

# Références relatives à la création de scripts ACL

Version : 14.1

Publié mercredi 19 juin 2019



# Table des matières

<b>Table des matières</b> .....	<b>3</b>
<b>Mise en route</b> .....	<b>13</b>
Principes de base pour la création de scripts Analytics .....	14
Commentaires .....	19
Types de données .....	21
Expressions .....	22
Définition de champs calculés avec des expressions .....	24
Fonctions .....	26
Variables .....	28
Structures des contrôles .....	31
Regroupement et exécution de boucles .....	34
Top 30 des fonctions Analytics .....	42
<b>Commandes</b> .....	<b>53</b>
Commande ACCEPT .....	54
Commande ACCESSDATA .....	59
Commande ACTIVATE .....	66
Commande AGE .....	68
Commande APPEND .....	73
Commande ASSIGN .....	82
Commande BENFORD .....	85
Commande CALCULATE .....	89
Commande CLASSIFY .....	91
Commande CLOSE .....	96
Commande CLUSTER .....	98
Commande COMMENT .....	101
Commande COUNT .....	103
Commande CREATE LAYOUT .....	106
Commande CROSSTAB .....	108
Commande CVSEVALUATE .....	112
Commande CVSPREPARE .....	116

Commande CVSSAMPLE .....	120
Commande DEFINE COLUMN .....	123
Commande DEFINE FIELD .....	126
Commande DEFINE FIELD ... COMPUTED .....	133
Commande DEFINE RELATION .....	139
Commande DEFINE REPORT .....	142
Commande DEFINE TABLE DB .....	143
Commande DEFINE VIEW .....	147
Commande DELETE .....	149
Commande DIALOG .....	153
Commande DIRECTORY .....	161
Commande DISPLAY .....	166
Commande DO REPORT .....	171
Commande DO SCRIPT .....	172
Commande DUMP .....	174
Commande DUPLICATES .....	176
Commande ESCAPE .....	181
Commande EVALUATE .....	183
Commande EXECUTE .....	188
Commande EXPORT .....	196
Commande EXTRACT .....	206
Commande FIELDSHIFT .....	211
Commande FIND .....	214
Commande FUZZYDUP .....	216
Commande FUZZYJOIN .....	222
Commande GAPS .....	228
Commande GROUP .....	232
Commande HELP .....	239
Commande HISTOGRAM .....	240
Commande IF .....	245
Commande IMPORT ACCESS .....	247
Commande IMPORT DELIMITED .....	250
Commande IMPORT EXCEL .....	258

Commande IMPORT GRCPROJECT .....	264
Commande IMPORT GRCRESULTS .....	270
Commande IMPORT LAYOUT .....	278
Commande IMPORT MULTIDELIMITED .....	280
Commande IMPORT MULTIEXCEL .....	288
Commande IMPORT ODBC .....	295
Commande IMPORT PDF .....	299
Commande IMPORT PRINT .....	309
Commande IMPORT SAP .....	318
Commande IMPORT XBRL .....	325
Commande IMPORT XML .....	330
Commande INDEX .....	335
Commande JOIN .....	338
Commande LIST .....	345
Commande LOCATE .....	348
Commande LOOP .....	351
Commande MERGE .....	353
Commande NOTES .....	357
Commande NOTIFY .....	359
Commande OPEN .....	362
Commande OUTLIERS .....	365
Commande PASSWORD .....	373
Commande PAUSE .....	375
Commande PREDICT .....	377
Commande PRINT .....	380
Commande PROFILE .....	382
Commande QUIT .....	384
Commande RANDOM .....	385
Commande RCOMMAND .....	388
Commande REFRESH .....	396
Commande RENAME .....	400
Commande REPORT .....	402
Commande RETRIEVE .....	406

Commande SAMPLE .....	408
Commande SAVE .....	417
Commande SAVE LAYOUT .....	419
Commande SAVE LOG .....	424
Commande SAVE TABLELIST .....	426
Commande SAVE WORKSPACE .....	428
Commande SEEK .....	430
Commande SEQUENCE .....	432
Commande SET .....	435
Commande SIZE .....	446
Commande SORT .....	451
Commande STATISTICS .....	456
Commande STRATIFY .....	460
Commande SUMMARIZE .....	466
Commande TOP .....	474
Commande TOTAL .....	475
Commande TRAIN .....	477
Commande VERIFY .....	483
<b>Fonctions .....</b>	<b>486</b>
Fonction ABS ( ) .....	487
Fonction AGE( ) .....	488
Fonction ALLTRIM( ) .....	493
Fonction ASCII( ) .....	495
Fonction AT( ) .....	497
Fonction BETWEEN( ) .....	500
Fonction BINTOSTR( ) .....	508
Fonction BIT( ) .....	510
Fonction BLANKS( ) .....	512
Fonction BYTE( ) .....	514
Fonction CDOW( ) .....	516
Fonction CHR( ) .....	519
Fonction CLEAN( ) .....	521
Fonction CMOY( ) .....	523

Fonction COS( ) .....	526
Fonction CTOD( ) .....	528
Fonction CTODT( ) .....	535
Fonction CTOT( ) .....	541
Fonction CUMIPMT( ) .....	546
Fonction CUMPRINC( ) .....	548
Fonction DATE( ) .....	550
Fonction DATETIME( ) .....	554
Fonction DAY( ) .....	559
Fonction DBYTE( ) .....	562
Fonction DEC( ) .....	564
Fonction DHEX( ) .....	567
Fonction DICECOEFFICIENT( ) .....	569
Fonction DIGIT( ) .....	575
Fonction DOW( ) .....	577
Fonction DTOU( ) .....	580
Fonction EBCDIC( ) .....	583
Fonction EFFECTIVE( ) .....	585
Fonction EOMONTH( ) .....	587
Fonction EXCLUDE( ) .....	590
Fonction EXP( ) .....	592
Fonction FILESIZE( ) .....	594
Fonction FIND( ) .....	596
Fonction FINDMULTI( ) .....	600
Fonction FREQUENCY( ) .....	605
Fonction FTYPE( ) .....	607
Fonction FVANNUITY( ) .....	610
Fonction FVLUMPSUM( ) .....	613
Fonction FVSCHEDULE( ) .....	616
Fonction GETOPTIONS( ) .....	618
Fonction GOMONTH( ) .....	620
Fonction HASH( ) .....	623
Fonction HEX( ) .....	628

Fonction HOUR()	630
Fonction HTOU()	633
Fonction INCLUDE()	635
Fonction INSERT()	637
Fonction INT()	639
Fonction IPMT()	640
Fonction ISBLANK()	642
Fonction ISDEFINED()	644
Fonction ISFUZZYDUP()	646
Fonction LAST()	651
Fonction LEADING()	653
Fonction LENGTH()	655
Fonction LEVDIST()	657
Fonction LOG()	661
Fonction LOWER()	663
Fonction LTRIM()	665
Fonction MAP()	667
Fonction MASK()	672
Fonction MATCH()	674
Fonction MAXIMUM()	682
Fonction MINIMUM()	685
Fonction MINUTE()	688
Fonction MOD()	691
Fonction MONTH()	693
Fonction NOMINAL()	696
Fonction NORMDIST()	698
Fonction NORMSINV()	700
Fonction NOW()	701
Fonction NPER()	702
Fonction OCCURS()	705
Fonction OFFSET()	707
Fonction OMIT()	709
Fonction PACKED()	713



Fonction PI( ) .....	715
Fonction PMT( ) .....	717
Fonction PPMT( ) .....	720
Fonction PROPER( ) .....	722
Fonction PROPERTIES( ) .....	724
Fonction PVANNUITY( ) .....	728
Fonction PVLUMPSUM( ) .....	731
Fonction PYDATE( ) .....	734
Fonction PYDATETIME( ) .....	736
Fonction PYLOGICAL( ) .....	738
Fonction PYNUMERIC( ) .....	740
Fonction PYSTRING( ) .....	742
Fonction PYTIME( ) .....	744
Fonction RAND( ) .....	746
Fonction RATE( ) .....	748
Fonction RDATE( ) .....	751
Fonction RDATETIME( ) .....	754
Fonction RECLLEN( ) .....	757
Fonction RECNO( ) .....	758
Fonction RECOFFSET( ) .....	760
Fonction REGEXFIND( ) .....	762
Fonction REGEXREPLACE( ) .....	770
Fonction REMOVE( ) .....	778
Fonction REPEAT( ) .....	781
Fonction REPLACE( ) .....	783
Fonction REVERSE( ) .....	787
Fonction RJUSTIFY( ) .....	788
Fonction RLOGICAL( ) .....	789
Fonction RNUMERIC( ) .....	792
Fonction ROOT( ) .....	795
Fonction ROUND( ) .....	797
Fonction RSTRING( ) .....	799
Fonction RTIME( ) .....	802

Fonction SECOND()	805
Fonction SHIFT()	808
Fonction SIN()	810
Fonction SOUNDEX()	812
Fonction SOUNDSLIKE()	816
Fonction SPLIT()	819
Fonction STOD()	823
Fonction STODT()	827
Fonction STOT()	832
Fonction STRING()	836
Fonction SUBSTR()	839
Fonction TAN()	842
Fonction TEST()	844
Fonction TIME()	846
Fonction TODAY()	851
Fonction TRANSFORM()	852
Fonction TRIM()	854
Fonction UNSIGNED()	856
Fonction UPPER()	858
Fonction UTOD()	860
Fonction VALUE()	864
Fonction VERIFY()	867
Fonction WORKDAY()	869
Fonction YEAR()	873
Fonction ZONED()	876
Fonction ZSTAT()	880
<b>Outils d'analyse</b>	<b>883</b>
Scripts d'outil d'analyse	884
En-têtes des outils d'analyse et balises	887
ANALYTIC	891
FILE	893
TABLE	895
FIELD	897

PARAM .....	899
PASSWORD .....	912
DATA .....	915
RESULT .....	919
PUBLISH .....	924
Développement d'outils d'analyse .....	925
Ajouter des en-têtes d'outils d'analyse .....	929
Bonnes pratiques en termes de développement d'outils d'analyse .....	933
Compresser des Apps Analyse .....	939
Exemples de scripts d'outils d'analyse (App Analyse) .....	942
<b>Annexe .....</b>	<b>947</b>
Configuration minimale requise .....	948
Installer ACL pour Windows .....	951
Configuration de Python pour être utilisé avec Analytics .....	953
Éditions Unicode et non Unicode .....	956
Convertir des outils d'analyse au format Unicode .....	957
Vérifier la compatibilité Unicode .....	960
Exécution de scripts R sur AX Serveur .....	963
Exécution de scripts Python sur AX Serveur .....	968
Codes d'erreur du moteur d'outil d'analyse .....	973
Variables créées via des commandes Analytics .....	980
Mots-clés réservés .....	985



# Mise en route

# Principes de base pour la création de scripts Analytics

ACLScript est un langage de commande qui vous permet de programmer et d'automatiser les commandes Analytics. La structure et les composants d'ACLScript sont simples, mais puissants.

## Remarque

Si vous êtes un vrai débutant en matière de création de scripts, pensez à visiter l'Académie pour suivre une formation de base avant de vous plonger dans le présent contenu. Pour les cours sur la création de scripts et l'utilisation d'Analytics, rendez-vous sur [www.highbond.com](http://www.highbond.com).

## Commandes

Chaque ligne d'un script exécute une commande ACLScript et commence par le nom de la commande. Une commande est une instruction permettant d'exécuter une opération dans Analytics.

Le nom de la commande est suivi d'un ou plusieurs paramètres spécifiés sous *nom\_paramètre valeur\_paramètre*.

## Astuce

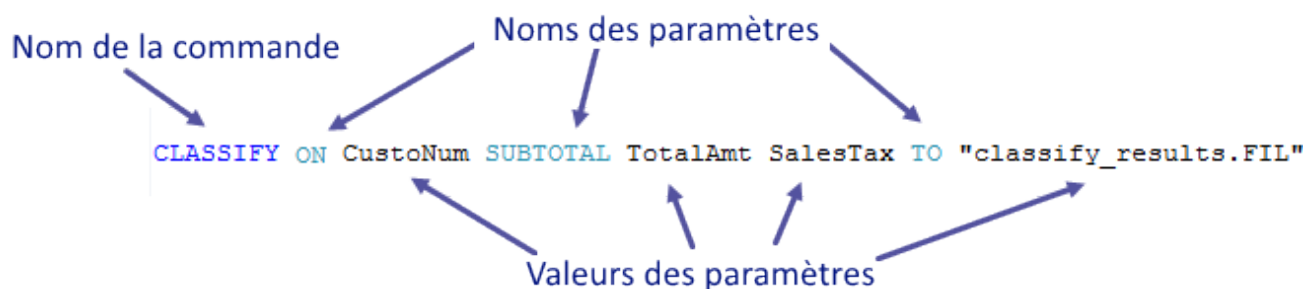
Selon la commande, certains paramètres sont obligatoires et d'autres sont facultatifs. Vous n'êtes pas obligé de spécifier des paramètres facultatifs. S'ils sont omis, la commande s'exécute sans eux. Toutefois, si vous omettez un paramètre obligatoire, Analytics utilise la valeur par défaut associée à ce paramètre.

## Exemple à l'aide de la commande CLASSIFY

L'exemple suivant affiche la commande CLASSIFY avec les paramètres suivants :

- ON : indique en fonction de quel champ de la table effectuer le classement
- SUBTOTAL : indique les champs facultatifs à sous-totaliser dans la sortie
- TO : indique la table sur laquelle écrire les résultats de la commande CLASSIFY

Remarquez comment chaque paramètre est suivi d'une ou plusieurs valeurs de paramètres :



## Remarques importantes concernant la syntaxe des commandes

- le premier mot d'une ligne de script doit être un nom de commande
- pour la plupart des commandes, l'ordre des paramètres suivant un nom de la commande n'a pas d'importance
- la plupart des commandes demandent à ce que vous ouvriez la table cible avant d'exécuter la commande, précédez ces commandes de `OPEN nom_table`

## Commentaires

Comme dans tout langage de script, il est possible d'ajouter des commentaires dans ACLScript avec le mot-clé `COMMENT`. Les commentaires facilitent la compréhension de votre code et sa communication auprès d'utilisateurs cherchant à lire, à utiliser ou à comprendre votre script. ACLScript prend en charge deux types de commentaires :

- **commentaires d'une seule ligne** : l'intégralité du texte qui suit `COMMENT` est ignorée jusqu'à ce que la fin de la ligne soit atteinte
- **blocs de commentaires sur plusieurs lignes** : commencent par `COMMENT` et les lignes qui suivent sont ignorées tant que le mot-clé `END` ou une ligne vide n'est pas atteint

Pour obtenir plus d'informations et des exemples, consultez la rubrique "Commentaires" Page 19.

## Types de données

ACLScript prend en charge quatre types de données de base :

- **logique** : le type de données le plus simple. Les données logiques expriment une valeur de vérité : vrai ou faux (« true or false »).
- **numériques** : contiennent des chiffres allant de 0 à 9 et, éventuellement, un signe négatif et une virgule décimale.
- **caractère** : une série d'un ou plusieurs caractères.
- **DateHeure** : une valeur date, DateHeure ou heure exprimée dans un format compatible.

Chaque type de données est traité différemment par Analytics et peut être utilisé dans différentes commandes et fonctions. Pour plus d'informations sur les types de données, consultez la rubrique "Types de données" Page 21.

## Expressions

Une expression est une instruction ayant une valeur. La forme d'expression la plus simple est une valeur littérale comme 2 ou "test". Toutefois, les expressions s'affichent généralement sous forme de calculs et peuvent être aussi complexes que toutes les combinaisons valides d'opérateurs, conditions, fonctions et valeurs que vous pouvez imaginer :

```
((2 + (3 - 2)) * 2) > ROOT(9;0)
```

Les expressions sont généralement utilisées dans Analytics pour remplir des champs calculés ou sous forme de valeur d'introduction pour la logique conditionnelle. Pour plus d'informations sur les expressions, consultez la section "Expressions" Page 22.

## Fonctions

Les fonctions sont des routines intégrées qui acceptent un certain nombre de paramètres et qui renvoient une seule valeur. Elles servent à manipuler le contenu des champs et les variables utilisés dans les commandes.

### Remarque

Les fonctions ne modifient pas les données dans les champs ; elles génèrent et renvoient une nouvelle valeur à partir d'un calcul ou d'un algorithme qui utilise les données des champs ou des variables comme entrées. Utilisez la valeur renvoyée par la fonction comme entrée pour une commande.

Les fonctions commencent par un nom de fonction suivi directement d'une parenthèse ouvrante, d'une liste séparée par des virgules d'au moins 0 valeurs transmises à la fonction sous forme d'arguments, ainsi que d'une parenthèse fermante.

## Exemple

La fonction `BETWEEN(valeur; min; max)` prend trois arguments et renvoie « vrai » si la valeur tombe dans la plage ou « faux » si elle ne tombe pas dans la plage :

- `valeur` : l'expression ou le champ à tester
- `min` : le minimum de la plage
- `max` : le maximum de la plage

```
BETWEEN(montant; 500; 5000)
```

Pour plus d'informations sur les fonctions, consultez la rubrique "Fonctions" Page 26.

## Variables

Une variable est un emplacement de stockage temporaire utilisé pour conserver une valeur. Un identificateur lui est associé, ce qui vous permet de faire référence à la valeur stockée en mémoire dans votre ordinateur et de l'utiliser.

ACLScript utilise la commande `ASSIGN` pour créer une variable et pour lui attribuer une valeur en même temps :



```
ASSIGN v_age_in_years = 3
```

Pour simplifier, vous pouvez omettre le mot-clé `ASSIGN` ; toutefois, `ASSIGN` est utilisé implicitement et la même commande s'exécute :

```
v_age_in_years = 3
```

### Remarque

ACLScript ne prend pas en charge les valeurs null. Toutes les variables doivent être associées à une valeur d'un des types de données pris en charge. L'interpréteur de script évalue le type de données à l'aide du format de date et du qualificateur que vous utilisez pour attribuer la valeur. Pour plus d'informations, consultez la section "Types de données" Page 21.

## Utilisation de variables

Une fois qu'une variable est créée, vous pouvez y faire référence à n'importe quel endroit où vous renvoyez à des noms de champs ou à des variables. Vous pouvez aussi lui réattribuer une nouvelle valeur à l'aide de la commande `ASSIGN`.

```
EXTRACT RECORD TO 'result.fil' IF age > v_age_in_years
v_age_in_years = 5
```

Vous pouvez aussi utiliser l'interpolation des chaînes ou la substitution de variable pour inclure une variable dans un littéral de chaîne en entourant le nom de la variable avec des caractères `%`. Lorsqu'Analytics trouve la variable de substitution, il remplace l'espace réservé par la valeur correspondante :

```
ASSIGN v_table = erp_data
OPEN %v_table%
```

Pour obtenir plus d'informations sur les variables, consultez la rubrique "Variables" Page 28.

## Structures des contrôles

Une structure de contrôle est un composant de script décidant de la direction à prendre compte tenu de certains paramètres. ACLScript fournit à la fois des structures de bouclage et des structures logiques conditionnelles.

### Logique conditionnelle

ACLScript met en œuvre une logique conditionnelle sous la forme d'une commande `IF` et sous la forme d'un paramètre facultatif dans plusieurs commandes du langage.

### Astuce

Utilisez la commande **IF** pour contrôler si une commande s'exécute ou non, mais utilisez le paramètre **IF** pour décider des enregistrements de la table par rapport auxquels exécuter la commande.

## Commande IF

```
IF v_compteur > 10 CLASSIFY ON num_client
```

## Paramètre IF

```
CLASSIFY ON num_client IF état = 'NY'
```

## Boucles

La commande **LOOP** fournit la structure de contrôle en boucle dans ACLScript. Cette commande traite les instructions contenues à l'intérieur de la boucle tant que l'expression du test de contrôle est évaluée sur vraie :

Pour obtenir plus d'informations sur les structures des contrôles, consultez la rubrique "Structures des contrôles" Page 31

# Commentaires

Comme dans tout langage de script, il est possible d'ajouter des commentaires dans ACLScript avec le mot-clé COMMENT. Les commentaires facilitent la compréhension de votre code et sa communication auprès d'utilisateurs cherchant à lire, à utiliser ou à comprendre votre script.

## Types de commentaires

ACLScript prend en charge deux types de commentaires :

- **commentaires d'une seule ligne** : l'intégralité du texte qui suit COMMENT est ignorée jusqu'à ce que la fin de la ligne soit atteinte
- **blocs de commentaires sur plusieurs lignes** : commencent par COMMENT et les lignes qui suivent sont ignorées tant que le mot-clé END ou une ligne vide n'est pas atteint

## Commentaires d'une seule ligne

Les commentaires d'une seule ligne permettent de décrire des étapes individuelles de votre script ou de décrire des variables :

```
COMMENT *** la date de début de la période d'analyse
v_Date_début = `20150101`
```

## Blocs de commentaires sur plusieurs lignes

Les blocs de commentaires sur plusieurs lignes servent à décrire des scripts ou des sections de script.

```
COMMENT
*****

** Cette section du script prépare les données pour l'importation
*****

END
```

## Blocs de commentaires d'en-tête

Il est recommandé d'inclure un bloc de commentaires d'en-tête qui contient des informations clés sur le script au début de chaque script :

```
COMMENT
*****
```

```
*** Nom du script : {ID_App}{Nom du script}
*** Paramètres : {Description détaillée}
*** Sortie : {Décrire les paramètres}
*** Rédigé par : {Nom}, Entreprise ABC, {Mois AAAA}
*** Modifié par : {Nom}, Entreprise ABC, objectif et logique du script
*** Version : 1.1.1 {app_ver.script_ver.defect.fix}
*****
END
```

# Types de données

ACLScript prend en charge quatre types de données de base : logique, numérique, caractère et DateHeure.

Type	Description	Limite	Qualificateur	Exemples
Caractère	Une série d'un ou plusieurs caractères.	32 767 octets	Guillemets simples ou guillemets doubles	<ul style="list-style-type: none"> <li>◦ 'John Doe'</li> <li>◦ "John Doe"</li> </ul>
Numérique	Les valeurs numériques contiennent des chiffres allant de 0 à 9 et, éventuellement, un signe négatif et une virgule décimale.	22 chiffres	Sans qualificateur	<ul style="list-style-type: none"> <li>◦ 100</li> <li>◦ -5</li> <li>◦ 5,01</li> <li>◦ 22222,1232</li> </ul>
DateHeure	Une valeur date, DateHeure ou heure exprimée dans un format pris en charge.	<ul style="list-style-type: none"> <li>◦ Minimum = 1900-01-01</li> <li>◦ Maximum = 9999-12-31</li> </ul>	<ul style="list-style-type: none"> <li>◦ Guillemets inversés</li> <li>◦ Au début : la lettre 't' ou une seule espace pour les valeurs heure</li> </ul>	<ul style="list-style-type: none"> <li>◦ `20160101`</li> <li>◦ `141231`</li> <li>◦ `t2359`</li> <li>◦ `20141231T2359-9`</li> <li>◦ `20141231 235959`</li> </ul>
Logique	<p>Le type de données le plus simple. Les données logiques expriment une valeur de vérité : vrai ou faux (« true or false »).</p> <p>Les opérateurs de comparaison comme =, &gt; et &lt; renvoient des valeurs logiques.</p>	<ul style="list-style-type: none"> <li>◦ T</li> <li>◦ F</li> </ul>	Sans qualificateur	ASSIGN v_vérité = 5 > 4 est évalué sur T

# Expressions

Une expression est une instruction ayant une valeur. La forme d'expression la plus simple est la valeur littérale. Toutefois, les expressions peuvent être aussi complexes que toutes les combinaisons légitimes d'opérateurs, conditions, fonctions et valeurs que vous pouvez imaginer.

## Composants des expressions

### Valeurs littérales

Une valeur littérale est une valeur écrite exactement comme elle est censée être interprétée, tel un caractère littéral 'ma valeur'. Pour obtenir des informations sur les valeurs littérales, consultez la rubrique "Types de données" Page précédente.

### Opérateurs

Les opérateurs sont des symboles qui indiquent à l'interpréteur de scripts d'effectuer une évaluation arithmétique, une évaluation de chaîne de caractères, une évaluation de comparaison ou une évaluation logique des valeurs spécifiées :

Type d'opérateur dans l'ordre de priorité	Opérateurs dans l'ordre de priorité	Exemples
Parenthèse	<ul style="list-style-type: none"> <li>○ () indique la priorité</li> <li>○ () opérateur de fonction</li> </ul>	$(5 + 3) * 2$
Unaire	<ul style="list-style-type: none"> <li>○ NOT logique</li> <li>○ - négation</li> </ul>	$v\_vérité = NOT (3 < 2)$
Arithmétique	<ul style="list-style-type: none"> <li>○ ^ exponentiation</li> <li>○ * multiplie, / divise</li> <li>○ + ajoute, - soustrait</li> </ul> <p><b>Remarque</b></p> <p>Les opérateurs de multiplication présentent des priorités égales les uns avec les autres ; ils sont évalués de gauche à droite.</p> <p>Les opérateurs d'addition présentent des priorités égales les uns avec les autres ; ils sont évalués de gauche à droite.</p>	$1 + 5 - 3 * 2$

Type d'opérateur dans l'ordre de priorité	Opérateurs dans l'ordre de priorité	Exemples
Chaîne de caractères	+ concatène	"C'est" + " mon script"
Comparatif	<ul style="list-style-type: none"> <li>○ &lt; inférieur à</li> <li>○ &gt; supérieur à</li> <li>○ = égalité</li> <li>○ &gt;= supérieur ou égal à</li> <li>○ &lt;= inférieur ou égal à</li> <li>○ &lt;&gt; différent de</li> </ul> <p><b>Remarque</b></p> <p>Les opérateurs de comparaison présentent des priorités égales les uns avec les autres ; ils sont évalués de gauche à droite.</p>	IF montant <> 100
Logique binaire	<ul style="list-style-type: none"> <li>○ AND ou &amp;</li> <li>○ OR ou  </li> </ul>	IF montant > 5 AND montant < 10

## Fonctions

Les expressions sont évaluées à l'aide des valeurs renvoyées par les fonctions. Les fonctions s'exécutent avec la plus haute priorité d'un composant d'une expression. Pour plus d'informations sur les fonctions, consultez la rubrique "Fonctions" Page 26.

## Exemples d'expressions

### Évaluée sur 6

$(2 + (3 - 2)) * 2$

### Évaluée sur « vrai » (true)

$((2 + (3 - 2)) * 2) > \text{ROOT}(9;0)$

### Évaluée sur 'ACLScript tutorial'

'AC' + 'LScri' + 'pt' + 'tutorial'

# Définition de champs calculés avec des expressions

Les **champs calculés** permettent de créer des champs de données supplémentaires dans la table active à l'aide d'une expression. Un champ calculé est un champ annexé à la table ouverte et qui est rempli avec la valeur de l'expression spécifiée.

## Syntaxe des champs calculés

```
DEFINE FIELD nom COMPUTED expression
```

- **nom** : nom du champ calculé à générer
- **expression** : calcul ayant servi à générer la valeur du champ

## Exemple de champ calculé

```
DEFINE FIELD c_nom_complet COMPUTED prénom + ' ' + nom
```

### Astuce

Ajoutez le préfixe **c\_** aux noms des champs calculés pour les identifier comme données calculées plutôt que comme données source d'origine.

## Définition des valeurs des champs calculés conditionnels

Vous pouvez aussi utiliser des conditions avec des champs calculés pour définir la valeur de différents cas :

```
DEFINE FIELD c_total COMPUTED  
  
montant * ca_taux_taxe IF state = 'CA'  
montant * ny_taux_taxe IF state = 'NY' OR state = 'NJ'  
montant * taux_général
```

Lorsque la première expression conditionnelle est évaluée sur vrai (« true »), la valeur spécifiée pour ce cas est utilisée. Dans cet exemple, `montant * taux_général` est la valeur par défaut utilisée lorsqu'aucune des expressions conditionnelles n'est évaluée sur vrai (« true »).



**Remarque**

Vous devez ajouter une ligne vide entre la commande de ligne et les conditions sauf si vous incluez les paramètres IF, WIDTH, PIC ou AS dans la commande DEFINE FIELD. Pour plus d'informations, consultez la section "Commande DEFINE FIELD . . . COMPUTED" Page 133.

# Fonctions

Les fonctions sont des routines intégrées qui acceptent un certain nombre de paramètres et qui renvoient une seule valeur. Elles servent à manipuler le contenu des champs et les variables utilisés dans les commandes.

## Remarque

Les fonctions ne modifient pas les données dans les champs ; elles génèrent et renvoient une nouvelle valeur à partir d'un calcul ou d'un algorithme qui utilise les données des champs ou des variables comme entrées. Utilisez la valeur renvoyée par la fonction comme entrée pour une commande.

## Syntaxe des fonctions

Les fonctions commencent par un nom de fonction suivi directement d'une parenthèse ouvrante, d'une liste séparée par des virgules d'au moins 0 valeurs transmises à la fonction sous forme d'arguments, ainsi que d'une parenthèse fermante.

## Exemple

La fonction `BETWEEN(valeur; min; max)` prend trois arguments et renvoie « vrai » si la valeur tombe dans la plage ou « faux » si elle ne tombe pas dans la plage :

- `valeur` : l'expression ou le champ à tester
- `min` : le minimum de la plage
- `max` : le maximum de la plage

```
BETWEEN(montant; 500; 5000)
```

## Arguments des fonctions

Un argument de fonction est une valeur d'introduction spécifique transmise à la fonction.

Les arguments des fonctions sont transmis aux fonctions via une liste d'arguments. Il s'agit d'une liste délimitée par des virgules contenant des valeurs littérales, des variables ou des expressions évaluées sur les valeurs du type de données des paramètres. Pour obtenir plus d'informations sur l'utilisation des types de données, consultez la rubrique "Types de données" Page 21.

**Remarque**

Si votre projet utilise des formats numériques européens ou que vous écrivez des scripts portables dans toutes les régions, séparez les arguments de fonction par un caractère espace plutôt que par une virgule sauf si vous transmettez une valeur numérique signée. Les fonctions acceptant des valeurs numériques signées nécessitent un délimiteur explicite.

## Fonctions et commandes

La différence entre une commande et une fonction est subtile, mais essentielle pour utiliser ACLScript :

Fonctions	Commandes
Utilisent des champs, des valeurs ou des enregistrements comme valeurs d'entrée et génèrent une nouvelle valeur renvoyée.	Utilisent les tables comme valeur d'entrée et génèrent de nouveaux enregistrements et de nouvelles tables.
Utilisées dans les expressions, champs calculés, valeurs de paramètres de commandes, variables et filtres pour aider à l'exécution des commandes et la modifier.	Utilisées pour analyser des données, importer des données et produire des résultats.
Ne peut pas constituer une étape indépendante dans un script.	Peut être une étape indépendante dans un script.

# Variables

Une variable est un emplacement de stockage temporaire utilisé pour conserver une valeur. Un identificateur lui est associé, ce qui vous permet de faire référence à la valeur stockée en mémoire dans votre ordinateur et de l'utiliser.

## Fonctionnement des variables dans ACLScript

### Création d'une variable et affectation d'une valeur

ACLScript utilise la commande `ASSIGN` pour créer une variable et pour lui attribuer une valeur en même temps :

```
ASSIGN v_age_in_years = 3
```

Pour simplifier, vous pouvez omettre le mot-clé `ASSIGN` ; toutefois, `ASSIGN` est utilisé implicitement et la même commande s'exécute :

```
v_age_in_years = 3
```

#### Remarque

ACLScript ne prend pas en charge les valeurs null. Toutes les variables doivent être associées à une valeur d'un des types de données pris en charge. L'interpréteur de script évalue le type de données à l'aide du format de date et du qualificateur que vous utilisez pour attribuer la valeur. Pour plus d'informations, consultez la section "Types de données" Page 21.

## Utilisation de variables

Une fois qu'une variable est créée, vous pouvez y faire référence à n'importe quel endroit où vous renvoyez à des noms de champs ou à des variables. Vous pouvez aussi lui réattribuer une nouvelle valeur à l'aide de la commande `ASSIGN`.

```
EXTRACT RECORD TO 'result.fil' IF age > v_age_in_years  
v_age_in_years = 5
```

Vous pouvez aussi utiliser l'interpolation des chaînes ou la substitution de variable pour inclure une variable dans un littéral de chaîne en entourant le nom de la variable avec des caractères `%`.

Lorsqu'Analytics trouve la variable de substitution, il remplace l'espace réservé par la valeur correspondante :

```
ASSIGN v_table = erp_data
OPEN %v_table%
```

## Types de variables

Analytics utilise les types de variables suivants :

- **les variables générées par le système** : créées automatiquement après l'exécution d'une commande
- **les variables permanentes** : restent dans la mémoire de votre ordinateur jusqu'à ce que vous les supprimiez et continuent d'exister après la fermeture du projet Analytics

### Remarque

Pour définir une variable permanente, ajoutez le préfixe '\_' devant l'identificateur : `_v_nom_société = 'Acme'`.

- **les variables de session** : restent en mémoire dans votre ordinateur jusqu'à ce que vous les supprimiez ou jusqu'à ce que le projet Analytics soit fermé

## Identificateurs de variables

Les identificateurs de variables ne respectent pas la casse et suivent certaines conventions associées au type de variable :

- les identificateurs des variables générées par le système utilisent des noms tout en majuscules : `OUTPUTFOLDER`
- les identificateurs des variables permanentes présentent le préfixe « \_ » : `_v_permanent`
- les identificateurs de variables de session utilisent le format `v_nomvar` par convention, mais vous n'êtes pas tenu de respecter cette convention de nom

## Affichage des valeurs des variables

Pendant le développement du script ou pendant le débogage, il peut s'avérer utile de suivre les valeurs des variables à mesure que le script s'exécute. Pour capturer les valeurs des variables dans le fichier trace du script, utilisez la commande `DISPLAY` :

```
DISPLAY v_age_en_années
```

À chaque fois que le script tombe sur cette commande, il l'écrit dans le fichier trace. Pour afficher la valeur de la variable à ce stade de l'exécution du script, cliquez sur son entrée dans la trace.

### Astuce

Vous pouvez aussi utiliser des variables pour aider au débogage en insérant des points d'arrêt dans votre script et en contrôlant les valeurs des variables dans l'onglet **Variables** du **Navigateur**.

# Structures des contrôles

Une structure de contrôle est un composant de script décidant de la direction à prendre compte tenu de certains paramètres. ACLScript fournit à la fois des structures de bouclage et des structures logiques conditionnelles IF.

## Logique conditionnelle à l'aide de IF

ACLScript met en œuvre une logique conditionnelle sous la forme d'une commande IF et sous la forme d'un paramètre facultatif dans plusieurs commandes du langage :

- **commande** : contrôle l'exécution ou l'inexécution d'une commande
- **paramètre** : décide des enregistrements d'une table par rapport auxquels exécuter la commande

## La commande IF

Lorsque vous utilisez la commande IF, vous spécifiez une expression conditionnelle suivie de la commande à exécuter si l'expression est évaluée sur « true » (vrai) :

```
IF v_compteur > 10 CLASSIFY ON num_client
```

Cette structure conditionnelle contrôle quel code s'exécute ; vous pouvez donc utiliser la commande IF lorsque vous souhaitez traiter une table entière basée sur l'expression de test. Si l'expression est évaluée sur « true », la commande est exécutée sur tous les enregistrements de la table. Pour obtenir plus d'informations sur la commande IF, consultez la rubrique "Commande IF" Page 245.

## Paramètre IF

Un grand nombre de commandes accepte un paramètre IF facultatif que vous pouvez utiliser pour filtrer les enregistrements dans lesquels la commande s'exécute :

```
CLASSIFY ON num_client IF état = 'NY'
```

Lorsque cette instruction s'exécute, le script classe tous les enregistrements dans la table pour lesquels la valeur du champ état est 'NY'.

## Boucles

### La commande LOOP

La commande LOOP fournit la structure de contrôle en boucle dans ACLScript.

### Remarque

La commande `LOOP` doit s'exécuter à l'intérieur de la commande `GROUP`, elle ne peut pas s'exécuter de façon autonome.

Cette commande traite les instructions contenues à l'intérieur de la boucle tant que l'expression `WHILE` spécifiée est vraie :

```
ASSIGN v_compteur = 10
GROUP
  LOOP WHILE v_compteur > 0
    v_total = v_total + montant
    v_compteur = v_compteur - 1
  END
END
```

Cette structure effectue 10 itérations et ajoute la valeur du champ `montant` à la variable `v_total`. À la fin de chaque itération, la variable `v_counter` est décrétementée de 1, puis testée dans l'expression `WHILE`. Une fois que l'expression est évaluée sur faux, la boucle se termine et le script se poursuit.

Une fois la boucle terminée, `v_total` conserve la somme des champs `montant` des 10 enregistrements.

Pour obtenir plus d'informations sur les boucles, consultez la rubrique "Commande `LOOP`" Page 351.

## LOOPING avec un sous-script

Il arrive que la commande `LOOP` ne fournisse pas exactement la fonctionnalité d'exécution en boucle dont vous pouvez avoir besoin. Dans ce cas, vous pouvez aussi appeler un script Analytics distinct pour exécuter une boucle à l'aide de la commande `DO SCRIPT` : `DO SCRIPT NomScript WHILE TestConditionnel`.

Vous pouvez utiliser l'une de ces méthodes courantes pour choisir quand terminer votre boucle :

- **drapeau** : la boucle se poursuit jusqu'à ce que la variable logique `drapeau` soit définie sur `FALSE`
- **compteur** : la boucle se poursuit jusqu'à ce qu'une variable incrémentant ou décrétementante croise un seuil conditionnel

Pour obtenir plus d'informations sur l'appel de sous-scripts, consultez la rubrique "Commande `DO SCRIPT`" Page 172.

### Exemple

Vous devez importer tous les fichiers CSV dans le dossier `C:\data` de votre projet. Vous pouvez utiliser la commande `DIRECTORY` pour obtenir une liste des fichiers du dossier. Toutefois, vous ne pouvez pas utiliser la commande `IMPORT` à l'intérieur de la structure `GROUP`. Vous devez trouver un autre moyen d'exécuter une boucle dans la table créée par `DIRECTORY`.

Pour ce faire, vous créez un script principal qui :



1. Exécute la commande DIRECTORY et enregistre les résultats dans une table.
2. Obtient le nombre d'enregistrements dans la table à utiliser comme compteur.
3. Appelle un sous-script une fois par enregistrement dans la table afin d'exécuter la commande IMPORT par rapport à l'enregistrement actuel.

## Script principal

```
COMMENT Script principal

DIRECTORY "C:\data\*.csv" TO T_Table_à_boucler
OPEN T_Table_à_boucler
COUNT
v_nb_enreg = COUNT1
v_compteur = 1
DO SCRIPT Importer_sous_script WHILE v_compteur <= v_nb_enreg
```

## Importer un sous-script

```
COMMENT Importer_sous_script

OPEN T_Table_à_boucler
LOCATE RECORD v_compteur

COMMENT code à importer fichier CSV dans l'enregistrement va ici...

ASSIGN v_compteur = v_compteur + 1
```

Les variables sont communes à tous les scripts s'exécutant dans le projet ; le script principal appelle donc le sous-script jusqu'à ce que la valeur de `v_compteur` dépasse la valeur de `v_nb_enreg`. À chaque exécution du sous-script, il incrémente `v_compteur`.

Cette structure vous permet d'appeler la commande `IMPORT` par rapport à chaque enregistrement tout en exécutant une boucle dans la table. À la fin du script principal, vous avez importé tous les fichiers CSV du dossier `C:\data`.

# Regroupement et exécution de boucles

Les commandes GROUP et LOOP fournissent deux façons d'exécuter des séries de commandes plusieurs fois. GROUP effectue une seule itération d'une ou plusieurs commandes sur chaque enregistrement. LOOP accomplit plusieurs itérations d'une série de commandes sur chaque enregistrement, et peut seulement être utilisée à l'intérieur d'un bloc GROUP.

## Exemple simple de GROUP

Voici une table de données de facture appelée **Trans\_Cf**. À l'aide de ces données, vous devez calculer un total cumulé de montants facturés :

Numéro_fournisseur	Nom_Fournisseur	Ville_fournisseur	Numéro_Facture	Date_Facture	Montant_Facture	Quantité	Coût_unitaire
11663	More Power Industries	Los Angeles	5981807	2000-11-17	618,30	90	6,87
13808	NOVATECH Wholesale	Des Moines	2275301	2000-11-17	6705,12	976	6,87
12433	Koro International	Sheveport	6585673	2000-11-17	7955,46	1158	6,87

Pour calculer ce montant, il vous faut utiliser la commande GROUP. À l'intérieur de chaque itération de GROUP, vous :

1. Calculez le total cumulé à partir de l'enregistrement actuel.
2. Importez le numéro de la facture, le montant, la date et le total cumulé dans une table de résultats.

```
OPEN Trans_Cf
```

```
COMMENT définit la valeur initiale du total cumulé sur zéro
ASSIGN v_running_total = 0,00
```

```
COMMENT itère sur chaque enregistrement de la table, puis calcule et extrait le total cumulé
GROUP
```

```
  ASSIGN v_running_total = v_running_total + Montant_Facture
  EXTRACT No_Facture, Montant_Facture, Date_Facture, v_running_total AS "Cumul" TO résultats1
END
```

Lorsque le script est exécuté, les commandes dans le bloc GROUP sont traitées sur chaque enregistrement de la table, du haut vers le bas, et le total cumulé est calculé et extrait. Si l'on pouvait décrire GROUP pendant son exécution, la procédure ressemblerait à ceci :

## Première itération de GROUP : total cumulé = 0,00 + 618,0

GROUP ajoute le montant de la facture du premier enregistrement au total cumulé initial de 0,00 et extrait les champs dans la table de résultats :

Numéro_fournisseur	Nom_Fournisseur	Ville_fournisseur	Numéro_Facture	Date_Facture	Montant_Facture	Quantité	Coût_unitaire
11663	More Power Industries	Los Angeles	5981807	2000-11-17	618,30	90	6,87
13808	NOVATECH Wholesale	Des Moines	2275301	2000-11-17	6705,12	976	6,87
12433	Koro International	Sheveport	6585673	2000-11-17	7955,46	1158	6,87

## Deuxième itération de GROUP : total cumulé = 618,30 + 6705,12

Le bloc GROUP ajoute le montant de la facture du deuxième enregistrement au nouveau total cumulé de 618,30 et extrait les champs dans la table de résultats :

Numéro_fournisseur	Nom_Fournisseur	Ville_fournisseur	Numéro_Facture	Date_Facture	Montant_Facture	Quantité	Coût_unitaire
11663	More Power Industries	Los Angeles	5981807	2000-11-17	618,30	90	6,87
13808	NOVATECH Wholesale	Des Moines	2275301	2000-11-17	6705,12	976	6,87
12433	Koro International	Sheveport	6585673	2000-11-17	7955,46	1158	6,87

## Troisième itération de GROUP : total cumulé = 7323,42 + 7955,46

Le bloc GROUP ajoute le montant de la facture du troisième enregistrement au nouveau total cumulé de 7323,42 et extrait les champs dans la table de résultats :

Numéro_fournisseur	Nom_Fournisseur	Ville_fournisseur	Numéro_Facture	Date_Facture	Montant_Facture	Quantité	Coût_unitaire
11663	More Power Industries	Los Angeles	5981807	2000-11-17	618,30	90	6,87
13808	NOVATECH Wholesale	Des Moines	2275301	2000-11-17	6705,12	976	6,87
12433	Koro International	Sheveport	6585673	2000-11-17	7955,46	1158	6,87

## Table de résultats finale

Une fois que GROUP a traité le dernier enregistrement de la table, vous obtenez la table de résultats suivante :

Numéro_Facture	Montant_Facture	Date_Facture	Total
5981807	618,30	2000-11-17	618,30
2275301	6705,12	2000-11-17	7323,42
6585673	7955,46	2000-11-17	15278,88

## Gérer différents cas à l'aide de GROUP IF

À l'aide de la même table **Trans\_Cf**, vous avez maintenant besoin de calculer les totaux cumulés pour trois types de facture :

- Valeur élevée (supérieure ou égale à 1000,00)
- Valeur moyenne (entre 100,00 et 1000,00)
- Valeur faible (moins de 100,00)

La commande GROUP fournit une structure IF/ELSE pour gérer différents cas. Vous fournissez des expressions conditionnelles à tester, et si un enregistrement est évalué sur vrai, la commande à l'intérieur du bloc s'exécute.

## Comment les cas sont testés

Les cas sont testés **de haut en bas** et un enregistrement ne peut être traité que par un bloc IF/ELSE. Le premier cas à être évalué sur vrai pour l'enregistrement est celui qui traite l'enregistrement :

1. Quand la commande GROUP traite le premier enregistrement, elle le teste avec la première condition IF ( $\text{Montant\_Facture} \geq 1000$ ). Si ce test est évalué sur vrai, le code pour ce cas est choisi et aucun autre cas n'est testé.

2. Si le premier cas est évalué sur faux, la condition ELSE IF suivante est testée (Montant\_Facture >= 100). De même, si ce test est évalué sur vrai, le code pour ce cas est choisi et aucun autre cas n'est testé.
3. Enfin, si aucun des cas IF ou ELSE IF n'est évalué sur vrai, le cas par défaut dans le bloc ELSE traite l'enregistrement.

### Remarque

Si un enregistrement est évalué sur vrai pour plusieurs cas, l'enregistrement est uniquement traité par le premier bloc IF/ELSE qui l'a testé. Les enregistrements ne sont jamais traités par plusieurs blocs IF/ELSE dans une commande GROUP.

OPEN Trans\_Cf

COMMENT définit les valeurs initiales des totaux cumulés

ASSIGN v\_running\_total\_hi = 0,00

ASSIGN v\_running\_total\_med = 0,00

ASSIGN v\_running\_total\_low = 0,00

COMMENT utilise GROUP IF pour exécuter différentes commandes ASSIGN et EXTRACT en fonction du montant de la facture

GROUP IF Montant\_Facture >= 1000

ASSIGN v\_running\_total\_hi = v\_running\_total\_hi + Montant\_Facture

EXTRACT No\_Facture, Montant\_Facture, Date\_facture, v\_running\_total\_hi AS "Total cumulé" TO results\_hi

ELSE IF Montant\_Facture >= 100

ASSIGN v\_running\_total\_med = v\_running\_total\_med + Montant\_Facture

EXTRACT No\_Facture, Montant\_Facture, Date\_facture, v\_running\_total\_med AS "Total cumulé" TO results\_med

ELSE

ASSIGN v\_running\_total\_low = v\_running\_total\_low + Montant\_Facture

EXTRACT No\_Facture, Montant\_Facture, Date\_facture, v\_running\_total\_low AS "Total cumulé" TO results\_low

END

Lorsque le script est exécuté, la commande GROUP teste le montant de facture pour chaque enregistrement. En fonction du montant, l'enregistrement est utilisé pour mettre à jour l'un des trois totaux cumulés (faible, moyen, élevé) et trois tables de résultats sont produites.

## LOOP à l'intérieur d'un GROUP

Lorsque vous utilisez GROUP pour traiter les enregistrements d'une table, vous pouvez utiliser une commande LOOP pour exécuter une série de commandes sur un même enregistrement plusieurs fois. LOOP est une deuxième itération se produisant à l'intérieur d'une itération de GROUP ; la commande s'exécute jusqu'à ce qu'une condition de test que vous avez spécifiée soit évaluée comme fausse.

## Utiliser LOOP pour fractionner un champ

La table suivante contient des données de facture et vous avez besoin d'isoler certaines informations pour des montants de facture par service. Une facture peut être reliée à plusieurs services et les codes des services sont enregistrés dans un format séparé par une virgule dans la table :

Numéro_fournisseur	Numéro_Facture	Date_Facture	Montant_Facture	Code_Département
11663	5981807	2000-11-17	618,30	CCD,RDR
13808	2275301	2000-11-17	6705,12	CCD
12433	6585673	2000-11-17	7955,46	CCD,LMO,RDR

Pour extraire les montants de facture par service, vous :

1. Utilisez une commande GROUP pour traiter la table, enregistrement par enregistrement.
2. Calculez le nombre de services ( $n$ ) associé à chaque enregistrement.
3. Utilisez la commande LOOP pour effectuer une itération sur l'enregistrement  $x$  fois afin d'extraire les données pour chaque service associé à l'enregistrement.

### Remarque

Vous devez augmenter la variable `v_counter` à l'intérieur de LOOP. Sinon, le test WHILE est toujours évalué sur vrai et le script entre dans une boucle infinie. Il est possible d'inclure une commande SET LOOP dans vos scripts pour vous prémunir contre les boucles infinies. Pour plus d'informations, consultez la section "Commande SET" Page 435.

```
COMMENT utilisez GROUP pour compter les virgules dans chaque champ de code de service afin
d'identifier le nombre de services associés à l'enregistrement exécutez "LOOP" sur chaque enre-
gistrement pour chaque code dans le champ, en extrayant chaque code dans son propre enre-
gistrement ENDGROUP v_department_count = OCCURS(Code_Département;',' ) v_counter = 0
LOOP WHILE v_counter <= v_department_count v_dept = SPLIT(Code_Département;',' , (v_counter
+ 1)) EXTRACT FIELDS No_Facture, Montant_Facture, v_dept AS "Service" TO result1 v_counter =
v_counter + 1 END END
```

Lorsque le script est exécuté, les commandes dans le bloc GROUP sont traitées sur chaque enregistrement de la table, de haut en bas. Pour chaque enregistrement, la commande LOOP effectue une itération sur l'enregistrement une fois par code de service dans la liste délimitées par une virgule, puis extrait un enregistrement. Si l'on pouvait décrire GROUP et LOOP pendant leur exécution, la procédure ressemblerait à ceci :

## Première itération de GROUP : 2 itérations de LOOP

Numéro_fournisseur	Numéro_Facture	Date_Facture	Montant_Facture	Code_Département
11663	5981807	2000-11-17	618,30	CCD,RDR
13808	2275301	2000-11-17	6705,12	CCD
12433	6585673	2000-11-17	7955,46	CCD,LMO,RDR

Pour le premier enregistrement dans la table, la valeur v\_department\_count est 1 donc LOOP effectuée deux itérations :

- Pour la première itération de LOOP :
  - v\_counter = 0
  - v\_depart = CCD

L'enregistrement suivant est extrait et la valeur v\_counter est augmentée de 1, c'est pourquoi LOOP effectuée une nouvelle itération :

5981807	618,30	CCD
---------	--------	-----

- Pour la deuxième itération de LOOP :
  - v\_counter = 1
  - v\_depart = RDR

L'enregistrement suivant est extrait et la valeur v\_counter est augmentée de 2, c'est pourquoi LOOP n'effectue plus d'itération et que GROUP passe à l'enregistrement suivant :

5981807	618,30	RDR
---------	--------	-----

## Deuxième itération de GROUP : 1 itération de LOOP

Numéro_fournisseur	Numéro_Facture	Date_Facture	Montant_Facture	Code_Département
11663	5981807	2000-11-17	618,30	CCD,RDR
13808	2275301	2000-11-17	6705,12	CCD
12433	6585673	2000-11-17	7955,46	CCD,LMO,RDR

Pour le deuxième enregistrement dans la table, la valeur v\_department\_count est 0 donc LOOP effectuée une seule itération :

- v\_counter = 0
- v\_depart = CCD

L'enregistrement suivant est extrait et la valeur v\_counter est augmentée de 1, c'est pourquoi LOOP n'effectue plus d'itération et que GROUP passe à l'enregistrement suivant :

2275301	6705,12	CCD
---------	---------	-----

## Troisième itération de GROUP : 3 itérations de LOOP

Numéro_fournisseur	Numéro_Facture	Date_Facture	Montant_Facture	Code_Département
11663	5981807	2000-11-17	618,30	CCD,RDR
13808	2275301	2000-11-17	6705,12	CCD
12433	6585673	2000-11-17	7955,46	CCD,LMO,RDR

Pour le troisième enregistrement dans la table, la valeur `v_department_count` est 2 donc LOOP effectuée trois itérations :

1. Pour la première itération de LOOP :

- `v_counter = 0`
- `v_depart = CCD`

L'enregistrement suivant est extrait et la valeur `v_counter` est augmentée de 1, c'est pourquoi LOOP effectuée une nouvelle itération :

6585673	7955,46	CCD
---------	---------	-----

2. Pour la deuxième itération de LOOP :

- `v_counter = 1`
- `v_depart = LMO`

L'enregistrement suivant est extrait et la valeur `v_counter` est augmentée de 1, c'est pourquoi LOOP effectuée une nouvelle itération :

6585673	7955,46	LMO
---------	---------	-----

3. Pour la troisième itération de LOOP :

- `v_counter = 2`
- `v_depart = RDR`

L'enregistrement suivant est extrait et la valeur `v_counter` est augmentée de 3, c'est pourquoi LOOP n'effectue plus d'itération et que GROUP atteint la fin de la table :

6585673	7955,46	RDR
---------	---------	-----

## Table de résultats finale

Une fois que GROUP a traité chaque enregistrement de la table, et que LOOP a itéré tous les codes de service, vous obtenez la table de résultats suivante :



Numéro_Facture	Montant_Facture	Département
5981807	618,30	CCD
5981807	618,30	RDR
2275301	6705,12	CCD
6585673	7955,46	CCD
6585673	7955,46	LMO
6585673	7955,46	RDR

# Top 30 des fonctions Analytics

Le top 30 des fonctions ACLScript s'avère utile pour différentes tâches. L'utilisation régulière de ces fonctions vous permet de préparer, d'analyser, de convertir et d'harmoniser les données dans vos scripts.

## Suppression des espaces de début et de fin

Les champs de type caractère dans les tables Analytics contiennent souvent des espaces de début ou de fin, car les champs présentent des largeurs de longueur fixe. Lorsque vous devez effectuer une opération à l'aide des données d'un champ de type caractère, vous pouvez supprimer ces espaces pour que la chaîne de caractères contienne uniquement les données réelles.

### ALLTRIM( )

Renvoie une chaîne dont les espaces de début et de fin ont été supprimés de la chaîne d'entrée.

#### Remarque

Il est recommandé d'utiliser ALLTRIM( ) sur n'importe quel champ de type caractère que vous utilisez comme entrée pour une autre fonction afin qu'aucune espace de début ou de fin n'affecte la valeur renvoyée.

#### Exemple

Le champ Numéro\_fournisseur contient la valeur " 1254". Vous devez supprimer cette espace supplémentaire de Numéro\_fournisseur pour pouvoir harmoniser ce champ avec les données d'une autre table.

```
COMMENT renvoie "1254"  
ALLTRIM(Numéro_fournisseur)
```

## Synchronisation de la casse alphabétique

La comparaison des chaînes dans Analytics est sensible à la casse. Il est donc utile de synchroniser la casse de toutes les données dans un champ avant d'effectuer toute comparaison, jointure ou relation à l'aide des données.

### UPPER( )

Renvoie une chaîne dont les caractères alphabétiques ont été convertis en majuscules.

#### Exemple

Le champ Nom contient la valeur "Smith". Vous devez mettre cette valeur en majuscules pour pouvoir la

comparer avec une valeur en majuscules d'une autre table.

```
COMMENT renvoie "SMITH"
UPPER(Nom)
```

## LOWER( )

Renvoie une chaîne dont les caractères alphabétiques ont été convertis en minuscules.

### Exemple

Le champ Nom contient la valeur "Smith". Vous devez mettre cette valeur en minuscules pour pouvoir la comparer avec une valeur en minuscules d'une autre table.

```
COMMENT renvoie "smith"
LOWER(Nom)
```

## PROPER( )

Renvoie une chaîne dans laquelle le premier caractère de chaque mot est en majuscule, tandis que les caractères restants sont définis en minuscules.

### Exemple

Le champ Nom contient la valeur "smith". Vous devez l'afficher sous forme de nom propre dans votre sortie.

```
COMMENT renvoie "Smith"
PROPER(Nom)
```

# Calcul et séparation de chaînes de caractères

Lorsque vous devez extraire un segment de données d'une longue chaîne de caractère ou lorsque vous devez tester certaines informations dans cette chaîne comme sa longueur ou son contenu, utilisez ces fonctions.

## SUBSTR( )

Renvoie une sous-chaîne indiquée d'une chaîne de caractères.

### Exemple

Le champ GL\_Code\_Compte contient la valeur "001-458-873-99". Vous devez extraire les trois premiers octets - ou caractères - de la chaîne de caractères.

```
COMMENT renvoie "001"
ASSIGN v_pos_début = 1
ASSIGN v_longueur = 3
SUBSTR(GL_Code_Compte; v_pos_début; v_longueur)
```

## LAST()

Renvoie un nombre spécifique de caractères à partir de la fin d'une chaîne.

### Exemple

Le champ `GL_Code_Compte` contient la valeur "001-458-873-99". Vous devez extraire les deux derniers octets - ou caractères - de la chaîne de caractères.

```
COMMENT renvoie "99"
ASSIGN v_car_num = 2
LAST(GL_Code_Compte; v_car_num)
```

## SPLIT()

Renvoie un segment spécifié d'une chaîne de caractères.

### Exemple

Le champ `GL_Code_Compte` contient la valeur "001-458-873-99". Vous devez extraire le deuxième segment du code de la chaîne de caractères.

```
COMMENT renvoie "458"
ASSIGN v_délimiteur = "-"
ASSIGN v_num_segment = 2
SPLIT(GL_Code_Compte; v_délimiteur; v_num_segment)
```

## AT()

Renvoie un nombre indiquant l'emplacement de début d'une occurrence spécifique d'une sous-chaîne dans une valeur de caractère.

### Exemple

Le champ `GL_Code_Compte` contient la valeur "001-458-873-99". Vous devez déterminer la position d'octet de départ de la valeur "458" pour tester si le deuxième segment du code GL est "458" (position de départ "5").

```
COMMENT renvoie "5"
ASSIGN v_occurrence = 1
ASSIGN v_sous-chaîne = "458"
AT(v_occurrence; v_sous-chaîne; GL_Code_Compte)
```

## OCCURS( )

Renvoie un décompte du nombre de fois qu'une sous-chaîne se produit dans une valeur de caractères indiquée.

### Exemple

Le champ `GL_Code_Compte` contient la valeur "001-458-873-99". Vous devez déterminer si le code GL est mis en forme correctement en vérifiant que les données contiennent les trois caractères traits d'union.

```
COMMENT renvoie "3"
ASSIGN v_sous-chaîne = "-"
OCCURS(GL_Code_Compte; v_sous-chaîne)
```

## LENGTH( )

Renvoie le nombre de caractères dans une chaîne.

### Exemple

Le champ `GL_Code_Compte` contient la valeur "001-458-873-99". Vous devez déterminer si le code GL est mis en forme correctement en vérifiant que les données contiennent 14 caractères.

```
COMMENT renvoie "14"
LENGTH(GL_Code_Compte)
```

## Conversion des types de données

Selon la source de données et les instructions d'importation ayant généré la table Analytics, il se peut que vous deviez convertir les valeurs d'un champ d'un type de données à un autre pour qu'une opération soit possible. Par exemple, pour effectuer des calculs sur les données importées sous forme de caractères ("12345"), vous devez les convertir en données numériques.

## STRING( )

Convertit une valeur numérique en chaîne de caractères.

### Exemple

Le champ `Montant_Facture` contient la valeur `12345,67`. Vous devez la convertir en données de type caractère.

```
COMMENT renvoie "12345,67"
ASSIGN v_longueur_chaine = 8
STRING(Montant_facture; v_longueur_chaine)
```

## VALUE( )

Convertit une chaîne de caractères en valeur numérique.

### Astuce

`VALUE( )` est souvent utilisée avec `ZONED( )` pour ajouter des zéros de début.

### Exemple

Le champ `Montant_Facture` contient la valeur `"12345,67"`. Vous devez la convertir en données numériques.

```
COMMENT renvoie 12345,67
VALUE(Montant_Facture)
```

## CTOD( )

Convertit une valeur de date au format Caractère ou Numérique en date. Cette fonction peut également extraire la date d'une valeur de type DateHeure au format Caractère ou Numérique et la renvoyer sous forme de date. Abréviation de "Caractère à Date".

### Exemple

Le champ `Date_soumission` contient la valeur `"25 avril 2016"`. Vous devez la convertir en données DateHeure.

```
COMMENT renvoie `20160425`
ASSIGN v_format_date = "mmm jj, aaaa"
CTOD(Date_soumission; v_format_date)
```

## DATE( )

Extrait la date d'une date ou d'une DateHeure spécifiée et la renvoie sous forme de chaîne de caractères. Peut également renvoyer la date du système d'exploitation en cours d'utilisation.

### Exemple

Le champ `Date_soumission` contient la valeur ``20160425``. Vous devez la convertir en données de type

caractère.

```
COMMENT renvoie "04/25/2016"
ASSIGN v_format_date = "MM/JJ/AAAA"
DATE(Date_soumission; v_format_date)
```

## Ajout de zéros de début

Convertissez des données numériques en données de type caractère et ajoutez des zéros de début à la sortie lorsque vous devez harmoniser des champs nécessitant des zéros de début.

### ZONED( )

Convertit les données numériques en caractères et ajoute des zéros de tête à la sortie.

#### Exemple

Le champ Numéro\_employé contient la valeur "254879". Vous devez convertir la valeur en chaînes de 10 chiffres avec des zéros de début.

#### Astuce

Vous devez utiliser la fonction VALUE() pour convertir le caractère en données numériques avant d'utiliser ces dernières comme entrée de ZONED().

```
COMMENT renvoie "0000254879"
ASSIGN v_longueur_chaine = 10
ASSIGN v_nombre_decimales = 0
ZONED(VALUE(Numéro_employé; v_nombre_decimales); v_longueur_chaine)
```

### BINTOSTR( )

Renvoie des données caractère Unicode converties à partir de données caractère ZONED ou EBCDIC. Abréviation de "Binaire à Chaîne".

#### Remarque

Édition Unicode uniquement Pour les éditions non-unicode, voir ZONED() ci-dessus.

#### Exemple

Le champ Numéro\_employé contient la valeur "254879". Vous devez convertir la valeur en chaînes de 10 chiffres avec des zéros de début.

**Astuce**

Vous devez utiliser la fonction `VALUE()` pour convertir le caractère en données numériques avant d'utiliser ces dernières comme entrée de `ZONED()`. Vous pouvez ensuite utiliser `BINTOSTR()` pour convertir les données ASCII qui proviennent de `ZONED()` en données Unicode.

```
COMMENT renvoie "0000254879"
ASSIGN v_longueur_chaine = 10
ASSIGN v_nombre_decimales = 0
ASSIGN v_type_chaine = "A"
BINTOSTR(ZONED(VALUE(Numéro_employé; v_nombre_decimales); v_longueur_chaine); v_
type_chaine)
```

## Extraction de parties DateHeure

Utilisez ces fonctions pour isoler et extraire des composants spécifiques d'une valeur DateHeure.

### MONTH( )

Extrait le mois d'une date ou d'une DateHeure spécifiée et le renvoie sous forme de valeur numérique (1 à 12).

#### Exemple

Le champ `Date_transaction` contient la valeur `20160815 100252`. Vous devez extraire le mois sous forme de données de type caractère avec un zéro de début.

```
COMMENT renvoie "08"
ASSIGN v_longueur_chaine = 2
ZONED(MONTH(Date_Transaction); v_longueur_chaine)
```

### DAY( )

Extrait le jour du mois d'une date ou d'une DateHeure spécifiée et le renvoie sous forme de valeur numérique (1 à 31).

#### Exemple

Le champ `Date_transaction` contient la valeur `20160815 100252`. Vous devez extraire le jour sous forme de données de type caractère.



```
COMMENT renvoie "15"
ASSIGN v_longueur_chaine = 2
STRING(DAY(Date_transaction); v_longueur_chaine)
```

## YEAR( )

Extrait l'année d'une date ou d'une DateHeure spécifiée et la renvoie sous forme de valeur numérique à l'aide du format AAAA.

### Exemple

Le champ `Date_transaction` contient la valeur `20160815 100252`. Vous devez extraire l'année sous forme de valeur numérique.

```
COMMENT renvoie 2016
YEAR(Date_transaction)
```

## HOUR( )

Extrait l'heure d'une heure ou d'une DateHeure spécifiée et la renvoie sous forme de valeur numérique à l'aide de l'horloge de 24 heures.

### Exemple

Le champ `Date_transaction` contient la valeur `20160815 100252`. Vous devez extraire les heures sous forme de valeur numérique.

```
COMMENT renvoie 10
HOUR(Date_transaction)
```

## MINUTE( )

Extrait les minutes d'une heure ou d'une DateHeure spécifiée et la renvoie sous forme de valeur numérique.

### Exemple

Le champ `Date_transaction` contient la valeur `20160815 100252`. Vous devez extraire les minutes sous forme de valeur numérique.

```
COMMENT renvoie 2
MINUTE(Date_transaction)
```

## SECOND( )

Extrait les secondes d'une heure ou d'une DateHeure spécifiée et les renvoie sous forme de valeur numérique.

### Exemple

Le champ Date\_transaction contient la valeur `20160815 100252`. Vous devez extraire les secondes sous forme de valeur numérique.

```
COMMENT renvoie 52  
SECOND(Date_transaction)
```

## CDOW( )

Revoie le nom du jour de la semaine de la date ou de la DateHeure indiquée. Abréviation de "Jour de la semaine au format caractère".

### Exemple

Le champ Date\_transaction contient la valeur `20160815 100252`. Vous devez extraire le nom du jour sous forme de données de type caractère.

```
COMMENT renvoie "Lun"  
CDOW(Date_transaction; 3)
```

## CMOY( )

Revoie le nom du mois de l'année de la date ou de la DateHeure indiquée. Abréviation de "Mois de l'année au format caractère".

### Exemple

Le champ Date\_transaction contient la valeur `20160815 100252`. Vous devez extraire le nom du mois sous forme de données de type caractère.

```
COMMENT renvoie "Août"  
CMOY(Date_transaction; 3)
```

## Manipulation des chaînes

Supprimez ou remplacez des segments de champs de type caractère à l'aide de ces fonctions.

## INCLUDE( )

Renvoie une chaîne qui inclut uniquement les caractères spécifiés.

### Exemple

Le champ Adresse contient la valeur « Entreprise 12345 ABC ». Vous devez extraire le numéro de l'adresse et exclure le nom de la société.

```
COMMENT renvoie "12345"
ASSIGN v_car_à_retourner = "0123456789"
INCLUDE(Adresse; v_car_à_retourner)
```

## EXCLUDE( )

Renvoie une chaîne qui exclut les caractères spécifiés.

### Exemple

Le champ Adresse contient la valeur « Entreprise 12345 ABC ». Vous devez extraire le nom de la société et exclure le numéro de l'adresse.

```
COMMENT renvoie « Entreprise ABC »
ASSIGN v_car_à_exclure = "0123456789"
EXCLUDE(Adresse; v_car_à_exclure)
```

## REPLACE( )

Remplace toutes les instances d'une chaîne de caractères indiquée par une nouvelle chaîne de caractères.

### Exemple

Le champ Adresse contient la valeur « Acme&Fils ». Vous devez remplacer le caractère "&" par le mot " et ".

```
COMMENT renvoie "12345 Acme et Fils"
ASSIGN v_car_cible = "&"
ASSIGN v_car_remplacement = " et "
REPLACE(Adresse; v_car_cible; v_car_remplacement)
```

## OMIT( )

Renvoie une chaîne dont une ou plusieurs sous-chaînes spécifiques ont été supprimées.

### Exemple

Le champ Adresse contient la valeur « 12345 Fake St ». Vous devez extraire l'adresse sans le suffixe de la rue.

```
COMMENT renvoie "12345 Fake"
ASSIGN v_car_à_omettre = "St"
OMIT(Adresse; v_car_à_omettre)
```

## REVERSE( )

Renvoie une chaîne dont l'ordre des caractères est inversé.

### Exemple

Le champ Ligne\_état contient la valeur "001 Correction 5874,39 CR ". Vous devez inverser la valeur et ignorer les espaces de début ou de fin.

```
COMMENT renvoie "RC 93.4785 noitcerroC 100"
REVERSE(ALLTRIM(Ligne_état))
```

## BLANKS( )

Renvoie une chaîne contenant un nombre spécifique d'espaces.

### Exemple

Vous devez créer un champ calculé pour un nom de région à partir d'une valeur dans le champ code\_région. Vous devez vous assurer que la valeur par défaut que vous spécifiez à la fin de la commande est au moins aussi longue que la plus longue valeur d'introduction.

```
COMMENT BLANKS retourne une chaîne de 8 caractères " "
ASSIGN v_longueur = 8
DEFINE FIELD région COMPUTED

"Sud" IF code_région = 1
"Nord" IF code_région = 2
"Est" IF code_région = 3
"Ouest" IF code_région = 4
BLANKS(v_longueur)
```

# Commandes

# Commande ACCEPT

Crée une boîte de dialogue invitant de manière interactive les utilisateurs à saisir une ou plusieurs valeurs d'introduction de script. Chaque valeur d'introduction est stockée dans une variable nommée de type caractère.

## Remarque

L'utilisation de la commande ACCEPT pour saisir des mots de passe n'est pas sécurisée. Il faudrait utiliser la "Commande PASSWORD" Page 373 à la place.

La commande ACCEPT n'est pas prise en charge par les outils d'analyse d'AX Serveur.

Vous pouvez créer une boîte de dialogue interactive plus avancée avec la "Commande DIALOG" Page 153.

## Syntaxe

```
ACCEPT {texte_message <FIELDS catégorie_élément_projet> TO nom_variable} <...n>
```

## Paramètres

Nom	Description
<i>texte_message</i>	<p>Libellé affiché dans la boîte de dialogue qui invite l'utilisateur à intervenir. Doit être une chaîne entre guillemets ou une variable de type caractère.</p> <p>Lorsque vous saisissez plusieurs invites, vous pouvez les séparer par des virgules. L'utilisation de virgules n'est pas obligatoire, mais cela améliore la lisibilité du script :</p> <pre>ACCEPT "Préciser une date de début ." TO v_date_début, "Préciser une date de fin ." TO v_date_fin</pre>
FIELDS <i>catégorie_élément_projet</i> optionnel	<p>Crée une liste déroulante d'éléments de projet pour la saisie de l'utilisateur plutôt qu'une zone de texte. L'utilisateur peut sélectionner un seul élément de projet, un seul champ ou une seule variable dans la liste.</p> <p><i>catégorie_élément_projet</i> indique les types d'élément à afficher dans la liste. Par exemple, saisir <i>xf</i> affiche toutes les tables du projet dans la liste. Entourez <i>project_item_category</i> de guillemets :</p> <pre>FIELDS "xf"</pre> <p>Pour obtenir les codes que vous avez utilisés pour spécifier des catégories, reportez-vous à la section "Codes pour les catégories d'élément de projet" Page 57.</p> <p>Vous pouvez spécifier plus d'un code dans la même invite, mais vous ne pouvez pas mélanger des éléments de projet, des champs ou des variables.</p>

Nom	Description
TO <i>nom_variable</i>	<p>Nom de la variable de type de caractère à utiliser pour stocker l'entrée utilisateur. Si la variable n'existe pas, elle est créée.</p> <p>Si la variable existe déjà, sa valeur actuelle est affichée dans la boîte de dialogue comme valeur par défaut.</p> <p><b>Remarque</b></p> <p>Vous ne pouvez pas utiliser des caractères qui n'existent pas en anglais, tels que é, dans les noms de variables utilisées pour substituer des variables. Les noms de variables contenant des caractères n'existant pas en anglais provoqueront l'échec du script.</p> <p>La commande ACCEPT crée uniquement des variables de type caractère. Si vous devez avoir des entrées d'un autre type de données, vous devez convertir la variable caractère dans le type requis dans le traitement suivant du script. Pour plus d'informations, consultez la section "Type de données d'entrée" Page 58.</p>

## Exemples

### L'utilisateur est invité à sélectionner la table Analytics à ouvrir

Il vous faut une boîte de dialogue qui invite l'utilisateur à sélectionner le nom de la table à ouvrir. Le script ouvre ensuite la table sélectionnée par l'utilisateur :

```
ACCEPT "Sélectionne la table à ouvrir : " FIELDS "xf" TO v_nom_table
OPEN %v_nom_table%
```

Les symboles de pourcentage sont requis, car ils indiquent que le nom de la table à ouvrir est stocké dans la variable *v\_nom\_table*. Si les symboles de pourcentage sont omis, le script tente d'ouvrir une table nommée "v\_nom\_table".

### Utilisation de plusieurs boîtes de dialogue pour recueillir les entrées requises

Vous souhaitez créer une boîte de dialogue distincte pour chaque valeur que l'utilisateur du script doit saisir.

Utilisez une seule chaîne d'invite dans chaque instance de la commande ACCEPT. Le script génère des boîtes de dialogue séparées pour indiquer chacun des éléments suivants :

- un nom de table
- un champ sur lequel effectuer l'échantillonnage
- un intervalle d'échantillonnage
- une valeur de départ aléatoire

```
ACCEPT "Saisir le nom de la table à analyser" TO v_nom_table
OPEN %v_nom_table%
ACCEPT "Sélectionner le champ à échantillonner" FIELDS "N" TO v_champ_à_échantillonner
ACCEPT "Saisir l'intervalle d'échantillonnage" TO v_intervalle_échantillonnage
ACCEPT "Saisir la valeur de départ aléatoire" TO v_valeur_départ_aléatoire
SAMPLE ON %v_champ_à_échantillonner% INTERVAL v_intervalle_échantillonnage FIXED v_valeur_départ_aléatoire RECORD TO Sortie_échantillonnage OPEN
```

## Lorsque le script s'exécute

1. La première boîte de dialogue demande le nom de la table.
2. La deuxième boîte de dialogue, avec FIELDS "N", demande une sélection de champ à partir d'une liste déroulante de champs numériques.
3. La troisième boîte de dialogue demande la valeur de l'intervalle.
4. La quatrième boîte de dialogue demande la valeur de départ aléatoire.

## Utilisation d'une boîte de dialogue unique avec plusieurs invites pour recueillir les entrées requises

Vous souhaitez créer une seule boîte de dialogue pour toutes les valeurs que l'utilisateur du script doit saisir.

Vous utilisez plusieurs invites séparées par des virgules dans la commande ACCEPT pour demander à l'utilisateur plusieurs valeurs d'entrée. La même boîte de dialogue contient des invites pour la date de départ et pour la date de fin d'une plage de dates :

```
ACCEPT "Préciser une date de début :." TO v_date_début, "Préciser une date de fin :." TO v_date_fin
```

# Remarques

## Interactivité

La commande ACCEPT permet de créer un script interactif. Pendant le traitement de la commande ACCEPT, le script s'interrompt et une boîte de dialogue invite l'utilisateur à saisir des entrées utilisées par Analytics lors d'un traitement ultérieur.

Vous pouvez créer des boîtes de dialogue indépendantes qui invitent l'utilisateur à entrer un seul élément à la fois ou bien créer une seule boîte de dialogue qui invite à entrer plusieurs éléments.

## DIALOG et ACCEPT

La commande DIALOG vous permet de créer une boîte de dialogue interactive plus avancée qui peut avoir un ou plusieurs des types de contrôles suivants :



- zone de texte
- case à cocher
- Cases d'option
- liste déroulante de valeurs personnalisées
- liste de projets

Vous pouvez également personnaliser la mise en page de la boîte de dialogue. Pour plus d'informations, consultez la section "Commande DIALOG" Page 153.

## Codes pour les catégories d'élément de projet

Utilisez les codes suivants pour indiquer la catégorie d'élément de projet à afficher dans une liste déroulante.

### Catégories de projet

Code	Catégorie
xf	Tables
xb	Scripts
xi	Index
xr	Vues et rapports
xw	Environnements_de_travail

### Catégories de champ

Code	Catégorie
C	Champs de type caractère
N	Champs numériques
D	Champs de type DateHeure
L	Champs logiques

### Catégories de variable

Code	Catégorie
c	Variables de type caractère

Code	Catégorie
n	Variables numériques
d	Variables de type DateHeure
l	Variables logiques

## Type de données d'entrée

ACCEPT stocke l'entrée utilisateur dans une ou plusieurs variables de type caractère. Si une saisie numérique ou de type DateHeure est requise, vous pouvez utiliser les fonctions VALUE( ) ou CTOD( ) pour convertir le contenu d'une variable de type caractère en valeur numérique ou en valeur DateHeure :

```
SET FILTER TO BETWEEN(%v_champ_date%; CTOD(%v_date_début%); CTOD(%v_date_fin%))
```

Dans l'exemple, les dates de début et de fin pour ce filtre sont stockées sous type de valeurs caractère. Elles doivent être converties en valeurs date afin d'être utilisées avec un champ date utilisant un type de données DateHeure.

Entourer le nom de la variable par des symboles de pourcentage (%) remplace la valeur de type caractère contenue par la variable pour le nom de la variable. La fonction CTOD( ) convertit alors la valeur de type caractère en une valeur de date.

## Place de la commande ACCEPT

Dans la mesure du possible, il est recommandé de placer toutes les commandes ACCEPT au début d'un script. Si vous demandez la saisie de toutes les données au début, le script peut s'exécuter sans entrave une fois les informations requises saisies par l'utilisateur.

### Remarque

Vous ne pouvez pas utiliser la commande ACCEPT dans la commande GROUP.

# Commande ACCESSDATA

Importe des données à partir de diverses sources de données compatibles ODBC.

La forme de la commande est ACCESSDATA64 ou ACCESSDATA32 selon que vous utilisez un pilote ODBC 64 bits ou 32 bits.

## Syntaxe

```
{ACCESSDATA64 | ACCESSDATA32} {CONNECTOR | ODBC {"Pilote"|"Dsn"|"Fichier"}} NAME
valeur <USER id_utilisateur> <PASSWORD num | PROMPT_PASSWORD> TO nom_table
CHARMAX longueur_champ_max MEMOMAX longueur_champ_max <ALLCHARACTER>
SOURCE (paramètres_connexion) <HASH(valeur_salt; champs)>
SQL_QUERY(
(syntaxe_SQL)
)END_QUERY
```

## Paramètres

Nom	Description
CONNECTOR   ODBC {"Pilote" "Dsn" "Fichier"}	Type de connexion ODBC que vous souhaitez effectuer : <ul style="list-style-type: none"> <li>○ <b>CONNECTOR</b> : connectez-vous à l'aide d'un connecteur de données Analytics natif</li> <li>○ <b>ODBC "Pilote"</b> : connectez-vous à l'aide d'un pilote ODBC Windows installé sur votre ordinateur</li> <li>○ <b>ODBC "Dsn"</b> : connectez-vous à l'aide d'un DSN (nom de la source de données) enregistré sur votre ordinateur</li> <li>○ <b>ODBC "Fichier"</b> : connectez-vous à l'aide d'un fichier DSN (un fichier .dsn enregistré)</li> </ul>
NAME <i>valeur</i>	Nom du connecteur de données Analytics, du pilote ODBC ou du DSN. Par exemple : <ul style="list-style-type: none"> <li>○ NAME "Amazon Redshift"</li> <li>○ NAME "Pilote Microsoft Access (*.mdb, *.accdb)"</li> <li>○ NAME "Mon DSN Excel"</li> <li>○ NAME "excel.dsn"</li> </ul>
USER <i>id_utilisateur</i> optionnel	ID utilisateur pour les sources de données exigeant un ID utilisateur.
PASSWORD <i>num</i>   PROMPT_PASSWORD optionnel	Pour les sources de données nécessitant un mot de passe : <ul style="list-style-type: none"> <li>○ <b>PASSWORD num</b> : utilisez la définition de mot de passe numérotée</li> <li>○ <b>PROMPT_PASSWORD</b> : affiche une invite de mot de passe</li> </ul> L'invite de mot de passe permet aussi de modifier <i>id_utilisateur</i> .

Nom	Description
	<p>Si vous utilisez <code>PASSWORD num</code>, vous devez spécifier une définition de mot de passe créée auparavant. Pour plus d'informations, consultez la section "Commande <code>PASSWORD</code>" Page 373 et "Commande <code>SET</code>" Page 435.</p> <p><b>Astuce</b></p> <p>L'utilisation de la commande <code>PASSWORD</code> conjointement avec <code>PASSWORD num</code> ressemble à l'utilisation du paramètre <code>PROMPT_PASSWORD</code>. Les deux méthodes demandent un mot de passe à l'utilisateur. <code>PROMPT_PASSWORD</code> présente l'intérêt d'autoriser la mise à jour de <code>id_utilisateur</code>.</p>
<p><code>TO nom_table</code></p>	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li>◦ <b><code>nom_table</code></b> : enregistre les résultats dans une table Analytics</li> </ul> <p>Indiquez <code>nom_table</code> sous forme de chaîne entre guillemets avec une extension de fichier <code>.FIL</code>. Par exemple : <code>TO "Sortie.FIL"</code></p> <p>Par défaut, le fichier de données de table (<code>.FIL</code>) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• <code>TO "C:\Sortie.FIL"</code></li> <li>• <code>TO "Résultats\Sortie.FIL"</code></li> </ul> <p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alphanumériques, sans l'extension <code>.FIL</code>. Le nom peut inclure le caractère de soulignement (<code>_</code>), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>
<p><code>CHARMAX longueur_champ_max</code></p>	<p>La longueur maximale en caractères d'un champ de la table Analytics qui provient d'une donnée de type caractère dans la source que vous importez.</p> <p>La valeur par défaut est 50. Les données dépassant la longueur de champ maximale sont tronquées lors de l'importation dans Analytics.</p>
<p><code>MEMOMAX longueur_champ_max</code></p>	<p>Longueur maximale en caractères des champs de texte, de commentaire ou mémo que vous importez.</p> <p>La valeur par défaut est 100. Les données dépassant la longueur de champ maximale sont tronquées lors de l'importation dans Analytics.</p>
<p><code>ALLCHARACTER</code> optionnel</p>	<p>Attribue automatiquement le type de données caractère à tous les champs importés.</p> <p>Une fois que les données sont dans Analytics, vous pouvez affecter différents types (Numérique ou DateHeure par exemple) de données aux champs et en spécifier les informations de format.</p> <p><b>Astuce</b></p> <p><code>ALLCHARACTER</code> s'avère utile si vous importez une table contenant des valeurs d'identifiants numériques. Vous pouvez utiliser <code>ALLCHARACTER</code> pour éviter que Analytics attribue automatiquement le type de données Numérique aux valeurs devant utiliser le type de données Caractère.</p>

Nom	Description
SOURCE <i>paramètres_ connexion</i>	Paramètres de connexion (chaîne de connexion) requis pour se connecter à la source de données.
HASH( <i>valeur_salt</i> , <i>champs</i> ) optionnel	<p>Importe les champs spécifiés sous forme de valeurs de hachage cryptographiques. Les valeurs de hachage sont des transformations à sens unique. Elles ne peuvent pas être décodées une fois les champs importés :</p> <ul style="list-style-type: none"> <li>◦ <b>valeur_salt</b> : chaîne de caractères alphanumériques concaténée aux valeurs des données source pour renforcer le hachage des valeurs dans les champs. Saisissez la valeur de hachage sous forme de chaîne entre guillemets.  La valeur salt est limitée à 128 caractères. N'utilisez aucun des caractères suivants : ( ) "</li> <li>◦ <b>champs</b> : liste d'un ou plusieurs champs à hacher. Saisissez les champs sous forme de chaîne entre guillemets et séparez chaque champ par un point-virgule.</li> </ul> <p>Vous devez indiquer le nom du champ que vous voyez dans l'aperçu de la fenêtre Accès aux données et dans la zone de transit, et non le nom du champ physique dans la source de données.</p> <p><b>Remarque</b></p> <p>Le nom du champ affiché dans l'aperçu de la fenêtre Accès aux données est la valeur d'alias du champ dans la requête SQL ("nom_champ" AS "alias"). Vous devez utiliser la valeur de l'alias pour référencer des champs.</p> <pre>HASH("QZ3x7"; "SSN_NO; CC_NO; Nom")</pre> <p>Pour obtenir des informations sur la comparaison de valeurs hachées pendant l'importation avec des valeurs hachées dans ACLScript, consultez la rubrique "Comparaison de données hachées à l'aide d'ACCESSDATA avec des données hachées à l'aide de la fonction HASH( ) d'ACLScript" Page 65.</p>
SQL_QUERY ( <i>syntaxe_ SQL</i> ) END_QUERY	<p>Instruction IMPORT de SQL.</p> <p>Tout ce qui se trouve entre parenthèses fait partie de la requête SQL et doit être une requête SQL valide.</p>

## Exemples

### Importation de données à l'aide d'un connecteur de données Analytics natif

Vous devez importer des données du service de données cloud Amazon Redshift. Pour ce faire, vous utilisez le connecteur de données Analytics Amazon Redshift :

```
ACCESSDATA64 CONNECTOR NAME "Amazon Redshift" USER "utilisateur_ACL" PROMPT_
PASSWORD TO "Historique_Autorisations.FIL" CHARMAX 50 MEMOMAX 100
```

```

SOURCE( bool-
saschar-
=0;c-
ache-
size-
=100-
;da-
taba-
se=-
usa-
ge;-
decla-
refetch-
mode=0;maxbytea=255;maxlongvarchar=8190;maxvarchar=255;port=5439;servername=acl_
test.-
high-
bond.-
com;-
singlerowmode=1;sslmode=require;textaslongvarchar=0;usemultiplestatements=0;useunicode=1)
SQL_QUERY(
SELECT
"historique_autorisations"."organisation" AS "organisation",
"historique_autorisations"."email_utilisateur" AS "email_utilisateur",
"historique_autorisations"."id_plan" AS "id_plan",
"historique_autorisations"."date_début" AS "date_début",
"historique_autorisations"."date_fin" AS "date_fin"
FROM
"prm"."historique_autorisations" "historique_autorisations"
)) END_QUERY

```

## Importation de données à l'aide d'un pilote ODBC Windows

Vous devez importer des données à partir d'une base de données Microsoft Access. Pour ce faire, vous utilisez un pilote ODBC Windows pour vous connecter à MS Access et effectuer l'importation :

```

ACCESSDATA32 ODBC "Pilote" NAME "Pilote Microsoft Access (*.mdb)" TO "Factures.FIL"
CHARMAX 50 MEMOMAX 100
SOURCE( dbq=C:\Users\lachlan_murray\Documents\ACL Data\Exemple de fichiers de don-
nées\Sample.mdb;defaultdir=C:\Users\lachlan_murray\Documents\ACL Data\Exemple de fichiers de
données;driverid=281;fil=MS Access;-
max-
buf-
fersize=2048;maxscanrows=8;pagetimeout=5;safetransactions=0;threads=3;usercommitsync=Yes)
SQL_QUERY(

```

```

SELECT
  `Client`.`ID_Client` AS `ID_Client`,
  `Client`.`Société` AS `Société`,
  `Client`.`Adresse` AS `Adresse`,
  `Client`.`Ville` AS `Ville`,
  `Client`.`Région` AS `Région`,
  `Client`.`CodePostal` AS `CodePostal`,
  `Client`.`Pays` AS `Pays`,
  `Client`.`Téléphone` AS `Téléphone`,
  `Commandes`.`ID_commande` AS `ID_commande`,
  `Commandes`.`ID_Client` AS `ID_Client_Commandes`,
  `Commandes`.`IDProd` AS `IDProd`,
  `Commandes`.`DateCommande` AS `DateCommande`,
  `Commandes`.`Quantité` AS `Quantité`,
  `Produit`.`IDProd` AS `IDProd_Produit`,
  `Produit`.`Nom_de_produit` AS `Nom_de_produit`,
  `Produit`.`Prix_unitaire` AS `Prix_unitaire`,
  `Produit`.`Descript` AS `Descript`,
  `Produit`.`ShipWt` AS `ShipWt`
FROM
  (`Client` `Client`
INNER JOIN
  `Commandes` `Commandes`
    ON `Client`.`ID_Client` = `Commandes`.`ID_Client`
  )
INNER JOIN
  `Produit` `Produit`
    ON `Commandes`.`IDProd` = `Produit`.`IDProd`
WHERE
  (
    `Client`.`Région` = 'BC'
  OR `Client`.`Région` = 'WA'
  )
)END_QUERY

```

## Importation de données à l'aide d'un DSN (nom de source de données) Windows

Vous devez importer des données à partir d'un fichier Microsoft Excel. Pour ce faire, vous utilisez un DSN Windows pour vous connecter à Excel et effectuer l'importation :

```

ACCESSDATA32 ODBC "Dsn" NAME "Fichiers Excel" TO "Trans_Avril_15_limite.FIL" CHARMAX 50
MEMOMAX 100

```

```

SOURCE( dbq=C:\Users\lachlan_murray\Documents\ACL Data\Exemple de fichiers de don-
nées\Trans_Avril.xls;defaultdir=C:\Users\lachlan_murray\Documents\ACL Data\Exemple de fichiers
de données;driverid=1046;maxbufferize=2048;pagetimeout=5)
SQL_QUERY(
  SELECT
    `Trans_Avr_`.`NUMCARTE` AS `NUMCARTE`,
    `Trans_Avr_`.`MONTANT` AS `MONTANT`,
    `Trans_Avr_`.`DATE_TRANS` AS `DATE_TRANS`,
    `Trans_Avr_`.`CODES` AS `CODES`,
    `Trans_Avr_`.`NUMCLIENT` AS `NUMCLIENT`,
    `Trans_Avr_`.`DESCRIPTION` AS `DESCRIPTION`
  FROM
    `Trans_Avr$`.`Trans_Avr_`
  WHERE
    (
      `Trans_Avr_`.`DATE_TRANS` <= {ts '2003-04-15 00:00:00'}
    )
)END_QUERY

```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Création de paramètres de connexion ODBC et d'instructions IMPORT SQL

Les paramètres de connexion ODBC ainsi que les instructions IMPORT SQL sont souvent assez longs et complexes, comme illustré dans les exemples.

Le moyen le plus simple de créer ces portions de la commande ACCESSDATA consiste à utiliser dans un premier temps la fenêtre d'accès aux données dans Analytics pour vous connecter à la source de données cible, puis à importer les données dans un deuxième temps. Vous pouvez ensuite copier l'intégralité de la commande ACCESSDATA à partir de la trace, y compris les paramètres de connexion et l'instruction IMPORT, et personnaliser la commande selon vos besoins.

## Valeur de mot de passe supprimée

Lorsque vous utilisez la fenêtre Accès aux données dans Analytics pour exécuter la commande ACCESSDATA et que vous fournissez un mot de passe, la valeur du mot de passe n'est pas écrite dans la trace. À la place, le paramètre PROMPT\_PASSWORD est substitué.



## Fichiers traces ACCESSDATA

Deux fichiers traces enregistrent les transactions associées à la commande ACCESSDATA. Ils peuvent être utilisés pour résoudre des problèmes en cas d'échec de connexion des données :

- **ServerDataAccess.log** : enregistre toutes les activités et erreurs avant l'importation des données

Emplacement : `C:\Users\<compte utilisateur>\AppData\Local\ACL\ACL for Windows\Data Access\ServerDataAccess.log`

### Remarque

Le « Serveur » dans `ServerDataAccess.log` fait référence au composant d'accès aux données d'Analytics s'exécutant en local sur l'ordinateur où Analytics est installé.

- **DataAccess.log** : enregistre les informations sur l'opération d'importation et le projet Analytics dans lequel vous importez des données

Emplacement : `..\<dossier de projet Analytics>\DataAccess.log`

## Comparaison de données hachées à l'aide d'ACCESSDATA avec des données hachées à l'aide de la fonction HASH( ) d'ACLScript

Même si vous ne pouvez pas lire les valeurs brutes des données hachées, elles restent utiles pour combiner ou analyser des données.

Si vous souhaitez comparer des valeurs hachées par ACCESSDATA pendant l'importation avec des valeurs hachées à l'aide de la fonction HASH( ) d'ACLScript, vous devez convertir les champs numériques ou DateHeure Analytics en valeurs de type caractère et supprimer l'intégralité des espaces de fin et de début avant de hacher les données.

Les champs DateHeure doivent utiliser les formats suivants lorsqu'ils sont convertis en caractère :

- **DateHeure** : "AAAA-MM-JJ hh:mm:ss"
- **Date** : "AAAA-MM-JJ"
- **Heure** : "hh:mm:ss"

L'exemple qui suit utilise les fonctions STRING( ) et ALLTRIM( ) pour convertir un champ numéro de carte de crédit en données de type caractère avant de hacher la valeur à l'aide de la fonction HASH( ) d'ACLScript :

```
COMMENT ACL HASH fonction utilisée après l'importation des données
HASH(ALLTRIM(STRING(CC_No; 16))); "QZ3x7")
```

Une fois que vous avez haché les valeurs Analytics, vous pouvez les comparer aux valeurs hachées dans le cadre de l'importation de commande ACCESSDATA.

# Commande ACTIVATE

Ajoute des définitions de champ stockées dans un environnement de travail Analytics au jeu de définitions de champs existant dans un format de table Analytics.

## Syntaxe

```
ACTIVATE <WORKSPACE> nom_environnement <OK>
```

## Paramètres

Nom	Description
WORKSPACE <i>nom_environnement</i>	Nom de l'environnement de travail à activer.
OK optionnel	Supprime ou écrase des éléments sans demander de confirmation. Si un champ de la table porte le même nom qu'un champ de l'environnement de travail activé, il sera remplacé sans confirmation. Vous ne pouvez pas remplacer un champ référencé par un champ calculé.

## Exemples

### Activation d'un environnement de travail dans votre projet Analytics

Vous activez l'environnement de travail **FormulesComplexes** :

```
ACTIVATE WORKSPACE FormulesComplexes OK
```

### Activation d'un environnement de travail enregistré sous un fichier .wsp dans le même dossier que votre projet Analytics

Vous activez l'environnement de travail **FormulesComplexes** qui a été enregistré dans un fichier .wsp :

```
ACTIVATE WORKSPACE FormulesComplexes.WSP OK
```

# Remarques

## Fonctionnement

ACTIVATE rend les définitions de champ de l'environnement de travail disponibles pour la table active. Une fois un environnement de travail activé, ses champs peuvent être utilisés avec la table active jusqu'à sa fermeture.

## Modification des formats de table

Les champs de l'environnement de travail sont définitivement ajoutés au format de table si :

- vous modifiez le format de table après avoir activé un environnement de travail
- vous effectuez une modification entraînant l'enregistrement du format de table

Une fois que les champs de l'environnement de travail sont enregistrés dans le format de table, vous pouvez :

1. Utilisez la commande DEFINE COLUMN pour ajouter les champs à une vue.
2. Utilisez la commande SAVE pour enregistrer vos modifications.

# Commande AGE

Regroupe les enregistrements en périodes d'antériorité à partir des valeurs dans un champ date ou DateHeure. Compte le nombre d'enregistrements dans chaque période et calcule également le sous-total des champs numériques spécifiés pour chaque période.

## Syntaxe

```
AGE <ON> champ_date <CUTOFF date_référence> <INTERVAL jours <;...n>> <SUPPRESS>
<SUBTOTAL champ_numérique <...n>|SUBTOTAL ALL> <IF test> <WHILE test> <FIRST
plage|NEXT plage> <TO {SCREEN|nom_fichier|GRAPH|PRINT}> <KEY champ_fractionnement>
<HEADER texte_en-tête> <FOOTER texte_pied_page> <APPEND> <LOCAL> <STATISTICS>
```

## Paramètres

Nom	Description
ON <i>champ_date</i>	Nom du champ de date ou DateHeure ou de l'expression à classer par antériorité. Bien que vous puissiez classer par antériorité un champ de DateHeure, seule la portion date des valeurs DateHeure est prise en compte. La portion heure est ignorée. Vous ne pouvez pas faire un classement par antériorité sur des données de temps uniquement.
CUTOFF <i>date_référence</i> optionnel	Date à laquelle les valeurs dans <i>champ_date</i> sont comparées. Vous devez indiquer la <i>date_référence</i> sous forme de chaîne sans guillemets au format AAMMJJ ou AAAAMMJJ, quel que soit le format du champ de date. Par exemple : CUTOFF 20141231 Si vous ignorez CUTOFF, c'est la date système actuelle qui est utilisée comme date de référence.
INTERVAL <i>jours &lt;;...n&gt;</i> optionnel	Intervalle de date (c'est-à-dire le nombre de jours) servant à calculer les périodes d'antériorité. <i>jours</i> représente le début de chaque période d'antériorité mesurée à rebours à partir de la <i>date_référence</i> : <ul style="list-style-type: none"> <li>la première valeur <i>jours</i> identifie le début de la première période d'antériorité</li> <li>une première valeur <i>jours</i> de « 0 » indique que la première période d'antériorité commence à la <i>date_référence</i> spécifiée</li> <li>la dernière valeur <i>jours</i> identifie la fin de la dernière période d'antériorité</li> </ul> Vous devez indiquer les intervalles sous forme de chaîne de valeurs séparées par des virgules, sans guillemets : <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <pre>INTERVAL 0;90;180;270;365</pre> </div>

Nom	Description
	<p>Les périodes d'antériorité par défaut sont 0; 30; 60; 90; 120 et 10 000 jours. Un intervalle de 10 000 jours sert à isoler les enregistrements comportant des dates qui sont probablement non valides.</p> <p>Si nécessaire, ces intervalles de dates peuvent être personnalisés pour refléter d'autres rapports internes de classement par antériorité.</p>
SUPPRESS optionnel	Supprime les dates de la sortie de la commande ne tombant pas dans la période d'antériorité.
SUBTOTAL <i>champ_numérique &lt;...n&gt;</i>   SUBTOTAL ALL optionnel	<p>Un ou plusieurs champs ou expressions numériques à sous-totaliser pour chaque groupe.</p> <p>Les champs multiples doivent être séparés par des espaces. Indiquez ALL pour calculer le sous-total de tous les champs numériques de la table.</p>
IF <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.</p> <p><b>Remarque</b></p> <p>La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p>
WHILE <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.</p> <p><b>Remarque</b></p> <p>Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.</p>
FIRST <i>plage</i>   NEXT <i>plage</i> optionnel	<p>Le nombre d'enregistrements à traiter :</p> <ul style="list-style-type: none"> <li><b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li><b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> </ul> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter.</p> <p>Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p>
TO SCREEN   <i>nom_fichier</i>   GRAPH   PRINT	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li><b>SCREEN</b> : affiche les résultats dans la zone d'affichage d'Analytics.</li> <li><b><i>nom_fichier</i></b> : enregistre les résultats dans un fichier</li> </ul> <p>Indiquez <i>nom_fichier</i> sous forme de chaîne entre guillemets avec l'extension de fichier appropriée. Par exemple : TO "Sortie.TXT"</p> <p>Par défaut, le fichier est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>TO "C:\Sortie.TXT"</li> </ul>

Nom	Description
	<ul style="list-style-type: none"> <li>• TO "Résultats\Sortie.TXT"</li> <li>◦ <b>GRAPH</b> : affiche les résultats dans un graphique dans la zone d'affichage d'Analytics</li> <li>◦ <b>PRINT</b> : envoie les résultats vers l'imprimante par défaut</li> </ul>
KEY <i>champ_fractionnement</i> optionnel	<p>Champ ou expression regroupant les calculs de sous-totaux. Un sous-total est calculé chaque fois que la valeur de <i>champ_fractionnement</i> change.</p> <p><i>champ_fractionnement</i> doit être un champ ou une expression de type caractère. Vous ne pouvez indiquer qu'un seul champ, mais vous pouvez utiliser une expression contenant plusieurs champs.</p>
HEADER <i>texte_en-tête</i> optionnel	<p>Texte à insérer en haut de chaque page d'un rapport.</p> <p><i>texte_en-tête</i> doit être indiqué sous forme de chaîne entre guillemets. La valeur remplace la variable système Analytics HEADER.</p>
FOOTER <i>texte_pied_page</i> optionnel	<p>Texte à insérer en bas de chaque page d'un rapport.</p> <p><i>texte_pied_page</i> doit être indiqué sous forme de chaîne entre guillemets. La valeur remplace la variable système Analytics FOOTER.</p>
APPEND optionnel	<p>Ajoute la sortie de commande à la fin d'un fichier existant au lieu de remplacer ce fichier.</p> <p><b>Remarque</b></p> <p>Vous devez vous assurer que la structure de la sortie de la commande et du fichier existant est identique :</p> <ul style="list-style-type: none"> <li>• les mêmes champs</li> <li>• le même ordre des champs</li> <li>• les champs correspondants ont la même longueur</li> <li>• les champs correspondants ont le même type de données</li> </ul> <p>Analytics ajoute la sortie à un fichier existant sans tenir compte de sa structure, ce qui peut désordonner les données si la structure de la sortie et du fichier existant ne correspond pas.</p>
LOCAL optionnel	<p>Enregistre le fichier de sortie au même emplacement que le projet Analytics.</p> <p><b>Remarque</b></p> <p>Applicable uniquement lorsque vous exécutez la commande sur une table de serveur avec un fichier de sortie qui est une table Analytics.</p>
STATISTICS optionnel	<p><b>Remarque</b></p> <p>Ne peut pas être utilisée sauf si SUBTOTAL est également spécifié.</p> <p>Calcule les valeurs moyenne, minimale et maximale pour tous les champs SUBTOTAL.</p>

## Exemples

### Classer par antériorité des factures avec des montants sous-totalisés

Vous souhaitez classer par antériorité une table Comptes clients en fonction du champ **Date\_facture** et sous-totaliser le champ **Montant\_facture**.

Les factures sont regroupées par périodes de 30 jours :

- à compter de la date limite jusqu'à 29 jours avant
- à compter des 30 jours précédents jusqu'à 59 jours précédents
- etc.

Les résultats incluent le montant total des factures en attente pour chaque période :

```
OPEN Comptes_clients  
AGE ON Date_Facture CUTOFF 20141231 INTERVAL 0;30;60;90;120;10000 SUBTOTAL Montant_  
Facture TO SCREEN
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Périodes de classement par antériorité

La commande AGE sert à regrouper les enregistrements en périodes d'antériorité à partir des valeurs dans un champ date ou DateHeure. Les résultats de sortie contiennent un seul enregistrement pour chaque période, avec le compte du nombre d'enregistrements de la table source inclus dans chaque période.

## Mesure des intervalles

Les périodes d'antériorité sont basées sur des intervalles de date (c'est-à-dire le nombre de jours) mesurés à partir de la date système actuelle en remontant dans le temps ou à partir d'une date de référence que vous spécifiez, par exemple, une date de fin de période fiscale.

## Périodes futures

Vous pouvez créer des périodes d'antériorité plus récentes que la date de référence en saisissant des valeurs négatives pour les intervalles de date. Par exemple, ce qui suit crée des périodes d'antériorité à partir de la date limite et jusqu'à la date limite :

```
INTERVAL -60;-30;0;30;60;90
```

Cette approche crée un profil de date de tous les enregistrements d'une table utilisant différents points dans le temps.

## Cas d'utilisations courantes

Les utilisations courantes du classement par antériorité comprennent l'analyse des tendances de vente, l'observation des volumes de transactions et le regroupement des factures en fonction du nombre de jours échus.

Analytics crée automatiquement une ou deux périodes d'antériorité pour les dates situées en dehors des périodes d'antériorité spécifiées, en supposant que vous n'utilisez pas la commande SUPPRESS.



# Commande APPEND

Associe des enregistrements provenant de plusieurs tables Analytics en les ajoutant à une table Analytics.

## Syntaxe

```
APPEND table_1; table_2; <...n> TO nom_table <COMMONFIELDS> <OPEN> <ASCHAR>
<ALLCHAR>
```

## Paramètres

Nom	Description
<i>table_1</i> ; <i>table_2</i> ; <...n>	<p>Les tables à concaténer.</p> <p>Les enregistrements de chaque table sont concaténés dans l'ordre dans lequel vous avez indiqué les tables. La table de sortie contient les enregistrements de la <i>table_1</i>, suivis des enregistrements de la <i>table_2</i> et ainsi de suite.</p> <p>Les tables source peuvent avoir des structures d'enregistrement identiques ou différentes et peuvent être triées ou non.</p>
TO <i>nom_table</i>	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li>◦ <b><i>nom_table</i></b> : enregistre les résultats dans une table Analytics</li> </ul> <p>Indiquez <i>nom_table</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple : TO "Sortie.FIL"</p> <p>Par défaut, le fichier de données de table (.FIL) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.FIL"</li> <li>• TO "Résultats\Sortie.FIL"</li> </ul> <p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alpha-numériques, sans l'extension .FIL. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>
COMMONFIELDS optionnel	<p>Seuls les champs communs à toutes les tables en cours de concaténation sont inclus dans la table de sortie.</p> <p>Si COMMONFIELDS est omis, tous les champs de toutes les tables figurent dans la table de sortie. Des valeurs vides apparaissent dans la table de sortie là où aucun champ n'existe dans les tables source.</p>

Nom	Description
	<p><b>Astuce</b></p> <p>Pour voir les schémas et captures d'écran illustrant les deux options, reportez-vous à la rubrique <a href="#">Ajout de tables</a>.</p> <p><b>Remarque</b></p> <p>La commande APPEND ne prend pas en charge la concaténation de champs calculés. Pour plus d'informations, consultez la section "Les champs calculés ne sont pas pris en charge" Page 78.</p> <p><b>Qu'est-ce qui rend un champ « commun » ?</b></p> <p>Pour que les champs soient considérés comme communs, ils doivent :</p> <ul style="list-style-type: none"> <li>◦ être présents dans chaque table source</li> <li>◦ avoir un nom physique identique</li> <li>◦ appartenir à la même catégorie de données : <ul style="list-style-type: none"> <li>• Caractère</li> <li>• Numérique</li> <li>• DateHeure</li> <li>• Logique</li> </ul> </li> </ul> <p><b>Nom identique, catégorie de données différente</b></p> <p>Si deux champs ont un nom identique mais appartiennent à différentes catégories de données, un message d'erreur apparaît et la commande APPEND n'est pas exécutée.</p> <p>Le message d'erreur contient tous les conflits liés aux catégories de données dans l'ensemble des tables spécifiées par APPEND. Le message est enregistré dans la trace des commandes.</p> <p><b>Remarque</b></p> <p>Vous pouvez éviter cette situation en utilisant ASCHAR ou ALLCHAR pour harmoniser les catégories de données.</p>
OPEN optionnel	Ouvre la table créée par la commande après l'exécution de la commande. Valide uniquement lorsque la commande crée une table de sortie.
ASCHAR optionnel	<p>Harmonise les champs ayant des noms identiques mais des catégories de données différentes en convertissant des champs non caractère dans la catégorie de données caractère.</p> <p>Par exemple, vous concaténez deux tables dans lesquelles le champ ID_Employé présente des données de type caractère dans une table et des données numériques dans l'autre. Le champ numérique ID_Employé est converti en données de type caractère et les deux champs sont concaténés sans erreur.</p> <p>ASCHAR est ignoré si ALLCHAR est également spécifié.</p>
ALLCHAR optionnel	<p>Convertit dans la catégorie de données de type caractère tous les champs qui ne sont pas de type caractère dans toutes les tables en train d'être concaténées.</p> <p>Cette conversion globale en données de type caractère vérifie que tous les champs ayant un nom identique sont concaténés sans erreur.</p>

Nom	Description
	<p><b>Remarque</b></p> <p>Après la concaténation, vous pouvez modifier la catégorie de données d'un champ concaténé dans son ensemble si celle-ci est adaptée aux données contenues dans le champ.</p>

## Exemples

### Concaténer trois tables de transactions mensuelles

L'exemple ci-dessous concatène trois tables de transactions mensuelles et sort une table de transactions trimestrielle incluant uniquement tous les champs des trois tables source :

```
APPEND Trans_Jan; Trans_Fév; Trans_Mar TO Trans_Q1
```

### Concaténer trois tables employés et inclure uniquement les champs communs

L'exemple ci-dessous ajoute trois tables Employés des divisions et sort une table maîtresse employés incluant uniquement les champs communs des trois tables source :

```
APPEND Employés_central; Employés_est; Employés_ouest TO Employés_maîtresse  
COMMONFIELDS
```

### Concaténer trois tables employés et harmoniser les champs comportant différentes catégories de données

Les exemples ci-dessous concatènent trois tables employés de division dans lesquelles certains champs portant des noms identiques utilisent différentes catégories de données.

Le premier exemple convertit des champs non caractères dans la catégorie de données caractère uniquement si cela est nécessaire pour des questions d'harmonisation :

```
APPEND Employés_central; Employés_est; Employés_ouest TO Employés_maîtresse ASCHAR
```

Le deuxième exemple convertit tous les champs non caractères dans la catégorie de données caractère si cela est nécessaire pour des questions d'harmonisation ou non :

```
APPEND Employés_central; Employés_est; Employés_ouest TO Employés_maîtresse ASCHAR
```

# Remarques

## Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Fonctionnement

La commande APPEND associe les enregistrements provenant de plusieurs tables en les concaténant et en créant une nouvelle table. Par « concaténer », on entend ajouter un groupe d'enregistrements à la suite d'un autre groupe d'enregistrements.

Les champs des tables source ayant des noms physiques et des catégories de données identiques sont ajoutés directement les uns aux autres.

Les champs ayant des noms physiques uniques dans toutes les tables source sont ajoutés à la table de sortie, mais ils ne sont pas directement ajoutés aux autres champs.

## Astuce

Si vous souhaitez concaténer directement des champs ayant des noms incohérents, standardisez les noms physiques des champs dans les formats de table avant de les concaténer. (Cela suppose que les champs appartiennent à la même catégorie de données ou que vous utilisez ASCHAR ou ALLCHAR pour harmoniser les catégories de données des champs.)

## Quand utiliser la commande APPEND

Utilisez APPEND lorsque vous souhaitez combiner des données provenant de plusieurs tables avec une structure identique ou similaire. Par exemple, APPEND est un bon choix pour combiner des tables mensuelles ou trimestrielles dans une table annuelle.

## Astuce

Une seule exécution de la commande APPEND peut remplacer plusieurs exécutions de la commande EXTRACT avec l'option APPEND.

## N'est pas un substitut à JOIN ou DEFINE RELATION

En général, APPEND n'est pas un substitut aux commandes JOIN ou DEFINE RELATION car elle ne vous permet pas d'inclure ou d'exclure des enregistrements selon des valeurs avec ou sans correspondance dans un champ clé commun. Avec APPEND, tous les enregistrements de chaque table source sont inclus dans la table de sortie.

## Concaténer des tables complètement différentes

Vous pouvez concaténer des tables complètement différentes, c'est-à-dire plusieurs tables qui n'ont aucun champ en commun. Bien qu'il ne s'agisse pas de la principale utilisation prévue pour la commande

APPEND, dans certains cas, concaténer des tables complètement différentes répond à un objectif analytique.

## Concaténer des champs DateHeure

Pour qu'au moins deux champs DateHeure soient concaténés, les conditions suivantes doivent être remplies :

- des noms physiques identiques
- une catégorie de données identique (DateHeure)
- des sous-types de données identiques (date, DateHeure ou heure)
- utilisation identique d'un indicateur de fuseau horaire (utilisé ou inutilisé par tous les champs concaténés)

Si deux champs DateHeure ont un nom identique mais qu'ils ne répondent pas à l'une des autres conditions, un message d'erreur apparaît et la commande APPEND n'est pas exécutée.

Le message d'erreur contient toutes les conditions non remplies dans l'ensemble des tables spécifiées par APPEND. Le message est enregistré dans la trace des commandes.

### Remarque

Vous pouvez harmoniser des champs DateHeure non similaires en les convertissant dans la catégorie de données caractère, puis les concaténer. Cette méthode vous permet de combiner les données dans une seule table. Toutefois, selon la nature des données source, il se peut que vous ne puissiez pas reconverter ensuite les données converties en données DateHeure.

## Harmonisation automatique

Dans certains cas, la commande APPEND harmonise automatiquement les champs pour les concaténer :

Catégorie de données des champs	Harmonisation effectuée
Caractère	<ul style="list-style-type: none"> <li>○ Les longueurs de champs différentes sont harmonisées.</li> <li>○ Les types de données caractère (par ex., Custom, PCASCII et EBCDIC) qui diffèrent sont harmonisés en convertissant les champs dans le type de données ASCII ou UNICODE.</li> </ul>
Numérique	<ul style="list-style-type: none"> <li>○ Les longueurs de champs différentes sont harmonisées. Les champs sont convertis dans le type de données ACL.</li> <li>○ Les quantités de décimales définies qui diffèrent sont harmonisées. Le nombre de décimales est standardisé par rapport au plus grand nombre de décimales ; des zéros non significatifs sont ajoutés aux valeurs numériques le cas échéant. Les champs sont convertis dans le type de données ACL.</li> <li>○ Les types de données numériques (par ex., Print, Float, EBCDIC et Micro) qui diffèrent sont harmonisés en convertissant les champs dans le type de données ACL.</li> </ul>
DateHeure	<ul style="list-style-type: none"> <li>○ Dans les données source, les formats date, DateHeure et heure qui diffèrent sont harmonisés en convertissant les champs aux formats Analytics par défaut : <ul style="list-style-type: none"> <li>• AAAAMMJJ</li> <li>• AAAAMMJJ hh:mm:ss</li> </ul> </li> </ul>

Catégorie de données des champs	Harmonisation effectuée
	<ul style="list-style-type: none"> <li>• hh:mm:ss</li> </ul>

## Situations où l'harmonisation automatique n'est pas effectuée

Analytics n'harmonise pas les champs automatiquement dans les situations suivantes. Un message d'erreur s'affiche et l'opération de concaténation n'est pas exécutée.

- Deux champs ayant un nom identique appartiennent à des catégories de données différentes.
- Deux champs DateHeure ayant un nom identique appartiennent à des sous-types de DateHeure différents (date, DateHeure ou heure).
- Deux champs DateHeure ayant un nom identique utilisent de manière incohérente l'indicateur de fuseau horaire.

### Remarque

Vous trouverez ci-dessus des explications concernant l'harmonisation - spécifiée par l'utilisateur - des champs ayant des noms identiques mais des catégories de données différentes. Pour plus d'informations, consultez la section "ASCHAR" Page 74 et "ALLCHAR" Page 74.

## Les champs calculés ne sont pas pris en charge

La commande APPEND ne prend pas en charge la concaténation de champs calculés. Lorsque vous concaténez des tables, les champs calculés des tables source sont exclus automatiquement de la table de sortie.

Si un champ calculé dans une table source porte le même nom qu'un champ physique dans une autre table source, un message d'erreur apparaît et la commande APPEND n'est pas exécutée.

### Astuce

Vous pouvez concaténer un champ calculé en commençant par l'extraire pour convertir ce champ en un champ physique. (Pour plus d'informations, consultez "Commande EXTRACT" Page 206.) Vous utilisez alors la table extraite dans l'opération de concaténation.

Une autre méthode consiste à recréer le champ calculé dans la table de sortie concaténée.

## Les champs Notes d'enregistrement ne sont pas pris en charge

La commande APPEND ne prend pas en charge la concaténation de champs Notes d'enregistrement. Lorsque vous concaténez des tables, les champs Notes d'enregistrement des tables source sont exclus automatiquement de la table de sortie.

Si un champ Note d'enregistrement dans une table source porte le même nom qu'un champ physique dans une autre table source, un message d'erreur apparaît et la commande APPEND n'est pas exécutée.

Un champ Note d'enregistrement est généré automatiquement par Analytics lorsque vous ajoutez une note à un enregistrement.

## Long. de l'enreg.

Si vous incluez tous les champs de toutes les tables source lors de la concaténation, la longueur de l'enregistrement dans la table de sortie peut être plus grande que l'enregistrement le plus long dans les tables source.

Un message d'erreur apparaît si la longueur de l'enregistrement de sortie dépasse la longueur maximale autorisée par Analytics qui est de 32 Ko.

## Concaténation et décimales

Un comportement spécifique gère la concaténation des champs numériques incluant des décimales.

### Le paramètre Decimal

La commande APPEND utilise le nombre de décimales défini dans le paramètre **Dec** dans la définition de champ du format de table.

#### Remarque

Il se peut que le paramètre **Dec** soit différent du nombre réel de décimales dans les données source. Les décimales dépassant le paramètre **Dec** ne sont pas définies et sont arrondies dans les calculs.

### Paramètres Decimal incohérents

Si les champs numériques concaténés présentent des paramètres **Dec** incohérents, les champs sont convertis dans le type de données ACL et ils sont harmonisés automatiquement sur le paramètre **Dec** le plus long.

Les décimales des fichiers de données source dépassant le paramètre **Dec** le plus long sont **exclues** de la table de sortie générée par APPEND.

### Paramètre Decimal cohérent

Si les champs numériques ajoutés ont un paramètre **Dec** cohérent, aucune conversion ni harmonisation des types de données n'a lieu.

Les décimales des fichiers de données source dépassant le paramètre **Dec** sont **incluses** dans la table de sortie générée par APPEND.

# Tri

Les éventuels ordres de tri existants dans les tables source sont conservés séparément dans les jeux d'enregistrements respectifs de la table de sortie.

Même si les enregistrements de toutes les tables source sont triés, la table de sortie obtenue est considérée comme non triée car les enregistrements source sont concaténés en tant que groupe sans tenir compte d'un quelconque ordre de tri existant dans les autres tables source.

Par exemple, si vous ajoutez des tables mensuelles ou trimestrielles pour créer une table annuelle, tout tri interne sur les données mensuelles ou trimestrielles est conservé. Si nécessaire, vous pouvez trier la table de sortie après l'exécution de l'opération de concaténation.

## Fonctionnement de l'ordre des champs

### Champs communs

Les champs communs des tables source ne doivent pas forcément se trouver dans le même ordre pour être concaténés.

Par exemple, ces champs sont concaténés correctement même s'ils sont dans un ordre différent :

Table	Champs
<i>table_1</i>	Nom   Prénom   Deuxième_prénom
<i>table_2</i>	Prénom   Deuxième_prénom   Nom

La première table indiquée dans la commande APPEND détermine l'ordre des champs dans la table de sortie. Par conséquent, dans l'exemple ci-dessous, l'ordre dans la table de sortie est :

- Nom | Prénom | Deuxième\_prénom

### Champs non communs

Les champs non communs des tables source s'affichent dans la table de sortie dans leur ordre d'apparition dans le groupe sélectionné des tables source.

Par exemple, lors de la concaténation de ces deux tables :

Table	Champs
<i>table_1</i>	Titre   Nom   Prénom   Deuxième_prénom
<i>table_2</i>	Prénom   Deuxième_prénom   Nom   Date_de_naissance

l'ordre dans la table de sortie est :

- Titre | Nom | Prénom | Deuxième\_prénom | Date\_de\_naissance



## Titre de la colonne

Les titres de colonnes des tables source apparaissent dans la table de sortie. Si plusieurs tables source ont un titre de colonne différent pour le même champ, le titre de la première table sélectionnée est prioritaire.

# Commande ASSIGN

Crée une variable et lui affecte une valeur.

## Syntaxe

```
ASSIGN nom_variable = valeur <IF test>
```

### Astuce

Vous pouvez ignorer le mot-clé ASSIGN, car Analytics interprète automatiquement la syntaxe suivante comme une opération d'attribution :

```
nom_variable = valeur
```

## Paramètres

Nom	Description
<i>nom_variable</i>	<p>Nom de la variable à laquelle la valeur doit être attribuée. Si la variable n'existe pas, elle est créée. Si la variable existe déjà, elle est mise à jour avec la nouvelle valeur.</p> <p>N'utilisez pas de caractères n'existant pas en anglais, comme « é », dans les noms des variables. Les noms de variables contenant des caractères n'existant pas en anglais provoqueront l'échec des scripts.</p> <p><b>Remarque</b></p> <p>La longueur des noms de variables est limitée à 31 caractères alpha-numériques. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>
<i>valeur</i>	Valeur à attribuer à la variable. Si une nouvelle variable est créée, le type de variable est basé sur le type de données dans <i>valeur</i> .
IF <i>test</i> Facultatif	Une expression conditionnelle qui doit être vraie pour créer la variable ou attribuer la valeur à la variable.

## Exemples

### Attribution d'une valeur à une variable

Vous attribuez la valeur du champ **Montant** de l'enregistrement actuel à une variable appelée *v\_montant\_actuel*. *v\_montant\_actuel* étant une variable, sa valeur n'est pas modifiée, sauf si une autre commande ASSIGN la modifie explicitement :

```
ASSIGN v_montant_actuel = Montant
```

## Attribution conditionnelle d'une valeur à une variable

Vous souhaitez actualiser la valeur d'une variable nommée *v\_quantité* sur 1, mais uniquement si la valeur d'une autre variable appelée *v\_compteur* est inférieure à 10.

Si *v\_compteur* est supérieur ou égal à 10, aucune valeur n'est attribuée et la valeur de *v\_quantité* reste inchangée.

Remarquez que le mot-clé ASSIGN facultatif est ignoré :

```
v_quantité = 1 IF v_compteur < 10
```

## Remarques

### Durée des variables

Les variables dont le nom n'est pas précédé d'un trait de soulignement sont conservées uniquement pour la durée de la session Analytics active.

Pour enregistrer de manière définitive la variable avec un projet Analytics, faites précéder son nom d'un trait de soulignement :

```
ASSIGN valeur = _nom_variable
```

## Réattribution des variables utilisées dans un champ calculé ou dans GROUP

Si vous affectez une valeur à une variable existante dans les situations suivantes, la nouvelle valeur est affectée mais la longueur et le nombre de décimales de la valeur précédente sont conservés :

- variables utilisées dans des champs calculés
- variables réaffectées à l'intérieur d'un GROUP

La longueur de la nouvelle valeur est tronquée ou complétée, et les décimales sont ajustées si nécessaire.

Si vous réaffectez une variable dans tout autre contexte, la valeur précédente ainsi que sa longueur et ses spécifications décimales sont écrasées.

## Variables créées via des commandes Analytics

Lorsque certaines commandes sont exécutées, via la saisie d'informations dans les boîtes de dialogue Analytics ou via l'exécution de scripts, des variables système sont automatiquement créées par Analytics. Vous pouvez utiliser ces variables, et les valeurs qu'elles contiennent, lors de l'exécution des commandes Analytics subséquentes.

La valeur d'une variable système est remplacée par une valeur mise à jour si vous exécutez à nouveau la même commande.

Pour plus d'informations, consultez la section "Variables créées via des commandes Analytics" Page 980.

# Commande BENFORD

Compte le nombre d'occurrences de chaque chiffre de tête (1-9) ou combinaison de chiffres de tête dans un champ, et compare le compte réel au compte probable. Le compte probable est obtenu en utilisant la formule Benford.

## Syntaxe

```
BENFORD <ON> champ_numérique <LEADING n> <IF test> <BOUNDS> <TO {SCREEN|nom_table|GRAPH|PRINT}> <HEADER texte_en-tête> <FOOTER texte_pied_page> <WHILE test> <FIRST plage|NEXT plage> <APPEND> <OPEN> <LOCAL>
```

## Paramètres

Nom	Description
ON <i>champ_numérique</i>	<p>Champ numérique à analyser.</p> <p><b>Remarque</b></p> <p>Sélectionnez un champ contenant des « nombres apparaissant naturellement », par exemple, des montants de transactions. L'analyse de Benford ne convient pas aux données numériques limitées d'une manière ou d'une autre.</p> <p>Pour plus d'informations, consultez la section "Quelles données puis-je tester à l'aide d'une analyse de Benford ?" Page 87</p>
LEADING <i>n</i> optionnel	<p>Nombre de chiffres de tête à analyser. La valeur de <i>n</i> doit être comprise entre 1 et 6.</p> <p>Si LEADING est ignoré, c'est la valeur par défaut de 1 qui est utilisée.</p>
IF <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.</p> <p><b>Remarque</b></p> <p>La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p>
BOUNDS optionnel	<p>Inclut les valeurs calculées supérieures et inférieures dans les résultats de sortie.</p> <p>Si le compte réel de plusieurs chiffres ou combinaisons de chiffres dans les résultats de sortie dépasse l'une de ces limites, il se peut que les données aient été manipulées. Elles doivent donc faire l'objet d'une étude plus approfondie.</p>
TO SCREEN   <i>nom_table</i>   GRAPH   PRINT	Emplacement vers lequel envoyer les résultats de la commande :

Nom	Description
optionnel	<ul style="list-style-type: none"> <li>◦ <b>SCREEN</b> : affiche les résultats dans la zone d'affichage d'Analytics.</li> <li>◦ <b>nom_table</b> : enregistre les résultats dans une table Analytics</li> </ul> <p>Indiquez <i>nom_table</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple : TO "Sortie.FIL"</p> <p>Par défaut, le fichier de données de table (.FIL) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.FIL"</li> <li>• TO "Résultats\Sortie.FIL"</li> </ul> <p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alphanumériques, sans l'extension .FIL. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p> <ul style="list-style-type: none"> <li>◦ <b>GRAPH</b> : affiche les résultats dans un graphique dans la zone d'affichage d'Analytics</li> <li>◦ <b>PRINT</b> : envoie les résultats vers l'imprimante par défaut</li> </ul>
HEADER <i>texte_en-tête</i> optionnel	Texte à insérer en haut de chaque page d'un rapport. <i>texte_en-tête</i> doit être indiqué sous forme de chaîne entre guillemets. La valeur remplace la variable système Analytics HEADER.
FOOTER <i>texte_pied_page</i> optionnel	Texte à insérer en bas de chaque page d'un rapport. <i>texte_pied_page</i> doit être indiqué sous forme de chaîne entre guillemets. La valeur remplace la variable système Analytics FOOTER.
WHILE <i>test</i> optionnel	Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.  <b>Remarque</b> Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.
FIRST <i>plage</i>   NEXT <i>plage</i> optionnel	Le nombre d'enregistrements à traiter : <ul style="list-style-type: none"> <li>◦ <b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li>◦ <b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> </ul> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter.</p> <p>Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p>
APPEND optionnel	Ajoute la sortie de commande à la fin d'un fichier existant au lieu de remplacer ce fichier.

Nom	Description
	<p><b>Remarque</b></p> <p>Vous devez vous assurer que la structure de la sortie de la commande et du fichier existant est identique :</p> <ul style="list-style-type: none"> <li>• les mêmes champs</li> <li>• le même ordre des champs</li> <li>• les champs correspondants ont la même longueur</li> <li>• les champs correspondants ont le même type de données</li> </ul> <p>Analytics ajoute la sortie à un fichier existant sans tenir compte de sa structure, ce qui peut désordonner les données si la structure de la sortie et du fichier existant ne correspond pas.</p>
OPEN optionnel	Ouvre la table créée par la commande après l'exécution de la commande. Valide uniquement lorsque la commande crée une table de sortie.
LOCAL optionnel	Enregistre le fichier de sortie au même emplacement que le projet Analytics. <p><b>Remarque</b></p> <p>Applicable uniquement lorsque vous exécutez la commande sur une table de serveur avec un fichier de sortie qui est une table Analytics.</p>

## Exemples

### Sortie graphique des résultats

Vous exécutez la commande BENFORD vis-à-vis du champ **Montant** et vous sortez les résultats sur un graphique :

```
BENFORD ON Montant LEADING 2 BOUNDS TO GRAPH
```

## Remarques

### Quelles données puis-je tester à l'aide d'une analyse de Benford ?

Vous devez uniquement utiliser une analyse de Benford pour tester des données numériques composées de « nombres apparaissant naturellement », tels que les montants comptables, les montants des transactions, les dépenses ou les numéros d'adresse. L'analyse de Benford ne convient pas aux données numériques limitées d'une manière ou d'une autre.

Suivez ces indications pour identifier les données numériques qui conviennent à l'analyse de Benford :

- **Taille du jeu de données** : Le jeu de données doit être suffisamment grand pour soutenir une répartition valide. Il se peut que les résultats d'une analyse de Benford ne soient pas très fiables pour des ensembles de moins de 500 enregistrements.
- **Condition du chiffre de tête** : Tous les chiffres compris entre 1 et 9 doivent avoir la possibilité d'apparaître comme chiffre de tête.
- **Condition de la combinaison des chiffres de tête** : Tous les chiffres compris entre 0 à 9 doivent avoir la possibilité d'apparaître comme deuxième chiffre de tête et comme un chiffre supplémentaire en cours d'analyse.
- **Données limitées** : Les données numériques qui sont affectées ou générées selon un modèle prédéterminé ne conviennent pas à l'analyse de Benford. Par exemple, n'utilisez pas Benford pour analyser les éléments suivants :
  - les numéros de chèques ou numéros de facture séquentiels
  - les numéros de sécurité sociale ou les numéros de téléphone qui correspondent à un modèle spécifique
  - tout modèle de numérotation présentant une plage empêchant l'apparition de certains chiffres
- **Nombres aléatoires** : Les nombres générés par un générateur de nombres aléatoires ne conviennent pas à l'analyse de Benford.



# Commande CALCULATE

Calcule la valeur d'une ou plusieurs expressions.

## Syntaxe

```
CALCULATE expression <AS libellé_résultat> <;...n>
```

## Paramètres

Nom	Description
<i>expression</i>	<p>Expression à calculer.</p> <p>L'expression peut être de l'un des quatre types suivants :</p> <ul style="list-style-type: none"> <li>○ caractère</li> <li>○ numérique</li> <li>○ DateHeure</li> <li>○ logique</li> </ul> <p>Séparez plusieurs expressions par un point-virgule :</p> <pre>CALCULATE 4,7 * 18,5; 1 + 2; "a" + "b"</pre>
<i>AS libellé_résultat</i> optionnel	<p>Nom du résultat affiché à l'écran et enregistré dans la trace de la commande Analytics.</p> <p><i>libellé_résultat</i> doit être une chaîne entre guillemets ou une expression de caractères valide.</p> <p>Si ce paramètre n'est pas spécifié, l'expression calculée est utilisée comme nom du résultat.</p>

## Exemples

### Exécution d'un calcul simple

Vous utilisez CALCULATE pour multiplier 4,70 par 18,50, ce qui renvoie le résultat 86,95 :

```
CALCULATE 4,70 * 18,50
```

### Nommer les résultats d'un calcul

Vous utilisez CALCULATE pour extraire la marge brute de l'enregistrement actuellement sélectionné à

l'aide des champs définis précédemment pour le prix de vente et le coût unitaire :

```
CALCULATE Prix_vente - Coût_unitaire AS "Marge"
```

Le résultat est identifié à l'écran et dans la trace par « Marge ».

## Remarques

### Fonctionnement

CALCULATE offre les fonctionnalités d'une calculatrice et permet d'accéder aux fonctions Analytics, aux variables et aux données de l'enregistrement actuellement sélectionné.

### Sortie de la commande

Selon l'endroit où vous exécutez CALCULATE, les résultats sortent à différents emplacements :

- **À partir de la ligne de commande** : le résultat s'affiche à l'écran
- **À partir d'un script** : le résultat est enregistré dans la trace

La valeur *libellé\_résultat* n'est pas une variable que vous pouvez utiliser dans un script. Elle permet uniquement d'identifier le calcul sur l'écran ou dans la trace.

### Nombre de décimales dans la sortie

Dans un calcul numérique, le résultat présente autant de décimales que l'expression qui en contient le plus grand nombre.

Renvoie 1 :

```
CALCULATE 365/52/7
```

Renvoie 1,0027 :

```
CALCULATE 365,0000/52/7
```

### Utilisation des saisies des tables

Si l'expression contient une valeur de champ, la table à laquelle le champ appartient doit être ouverte. Les commandes FIND, SEEK ou LOCATE permettent d'accéder à l'enregistrement qui doit être analysé par CALCULATE.

# Commande CLASSIFY

Regroupe les enregistrements en fonction des valeurs identiques d'un champ caractère ou numérique. Compte le nombre d'enregistrements dans chaque groupe et calcule également le sous-total des champs numériques spécifiés pour chaque groupe.

## Syntaxe

```
CLASSIFY <ON> champ_clé <SUBTOTAL champ_numérique <...n>|SUBTOTAL ALL>
<INTERVALS nombre> <SUPPRESS> <TO {SCREEN|nom_table|GRAPH|PRINT}> <IF test>
<WHILE test> <FIRST plage|NEXT plage> <HEADER texte_en-tête> <FOOTER texte_pied_page>
<KEY champ_fractionnement> <OPEN> <APPEND> <LOCAL> <STATISTICS>
```

## Paramètres

Nom	Description
ON <i>champ_clé</i>	<p>Champ caractère ou numérique à classer.</p> <p>La longueur maximale des champs clés est de 2 048 caractères.</p> <p>Si vous souhaitez classer une table à l'aide d'un champ clé contenant plus de 2 048 caractères, utilisez la commande SUMMARIZE, opération qui ne limite pas la longueur du champ clé. Cela ne limite pas la longueur du champ clé.</p>
SUBTOTAL <i>champ_numérique</i> <...n>   SUBTOTAL ALL optionnel	<p>Un ou plusieurs champs ou expressions numériques à sous-totaliser pour chaque groupe.</p> <p>Les champs multiples doivent être séparés par des espaces. Indiquez ALL pour calculer le sous-total de tous les champs numériques de la table.</p>
INTERVALS <i>nombre</i> optionnel	<p>Nombre maximal de groupes dans le résultat de sortie.</p> <p>Si le nombre de jeux de valeurs identiques dans le champ en cours de classement dépasse la valeur maximale spécifiée, les jeux sont utilisés à partir du haut de la colonne.</p> <p>Les jeux dépassant la valeur maximale sont regroupés dans un groupe intitulé « OTHER ».</p> <p>Si INTERVALS est ignoré, un groupe est créé pour chaque jeu de valeurs identiques dans le champ en cours de classement.</p> <p><b>Remarque</b></p> <p>Ce paramètre n'est pas disponible dans l'interface utilisateur Analytics et peut uniquement être utilisé en tant que partie intégrante d'une syntaxe ACLScript dans un script ou la ligne de commande.</p>

Nom	Description
SUPPRESS optionnel	<p><b>Remarque</b></p> <p>Ne peut pas être utilisé sauf si INTERVALS est également spécifié. SUPPRESS n'est pas disponible dans l'interface utilisateur Analytics et peut uniquement être utilisé en tant que partie intégrante d'une syntaxe ACLScript dans un script ou la ligne de commande.</p> <p>Exclut les jeux de valeurs identiques dépassant le maximum spécifié par INTERVALS de la sortie de commande.</p>
TO SCREEN   <i>nom_table</i>   GRAPH   PRINT	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li>◦ <b>SCREEN</b> : affiche les résultats dans la zone d'affichage d'Analytics.</li> <li>◦ <b><i>nom_table</i></b> : enregistre les résultats dans une table Analytics</li> </ul> <p>Indiquez <i>nom_table</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple : TO "Sortie.FIL"</p> <p>Par défaut, le fichier de données de table (.FIL) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.FIL"</li> <li>• TO "Résultats\Sortie.FIL"</li> </ul> <p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alpha-numériques, sans l'extension .FIL. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p> <ul style="list-style-type: none"> <li>◦ <b>GRAPH</b> : affiche les résultats dans un graphique dans la zone d'affichage d'Analytics</li> <li>◦ <b>PRINT</b> : envoie les résultats vers l'imprimante par défaut</li> </ul>
IF <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.</p> <p><b>Remarque</b></p> <p>La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p>
WHILE <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.</p> <p><b>Remarque</b></p> <p>Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.</p>
FIRST <i>plage</i>   NEXT <i>plage</i> optionnel	<p>Le nombre d'enregistrements à traiter :</p> <ul style="list-style-type: none"> <li>◦ <b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li>◦ <b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné</li> </ul>

Nom	Description
	<p>tionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</p> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter.</p> <p>Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p>
HEADER <i>texte_en-tête</i> optionnel	<p>Texte à insérer en haut de chaque page d'un rapport.</p> <p><i>texte_en-tête</i> doit être indiqué sous forme de chaîne entre guillemets. La valeur remplace la variable système Analytics HEADER.</p>
FOOTER <i>texte_pied_page</i> optionnel	<p>Texte à insérer en bas de chaque page d'un rapport.</p> <p><i>texte_pied_page</i> doit être indiqué sous forme de chaîne entre guillemets. La valeur remplace la variable système Analytics FOOTER.</p>
KEY <i>champ_fractionnement</i> optionnel	<p>Champ ou expression regroupant les calculs de sous-totaux. Un sous-total est calculé chaque fois que la valeur de <i>champ_fractionnement</i> change.</p> <p><i>champ_fractionnement</i> doit être un champ ou une expression de type caractère. Vous ne pouvez indiquer qu'un seul champ, mais vous pouvez utiliser une expression contenant plusieurs champs.</p>
OPEN optionnel	<p>Ouvre la table créée par la commande après l'exécution de la commande. Valide uniquement lorsque la commande crée une table de sortie.</p>
APPEND optionnel	<p>Ajoute la sortie de commande à la fin d'un fichier existant au lieu de remplacer ce fichier.</p> <p><b>Remarque</b></p> <p>Vous devez vous assurer que la structure de la sortie de la commande et du fichier existant est identique :</p> <ul style="list-style-type: none"> <li>• les mêmes champs</li> <li>• le même ordre des champs</li> <li>• les champs correspondants ont la même longueur</li> <li>• les champs correspondants ont le même type de données</li> </ul> <p>Analytics ajoute la sortie à un fichier existant sans tenir compte de sa structure, ce qui peut désordonner les données si la structure de la sortie et du fichier existant ne correspond pas.</p>
LOCAL optionnel	<p>Enregistre le fichier de sortie au même emplacement que le projet Analytics.</p> <p><b>Remarque</b></p> <p>Applicable uniquement lorsque vous exécutez la commande sur une table de serveur avec un fichier de sortie qui est une table Analytics.</p>
STATISTICS optionnel	<p><b>Remarque</b></p> <p>Ne peut pas être utilisée sauf si SUBTOTAL est également spécifié.</p> <p>Calcule les valeurs moyenne, minimale et maximale pour tous les champs SUBTOTAL.</p>

# Exemples

## Montant total des transactions par client

Vous souhaitez classer une table Comptes clients en fonction du champ **Numéro\_client** et sous-totaliser le champ **Montant\_trans**. Les résultats de sortie sont regroupés par client et incluent le montant total des transactions pour chaque client :

```
OPEN Comptes_clients
CLASSIFY ON Numéro_Client SUBTOTAL Montant_Trans TO "Total_Clients.FIL"
```

## Montants total, moyen, minimum et maximum des transactions par client

Comme dans l'exemple précédent, vous classez une table Comptes clients en fonction du champ **Numéro\_client** et vous sous-totalisez le champ **Montant\_trans**.

Maintenant, vous incluez STATISTICS pour calculer les montants moyen, minimum et maximum des transactions de chaque client :

```
OPEN Comptes_clients
CLASSIFY ON Numéro_Client SUBTOTAL Montant_Trans TO "Stats_Clients.FIL" STATISTICS
```

## Montants de facture identiques

Vous devez identifier les montants des factures qui apparaissent plusieurs fois dans la table **Comptes\_clients**.

Pour ce faire, vous classez la table en fonction du champ **Montant\_facture**. Les résultats de la sortie sont regroupés par montant de facture, avec un compte associé que vous pouvez utiliser pour identifier des montants de facture qui apparaissent plusieurs fois :

```
OPEN Trans_Cf
CLASSIFY ON Montant_Facture TO "Montants_facture_groupés.FIL" OPEN
SET FILTER TO COUNT > 1
```

# Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Fonctionnement

CLASSIFY regroupe les enregistrements présentant la même valeur dans un champ caractère ou numérique.

La sortie contient un seul enregistrement pour chaque groupe, avec le compte du nombre d'enregistrements de la table source inclus dans le groupe.

## Tri et CLASSIFY

CLASSIFY peut traiter les données triées ou non triées. La sortie est automatiquement triée dans l'ordre croissant.

## Noms des champs des sous-totaux et des statistiques générés automatiquement

Si vous utilisez STATISTICS pour effectuer des calculs statistiques sur un ou plusieurs champs SUBTOTAL, et que vous produisez les résultats vers une table Analytics, les champs générés automatiquement par les paramètres ont les noms suivants :

Description du champ généré automatiquement	Nom du champ dans la table de sortie	Autre titre de la colonne (nom d'affichage) dans la table de sortie
Champ Sous-total	<i>nom du champ sous-totalisé dans la table source</i>	<b>Total</b> + <i>autre titre de la colonne sous-totalisé dans la table source</i>
Champ Moyenne	<b>a_</b> <i>nom du champ sous-totalisé dans la table source</i>	<b>Moyenne</b> + <i>autre titre de la colonne sous-totalisé dans la table source</i>
Champ Minimum	<b>m_</b> <i>nom du champ sous-totalisé dans la table source</i>	<b>Minimum</b> + <i>autre titre de la colonne sous-totalisé dans la table source</i>
Champ Maximum	<b>x_</b> <i>nom du champ sous-totalisé dans la table source</i>	<b>Maximum</b> + <i>autre titre de la colonne sous-totalisé dans la table source</i>

# Commande CLOSE

Ferme une table Analytics, un fichier d'index ou un fichier trace, ou arrête une session d'**enregistrement de script**.

## Syntaxe

```
CLOSE <nom_table|PRIMARY|SECONDARY|INDEX|LOG|LEARN>
```

## Paramètres

Nom	Description
<p><i>nom_table</i>  PRIMARY SECONDARY INDEX LOG LEARN optionnel</p>	<p>L'élément à fermer :</p> <ul style="list-style-type: none"> <li>○ <b>nom_table</b> : nom de la table Analytics à fermer</li> <li>○ <b>PRIMARY</b> : ferme la table Analytics principale</li> </ul> <p>L'utilisation de CLOSE sans paramètre entraîne aussi la fermeture de la table principale.</p> <ul style="list-style-type: none"> <li>○ <b>SECONDARY</b> : ferme la table Analytics secondaire</li> <li>○ <b>INDEX</b> : ferme l'index actuel appliqué à la table Analytics</li> <li>○ <b>LOG</b> : renvoie le fichier trace à la trace de commande par défaut, une fois la commande SET LOG utilisée pour indiquer un autre fichier trace</li> <li>○ <b>LEARN</b> : arrête une session d'<b>enregistrement de script</b> active et vous invite à enregistrer le fichier de script dans lequel la session a été enregistrée.</li> </ul> <p>LEARN peut être utilisé dans les scripts, mais il est conçu pour une utilisation dans la ligne de commande. L'<b>enregistrement de script</b> enregistre la syntaxe ACLScript pour les commandes qui sont exécutées à l'aide de boîtes de dialogue dans l'interface utilisateur Analytics.</p>

## Exemples

### Fermeture d'une table par nom

Vous souhaitez fermer une table nommée **Inventaire** :

```
CLOSE Inventaire
```



## Fermeture d'une table par type

Vous souhaitez fermer la table secondaire active :

```
CLOSE SECONDARY
```

## Restauration de la trace de commande Analytics par défaut

Vous souhaitez restaurer la trace de commande par défaut après avoir utilisé un fichier de trace distinct pour capturer la phase de vérification des données d'un script :

```
SET LOG TO "PhaseVérificationDonnées.log"  
COMMENT Exécuter commandes vérification données  
CLOSE LOG
```

# Remarques

## Quand utiliser CLOSE

En général, vous ne devez pas fermer les tables Analytics. La table Analytics active se ferme automatiquement lorsque vous ouvrez une autre table. La table principale se ferme automatiquement avant d'exécuter les commandes OPEN ou QUIT.

CLOSE ne peut pas être utilisée pour fermer un projet Analytics. Utilisez QUIT à la place.

## Champs et tables associés

Lorsque vous fermez une table principale ou secondaire, toutes les définitions de champ associées sont supprimées de la mémoire. Toute modification apportée au format de table est enregistrée avant la fermeture de la table.

Si vous avez défini des relations entre les tables dans un projet Analytics, la commande CLOSE ferme les tables principales et secondaires. Elle ferme aussi les tables associées.

# Commande CLUSTER

Regroupe les enregistrements en clusters selon des valeurs similaires dans un ou plusieurs champs numériques. Les clusters peuvent être unidimensionnels ou multidimensionnels.

## Syntaxe

```
CLUSTER ON champ_fclé <...n> KVALUE nombre_de_clusters ITERATIONS nombre_itérations
INITIALIZATIONS nombre_initialisations <SEED valeur_départ> <OTHER champ <...n>> TO nom_table
<IF test> <WHILE test> <FIRST plage|NEXT plage> OPEN {aucun_mot_clé|NOCENTER|NOSCALE}
```

## Paramètres

Nom	Description
ON <i>champ_clé</i> <...n>	Un ou plusieurs champs numériques à regrouper. Les champs multiples doivent être séparés par des espaces.
KVALUE <i>nombre_de_clusters</i>	Nombre de clusters générés dans les résultats de sortie.
ITERATIONS <i>nombre_itérations</i>	Nombre maximum de fois où le calcul des clusters est ré-effectué.
INITIALIZATIONS <i>nombre_initialisations</i>	Nombre de fois permettant de générer un premier ensemble de centroïdes aléatoires.
SEED <i>valeur_départ</i> optionnel	Valeur de départ à utiliser pour lancer le générateur de nombres aléatoires dans Analytics. Si vous omettez SEED, Analytics sélectionne aléatoirement la valeur de départ.
OTHER <i>champ</i> <...n> optionnel	Un ou plusieurs champs supplémentaires à inclure dans la sortie.  <div style="border-left: 2px solid #0056b3; padding-left: 10px; margin-left: 20px;"> <p><b>Remarque</b></p> <p>Les champs clés sont inclus automatiquement dans la table de sortie et ne doivent pas être spécifiés à l'aide de OTHER.</p> </div>
TO <i>nom_table</i>	Emplacement vers lequel envoyer les résultats de la commande : <ul style="list-style-type: none"> <li>◦ <b><i>nom_table</i></b> : enregistre les résultats dans une table Analytics Indiquez <i>nom_table</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple : TO "Sortie.FIL"</li> </ul> Par défaut, le fichier de données de table (.FIL) est enregistré dans le dossier conte-

Nom	Description
	<p>nant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.FIL"</li> <li>• TO "Résultats\Sortie.FIL"</li> </ul> <p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alpha-numériques, sans l'extension .FIL. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>
<p>IF <i>test</i> optionnel</p>	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.</p> <p><b>Remarque</b></p> <p>La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p>
<p>WHILE <i>test</i> optionnel</p>	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.</p> <p><b>Remarque</b></p> <p>Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.</p>
<p>FIRST <i>plage</i>   NEXT <i>plage</i> optionnel</p>	<p>Le nombre d'enregistrements à traiter :</p> <ul style="list-style-type: none"> <li>◦ <b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li>◦ <b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> </ul> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter.</p> <p>Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p>
<p>OPEN optionnel</p>	<p>Ouvre la table créée par la commande après l'exécution de la commande. Valide uniquement lorsque la commande crée une table de sortie.</p>
<p><i>aucun_mot_clé</i>   NOCENTER   NOSCALE</p>	<p>Méthode permettant de standardiser les valeurs numériques des champs clés.</p> <ul style="list-style-type: none"> <li>◦ <b><i>aucun_mot_clé</i></b> : centre les valeurs de champ clé aux alentours de zéro (0) et met à l'échelle les valeurs sur l'écart à l'unité lors du calcul des clusters</li> <li>◦ <b>NOCENTER</b> : met à l'échelle les valeurs de champ clé sur l'écart à l'unité lors du calcul des clusters, mais ne centre pas les valeurs aux alentours de zéro (0)</li> <li>◦ <b>NOSCALE</b> : utilise les valeurs de champ clé brutes, non mises à l'échelle, lors du calcul des clusters</li> </ul>

# Exemples

## Exécuter la commande Cluster sur le montant des factures

En plus de stratifier une table Comptabilité client sur le champ **Montant\_facture**, vous décidez également d'exécuter la commande Cluster sur le même champ.

- La stratification regroupe les montants par couche présentant des limites numériques prédéfinies, par exemple, des intervalles de 1000 \$.
- Le regroupement en clusters permet de détecter les regroupements organiques de montants existant dans les données sans que vous ayez à décider d'avance des limites numériques.

```
Ouvrir Ar  
CLUSTER ON Montant_Facture KVALUE 8 ITERATIONS 30 INITIALIZATIONS 10 OTHER Num  
Date d'échéance Réf Type TO "Factures_en_cluster" NOSCALE
```

Pour connaître rapidement le nombre d'enregistrements contenus dans chaque cluster de sortie, vous classez la table de sortie Factures\_en\_cluster sur le champ **Cluster**.

```
OPEN Factures_en_cluster  
CLASSIFY ON Cluster TO SCREEN
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

# Commande COMMENT

Ajoute un commentaire d'explication à un script sans affecter le traitement.

## Syntaxe

### Commentaires sur une seule ligne

```
COMMENT texte_commentaire
```

### commentaires multilignes

```
COMMENT
texte_commentaire
<...n>
<END>
```

#### Remarque

N'utilisez pas le caractère caret ^ avant des lignes de texte de commentaire. Le caret a une utilisation spéciale dans le fichier de projet .acl et le texte de commentaire n'est pas enregistré si vous le faites précéder d'un caret.

## Paramètres

Nom	Description
<i>texte_commentaire</i>	Commentaire que vous ajoutez. <ul style="list-style-type: none"> <li>◦ <b>commentaire sur une seule ligne</b> : saisir le texte entier du commentaire sans saut de ligne</li> <li>◦ <b>commentaire multiligne</b> : saisir autant de lignes de commentaire que nécessaire à partir de la ligne qui suit immédiatement la commande COMMENT</li> </ul> Terminer un commentaire multiligne avec le mot-clé END sur une ligne séparée ou sur une ligne vide
END optionnel	Fin d'une commande COMMENT multiligne. Si vous utilisez END, cette instruction facultative doit être saisie sur la ligne suivant immédiatement la dernière ligne du commentaire. Si vous ignorez END, une ligne vide doit suivre la dernière ligne du commentaire.

# Exemples

## Commentaires sur une seule ligne

Vous utilisez des commentaires d'une seule ligne avant les commandes pour ajouter de la documentation destinée aux utilisateurs futurs qui mettront le script à jour :

```
COMMENT Générer l'écart type et la moyenne.  
STATISTICS ON %v_amt% STD TO SCREEN NUMBER 5  
COMMENT Créer des champs pour stocker l'écart type et la moyenne.  
DEFINE FIELD Ecart_Type COMPUTED STDDEV1  
DEFINE FIELD Moyenne COMPUTED AVERAGE1
```

## Commentaire multiligne

Vous commencez chaque script que vous rédigez par un commentaire multiligne expliquant l'objectif du script :

```
COMMENT  
Cet outil d'analyse identifie les enregistrements multiples ayant des identifiants de créateur de transactions communs  
(comme l'identifiant de fournisseur ou l'identifiant marchand)  
où les valeurs des dates de transaction sont égales ou ont un jour d'écart.  
Cet outil d'analyse peut être utilisé pour les factures ventilées, les bons de commande ventilés,  
les réquisitions ventilées et les transactions par carte d'entreprise ventilées.  
END
```

# Remarques

## Quand utiliser COMMENT

COMMENT permet d'inclure des informations relatives à la portée d'un script, à la logique utilisée et d'autres informations telles que les entrées requises pour le script et l'objectif de chaque variable que vous définissez.

Les commentaires d'un script sont écrits dans la trace de la commande Analytics à chaque exécution du script.

# Commande COUNT

Compte le nombre total d'enregistrements dans la vue active ou uniquement les enregistrements qui remplissent la condition spécifiée.

## Syntaxe

```
COUNT <IF test> <WHILE test> <FIRST plage|NEXT plage>
```

## Paramètres

Nom	Description
IF <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.</p> <p><b>Remarque</b></p> <p>La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p>
WHILE <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.</p> <p><b>Remarque</b></p> <p>Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.</p>
FIRST <i>plage</i>   NEXT <i>plage</i> optionnel	<p>Le nombre d'enregistrements à traiter :</p> <ul style="list-style-type: none"> <li>○ <b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li>○ <b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> </ul> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter.</p> <p>Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p>

# Variables de sortie Analytics

Nom	Contient
COUNT $n$	<p>Nombre d'enregistrements calculé par la commande.</p> <ul style="list-style-type: none"> <li>Si le nom de variable est COUNT1, elle stocke le nombre d'enregistrements de la dernière commande exécutée.</li> <li>Si le nom de variable est COUNT<math>n</math>, où <math>n</math> est supérieur à 1, la variable stocke le nombre d'enregistrements d'une commande exécutée dans une commande GROUP.</li> </ul> <p>La valeur <math>n</math> est attribuée en fonction du numéro de ligne de la commande dans GROUP. Par exemple, si la commande se situe une ligne sous de la commande GROUP, la valeur COUNT2 lui est attribuée. Si la commande se situe quatre lignes sous de la commande GROUP, la valeur COUNT5 lui est attribuée.</p>

## Exemples

### Stockage de COUNT1

Le résultat de la commande COUNT est stocké dans la variable de sortie COUNT1. Vous pouvez récupérer et stocker cette valeur dans une variable définie par l'utilisateur.

La commande COUNT remplace la variable COUNT1 chaque fois qu'elle est exécutée. Par conséquent, la valeur doit être stockée dans une variable définie par l'utilisateur avant la deuxième exécution de la commande, et après l'application du filtre à la table :

```
OPEN AdresseClient
COUNT
TotalEnr = COUNT1
SET FILTER TO DateModification > '20100101'
COUNT
TotalEnrFiltrés = COUNT1
```

## Remarques

### Quand utiliser COUNT

La commande COUNT permet de compter le nombre d'enregistrements dans une table Analytics ou le nombre d'enregistrements qui remplissent une condition test particulière. Si aucun test n'est spécifié, le nombre total d'enregistrements de la table Analytics s'affiche.



## Impact des filtres sur COUNT

Si un filtre est appliqué à une vue, la commande compte le nombre d'enregistrements restant dans la vue une fois la condition de filtrage appliquée.

# Commande CREATE LAYOUT

Crée un format de table Analytics vide, qui peut être requis dans certaines situations de création de scripts.

## Syntaxe

```
CREATE LAYOUT nom_format WIDTH caractères <RECORD 0|RECORD 1>
```

## Paramètres

Nom	Description
<i>nom_format</i>	Nom du format.
WIDTH <i>caractères</i>	Longueur de l'enregistrement en caractères.
RECORD 0   RECORD 1 optionnel	<ul style="list-style-type: none"> <li>Si vous spécifiez RECORD 0 ou si vous omettez ce paramètre, le format de table est créé sans enregistrements ou fichier de données source.</li> <li>Si vous spécifiez RECORD 1, le format de table est créé avec un seul enregistrement vide et un fichier de données source nommé <i>nom_format.fil</i>.</li> </ul>

## Exemples

### Création d'un format de table vide sans enregistrements

Un format de table vide d'une longueur d'enregistrements de 100 caractères est créé :

```
CREATE LAYOUT table_vide WIDTH 100
```

### Création d'un format de table vide avec un enregistrement

Vous créez :

- un format de table vide avec un enregistrement vide
- une longueur d'enregistrement de 50 caractères
- un fichier de données Analytics associé appelé `table_vide.fil`

```
CREATE LAYOUT table_vide WIDTH 50 RECORD 1
```

# Remarques

Le format de table vide est créé avec un seul champ de type caractère appelé **Champ\_1**. La longueur de champ est identique à la longueur d'enregistrement spécifiée avec WIDTH.

## Remarque

Cette commande n'est pas compatible avec une utilisation dans les outils d'analyse Analytics exécutés sur AX Serveur.

# Commande CROSSTAB

Regroupe les enregistrements basés sur des combinaisons identiques de valeurs d'au moins deux champs de type caractère ou numérique et affiche les groupes créés dans une grille de lignes et de colonnes. Compte le nombre d'enregistrements dans chaque groupe et calcule également le sous-total des champs numériques spécifiés pour chaque groupe.

## Syntaxe

```
CROSSTAB <ON> champ_ligne <...n> COLUMNS champ_colonne <SUBTOTAL champ_numérique <...n> | SUBTOTAL ALL> TO {SCREEN|nom_table|nom_fichier|GRAPH|PRINT} <IF test>
<WHILE test> <FIRST plage|NEXT plage> <APPEND> <COUNT> <OPEN> <LOCAL>
<HEADER texte_en-tête> <FOOTER texte_pied_page>
```

## Paramètres

Nom	Description
ON <i>champ_ligne</i> <...n>	Le champ ou l'expression à utiliser pour les lignes dans la grille de lignes et de colonnes ainsi créée. Vous pouvez spécifier un ou plusieurs champs ou expressions servant de base pour les lignes.
COLUMNS <i>champ_colonne</i>	Le champ ou l'expression à utiliser pour les colonnes dans la grille de lignes et de colonnes ainsi créée. Vous ne pouvez définir qu'un seul champ ou une seule expression pour les colonnes.
SUBTOTAL <i>champ_numérique</i> <...n>   SUBTOTAL ALL optionnel	Un ou plusieurs champs ou expressions numériques à sous-totaliser pour chaque groupe.  Les champs multiples doivent être séparés par des espaces. Indiquez ALL pour calculer le sous-total de tous les champs numériques de la table.
TO SCREEN   <i>nom_table</i>   <i>nom_fichier</i>   GRAPH   PRINT	Emplacement vers lequel envoyer les résultats de la commande : <ul style="list-style-type: none"> <li>◦ <b>SCREEN</b> : affiche les résultats dans la zone d'affichage d'Analytics.</li> <li>◦ <b><i>nom_table</i></b> : enregistre les résultats dans une table Analytics</li> </ul> <p>Indiquez <i>nom_table</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple : TO "Sortie.FIL"</p> <p>Par défaut, le fichier de données de table (.FIL) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.FIL"</li> <li>• TO "Résultats\Sortie.FIL"</li> </ul>

Nom	Description
	<p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alphanumériques, sans l'extension .FIL. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p> <ul style="list-style-type: none"> <li>◦ <b>nom_fichier</b> : enregistre les résultats dans un fichier</li> </ul> <p>Indiquez <i>nom_fichier</i> sous forme de chaîne entre guillemets avec l'extension de fichier appropriée. Par exemple : TO "Sortie.TXT"</p> <p>Par défaut, le fichier est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.TXT"</li> <li>• TO "Résultats\Sortie.TXT"</li> </ul> <ul style="list-style-type: none"> <li>◦ <b>GRAPH</b> : affiche les résultats dans un graphique dans la zone d'affichage d'Analytics</li> <li>◦ <b>PRINT</b> : envoie les résultats vers l'imprimante par défaut</li> </ul>
IF <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.</p> <p><b>Remarque</b></p> <p>La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p>
WHILE <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.</p> <p><b>Remarque</b></p> <p>Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.</p>
FIRST <i>plage</i>   NEXT <i>plage</i> optionnel	<p>Le nombre d'enregistrements à traiter :</p> <ul style="list-style-type: none"> <li>◦ <b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li>◦ <b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> </ul> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter.</p> <p>Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p>
APPEND optionnel	<p>Ajoute la sortie de commande à la fin d'un fichier existant au lieu de remplacer ce fichier.</p>

Nom	Description
	<p><b>Remarque</b></p> <p>Vous devez vous assurer que la structure de la sortie de la commande et du fichier existant est identique :</p> <ul style="list-style-type: none"> <li>• les mêmes champs</li> <li>• le même ordre des champs</li> <li>• les champs correspondants ont la même longueur</li> <li>• les champs correspondants ont le même type de données</li> </ul> <p>Analytics ajoute la sortie à un fichier existant sans tenir compte de sa structure, ce qui peut désordonner les données si la structure de la sortie et du fichier existant ne correspond pas.</p>
COUNT optionnel	<p>Inclut le nombre d'enregistrements sous forme de colonnes. Les comptes sont particulièrement utiles lors de l'utilisation de SUBTOTAL.</p> <p>Les comptes sont inclus automatiquement si vous ne sélectionnez pas de champs de sous-total.</p>
OPEN optionnel	Ouvre la table créée par la commande après l'exécution de la commande. Valide uniquement lorsque la commande crée une table de sortie.
LOCAL optionnel	<p>Enregistre le fichier de sortie au même emplacement que le projet Analytics.</p> <p><b>Remarque</b></p> <p>Applicable uniquement lorsque vous exécutez la commande sur une table de serveur avec un fichier de sortie qui est une table Analytics.</p>
HEADER <i>texte_en-tête</i> optionnel	<p>Texte à insérer en haut de chaque page d'un rapport.</p> <p><i>texte_en-tête</i> doit être indiqué sous forme de chaîne entre guillemets. La valeur remplace la variable système Analytics HEADER.</p>
FOOTER <i>texte_pied_page</i> optionnel	<p>Texte à insérer en bas de chaque page d'un rapport.</p> <p><i>texte_pied_page</i> doit être indiqué sous forme de chaîne entre guillemets. La valeur remplace la variable système Analytics FOOTER.</p>

## Exemples

### Tabulation croisée d'une table Comptes clients avec SUBTOTAL

Vous souhaitez effectuer une tabulation croisée sur une table Comptes clients par les champs **Numéro client** et **Type de transaction**. Vous souhaitez également sous-totaliser le champ **Montant des transactions**.

La sortie est regroupée par client et, dans chaque client, par type de transaction. Elle inclut le montant total des transactions pour chaque client pour chaque type de transaction :

```
OPEN Comptes_clients
CROSSTAB ON Numéro_Client COLUMNS Type_Trans SUBTOTAL Montant_Trans COUNT TO
SCREEN
```

## Appliquer une tabulation croisée sur une table Comptes clients pour rechercher des transactions en doublons

Vous devez trouver des preuves de vos transactions en doublons dans une table Comptes clients.

Pour ce faire, vous appliquez une tabulation croisée à une table Comptes clients en fonction des champs **Montant des transactions** et **Type de transaction**. La sortie regroupe et compte les montants identiques de transaction pour chaque type de transaction :

```
OPEN Comptes_clients
CROSSTAB ON Montant_Trans COLUMNS Type_Trans TO SCREEN
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Fonctionnement

CROSSTAB regroupe les enregistrements présentant la même combinaison de valeurs dans au moins deux champs de type caractère ou numérique.

La sortie contient un quadrillage de lignes et de colonnes similaire à un tableau croisé dynamique. Elle contient une seule intersection ligne-colonne pour chaque groupe, avec le compte du nombre d'enregistrements de la table source inclus dans le groupe.

## Tri et CROSSTAB

CROSSTAB peut traiter les données triées ou non triées. Le *champ\_ligne* et le *champ\_colonne* dans la sortie sont automatiquement triés dans l'ordre croissant.

Si vous spécifiez plusieurs *champ\_ligne*, les champs utilisent un tri imbriqué, commençant au premier *champ\_ligne* spécifié.

# Commande CVSEVALUATE

Pour l'échantillonnage de variables classiques : fournit quatre méthodes différentes pour projeter les résultats de l'analyse de l'échantillon sur l'intégralité de la population.

## Syntaxe

```
CVSEVALUATE BOOKED champ_valeur_comptable AUDITED champ_valeur_audit ETYPE
{MPU|DIFFERENCE|RATIO SEPARATE|RATIO COMBINED} STRATA valeur_limite <;...n>
POPULATION compte_couche; valeur_comptable_couche <;...n> CONFIDENCE niveau_confiance
CUTOFF valeur; compte_couche_certitude; valeur_comptable_couche_certitude ERRORLIMIT
nombre PLIMIT {BOTH|UPPER|LOWER} <TO {SCREEN|nom_fichier}>
```

## Paramètres

### Remarque

Si vous utilisez les résultats de sortie des commandes CVSPREPARE et CVSSAMPLE comme entrée pour la commande CVSEVALUATE, un certain nombre de valeurs de paramètres est déjà spécifié et stocké dans des variables. Pour plus d'informations, consultez la section "Commande CVSPREPARE" Page 116 et "Commande CVSSAMPLE" Page 120.

N'incluez pas les séparateurs des milliers ou les symboles de pourcentage lorsque vous saisissez des valeurs.

Nom	Description
BOOKED <i>champ_valeur_comptable</i>	Champ valeur comptable numérique à utiliser dans l'évaluation.
AUDITED <i>champ_valeur_audit</i>	Champ valeur d'audit numérique à utiliser dans l'évaluation.
ETYPE MPU   DIFFERENCE   RATIO SEPARATE   RATIO COMBINED	Type d'estimation à utiliser : <ul style="list-style-type: none"> <li>• MPU (moyenne unitaire)</li> <li>• Différence</li> <li>• Ratio distinct</li> <li>• Ratio combiné</li> </ul> <p>Pour plus d'informations, consultez la section "Quel type d'estimation dois-je utiliser ?" Page 114</p>
STRATA <i>valeur_limite</i> <;...n>	Valeurs limites supérieures à utiliser pour stratifier le <i>champ_valeur_comptable</i> .



Nom	Description
POPULATION <i>compte_couche</i> ; <i>valeur_couche</i> <;...n>	Nombre d'enregistrements et valeur totale de chaque couche dans le <i>champ_valeur_comptable</i> .
CONFIDENCE <i>niveau_confiance</i>	Niveau de confiance utilisé pendant l'étape de préparation de l'échantillonnage de variables classiques.
CUTOFF <i>valeur</i> ; <i>compte_couche_certitude</i> ; <i>valeur_comptable_couche_certitude</i>	<ul style="list-style-type: none"> <li>◦ <b>valeur</b> : valeur limite de couche de certitude utilisée pendant l'étape de préparation et d'échantillonnage de l'échantillon de variables classiques</li> <li>◦ <b>compte_couche_certitude</b> : nombre d'enregistrements se trouvant dans la couche de certitude</li> <li>◦ <b>valeur_comptable_couche_certitude</b> : valeur comptable totale des enregistrements dans la couche de certitude</li> </ul>
ERRORLIMIT <i>nombre</i>	<p>Nombre minimal d'erreurs que vous prévoyez dans l'échantillon.</p> <p><b>Remarque</b></p> <p>Si le nombre réel d'erreurs que vous avez trouvées lorsque vous avez analysé l'échantillon est inférieur au <i>nombre</i> ERRORLIMIT, la seule méthode d'évaluation possible est la moyenne unitaire.</p>
PLIMIT BOTH   UPPER   LOWER	<p>Type de limite de précision à utiliser :</p> <ul style="list-style-type: none"> <li>◦ Les deux</li> <li>◦ Supérieure</li> <li>◦ Inférieure</li> </ul> <p>Pour plus d'informations, consultez la section "Commande CVSPREPARE" Page 116.</p>
TO SCREEN   <i>nom_fichier</i>	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li>◦ <b>SCREEN</b> : affiche les résultats dans la zone d'affichage d'Analytics.</li> <li>◦ <b>nom_fichier</b> : enregistre les résultats dans un fichier</li> </ul> <p>Indiquez <i>nom_fichier</i> sous forme de chaîne entre guillemets avec l'extension de fichier appropriée. Par exemple : TO "Sortie.TXT"</p> <p>Par défaut, le fichier est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.TXT"</li> <li>• TO "Résultats\Sortie.TXT"</li> </ul>

## Exemples

### Projeter les erreurs détectées dans les données échantillonnées sur l'intégralité de la population

Vous avez effectué vos tests sur les données échantillonnées et vous avez enregistré les anomalies

détectées. Vous pouvez désormais projeter les erreurs que vous avez détectées sur l'intégralité de la population.

L'exemple ci-dessous utilise le type d'estimation Différence pour projeter les résultats de l'analyse de l'échantillon sur la population entière :

```
CVSEVALUATE BOOKED montant_facture AUDITED AUDIT_VALUE ETYPE DIFFERENCE
STRATA 4376,88;9248,74;16904,52;23864,32 POPULATION
1279;3382131,93;898;5693215,11;763;9987014,57;627;12657163,59;479;13346354,63
CONFIDENCE 95,00 CUTOFF 35000,00;36;1334318,88 ERRORLIMIT 6 PLIMIT BOTH TO
SCREEN
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Quel type d'estimation dois-je utiliser ?

Le type d'estimation à utiliser dépend de la nature des données : les valeurs comptables de l'échantillon, les valeurs d'audit de l'échantillon et les relations entre elles.

### Instructions

Les recommandations ci-dessous vous aident à choisir un type d'estimation. Vous pouvez répéter l'étape d'évaluation avec différents types d'estimation, puis comparer les résultats de chacun.

Type d'estimation	Présence d'anomalies	Taille des anomalies	Signe des valeurs comptables	Comparaison des ratios de couche
Moyenne unitaire	<p><b>Aucune anomalie ou très peu d'anomalies</b></p> <p>Le seul type d'estimation valide en l'absence d'anomalie ou en leur présence exceptionnelle, dans la population de l'échantillon auditée.</p>	SO	SO	SO
Différence	<p><b>Anomalies requises</b></p> <p>Requiert un certain nombre d'anomalies</p>	<p><b>Anomalies non proportionnelles</b></p> <p>Plus adapté lorsque</p>	SO	SO

Type d'estimation	Présence d'anomalies	Taille des anomalies	Signe des valeurs comptables	Comparaison des ratios de couche
	<p>dans la population de l'échantillon audité.</p> <p>Par exemple, au moins 5 % des échantillons contiennent des anomalies.</p>	<p>les anomalies sont non proportionnelles : la taille d'une anomalie n'est pas relative à la taille de la valeur comptable associée.</p> <p>Autrement dit, les petites et grandes valeurs comptables peuvent présenter de petites ou de grosses anomalies.</p>		
Ratio distinct		<p><b>Anomalies proportionnelles</b></p> <p>Plus adapté lorsque les anomalies sont proportionnelles : la taille d'une anomalie est relative à la taille de la valeur comptable associée.</p> <p>Autrement dit, les petites valeurs comptables présentent de petites anomalies et les grandes valeurs comptables présentent de grosses anomalies.</p>	<p><b>Les valeurs comptables ont le même signe</b></p> <p>Toutes les valeurs comptables de l'échantillon ont le même signe : soit elles sont toutes positives soit elles sont toutes négatives.</p>	<p><b>Ratios variés</b></p> <p>Plus adapté lorsque le ratio de la valeur moyenne d'audit de l'échantillon par rapport à la moyenne des valeurs comptables de l'échantillon varie grandement entre les couches.</p>
Ratio combiné				<p><b>Ratios cohérents</b></p> <p>Plus adapté lorsque le ratio de la valeur moyenne d'audit de l'échantillon par rapport à la valeur comptable moyenne de l'échantillon est relativement cohérent entre les différentes couches.</p>

# Commande CVSPREPARE

Stratifie une population et calcule une taille d'échantillon statistiquement valide pour chaque couche dans le cas de l'échantillonnage de variables classiques.

## Syntaxe

```
CVSPREPARE <ON> champ_valeur_comptable NUMSTRATA nombre MINIMUM taille_échantillon_couches_minimale PRECISION valeur CONFIDENCE niveau_confiance <CUTOFF valeur> NCELLS nombre PLIMIT {BOTH|UPPER|LOWER} <ERRORLIMIT nombre <MINSAMPSIZE taille_échantillon_minimale> TO SCREEN|nom_fichier}
```

## Paramètres

### Remarque

N'incluez pas les séparateurs des milliers ou les symboles de pourcentage lorsque vous saisissez des valeurs.

Nom	Description
ON <i>valeur_champ_comptable</i>	Champ valeur comptable numérique à utiliser pour préparer l'échantillonnage de variables classiques.
NUMSTRATA <i>nombre</i>	<p>Nombre de couches à utiliser pour stratifier numériquement le <i>champ_valeur_comptable</i>.</p> <p>Le nombre minimal de couches est 1 et le nombre maximal est 256.</p> <p>Si vous indiquez NUMSTRATA 1 mais pas de CUTOFF, la population reste non stratifiée avant l'extraction d'un échantillon.</p> <p><b>Remarque</b></p> <p>Le nombre de couches ne peut pas dépasser 50 % du nombre de cellules spécifiées pour NCELLS.</p>
MINIMUM <i>taille_échantillon_couches_minimale</i>	<p>Nombre minimum d'enregistrements à échantillonner dans chaque couche.</p> <p>Laissez la valeur de zéro (0) par défaut si vous n'avez aucune raison particulière de préciser un nombre minimum.</p>
PRECISION <i>valeur</i>	<p>Le montant monétaire qui est la différence entre l'anomalie acceptable et l'anomalie probable dans le compte.</p> <ul style="list-style-type: none"> <li><b>Anomalie acceptable</b> : le montant total maximal de l'anomalie que peut présenter le champ de l'échantillon sans que cela soit considéré comme étant une anomalie significative</li> <li><b>Anomalie probable</b> : le montant total de l'anomalie que vous prévoyez de voir conte-</li> </ul>

Nom	Description
	<p>nir le champ de l'échantillon</p> <p>La précision définit la plage d'acceptabilité pour qu'un compte soit considéré comme évalué justement.</p> <p>Réduire la précision diminue la plage d'acceptabilité (la marge d'erreur) nécessitant une plus grande taille d'échantillon.</p>
CONFIDENCE <i>niveau_ confiance</i>	<p>Le niveau de confiance que vous souhaitez attribuer au fait que l'échantillon créé est représentatif de la population totale.</p> <p>Par exemple, saisir 95 signifie que vous souhaitez être sûr 95 % du temps que l'échantillon sera bien représentatif. La confiance est le complément du « risque d'échantillonnage ». Un niveau de confiance de 95 % équivaut à un risque d'échantillonnage de 5 %.</p> <ul style="list-style-type: none"> <li>○ Si PLIMIT est BOTH, le niveau de confiance minimum est de 10 % et le niveau de confiance maximum est de 99,5 %.</li> <li>○ Si PLIMIT est UPPER ou LOWER, le niveau de confiance minimum est de 55 % et le niveau de confiance maximum est de 99,5 %.</li> </ul>
CUTOFF <i>valeur</i> optionnel	<p>Valeur limite de couche de certitude.</p> <p>Les montants dans <i>champ_valeur_comptable</i> qui sont supérieurs ou égaux à la valeur limite sont sélectionnés et inclus automatiquement dans l'échantillon.</p> <p>Si vous ignorez CUTOFF, une valeur limite par défaut égale au montant maximum dans le <i>champ_valeur_comptable</i> est utilisée.</p>
NCELLS <i>nombre</i>	<p>Nombre de cellules à utiliser pour pré-stratifier le <i>champ_valeur_comptable</i>.</p> <p>Les cellules sont des divisions numériques plus étroites que les couches. La pré-stratification fait partie d'un processus interne optimisant la position des limites des couches. Les cellules ne sont pas conservées dans la sortie stratifiée définitive.</p> <p>Le nombre minimal de cellules est 2 et le nombre maximal est 999.</p> <p><b>Remarque</b></p> <p>Le nombre de cellules doit être au moins le double (2 x) du nombre de couches indiqué pour NUMSTRATA.</p>
PLIMIT BOTH   UPPER   LOWER	<p>Type de limite de précision à utiliser.</p> <ul style="list-style-type: none"> <li>○ BOTH : spécifiez cette option si : <ul style="list-style-type: none"> <li>• le compte dans son ensemble peut être surestimé ou sous-estimé</li> <li>• vous êtes intéressé pour estimer si l'anomalie dans un des deux sens dépasse la PRECISION indiquée</li> </ul> </li> <li>○ UPPER : spécifiez cette option si : <ul style="list-style-type: none"> <li>• le compte dans son ensemble est susceptible d'être sous-estimé</li> <li>• vous êtes intéressé pour estimer si le montant total de la sous-évaluation dépasse la PRECISION indiquée</li> </ul> </li> <li>○ LOWER : spécifiez cette option si : <ul style="list-style-type: none"> <li>• le compte dans son ensemble est susceptible d'être surestimé</li> <li>• vous êtes intéressé pour estimer si le montant total de la surévaluation dépasse la PRECISION indiquée</li> </ul> </li> </ul> <p><b>Attention</b></p> <p>Indiquez BOTH si vous ne savez pas quelle option indiquer.</p>

Nom	Description
ERRORLIMIT <i>nombre</i>	<p>Nombre minimal d'erreurs que vous prévoyez dans l'échantillon.</p> <p><b>Remarque</b></p> <p>Si le nombre réel d'erreurs que vous trouvez lorsque vous analysez l'échantillon est inférieur au <i>nombre</i> ERRORLIMIT, la seule méthode d'évaluation possible est la moyenne unitaire.</p>
MINSAMPSIZE <i>taille_échantillon_minimum</i> optionnel	<p>Nombre minimum d'enregistrements à échantillonner dans l'intégralité de la population.</p> <p>Laissez la valeur de zéro (0) par défaut si vous n'avez aucune raison particulière de préciser un nombre minimum.</p>
TO SCREEN   <i>nom_fichier</i>	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li>o <b>SCREEN</b> : affiche les résultats dans la zone d'affichage d'Analytics.</li> <li>o <b><i>nom_fichier</i></b> : enregistre les résultats dans un fichier</li> </ul> <p>Indiquez <i>nom_fichier</i> sous forme de chaîne entre guillemets avec l'extension de fichier appropriée. Par exemple : TO "Sortie.TXT"</p> <p>Par défaut, le fichier est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.TXT"</li> <li>• TO "Résultats\Sortie.TXT"</li> </ul>

## Variables de sortie Analytics

Nom	Contient
CONFIDENCE	Niveau de confiance indiqué par l'utilisateur.
ERRLIMIT	Nombre minimum d'erreurs spécifié par l'utilisateur.
NSTRATA	Nombre de couches spécifié par l'utilisateur.
PLIMIT	Type de limite de précision spécifiée par l'utilisateur.
S_TOP	Valeur de la limite de la couche de certitude spécifiée par l'utilisateur ; si aucune valeur n'a été indiquée, il s'agit de la limite supérieure de la couche supérieure calculée par la commande.
SAMPLEFIELD	Champ de valeur comptable spécifié par l'utilisateur.
SBOTTOM	Limite inférieure de la couche inférieure calculée par la commande.
SBOUNDARY	Toutes les limites supérieures des couches calculées par la commande, et S_TOP. Ne stocke pas SBOTTOM.

Nom	Contient
SPOPULATION	Le compte du nombre d'enregistrements dans chaque couche et la valeur monétaire totale de chaque couche. Exclut les éléments supérieurs à la limite de couche de certitude
SSAMPLE	Taille d'échantillon de chaque couche calculée par la commande.

## Exemples

### Préparer un échantillon de variables classiques

Vous avez décidé d'utiliser un échantillonnage de variables classiques pour estimer le montant total d'anomalie monétaire dans un compte contenant des factures.

Avant d'extraire l'échantillon, vous devez d'abord stratifier la population et calculer une taille d'échantillon statistiquement valide pour chaque couche.

Vous souhaitez être sûr 95 % du temps que l'échantillon extrait par Analytics sera bien représentatif de l'ensemble de la population.

À l'aide du niveau de confiance que vous avez spécifié, l'exemple ci-dessous stratifie une table à partir du champ **Montant\_Facture** et calcule la taille d'échantillon correspondant à chaque couche et à la couche de certitude :

```
CVSPREPARE ON Montant_Facture NUMSTRATA 5 MINIMUM 0 PRECISION 928003,97
CONFIDENCE 95,00 CUTOFF 35000 NCELLS 50 PLIMIT BOTH ERRORLIMIT 6 MINSAMPSIZE 0
TO SCREEN
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

### Limitation de longueur numérique

Plusieurs calculs internes ont eu lieu pendant l'étape de préparation de l'échantillonnage de variables classiques. Ces calculs prennent en charge les nombres d'une longueur maximale de 17 chiffres. Si le résultat d'un calcul dépasse 17 chiffres, ce résultat ne figure pas dans la sortie et vous ne pouvez pas poursuivre le processus d'échantillonnage.

Notez que les nombres des données source d'une longueur inférieure à 17 chiffres peuvent générer des résultats de calculs internes dépassant les 17 chiffres.

# Commande CVSSAMPLE

Extrait un échantillon d'enregistrements à l'aide de la méthode de l'échantillonnage de variables classiques.

## Syntaxe

```
CVSSAMPLE ON champ_valeur_comptable NUMSTRATA nombre <SEED valeur_départ>
CUTOFF valeur STRATA valeur_limite <;...n> SAMPLESIZE nombre <;...n> POPULATION
compte_couche; valeur_couche <;...n> TO nom_table
```

## Paramètres

### Remarque

Si vous utilisez les résultats de sortie de la commande CVSPREPARE comme entrée pour la commande CVSSAMPLE, un certain nombre de valeurs de paramètres est déjà spécifié et stocké dans des variables. Pour plus d'informations, consultez la section "Commande CVSPREPARE" Page 116.

N'incluez pas les séparateurs des milliers ou les symboles de pourcentage lorsque vous saisissez des valeurs.

Nom	Description
ON <i>valeur_champ_comptable</i>	Champ valeur comptable numérique à utiliser comme base pour l'échantillon.
NUMSTRATA <i>nombre</i>	Nombre de couches à utiliser pour stratifier le <i>champ_valeur_comptable</i> .
SEED <i>valeur_départ</i> optionnel	Valeur de départ à utiliser pour lancer le générateur de nombres aléatoires dans Analytics. Si vous omettez SEED, Analytics sélectionne aléatoirement la valeur de départ.
CUTOFF <i>valeur</i>	Valeur limite de couche de certitude. Les montants dans <i>champ_valeur_comptable</i> qui sont supérieurs ou égaux à la <i>valeur</i> limite sont sélectionnés et inclus automatiquement dans l'échantillon.
STRATA <i>valeur_limite</i> <;... <i>n</i> >	Valeurs limites supérieures à utiliser pour stratifier le <i>champ_valeur_comptable</i> .
SAMPLESIZE <i>nombre</i> <;... <i>n</i> >	Nombre d'enregistrements à échantillonner dans chaque couche.



Nom	Description
POPULATION <i>compte_couche; valeur_couche</i> <;...n>	Nombre d'enregistrements dans chaque couche et valeur totale de chaque couche.
TO <i>nom_table</i>	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li>◦ <b>nom_table</b> : enregistre les résultats dans une table Analytics</li> </ul> <p>Indiquez <i>nom_table</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple : TO "Sortie.FIL"</p> <p>Par défaut, le fichier de données de table (.FIL) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.FIL"</li> <li>• TO "Résultats\Sortie.FIL"</li> </ul> <p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alphanumériques, sans l'extension .FIL. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>

## Variables de sortie Analytics

Nom	Contient
S_TOPEV	<p>Valeur de la limite de la couche de certitude spécifiée par l'utilisateur ; si aucune valeur n'a été indiquée, il s'agit de la limite supérieure de la couche supérieure calculée précédemment par la commande CVSPREPARE.</p> <p>Stocke aussi le compte du nombre d'enregistrements de la couche de certitude et sa valeur monétaire totale.</p>
SBOTTOMEV	Limite inférieure de la couche inférieure calculée par la commande.
BOUNDARYEV	Toutes les limites supérieures de couches préremplies par la commande ou spécifiées par l'utilisateur. Ne stocke pas S_TOPEV ou SBOTTOMEV.
SPOPULATION	Le compte du nombre d'enregistrements dans chaque couche et la valeur monétaire totale de chaque couche. Exclut les éléments supérieurs à la limite de couche de certitude

## Exemples

### Extraire un échantillon de variables classiques

Vous allez utiliser un échantillonnage de variables classiques pour estimer le montant total d'anomalie monétaire dans un compte contenant des factures.

Après avoir stratifié la population et calculé une taille d'échantillon statistiquement valide pour chaque couche, vous voilà prêt à extraire l'échantillon.

L'exemple ci-dessous extrait un échantillon stratifié des enregistrements en fonction du champ **montant\_facture** et sort les enregistrements échantillonnés dans la table `Factures_échantillon` :

```
CVSSAMPLE ON Montant_Facture NUMSTRATA 5 SEED 12345 CUTOFF 35000,00 STRATA
4376,88;9248,74;16904,52;23864,32;35000,00 SAMPLESIZE 37;36;49;36;39 POPULATION
1279;3382131,93;898;5693215,11;763;9987014,57;627;12657163,59;479;13346354,63 TO "Fac-
tures_échantillon"
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Champs générés par le système

Analytics génère automatiquement quatre champs et les ajoute à la table de sortie de l'échantillon. Pour chaque enregistrement de l'échantillon, les champs contiennent les informations descriptives suivantes :

- **STRATUM** : le numéro de la couche à laquelle l'enregistrement est attribué
- **ORIGIN\_RECORD\_NUMBER** : numéro d'enregistrement d'origine dans la table de données source
- **SELECTION\_ORDER** : sur une base par couche : l'ordre dans lequel l'enregistrement a été sélectionné de manière aléatoire
- **SAMPLE\_RECORD\_NUMBER** : le numéro d'enregistrement dans la table de sortie des échantillons

# Commande DEFINE COLUMN

Crée et ajoute une ou plusieurs colonnes à une vue existante.

## Syntaxe

```
DEFINE COLUMN nom_vue nom_champ <AS nom_affichage> <POSITION n> <WIDTH caractères>
<PIC format> <SORT|SORT D> <KEY> <PAGE> <NODUPS> <NOZEROS> <LINE n>
```

## Paramètres

Nom	Description
<i>nom_vue</i>	Vue à laquelle la colonne doit être ajoutée.
<i>nom_champ</i>	Champ pour lequel la colonne doit être créée. Pour utiliser un champ d'une table associée, spécifiez le nom du champ sous la forme <i>nom_table.nom_champ</i> .
AS <i>nom_affichage</i> optionnel	Nom d'affichage (titre de la colonne) pour le champ dans la vue. Si vous souhaitez que le nom d'affichage soit identique au champ nom, n'utilisez pas AS. Indiquez le <i>nom_affichage</i> sous forme de chaîne entre guillemets. Utilisez un point-virgule (;) entre les mots si vous souhaitez insérer un saut de ligne dans le titre de la colonne.
POSITION <i>n</i> optionnel	Position numérique, de gauche à droite, de la colonne : <ul style="list-style-type: none"> <li>si vous ignorez ce paramètre, la colonne est placée à l'extrême droite lors de son ajout</li> <li>si un numéro de position est manquant, les positions des colonnes sont ajustées afin de les organiser par ordre séquentiel</li> <li>si une position est déjà utilisée, la nouvelle colonne est placée à gauche de la colonne occupant la position</li> </ul>
WIDTH <i>caractères</i> optionnel	Largeur d'affichage du champ en nombre de caractères. La valeur indiquée contrôle la largeur d'affichage du champ dans les vues et rapports Analytics. La largeur d'affichage n'altère jamais les données ; elle peut toutefois masquer des données si elle est plus courte que la longueur du champ. Si vous ignorez WIDTH, la largeur d'affichage est définie sur la largeur de caractères spécifiée pour le champ dans le format de table.

Nom	Description
	<p><b>Remarque</b></p> <p>Les caractères spécifiés par WIDTH sont des caractères à largeur fixe. Le même espace est attribué à chaque caractère, quelle que soit la longueur du caractère réel.</p> <p>Par défaut, les vues dans Analytics utilisent une largeur proportionnelle qui ne correspond pas à l'espacement des caractères à largeur fixe.</p> <p>Si vous souhaitez une correspondance un à un entre la valeur WIDTH et les caractères de la vue, vous pouvez modifier le paramètre <b>Police proportionnelle</b> dans la boîte de dialogue <b>Options</b> sur une police à chasse fixe comme Courier New.</p>
PIC <i>format</i> optionnel	<p><b>Remarque</b></p> <p>S'applique aux champs numériques ou DateHeure.</p> <ul style="list-style-type: none"> <li>◦ <b>champs numériques</b> : format d'affichage des valeurs numériques dans les vues et rapports Analytics</li> <li>◦ <b>champs DateHeure</b> : format physique des valeurs DateHeure dans les données source (ordre des caractères date et heure, séparateurs, etc.)</li> </ul> <p><b>Remarque</b></p> <p>Pour les champs DateHeure, le <i>format</i> doit correspondre exactement au format physique dans les données source. Par exemple, si les données source sont 12/31/2014, vous devez saisir la date au format "MM/JJ/AAAA".</p> <p>Le <i>format</i> doit être placé entre guillemets.</p>
SORT   SORT D optionnel	<p>Trie la colonne :</p> <ul style="list-style-type: none"> <li>◦ <b>ordre croissant</b> : SORT</li> <li>◦ <b>ordre décroissant</b> : SORT D</li> </ul>
KEY optionnel	<p>La colonne est désignée comme champ de fractionnement dans les rapports. Les rapports sont sous-totalisés et sous-divisés lorsque la valeur de la colonne diffère. Les limitations ci-dessous s'appliquent aux champs de fractionnement :</p> <ul style="list-style-type: none"> <li>◦ doit être un champ ou une expression de type caractère</li> <li>◦ si un champ de fractionnement est défini dans la vue, il doit correspondre à la colonne la plus à gauche</li> <li>◦ la dernière colonne dans la vue ne peut pas être un champ de fractionnement</li> <li>◦ si plusieurs champs de fractionnement existent, toutes les colonnes à gauche d'un champ de fractionnement supplémentaire doivent également être des champs de fractionnement</li> </ul>
PAGE optionnel	<p>Insère un saut de page chaque fois que la valeur du champ de fractionnement diffère.</p>
NODUPS optionnel	<p>Substitue des valeurs vides aux valeurs répétées dans le champ.</p> <p>Par exemple, si le nom d'un client est répertorié pour chaque enregistrement de facture, le rapport est plus lisible s'il montre uniquement la première occurrence de chaque nom de client.</p>

Nom	Description
NOZEROS optionnel	Substitue des valeurs vides aux valeurs nulles dans le champ. Par exemple, si, dans un rapport, un champ contient un grand nombre de zéros, la lecture du rapport est facilitée si ce dernier n'affiche que les valeurs non nulles.
LINE <i>n</i> optionnel	Nombre de lignes dans la colonne. Si aucune valeur n'est indiquée, la colonne comprend par défaut une seule ligne. La valeur <i>n</i> doit être comprise entre 2 et 60.

## Exemples

### Définir une vue avec 6 colonnes

Une fois la table **AR** ouverte, vous définissez une vue **Rapport\_AR** et vous définissez 6 colonnes. Les colonnes s'affichent dans l'ordre indiqué :

```
OPEN Comptes_clients
DEFINE VIEW Rapport_AR OK
DEFINE COLUMN Rapport_AR No AS "Numéro client" WIDTH 7 KEY
DEFINE COLUMN Rapport_AR Date AS "Date facture" WIDTH 10
DEFINE COLUMN Rapport_AR Echéance AS "Date échéance" WIDTH 10
DEFINE COLUMN Rapport_AR Référence AS "Numéro référence" WIDTH 6
DEFINE COLUMN Rapport_AR Type AS "Type transaction" WIDTH 5
DEFINE COLUMN Rapport_AR Montant AS "Montant transaction" WIDTH 12 PIC "-9999999999,99"
```

# Commande DEFINE FIELD

Définit un champ de données physique dans un format de table Analytics.

## Syntaxe

```
DEFINE FIELD nom_champ type_données position_départ longueur <décimales|format_date>
<NDATETIME> <PIC format> <AS nom_affichage> <WIDTH caractères> <SUPPRESS> <note_
champ>
```

## Paramètres

Nom	Description
<i>nom_champ</i>	<p>Nom du champ.</p> <p><b>Remarque</b></p> <p>La longueur des noms des champs ne doit pas dépasser 256 caractères alphanumériques majuscules et minuscules. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p> <p>Un certain nombre de mots-clés réservés ne peuvent pas être utilisés comme noms de champs dans Analytics. Pour consulter la liste complète, voir "Mots-clés réservés" Page 985.</p>
<i>type_données</i>	<p>Type de données à utiliser pour l'interprétation des données. Pour obtenir la liste des types de données pris en charge, consultez la rubrique "Types de données prises en charge" Page 131.</p> <p>Par exemple, les numéros de factures peuvent être stockés en tant que valeurs numériques dans la source de données. Pour que ces valeurs soient traitées en tant que chaînes de caractères plutôt que comme nombres, vous pouvez définir le champ en tant que données de type caractère à la place.</p>
<i>position_départ</i>	La position d'octet de départ du champ dans le fichier de données Analytics.

Nom	Description						
	<p><b>Remarque</b></p> <table border="1" data-bbox="570 312 1349 533"> <tr> <td data-bbox="570 312 1037 375">Analytics non Unicode</td> <td data-bbox="1037 312 1349 375">1 octet = 1 caractère</td> </tr> <tr> <td data-bbox="570 375 1037 470">Analytics Unicode, données ASCII étendu (ANSI)</td> <td data-bbox="1037 375 1349 470">1 octet = 1 caractère</td> </tr> <tr> <td data-bbox="570 470 1037 533">Analytics Unicode, données Unicode</td> <td data-bbox="1037 470 1349 533">2 octets = 1 caractère</td> </tr> </table> <p>Pour les données Unicode, normalement, vous devez indiquer une position d'octet de départ avec un nombre impair. Indiquer une position de départ avec un chiffre pair peut entraîner l'affichage incorrect des caractères.</p>	Analytics non Unicode	1 octet = 1 caractère	Analytics Unicode, données ASCII étendu (ANSI)	1 octet = 1 caractère	Analytics Unicode, données Unicode	2 octets = 1 caractère
Analytics non Unicode	1 octet = 1 caractère						
Analytics Unicode, données ASCII étendu (ANSI)	1 octet = 1 caractère						
Analytics Unicode, données Unicode	2 octets = 1 caractère						
<i>longueur</i>	<p>Longueur du champ en octets.</p> <p><b>Remarque</b></p> <table border="1" data-bbox="570 785 1349 1005"> <tr> <td data-bbox="570 785 1037 848">Analytics non Unicode</td> <td data-bbox="1037 785 1349 848">1 octet = 1 caractère</td> </tr> <tr> <td data-bbox="570 848 1037 942">Analytics Unicode, données ASCII étendu (ANSI)</td> <td data-bbox="1037 848 1349 942">1 octet = 1 caractère</td> </tr> <tr> <td data-bbox="570 942 1037 1005">Analytics Unicode, données Unicode</td> <td data-bbox="1037 942 1349 1005">2 octets = 1 caractère</td> </tr> </table> <p>Pour les données Unicode, spécifiez uniquement un nombre d'octets pair. Indiquer un nombre d'octets impair peut entraîner l'affichage incorrect des caractères.</p>	Analytics non Unicode	1 octet = 1 caractère	Analytics Unicode, données ASCII étendu (ANSI)	1 octet = 1 caractère	Analytics Unicode, données Unicode	2 octets = 1 caractère
Analytics non Unicode	1 octet = 1 caractère						
Analytics Unicode, données ASCII étendu (ANSI)	1 octet = 1 caractère						
Analytics Unicode, données Unicode	2 octets = 1 caractère						
<i>décimales</i> optionnel	Nombre de décimales contenues par les champs numériques						
<i>format_date</i> optionnel	<p>Format de date dans les champs des données sources.</p> <p>Pour les champs de type DateHeure ou heure, utilisez PIC <i>format</i>. Vous pouvez également utiliser PIC <i>format</i> pour les champs de date.</p> <p>Si les données sources comprennent des séparateurs comme des barres obliques, vous devez les inclure dans <i>format_date</i>. Par exemple, si les données source sont 12/31/2014, vous devez saisir la date au format MM/JJ/AAAA. N'entourez pas <i>format_date</i> de guillemets.</p>						
NDATETIME optionnel	<p>Les valeurs de date, DateHeure ou heure stockées dans un champ numérique sont traitées comme des données DateHeure.</p> <p>NDATETIME requiert que vous indiquiez également le format de la DateHeure source à l'aide de PIC <i>format</i>.</p>						
PIC <i>format</i> optionnel	<p><b>Remarque</b></p> <p>S'applique aux champs numériques ou DateHeure.</p> <ul style="list-style-type: none"> <li>○ <b>champs numériques</b> : format d'affichage des valeurs numériques dans les vues et rapports Analytics</li> </ul>						

Nom	Description
	<ul style="list-style-type: none"> <li>◦ <b>champs DateHeure</b> : format physique des valeurs DateHeure dans les données source (ordre des caractères date et heure, séparateurs, etc.)</li> </ul> <p><b>Remarque</b></p> <p>Pour les champs DateHeure, le <i>format</i> doit correspondre exactement au format physique dans les données source. Par exemple, si les données source sont 12/31/2014, vous devez saisir la date au format "MM/JJ/AAAA".</p> <p>Le <i>format</i> doit être placé entre guillemets.</p>
AS <i>nom_affichage</i> optionnel	<p>Nom d'affichage (titre de la colonne) pour le champ dans la vue. Si vous souhaitez que le nom d'affichage soit identique au champ nom, n'utilisez pas AS.</p> <p>Indiquez le <i>nom_affichage</i> sous forme de chaîne entre guillemets. Utilisez un point-virgule (;) entre les mots si vous souhaitez insérer un saut de ligne dans le titre de la colonne.</p>
WIDTH <i>caractères</i> optionnel	<p>Largeur d'affichage du champ en nombre de caractères.</p> <p>La valeur indiquée contrôle la largeur d'affichage du champ dans les vues et rapports Analytics. La largeur d'affichage n'altère jamais les données ; elle peut toutefois masquer des données si elle est plus courte que la longueur du champ.</p> <p>La largeur d'affichage ne peut pas être inférieure à la longueur de <i>nom_champ</i> ou de <i>nom_affichage</i>.</p> <p>Si vous ignorez WIDTH, la largeur d'affichage est définie sur la longueur du champ en caractères.</p> <p><b>Remarque</b></p> <p>Les caractères spécifiés par WIDTH sont des caractères à largeur fixe. Le même espace est attribué à chaque caractère, quelle que soit la longueur du caractère réel.</p> <p>Par défaut, les vues dans Analytics utilisent une largeur proportionnelle qui ne correspond pas à l'espacement des caractères à largeur fixe.</p> <p>Si vous souhaitez une correspondance un à un entre la valeur WIDTH et les caractères de la vue, vous pouvez modifier le paramètre <b>Police proportionnelle</b> dans la boîte de dialogue <b>Options</b> sur une police à chasse fixe comme Courier New.</p>
SUPPRESS optionnel	<p>S'applique uniquement aux champs numériques.</p> <p>Supprime la totalisation automatique d'un champ numérique dans les rapports Analytics.</p> <p>La totalisation de certains champs numériques n'est pas appropriée : par exemple, un champ coût unitaire ou un champ taux de remise.</p>
<i>note_champ</i> optionnel	<p>Commentaire de champ qui est ajouté à la définition de champ dans le format de table.</p> <p><i>note_champ</i> doit être le dernier paramètre, après tous les autres paramètres obligatoires et facultatifs. Le texte ne peut pas être contenu sur plusieurs lignes. Les guillemets ne sont pas obligatoires.</p>



# Exemples

## Définition d'un champ de type caractère

Définit un champ de type caractère appelé **ProdDesc**. Le titre de la colonne dans la vue est **Description produit**.

### Analytics non Unicode

- Commence à : octet 12 (position de caractère 12)
- Longueur : 24 octets (24 caractères)

```
DEFINE FIELD ProdDesc ASCII 12 24 AS "Description produit"
```

### Analytics Unicode, données ASCII étendu (ANSI)

- Commence à : octet 12
- Longueur : 24 octets (24 caractères)

```
DEFINE FIELD ProdDesc ASCII 12 24 AS "Description produit"
```

### Analytics Unicode, données Unicode

- Commence à : octet 13
- Longueur : 48 octets (24 caractères)

```
DEFINE FIELD ProdDesc UNICODE 13 48 AS "Description produit"
```

## Définition d'un champ numérique

Définit un champ numérique appelé **Quantité disponible**. Dans la vue, la colonne utilise le format d'affichage spécifié, et le titre est **Quantité disponible**.

- Commence à : octet 61
- Longueur : 10 octets
- Places décimales : aucune

```
DEFINE FIELD QtyOH NUMERIC 61 10 0 PIC "(9.999.999)" AS "Quantité disponible"
```

## Définition d'un champ DateHeure à partir de données de type caractères

Dans les données caractères source, les deux premiers exemples ci-dessous définissent un champ DateHeure appelé **Date\_transaction**. Dans les données source, le format de date est JJ/MM/AAAA. Aucun

titre de colonne n'est spécifié ; par conséquent, le titre de colonne utilise par défaut le nom du champ.

- Commence à : octet 20
- Longueur : 10 octets

Ici, le format de date est spécifié à l'aide de *format\_date* :

```
DEFINE FIELD Date_transaction DATETIME 20 10 JJ/MM/AAAA
```

Ici, le format de date est spécifié à l'aide de PIC *format* :

```
DEFINE FIELD Date_transaction DATETIME 20 10 PIC "DD/MM/YYYY"
```

Lors de la définition des champs DateHeure incluant les données heure, vous devez utiliser PIC *format*. L'exemple suivant définit un champ DateHeure appelé **e-mail\_horodatage**. Dans les données source, le format DateHeure est AAAA/MM/JJ hh:mm:ss-hh:mm.

- Commence à : octet 1
- Longueur : 25 octets

```
DEFINE FIELD e-mail_horodatage DATETIME 1 25 PIC "YYYY/MM/DD hh:mm:ss-hh:mm"
```

## Définition d'un champ DateHeure à partir de données numériques

À partir des données numériques source, définit un champ DateHeure appelé **Reçu\_horodatage** ayant le format de DateHeure spécifié dans les données source.

- Commence à : octet 15
- Longueur : 15 octets

```
DEFINE FIELD Reçu_horodatage DATETIME 15 15 PIC "YYYYMMDD,hhmmss"
```

## Définition d'un champ DateHeure « numérique »

À partir des données numériques source, définit un champ numérique appelé **Reçu\_horodatage** ayant le format de DateHeure spécifié dans les données source.

Le paramètre NDATEIME permet aux valeurs de DateHeure stockées dans le champ numérique d'être traitées comme des données de DateHeure par Analytics.

- Commence à : octet 15
- Longueur : 15 octets
- Places décimales : 6

```
DEFINE FIELD Reçu_horodatage PRINT 15 15 6 NDATEIME PIC "YYYYMMDD.hhmmss"
```

## Définir un champ de données physique qui lit des données compressées

## de l'ordinateur central

Vous pouvez utiliser l'option `NDATETIME` pour créer un champ de données physique qui lit les valeurs de date d'un champ numérique compressé.

Analytics ne peut pas reconnaître une date dans un nombre dont le taux de compression est inférieur à un octet par chiffre et qui n'affiche pas de format de date. Par conséquent, vous devez « décompresser » la valeur avec `NDATETIME` afin d'obtenir le nombre total de chiffres, puis spécifier le format de date à l'aide de `PIC`.

Pour indiquer précisément quels chiffres représentent le jour, le mois et l'année, vous indiquez le même format de date que celui dans le format d'enregistrement compressé :

```
DEFINE FIELD nom_champ_date NUMERIC 1 8 0 NDATETIME PIC "AAAAMMJJ"
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Écraser des champs dans un script

Il est possible d'écraser un champ dans un format de table en définissant un champ utilisant le même nom que le champ existant. Si `SET SAFETY` est sur `ON`, une boîte de dialogue de confirmation apparaît avant d'écraser le champ existant.

Pour éviter d'interrompre un script, vous pouvez définir `SET SAFETY` sur `OFF`. Le champ existant est écrasé sans confirmation supplémentaire.

## Types de données prises en charge

Catégorie de données	Type de données
Caractère	ASCII
	CUSTOM
	EBCDIC
	NOTE
	PCASCII
	UNICODE

Catégorie de données	Type de données
Numérique	ACCPAC
	ACL
	BASIC
	BINARY
	FLOAT
	HALFBYTE
	IBMFLOAT
	MICRO
	NUMERIC
	PACKED
	PRINT
	UNISYS
	UNSIGNED
	VAXFLOAT
ZONED	
DateHeure	DATETIME
Logique	LOGICAL

# Commande DEFINE FIELD ... COMPUTED

Définit un champ calculé dans un format de table Analytics.

## Syntaxe

Pour définir un champ calculé :

```
DEFINE FIELD nom_champ COMPUTED expression
```

Pour définir un champ calculé avec des paramètres facultatifs :

```
DEFINE FIELD nom_champ COMPUTED  
<IF test> <STATIC> <PIC format> <AS nom_affichage> <WIDTH caractères> <SUPPRESS>  
<note_champ>  
expression
```

Pour définir un champ calculé conditionnel :

```
DEFINE FIELD nom_champ COMPUTED  
*** BLANK_LINE ***  
valeur IF condition  
<valeur IF condition>  
<...n>  
valeur_par_défaut
```

Pour définir un champ calculé conditionnel avec des paramètres facultatifs :

```
DEFINE FIELD nom_champ COMPUTED  
<IF test> <STATIC> <PIC format> <AS nom_affichage> <WIDTH caractères> <SUPPRESS>  
<note_champ>  
valeur IF condition  
<valeur IF condition>  
<...n>  
valeur_par_défaut
```

**Remarque**

La syntaxe multiligne doit présenter exactement la même structure que la syntaxe illustrée dans la syntaxe générique ci-dessus et dans les exemples ci-dessous.

## Paramètres

Nom	Description
<i>nom_champ</i>	<p>Le nom du champ calculé.</p> <p><b>Remarque</b></p> <p>La longueur des noms des champs ne doit pas dépasser 256 caractères alphanumériques majuscules et minuscules. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p> <p>Un certain nombre de mots-clés réservés ne peuvent pas être utilisés comme noms de champs dans Analytics. Pour consulter la liste complète, voir "Mots-clés réservés" Page 985.</p>
<i>expression</i>	Une expression Analytics valide qui définit la valeur du champ calculé.
IF <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.</p> <p><b>Remarque</b></p> <p>La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p>
STATIC optionnel	<p>Le champ affiche la même valeur sur chaque ligne de la table jusqu'à ce qu'une nouvelle valeur soit rencontrée.</p> <p>Par exemple, les données source contiennent un champ Nom de famille dont :</p> <ul style="list-style-type: none"> <li>○ le premier enregistrement affiche la valeur « Smith »</li> <li>○ les cinq enregistrements suivants affichent des lignes vides</li> <li>○ le septième enregistrement affiche la valeur « Wong »</li> </ul> <p>Dans ce cas, « Smith » s'affiche sur six lignes consécutives, puis « Wong » s'affiche sur la septième ligne.</p>
PIC <i>format</i> optionnel	<p><b>Remarque</b></p> <p>S'applique aux champs numériques uniquement.</p> <p>Format d'affichage des valeurs numériques dans les vues et rapports Analytics.</p> <p>Le <i>format</i> doit être placé entre guillemets.</p>
AS <i>nom_affichage</i> optionnel	<p>Nom d'affichage (titre de la colonne) pour le champ dans la vue. Si vous souhaitez que le nom d'affichage soit identique au champ nom, n'utilisez pas AS.</p> <p>Indiquez le <i>nom_affichage</i> sous forme de chaîne entre guillemets. Utilisez un point-virgule (;) entre les mots si vous souhaitez insérer un saut de ligne dans le titre de la</p>

Nom	Description
	colonne.
WIDTH <i>caractères</i> optionnel	<p>Largeur d'affichage du champ en nombre de caractères.</p> <p>La valeur indiquée contrôle la largeur d'affichage du champ dans les vues et rapports Analytics. La largeur d'affichage n'altère jamais les données ; elle peut toutefois masquer des données si elle est plus courte que la longueur du champ.</p> <p>La largeur d'affichage ne peut pas être inférieure à la longueur de <i>nom_champ</i> ou de <i>nom_affichage</i>.</p> <p>Si vous ignorez WIDTH, la largeur d'affichage est définie sur la longueur du champ en caractères.</p> <p><b>Remarque</b></p> <p>Les caractères spécifiés par WIDTH sont des caractères à largeur fixe. Le même espace est attribué à chaque caractère, quelle que soit la longueur du caractère réel.</p> <p>Par défaut, les vues dans Analytics utilisent une largeur proportionnelle qui ne correspond pas à l'espacement des caractères à largeur fixe.</p> <p>Si vous souhaitez une correspondance un à un entre la valeur WIDTH et les caractères de la vue, vous pouvez modifier le paramètre <b>Police proportionnelle</b> dans la boîte de dialogue <b>Options</b> sur une police à chasse fixe comme Courier New.</p>
SUPPRESS optionnel	<p>S'applique aux champs numériques uniquement.</p> <p>Supprime la totalisation automatique des champs calculés numériques dans les rapports Analytics.</p> <p>La totalisation de certains champs numériques n'est pas appropriée : par exemple, un champ coût unitaire ou un champ taux de remise.</p>
<i>note_champ</i> optionnel	<p>Commentaire de champ qui est ajouté à la définition de champ dans le format de table.</p> <p><i>note_champ</i> doit être le dernier paramètre, après tous les autres paramètres obligatoires et facultatifs. Le texte ne peut pas être contenu sur plusieurs lignes. Les guillemets ne sont pas obligatoires.</p>
<i>valeur IF condition</i>	<p>Champ calculé conditionnel uniquement.</p> <ul style="list-style-type: none"> <li>○ <b>valeur</b> : la valeur ou l'expression du champ calculé à utiliser lorsque la <i>condition</i> indiquée est vraie</li> <li>○ <b>condition</b> : le test logique qui est évalué</li> </ul>
<i>valeur_par_défaut</i>	<p>Champ calculé conditionnel uniquement.</p> <p>La valeur ou l'expression à utiliser dans le champ calculé si aucune condition évaluée n'est vraie.</p>

Nom	Description
	<p><b>Remarque</b></p> <p>La précision décimale de toutes les valeurs calculées numériques est déterminée par la précision de <i>valeur_défait</i>. Par exemple, si vous indiquez une valeur par défaut de 0,00, toutes les valeurs calculées sont déterminées avec deux décimales, et sont arrondies si nécessaire. Pour une plus grande précision, augmentez le nombre de décimales dans <i>valeur_défait</i>.</p>

## Exemples

### Définition d'un champ calculé

Vous définissez un champ calculé appelé **Valeur** en tant que produit des champs **Coût** et **Quantité** :

```
DEFINE FIELD Valeur COMPUTED Coût * Quantité
```

### Définition d'un champ calculé avec des options

Vous définissez un champ calculé nommé **Valeur\_03**, où plusieurs options sont définies. Vous incluez une condition IF qui limite les enregistrements à traiter par le champ calculé :

```
DEFINE FIELD Valeur_03 COMPUTED
IF Product_Class = "03" PIC "($9.999.999,99)" AS "Valeur Prod Class 3" La valeur est le coût multiplié par la quantité
Coût * Quantité
```

### Définir un champ calculé conditionnel

Vous définissez un champ calculé conditionnel nommé **Taxe\_ventes** qui calcule une autre taxe sur les ventes selon l'état d'exécution de la transaction. Les transactions exécutées en dehors de ces trois états présentent une taxe sur les ventes par défaut de 0,00 \$.

#### Remarque

La deuxième ligne doit rester vide, car il n'y a pas de paramètres facultatifs.

```
DEFINE FIELD Taxe_ventes COMPUTED

0,0750 * Montant_vente IF Etat = "CA"
0,0400 * Montant_vente IF Etat = "NY"
```



```
0,0625 * Montant_vente IF Etat = "TX"
0,00
```

## Définir un champ calculé conditionnel avec des options

Vous définissez un champ calculé conditionnel nommé **Taxe\_ventes\_100** qui calcule une autre taxe sur les ventes selon l'état d'exécution de la transaction. Le champ calcule uniquement la taxe sur des montants supérieurs ou égaux à 100 \$.

Les transactions exécutées en dehors de ces trois états présentent une taxe sur les ventes par défaut de 0,00 \$.

### Remarque

Lorsque vous spécifiez des paramètres facultatifs, ne laissez pas de lignes vides.

```
DEFINE FIELD Taxe_ventes_100 COMPUTED
IF Montant_vente >= 100
0,0750 * Montant_vente IF Etat = "CA"
0,0400 * Montant_vente IF Etat = "NY"
0,0625 * Montant_vente IF Etat = "TX"
0,00
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Deux types de champs calculés

Il existe deux types de champs calculés :

- **champ calculé standard**

Un champ calculé standard effectue le même calcul sur chaque enregistrement d'une table.

Par exemple, dans une table Inventaire, vous pourriez créer un champ calculé qui multiplie la valeur du champ Coût par la valeur du champ Quantité pour calculer la Valeur d'inventaire au Coût pour chaque enregistrement.

- **champ calculé conditionnel**

Un champ calculé conditionnel est capable d'effectuer différents calculs sur les enregistrements d'une table, à partir d'un ensemble de conditions que vous spécifiez. Le calcul effectué sur un enregistrement dépend de la condition remplie par l'enregistrement.

Par exemple, dans une table Transactions, vous pourriez créer un champ calculé conditionnel qui calcule la TVA à l'aide d'un taux ajusté selon l'état où a été effectuée la transaction. Des conditions comme IF Etat = "CA" et IF Etat = "NY" testeraient chaque enregistrement pour identifier quel taux de TVA utiliser.

## Instructions pour la création d'un champ calculé conditionnel

### Remarque

Lorsque vous définissez un champ calculé conditionnel, si vous n'indiquez pas de paramètres facultatifs sur la deuxième ligne, vous devez la laisser vide.

Outre une valeur par défaut, les champs calculés conditionnels nécessitent au moins une valeur conditionnelle. Vous devez utiliser la syntaxe multiligne suivante pour définir un champ calculé conditionnel :

- les paramètres facultatifs s'affichent sur la deuxième ligne ;
- s'il n'y a pas de paramètres facultatifs, la deuxième ligne doit rester vide ;
- la première instruction de condition s'affiche sur la troisième ligne ;
- chaque instruction de condition supplémentaire nécessite une ligne séparée ;
- la valeur par défaut s'affiche sur la dernière ligne.

## Écrasement des définitions de champ

Il est possible d'écraser une définition de champ dans un format de table en définissant un champ utilisant le même nom que le champ existant.

Si SET SAFETY est sur ON, Analytics affiche une boîte de dialogue de confirmation avant d'écraser le champ existant. Pour éviter l'interruption d'un script, vous pouvez définir SET SAFETY sur OFF, et Analytics remplacera le champ existant sans demander de confirmation.

# Commande DEFINE RELATION

Définit une relation entre deux tables Analytics.

## Remarque

Vous pouvez associer jusqu'à 18 tables Analytics, accéder aux données de toute combinaison de champs dans les tables associées et les analyser comme si elles existaient dans une seule et même table. Vous devez indiquer une commande DEFINE RELATION distincte pour chaque paire de tables associées.

## Syntaxe

```
DEFINE RELATION champ_clé WITH nom_table_associée INDEX nom_index <AS nom_relation>
```

## Paramètres

Nom	Description
<i>champ_clé</i>	<p>Champ clé de la table parent.</p> <p>Vous ne pouvez sélectionner qu'un seul champ clé par relation.</p> <p><b>Remarque</b></p> <p>Lors de la création de relations entre les tables parents et les tables petits-enfants, vous devez indiquer nom de champ clé complet au format <i>nom_table.nom_champ</i>.</p> <p>Dans "Associer trois tables" Page suivante, consultez la section : <a href="#">Justificatifs créés par</a></p>
WITH <i>nom_table_associée</i>	Nom de la table associée.
INDEX <i>nom_index</i>	<p>Nom de l'index du champ clé de la table associée.</p> <p>Vous devez indexer la table associée sur le champ clé avant d'associer la table.</p>
AS <i>nom_relation</i> optionnel	<p>Nom unique pour la relation.</p> <p>Par défaut, le nom de la table enfant est utilisé comme nom de relation. Si vous définissez des relations supplémentaires avec la même table enfant, vous devez spécifier un nom unique.</p>

## Exemples

### Associer deux tables

Dans l'exemple suivant, la table ouverte est associée à la table **Client** en utilisant le champ de numéro client (**CustNum**) en tant que champ clé :

```
DEFINE RELATION CustNum WITH Client INDEX Client_sur_CustNum
```

**Client\_sur\_CustNum** est le nom de l'index de table enfant sur le champ clé. Un index de table enfant est requis lorsque vous associez des tables.

Si l'index de table enfant n'existe pas déjà lorsque vous exécutez la commande DEFINE RELATION, un message d'erreur apparaît et la relation n'est pas effectuée.

### Astuce

Si vous définissez une relation dans l'interface utilisateur Analytics, l'index de table enfant est créé automatiquement pour vous.

## Créer un index de table enfant avant d'associer deux tables

Si besoin, vous pouvez créer un index de table enfant juste avant d'associer deux tables. L'exemple ci-dessous montre la création d'un index sur la table enfant **Client** avant d'associer la table **Ar** à la table **Client**.

```
OPEN Client
INDEX ON CustNum TO Client_sur_CustNum
Ouvrir Ar
DEFINE RELATION CustNum WITH Client INDEX Client_sur_CustNum
```

## Associer trois tables

Dans l'exemple suivant, trois tables sont associées dans l'exemple de projet **ACL\_Rockwood.ACL** :

- **éléments\_justificatifs** : table parent
- **Justificatifs** : table enfant
- **Employés** : table petit-enfant

En utilisant la table **Justificatifs** comme table intermédiaire dans la relation, vous pouvez associer chaque élément justificatif à l'employé l'ayant traité.

```
OPEN Justificatifs
INDEX ON numéro_justificatif TO "Justificatifs_par_numéro_justificatif"
OPEN éléments_justificatifs
DEFINE RELATION numéro_justificatif WITH Justificatifs INDEX Justificatifs_par_numéro_justificatif
OPEN Employés
INDEX ON numéro_employé TO "Employés_par_numéro_employé"
OPEN éléments_justificatifs
DEFINE RELATION Justificatifs.créés_par WITH Employés INDEX Employés_par_numéro_employé
```

## Explication de la logique de la syntaxe

1. Ouvrez la table **Justificatifs** et indexez-la sur le champ **numéro\_justificatif**.
2. Ouvrez la table **éléments\_preuves** et associez-la à la table **Justificatifs** à l'aide du champ clé **numéro\_justificatif**.
3. Ouvrez la table **Employés** et indexez-la sur le champ **numéro\_employé**.
4. Ouvrez la table **éléments\_justificatifs** et associez-la à la table **Employés** à l'aide du champ clé **Justificatifs.créés\_par**.

### Remarque

**Justificatifs.créés\_par** est disponible comme champ clé dans la deuxième relation car vous avez déjà associé **éléments\_preuves** et **Justificatifs** dans la première relation.

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

# Commande DEFINE REPORT

Crée une vue ou ouvre une vue existante.

## Syntaxe

```
DEFINE REPORT nom_vue
```

## Paramètres

Nom	Description
<i>nom_vue</i>	Nom d'une nouvelle vue ou d'une vue existante. <ul style="list-style-type: none"><li>◦ <b>nouvelle vue</b> : crée une vue vide avec le nom spécifié dans la table ouverte Les espaces dans le nom de la vue sont remplacés par des caractères de soulignement.</li><li>◦ <b>vue existante</b> : ouvre la vue indiquée dans la table ouverte</li></ul>

## Exemples

### Création d'une nouvelle vue

Vous créez une nouvelle table nommée **examen\_T4\_AR** :

```
DEFINE REPORT examen_T4_AR
```

# Commande DEFINE TABLE DB

Définit une table de serveur Analytics en se connectant à une table de base de données à l'aide du Connecteur AX. Vous pouvez vous connecter à une base de données Microsoft SQL Server, Oracle ou DB2.

## Syntaxe

```
DEFINE TABLE DB {SOURCE profil_base_données <PASSWORD num> <PASSWORD num> |
SERVER profil_serveur <PASSWORD num>} <FORMAT nom_format> SCHEMA schéma <TITLED
nom_table_act> <PRIMARY|SECONDARY> DBTABLE nom_table_db FIELDS {noms_champs|ALL}
<...n> <WHERE condition> <ORDER noms_champ>
```

## Paramètres

<p>SOURCE <i>profil_base_données</i></p>	<p>Profil de base de données Analytics à utiliser pour accéder au moteur de base de données.</p> <p>Les profils de bases de données incluent les informations requises pour se connecter au moteur de base de données, par exemple :</p> <ul style="list-style-type: none"> <li>○ une référence au profil du serveur associé ;</li> <li>○ le type de base de données ;</li> <li>○ le nom de la base de données ;</li> <li>○ les informations du compte utilisateur.</li> </ul> <p><b>Remarque</b></p> <p>DEFINE TABLE DB prend en charge les connexions vers les bases de données suivantes uniquement : Microsoft SQL Server, Oracle ou DB2.</p>
<p>PASSWO- RD <i>num</i> optionnel</p>	<p>Définition de mot de passe à utiliser.</p> <p>Vous n'utilisez pas PASSWORD <i>num</i> pour inviter à saisir ou pour spécifier un mot de passe réel. La définition de mot de passe renvoie à un mot de passe fourni ou défini auparavant à l'aide de la commande PASSWORD, de la commande SET PASSWORD ou de la balise d'outil d'analyse PASSWORD.</p> <p><i>num</i> correspond au numéro de la définition de mot de passe. Par exemple, si deux mots de passe ont été fournis ou définis auparavant dans un script, ou lors de la planification d'un outil d'analyse, PASSWORD 2 indique que le mot de passe n° 2 est utilisé.</p> <p>Pour plus d'informations sur la fourniture ou définition de mots de passe, consultez les rubriques :</p> <ul style="list-style-type: none"> <li>• "Commande PASSWORD" Page 373</li> <li>• "Commande SET" Page 435</li> <li>• <a href="#">Balise d'outil d'analyse PASSWORD</a></li> </ul> <p>Pour plus d'informations sur la fourniture ou définition de mots de passe, consultez les rubriques :</p> <ul style="list-style-type: none"> <li>• <a href="#">Commande PASSWORD</a></li> <li>• <a href="#">Commande SET</a></li> <li>• <a href="#">Balise d'outil d'analyse PASSWORD</a></li> </ul> <p>Le mot de passe est uniquement requis si le profil de base de données ne contient aucun mot de passe</p>

	enregistré. Utilisez PASSWORD deux fois après le mot-clé SOURCE. Le premier mot de passe vous connecte au serveur et le deuxième à la base de données.
SERVER <i>profil_serveur</i>	N'est plus utilisé. Avant la version 10.0 d'Analytics, était utilisé pour la connexion à ACL Édition Serveur pour z/OS. À compter de la version 10.0 d'Analytics, ACL Édition Serveur pour z/OS n'est plus inclus.
FORMAT <i>nom_format</i> optionnel	Le nom d'une table Analytics ou d'un fichier de format de table (.layout) avec le format de table que vous souhaitez utiliser.
SCHEMA <i>schéma</i>	Schéma auquel se connecter. Vous devez placer le nom de schéma entre guillemets.
TITLED <i>nom_table_acl</i> optionnel	Nom de la table Analytics à créer. <i>nom_table_acl</i> doit être une chaîne entre guillemets. Si vous ignorez TITLED, Analytics utilise le nom de table de la base de données. Lorsque vous accédez à plusieurs tables à la fois, Analytics utilise le nom de la première table.
PRIMARY   SECONDARY optionnel	Utilisez la table en tant que table principale ou table secondaire dans les commandes multifichiers. Si aucune option n'est indiquée, la valeur par défaut PRIMARY est utilisée.
DBTABLE <i>table_base_données</i>	Table de base de données à laquelle vous souhaitez accéder. <i>table_base_données</i> doit être une chaîne entre guillemets.
FIELDS <i>noms_champs</i>   ALL	<p>Les champs à inclure dans la sortie :</p> <ul style="list-style-type: none"> <li>◦ <b>FIELDS <i>noms_champ</i></b> : utilisez les champs spécifiés <i>noms_champ</i> doit être une chaîne entre guillemets.</li> <li>◦ <b>ALL</b> : utilisez tous les champs dans la table</li> </ul> <p>Pour utiliser les champs à partir de plusieurs tables :</p> <ol style="list-style-type: none"> <li>a. Saisissez le nom de la première table suivi par les champs de cette table.</li> <li>b. Saisissez le nom de la table suivante suivi par les champs de cette table.</li> <li>c. Pour chaque table supplémentaire, répétez l'étape b.</li> </ol> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <pre>DBTABLE "DSN1310" FIELDS "Champ_A Champ_B Champ_C" DBTABLE "DSN2516" FIELDS "Champ_L Champ_M Champ_N"</pre> </div> <p><b>Remarque</b></p> <p>Le Connecteur AX permet d'accéder à un nombre illimité de tables associées, mais nous recommandons de limiter ce nombre à cinq. La durée de traitement augmente lorsque vous accédez à plusieurs tables.</p>



WHERE <i>condition</i> optionnel	<p>Une instruction SQL WHERE qui limite les données aux enregistrements qui remplissent la condition indiquée.</p> <p>Vous devez utiliser une syntaxe SQL valide entrée en tant que chaîne entre guillemets.</p> <p>Lorsque vous joignez des tables, Analytics affiche la condition de jonction dans la clause WHERE :</p> <pre>"Table_1.First_name = Table_2.First_name"</pre>
ORDER <i>noms_</i> <i>champ</i> optionnel	<p>Les champs que le moteur de base de données utilise pour trier les enregistrements. <i>noms_champ</i> doit être une chaîne entre guillemets.</p> <p>L'exécution de la commande demande plus de temps lors du tri des enregistrements. Utilisez ORDER uniquement lorsque le tri est important.</p>

## Exemples

### Exemple

Vous souhaitez accéder aux données à partir d'une base de données Microsoft SQL Server via le Connecteur AX. Pour ce faire, vous utilisez la commande DEFINE TABLE DB. Vous incluez le paramètre SOURCE pour vous connecter au Connecteur AX via un profil de base de données :

```
DEFINE TABLE DB SOURCE "Audit_ServeurSQL" SCHEMA "RH" TITLED "Paie" DBTABLE "RH.employé" FIELDS "IDemployé" DBTABLE "RH.HistoriquePaieEmployé" FIELDS "Taux FréquencePaie" WHERE "RH.employé.IDemployé=RH.HistoriquePaieEmployé.IDemployé"
```

## Remarques

### Fonctionnement

La table de serveur Analytics est définie en tant que requête utilisant un profil de base de données pour se connecter à une table de base de données.

## Supprimer la partie correspondant à l'heure des valeurs DateHeure

Préfacez la commande DEFINE TABLE DB avec la commande SET SUPPRESSTIME pour supprimer la portion correspondant à l'heure des valeurs DateHeure.

L'utilisation de SET SUPPRESSTIME ON est destinée aux scripts antérieurs à la version 10.0 d'Analytics qui supposent que la portion heure des valeurs DateHeure sera tronquée. Si SET SUPPRESSTIME ON n'est pas ajoutée à ces scripts, ils ne pourront pas s'exécuter dans la version avec DateHeure activée d'Analytics.

Pour plus d'informations, consultez la rubrique "SET SUPPRESSTIME" dans "Commande SET"  
Page 435.

# Commande DEFINE VIEW

Définit une nouvelle vue ou remplace une vue existante.

## Syntaxe

```
DEFINE VIEW nom_vue <RLINES n> <ALL> <SUPPRESS> <SUMMARIZED> <IF test> <WHILE test> <HEADER texte_en-tête> <FOOTER texte_pied_page> <TO nom_fichier_rapport <HTML>> <OK>
```

## Paramètres

Nom	Description
<i>nom_vue</i>	Nom de la vue à créer ou à remplacer.  <b>Remarque :</b> La longueur des noms de vues est limitée à 64 caractères alpha-numériques. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.
RLINES <i>n</i> optionnel	Interligne utilisé pour les enregistrements de détail dans les vues et les rapports. Les lignes de détail sont formatées par défaut avec un interligne simple.
ALL optionnel	Tous les champs dans le format de table Analytics actif sont ajoutés à la vue.
SUPPRESS optionnel	Supprime les lignes de détail vides dans les rapports générés dans la vue. Lors de la génération du rapport, les lignes de détail vides sont ignorées dans la sortie. Cette option s'applique aux rapports basés sur des vues multilignes.
SUMMARIZED optionnel	Indique que les rapports générés dans la vue doivent inclure les sous-totaux et les totaux, mais pas les lignes de détail.  Les sous-totaux sont générés en fonction des champs de fractionnement définis dans la vue. Si cette option n'est pas sélectionnée, le rapport contient les lignes de détail et les sous-totaux de chaque champ de fractionnement spécifié.
IF <i>test</i> optionnel	Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.

Nom	Description
	<p><b>Remarque</b></p> <p>La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p>
WHILE <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.</p> <p><b>Remarque</b></p> <p>Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.</p>
HEADER <i>texte_en-tête</i> optionnel	<p>Texte à insérer en haut de chaque page d'un rapport.</p> <p><i>texte_en-tête</i> doit être indiqué sous forme de chaîne entre guillemets. La valeur remplace la variable système Analytics HEADER.</p>
FOOTER <i>texte_pied_page</i> optionnel	<p>Texte à insérer en bas de chaque page d'un rapport.</p> <p><i>texte_pied_page</i> doit être indiqué sous forme de chaîne entre guillemets. La valeur remplace la variable système Analytics FOOTER.</p>
TO <i>nom_fichier_rapport</i> HTML optionnel	<p>Nom de fichier et type des rapports créés dans cette vue.</p> <p>Utilisez le mot-clé HTML pour enregistrer les rapports générés dans cette vue en tant que fichiers HTML (.htm). Par défaut, les rapports générés sont rendus en tant que fichiers texte ASCII.</p>
OK optionnel	<p>Supprime ou écrase des éléments sans demander de confirmation.</p>

## Exemples

### Création d'une vue

Vous ouvrez la table **Ar** et créez une vue appelée **Rapport\_AR**, qui inclut tous les champs dans le format de table :

```
OPEN Comptes_clients
DEFINE VIEW Rapport_AR HEADER "Rapport AR" ALL OK
```

# Commande DELETE

Supprime un élément de projet Analytics, un champ d'un format de table, une variable, une ou plusieurs entrées d'historique de la table, une relation entre des tables ou un fichier dans un dossier Windows. Supprime aussi une colonne d'une vue.

## Syntaxe

Objectif	Syntaxe
Pour supprimer un élément de projet Analytics	<code>DELETE <i>type_élément nom_élément</i> &lt;OK&gt;</code>
Pour supprimer un champ d'un format de table	<code>DELETE <i>nom_champ</i> &lt;OK&gt;</code>
Pour supprimer une colonne d'une vue	<code>DELETE COLUMN <i>nom_vue nom_champ</i> &lt;ALL&gt; &lt;OK&gt;</code>
Pour supprimer une ou toutes les variables	<code>DELETE {<i>nom_variable</i> ALL} &lt;OK&gt;</code>
Pour supprimer l'historique de la table Analytics active	<code>DELETE HISTORY &lt;<i>conserver_entrées_historique</i>&gt; &lt;OK&gt;</code>
Pour supprimer une relation entre deux tables	<code>DELETE RELATION &lt;<i>nom_table_enfant</i> <i>nom_relation</i>&gt; &lt;OK&gt;</code>
Pour supprimer un fichier	<code>DELETE <i>nom_fichier</i> &lt;OK&gt;</code>
Pour supprimer toutes les notes d'enregistrement, et le champ <b>NoteEnregistrement</b> généré automatiquement, de la table ouverte	<code>DELETE NOTES &lt;OK&gt;</code>

## Paramètres

Nom	Description
<i>type_élément nom_élé-</i>	Type et nom de l'élément à supprimer.

Nom	Description
<i>ment</i>	<p>Indiquez l'un des types d'élément suivants :</p> <ul style="list-style-type: none"> <li>○ <b>FOLDER</b> : le dossier de projet indiqué et l'intégralité de son contenu</li> <li>○ <b>FORMAT</b> : le format de table spécifiée, l'ensemble de ses vues et ses index et relations associés</li> </ul> <p>Tous les autres formats de table concernant la table associée sont conservés.</p> <p>Le fichier de données (.fil) associé au format de table n'est pas supprimé si l'option <b>Supprimer le fichier de données avec la table</b> n'est pas sélectionnée dans l'onglet <b>Table</b> de la boîte de dialogue <b>Options (Outils &gt;Options)</b>.</p> <p>Vous pouvez également utiliser la commande SET DELETE_FILE {ON OFF} dans un script ou dans la ligne de commande pour activer ou désactiver cette option. Pour plus d'informations, consultez la section "Commande SET" Page 435.</p> <p><b>Attention</b></p> <p>Redoublez de prudence lorsque vous activez l'option <b>Supprimer le fichier de données avec la table</b>. Il se peut que ce soit le fichier de données d'origine qui soit supprimé avec le format de table !</p> <p>Les fichiers de données sont supprimés de manière définitive. Ils ne sont pas envoyés dans la Corbeille de Windows.</p> <ul style="list-style-type: none"> <li>○ <b>REPORT</b> : la vue spécifiée</li> </ul> <p>Vous ne pouvez pas supprimer une vue si elle est active.</p> <ul style="list-style-type: none"> <li>○ <b>COLUMN</b> : la colonne spécifiée</li> <li>○ <b>SCRIPT (ou BATCH)</b> : le script indiqué</li> <li>○ <b>WORKSPACE</b> : l'environnement de travail spécifié</li> <li>○ <b>INDEX</b> : l'index spécifié</li> <li>○ <b>NOTES</b> : toutes les notes d'enregistrement de la table ouverte, et le champ <b>NoteEnregistrement</b> du format de table</li> </ul>
<i>nom_champ</i> ALL	<p><b>Supprimer un champ</b></p> <p>Nom du champ à supprimer dans le format de table Analytics actif.</p> <p>Vous pouvez supprimer un champ d'un format de table, même si le champ est inclus dans la vue actuelle.</p> <p><b>Remarque</b></p> <p>Vous ne pouvez pas supprimer un champ référencé par un champ calculé, sauf si vous supprimez d'abord le champ calculé.</p> <p><b>Supprimer une colonne</b></p> <p>Nom de la colonne qui doit être supprimée dans la vue indiquée.</p> <p><b>Remarque</b></p> <p>Utilisez le nom de champ physique, pas le nom d'affichage de la colonne.</p> <ul style="list-style-type: none"> <li>○ <b>ALL inclus</b> : supprime toutes les occurrences de la colonne dans la vue</li> <li>○ <b>ALL ignoré</b> : supprimer la première (tout à gauche) occurrence de la colonne de la vue</li> </ul>
<i>nom_vue</i>	Nom de la vue dans laquelle une colonne doit être supprimée.

Nom	Description
<code>nom_variable</code>   ALL	<p>Nom de la variable à supprimer. Utilisez ALL pour supprimer toutes les variables.</p> <p>Si vous indiquez ALL, toutes les occurrences des types de variables suivants sont supprimées du projet :</p> <ul style="list-style-type: none"> <li>variables système</li> <li>variables temporaires définies par l'utilisateur</li> <li>variables permanentes définies par l'utilisateur</li> </ul> <p><b>Remarque</b></p> <p>Vous ne pouvez pas supprimer une variable référencée par un champ calculé, sauf si vous supprimez d'abord le champ calculé.</p>
HISTORY <code>conserver_entrées_historique</code>	<p>Supprime toutes les entrées d'historique de la table, à l'exception du nombre d'entrées les plus récentes indiqué par <code>conserver_entrées_historique</code>.</p> <p>Omettez <code>conserver_entrées_historique</code> pour supprimer toutes les entrées.</p>
RELATION <code>nom_table_enfant</code>   <code>nom_relation</code> >	<p>Supprime toute relation qui n'a aucune relation associée ni aucun champ associé référencé dans la vue active ou dans un champ calculé actif.</p> <p>Utilisez les options pour indiquer quelle relation supprimer :</p> <ul style="list-style-type: none"> <li><b><code>nom_table_enfant</code></b> : à utiliser si la relation n'a pas de nom précis (nom par défaut lorsqu'une relation est créée)</li> <li><b><code>nom_relation</code></b> : à utiliser si la relation a été spécifiquement nommée lors de sa création. Sinon, utilisez <code>nom_table_enfant</code></li> </ul> <p>Si vous n'utilisez pas l'une de ces options, la dernière relation ayant été définie est supprimée.</p>
<code>nom_fichier</code>	<p>Nom d'un champ physique à supprimer.</p> <p>Vous pouvez indiquer un chemin absolu ou relatif vers un fichier que vous souhaitez supprimer. Si le chemin d'accès contient des espaces, placez-le entre guillemets doubles.</p>
OK optionnel	Supprime les éléments sans présenter de boîte de dialogue de confirmation.

## Exemples

### Suppression d'un champ date

Vous supprimez le champ **Date** du format de table associé à la table **Ar** :

```
OPEN Comptes_clients
DELETE Date
```

### Suppression de plusieurs colonnes d'une vue

Vous supprimez deux colonnes de la vue **Rapport\_AR** associée à la table **Ar**. Vous spécifiez OK pour les

deux commandes DELETE afin de ne pas afficher d'invite de confirmation lors de l'exécution du script :

```
OPEN Comptes_clients  
DELETE COLUMN Rapport_AR Date OK  
DELETE COLUMN Rapport_AR Date_Facture OK
```



# Commande DIALOG

Crée une boîte de dialogue personnalisée invitant de manière interactive les utilisateurs à saisir une ou plusieurs valeurs d'introduction de script. Chaque valeur d'introduction est stockée dans une variable nommée.

## Remarque

L'utilisation de la commande DIALOG pour saisir des mots de passe n'est pas sécurisée. Il faudrait utiliser la "Commande PASSWORD" Page 373 à la place.

La commande DIALOG n'est pas prise en charge par les outils d'analyse d'AX Serveur.

Vous pouvez créer une boîte de dialogue interactive de base avec la "Commande ACCEPT" Page 54.

## Astuce

La façon la plus simple de créer des boîtes de dialogue personnalisées est d'utiliser le **Générateur de boîte de dialogue**. Pour plus d'informations, consultez la section [Création de boîtes de dialogue personnalisées](#).

## Syntaxe

```
DIALOG (DIALOG TITLE texte_titre WIDTH pixels HEIGHT pixels) (BUTTONSET TITLE "&OK;&Annuler" AT pos_x pos_y <WIDTH pixels> <HEIGHT pixels> DEFAULT num_élément <HORZ>) <[syntaxe_libellé][syntaxe_zone_texte][syntaxe_case_cocher][syntaxe_bouton_option][syntaxe_liste_déroulante][syntaxe_liste_élément_projet]> <...n>
```

```
syntaxe_libellé ::=
(TEXT TITLE texte_titre AT pos_x pos_y <WIDTH pixels> <HEIGHT pixels> <CENTER|RIGHT>)
```

```
syntaxe_zone_texte ::=
(EDIT TO nom_var AT pos_x pos_y <WIDTH pixels> <HEIGHT pixels> <DEFAULT chaîne>)
```

```
syntaxe_case_cocher ::=
(CHECKBOX TITLE texte_titre TO nom_var AT pos_x pos_y <WIDTH pixels> <HEIGHT pixels> <CHECKED>)
```

```
syntaxe_bouton_option ::=
(RADIOBUTTON TITLE liste_valeur TO nom_var AT pos_x pos_y <WIDTH pixels> <HEIGHT pixels> <DEFAULT num_élément> <HORZ>)
```

```

syntaxe_liste_déroulante ::=
(DROPDOWN TITLE liste_valeur TO nom_var AT pos_x pos_y <WIDTH pixels> <HEIGHT pixels>
<DEFAULT num_élément>)

```

```

syntaxe_liste_élément_projet ::=
(ITEM TITLE catégorie_élément_projet TO nom_var AT pos_x pos_y <WIDTH pixels> <HEIGHT pixels>
<DEFAULT chaîne>)

```

# Paramètres

## Paramètres généraux

Nom	Description
DIALOG TITLE <i>texte_titre</i>	Crée la boîte de dialogue principale et le titre de la boîte de dialogue. <i>texte_titre</i> doit être indiqué sous forme de chaîne entre guillemets.
BUTTONSET TITLE "&OK;&Annuler"	Les libellés des boutons <b>OK</b> et <b>Annuler</b> dans la boîte de dialogue. Généralement, les valeurs ne doivent pas être modifiées, mais, le cas échéant, assurez-vous que la valeur positive précède la valeur négative. Par exemple : "&Oui;&Non"
WIDTH <i>pixels</i>	Largeur du contrôle individuel, ou largeur de la boîte de dialogue (si indiquée), du contrôle DIALOG. La valeur est indiquée en pixels. Si aucune valeur n'est indiquée pour un contrôle, la largeur est calculée en fonction de la valeur la plus longue contenue par le contrôle.
HEIGHT <i>pixels</i>	Hauteur du contrôle individuel, ou hauteur de la boîte de dialogue (si indiquée), du contrôle DIALOG. La valeur est indiquée en pixels.
AT <i>pos_xpos_y</i>	Emplacement du coin supérieur gauche du contrôle dans la boîte de dialogue personnalisée. <ul style="list-style-type: none"> <li><i>pos_x</i> est la distance horizontale en pixels à partir du côté gauche de la boîte de dialogue.</li> <li><i>pos_y</i> est la distance verticale en pixels à partir du haut de la boîte de dialogue.</li> </ul>
DEFAULT <i>num_élément</i>	Valeur numérique correspondant à la valeur BUTTONSET que vous souhaitez sélectionner en tant qu'élément par défaut. Par exemple, si les valeurs BUTTONSET sont « &OK;&Annuler », indiquez DEFAULT 1 pour sélectionner OK par défaut.
HORZ optionnel	Affiche les valeurs du contrôle BUTTONSET à l'horizontale. Les valeurs s'affichent verticalement par défaut.

### Remarque

Pour la plupart des types de contrôle, la commande DIALOG crée une variable pour stocker l'entrée utilisateur. Vous ne pouvez pas utiliser des caractères qui n'existent pas en anglais, tels que é, dans les noms de variables utilisées pour substituer des variables. Les noms de variables contenant des caractères n'existant pas en anglais provoqueront l'échec du script.

Par défaut, certaines des variables DIALOG sont créées en tant que variables de type caractère. Si vous utilisez une variable de type caractère pour stocker des valeurs de type numérique ou DateHeure, vous devez convertir la variable dans le type de données requis dans le traitement suivant du script. Pour plus d'informations, consultez la section "Type de données d'entrée" Page 159.

## Paramètres de libellé

Nom	Description
TEXT	Crée un libellé de texte pour identifier, informer ou donner une instruction.
TITLE <i>texte_titre</i>	Libellé du contrôle. <i>texte_titre</i> doit être indiqué sous forme de chaîne entre guillemets.
CENTER   RIGHT optionnel	Alignement du texte dans le contrôle. Si vous ignorez CENTER ou RIGHT, l'alignement à gauche est utilisé par défaut.

## Paramètres de zone de texte

Nom	Description
EDIT	Crée une zone de texte pour une saisie de l'utilisateur.
TO <i>nom_var</i>	Nom de la variable de type caractère qui stocke la saisie spécifiée par l'utilisateur. Si la variable existe déjà, la valeur spécifiée est attribuée. Si la variable n'existe pas, elle est créée, puis la valeur spécifiée est attribuée.
DEFAULT <i>chaîne</i> optionnel	Chaîne de texte par défaut à afficher dans le contrôle. <i>chaîne</i> doit être indiquée sous forme de chaîne entre guillemets.

## Paramètres de case à cocher

Nom	Description
CHECKBOX	Crée une case à cocher pour présenter une option à l'utilisateur.

Nom	Description
TITLE <i>texte_titre</i>	Libellé du contrôle. <i>texte_titre</i> doit être indiqué sous forme de chaîne entre guillemets.
TO <i>nom_var</i>	Nom de la variable logique qui stocke la valeur Vrai ou Faux spécifiée par l'utilisateur. Si la variable existe déjà, la valeur spécifiée est attribuée. Si la variable n'existe pas, elle est créée, puis la valeur spécifiée est attribuée.
CHECKED optionnel	Définit le contrôle à activer par défaut.

## Paramètres de case d'option

Nom	Description
RADIOBUTTON	Crée des cases d'option pour présenter à l'utilisateur des options s'excluant mutuellement.
TITLE <i>liste_valeur</i>	La liste de valeurs affichées pour le contrôle. Les valeurs doivent être indiquées sous forme de chaîne entre guillemets. Séparez chaque valeur par un point-virgule (;).
TO <i>nom_var</i>	Le nom de la variable numérique qui stocke la position numérique de la valeur de la case d'option sélectionnée par l'utilisateur. Si la variable existe déjà, la valeur spécifiée est attribuée. Si la variable n'existe pas, elle est créée, puis la valeur spécifiée est attribuée.
DEFAULT <i>num_élément</i> optionnel	Valeur numérique correspondant à l'élément de liste que vous souhaitez sélectionner en tant qu'élément par défaut. Par exemple, si la <i>liste_valeur</i> est "Rouge;Vert;Bleu", indiquez DEFAULT 2 pour sélectionner Vert par défaut.
HORZ optionnel	Affiche les valeurs du contrôle à l'horizontale. Les valeurs s'affichent verticalement par défaut.

## Paramètres de la liste déroulante

Nom	Description
DROPDOWN	Crée une liste déroulante pour présenter une liste d'options à l'utilisateur.
TITLE <i>liste_valeur</i>	La liste de valeurs affichées pour le contrôle. Les valeurs doivent être indiquées sous forme de chaîne entre guillemets. Séparez chaque valeur par un point-virgule (;).

Nom	Description
TO <i>nom_var</i>	Nom de la variable de type caractère qui stocke la valeur de liste déroulante sélectionnée par l'utilisateur.  Si la variable existe déjà, la valeur spécifiée est attribuée. Si la variable n'existe pas, elle est créée, puis la valeur spécifiée est attribuée.
DEFAULT <i>num_élément</i> optionnel	Valeur numérique correspondant à l'élément de liste que vous souhaitez sélectionner en tant qu'élément par défaut.  Par exemple, si la <i>liste_valeur</i> est "Rouge;Vert;Bleu", indiquez DEFAULT 2 pour sélectionner Vert par défaut lorsque la liste déroulante s'affiche.

## Paramètres de liste d'éléments de projet

Nom	Description
ITEM	Crée une liste d'éléments de projet pour présenter à l'utilisateur une liste d'éléments de projet Analytics, comme des champs.
TITLE <i>catégorie_élément_projet</i>	Catégorie d'élément de projet à inclure dans le contrôle.  Vous pouvez indiquer une ou plusieurs catégories. L'utilisateur peut sélectionner une valeur unique dans la liste des éléments du projet.  Entourez <i>catégorie_élément_projet</i> de guillemets, sans espace ni signe de ponctuation entre les catégories.  Pour obtenir les codes que vous avez utilisés pour spécifier des catégories, reportez-vous à la section "Codes pour les catégories d'élément de projet" Page suivante.  <b>Remarque</b> Ne mélangez pas des catégories différentes dans le même contrôle ITEM, sauf si vous avez une raison de le faire. Par exemple, ne mélangez pas les tables et les champs. La liste d'éléments de projet ainsi créée va peut-être mettre l'utilisateur en difficulté.
TO <i>nom_var</i>	Nom de la variable de type caractère qui stocke le nom de l'élément de projet sélectionné par l'utilisateur.  Si la variable existe déjà, la valeur spécifiée est attribuée. Si la variable n'existe pas, elle est créée, puis la valeur spécifiée est attribuée.
DEFAULT <i>chaîne</i> optionnel	Nom exact de l'élément de projet que vous souhaitez sélectionner par défaut.  <i>chaîne</i> doit être indiquée sous forme de chaîne entre guillemets.

## Exemples

### Inviter l'utilisateur à utiliser une table et un script

Dans votre script, vous devez inviter l'utilisateur à sélectionner la table et le script Analytics à utiliser pour exécuter une analyse.

Vous indiquez que la table **Inventaire\_Metaphore\_2012** du projet `ACL_Demo.ac1` est sélectionnée par défaut comme table Analytics, mais que l'utilisateur peut sélectionner n'importe quelle table dans le projet.

Le script à exécuter doit également être sélectionné dans la liste de scripts du projet Analytics :

```
DIALOG (DIALOG TITLE "Analyse inventaire" WIDTH 500 HEIGHT 200 )(BUTTONSET TITLE
"&OK;&Annuler" AT 370 12 DEFAULT 1 ) (TEXT TITLE "Choisissez les éléments de projet Analytics
à analyser." AT 50 16 ) (TEXT TITLE "Table:" AT 50 50 ) (ITEM TITLE "f" TO "v_table" AT 50 70
DEFAULT "Inventaire_Metaphore_2012" ) (TEXT TITLE "Script :" AT 230 50 ) (ITEM TITLE "b" TO
"v_script" AT 230 70 )
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Interactivité

La commande DIALOG permet de créer un script interactif. Pendant le traitement de la commande DIALOG, le script s'interrompt et une boîte de dialogue invite l'utilisateur à saisir des entrées utilisées par Analytics lors d'un traitement ultérieur.

Vous pouvez créer des boîtes de dialogue indépendantes qui invitent l'utilisateur à entrer un seul élément à la fois ou bien créer une seule boîte de dialogue qui invite à entrer plusieurs éléments.

## ACCEPT et DIALOG

La commande ACCEPT vous permet de créer une boîte de dialogue interactive de base qui peut avoir un ou plusieurs des types de contrôles suivants :

- zone de texte
- liste de projets

Pour une interactivité de base, ACCEPT peut suffire. Pour plus d'informations, consultez la section "Commande ACCEPT" Page 54.

## Codes pour les catégories d'élément de projet

Utilisez les codes suivants pour indiquer la catégorie d'élément de projet à afficher dans une liste d'éléments de projet.

## Catégories de projet

Code	Catégorie
f	Tables
Md(s)	Scripts
i	Index
r	Vues et rapports
w	Environnements_de_travail

## Catégories de champ

Code	Catégorie
C	Champs de type caractère
N	Champs numériques
D	Champs de type DateHeure
L	Champs logiques

## Catégories de variable

Code	Catégorie
c	Variables de type caractère
n	Variables numériques
d	Variables de type DateHeure
l	Variables logiques

## Type de données d'entrée

Certains des contrôles de la commande DIALOG stockent l'entrée de l'utilisateur dans des variables de type caractère. Si une saisie numérique ou de type DateHeure est requise, vous pouvez utiliser les fonctions VALUE() ou CTOD() pour convertir le contenu d'une variable de type caractère en valeur numérique ou en valeur DateHeure :

```
SET FILTER TO BETWEEN(%v_champ_date%; CTOD(%v_date_début%); CTOD(%v_date_fin%))
```

Dans l'exemple, les dates de début et de fin pour ce filtre sont stockées sous type de valeurs caractère. Elles doivent être converties en valeurs date afin d'être utilisées avec un champ date utilisant un type de données DateHeure.

Entourer le nom de la variable par des symboles de pourcentage (%) remplace la valeur de type caractère contenue par la variable pour le nom de la variable. La fonction CTOD( ) convertit alors la valeur de type caractère en une valeur de date.

## Place de la commande DIALOG

Dans la mesure du possible, il est recommandé de placer toutes les commandes DIALOG au début d'un script. Si vous demandez la saisie de toutes les données au début, le script peut s'exécuter sans entrave une fois les informations requises saisies par l'utilisateur.

### Remarque

Vous ne pouvez pas utiliser la commande DIALOG dans la commande GROUP.



# Commande DIRECTORY

Génère une liste de fichiers et de dossiers dans le répertoire spécifié.

## Syntaxe

```
DIRECTORY <spéc_fichier> <SUPPRESS> <SUBDIRECTORY> <APPEND> <TO nom_
table|nom_fichier>
```

## Paramètres

Nom	Description
<i>spéc_fichier</i> optionnel	<p>Dossier Windows ou fichiers à afficher et pour lesquels les informations doivent être affichées.</p> <p>Vous pouvez utiliser le caractère générique astérisque (*) pour obtenir les noms de tous les fichiers qui ont une extension spécifique, de tous les fichiers qui commencent par une chaîne particulière, ou de tous les fichiers d'un répertoire. Par exemple :</p> <ul style="list-style-type: none"> <li>◦ *.fil : répertorie tous les noms de fichiers avec l'extension .fil (fichiers de données Analytics)</li> <li>◦ Inv*.* : inclut tous les fichiers commençant par « Inv », quelle que soit l'extension de fichier.</li> <li>◦ Résultats\* ou Résultats\*.* permet d'obtenir les noms de tous les fichiers du répertoire Résultats.</li> </ul> <p>Pour restreindre le nombre de fichiers répertoriés dans un dossier particulier, vous pouvez spécifier un chemin d'accès relativement au dossier de projet Analytics ou spécifier un chemin d'accès complet. Par exemple :</p> <ul style="list-style-type: none"> <li>◦ Résultats\*.* : affiche le contenu du sous-répertoire Résultats dans le dossier du projet Analytics</li> <li>◦ C:\ACL Data\Résultats\*.* : affiche le contenu du dossier spécifié</li> </ul> <p><b>Remarque</b></p> <p>Le caractère générique ne peut pas être utilisé dans les niveaux intermédiaires d'un chemin de fichier spécifié. Il peut être uniquement utilisé au dernier niveau du chemin, comme indiqué ci-dessus.</p> <p>Les chemins d'accès ou noms de fichiers contenant des espaces doivent être entourés de doubles guillemets.</p> <p>Si vous utilisez <i>spéc_fichier</i>, ce paramètre doit être placé avant n'importe quel autre paramètre. Si <i>spéc_fichier</i> s'affiche à une autre place, la commande DIRECTORY n'est pas traitée et une erreur est générée.</p> <p>Si vous ignorez <i>spéc_fichier</i>, vous obtenez les noms de tous les fichiers du dossier contenant le projet Analytics. Vous ne pouvez utiliser aucun des autres paramètres si vous ignorez <i>spéc_fichier</i>.</p>

Nom	Description
SUPPRESS optionnel	Supprime les informations sur le chemin d'accès dans la sortie, en laissant uniquement les noms de fichier et les propriétés.
SUBDIRECTORY optionnel	Inclut le contenu des sous-répertoires dans la liste des fichiers du répertoire.  Par exemple, si <i>spéc_fichier</i> indique Résultats\*.fil, les fichiers .fil sont recherchés dans le dossier Résultats et dans tous les sous-répertoires du dossier Résultats.  Selon le nombre de sous-répertoires et de fichiers à rechercher, l'utilisation de SUBDIRECTORY peut ralentir l'analyse des sous-répertoires. Analytics affiche une boîte de dialogue qui indique la progression de la commande.
APPEND optionnel	Ajoute la sortie de commande à la fin d'un fichier existant au lieu de remplacer ce fichier.  <b>Remarque</b>  Vous devez vous assurer que la structure de la sortie de la commande et du fichier existant est identique : <ul style="list-style-type: none"> <li>• les mêmes champs</li> <li>• le même ordre des champs</li> <li>• les champs correspondants ont la même longueur</li> <li>• les champs correspondants ont le même type de données</li> </ul> Analytics ajoute la sortie à un fichier existant sans tenir compte de sa structure, ce qui peut désordonner les données si la structure de la sortie et du fichier existant ne correspond pas.
TO <i>nom_table</i>   <i>nom_fichier</i> optionnel	Emplacement vers lequel envoyer les résultats de la commande : <ul style="list-style-type: none"> <li>◦ <b>nom_table</b> : enregistre les résultats dans une table Analytics  Indiquez <i>nom_table</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple : TO "Sortie.FIL"  Par défaut, le fichier de données de table (.FIL) est enregistré dans le dossier contenant le projet Analytics.  Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant : <ul style="list-style-type: none"> <li>• TO "C:\Sortie.FIL"</li> <li>• TO "Résultats\Sortie.FIL"</li> </ul> <b>Remarque</b>  La longueur des noms des tables est limitée à 64 caractères alphanumériques, sans l'extension .FIL. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</li> </ul> <ul style="list-style-type: none"> <li>◦ <b>nom_fichier</b> : enregistre les résultats dans un fichier  Indiquez <i>nom_fichier</i> sous forme de chaîne entre guillemets avec l'extension de fichier appropriée. Par exemple : TO "Sortie.TXT"  Par défaut, le fichier est enregistré dans le dossier contenant le projet Analytics.  Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier dans un autre dossier existant : <ul style="list-style-type: none"> <li>• TO "C:\Sortie.TXT"</li> </ul> </li> </ul>

Nom	Description
	<ul style="list-style-type: none"> <li>• TO "Résultats\Sortie.TXT"</li> </ul> <p>Si vous ignorez TO, la liste des fichiers du répertoire s'affiche dans la zone d'affichage Analytics.</p>

## Exemples

### Différentes options pour dresser la liste des fichiers

Il est utile d'avoir une liste de fichiers pour des enquêtes ponctuelles et pour les intégrer à un script.

Un certain nombre d'options différentes permettant de dresser une liste de fichiers avec la commande DIRECTORY apparaît ci-dessous.

#### Lister tous les fichiers

Liste tous les fichiers du dossier contenant le projet Analytics :

```
DIRECTORY
```

#### Lister tous les fichiers d'un type spécifique

Liste tous les fichiers .fil (fichiers de données Analytics) du dossier contenant le projet Analytics :

```
DIRECTORY *.fil
```

#### Utiliser des caractères génériques pour lister des fichiers

Liste tous les noms de fichiers commençant par "Inv" dans le dossier contenant le projet Analytics :

```
DIRECTORY Inv*.*
```

#### Lister tous les fichiers d'un sous-répertoire relatif au dossier de projet Analytics

Liste tous les fichiers du sous-répertoire `Résultats` dans le dossier contenant le projet Analytics :

```
DIRECTORY "Résultats\*"
```

#### Lister tous les fichiers dans un dossier spécifié

Liste tous les fichiers dans le sous-répertoire `Résultats` :

```
DIRECTORY "C:\ACL Data\Résultats\*"
```

## Lister tous les fichiers d'un type spécifique à un emplacement indiqué

Liste tous les fichiers .fil (fichiers de données Analytics) du dossier spécifié ainsi que tous les sous-répertoires :

```
DIRECTORY "C:\ACL Data\Résultats\*.fil" SUBDIRECTORY
```

## Lister tous les fichiers dans un dossier spécifié et sortir la liste dans une table Analytics

Liste tous les fichiers du dossier `Résultats` et sort la liste dans une table Analytics dans le dossier contenant le projet Analytics :

```
DIRECTORY "C:\ACL Data\Résultats\*" TO Contenu_Dossier_Résultats.fil
```

La nouvelle table `Contenu_Dossier_Résultats` est ajoutée au projet ouvert.

## Lister tous les fichiers dans un dossier et sortir la liste dans une table Analytics d'un autre dossier

Liste tous les fichiers du dossier `ACL Data\Résultats` et sort la liste dans une table Analytics dans le dossier `GL Audit 2014\Résultats` :

```
DIRECTORY "C:\ACL Data\Résultats\*" TO "C:\ACL Projects\GL Audit 2014\Résultats\Contenu_Dossier_Résultats.fil"
```

La nouvelle table `Contenu_Dossier_Résultats` est ajoutée au projet ouvert. Le fichier de données associé (`Contenu_Dossier_Résultats.fil`) est créé dans le dossier de sortie spécifié, qui peut être ou non le dossier contenant le projet Analytics.

# Remarques

## Propriétés affichées par DIRECTORY

La commande DIRECTORY est semblable à la commande DIR de Windows. En plus d'obtenir les noms des fichiers et des sous-répertoires dans un dossier, la commande DIRECTORY permet également d'afficher les propriétés de fichiers et de répertoires suivantes :

<ul style="list-style-type: none"> <li>○ Taille du fichier</li> <li>○ Attributs</li> </ul>	<ul style="list-style-type: none"> <li>○ Date de création</li> <li>○ Heure de création</li> </ul>	<ul style="list-style-type: none"> <li>○ Date d'accès</li> <li>○ Heure d'accès</li> </ul>	<ul style="list-style-type: none"> <li>○ Date de modification</li> <li>○ Heure de modification</li> <li>○ le nombre total de fichiers et de réper-</li> </ul>
--	---	---	---

			toires qui correspondent aux critères spécifiés
--	--	--	---

## Utilisations de DIRECTORY dans un script

Lorsqu'elle est utilisée dans un script, la commande DIRECTORY donne la possibilité d'examiner le système de fichiers. Par exemple, vous pourriez utiliser DIRECTORY en association avec d'autres commandes pour détecter la présence ou l'absence de fichiers, vérifier la taille d'un fichier ou prendre des décisions en fonction d'autres propriétés de fichiers.

## Sortie des résultats de DIRECTORY

Vous pouvez exécuter la commande à partir de la ligne de commande pour afficher une liste de fichiers contenus dans un répertoire à l'écran, ou sauvegarder la liste dans une table Analytics ou un fichier .txt.

## Comment ouvrir les résultats de table de DIRECTORY

La commande DIRECTORY n'inclut pas le paramètre OPEN. Si vous utilisez la commande dans un script et que vous envoyez les résultats vers une table Analytics, et que vous souhaitez ouvrir la table créée, suivez la commande DIRECTORY de la commande OPEN. Par exemple :

```
DIRECTORY "C:\ACL Data\Résultats\*" TO Contenu_Dossier_Résultats.fil
OPEN Contenu_Dossier_Résultats
```

# Commande DISPLAY

Affiche des informations sur le type d'élément Analytics spécifié. Peut également afficher le résultat d'une expression ou la sortie d'une fonction.

## Syntaxe et paramètres

Syntaxe	Objectif
DISPLAY	Affiche les définitions des champs et toutes les tables enfant associées pour la table Analytics active
DISPLAY OPEN	Affiche une liste des tables et fichiers de projet Analytics ouverts. <ul style="list-style-type: none"> <li>◦ <b>tables Analytics</b> : affiche le nom du fichier de données source, pas le nom du format de table.</li> <li>◦ <b>mode table multiple</b> : le fichier de données source identifié comme PRIMARY est associé à la table actuellement active.</li> <li>◦ <b>tables liées</b> : si la table mère est ouverte, affiche le fichier de données source pour les tables mère et enfant, même si la table enfant n'est pas ouverte dans l'onglet Afficher.</li> </ul>
DISPLAY {<PRIMARY> SECONDARY}	Affiche le nom et les informations de format de table pour la table principale ou secondaire. <ul style="list-style-type: none"> <li>◦ <b>PRIMARY (ou aucun mot-clé spécifié)</b> : afficher des informations pour la table active</li> <li>◦ <b>SECONDARY</b> : afficher des informations pour la table secondaire.</li> </ul> <p>En mode table multiple, SECONDARY renvoie à la table secondaire associée à la table actuellement active.</p> <p>Les informations affichées incluent :</p> <ul style="list-style-type: none"> <li>◦ le nom du format de table</li> <li>◦ le nom du fichier de données source</li> <li>◦ les relations entre la table et les autres tables</li> <li>◦ les informations de définition de champ du format de table</li> </ul>
DISPLAY HISTORY	Affiche l'historique de la table Analytics actuellement active. <p><b>Remarque</b></p> <p>Une table peut ou non avoir un historique de tables associé.</p>
DISPLAY RELATION	Affiche les informations de relation pour la table Analytics active : <ul style="list-style-type: none"> <li>◦ les noms des tables enfants</li> <li>◦ les noms des champs clés</li> <li>◦ les noms des index</li> </ul>

Syntaxe	Objectif
DISPLAY { <i>nom_variable</i>  VARIABLES}	<p>Affiche la valeur d'une seule variable ou de toutes les variables.</p> <ul style="list-style-type: none"> <li>◦ <b>nom_variable</b> : nom d'une seule variable dont afficher la valeur.</li> <li>◦ <b>VARIABLES</b> : afficher les valeurs actuelles de toutes les variables système et définies par l'utilisateur, et la mémoire disponible pour enregistrer les variables.</li> </ul>
DISPLAY VERSION	<p>Affiche des informations sous le format suivant concernant la version d'Analytics qui est installée :</p> <ul style="list-style-type: none"> <li>◦ <b>Version</b> : <i>numéro de version principale.numéro de version secondaire</i></li> <li>◦ <b>Correctif</b> : <i>numéro de correctif</i></li> <li>◦ <b>Type</b> : édition 000 (non-Unicode) ou 001 (Unicode) d'Analytics</li> <li>◦ <b>Build</b> : <i>numéro de build du logiciel</i></li> </ul>
DISPLAY {DATE TIME}	<p>Affiche la date et l'heure actuelles du système d'exploitation.</p> <p><b>DATE   TIME</b> : indiquez l'un ou l'autre mot-clé. Les deux mots-clés font la même chose.</p>
DISPLAY {FREE SPACE}	<p>Affiche la quantité de mémoire physique (RAM) disponible qui peut être utilisée par Analytics.</p> <p>La quantité affichée ne comprend pas la mémoire réservée aux variables. Par défaut, Analytics réserve 60 Ko de mémoire physique pour stocker les variables mais la quantité augmente automatiquement en fonction des besoins.</p> <p><b>FREE   SPACE</b> : indiquez l'un ou l'autre mot-clé. Les deux mots-clés font la même chose.</p>
DISPLAY <i>expression</i>	<p>Affiche le résultat d'une expression.</p> <p><b>expression</b> : l'expression dont afficher le résultat.</p>
DISPLAY <i>fonction</i>	<p>Affiche la sortie d'une fonction.</p> <p><b>fonction</b> : la fonction dont il faut afficher la sortie.</p>

## Exemples

### Afficher le format d'une table Analytics

Afficher le format d'une table peut s'avérer utile dans de nombreuses situations. Par exemple, vous voudrez peut-être combiner au moins deux tables et vous devez examiner les longueurs des champs et les types de données.

L'exemple ci-dessous affiche le format de la table `Trans_Cf` :

```
OPEN Trans_Cf
DISPLAY
```

La commande DISPLAY produit la sortie à l'écran affichée ci-dessous.

### Remarque

Si vous saisissez DISPLAY directement dans la ligne de commande Analytics, la sortie s'affiche immédiatement.

Si vous exécutez DISPLAY dans un script, double-cliquez sur l'entrée DISPLAY correspondante dans la trace des commandes pour afficher la sortie.

## Sortie vers l'écran

### Relation

'Fournisseur' associé par 'No\_Fournisseur' à l'aide de l'index 'Fournisseur\_sur\_No\_Fournisseur'

### Fichier

'Trans\_Cf.fil' (format 'Ap\_Trans') est votre fichier PRINCIPAL.

La longueur de l'enregistrement est 59

### Champs

Nom	Type	Début	Longueur	Décimales	Explication du champ
No_Fournisseur	ASCII	1	5		AS "Fournisseur;Numéro" WIDTH 7
No_Facture	ASCII	6	15		AS "Facture;Numéro"
Date_Facture	DATETIME	21	8		PICTURE "MM/DD/YY" AS "Facture;Date" WIDTH 8
Montant_Facture	NUMERIC	29	12	2	PICTURE "(9.999.999,99)" AS "Facture;Montant" WIDTH 12
NumProd	ASCII	41	9		AS "Produit;Numéro"
Quantité	MICRO	50	4	0	PICTURE "(9.999.999)"
Coût_unitaire	NUMERIC	54	6	2	PICTURE "(9.999.999)" AS "Unitaire;Coût" SUPPRESS

## Afficher les valeurs de toutes les variables dans un projet Analytics

DISPLAY VARIABLES génère les mêmes informations qui s'affichent dans l'onglet **Variables** du **Navigateur**. Un des avantages de l'utilisation de DISPLAY VARIABLES est que vous pouvez copier-coller les informations affichées.



L'exemple ci-dessous crée deux variables définies par l'utilisateur et deux variables système, puis affiche les valeurs des variables :

```
ASSIGN v_nom_table = "Trans_Cf"
ASSIGN v_nom_champ = "Montant_Facture"
OPEN %v_nom_table%
TOTAL FIELDS %v_nom_champ%
DISPLAY VARIABLES
```

La commande DISPLAY produit la sortie à l'écran affichée ci-dessous.

### Remarque

Si vous saisissez DISPLAY VARIABLES directement dans la ligne de commande Analytics, la sortie s'affiche immédiatement.

Si vous exécutez DISPLAY VARIABLES dans un script, double-cliquez sur l'entrée DISPLAY VARIABLES correspondante dans la trace des commandes pour afficher la sortie.

## Sortie vers l'écran

Nom	Type	Valeur
TOTAL1	N	278.641,33
OUTPUTFOLDER	C	"/Tables/Comptes fournisseurs"
v_nom_champ	C	"Montant_Facture"
v_nom_table	C	"Trans_Cf"

## Afficher le résultat d'une expression

Pour l'enregistrement sélectionné, l'exemple ci-dessous affiche le résultat apparaissant suite à la multiplication de la valeur dans le champ Prix\_vente par la valeur dans le champ Quantité\_disponible :

```
DISPLAY Prix_vente * Quantité_disponible
```

## Afficher la sortie d'une fonction

Pour l'enregistrement sélectionné, l'exemple ci-dessous affiche le nombre de jours s'étant écoulés depuis la date dans le champ Date\_facture :

```
DISPLAY AGE(Date_Facture)
```

# Remarques

## Emplacement des résultats de la commande

**DISPLAY exécuté à partir de la ligne de commande Analytics** : les résultats s'affichent à l'écran.

**DISPLAY exécuté dans un script** : les résultats sont écrits dans la trace de commande Analytics. Vous pouvez double-cliquer sur l'entrée dans la trace de commande pour afficher les résultats à l'écran.

# Commande DO REPORT

Génère le rapport Analytics indiqué.

## Syntaxe

```
DO REPORT nom_rapport
```

## Paramètres

Nom	Description
<i>nom_rapport</i>	Nom de la vue à générer et à imprimer sous forme de rapport.

## Exemple

### Impression de la vue par défaut

Vous ouvrez la table **Trans\_CF** et imprimez la vue par défaut :

```
OPEN Trans_CF
DO REPORT Vue_Défaut
```

## Remarques

### Exécution de DO REPORT dans la ligne de commande par rapport à un script

Les paramètres utilisés pour imprimer le rapport dépendent de l'endroit où vous exécutez la commande :

- **dans la ligne de commande** : la boîte de dialogue **Imprimer** s'ouvre pour que vous sélectionniez les pages à imprimer et pour configurer d'autres options pour le rapport
- **dans un script** : le rapport est imprimé immédiatement en fonction de ses paramètres par défaut

# Commande DO SCRIPT

Exécute un script secondaire, ou un script externe, depuis un script Analytics.

## Syntaxe

```
DO <SCRIPT> nom_script{<IF test>|<WHILE test>}
```

## Paramètres

Nom	Description
SCRIPT <i>nom_script</i>	<p>Nom du script à exécuter. Vous pouvez exécuter des scripts secondaires dans le projet Analytics ou des scripts externes stockés dans des fichiers texte portant des extensions comme .aclscript, .txt ou .bat.</p> <p>Vous pouvez spécifier un chemin d'accès au fichier d'un script externe. Vous devez entourer le chemin de guillemets s'il contient des espaces.</p> <p><b>Remarque</b></p> <p>Vous ne pouvez pas appeler un script qui est en cours d'exécution. Par exemple, si le ScriptA appelle le ScriptB, le ScriptB ne peut pas appeler le ScriptA. Le ScriptA est en cours d'exécution alors qu'il attend la fin du ScriptB.</p>
IF <i>test</i> optionnel	<p>Expression conditionnelle qui est évaluée une fois afin de déterminer si le script doit être exécuté. Si la condition évaluée est vraie, le script est exécuté. Dans le cas contraire, il ne l'est pas.</p> <p>Ne peut pas être utilisé avec WHILE dans la même commande. Si les deux sont utilisés, WHILE est ignoré lors du traitement du script. Un commentaire est saisi dans la trace, mais l'exécution du script n'est pas arrêtée.</p>
WHILE <i>test</i> optionnel	<p>Expression conditionnelle évaluée après l'exécution du script afin de déterminer si le script doit être de nouveau exécuté. Si la condition est vraie, le script est exécuté. Dans le cas contraire, il ne l'est pas.</p> <p><b>Remarque</b></p> <p>Si vous utilisez WHILE, assurez-vous que votre test finit par être évalué sur faux. Si cela n'est pas le cas, le script entre dans une boucle infinie. En cas de boucle infinie, appuyez sur la touche <b>Échap</b> pour annuler le traitement du script.</p> <p>Ne peut pas être utilisé avec IF dans la même commande. Si les deux sont utilisés, WHILE est ignoré lors du traitement du script. Un commentaire est saisi dans la trace, mais l'exécution du script n'est pas arrêtée.</p>

# Exemples

## Exécution d'un sous-script de manière répétée jusqu'à la validation de l'entrée

Vous avez un sous-script qui recueille les saisies de l'utilisateur à l'aide d'une boîte de dialogue. Il fait ce qui suit :

1. Invite l'utilisateur à saisir les valeurs requises.
2. Vérifie la saisie de l'utilisateur.
3. Définit la variable *v\_validée* sur vrai lorsque les valeurs de saisie sont validées.

Pour vous assurer que l'utilisateur saisisse une entrée valide, vous utilisez DO SCRIPT et vous incluez une condition WHILE pour que le script répète cette commande jusqu'à la validation de la saisie. Une fois la valeur de la variable modifiée, le script principal passe à la commande suivante :

```
DO SCRIPT ObtenirEntréeUtilisateur WHILE v_validée = F
```

## Exécution d'un sous-script à partir d'un emplacement partagé

Vous conservez les sous-scripts des utilitaires dans un emplacement partagé. Lorsque vous avez besoin de l'un d'entre eux pendant une analyse, vous y faites référence à l'aide du chemin complet vers l'emplacement partagé :

```
DO SCRIPT "C:\Mes scripts utilitaires\ObtenirEntréeUtilisateur.acscript" WHILE v_validée = F
```

# Remarques

## Commandes associées

DO SCRIPT est équivalente à la commande DO BATCH figurant dans les scripts créés avec les versions antérieures d'Analytics.

Vous ne pouvez pas inclure la commande DO SCRIPT dans une commande GROUP.

## Utilité d'un script externe

Il peut être utile de stocker un script en externe et de l'appeler depuis un script Analytics si vous souhaitez réutiliser le même sous-script dans différents scripts et projets Analytics.

Vous pouvez stocker une seule copie du script à un seul emplacement et y apporter des mises à jour à un endroit, plutôt que de le conserver à plusieurs emplacements.

# Commande DUMP

Affiche le contenu d'un fichier ou l'enregistrement actuel dans les codages de caractère Hexadécimal, ASCII et EBCDIC.

## Remarque

Cette commande peut être saisie uniquement sur la ligne de commande. Elle ne peut pas être utilisée dans des scripts.

## Syntaxe

```
DUMP {RECORD|nom_fichier} <SKIP octets> <COLUMN octets> <HORIZONTAL>
```

## Paramètres

Nom	Description
RECORD	Affiche le contenu de l'enregistrement sélectionné. Requis si vous ne spécifiez pas de <i>nom_fichier</i> .
<i>nom_fichier</i>	Nom du fichier que vous souhaitez afficher. Requis si vous ne spécifiez pas RECORD.  <div style="border-left: 2px solid #0056b3; padding-left: 10px; margin-left: 20px;"> <p><b>Remarque</b></p> <p>Pour afficher les codages de caractères concernant une table Analytics, vous devez indiquer le nom du fichier de données source et l'extension de fichier. Par exemple : <code>Trans_Cf.fil</code></p> </div>
SKIP <i>octets</i> optionnel	Nombre d'octets à contourner avant le début du vidage. La valeur par défaut est 0.
COLUMN <i>octets</i> optionnel	Dans la sortie : largeur des colonnes en nombre d'octets.  <div style="border-left: 2px solid #0056b3; padding-left: 10px; margin-left: 20px;"> <p><b>Remarque</b></p> <p>Le nombre indiqué par <i>octets</i> renvoie aux octets contenus par l'enregistrement ou la table Analytics.</p> <p>Il se peut que les caractères codés dans la sortie n'aient pas de relation un-à-un avec les caractères dans la vue. Par exemple, l'encodage hexadécimal pour le nombre 1 est 31.</p> </div> <p>Le nombre par défaut est 16 octets pour chaque colonne dans un affichage vertical et 64 octets pour une seule colonne dans un affichage horizontal. Le nombre maximum d'octets que vous pouvez spécifier est 255.</p>

Nom	Description
HORIZONTAL optionnel	Affiche les encodages de caractères en lignes horizontales plutôt qu'en blocs verticaux parallèles (par défaut).

## Exemples

### Afficher l'encodage des caractères de la table Inventaire

L'exemple ci-dessous affiche l'encodage des caractères hexadécimaux, ASCII et EBCDIC des données dans la table Inventaire. La sortie est organisée en lignes horizontales (l'encodage hexadécimal utilise une ligne double). Chaque ligne représente 97 octets de données dans la table Analytics :

```
DUMP Inventaire.fil COLUMN 97 HORIZONTAL
```

# Commande DUPLICATES

Détecte si des valeurs dupliquées ou des enregistrements entièrement dupliqués existent dans une table Analytics.

## Syntaxe

```
DUPLICATES <ON> {champ_clé<D> <...n>|ALL} <OTHER champ<...n>|OTHER ALL>
<UNFORMATTED> <TO {SCREEN|nom_table|nom_fichier|PRINT}> <APPEND> <IF test>
<WHILE test> <FIRST plage|NEXT plage> <HEADER texte_en-tête> <FOOTER texte_pied_page>
<PRESORT> <OPEN> <LOCAL> <ISOLOCALE code_paramètres_régionaux>
```

## Paramètres

Nom	Description
ON <i>champ_clé</i> D <...n>   ALL	<p>Le ou les champs clés ou bien l'expression à tester pour la recherche de doublons.</p> <ul style="list-style-type: none"> <li>◦ <b>champ_clé</b> : utilisez le(s) champ(s) spécifié(s)</li> </ul> <p>Si vous effectuez le test en fonction de plusieurs champs, les enregistrements identifiés comme doublons nécessitent des valeurs identiques dans chaque champ spécifié.</p> <p>Incluez D pour trier le champ clé dans l'ordre décroissant. L'ordre de tri par défaut est croissant.</p> <ul style="list-style-type: none"> <li>◦ <b>ALL</b> : utilisez tous les champs de la table</li> </ul> <p>Si vous effectuez le test en fonction de tous les champs d'une table, les enregistrements identifiés comme doublons doivent être entièrement identiques.</p> <p>Un ordre de tri croissant est la seule option pour ALL.</p> <p><b>Remarque</b></p> <p>Les portions indéfinies des enregistrements ne sont pas testées.</p>
OTHER <i>champ</i> <...n>   OTHER ALL optionnel	<p>Un ou plusieurs champs supplémentaires à inclure dans la sortie.</p> <ul style="list-style-type: none"> <li>◦ <b>champ &lt;...n&gt;</b> : incluez le(s) champ(s) spécifié(s)</li> <li>◦ <b>ALL</b> : incluez tous les champs de la table qui ne sont pas spécifiés comme champs clés</li> </ul>
UNFORMATTED optionnel	<p>Supprime les en-têtes et les sauts de page lorsque les résultats sont renvoyés dans un fichier.</p>
TO SCREEN   <i>nom_table</i>   <i>nom_fichier</i>   PRINT	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li>◦ <b>SCREEN</b> : affiche les résultats dans la zone d'affichage d'Analytics.</li> </ul>



Nom	Description
optionnel	<ul style="list-style-type: none"> <li>◦ <b>nom_table</b> : enregistre les résultats dans une table Analytics Indiquez <i>nom_table</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple : TO "Sortie.FIL"  Par défaut, le fichier de données de table (.FIL) est enregistré dans le dossier contenant le projet Analytics.  Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant : <ul style="list-style-type: none"> <li>• TO "C:\Sortie.FIL"</li> <li>• TO "Résultats\Sortie.FIL"</li> </ul> <div style="border-left: 2px solid blue; padding-left: 10px; margin-left: 20px;"> <p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alpha-numériques, sans l'extension .FIL. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p> </div> </li> <li>◦ <b>nom_fichier</b> : enregistre les résultats dans un fichier Indiquez <i>nom_fichier</i> sous forme de chaîne entre guillemets avec l'extension de fichier appropriée. Par exemple : TO "Sortie.TXT"  Par défaut, le fichier est enregistré dans le dossier contenant le projet Analytics.  Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier dans un autre dossier existant : <ul style="list-style-type: none"> <li>• TO "C:\Sortie.TXT"</li> <li>• TO "Résultats\Sortie.TXT"</li> </ul> </li> <li>◦ <b>PRINT</b> : envoie les résultats vers l'imprimante par défaut</li> </ul>
APPEND optionnel	<p>Ajoute la sortie de commande à la fin d'un fichier existant au lieu de remplacer ce fichier.</p> <div style="border-left: 2px solid blue; padding-left: 10px; margin-left: 20px;"> <p><b>Remarque</b></p> <p>Vous devez vous assurer que la structure de la sortie de la commande et du fichier existant est identique :</p> <ul style="list-style-type: none"> <li>• les mêmes champs</li> <li>• le même ordre des champs</li> <li>• les champs correspondants ont la même longueur</li> <li>• les champs correspondants ont le même type de données</li> </ul> <p>Analytics ajoute la sortie à un fichier existant sans tenir compte de sa structure, ce qui peut désordonner les données si la structure de la sortie et du fichier existant ne correspond pas.</p> </div>
IF test optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.</p> <div style="border-left: 2px solid blue; padding-left: 10px; margin-left: 20px;"> <p><b>Remarque</b></p> <p>La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p> </div>
WHILE test	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La</p>

Nom	Description
optionnel	<p>commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.</p> <p><b>Remarque</b></p> <p>Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.</p>
FIRST <i>plage</i>   NEXT <i>plage</i> optionnel	<p>Le nombre d'enregistrements à traiter :</p> <ul style="list-style-type: none"> <li>○ <b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li>○ <b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> </ul> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter.</p> <p>Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p>
HEADER <i>texte_en-tête</i> optionnel	<p>Texte à insérer en haut de chaque page d'un rapport.</p> <p><i>texte_en-tête</i> doit être indiqué sous forme de chaîne entre guillemets. La valeur remplace la variable système Analytics HEADER.</p>
FOOTER <i>texte_pied_page</i> optionnel	<p>Texte à insérer en bas de chaque page d'un rapport.</p> <p><i>texte_pied_page</i> doit être indiqué sous forme de chaîne entre guillemets. La valeur remplace la variable système Analytics FOOTER.</p>
PRESORT optionnel	<p>Trie la table par le champ clé avant d'exécuter la commande.</p> <p><b>Remarque</b></p> <p>Vous ne pouvez pas utiliser PRESORT dans la commande GROUP.</p>
OPEN optionnel	<p>Ouvre la table créée par la commande après l'exécution de la commande. Valide uniquement lorsque la commande crée une table de sortie.</p>
LOCAL optionnel	<p>Enregistre le fichier de sortie au même emplacement que le projet Analytics.</p> <p><b>Remarque</b></p> <p>Applicable uniquement lorsque vous exécutez la commande sur une table de serveur avec un fichier de sortie qui est une table Analytics.</p>
ISOLocale <i>code_paramètres_régionaux</i> optionnel	<p><b>Remarque</b></p> <p>Applicable dans l'édition Unicode d'Analytics uniquement.</p> <p>Paramètre système local au format <i>langue_pays</i>. Par exemple, pour utiliser le français canadien, saisissez fr_ca.</p> <p>Utilisez les codes suivants :</p> <ul style="list-style-type: none"> <li>○ <b>langue</b> : code linguistique de la norme ISO 639</li> <li>○ <b>pays</b> : code pays de la norme ISO 3166</li> </ul> <p>Si vous n'indiquez pas de code pays, c'est le pays par défaut pour la langue qui est utilisé.</p>

Nom	Description
	Si vous n'utilisez pas ISOLOCALE, ce sont les paramètres système régionaux par défaut qui sont utilisés.

## Variables de sortie Analytics

Nom	Contient
GAPDUP <i>n</i>	Nombre total d'omissions, de doublons ou de groupes de doublons approximatifs identifiés par la commande.

## Exemples

### Test de recherche de valeurs en doublons dans un champ

L'exemple suivant :

- recherche par des tests des valeurs en doublons dans le champ **No\_Facture**
- sort des enregistrements contenant des numéros de facture en doublons dans une nouvelle table Analytics

```

DUPLICATES ON No_Facture OTHER Numéro_fournisseur Date_facture Montant_Facture
PRESORT TO "Factures_Doublons.FIL"

```

### Test de recherche de valeurs en doublons dans au moins deux champs associés

L'exemple suivant :

- recherche par des tests des combinaisons de valeurs en doublons dans les champs **No\_Facture** et **Numéro\_fournisseur**
- sort des enregistrements contenant le même numéro de facture et le même numéro fournisseur dans une nouvelle table Analytics

La différence entre ce test et le précédent, c'est qu'un numéro de facture identique provenant de deux fournisseurs différents n'est pas signalé comme faux positif.

```

DUPLICATES ON No_Facture Numéro_fournisseur OTHER Date_facture Montant_Facture
PRESORT TO "Factures_Doublons.FIL"

```

### Test de recherche d'enregistrements en doublons

Les exemples suivants :

- recherchent, par des tests, des valeurs en doublons dans chaque champ d'une table Inventaire
- sortent tous les champs exactement identiques dans une nouvelle table Analytics

```
DUPLICATES ON ProdNum ProdClass Emplacement ProdDesc ProdStatus UnitCost CostDate  
SalePrice PriceDate PRESORT TO "Éléments_inventaire_doublons.FIL"
```

Vous pouvez simplifier la syntaxe en utilisant ALL :

```
DUPLICATES ON ALL PRESORT TO "Éléments_inventaire_doublons.FIL"
```

# Commande ESCAPE

Arrête le script en cours de traitement, ou tous les scripts, sans quitter Analytics.

## Syntaxe

```
ESCAPE <ALL> <IF test>
```

## Paramètres

Nom	Description
ALL optionnel	Met fin à tous les scripts actifs. S'il est ignoré, le script actif prend fin.
IF <i>test</i> optionnel	Test qui doit être évalué comme vrai avant l'exécution de la commande. Si le test est évalué comme faux, la commande n'est pas exécutée.

## Exemples

### Mettre fin à un script de manière conditionnelle

Vous comptez le nombre d'enregistrements contenu dans une table, et vous utilisez la commande ESCAPE pour arrêter l'exécution du script si le nombre d'enregistrements est inférieur à 100 :

```
COUNT  
ESCAPE IF COUNT1 < 100
```

## Remarques

### Quand utiliser ESCAPE

La commande ESCAPE permet d'arrêter l'exécution d'un script ou d'un sous-script basé sur une condition, ou d'arrêter l'exécution de tous les scripts en cours.

## Utilisation d'ESCAPE dans les sous-scripts

Si vous exécutez ESCAPE à l'intérieur d'un sous-script, le sous-script arrête de s'exécuter et le script principal reprend son traitement à partir de la commande DO SCRIPT ayant appelé le sous-script.

Si vous incluez l'option ALL à la commande ESCAPE dans le sous-script, le sous-script et le script principal arrêtent tous les deux leur traitement :

```
ESCAPE ALL
```

# Commande EVALUATE

Pour l'échantillonnage par enregistrement ou par unité monétaire : projette les erreurs détectées dans les données échantillonnées sur la population entière et calcule les limites supérieures sur le taux d'écart ou le montant des anomalies.

Échantillonnage par enregistrement Échantillonnage par unité monétaire

## Syntaxe

```
EVALUATE RECORD CONFIDENCE niveau_confiance SIZE taille_échantillon
ERRORLIMIT nombre_d'erreurs <TO {SCREEN|nom_fichier}>
```

## Paramètres

### Remarque

N'incluez pas les séparateurs des milliers ou les symboles de pourcentage lorsque vous saisissez des valeurs.

Nom	Description
RECORD	Évaluez les erreurs détectées dans un échantillon par enregistrement.
CONFIDENCE <i>niveau_confiance</i>	Le même niveau de confiance que vous avez spécifié lorsque vous avez calculé la taille de l'échantillon.
SIZE <i>taille_échantillon</i>	Le nombre d'enregistrements contenus dans l'échantillon.  <div style="border-left: 2px solid #0056b3; padding-left: 10px; margin-left: 20px;"> <p><b>Remarque</b></p> <p>Indiquez la taille réelle de l'échantillon tel qu'il a été extrait, cette taille peut être différente de la taille de l'échantillon calculée initialement par Analytics.</p> </div>
ERRORLIMIT <i>nombre_d'erreurs</i>	Le nombre total d'erreurs ou d'écarts détectés dans l'échantillon.
TO SCREEN   <i>nom_fichier</i>	Emplacement vers lequel envoyer les résultats de la commande : <ul style="list-style-type: none"> <li>◦ <b>SCREEN</b> : affiche les résultats dans la zone d'affichage d'Analytics.</li> <li>◦ <b><i>nom_fichier</i></b> : enregistre les résultats dans un fichier</li> </ul> <p>Indiquez <i>nom_fichier</i> sous forme de chaîne entre guillemets avec l'extension de fichier appropriée. Par exemple : TO "Sortie.TXT"</p> <p>Par défaut, le fichier est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier dans un autre</p>

Nom	Description
	dossier existant : <ul style="list-style-type: none"> <li>• TO "C:\Sortie.TXT"</li> <li>• TO "Résultats\Sortie.TXT"</li> </ul>

## Variables de sortie Analytics

Nom	Contient
MLE <sub>n</sub>	Taux de fréquence de la limite d'erreur supérieure (taux d'écart supérieur calculé) calculé par la commande.

## Exemples

### Projeter les erreurs détectées dans les données échantillonnées sur l'intégralité de la population

Vous avez effectué vos tests sur les données échantillonnées et vous avez enregistré les écarts de contrôle détectés. Vous pouvez désormais projeter les erreurs que vous avez détectées sur l'intégralité de la population.

L'exemple ci-dessous prévoit deux erreurs dans les données échantillonnées sur la population totale ; il calcule un **taux de fréquence de limite d'erreur supérieure** (taux d'écart supérieur calculé) de 6,63 %.

```
EVALUATE RECORD CONFIDENCE 95 SIZE 95 ERRORLIMIT 2 TO SCREEN
```

Pour obtenir des explications détaillées sur le calcul des valeurs par Analytics lors de l'évaluation des erreurs, consultez [Évaluation des erreurs dans un échantillon par enregistrement](#).

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).



# Syntaxe

```
EVALUATE MONETARY CONFIDENCE niveau_confiance <ERRORLIMIT valeur_comptable; montant_anomalie <;...n>> INTERVAL valeur_intervalle <TO {SCREEN|nom_fichier}>
```

## Paramètres

### Remarque

N'incluez pas les séparateurs des milliers ou les symboles de pourcentage lorsque vous saisissez des valeurs.

Nom	Description
MONETARY	Évaluez les erreurs détectées dans un échantillon par unité monétaire.
CONFIDENCE <i>niveau_confiance</i>	Le même niveau de confiance que vous avez spécifié lorsque vous avez calculé la taille de l'échantillon.
ERRORLIMIT <i>valeur_comptable;montant_anomalie</i>	<p>Toutes les anomalies que vous avez détectées dans l'échantillon.</p> <p>Indiquez la valeur comptable du montant et le montant de l'anomalie, séparés par un point-virgule. Par exemple, si un montant présente une valeur comptable de 1 000 \$ et une valeur d'audit de 930 \$, indiquez 1000;70.</p> <p>Indiquez les surévaluations sous forme de montants positifs et les sous-évaluations sous forme de montants négatifs. Par exemple, si un montant présente une valeur comptable de 1 250 \$ et une valeur d'audit de 1 450 \$, indiquez 1250;-200.</p> <p>Séparez plusieurs paires <i>valeur_comptable; montant_anomalie</i> par un point-virgule :</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>1000;70;1250;-200</p> </div>
INTERVAL <i>valeur_intervalle</i>	<p>La valeur de l'intervalle que vous avez utilisée lorsque vous avez extrait l'échantillon.</p> <p><b>Remarque</b></p> <p>La valeur de l'intervalle que vous avez utilisée peut différer de la valeur de l'intervalle initialement calculée par Analytics.</p>
TO SCREEN   <i>nom_fichier</i>	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li>○ <b>SCREEN</b> : affiche les résultats dans la zone d'affichage d'Analytics.</li> <li>○ <b><i>nom_fichier</i></b> : enregistre les résultats dans un fichier</li> </ul> <p>Indiquez <i>nom_fichier</i> sous forme de chaîne entre guillemets avec l'extension de fichier appropriée. Par exemple : TO "Sortie.TXT"</p> <p>Par défaut, le fichier est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier dans un autre</p>

Nom	Description
	dossier existant : <ul style="list-style-type: none"> <li>TO "C:\Sortie.TXT"</li> <li>TO "Résultats\Sortie.TXT"</li> </ul>

## Variables de sortie Analytics

Nom	Contient
MLE <sub>n</sub>	Montant de l'erreur la plus probable (anomalie projetée) calculée par la commande.
UEL <sub>n</sub>	Montant de la limite d'erreur supérieure (limite d'anomalie supérieure) calculée par la commande.

## Exemples

### Projeter les erreurs détectées dans les données échantillonnées sur l'intégralité de la population

Vous avez effectué vos tests sur les données échantillonnées et vous avez enregistré les anomalies détectées. Vous pouvez désormais projeter les erreurs que vous avez détectées sur l'intégralité de la population.

L'exemple ci-dessous prévoit la détection de trois erreurs dans les données de l'échantillon sur la population entière, et calcule plusieurs valeurs, dont :

- **Précision de base** : marge de tolérance de base autorisée pour le risque d'échantillonnage (18 850,00)
- **Erreur la plus probable** : montant des anomalies prévu pour la population (1 201,69)
- **Limite d'erreur supérieure** : limite d'anomalie supérieure pour la population (22 624,32)

```
EVALUATE MONETARY CONFIDENCE 95 ERRORLIMIT 1000;70;1250;-200;3200;900
INTERVAL 6283,33 TO SCREEN
```

Pour obtenir des explications détaillées sur le calcul des valeurs par Analytics lors de l'évaluation des erreurs, consultez [Évaluation des erreurs dans un échantillon par unité monétaire](#).

# Remarques

## Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

# Commande EXECUTE

Exécute une application ou un processus externe par rapport à Analytics. Émule la commande Exécuter de Windows. Peut être utilisée pour interagir avec l'invite de commande de Windows.

## Remarque

La commande EXECUTE vous donne la possibilité d'interagir avec le système d'exploitation et avec des applications externes à Analytics, c'est pourquoi des problèmes techniques qui se situent au-delà de l'étendue de la fonctionnalité native d'Analytics peuvent survenir.

L'Assistance peut vous aider à utiliser la commande EXECUTE à l'intérieur d'Analytics, mais les problèmes qui surviennent avec des processus et des applications externes à Analytics ne sont pas pris en charge dans le cadre de l'Assistance.

## Syntaxe

```
EXECUTE Syntaxe_commande_Exécuter_Windows <ASYN<
```

## Paramètres

Nom	Description
<i>Syntaxe_commande_Exécuter_Windows</i>	<p>Nom de l'application à exécuter, du dossier ou du fichier à ouvrir, ou de la commande à exécuter, suivi de tous les arguments ou interrupteurs de commande requis.</p> <p>Requiert une syntaxe de commande Exécuter de Windows valide entourée de guillemets.</p>
ASYN< optionnel	<p>Exécute la commande en mode asynchrone.</p> <p>En mode asynchrone, le script Analytics continue de s'exécuter sans attendre la fin du processus lancé par la commande EXECUTE.</p> <p>Si vous omettez ASYN&lt;, le processus lancé par la commande EXECUTE doit s'achever avant que le script Analytics continue. Analytics est inaccessible pendant l'exécution des processus externes.</p> <p><b>Remarque</b></p> <p>Lors de l'exécution de la commande EXECUTE à partir de la ligne de commande d'Analytics, vous devez indiquer ASYN&lt;.</p>

# Variables de sortie Analytics

Nom	Contient
RETURN_CODE	<p>Code renvoyé par une application externe ou une exécution de processus à l'aide de la commande EXECUTE.</p> <p><b>Que sont les codes de retour ?</b></p> <p>Les codes de retour sont des nombres générés par l'application ou le processus externe et ils sont renvoyés à Analytics pour indiquer l'issue du processus externe. Analytics ne génère pas le code de retour, il ne fait que le recevoir.</p> <p><b>Codes de retour types</b></p> <p>Les codes de retour types sont des valeurs entières qui mappent vers des notifications ou des messages d'erreur particuliers. Par exemple, le code de retour « 0 » pourrait signifier « L'opération a été effectuée avec succès ». Le code de retour « 2 » pourrait signifier « Le système ne peut pas trouver le fichier spécifié ».</p> <p><b>Signification de certains codes de retour</b></p> <p>Les codes de retour spécifiques et leurs significations varient en fonction de l'application ou du processus externe. Les listes des codes de retour, aussi appelés « codes d'erreur » ou « codes de sortie », et leurs significations, figurent souvent dans la documentation de l'application externe associée. Vous trouverez également les listes des codes de retour sur Internet.</p> <p><b>Variable créée en mode par défaut uniquement</b></p> <p>La variable RETURN_CODE est créée lorsque la commande EXECUTE est exécutée dans le mode par défaut. La variable n'est pas créée lorsque la commande est exécutée en mode asynchrone.</p>

## Exemples

### Ouvrir une application

Ouvre Microsoft Excel :

```
EXECUTE "Excel"
```

Ouvre Adobe Acrobat Reader :

```
EXECUTE "AcroRd32.exe"
```

### Fermer une application

Ferme Microsoft Excel :

```
EXECUTE "TASKKILL /f /im Excel.exe"
```

### Remarque

Utilisez le bouton `/f` avec prudence. Cela force la fermeture d'une application sans présentation de boîtes de dialogue, comme celles qui permettent la sauvegarde des modifications.

## Ouvrir un fichier

Ouvre le classeur Excel `AP_Trans.xlsx` :

```
EXECUTE "'C:\Projets ACL\Données source\AP_Trans.xlsx'"
```

## Créer un nouveau dossier

Crée un nouveau dossier appelé `Données source` :

```
EXECUTE 'cmd /c MD "C:\Projets ACL\Données source"'
```

## Exécuter des scripts externes ou des fichiers de commandes non-Analytics (.bat)

Exécute le script `Mon_Lot.bat` :

```
EXECUTE "'C:\Projets ACL\Fichiers de lot\Mon_Lot.bat'"
```

## Transmettre des paramètres vers un fichier de commandes qui n'appartient pas à Analytics

Transmet deux paramètres à `Mon_Lot.bat`. Les paramètres peuvent être des valeurs littérales ou des variables Analytics :

```
EXECUTE "'C:\Projets ACL\Fichiers de lot\Mon_Lot.bat" param1%v_param2%'
```

## Exécuter des scripts Analytics dans d'autres projets Analytics

Exécute `"AP_Trans_script"` dans `AP Trans Tests.acl`"

```
EXECUTE 'aclwin.exe "C:\ACL Projects\AP Trans Tests.acl" /b AP_Trans_script'
```

### Remarque

L'exécution d'un script Analytics dans un autre projet lance une seconde instance d'Analytics. Le script dans le second projet devrait se terminer par la commande QUIT de sorte que la seconde instance d'Analytics se ferme et que le contrôle soit renvoyé à l'instance initiale d'Analytics.

## Incorporer un délai d'attente dans un script Analytics

Les deux exemples créent un délai d'attente de 30 secondes :

```
EXECUTE "TIMEOUT /t 30"
```

```
EXECUTE "cmd /c PING -n 31 127,0,0,1 > nul"
```

## Remarques

### Utiliser EXECUTE pour exécuter des tâches utiles

La commande EXECUTE vous permet d'exécuter des commandes Windows et DOS depuis la ligne de commande d'Analytics ou à partir d'un script Analytics.

Vous pouvez utiliser cette capacité pour augmenter l'automatisation des scripts Analytics en réalisant une variété de tâches utiles qui ne peuvent pas être réalisées en utilisant la syntaxe ACLScript seule.

### Exemples de tâches pouvant être commencées à l'aide d'EXECUTE

Ouverture des autres programmes et applications et exécution des tâches requises par le script d'Analytics	Paramètres de passage vers un fichier de lot	Accès aux données à partir d'emplacements réseau	Incorporation de listes de comptes Active Directory
Ouverture de n'importe quel fichier dans son application par défaut	Exécuter des scripts Analytics dans d'autres projets Analytics	Utilisation du FTP pour accéder aux données à partir d'emplacements distants	Intégration avec VBScript
Exécution de tâches d'administration relatives aux fichiers et aux dossiers comme la copie, le déplacement, la création, la suppression ou la comparaison de fichiers ou de dossiers qui existent en dehors d'Analytics	Incorporation de délais d'attente dans les scripts Analytics	Compression ou décompression Zip de données	Intégration avec des bases de données SQL

Exécuter des scripts externes ou des fichiers de commandes non-Analytics (.bat)	Incorporation de la planification des tâches de Windows dans les scripts Analytics	Chiffrement ou déchiffrement de données	Ouverture de pages Web
<p><b>Remarque</b></p> <p>Les informations spécifiques concernant la manière d'effectuer ces tâches sortent du champ d'application de la documentation d'aide de Galvanize. Pour obtenir de l'aide, veuillez consulter la documentation du système d'exploitation Windows qui convient, ou toute autre documentation de tiers.</p>			

## Mode par défaut et mode asynchrone

Vous pouvez exécuter la commande EXECUTE dans le mode par défaut ou dans le mode asynchrone :

- **Mode par défaut** : le processus commencé par la commande EXECUTE doit être terminé avant que le script Analytics puisse continuer.

Analytics est inaccessible pendant l'exécution des processus externes.

- **Mode asynchrone** : le script Analytics continue de s'exécuter sans attendre la fin du processus lancé par la commande EXECUTE.

Analytics reste accessible pendant l'exécution des processus externes.

Si vous spécifiez ASYNC, la commande EXECUTE s'exécute en mode asynchrone.

### Quel mode dois-je utiliser ?

Lorsque vous créez des scripts Analytics qui utilisent la commande EXECUTE vous devez prendre en compte le mode d'exécution le plus adapté.

Utiliser le mode par défaut	Utiliser le mode asynchrone / ASYNC
<ul style="list-style-type: none"> <li>○ tâches d'administration par fichier et par dossier</li> <li>○ spécifier des périodes d'attente</li> <li>○ toute tâche dont les tâches suivantes dépendent</li> <li>○ l'exécution d'un script qui suit dépend du résultat dans la variable RETURN_CODE.</li> </ul>	<ul style="list-style-type: none"> <li>○ les tâches externes entraînent l'ouverture d'une interface d'application ou d'une boîte de dialogue contextuelle</li> </ul>

### Scripts Analytics s'exécutant sans assistance

Si vous souhaitez qu'un script Analytics contenant la commande EXECUTE s'exécute sans assistance, utilisez l'une des méthodes suivantes :

- utiliser le mode asynchrone pour des tâches qui entraînent l'ouverture d'une interface d'application ou d'une boîte de dialogue contextuelle
- éviter d'ouvrir des éléments de l'interface dans des scripts s'exécutant sans assistance



### Remarque

Tant que les éléments de l'interface ne sont pas fermés, ils représentent des processus qui sont toujours en cours d'exécution. Si ces éléments d'interface sont ouverts avec EXECUTE dans le mode par défaut, ils évitent l'exécution des lignes suivantes dans un script Analytics et entraînent le blocage du script.

## Commande EXECUTE dans les scripts d'outils d'analyse

Si vous souhaitez utiliser la commande EXECUTE dans les scripts d'outils d'analyse de Robots ou d'AX Serveur, vous devez observer une configuration spécifique pour permettre l'exécution de la commande. Pour plus d'informations, consultez :

- **Robots** : [Configuration d'un agent Robots](#)
- **Serveur AX** : [Paramètres d'AX Server](#)

## Guillemets

La syntaxe de la commande Exécuter de Windows que vous utilisez avec la commande EXECUTE doit être entourée de guillemets simples ou doubles.

L'exemple ci-dessous utilise la commande MD de Windows pour créer un nouveau répertoire :

```
EXECUTE 'cmd /c md C:\Nouveau_Répertoire_Données'
```

## Guillemets imbriqués

Si des chemins de fichiers dans la syntaxe de la commande Exécuter contiennent des espaces, alors les chemins de fichiers doivent également être entre guillemets.

Pour mettre des chemins entre guillemets, vous avez deux possibilités :

- **Guillemets doubles à l'intérieur de guillemets simples** : Entourez la chaîne de la commande d'exécution avec des guillemets simples et encadrez les chemins internes avec des guillemets doubles :

```
EXECUTE 'cmd /c md "C:\Nouveau Répertoire Données"
```

Vous trouverez peut-être cette méthode plus facile à lire que la seconde méthode.

### Remarque

L'inversion de l'ordre de l'imbrication - l'utilisation de guillemets doubles pour entourer la totalité de la chaîne, et de guillemets simples pour entourer les chemins de fichiers - ne fonctionne pas.

- **Deux guillemets doubles** : Entourez la chaîne de la commande d'exécution avec des guillemets doubles et encadrez les chemins internes avec deux guillemets doubles :

```
EXECUTE "cmd /c md ""C:\Nouveau Répertoire Données"""
```

Si vous utilisez cette deuxième méthode, les deux guillemets doubles utilisés en interne doivent être directement adjacents et il ne doit pas y avoir d'espace entre eux.

## Commandes internes et externes

Les commandes Windows peuvent être **internes** ou **externes**.

- Les **commandes internes** : peuvent être exécutées à partir de l'invite de commande uniquement, ce qui signifie que vous devez ouvrir la session de commandes à l'aide de `cmd /c` ou de `cmd /k` avant d'indiquer la commande.
- Les **commandes externes** : peuvent être exécutées à partir de l'invite de commande ou directement à l'aide de la commande EXECUTE, ce qui signifie que l'ouverture de la session de commandes est une option, mais qu'elle n'est pas requise.

L'exemple ci-dessous utilise la commande Windows DIR interne (affichage du contenu d'un répertoire), et la commande Windows COMP externe (comparaison de deux fichiers), pour illustrer la différence :

```
EXECUTE 'cmd /k dir "C:\ACL DATA\Exemples de fichiers de données"'
EXECUTE 'comp C:\Fichier_1.txt C:\Fichier_2.txt'
```

Vous pouvez éviter cette complication en créant des scripts externes ou des fichiers de lot pour contenir les commandes Windows, et en utilisant la commande EXECUTE uniquement pour démarrer le fichier de lot. Par exemple :

```
EXECUTE 'C:\Mon_Lot.bat'
```

## Syntaxe de commande à exécution multi-lignes

La commande EXECUTE ne prend pas en charge la commande Exécuter multi-ligne. Pour intégrer des commandes Exécuter multi-lignes dans un script Analytics, utilisez l'une des méthodes suivantes :

Méthode	Exemple
Répétez la commande EXECUTE pour chaque commande Exécuter.	<pre>EXECUTE 'cmd /c md "C:\Nouveau Répertoire Données"' EXECUTE 'cmd /c copy C:\Fichier_1.txt "C:\Nouveau Répertoire Données"'</pre>

Méthode	Exemple
Com- binez des com- mandes d'exécuti- on avec &.	<pre>EXECUTE 'cmd /c md "C:\Nouveau Répertoire Données" &amp; copy C:\Fichier_1.txt "C:\Nouveau Répertoire Données"'</pre>
Créez un script externe ou un fichier de com- mandes qui contient des com- mandes d'exécuti- on sur plusieurs lignes, puis uti- lisez la com- mande EXECUT- E uni- quement pour lan- cer le fichier de com- mandes.	<pre>EXECUTE 'C:\Mon_Lot.bat'</pre>

# Commande EXPORT

Exporte des données depuis Analytics vers le format de fichier spécifié ou vers Résultats dans HighBond.

## Syntaxe

```
EXPORT {<FIELDS> nom_champ<AS nom_export> <...n>|<FIELDS> ALL} <UNICODE> type_
export<SCHEMA> PASSWORD num TO {nom_fichier|aclgrc_id} <OVERWRITE> <IF test>
<WHILE test> <{FIRST plage|NEXT plage}> <APPEND> <KEEPTITLE> <SEPARATOR
caractère> <QUALIFIER caractère> <WORKSHEET nom_feuille_de_calcul> <DISPLAYNAME>
```

## Paramètres

Nom	Description
FIELDS <i>nom_champ</i> AS <i>nom_export</i> <...n>   FIELDS ALL	<p>Champs à exporter.</p> <ul style="list-style-type: none"> <li>◦ <b>nom_champ</b> : exportez le(s) champ(s) spécifié(s) Séparez les noms des champs par des espaces. Si nécessaire, vous pouvez inclure un autre nom pour le champ du fichier d'exportation à l'aide de AS <i>nom_export</i>. Entourez <i>nom_export</i> de guillemets. Si vous effectuez une exportation vers ACLGRC (HighBond), il est possible de combiner AS au paramètre DISPLAYNAME. Pour plus d'informations, consultez la rubrique "Comment DISPLAYNAME interagit avec AS lors de l'exportation dans HighBond Résultats" Page 205.</li> <li>◦ <b>ALL</b> : exportez tous les champs de la table</li> </ul>
UNICODE optionnel	<p>Disponible dans l'édition Unicode d'Analytics uniquement. S'applique uniquement aux fichiers texte (ASCII), aux fichiers texte délimités (DELIMITED), et aux fichiers XML ainsi qu'à la sortie du Presse-papiers Windows (CLIPBOARD).</p> <p>Exporte les données Analytics en appliquant le codage de caractères Unicode UTF-16 Little Endian (petit-boutiste).</p> <ul style="list-style-type: none"> <li>◦ <b>Indiquez UNICODE</b> : si les données que vous exportez contiennent des caractères non pris en charge par le format ASCII (ANSI) étendu</li> <li>◦ <b>Ne spécifiez pas UNICODE</b> : si tous les caractères dans les données que vous exportez sont pris en charge par le format ASCII (ANSI) étendu</li> </ul> <p>Les données exportées sont codées au format ASCII (ANSI) étendu.</p> <p><b>Remarque</b> Les caractères non pris en charge sont omis du fichier exporté.</p> <p>Pour plus d'informations, consultez la rubrique <a href="#">Produits ACL Unicode</a>.</p>

Nom	Description
<i>type_export</i>	<p>Format de fichier de sortie ou destination à l'aide de l'une des options suivantes :</p> <ul style="list-style-type: none"> <li>◦ <b>ACCESS</b> : Fichier de base de données Microsoft Access (.mdb) Par défaut, les données sont exportées au format Unicode.</li> <li>◦ <b>ACLGRC</b> : Résultats dans HighBond</li> <li>◦ <b>ASCII</b> : Texte brut ASCII (.txt)</li> <li>◦ <b>CLIPBOARD</b> : Presse-papiers Windows</li> <li>◦ <b>DBASE</b> : Fichier compatible dBASE (.dbf)</li> <li>◦ <b>DELIMITED</b> : Fichier texte délimité (.del)</li> <li>◦ <b>EXCEL</b> : Fichier Microsoft Excel (.xls) compatible avec Excel 1997 à 2003</li> <li>◦ <b>JSON</b> : Fichier JSON (.json)</li> <li>◦ <b>LOTUS</b> : Fichiers Lotus 123</li> <li>◦ <b>WDPF6</b> : Fichier Wordperfect 6</li> <li>◦ <b>WORD</b> : Fichier MS Word (.doc)</li> <li>◦ <b>WP</b> : Fichier Wordperfect</li> <li>◦ <b>XLS21</b> : Fichier Microsoft Excel version 2.1</li> <li>◦ <b>XLSX</b> : Fichier Microsoft Excel .xlsx Par défaut, les données sont exportées au format Unicode.</li> <li>◦ <b>XML</b> : Fichier XML (.xml)</li> </ul>
SCHEMA optionnel	<p>S'applique uniquement à la sortie de fichier XML.</p> <p>Inclure le schéma XML dans le fichier XML exporté. Le schéma XML contient des méta-données qui décrivent la structure du fichier XML, notamment le type de données des champs.</p> <p>Vous pouvez valider le fichier par rapport au schéma une fois que le fichier a été exporté.</p>
PASSWORD <i>num</i>	<p><b>Remarque</b> S'applique à HighBond Résultats uniquement.</p> <p>Définition de mot de passe à utiliser.</p> <p>Vous n'utilisez pas PASSWORD <i>num</i> pour inviter à saisir ou pour spécifier un mot de passe réel. La définition de mot de passe renvoie à un mot de passe fourni ou défini auparavant à l'aide de la commande PASSWORD, de la commande SET PASSWORD ou de la balise d'outil d'analyse PASSWORD.</p> <p><i>num</i> correspond au numéro de la définition de mot de passe. Par exemple, si deux mots de passe ont été fournis ou définis auparavant dans un script, ou lors de la planification d'un outil d'analyse, PASSWORD 2 indique que le mot de passe n° 2 est utilisé.</p> <p>Pour plus d'informations sur la fourniture ou définition de mots de passe, consultez les rubriques :</p> <ul style="list-style-type: none"> <li>• "Commande PASSWORD" Page 373</li> <li>• "Commande SET" Page 435</li> <li>• <a href="#">Balise d'outil d'analyse PASSWORD</a></li> </ul> <p>Pour plus d'informations sur la fourniture ou définition de mots de passe, consultez les rubriques :</p> <ul style="list-style-type: none"> <li>• <a href="#">Commande PASSWORD</a></li> <li>• <a href="#">Commande SET</a></li> </ul>

Nom	Description							
	<ul style="list-style-type: none"> <li>• <a href="#">Balise d'outil d'analyse PASSWORD</a></li> </ul> <p>PASSWORD <i>num</i> doit être placé directement avant TO ou à la fin de chaîne de caractères de la syntaxe de commande. Le mot de passe est un jeton d'accès à HighBond. Pour plus d'informations, consultez la section "Exportation dans HighBond Résultats" Page 203.</p> <p><b>Remarque</b></p> <p>Selon l'environnement d'exécution du script, PASSWORD peut être requis ou non :</p> <table border="1" data-bbox="589 564 1292 1050"> <tr> <td data-bbox="589 564 940 753">Analytics (activation en ligne)</td> <td data-bbox="940 564 1292 753">PASSWORD n'est pas obligatoire.  Le jeton d'accès HighBond de l'utilisateur actuel est utilisé automatiquement.</td> </tr> <tr> <td data-bbox="589 753 940 858">Analytics (activation hors ligne)</td> <td data-bbox="940 753 1292 858" rowspan="4">PASSWORD est obligatoire.</td> </tr> <tr> <td data-bbox="589 858 940 926">Robots</td> </tr> <tr> <td data-bbox="589 926 940 989">Analytics Exchange</td> </tr> <tr> <td data-bbox="589 989 940 1050">Fenêtre App Analyse</td> </tr> </table>	Analytics (activation en ligne)	PASSWORD n'est pas obligatoire.  Le jeton d'accès HighBond de l'utilisateur actuel est utilisé automatiquement.	Analytics (activation hors ligne)	PASSWORD est obligatoire.	Robots	Analytics Exchange	Fenêtre App Analyse
Analytics (activation en ligne)	PASSWORD n'est pas obligatoire.  Le jeton d'accès HighBond de l'utilisateur actuel est utilisé automatiquement.							
Analytics (activation hors ligne)	PASSWORD est obligatoire.							
Robots								
Analytics Exchange								
Fenêtre App Analyse								
TO <i>nom_fichier</i>   <i>aclgrc_id</i>	<p>Destination de l'exportation :</p> <ul style="list-style-type: none"> <li>◦ <b>TO <i>nom_fichier</i></b> : exporter les données dans un fichier</li> </ul> <p>Si besoin, vous pouvez inclure un chemin d'accès absolu ou relatif, mais le dossier Windows doit déjà exister. Vous devez indiquer la valeur <i>nom_fichier</i> sous forme de chaîne entre guillemets.</p> <ul style="list-style-type: none"> <li>◦ <b>TO <i>aclgrc_id</i></b> : exporter des données vers HighBond Résultats</li> </ul> <p>La valeur <i>aclgrc_id</i> doit inclure le numéro d'ID du test de contrôle, et si vous effectuez votre exportation vers un data center qui n'est pas en Amérique du Nord, cette valeur doit inclure le code du data center. La valeur <i>aclgrc_id</i> doit être entourée de guillemets.</p> <p>Le numéro ID du test de contrôle et le code du centre de données doivent être séparés par une arobase (@). Par exemple, TO "99@eu".</p> <p>Si vous ne connaissez pas le numéro d'ID du test de contrôle, utilisez l'interface utilisateur d'Analytics pour commencer une exportation dans Résultats. Annulez l'exportation après avoir identifié le numéro d'ID du test de contrôle. Pour obtenir plus d'informations, consultez la section <a href="#">Exportation des exceptions vers ACL GRC</a>.</p> <p>Le code du data center indique vers quel serveur régional d'HighBond vous exportez les données :</p> <ul style="list-style-type: none"> <li>• ap : Asie-Pacifique</li> <li>• au : Australie</li> <li>• ca : Canada</li> <li>• eu : Europe</li> </ul>							

Nom	Description
	<ul style="list-style-type: none"> <li>• us : Amérique du Nord</li> </ul> <p>Vous pouvez uniquement utiliser le ou les codes de data center autorisés pour l'installation d'HighBond de votre organisation. Le data center nord américain est le data center par défaut, vous n'êtes donc pas obligé d'indiquer "@us".</p>
<p>OVERWRITE optionnel</p>	<p>S'applique à l'exportation vers HighBond Résultats uniquement.</p> <p>Des données existantes dans le test de contrôle cible (table) sont écrasées par les données exportées. Vous devez avoir un rôle de Responsable professionnel dans la collection cible pour écraser des données.</p> <p>Si vous omettez OVERWRITE et que les données existent déjà dans le test de contrôle cible (table), les données exportées sont ajoutées aux données existantes. Pour plus d'informations sur l'ajout dans Résultats, consultez les « Remarques » ci-dessous.</p> <p>Les interprétations associées au test de contrôle cible (table) s'actualisent de manière dynamique pour refléter les données importées, qu'il s'agisse d'un écrasement ou d'un ajout.</p>
<p>IF test optionnel</p>	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.</p> <p><b>Remarque</b></p> <p>La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p>
<p>WHILE test optionnel</p>	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.</p> <p><b>Remarque</b></p> <p>Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.</p>
<p>FIRST <i>plage</i>   NEXT <i>plage</i> optionnel</p>	<p>Le nombre d'enregistrements à traiter :</p> <ul style="list-style-type: none"> <li>◦ <b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li>◦ <b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> </ul> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter.</p> <p>Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p>
<p>APPEND optionnel</p>	<p>S'applique uniquement aux fichiers texte (ASCII) et fichiers texte délimités (DELIMITED).</p> <p>Ajoute la sortie de commande à la fin d'un fichier existant au lieu de remplacer ce fichier.</p>

Nom	Description
	<p><b>Remarque</b></p> <p>Vous devez vous assurer que la structure de la sortie de la commande et du fichier existant est identique :</p> <ul style="list-style-type: none"> <li>• les mêmes champs</li> <li>• le même ordre des champs</li> <li>• les champs correspondants ont la même longueur</li> <li>• les champs correspondants ont le même type de données</li> </ul> <p>Analytics ajoute la sortie à un fichier existant sans tenir compte de sa structure, ce qui peut désordonner les données si la structure de la sortie et du fichier existant ne correspond pas.</p>
KEEPTITLE optionnel	<p>S'applique uniquement aux fichiers texte (ASCII) et fichiers texte délimités (DELIMITED).</p> <p>Inclure les noms des champs Analytics avec les données exportées. Si vous ignorez ce paramètre, aucun nom de champ n'apparaît dans le fichier de sortie.</p>
SEPARATOR <i>caractère</i> optionnel	<p>S'applique uniquement aux fichiers texte délimités (DELIMITED).</p> <p>Caractère à utiliser comme séparateur de champs. Vous devez indiquer le caractère sous forme de chaîne entre guillemets.</p> <p>Par défaut, Analytics utilise une virgule.</p>
QUALIFIER <i>caractère</i> optionnel	<p>S'applique uniquement aux fichiers texte délimités (DELIMITED).</p> <p>Caractère à utiliser en tant qu'identificateur de texte pour renvoyer à la ligne et identifier des valeurs de champs. Vous devez indiquer le caractère sous forme de chaîne entre guillemets.</p> <p>Par défaut, Analytics utilise des guillemets doubles.</p>
WORKSHEET <i>nom_ feuille_de_calcul</i> optionnel	<p>S'applique uniquement aux fichiers Microsoft Excel (.xlsx).</p> <p>Nom de la feuille de calcul Excel créée dans un nouveau fichier Excel ou dans un fichier Excel existant.</p> <p>Par défaut, Analytics utilise le nom de la table Analytics que vous êtes en train d'exporter comme nom de la feuille de calcul.</p> <p><i>nom_feuille_de_calcul</i> peut contenir uniquement des caractères alphanumériques ou le caractère de soulignement ( _ ). Le nom ne peut pas contenir de caractères spéciaux ni d'espaces et il ne peut pas commencer par un chiffre. Il n'est pas obligatoire de placer la valeur entre guillemets.</p> <p>Pour plus d'informations sur l'écrasement de classeurs et de feuilles de calcul Excel lors de l'exportation, consultez la rubrique "Paramètre WORKSHEET et écrasement" Page 203.</p>
DISPLAYNAME optionnel	<p>S'applique à ACLGRC (HighBond) uniquement.</p> <p>Exporte les noms des champs sous forme de noms de champs et exporte les noms d'affichage sous forme de noms d'affichage. Ainsi, les noms d'affichage apparaissent dans les en-têtes des colonnes dans Résultats sans incidence sur le nom de champ réel.</p>



Nom	Description
	Il est possible de combiner DISPLAYNAME à AS. Pour plus d'informations, consultez la section "Comment DISPLAYNAME interagit avec AS lors de l'exportation dans HighBond Résultats" Page 205.

## Exemples

### Exportation de données vers un fichier Excel .xlsx

Vous exportez des champs spécifiques à partir de la table **Fournisseur** vers un fichier Excel .xlsx :

```
OPEN Fournisseur
EXPORT FIELDS No_Fournisseur Nom_Fournisseur Ville_Fournisseur XLSX TO "ExportFournisseur"
```

### Exportation de données vers un fichier Excel .xlsx et choix d'un nom de feuille de calcul

Vous exportez des champs spécifiques de la table **Fournisseur** vers une feuille de calcul nommée **Fournisseurs\_États-Unis** dans un fichier Excel .xlsx :

```
OPEN Fournisseur
EXPORT FIELDS No_Fournisseur Nom_Fournisseur Ville_Fournisseur XLSX TO "ExportFournisseur" WORKSHEET Fournisseurs_États-Unis
```

### Exportation de tous les champs vers un fichier délimité

Vous exportez tous les champs à partir de la table **Fournisseur** vers un fichier délimité :

```
OPEN Fournisseur
EXPORT FIELDS ALL DELIMITED TO "ExportFournisseur"
```

### Exportation de données vers plusieurs fichiers délimités à l'aide de GROUP

Vous exportez des champs spécifiques à partir de la table **Fournisseur** vers deux fichiers délimités :

- un fichier pour les noms des fournisseurs de « A » à « M »
- un fichier pour les noms des fournisseurs de « N » à « Z »

À l'aide de la commande GROUP, vous testez le nom du fournisseur de chaque enregistrement avec une condition IF :

```
GROUP
EXPORT FIELDS No_Fournisseur Nom_Fournisseur DELIMITED TO "AàM" IF BETWEEN
(UPPER(Nom_Fournisseur); "A"; "M")
EXPORT FIELDS No_Fournisseur Nom_Fournisseur DELIMITED TO "NàZ" IF BETWEEN
(UPPER(Nom_Fournisseur); "N"; "Z")
END
```

## Exporter des données vers HighBond Résultats

Vous exportez des champs spécifiques de la table **Exceptions\_Ar** vers HighBond Résultats. Vous écrasez les données existante dans le test de contrôle cible (table) :

```
OPEN Exceptions_Ar
EXPORT FIELDS Aucune date d'échéance Réf Montant Type ACLGRC PASSWORD 1 TO
"10926@us" OVERWRITE
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Utilisation d'EXPORT avec la commande GROUP

Avec la plupart des formats d'exportation, vous pouvez exporter les données simultanément vers plusieurs fichiers à l'aide de la commande GROUP.

Un seul fichier peut être créé à la fois lorsque vous exportez des données vers Microsoft Excel et Microsoft Access.

## Export vers Excel

Les limitations suivantes s'appliquent lors de l'exportation de données vers un fichier Excel :

<p>Nombre d'enregistrements</p>	<ul style="list-style-type: none"> <li>○ <b>Excel 2007 et versions suivantes (*.xlsx)</b> : un maximum de 1 048 576 enregistrements</li> <li>○ <b>Excel 97 et 2003</b> : un maximum de 65 536 enregistrements</li> </ul> <p>Les tables Analytics qui dépassent ces valeurs maximales sont exportées, mais les enregistrements en excès sont ignorés et ne sont pas exportés.</p>
<p>Longueur des champs</p>	<ul style="list-style-type: none"> <li>○ aucune limite de longueur pour un champ spécifique</li> <li>○ les longueurs des champs spécifiques ne peuvent dépasser la limite de longueur globale des enregistrements qui est de 32 Ko</li> </ul> <p>(32 765 caractères dans Analytics non-Unicode, 16 382 caractères dans Analytics)</p>

	Unicode) <ul style="list-style-type: none"> <li>◦ dans le cas d'Excel 2.1, un maximum de 247 caractères</li> </ul>
Longueur des noms des champs	<ul style="list-style-type: none"> <li>◦ un maximum de 64 caractères</li> <li>◦ dans le cas d'Excel 2.1, un maximum de 248 caractères</li> </ul>

## Paramètre WORKSHEET et écrasement

Le résultat de l'utilisation ou non du paramètre WORKSHEET lors de l'exportation à partir d'une table Analytics vers un fichier Excel est expliqué ci-dessous :

Correspondance	Description	Paramètre WORKSHEET utilisé	Paramètre WORKSHEET non utilisé
Aucun nom de fichier Excel correspondant	<ul style="list-style-type: none"> <li>◦ TO <i>nomdefichier</i> la valeur ne correspond à aucun nom de fichier Excel existant</li> </ul>	Un nouveau fichier Excel est créé, avec une feuille de calcul portant le nom spécifié	Un nouveau fichier Excel est créé, avec une feuille de calcul utilisant le nom de la table Analytics exportée
Nom de fichier Excel correspondant Aucun	<ul style="list-style-type: none"> <li>◦ TO <i>nomdefichier</i> la valeur, et un nom de fichier Excel existant, sont identiques</li> <li>◦ WORKSHEET <i>nom_feuillecalcul</i> ne correspond à aucun nom de feuille de calcul dans le fichier Excel</li> </ul>	Une feuille de calcul avec le nom spécifié est ajoutée au fichier Excel existant	Le fichier Excel existant est écrasé par un nouveau fichier Excel, comportant une feuille de calcul qui utilise le nom de la table Analytics exportée
Nom de fichier Excel et nom de feuille de calcul correspondants	<ul style="list-style-type: none"> <li>◦ TO <i>nomdefichier</i> la valeur, et un nom de fichier Excel existant, sont identiques</li> <li>◦ WORKSHEET <i>nom_feuillecalcul</i> correspond à un nom de feuille de calcul dans le fichier Excel</li> </ul>	<p>Une feuille de calcul portant le nom spécifié écrase la feuille de calcul existante si elle a été créée dans Analytics à l'origine.</p> <p>Un message d'erreur s'affiche et l'opération d'exportation est annulée si la feuille de calcul existante a été créée directement dans Excel à l'origine.</p>	Le fichier Excel existant est écrasé par un nouveau fichier Excel, comportant une feuille de calcul qui utilise le nom de la table Analytics exportée

## Exportation dans HighBond Résultats

Le tableau ci-dessous contient des informations supplémentaires sur l'exportation dans un test de contrôle dans Résultats.

Élément	Détails
<b>Autorisations requises</b>	<p>La fonctionnalité permettant d'exporter les résultats dans un test de contrôle dans Résultats nécessite une affectation de rôle HighBond spécifique ou des privilèges d'administrateur :</p> <ul style="list-style-type: none"> <li>○ Les utilisateurs ayant un rôle d'Utilisateur professionnel ou de Responsable professionnel pour une collection de Résultats peuvent exporter les résultats vers n'importe quel test de contrôle dans la collection.</li> </ul> <p><b>Remarque</b></p> <p>Seuls les utilisateurs ayant le rôle Responsable professionnel peuvent exporter et écraser les données existantes dans un test de contrôle.</p> <ul style="list-style-type: none"> <li>○ Les administrateurs système d'HighBond et les administrateurs de Résultats reçoivent automatiquement le rôle de Responsable professionnel dans toutes les collections des organisations HighBond qu'ils gèrent.</li> </ul>
<b>Limites d'exportation</b>	<p>Les limites suivantes s'appliquent lors de l'exportation vers un test de contrôle :</p> <ul style="list-style-type: none"> <li>○ 100 000 enregistrements maximum par exportation</li> <li>○ 100 000 enregistrements maximum par test de contrôle</li> <li>○ 500 champs maximum par enregistrement</li> <li>○ 256 caractères maximum par champ</li> </ul> <p>Vous pouvez lancer plusieurs exportations vers un même test de contrôle, mais vous ne devez pas sortir de ces limites générales.</p>
<b>Ajout de champs</b>	<p>Indépendamment de leur ordre dans une table Analytics, les champs exportés sont ajoutés aux champs existants dans un test de contrôle si leurs noms de champs physiques correspondent.</p> <p>Dans Analytics, le nom de champ physique correspond au nom dans le format de table. Les champs exportés dont le nom ne correspond pas à celui des champs existants s'ajoutent sous forme de colonnes supplémentaires à la table dans Résultats.</p> <p>Les noms d'affichage pour les champs dans Analytics et dans Résultats ne sont pas pris en compte. Toutefois, si vous utilisez le paramètre <i>AS nom_export</i> facultatif, la valeur <i>nom_export</i> est utilisée comme nom de champ physique si vous n'utilisez pas <i>DISPLAYNAME</i>.</p> <p>Lorsque vous concaténez des données aux champs du questionnaire, le nom d'affichage de la colonne dans Résultats reste le nom spécifié dans la configuration du questionnaire.</p> <p><b>Remarque</b></p> <p>Si vous bouclez les données entre Résultats et Analytics et que les données apparaissent mal alignées dans Résultats, c'est que vous n'avez probablement pas fait correspondre les noms des champs.</p> <p>Pour plus d'informations, consultez la section "Considérations générales sur les noms des champs lors de l'importation et de l'exportation des données de Résultats" Page 275.</p>
<b>Création d'une définition de mot de passe et spécification d'une valeur de mot de passe</b>	<p>Commande PASSWORD</p> <p>Si vous utilisez la commande PASSWORD pour créer la définition de mot de passe numéroté pour la connexion à HighBond, aucune valeur de mot de passe n'est indiquée, et l'utilisateur est invité à saisir le mot de passe lorsque le script tente de se connecter.</p> <p>Pour plus d'informations, consultez la section "Commande PASSWORD" Page 373.</p> <p>Commande SET PASSWORD</p>

Élément	Détails
	<p>Si vous utilisez la commande SET PASSWORD pour créer la définition de mot de passe numéroté pour la connexion à HighBond, une valeur de mot de passe est indiquée, et l'utilisateur n'est pas invité à saisir le mot de passe, ce qui convient aux scripts conçus pour s'exécuter sans surveillance.</p> <p>Pour plus d'informations, consultez la rubrique <a href="#">Commande SET PASSWORD</a>.</p> <p>Jeton d'accès à HighBond</p> <p>Quelle que soit la méthode que vous utilisez pour créer la définition de mot de passe, la valeur de mot de passe requise est un jeton d'accès à HighBond :</p> <ul style="list-style-type: none"> <li>○ <b>Méthode PASSWORD</b> - Les utilisateurs peuvent acquérir un jeton d'accès en sélectionnant <b>Outils Jeton d'accès à HighBond</b>, puis en se connectant à HighBond. Un jeton d'accès est renvoyé, et les utilisateurs peuvent le copier et le coller dans la demande de mot de passe.</li> <li>○ <b>Méthode SET PASSWORD</b> : pour insérer un jeton d'accès dans la syntaxe de la commande SET PASSWORD dans un script Analytics, cliquez avec le bouton droit de la souris dans l'<b>Éditeur de script</b>, sélectionnez <b>Insérer &gt; Jeton HighBond</b>, puis connectez-vous à HighBond. Un jeton d'accès est inséré dans le script à l'emplacement du curseur.</li> </ul> <p><b>Attention</b></p> <p>Le jeton d'accès renvoyé correspond au compte utilisé pour se connecter à HighBond. En tant que rédacteur de scripts, il se peut qu'il ne soit pas très approprié d'utiliser votre propre jeton d'accès si vous écrivez un script qui sera utilisé par d'autres personnes.</p>

## Comment DISPLAYNAME interagit avec AS lors de l'exportation dans HighBond Résultats

La matrice ci-dessous illustre comment le paramètre DISPLAYNAME interagit avec AS lors de l'exportation des noms des champs d'Analytics vers Résultats.

	Sans AS	Avec AS
Sans DISPLAYNAME	Le nom du champ et le nom d'affichage dans Résultats sont identiques au nom du champ dans Analytics.	Le nom du champ et le nom d'affichage dans Résultats sont identiques au nom d'affichage dans le paramètre AS.
Avec DISPLAYNAME	Le nom du champ dans Résultats et celui dans Analytics sont identiques. Le nom d'affichage dans Résultats et celui dans Analytics sont identiques	Le nom du champ dans Résultats et celui dans Analytics sont identiques. Le nom d'affichage dans Résultats est identique au nom d'affichage dans le paramètre AS.

# Commande EXTRACT

Extrait des données d'une table Analytics et les sort dans une nouvelle table Analytics, ou les ajoute à une table Analytics existante. Vous pouvez extraire des enregistrements entiers ou des champs sélectionnés.

## Syntaxe

```
EXTRACT {RECORD | FIELDS nom_champ<AS nom_affichage> <...n> |FIELDS ALL} TO nom_table <IF test> <WHILE test> <FIRST plage|NEXT plage> <EOF> <APPEND> <OPEN> <LOCAL>
```

## Paramètres

Nom	Description
RECORD   FIELDS <i>nom_champ</i>   FIELDS ALL	<p>Les champs à inclure dans la sortie :</p> <ul style="list-style-type: none"> <li>○ <b>RECORD</b> : utilisez l'intégralité de l'enregistrement dans le fichier de données source : tous les champs de la table et toutes les parties indéfinies de l'enregistrement</li> </ul> <p>Les champs sont utilisés dans leur ordre d'apparition dans le format de table.</p> <p>Conserve les champs calculés.</p> <ul style="list-style-type: none"> <li>○ <b>FIELDS <i>nom_champ</i></b> : utilisez les champs spécifiés</li> </ul> <p>Les champs sont utilisés dans l'ordre dans lequel vous les indiquez.</p> <p>Convertit les champs calculés en champs physiques du type de données approprié dans la table de destination : ASCII ou Unicode (selon l'édition d'Analytics), ACL (type de données numérique natif), DateHeure ou Logique. Remplit les champs physiques avec les valeurs calculées réelles.</p> <ul style="list-style-type: none"> <li>○ <b>FIELDS ALL</b> : utilisez tous les champs dans la table</li> </ul> <p>Les champs sont utilisés dans leur ordre d'apparition dans le format de table.</p> <p>Convertit les champs calculés en champs physiques du type de données approprié dans la table de destination : ASCII ou Unicode (selon l'édition d'Analytics), ACL (type de données numérique natif), DateHeure ou Logique. Remplit les champs physiques avec les valeurs calculées réelles.</p>
AS <i>nom_affichage</i> optionnel	<p>Utilisé uniquement dans le cas d'une extraction à l'aide de FIELDS <i>nom_champ</i>.</p> <p>Le nom d'affichage (titre de la colonne) pour le champ dans la vue dans la nouvelle table Analytics. Si vous souhaitez que le nom d'affichage soit identique au nom du champ ou à un nom d'affichage existant dans la table source, n'utilisez pas AS.</p> <p>Indiquez le <i>nom_affichage</i> sous forme de chaîne entre guillemets. Utilisez un point-virgule (;) entre les mots si vous souhaitez insérer un saut de ligne dans le titre de la colonne.</p>

Nom	Description
	<p><b>Remarque</b></p> <p>AS fonctionne uniquement dans le cas de l'extraction dans une nouvelle table. Si vous effectuez un ajout dans une table existante, les autres titres de colonne de la table existante ont la priorité.</p>
TO <i>nom_table</i>	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li>◦ <b>nom_table</b> : enregistre les résultats dans une table Analytics</li> </ul> <p>Indiquez <i>nom_table</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple : TO "Sortie.FIL"</p> <p>Par défaut, le fichier de données de table (.FIL) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.FIL"</li> <li>• TO "Résultats\Sortie.FIL"</li> </ul> <p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alphanumériques, sans l'extension .FIL. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>
IF <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.</p> <p><b>Remarque</b></p> <p>La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p>
WHILE <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.</p> <p><b>Remarque</b></p> <p>Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.</p>
FIRST <i>plage</i>   NEXT <i>plage</i> optionnel	<p>Le nombre d'enregistrements à traiter :</p> <ul style="list-style-type: none"> <li>◦ <b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li>◦ <b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> </ul> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter.</p> <p>Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p>
EOF optionnel	<p>Réexécutez la commande une fois la fin du fichier atteinte.</p> <p>Cela garantit que le dernier enregistrement de la table est traité quand il se trouve à</p>

Nom	Description
	l'intérieur d'une commande GROUP. Utilisez le paramètre de traitement de fin de fichier uniquement si tous les champs sont calculés en se référant à des enregistrements antérieurs.
APPEND optionnel	<p>Ajoute la sortie de commande à la fin d'un fichier existant au lieu de remplacer ce fichier.</p> <p><b>Remarque</b></p> <p>Vous devez vous assurer que la structure de la sortie de la commande et du fichier existant est identique :</p> <ul style="list-style-type: none"> <li>• les mêmes champs</li> <li>• le même ordre des champs</li> <li>• les champs correspondants ont la même longueur</li> <li>• les champs correspondants ont le même type de données</li> </ul> <p>Analytics ajoute la sortie à un fichier existant sans tenir compte de sa structure, ce qui peut désordonner les données si la structure de la sortie et du fichier existant ne correspond pas.</p>
OPEN optionnel	Ouvre la table créée par la commande après l'exécution de la commande. Valide uniquement lorsque la commande crée une table de sortie.
LOCAL optionnel	<p>Enregistre le fichier de sortie au même emplacement que le projet Analytics.</p> <p><b>Remarque</b></p> <p>Applicable uniquement lorsque vous exécutez la commande sur une table de serveur avec un fichier de sortie qui est une table Analytics.</p>

## Exemples

### Extraction de tous les enregistrements d'une table dans une nouvelle table

Vous créez un doublon exact de la table **C\_Client** par l'extraction de tous les enregistrements dans une nouvelle table Analytics. Les champs calculés sont conservés comme champs calculés :

```
OPEN C_Client
EXTRACT RECORD TO "C_Client_2"
```

### Extraction de tous les champs d'une table vers une nouvelle table

Vous extrayez tous les champs définis de la table **C\_Client** vers une nouvelle table Analytics. Les champs calculés sont convertis en champs physiques et ils sont renseignés avec les valeurs calculées réelles :



```
OPEN C_Client
EXTRACT FIELDS ALL TO "C_Client_2"
```

## Extraction de tous les enregistrements d'une table et concaténation de ces derniers dans une table existante

Vous extrayez tous les enregistrements dans la table **C\_Client** et vous les concaténez comme un groupe à la fin de la table **C\_Client\_Principal** :

```
OPEN C_Client
EXTRACT RECORD TO "C_Client_Principal" APPEND
```

## Extraction de tous les enregistrements d'une table et concaténation de ces derniers dans une table existante d'un dossier différent

Vous extrayez tous les enregistrements dans la table **C\_Client** et vous les concaténez comme un groupe à la fin de la table **C\_Client\_Principal**, qui est un dossier différent du dossier du projet Analytics :

```
OPEN C_Client
EXTRACT RECORD TO "C:\Utilisateurs\Données client\C_Client_Principal" APPEND
```

## Extraction d'un sous-ensemble des champs d'une table vers une nouvelle table

Vous extrayez trois champs de la table **C\_Client** vers une nouvelle table Analytics :

```
OPEN C_Client
EXTRAIRE FIELDS Date Échéance Nom TO "C_Dates_Client.fil"
```

## Création de noms d'affichage pour des champs extraits

Vous extrayez trois champs de la table **C\_Client** et vous créez des noms d'affichage pour les champs dans la nouvelle table Analytics :

```
OPEN C_Client
EXTRACT FIELDS Nom AS "Client;Nom" Due AS "Échéance;Date" Date AS "Facture;Date" TO "C_Dates_Client.fil"
```

## Extraction de champs à partir d'une condition

Vous extrayez trois champs de la table **C\_Client** vers une nouvelle table Analytics si la date dans la colonne **Échéance** est antérieure au 1er juillet 2014 :

```
OPEN C_Client  
EXTRACT FIELDS Date Échéance Nom IF Échéance < `20140701` TO "Impayé.fil"
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## EXTRACT et copie d'une table

EXTRACT crée un nouveau fichier de données source (.fil) ainsi qu'un nouveau format de table.

Copier une table à l'aide du **Navigateur (Modifier > Copier)** crée un nouveau format de table restant associé avec le fichier de données source d'origine. Cela ne crée pas de nouveau fichier de données.

# Commande FIELDSHIFT

Déplace la position de départ d'une définition de champ dans un format de table.

## Syntaxe

```
FIELDSHIFT START position_départ COLUMNS octets_à_décaler <FILTER nom_filtre_données>
<OK>
```

## Paramètres

Nom	Description						
START <i>position_départ</i>	<p>Position de l'octet de départ de la première définition de champ que vous souhaitez décaler.</p> <p>Toutes les définitions de champs à droite de la définition de champ spécifiée sont aussi décalées.</p> <p>Si vous indiquez une position d'octet qui n'est pas une position de départ, c'est la position d'octet de départ suivante qui est utilisée.</p> <p><b>Remarque</b></p> <table border="1"> <tbody> <tr> <td>Analytics non Unicode</td> <td>1 octet = 1 caractère</td> </tr> <tr> <td>Analytics Unicode, données ASCII étendu (ANSI)</td> <td>1 octet = 1 caractère</td> </tr> <tr> <td>Analytics Unicode, données Unicode</td> <td>2 octets = 1 caractère</td> </tr> </tbody> </table> <p>Pour les données Unicode, normalement, vous devez indiquer une position d'octet de départ avec un nombre impair. Indiquer une position de départ avec un chiffre pair peut entraîner l'affichage incorrect des caractères.</p>	Analytics non Unicode	1 octet = 1 caractère	Analytics Unicode, données ASCII étendu (ANSI)	1 octet = 1 caractère	Analytics Unicode, données Unicode	2 octets = 1 caractère
Analytics non Unicode	1 octet = 1 caractère						
Analytics Unicode, données ASCII étendu (ANSI)	1 octet = 1 caractère						
Analytics Unicode, données Unicode	2 octets = 1 caractère						
COLUMNS <i>octets_à_décaler</i>	<p>Nombre d'octets dont décaler la définition de champ.</p> <p>Saisissez un nombre positif pour décaler une définition de champ vers la droite. Saisissez un nombre négatif pour décaler une définition de champ vers la gauche.</p>						

Nom	Description						
	<p><b>Remarque</b></p> <table border="1"> <tr> <td>Analytics non Unicode</td> <td>1 octet = 1 caractère</td> </tr> <tr> <td>Analytics Unicode, données ASCII étendu (ANSI)</td> <td>1 octet = 1 caractère</td> </tr> <tr> <td>Analytics Unicode, données Unicode</td> <td>2 octets = 1 caractère</td> </tr> </table> <p>Pour les données Unicode, spécifiez uniquement un nombre d'octets pair. Indiquer un nombre d'octets impair peut entraîner l'affichage incorrect des caractères.</p>	Analytics non Unicode	1 octet = 1 caractère	Analytics Unicode, données ASCII étendu (ANSI)	1 octet = 1 caractère	Analytics Unicode, données Unicode	2 octets = 1 caractère
Analytics non Unicode	1 octet = 1 caractère						
Analytics Unicode, données ASCII étendu (ANSI)	1 octet = 1 caractère						
Analytics Unicode, données Unicode	2 octets = 1 caractère						
FILTER <i>nom_filtre_données</i> optionnel	Nom du filtre qui identifie les définitions de champs associés à une définition d'enregistrement particulière.						
OK optionnel	Supprime ou écrase des éléments sans demander de confirmation.						

## Exemples

### Décaler des définitions de champ

Vous décalez la définition de champ à partir de l'octet 11, et vous décalez les définitions de champs suivantes de 4 octets vers la droite :

```
FIELDSHIFT START 11 COLUMNS 4
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

### Les définitions de champs décalées doivent rester à l'intérieur de la longueur d'enregistrement

Lorsque vous décalez une ou plusieurs définitions de champs vers la droite ou vers la gauche, les champs ne peuvent pas dépasser la longueur d'enregistrement dans un sens ou dans l'autre.

Retenez bien que FIELDSHIFT déplace à la fois la définition de champ spécifiée, mais aussi les définitions de champs à droite de la définition ayant été spécifiée. Si le bloc de définitions décalées dépasse la longueur d'enregistrement dans un sens ou dans l'autre, un message d'erreur s'affiche et la commande n'est pas exécutée.

#### **Astuce**

Si le message d'erreur s'affiche parce que vous dépassez la fin de l'enregistrement, tentez de supprimer la dernière définition de champ pour laisser de la place afin que les définitions des champs soient décalées.

# Commande FIND

Recherche, dans un champ caractère indexé, la première valeur qui correspond à la chaîne de caractères spécifiée.

## Remarque

La commande FIND et la fonction FIND( ) sont deux fonctionnalités Analytics distinctes présentant des différences significatives. Pour plus d'informations sur la fonction, consultez la rubrique "Fonction FIND( )" Page 596.

## Syntaxe

```
FIND rechercher_valeur
```

## Paramètres

Nom	Description
<i>rechercher_valeur</i>	Chaîne de caractères à rechercher. <i>rechercher_valeur</i> respecte la casse et ne peut pas inclure des espaces de début. Ne mettez pas la valeur entre guillemets sauf si ceux-ci font partie des données recherchées.

## Exemples

### Rechercher une valeur spécifique

Vous souhaitez trouver la première valeur du champ de type caractère **Num\_Carte** qui correspond exactement à "8590124" ou qui commence par ces chiffres.

Tout d'abord, vous indexez le champ **Num\_Carte** dans l'ordre croissant. Puis, vous exécutez FIND :

```
INDEX ON Num_Carte TO "NumCarte" OPEN
SET INDEX TO "NumCarte"
FIND 8590124
```

# Remarques

## Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Quand utiliser FIND

Utilisez la commande FIND pour accéder directement au premier enregistrement d'une table qui contient la valeur du paramètre *rechercher\_valeur* dans le champ caractère spécifié.

## Condition préalable pour INDEX

Pour utiliser la commande, la table dans laquelle vous effectuez la recherche doit être indexée sur un champ caractère dans l'ordre croissant.

Si plusieurs champs de type caractère sont indexés dans l'ordre croissant, seul le premier champ indiqué dans l'index est exploré. La commande ne peut pas être utilisée pour rechercher dans des champs d'index qui ne soient pas de type caractère ni des champs caractère indexés dans l'ordre décroissant.

## Rapprochement partiel

Le rapprochement partiel est pris en charge. La valeur de recherche peut être contenue par une valeur plus longue dans le champ indexé. Toutefois, la valeur de recherche doit apparaître au début du champ pour constituer un rapprochement.

## Sortie FIND dépendante de la correspondance

La commande FIND produit l'un des résultats suivants, selon la détection de la valeur de recherche ou non :

- **la valeur recherchée est trouvée** : le premier enregistrement correspondant de la table est sélectionné
- **la valeur de recherche n'est pas trouvée** : la table est placée sur le premier enregistrement présentant une valeur supérieure à la valeur de recherche

Si aucune valeur du champ indexé n'est supérieure à la valeur de recherche, la table est placée sur le premier enregistrement. Dans les deux cas, le message « Aucune correspondance de clé n'a été trouvée » s'affiche.

La commande FIND n'est pas affectée par l'option **Comparaisons exactes de caractères** (SET EXACT ON/OFF).

# Commande FUZZYDUP

Détecte des valeurs presque identiques (doublons approximatifs) dans un champ de type caractère.

## Remarque

Pour utiliser les correspondances approximatives afin de combiner des champs de deux tables Analytics dans une nouvelle table Analytics unique, reportez-vous à la rubrique "Commande FUZZYJOIN" Page 222.

## Syntaxe

```
FUZZYDUP ON champ_clé <OTHER champs> LEVDISTANCE valeur <DIFFPCT valeur>
<RESULTSIZ pourcentage> <EXACT> <IF test> TO nom_table <LOCAL> <OPEN>
```

## Paramètres

Nom	Description
ON <i>champ_clé</i>	Le champ ou l'expression de type caractère dans lesquels rechercher les doublons approximatifs.
OTHER <i>champs</i> optionnel	Liste de champs ou d'expressions à inclure dans la sortie.
LEVDISTANCE <i>valeur</i>	<p>La distance Levenshtein maximale autorisée entre deux chaînes pour qu'elles soient identifiées en tant que doublons approximatifs et pour qu'elles soient incluses dans les résultats.</p> <p>La valeur LEVDISTANCE ne peut pas être inférieure à 1 ni supérieure à 10. Le fait d'augmenter la valeur LEVDISTANCE augmente le nombre de résultats en incluant des valeurs d'un niveau d'approximation supérieur, c'est-à-dire des valeurs qui diffèrent davantage les unes des autres.</p> <p>Pour plus d'informations, consultez la section "Comportement de FUZZYDUP" Page 219.</p>
DIFFPCT <i>pourcentage</i> optionnel	<p>Un seuil délimitant le « pourcentage de différence » ou la proportion d'une chaîne pouvant être différente.</p> <p>Le pourcentage qui résulte d'un calcul Analytics interne effectué sur des paires de doublons approximatifs potentiels doit être inférieur ou égal à la valeur DIFFPCT de la paire à inclure dans les résultats. La valeur DIFFPCT ne peut pas être inférieure à 1 ni supérieure à 99.</p> <p>Si DIFFPCT est ignoré, le seuil est désactivé et le pourcentage de différence n'est pas pris en compte lors du traitement de la commande FUZZYDUP.</p>



Nom	Description
	<p>Pour plus d'informations, consultez la section "Comportement de FUZZYDUP" Page 219.</p>
<p>RESULTSIZE <i>pourcentage</i> optionnel</p>	<p>Taille maximale de l'ensemble des résultats de sortie sous forme de pourcentage du nombre d'enregistrements dans le champ clé.</p> <p>Par exemple, dans le cas d'un champ clé comportant 50 000 enregistrements, un RESULTSIZE de 3 arrêterait le traitement lorsque les résultats dépasseraient 1500 doublons approximatifs (50 000 x 0,03). Aucune table de sortie n'est générée si le traitement est arrêté.</p> <p>La valeur RESULTSIZE ne peut pas être inférieure à 1 ni supérieure à 1 000 (mille) pour cent. La limite de 1000 % doit s'adapter à la nature de la correspondance plusieurs-à-plusieurs, qui peut générer des résultats plus nombreux que l'ensemble de données de test d'origine.</p> <p>Si RESULTSIZE est ignoré, le seuil est désactivé et la taille du résultat n'est pas prise en compte lors du traitement de la commande FUZZYDUP.</p> <p><b>Attention</b></p> <p>Ignorer RESULTSIZE peut produire un jeu de résultats trop grand, lent à traiter et excédant la mémoire disponible, ce qui provoque l'arrêt du traitement. Ignorez RESULTSIZE uniquement si vous êtes sûr(e) de produire des résultats de taille gérable.</p>
<p>EXACT optionnel</p>	<p>Inclut les doublons exacts ainsi que les doublons approximatifs dans les résultats.</p>
<p>IF <i>test</i> optionnel</p>	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.</p> <p><b>Remarque</b></p> <p>La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p>
<p>TO <i>nom_table</i></p>	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li>◦ <b>nom_table</b> : enregistre les résultats dans une table Analytics</li> </ul> <p>Indiquez <i>nom_table</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple : TO "Sortie.FIL"</p> <p>Par défaut, le fichier de données de table (.FIL) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.FIL"</li> <li>• TO "Résultats\Sortie.FIL"</li> </ul>

Nom	Description
	<p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alpha-numériques, sans l'extension .FIL. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>
LOCAL optionnel	<p>Enregistre le fichier de sortie au même emplacement que le projet Analytics.</p> <p><b>Remarque</b></p> <p>Applicable uniquement lorsque vous exécutez la commande sur une table de serveur avec un fichier de sortie qui est une table Analytics.</p>
OPEN optionnel	<p>Ouvre la table créée par la commande après l'exécution de la commande. Valide uniquement lorsque la commande crée une table de sortie.</p>

## Variables de sortie Analytics

Nom	Contient
GAPDUP $n$	Nombre total d'omissions, de doublons ou de groupes de doublons approximatifs identifiés par la commande.

## Exemples

### Recherche de doublons approximatifs dans un champ de nom

Les doublons approximatifs sont recherchés dans un champ de nom (le champ **Nom** de la table « **Liste\_Employés** » dans `ACL DATA\Exemples de fichiers de données\Données_Employés_Metaphor.ACL`). Les résultats sont renvoyés vers une nouvelle table Analytics.

- En plus du champ test, d'autres champs sont inclus dans les résultats.
- La distance Levenshtein maximale autorisée est 1.
- La marge de différence d'une chaîne est limitée à 50 %.
- La taille des résultats est limitée à 20 % de la taille du champ test.
- En plus de doublons approximatifs, des doublons exacts sont inclus.

```

FUZZYDUP ON Nom OTHER Prénom NoEmp LEVDISTANCE 1 DIFFPCT 50 RESULTSIZE 20
EXACT TO "Nom_Approximatif" OPEN

```

# Remarques

## Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Fonctionnement

La commande FUZZYDUP recherche des valeurs presque identiques (des doublons approximatifs) ou repère les incohérences orthographiques dans les données saisies manuellement.

Contrairement à la fonction ISFUZZYDUP( ), qui identifie une liste exhaustive de doublons approximatifs pour une valeur de caractère unique, la commande FUZZYDUP identifie tous les doublons approximatifs d'un champ, les organise en groupes et renvoie des résultats non-exhaustifs.

## Qu'entend-on par « non-exhaustif » ?

Non-exhaustif signifie que dans les résultats, les groupes de doublons approximatifs individuels peuvent ne pas inclure tous les doublons approximatifs d'un champ test qui se situent dans la marge de différence définie pour le propriétaire du groupe. Cependant, si un propriétaire de groupe est un doublon approximatif d'une autre valeur du champ test, les deux valeurs sont affichées ensemble dans un groupe, dans les résultats.

Si la génération d'une liste exhaustive de doublons approximatifs pour une valeur spécifique du champ de test est importante pour votre analyse, vous pouvez utiliser la fonction ISFUZZYDUP( ) pour le faire.

## Comportement de FUZZYDUP

Dans la commande FUZZYDUP, vous pouvez spécifier deux paramètres qui contrôlent la marge de différence entre les doublons approximatifs et la taille des résultats :

- LEVDISTANCE
- DIFFPCT

Vous devrez peut-être essayer différentes combinaisons de paramètres pour ces deux paramètres afin de trouver la combinaison la mieux adaptée à un ensemble de données spécifique.

### LEVDISTANCE (distance Levenshtein)

Lors du traitement des données, la commande FUZZYDUP calcule la distance Levenshtein entre chaque paire de chaînes évaluée dans le champ test, puis calcule le pourcentage de différence. La distance Levenshtein représente le nombre minimal de modifications d'un caractère unique requises pour rendre une chaîne identique à une autre chaîne. Pour plus d'informations, consultez la section "Fonction LEVDIST( )" Page 657.

## DIFFPCT (pourcentage de différence)

Le pourcentage de différence correspond au pourcentage de la plus courte des deux chaînes évaluées et qui présente des dissemblances. Le pourcentage de différence est le résultat du calcul Analytics interne suivant (ce calcul se base sur la distance Levenshtein séparant les deux chaînes) :

*Distance Levenshtein / nombre de caractères dans la chaîne plus courte × 100 = pourcentage de différence*

### Plus d'informations

Pour plus d'informations sur les paramètres de différence entre les doublons approximatifs, le contrôle de la taille du résultat et les groupes de doublons approximatifs, consultez la [Présentation des doublons approximatifs](#).

## Sensible à la casse

La commande FUZZYDUP n'est pas sensible à la casse (par ex., « MARTIN » est équivalent à « martin »).

## Troncage automatique des espaces de fin

La commande FUZZYDUP tronque automatiquement les espaces de fin dans *champ\_clé*. Vous n'avez donc pas besoin d'utiliser la fonction TRIM( ) ou ALLTRIM( ) lorsque vous renseignez un seul champ comme *champ\_clé*.

Si vous concaténez des champs pour *key\_field*, vous devez utiliser ALLTRIM( ), comme indiqué ci-dessous.

## Améliorer l'efficacité de FUZZYDUP

### Concaténation de champs

La concaténation de deux champs test ou plus peut améliorer l'efficacité de la commande FUZZYDUP en augmentant le degré d'unicité des valeurs test.

Par exemple :

```
FUZZYDUP ON ALLTRIM(Prénom)+ALLTRIM(Nom) OTHER Prénom Nom NoEmp
LEVDISTANCE 4 DIFFPCT 50 RESULTSIZE 20 EXACT TO "Nom_Prénom_Approximatif" OPEN
```

### Fonction OMIT( )

La fonction OMIT( ) peut également améliorer l'efficacité de la commande en supprimant les éléments génériques tels que « Corporation » ou « Inc. ». des valeurs de champ.

La suppression des éléments génériques permet d'axer la comparaison FUZZYDUP sur la portion des chaînes qui peut inclure une différence significative.

Pour plus d'informations, consultez la section "Fonction OMIT( )" Page 709.

## Autres méthodes de comparaison de chaînes de caractères

- La fonction **DICECOEFFICIENT( )** : fournit une méthode de comparaison des chaînes qui ne met pas l'accent sur la position des caractères ou des blocs de caractères voire les ignore complètement.
- Les fonctions **SOUNDSLIKE( )** et **SOUNDEX( )** : fournissent une méthode de comparaison des chaînes de type phonétique (son) plutôt qu'orthographique.

# Commande FUZZYJOIN

Utilise les correspondances approximatives pour combiner des champs de deux tables Analytics dans une nouvelle table Analytics unique.

## Remarque

Pour détecter des valeurs presque identiques (doublons approximatifs) dans un seul champ de type caractère, consultez la section "Commande FUZZYDUP" Page 216.

Pour connaître les différentes options lors de la jointure de tables à l'aide des valeurs de champs clés correspondant exactement, consultez la rubrique "Commande JOIN" Page 338.

## Syntaxe

```
FUZZYJOIN {DICE PERCENT pourcentage NGRAM longueur_n-grammes|LEVDISTANCE
DISTANCE valeur} PKEY champ_clé_principaleSKEY champ_clé_secondaire {FIELDS champs_principaux|FIELDS ALL} <WITH champs_secondaires|WITH ALL> <IF test> <OPEN>
<FIRSTMATCH> TO nom_table <WHILE test> <FIRST plage|NEXT plage> <APPEND>
```

## Remarque

Vous ne pouvez pas exécuter la commande FUZZYJOIN localement par rapport à une table de serveur.

Vous devez indiquer en entier le nom de la commande FUZZYJOIN. Vous ne pouvez pas l'abréger.

## Paramètres

Nom	Description
DICE PERCENT <i>pourcentage</i> NGRAM <i>longueur_n-grammes</i>   LEVDISTANCE DISTANCE <i>valeur</i>	<p>L'algorithme de correspondance approximative à utiliser.</p> <p><b>DICE</b> : utilise l'algorithme du coefficient de Dice</p> <ul style="list-style-type: none"> <li> <b>PERCENT <i>pourcentage</i></b> : le coefficient minimum autorisé de deux chaînes de caractères pour qu'elles puissent être qualifiées de correspondance approximative           <p>Spécifiez un nombre décimal, compris entre 0,0000 et 1,0000 (par exemple, 0,7500). Utilisez quatre décimales au maximum.</p> <p>Le fait de diminuer cette valeur augmente le nombre de correspondances en incluant des correspondances d'un niveau d'approximation supérieur, c'est-à-dire des chaînes de caractères qui diffèrent davantage les unes des autres.</p> </li> <li> <b>NGRAM <i>longueur_n-grammes</i></b> : la longueur <i>n-grammes</i> à utiliser           <p>Spécifiez un nombre entier, 1 ou plus.</p> </li> </ul>

Nom	Description
	<p>Augmenter la longueur <math>n</math>-grammes rend encore plus strict le critère de similarité entre deux chaînes de caractères. Les</p> <p><math>N</math>-grammes sont les sous-chaînes (blocs de caractères) qui se chevauchent et parmi lesquelles les chaînes de comparaison sont divisées dans le cadre du calcul du coefficient de Dice.</p> <p><b>Remarque</b></p> <p>Lorsque vous spécifiez DICE, la commande FUZZYJOIN utilise la fonction DICECOEFFICIENT( ) dans une instruction IF pour joindre conditionnellement des valeurs de champs clés. Pour des informations détaillées sur la fonction, consultez la section "Fonction DICECOEFFICIENT( )" Page 569.</p> <p><b>LEVDISTANCE</b> : utilise l'algorithme de distance Levenshtein</p> <ul style="list-style-type: none"> <li>◦ <b>DISTANCE valeur</b> : la distance de Levenshtein maximale permise entre deux chaînes de caractères pour qu'elles puissent être qualifiées de correspondance approximative</li> </ul> <p>Spécifiez un nombre entier, 1 ou plus.</p> <p>Le fait d'augmenter cette valeur augmente le nombre de correspondances en incluant des correspondances d'un niveau d'approximation supérieur, c'est-à-dire des chaînes de caractères qui diffèrent davantage les unes des autres.</p> <p><b>Remarque</b></p> <p>Lorsque vous spécifiez LEVDISTANCE, la commande FUZZYJOIN utilise la fonction LEVDIST( ) dans une instruction IF pour joindre conditionnellement des valeurs de champs clés. Pour des informations détaillées sur la fonction, consultez la section "Fonction LEVDIST( )" Page 657.</p> <p>Contrairement à la fonction, l'algorithme de distance Levenshtein de la commande FUZZYJOIN tronque automatiquement les espaces de début et de fin, et n'est pas sensible à la casse.</p>
PKEY <i>champ_clé_principale</i>	<p>Le champ clé, ou l'expression clé, de type caractère dans la table principale.</p> <p>Vous ne pouvez spécifier qu'un seul champ clé principal.</p>
SKEY <i>champ_clé_secondaire</i>	<p>Le ou les champs clés, ou l'expression clé, de type caractère dans la table secondaire.</p> <p>Vous ne pouvez spécifier qu'un seul champ clé secondaire.</p>
FIELDS <i>champs_principaux</i>   FIELDS ALL	<p>Les champs ou expressions de la table principale à inclure dans la table de sortie jointe.</p> <ul style="list-style-type: none"> <li>◦ <b>champs_principaux</b> : incluez le(s) champ(s) spécifié(s)</li> <li>◦ <b>ALL</b> : incluez tous les champs de la table</li> </ul> <p><b>Remarque</b></p> <p>Vous devez indiquer de manière explicite le champ clé principal si vous voulez l'intégrer dans la table jointe. Le fait d'indiquer ALL permet aussi de l'inclure.</p>
WITH <i>champs_secondaire</i>	<p>Les champs ou expressions de la table secondaire à inclure dans la table de sortie</p>

Nom	Description
<i>dares</i>   WITH ALL optionnel	<p>jointe.</p> <ul style="list-style-type: none"> <li>◦ <b>champs_secondaires</b> : incluez le(s) champ(s) spécifié(s)</li> <li>◦ <b>ALL</b> : incluez tous les champs de la table</li> </ul> <p><b>Remarque</b></p> <p>Vous devez indiquer de manière explicite le champ clé secondaire si vous voulez l'intégrer dans la table jointe. Le fait d'indiquer ALL permet aussi de l'inclure.</p>
IF <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.</p> <p><b>Remarque</b></p> <p>La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p> <p><b>Remarque</b></p> <p>La condition IF peut faire référence à la table principale, à la table secondaire ou aux deux.</p>
OPEN optionnel	<p>Ouvre la table créée par la commande après l'exécution de la commande. Valide uniquement lorsque la commande crée une table de sortie.</p>
FIRSTMATCH optionnel	<p>Spécifie que chaque valeur de clé primaire n'est jointe qu'à la première occurrence d'une correspondance de clé secondaire.</p> <p>S'il se trouve que la première occurrence est une correspondance exacte, les correspondances approximatives suivantes pour la valeur de clé primaire ne sont pas incluses dans la table de sortie jointe.</p> <p>Si vous omettez FIRSTMATCH, le comportement par défaut de FUZZYJOIN consiste à joindre chaque valeur de clé primaire à toutes les occurrences des correspondances de clés secondaires.</p> <p>FIRSTMATCH est utile si vous voulez seulement savoir s'il existe des correspondances, exactes ou approximatives, entre deux tables, et si vous souhaitez réduire le temps de traitement nécessaire pour identifier toutes les correspondances.</p> <p>Vous pouvez également utiliser FIRSTMATCH si vous êtes certain qu'il n'existe qu'une seule correspondance dans la table secondaire pour chaque valeur de clé primaire.</p> <p><b>Remarque</b></p> <p>FIRSTMATCH n'est disponible qu'en tant que paramètre ACLScript. L'option n'est pas disponible dans l'interface utilisateur Analytics.</p>
TO <i>nom_table</i>	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li>◦ <b>nom_table</b> : enregistre les résultats dans une table Analytics</li> </ul> <p>Indiquez <i>nom_table</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple : TO "Sortie.FIL"</p> <p>Par défaut, le fichier de données de table (.FIL) est enregistré dans le dossier contenant le projet Analytics.</p>



Nom	Description
	<p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.FIL"</li> <li>• TO "Résultats\Sortie.FIL"</li> </ul> <p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alphanumériques, sans l'extension .FIL. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>
WHILE <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.</p> <p><b>Remarque</b></p> <p>Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.</p>
FIRST <i>plage</i>   NEXT <i>plage</i> optionnel	<p>Le nombre d'enregistrements à traiter :</p> <ul style="list-style-type: none"> <li>◦ <b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li>◦ <b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> </ul> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter.</p> <p>Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p>
APPEND optionnel	<p>Ajoute la sortie de commande à la fin d'un fichier existant au lieu de remplacer ce fichier.</p> <p><b>Remarque</b></p> <p>Vous devez vous assurer que la structure de la sortie de la commande et du fichier existant est identique :</p> <ul style="list-style-type: none"> <li>• les mêmes champs</li> <li>• le même ordre des champs</li> <li>• les champs correspondants ont la même longueur</li> <li>• les champs correspondants ont le même type de données</li> </ul> <p>Analytics ajoute la sortie à un fichier existant sans tenir compte de sa structure, ce qui peut désordonner les données si la structure de la sortie et du fichier existant ne correspond pas.</p>
ISOLOCALE <i>code_paramètres_régionaux</i> optionnel	<p><b>Remarque</b></p> <p>Applicable dans l'édition Unicode d'Analytics uniquement.</p> <p>Paramètre système local au format <i>langue_pays</i>. Par exemple, pour utiliser le français canadien, saisissez fr_ca.</p> <p>Utilisez les codes suivants :</p> <ul style="list-style-type: none"> <li>◦ <b>langue</b> : code linguistique de la norme ISO 639</li> <li>◦ <b>pays</b> : code pays de la norme ISO 3166</li> </ul>

Nom	Description
	<p>Si vous n'indiquez pas de code pays, c'est le pays par défaut pour la langue qui est utilisé.</p> <p>Si vous n'utilisez pas ISOLOCALE, ce sont les paramètres système régionaux par défaut qui sont utilisés.</p>

## Exemples

### Utilisez les correspondances approximatives pour joindre deux tables afin de détecter les employés susceptibles d'être des fournisseurs

L'exemple ci-dessous joint les tables Empmast et Fournisseur en utilisant l'adresse comme champ clé commun (champs Adresse et Rue\_Fournisseur).

La commande FUZZYJOIN crée une nouvelle table avec des enregistrements primaires et secondaires qui sont des correspondances exactes ou approximatives. Le résultat est une liste de tous les employés et fournisseurs ayant soit une adresse identique, soit une adresse similaire.

#### FUZZYJOIN avec l'algorithme du coefficient de Dice

```
OPEN Empmast PRIMARY
OPEN Fournisseur SECONDARY
FUZZYJOIN DICE PERCENT 0,8000 NGRAM 2 PKEY Adresse SKEY Rue_Fournisseur FIELDS
Numéro_Employé Prénom Nom Adresse WITH Numéro_fournisseur Nom_fournisseur Rue_Four-
nisseur OPEN TO "Correspondance_employé_fournisseur"
```

#### FUZZYJOIN avec l'algorithme de distance Levenshtein

```
OPEN Empmast PRIMARY
OPEN Fournisseur SECONDARY
FUZZYJOIN LEVDISTANCE DISTANCE 5 PKEY Adresse SKEY Rue_Fournisseur FIELDS
Numéro_Employé Prénom Nom Adresse WITH Numéro_fournisseur Nom_fournisseur Rue_Four-
nisseur OPEN TO "Correspondance_employé_fournisseur"
```

#### Inclure tous les champs

Cette version de la commande FUZZYJOIN inclut tous les champs des tables principale et secondaires dans la table de sortie jointe.

```
OPEN Empmast PRIMARY  
OPEN Fournisseur SECONDARY  
FUZZYJOIN LEVDISTANCE DISTANCE 5 PKEY Adresse SKEY Rue_Fournisseur FIELDS ALL  
WITH ALL OPEN TO "Correspondance_employé_fournisseur"
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Respect de la casse

La commande FUZZYJOIN n'est pas sensible à la casse, quel que soit l'algorithme de correspondances approximatives que vous utilisez. "SMITH" est donc équivalent à "smith".

## Espaces de début et de fin

La commande FUZZYJOIN tronque les espaces de début et de fin dans les champs de manière automatique, quel que soit l'algorithme de correspondances approximatives que vous utilisez. Il n'est pas nécessaire d'utiliser les fonctions TRIM( ) ou ALLTRIM( ) pour spécifier les champs clés principaux et secondaires.

# Commande GAPS

Détecte si un champ numérique ou DateHeure dans une table Analytics contient une ou plusieurs omissions dans des données séquentielles.

## Syntaxe

```
GAPS <ON> champ_clé <D> <UNFORMATTED> <PRESORT> <MISSING limite>
<HEADER texte_en-tête> <FOOTER texte_pied_page> <IF test> <WHILE test> <FIRST
plage|NEXT plage> <TO {SCREEN|nom_table|nom_fichier|PRINT}> <APPEND> <LOCAL>
<OPEN>
```

## Paramètres

Nom	Description
ON <i>champ_clé</i> D	Champs ou expressions dans lesquels rechercher des omissions. Incluez D pour trier le champ clé dans l'ordre décroissant. L'ordre de tri par défaut est croissant.
UNFORMATTED optionnel	Supprime les en-têtes et les sauts de page lorsque les résultats sont renvoyés dans un fichier.
PRESORT optionnel	Trie la table par le champ clé avant d'exécuter la commande.  <div style="border-left: 2px solid #0056b3; padding-left: 10px; margin-left: 20px;"> <p><b>Remarque</b></p> <p>Vous ne pouvez pas utiliser PRESORT dans la commande GROUP.</p> </div>
MISSING <i>limite</i> optionnel	Les résultats de sortie contiennent des éléments manquants individuels plutôt que des plages d'omission.  La valeur <i>limite</i> indique le nombre maximal d'éléments manquants à rapporter pour chaque omission identifiée. La valeur par défaut est 5. Si la limite est dépassée pour une omission particulière, les éléments manquants sont rapportés sous forme de plage pour cette omission.  La valeur <i>limite</i> ne restreint pas le nombre total d'éléments manquants rapportés, mais uniquement le nombre d'éléments manquants rapportés dans une omission spécifique.
HEADER <i>texte_en-tête</i> optionnel	Texte à insérer en haut de chaque page d'un rapport.  <i>texte_en-tête</i> doit être indiqué sous forme de chaîne entre guillemets. La valeur remplace la variable système Analytics HEADER.
FOOTER <i>texte_pied_page</i>	Texte à insérer en bas de chaque page d'un rapport.

Nom	Description
optionnel	<i>texte_pied_page</i> doit être indiqué sous forme de chaîne entre guillemets. La valeur remplace la variable système Analytics FOOTER.
IF <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.</p> <p><b>Remarque</b></p> <p>La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p>
WHILE <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.</p> <p><b>Remarque</b></p> <p>Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.</p>
FIRST <i>plage</i>   NEXT <i>plage</i> optionnel	<p>Le nombre d'enregistrements à traiter :</p> <ul style="list-style-type: none"> <li>◦ <b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li>◦ <b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> </ul> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter.</p> <p>Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p>
TO SCREEN   <i>nom_table</i>   <i>nom_fichier</i>   PRINT optionnel	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li>◦ <b>SCREEN</b> : affiche les résultats dans la zone d'affichage d'Analytics.</li> <li>◦ <b><i>nom_table</i></b> : enregistre les résultats dans une table Analytics</li> </ul> <p>Indiquez <i>nom_table</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple : TO "Sortie.FIL"</p> <p>Par défaut, le fichier de données de table (.FIL) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.FIL"</li> <li>• TO "Résultats\Sortie.FIL"</li> </ul> <p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alphanumériques, sans l'extension .FIL. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p> <ul style="list-style-type: none"> <li>◦ <b><i>nom_fichier</i></b> : enregistre les résultats dans un fichier</li> </ul> <p>Indiquez <i>nom_fichier</i> sous forme de chaîne entre guillemets avec l'extension de fichier appropriée. Par exemple : TO "Sortie.TXT"</p>

Nom	Description
	<p>Par défaut, le fichier est enregistré dans le dossier contenant le projet Analytics. Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.TXT"</li> <li>• TO "Résultats\Sortie.TXT"</li> <li>○ <b>PRINT</b> : envoie les résultats vers l'imprimante par défaut</li> </ul>
APPEND optionnel	<p>Ajoute la sortie de commande à la fin d'un fichier existant au lieu de remplacer ce fichier.</p> <p><b>Remarque</b></p> <p>Vous devez vous assurer que la structure de la sortie de la commande et du fichier existant est identique :</p> <ul style="list-style-type: none"> <li>• les mêmes champs</li> <li>• le même ordre des champs</li> <li>• les champs correspondants ont la même longueur</li> <li>• les champs correspondants ont le même type de données</li> </ul> <p>Analytics ajoute la sortie à un fichier existant sans tenir compte de sa structure, ce qui peut désordonner les données si la structure de la sortie et du fichier existant ne correspond pas.</p>
LOCAL optionnel	<p>Enregistre le fichier de sortie au même emplacement que le projet Analytics.</p> <p><b>Remarque</b></p> <p>Applicable uniquement lorsque vous exécutez la commande sur une table de serveur avec un fichier de sortie qui est une table Analytics.</p>
OPEN optionnel	<p>Ouvre la table créée par la commande après l'exécution de la commande. Valide uniquement lorsque la commande crée une table de sortie.</p>

## Variables de sortie Analytics

Nom	Contient
GAPDUP $n$	Nombre total d'omissions, de doublons ou de groupes de doublons approximatifs identifiés par la commande.

## Exemples

### Recherche de numéro de facture manquant

Vous utilisez GAPS pour vérifier qu'aucun numéro de facture ne manque dans une table **Factures** :

```
OPEN Factures  
GAPS ON Num_Fac PRESORT TO "Omissions_Factures.fil"
```

## Remarques

### Utilisation de GAPS sur des champs de type caractère

En plus de tester les champs numériques ou DateHeure, vous pouvez aussi rechercher des omissions dans des données numériques apparaissant dans un champ de type caractère. Par exemple, vous pouvez faire un test sur les numéros de chèque, qui sont généralement mis en forme en tant que données de type caractère.

Si des lettres et des nombres apparaissent en même temps dans un champ de type caractère, seuls les nombres sont testés, les lettres sont ignorées.

# Commande GROUP

Exécute une ou plusieurs commandes ACLScript sur un enregistrement avant de passer à l'enregistrement suivant dans la table, avec uniquement une seule transmission dans la table. L'exécution de la commande peut être contrôlée par des conditions.

## Syntaxe

```
GROUP <IF test> <WHILE test> <FIRST plage|NEXT plage>
  commande
  <...n>
<ELSE IF test>
  commande
  <...n>
<ELSE>
  commande
  <...n>
END
```

### Remarque

Certaines commandes Analytics ne peuvent pas être utilisées avec la commande GROUP. Pour plus d'informations, consultez la section "Commandes utilisables dans la commande GROUP" Page 236.

## Paramètres

Nom	Description
IF <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.</p> <p><b>Remarque</b> La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p>
WHILE <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.</p> <p><b>Remarque</b> Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.</p>



Nom	Description
FIRST <i>plage</i>   NEXT <i>plage</i> optionnel	<p>Le nombre d'enregistrements à traiter :</p> <ul style="list-style-type: none"> <li>◦ <b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li>◦ <b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> </ul> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter.</p> <p>Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p>
<i>commande &lt;...n&gt;</i>	<p>Une ou plusieurs commandes ACLScript à exécuter à l'intérieur de GROUP. Pour obtenir la liste complète des commandes prises en charge à l'intérieur de GROUP, consultez la rubrique "Commandes utilisables dans la commande GROUP" Page 236.</p> <p>Si IF ou ELSE IF le précède, le test doit être évalué sur « true ».</p> <p>Lorsque la commande est répertoriée sous ELSE, elle est exécutée si des enregistrements n'ont été traités par aucune commande précédente. Vous pouvez inclure plusieurs commandes, chacune commençant sur une ligne séparée.</p>
ELSE IF <i>test</i> optionnel	<p>Ouvre un bloc ELSE IF pour la commande GROUP. La condition teste les enregistrements qui ne correspondent pas au test de la commande GROUP ou à aucun test ELSE IF précédent.</p> <p>Vous pouvez inclure plusieurs tests ELSE IF. Ils sont évalués de haut en bas jusqu'à ce qu'un enregistrement soit évalué comme vrai et les commandes suivant cette instruction ELSE IF sont exécutées.</p>
ELSE optionnel	<p>Ouvre un bloc ELSE pour la commande GROUP. Les commandes suivantes sont exécutées pour les enregistrements évalués comme faux dans tous les tests antérieurs.</p>
END	<p>Fin de la commande GROUP.</p>

## Exemples

### GROUP S=simple

Les groupes simples commencent par une commande GROUP, suivie d'une série de commandes, et se terminent par une commande END :

```
GROUP
COUNT
HISTOGRAM ON Quantité MINIMUM 0 MAXIMUM 100 INTERVALS 10
CLASSIFY ON Emplacement SUBTOTAL Quantité
END
```

### GROUP IF

Les groupes conditionnels exécutent des commandes qui dépendent de l'état vrai ou faux d'une condition. La commande GROUP suivante est exécutée uniquement sur les enregistrements dont la valeur du champ **Classe\_produits** est inférieure à 5 :

```
GROUP IF Classe_produits < "05"
COUNT
HISTOGRAM ON Quantité MINIMUM 0 MAXIMUM 100 INTERVALS 10
CLASSIFY ON Emplacement SUBTOTAL Quantité
END
```

## GROUP IF ...ELSE

Les enregistrements qui ne remplissent pas la condition de test sont ignorés, sauf si un bloc ELSE est inclus.

L'instruction ELSE peut être suivie d'autant de commandes que vous le souhaitez. Dans l'exemple suivant, le total du champ **Quantité** est calculé pour tous les enregistrements qui ne remplissent pas la condition :

```
GROUP IF Classe_produits < "05"
COUNT
HISTOGRAM ON Quantité MINIMUM 0 MAXIMUM 100 INTERVALS 10
CLASSIFY ON Emplacement SUBTOTAL Quantité
ELSE
TOTAL Quantité
END
```

## GROUP IF...ELSE IF...ELSE

Vous pouvez inclure plusieurs blocs ELSE IF dans un groupe, à condition que chaque block ELSE IF contienne un test différent. Dans l'exemple suivant, les blocs ELSE IF et le bloc ELSE génèrent quatre totaux :

```
GROUP IF Classe_produits < "05"
COUNT
HISTOGRAM ON Quantité MINIMUM 0 MAXIMUM 100 INTERVALS 10
CLASSIFY ON Emplacement SUBTOTAL Quantité
ELSE IF Classe_produits = "05"
TOTAL Quantité
ELSE IF Classe_produits = "06"
TOTAL Quantité
ELSE IF Classe_produits = "07"
TOTAL Quantité
ELSE
```

```
TOTAL Quantité
END
```

## Commandes GROUP imbriquées

Les groupes imbriqués sont des groupes insérés dans d'autres groupes. Ils permettent de contrôler efficacement quelles commandes sont exécutées, et pour quels enregistrements. La plupart des applications ne nécessitent pas de fonctionnalités aussi avancées, mais elles sont disponibles.

Comme dans d'autres groupes, utilisez la commande END pour terminer un groupe imbriqué. Analytics commence le traitement des données uniquement lorsque toutes les commandes du groupe sont terminées :

```
GROUP IF Classe_produits < "05"
  COUNT
  STRATIFY ON Quantité SUBTOTAL Quantité MIN 0 MAX 100 INT 10
  GROUP IF Quantité > 0
    STATISTICS ON Quantité
    HISTOGRAM ON Quantité
  END
ELSE
  TOTAL Quantité
END
```

Dans cet exemple, toutes les commandes, de la commande COUNT à la commande GROUP suivante incluse, sont exécutées uniquement si la valeur **Classe\_produits** est inférieure à 5.

Les commandes STATISTICS et HISTOGRAM sont exécutées si la valeur **Quantité** est supérieure à zéro. Cependant, la deuxième commande GROUP étant imbriquée, les commandes STATISTICS et HISTOGRAM sont exécutées uniquement pour les enregistrements qui remplissent les conditions **Classe\_produits** < "05" et **Quantité** > 0.

## Génération de variables système à l'intérieur d'un GROUP

Vous pouvez utiliser GROUP pour créer plusieurs variables système pour une seule commande.

Normalement, lorsque vous exécutez une commande comme TOTAL, COUNT ou STATISTICS, une seule variable système est générée. Chaque fois que vous exécutez la commande, vous écrasez la valeur de la dernière exécution de la commande. Les commandes qui s'exécutent dans un GROUP créent une variable spécifique pour chaque instance de la commande à l'intérieur du GROUP.

Dans cet exemple, la commande TOTAL calcule la somme du champ **Montant** pour chaque classe de produit dans la table **Trans\_Metaphor\_2002**. Lorsque le code s'exécute, les variables suivantes sont générées et peuvent être utilisées dans les commandes suivantes après le GROUP :

- **TOTAL2** : la somme du champ **Montant** pour la classe de produit 03
- **TOTAL3** : la somme du champ **Montant** pour la classe de produit 05

- **TOTAL4** : la somme du champ **Montant** pour la classe de produit 08
- **TOTAL5** : la somme du champ **Montant** pour la classe de produit 09

```

OPEN Trans_Metaphor_2002
GROUP
TOTAL AMOUNT IF PRODCLS = "03"
TOTAL AMOUNT IF PRODCLS = "05"
TOTAL AMOUNT IF PRODCLS = "08"
TOTAL AMOUNT IF PRODCLS = "09"
END
OPEN Trans_Metaphor_2002

```

## Remarques

### Astuce

Pour voir le tutoriel détaillé sur les commandes LOOP et GROUP, reportez-vous à la rubrique "Regroupement et exécution de boucles" Page 34.

## Commandes utilisables dans la commande GROUP

Le tableau ci-dessous répertorie les commandes Analytics pouvant être utilisées à l'intérieur de la commande GROUP.

Si une commande ne figure pas ci-après, elle ne peut pas être utilisée à l'intérieur de GROUP.

AGE	ASSIGN	BENFORD
CLASSIFY	COMMENT	COUNT
CROSSTAB	DUPLICATES	EXPORT
EXTRACT	GAPS	GROUP
HISTOGRAM	JOIN	LIST
LOOP	MERGE	PROFILE
REPORT	SEQUENCE	STATISTICS
STRATIFY	SUMMARIZE	TOTAL
VERIFY		

## Regroupement et exécution de boucles

La commande GROUP permet d'exécuter plusieurs commandes sur un enregistrement avant de passer à l'enregistrement suivant dans la table, ce qui peut réduire considérablement la durée de traitement.

Vous pouvez utiliser la commande LOOP dans la commande GROUP afin d'exécuter plusieurs fois une série de commandes dans un enregistrement.

## Utilisation de variables avec GROUP

### Variables définies par l'utilisateur

Pour utiliser une variable dans la commande GROUP, définissez la variable avant de saisir le bloc GROUP.

#### Remarque

Pendant que vous pouvez initialiser et définir une variable à l'intérieur d'un bloc GROUP, cela n'est pas recommandé. Les variables initialisées à l'intérieur d'un GROUP peuvent entraîner des résultats inattendus pendant l'utilisation.

À l'intérieur d'un groupe, vous pouvez évaluer les variables à l'aide de la substitution de variable. La valeur de la variable reste identique à la valeur fournie au moment de la saisie de GROUP.

Il est impossible de définir une variable à l'intérieur d'un GROUP pour ensuite y faire référence à l'aide de la substitution de variable :

```
ASSIGN v_test = "hello"
GROUP
  ASSIGN v_test2 = "%v_test% world"
  COMMENT cela serait non valide : v_test3 = "%v_test2% again"
END
```

### Variables définies par le système

Certaines commandes comme TOTAL et STATISTICS génèrent des variables système basées sur des calculs que les commandes effectuent. Si vous utilisez un GROUP pour exécuter ces commandes, les variables système créées sont numérotées dans l'ordre consécutif, à partir du numéro de ligne de la commande à l'intérieur du GROUP (à l'exception des lignes vides) et jusqu'à  $n$ . La valeur  $n$  augmente de 1 pour chaque numéro de ligne dans le GROUP.

#### Remarque

Vous devez attendre la fin de l'exécution de GROUP avant d'utiliser des variables générées par le système qui sont créées à l'intérieur de GROUP. La commande doit s'exécuter pour chaque enregistrement de la table avant que la variable soit disponible. Utilisez ces variables après le mot-clé END de fermeture du GROUP.

Dans l'exemple suivant, la première commande TOTAL génère la variable TOTAL2 et la deuxième génère TOTAL4. Ces deux variables sont disponibles pour être utilisées dans les commandes suivantes une fois le GROUP exécuté :

```
GROUP
TOTAL Réduction IF Ordre_Priorité = "Faible"
ASSIGN v_var = "test"
TOTAL Réduction IF Ordre_Priorité = "Élevé"
END
```

## Remarques sur la syntaxe

- La syntaxe multiligne répertoriée pour la commande GROUP est requise. Par conséquent, il n'est pas possible de saisir la commande GROUP dans la ligne de commande ACL.
- Chaque commande GROUP doit se terminer par une commande END.
- Lorsque vous utilisez la commande GROUP dans vos scripts, vous pouvez améliorer la lisibilité du bloc de commandes en appliquant un retrait aux commandes répertoriées dans le groupe.

# Commande HELP

Lance la documentation d'aide pour Analytics dans un navigateur.

## Syntaxe

```
HELP
```

# Commande HISTOGRAM

Regroupe des enregistrements en fonction des valeurs d'un champ caractère ou numérique, compte le nombre d'enregistrements dans chaque groupe et affiche les groupes et les comptes dans un graphe à barres.

## Syntaxe

```
HISTOGRAM {<ON> champ_caractère|<ON> champ_numérique MINIMUM valeur MAXIMUM
valeur {<INTERVALS nombre>|FREE valeur_intervalle <...n> dernier_intervalle}} <TO
{SCREEN|nom_fichier|GRAPH|PRINT}> <IF test> <WHILE test> <FIRST plage|NEXT plage>
<HEADER texte_en_tête> <FOOTER texte_pied_page> <KEY champ_fractionnement>
<SUPPRESS> <COLUMNS nombre> <APPEND> <LOCAL> <OPEN>
```

## Paramètres

Nom	Description
ON <i>champ_caractère</i>	Expression ou champ de type caractère à utiliser pour l'histogramme.
ON <i>champ_numérique</i>	Expression ou champ numérique à utiliser pour l'histogramme.
MINIMUM <i>valeur</i>	S'applique aux champs numériques uniquement. Valeur minimale du premier intervalle numérique. MINIMUM est facultatif si vous utilisez FREE, sinon il est obligatoire.
MAXIMUM <i>valeur</i>	S'applique aux champs numériques uniquement. Valeur maximale du dernier intervalle numérique. MAXIMUM est facultatif si vous utilisez FREE, sinon il est obligatoire.
INTERVALS <i>nombre</i> optionnel	S'applique aux champs numériques uniquement. Le nombre d'intervalles à taille égale générés par Analytics sur la plage spécifiée par les valeurs MINIMUM et MAXIMUM. Si vous n'indiquez pas de nombre d'intervalles, c'est la valeur par défaut qui est utilisée. La valeur par défaut est spécifiée par le nombre d' <b>Intervalles</b> dans l'onglet <b>Commande</b> de la boîte de dialogue <b>Options</b> .
FREE <i>valeur_intervalle</i> <...n> <i>dernier_intervalle</i> optionnel	S'applique aux champs numériques uniquement. Crée des intervalles de taille personnalisée en spécifiant le point de départ de chaque intervalle et le point de fin du dernier intervalle. Si vous spécifiez les valeurs MINIMUM et MAXIMUM, ces valeurs se trouvent au point



Nom	Description
	<p>de départ du premier intervalle et au point de fin du dernier intervalle ; chaque <i>valeur_intervalle</i> crée un nouvel intervalle à l'intérieur de la plage. Les valeurs des intervalles que vous spécifiez doivent être supérieures à la valeur MINIMUM et inférieures ou égales à la valeur MAXIMUM.</p> <p>Les valeurs des intervalles doivent suivre l'ordre numérique et ne peuvent pas contenir de doublons :</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;">FREE -1000; 0; 1000; 2000; 3000</div> <p>Si vous indiquez à la fois FREE et INTERVALS, INTERVALS est ignoré.</p>
TO SCREEN   <i>nom_fichier</i>   GRAPH   PRINT	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li>◦ <b>SCREEN</b> : affiche les résultats dans la zone d'affichage d'Analytics.</li> <li>◦ <b><i>nom_fichier</i></b> : enregistre les résultats dans un fichier</li> </ul> <p>Indiquez <i>nom_fichier</i> sous forme de chaîne entre guillemets avec l'extension de fichier appropriée. Par exemple : TO "Sortie.TXT"</p> <p>Par défaut, le fichier est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.TXT"</li> <li>• TO "Résultats\Sortie.TXT"</li> </ul> <ul style="list-style-type: none"> <li>◦ <b>GRAPH</b> : affiche les résultats dans un graphique dans la zone d'affichage d'Analytics</li> <li>◦ <b>PRINT</b> : envoie les résultats vers l'imprimante par défaut</li> </ul> <p><b>Remarque</b></p> <p>La sortie des résultats de l'histogramme dans un fichier s'affiche en tant que représentation textuelle d'un graphe à barres.</p>
IF <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.</p> <p><b>Remarque</b></p> <p>La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p>
WHILE <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.</p> <p><b>Remarque</b></p> <p>Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.</p>
FIRST <i>plage</i>   NEXT <i>plage</i> optionnel	<p>Le nombre d'enregistrements à traiter :</p> <ul style="list-style-type: none"> <li>◦ <b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li>◦ <b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné</li> </ul>

Nom	Description
	<p>tionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</p> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter.</p> <p>Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p>
HEADER <i>texte_en-tête</i> optionnel	<p>Texte à insérer en haut de chaque page d'un rapport.</p> <p><i>texte_en-tête</i> doit être indiqué sous forme de chaîne entre guillemets. La valeur remplace la variable système Analytics HEADER.</p>
FOOTER <i>texte_pied_page</i> optionnel	<p>Texte à insérer en bas de chaque page d'un rapport.</p> <p><i>texte_pied_page</i> doit être indiqué sous forme de chaîne entre guillemets. La valeur remplace la variable système Analytics FOOTER.</p>
KEY <i>champ_fractionnement</i> optionnel	<p>Champ ou expression regroupant les calculs de sous-totaux. Un sous-total est calculé chaque fois que la valeur de <i>champ_fractionnement</i> change.</p> <p><i>champ_fractionnement</i> doit être un champ ou une expression de type caractère. Vous ne pouvez indiquer qu'un seul champ, mais vous pouvez utiliser une expression contenant plusieurs champs.</p>
SUPPRESS optionnel	<p>Les valeurs supérieures à la valeur MAXIMUM et inférieures à la valeur MINIMUM sont exclues de la sortie de commande.</p>
COLUMNS <i>nombre</i> optionnel	<p>Longueur de l'axe des abscisses dans la représentation textuelle du graphe à barres si vous sortez les résultats de l'histogramme dans un fichier texte.</p> <p>La valeur nombre correspond au nombre d'espaces caractères (colonnes de texte) à utiliser pour les étiquettes de l'axe des abscisses (et de l'axe des ordonnées). Si vous omettez COLUMNS, la valeur par défaut de 78 espaces caractères est utilisée.</p>
APPEND optionnel	<p>Ajoute la sortie de commande à la fin d'un fichier existant au lieu de remplacer ce fichier.</p> <p><b>Remarque</b></p> <p>Vous devez vous assurer que la structure de la sortie de la commande et du fichier existant est identique :</p> <ul style="list-style-type: none"> <li>• les mêmes champs</li> <li>• le même ordre des champs</li> <li>• les champs correspondants ont la même longueur</li> <li>• les champs correspondants ont le même type de données</li> </ul> <p>Analytics ajoute la sortie à un fichier existant sans tenir compte de sa structure, ce qui peut désordonner les données si la structure de la sortie et du fichier existant ne correspond pas.</p>
LOCAL optionnel	<p>Enregistre le fichier de sortie au même emplacement que le projet Analytics.</p> <p><b>Remarque</b></p> <p>Applicable uniquement lorsque vous exécutez la commande sur une table de serveur avec un fichier de sortie qui est une table Analytics.</p>

Nom	Description
OPEN optionnel	Ouvre la table créée par la commande après l'exécution de la commande. Valide uniquement lorsque la commande crée une table de sortie.

## Exemples

### Histogramme de base pour le salaire horaire

Vous utilisez HISTOGRAM pour créer un graphe indiquant la distribution des salaires de 0 à 100 euros par heure :

```
HISTOGRAM ON Taux MINIMUM 0 MAXIMUM 100 TO GRAPH
```

### Histogramme avec des intervalles définis pour le salaire horaire

Suite à l'exemple précédent, vous utilisez HISTOGRAM pour indiquer les plages de manière plus précise dans le graphique.

La plupart des salaires se situent entre 20 et 50 dollars par heure. Le graphique inclut le nombre d'intervalles suivant :

- trois dans la plage 20 à 50
- un pour la plage 0-20
- un pour la plage 50-100
- un pour la plage > 100

```
HISTOGRAM ON Taux MINIMUM 0 MAXIMUM 100 FREE 20;30;40;50;100 TO GRAPH
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Remplissage des valeurs les plus faibles et les plus élevées

Vous pouvez exécuter les commandes STATISTICS ou PROFILE sur un champ numérique avant l'exécution de la commande HISTOGRAM afin de remplir automatiquement les valeurs des paramètres MINIMUM et MAXIMUM avec la valeur la plus faible et la plus élevée dans le champ.

## Commandes associées

La création d'un histogramme avec un champ caractère revient à un classement. La création d'un histogramme avec un champ numérique revient à une stratification.

À la différence des autres opérations de regroupements dans Analytics, les histogrammes ne prennent pas en charge la sous-totalisation de champs numériques.

# Commande IF

Spécifie une condition qui doit être évaluée sur vrai pour exécuter une commande.

## Syntaxe

```
IF test commande
```

## Paramètres

Nom	Description
<i>rechercher</i>	Condition à satisfaire pour que <i>command</i> soit exécuté.
<i>commande</i>	Toute commande ACLScript valide à exécuter si <i>test</i> est évaluée sur « true ».

## Exemples

### Exécution conditionnelle d'une commande

Vous souhaitez utiliser CLASSIFY sur une table, mais uniquement si la variable *v\_compteur* est supérieure à dix :

```
IF v_compteur > 10 CLASSIFY ON Emplacement TO "Nombre_par_emplacement.fil" OPEN
```

### Exécution d'une commande à partir d'une décision de l'utilisateur

Vous voulez laisser l'utilisateur du script décider de classer ou pas la table.

Dans votre script, vous incluez une boîte de dialogue avec une case à cocher qui, si elle est sélectionnée, permet à la commande CLASSIFY de s'exécuter. La case à cocher stocke une valeur d'introduction Vrai ou Faux dans la variable logique *v\_caseacoher\_classer*.

Vous utilisez un test IF pour déterminer la valeur de *v\_caseacoher\_classer*, et si sa valeur est définie sur Vrai, alors CLASSIFY exécute :

```
IF v_caseacoher_classer=T CLASSIFY ON Emplacement TO "Nombre_par_emplacement.fil" OPEN
```

# Remarques

## Commande IF et paramètre IF

La logique de la commande IF est différente du paramètre IF qui est pris en charge par la plupart des commandes :

- La **commande IF** : détermine si la commande associée est exécutée ou non, selon la valeur de l'expression test
- Le **paramètre IF** : détermine si la commande est exécutée par rapport à chaque enregistrement dans une table Analytics à partir de la valeur de l'expression de test

## La prise de décisions dans les scripts

Dans un script, vous pouvez saisir une série de tests de la commande IF et exécuter plusieurs commandes basées sur les résultats. La commande IF peut aussi être utilisée pour tester la valeur d'une variable afin de déterminer si un traitement supplémentaire doit être exécuté.

# Commande IMPORT ACCESS

Crée une table Analytics en définissant et en important un fichier de base de données Microsoft Access.

## Syntaxe

```
IMPORT ACCESS TO table <PASSWORD num> nom_fichier_importation FROM nom_fichier_source TABLE nom_table_entrée CHARMAX longueur_champ_max MEMOMAX longueur_champ_max
```

## Paramètres

Nom	Description
TO <i>table</i>	<p>Nom de la table Analytics dans laquelle les données doivent être importées.</p> <p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alpha-numériques. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>
PASSWORD <i>num</i> optionnel	<p>Uniquement utilisé avec les fichiers Access protégés par mot de passe.</p> <p>Définition de mot de passe à utiliser.</p> <p>Vous n'utilisez pas PASSWORD <i>num</i> pour inviter à saisir ou pour spécifier un mot de passe réel. La définition de mot de passe renvoie à un mot de passe fourni ou défini auparavant à l'aide de la commande PASSWORD, de la commande SET PASSWORD ou de la balise d'outil d'analyse PASSWORD.</p> <p><i>num</i> correspond au numéro de la définition de mot de passe. Par exemple, si deux mots de passe ont été fournis ou définis auparavant dans un script, ou lors de la planification d'un outil d'analyse, PASSWORD 2 indique que le mot de passe n° 2 est utilisé.</p> <p>Pour plus d'informations sur la fourniture ou définition de mots de passe, consultez les rubriques :</p> <ul style="list-style-type: none"> <li>• "Commande PASSWORD" Page 373</li> <li>• "Commande SET" Page 435</li> <li>• <a href="#">Balise d'outil d'analyse PASSWORD</a></li> </ul> <p>Pour plus d'informations sur la fourniture ou définition de mots de passe, consultez les rubriques :</p> <ul style="list-style-type: none"> <li>• <a href="#">Commande PASSWORD</a></li> <li>• <a href="#">Commande SET</a></li> <li>• <a href="#">Balise d'outil d'analyse PASSWORD</a></li> </ul>
<i>nom_fichier_importation</i>	Nom du fichier de données Analytics à créer.

Nom	Description
	<p>Indiquez <i>nom_fichier_importation</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple, "Factures.FIL".</p> <p>Par défaut, le fichier de données (.FIL) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>o "C:\données\Factures.FIL"</li> <li>o "données\Factures.FIL"</li> </ul>
FROM <i>nom_fichier_source</i>	<p>Nom du fichier de données source. <i>nom_fichier_source</i> doit être une chaîne entre guillemets.</p> <p>Si le fichier de données source n'est pas enregistré dans le même répertoire que le projet Analytics, vous devez utiliser un chemin d'accès absolu ou un chemin d'accès relatif pour indiquer l'emplacement du fichier :</p> <ul style="list-style-type: none"> <li>o "C:\données\<i>nom_fichier_source</i>"</li> <li>o "données<i>nom_fichier_source</i>"</li> </ul>
TABLE <i>nom_table_entrée</i>	Nom de la table dans le fichier de base de données Microsoft Access à importer.
CHARMAX <i>longueur_champ_max</i>	<p>La longueur maximale en caractères d'un champ de la table Analytics qui provient d'une donnée de type caractère dans la source que vous importez.</p> <p>Vous pouvez indiquer de 1 à 255 caractères.</p>
MEMOMAX <i>longueur_champ_max</i>	<p>Longueur maximale en caractères des champs de texte, de commentaire ou mémo que vous importez.</p> <p>Vous pouvez indiquer de 1 à 32 767 caractères (Analytics non Unicode) ou 16 383 caractères (Analytics Unicode).</p>

## Exemples

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

### Importation dans une table

Vous disposez d'un fichier Microsoft Access appelé `Codes_acceptables.mdb`. Vous devez importer la table [`Codes_Acceptable`] à partir du fichier dans Analytics. Pour ce faire, vous utilisez la commande suivante et vous créez une table appelée `codes_acc` dans Analytics.

La longueur des champs caractères ou mémo importés est définie sur la longueur de la valeur la plus longue dans le champ ou bien sur le nombre de caractères maximal spécifié si cette valeur est plus courte :



```
SET ECHO NONE
SET PASSWORD 1 TO "qr347wx"
SET ECHO ON
IMPORT ACCESS TO codes_acc PASSWORD 1 "C:\ACL DATA\Exemples de fichiers de données\codes_acc.fil" FROM "Codes_acceptables.mdb" TABLE "[Codes_acceptables]" CHARMAX 60
MEMOMAX 70
```

# Commande IMPORT DELIMITED

Crée une table Analytics en définissant et en important un fichier texte délimité.

## Syntaxe

```
IMPORT DELIMITED TO table nom_fichier_importation FROM nom_fichier_source <SERVER
nom_profil> encodage_car_source SEPARATOR {car|TAB|SPACE} QUALIFIER {car|NONE}
CONSECUTIVE STARTLINE numéro_ligne <KEEPTITLE> <CRCLEAR> <LFCLEAR>
<REPLACENULL> <ALLCHAR> {ALLFIELDS|[syntaxe_champ] <...n> <IGNORE num_champ>
<...n>}
```

```
syntaxe_champ ::=
FIELD nom type AT position_départ DEC valeur WID octets PIC format AS nom_affichage
```

## Paramètres

Nom	Description
TO <i>table</i>	<p>Nom de la table Analytics dans laquelle les données doivent être importées.</p> <p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alpha-numériques. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>
<i>nom_fichier_importation</i>	<p>Nom du fichier de données Analytics à créer.</p> <p>Indiquez <i>nom_fichier_importation</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple, "Factures.FIL".</p> <p>Par défaut, le fichier de données (.FIL) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>○ "C:\données\Factures.FIL"</li> <li>○ "données\Factures.FIL"</li> </ul>
FROM <i>nom_fichier_source</i>	<p>Nom du fichier de données source. <i>nom_fichier_source</i> doit être une chaîne entre guillemets.</p> <p>Si le fichier de données source n'est pas enregistré dans le même répertoire que le projet Analytics, vous devez utiliser un chemin d'accès absolu ou un chemin d'accès relatif pour indiquer l'emplacement du fichier :</p>

Nom	Description															
	<ul style="list-style-type: none"> <li>o "C:\données\nom_fichier_source"</li> <li>o "donnéesnom_fichier_source"</li> </ul>															
SERVER <i>nom_profil</i> optionnel	Le nom de profil AX Serveur qui contient les données que vous souhaitez importer.															
<i>encodage_car_source</i>	<p>Jeu de caractères et encodage des données source.</p> <p>Selon l'édition d'Analytics que vous utilisez et selon l'encodage des données source, indiquez le code approprié :</p> <table border="1"> <thead> <tr> <th>Code</th> <th>Édition Ana-lytics</th> <th>Encodage des données source</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Édition non Uni-code</td> <td>toutes les données</td> </tr> <tr> <td>0</td> <td>édition Unicode</td> <td>données ASCII</td> </tr> <tr> <td>2</td> <td>édition Unicode</td> <td>données Unicode, codage UTF-16 LE</td> </tr> <tr> <td><b>3</b> <i>code_numérique</i></td> <td>édition Unicode</td> <td> <p>données Unicode qui n'utilisent pas le codage UTF-16 Little Endian</p> <p>Pour déterminer le code numérique correspondant à l'encodage des données source, effectuez une importation à l'aide de l'<b>Assistant de définition des données</b>, sélectionnez l'option <b>Texte codé</b> et recherchez l'encodage correspondant dans la liste déroulante.</p> <p>Pour définir le code, indiquez 3, suivi d'un espace, et du code numérique.</p> </td> </tr> </tbody> </table>	Code	Édition Ana-lytics	Encodage des données source	0	Édition non Uni-code	toutes les données	0	édition Unicode	données ASCII	2	édition Unicode	données Unicode, codage UTF-16 LE	<b>3</b> <i>code_numérique</i>	édition Unicode	<p>données Unicode qui n'utilisent pas le codage UTF-16 Little Endian</p> <p>Pour déterminer le code numérique correspondant à l'encodage des données source, effectuez une importation à l'aide de l'<b>Assistant de définition des données</b>, sélectionnez l'option <b>Texte codé</b> et recherchez l'encodage correspondant dans la liste déroulante.</p> <p>Pour définir le code, indiquez 3, suivi d'un espace, et du code numérique.</p>
Code	Édition Ana-lytics	Encodage des données source														
0	Édition non Uni-code	toutes les données														
0	édition Unicode	données ASCII														
2	édition Unicode	données Unicode, codage UTF-16 LE														
<b>3</b> <i>code_numérique</i>	édition Unicode	<p>données Unicode qui n'utilisent pas le codage UTF-16 Little Endian</p> <p>Pour déterminer le code numérique correspondant à l'encodage des données source, effectuez une importation à l'aide de l'<b>Assistant de définition des données</b>, sélectionnez l'option <b>Texte codé</b> et recherchez l'encodage correspondant dans la liste déroulante.</p> <p>Pour définir le code, indiquez 3, suivi d'un espace, et du code numérique.</p>														
SEPARATOR <i>car</i>   TAB   SPACE	<p>Le caractère de séparation (délimiteur) utilisé entre les champs dans les données source. Vous devez indiquer le caractère sous forme de chaîne entre guillemets.</p> <p>Vous pouvez indiquer une tabulation ou une espace comme séparateur en saisissant le caractère entre guillemets doubles ou en utilisant un mot clé :</p> <ul style="list-style-type: none"> <li>o SEPARATOR " " ou SEPARATOR TAB</li> <li>o SEPARATOR "" ou SEPARATOR SPACE</li> </ul>															
QUALIFIEUR <i>car</i>   NONE	<p>Caractère d'identificateur de texte utilisé dans les données source pour renvoyer à la ligne et identifier des valeurs de champs. Vous devez indiquer le caractère sous forme de chaîne entre guillemets.</p> <p>Pour définir le caractère guillemet double comme identificateur de texte, placez le caractère à l'intérieur de guillemets simples : QUALIFIEUR "".</p> <p>Vous pouvez spécifier qu'il n'y a pas de qualificatif de texte à l'aide de l'une de ces deux méthodes :</p> <ul style="list-style-type: none"> <li>o QUALIFIEUR ""</li> <li>o QUALIFIEUR NONE</li> </ul>															

Nom	Description
CONSECUTIVE optionnel	Les identificateurs de texte consécutifs sont traités en tant qu'identificateur unique.
STARTLINE <i>numéro_ligne</i>	Ligne à laquelle les données commencent. Par exemple, si les quatre premières lignes de données contiennent des informations d'en-tête que vous ne souhaitez pas inclure, indiquez 5 pour <i>numéro_ligne</i> .
KEEPTITLE optionnel	Traitez le numéro de ligne spécifié par STARTLINE comme noms de champs plutôt que comme données. Si vous ignorez KEEPTITLE, les noms des champs génériques sont utilisés. Si vous spécifiez individuellement la syntaxe FIELD, FIELD <i>nom</i> est prioritaire sur les valeurs de la première ligne dans le fichier délimité. Dans cette situation, KEEPTITLE empêche l'importation des valeurs de la première ligne.
CRCLEAR optionnel	Remplace les caractères de retour chariot qui se trouvent entre les identificateurs de texte par des caractères espace. Vous devez indiquer QUALIFIER avec une valeur <i>char</i> pour utiliser CRCLEAR. Si vous utilisez CRCLEAR et LFCLEAR, c'est CRCLEAR qui doit apparaître en premier.
LFCLEAR optionnel	Remplace les caractères de saut de ligne qui se trouvent entre les identificateurs de texte par des caractères espace. Vous devez indiquer QUALIFIER avec une valeur <i>char</i> pour utiliser LFCLEAR. Si vous utilisez CRCLEAR et LFCLEAR, c'est CRCLEAR qui doit apparaître en premier.
REPLACENULL optionnel	Remplace les caractères NUL qui se trouvent dans le fichier délimité par des espaces. Le nombre de caractères NUL remplacés est enregistré dans la trace.
ALLCHAR optionnel	Le type de données caractère est automatiquement attribué à tous les champs importés.  <div style="border-left: 2px solid green; padding-left: 10px; margin-left: 20px;"> <p><b>Astuce</b></p> <p>Affecter le type de données caractère à tous les champs importés simplifie le processus d'importation des fichiers texte délimités. Une fois que les données sont dans Analytics, vous pouvez affecter différents types (Numérique ou DateHeure par exemple) de données aux champs et en spécifier les informations de format.</p> <p>ALLCHAR s'avère utile si vous importez une table dans laquelle le type de données Numérique est affecté automatiquement aux champs des identificateurs par Analytics alors qu'en réalité, ils devraient utiliser le type de données Caractère.</p> </div>
ALLFIELDS	Tous les champs du fichier de données source sont importés. Pour obtenir des informations sur l'affectation des types de données par Analytics lorsque vous utilisez ALLFIELDS, consultez la section "Remarques" Page 255.  <div style="border-left: 2px solid blue; padding-left: 10px; margin-left: 20px;"> <p><b>Remarque</b></p> <p>Si vous spécifiez ALLFIELDS, n'indiquez aucune syntaxe FIELD individuelle ni IGNORE.</p> </div>

Nom	Description				
FIELD <i>nom type</i>	<p>Les champs individuels à importer à partir du fichier de données source, y compris le nom et le type de données du champ. Pour empêcher l'importation d'un champ, ne l'indiquez pas.</p> <p>Pour plus d'informations sur le <i>type</i>, consultez la rubrique "Identificateurs des types de données de champ" Page 256.</p> <p><b>Remarque</b> <i>type</i> est ignoré si vous spécifiez ALLCHAR.</p>				
AT <i>position_départ</i>	<p>La position d'octet de départ du champ dans le fichier de données Analytics.</p> <p><b>Remarque</b></p> <table border="1"> <tbody> <tr> <td>Analytics non Unicode</td> <td>1 octet = 1 caractère</td> </tr> <tr> <td>Analytics Unicode</td> <td>2 octets = 1 caractère</td> </tr> </tbody> </table> <p>Dans Analytics Unicode, normalement vous devez indiquer une position d'octet de départ avec un nombre impair. Indiquer une position de départ avec un chiffre pair peut entraîner l'affichage incorrect des caractères.</p>	Analytics non Unicode	1 octet = 1 caractère	Analytics Unicode	2 octets = 1 caractère
Analytics non Unicode	1 octet = 1 caractère				
Analytics Unicode	2 octets = 1 caractère				
DEC <i>valeur</i>	<p>Nombre de décimales contenues par les champs numériques</p> <p><b>Remarque</b> DEC est ignoré si vous spécifiez ALLCHAR.</p>				
WID <i>octets</i>	<p>La longueur du champ en octets dans le format de la table Analytics.</p> <p><b>Remarque</b></p> <table border="1"> <tbody> <tr> <td>Analytics non Unicode</td> <td>1 octet = 1 caractère</td> </tr> <tr> <td>Analytics Unicode</td> <td>2 octets = 1 caractère</td> </tr> </tbody> </table> <p>Dans Analytics Unicode, spécifiez uniquement un nombre d'octets pair. Indiquer un nombre d'octets impair peut entraîner l'affichage incorrect des caractères.</p>	Analytics non Unicode	1 octet = 1 caractère	Analytics Unicode	2 octets = 1 caractère
Analytics non Unicode	1 octet = 1 caractère				
Analytics Unicode	2 octets = 1 caractère				
PIC <i>format</i>	<p><b>Remarque</b> S'applique aux champs numériques ou DateHeure.</p> <ul style="list-style-type: none"> <li>◦ <b>champs numériques</b> : format d'affichage des valeurs numériques dans les vues et rapports Analytics</li> <li>◦ <b>champs DateHeure</b> : format physique des valeurs DateHeure dans les données source (ordre des caractères date et heure, séparateurs, etc.)</li> </ul> <p><b>Remarque</b> Pour les champs DateHeure, le <i>format</i> doit correspondre exactement au format physique dans les données source. Par exemple, si les données source sont 12/31/2014, vous devez saisir la date au format "MM/JJ/AAAA".</p>				

Nom	Description
	<p>Le <i>format</i> doit être placé entre guillemets.</p> <p><b>Remarque</b> PIC est ignoré si vous spécifiez ALLCHAR.</p>
AS <i>nom_affichage</i>	<p>Le nom d'affichage (titre de la colonne) pour le champ dans la vue dans la nouvelle table Analytics.</p> <p>Indiquez le <i>nom_affichage</i> sous forme de chaîne entre guillemets. Utilisez un point-virgule (;) entre les mots si vous souhaitez insérer un saut de ligne dans le titre de la colonne.</p> <p>AS est requis lors de la définition de FIELD. Pour que le nom d'affichage soit identique au champ nom, saisissez une valeur <i>nom_affichage</i> vide à l'aide de la syntaxe suivante : AS "". Vérifiez qu'il n'y a aucune espace entre les deux guillemets doubles.</p>
IGNORE <i>num_champ</i> <...n> optionnel	<p>Exclut le champ du format de table.</p> <p><i>num_champ</i> indique la position du champ dans les données source. Par exemple, IGNORE 5 exclut le cinquième champ des données de la source du format de la table Analytics.</p> <p><b>Remarque</b> Les données du champ sont toujours importées, mais ne sont pas définies et ne s'affichent pas dans la nouvelle table Analytics. Les données peuvent être définies ultérieurement, si nécessaire, et ajoutées à la table.</p> <p>Pour exclure complètement un champ de l'importation, ne le spécifiez pas lorsque vous indiquez les champs individuellement.</p>

## Exemples

### Importer tous les champs

Vous importez tous les champs à partir d'un fichier délimité composé de valeurs séparées par des virgules dans une table Analytics appelée **Employés**. Le fichier utilise des guillemets doubles comme identificateur de texte. Les types de données sont affectés automatiquement en fonction de l'ensemble des règles décrites dans "Remarques" Page opposée :

```
IMPORT DELIMITED TO Employés "Employés.fil" FROM "Employés.csv" 0 SEPARATOR ","
QUALIFIER "" CONSECUTIVE STARTLINE 1 KEPTITLE ALLFIELDS
```

### Importer tous les champs, affecter automatiquement un type de données caractère

Vous importez tous les champs à partir d'un fichier délimité composé de valeurs séparées par des virgules dans une table Analytics appelée **Employés**. Le fichier utilise des guillemets doubles comme identificateur

de texte. Le type de données caractère est automatiquement attribué à tous les champs importés :

```
IMPORT DELIMITED TO Employés "Employés.fil" FROM "Employés.csv" 0 SEPARATOR ","
QUALIFIER "" CONSECUTIVE STARTLINE 1 KEPTITLE ALLCHAR ALLFIELDS
```

## Importer les champs spécifiés, affecter automatiquement un type de données caractère

Vous importez les champs spécifiés à partir d'un fichier délimité par des tabulations dans une table Analytics appelée **Employés**. Le fichier utilise des guillemets doubles comme identificateur de texte. Le type de données caractère est automatiquement attribué à tous les champs importés :

```
IMPORT DELIMITED TO Employés "Employés.fil" FROM "Employés.csv" 0 SEPARATOR TAB
QUALIFIER "" CONSECUTIVE STARTLINE 1 KEPTITLE ALLCHAR FIELD "Prénom" C AT 1 DEC 0 WID 25 PIC "" AS "Prénom" FIELD "Nom" C AT 26 DEC 0 WID 25 PIC "" AS "Nom" FIELD
"NumCarte" C AT 51 DEC 0 WID 16 PIC "" AS "Num Carte" FIELD "NoEmp" C AT 67 DEC 0 WID 6 PIC "" AS "No Emp" FIELD "DateEmb" C AT 73 DEC 0 WID 10 PIC "" AS "Date Emb" FIELD "Salaire"
C AT 83 DEC 0 WID 5 PIC "" AS "" FIELD "Bonus_2016" C AT 88 DEC 0 WID 10 PIC "" AS "Bonus
2016"
```

## Importer les champs spécifiés, affecter individuellement des types de données

Vous importez les champs spécifiés depuis un fichier délimité par des points-virgules vers une table Analytics nommée **Employés**. Le fichier n'utilise aucun identificateur de texte. Vous spécifiez le type de données de chaque champ importé :

```
IMPORT DELIMITED TO Employés "Employés.fil" FROM "Employés.csv" 0 SEPARATOR ";"
QUALIFIER "" CONSECUTIVE STARTLINE 1 KEPTITLE FIELD "Prénom" C AT 1 DEC 0 WID 25 PIC "" AS "Prénom" FIELD "Nom" C AT 26 DEC 0 WID 25 PIC "" AS "Nom" FIELD "NumCarte" C AT
51 DEC 0 WID 16 PIC "" AS "Num Carte" FIELD "NoEmp" C AT 67 DEC 0 WID 6 PIC "" AS "No Emp"
FIELD "DateEmb" D AT 73 DEC 0 WID 10 PIC "MM/DD/YYYY" AS "Date Emb" FIELD "Salaire" N AT
83 DEC 0 WID 5 PIC "" AS "" FIELD "Bonus_2016" N AT 88 DEC 2 WID 10 PIC "" AS "Bonus 2016"
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Méthode d'affectation des types de données par Analytics lorsque vous utilisez ALLFIELDS

Lorsque vous utilisez le paramètre ALLFIELDS, plutôt que de définir les champs individuellement, Analytics examine un sous-ensemble d'enregistrements au début du fichier délimité et affecte les types de données en fonction de l'ensemble des règles décrites ci-dessous.

Une fois que les données sont dans Analytics, vous pouvez affecter différents types (Numérique ou DateHeure par exemple) de données aux champs et en spécifier les informations de format.

Description des valeurs des champs dans le fichier délimité	Exemples	Type de données affecté
Valeurs entourées de qualificatifs de texte	"Fournisseurs ABC" "6990,75"	Caractère
Les valeurs incluent un caractère non numérique n'importe où dans le champ, à l'exception des virgules et des points utilisés comme séparateurs numériques et du signe négatif (-)	\$995 (995)	Caractère
Les valeurs incluent uniquement les nombres, les séparateurs numériques ou le signe négatif	6990,75 -6990,75 995	Numérique
Une ou plusieurs valeurs vides se trouvent dans un champ		Caractère
Les valeurs DateHeure avec des séparateurs ou les mois en lettres	2016/12/31 31 déc 2016	Caractère
Valeurs DateHeure qui sont toutes des chiffres	20161231	Numérique

## Identificateurs des types de données de champ

Le tableau ci-dessous répertorie les lettres que vous devez utiliser lorsque vous indiquez *type* pour FIELD. Chaque lettre correspond à un type de données.

Par exemple, si vous définissez un champ Nom, qui utilise un type de données caractère, vous devez indiquer « C » : FIELD "Nom" C.

### Remarque

Lorsque vous utilisez l'**Assistant de définition de données** pour définir une table qui comprend des champs EBCDIC, Unicode ou ASCII, ces derniers se voient automatiquement attribuer la lettre « C » (pour le type CHARACTER).

Lorsque vous saisissez manuellement une instruction IMPORT ou que vous modifiez une instruction IMPORT existante, vous pouvez remplacer les lettres « E » ou « U » par des champs EBCDIC ou Unicode.



Lettre	Type de données
A	ACL
B	BINARY
C	CHARACTER
D	DATETIME
E	EBCDIC
F	FLOAT
G	ACCPAC
I	IBMFLOAT
K	UNSIGNED
L	LOGICAL
N	PRINT
P	PACKED
Q	BASIC
R	MICRO
S	CUSTOM
T	PCASCII
U	UNICODE
V	VAXFLOAT
X	NUMERIC
Y	UNISYS
Z	ZONED

# Commande IMPORT EXCEL

Crée une table Analytics en définissant et en important une feuille de calcul Microsoft Excel ou une page nommée.

## Syntaxe

```
IMPORT EXCEL TO table nom_fichier_importation FROM nom_fichier_source TABLE feuille_calcul_ou_plage_nommée_entrée <KEEPTITLE> {ALLFIELDS|CHARMAX longueur_champ_max|  
[syntaxe_champ] <...n> <IGNORE num_champ> <...n>} <OPEN>
```

```
syntaxe_champ ::=  
FIELD nom type {PIC format|WID caractères DEC valeur} AS nom_affichage
```

## Paramètres

Nom	Description
TO <i>table</i>	<p>Nom de la table Analytics dans laquelle les données doivent être importées.</p> <p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alphanumériques. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>
<i>nom_fichier_importation</i>	<p>Nom du fichier de données Analytics à créer.</p> <p>Indiquez <i>nom_fichier_importation</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple, "Factures.FIL".</p> <p>Par défaut, le fichier de données (.FIL) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>o "C:\données\Factures.FIL"</li> <li>o "données\Factures.FIL"</li> </ul>
FROM <i>nom_fichier_source</i>	<p>Nom du fichier de données source. <i>nom_fichier_source</i> doit être une chaîne entre guillemets.</p> <p>Si le fichier de données source n'est pas enregistré dans le même répertoire que le projet Analytics, vous devez utiliser un chemin d'accès absolu ou un chemin d'accès relatif pour indiquer l'emplacement du fichier :</p>

Nom	Description
	<ul style="list-style-type: none"> <li>o "C:\données\nom_fichier_source"</li> <li>o "donnéesnom_fichier_source"</li> </ul>
TABLE <i>feuille_calcul_ou_plage_nommée</i>	<p>La feuille de calcul Microsoft Excel ou la plage nommée dans le fichier de données source à importer :</p> <ul style="list-style-type: none"> <li>o vous devez saisir un signe « \$ » à la fin du nom d'une feuille de calcul</li> <li>o <i>feuille_calcul_ou_plage_nommée</i> doit être indiqué sous forme de chaîne entre guillemets</li> </ul>
KEEPTITLE optionnel	<p>Traitez la première ligne de données en tant que noms de champs plutôt que comme données. Si vous ignorez ce paramètre, les noms des champs génériques sont utilisés.</p> <p>Si vous définissez les champs individuellement, KEEPTITLE doit apparaître avant le premier FIELD.</p>
ALLFIELDS	<p>Tous les champs du fichier de données source sont importés.</p> <p><b>Remarque</b> Si vous spécifiez ALLFIELDS, n'indiquez aucune syntaxe FIELD individuelle ni IGNORE.</p>
CHARMAX <i>longueur_champ_max</i>	<p>S'applique uniquement lorsque vous ne définissez pas les champs individuellement.</p> <p>La longueur maximale en caractères d'un champ de la table Analytics qui provient d'une donnée de type caractère dans le fichier de données source.</p>
FIELD <i>nom type</i>	<p>Les champs individuels à importer à partir du fichier de données source, y compris le nom et le type de données du champ. Pour empêcher l'importation d'un champ, ne l'indiquez pas.</p> <p>Pour plus d'informations sur le <i>type</i>, consultez la rubrique "Identificateurs des types de données de champ" Page 262.</p>
PIC <i>format</i>	<p><b>Remarque</b> S'applique aux champs numériques ou DateHeure.</p> <ul style="list-style-type: none"> <li>o <b>champs numériques</b> : format d'affichage des valeurs numériques dans les vues et rapports Analytics</li> <li>o <b>champs DateHeure</b> : format physique des valeurs DateHeure dans les données source (ordre des caractères date et heure, séparateurs, etc.)</li> </ul> <p><b>Remarque</b> Pour les champs DateHeure, le <i>format</i> doit correspondre exactement au format physique dans les données source. Par exemple, si les données source sont 12/31/2014, vous devez saisir la date au format "MM/JJ/AAAA".</p> <p>Le <i>format</i> doit être placé entre guillemets.</p>
WID <i>caractères</i>	<p>La longueur du champ en caractères dans le format de la table Analytics.</p>
DEC <i>valeur</i>	<p>Nombre de décimales contenues par les champs numériques</p>

Nom	Description
AS <i>nom_affichage</i>	<p>Le nom d'affichage (titre de la colonne) pour le champ dans la vue dans la nouvelle table Analytics.</p> <p>Indiquez le <i>nom_affichage</i> sous forme de chaîne entre guillemets. Utilisez un point-virgule (;) entre les mots si vous souhaitez insérer un saut de ligne dans le titre de la colonne.</p> <p>AS est requis lors de la définition de FIELD. Pour que le nom d'affichage soit identique au champ nom, saisissez une valeur <i>nom_affichage</i> vide à l'aide de la syntaxe suivante : AS "". Vérifiez qu'il n'y a aucune espace entre les deux guillemets doubles.</p>
IGNORE <i>num_champ</i> <...n> optionnel	<p>Exclut le champ du format de table.</p> <p><i>num_champ</i> indique la position du champ dans les données source. Par exemple, IGNORE 5 exclut le cinquième champ des données de la source du format de la table Analytics.</p>
OPEN optionnel	<p>Ouvre la table créée par la commande après l'exécution de la commande. Valide uniquement lorsque la commande crée une table de sortie.</p>

## Exemples

### Importation des champs spécifiés

Vous effectuez une importation définissant une nouvelle table Analytics appelée **Cartes\_crédit**. Elle utilise la première ligne de données Excel comme noms de champs.

La table Analytics définit et inclut trois champs de la table source, mais elle exclut les champs restants :

```
IMPORT EXCEL TO Cartes_Crédit "Cartes Crédit.fil" FROM "Cartes_Crédit_Metaphor.xls" TABLE
"Cartes_Crédit_Entreprise$" KEEPTITLE FIELD "NUMCARTE" N WID 16 DEC 0 AS "Numéro de
carte" FIELD "DATEEXP" D WID 10 PIC "AAAA-MM-JJ" AS "Date Expiration" FIELD
"MONTECHU" N WID 6 DEC 2 AS "Montant Échu" IGNORE 2 IGNORE 3 IGNORE 5 IGNORE 6
IGNORE 7 IGNORE 9 IGNORE 10 IGNORE 11 IGNORE 12
```

### Importer tous les champs

Vous effectuez une importation définissant une nouvelle table Analytics appelée **Transactions\_mai**. Elle utilise la première ligne de données Excel comme noms de champs.

La table Analytics inclut tous les champs de la table source et elle utilise les définitions de champs par défaut.

## Longueur de champ définie sur la valeur la plus longue

Dans le premier exemple, les champs à l'origine des données de type caractère dans le fichier de données source sont définis sur la longueur de la valeur la plus longue dans le champ :

```
IMPORT EXCEL TO Transactions_Mai "Transactions_Mai.fil" FROM "Trans_Mai.xls" TABLE "Trans1_Mai$" KEeptITLE ALLFIELDS
```

## Longueur de champ limitée

Dans le deuxième exemple, les champs à l'origine des données de type caractère dans le fichier de données source sont définis sur la longueur de la valeur la plus longue dans le champ ou sur la valeur CHARMAX de 100 caractères si elle est plus courte :

```
IMPORT EXCEL TO Transactions_Mai "Transactions_Mai.fil" FROM "Trans_Mai.xls" TABLE "Trans1_Mai$" KEeptITLE CHARMAX 100
```

# Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Définissez les champs un à un ou bien importez tous les champs à l'aide d'une définition par défaut

Lorsque vous importez un fichier Excel dans une table Analytics, vous pouvez utiliser les paramètres FIELD pour définir chaque champ individuellement ou vous pouvez utiliser le paramètre ALLFIELDS ou le paramètre CHARMAX pour importer tous les champs avec les définitions de champ Analytics par défaut.

## Taille maximale de l'importation de données

### Format de fichier .xlsx ou .xlsm

Le nombre maximal de colonnes Excel, et le nombre maximal de caractères dans un champ, que vous pouvez importer à partir de fichiers .xlsx ou .xlsm n'est pas limité à un nombre spécifique.

L'importation de ces types de fichiers Excel est régie par la limite de longueur d'enregistrement dans les fichiers de données Analytics (.fil) de 32 Ko. Si un enregistrement dans le fichier Excel source crée un enregistrement Analytics de longueur supérieure à 32 Ko, l'importation échoue.

## Format de fichier .xls

L'importation de fichiers .xls (Excel 97-2003) utilise un autre type de traitement ; elle est soumise à un maximum de :

- 255 colonnes
- 255 caractères par champ
- 32 Ko par enregistrement
- 65 000 lignes

## Identificateurs des types de données de champ

Le tableau ci-dessous répertorie les lettres que vous devez utiliser lorsque vous indiquez *type* pour FIELD. Chaque lettre correspond à un type de données.

Par exemple, si vous définissez un champ Nom, qui utilise un type de données caractère, vous devez indiquer « C » : FIELD "Nom" C.

### Remarque

Lorsque vous utilisez l'**Assistant de définition de données** pour définir une table qui comprend des champs EBCDIC, Unicode ou ASCII, ces derniers se voient automatiquement attribuer la lettre « C » (pour le type CHARACTER).

Lorsque vous saisissez manuellement une instruction IMPORT ou que vous modifiez une instruction IMPORT existante, vous pouvez remplacer les lettres « E » ou « U » par des champs EBCDIC ou Unicode.

Lettre	Type de données
A	ACL
B	BINARY
C	CHARACTER
D	DATETIME
E	EBCDIC
F	FLOAT
G	ACCPAC
I	IBMFLOAT
K	UNSIGNED
L	LOGICAL

Lettre	Type de données
N	PRINT
P	PACKED
Q	BASIC
R	MICRO
S	CUSTOM
T	PCASCII
U	UNICODE
V	VAXFLOAT
X	NUMERIC
Y	UNISYS
Z	ZONED

# Commande IMPORT GRCPROJECT

Crée une table Analytics en important une table HighBond Projets.

## Syntaxe

```
IMPORT GRCPROJECT TO table nom_fichier_importation PASSWORD num FROM id_org/id_type
<FIELD nom AS nom_affichage <...n>>
```

## Paramètres

Nom	Description
TO <i>table</i>	<p>Nom de la table Analytics dans laquelle les données doivent être importées.</p> <p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alpha-numériques. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>
<i>nom_fichier_importation</i>	<p>Nom du fichier de données Analytics à créer.</p> <p>Indiquez <i>nom_fichier_importation</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple, "Factures.FIL".</p> <p>Par défaut, le fichier de données (.FIL) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>○ "C:\données\Factures.FIL"</li> <li>○ "données\Factures.FIL"</li> </ul>
PASSWORD <i>num</i>	<p>Définition de mot de passe à utiliser.</p> <p>Vous n'utilisez pas PASSWORD <i>num</i> pour inviter à saisir ou pour spécifier un mot de passe réel. La définition de mot de passe renvoie à un mot de passe fourni ou défini auparavant à l'aide de la commande PASSWORD, de la commande SET PASSWORD ou de la balise d'outil d'analyse PASSWORD.</p> <p><i>num</i> correspond au numéro de la définition de mot de passe. Par exemple, si deux mots de passe ont été fournis ou définis auparavant dans un script, ou lors de la planification d'un outil d'analyse, PASSWORD 2 indique que le mot de passe n° 2 est utilisé.</p> <p>Pour plus d'informations sur la fourniture ou définition de mots de passe, consultez les rubriques :</p>



Nom	Description							
	<ul style="list-style-type: none"> <li>• "Commande PASSWORD" Page 373</li> <li>• "Commande SET" Page 435</li> <li>• <a href="#">Balise d'outil d'analyse PASSWORD</a></li> </ul> <p>Pour plus d'informations sur la fourniture ou définition de mots de passe, consultez les rubriques :</p> <ul style="list-style-type: none"> <li>• <a href="#">Commande PASSWORD</a></li> <li>• <a href="#">Commande SET</a></li> <li>• <a href="#">Balise d'outil d'analyse PASSWORD</a></li> </ul> <p><b>Remarque</b></p> <p>Selon l'environnement d'exécution du script, PASSWORD peut être requis ou non :</p> <table border="1" data-bbox="570 653 1271 1140"> <tbody> <tr> <td data-bbox="570 653 920 842">Analytics (activation en ligne)</td> <td data-bbox="920 653 1271 842">PASSWORD n'est pas obligatoire.  Le jeton d'accès HighBond de l'utilisateur actuel est utilisé automatiquement.</td> </tr> <tr> <td data-bbox="570 842 920 947">Analytics (activation hors ligne)</td> <td data-bbox="920 842 1271 947" rowspan="4">PASSWORD est obligatoire.</td> </tr> <tr> <td data-bbox="570 947 920 1010">Robots</td> </tr> <tr> <td data-bbox="570 1010 920 1073">Analytics Exchange</td> </tr> <tr> <td data-bbox="570 1073 920 1140">Fenêtre App Analyse</td> </tr> </tbody> </table>	Analytics (activation en ligne)	PASSWORD n'est pas obligatoire.  Le jeton d'accès HighBond de l'utilisateur actuel est utilisé automatiquement.	Analytics (activation hors ligne)	PASSWORD est obligatoire.	Robots	Analytics Exchange	Fenêtre App Analyse
Analytics (activation en ligne)	PASSWORD n'est pas obligatoire.  Le jeton d'accès HighBond de l'utilisateur actuel est utilisé automatiquement.							
Analytics (activation hors ligne)	PASSWORD est obligatoire.							
Robots								
Analytics Exchange								
Fenêtre App Analyse								
FROM <i>id_org</i> / <i>id_type</i>	<p>Organisation et type d'information définissant les données en cours d'importation :</p> <ul style="list-style-type: none"> <li>◦ <b><i>id_org</i></b> : l'organisation Projets à partir de laquelle vous importez des données</li> <li>◦ <b><i>id_type</i></b> : le type d'information que vous importez</li> </ul> <p>Les valeurs <i>id_org</i> et <i>id_type</i> doivent être séparées par une barre oblique, sans espace intermédiaire : FROM "125@eu/audits".</p> <p>La chaîne de caractères complète doit être entourée de guillemets.</p> <p><b>ID d'organisation</b></p> <p><i>id_org</i> doit inclure le numéro ID de l'organisation, et si vous effectuez l'importation à partir d'un data center qui n'est pas en Amérique du Nord, cette valeur doit inclure le code du data center. Le numéro ID de l'organisation et le code du data center doivent être séparés par une arobase (@) : FROM "125@eu".</p> <p>Le code du data center indique à partir de quel serveur régional d'HighBond vous importez les données :</p> <ul style="list-style-type: none"> <li>◦ ap : Asie-Pacifique</li> <li>◦ au : Australie</li> <li>◦ ca : Canada</li> <li>◦ eu : Europe</li> <li>◦ us : Amérique du Nord</li> </ul>							

Nom	Description
	<p>Vous pouvez uniquement utiliser le ou les codes de data center autorisés pour l'installation d'HighBond de votre organisation. Le data center nord américain est le data center par défaut, vous n'êtes donc pas obligé d'indiquer "@us".</p> <p>Si vous ne connaissez pas le numéro d'ID de l'organisation, utilisez l'interface utilisateur d'Analytics pour importer une table depuis Projets. Le numéro d'ID de l'organisation se trouve dans la commande dans la trace. Pour plus d'informations, consultez la rubrique <a href="#">Définition des données ACL GRC</a>.</p> <p><b>ID type</b></p> <p><i>id_type</i> le type d'information que vous importez. Les informations dans Projets sont contenues dans une série de tables associées.</p> <p>Pour <i>id_type</i>, utilisez l'une des valeurs ci-après. Saisissez la valeur telle qu'elle apparaît et incluez des tirets bas, le cas échéant :</p> <ul style="list-style-type: none"> <li>○ audits : Projets</li> <li>○ control_test_plans : Plans de test de contrôle</li> <li>○ control_tests : Test de contrôle</li> <li>○ controls : Contrôles</li> <li>○ finding_actions : Actions</li> <li>○ findings : Problèmes</li> <li>○ mitigations : Associations du contrôle de risque</li> <li>○ narratives : Descriptions</li> <li>○ objectives : Objectifs</li> <li>○ risks : Risques</li> <li>○ walkthroughs : Revues générales</li> </ul> <p><b>Astuce</b></p> <p>Pour plus d'informations sur la façon dont les tables de Projets sont associées et sur les champs clés que vous pouvez utiliser pour joindre les tables une fois qu'elles ont été importées dans Analytics, consultez la rubrique <a href="#">Définir les données ACL GRC</a>.</p>
<p>FIELD <i>nom</i> AS <i>nom_affichage</i> &lt;...n&gt; optionnel</p>	<p>Champs individuels dans les données source à importer. Indiquez le nom.</p> <p>Si vous ignorez FIELD, tous les champs sont importés.</p> <ul style="list-style-type: none"> <li>○ <i>nom</i> doit correspondre exactement au nom du champ physique dans la table Projets, avec la casse correspondante</li> <li>○ <i>nom_affichage</i> (autre titre de colonne) est le nom d'affichage du champ dans la vue dans la nouvelle table Analytics. Vous devez indiquer un nom d'affichage pour chaque FIELD <i>nom</i>. Indiquez le <i>nom_affichage</i> sous forme de chaîne entre guillemets.</li> </ul> <p>Utilisez un point-virgule (;) entre les mots si vous souhaitez insérer un saut de ligne dans le titre de la colonne.</p> <p>À la différence de certaines autres commandes IMPORT dans Analytics, vous ne pouvez pas spécifier de <i>nom_affichage</i> vide comme moyen d'utiliser le nom FIELD comme nom d'affichage.</p>

Nom	Description
	<p><b>Astuce</b></p> <p>Pour obtenir les noms de champs physiques, utilisez l'interface utilisateur Analytics pour importer la table appropriée à partir de Projets. Les noms de champs physiques se trouvent dans la commande dans la trace.</p> <p>Les importations suivantes peuvent être rédigées à l'aide d'un script.</p>

## Exemples

### Importer tous les champs à partir de la table Projets

Vous importez tous les champs de la table **Projets** pour tous les projets actifs relevant de l'organisation 286 vers une table Analytics nommée **Tous\_projets**. Vous incluez une définition de mot de passe numérotée pour authentifier la connexion :

```
IMPORT GRCPROJECT TO Tous_projets "C:\ACL GRC Project Data\Tous_projets.fil" PASSWORD
1 FROM "286@us/audits"
```

### Importer certains champs à partir de la table Projets

Vous importez certains champs de la table **Projets** pour tous les projets actifs relevant de l'organisation 286 vers une table Analytics nommée **Tous\_projets** :

```
IMPORT GRCPROJECT TO Tous_projets "C:\ACL GRC Project Data\Tous_projets.fil" FROM
"286@us/audits" FIELD "id" AS "Id" FIELD "description" AS "Description" FIELD "nom" AS "Nom"
FIELD "date_début" AS "Date de début" FIELD "état" AS "État" FIELD "créé_le" AS "Créé à"
```

### Importer tous les champs à partir de la table Problèmes

Vous importez tous les champs de la table **Problèmes** pour tous les projets actifs relevant de l'organisation 286 vers une table Analytics nommée **Tous\_problèmes** :

```
IMPORT GRCPROJECT TO Tous_problèmes "C:\ACL GRC Project Data\Tous_problèmes.fil" FROM
"286@us/découvertes"
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

# Création d'une définition de mot de passe et spécification d'une valeur de mot de passe

## Commande PASSWORD

Si vous utilisez la commande PASSWORD pour créer la définition de mot de passe numéroté pour la connexion à HighBond, aucune valeur de mot de passe n'est indiquée, et l'utilisateur est invité à saisir le mot de passe lorsque le script tente de se connecter.

Pour plus d'informations, consultez la section "Commande PASSWORD" Page 373.

## Commande SET PASSWORD

Si vous utilisez la commande SET PASSWORD pour créer la définition de mot de passe numéroté pour la connexion à HighBond, une valeur de mot de passe est indiquée, et l'utilisateur n'est pas invité à saisir le mot de passe, ce qui convient aux scripts conçus pour s'exécuter sans surveillance.

Pour plus d'informations, consultez la rubrique [Commande SET PASSWORD](#).

## Jeton d'accès à HighBond

Quelle que soit la méthode que vous utilisez pour créer la définition de mot de passe, la valeur de mot de passe requise est un jeton d'accès à HighBond :

- **Méthode PASSWORD** - Les utilisateurs peuvent acquérir un jeton d'accès en sélectionnant **Outils Jeton d'accès à HighBond**, puis en se connectant à HighBond. Un jeton d'accès est renvoyé, et les utilisateurs peuvent le copier et le coller dans la demande de mot de passe.
- **Méthode SET PASSWORD** : pour insérer un jeton d'accès dans la syntaxe de la commande SET PASSWORD dans un script Analytics, cliquez avec le bouton droit de la souris dans l'**Éditeur de script**, sélectionnez **Insérer > Jeton HighBond**, puis connectez-vous à HighBond. Un jeton d'accès est inséré dans le script à l'emplacement du curseur.

### Attention

Le jeton d'accès renvoyé correspond au compte utilisé pour se connecter à HighBond. En tant que rédacteur de scripts, il se peut qu'il ne soit pas très approprié d'utiliser votre propre jeton d'accès si vous écrivez un script qui sera utilisé par d'autres personnes.

# Fonctionnalité de débogage pour les importations

Une fonctionnalité de débogage simple existe pour les importations à partir d'HighBond.

Les données importées sont stockées temporairement dans un fichier intermédiaire JSON dans le dossier contenant le projet Analytics cible. Dans un dossier contenant un projet Analytics, vous pouvez créer un fichier texte entraînant la conservation du fichier JSON, plutôt que sa suppression après l'importation des données dans Analytics.

- **Le fichier JSON est présent** : Si l'importation à partir d'HighBond échoue, mais que le fichier JSON est présent sur votre ordinateur, vous savez que le problème est du côté d'Analytics et non du côté d'HighBond.
- **Le fichier JSON n'est pas présent** : Si l'importation à partir d'HighBond échoue, et que le fichier JSON n'est pas présent sur votre ordinateur, vous savez que le problème est du côté d'HighBond.

Ces informations peuvent être utiles pour résoudre des problèmes.

## Configurer la conservation du fichier intermédiaire JSON

Dans le dossier contenant le projet Analytics cible, créez un fichier texte vide comportant ce nom exactement : `_grc_import_debug.txt`

Lorsque vous importez à partir de Résultats ou de Projets vers HighBond, le fichier intermédiaire JSON est conservé avec le nom `results.json`. Le fichier est écrasé à chaque nouvelle importation à partir d'HighBond.

# Commande IMPORT GRCRESULTS

Crée une table Analytics en important une interprétation ou une table HighBond Résultats.

## Syntaxe

```
IMPORT GRCRESULTS TO table nom_fichier_importation <PASSWORD num> FROM Chemin_ressources_Résultats <FIELD nom AS nom_affichage <...n>>
```

## Paramètres

Nom	Description
TO <i>table</i>	<p>Nom de la table Analytics dans laquelle les données doivent être importées.</p> <p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alpha-numériques. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>
<i>nom_fichier_importation</i>	<p>Nom du fichier de données Analytics à créer.</p> <p>Indiquez <i>nom_fichier_importation</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple, "Factures.FIL".</p> <p>Par défaut, le fichier de données (.FIL) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>o "C:\données\Factures.FIL"</li> <li>o "données\Factures.FIL"</li> </ul>
PASSWORD <i>num</i>	<p>Définition de mot de passe à utiliser.</p> <p>Vous n'utilisez pas PASSWORD <i>num</i> pour inviter à saisir ou pour spécifier un mot de passe réel. La définition de mot de passe renvoie à un mot de passe fourni ou défini auparavant à l'aide de la commande PASSWORD, de la commande SET PASSWORD ou de la balise d'outil d'analyse PASSWORD.</p> <p><i>num</i> correspond au numéro de la définition de mot de passe. Par exemple, si deux mots de passe ont été fournis ou définis auparavant dans un script, ou lors de la planification d'un outil d'analyse, PASSWORD 2 indique que le mot de passe n° 2 est utilisé.</p> <p>Pour plus d'informations sur la fourniture ou définition de mots de passe, consultez les rubriques :</p>

Nom	Description							
	<ul style="list-style-type: none"> <li>• "Commande PASSWORD" Page 373</li> <li>• "Commande SET" Page 435</li> <li>• <a href="#">Balise d'outil d'analyse PASSWORD</a></li> </ul> <p>Pour plus d'informations sur la fourniture ou définition de mots de passe, consultez les rubriques :</p> <ul style="list-style-type: none"> <li>• <a href="#">Commande PASSWORD</a></li> <li>• <a href="#">Commande SET</a></li> <li>• <a href="#">Balise d'outil d'analyse PASSWORD</a></li> </ul> <p><b>Remarque</b></p> <p>Selon l'environnement d'exécution du script, PASSWORD peut être requis ou non :</p> <table border="1" data-bbox="570 653 1271 1140"> <tbody> <tr> <td data-bbox="570 653 922 842">Analytics (activation en ligne)</td> <td data-bbox="922 653 1271 842">PASSWORD n'est pas obligatoire.  Le jeton d'accès HighBond de l'utilisateur actuel est utilisé automatiquement.</td> </tr> <tr> <td data-bbox="570 842 922 947">Analytics (activation hors ligne)</td> <td data-bbox="922 842 1271 947" rowspan="4">PASSWORD est obligatoire.</td> </tr> <tr> <td data-bbox="570 947 922 1010">Robots</td> </tr> <tr> <td data-bbox="570 1010 922 1073">Analytics Exchange</td> </tr> <tr> <td data-bbox="570 1073 922 1140">Fenêtre App Analyse</td> </tr> </tbody> </table>	Analytics (activation en ligne)	PASSWORD n'est pas obligatoire.  Le jeton d'accès HighBond de l'utilisateur actuel est utilisé automatiquement.	Analytics (activation hors ligne)	PASSWORD est obligatoire.	Robots	Analytics Exchange	Fenêtre App Analyse
Analytics (activation en ligne)	PASSWORD n'est pas obligatoire.  Le jeton d'accès HighBond de l'utilisateur actuel est utilisé automatiquement.							
Analytics (activation hors ligne)	PASSWORD est obligatoire.							
Robots								
Analytics Exchange								
Fenêtre App Analyse								
FROM <i>Chemin_res-sources_résultats</i>	<p>Chemin d'accès permettant d'accéder aux données que vous importez.</p> <p>La forme du chemin d'accès varie selon les données que vous importez. Pour obtenir des informations détaillées sur la forme du chemin d'accès, reportez-vous à la rubrique "Chemin d'accès Résultats" Page 273.</p> <p><b>Remarque</b></p> <p>La forme du chemin d'accès Résultats est fournie par une API et est susceptible d'être modifiée. Le moyen le plus simple et le plus fiable d'acquérir la syntaxe actuelle qui convient pour le chemin consiste à effectuer une importation manuelle des données cibles et de copier le chemin depuis la trace des commandes.</p>							
FIELD <i>nom</i> AS <i>nom_affichage &lt;...n&gt;</i> optionnel	<p>Champs individuels dans les données source à importer. Indiquez le nom.</p> <p>Si vous ignorez FIELD, tous les champs sont importés.</p> <p><b>Nom</b></p> <p><i>nom</i> doit correspondre exactement au nom du champ physique dans la table Résultats, avec la casse correspondante. Pour afficher le nom de champ physique, effectuez l'une des actions suivantes :</p> <ul style="list-style-type: none"> <li>◦ Dans Résultats, cliquez sur un en-tête de colonne dans la <b>Vue de la table</b>. Le nom</li> </ul>							

Nom	Description
	<p>du champ physique apparaît après <b>Nom du champ</b>.</p> <ul style="list-style-type: none"> <li>Dans l'interface utilisateur d'Analytics, lorsque vous importez une table Résultats, le nom du champ physique apparaît entre parenthèses après le nom d'affichage dans la boîte de dialogue vous permettant de sélectionner des champs.</li> </ul> <p><b>Remarque</b></p> <p>Le nom du champ physique Résultats n'est pas le nom d'affichage utilisé dans les en-têtes des colonnes dans la <b>Vue de la table</b>.</p> <p>Voir aussi "Considérations générales sur les noms des champs lors de l'importation et de l'exportation des données de Résultats" Page 275.</p> <p><b>Nom d'affichage</b></p> <p><i>nom_affichage</i> (autre titre de colonne) est le nom d'affichage du champ dans la vue dans la nouvelle table Analytics. Vous devez indiquer un nom d'affichage pour chaque FIELD <i>nom</i>. Indiquez le <i>nom_affichage</i> sous forme de chaîne entre guillemets.</p> <p>Utilisez un point-virgule (;) entre les mots si vous souhaitez insérer un saut de ligne dans le titre de la colonne.</p> <p>À la différence de certaines autres commandes IMPORT dans Analytics, vous ne pouvez pas spécifier de <i>nom_affichage</i> vide comme moyen d'utiliser le nom FIELD comme nom d'affichage.</p>

## Exemples

### Importation des champs spécifiés d'une table de Résultats

Vous importez des champs spécifiés d'une table de Résultats vers une table Analytics nommée **Exceptions T et E** :

```
IMPORT GRCRESULTS TO Exceptions_T_et_E "C:\Analyse secondaire\Exceptions_T_et_E.fil"
PASSWORD 1 FROM "résultats/api/orgs/11594/tests_contrôle/185699/exceptions" FIELD "meta-
data.status" AS "Status" FIELD "EmpNo" AS "Numéro de l'employé" FIELD "DATE" AS "Date"
FIELD "CARDNUM" AS "Numéro de carte" FIELD "CODES" AS "Codes MC" FIELD "AMOUNT" AS
"Montant" FIELD "DESCRIPTION" AS "Description"
```

### Importation de tous les champs d'une table de Résultats

Vous importez tous les champs d'une table de Résultats vers une table Analytics nommée **Exceptions T et E** :

```
IMPORT GRCRESULTS TO Exceptions_T_et_E "C:\Analyse secondaire\Exceptions_T_et_E.fil"
PASSWORD 1 FROM "résultats/api/orgs/11594/tests_contrôle/185699/exceptions"
```

### Importation des données à partir d'une interprétation dans Résultats



Vous importez une interprétation de Résultats vers une table Analytics nommée **Exceptions T et E filtrées** :

```
IMPORT GRCRESULTS TO Exceptions_T_et_E_filtrées "C:\Analyse secondaire\Exceptions_T_et_E_filtrées.fil" FROM "résultats/api/orgs/11594/tests_contrôle/185699/interprétations/22699/exceptions"
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Conserver l'ordre des tris et les filtres

Lorsque vous importez des données à partir de Résultats, la personnalisation de données, comme les tris ou les filtres, est conservée ou ignorée dans la table Analytics ainsi créée selon le mode d'importation des données :

- **importer une table** : la personnalisation des données est ignorée. Toutes les données de la table sont importées, à l'exception des champs que vous choisissez d'omettre.
- **importer une interprétation** : la personnalisation des données est conservée

## Chemin d'accès Résultats

### Remarque

La forme du chemin d'accès Résultats est fournie par une API et est susceptible d'être modifiée. Le moyen le plus simple et le plus fiable d'acquérir la syntaxe actuelle qui convient pour le chemin consiste à effectuer une importation manuelle des données cibles et de copier le chemin depuis la trace des commandes.

Le chemin d'accès Résultats dans le paramètre FROM prend la forme générale suivante :

```
FROM "résultats <-code région>/api/orgs/<ID org>/tests_contrôle/<ID de test de contrôle>/exceptions"
```

Par exemple : FROM "résultats/api/orgs/11594/tests\_contrôle/4356/exceptions"

L'ID org s'affiche dans la barre d'adresse du navigateur lorsque vous vous connectez à la Barre de lancement. L'ID du test de contrôle et l'ID de l'interprétation s'affichent dans la barre d'adresse lorsque vous affichez ces tables dans Résultats.

Le tableau ci-dessous indique toutes les variations du chemin d'accès Résultats.

Pour l'importation :	Utilisez cette forme de chemin d'accès Résultats :
Données des tests de contrôle (table)	FROM "résultats/api/orgs/11594/tests_contrôle/4356/exceptions"

Pour l'importation :	Utilisez cette forme de chemin d'accès Résultats :
Piste d'audit des tests de contrôle (table)	FROM "résultats/api/orgs/11594/tests_contrôle/4356/piste_audit"
Commentaires des tests de contrôle (table)	FROM "résultats/api/orgs/11594/tests_contrôle/4356/commentaires"
Interprétation	FROM "résultats/api/orgs/11594/tests_contrôle/4356/interprétations/1192/exceptions"
Données d'une région HighBond autre que la région par défaut (us)	<ul style="list-style-type: none"> <li>○ <b>Asie-Pacifique</b> : FROM "résultats-ap/api/orgs/11594/tests_contrôle/4356/exceptions"</li> <li>○ <b>Australie</b> : FROM "résultats-au/api/orgs/11594/tests_contrôle/4356/exceptions"</li> <li>○ <b>Canada</b> : FROM "résultats-ca/api/orgs/11594/tests_contrôle/4356/exceptions"</li> <li>○ <b>Europe</b> : FROM "résultats-eu/api/orgs/11594/tests_contrôle/4356/exceptions"</li> </ul>

## Colonnes d'informations générées par le système

Lorsque vous importez des données à partir de Résultats, vous avez la possibilité d'importer une ou plusieurs colonnes d'informations générées par le système parmi celles ci-dessous.

Les colonnes générées par le système :

- font partie des tables Résultats et contiennent des informations de traitement liées aux enregistrements individuels OU
- représentent des informations supplémentaires : nom de la collection, nom de la table ou numéro d'ID d'enregistrement

Vous devez spécifier les noms des champs des colonnes générées par le système exactement comme ils apparaissent ci-dessous. Les noms d'affichage par défaut s'appliquent lorsque vous effectuez une importation depuis Résultats via l'interface utilisateur Analytics. Vous êtes libre de modifier les noms d'affichage si vous écrivez un script pour le processus d'importation.

Nom de champ	Nom d'affichage par défaut
metadata.priority	Priorité
metadata.status	Statut
metadata.publish_date	Publié(e)
metadata.publisher	Nom de l'éditeur
metadata.assignee	Assigné
metadata.group	Groupe
metadata.updated_at	Mis(e) à jour
metadata.closed_at	Fermé(e)

Nom de champ	Nom d'affichage par défaut
extras.collection	Collection
extras.results_table	Table de résultats
extras.record_id	ID enregistrement

## Considérations générales sur les noms des champs lors de l'importation et de l'exportation des données de Résultats

Si vous exécutez des allers-retours de données entre Résultats et Analytics, vous devez vous assurer que tous les noms de champ dans la table de Résultats répondent aux conditions préalables des noms de champ Analytics. Si cela n'est pas le cas, vous risquez de ne pas aligner correctement les données entre Analytics et Résultats.

Par exemple, les caractères spéciaux des noms de champs de Résultats sont automatiquement convertis en caractères de soulignement lorsqu'ils sont importés dans Analytics, ce qui signifie que les noms de champ ne correspondent plus aux noms d'origine dans Résultats. Si vous réexportez ensuite les données Analytics dans la table d'origine dans Résultats, les champs ne correspondent plus comme il faut.

Pour éviter ce problème avec les données avec lesquelles vous prévoyez de faire des allers-retours, assurez-vous qu'elles répondent aux conditions préalables des noms de champ Analytics avant de charger les données dans Résultats à partir de fichiers CSV ou Excel :

- ne contiennent aucun caractère spécial ni espace
- ne commencent pas par un chiffre
- contiennent uniquement des caractères alphanumériques ou le caractère de soulignement ( \_ )

## Création d'une définition de mot de passe et spécification d'une valeur de mot de passe

### Commande PASSWORD

Si vous utilisez la commande PASSWORD pour créer la définition de mot de passe numéroté pour la connexion à HighBond, aucune valeur de mot de passe n'est indiquée, et l'utilisateur est invité à saisir le mot de passe lorsque le script tente de se connecter.

Pour plus d'informations, consultez la section "Commande PASSWORD" Page 373.

### Commande SET PASSWORD

Si vous utilisez la commande SET PASSWORD pour créer la définition de mot de passe numéroté pour la connexion à HighBond, une valeur de mot de passe est indiquée, et l'utilisateur n'est pas invité à saisir le mot de passe, ce qui convient aux scripts conçus pour s'exécuter sans surveillance.

Pour plus d'informations, consultez la rubrique [Commande SET PASSWORD](#).

Jeton d'accès à HighBond

Quelle que soit la méthode que vous utilisez pour créer la définition de mot de passe, la valeur de mot de passe requise est un jeton d'accès à HighBond :

- **Méthode PASSWORD** - Les utilisateurs peuvent acquérir un jeton d'accès en sélectionnant **Outils Jeton d'accès à HighBond**, puis en se connectant à HighBond. Un jeton d'accès est renvoyé, et les utilisateurs peuvent le copier et le coller dans la demande de mot de passe.
- **Méthode SET PASSWORD** : pour insérer un jeton d'accès dans la syntaxe de la commande SET PASSWORD dans un script Analytics, cliquez avec le bouton droit de la souris dans l'**Éditeur de script**, sélectionnez **Insérer > Jeton HighBond**, puis connectez-vous à HighBond. Un jeton d'accès est inséré dans le script à l'emplacement du curseur.

### Attention

Le jeton d'accès renvoyé correspond au compte utilisé pour se connecter à HighBond. En tant que rédacteur de scripts, il se peut qu'il ne soit pas très approprié d'utiliser votre propre jeton d'accès si vous écrivez un script qui sera utilisé par d'autres personnes.

## Fonctionnalité de débogage pour les importations

Une fonctionnalité de débogage simple existe pour les importations à partir d'HighBond.

Les données importées sont stockées temporairement dans un fichier intermédiaire JSON dans le dossier contenant le projet Analytics cible. Dans un dossier contenant un projet Analytics, vous pouvez créer un fichier texte entraînant la conservation du fichier JSON, plutôt que sa suppression après l'importation des données dans Analytics.

- **Le fichier JSON est présent** : Si l'importation à partir d'HighBond échoue, mais que le fichier JSON est présent sur votre ordinateur, vous savez que le problème est du côté d'Analytics et non du côté d'HighBond.
- **Le fichier JSON n'est pas présent** : Si l'importation à partir d'HighBond échoue, et que le fichier JSON n'est pas présent sur votre ordinateur, vous savez que le problème est du côté d'HighBond.

Ces informations peuvent être utiles pour résoudre des problèmes.

### Configurer la conservation du fichier intermédiaire JSON

Dans le dossier contenant le projet Analytics cible, créez un fichier texte vide comportant ce nom exactement : `_grc_import_debug.txt`

Lorsque vous importez à partir de Résultats ou de Projets vers HighBond, le fichier intermédiaire JSON est conservé avec le nom `results.json`. Le fichier est écrasé à chaque nouvelle importation à partir d'HighBond.

## Importation de tables volumineuses

Il se peut que les tables comportant un grand nombre de champs n'importent pas correctement à l'aide d'une seule commande IMPORT GRCRESULTS. Si vous devez travailler avec une seule table contenant un grand nombre de champs en dehors de Résultats, utilisez l'une des méthodes suivantes :

- **Diviser la table** : utilisez au moins deux commandes IMPORT GRCRESULTS pour importer un sous-ensemble de champs, puis joignez les tables ainsi créées dans Analytics à l'aide de la

commande JOIN

- **Exporter la table dans un fichier** : utilisez l'exportation au format CSV, puis importez le fichier ainsi créé dans Analytics à l'aide de la commande IMPORT DELIMITED

# Commande IMPORT LAYOUT

Importe un fichier de format de table externe (.layout) vers un projet Analytics.

## Remarque :

Avant la version 11 d'Analytics, les fichiers de format de table externes utilisaient une extension de fichier .fmt. Vous pouvez toujours importer un fichier de format de table avec une extension .fmt en indiquant l'extension manuellement.

## Syntaxe

```
IMPORT LAYOUT fichier_format_externe TO nom_format_table
```

## Paramètres

Nom	Description
<i>fichier_format_externe</i>	<p>Nom du fichier de format de table externe. Si le nom du fichier ou le chemin inclut des espaces, le nom ou le chemin doit être indiqué entre guillemets, par exemple, "Trans Cf.layout".</p> <p>L'extension de fichier .layout est utilisée par défaut et elle ne doit pas être nécessairement spécifiée. Si besoin, vous pouvez utiliser une autre extension de fichier, par exemple .fmt.</p> <p>Si le fichier de format ne se trouve pas dans le même dossier que le projet Analytics, vous devez utiliser un chemin absolu ou relatif pour spécifier l'emplacement du fichier, par exemple, "C:\Formats enregistrés\Trans_Cf.layout" ou "Formats enregistrés\Trans_Cf.layout".</p>
TO <i>nom_format_table</i>	<p>Le nom du format de table importé dans le projet Analytics, par exemple, "Trans Cf Mai". Vous devez indiquer un <i>nom_format_table</i> sous forme de chaîne entre guillemets s'il contient des espaces. Vous pouvez spécifier un <i>nom_format_table</i> différent du nom du <i>fichier_format_externe</i>.</p>

## Exemple

### Exportation d'un fichier de format de table externe

Vous importez un fichier de format de table externe nommé **Trans\_Cf.layout** et créez un nouveau format de table nommé **Trans\_Cf\_Mai** dans le projet Analytics :

```
IMPORT LAYOUT "C:\Formats enregistrés\Trans_Cf.layout" TO "Trans_Cf_Mai"
```

## Remarques

### Quand utiliser IMPORT LAYOUT

L'importation d'un fichier de format de table externe et son association à un fichier de données peuvent vous éviter d'avoir à créer un format de table depuis le début :

- Si le format de table importé spécifie une association avec un fichier de données Analytics (.fil) particulier et qu'un fichier de données au même nom existe dans le dossier contenant le projet, le format de table importé est automatiquement associé au fichier de données dans le dossier.
- Si le dossier de projet ne contient aucun fichier de données, vous devez lier le format de table importé à une nouvelle source de données.

### Les formats de table et les fichiers de données source doivent correspondre

Le format de table importé et le fichier de données avec lequel il est associé doivent correspondre. La structure des données dans le fichier de données doit correspondre aux définitions de champs spécifiées par les métadonnées du format de table.

La structure de données fait référence aux éléments de données (champs) contenus dans un fichier de données, au nombre et à l'ordre des champs, au type de données et à la longueur des champs. Si le format de table et le fichier de données ne correspondent pas, il en résulte des données désordonnées ou manquantes.

# Commande IMPORT MULTIDELIMITED

Crée plusieurs tables Analytics en définissant et en important plusieurs fichiers délimités.

## Syntaxe

```
IMPORT MULTIDELIMITED <TO dossier_import> FROM {nom_fichier_source|dossier_source}
encodage_car_source SEPARATOR {car|TAB|SPACE} QUALIFIER {car|NONE} CONSECUTIVE
STARTLINE numéro_ligne <KEEPTITLE> <CRCLEAR> <LFCLEAR> <REPLACENULL>
<ALLCHAR>
```

### Remarque

Vous devez indiquer les paramètres IMPORT MULTIDELIMITED exactement dans le même ordre que ci-dessus et dans le tableau ci-dessous.

Pour importer proprement plusieurs fichiers délimités, la structure de tous les fichiers doit être cohérente avant l'importation.

Pour plus d'informations, consultez la section "Structure de fichiers cohérente requise" Page 285.

## Paramètres

Nom	Description
TO <i>dossier_import</i> optionnel	<p>Dossier dans lequel importer les données.</p> <p>Pour indiquer le dossier, utilisez un chemin d'accès au fichier absolu ou un chemin d'accès au fichier relatif au dossier contenant le projet Analytics. Indiquez <i>dossier_import</i> sous forme de chaîne entre guillemets.</p> <p><b>Exemple</b></p> <pre>TO "C:\Point of Sale Audit\Données\Données de travail transaction"</pre> <pre>TO "Données\Données de travail transaction"</pre> <p>Si vous omettez TO, les données sont importées dans le dossier contenant le projet Analytics.</p>
FROM <i>nom_fichier_</i>	Nom des fichiers de données source ou du dossier contenant les fichiers de données



Nom	Description
<p><i>source</i>   <i>dossier_source</i></p>	<p>source.</p> <p>Indiquez <i>nom_fichier_source</i> ou <i>dossier_source</i> sous forme de chaîne entre guillemets.</p> <p>La commande prend en charge l'importation de quatre types de fichiers délimités :</p> <ul style="list-style-type: none"> <li>○ *.csv</li> <li>○ *.dat</li> <li>○ *.del</li> <li>○ *.txt</li> </ul> <p>Fichiers de données source dans le dossier de projet Analytics racine</p> <p>Pour indiquer plusieurs fichiers, utilisez un caractère générique (*) au lieu de caractères uniques dans les noms de fichiers. Le caractère générique représente zéro (0) occurrence ou plus de n'importe quel lettre, chiffre ou caractère spécial.</p> <p><b>Exemple</b></p> <pre style="border: 1px solid #ccc; padding: 5px;">FROM "Transactions_FY*.csv"</pre> <p>sélectionne :</p> <p><i>Transactions_FY18.csv</i></p> <p><i>Transactions_FY17.csv</i></p> <p>Vous pouvez utiliser un caractère générique dans plusieurs emplacements d'un nom de fichier et dans une extension de fichier.</p> <p><b>Exemple</b></p> <pre style="border: 1px solid #ccc; padding: 5px;">FROM "Transactions_FY*.*"</pre> <p>sélectionne :</p> <p><i>Transactions_FY18.txt</i></p> <p><i>Transactions_FY17.csv</i></p> <p>Fichiers de données source absents du dossier de projet Analytics racine</p> <p>Si les fichiers de données source ne sont pas enregistrés dans le même dossier que le projet Analytics, vous devez utiliser un chemin d'accès absolu ou un chemin d'accès relatif au dossier contenant le projet pour indiquer l'emplacement des fichiers.</p> <p><b>Exemple</b></p> <pre style="border: 1px solid #ccc; padding: 5px;">FROM "C:\Point of Sale Audit\Données\Fichiers maîtres Transaction\Transactions_FY*.csv"</pre> <pre style="border: 1px solid #ccc; padding: 5px;">FROM "Données\Fichiers maîtres Transaction\Transactions_FY*.csv"</pre> <p>Dossier contenant les fichiers de données source</p> <p>Au lieu d'indiquer des noms de fichier, vous pouvez simplement indiquer le nom du dossier contenant plusieurs fichiers de données source. Tous les fichiers délimités pris en charge dans le dossier sont importés (*.csv, *.dat, *.del, *.txt).</p>

Nom	Description															
	<p>Pour indiquer un dossier de données source, utilisez un chemin d'accès au fichier absolu ou un chemin d'accès au fichier relatif au dossier contenant le projet Analytics.</p> <p><b>Exemple</b></p> <pre>FROM "C:\Point of Sale Audit\Données\Fichiers maîtres Transaction"</pre> <pre>FROM "Données\Fichiers maîtres Transaction"</pre>															
<p><i>encodage_car_source</i></p>	<p>Jeu de caractères et encodage des données source.</p> <p>Selon l'édition d'Analytics que vous utilisez et selon l'encodage des données source, indiquez le code approprié :</p> <table border="1" data-bbox="467 716 1430 1354"> <thead> <tr> <th data-bbox="467 716 662 814">Code</th> <th data-bbox="662 716 857 814">Édition Analytics</th> <th data-bbox="857 716 1430 814">Encodage des données source</th> </tr> </thead> <tbody> <tr> <td data-bbox="467 814 662 905">0</td> <td data-bbox="662 814 857 905">Édition non Unicode</td> <td data-bbox="857 814 1430 905">toutes les données</td> </tr> <tr> <td data-bbox="467 905 662 972">0</td> <td data-bbox="662 905 857 972">édition Unicode</td> <td data-bbox="857 905 1430 972">données ASCII</td> </tr> <tr> <td data-bbox="467 972 662 1039">2</td> <td data-bbox="662 972 857 1039">édition Unicode</td> <td data-bbox="857 972 1430 1039">données Unicode, codage UTF-16 LE</td> </tr> <tr> <td data-bbox="467 1039 662 1354">3 <i>code numérique</i></td> <td data-bbox="662 1039 857 1354">édition Unicode</td> <td data-bbox="857 1039 1430 1354"> <p>données Unicode qui n'utilisent pas le codage UTF-16 Little Endian</p> <p>Pour déterminer le code numérique correspondant à l'encodage des données source, effectuez une importation à l'aide de l'<b>Assistant de définition des données</b>, sélectionnez l'option <b>Texte codé</b> et recherchez l'encodage correspondant dans la liste déroulante.</p> <p>Pour définir le code, indiquez 3, suivi d'un espace, et du code numérique.</p> </td> </tr> </tbody> </table> <p><b>Remarque</b></p> <p>Si vous n'indiquez pas de code, Analytics non Unicode utilise automatiquement 0 et Analytics Unicode utilise automatiquement 2 .</p>	Code	Édition Analytics	Encodage des données source	0	Édition non Unicode	toutes les données	0	édition Unicode	données ASCII	2	édition Unicode	données Unicode, codage UTF-16 LE	3 <i>code numérique</i>	édition Unicode	<p>données Unicode qui n'utilisent pas le codage UTF-16 Little Endian</p> <p>Pour déterminer le code numérique correspondant à l'encodage des données source, effectuez une importation à l'aide de l'<b>Assistant de définition des données</b>, sélectionnez l'option <b>Texte codé</b> et recherchez l'encodage correspondant dans la liste déroulante.</p> <p>Pour définir le code, indiquez 3, suivi d'un espace, et du code numérique.</p>
Code	Édition Analytics	Encodage des données source														
0	Édition non Unicode	toutes les données														
0	édition Unicode	données ASCII														
2	édition Unicode	données Unicode, codage UTF-16 LE														
3 <i>code numérique</i>	édition Unicode	<p>données Unicode qui n'utilisent pas le codage UTF-16 Little Endian</p> <p>Pour déterminer le code numérique correspondant à l'encodage des données source, effectuez une importation à l'aide de l'<b>Assistant de définition des données</b>, sélectionnez l'option <b>Texte codé</b> et recherchez l'encodage correspondant dans la liste déroulante.</p> <p>Pour définir le code, indiquez 3, suivi d'un espace, et du code numérique.</p>														
<p>SEPARATOR <i>car</i>   TAB   SPACE</p>	<p>Le caractère de séparation (délimiteur) utilisé entre les champs dans les données source. Vous devez indiquer le caractère sous forme de chaîne entre guillemets.</p> <p>Vous pouvez indiquer une tabulation ou une espace comme séparateur en saisissant le caractère entre guillemets doubles ou en utilisant un mot clé :</p> <ul style="list-style-type: none"> <li>○ SEPARATOR " " ou SEPARATOR TAB</li> <li>○ SEPARATOR " " ou SEPARATOR SPACE</li> </ul>															
<p>QUALIFIER <i>car</i>   NONE</p>	<p>Caractère d'identificateur de texte utilisé dans les données source pour renvoyer à la ligne et identifier des valeurs de champs. Vous devez indiquer le caractère sous forme de chaîne entre guillemets.</p>															

Nom	Description
	<p>Pour définir le caractère guillemet double comme identificateur de texte, placez le caractère à l'intérieur de guillemets simples : QUALIFIER "".</p> <p>Vous pouvez spécifier qu'il n'y a pas de qualificatif de texte à l'aide de l'une de ces deux méthodes :</p> <ul style="list-style-type: none"> <li>○ QUALIFIER ""</li> <li>○ QUALIFIER NONE</li> </ul>
CONSECUTIVE optionnel	Les identificateurs de texte consécutifs sont traités en tant qu'identificateur unique.
STARTLINE <i>numéro_</i> <i>ligne</i>	<p>Ligne à laquelle les données commencent.</p> <p>Par exemple, si les quatre premières lignes de données contiennent des informations d'en-tête que vous ne souhaitez pas inclure, indiquez 5 pour <i>numéro_ligne</i>.</p> <p><b>Remarque</b></p> <p>Idéalement, la ligne de départ des données devrait être la même dans tous les fichiers délimités que vous importez avec une seule exécution d'IMPORT MULTIDELIMITED.</p> <p>Si les lignes de départ sont différentes, consultez la rubrique "Structure de fichiers cohérente requise" Page 285.</p>
KEEPTITLE optionnel	<p>Traitez le numéro de ligne spécifié par STARTLINE comme noms de champs plutôt que comme données. Si vous ignorez KEEPTITLE, les noms des champs génériques sont utilisés.</p> <p><b>Remarque</b></p> <p>Les noms des champs doivent se trouver sur le même numéro de ligne dans tous les fichiers délimités que vous importez avec une seule exécution d'IMPORT MULTIDELIMITED.</p> <p>Si les noms de champ se trouvent sur différents numéros de ligne, consultez la rubrique "Structure de fichiers cohérente requise" Page 285.</p>
CRCLEAR optionnel	<p>Remplace les caractères de retour chariot qui se trouvent entre les identificateurs de texte par des caractères espace. Vous devez indiquer QUALIFIER avec une valeur <i>char</i> pour utiliser CRCLEAR.</p> <p>Si vous utilisez CRCLEAR et LFCLEAR, c'est CRCLEAR qui doit apparaître en premier.</p>
LFCLEAR optionnel	<p>Remplace les caractères de saut de ligne qui se trouvent entre les identificateurs de texte par des caractères espace. Vous devez indiquer QUALIFIER avec une valeur <i>char</i> pour utiliser LFCLEAR.</p> <p>Si vous utilisez CRCLEAR et LFCLEAR, c'est CRCLEAR qui doit apparaître en premier.</p>
REPLACENULL optionnel	Remplace les caractères NUL qui se trouvent dans le fichier délimité par des espaces. Le nombre de caractères NUL remplacés est enregistré dans la trace.
ALLCHAR optionnel	Le type de données caractère est automatiquement attribué à tous les champs importés.

Nom	Description
	<p><b>Astuce</b></p> <p>Affecter le type de données caractère à tous les champs importés simplifie le processus d'importation des fichiers texte délimités. Une fois que les données sont dans Analytics, vous pouvez affecter différents types (Numérique ou DateHeure par exemple) de données aux champs et en spécifier les informations de format.</p> <p>ALLCHAR s'avère utile si vous importez une table dans laquelle le type de données Numérique est affecté automatiquement aux champs des identificateurs par Analytics alors qu'en réalité, ils devraient utiliser le type de données Caractère.</p>

## Exemples

Dans les exemples qui suivent, on suppose que les données des transactions mensuelles sont stockées dans 12 fichiers délimités :

- `Transactions_Jan.csv` vers `Transactions_Dec.csv`

### Remarque

Une table Analytics distincte est créée pour fichier délimité que vous importez.

### Importer tous les fichiers délimités

Vous voulez importer les 12 fichiers délimités. Vous utilisez le symbole caractère générique (\*) à l'emplacement du mois dans le nom de chaque fichier

Analytics essaye d'attribuer le bon type de données à chaque champ.

```
IMPORT MULTIDELIMITED FROM "Transactions_*.csv" 0 SEPARATOR "," QUALIFIER ""
CONSECUTIVE STARTLINE 1 KEPTITLE
```

### Importer tous les fichiers délimités en tant que données de type caractère.

Cet exemple est identique au précédent, tout en montrant également comment Analytics attribue automatiquement le type de champ Caractère à tous les champs importés.

```
IMPORT MULTIDELIMITED FROM "Transactions_*.csv" 0 SEPARATOR "," QUALIFIER ""
CONSECUTIVE STARTLINE 1 KEPTITLE ALLCHAR
```

### Importer tous les fichiers délimités dans le dossier défini

Vous souhaitez importer tous les fichiers délimités dans le dossier `C:\Point of sale audit\Data\Fichiers maîtres Transaction`.

```
IMPORT MULTIDELIMITED FROM "C:\Point of sale audit\Data\Transaction master files" 0
SEPARATOR "," QUALIFIER "" CONSECUTIVE STARTLINE 1 KEPTITLE
```

## Importez tous les fichiers délimités dans le dossier spécifié et enregistrez les tables Analytics dans un autre dossier

Cet exemple est le même que le précédent, mais au lieu d'enregistrer les tables Analytics dans le dossier de projet racine, vous voulez les enregistrer dans le dossier `C:\Point of Sale Audit\Données\Données de travail transaction`.

```
IMPORT MULTIDELIMITED FROM "C:\Point of sale audit\Data\Transaction working data" FROM
"C:\Point of sale audit\Data\Transaction master files" 0 SEPARATOR "," QUALIFIER ""
CONSECUTIVE STARTLINE 1 KEPTITLE
```

## Remarques

### Structure de fichiers cohérente requise

Pour importer proprement un groupe de fichiers délimités à l'aide d'IMPORT MULTIDELIMITED, la structure de tous les fichiers du groupe doit être cohérente.

Vous pouvez importer des fichiers délimités dont la structure est incohérente, puis effectuer le nettoyage des données et leur standardisation dans Analytics. Toutefois, cette approche peut être génératrice de pas mal de travail. Dans la plupart des cas, il est plus simple d'assurer la cohérence des fichiers délimités avant l'importation.

Pour importer proprement plusieurs fichiers délimités, les éléments suivants doivent être cohérents sur l'ensemble des fichiers :

Élément	Mot-clé ACLS-cript	Problème	Solution
Jeu de caractères et encodage des données source	<i>code numérique</i>	(édition Unicode d'Analytics uniquement) Les fichiers délimités source utilisent différents encodages de caractère. Par exemple, certains fichiers ont un encodage ASCII alors que d'autres un encodage Unicode.	Regroupez les fichiers source par type d'encodage et lancez un import distinct pour chaque groupe.
Caractère délimiteur	SEPARATOR	Les fichiers délimités source utilisent un caractère de séparation différent (séparateur) entre les champs.	Effectuez l'une des actions suivantes : <ul style="list-style-type: none"> <li>Standardisez le séparateur de champs dans les fichiers source avant de les importer.</li> <li>Regroupez les fichiers source par séparateur de champs et lancez un</li> </ul>

Élément	Mot-clé ACLS-cript	Problème	Solution
			import distinct pour chaque groupe.
Caractère des identificateurs de texte	QUALIFIER	Les fichiers délimités source utilisent un caractère d'identification de texte pour renvoyer à la ligne et identifier des valeurs de champs.	Effectuez l'une des actions suivantes : <ul style="list-style-type: none"> <li>Standardisez l'identificateur de texte dans les fichiers source avant de les importer.</li> <li>Regroupez les fichiers source par identificateur de texte et lancez un import distinct pour chaque groupe.</li> </ul>
Ligne de départ des données	STARTLINE	Les fichiers délimités source ont différentes lignes de départ des données.	Effectuez l'une des actions suivantes : <ul style="list-style-type: none"> <li>Standardisez la ligne de départ dans les fichiers source avant de les importer.</li> <li>Regroupez les fichiers source ayant des lignes de départ identiques et lancez un import distinct pour chaque groupe.</li> <li>Faites en sorte que <i>numéro_ligne</i> soit égal à la plus petite ligne de départ parmi tous les fichiers. Une fois que les fichiers ont été importés dans les tables Analytics, vous pouvez utiliser "Commande EXTRACT" Page 206 pour extraire uniquement les enregistrements de toutes les tables contenant des informations d'en-tête indésirables.</li> </ul>
Nom des champs	KEEPTITLE	Les fichiers délimités source ont des noms de champs sur différents numéros de ligne.	Effectuez l'une des actions suivantes : <ul style="list-style-type: none"> <li>Standardisez les numéros de ligne avec les noms de champs dans les fichiers source avant de les importer.</li> <li>Regroupez les fichiers source ayant des noms de champs sur les mêmes numéros de ligne et lancez un import distinct pour chaque groupe.</li> </ul>
Nom des champs	KEEPTITLE	Des fichiers délimités source ont des noms de champ et d'autres pas.	Effectuez l'une des actions suivantes : <ul style="list-style-type: none"> <li>Ajoutez les noms de champ aux fichiers source qui le nécessitent avec d'importer tous les fichiers.</li> <li>Regroupez les fichiers source ayant des noms de champs, et ceux qui n'en ont pas, et lancez un import distinct pour chaque groupe.</li> <li>Otez l'option KEEPTITLE afin d'importer tous les fichiers en utilisant des noms de champ générique. Une</li> </ul>

Élément	Mot-clé ACLS-cript	Problème	Solution
			<p>fois que les fichiers ont été importés dans les tables Analytics, vous pouvez utiliser "Commande EXTRACT" Page 206 pour extraire uniquement les données que vous souhaitez depuis n'importe quelle table.</p>

## Commandes IMPORT DELIMITED multiples

La commande IMPORT MULTIDELIMITED exécute en fait plusieurs commandes IMPORT MULTIDELIMITED individuelles, une pour chaque fichier importé. Si vous double-cliquez sur l'entrée IMPORT MULTIDELIMITED dans le fichier journal, chaque commande IMPORT MULTIDELIMITED s'affiche individuellement dans la zone d'affichage.

## Combiner plusieurs fichiers délimités après les avoir importés

Après avoir importé plusieurs fichiers délimités dans des tables Analytics individuelles, vous voudrez peut-être les combiner en une seule table Analytics. Par exemple, vous pouvez combiner les données de douze tables mensuelles en une seule table annuelle contenant toutes les données.

Pour obtenir des informations sur la combinaison de plusieurs tables Analytics, consultez la rubrique "Commande APPEND" Page 73.

# Commande IMPORT MULTIEXCEL

Crée plusieurs tables Analytics en définissant et en important plusieurs pages nommées ou feuilles de calcul Microsoft Excel.

## Syntaxe

```
IMPORT MULTIEXCEL <TO dossier_import> FROM {nom_fichier_source|dossier_source} TABLE
saisie_feuilles_calcul_ou_plages_nommées <PREFIX> <KEEPTITLE> <CHARMAX longueur_
champ_max>
```

### Remarque

Vous devez indiquer les paramètres IMPORT MULTIEXCEL exactement dans le même ordre que ci-dessus et dans le tableau ci-dessous.

## Paramètres

Nom	Description
TO <i>dossier_import</i> optionnel	<p>Dossier dans lequel importer les données.</p> <p>Pour indiquer le dossier, utilisez un chemin d'accès au fichier absolu ou un chemin d'accès au fichier relatif au dossier contenant le projet Analytics. Indiquez <i>dossier_import</i> sous forme de chaîne entre guillemets.</p> <p><b>Exemple</b></p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">TO "C:\Point of Sale Audit\Données\Données de travail transaction"</div> <div style="border: 1px solid #ccc; padding: 5px;">TO "Données\Données de travail transaction"</div> <p>Si vous omettez TO, les données sont importées dans le dossier contenant le projet Analytics.</p>
FROM <i>nom_fichier_source</i>   <i>dossier_source</i>	<p>Nom du ou des fichiers de données source, ou du dossier contenant le(s) fichier(s) de données source.</p> <p>Indiquez <i>nom_fichier_source</i> ou <i>dossier_source</i> sous forme de chaîne entre guillemets.</p> <p><b>Fichier(s) de données source dans le dossier de projet Analytics racine</b></p> <ul style="list-style-type: none"> <li>◦ <b>un seul fichier Excel</b> Indiquez le nom et l'extension de fichier complets.</li> </ul> <p><b>Exemple</b></p>



Nom	Description
	<div data-bbox="459 268 1425 342" style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;">FROM "Transactions_FY18.xlsx"</div> <ul style="list-style-type: none"> <li data-bbox="430 363 711 390">○ <b>plusieurs fichiers Excel</b></li> </ul> <p data-bbox="459 405 1406 491">Pour indiquer plusieurs fichiers, utilisez un caractère générique (*) au lieu de caractères uniques dans les noms de fichiers. Le caractère générique représente zéro (0) occurrence ou plus de n'importe quel lettre, chiffre ou caractère spécial.</p> <p data-bbox="459 512 581 543"><b>Exemple</b></p> <div data-bbox="459 564 1425 638" style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;">FROM "Transactions_FY*.xlsx"</div> <p data-bbox="459 659 597 686">sélectionne :</p> <p data-bbox="459 701 766 728">Transactions_FY18.xlsx</p> <p data-bbox="459 743 766 770">Transactions_FY17.xlsx</p> <p data-bbox="459 785 1406 842">Vous pouvez utiliser un caractère générique dans plusieurs emplacements d'un nom de fichier et dans une extension de fichier.</p> <p data-bbox="459 863 581 894"><b>Exemple</b></p> <div data-bbox="459 915 1425 989" style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;">FROM "Transactions_FY*.*"</div> <p data-bbox="459 1010 597 1037">sélectionne :</p> <p data-bbox="459 1052 766 1079">Transactions_FY18.xlsx</p> <p data-bbox="459 1094 750 1121">Transactions_FY17.xls</p> <p data-bbox="423 1146 1396 1213"><b>Fichier(s) de données source absent(s) du dossier de projet Analytics racine</b></p> <p data-bbox="423 1241 1396 1327">Si le(s) fichier(s) de données source ne sont pas enregistrés dans le même dossier que le projet Analytics, vous devez utiliser un chemin d'accès absolu ou un chemin d'accès relatif au fichier contenant le projet pour indiquer l'emplacement du fichier :</p> <p data-bbox="423 1348 548 1379"><b>Exemple</b></p> <div data-bbox="423 1400 1425 1499" style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;">FROM "C:\Point of Sale Audit\Données\Fichiers maîtres Transaction\Transactions_FY18.xlsx"</div> <div data-bbox="423 1520 1425 1593" style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;">FROM "Données\Fichiers maîtres Transaction\Transactions_FY*.xlsx"</div> <p data-bbox="423 1619 1156 1650"><b>Dossier contenant le(s) fichier(s) de données source</b></p> <p data-bbox="423 1677 1382 1734">Au lieu d'indiquer un nom de fichier, vous pouvez simplement indiquer le nom du dossier contenant un ou plusieurs fichiers de données source.</p> <p data-bbox="423 1749 1419 1806">Pour indiquer un dossier de données source, utilisez un chemin d'accès au fichier absolu ou un chemin d'accès au fichier relatif au dossier contenant le projet Analytics.</p>

Nom	Description
	<p><b>Exemple</b></p> <pre>FROM "C:\Point of Sale Audit\Données\Fichiers maîtres Transaction"</pre> <pre>FROM "Données\Fichiers maîtres Transaction"</pre> <p><b>Remarque</b></p> <p>Lorsque vous indiquez un dossier, une feuille de calcul de fichier Excel dans le dossier est importée si le nom de la feuille de calcul correspond à la valeur TABLE.</p>
<p>TABLE <i>saisie_feuilles_calcul_ou_plages_nommées</i></p>	<p>Nom des feuilles de calcul ou plages nommées à importer. Une table Analytics distincte est créée pour chaque feuille de calcul ou plage nommée importée.</p> <p>Indiquez <i>saisie_feuilles_calcul_ou_plages_nommées</i> sous forme de chaîne entre guillemets.</p> <p>Dans les noms des feuilles de calcul ou des plages, utilisez un caractère générique (*) au lieu de caractères uniques.</p> <p>Par exemple, "Trans_*\$" sélectionne les feuilles de calcul suivantes :</p> <ul style="list-style-type: none"> <li>○ Trans_Jan</li> <li>○ Trans_Fév</li> <li>○ Trans_Mar</li> <li>○ et ainsi de suite</li> </ul> <p><b>Remarque</b></p> <p>Le caractère générique (*) représente zéro (0) occurrence ou plus de n'importe quel lettre, chiffre ou caractère spécial.</p> <p>Vous pouvez utiliser un caractère générique dans plusieurs emplacements.</p> <p>Par exemple, *Trans*\$ sélectionne :</p> <ul style="list-style-type: none"> <li>• Trans_Jan</li> <li>• Jan_Trans</li> </ul> <p><b>Signification du symbole dollar (\$)</b></p> <p>Dans un fichier Excel, les feuilles de calcul sont identifiées par un symbole dollar (\$) ajouté au nom de la feuille de calcul (Trans_Jan\$). Le symbole dollar n'est pas visible dans Excel.</p> <p>Les plages nommées sont identifiées par l'absence d'un symbole dollar (Trans_Jan_commercial).</p> <p>Il n'est pas obligatoire d'indiquer le symbole dollar pour utiliser IMPORT MULTIEXCEL. Toutefois, vous devez l'inclure ou l'exclure dans les situations suivantes :</p> <ul style="list-style-type: none"> <li>○ <b>Inclure "\$"</b> : si vous souhaitez importer uniquement des feuilles de calcul, et aucune plage nommée, incluez le symbole dollar à la fin du nom de la feuille de calcul</li> <li>○ <b>Exclure "\$"</b> : si vous souhaitez importer des plages nommées ou des feuilles de calcul et des plages nommées dans une seule opération d'importation, n'incluez pas le symbole dollar</li> </ul>
<p>PREFIX</p>	<p>Ajoutez comme préfixe le nom du fichier Excel au nom des tables Analytics.</p>

Nom	Description
optionnel	<p><b>Astuce</b></p> <p>Si les feuilles de calcul de différents fichiers portent le même nom, ajouter comme préfixe le nom du fichier Excel vous permet d'éviter des conflits de noms de table.</p>
KEEPTITLE optionnel	<p>Traitez la première ligne de données en tant que noms de champs plutôt que comme données. Si vous ignorez ce paramètre, les noms des champs génériques sont utilisés.</p> <p><b>Remarque</b></p> <p>L'ensemble des premières lignes des feuilles de calcul et des plages nommées que vous importez doit utiliser une approche cohérente. Les premières lignes doivent être soit des noms de champs soit des données dans tous les jeux de données. Évitez de mélanger les deux dans une seule et même opération d'importation.</p> <p>Si les jeux de données adoptent une approche incohérente pour les premières lignes, utilisez deux opérations d'importation distinctes.</p>
CHARMAX <i>longueur_champ_max</i> optionnel	<p>La longueur maximale en caractères d'un champ d'une table Analytics qui provient d'une donnée de type caractère dans un fichier de données source.</p>

## Exemples

Dans les exemples qui suivent, on suppose que les données des transactions mensuelles pendant trois ans sont stockées dans trois fichiers Excel :

- `Transactions_FY18.xlsx`
- `Transactions_FY17.xlsx`
- `Transactions_FY16.xlsx`

Chaque fichier Excel comporte 12 feuilles de calcul, une pour chaque mois de l'année. Les feuilles de calcul incluent aussi certaines plages nommées identifiant différents sous-ensembles de transactions.

### Remarque

Une table Analytics distincte est créée pour chaque feuille de calcul ou plage nommée que vous importez.

## Importer des feuilles de calcul

### Importer toutes les feuilles de calcul FY18

Vous souhaitez importer les 12 feuilles de calcul mensuelles du fichier Excel FY18 et ignorer les plages nommées.

- vous utilisez le symbole caractère générique (\*) à l'emplacement du mois dans le nom de chaque feuille de calcul
- incluez le symbole dollar (\$) à la fin du nom de la feuille de calcul pour sélectionner uniquement des feuilles de calcul, et aucune plage nommée

```
IMPORT MULTIXCEL FROM "Transactions_FY18.xlsx" TABLE "Trans_*$"
```

## Importez toutes les feuilles de calcul FY18, conservez les noms des champs et indiquez la longueur maximale des champs caractère

Cet exemple est le même que le précédent, mais vous souhaitez conserver les noms de champs des fichiers Excel ainsi que limiter la longueur des champs caractère.

- vous incluez KEeptITLE pour utiliser la première ligne de données Excel comme noms de champ
- vous incluez CHARMAX 50 pour que les champs provenant de données de type caractère dans le fichier Excel aient une longueur limitée à 50 caractères dans la table Analytics ainsi créée

```
IMPORT MULTIXCEL FROM "Transactions_FY18.xlsx" TABLE "Trans_*$" KEeptITLE  
CHARMAX 50
```

## Importer toutes les feuilles de calcul de l'ensemble des trois fichiers

Vous souhaitez importer les 36 feuilles de calcul mensuelles des trois fichiers Excel et vous souhaitez ignorer les pages nommées.

- vous utilisez le symbole caractère générique (\*) à l'emplacement du mois dans le nom de chaque feuille de calcul
- incluez le symbole dollar (\$) à la fin du nom de la feuille de calcul pour sélectionner uniquement des feuilles de calcul, et aucune plage nommée
- vous utilisez le symbole caractère générique (\*) à l'emplacement de l'année dans le nom de chaque fichier Excel
- afin de réduire toute possibilité de conflits de noms, vous utilisez PREFIX pour ajouter le nom du fichier Excel source en guise de préfixe à chaque nom de table Analytics

```
IMPORT MULTIXCEL FROM "Transactions_FY*.xlsx" TABLE "Trans_*$" PREFIX
```

## Importer des plages nommées

### Importer toutes les plages nommées "Transaction\_commerciale" FY18

Vous souhaitez importer toutes les plages nommées "Transaction\_commerciale" du fichier Excel FY18 et ignorer les feuilles de calcul et les autres plages nommées.

- vous utilisez le symbole de caractère générique (\*) à l'emplacement d'un identifiant unique dans les noms des différentes plages

- vous excluez le symbole dollar (\$) pour que les plages nommées puissent être sélectionnées

```
IMPORT MULTIEXCEL FROM "Transactions_FY18.xlsx" TABLE "Transaction_commerciale_**"
```

## Importer les feuilles de calcul et les plages nommées

### Importer toutes les feuilles de calcul et toutes les plages nommées FY18

Vous souhaitez importer les 12 feuilles de calcul mensuelles et toutes les plages nommées du fichier Excel FY18.

- avec TABLE, vous utilisez uniquement le symbole caractère générique (\*) pour que toutes les feuilles de calcul et plages nommées du fichier soient sélectionnées
- vous excluez le symbole dollar (\$) pour que les plages nommées puissent être sélectionnées

```
IMPORT MULTIEXCEL FROM "Transactions_FY18.xlsx" TABLE "**"
```

## Gérer des répertoires

### Importer toutes les feuilles de calcul de tous les fichiers Excel dans le dossier spécifié

Vous souhaitez importer toutes les feuilles de calcul de tous les fichiers Excel dans le dossier `C:\Point of Sale Audit\Données\Fichiers maîtres Transaction`.

- avec TABLE, vous utilisez uniquement le symbole caractère générique (\*) pour que toutes les feuilles de calcul de chaque fichier soient sélectionnées et vous utilisez le symbole dollar (\$) pour que seules les feuilles de calcul soient sélectionnées, et aucune plage nommée
- afin de réduire toute possibilité de conflits de noms, vous utilisez PREFIX pour ajouter le nom du fichier Excel source en guise de préfixe à chaque nom de table Analytics

```
IMPORT MULTIEXCEL FROM "C:\Point of Sale Audit\Données\Fichiers maîtres Transaction" TABLE "$*" PREFIX
```

### Importez toutes les feuilles de calcul de tous les fichiers Excel dans le dossier spécifié et enregistrez les tables Analytics dans un autre dossier

Cet exemple est le même que le précédent, mais au lieu d'enregistrer les tables Analytics dans le dossier de projet racine, vous voulez les enregistrer dans le dossier `C:\Point of Sale Audit\Données\Données de travail transaction`.

```
IMPORT MULTIEXCEL TO "C:\Point of Sale Audit\Données\Données de travail transaction" FROM "C:\Point of Sale Audit\Données\Fichiers maîtres Transaction" TABLE "$*" PREFIX
```

# Remarques

## Commandes IMPORT EXCEL multiples

La commande IMPORT MULTIEXCEL exécute en fait plusieurs commandes IMPORT EXCEL individuelles, une pour chaque feuille de calcul importée. Si vous double-cliquez sur l'entrée IMPORT MULTIEXCEL dans la trace, chaque commande IMPORT EXCEL s'affiche individuellement dans la zone d'affichage.

## La dernière table importée est ouverte automatiquement

IMPORT MULTIEXCEL ne prend pas en charge le mot-clé OPEN. Toutefois, après l'exécution de la commande, la dernière table importée s'ouvre automatiquement.

## Combiner plusieurs feuilles de calcul après les avoir importées

Après avoir importé plusieurs feuilles de calcul dans des tables Analytics individuelles, vous voudrez peut-être les combiner en une seule table Analytics. Par exemple, vous pouvez combiner les données de douze tables mensuelles en une seule table annuelle contenant toutes les données.

Pour obtenir des informations sur la combinaison de plusieurs tables Analytics, consultez la rubrique "Commande APPEND" Page 73.

# Commande IMPORT ODBC

Crée une table Analytics en définissant et en important des données à partir d'une source de données ODBC.

ODBC signifie « Open Database Connectivity », c'est une méthode standard d'accès aux bases de données.

## Syntaxe

```
IMPORT ODBC SOURCE nom_source TABLE nom_table <QUALIFIER qualificatif_données>
<OWNER nom_utilisateur> <USERID id_utilisateur> <PASSWORD num> <WHERE clause_where>
<TO nom_table> <WIDTH longueur_champ_max> <MAXIMUM longueur_champ_max>
<FIELDS champ <;...n>>
```

## Paramètres

Nom	Description
SOURCE <i>nom_source</i>	<p>Nom de la source de données (DSN) ODBC à laquelle se connecter. Le DSN doit déjà exister et être configuré correctement.</p> <p><b>Remarque</b></p> <p>Vous êtes limité aux sources de données qui utilisent les pilotes ODBC Windows qui sont installés sur votre ordinateur. Les connecteurs de données Analytics natifs pouvant être utilisés avec la commande ACCESSDATA ne sont peut-être pas disponibles avec IMPORT ODBC.</p>
TABLE <i>nom_table</i>	<p>Nom de table de la source de données ODBC à partir de laquelle les données doivent être importées.</p> <p><i>nom_table</i> se rapporte généralement à une table de base de données dans les données sources, mais il peut renvoyer à tout élément importé en tant que table par Analytics. Par exemple, si vous utilisez Microsoft Text Driver, <i>nom_table</i> se rapporte au fichier texte à partir duquel vous souhaitez importer les données.</p>
QUALIFIER <i>qualificatif_données</i> optionnel	<p>Caractère à utiliser en tant qu'identificateur de texte pour renvoyer à la ligne et identifier des valeurs de champs. Vous devez indiquer le caractère sous forme de chaîne entre guillemets.</p> <p>Utilisez des guillemets simples pour spécifier le caractère guillemet double : "".</p>
OWNER <i>nom_utilisateur</i> optionnel	<p>Nom du compte utilisateur de base de données propriétaire de la table à laquelle vous vous connectez.</p>
USERID <i>id_utilisateur</i>	<p>Nom d'utilisateur permettant d'accéder à la source de données.</p>

Nom	Description
optionnel	
PASSWORD <i>num</i> optionnel	<p>Définition de mot de passe à utiliser.</p> <p>Vous n'utilisez pas PASSWORD <i>num</i> pour inviter à saisir ou pour spécifier un mot de passe réel. La définition de mot de passe renvoie à un mot de passe fourni ou défini auparavant à l'aide de la commande PASSWORD, de la commande SET PASSWORD ou de la balise d'outil d'analyse PASSWORD.</p> <p><i>num</i> correspond au numéro de la définition de mot de passe. Par exemple, si deux mots de passe ont été fournis ou définis auparavant dans un script, ou lors de la planification d'un outil d'analyse, PASSWORD 2 indique que le mot de passe n° 2 est utilisé.</p> <p>Pour plus d'informations sur la fourniture ou définition de mots de passe, consultez les rubriques :</p> <ul style="list-style-type: none"> <li>• "Commande PASSWORD" Page 373</li> <li>• "Commande SET" Page 435</li> <li>• <a href="#">Balise d'outil d'analyse PASSWORD</a></li> </ul> <p>Pour plus d'informations sur la fourniture ou définition de mots de passe, consultez les rubriques :</p> <ul style="list-style-type: none"> <li>• <a href="#">Commande PASSWORD</a></li> <li>• <a href="#">Commande SET</a></li> <li>• <a href="#">Balise d'outil d'analyse PASSWORD</a></li> </ul>
WHERE <i>clause_where</i> optionnel	<p>Instruction SQL WHERE limitant le nombre d'enregistrements renvoyé à partir d'un critère que vous spécifiez. Doit être une instruction SQL valide et doit être saisie sous forme de chaîne entre guillemets :</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>WHERE "SALARY &gt; 50000".</p> </div>
TO <i>nom_table</i> optionnel	<p>Nom du fichier de données Analytics à créer.</p> <p>Indiquez <i>nom_table</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple, TO "Factures.FIL".</p> <p>Par défaut, le fichier de données (.FIL) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>◦ TO "C:\données\Factures.FIL"</li> <li>◦ TO "données\Factures.FIL".</li> </ul>
WIDTH <i>longueur_champ_max</i> optionnel	<p>La longueur maximale en caractères d'un champ de la table Analytics qui provient d'une donnée de type caractère dans la source que vous importez.</p> <p>Vous pouvez saisir une valeur comprise entre 1 et 254. La valeur par défaut est 50. Les données dépassant la longueur de champ maximale sont tronquées lors de l'importation dans Analytics.</p>
MAXIMUM <i>longueur_champ_max</i>	<p>Longueur maximale en caractères des champs de texte, de commentaire ou mémo que vous importez.</p>



Nom	Description
optionnel	Vous pouvez saisir une valeur comprise entre 1 et 1 100. La valeur par défaut est 100. Les données dépassant la longueur de champ maximale sont tronquées lors de l'importation dans Analytics.
FIELDS <i>champ &lt;:...n&gt;</i> optionnel	<p>Champs individuels dans les données source à importer. Indiquez le nom.</p> <p>Si vous en indiquez plusieurs, chaque champ doit être séparé par une virgule. Si vous ignorez FIELDS, tous les champs sont importés.</p> <p>Les noms de champ entre guillemets sont sensibles à la casse. Si vous utilisez des guillemets, la casse des noms de champ de FIELDS et de la source de données ODBC doit être identique. Si vous utilisez des guillemets et que la casse des noms est différente, les champs ne sont pas importés.</p> <p><b>Remarque</b></p> <p>FIELDS doit être placé en dernier dans les paramètres IMPORT ODBC. Si FIELDS n'est pas placé en dernier, la commande échoue.</p>

## Exemples

### Importation de données à partir de SQL Server

Vous importez des données d'une base de données SQL Server vers une table Analytics appelée **Trans\_Déc11** :

```
IMPORT ODBC SOURCE "AuditServeurSQL" TABLE "Transactions" OWNER "audit" TO "C:\ACL
DATA\Trans_Déc11.FIL" WIDTH 100 MAXIMUM 200 FIELDS
"NUMCARTE","LIMCRED","NUMCLIENT","MONTECHU"
```

## Remarques

### Ancienne méthode de connexion aux sources de données ODBC

La commande IMPORT ODBC est l'ancienne méthode de connexion aux sources de données compatibles ODBC d'Analytics. La nouvelle méthode de connexion aux sources de données ODBC utilise la fenêtre Accès aux données et la commande ACCESSDATA.

Vous pouvez continuer à utiliser IMPORT ODBC dans Analytics. Toutefois, cette méthode de connexion est maintenant uniquement disponible dans les scripts et à partir de la ligne de commande Analytics. Vous ne pouvez plus accéder à cette méthode de connexion dans l'**Assistant de définition de données**.

## Supprimer la partie correspondant à l'heure des valeurs DateHeure

Lorsque vous utilisez la commande IMPORT ODBC pour définir une table Analytics, vous pouvez supprimer la portion correspondant à l'heure des valeurs DateHeure en préfaçant la commande avec la commande SET SUPPRESSTIME ON.

Cette fonctionnalité permet de réintégrer des scripts Analytics écrits avant la version 10.0 d'Analytics, lorsque la partie correspondant à l'heure des valeurs DateHeure était tronquée automatiquement. Si SET SUPPRESSTIME ON n'est pas ajoutée à ces scripts, ils ne s'exécuteront pas dans la version avec DateHeure activée d'Analytics.

Pour plus d'informations, consultez la rubrique "SET SUPPRESSTIME" dans "Commande SET" Page 435.

# Commande IMPORT PDF

Crée une table Analytics en définissant et en important un fichier Adobe PDF.

## Syntaxe

```
IMPORT PDF TO table <PASSWORD num> nom_fichier_importation FROM nom_fichier_source
<SERVER nom_profil> ignorer_longueur <PARSER "VPDF"> <PAGES plage_page> {[syntaxe_enre-
gistrement] [syntaxe_champ] <...n>} <...n>
```

```
syntaxe_enregistrement ::=
RECORD nom_enregistrement type_enregistrement lignes_dans_enregistrement transparent [syn-
taxe_test] <...n>
```

```
syntaxe_test ::=
TEST inclure_exclure type_correspondance AT ligne_départ,position_départ,plage logique texte
```

```
syntaxe_champ ::=
FIELD nom type AT ligne_départ,position_départ SIZE longueur,lignes_dans_champ DEC valeur WID
octets PIC format AS nom_affichage
```

## Paramètres

### Paramètres généraux

Nom	Description
TO <i>table</i>	<p>Nom de la table Analytics dans laquelle les données doivent être importées.</p> <p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alphanumériques. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>
PASSWORD <i>num</i> optionnel	<p>Fichiers PDF protégés par mot de passe.</p> <p>Définition de mot de passe à utiliser.</p> <p>Vous n'utilisez pas PASSWORD <i>num</i> pour inviter à saisir ou pour spécifier un mot de passe réel. La définition de mot de passe renvoie à un mot de passe fourni ou défini</p>

Nom	Description
	<p>auparavant à l'aide de la commande PASSWORD, de la commande SET PASSWORD ou de la balise d'outil d'analyse PASSWORD.</p> <p><i>num</i> correspond au numéro de la définition de mot de passe. Par exemple, si deux mots de passe ont été fournis ou définis auparavant dans un script, ou lors de la planification d'un outil d'analyse, PASSWORD 2 indique que le mot de passe n° 2 est utilisé.</p> <p>Pour plus d'informations sur la fourniture ou définition de mots de passe, consultez les rubriques :</p> <ul style="list-style-type: none"> <li>• "Commande PASSWORD" Page 373</li> <li>• "Commande SET" Page 435</li> <li>• <a href="#">Balise d'outil d'analyse PASSWORD</a></li> </ul> <p>Pour plus d'informations sur la fourniture ou définition de mots de passe, consultez les rubriques :</p> <ul style="list-style-type: none"> <li>• <a href="#">Commande PASSWORD</a></li> <li>• <a href="#">Commande SET</a></li> <li>• <a href="#">Balise d'outil d'analyse PASSWORD</a></li> </ul>
<i>nom_fichier_importation</i>	<p>Nom du fichier de données Analytics à créer.</p> <p>Indiquez <i>nom_fichier_importation</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple, "Factures.FIL".</p> <p>Par défaut, le fichier de données (.FIL) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>◦ "C:\données\Factures.FIL"</li> <li>◦ "données\Factures.FIL"</li> </ul>
FROM <i>nom_fichier_source</i>	<p>Nom du fichier de données source. <i>nom_fichier_source</i> doit être une chaîne entre guillemets.</p> <p>Si le fichier de données source n'est pas enregistré dans le même répertoire que le projet Analytics, vous devez utiliser un chemin d'accès absolu ou un chemin d'accès relatif pour indiquer l'emplacement du fichier :</p> <ul style="list-style-type: none"> <li>◦ "C:\données\<i>nom_fichier_source</i>"</li> <li>◦ "données<i>nom_fichier_source</i>"</li> </ul>
SERVER <i>nom_profil</i> optionnel	<p>Nom de profil du serveur contenant les données à importer.</p>
<i>ignorer_longueur</i> optionnel	<p>Nombre d'octets à ignorer au début du fichier.</p> <p>Par exemple, si les 32 premiers octets contiennent des informations d'en-tête, spécifiez une valeur de 32 pour ignorer_longueur afin que ces informations soient omises.</p> <p><b>Remarque</b></p> <p>Pour les données Unicode, spécifiez uniquement un nombre d'octets pair. Indiquer un nombre impair d'octets peut entraîner des problèmes lors du traitement ultérieur des données importées.</p>

Nom	Description
PARSER "VPDF" optionnel	Utilisez l'analyseur VeryPDF pour analyser le fichier PDF pendant le processus de définition des fichiers.  Si vous ignorez PARSEUR, l'analyseur Xpdf par défaut est utilisé.  Si vous importez le PDF pour la première fois et que vous n'avez aucune raison de faire autrement, utilisez l'analyseur Xpdf par défaut. Si vous avez déjà rencontré des problèmes d'alignement des données lors de l'utilisation de Xpdf avec le fichier, utilisez l'analyseur VeryPDF pour voir si les résultats d'analyse sont meilleurs.
PAGES <i>plage_page</i> optionnel	Les pages à inclure si vous ne souhaitez pas importer toutes les pages du fichier PDF. <i>plage_page</i> doit être indiqué sous forme de chaîne entre guillemets.  Vous pouvez indiquer : <ul style="list-style-type: none"> <li>○ pages individuelles séparées par des virgules (1,3,5)</li> <li>○ étendues de pages (2-7)</li> <li>○ une combinaison de pages et de plages (1, 3, 5-7, 11)</li> </ul> Si vous omettez PAGES, toutes les pages du fichier PDF sont importées.

## Paramètre RECORD

Informations de définition générales des enregistrements.

### Remarque

Certaines informations de définition d'enregistrement sont indiquées à l'aide de codes numériques mappant vers les options dans l'Assistant de définition de données.

**Dans les scripts, indiquez le code numérique et non le nom de l'option.**

Nom	Description
RECORD <i>nom_enregistrement</i>	Nom de l'enregistrement dans l'Assistant de définition de données.  Il est obligatoire d'indiquer <i>nom_enregistrement</i> dans la commande IMPORT PDF, mais la valeur <i>nom_enregistrement</i> n'apparaît pas dans la table Analytics créée.  Dans l'Assistant de définition de données, Analytics fournit les noms par défaut basés sur le type d'enregistrement : <ul style="list-style-type: none"> <li>○ Détail</li> <li>○ En-tête</li> <li>○ Pied de page</li> </ul> Vous pouvez utiliser les noms par défaut ou indiquez des noms différents.
<i>type_enregistrement</i>	Les trois types d'enregistrement possibles lors de la définition d'un fichier PDF : <ul style="list-style-type: none"> <li>○ 0 : détail</li> <li>○ 1 : en-tête</li> <li>○ 2 : pied de page</li> </ul>

Nom	Description
	<p><b>Remarque</b></p> <p>Vous pouvez définir plusieurs jeux d'enregistrements d'en-tête et de pied de page dans une seule exécution d'IMPORT PDF, mais vous pouvez définir uniquement un seul jeu d'enregistrements de détail.</p>
<i>lignes_dans_enregistrement</i>	<p>Nombre de lignes occupées par un enregistrement dans le fichier PDF.</p> <p>Vous pouvez définir des enregistrements sur une seule ligne ou sur plusieurs lignes à associer aux données dans le fichier PDF.</p>
<i>transparent</i>	<p>Paramètre de transparence pour un enregistrement d'en-tête.</p> <p><b>Remarque</b></p> <p>S'applique aux enregistrements d'en-tête uniquement.</p> <ul style="list-style-type: none"> <li>○ 0 : pas transparent</li> <li>○ 1 : transparent</li> </ul> <p>Les enregistrements d'en-tête transparents ne ventilent pas les enregistrements de détail multilignes.</p> <p>Si un enregistrement d'en-tête ventile un enregistrement de détail multiligne dans le fichier source PDF, ce qui peut se produire à l'endroit d'un saut de page, indiquer 1 (transparent) unifie l'enregistrement de détail dans la table Analytics créée.</p>

## Paramètre TEST

Critères de définition d'un jeu d'enregistrements dans le fichier PDF. Vous pouvez avoir une ou plusieurs occurrences de TEST (8 max.) pour chaque occurrence de RECORD.

### Remarque

Certains critères sont indiqués à l'aide des codes numériques mappant vers les options dans l'Assistant de définition de données (les noms des options sont indiqués entre parenthèses ci-dessous).

**Dans les scripts, indiquez le code numérique et non le nom de l'option.**

Nom	Description
TEST <i>inclure_exclure</i>	<p>Comment traiter la correspondance des données :</p> <ul style="list-style-type: none"> <li>○ 0 : (<b>Inclure</b>) les données remplissant les critères sont incluses dans le jeu d'enregistrements</li> <li>○ 1 : (<b>Exclure</b>) les données remplissant les critères sont exclues du jeu d'enregistrements</li> </ul>
<i>type_correspondance</i>	<p>Type de correspondance à effectuer :</p> <ul style="list-style-type: none"> <li>○ 0 : (<b>Corresp. exacte</b>) : les enregistrements correspondants doivent contenir le caractère ou la chaîne de caractères à la ligne de départ spécifiée, à partir de la position spécifiée</li> <li>○ 2 : (<b>Alpha</b>) les enregistrements correspondants doivent contenir un ou plusieurs</li> </ul>

Nom	Description				
	<p>caractères alphabétiques à la ligne de départ spécifiée, à la position de départ spécifiée ou à toutes les positions de la plage spécifiée</p> <ul style="list-style-type: none"> <li>◦ 3 : (<b>Numérique</b>) les enregistrements correspondants doivent contenir un ou plusieurs caractères numériques à la Ligne spécifiée, à la position de départ spécifiée ou à toutes les positions de la plage spécifiée</li> <li>◦ 4 : (<b>Vide</b>) les enregistrements correspondants doivent contenir un ou plusieurs espaces vides à la ligne de départ spécifiée, à la position de départ spécifiée ou à toutes les positions de la plage spécifiée</li> <li>◦ 5 : (<b>Non vide</b>) les enregistrements correspondants doivent contenir un ou plusieurs caractères non vides (y compris les caractères spéciaux) à la ligne de départ spécifiée, à la position de départ spécifiée ou à toutes les positions de la plage spécifiée</li> <li>◦ 7 : (<b>Rechercher dans la ligne</b>) les enregistrements correspondants doivent contenir le caractère ou la chaîne de caractères n'importe où sur la Ligne de départ spécifiée</li> <li>◦ 8 : (<b>Rechercher dans la plage</b>) les enregistrements correspondants doivent contenir le caractère ou la chaîne de caractères spécifié, à la ligne de départ spécifiée, n'importe où dans la plage spécifiée</li> <li>◦ 10 : (<b>Modèle de texte</b>) les enregistrements correspondants doivent contenir les caractères correspondant au schéma de caractères spécifié, à la ligne de départ spécifiée, à partir de la position spécifiée</li> </ul>				
<p>AT <i>ligne_départ</i>, <i>position_départ</i>, <i>plage</i></p>	<ul style="list-style-type: none"> <li>◦ <b>ligne_départ</b> : ligne d'un enregistrement auquel les critères s'appliquent</li> </ul> <p>Par exemple, si vous créez un modèle de texte qui fait correspondre les codes postaux et que ceux-ci apparaissent à la troisième ligne d'un enregistrement d'adresse de trois lignes, vous devez spécifier 3 dans <i>ligne_départ</i>.</p> <p><b>Remarque</b></p> <p>Pour les enregistrements d'une seule ligne, la valeur <i>ligne_départ</i> est toujours 1.</p> <ul style="list-style-type: none"> <li>◦ <b>position_départ</b> : position d'octet de départ dans le fichier PDF pour la comparaison vis-à-vis des critères</li> <li>◦ <b>plage</b> : nombre d'octets à partir de la position d'octet de départ dans le fichier PDF à utiliser dans la comparaison vis-à-vis des critères</li> </ul> <p>Si vous utilisez une position d'octet de départ uniquement, sans plage, indiquez 0 pour <i>plage</i>.</p> <p><b>Remarque</b></p> <table border="1" data-bbox="604 1377 1354 1507"> <tbody> <tr> <td data-bbox="604 1377 1052 1440">Analytics non Unicode</td> <td data-bbox="1052 1377 1354 1440">1 octet = 1 caractère</td> </tr> <tr> <td data-bbox="604 1440 1052 1507">Analytics Unicode</td> <td data-bbox="1052 1440 1354 1507">2 octets = 1 caractère</td> </tr> </tbody> </table>	Analytics non Unicode	1 octet = 1 caractère	Analytics Unicode	2 octets = 1 caractère
Analytics non Unicode	1 octet = 1 caractère				
Analytics Unicode	2 octets = 1 caractère				
<p><i>logique</i></p>	<p>Relations logiques entre les critères :</p> <ul style="list-style-type: none"> <li>◦ 0 : (<b>Et</b>) le critère actuel et le critère suivant sont associés avec un AND logique</li> <li>◦ 1 : (<b>Ou</b>) le critère actuel et le critère suivant sont associés avec un OR logique</li> <li>◦ 4 : (<b>Nouveau groupe &gt; Et</b>) le critère actuel est le dernier d'un groupe de critères logiques et le groupe actuel et le groupe suivant sont liés avec un AND logique</li> <li>◦ 5 : (<b>Nouveau groupe &gt; Ou</b>) le critère actuel est le dernier d'un groupe de critères logiques et le groupe actuel et le groupe suivant sont liés avec un OR logique</li> <li>◦ 7 : (<b>Fin</b>) le critère actuel est le dernier d'un groupe de critères logiques</li> </ul>				
<p><i>texte</i></p>	<p>Caractères littéraux ou génériques par rapport auxquels les associer :</p>				

Nom	Description
	<ul style="list-style-type: none"> <li>○ <b>Pour Corresp. exacte, Rechercher dans la ligne ou Rechercher dans la plage</b> : indique le caractère ou la chaîne de caractères qui identifie de manière unique le jeu d'enregistrements dans le fichier PDF.</li> <li>○ <b>Pour Modèle de texte</b> : indique le schéma de caractères qui identifie de manière unique le jeu d'enregistrements dans le fichier PDF</li> </ul> <p>L'option <b>Modèle de texte</b> utilise la même syntaxe que la "Fonction MAP()" Page 667.</p> <p>Pour les autres correspondances, <i>texte</i> est une chaîne de caractères vide "".</p>

## Paramètres FIELD

Informations de définition de champ.

Nom	Description				
FIELD <i>nom type</i>	<p>Les champs individuels à importer à partir du fichier de données source, y compris le nom et le type de données du champ. Pour empêcher l'importation d'un champ, ne l'indiquez pas.</p> <p>Pour plus d'informations sur le <i>type</i>, consultez la rubrique "Identificateurs des types de données de champ" Page 306.</p>				
AT <i>ligne_départ, position_départ</i>	<ul style="list-style-type: none"> <li>○ <b>ligne_départ</b> : ligne de départ du champ dans l'enregistrement dans le fichier PDF</li> </ul> <p>Pour les enregistrements multilignes dans un fichier PDF, <i>ligne_départ</i> vous permet de commencer un champ à n'importe quelle ligne de l'enregistrement. <i>ligne_départ</i> est toujours 1 si <i>lignes_dans_enregistrement</i> est 1.</p> <ul style="list-style-type: none"> <li>○ <b>position_départ</b> : position d'octet de départ du champ dans le fichier PDF</li> </ul> <p><b>Remarque</b></p> <table border="1"> <tbody> <tr> <td>Analytics non Unicode</td> <td>1 octet = 1 caractère</td> </tr> <tr> <td>Analytics Unicode</td> <td>2 octets = 1 caractère</td> </tr> </tbody> </table> <p>Dans Analytics Unicode, normalement vous devez indiquer une position d'octet de départ avec un nombre impair. Indiquer une position de départ avec un chiffre pair peut entraîner l'affichage incorrect des caractères.</p>	Analytics non Unicode	1 octet = 1 caractère	Analytics Unicode	2 octets = 1 caractère
Analytics non Unicode	1 octet = 1 caractère				
Analytics Unicode	2 octets = 1 caractère				
SIZE <i>longueur, lignes_dans_champ</i>	<ul style="list-style-type: none"> <li>○ <b>longueur</b> : longueur du champ en octets dans le format de la table Analytics</li> </ul> <p><b>Remarque</b></p> <table border="1"> <tbody> <tr> <td>Analytics non Unicode</td> <td>1 octet = 1 caractère</td> </tr> <tr> <td>Analytics Unicode</td> <td>2 octets = 1 caractère</td> </tr> </tbody> </table> <p>Dans Analytics Unicode, spécifiez uniquement un nombre d'octets pair. Indiquer un nombre d'octets impair peut entraîner l'affichage incorrect des caractères.</p>	Analytics non Unicode	1 octet = 1 caractère	Analytics Unicode	2 octets = 1 caractère
Analytics non Unicode	1 octet = 1 caractère				
Analytics Unicode	2 octets = 1 caractère				



Nom	Description
	<ul style="list-style-type: none"> <li>◦ <b>lignes_dans_champ</b> : nombre de lignes occupées par une seule valeur de champ dans le fichier PDF</li> </ul> <p>Vous pouvez définir des champs unilignes ou multilignes pour associer les données du fichier.</p> <p><b>Remarque</b></p> <p>Le nombre de lignes spécifié pour un champ ne peut pas dépasser le nombre de lignes spécifié pour l'enregistrement contenant le champ.</p>
DEC <i>valeur</i>	Nombre de décimales contenues par les champs numériques
WID <i>octets</i>	<p>Largeur d'affichage du champ en nombre d'octets.</p> <p>La valeur indiquée contrôle la largeur d'affichage du champ dans les vues et rapports Analytics. La largeur d'affichage n'altère jamais les données ; elle peut toutefois masquer des données si elle est plus courte que la longueur du champ.</p>
PIC <i>format</i>	<p><b>Remarque</b></p> <p>S'applique aux champs numériques ou DateHeure.</p> <ul style="list-style-type: none"> <li>◦ <b>champs numériques</b> : format d'affichage des valeurs numériques dans les vues et rapports Analytics</li> <li>◦ <b>champs DateHeure</b> : format physique des valeurs DateHeure dans les données source (ordre des caractères date et heure, séparateurs, etc.)</li> </ul> <p><b>Remarque</b></p> <p>Pour les champs DateHeure, le <i>format</i> doit correspondre exactement au format physique dans les données source. Par exemple, si les données source sont 12/31/2014, vous devez saisir la date au format "MM/JJ/AAAA".</p> <p>Le <i>format</i> doit être placé entre guillemets.</p>
AS <i>nom_affichage</i>	<p>Le nom d'affichage (titre de la colonne) pour le champ dans la vue dans la nouvelle table Analytics.</p> <p>Indiquez le <i>nom_affichage</i> sous forme de chaîne entre guillemets. Utilisez un point-virgule (;) entre les mots si vous souhaitez insérer un saut de ligne dans le titre de la colonne.</p> <p>AS est requis lors de la définition de FIELD. Pour que le nom d'affichage soit identique au champ nom, saisissez une valeur <i>nom_affichage</i> vide à l'aide de la syntaxe suivante : AS "". Vérifiez qu'il n'y a aucune espace entre les deux guillemets doubles.</p>

## Exemples

### Importer des données à partir d'une page spécifique d'un fichier PDF

Vous importez des données à partir de la page 1 d'un fichier PDF protégé par mot de passe, `Four-nisseurs.pdf` :

Un jeu d'enregistrements de détail, avec trois champs, est créé dans la table Analytics créée, **Liste\_Fournisseurs** :

```
IMPORT PDF TO Liste_Fournisseurs PASSWORD 1 "Liste_Fournisseurs.FIL" FROM "Liste_Fournisseurs.pdf" 2 PAGES "1" RECORD "Détail" 0 1 0 TEST 0 3 AT 1,1,0 7 "" FIELD "Numéro_fournisseur" C AT 1,1 SIZE 10,1 DEC 0 WID 10 PIC "" AS "" FIELD "Nom_Fournisseur" C AT 1,33 SIZE 58,1 DEC 0 WID 58 PIC "" AS "" FIELD "Dernière_Date_active" D AT 1,277 SIZE 20,1 DEC 0 WID 20 PIC "JJ/MM/AAAA" AS ""
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Dépannage pour les importations PDF dans l'édition Unicode d'Analytics

Si vous rencontrez des difficultés lorsque vous importez un fichier PDF à l'aide de l'édition Unicode d'Analytics, il se peut que le problème soit associé aux spécifications des longueurs :

- Si des caractères de langues étrangères apparaissent anormalement ou si le format de la table Analytics créée est asymétrique, vérifiez que la *longueur* de SIZE est bien définie sur un nombre pair.

Indiquer un nombre impair d'octets pour la *longueur* de SIZE peut entraîner des problèmes lors du traitement ultérieur des données importées.

- Si la table Analytics est créée, mais qu'elle ne contient aucun enregistrement, essayez de définir *ignorer\_longueur* sur 2 ou sur un autre nombre pair si des données d'en-tête se trouvent au début du fichier que vous souhaitez ignorer.

## Identificateurs des types de données de champ

Le tableau ci-dessous répertorie les lettres que vous devez utiliser lorsque vous indiquez *type* pour FIELD. Chaque lettre correspond à un type de données.

Par exemple, si vous définissez un champ Nom, qui utilise un type de données caractère, vous devez indiquer « C » : FIELD "Nom" C.

### Remarque

Lorsque vous utilisez l'**Assistant de définition de données** pour définir une table qui comprend des champs EBCDIC, Unicode ou ASCII, ces derniers se voient automatiquement attribuer la lettre « C » (pour le type CHARACTER).

Lorsque vous saisissez manuellement une instruction IMPORT ou que vous modifiez une instruction IMPORT existante, vous pouvez remplacer les lettres « E » ou « U » par des champs EBCDIC ou Unicode.

Lettre	Type de données
A	ACL
B	BINARY
C	CHARACTER
D	DATETIME
E	EBCDIC
F	FLOAT
G	ACCPAC
I	IBMFLOAT
K	UNSIGNED
L	LOGICAL
N	PRINT
P	PACKED
Q	BASIC
R	MICRO
S	CUSTOM
T	PCASCII
U	UNICODE
V	VAXFLOAT
X	NUMERIC

Lettre	Type de données
Y	UNISYS
Z	ZONED

# Commande IMPORT PRINT

Crée une table Analytics en définissant et en important un fichier image d'impression (rapport).

## Syntaxe

```
IMPORT PRINT TO table nom_fichier_importation FROM nom_fichier_source <SERVER nom_profil>
valeur_jeu_caractères <numéro_page_code> {[syntaxe_enregistrement] [syntaxe_champ] <...n>}
<...n>
```

```
syntaxe_enregistrement ::=
RECORD nom_enregistrement type_enregistrement lignes_dans_enregistrement transparent [syn-
taxe_test] <...n>
```

```
syntaxe_test ::=
TEST inclure_exclure type_correspondance AT ligne_départ,position_départ,plage logique texte
```

```
syntaxe_champ ::=
FIELD nom type AT ligne_départ,position_départ SIZE longueur,lignes_dans_champ DEC valeur WID
octets PIC format AS nom_affichage
```

## Paramètres

### Paramètres généraux

Nom	Description
TO <i>table</i>	<p>Nom de la table Analytics dans laquelle les données doivent être importées.</p> <p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alpha-numériques. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>
<i>nom_fichier_importation</i>	<p>Nom du fichier de données Analytics à créer.</p> <p>Indiquez <i>nom_fichier_importation</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple, "Factures.FIL".</p> <p>Par défaut, le fichier de données (.FIL) est enregistré dans le dossier contenant le</p>

Nom	Description
	<p>projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>o "C:\données\Factures.FIL"</li> <li>o "données\Factures.FIL"</li> </ul>
FROM <i>nom_fichier_source</i>	<p>Nom du fichier de données source. <i>nom_fichier_source</i> doit être une chaîne entre guillemets.</p> <p>Si le fichier de données source n'est pas enregistré dans le même répertoire que le projet Analytics, vous devez utiliser un chemin d'accès absolu ou un chemin d'accès relatif pour indiquer l'emplacement du fichier :</p> <ul style="list-style-type: none"> <li>o "C:\données\<i>nom_fichier_source</i>"</li> <li>o "données<i>nom_fichier_source</i>"</li> </ul>
SERVER <i>nom_profil</i> optionnel	Nom de profil du serveur contenant les données à importer.
<i>valeur_jeu_caractères</i>	<p>L'ensemble de caractères utilisé pour coder le fichier image d'impression (rapport). Les valeurs suivantes sont prises en charge :</p> <ul style="list-style-type: none"> <li>o 0 : ASCII</li> <li>o 1 : EBCDIC</li> <li>o 2 : Unicode</li> <li>o 3 : Texte codé</li> </ul>
<i>numéro_page_code</i> optionnel	Si vous avez indiqué 3 (texte codé) pour <i>valeur_jeu_caractères</i> , saisissez également un numéro de page codé.

## Paramètre RECORD

Informations de définition générales des enregistrements.

### Remarque

Certaines informations de définition d'enregistrement sont indiquées à l'aide de codes numériques mappant vers les options dans l'Assistant de définition de données.

**Dans les scripts, indiquez le code numérique et non le nom de l'option.**

Nom	Description
RECORD <i>nom_enregistrement</i>	<p>Nom de l'enregistrement dans l'Assistant de définition de données.</p> <p>Il est obligatoire d'indiquer <i>nom_enregistrement</i> dans la commande IMPORT PRINT, mais la valeur <i>nom_enregistrement</i> n'apparaît pas dans la table Analytics créée.</p> <p>Dans l'Assistant de définition de données, Analytics fournit les noms par défaut basés sur le type d'enregistrement :</p> <ul style="list-style-type: none"> <li>o Détail</li> </ul>

Nom	Description
	<ul style="list-style-type: none"> <li>○ En-tête<i>n</i></li> <li>○ Pied de page<i>n</i></li> </ul> <p>Vous pouvez utiliser les noms par défaut ou indiquez des noms différents.</p>
<i>type_enregistrement</i>	<p>Les trois types d'enregistrement possibles lors de la définition d'un fichier Image d'impression :</p> <ul style="list-style-type: none"> <li>○ 0 : détail</li> <li>○ 1 : en-tête</li> <li>○ 2 : pied de page</li> </ul> <p><b>Remarque</b></p> <p>Vous pouvez définir plusieurs jeux d'enregistrements d'en-tête et de pied de page dans une seule exécution d'IMPORT PRINT, mais vous pouvez définir un seul jeu d'enregistrements de détail uniquement.</p>
<i>lignes_dans_enregistrement</i>	<p>Nombre de lignes occupées par un enregistrement dans le fichier Image d'impression.</p> <p>Vous pouvez définir des enregistrements sur une seule ligne ou sur plusieurs lignes à associer aux données dans le fichier.</p>
<i>transparent</i>	<p>Paramètre de transparence pour un enregistrement d'en-tête.</p> <p><b>Remarque</b></p> <p>S'applique aux enregistrements d'en-tête uniquement.</p> <ul style="list-style-type: none"> <li>○ 0 : pas transparent</li> <li>○ 1 : transparent</li> </ul> <p>Les enregistrements d'en-tête transparents ne ventilent pas les enregistrements de détail multilignes.</p> <p>Si un enregistrement d'en-tête ventile un enregistrement de détail multiligne dans le fichier source Image d'impression, ce qui peut se produire à l'endroit d'un saut de page, indiquer 1 (transparent) unifie l'enregistrement de détail dans la table Analytics créée.</p>

## Paramètre TEST

Critères de définition d'un jeu d'enregistrements dans le fichier Image d'impression. Vous pouvez avoir une ou plusieurs occurrences de TEST (8 max.) pour chaque occurrence de RECORD.

### Remarque

Certains critères sont indiqués à l'aide des codes numériques mappant vers les options dans l'Assistant de définition de données (les noms des options sont indiqués entre parenthèses ci-dessous).

**Dans les scripts, indiquez le code numérique et non le nom de l'option.**

Nom	Description
TEST <i>inclure_exclure</i>	Comment traiter la correspondance des données :

Nom	Description
	<ul style="list-style-type: none"> <li>◦ 0 : (<b>Inclure</b>) les données remplissant les critères sont incluses dans le jeu d'enregistrements</li> <li>◦ 1 : (<b>Exclure</b>) les données remplissant les critères sont exclues du jeu d'enregistrements</li> </ul>
<i>type_correspondance</i>	<p>Type de correspondance à effectuer :</p> <ul style="list-style-type: none"> <li>◦ 0 : (<b>Corresp. exacte</b>) : les enregistrements correspondants doivent contenir le caractère ou la chaîne de caractères à la ligne de départ spécifiée, à partir de la position spécifiée</li> <li>◦ 2 : (<b>Alpha</b>) les enregistrements correspondants doivent contenir un ou plusieurs caractères alphabétiques à la ligne de départ spécifiée, à la position de départ spécifiée ou à toutes les positions de la plage spécifiée</li> <li>◦ 3 : (<b>Numérique</b>) les enregistrements correspondants doivent contenir un ou plusieurs caractères numériques à la Ligne spécifiée, à la position de départ spécifiée ou à toutes les positions de la plage spécifiée</li> <li>◦ 4 : (<b>Vide</b>) les enregistrements correspondants doivent contenir un ou plusieurs espaces vides à la ligne de départ spécifiée, à la position de départ spécifiée ou à toutes les positions de la plage spécifiée</li> <li>◦ 5 : (<b>Non vide</b>) les enregistrements correspondants doivent contenir un ou plusieurs caractères non vides (y compris les caractères spéciaux) à la ligne de départ spécifiée, à la position de départ spécifiée ou à toutes les positions de la plage spécifiée</li> <li>◦ 7 : (<b>Rechercher dans la ligne</b>) les enregistrements correspondants doivent contenir le caractère ou la chaîne de caractères n'importe où sur la Ligne de départ spécifiée</li> <li>◦ 8 : (<b>Rechercher dans la plage</b>) les enregistrements correspondants doivent contenir le caractère ou la chaîne de caractères spécifié, à la ligne de départ spécifiée, n'importe où dans la plage spécifiée</li> <li>◦ 10 : (<b>Modèle de texte</b>) les enregistrements correspondants doivent contenir les caractères correspondant au schéma de caractères spécifié, à la ligne de départ spécifiée, à partir de la position spécifiée</li> </ul>
<i>AT ligne_départ, position_départ, plage</i>	<ul style="list-style-type: none"> <li>◦ <b>ligne_départ</b> : ligne d'un enregistrement auquel les critères s'appliquent</li> </ul> <p>Par exemple, si vous créez un modèle de texte qui fait correspondre les codes postaux et que ceux-ci apparaissent à la troisième ligne d'un enregistrement d'adresse de trois lignes, vous devez spécifier 3 dans <i>ligne_départ</i>.</p> <div style="border-left: 2px solid blue; padding-left: 10px; margin-left: 20px;"> <p><b>Remarque</b></p> <p>Pour les enregistrements d'une seule ligne, la valeur <i>ligne_départ</i> est toujours 1.</p> </div> <ul style="list-style-type: none"> <li>◦ <b>position_départ</b> : position d'octet de départ dans le fichier Image d'impression pour la comparaison vis-à-vis des critères</li> <li>◦ <b>plage</b> : nombre d'octets à partir de la position d'octet de départ dans le fichier Image d'impression à utiliser dans la comparaison vis-à-vis des critères</li> </ul> <p>Si vous utilisez une position d'octet de départ uniquement, sans plage, indiquez 0 pour <i>plage</i>.</p>



Nom	Description						
	<p><b>Remarque</b></p> <table border="1"> <tr> <td>Analytics non Unicode</td> <td>1 octet = 1 caractère</td> </tr> <tr> <td>Analytics Unicode, données ASCII étendu (ANSI)</td> <td>1 octet = 1 caractère</td> </tr> <tr> <td>Analytics Unicode, données Unicode</td> <td>2 octets = 1 caractère</td> </tr> </table> <p>Pour les données Unicode, <i>plage</i> doit être un nombre d'octets pair. Par exemple, 50,59 (10 octets). Spécifier un nombre d'octets impair peut empêcher la bonne correspondance vis-à-vis des critères.</p>	Analytics non Unicode	1 octet = 1 caractère	Analytics Unicode, données ASCII étendu (ANSI)	1 octet = 1 caractère	Analytics Unicode, données Unicode	2 octets = 1 caractère
Analytics non Unicode	1 octet = 1 caractère						
Analytics Unicode, données ASCII étendu (ANSI)	1 octet = 1 caractère						
Analytics Unicode, données Unicode	2 octets = 1 caractère						
<i>logique</i>	<p>Relations logiques entre les critères :</p> <ul style="list-style-type: none"> <li>○ 0 : <b>(Et)</b> le critère actuel et le critère suivant sont associés avec un AND logique</li> <li>○ 1 : <b>(Ou)</b> le critère actuel et le critère suivant sont associés avec un OR logique</li> <li>○ 4 : <b>(Nouveau groupe &gt; Et)</b> le critère actuel est le dernier d'un groupe de critères logiques et le groupe actuel et le groupe suivant sont liés avec un AND logique</li> <li>○ 5 : <b>(Nouveau groupe &gt; Ou)</b> le critère actuel est le dernier d'un groupe de critères logiques et le groupe actuel et le groupe suivant sont liés avec un OR logique</li> <li>○ 7 : <b>(Fin)</b> le critère actuel est le dernier d'un groupe de critères logiques</li> </ul>						
<i>texte</i>	<p>Caractères littéraux ou génériques par rapport auxquels les associer :</p> <ul style="list-style-type: none"> <li>○ <b>Pour Corresp. exacte, Rechercher dans la ligne ou Rechercher dans la plage</b> : indique le caractère ou la chaîne de caractères qui identifie de manière unique le jeu d'enregistrements dans le fichier image d'impression</li> <li>○ <b>Pour Modèle de texte</b> : indique le schéma de caractères qui identifie de manière unique le jeu d'enregistrements dans le fichier image d'impression</li> </ul> <p>L'option <b>Modèle de texte</b> utilise la même syntaxe que la "Fonction MAP( )" Page 667.</p> <p>Pour les autres correspondances, <i>texte</i> est une chaîne de caractères vide "".</p>						

## Paramètres FIELD

Informations de définition de champ.

Nom	Description
FIELD <i>nom type</i>	<p>Les champs individuels à importer à partir du fichier de données source, y compris le nom et le type de données du champ. Pour empêcher l'importation d'un champ, ne l'indiquez pas.</p> <p>Pour plus d'informations sur le <i>type</i>, consultez la rubrique "Identificateurs des types de données de champ" Page 316.</p>
AT <i>ligne_départ, position_départ</i>	<ul style="list-style-type: none"> <li>○ <b>ligne_départ</b> : ligne de départ du champ dans l'enregistrement dans le fichier Image d'impression</li> </ul> <p>Pour les enregistrements multilignes dans un fichier PDF, <i>ligne_départ</i> vous permet de commencer un champ à n'importe quelle ligne de l'enregistrement. <i>ligne_départ</i></p>

Nom	Description						
	<p>est toujours 1 si <i>lignes_dans_enregistrement</i> est 1.</p> <ul style="list-style-type: none"> <li>◦ <b>position_départ</b> : position d'octet de départ du champ dans le fichier Image d'impression</li> </ul> <p><b>Remarque</b></p> <table border="1"> <tr> <td>Analytics non Unicode</td> <td>1 octet = 1 caractère</td> </tr> <tr> <td>Analytics Unicode, données ASCII étendu (ANSI)</td> <td>1 octet = 1 caractère</td> </tr> <tr> <td>Analytics Unicode, données Unicode</td> <td>2 octets = 1 caractère</td> </tr> </table>	Analytics non Unicode	1 octet = 1 caractère	Analytics Unicode, données ASCII étendu (ANSI)	1 octet = 1 caractère	Analytics Unicode, données Unicode	2 octets = 1 caractère
Analytics non Unicode	1 octet = 1 caractère						
Analytics Unicode, données ASCII étendu (ANSI)	1 octet = 1 caractère						
Analytics Unicode, données Unicode	2 octets = 1 caractère						
SIZE <i>longueur, lignes_dans_champ</i>	<ul style="list-style-type: none"> <li>◦ <b>longueur</b> : longueur du champ en octets dans le format de la table Analytics</li> </ul> <p><b>Remarque</b></p> <table border="1"> <tr> <td>Analytics non Unicode</td> <td>1 octet = 1 caractère</td> </tr> <tr> <td>Analytics Unicode, données ASCII étendu (ANSI)</td> <td>1 octet = 1 caractère</td> </tr> <tr> <td>Analytics Unicode, données Unicode</td> <td>2 octets = 1 caractère</td> </tr> </table> <p>Pour les données Unicode, spécifiez uniquement un nombre d'octets pair. Indiquer un nombre d'octets impair peut entraîner l'affichage incorrect des caractères.</p> <ul style="list-style-type: none"> <li>◦ <b>lignes_dans_champ</b> : nombre de lignes occupées par une seule valeur de champ dans le fichier Image d'impression</li> </ul> <p>Vous pouvez définir des champs unilignes ou multilignes pour associer les données du fichier.</p> <p><b>Remarque</b></p> <p>Le nombre de lignes spécifié pour un champ ne peut pas dépasser le nombre de lignes spécifié pour l'enregistrement contenant le champ.</p>	Analytics non Unicode	1 octet = 1 caractère	Analytics Unicode, données ASCII étendu (ANSI)	1 octet = 1 caractère	Analytics Unicode, données Unicode	2 octets = 1 caractère
Analytics non Unicode	1 octet = 1 caractère						
Analytics Unicode, données ASCII étendu (ANSI)	1 octet = 1 caractère						
Analytics Unicode, données Unicode	2 octets = 1 caractère						
DEC <i>valeur</i>	Nombre de décimales contenues par les champs numériques						
WID <i>octets</i>	<p>Largeur d'affichage du champ en nombre d'octets.</p> <p>La valeur indiquée contrôle la largeur d'affichage du champ dans les vues et rapports Analytics. La largeur d'affichage n'altère jamais les données ; elle peut toutefois masquer des données si elle est plus courte que la longueur du champ.</p>						
PIC <i>format</i>	<p><b>Remarque</b></p> <p>S'applique aux champs numériques ou DateHeure.</p> <ul style="list-style-type: none"> <li>◦ <b>champs numériques</b> : format d'affichage des valeurs numériques dans les vues et rapports Analytics</li> <li>◦ <b>champs DateHeure</b> : format physique des valeurs DateHeure dans les données source (ordre des caractères date et heure, séparateurs, etc.)</li> </ul>						

Nom	Description
	<p><b>Remarque</b></p> <p>Pour les champs DateHeure, le <i>format</i> doit correspondre exactement au format physique dans les données source. Par exemple, si les données source sont 12/31/2014, vous devez saisir la date au format "MM/JJ/AAAA".</p> <p>Le <i>format</i> doit être placé entre guillemets.</p>
AS <i>nom_affichage</i>	<p>Le nom d'affichage (titre de la colonne) pour le champ dans la vue dans la nouvelle table Analytics.</p> <p>Indiquez le <i>nom_affichage</i> sous forme de chaîne entre guillemets. Utilisez un point-virgule (;) entre les mots si vous souhaitez insérer un saut de ligne dans le titre de la colonne.</p> <p>AS est requis lors de la définition de FIELD. Pour que le nom d'affichage soit identique au champ nom, saisissez une valeur <i>nom_affichage</i> vide à l'aide de la syntaxe suivante : AS "". Vérifiez qu'il n'y a aucune espace entre les deux guillemets doubles.</p>

## Exemples

### Importation de données à partir d'un fichier image d'impression (fichier de rapport)

Vous importez des données à partir du fichier Image d'impression (Rapport), `Report.txt`.

Un enregistrement d'en-tête et un jeu d'enregistrements de détail, avec cinq champs, est créé dans la table Analytics créée, `Rapport_stocks` :

```
IMPORT PRINT TO Rapport_stocks "Rapport_stocks.FIL" FROM "Rapport.txt" 0 RECORD "Entête1"
1 1 0 TEST 0 0 AT 1,17,0 7 ":" FIELD "Champ_1" C AT 1,19 SIZE 2,1 DEC 0 WID 2 PIC "" AS "Classe
prod" FIELD "Champ_2" C AT 1,24 SIZE 31,1 DEC 0 WID 31 PIC "" AS "Description prod" RECORD
"Détail" 0 1 0 TEST 0 0 AT 1,59,59 7 ":" FIELD "Champ_3" X AT 1,6 SIZE 9,1 DEC 0 WID 9 PIC "" AS
"ID élément" FIELD "Champ_4" C AT 1,16 SIZE 24,1 DEC 0 WID 24 PIC "" AS "Desc. élément" FIELD
"Champ_5" N AT 1,40 SIZE 10,1 DEC 0 WID 10 PIC "" AS "Disponible" FIELD "Champ_6" N AT 1,50
SIZE 12,1 DEC 2 WID 12 PIC "" AS "Coût" FIELD "Champ_7" N AT 1,62 SIZE 12,1 DEC 2 WID 12
PIC "" AS "Total"
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Identificateurs des types de données de champ

Le tableau ci-dessous répertorie les lettres que vous devez utiliser lorsque vous indiquez *type* pour FIELD. Chaque lettre correspond à un type de données.

Par exemple, si vous définissez un champ Nom, qui utilise un type de données caractère, vous devez indiquer « C » : FIELD "Nom" C.

### Remarque

Lorsque vous utilisez l'**Assistant de définition de données** pour définir une table qui comprend des champs EBCDIC, Unicode ou ASCII, ces derniers se voient automatiquement attribuer la lettre « C » (pour le type CHARACTER).

Lorsque vous saisissez manuellement une instruction IMPORT ou que vous modifiez une instruction IMPORT existante, vous pouvez remplacer les lettres « E » ou « U » par des champs EBCDIC ou Unicode.

Lettre	Type de données
A	ACL
B	BINARY
C	CHARACTER
D	DATETIME
E	EBCDIC
F	FLOAT
G	ACCPAC
I	IBMFLOAT
K	UNSIGNED
L	LOGICAL
N	PRINT
P	PACKED
Q	BASIC
R	MICRO
S	CUSTOM

Lettre	Type de données
T	PCASCII
U	UNICODE
V	VAXFLOAT
X	NUMERIC
Y	UNISYS
Z	ZONED

# Commande IMPORT SAP

Crée une table Analytics en important des données à partir d'un système SAP à l'aide de *Direct Link*.

## Syntaxe

```
IMPORT SAP PASSWORD num TO nom_table SAP SOURCE "SAP AGENT" détails_importation
```

## Paramètres

Nom	Description
PASSWORD <i>num</i>	<p>Définition de mot de passe à utiliser.</p> <p>Vous n'utilisez pas PASSWORD <i>num</i> pour inviter à saisir ou pour spécifier un mot de passe réel. La définition de mot de passe renvoie à un mot de passe fourni ou défini auparavant à l'aide de la commande PASSWORD, de la commande SET PASSWORD ou de la balise d'outil d'analyse PASSWORD.</p> <p><i>num</i> correspond au numéro de la définition de mot de passe. Par exemple, si deux mots de passe ont été fournis ou définis auparavant dans un script, ou lors de la planification d'un outil d'analyse, PASSWORD 2 indique que le mot de passe n° 2 est utilisé.</p> <p>Pour plus d'informations sur la fourniture ou définition de mots de passe, consultez les rubriques :</p> <ul style="list-style-type: none"> <li>• "Commande PASSWORD" Page 373</li> <li>• "Commande SET" Page 435</li> <li>• <a href="#">Balise d'outil d'analyse PASSWORD</a></li> </ul> <p>Pour plus d'informations sur la fourniture ou définition de mots de passe, consultez les rubriques :</p> <ul style="list-style-type: none"> <li>• <a href="#">Commande PASSWORD</a></li> <li>• <a href="#">Commande SET</a></li> <li>• <a href="#">Balise d'outil d'analyse PASSWORD</a></li> </ul> <p><b>Remarque</b></p> <p>Le mot de passe est utilisé pour accéder au système SAP.</p>
TO <i>nom_table</i>	<p>Nom de la table Analytics dans laquelle les données doivent être importées.</p> <p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alphanumériques. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>
SAP SOURCE	<p>Requis pour importer des données SAP. "SAP AGENT" est la seule option disponible.</p>

Nom	Description
"SAP AGENT"	
<i>détails_importation</i>	<p>Les détails de la requête. Doit être placée entre des balises &lt;q&gt;&lt;/q&gt;. Elle utilise les balises répertoriées dans "Balises de requête Direct Link" Page 321 pour définir la requête.</p> <p>La taille physique de cette valeur peut atteindre 16 Ko.</p>

## Exemples

### Effectuer une requête sur plusieurs tables

Dans l'exemple suivant, une requête sur plusieurs tables est effectuée en utilisant la commande IMPORT SAP.

La création d'une chaîne de requête valide nécessite une imbrication et une séquence de balises correctes. Les balises de l'exemple sont ordonnées et imbriquées correctement. Reportez-vous à cet exemple pour déterminer l'ordre et l'imbrication appropriés des balises de requête IMPORT SAP.

#### Remarque

Pour augmenter sa lisibilité, cet exemple est mis en forme sur plusieurs lignes. Dans votre script, la commande et la chaîne de requête doivent être saisies sans saut de ligne.

#### Astuce

La syntaxe d'une chaîne de requête IMPORT SAP est généralement complexe. Pour ajouter des commandes IMPORT SAP avec des chaînes de requête à vos scripts, la meilleure méthode consiste à copier une commande IMPORT SAP existante depuis l'onglet **Trace** vers Analytics, puis de modifier les balises de requête selon vos besoins.

```
IMPORT SAP PASSWORD 1 TO Doc_achat SAP SOURCE "SAP AGENT"
<q version="6.0">
  <s>0</s>
  <d>IDES</d>
  <u>mzunini</u>
  <c>800</c>
  <lg>en</lg>
  <cf>C:\ACL Data\Doc_achat.fil</cf>
  <sf>E:\Data\DL_JSMITH111107.DAT</sf>
  <jcount>11110701</jcount>
  <jname>DL_JSMITH111107.DAT</jname>
  <dl>75</dl>
  <m>2</m>
```

```
<dt>20140321</dt>
<tm>033000</tm>
<r>500</r>
<ar>0</ar>
<e>500</e>
<ts>
  <t>
    <n>EKKO</n>
    <a>T00001</a>
    <td>En-tête document achat</td>
    <fs>
      <f>EBELN</f>
      <f>BUKRS</f>
      <f>BSTYP</f>
      <f>BSART</f>
      <f>STATU</f>
      <f>WKURS</f>
    </fs>
    <wc>
      <w>
        <f>BUKRS</f>
        <o>0</o>
        <l>1000</l>
        <h></h>
      </w>
    </wc>
  </t>
  <t>
    <n>EKPO</n>
    <a>T00002</a>
    <td>Élément document achat</td>
    <fs>
      <f>EBELP</f>
      <f>WERKS</f>
      <f>MENGE</f>
      <f>BRTWR</f>
    </fs>
    <wc></wc>
  </t>
</ts>
<js>
<jc>
  <pt>
    <pa>T00001</pa>
```



```

<pf>EBELN</pf>
</pt>
<ct>
  <ca>T00002</ca>
  <cf>EBELN</cf>
</ct>
</jc>
</js>
</q>

```

## Remarques

La commande IMPORT SAP est prise en charge uniquement si Direct Link est installé et configuré.

Le tableau dans "Balises de requête Direct Link" bas répertorie les balises pouvant être incluses dans le paramètre *détails\_importation*. La colonne **Obligatoire** utilise les valeurs suivantes pour indiquer lorsque les balises doivent être présentes :

- **Y** : Obligatoire
- **N** : Facultatif
- **M** : Obligatoire uniquement pour les requêtes sur plusieurs tables
- **B** : Obligatoire, mais aucune valeur ne doit être transmise
- **W** : Facultative lorsque des filtres sont utilisés
- **S** : Obligatoire lorsque le mode planifié est spécifié

## Balises de requête Direct Link

Nom	Balise	Obligatoire	Description
Alias de table	<a>	M	L'alias qui identifie la table de façon unique dans la requête. Une même table peut ainsi être utilisée plusieurs fois. La longueur maximale est de 6 caractères.
Toutes les lignes	<ar>	Y	Indique que toutes les lignes correspondantes doivent être renvoyées avec le jeu de résultats de la requête. Les valeurs correctes sont : <b>1</b> : Remplace le nombre d'enregistrements spécifié dans la balise <r> (Nombre maximal de lignes) <b>0</b> : Renvoie le nombre d'enregistrements spécifié dans la balise <r> (Nombre maximal de lignes) Cette balise est toujours placée après les balises <r></r>.
Client	<c>	N	Le client au sein du système SAP.

Nom	Balise	Obligatoire	Description
Alias de table enfant	<ca>	M	L'alias de la table enfant.
Champ de table enfant	<cf>	M	Le champ de la table enfant sur lequel est basée la condition de jonction.
Nom de fichier client	<cf>	Y	Identifie le fichier cible sur le système client dans lequel les résultats de la requête sont stockés.
Table enfant	<ct>	M	La table enfant dans la condition de jonction.
Destination	<d>	N	Identifie une destination dans le fichier bibliothèque SAP RFC ( <code>sapn-wrfc.ini</code> ) qui est utilisé pour localiser un système SAP.
Longueur de données	<dl>	B	Le nombre de caractères sur chaque ligne, y compris les caractères retour chariot et saut de ligne marquant la fin de l'enregistrement (CR+LF ou les caractères hexadécimaux 0D+0A).
Date	<dt>	S	Obligatoire lors de l'utilisation du mode planifié. Indique l'heure d'exécution de la tâche SAP. Doit être au format AAAAMMJJ. Par exemple, le 31 décembre 2014 doit être indiqué sous la forme 20141231.
Lignes attendues	<e>	B	Le nombre attendu de lignes renvoyées par la requête.
Nom de champ	<f>	Y	Le nom du champ d'origine.
Champ de filtre	<f>	W	Le nom du champ d'origine auquel s'applique le filtre.
Champs	<fs>	Y	La liste des champs de la table qui sont renvoyés avec les résultats de requête.
Valeur élevée	<h>	W	Contient la valeur élevée lors de l'utilisation de l'opérateur Entre. Ignoré lors de l'utilisation d'un autre opérateur.
Condition de jonction	<jc>	M	La condition de jonction.
Nombre de tâches	<jcount>	B	Utilisé en interne par SAP pour identifier une requête en mode Fond.
Nom de tâche	<jname>	B	Utilisé en interne par SAP pour identifier une requête en mode Fond.
Relations de jonctions	<js>	Y	La liste des conditions de jonction qui lient les tables dans la requête.
Commutateur de jointure	<jw>	N	L'équivalent numérique du type de commutateur de jointure énuméré. Les valeurs correctes sont :

Nom	Balise	Obligatoire	Description
			<b>0</b> : Jointure fermée <b>1</b> : Jointure gauche extérieure
Valeur faible	<l>	W	Contient la valeur la plus faible lors de l'utilisation de l'opérateur Entre ou la valeur d'utilisation d'un autre opérateur.
Langue	<lg>	Y	L'identificateur de langue utilisé pour déterminer le paramètre régional des champs de base de données SAP.
Mode	<m>	Y	L'équivalent numérique du type énuméré en mode de soumission. Les valeurs correctes sont : <b>0</b> : Extraire maintenant <b>1</b> : Fond <b>2</b> : Planifié
Nom de table	<n>	Y	Le nom de table d'origine.
Opérateur	<o>	W	L'équivalent numérique du type d'opérateur énuméré. Les valeurs correctes sont : <b>0</b> : Égal à (=) <b>1</b> : Différent de (<>) <b>2</b> : Inférieur à (<) <b>3</b> : Inférieur ou égal à (<=) <b>4</b> : Supérieur à (>) <b>5</b> : Supérieur ou égal à (>=) <b>6</b> : Entre <b>7</b> : Contient
Alias de table parent	<pa>	M	L'alias de la table parent.
Champ de table parent	<pf>	M	Le champ de la table parent sur lequel est basée la condition de jonction.
Table parent	<pt>	M	La table parent dans la condition de jonction.
Requête	<q>	Y	Entoure une requête.
Nombre maximal de lignes	<r>	Y	Le nombre maximal de lignes que la requête doit renvoyer.
Sélectionné	<s>	Y	Si la balise <s> figure sous la balise <f>, elle indique si le champ est renvoyé ou non avec le jeu de résultats de la requête.

Nom	Balise	Obligatoire	Description
Système	<s>	Y	Si la balise <s> figure sous la balise <q>, elle identifie le type de système auquel cette requête se rapporte (actuellement, seul SAP est pris en charge).
Nom de fichier de serveur	<sf>	B	Identifie le fichier sur le serveur qui héberge les résultats d'une requête en mode Fond.
Nom du groupe de serveurs	<sg>	N	Nom du groupe de serveurs. 20 caractères maximum.
Nom du serveur	<sn>	N	Nom du serveur. 20 caractères maximum.
Table	<t>	Y	La table.
Description de table	<td>	Y	La description de table du dictionnaire de données SAP. Elle doit toujours figurer sous la balise <a>.
Durée	<tm>	S	Obligatoire lors de l'utilisation du mode planifié. Indique l'heure d'exécution de la tâche SAP.  Doit être au format hhmmss. Par exemple, pour indiquer 2 h 30 de l'après-midi, il faut utiliser 143000.
Tables	<ts>	Y	La liste des tables à partir desquelles la requête extrait des données.
Type de table	<tt>	Y	Le type de table SAP. Les valeurs correctes sont : <b>0</b> : en cluster <b>1</b> : transparent <b>2</b> : en pool <b>3</b> : vue
Nom d'utilisateur	<u>	N	Le nom de connexion de l'utilisateur.
Filtrer	<w>	W	Le filtre appliqué aux données de la table.
Filtres	<wc>	W	La liste des filtres qui s'appliquent aux données incluses dans la table.
Commutateur de filtres	<ws>	N	L'équivalent numérique du type de commutateur de filtres énuméré. Les valeurs correctes sont : <b>0</b> : (Ou) Et (Ou) <b>1</b> : (Et) Ou (Et)

# Commande IMPORT XBRL

Crée une table Analytics en définissant et en important un fichier XBRL.

## Syntaxe

```
IMPORT XBRL TO table nom_fichier_importation FROM nom_fichier_source CONTEXT nom_contexte <...n> [syntaxe_champ] <...n> <IGNORE num_champ> <...n>
```

```
syntaxe_champ ::=  
FIELD nom type AT position_départ DEC valeur WID octets PIC format AS nom_affichage
```

## Paramètres

Nom	Description
TO <i>table</i>	<p>Nom de la table Analytics dans laquelle les données doivent être importées.</p> <p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alphanumériques. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>
<i>nom_fichier_importation</i>	<p>Nom du fichier de données Analytics à créer.</p> <p>Indiquez <i>nom_fichier_importation</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple, "Factures.FIL".</p> <p>Par défaut, le fichier de données (.FIL) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>◦ "C:\données\Factures.FIL"</li> <li>◦ "données\Factures.FIL"</li> </ul>
FROM <i>nom_fichier_source</i>	<p>Nom du fichier de données source. <i>nom_fichier_source</i> doit être une chaîne entre guillemets.</p> <p>Si le fichier de données source n'est pas enregistré dans le même répertoire que le projet Analytics, vous devez utiliser un chemin d'accès absolu ou un chemin d'accès relatif pour indiquer l'emplacement du fichier :</p> <ul style="list-style-type: none"> <li>◦ "C:\données\<i>nom_fichier_source</i>"</li> <li>◦ "données<i>nom_fichier_source</i>"</li> </ul>

Nom	Description				
CONTEXT <i>nom_contexte</i>	Contexte XBRL à partir duquel la table doit être définie. Si vous indiquez plusieurs contextes, ils doivent tous être de même type (instantané, période ou toujours).				
FIELD <i>nom type</i>	<p>Les champs individuels à importer à partir du fichier de données source, y compris le nom et le type de données du champ. Pour empêcher l'importation d'un champ, ne l'indiquez pas.</p> <p>Pour plus d'informations sur le <i>type</i>, consultez la rubrique "Identificateurs des types de données de champ" Page opposée.</p>				
AT <i>position_départ</i>	<p>La position d'octet de départ du champ dans le fichier de données Analytics.</p> <p><b>Remarque</b></p> <table border="1"> <tr> <td>Analytics non Unicode</td> <td>1 octet = 1 caractère</td> </tr> <tr> <td>Analytics Unicode</td> <td>2 octets = 1 caractère</td> </tr> </table> <p>Dans Analytics Unicode, normalement vous devez indiquer une position d'octet de départ avec un nombre impair. Indiquer une position de départ avec un chiffre pair peut entraîner l'affichage incorrect des caractères.</p>	Analytics non Unicode	1 octet = 1 caractère	Analytics Unicode	2 octets = 1 caractère
Analytics non Unicode	1 octet = 1 caractère				
Analytics Unicode	2 octets = 1 caractère				
DEC <i>valeur</i>	Nombre de décimales contenues par les champs numériques				
WID <i>octets</i>	<p>La longueur du champ en octets dans le format de la table Analytics.</p> <p><b>Remarque</b></p> <table border="1"> <tr> <td>Analytics non Unicode</td> <td>1 octet = 1 caractère</td> </tr> <tr> <td>Analytics Unicode</td> <td>2 octets = 1 caractère</td> </tr> </table> <p>Dans Analytics Unicode, spécifiez uniquement un nombre d'octets pair. Indiquer un nombre d'octets impair peut entraîner l'affichage incorrect des caractères.</p>	Analytics non Unicode	1 octet = 1 caractère	Analytics Unicode	2 octets = 1 caractère
Analytics non Unicode	1 octet = 1 caractère				
Analytics Unicode	2 octets = 1 caractère				
PIC <i>format</i>	<p><b>Remarque</b></p> <p>S'applique aux champs numériques ou DateHeure.</p> <ul style="list-style-type: none"> <li>◦ <b>champs numériques</b> : format d'affichage des valeurs numériques dans les vues et rapports Analytics</li> <li>◦ <b>champs DateHeure</b> : format physique des valeurs DateHeure dans les données source (ordre des caractères date et heure, séparateurs, etc.)</li> </ul> <p><b>Remarque</b></p> <p>Pour les champs DateHeure, le <i>format</i> doit correspondre exactement au format physique dans les données source. Par exemple, si les données source sont 12/31/2014, vous devez saisir la date au format "MM/JJ/AAAA".</p> <p>Le <i>format</i> doit être placé entre guillemets.</p>				
AS <i>nom_affichage</i>	Le nom d'affichage (titre de la colonne) pour le champ dans la vue dans la nouvelle table				

Nom	Description
	<p>Analytics.</p> <p>Indiquez le <i>nom_affichage</i> sous forme de chaîne entre guillemets. Utilisez un point-virgule (;) entre les mots si vous souhaitez insérer un saut de ligne dans le titre de la colonne.</p> <p>AS est requis lors de la définition de FIELD. Pour que le nom d'affichage soit identique au champ nom, saisissez une valeur <i>nom_affichage</i> vide à l'aide de la syntaxe suivante : AS "". Vérifiez qu'il n'y a aucune espace entre les deux guillemets doubles.</p>
IGNORE <i>num_champ</i> optionnel	<p>Exclut un champ du format de table.</p> <p><i>num_champ</i> indique la position du champ dans les données source. Par exemple, IGNORE 5 exclut le cinquième champ des données de la source du format de la table Analytics.</p>

## Exemples

### Importer un fichier XBRL dans une table Analytics

Vous importez des données à partir du contexte **Actuel\_Le** d'un fichier XBRL vers une table Analytics appelée **Financier** :

```
IMPORT XBRL TO Financier "Financier.fil" FROM "RapportFinancierXBRL.xml" CONTEXT "Actuel_Le" FIELD "Elément" C AT 1 DEC 0 WID 57 PIC "" AS "" FIELD "Valeur" X AT 58 DEC 0 WID 7 PIC "" AS "" IGNORE 1 IGNORE 3
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

### Identificateurs des types de données de champ

Le tableau ci-dessous répertorie les lettres que vous devez utiliser lorsque vous indiquez *type* pour FIELD. Chaque lettre correspond à un type de données.

Par exemple, si vous définissez un champ Nom, qui utilise un type de données caractère, vous devez indiquer « C » : FIELD "Nom" C.

### Remarque

Lorsque vous utilisez l'**Assistant de définition de données** pour définir une table qui comprend des champs EBCDIC, Unicode ou ASCII, ces derniers se voient automatiquement attribuer la lettre « C » (pour le type CHARACTER).

Lorsque vous saisissez manuellement une instruction IMPORT ou que vous modifiez une instruction IMPORT existante, vous pouvez remplacer les lettres « E » ou « U » par des champs EBCDIC ou Unicode.

Lettre	Type de données
A	ACL
B	BINARY
C	CHARACTER
D	DATETIME
E	EBCDIC
F	FLOAT
G	ACCPAC
I	IBMFLOAT
K	UNSIGNED
L	LOGICAL
N	PRINT
P	PACKED
Q	BASIC
R	MICRO
S	CUSTOM
T	PCASCII
U	UNICODE
V	VAXFLOAT
X	NUMERIC



Lettre	Type de données
Y	UNISYS
Z	ZONED

# Commande IMPORT XML

Crée une table Analytics en définissant et en important un fichier XML.

## Syntaxe

```
IMPORT XML TO table nom_fichier_importation FROM nom_fichier_source [syntaxe_champ] <...n>
```

```
syntaxe_champ ::=
FIELD nom type AT position_départ DEC valeur WID octets PIC format AS nom_affichage RULE
expression_xpath
```

## Paramètres

Nom	Description
TO <i>table</i>	<p>Nom de la table Analytics dans laquelle les données doivent être importées.</p> <p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alpha-numériques. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>
<i>nom_fichier_importation</i>	<p>Nom du fichier de données Analytics à créer.</p> <p>Indiquez <i>nom_fichier_importation</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple, "Factures.FIL".</p> <p>Par défaut, le fichier de données (.FIL) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>○ "C:\données\Factures.FIL"</li> <li>○ "données\Factures.FIL"</li> </ul>
FROM <i>nom_fichier_source</i>	<p>Nom du fichier de données source. <i>nom_fichier_source</i> doit être une chaîne entre guillemets.</p> <p>Si le fichier de données source n'est pas enregistré dans le même répertoire que le projet Analytics, vous devez utiliser un chemin d'accès absolu ou un chemin d'accès relatif pour indiquer l'emplacement du fichier :</p> <ul style="list-style-type: none"> <li>○ "C:\données\<i>nom_fichier_source</i>"</li> <li>○ "données\<i>nom_fichier_source</i>"</li> </ul>

Nom	Description				
FIELD <i>nom type</i>	<p>Les champs individuels à importer à partir du fichier de données source, y compris le nom et le type de données du champ. Pour empêcher l'importation d'un champ, ne l'indiquez pas.</p> <p>Pour plus d'informations sur le <i>type</i>, consultez la rubrique "Identificateurs des types de données de champ" Page suivante.</p>				
AT <i>position_départ</i>	<p>La position d'octet de départ du champ dans le fichier de données Analytics.</p> <p><b>Remarque</b></p> <table border="1"> <tbody> <tr> <td>Analytics non Unicode</td> <td>1 octet = 1 caractère</td> </tr> <tr> <td>Analytics Unicode</td> <td>2 octets = 1 caractère</td> </tr> </tbody> </table> <p>Dans Analytics Unicode, normalement vous devez indiquer une position d'octet de départ avec un nombre impair. Indiquer une position de départ avec un chiffre pair peut entraîner l'affichage incorrect des caractères.</p>	Analytics non Unicode	1 octet = 1 caractère	Analytics Unicode	2 octets = 1 caractère
Analytics non Unicode	1 octet = 1 caractère				
Analytics Unicode	2 octets = 1 caractère				
DEC <i>valeur</i>	Nombre de décimales contenues par les champs numériques				
WID <i>octets</i>	<p>La longueur du champ en octets dans le format de la table Analytics.</p> <p><b>Remarque</b></p> <table border="1"> <tbody> <tr> <td>Analytics non Unicode</td> <td>1 octet = 1 caractère</td> </tr> <tr> <td>Analytics Unicode</td> <td>2 octets = 1 caractère</td> </tr> </tbody> </table> <p>Dans Analytics Unicode, spécifiez uniquement un nombre d'octets pair. Indiquer un nombre d'octets impair peut entraîner l'affichage incorrect des caractères.</p>	Analytics non Unicode	1 octet = 1 caractère	Analytics Unicode	2 octets = 1 caractère
Analytics non Unicode	1 octet = 1 caractère				
Analytics Unicode	2 octets = 1 caractère				
PIC <i>format</i>	<p><b>Remarque</b></p> <p>S'applique aux champs numériques ou DateHeure.</p> <ul style="list-style-type: none"> <li>◦ <b>champs numériques</b> : format d'affichage des valeurs numériques dans les vues et rapports Analytics</li> <li>◦ <b>champs DateHeure</b> : format physique des valeurs DateHeure dans les données source (ordre des caractères date et heure, séparateurs, etc.)</li> </ul> <p><b>Remarque</b></p> <p>Pour les champs DateHeure, le <i>format</i> doit correspondre exactement au format physique dans les données source. Par exemple, si les données source sont 12/31/2014, vous devez saisir la date au format "MM/JJ/AAAA".</p> <p>Le <i>format</i> doit être placé entre guillemets.</p>				
AS <i>nom_affichage</i>	<p>Le nom d'affichage (titre de la colonne) pour le champ dans la vue dans la nouvelle table Analytics.</p> <p>Indiquez le <i>nom_affichage</i> sous forme de chaîne entre guillemets. Utilisez un point-vir-</p>				

Nom	Description
	<p>gule (;) entre les mots si vous souhaitez insérer un saut de ligne dans le titre de la colonne.</p> <p>AS est requis lors de la définition de FIELD. Pour que le nom d'affichage soit identique au champ nom, saisissez une valeur <i>nom_affichage</i> vide à l'aide de la syntaxe suivante : AS "". Vérifiez qu'il n'y a aucune espace entre les deux guillemets doubles.</p>
RULE <i>expression_xpath</i>	<p>L'expression XPath utilisée pour sélectionner le contenu de champs dans le fichier XML.</p> <p>XPath est un mode d'accès standard aux données de fichiers XML. Par exemple, acct/-titre/text() récupère le texte de la balise &lt;title&gt; dans le fichier XML.</p>

## Exemples

### Importer des données d'un fichier XML vers une table Analytics

Vous importez des données à partir d'un fichier XML vers une table Analytics nommée **Employés** :

```
IMPORT XML TO Employés "Employés.fil" FROM "emp.XML" FIELD "NoEmp" C AT 1 DEC 0 WID 6
PIC "" AS "" RULE "/RECORDS/RECORD/NoEmp/text()" FIELD "Premier" C AT 7 DEC 0 WID 13
PIC "" AS "" RULE "/RECORDS/RECORD/Premier/text()" FIELD "Dernier" C AT 20 DEC 0 WID 20
PIC "" AS "" RULE "/RECORDS/RECORD/Dernier/text()" FIELD "DateEmb" D AT 40 DEC 0 WID
10 PIC "AAAA-MM-JJ" AS "" RULE "/RECORDS/RECORD/DateEmb/text()" FIELD "Salaire" N AT
50 DEC 2 WID 8 PIC "" AS "" RULE "/RECORDS/RECORD/Salaire/text()"
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Identificateurs des types de données de champ

Le tableau ci-dessous répertorie les lettres que vous devez utiliser lorsque vous indiquez *type* pour FIELD. Chaque lettre correspond à un type de données.

Par exemple, si vous définissez un champ Nom, qui utilise un type de données caractère, vous devez indiquer « C » : FIELD "Nom" C.

### Remarque

Lorsque vous utilisez l'**Assistant de définition de données** pour définir une table qui comprend des champs EBCDIC, Unicode ou ASCII, ces derniers se voient automatiquement attribuer la lettre « C » (pour le type CHARACTER).

Lorsque vous saisissez manuellement une instruction IMPORT ou que vous modifiez une instruction IMPORT existante, vous pouvez remplacer les lettres « E » ou « U » par des champs EBCDIC ou Unicode.

Lettre	Type de données
A	ACL
B	BINARY
C	CHARACTER
D	DATETIME
E	EBCDIC
F	FLOAT
G	ACCPAC
I	IBMFLOAT
K	UNSIGNED
L	LOGICAL
N	PRINT
P	PACKED
Q	BASIC
R	MICRO
S	CUSTOM
T	PCASCII
U	UNICODE
V	VAXFLOAT
X	NUMERIC

Lettre	Type de données
Y	UNISYS
Z	ZONED

# Commande INDEX

Crée un index pour une table Analytics permettant d'accéder aux enregistrements par ordre séquentiel plutôt que physique.

## Syntaxe

```
INDEX <ON> {champ_clé <D> <...n>|ALL} TO nom_fichier <IF test> <WHILE test> <FIRST
plage|NEXT plage> <OPEN> <ISOLOCALE code_paramètres_régionaux>
```

## Paramètres

Nom	Description
ON <i>champ_clé</i> D <...n>   ALL	<p>Le ou les champs clés ou bien l'expression à utiliser pour l'indexation.</p> <p>Vous pouvez indexer en fonction de tout type de champ, y compris les champs calculés et les expressions ad hoc, et ce quel que soit le type de données.</p> <ul style="list-style-type: none"> <li>◦ <b>champ_clé</b> : utilisez le(s) champ(s) spécifié(s) Si vous indexez en fonction de plusieurs champs, vous créez une indexation imbriquée dans la table. L'ordre de l'imbrication suit l'ordre dans lequel vous indiquez les champs. Incluez D pour indexer le champ clé dans l'ordre décroissant. L'ordre d'indexation par défaut est croissant.</li> <li>◦ <b>ALL</b> : utilisez tous les champs dans la table Si vous indexez en fonction de tous les champs dans une table, vous créez une indexation imbriquée. L'ordre de l'imbrication suit l'ordre d'apparition des champs dans le format de table. Un ordre d'indexation croissant est la seule option pour ALL.</li> </ul>
TO <i>nom_fichier</i>	<p>Le nom de l'index et du fichier d'index associé. Le fichier d'index est créé avec une extension .INX.</p> <p><b>Remarque</b> Dans l'interface utilisateur Analytics, la longueur des noms des index est limitée à 64 caractères alphanumériques. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>
IF <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.</p>

Nom	Description
	<p><b>Remarque</b></p> <p>La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p>
WHILE <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.</p> <p><b>Remarque</b></p> <p>Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.</p>
FIRST <i>plage</i>   NEXT <i>plage</i> optionnel	<p>Le nombre d'enregistrements à traiter :</p> <ul style="list-style-type: none"> <li>◦ <b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li>◦ <b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> </ul> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter.</p> <p>Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p>
OPEN optionnel	<p>Ouvrez la table et appliquez l'index à la table.</p>
ISOLocale <i>code_paramètres_régionaux</i> optionnel	<p><b>Remarque</b></p> <p>Applicable dans l'édition Unicode d'Analytics uniquement.</p> <p>Paramètre système local au format <i>langue_pays</i>. Par exemple, pour utiliser le français canadien, saisissez fr_ca.</p> <p>Utilisez les codes suivants :</p> <ul style="list-style-type: none"> <li>◦ <b>langue</b> : code linguistique de la norme ISO 639</li> <li>◦ <b>pays</b> : code pays de la norme ISO 3166</li> </ul> <p>Si vous n'indiquez pas de code pays, c'est le pays par défaut pour la langue qui est utilisé.</p> <p>Si vous n'utilisez pas ISOLocale, ce sont les paramètres système régionaux par défaut qui sont utilisés.</p>

## Exemples

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

### Créer un index et ouvrir la table



Dans la table Fournisseur, vous créez un index sur le champ **Ville fournisseur** et ouvrez la table :

```
OPEN Fournisseur
INDEX ON Ville_fournisseur to "IndexVille" OPEN
```

## Créer un index et l'appliquer sur une table

Dans la table Fournisseur, vous créez un index sur le champ **Ville fournisseur**. Ultérieurement, vous appliquez l'index à la table :

```
OPEN Fournisseur
INDEX ON Ville_fournisseur to "IndexVille"
,
,
,
SET INDEX TO "IndexVille"
```

# Commande JOIN

Combine des champs de deux tables Analytics dans une nouvelle table Analytics unique.

## Remarque

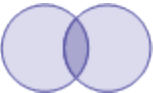
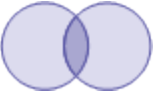
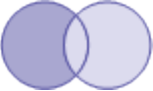
Pour utiliser les correspondances approximatives afin de joindre des tables, consultez la rubrique "Commande FUZZYJOIN" Page 222.

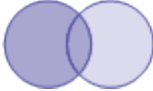
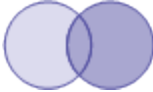
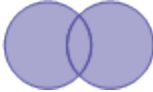
## Syntaxe

```
JOIN {PKEY champs_clés_principaux|PKEY ALL} {FIELDS champs_principaux|FIELDS ALL}
{SKEY champs_clés_secondaires|SKEY ALL} <WITH champs_secondaires|WITH ALL> {aucun_
mot_clé|MANY|UNMATCHED|PRIMARY|SECONDARY|PRIMARY SECONDARY} <IF test> TO
nom_table <LOCAL> <OPEN> <WHILE test> <FIRST plage|NEXT plage> <APPEND>
<PRESORT> <SECSORT> <ISOLocale code_paramètres_régionaux>
```

## Paramètres

Nom	Description
PKEY <i>champs_clés_principaux</i>   PKEY ALL	<p>Le ou les champs clés, ou expression clé, dans la table principale.</p> <ul style="list-style-type: none"> <li>◦ <b><i>champs_clés_principaux</i></b> : utilisez le(s) champ(s) spécifié(s)</li> <li>◦ <b>ALL</b> : utilisez tous les champs dans la table</li> </ul>
FIELDS <i>champs_principaux</i>   FIELDS ALL	<p>Les champs ou expressions de la table principale à inclure dans la table de sortie jointe.</p> <ul style="list-style-type: none"> <li>◦ <b><i>champs_principaux</i></b> : incluez le(s) champ(s) spécifié(s)</li> <li>◦ <b>ALL</b> : incluez tous les champs de la table</li> </ul> <p><b>Remarque</b></p> <p>Vous devez indiquer de manière explicite le(s) champ(s) clé(s) principal (aux) si vous voulez les intégrer dans la table jointe. Le fait d'indiquer ALL permet aussi de les inclure.</p>
SKEY <i>champs_clés_secondaires</i>   SKEY ALL	<p>Le ou les champs clés, ou l'expression clé, dans la table secondaire.</p> <ul style="list-style-type: none"> <li>◦ <b><i>champs_clés_secondaires</i></b> : utilisez le(s) champ(s) spécifié(s)</li> <li>◦ <b>ALL</b> : utilisez tous les champs dans la table</li> </ul>
WITH <i>champs_secondaires</i>   WITH ALL optionnel	<p>Les champs ou expressions de la table secondaire à inclure dans la table de sortie jointe.</p> <ul style="list-style-type: none"> <li>◦ <b><i>champs_secondaires</i></b> : incluez le(s) champ(s) spécifié(s)</li> <li>◦ <b>ALL</b> : incluez tous les champs de la table</li> </ul>

Nom	Description												
	<p><b>Remarque</b></p> <p>Vous devez indiquer de manière explicite le(s) champ(s) clé(s) secondaire(s) si vous voulez les intégrer dans la table jointe. Le fait d'indiquer ALL permet aussi de les inclure.</p> <p>Vous ne pouvez pas indiquer WITH si vous utilisez le type de jointure UNMATCHED.</p>												
<p><i>aucun_mot_clé</i>   MANY   UNMATCHED   PRIMARY   SECONDARY   PRIMARY SECONDARY</p>	<p>Type de jointure à effectuer.</p> <p><b><i>aucun_mot_clé</i> (omettez tous les mots-clés du type de jointure)</b></p>  <table border="1" data-bbox="483 737 1425 972"> <tr> <td data-bbox="483 737 951 835">La table de sortie jointe contient :</td> <td data-bbox="951 737 1425 835">Option correspondante dans la boîte de dialogue Joindre</td> </tr> <tr> <td data-bbox="483 835 951 972"> <ul style="list-style-type: none"> <li>tous les enregistrements primaires correspondants et le premier enregistrement secondaire correspondant</li> </ul> </td> <td data-bbox="951 835 1425 972"> <p><b>Primaires et secondaires correspondants</b></p> <p><b>(1re correspondance secondaire)</b></p> </td> </tr> </table> <p><b>MANY</b></p>  <table border="1" data-bbox="483 1167 1425 1436"> <tr> <td data-bbox="483 1167 951 1266">La table de sortie jointe contient :</td> <td data-bbox="951 1167 1425 1266">Option correspondante dans la boîte de dialogue Joindre</td> </tr> <tr> <td data-bbox="483 1266 951 1436"> <ul style="list-style-type: none"> <li>tous les enregistrements primaires et secondaires correspondants</li> <li>un enregistrement pour chaque correspondance entre la table principale et les tables secondaires</li> </ul> </td> <td data-bbox="951 1266 1425 1436"> <p><b>Primaires et secondaires correspondants</b></p> <p><b>(toutes les correspondances secondaires)</b></p> </td> </tr> </table> <p><b>UNMATCHED</b></p>  <table border="1" data-bbox="483 1631 1425 1816"> <tr> <td data-bbox="483 1631 951 1730">La table de sortie jointe contient :</td> <td data-bbox="951 1631 1425 1730">Option correspondante dans la boîte de dialogue Joindre</td> </tr> <tr> <td data-bbox="483 1730 951 1816"> <ul style="list-style-type: none"> <li>enregistrements primaires non correspondants</li> </ul> </td> <td data-bbox="951 1730 1425 1816"> <p><b>Primaires non correspondants</b></p> </td> </tr> </table>	La table de sortie jointe contient :	Option correspondante dans la boîte de dialogue Joindre	<ul style="list-style-type: none"> <li>tous les enregistrements primaires correspondants et le premier enregistrement secondaire correspondant</li> </ul>	<p><b>Primaires et secondaires correspondants</b></p> <p><b>(1re correspondance secondaire)</b></p>	La table de sortie jointe contient :	Option correspondante dans la boîte de dialogue Joindre	<ul style="list-style-type: none"> <li>tous les enregistrements primaires et secondaires correspondants</li> <li>un enregistrement pour chaque correspondance entre la table principale et les tables secondaires</li> </ul>	<p><b>Primaires et secondaires correspondants</b></p> <p><b>(toutes les correspondances secondaires)</b></p>	La table de sortie jointe contient :	Option correspondante dans la boîte de dialogue Joindre	<ul style="list-style-type: none"> <li>enregistrements primaires non correspondants</li> </ul>	<p><b>Primaires non correspondants</b></p>
La table de sortie jointe contient :	Option correspondante dans la boîte de dialogue Joindre												
<ul style="list-style-type: none"> <li>tous les enregistrements primaires correspondants et le premier enregistrement secondaire correspondant</li> </ul>	<p><b>Primaires et secondaires correspondants</b></p> <p><b>(1re correspondance secondaire)</b></p>												
La table de sortie jointe contient :	Option correspondante dans la boîte de dialogue Joindre												
<ul style="list-style-type: none"> <li>tous les enregistrements primaires et secondaires correspondants</li> <li>un enregistrement pour chaque correspondance entre la table principale et les tables secondaires</li> </ul>	<p><b>Primaires et secondaires correspondants</b></p> <p><b>(toutes les correspondances secondaires)</b></p>												
La table de sortie jointe contient :	Option correspondante dans la boîte de dialogue Joindre												
<ul style="list-style-type: none"> <li>enregistrements primaires non correspondants</li> </ul>	<p><b>Primaires non correspondants</b></p>												

Nom	Description												
	<p><b>PRIMARY</b></p>  <table border="1" data-bbox="500 491 1446 732"> <tr> <td data-bbox="500 491 971 585">La table de sortie jointe contient :</td> <td data-bbox="971 491 1446 585">Option correspondante dans la boîte de dialogue Joindre</td> </tr> <tr> <td data-bbox="500 585 971 732"> <ul style="list-style-type: none"> <li>tous les enregistrements primaires (correspondants et non correspondants) et le premier enregistrement secondaire correspondant</li> </ul> </td> <td data-bbox="971 585 1446 732"><b>Tous les primaires et tous les secondaires correspondants</b></td> </tr> </table> <p><b>Remarque</b> Le mot-clé BOTH revient à spécifier PRIMARY.</p> <p><b>SECONDARY</b></p>  <table border="1" data-bbox="500 1016 1446 1352"> <tr> <td data-bbox="500 1016 971 1110">La table de sortie jointe contient :</td> <td data-bbox="971 1016 1446 1110">Option correspondante dans la boîte de dialogue Joindre</td> </tr> <tr> <td data-bbox="500 1110 971 1352"> <ul style="list-style-type: none"> <li>tous les enregistrements secondaires (correspondants et non correspondants) et tous les enregistrements primaires correspondants</li> </ul> <p>Seule la première instance des correspondances secondaires en doublon est jointe à un enregistrement primaire.</p> </td> <td data-bbox="971 1110 1446 1352"><b>Tous les secondaires et tous les primaires correspondants</b></td> </tr> </table> <p><b>PRIMARY SECONDARY</b></p>  <table border="1" data-bbox="500 1551 1446 1814"> <tr> <td data-bbox="500 1551 971 1646">La table de sortie jointe contient :</td> <td data-bbox="971 1551 1446 1646">Option correspondante dans la boîte de dialogue Joindre</td> </tr> <tr> <td data-bbox="500 1646 971 1814"> <ul style="list-style-type: none"> <li>tous les enregistrements primaires et secondaires, correspondants et non correspondants</li> </ul> <p>Seule la première instance des cor-</p> </td> <td data-bbox="971 1646 1446 1814"><b>Tous les primaires et secondaires</b></td> </tr> </table>	La table de sortie jointe contient :	Option correspondante dans la boîte de dialogue Joindre	<ul style="list-style-type: none"> <li>tous les enregistrements primaires (correspondants et non correspondants) et le premier enregistrement secondaire correspondant</li> </ul>	<b>Tous les primaires et tous les secondaires correspondants</b>	La table de sortie jointe contient :	Option correspondante dans la boîte de dialogue Joindre	<ul style="list-style-type: none"> <li>tous les enregistrements secondaires (correspondants et non correspondants) et tous les enregistrements primaires correspondants</li> </ul> <p>Seule la première instance des correspondances secondaires en doublon est jointe à un enregistrement primaire.</p>	<b>Tous les secondaires et tous les primaires correspondants</b>	La table de sortie jointe contient :	Option correspondante dans la boîte de dialogue Joindre	<ul style="list-style-type: none"> <li>tous les enregistrements primaires et secondaires, correspondants et non correspondants</li> </ul> <p>Seule la première instance des cor-</p>	<b>Tous les primaires et secondaires</b>
La table de sortie jointe contient :	Option correspondante dans la boîte de dialogue Joindre												
<ul style="list-style-type: none"> <li>tous les enregistrements primaires (correspondants et non correspondants) et le premier enregistrement secondaire correspondant</li> </ul>	<b>Tous les primaires et tous les secondaires correspondants</b>												
La table de sortie jointe contient :	Option correspondante dans la boîte de dialogue Joindre												
<ul style="list-style-type: none"> <li>tous les enregistrements secondaires (correspondants et non correspondants) et tous les enregistrements primaires correspondants</li> </ul> <p>Seule la première instance des correspondances secondaires en doublon est jointe à un enregistrement primaire.</p>	<b>Tous les secondaires et tous les primaires correspondants</b>												
La table de sortie jointe contient :	Option correspondante dans la boîte de dialogue Joindre												
<ul style="list-style-type: none"> <li>tous les enregistrements primaires et secondaires, correspondants et non correspondants</li> </ul> <p>Seule la première instance des cor-</p>	<b>Tous les primaires et secondaires</b>												

Nom	Description				
	<table border="1" data-bbox="483 268 1425 457"> <tr> <td data-bbox="483 268 951 363">La table de sortie jointe contient :</td> <td data-bbox="951 268 1425 363">Option correspondante dans la boîte de dialogue Joindre</td> </tr> <tr> <td data-bbox="483 363 951 457">respondances secondaires en doublon est jointe à un enregistrement primaire.</td> <td data-bbox="951 363 1425 457"></td> </tr> </table>	La table de sortie jointe contient :	Option correspondante dans la boîte de dialogue Joindre	respondances secondaires en doublon est jointe à un enregistrement primaire.	
La table de sortie jointe contient :	Option correspondante dans la boîte de dialogue Joindre				
respondances secondaires en doublon est jointe à un enregistrement primaire.					
IF test optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.</p> <p><b>Remarque</b></p> <p>La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p> <p><b>Remarque</b></p> <p>Pour la plupart des types de jointure, la condition IF ne s'applique que dans la table principale.</p> <p>La seule exception est une jointure ACL de type plusieurs-à-plusieurs, dans laquelle la condition IF peut aussi faire référence à la table secondaire.</p>				
TO <i>nom_table</i>	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li>◦ <b>nom_table</b> : enregistre les résultats dans une table Analytics</li> </ul> <p>Indiquez <i>nom_table</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple : TO "Sortie.FIL"</p> <p>Par défaut, le fichier de données de table (.FIL) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.FIL"</li> <li>• TO "Résultats\Sortie.FIL"</li> </ul> <p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alpha-numériques, sans l'extension .FIL. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>				
LOCAL optionnel	<p>Enregistre le fichier de sortie au même emplacement que le projet Analytics.</p> <p><b>Remarque</b></p> <p>Applicable uniquement lorsque vous exécutez la commande sur une table de serveur avec un fichier de sortie qui est une table Analytics.</p>				
OPEN optionnel	<p>Ouvre la table créée par la commande après l'exécution de la commande. Valide uniquement lorsque la commande crée une table de sortie.</p>				

Nom	Description
WHILE <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.</p> <p><b>Remarque</b></p> <p>Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.</p>
FIRST <i>plage</i>   NEXT <i>plage</i> optionnel	<p>Le nombre d'enregistrements à traiter :</p> <ul style="list-style-type: none"> <li>○ <b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li>○ <b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> </ul> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter.</p> <p>Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p>
APPEND optionnel	<p>Ajoute la sortie de commande à la fin d'un fichier existant au lieu de remplacer ce fichier.</p> <p><b>Remarque</b></p> <p>Vous devez vous assurer que la structure de la sortie de la commande et du fichier existant est identique :</p> <ul style="list-style-type: none"> <li>• les mêmes champs</li> <li>• le même ordre des champs</li> <li>• les champs correspondants ont la même longueur</li> <li>• les champs correspondants ont le même type de données</li> </ul> <p>Analytics ajoute la sortie à un fichier existant sans tenir compte de sa structure, ce qui peut désordonner les données si la structure de la sortie et du fichier existant ne correspond pas.</p>
PRESORT optionnel	<p>Trie la table principale par le champ clé primaire avant d'exécuter la commande.</p> <p><b>Remarque</b></p> <p>Vous ne pouvez pas utiliser PRESORT dans la commande GROUP.</p> <p><b>Indexation au lieu du tri</b></p> <p>La table primaire peut être indexée plutôt que triée. Avec les grandes tables, indexer plutôt que trier peut permettre de diminuer la durée requise pour joindre les tables.</p> <p>Si vous joignez deux tables à l'aide d'un champ clé commun indexé, ignorez PRESORT et SECSORT.</p>
SECSORT optionnel	<p>Trie la table secondaire par le champ clé secondaire avant d'exécuter la commande.</p> <p><b>Remarque</b></p> <p>Vous ne pouvez pas utiliser SECSORT dans la commande GROUP.</p> <p><b>Indexation au lieu du tri</b></p> <p>La table secondaire peut être indexée plutôt que triée. Avec les grandes tables, indexer</p>

Nom	Description
	<p>plutôt que trier peut permettre de diminuer la durée requise pour joindre les tables.</p> <p>Si vous joignez deux tables à l'aide d'un champ clé commun indexé, ignorez PRESORT et SECSORT.</p>
ISOLOCALE <i>code_paramètres_régionaux</i> optionnel	<p><b>Remarque</b> Applicable dans l'édition Unicode d'Analytics uniquement.</p> <p>Paramètre système local au format <i>langue_pays</i>. Par exemple, pour utiliser le français canadien, saisissez <i>fr_ca</i>.</p> <p>Utilisez les codes suivants :</p> <ul style="list-style-type: none"> <li>○ <b>langue</b> : code linguistique de la norme ISO 639</li> <li>○ <b>pays</b> : code pays de la norme ISO 3166</li> </ul> <p>Si vous n'indiquez pas de code pays, c'est le pays par défaut pour la langue qui est utilisé.</p> <p>Si vous n'utilisez pas ISOLOCALE, ce sont les paramètres système régionaux par défaut qui sont utilisés.</p>

## Exemples

### Joindre deux tables pour détecter les employés susceptibles d'être aussi des fournisseurs

L'exemple ci-dessous joint les tables Empmast et Fournisseur en utilisant l'adresse comme champ clé commun (champs Adresse et Rue\_Fournisseur).

La commande JOIN crée une nouvelle table avec les enregistrements primaires et secondaires correspondants, ce qui génère la liste des employés et des fournisseurs ayant la même adresse.

```
OPEN Empmast PRIMARY
OPEN Fournisseur SECONDARY
JOIN PKEY Adresse FIELDS Empno Premier Dernier Adresse SKEY Rue_Fournisseur WITH No_
Fournisseur Nom_Fournisseur Rue_Fournisseur "Correspondance_employé_fournisseur" OPEN
PRESORT SECSORT
```

Cette version de la commande JOIN inclut tous les champs des tables principale et secondaires dans la table de sortie jointe.

```
OPEN Empmast PRIMARY
OPEN Fournisseur SECONDARY
JOIN PKEY Adresse FIELDS ALL SKEY Rue_Fournisseur WITH ALL TO "Correspondance_
employé_fournisseur" OPEN PRESORT SECSORT
```

### Joindre deux tables pour découvrir des enregistrements de créances clients

## sans client qui correspond

Dans l'exemple ci-dessous, les tables Ar et Client sont jointes en utilisant le numéro de client (NumClient) comme champ clé commun.

La commande JOIN utilise le type de jointure UNMATCHED pour créer une table d'enregistrements primaires non correspondants, qui renvoie une liste d'enregistrements Ar qui ne sont associés à aucun enregistrement Client.

```
OPEN Ar PRIMARY
OPEN Client SECONDARY
JOIN PKEY NumClient FIELDS NumClient Echéance Montant SKEY NumClient UNMATCHED TO
"ClientIntrouvable.fil" OPEN PRESORT SECSORT
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).



# Commande LIST

Sort les données d'un ou plusieurs champs d'une table Analytics dans un affichage mis en forme en colonnes.

## Syntaxe

```
LIST {FIELDS nom_champ <AS nom_affichage> <...n>|FIELDS ALL} <LINE numéro liste_champs>
<TO {SCREEN|nom_fichier|PRINT}> <UNFORMATTED> <IF test> <WHILE test> <FIRST
page|NEXT page> <HEADER texte_en-tête> <FOOTER texte_pied_page> <SKIP lignes> <EOF>
<APPEND>
```

## Paramètres

Nom	Description
FIELDS <i>nom_champ</i> <...n>   FIELDS ALL	Les champs à inclure dans la sortie : <ul style="list-style-type: none"> <li>◦ <b>FIELDS <i>nom_champ</i></b> : utilisez les champs spécifiés</li> <li>◦ <b>FIELDS ALL</b> : utilisez tous les champs dans la table</li> </ul>
AS <i>nom_affichage</i> optionnel	Utilisé uniquement dans le cas d'une énumération des données à l'aide de FIELDS <i>nom_champ</i> .  Nom d'affichage (titre de la colonne) pour le champ dans la sortie. Si vous souhaitez que le nom d'affichage soit identique au nom du champ ou à un nom d'affichage existant dans la table source, n'utilisez pas AS.  Indiquez le <i>nom_affichage</i> sous forme de chaîne entre guillemets. Utilisez un point-virgule (;) entre les mots si vous souhaitez insérer un saut de ligne dans le titre de la colonne.
LINE <i>numéro liste_champs</i> optionnel	Plusieurs lignes sont utilisées dans la sortie pour chaque enregistrement : <ul style="list-style-type: none"> <li>◦ <b>numéro</b> : numéro de ligne ; doit être compris entre 2 et 60 inclus</li> <li>◦ <b>liste_champs</b> : champs à inclure dans cette ligne</li> </ul>
TO SCREEN   <i>nom_fichier</i>   PRINT optionnel	Emplacement vers lequel envoyer les résultats de la commande : <ul style="list-style-type: none"> <li>◦ <b>SCREEN</b> : affiche les résultats dans la zone d'affichage d'Analytics.</li> <li>◦ <b><i>nom_fichier</i></b> : enregistre les résultats dans un fichier</li> </ul> <p>Indiquez <i>nom_fichier</i> sous forme de chaîne entre guillemets avec l'extension de fichier appropriée. Par exemple : TO "Sortie.TXT"</p> <p>Par défaut, le fichier est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier dans un autre dossier existant :</p>

Nom	Description
	<ul style="list-style-type: none"> <li>• TO "C:\Sortie.TXT"</li> <li>• TO "Résultats\Sortie.TXT"</li> <li>◦ <b>PRINT</b> : envoie les résultats vers l'imprimante par défaut</li> </ul>
UNFORMATTED optionnel	La sortie s'affiche sous forme de texte sans mise en forme. La sortie est identique à celle créée par la commande EXPORT ASCII. Les données non formatées peuvent être renvoyées dans un fichier pour un traitement ultérieur par d'autres logiciels.
IF <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.</p> <p><b>Remarque</b></p> <p>La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p>
WHILE <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.</p> <p><b>Remarque</b></p> <p>Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.</p>
FIRST <i>plage</i>   NEXT <i>plage</i> optionnel	<p>Le nombre d'enregistrements à traiter :</p> <ul style="list-style-type: none"> <li>◦ <b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li>◦ <b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> </ul> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter.</p> <p>Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p>
HEADER <i>texte_en-tête</i> optionnel	<p>Texte à insérer en haut de chaque page d'un rapport.</p> <p><i>texte_en-tête</i> doit être indiqué sous forme de chaîne entre guillemets. La valeur remplace la variable système Analytics HEADER.</p>
FOOTER <i>texte_pied_page</i> optionnel	<p>Texte à insérer en bas de chaque page d'un rapport.</p> <p><i>texte_pied_page</i> doit être indiqué sous forme de chaîne entre guillemets. La valeur remplace la variable système Analytics FOOTER.</p>
SKIP <i>lignes</i> optionnel	Insère le nombre défini de lignes vides entre chaque enregistrement de la liste. Par exemple, LIST ALL SKIP 1 génère une liste avec un espacement double (une ligne vide entre chaque enregistrement).
EOF optionnel	<p>Réexécutez la commande une fois la fin du fichier atteinte.</p> <p>Cela garantit que le dernier enregistrement de la table est traité quand il se trouve à l'intérieur d'une commande GROUP. Utilisez le paramètre de traitement de fin de fichier uniquement si tous les champs sont calculés en se référant à des enregistrements antérieurs.</p>

Nom	Description
APPEND optionnel	Ajoute la sortie de commande à la fin d'un fichier existant au lieu de remplacer ce fichier.

## Exemples

### Listage des exceptions et enregistrement dans un fichier texte

Vous utilisez LIST pour créer un rapport qui répertorie les exceptions identifiées dans une table d'inventaire. Le rapport est enregistré en tant que fichier texte :

```
LIST Numéro_produit Description Quantité Coût_unitaire Valeur IF Quantité < 0 OR Coût_unitaire < 0
HEADER "Valeurs négatives" TO "Exceptions.txt"
```

## Remarques

### Quand utiliser LIST

Utilisez LIST pour imprimer des données, afficher des données à l'écran ou les exporter vers un fichier texte.

### Mise en forme et totaux

À moins de préciser UNFORMATTED, les informations suivantes figurent automatiquement :

- numéros de page
- date
- heure
- identification de l'utilisateur
- en-têtes des colonnes

Les colonnes numériques sont également totalisées automatiquement.

# Commande LOCATE

Recherche le premier enregistrement qui correspond à la valeur ou à la condition spécifiée ou accède au numéro d'enregistrement spécifié.

## Syntaxe

```
LOCATE {IF test <WHILE test> <FIRST plage|NEXT plage>|RECORD num}
```

## Paramètres

Nom	Description
IF <i>test</i>	Valeur ou condition à rechercher. Les valeurs littérales de caractère doivent être entre guillemets et les valeurs DateHeure doivent être entre guillemets inversés.
WHILE <i>test</i> optionnel	Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.  <div style="border-left: 2px solid #0056b3; padding-left: 10px; margin-left: 20px;"> <p><b>Remarque</b></p> <p>Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.</p> </div>
FIRST <i>plage</i>   NEXT <i>plage</i> optionnel	Le nombre d'enregistrements à traiter : <ul style="list-style-type: none"> <li>○ <b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li>○ <b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> </ul> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter.</p> <p>Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p>
RECORD <i>num</i>	Numéro d'enregistrement à rechercher.

## Exemples

### Rechercher le premier enregistrement correspondant à une valeur spécifiée

Dans les exemples suivants, la commande LOCATE est utilisée pour rechercher la première occurrence d'une valeur spécifique dans une table :

```
LOCATE IF Nom_Fournisseur = "Équipement uni"
```

```
LOCATE IF Nom_Fournisseur = "Uni"
```

```
LOCATE IF Montant_Facture > 1000
```

```
LOCATE IF Date_facture = `20141231`
```

## Rechercher le premier enregistrement correspondant à une condition ou à une expression spécifiée

Dans les exemples suivants, la commande LOCATE est utilisée pour rechercher la première occurrence d'une condition ou d'une expression spécifique dans une table :

```
LOCATE IF Nom_Fournisseur = "Équipement uni" AND Montant_Facture > 1000 AND Date_facture > `20140930`
```

```
LOCATE IF Ville_fournisseur = v_ville
```

## Rechercher un enregistrement par numéro d'enregistrement

Dans l'exemple suivant, la commande LOCATE est utilisée pour accéder à un enregistrement spécifique dans une table :

```
LOCATE RECORD 50
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Fonctionnement

Utilisez la commande LOCATE pour accéder directement au premier enregistrement d'une table correspondant à la valeur ou à la condition spécifiée.

Si la valeur ou la condition spécifiée est trouvée, le premier enregistrement correspondant de la table est sélectionné. Si la valeur ou la condition spécifiée n'est pas trouvée, la table est placée sur le premier enregistrement.

LOCATE permet aussi de passer directement à un numéro d'enregistrement spécifique

## Comparaison de LOCATE avec FIND et SEEK

À la différence des commandes FIND et SEEK, la commande LOCATE ne se limite pas à la recherche dans une table indexée ou dans un champ caractère unique. À l'aide de la commande LOCATE, vous pouvez rechercher n'importe quel type de littéral ou d'expression utilisant n'importe quel type de données ou différents types de données.

Lorsqu'elle est utilisée pour effectuer une recherche dans une table non indexée, la commande LOCATE peut être beaucoup plus lente que FIND et SEEK, car elle doit traiter chaque enregistrement de la table de manière séquentielle. La durée de traitement requise dépend de la taille de la table, de l'emplacement d'un enregistrement correspondant et de la réduction ou non du périmètre de la recherche à l'aide de WHILE, FIRST ou NEXT.

## Rapprochement partiel pris en charge

Le rapprochement partiel est pris en charge pour les recherches de caractères. La valeur de recherche peut être contenue par une valeur plus longue dans le ou les champs dans lesquels la recherche est effectuée. Toutefois, les valeurs de recherche doivent apparaître au début des champs pour pouvoir constituer un rapprochement.

### Activation ou désactivation du rapprochement partiel

Vous pouvez activer ou désactiver le rapprochement partiel à l'aide de la commande SET ou d'un paramètre dans la boîte de dialogue **Options** :

Activer le rapprochement partiel	Désactiver le rapprochement partiel
<p><b>Spécifiez</b> : SET EXACT OFF</p> <p>ou</p> <p><b>Décochez</b> : <b>Comparaisons exactes de caractères</b> dans la boîte de dialogue <b>Options (Outils &gt; Options &gt; Table)</b></p> <p><b>Résultat</b> : La valeur de recherche peut être contenue par une valeur plus longue dans le ou les champs dans lesquels la recherche est effectuée. La valeur de recherche doit apparaître au début d'un champ pour constituer un rapprochement.</p>	<p><b>Spécifiez</b> : SET EXACT ON</p> <p>ou</p> <p><b>Sélectionnez</b> : <b>Comparaisons exactes de caractères</b> dans la boîte de dialogue <b>Options (Outils &gt; Options &gt; Table)</b></p> <p><b>Résultat</b> : La valeur de recherche doit correspondre exactement à une valeur dans un champ pour constituer un rapprochement.</p>

Pour plus d'informations sur SET EXACT, consultez la rubrique "Commande SET" Page 435.

Pour plus d'informations sur l'option **Comparaisons exactes de caractères**, consultez la rubrique [Onglet Table \(boîte de dialogue Options\)](#).

# Commande LOOP

Exécute une série de commandes ACLScript de manière répétée alors qu'une condition spécifiée est évaluée sur vrai.

## Remarque

La commande LOOP doit être insérée à l'intérieur de la commande GROUP.

## Syntaxe

```
LOOP WHILE test
  commande
  <...n>
END
```

## Paramètres

Nom	Description
WHILE <i>test</i>	Le test qui doit renvoyer la valeur « vrai » avant l'exécution des commandes dans la commande LOOP. Si le test renvoie « vrai », les commandes sont exécutées à répétition jusqu'à ce que le test renvoie « faux ».
<i>commande</i> <... <i>n</i> >	Une ou plusieurs commande à exécuter. Vous pouvez saisir plusieurs commandes au sein de la commande LOOP. Chaque commande doit commencer sur une nouvelle ligne.
END	Fin de la commande LOOP.

## Exemples

### Diviser un champ délimité par des virgules

Une table contient des données de facture et vous avez besoin d'isoler certaines informations pour des montants de facture par service. Une facture peut être reliée à plusieurs services et les codes des services sont enregistrés dans un format séparé par une virgule dans la table.

Pour extraire les montants de facture par service, vous :

1. Utilisez une commande GROUP pour traiter la table, enregistrement par enregistrement.
2. Calculez le nombre de services (n) associé à chaque enregistrement.

3. Utilisez la commande LOOP pour effectuer une itération sur l'enregistrement x fois afin d'extraire les données pour chaque service associé à l'enregistrement.

```
COMMENT utilisez GROUP pour compter les virgules dans chaque champ de code de service afin d'identifier le nombre de services associés à l'enregistrement exécutez "LOOP" sur chaque enregistrement pour chaque code dans le champ, en extrayant chaque code dans son propre enregistrement ENDGROUP v_department_count = OCCURS(Code_Département; ',') v_counter = 0 LOOP WHILE v_counter <= v_department_count v_dept = SPLIT(Code_Département; ',', (v_counter + 1)) EXTRACT FIELDS No_Facture, Montant_Facture, v_dept AS "Service" TO result1 v_counter = v_counter + 1 END END
```

## Remarques

### Astuce

Pour voir le tutoriel détaillé sur les commandes LOOP et GROUP, reportez-vous à la rubrique "Regroupement et exécution de boucles" Page 34.

## Quand utiliser LOOP

Des boucles sont souvent utilisées lorsqu'un enregistrement contient des répétitions de segments de données que vous souhaitez traiter.

## Fonctionnement

Chaque commande LOOP doit indiquer une condition WHILE à tester, et se terminer par une instruction END. Les commandes situées entre LOOP et END sont exécutées à répétition sur l'enregistrement actuel, tant que le test spécifié renvoie « vrai ».

Si le test renvoie initialement « faux », les commandes ne sont pas exécutées.

## Éviter des boucles infinies

Pour éviter de créer une boucle infinie, assurez-vous que le test puisse toujours se révéler faux. Vous pouvez également utiliser la commande SET LOOP pour éviter une boucle infinie.



# Commande MERGE

Combine les enregistrements provenant de deux tables Analytics triées avec une structure identique dans une nouvelle Analytics utilisant le même ordre de tri que les tables d'origine.

## Syntaxe

```
MERGE {ON champs_clés|PKEY champs_clés_principauxSKEY champs_clés_secondaires} <IF
test> TO nom_table<LOCAL> <OPEN> <WHILE test> <FIRST plage|NEXT plage> <APPEND>
<PRESORT> <ISOLOCALE code_paramètres_régionaux>
```

## Paramètres

Nom	Description
ON <i>champs_clés</i>   PKEY <i>champs_clés_principaux</i> SKEY <i>champs_clés_secondaires</i>	<p><b>Remarque</b></p> <p>Seuls les champs caractère ou les champs calculés caractère peuvent être utilisés comme champs clés dans MERGE.</p> <ul style="list-style-type: none"> <li>○ <b>ON <i>champs_clés</i></b> : le(s) champ(s) clé(s) à fusionner si les champs clés correspondants dans les tables principale et secondaire portent le même nom Si les champs correspondants ont des noms différents ou s'il s'agit d'expressions plutôt que de champs physiques réels, vous devez utiliser PKEY et SKEY.</li> <li>○ <b>PKEY <i>champs_clés_principaux</i></b> : le ou les champs clés, ou expression clé, dans la table principale</li> <li>○ <b>SKEY <i>champs_clés_principaux</i></b> : le ou les champs clés, ou l'expression clé, dans la table secondaire</li> </ul> <p><b>Conditions préalables au tri</b></p> <p>Les champs clés dans les tables primaire et secondaire doivent être triés dans l'ordre croissant. Si les champs clés de l'une ou des deux tables ne sont pas triés, ou s'ils sont triés dans l'ordre décroissant, la commande MERGE échoue.</p> <p>Vous pouvez utiliser PRESORT pour trier les champs clés de la table primaire. Si les champs clés de la table secondaire ne sont pas triés, vous devez d'abord les trier lors d'une opération de tri distincte avant d'effectuer la fusion.</p> <p><b>Indexation au lieu du tri</b></p> <p>Les tables principale et secondaire peuvent être indexées plutôt que triées. Avec les grandes tables, indexer plutôt que trier peut permettre de diminuer la durée requise pour fusionner les tables.</p>
IF <i>test</i> optionnel	Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.

Nom	Description
	<p><b>Remarque</b></p> <p>La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p>
TO <i>nom_table</i>	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li>◦ <b>nom_table</b> : enregistre les résultats dans une table Analytics</li> </ul> <p>Indiquez <i>nom_table</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple : TO "Sortie.FIL"</p> <p>Par défaut, le fichier de données de table (.FIL) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.FIL"</li> <li>• TO "Résultats\Sortie.FIL"</li> </ul> <p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alpha-numériques, sans l'extension .FIL. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>
LOCAL optionnel	<p>Enregistre le fichier de sortie au même emplacement que le projet Analytics.</p> <p><b>Remarque</b></p> <p>Applicable uniquement lorsque vous exécutez la commande sur une table de serveur avec un fichier de sortie qui est une table Analytics.</p>
OPEN optionnel	<p>Ouvre la table créée par la commande après l'exécution de la commande. Valide uniquement lorsque la commande crée une table de sortie.</p>
WHILE <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.</p> <p><b>Remarque</b></p> <p>Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.</p>
FIRST <i>plage</i>   NEXT <i>plage</i> optionnel	<p>Le nombre d'enregistrements à traiter :</p> <ul style="list-style-type: none"> <li>◦ <b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li>◦ <b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> </ul> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter.</p> <p>Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p>

Nom	Description
APPEND optionnel	<p>Ajoute la sortie de commande à la fin d'un fichier existant au lieu de remplacer ce fichier.</p> <p><b>Remarque</b></p> <p>Vous devez vous assurer que la structure de la sortie de la commande et du fichier existant est identique :</p> <ul style="list-style-type: none"> <li>• les mêmes champs</li> <li>• le même ordre des champs</li> <li>• les champs correspondants ont la même longueur</li> <li>• les champs correspondants ont le même type de données</li> </ul> <p>Analytics ajoute la sortie à un fichier existant sans tenir compte de sa structure, ce qui peut désordonner les données si la structure de la sortie et du fichier existant ne correspond pas.</p>
PRESORT optionnel	<p>Trie la table principale par le champ clé primaire avant d'exécuter la commande.</p> <p><b>Remarque</b></p> <p>Vous ne pouvez pas utiliser PRESORT dans la commande GROUP.</p> <p>Omettez PRESORT :</p> <ul style="list-style-type: none"> <li>◦ Si le champ clé primaire est déjà trié</li> <li>◦ Si vous fusionnez deux tables à l'aide d'un champ clé commun indexé</li> </ul>
ISOLOCALE <i>code_paramètres_régionaux</i> optionnel	<p><b>Remarque</b></p> <p>Applicable dans l'édition Unicode d'Analytics uniquement.</p> <p>Paramètre système local au format <i>langue_pays</i>. Par exemple, pour utiliser le français canadien, saisissez <i>fr_ca</i>.</p> <p>Utilisez les codes suivants :</p> <ul style="list-style-type: none"> <li>◦ <b>langue</b> : code linguistique de la norme ISO 639</li> <li>◦ <b>pays</b> : code pays de la norme ISO 3166</li> </ul> <p>Si vous n'indiquez pas de code pays, c'est le pays par défaut pour la langue qui est utilisé.</p> <p>Si vous n'utilisez pas ISOLOCALE, ce sont les paramètres système régionaux par défaut qui sont utilisés.</p>

## Exemples

### Fusion de tables ayant des noms de champs clés identiques

L'exemple suivant fusionne deux tables ayant des noms de champs clés identiques :

```
OPEN Position_Employés_1 PRIMARY
OPEN Position_Employés_2 SECONDARY
MERGE ON Nom TO "TousEmployés" PRESORT
```

### Fusion de tables ayant des noms de champs clés différents

L'exemple suivant fusionne deux tables ayant des noms de champs clés différents :

```
OPEN Position_Employés_1 PRIMARY  
OPEN Position_Employés_2 SECONDARY  
MERGE PKEY Nom SKEY Nom TO "TousEmployés" PRESORT
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Alternatives à la fusion

L'exécution correcte d'une fusion peut s'avérer délicate. Vous pouvez obtenir le même résultat en procédant à une concaténation, à une extraction et à un ajout, puis à un tri des données.

Pour plus d'informations, consultez les sections "Commande APPEND" Page 73 et "Commande EXTRACT" Page 206.

Si les deux tables sources sont déjà triées, la fusion est plus efficace et peut s'opérer plus rapidement.

# Commande NOTES

Crée, modifie ou supprime une note associée à un enregistrement individuel dans une table Analytics.

## Syntaxe

```
NOTES <IF test> <TEXT texte_note> <APPEND> <CLEAR>
```

## Paramètres

Nom	Description
IF <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.</p> <p><b>Remarque</b></p> <p>La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p> <ul style="list-style-type: none"> <li>Si vous n'indiquez pas de test IF, le texte de la note est ajouté à chaque enregistrement de la table</li> <li>Si vous indiquez un test IF et CLEAR, les notes des enregistrements qui remplissent la condition sont supprimées</li> </ul>
TEXT <i>texte_note</i> optionnel	Texte à ajouter sous forme de note. <i>texte_note</i> doit être soit une chaîne entre guillemets, soit une expression caractères.
APPEND optionnel	Le texte de la note est ajouté à la fin de toutes les notes existantes. Si vous omettez ce paramètre, toutes les notes existantes sont écrasées.
EFFACER optionnel	Les notes sont supprimées. Même si toutes les notes des enregistrements d'un tableau sont supprimées, le champ auto-généré <b>RecordNote</b> n'est pas supprimé du format de la table.

## Exemples

### Ajout de la même note à plusieurs enregistrements

Toutes les notes existantes pour les enregistrements indiqués sont écrasées :

```
NOTES IF MATCH(RECNO();1;3;5;7) TEXT "texte note"
```

## Ajout ou annexion de la même note à plusieurs enregistrements

Le nouveau texte de note est ajouté aux notes existantes pour les enregistrements indiqués :

```
NOTES IF MATCH(RECNO();1;3;5;7) TEXT "texte note" APPEND
```

## Suppression de notes à partir d'enregistrements multiples

Toutes les notes d'enregistrements dans la table sont supprimées :

```
NOTES CLEAR
```

Les notes des enregistrements spécifiés sont supprimées :

```
NOTES IF MATCH(RECNO();1;3;5;7) CLEAR
```

Les notes des enregistrements 1 à 100 sont supprimées :

```
NOTES IF RECNO() <= 100 CLEAR
```

# Remarques

## Suppression du champ NoteEnregistrement

Pour supprimer le champ **NoteEnregistrement** du format de la table et toutes les notes de la table, utilisez la commande DELETE NOTES sans aucune option spécifiée.

# Commande NOTIFY

Envoie un message de notification par e-mail.

## Syntaxe

```
NOTIFY USER nom_utilisateur<PASSWORD mdp> MAILBOX nom_chemin ADDRESS destinataire
<CC destinataire_cc> <BCC destinataire_cci> <SUBJECT objet> MESSAGE message
<ATTACHMENT nom_chemin>
```

## Paramètres

Nom	Description
USER <i>nom_utilisateur</i>	Adresse électronique de l'expéditeur.
PASSWORD <i>mdp</i> optionnel	Mot de passe du serveur de messagerie.
MAILBOX <i>nom_chemin</i>	Nom du serveur SMTP à utiliser pour envoyer le courrier électronique. Par exemple : <div style="border: 1px solid black; padding: 5px; margin: 5px 0;">MAILBOX "mailserver.acl.com"</div>
ADDRESS <i>destinataire</i>	Adresse électronique d'un ou plusieurs destinataires. Séparez plusieurs adresses électroniques par des virgules. Saisissez 1020 caractères au maximum.
CC <i>destinataire_cc</i> optionnel	Adresse électronique d'un ou plusieurs destinataires mis en copie carbone. Séparez plusieurs adresses électroniques par des virgules. Saisissez 1020 caractères au maximum.
BCC <i>destinataire_cci</i> optionnel	Adresse électronique d'un ou plusieurs destinataires mis en copie carbone invisible. Séparez plusieurs adresses électroniques par des virgules.
SUBJECT <i>objet</i> optionnel	Ligne d'objet du message électronique.
MESSAGE <i>message</i>	Corps de texte du message électronique. Le message est en texte brut et ne prend pas en charge HTML. Si vous souhaitez insérer un saut de ligne dans votre message, utilisez deux caractères caret : ^^.

Nom	Description
ATTACHMENT <i>nom_chemin</i> optionnel	<p>Chemin d'accès et nom de fichier d'une ou de plusieurs pièces jointes. Doit être une chaîne entre guillemets.</p> <p>Indiquez plusieurs pièces jointes en saisissant les fichiers dans une liste de valeurs séparées par des virgules pour <i>nom_chemin</i> :</p> <pre>ATTACHMENT "résultat1,résultat2"</pre>

## Exemples

### Envoi d'un e-mail de rapport d'erreurs

Vous exécutez un script, et vous souhaitez envoyer un e-mail de notification si le script échoue. À l'aide de NOTIFY, vous définissez le message électronique et incluez deux pièces jointes :

- le fichier trace
- un fichier .fil contenant les erreurs enregistrées

```
NOTIFY USER "support@societe.com" MAILBOX "mail.societe.com" ADDRESS "script_admin@a-cl.com" SUBJECT "Rapport d'erreur" MESSAGE "Echec du traitement du script. Détails joints." ATTACHMENT "Erreurs.fil,ACL_Demo.log"
```

## Remarques

### Destinataires et pièces jointes

Vous pouvez utiliser la commande NOTIFY pour envoyer des messages de notification à un ou plusieurs destinataires. Les messages peuvent inclure des fichiers de données et des projets Analytics en pièces jointes.

La commande NOTIFY peut être utilisée pour informer les personnes concernées de l'échec inattendu d'un script.

### Protocoles et ports

La commande peut être utilisée avec n'importe quel serveur de messagerie qui prend en charge le protocole SMTP (Simple Mail Transfer Protocol), utilisé par Microsoft Exchange et de nombreux serveurs de messagerie. La commande NOTIFY peut également être utilisée avec d'anciennes applications de courrier électronique (Microsoft et autres) qui envoient le courrier électronique localement.

NOTIFY utilise le port 25, alors ce port doit être ouvert ou la commande échoue.



## Gestion des erreurs

Si Analytics n'est pas en mesure de se connecter avec le serveur de messagerie, il réalise cinq tentatives supplémentaires pour se connecter, avec une pause de 10 secondes entre chaque tentative. Si aucune tentative de connexion n'aboutit, la commande NOTIFY est annulée (un message est ajouté à la trace), mais le traitement du script se poursuit.

Vous pouvez utiliser la commande SET pour modifier ce comportement par défaut. Vous pouvez spécifier un nombre différent de tentatives de connexion et une durée différente entre chaque tentative ou bien vous pouvez désactiver des tentatives de connexion supplémentaires. Vous pouvez aussi spécifier l'arrêt du traitement du script par Analytics en cas d'annulation de la commande NOTIFY. Pour plus d'informations, consultez la section "Commande SET" Page 435.

Un destinataire de message électronique incorrect n'est pas considéré comme un échec de la commande NOTIFY et n'entraîne pas l'arrêt d'un script quel que soit le paramètre associé.

# Commande OPEN

Ouvre une table Analytics et le fichier de données associé.

## Syntaxe

```
OPEN {nom_table|fichier_données<FORMAT nom_format>}<BUFFERLENGTH longueur>
<CRLF> <DBASE> <INDEX fichier_index> <PRIMARY|SECONDARY> <SKIP octets>
<RELATION champ_clé>
```

## Paramètres

Nom	Description
<i>nom_table</i>	Nom de la table Analytics à ouvrir.
<i>fichier_données</i>	Fichier de données à associer à la table indiquée par FORMAT <i>nom_format</i> . Analytics considère que le fichier a une extension .fil si aucune n'est spécifiée. Pour ouvrir un fichier sans extension, ajoutez un point (.) à la fin du nom de fichier.
FORMAT <i>nom_format</i> optionnel	Format de table Analytics à appliquer au fichier de données que vous ouvrez comme table.
BUFFERLENGTH <i>n</i> optionnel	Longueur en octets de la zone tampon d'entrée à allouer à la table. La valeur par défaut est 33 000 octets.  L'augmentation de la taille des zones tampons peut accélérer la vitesse de traitement au détriment de la disponibilité de la mémoire (RAM) pour le stockage des commandes Analytics.  Lors de la lecture de blocs IBM de longueur variable, si certains dépassent la longueur de la mémoire tampon, Analytics affiche un message d'erreur et arrête le traitement. La valeur par défaut est définie dans le champ <b>Taille mémoire tampon</b> de l'onglet <b>Table</b> dans la boîte de dialogue <b>Options</b> .  La longueur par défaut permettant de gérer la plupart des situations, la modification de BUFFERLENGTH <i>n</i> est rarement nécessaire.
CRLF facultatif	Indique qu'un fichier ASCII de longueur variable doit être lu. Analytics s'ajuste automatiquement aux variations de longueur d'enregistrement.  Par défaut, les fichiers sont considérés comme des fichiers de longueur fixe.
DBASE optionnel	Indique que la source de données est un fichier dBASE. Analytics reconnaît le type de fichier dBASE et crée automatiquement une table à partir de sa description. Peut être ignoré pour les fichiers dBASE comportant une extension .dbf.

Nom	Description						
INDEX <i>fichier_index</i> facultatif	Fichier d'index à appliquer à la table lorsqu'elle est ouverte.  Le système considère que le nom de fichier d'index a une extension .inx si aucune n'est spécifiée. La commande INDEX peut être spécifiée avec des tables principales ou secondaires.						
PRIMARY   SECONDARY optionnel	Indique si une table est ouverte en tant que table principale ou secondaire. Si ce paramètre est ignoré, la table est ouverte en tant que table principale.						
SKIP <i>octets</i> facultatif	Nombre d'octets à ignorer au début physique de la table.  La commande SKIP peut être utilisée pour ignorer les enregistrements d'en-tête ou les sections de début de table qui ne respectent pas la structure du reste de la table. Si ce paramètre est ignoré, la table est lue en commençant au premier octet.  <b>Remarque</b> <table border="1"> <tbody> <tr> <td>Analytics non Unicode</td> <td>1 octet = 1 caractère</td> </tr> <tr> <td>Analytics Unicode, données ASCII étendu (ANSI)</td> <td>1 octet = 1 caractère</td> </tr> <tr> <td>Analytics Unicode, données Unicode</td> <td>2 octets = 1 caractère</td> </tr> </tbody> </table> Pour les données Unicode, spécifiez uniquement un nombre d'octets pair. Indiquer un nombre d'octets impair peut entraîner l'affichage incorrect des caractères.	Analytics non Unicode	1 octet = 1 caractère	Analytics Unicode, données ASCII étendu (ANSI)	1 octet = 1 caractère	Analytics Unicode, données Unicode	2 octets = 1 caractère
Analytics non Unicode	1 octet = 1 caractère						
Analytics Unicode, données ASCII étendu (ANSI)	1 octet = 1 caractère						
Analytics Unicode, données Unicode	2 octets = 1 caractère						
RELATION <i>champ_clé</i> facultatif	Indique que la table doit être ouverte en tant que table associée ad hoc. Analytics ne conserve pas cette relation lorsque la table est fermée.  Vous devez également indiquer le paramètre INDEX lorsque vous utilisez RELATION. <i>champ_clé</i> est l'expression ou le champ clé utilisé pour créer la relation entre deux tables.						

## Exemples

### Ouverture d'une table en indiquant un format de table

Vous ouvrez la table `Avril_2012` à l'aide du format de table `Mars_2012` :

```
OPEN Avril_2012 FORMAT Mars_2012
```

### Ouverture d'un fichier dBASE

Vous ouvrez un fichier dBASE appelé `Inventaire.dbf` pour lequel aucune table n'existe :

```
OPEN Inventaire
```

## Ouverture d'une table et application d'un index pré-existant

Pour ouvrir une table principale ou secondaire et appliquer un index pré-existant de la table, utilisez la syntaxe suivante :

```
OPEN Comptes_clients INDEX Numéro_client_AR
```

```
OPEN Client SECONDARY INDEX Numéro_client
```

## Ouverture d'une table et création d'une relation ad hoc avec une autre table

Vous devez établir une relation temporaire entre une table ouverte nommée **Clients** (table principale) et une table nommée **Comptes\_clients** (table secondaire).

Vous utilisez un index nommé **Index\_client** et un champ clé dans la table principale appelé **Nom** :

```
OPEN Comptes_clients INDEX Index_client RELATION Nom
```

# Commande OUTLIERS

Identifie les valeurs statistiques aberrantes dans un champ numérique. Des valeurs aberrantes peuvent être identifiées pour le champ dans son ensemble ou pour des groupes distincts basés sur les valeurs identiques contenues dans un ou plusieurs champs clés de type caractère, numérique ou DateHeure.

## Syntaxe

```
OUTLIERS {AVERAGE|MEDIAN} {PKEY champ_clé<...n>|NOKEY} ON champ_numérique
<OTHER champ<...n>> NUMSTDEV nombre_écarts_types<IF test> <TO {SCREEN|nom_table}>
<PRESORT> <WHILE test> <FIRST plage|NEXT plage> <OPEN>
```

### Remarque

Vous ne pouvez pas exécuter la commande OUTLIERS localement par rapport à une table de serveur.

Vous devez indiquer en entier le nom de la commande OUTLIERS. Vous ne pouvez pas l'abréger.

## Paramètres

Nom	Description
AVERAGE   MEDIAN	<p>Méthode de calcul du point central des valeurs dans <i>champ_numérique</i> (le champ de la valeur aberrante).</p> <ul style="list-style-type: none"> <li>○ <b>AVERAGE</b> : calculez la moyenne des valeurs</li> <li>○ <b>MEDIAN</b> : calculez la médiane des valeurs</li> </ul> <p>Le point central est calculé pour l'un ou l'autre des éléments suivants :</p> <ul style="list-style-type: none"> <li>○ soit pour le champ numérique dans son ensemble</li> <li>○ soit pour les valeurs numériques de chaque groupe de champs clés</li> </ul> <p>Le point central est utilisé par la suite pour calculer l'écart type du champ numérique ou de chaque groupe.</p> <p><b>Remarque</b></p> <p>Si vous indiquez MEDIAN, <i>champ_numérique</i> doit être trié. Utilisez PRESORT si <i>champ_numérique</i> n'est pas déjà trié.</p> <p><b>Astuce</b></p> <p>Si les données dans lesquelles vous recherchez des valeurs aberrantes sont sensiblement faussées, MEDIAN peut produire des résultats plus représentatifs du volume de données.</p>
PKEY <i>champ_clé</i> <...n>	Si vous indiquez PKEY, les valeurs aberrantes sont identifiées au niveau du groupe. Si

Nom	Description
NOKEY	<p>vous indiquez NOKEY, les valeurs aberrantes sont identifiées au niveau du champ.</p> <ul style="list-style-type: none"> <li> <b>PKEY <i>champ_clé</i></b> : champ(s) à utiliser pour regrouper les données dans la table            Les champs clés peuvent être de type caractère, numérique ou DateHeure. Les champs multiples doivent être séparés par des espaces et il peut s'agir de différents types de données.            Si vous spécifiez plusieurs champs, c'est que vous avez créé des groupes imbriqués. L'imbrication suit l'ordre dans lequel vous indiquez les champs.            Pour chaque groupe de champs clés, un écart type est calculé pour les valeurs numériques du groupe dans <i>champ_numérique</i>. L'écart type du groupe est utilisé comme référence pour identifier les valeurs aberrantes du groupe.           <div style="border-left: 2px solid #0056b3; padding-left: 10px; margin: 10px 0;"> <p><b>Remarque</b></p> <p>Le(s) champ(s) clé(s) doivent être triés. Utilisez PRESORT si un ou plusieurs champs ne sont pas déjà triés.</p> </div> </li> <li> <b>NOKEY</b> : ne regroupe pas les données dans la table            Un écart type est calculé pour <i>champ_numérique</i> dans sa globalité. L'écart type du champ est utilisé comme référence pour identifier les valeurs aberrantes du champ.         </li> </ul>
ON <i>champ_numérique</i>	<p>Champ numérique à examiner pour y rechercher des valeurs aberrantes. Vous ne pouvez examiner qu'un seul champ à la fois.</p> <p>Les valeurs aberrantes sont les valeurs sortant des limites supérieure et inférieure définies par l'écart type du champ ou groupe ou par un multiple spécifié de l'écart type.</p>
OTHER <i>champ &lt;...n&gt;</i> optionnel	<p>Un ou plusieurs champs supplémentaires à inclure dans la sortie.</p> <div style="border-left: 2px solid #0056b3; padding-left: 10px; margin: 10px 0;"> <p><b>Remarque</b></p> <p>Les champs clés et le champ de la valeur aberrante sont inclus automatiquement dans la table de sortie et ne doivent pas être spécifiés à l'aide de OTHER.</p> </div>
NUMSTDEV <i>nombre_écarts_types</i>	<p>Dans <i>champ_numérique</i>, nombre d'écarts types compris entre la moyenne ou la médiane et les limites supérieure et inférieure de valeurs aberrantes. Vous pouvez spécifier un entier positif ou une valeur numérique décimale (1; 1,5; 2 . . . )</p> <p>La formule permettant de créer des limites de valeurs aberrantes est la suivante :</p> <p>moyenne/médiane ± (<i>nombre_écarts_types</i> * écart type)</p> <div style="border-left: 2px solid #0056b3; padding-left: 10px; margin: 10px 0;"> <p><b>Remarque</b></p> <p>L'écart type est un indicateur de dispersion d'un jeu de données, c'est-à-dire qu'il indique dans quelle mesure les valeurs s'évalent. Le calcul des valeurs aberrantes utilise l'écart type de la population.</p> </div> <p><b>Exemple de limites de valeurs aberrantes</b></p> <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <p>NUMSTDEV 2</p> </div> <p>définit, pour <i>champ_numérique</i> dans son ensemble, ou pour chaque groupe de champs clés :</p>

Nom	Description				
	<ul style="list-style-type: none"> <li>une limite supérieure de valeurs aberrantes située à 2 écarts types au-dessus de la moyenne ou de la médiane moyenne/médiane + (2 * écarts types)</li> <li>une limite inférieure de valeurs aberrantes située à 2 écarts types en dessous de la moyenne ou de la médiane moyenne/médiane - (2 * écarts types)</li> </ul> <p>Toutes les valeurs qui sont supérieures à la limite supérieure ou qui sont inférieures à la limite inférieure figurent dans les valeurs aberrantes dans les résultats de sortie.</p> <p><b>Remarque</b> Pour le même jeu de données, à mesure que vous augmentez la valeur dans <i>nombre_écarts_types</i>, vous diminuez peut-être le nombre de valeurs aberrantes renvoyé.</p>				
IF test optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.</p> <p><b>Remarque</b> La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p>				
TO SCREEN   <i>nom_table</i> optionnel	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li><b>SCREEN</b> : affiche les résultats dans la zone d'affichage d'Analytics.</li> <li><b><i>nom_table</i></b> : enregistre les résultats dans une table Analytics</li> </ul> <p>Indiquez <i>nom_table</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple : TO "Sortie.FIL"</p> <p>Par défaut, le fichier de données de table (.FIL) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>TO "C:\Sortie.FIL"</li> <li>TO "Résultats\Sortie.FIL"</li> </ul> <p><b>Remarque</b> La longueur des noms des tables est limitée à 64 caractères alpha-numériques, sans l'extension .FIL. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>				
PRESORT optionnel	<p>Effectue une opération de tri avant l'exécution de la commande.</p> <table border="1" data-bbox="483 1633 1446 1833"> <thead> <tr> <th data-bbox="483 1633 964 1696">Si vous indiquez PRESORT et:</th> <th data-bbox="964 1633 1446 1696">Trie par :</th> </tr> </thead> <tbody> <tr> <td data-bbox="483 1696 964 1833">PKEY, AVERAGE</td> <td data-bbox="964 1696 1446 1833"> <ul style="list-style-type: none"> <li>champ(s) clé(s)</li> <li>champ(s) clé(s) <i>champ_numérique</i> (si <i>champ_numérique</i> est calculé)</li> </ul> </td> </tr> </tbody> </table>	Si vous indiquez PRESORT et:	Trie par :	PKEY, AVERAGE	<ul style="list-style-type: none"> <li>champ(s) clé(s)</li> <li>champ(s) clé(s) <i>champ_numérique</i> (si <i>champ_numérique</i> est calculé)</li> </ul>
Si vous indiquez PRESORT et:	Trie par :				
PKEY, AVERAGE	<ul style="list-style-type: none"> <li>champ(s) clé(s)</li> <li>champ(s) clé(s) <i>champ_numérique</i> (si <i>champ_numérique</i> est calculé)</li> </ul>				

Nom	Description										
	<table border="1"> <thead> <tr> <th>Si vous indiquez PRESORT et:</th> <th>Trie par :</th> </tr> </thead> <tbody> <tr> <td></td> <td> <p><b>Remarque</b></p> <p>Le tri d'un <i>champ_numérique</i> calculé est une condition technique interne d'Analytics.</p> </td> </tr> <tr> <td>PKEY, MEDIAN</td> <td>champ(s) dé(s) puis par <i>champ_numérique</i></td> </tr> <tr> <td>NOKEY, AVERAGE</td> <td>aucun tri</td> </tr> <tr> <td>NOKEY, MEDIAN</td> <td><i>champ_numérique</i></td> </tr> </tbody> </table> <p><b>Astuce</b></p> <p>Si le(s) champ(s) approprié(s) de la table d'entrée sont déjà triés, vous pouvez gagner du temps de traitement en évitant d'indiquer PRESORT.</p> <p><b>Remarque</b></p> <p>Vous ne pouvez pas utiliser PRESORT dans la commande GROUP.</p>	Si vous indiquez PRESORT et:	Trie par :		<p><b>Remarque</b></p> <p>Le tri d'un <i>champ_numérique</i> calculé est une condition technique interne d'Analytics.</p>	PKEY, MEDIAN	champ(s) dé(s) puis par <i>champ_numérique</i>	NOKEY, AVERAGE	aucun tri	NOKEY, MEDIAN	<i>champ_numérique</i>
Si vous indiquez PRESORT et:	Trie par :										
	<p><b>Remarque</b></p> <p>Le tri d'un <i>champ_numérique</i> calculé est une condition technique interne d'Analytics.</p>										
PKEY, MEDIAN	champ(s) dé(s) puis par <i>champ_numérique</i>										
NOKEY, AVERAGE	aucun tri										
NOKEY, MEDIAN	<i>champ_numérique</i>										
WHILE <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.</p> <p><b>Remarque</b></p> <p>Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.</p>										
FIRST <i>plage</i>   NEXT <i>plage</i> optionnel	<p>Le nombre d'enregistrements à traiter :</p> <ul style="list-style-type: none"> <li><b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li><b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> </ul> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter.</p> <p>Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p>										
OPEN optionnel	<p>Ouvre la table créée par la commande après l'exécution de la commande. Valide uniquement lorsque la commande crée une table de sortie.</p>										

## Exemples

### Identifier les montants de transactions sortant de l'ordinaire

Vous souhaitez identifier les montants des transactions sortant de l'ordinaire dans toute la table **Ar** dans



### Exemple de projet.acl.

Vous décidez de définir les limites des valeurs aberrantes sur 3 fois l'écart type du champ **Montant**. Le test renvoie 16 valeurs aberrantes dans la table de 772 enregistrements.

```
OPEN Comptes_clients
OUTLIERS AVERAGE NOKEY ON Montant NUMSTDEV 3 PRESORT TO "Valeurs_aberrantes_
Client.fil" OPEN
```

Vous réitérez le test, mais vous augmentez le multiple de l'écart type en le faisant passer à 3,5. Le test renvoie désormais 6 valeurs aberrantes car les limites des valeurs aberrantes sont bien plus loin du point centrale des valeurs dans le champ **Montant**.

```
OPEN Comptes_clients
OUTLIERS AVERAGE NOKEY ON Montant NUMSTDEV 3,5 PRESORT TO "Valeurs_aberrantes_
Client.fil" OPEN
```

## Identifier les montants des transactions sortant de l'ordinaire pour chaque client

Pour chaque client de la table **Ar** dans [Exemple de projet.acl](#), vous souhaitez identifier les montants des transactions sortant de l'ordinaire.

Vous décidez de définir les limites des valeurs aberrantes sur 3 fois l'écart type du groupe de transactions de chaque client.

```
OPEN Comptes_clients
OUTLIERS AVERAGE PKEY Num ON Montant NUMSTDEV 3 PRESORT TO "Valeurs_aberrantes_
Comptabilité_Client.fil" OPEN
```

Le test renvoie 7 valeurs aberrantes. L'écart type et la moyenne sont mentionnés pour chaque groupe de transactions de chaque client :

	Numéro client (No)	Montant de trans	STDEV	AVERAGE	Numéro de groupe
1	065003	4 954,64	1015,58	833,83	1
2	<b>262001</b>	3 567,34	772,44	438,81	2
3	<b>262001</b>	(2 044,82)	772,44	438,81	2
4	376005	(931,55)	411,18	484,57	3
5	501657	5 549,19	1332,80	441,14	4
6	811002	3 409,82	634,20	672,10	5

	Numéro client (No)	Montant de trans	STDEV	AVERAGE	Numéro de groupe
7	925007	3 393,87	736,48	906,16	6

## Identification des valeurs aberrantes pour le client 262001

Le client 262001 présente 101 transactions dans la table Ar ; deux d'entre elles sont signalées comme étant des valeurs aberrantes car elles dépassent les limites des valeurs aberrantes correspondant à ce client-là :

Valeur aberrante	Limite inférieure	Limite supérieure	Valeur aberrante
(2 044,82)	(1 878,51)	2 756,13	3 567,34

## Mode de calcul des limites des valeurs aberrantes pour le client 262001

Les limites des valeurs aberrantes correspondent à la moyenne de toutes les transactions du client 262001, plus ou moins le multiple spécifié de l'écart type des transactions :

Moyenne de toutes les transactions du client 262001	438,81
Multiple spécifié de l'écart type	3
Écart type des transactions	772,44
	$438,81 \pm (3 * 772,44)$ $= 438,81 \pm 2317,32$ $= (1878,51) \text{ (limite inférieure)}$ $= 2756,13 \text{ (limite supérieure)}$

## Utilisation de MEDIAN pour identifier les montants des transactions sortant de l'ordinaire pour chaque client

Vous utilisez MEDIAN, plutôt que AVERAGE, pour effectuer le même test de recherche de valeurs aberrantes que celui effectué dans l'exemple précédent.

```
OPEN Comptes_clients
OUTLIERS MEDIAN PKEY Num ON Montant NUMSTDEV 3 PRESORT TO "Valeurs_aberrantes_
Comptabilité_Client_Médiane.fil" OPEN
```

Le test renvoie 10 valeurs aberrantes au lieu des 7 renvoyées lors du test précédent. Selon la nature des données, MEDIAN et AVERAGE peuvent renvoyer des résultats quelque peu différents :

	Numéro client (No)	Montant de trans	STDEV	MEDIAN	Numéro de groupe
1	065003	4 954,64	1015,58	663,68	1
2	262001	(2 044,82)	772,44	450,67	2
3	262001	3 567,34	772,44	450,67	2
4	376005	(931,55)	411,18	517,16	3
5	501657	4 426,14	1332,80	146,80	4
6	501657	5 549,19	1332,80	146,80	4
7	811002	3 409,82	634,20	624,53	5
8	925007	2 972,78	736,48	717,88	6
9	925007	3 030,71	736,48	717,88	6
10	925007	3 393,87	736,48	717,88	6

## Mode de calcul des limites des valeurs aberrantes pour chaque client

Les limites des valeurs aberrantes correspondent à la valeur médiane des transactions de chaque client, plus ou moins le multiple spécifié de l'écart type des transactions.

Par exemple, pour le client 262001 :  $450,67 \pm (3 * 772,44)$

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Ajouter des champs de limites de valeurs aberrantes à la table de résultats

Analytics ajoute automatiquement les champs calculés **STDEV** et **AVERAGE** ou **MEDIAN** à la table de résultats de valeurs aberrantes. Par ailleurs, vous trouverez peut-être utile d'ajouter deux champs calculés affichant les limites des valeurs aberrantes utilisées pour identifier les valeurs aberrantes dans la table de résultats.

1. Ouvrez la table Résultats des valeurs aberrantes.
2. Collez cette expression dans la ligne de commande Analytics, modifiez-la si besoin, et appuyez sur Entrée :

```
DEFINE FIELD Limite_inférieure COMPUTED AVERAGE - (nombre_écarts_types * STDEV)
```

- Pour *nombre\_écarts\_types*, substituez le multiple de l'écart type réel que vous avez utilisé.
  - Si vous avez utilisé la médiane au lieu de la moyenne comme point central, substituez MEDIAN à AVERAGE.
3. Collez cette expression dans la ligne de commande Analytics, modifiez-la si besoin, et appuyez sur Entrée :

```
DEFINE FIELD Limite_supérieure COMPUTED AVERAGE + (nombre_écarts_types * STDEV)
```

- Pour *nombre\_écarts\_types*, substituez le multiple de l'écart type réel que vous avez utilisé.
  - Si vous avez utilisé la médiane au lieu de la moyenne comme point central, substituez MEDIAN à AVERAGE.
4. Cliquez dans la vue avec le bouton droit de la souris et sélectionnez **Ajouter des colonnes**.
  5. Dans la liste **Champs disponibles**, double-cliquez sur **Limite\_inférieure** et sur **Limite\_supérieure** pour les ajouter à la liste **Champs sélectionnés**.
  6. Cliquez sur **OK**.
  7. Facultatif. Remplacez les champs ajoutés en faisant glisser les en-têtes des colonnes.

# Commande PASSWORD

Crée une définition de mot de passe, sans valeur de mot de passe, qui invite les utilisateurs à fournir un mot de passe pendant qu'un script s'exécute.

## Syntaxe

```
PASSWORD num <invite>
```

## Paramètres

Nom	Description
<i>num</i>	Une valeur entre 1 et 10 qui identifie la définition de mot de passe.
<i>invite</i> optionnel	Une expression de type caractère valide à afficher dans la boîte de dialogue utilisée pour inviter à saisir le mot de passe. Placez les chaînes de caractères littérales entre guillemets.  Si <i>invite</i> est ignoré, une boîte de dialogue par défaut sans message s'affiche.

## Exemples

### Invitation à saisir un mot de passe

Vous utilisez la commande PASSWORD pour inviter l'utilisateur à saisir les trois mots de passe obligatoires dans un script. Une fois les mots de passe saisis, le script peut poursuivre le traitement sans interruption.

```
PASSWORD 1 "Saisissez le mot de passe de la base de données Comptes clients"
PASSWORD 2 "Saisissez le mot de passe de la base de données Comptes fournisseurs"
PASSWORD 3 "Saisissez le mot de passe de la base de données Client"
```

### Spécification d'un mot de passe lors de l'actualisation d'une table Analytics

Vous combinez la commande PASSWORD avec la commande REFRESH pour mettre à jour un fichier de données protégé par mot de passe :

```
PASSWORD 1 "Mot de passe :"  
REFRESH Abc PASSWORD 1
```

### Spécification d'un mot de passe pour définir une table de serveur

Vous utilisez la commande PASSWORD avec la commande DEFINE TABLE DB pour définir une table de serveur via le Connecteur AX, qui requiert un mot de passe pour le profil de base de données et un deuxième mot de passe pour le profil du serveur associé :

```
DEFINE TABLE DB SOURCE ProfilBD_Inventaire PASSWORD 9 PASSWORD 3
```

## Remarques

### Quand utiliser PASSWORD

Utilisez la commande PASSWORD pour inviter un utilisateur à saisir le mot de passe avant qu'un script accède, importe ou actualise des données protégées par mot de passe.

Vous pouvez créer jusqu'à dix définitions de mots de passe différentes dans un script.

PASSWORD est utile dans les cas suivants :

- vous souhaitez éviter de saisir un vrai mot de passe dans un script, ce qui est exigé par la commande SET PASSWORD
- les utilisateurs individuels doivent saisir des mots de passe distincts

### Stockage des mots de passe

Les mots de passe saisis par les utilisateurs sont stockés temporairement et en toute sécurité dans la mémoire.

Lorsqu'un utilisateur saisit un mot de passe dans la boîte de dialogue de l'invite, les caractères sont masqués avec des astérisques (\*). Le mot de passe ne figure pas dans le script ni dans la trace.

### Stockage des mots de passe pour les outils d'analyse basés sur serveur

La commande PASSWORD n'est pas prise en charge par les outils d'analyse dans Robots ou sur AX Serveur ni par les scripts de serveur hérités.

Vous pouvez utiliser la balise PASSWORD pour inviter un utilisateur à saisir un mot de passe lorsqu'un utilisateur planifie un outil d'analyse dans Robots ou sur AX Serveur.

Vous pouvez utiliser la commande SET PASSWORD pour spécifier des mots de passe dans des scripts de serveur hérités.

# Commande PAUSE

Interrompt un script et affiche des informations destinées aux utilisateurs dans une boîte de dialogue.

## Syntaxe

```
PAUSE message <IF test>
```

## Paramètres

Nom	Description
<i>message</i>	Message à afficher dans la boîte de dialogue. La longueur maximale est de 199 caractères.  <i>message</i> doit être entre guillemets. Si le message contient des guillemets doubles, entourez-le de guillemets simples.
IF <i>test</i> optionnel	Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.  <b>Remarque</b> La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).

## Exemples

### Affichage d'un message d'erreur

Il faut que la saisie de l'utilisateur réponde à des conditions spécifiques. Lorsque vous détectez que l'entrée ne répond pas à ces conditions préalables, vous utilisez la commande PAUSE et affichez un message d'erreur dans une boîte de dialogue :

```
PAUSE "La classe de produit doit être une valeur à deux chiffres."
```

# Remarques

## Quand utiliser PAUSE

Utilisez PAUSE pour afficher des messages en lecture seule sur l'écran pendant l'exécution d'un script. Vous pouvez afficher des messages d'erreur ou des informations, telles que le résultat d'une opération d'analyse.

## Fonctionnement

Bien que la boîte de dialogue de message s'affiche, l'exécution du script est interrompue et reprend seulement une fois que l'utilisateur clique sur **OK** pour fermer la boîte de dialogue de message. C'est pourquoi vous ne pouvez pas utiliser PAUSE dans des scripts ou des outils d'analyse qui doivent s'exécuter sans assistance.

## Limites

PAUSE présente les limites suivantes :

- PAUSE ne peut pas figurer à l'intérieur de la commande GROUP
- la commande PAUSE ne peut pas être utilisée dans des outils d'analyse exécutés dans Robots ou sur AX Serveur



# Commande PREDICT

Applique un modèle prédictif à un jeu de données sans libellé pour prédire des classes ou des valeurs numériques associées à des enregistrements individuels.

## Syntaxe

```
PREDICT MODEL nom_modèle TO nom_table <IF test> <WHILE test> <FIRST plage|NEXT plage>
```

## Paramètres

Nom	Description
MODEL <i>nom_modèle</i>	<p>Nom du fichier de modèle à utiliser pour effectuer des prédictions sur des classes ou des valeurs. Vous utilisez un fichier de modèle généré préalablement par la commande TRAIN.</p> <p>Vous devez préciser l'extension de fichier <code>*.model</code>. Par exemple :</p> <pre>MODEL "Prédiction_prêt_par_défaut.model"</pre> <p><b>Remarque</b></p> <p>Le fichier de modèle doit avoir été entraîné sur un jeu de données contenant les mêmes champs que le jeu de données sans libellé (ou contenant en grande partie les mêmes champs).</p>
TO <i>nom_table</i>	<p>Nom de la table Analytics sortie par le processus de prédiction.</p> <p>La table contient les champs clés que vous avez spécifiés pendant le processus d'entraînement de l'apprentissage ainsi qu'un ou deux champs générés par le processus de prédiction :</p> <ul style="list-style-type: none"> <li>○ <b>Prédit</b> : classes ou valeurs numériques prédites associées à chaque enregistrement dans le jeu de données sans libellé</li> <li>○ <b>Probabilité</b> : (classification uniquement) probabilité qu'une classe ou valeur numérique prédite est exacte</li> </ul> <p>Indiquez <i>nom_table</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple : TO "Demandeurs_pret_prediction_defaut.FIL"</p> <p>Par défaut, le fichier de données de table (.FIL) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>○ TO "C:\Demandeurs_pret_prediction_defaut.FIL"</li> <li>○ TO "ML Predict output\Demandeurs_pret_prediction_defaut.FIL"</li> </ul>

Nom	Description
	<p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alpha-numériques, sans l'extension .FIL. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>
IF <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.</p> <p><b>Remarque</b></p> <p>La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p>
WHILE <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.</p> <p><b>Remarque</b></p> <p>Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.</p>
FIRST <i>plage</i>   NEXT <i>plage</i> optionnel	<p>Le nombre d'enregistrements à traiter :</p> <ul style="list-style-type: none"> <li>◦ <b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li>◦ <b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> </ul> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter.</p> <p>Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p>

## Exemples

### Utiliser un modèle de classification pour effectuer des prédictions

Vous entrez un modèle de classification dans la commande PREDICT pour effectuer des prédictions sur la valeur par défaut reçue par les demandeurs de prêt actuels s'ils reçoivent un prêt.

Auparavant, vous avez produit le modèle de classification à l'aide de la commande TRAIN avec un jeu de données de prêt historiques, y compris les informations par défaut concernant les prêts.

```
OPEN "Demandeurs_prêt_en_cours"
PREDICT MODEL "Prédiction_prêt_par_défaut.model" TO "Demandeurs_pret_prediction_
default.FIL"
```

### Utiliser un modèle de régression pour effectuer des prédictions

Vous entrez un modèle de régression dans la commande PREDICT pour faire des prédictions sur les futurs prix de vente des maisons.

Auparavant, vous aviez produit le modèle de régression à l'aide de la commande TRAIN avec un ensemble de données concernant les dernières ventes de maison, y compris le prix de vente.

```
OPEN "Evaluation_prix_maison"  
PREDICT MODEL "Prédiction_prix_maison.model" TO "Prix_maison_prédits.FIL"
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

# Commande PRINT

Imprime un fichier texte, un fichier trace Analytics ou un élément de projet Analytics qui a été exporté en tant que fichier externe (un script (.aclscript), une table (.layout) ou un environnement de travail (.wsp)). Vous pouvez également imprimer un graphe qui a été généré par une commande.

## Syntaxe

```
PRINT {nom_fichier|GRAPH}
```

## Paramètres

Nom	Description
<i>nom_fichier</i>   GRAPH	<p>Élément à imprimer :</p> <ul style="list-style-type: none"> <li> <b><i>nom_fichier</i></b> : chemin d'accès absolu ou relatif et nom du fichier à imprimer            Par exemple, "C:\ACL Data\Exemples de fichiers de données\ACL_Demo.log" ou "Exemples de fichiers de données\ACL_Demo.log".            Si le chemin d'accès ou le nom de fichier inclut des espaces, placez <i>nom_fichier</i> entre guillemets.         </li> <li> <b>GRAPH</b> : graphe précédemment renvoyé en tant que résultat d'une commande         </li> </ul>

## Exemples

### Impression d'un fichier trace

Pour imprimer le fichier trace du projet `ACL_Demo.acl`, indiquez la commande suivante :

```
PRINT "C:\ACL Data\Exemples de fichiers de données\ACL_Demo.log"
```

### Impression d'un graphe

Pour imprimer le graphe produit à partir de la commande BENFORD, indiquez les commandes suivantes :

```
OPEN TransAP_2002_Metaphor
BENFORD ON Montant_Facture LEADING 1 TO GRAPH
PRINT GRAPH
```

# Remarques

## Sélection d'une imprimante

L'imprimante utilisée est l'imprimante par défaut configurée dans Microsoft Windows. Pour modifier l'imprimante, changez l'imprimante par défaut dans Windows.

## Commandes associées

Pour imprimer le contenu d'une table Analytics dans un projet, utilisez la commande DO REPORT.

# Commande PROFILE

Génère une synthèse des statistiques pour un ou plusieurs champs ou expressions numériques dans une table Analytics.

## Syntaxe

```
PROFILE {<FIELDS> champ_numérique<...n>|<FIELDS> ALL} <IF test> <WHILE test> <FIRST
plage|NEXT plage>
```

## Paramètres

Nom	Description
FIELDS <i>champ_numérique</i> <...n>   FIELDS ALL	Indiquez des champs individuels à profiler ou spécifiez ALL pour profiler tous les champs numériques dans la table Analytics.
IF <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.</p> <p><b>Remarque</b></p> <p>La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p>
WHILE <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.</p> <p><b>Remarque</b></p> <p>Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.</p>
FIRST <i>plage</i>   NEXT <i>plage</i> optionnel	<p>Le nombre d'enregistrements à traiter :</p> <ul style="list-style-type: none"> <li>○ <b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li>○ <b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> </ul> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter.</p> <p>Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p>

# Exemples

## Profilage d'un seul champ

Vous profilez le champ **Salaire** :

```
OPEN Paie_Employé  
PROFILE FIELDS Salaire
```

La commande génère la sortie suivante :

Nom de champ	Valeur totale	Valeur absolue	Minimum	Maximum
<b>SALAIRE</b>	1 152 525	1 152 525	15 340	52 750

## Remarques

### Statistiques affichées dans la sortie

Les statistiques suivantes s'affichent pour chaque champ numérique ou expression numérique indiqué pour la commande :

- valeur totale
- valeur absolue
- valeur minimale
- valeur maximale

# Commande QUIT

Termine la session actuelle et ferme Analytics.

## Syntaxe

```
QUIT
```

## Exemples

### Vérifier si un fichier existe et fermer Analytics si ce n'est pas le cas

Vous avez créé un script à faire exécuter par d'autres utilisateurs, mais si un fichier requis est inexistant, vous voulez fermer Analytics.

L'exemple suivant vérifie l'existence du fichier `Inventaire.csv` requis, et Analytics se ferme s'il n'existe pas.

```
IF FILESIZE("Inventaire.csv") = -1 QUIT
```

### Fermer automatiquement Analytics à la fin d'un script

Le script ci-dessous totalise la table Inventaire et produit des résultats de sortie, puis ferme automatiquement Analytics :

```
OPEN Inventaire  
SUMMARIZE ON Emplacement ClsProd SUBTOTAL Valeur TO "Valeur_inventaire_par_classe_ emplacement.FIL" PRESORT CPERCENT  
QUIT
```

## Remarques

### Les modifications sont enregistrées

Lorsque QUIT est exécuté, toutes les tables Analytics ouvertes sont enregistrées et fermées avant l'arrêt. Si vous avez modifié la vue active ou un script sans enregistrer les modifications, Analytics vous invite à le faire avant la fermeture.



# Commande RANDOM

Génère un ensemble de nombres aléatoires.

## Syntaxe

```
RANDOM NUMBER n <SEED valeur_départ> MINIMUM valeur_min MAXIMUM valeur_max
<COLUMNS n> <UNIQUE> <SORTED> <TO {SCREEN|nom_fichier}> <APPEND>
```

## Paramètres

Nom	Description
NUMBER <i>n</i>	Taille du jeu de nombres aléatoires à générer. Il est possible de générer 32 767 nombres au maximum.
SEED <i>valeur_départ</i> optionnel	Valeur utilisée pour lancer un générateur de nombres aléatoires. Si vous indiquez une valeur de départ, il peut s'agir de n'importe quel nombre. Toute valeur de départ unique génère un jeu différent de nombres aléatoires. Si vous indiquez à nouveau la même valeur de départ, le même jeu de nombres aléatoires est généré. Il se peut que vous deviez régénérer le même jeu de nombres aléatoires si vous devez répliquer l'analyse. <ul style="list-style-type: none"> <li>○ <b>Valeur de départ</b> : indiquez de manière explicite une valeur de départ et enregistrez-la si vous voulez avoir la possibilité de répliquer un jeu particulier de nombres aléatoires.</li> <li>○ <b>Pas de valeur de départ</b> : saisissez une valeur de départ égale à « 0 » ou laissez-la vierge si vous voulez qu'Analytics sélectionne une valeur de départ de manière aléatoire.</li> </ul>
MINIMUM <i>valeur_min</i>	Le plus petit numéro du jeu de nombres aléatoires. Toute valeur ou expression numérique valide est autorisée.
MAXIMUM <i>valeur_max</i>	Le plus grand numéro du jeu de nombres aléatoires. Toute valeur ou expression numérique valide est autorisée.
COLUMNS <i>n</i> optionnel	Nombre de colonnes à utiliser pour afficher le jeu de nombres aléatoires. Si vous omettez COLUMNS, le nombre par défaut est 6 colonnes.
UNIQUE optionnel	Inclut seulement des nombres uniques dans le jeu de nombres aléatoires. Si vous ignorez UNIQUE, les valeurs en double sont autorisées dans le jeu de nombres aléatoires.

Nom	Description
	<p><b>Remarque</b></p> <p>N'indiquez pas UNIQUE lorsque la taille indiquée du jeu de nombres aléatoires dépasse 75 % de la plage comprise entre MINIMUM et MAXIMUM. Ce choix peut entraîner l'abandon de nombreuses sélections de nombres aléatoires.</p>
SORTED optionnel	<p>Affiche le jeu de nombres aléatoires dans l'ordre croissant.</p> <p>Si vous ignorez SORTED, les nombres s'affichent dans l'ordre de sélection aléatoire.</p>
TO SCREEN   <i>nom_fichier</i> optionnel	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li>◦ <b>SCREEN</b> : affiche les résultats dans la zone d'affichage d'Analytics.</li> <li>◦ <b><i>nom_fichier</i></b> : enregistre les résultats dans un fichier</li> </ul> <p>Indiquez <i>nom_fichier</i> sous forme de chaîne entre guillemets avec l'extension de fichier appropriée. Par exemple : TO "Sortie.TXT"</p> <p>Par défaut, le fichier est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.TXT"</li> <li>• TO "Résultats\Sortie.TXT"</li> </ul> <p>Si vous ignorez TO, le jeu de nombres aléatoires sort à l'écran.</p>
APPEND optionnel	<p>Ajoute la sortie de commande à la fin d'un fichier existant au lieu de remplacer ce fichier.</p> <p><b>Remarque</b></p> <p>Vous devez vous assurer que la structure de la sortie de la commande et du fichier existant est identique :</p> <ul style="list-style-type: none"> <li>• les mêmes champs</li> <li>• le même ordre des champs</li> <li>• les champs correspondants ont la même longueur</li> <li>• les champs correspondants ont le même type de données</li> </ul> <p>Analytics ajoute la sortie à un fichier existant sans tenir compte de sa structure, ce qui peut désordonner les données si la structure de la sortie et du fichier existant ne correspond pas.</p>

## Exemples

### Générer un fichier texte avec 100 nombres aléatoires

Vous souhaitez extraire 100 fichiers papier au hasard parmi un ensemble de fichiers dont la numérotation s'étend de 10 000 à 20 000.

Vous pouvez utiliser la commande RANDOM pour générer un fichier texte avec 100 nombres aléatoires compris entre 10 000 et 20 000. Vous extrayez ensuite les fichiers papier qui correspondent aux nombres

aléatoires. Les nombres sont organisés dans 10 colonnes, sont uniques et sont triés dans l'ordre croissant :

```
RANDOM NUMBER 100 SEED 45387 MINIMUM 10000 MAXIMUM 20000 COLUMNS 10 UNIQUE  
SORTED TO "Nombres_Aléatoires.txt"
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Algorithmes de nombres aléatoires

La commande RANDOM utilise l'algorithme de nombres aléatoires par défaut Analytics. Toutefois, à la différence de la commande SAMPLE, la commande RANDOM ne présente pas la possibilité d'utiliser l'algorithme de nombres aléatoires Mersenne-Twister.

# Commande RCOMMAND

Transmet une table Analytics à un script R externe sous forme de **cadre de données** et crée une nouvelle table dans le projet Analytics à l'aide de la sortie du script R externe.

## Syntaxe

```
RCOMMAND FIELDS champ<...n> RSCRIPT chemin_vers_script TO nom_table <IF test> <WHILE test> <FIRST plage|NEXT plage> <KEEPTITLE> <SEPARATOR caractère> <QUALIFIER caractère> <OPEN>
```

## Paramètres

Nom	Description
FIELDS <i>nom_champ</i> <... <i>n</i> >	<p>Champs ou expressions de la table Analytics source à inclure dans le cadre de données envoyé au script R.</p> <p>Selon l'édition d'Analytics que vous utilisez, il se peut que vous rencontriez des erreurs lors de l'envoi de données contenant certains caractères spéciaux vers R :</p> <ul style="list-style-type: none"> <li>◦ <b>non Unicode</b> : "\"</li> <li>◦ <b>Unicode</b> : "ÿ" ou "Ŝ"</li> <li>◦ <b>Les deux</b> : boîtes de dialogue écrivant des caractères comme des blocs, des carrés noirs et des barres verticales discontinues</li> </ul> <p><b>Remarque</b></p> <p>Les données de langues mixtes ne sont pas prises en charge, comme une table contenant à la fois des caractères japonais et des caractères chinois.</p>
RSCRIPT <i>chemin_vers_script</i>	Chemin d'accès complet ou relatif au script R sur le système de fichiers. Entourez <i>chemin_vers_script</i> de guillemets.
TO <i>nom_table</i>	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li>◦ <b><i>nom_table</i></b> : enregistre les résultats dans une table Analytics</li> </ul> <p>Indiquez <i>nom_table</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple : TO "Sortie.FIL"</p> <p>Par défaut, le fichier de données de table (.FIL) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.FIL"</li> <li>• TO "Résultats\Sortie.FIL"</li> </ul>

Nom	Description
	<p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alphanumériques, sans l'extension .FIL. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p> <p>La table de sortie est créée à partir du cadre de données ou de la matrice que le script R renvoie.</p>
IF <i>test</i> optionnel	Condition devant être remplie pour que l'enregistrement actuel soit traité. Le cadre de données transmis au script R contient uniquement les enregistrements qui remplissent la condition.
WHILE <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.</p> <p><b>Remarque</b></p> <p>Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.</p>
FIRST <i>plage</i>   NEXT <i>plage</i> optionnel	<p>Le nombre d'enregistrements à traiter :</p> <ul style="list-style-type: none"> <li>◦ <b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li>◦ <b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> </ul> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter.</p> <p>Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p> <p><b>Attention</b></p> <p>Dans la version actuelle, il existe un problème connu avec NEXT lors de l'exécution de RCOMMAND. Éviter d'utiliser cette option comme référence d'enregistrement peut rétablir la valeur sur le premier enregistrement quel que soit l'enregistrement sélectionné.</p>
KEEPTITLE optionnel	<p>Traitez la première ligne de données en tant que noms de champs plutôt que comme données. Si vous ignorez ce paramètre, les noms des champs génériques sont utilisés.</p> <p>Cette option est requise si vous souhaitez récupérer les données à l'aide des noms des colonnes dans le script R.</p>
SEPARATOR <i>caractère</i> optionnel	<p>Caractère à utiliser comme séparateur de champs. Vous devez indiquer le caractère sous forme de chaîne entre guillemets.</p> <p>Le caractère par défaut est une virgule.</p> <p><b>Remarque</b></p> <p>Évitez d'utiliser des caractères qui apparaissent dans les champs d'entrée. Si le caractère SEPARATOR apparaît dans les données d'introduction, les résultats peuvent être affectés.</p>

Nom	Description
QUALIFIER <i>caractère</i> optionnel	<p>Caractère à utiliser en tant qu'identificateur de texte pour renvoyer à la ligne et identifier des valeurs de champs. Vous devez indiquer le caractère sous forme de chaîne entre guillemets.</p> <p>Le caractère par défaut est un guillemet double.</p> <p><b>Remarque</b> Évitez d'utiliser des caractères qui apparaissent dans les champs d'entrée. Si le caractère QUALIFIER apparaît dans les données d'introduction, les résultats peuvent être affectés.</p>
OPEN optionnel	Ouvre la table créée par la commande après l'exécution de la commande. Valide uniquement lorsque la commande crée une table de sortie.

## Exemples

### Faire fonctionner R (Hello world)

Vous créez un script « hello world » pour tester votre connexion entre Analytics et R :

### Commande Analytics

```
RCOMMAND FIELDS "Bonjour"; ", le monde !" TO "r_résultat" RSCRIPT "C:\scripts\r_scripts\analyse.r"
```

### Script R (analyse.r)

```
srcTable<-acl.readData()

# créer une table à renvoyer dans ACL
output<-data.frame(
  c(srcTable[1,1]),
  c(srcTable[1,2])
)

# ajouter les noms des colonnes et renvoyer la table dans ACL
colnames(output) <- c("Bonjour";"Objet")
acl.output<-output
```

### Accès aux données des champs à l'aide des coordonnées des lignes et des colonnes

Vous envoyez un certain nombre de champs facture à un script R pour les faire analyser en dehors

d'Analytics :

## Commande Analytics

```
RCOMMAND FIELDS Code_Département Montant_Facture Date_Facture Numéro_Facture
Numéro_fournisseur TO "r_résultat" RSCRIPT "C:\scripts\r_scripts\analyse.r"
```

## Script R (analyse.r)

```
# Récupère le numéro de facture de la deuxième ligne du cadre de données dans le script R
srcTable<-acl.readData()
srcTable[2,4]
```

## Accès aux données des champs à l'aide des noms de colonnes

Vous envoyez un certain nombre de champs facture à un script R pour les faire analyser en dehors d'Analytics. Vous utilisez l'option KEeptitle pour que les colonnes puissent être référencées par nom dans R :

## Commande Analytics

```
RCOMMAND FIELDS Code_Département Montant_Facture Numéro_Facture TO "r_résultat"
RSCRIPT "C:\scripts\r_scripts\analyse.r" KEeptitle
```

## Script R (analyse.r)

```
# Récupère le numéro de facture de la deuxième ligne du cadre de données dans le script R
srcTable<-acl.readData()
srcTable["2","Numéro_Facture"]
```

## Envoi des enregistrements de factures dépassant une valeur de 1000,00 au script R

Vous envoyez un certain nombre de champs facture à un script R pour les faire analyser en dehors d'Analytics. Vous utilisez IF pour limiter le nombre d'enregistrements envoyés à R. Seuls les enregistrements dont le montant des factures dépasse 1000,00 sont envoyés :

## Commande Analytics

```
RCOMMAND FIELDS Code_Département Montant_Facture Numéro_Facture TO "r_résultat" IF Mon-
tant_Facture > 1000,00 RSCRIPT "C:\scripts\r_scripts\analyse.r" KEeptitle
```

## Script R (analyse.r)

```
# Récupère le numéro de facture de la deuxième ligne du cadre de données dans le script R
srcTable<-acl.readData()
srcTable["2","Numéro_Facture"]
```

## Envoi des enregistrements des factures et renvoi des montants de facture multipliés

Vous envoyez un certain nombre de champs facture à un script R pour les faire analyser en dehors d'Analytics. Le script R effectue une seule action par rapport à chaque cellule dans la colonne nommée :

### Commande Analytics

```
RCOMMAND FIELDS Code_Département Montant_Facture Numéro_Facture TO "r_résultat"
RSCRIPT "C:\scripts\r_scripts\analyse.r" KEeptITLE
```

## Script R (analyse.r)

```
# Renvoie une section de table ACL avec la valeur doublée
srcTable<-acl.readData()
acl.output<-srcTable["Montant_Facture"] * 2
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Référence à des données Analytics dans le script R

La table Analytics est transmise au script sous forme de **cadre de données** R. Les cadres de données sont des objets de données tabulaires pouvant contenir des colonnes de différents modes - ou types - de données.

Pour utiliser le cadre de données créé par Analytics dans un script R, appelez la fonction `acl.readData()` et stockez le cadre de données renvoyé dans une variable :

```
# stocke la table Analytics dans un cadre de données appelé maTable qui peut être référencé tout au long du script
maTable<-acl.readData()
```



Pour extraire les données d'une cellule dans le cadre de données, vous pouvez utiliser l'une des approches suivantes :

- Utilisation des coordonnées de ligne et de colonne :

```
# Extrait la valeur dans la première ligne et la deuxième colonne du cadre de données
maTable[1,2]
```

#### Remarque

Les coordonnées dépendent de l'ordre des champs indiqué dans la commande et non du format de table ou de la vue ouverte.

- À l'aide des noms de ligne et de colonne :

```
# Extrait la valeur de la première ligne et de la colonne « monTitreColonne » du cadre de données
maTable["1";"monTitreColonne"]
```

Vous devez indiquer l'option **KEEPTITLE** de la commande pour utiliser les noms de table.

Les lignes sont intitulées « 1 », « 2 », « 3 » et incrémentées en conséquence. Vous pouvez également utiliser une combinaison de noms et de coordonnées.

## Retransmission de données vers Analytics

Pour renvoyer un cadre de données ou une matrice dans Analytics et créer une nouvelle table, utilisez la syntaxe suivante :

```
# Retransmet le cadre de données de maNouvelleTable à Analytics pour créer une nouvelle table
acl.output<-maNouvelleTable
```

#### Remarque

Vous devez renvoyer un cadre de données ou une matrice à Analytics lorsque le script R se termine. Assurez-vous que les colonnes du cadre de données ou de la matrice contiennent uniquement des valeurs atomiques et non des listes, des matrices, des tableaux ou des objets non atomiques. Si les valeurs ne peuvent pas être traduites en types de données Analytics, la commande échoue.

## Mappages des types de données

Les types de données Analytics sont traduits en types de données R à l'aide d'un processus de traduction entre le projet Analytics et le script R :

Type de données Analytics	Type(s) de données R
Logique	Logique

Type de données Analytics	Type(s) de données R
Numérique	Numérique
Caractère	Caractère
DateHeure	Date, POSIXct, POSIXlt

## Limitation des performances et des tailles de fichiers

La durée d'exécution du script R et du traitement des données renvoyées est rallongée pour les données d'entrée supérieures à 1 Go. R ne prend pas en charge les fichiers d'entrée de 2 Go ou plus.

Le nombre d'enregistrements envoyés dans R influence aussi les performances. Dans le cas de deux tables présentant une taille de fichier identique mais un nombre d'enregistrements différent, le traitement de la table contenant le moins d'enregistrements sera plus rapide.

## Gestion de données de type caractères multi-octets

Si vous envoyez des données à R dans un jeu de caractères multioctets, comme le chinois, vous devez définir le système local de manière approprié dans votre script R. Pour réussir l'envoi d'une table contenant des données multioctets à R, la première ligne du script R doit contenir la fonction suivante :

```
# Exemple qui définit les paramètres régionaux en chinois
Sys.setlocale("LC_ALL","Chinois")
```

Pour obtenir plus d'informations sur `Sys.setlocale()`, voir la documentation R.

## Fichier trace R

Analytics enregistre les messages en langage R dans un fichier `aclrlang.log` dans le dossier de projet. Utilisez ce fichier trace pour le débogage des erreurs R

### Astuce

Ce fichier trace se trouve dans le dossier de résultats des tâches analytiques d'Analytics Exchange.

## Exécution de scripts R sur AX Serveur

Si vous rédigez une App Analyse à exécuter sur AX Serveur et que vous souhaitez travailler sur des scripts R externes :

1. Chargez le fichier sous forme de fichier associé à l'App Analyse.
2. Utilisez la balise d'analyse FILE pour identifier le(s) fichier(s).
3. Faites référence au(x) fichier(s) à l'aide du chemin d'accès relatif `./nom_fichier.r`.

**Remarque**

L'utilisation d'un fichier associé garantit que le compte du serveur d'application TomEE dispose des autorisations suffisantes pour accéder au fichier lors de l'exécution de R avec Analytics Exchange.

# Commande REFRESH

Met à jour les données d'une table Analytics depuis sa source de données associée.

## Syntaxe

```
REFRESH <nom_table> <PASSWORD num>
```

## Paramètres

Nom	Description
<i>nom_table</i> optionnel	Nom de la table Analytics à actualiser. Si vous n'indiquez pas de <i>nom_table</i> , la table ouverte est actualisée.
PASSWORD <i>num</i> optionnel	<p>Définition de mot de passe à utiliser.</p> <p>Vous n'utilisez pas PASSWORD <i>num</i> pour inviter à saisir ou pour spécifier un mot de passe réel. La définition de mot de passe renvoie à un mot de passe fourni ou défini auparavant à l'aide de la commande PASSWORD, de la commande SET PASSWORD ou de la balise d'outil d'analyse PASSWORD.</p> <p><i>num</i> correspond au numéro de la définition de mot de passe. Par exemple, si deux mots de passe ont été fournis ou définis auparavant dans un script, ou lors de la planification d'un outil d'analyse, PASSWORD 2 indique que le mot de passe n° 2 est utilisé.</p> <p>Pour plus d'informations sur la fourniture ou définition de mots de passe, consultez les rubriques :</p> <ul style="list-style-type: none"> <li>• "Commande PASSWORD" Page 373</li> <li>• "Commande SET" Page 435</li> <li>• <a href="#">Balise d'outil d'analyse PASSWORD</a></li> </ul> <p>Pour plus d'informations sur la fourniture ou définition de mots de passe, consultez les rubriques :</p> <ul style="list-style-type: none"> <li>• <a href="#">Commande PASSWORD</a></li> <li>• <a href="#">Commande SET</a></li> <li>• <a href="#">Balise d'outil d'analyse PASSWORD</a></li> </ul> <p><b>Remarque</b></p> <p>Le mot de passe est utilisé pour accéder au système de données source d'origine.</p> <p>Vous ne pouvez pas utiliser la commande REFRESH avec un mot de passe pour les sources de données sur fichier, à l'exception des PDF.</p>

# Exemples

## Actualisation d'une table sans mot de passe requis

Si un mot de passe n'est pas requis pour la source de données, spécifiez simplement la commande REFRESH et le nom de la table Analytics à actualiser.

```
REFRESH Factures
```

## Actualisation d'une table avec un mot de passe dans un script interactif

Si vous créez un script interactif, vous pouvez inviter l'utilisateur à saisir un mot de passe :

```
PASSWORD 1 "Saisissez votre mot de passe :"  
REFRESH Factures PASSWORD 1
```

Si vous actualisez une table ayant été importée à l'origine à partir d'une source de données protégée par un mot de passe à l'aide de la commande ACCESSDATA, l'invite de mot de passe est automatique et ne doit pas être spécifiée séparément :

```
REFRESH Factures
```

## Actualisation d'une table avec un mot de passe dans un script non interactif

Vous pouvez définir le mot de passe dans un script si vous ne souhaitez pas inviter l'utilisateur à saisir la valeur :

```
SET PASSWORD 1 TO "mot de passe"  
REFRESH Factures PASSWORD 1
```

L'inconvénient de cette méthode est que le mot de passe apparaît sous forme de texte en clair dans le script.

## Actualisation d'une table avec mot de passe dans un outil d'analyse AX Serveur

Si vous créez un outil d'analyse AX Serveur, vous pouvez inviter l'utilisateur à saisir le mot de passe lorsque l'outil d'analyse est planifié ou exécuté ponctuellement :

```
COMMENT  
//ANALYTIC Actualiser la table  
//PASSWORD 1 "Saisissez votre mot de passe :"
```

```
END  
REFRESH Factures PASSWORD 1
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Fonctionnement

La commande REFRESH met à jour le contenu d'une table en réexécutant la commande IMPORT ou ACCESSDATA, qui a été utilisée à l'origine pour définir et importer la table.

## REFRESH met à jour le contenu de la table uniquement

La commande REFRESH met uniquement à jour le contenu des champs existants dans une table Analytics. Elle ne peut pas mettre à jour un format de table Analytics.

Vous ne pouvez pas utiliser REFRESH si la structure des données source a changé : par exemple, si des champs ont été ajoutés ou supprimés. Vous devez réimporter les données.

## Sources de données prenant en charge l'actualisation

La commande REFRESH permet de mettre à jour le contenu d'une table Analytics créée à l'aide de l'une des commandes suivantes :

- IMPORT ACCESS
- IMPORT DELIMITED
- IMPORT EXCEL
- IMPORT ODBC (commande ODBC héritée)
- IMPORT PDF
- IMPORT PRINT
- IMPORT SAP
- IMPORT XBRL
- IMPORT XML
- ACCESSDATA (sources de données ODBC)

## REFRESH et ACCESSDATA

Les instructions suivantes s'appliquent lors de l'actualisation d'une table importée à partir d'une source de données ODBC à l'aide de la commande ACCESSDATA.

- **Ouvrir la table** : Si la table est ouverte lorsque vous l'actualisez, il vous faudra temporairement un espace disque égal à deux fois la taille de la table. Si votre espace disque est limité, fermez d'abord la table avant de l'actualiser.
- Les tables **Analytics 12** : ayant été importées à l'aide de la commande ACCESSDATA dans la version 12 d'Analytics ne sont pas actualisables, même si vous utilisez une version plus récente d'Analytics.

Si vous souhaitez pouvoir actualiser ces tables, ré-importez-les à l'aide d'Analytics 12.5 ou version ultérieure.

## REFRESH et mots de passe

Vous pouvez utiliser la commande REFRESH avec des sources de données protégées par mot de passe qui existent dans une base de données ou dans un service de données cloud.

Vous ne pouvez pas utiliser la commande REFRESH avec des sources de données sur fichiers protégées par mot de passe, comme des fichiers Excel. Les PDF protégés par mot de passe constituent la seule exception.

## REFRESH et la fenêtre App Analyse

N'utilisez pas la commande REFRESH dans les scripts que vous prévoyez d'exécuter dans la fenêtre App Analyse.

Selon la façon dont une table est importée, l'actualisation des données dans la table n'est pas prise en charge ou produit des résultats imprévisibles, en cas de tentative dans la fenêtre App Analyse.

Si vous souhaitez actualiser des données dans le cadre d'un script exécuté dans la fenêtre App Analyse, utilisez une commande IMPORT ou la commande ACCESSDATA, puis écrasez la table.

# Commande RENAME

Renomme un fichier ou un élément de projet Analytics.

## Syntaxe

```
RENAME type_élément nom <AS|TO> nouveau_nom <OK>
```

## Paramètres

Nom	Description
<i>type_élément nom</i>	<p>Le type et le nom de l'élément de projet ou du fichier à renommer.</p> <p><b>Remarque</b></p> <p>Dans la plupart des cas, vous ne pouvez pas renommer un élément ou un fichier actif, ouvert ou en cours d'utilisation.</p> <p>Spécifiez l'un des types valides suivants :</p> <ul style="list-style-type: none"> <li>○ <b>FIELD</b> : champ de données physiques, champ calculé ou variable <ul style="list-style-type: none"> <li>• La table contenant le champ doit être ouverte. Toutefois, la vue active ne peut pas inclure le champ.</li> <li>• Vous ne pouvez pas renommer un champ référencé par un champ calculé.</li> </ul> </li> <li>○ <b>FORMAT</b> : table Analytics</li> <li>○ <b>INDEX</b> : index</li> <li>○ <b>REPORT</b> : un rapport ou une vue</li> <li>○ <b>WORKSPACE</b> : environnement de travail</li> <li>○ <b>SCRIPT (ou BATCH)</b> : script</li> <li>○ <b>DATA</b> : fichier de données Analytics (.fil)</li> <li>○ <b>FILE</b> : fichier de données dans le système de fichiers</li> <li>○ <b>LOG</b> : fichier journal Analytics (.log)</li> <li>○ <b>TEXT</b> : fichier texte</li> </ul>
AS TO <i>nouveau_nom</i>	<p>Nouveau nom de l'élément de projet ou du fichier.</p> <p><b>Remarque</b></p> <p>Les limitations de longueur s'appliquent à la plupart des noms d'élément de projet Analytics. Pour plus d'informations, consultez la rubrique <a href="#">Limitations de caractères et de tailles dans Analytics</a>.</p>
OK optionnel	<p>Supprime ou écrase des éléments sans demander de confirmation.</p>



# Exemples

## Renommer un champ

Vous devez renommer le champ **NumProd** en **NuméroProduit**. Vous utilisez OK pour effectuer l'action sans confirmation supplémentaire :

```
OPEN Inventaire  
RENAME FIELD NumProd AS NuméroProduit OK
```

# Commande REPORT

Met en forme et génère un rapport à partir de la table Analytics ouverte.

## Syntaxe

```
REPORT <ON champ_fractionnement <PAGE> <NODUPS> <WIDTH caractères> <AS nom_affichage>> <...n> FIELD autres_champs <WIDTH caractères> <AS nom_affichage> <...n>
<SUPPRESS> <NOZEROS> <LINE n autres_champs> <PRESORT <champ_tri>> <...n>
<SUMMARIZED> <SKIP n> <EOF> <TO {SCREEN|PRINT|nom_fichier <HTML>}> <IF >test>
<WHILE test> <FIRST plage|NEXT plage> <HEADER texte_en-tête> <FOOTER texte_pied_page>
<APPEND>
```

## Paramètres

Nom	Description
ON <i>champ_fractionnement</i> PAGE NODUPS WIDTH <i>caractères</i> AS <i>nom_affichage</i> <...n> optionnel	<p>Le ou les champs de type caractère utilisés pour diviser le rapport en sections.</p> <p>Une nouvelle section de rapport et un nouveau sous-total sont créés à chaque fois que la valeur de <i>champ_fractionnement</i> change. Diviser les rapports en sections peut faciliter leur lecture rapide.</p> <ul style="list-style-type: none"> <li>◦ <b><i>champ_fractionnement</i></b> : champ à utiliser comme champ de fractionnement</li> </ul> <p>Pour exécuter un rapport à partir d'une vue (DO REPORT), le champ de fractionnement doit se trouver être le champ caractère le plus à gauche dans la vue.</p> <ul style="list-style-type: none"> <li>◦ <b>PAGE</b> : insère un saut de page lorsque la valeur champ_fractionnement diffère</li> <li>◦ <b>NODUPS</b> : supprime les valeurs d'affichage dupliquées dans le champ de fractionnement</li> </ul> <p>Par exemple, si le nom d'un client est répertorié pour chaque enregistrement de facture, vous pouvez rendre le rapport plus lisible en indiquant seulement la première occurrence du nom du client.</p> <ul style="list-style-type: none"> <li>◦ <b>WIDTH <i>caractères</i></b> : longueur de sortie du champ en nombre de caractères</li> <li>◦ <b>AS <i>nom_affichage</i></b> : nom d'affichage (titre de la colonne) pour le champ dans le rapport.</li> </ul> <p>Indiquez le <i>nom_affichage</i> sous forme de chaîne entre guillemets. Utilisez un point-virgule (;) entre les mots si vous souhaitez insérer un saut de ligne dans le titre de la colonne. Si vous souhaitez que le nom d'affichage soit identique au nom du champ ou à un nom d'affichage existant dans la table source, n'utilisez pas AS.</p> <p><b>Remarque</b></p> <p>Vous devez spécifier ON pour utiliser <i>champ_fractionnement</i>, PAGE, NODUPS ou PRESORT.</p>

Nom	Description
FIELD <i>autres_champs</i> WIDTH <i>caractères</i> AS <i>nom_affichage</i> <... <i>n</i> >	<p>Champs à inclure dans le rapport.</p> <ul style="list-style-type: none"> <li>◦ <b>WIDTH <i>caractères</i></b> : longueur de sortie du champ en nombre de caractères</li> <li>◦ <b>AS <i>nom_affichage</i></b> : nom d'affichage (titre de la colonne) pour le champ dans le rapport.</li> </ul> <p>Indiquez le <i>nom_affichage</i> sous forme de chaîne entre guillemets. Utilisez un point-virgule (;) entre les mots si vous souhaitez insérer un saut de ligne dans le titre de la colonne. Si vous souhaitez que le nom d'affichage soit identique au nom du champ ou à un nom d'affichage existant dans la table source, n'utilisez pas AS.</p> <p>Les mots-clés SUBTOTAL et ACCUMULATE sont des synonymes de FIELD et sont devenus obsolètes. Tous les champs numériques sont sous-totalisés automatiquement.</p> <p><b>Remarque</b> Les champs de fractionnement sont inclus automatiquement dans le rapport et ne doivent être spécifiés sous <i>autres_champs</i>.</p>
SUPPRESS optionnel	Exclut les lignes de détail vides du rapport.
NOZEROS optionnel	Substitue des valeurs vides aux valeurs nulles dans le champ. Par exemple, si, dans un rapport, un champ contient un grand nombre de zéros, la lecture du rapport est facilitée si ce dernier n'affiche que les valeurs non nulles.
LINE <i>n autres_champs</i> optionnel	Spécifie le nombre de lignes de sortie dans la colonne et les champs s'affichent sur le numéro de ligne <i>n</i> . Si aucune valeur n'est indiquée, la colonne comprend par défaut une seule ligne. La valeur <i>n</i> doit être comprise entre 2 et 60 inclus. Les titres des colonnes du rapport sont déterminés uniquement par les champs de la première ligne. <i>autres_champs</i> indique des expressions ou des champs appropriés pour le rapport.
PRESORT <i>champ_tri</i> <... <i>n</i> > optionnel	<ul style="list-style-type: none"> <li>◦ Trie <i>champ_fractionnement</i>, si un ou plusieurs champs de fractionnement sont spécifiés.</li> <li>◦ Trie <i>champ_tri</i>, si un ou plusieurs champs de tri sont spécifiés.</li> </ul> <p>PRESORT ne trie pas les champs répertoriés sous <i>autres_champs</i> sauf s'ils sont aussi répertoriés sous <i>champ_tri</i>.</p>
SUMMARIZED optionnel	Génère un rapport contenant uniquement des sous-totaux et des totaux, sans lignes de détail. Les sous-totaux sont générés pour les valeurs de l'unique champ de fractionnement. Si SUMMARIZED n'est pas spécifiée, Analytics génère un rapport contenant des lignes de détail, ainsi que le sous-total de chaque champ de fractionnement clé spécifié.
SKIP <i>n</i> optionnel	Insère des lignes vides entre les lignes de détail dans le rapport. <i>n</i> doit être un nombre entier indiquant le nombre de lignes à insérer. Par exemple, SKIP 1 génère un rapport avec un espacement double.
EOF	Réexécutez la commande une fois la fin du fichier atteinte.

Nom	Description
optionnel	Cela garantit que le dernier enregistrement de la table est traité quand il se trouve à l'intérieur d'une commande GROUP. Utilisez le paramètre de traitement de fin de fichier uniquement si tous les champs sont calculés en se référant à des enregistrements antérieurs.
TO SCREEN   PRINT  <i>nom_fichier</i> <HTML> optionnel	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li>◦ <b>SCREEN</b> : affiche les résultats dans la zone d'affichage d'Analytics.</li> <li>◦ <b><i>nom_fichier</i></b> : enregistre les résultats dans un fichier</li> </ul> <p>Indiquez <i>nom_fichier</i> sous forme de chaîne entre guillemets avec l'extension de fichier appropriée. Par exemple : TO "Sortie.TXT"</p> <p>Par défaut, le fichier est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.TXT"</li> <li>• TO "Résultats\Sortie.TXT"</li> </ul> <li>◦ <b>PRINT</b> : envoie les résultats vers l'imprimante par défaut</li> <p>Par défaut, les rapports rendus dans un fichier sont enregistrés en tant que fichiers texte ASCII. Indiquez HTML si vous souhaitez sortir le rapport sous forme de fichier HTML (.htm).</p> <p>Si vous ignorez TO, le rapport sort à l'écran.</p>
IF <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.</p> <p><b>Remarque</b></p> <p>La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p>
WHILE <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.</p> <p><b>Remarque</b></p> <p>Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.</p>
FIRST <i>plage</i>   NEXT <i>plage</i> optionnel	<p>Le nombre d'enregistrements à traiter :</p> <ul style="list-style-type: none"> <li>◦ <b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li>◦ <b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> </ul> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter.</p> <p>Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p>
HEADER <i>texte_en-tête</i> optionnel	<p>Texte à insérer en haut de chaque page d'un rapport.</p> <p><i>texte_en-tête</i> doit être indiqué sous forme de chaîne entre guillemets. La valeur rem-</p>

Nom	Description
	place la variable système Analytics HEADER.
FOOTER <i>texte_pied_page</i> optionnel	Texte à insérer en bas de chaque page d'un rapport. <i>texte_pied_page</i> doit être indiqué sous forme de chaîne entre guillemets. La valeur remplace la variable système Analytics FOOTER.
APPEND optionnel	<p>Ajoute la sortie de commande à la fin d'un fichier existant au lieu de remplacer ce fichier.</p> <p><b>Remarque</b></p> <p>Vous devez vous assurer que la structure de la sortie de la commande et du fichier existant est identique :</p> <ul style="list-style-type: none"> <li>• les mêmes champs</li> <li>• le même ordre des champs</li> <li>• les champs correspondants ont la même longueur</li> <li>• les champs correspondants ont le même type de données</li> </ul> <p>Analytics ajoute la sortie à un fichier existant sans tenir compte de sa structure, ce qui peut désordonner les données si la structure de la sortie et du fichier existant ne correspond pas.</p>

## Exemples

### Génération d'un rapport HTML

Vous générez un rapport depuis la table **Ar** puis vous le sortez sous forme de fichier HTML :

```
OPEN Comptes_clients
REPORT ON No FIELD Montant Type Échéance TO "C:\Reports\AR.htm" HTML
```

# Commande RETRIEVE

Récupère les résultats d'une requête *Direct Link* soumise pour un traitement en arrière-plan.

## Syntaxe

```
RETRIEVE nom_table PASSWORD num
```

## Paramètres

Nom	Description
<i>nom_table</i>	Nom de la table initialement créée dans Analytics par la requête Direct Link. La table doit déjà exister avant que vous puissiez utiliser RETRIEVE.
PASSWORD <i>num</i>	<p>Définition de mot de passe à utiliser.</p> <p>Vous n'utilisez pas PASSWORD <i>num</i> pour inviter à saisir ou pour spécifier un mot de passe réel. La définition de mot de passe renvoie à un mot de passe fourni ou défini auparavant à l'aide de la commande PASSWORD, de la commande SET PASSWORD ou de la balise d'outil d'analyse PASSWORD.</p> <p><i>num</i> correspond au numéro de la définition de mot de passe. Par exemple, si deux mots de passe ont été fournis ou définis auparavant dans un script, ou lors de la planification d'un outil d'analyse, PASSWORD 2 indique que le mot de passe n° 2 est utilisé.</p> <p>Pour plus d'informations sur la fourniture ou définition de mots de passe, consultez les rubriques :</p> <ul style="list-style-type: none"> <li>• "Commande PASSWORD" Page 373</li> <li>• "Commande SET" Page 435</li> <li>• <a href="#">Balise d'outil d'analyse PASSWORD</a></li> </ul> <p>Pour plus d'informations sur la fourniture ou définition de mots de passe, consultez les rubriques :</p> <ul style="list-style-type: none"> <li>• <a href="#">Commande PASSWORD</a></li> <li>• <a href="#">Commande SET</a></li> <li>• <a href="#">Balise d'outil d'analyse PASSWORD</a></li> </ul> <p><b>Remarque</b></p> <p>Le mot de passe est utilisé pour accéder au système SAP.</p>

## Exemples

### Récupération du résultat de la requête en mode Fond

Vous définissez le mot de passe puis vous récupérez le résultat de la requête en mode Fond pour une table Analytics nommée **Données\_DD02T** :

```
SET PASSWORD 1 TO "mdp"  
RETRIEVE Données_DD02T PASSWORD 1
```

## Remarques

### Avant de commencer

Cette commande est prise en charge uniquement si Direct Link est installé et configuré.

# Commande SAMPLE

Extrait un échantillon d'enregistrements à l'aide de la méthode d'échantillonnage par enregistrement ou par unité monétaire.

Échantillonnage par enregistrement Échantillonnage par unité monétaire

## Syntaxe

### Remarque

Cette syntaxe n'inclut pas les paramètres de filtrage (instructions IF) ou d'étendue car l'application de ces options compromet la validité d'un échantillon.

## Méthode de sélection par intervalle fixe

```
SAMPLE <ON> RECORD INTERVAL valeur_intervalle <FIXED valeur_initiale>
{RECORD|FIELDS nom_champ <...n>} TO nom_table <OPEN> <APPEND> <LOCAL>
```

## Méthode de sélection par cellule

```
SAMPLE <ON> RECORD CELL INTERVAL valeur_intervalle <RANDOM valeur_départ>
{RECORD|FIELDS nom_champ <...n>} TO nom_table <OPEN> <APPEND> <MERSENNE_
TWISTER> <LOCAL>
```

## Méthode de sélection aléatoire

```
SAMPLE <ON> RECORD NUMBER taille_échantillon <RANDOM valeur_départ> <ORDER>
{RECORD|FIELDS nom_champ <...n>} TO nom_table <OPEN> <APPEND> <MERSENNE_
TWISTER> <LOCAL>
```

## Paramètres

### Remarque

N'incluez pas les séparateurs des milliers lorsque vous saisissez des valeurs.



Nom	Description
ON RECORD	Utilisez l'échantillonnage par enregistrement.
INTERVAL <i>valeur_intervalle</i> FIXED <i>valeur_initiale</i>   CELL INTERVAL <i>valeur_intervalle</i>   NUMBER <i>taille_échantillon</i>	<p><b>INTERVAL <i>valeur_intervalle</i> FIXED <i>valeur_initiale</i></b></p> <p>Utilisez la méthode de sélection par <b>intervalle fixe</b>.</p> <p>Un enregistrement initial est sélectionné, et toutes les sélections suivantes sont séparées d'un intervalle ou d'une distance fixe : par exemple, tous les 20 enregistrements à partir de la première sélection.</p> <ul style="list-style-type: none"> <li>◦ <b>INTERVAL <i>valeur_intervalle</i></b> : indiquez la valeur d'intervalle générée en calculant la taille de l'échantillon</li> <li>◦ <b>FIXED <i>valeur_initiale</i></b> : indiquez le numéro de l'enregistrement initial sélectionné</li> </ul> <p>Si vous indiquez une <i>valeur_initiale</i> de zéro ('0') ou si vous omettez FIXED, Analytics sélectionne de manière aléatoire l'enregistrement initial.</p> <p><b>CELL INTERVAL <i>valeur_intervalle</i></b></p> <p>Utilisez la méthode de sélection des <b>cellules</b>.</p> <p>Le jeu de données est divisé en plusieurs cellules ou groupes de taille égale, et un enregistrement est sélectionné de façon aléatoire dans chaque cellule.</p> <p>La <i>valeur_intervalle</i> détermine la taille de chaque cellule. Indiquez la valeur d'intervalle générée en calculant la taille de l'échantillon.</p> <p><b>NUMBER <i>taille_échantillon</i></b></p> <p>Utilisez la méthode de sélection <b>aléatoire</b>.</p> <p>Tous les enregistrements sont sélectionnés de manière aléatoire dans le jeu de données complet.</p> <p>Indiquez la taille d'échantillon générée en calculant la taille de l'échantillon.</p>
RANDOM <i>valeur_départ</i> optionnel	<p><b>Remarque</b></p> <p>Méthodes de sélection aléatoire et par cellule uniquement.</p> <p>Valeur de départ à utiliser pour lancer le générateur de nombres aléatoires dans Analytics.</p> <p>Si vous spécifiez une valeur de zéro (« 0 ») ou si vous omettez RANDOM, Analytics sélectionne la valeur de départ.</p>
ORDER optionnel	<p><b>Remarque</b></p> <p>Méthode de sélection aléatoire uniquement.</p> <p>Vous pouvez uniquement utiliser ORDER lorsque vous spécifiez FIELDS.</p> <p>Ajoute le champ ORDER aux résultats de la sortie.</p> <p>Ce champ affiche l'ordre dans lequel chaque enregistrement est sélectionné de façon aléatoire.</p>
RECORD   FIELDS <i>nom_champ</i> <...n>	<ul style="list-style-type: none"> <li>◦ <b>RECORD</b> : l'intégralité de l'enregistrement est incluse dans la table de sortie</li> <li>◦ <b>FIELDS</b> : les champs individuels, plutôt que l'enregistrement entier, sont inclus dans la table de sortie</li> </ul>

Nom	Description
	Indiquez le(s) champ(s) ou expressions à inclure. Si vous indiquez plusieurs champs, ils doivent être séparés par des espaces.
TO <i>nom_table</i>	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li>◦ <b>nom_table</b> : enregistre les résultats dans une table Analytics</li> </ul> <p>Indiquez <i>nom_table</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple : TO "Sortie.FIL"</p> <p>Par défaut, le fichier de données de table (.FIL) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.FIL"</li> <li>• TO "Résultats\Sortie.FIL"</li> </ul> <p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alpha-numériques, sans l'extension .FIL. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>
OPEN optionnel	Ouvre la table créée par la commande après l'exécution de la commande. Valide uniquement lorsque la commande crée une table de sortie.
APPEND optionnel	<p>Ajoute la sortie de commande à la fin d'un fichier existant au lieu de remplacer ce fichier.</p> <p><b>Remarque</b></p> <p>Vous devez vous assurer que la structure de la sortie de la commande et du fichier existant est identique :</p> <ul style="list-style-type: none"> <li>• les mêmes champs</li> <li>• le même ordre des champs</li> <li>• les champs correspondants ont la même longueur</li> <li>• les champs correspondants ont le même type de données</li> </ul> <p>Analytics ajoute la sortie à un fichier existant sans tenir compte de sa structure, ce qui peut désordonner les données si la structure de la sortie et du fichier existant ne correspond pas.</p>
MERSENNE_TWISTER optionnel	<p><b>Remarque</b></p> <p>Méthodes de sélection aléatoire et par cellule uniquement.</p> <p>Le générateur de nombres aléatoires d'Analytics utilise l'algorithme Mersenne-Twister. Si vous ignorez MERSENNE_TWISTER, l'algorithme d'Analytics par défaut est utilisé.</p> <p><b>Remarque</b></p> <p>Vous devez uniquement utiliser l'algorithme Analytics par défaut si vous avez besoin d'une rétrocompatibilité avec les scripts Analytics ou les résultats d'échantillonnage créés avant Analytics version 12.</p>
LOCAL	Enregistre le fichier de sortie au même emplacement que le projet Analytics.

Nom	Description
optionnel	<p><b>Remarque</b></p> <p>Applicable uniquement lorsque vous exécutez la commande sur une table de serveur avec un fichier de sortie qui est une table Analytics.</p>

## Exemples

### Extraire un échantillon d'enregistrements

Vous allez utiliser l'échantillonnage par enregistrement pour estimer le taux d'écart par rapport au contrôle prescrit dans un compte contenant des factures.

Après avoir calculé une taille d'échantillon statistiquement valide, vous voilà prêt à extraire l'échantillon. Vous allez utiliser la méthode de sélection aléatoire.

L'exemple ci-dessous :

- Échantillonne la table Analytics ouverte
- Utilise la méthode de sélection aléatoire avec une valeur de départ de 123456
- Spécifie une taille d'échantillon de 95 enregistrements
- Inclut uniquement les champs spécifiés dans la table de sortie
- Indique que le générateur de nombres aléatoires d'Analytics utilise l'algorithme Mersenne-Twister

```
SAMPLE ON RECORD RANDOM 123456 NUMBER 95 FIELDS RefNum CustNum Amount Date
Type TO "C_échantillon_enregistrement" OPEN MERSENNE_TWISTER
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Syntaxe

### Remarque

Cette syntaxe n'inclut pas les paramètres de filtrage (instructions IF) ou d'étendue car l'application de ces options compromet la validité d'un échantillon.

## Méthode de sélection par intervalle fixe

```
SAMPLE <ON> champ_numérique_mus INTERVAL valeur_intervalle <FIXED valeur_initiale>
<CUTOFF valeur_limite_couche_supérieure> <SUBSAMPLE> <NOREPLACEMENT>
{RECORD|FIELDS nom_champ <...n>} TO nom_table <OPEN> <APPEND> <LOCAL>
```

## Méthode de sélection par cellule

```
SAMPLE <ON> champ_numérique_mus CELL INTERVAL intervalle_valeur <CUTOFF valeur_
limite_couche_supérieure> <RANDOM valeur_départ> <NOREPLACEMENT> {RECORD|FIELDS
nom_champ <...n>} TO nom_table <OPEN> <APPEND> <MERSENNE_TWISTER> <LOCAL>
```

## Méthode de sélection aléatoire

```
SAMPLE <ON> champ_numérique_mus NUMBER taille_échantillon POPULATION valeur_absolue
<RANDOM valeur_départ> <SUBSAMPLE> <NOREPLACEMENT> {RECORD|FIELDS nom_
champ <...n>} TO nom_table <OPEN> <APPEND> <MERSENNE_TWISTER> <LOCAL>
```

## Paramètres

### Remarque

N'incluez pas les séparateurs des milliers lorsque vous saisissez des valeurs.

Nom	Description
ON <i>champ_numérique_mus</i>	Utilisez l'échantillonnage par unité monétaire (MUS). <i>champ_numérique_mus</i> est le champ ou l'expression numérique à utiliser comme base pour l'échantillonnage.
INTERVAL <i>valeur_intervalle</i> FIXED <i>valeur_initiale</i>   CELL INTERVAL <i>valeur_intervalle</i>   NUMBER <i>taille_échantillon</i> POPULATION <i>valeur_absolue</i>	<p><b>INTERVAL <i>valeur_intervalle</i> FIXED <i>valeur_initiale</i></b></p> <p>Utilisez la méthode de sélection par <b>intervalle fixe</b>.</p> <p>Une unité monétaire initiale est sélectionnée, et toutes les sélections suivantes sont séparées d'un intervalle ou d'une distance fixe : par exemple, toutes les 5000 unités monétaires à partir de la première sélection.</p> <ul style="list-style-type: none"> <li>○ <b>INTERVAL <i>valeur_intervalle</i></b> : indiquez la valeur d'intervalle générée en calculant la taille de l'échantillon</li> <li>○ <b>FIXED <i>valeur_initiale</i></b> : indiquez l'unité monétaire initiale sélectionnée</li> </ul> <p>Si vous indiquez une <i>valeur_initiale</i> de zéro ('0') ou si vous omettez FIXED, Analytics sélectionne de manière aléatoire l'unité monétaire initiale.</p>

Nom	Description
	<p><b>CELL INTERVAL <i>valeur_intervalle</i></b></p> <p>Utilisez la méthode de sélection des <b>cellules</b>.</p> <p>Le jeu de données est divisé en plusieurs cellules ou groupes de taille égale, et une unité monétaire est sélectionnée de façon aléatoire dans chaque cellule.</p> <p>La <i>valeur_intervalle</i> détermine la taille de chaque cellule. Indiquez la valeur d'intervalle générée en calculant la taille de l'échantillon.</p> <p><b>NUMBER <i>taille_échantillon</i> POPULATION <i>valeur_absolue</i></b></p> <p>Utilisez la méthode de sélection <b>aléatoire</b>.</p> <p>Toutes les unités monétaires sont sélectionnées de manière aléatoire dans le jeu de données complet.</p> <ul style="list-style-type: none"> <li>○ <b>NUMBER <i>taille_échantillon</i></b> : indiquez la taille d'échantillon générée en calculant la taille de l'échantillon.</li> <li>○ <b>POPULATION <i>valeur_absolue</i></b> : indiquez la valeur absolue totale du <i>champ_numérique_mus</i> ; elle correspond à la population dans laquelle l'échantillon sera sélectionné.</li> </ul>
<p>CUTOFF <i>valeur_limite_couche_supérieure</i> optionnel</p>	<p><b>Remarque</b></p> <p>Méthodes de sélection à intervalle fixe et par sélection des cellules uniquement.</p> <p>Valeur limite de la couche supérieure.</p> <p>Les montants dans <i>champ_numérique_mus</i> qui sont supérieurs ou égaux à la valeur limite sont sélectionnés et inclus automatiquement dans l'échantillon.</p> <p>Si vous ignorez CUTOFF, une valeur limite par défaut égale à la <i>valeur_intervalle</i> est utilisée.</p>
<p>RANDOM <i>valeur_départ</i> optionnel</p>	<p><b>Remarque</b></p> <p>Méthodes de sélection aléatoire et par cellule uniquement.</p> <p>Valeur de départ à utiliser pour lancer le générateur de nombres aléatoires dans Analytics.</p> <p>Si vous spécifiez une valeur de zéro (« 0 ») ou si vous omettez RANDOM, Analytics sélectionne la valeur de départ.</p>
<p>SUBSAMPLE optionnel</p>	<p><b>Remarque</b></p> <p>Vous pouvez uniquement utiliser SUBSAMPLE lorsque vous spécifiez FIELDS.</p> <p>Ajoute le champ SUBSAMPLE aux résultats de la sortie.</p> <p>Si chaque montant d'un champ de l'échantillon représente un total de plusieurs transactions distinctes et que vous souhaitez effectuer des procédures d'audit sur une seule transaction de chaque montant total échantillonné, vous pouvez utiliser les valeurs du champ SUBSAMPLE pour sélectionner de façon aléatoire les transactions individuelles.</p> <p>Pour plus d'informations, consultez la rubrique <a href="#">Réalisation d'un échantillonnage par</a></p>

Nom	Description
	<a href="#">unité monétaire.</a>
NOREPLACEMENT optionnel	<p>Le même enregistrement n'est pas sélectionné plusieurs fois. Par conséquent, l'échantillon peut contenir moins d'enregistrements que le nombre calculé par la commande SIZE.</p> <p>Si NOREPLACEMENT est omis ou si vous spécifiez REPLACEMENT, les enregistrements peuvent être sélectionnés plusieurs fois.</p>
ORDER optionnel	<p><b>Remarque</b></p> <p>Méthode de sélection aléatoire uniquement.</p> <p>Vous pouvez uniquement utiliser ORDER lorsque vous spécifiez FIELDS.</p> <p>Ajoute le champ ORDER aux résultats de la sortie.</p> <p>Ce champ affiche l'ordre dans lequel chaque enregistrement est sélectionné de façon aléatoire.</p>
RECORD   FIELDS <i>nom_champ</i> <...n>	<ul style="list-style-type: none"> <li>◦ <b>RECORD</b> : l'intégralité de l'enregistrement est incluse dans la table de sortie</li> <li>◦ <b>FIELDS</b> : les champs individuels, plutôt que l'enregistrement entier, sont inclus dans la table de sortie</li> </ul> <p>Indiquez le(s) champ(s) ou expressions à inclure. Si vous indiquez plusieurs champs, ils doivent être séparés par des espaces.</p>
TO <i>nom_table</i>	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li>◦ <b><i>nom_table</i></b> : enregistre les résultats dans une table Analytics</li> </ul> <p>Indiquez <i>nom_table</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple : TO "Sortie.FIL"</p> <p>Par défaut, le fichier de données de table (.FIL) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.FIL"</li> <li>• TO "Résultats\Sortie.FIL"</li> </ul> <p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alpha-numériques, sans l'extension .FIL. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>
OPEN optionnel	Ouvre la table créée par la commande après l'exécution de la commande. Valide uniquement lorsque la commande crée une table de sortie.
APPEND optionnel	Ajoute la sortie de commande à la fin d'un fichier existant au lieu de remplacer ce fichier.

Nom	Description
	<p><b>Remarque</b></p> <p>Vous devez vous assurer que la structure de la sortie de la commande et du fichier existant est identique :</p> <ul style="list-style-type: none"> <li>• les mêmes champs</li> <li>• le même ordre des champs</li> <li>• les champs correspondants ont la même longueur</li> <li>• les champs correspondants ont le même type de données</li> </ul> <p>Analytics ajoute la sortie à un fichier existant sans tenir compte de sa structure, ce qui peut désordonner les données si la structure de la sortie et du fichier existant ne correspond pas.</p>
MERSENNE_TWISTER optionnel	<p><b>Remarque</b></p> <p>Méthodes de sélection aléatoire et par cellule uniquement.</p> <p>Le générateur de nombres aléatoires d'Analytics utilise l'algorithme Mersenne-Twister. Si vous ignorez MERSENNE_TWISTER, l'algorithme d'Analytics par défaut est utilisé.</p> <p><b>Remarque</b></p> <p>Vous devez uniquement utiliser l'algorithme Analytics par défaut si vous avez besoin d'une rétrocompatibilité avec les scripts Analytics ou les résultats d'échantillonnage créés avant Analytics version 12.</p>
LOCAL optionnel	<p>Enregistre le fichier de sortie au même emplacement que le projet Analytics.</p> <p><b>Remarque</b></p> <p>Applicable uniquement lorsque vous exécutez la commande sur une table de serveur avec un fichier de sortie qui est une table Analytics.</p>

## Exemples

### Extraire un échantillon par unité monétaire

Vous allez utiliser un échantillonnage par unité monétaire pour estimer le montant total d'anomalie monétaire dans un compte contenant des factures.

Après avoir calculé une taille d'échantillon statistiquement valide, vous voilà prêt à extraire l'échantillon. Vous allez utiliser la méthode de sélection par intervalle fixe.

L'exemple ci-dessous :

- Échantillonne la table Analytics ouverte à partir d'un champ Montant des transactions
- Utilise la méthode de sélection à intervalle fixe avec une valeur d'intervalle de 6 283,33 €
- Indique que le premier enregistrement sélectionné contient la 100 000e unité monétaire (le nombre de centimes dans 1 000 €)
- Utilise une limite de couche supérieure de 5 000 €
- Inclut l'intégralité de l'enregistrement dans la table de sortie

SAMPLE ON Amount INTERVAL 6283.33 FIXED 1000.00 CUTOFF 5000.00 RECORD TO "C\_échantillon\_unité\_monétaire" OPEN

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).



# Commande SAVE

Copie une table Analytics et l'enregistre sous un nom différent ou enregistre un projet Analytics.

## Syntaxe

Pour créer une copie d'une table Analytics et l'enregistrer sous un nom différent :

```
SAVE nouvelle_table FORMAT table_ACL
```

Pour enregistrer les modifications apportées au projet actuel :

```
SAVE
```

## Paramètres

Nom	Description
<i>nouvelle_table</i>	<p>Le nom de la nouvelle table Analytics à créer et à enregistrer.</p> <p><b>Remarque</b> La longueur des noms des tables est limitée à 64 caractères alpha-numériques. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>
FORMAT <i>table_ACL</i>	Nom de la table Analytics existante. Utilisez le nom du format de table et non le nom d'un fichier de données associé.

## Exemples

### Création d'une nouvelle table à partir d'une table existante

Vous créez une nouvelle table appelée **Fournisseurs\_Mars**, basée sur la table existante **Fournisseurs\_maître**. **Fournisseurs\_Mars** peut alors être liée au fichier de données des fournisseurs de mars :

```
SAVE Fournisseurs_Mars FORMAT Fournisseurs_maître
```

# Remarques

## Fonctionnement

SAVE FORMAT produit un résultat similaire au copier-coller d'une table Analytics dans l'onglet **Vue globale** du **Navigateur**. Une nouvelle table Analytics est créée et associée au même fichier de données ou à la même source de données que la table d'origine.

Si besoin, vous pouvez lier la table nouvellement créée à une autre source de données.

## Utilisation de SAVE pour éviter les invites

À certains moments, Analytics vous invite à enregistrer les modifications apportées au projet actuel. Pour éviter toute interruption lors de l'exécution des scripts, vous pouvez utiliser la commande SAVE pour enregistrer les modifications avant qu'Analytics vous y invite.

# Commande SAVE LAYOUT

Enregistre un format de table Analytics dans un fichier de format de table externe (.layout) ou enregistre les métadonnées de format de table dans une table Analytics.

## Remarque

Avant la version 11 d'Analytics, les fichiers de format de table externes utilisaient une extension de fichier .fmt. Vous pouvez toujours enregistrer un fichier de format de table avec une extension .fmt en indiquant l'extension manuellement.

## Syntaxe

```
SAVE LAYOUT {FILE|TABLE} TO {nom_fichier|nom_table}
```

## Paramètres

Nom	Description
FILE   TABLE	<ul style="list-style-type: none"> <li>○ <b>FILE</b> : enregistrer un format de table Analytics dans un fichier de format de table externe (.layout)</li> <li>○ <b>TABLE</b> : enregistrer les métadonnées d'un format de table dans une table Analytics (.fil)</li> </ul>
TO <i>nom_fichier</i>   <i>nom_table</i>	<p>Nom du fichier de sortie et emplacement de la sortie :</p> <ul style="list-style-type: none"> <li>○ <b><i>nom_fichier</i></b> : nom du fichier .layout</li> </ul> <p>Indiquez le <i>nom_fichier</i> sous forme de chaîne entre guillemets. Par exemple : TO "Trans_Cf.layout".</p> <p>L'extension de fichier .layout est utilisée par défaut, il n'est donc pas obligatoire de le mentionner.</p> <p>Par défaut, le fichier est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Trans_Cf.layout"</li> <li>• TO "Formats de table\Trans_Cf.layout"</li> </ul> <p><b>Remarque</b></p> <p>La longueur du nom du format de table est limitée à 64 caractères alphanumériques, sans l'extension .layout, pour éviter au nom d'être tronqué lorsque le format de table est réimporté dans Analytics.</p> <p>Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>

Nom	Description
	<ul style="list-style-type: none"> <li>◦ <b>nom_table</b> :nom de la table Analytics et du fichier .fil</li> </ul> <p>Indiquez le <i>nom_table</i> sous forme de chaîne entre guillemets avec une extension de fichier Par exemple : TO "Métadonnées_format_trans_cf.fil".</p> <p>L'extension de fichier .fil est utilisée par défaut, il n'est donc pas obligatoire de le mentionner.</p> <p>Par défaut, le fichier de données de table (.fil) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Métadonnées_format_trans_cf.fil"</li> <li>• TO "Métadonnées format\Métadonnées_format_trans_cf.fil"</li> </ul> <p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alpha-numériques, sans l'extension .FIL. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>

## Exemples

### Enregistrement d'un format de table dans un fichier de format de table externe (.layout)

Dans les exemples suivants, le format de table utilisé par la table ouverte est enregistré dans un fichier de format de table externe intitulé **Trans\_Cf.layout** :

Ici, le fichier de format de table est enregistré dans le dossier de projet Analytics :

```
SAVE LAYOUT FILE TO Trans_Cf.layout
```

Ici, le fichier de format de table est enregistré dans le dossier indiqué :

```
SAVE LAYOUT FILE TO "C:\ACL_DATA\AP Audit 2013\Trans_Cf.layout"
```

### Enregistrement d'une copie des métadonnées de format de table dans une nouvelle table Analytics

Dans les exemples suivants, une copie des métadonnées du format de table utilisé par la table ouverte est enregistrée dans une nouvelle table Analytics intitulée **Métadonnées\_format\_trans\_cf**.

Ici, la nouvelle table Analytics est enregistrée dans le dossier de projet Analytics :

```
SAVE LAYOUT TABLE TO Métadonnées_format_trans_cf
```

Ici, la nouvelle table Analytics est enregistrée dans le dossier indiqué :

```
SAVE LAYOUT TABLE TO "C:\ACL_DATA\AP Audit 2013\Métadonnées_format_trans_cf"
```

## Remarques

### SAVE LAYOUT fichier et table

La commande SAVE LAYOUT est utilisée à deux fins différentes :

- La commande **FILE** : enregistre le format de table de la table Analytics ouverte dans un fichier de format de table externe portant une extension `.layout`
- La commande **TABLE** : extrait les métadonnées du format de table pour la table Analytics ouverte et les enregistre dans une nouvelle table Analytics.

### SAVE LAYOUT FILE

#### Fonctionnement

La commande SAVE LAYOUT FILE enregistre le format de table de la table Analytics ouverte dans un fichier de format de table externe portant une extension `.layout`.

Un format de table contient des métadonnées fournissant une interprétation structurée des données brutes dans le fichier de données source. Un format de table ne contient pas de données source lui-même.

#### Quand utiliser SAVE LAYOUT FILE

L'enregistrement d'un format de table sous un fichier `.layout` rend le format de table et ses métadonnées portables et réutilisables.

Le fichier `.layout` peut être importé dans un projet Analytics et être associé à un fichier de données source correspondant. Les éléments de données dans le fichier de données source doit correspondre aux définitions de champs spécifiées par les métadonnées du format de table.

Par exemple, vous pourriez enregistrer le format de table d'un fichier de transactions du mois de mars et l'associer à un fichier de données source contenant des transactions du mois d'avril, en supposant que la structure des données dans les fichiers de données source du mois de mars et du mois d'avril est identique. Utilisés de la sorte, les fichiers `.layout` peuvent vous éviter de devoir créer un tout nouveau format de table.

Pour plus d'informations sur la structure des tables Analytics, consultez l'aide d'Analytics.

# SAVE LAYOUT TABLE

## Fonctionnement

La commande SAVE LAYOUT TABLE extrait les métadonnées du format de table pour la table Analytics ouverte et les enregistre dans une nouvelle table Analytics.

La nouvelle table n'est pas le format de table lui-même, mais plutôt une table Analytics ordinaire qui contient un résumé des métadonnées du format de table pour la table d'origine. L'accès à ce résumé dans un script Analytics vous permet de prendre des décisions quant au script à partir de ces informations.

Pour chaque champ de la table d'origine, les éléments suivants des métadonnées du format de table sont extraits dans une nouvelle table.

### Remarque

Les noms des champs de la nouvelle table sont toujours générés en anglais indépendamment de la version localisée d'Analytics que vous utilisez.

Nom du champ dans la nouvelle table	Métadonnées d'un format de table
nom_champ	Nom du champ
type_données	Type de données du champ
catégorie	Catégorie de données du champ
position_départ	Position de début du champ
longueur_champ	Longueur du champ
décimales	Nombre de décimales du champ (champs numériques uniquement)
format	Format du champ (champs de type DateHeure et numérique uniquement)
titre_colonne	Autre titre de colonne du champ
largeur_colonne	Largeur de la colonne dans la vue

## Informations supplémentaires

Champs calculés	Les champs calculés sont inclus dans les métadonnées extraites, mais l'expression utilisée par le champ calculé, et ses éventuelles conditions, ne sont pas enregistrées. La position de départ, la longueur du champ et les places des décimales ne sont pas enregistrées non plus pour les champs calculés.
Champs associés	Les champs associés ne sont pas inclus, car ils ne font pas partie du format de table.
Filtres au niveau du	Les filtres au niveau des champs et les commentaires des champs ne sont pas inclus.

champ Commentaires des champs	
Autre titre de colonne Largeur de colonne	Les valeurs enregistrées pour l'autre titre de colonne et la largeur de colonne correspondent à celles indiquées dans le format de table, et non aux valeurs au niveau de la vue pouvant être indiquées pour les colonnes.

# Commande SAVE LOG

Enregistre l'intégralité de la trace des commandes, ou les entrées de la trace pour la session Analytics active, dans un fichier externe.

## Syntaxe

```
SAVE LOG <SESSION> AS nom_fichier {<ASCII>|HTML} <OK>
```

## Paramètres

Nom	Description
SESSION optionnel	Seules les entrées de la trace correspondant à la session Analytics active sont enregistrées.
AS <i>nom_fichier</i>	<p>Nom du fichier de sortie.</p> <p>Indiquez le <i>nom_fichier</i> sous forme de chaîne entre guillemets. Par exemple : AS "Trace des commandes". Vous pouvez indiquer une extension de fichier (.txt ou .htm ou .html), mais ce n'est pas obligatoire.</p> <p>Par défaut, le fichier est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>○ AS "C:\Trace des commandes.TXT"</li> <li>○ AS "Résultats\Trace des commandes.TXT"</li> </ul>
ASCII   HTML	<p>Le format du fichier de sortie :</p> <ul style="list-style-type: none"> <li>○ <b>ASCII (ou aucun mot-clé)</b> : fichier ASCII en texte brut.</li> <li>○ <b>HTML</b> : fichier HTML.</li> </ul>
OK optionnel	Si un fichier portant le même nom que <i>nom_fichier</i> existe déjà, il est remplacé sans confirmation.

## Exemples

### Enregistrer la trace des commandes à partir de l'analyse des fournisseurs

Vous avez effectué une analyse de données sur le fichier fournisseurs du mois de mars et vous souhaitez



enregistrer la trace des commandes qui y est associée dans le cadre de vos documents de travail.

Dans l'exemple suivant, les entrées de la session Analytics en cours sont enregistrées dans un fichier HTML. Si un fichier portant le même nom existe déjà, il est remplacé sans confirmation :

```
SAVE LOG SESSION AS "C:\Trace_Comptes_fournisseurs_Mars.htm" HTML OK
```

# Commande SAVE TABLELIST

Enregistre la liste de toutes les tables d'un projet Analytics dans une table Analytics ou dans un fichier CSV.

## Syntaxe

```
SAVE TABLELIST {FILE|TABLE} TO {nom_table|nom_fichier}
```

## Paramètres

Nom	Description
FILE   TABLE	<ul style="list-style-type: none"> <li>La commande <b>FILE</b> : enregistre la liste de table dans un fichier CSV (.csv)</li> <li>La commande <b>TABLE</b> : enregistre la liste de table dans une table Analytics</li> </ul>
TO <i>nom_table</i>   <i>nom_fichier</i>	<p>Emplacement d'enregistrement de la liste de table :</p> <ul style="list-style-type: none"> <li><b><i>nom_table</i></b> : nom de la table de sortie Analytics et du fichier .fil associé lors de l'utilisation de TABLE</li> </ul> <p>L'extension de fichier .fil est utilisée par défaut et elle ne doit pas être nécessairement spécifiée. La table est enregistrée dans le même répertoire que le projet Analytics et ne peut pas être enregistrée dans un autre dossier.</p> <p><b>Remarque</b></p> <p>La longueur des noms des tables Analytics est limitée à 64 caractères alphanumériques. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p> <ul style="list-style-type: none"> <li><b><i>nom_fichier</i></b> : nom du fichier .csv lors de l'utilisation de FILE</li> </ul> <p>L'extension de fichier .csv est utilisée par défaut et elle ne doit pas être nécessairement spécifiée. Vous pouvez indiquer un chemin d'accès absolu ou relatif pour enregistrer le fichier CSV dans un dossier existant différent du dossier contenant le projet Analytics. Si vous spécifiez un chemin d'accès relatif, il est relatif au répertoire de travail Analytics.</p> <p>Vous devez indiquer les valeurs sous forme de chaînes entre guillemets si elles contiennent des espaces.</p>

## Exemples

### Création d'une nouvelle table

Vous créez une nouvelle table dans le projet Analytics appelé **Table\_liste\_complète** :

```
SAVE TABLELIST TABLE TO Table_liste_complète
```

## Création d'un fichier CSV

Vous créez un nouveau fichier CSV appelé `Table_liste_complète.csv` dans le dossier `C:\ACL Data` :

```
SAVE TABLELIST FILE TO "C:\ACL Data\Table_liste_complète"
```

# Remarques

## Colonnes de sortie

La table Analytics ou le fichier CSV de sortie contient trois colonnes :

- **nom\_table** : nom du format de table Analytics
- **type** : indique si la table Analytics est une table locale ou une table de serveur
- **Chemin\_fichier\_données** : chemin d'accès complet au fichier de données source

# Commande SAVE WORKSPACE

Crée et enregistre un environnement de travail.

## Syntaxe

```
SAVE WORKSPACE nom_environnement{nom_champ<...n>}
```

## Paramètres

Nom	Description
<i>nom_environnement</i>	Le nom de l'environnement de travail à créer et à ajouter au projet Analytics actuel.
<i>nom_champ</i> <... <i>n</i> >	Le nom du champ à ajouter à l'environnement de travail. Vous pouvez inclure plusieurs noms de champ séparés par des espaces.

## Exemple

### Activation d'un environnement de travail

Vous créez un environnement de travail nommé **Marge\_inventaire** avec deux champs calculés de la table **Metaphor\_Inventaire\_2002**. Puis vous activez l'environnement de travail pour que les champs soient disponibles dans la table **Inventaire** :

```
OPEN Inventaire_2002_Metaphor
SAVE WORKSPACE Marge_inventaire Marge_unitaire_brute Marge_unitaire_pourcentage
OPEN Inventaire
ACTIVATE WORKSPACE Marge_inventaire OK
```

## Remarques

### Les noms de champ servant à créer les champs calculés doivent correspondre

Les noms de champ que vous utilisez dans les expressions créant un champ calculé enregistré dans un environnement de travail doivent correspondre aux noms de champ de la table qui utilise l'environnement

de travail.

Par exemple, si un environnement de travail contient le champ calculé  $Valeur = Prix\_vente * Quantit $ , la table active doit  galement contenir des champs appel s **Prix\_vente** et **Quantit **.

# Commande SEEK

Recherche, dans un champ caractère indexé, la première valeur qui correspond à la chaîne ou à l'expression de caractères spécifiée.

## Syntaxe

```
SEEK expression_recherchée
```

## Paramètres

Nom	Description
<i>expression_recherchée</i>	Expression de type caractère à rechercher. Vous pouvez utiliser une chaîne entre guillemets, une variable de type caractère ou une expression de type caractère valide. L' <i>expression_recherchée</i> respecte la casse et peut inclure des espaces de début, qui sont traités comme des caractères.

## Exemples

### Rechercher la première valeur d'un champ qui correspond à une variable de type caractère

Le champ Num\_Carte a été défini en tant que champ caractère et il est indexé dans l'ordre croissant.

L'exemple ci-dessous permet de trouver la première valeur du champ qui correspond exactement à la valeur contenue dans la variable *num\_carte\_fou* qui commence par celle-ci.

```
INDEX ON Num_Carte TO "NumCarte" OPEN
SET INDEX TO "NumCarte"
SEEK v_num_carte
```

### Rechercher la première valeur d'un champ qui correspond à une chaîne de caractères

Le champ Num\_Carte a été défini en tant que champ caractère et il est indexé dans l'ordre croissant.

L'exemple ci-dessous permet de trouver la première valeur du champ qui correspond exactement à "AB-123" ou qui commence par ce littéral de caractère :

```
INDEX ON Num_Carte TO "NumCarte" OPEN
SET INDEX TO "NumCarte"
SEEK "AB-123"
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Fonctionnement

Utilisez la commande SEEK pour accéder directement au premier enregistrement d'une table qui contient la valeur du paramètre *expression\_recherchée* dans le champ caractère indexé.

- **Si l'*expression recherchée* est trouvée** : le premier enregistrement correspondant de la table est sélectionné.
- **Si l'*expression recherchée* n'est pas trouvée** : le message « Aucune correspondance de clé n'a été trouvée » s'affiche et la table est placée au premier enregistrement présentant une valeur supérieure à l'expression recherchée.

Si aucune valeur du champ indexé n'est supérieure à l'expression recherchée, la table est placée sur le premier enregistrement.

## Index requis

Pour utiliser SEEK afin de rechercher un champ de type caractère, vous devez d'abord indexer le champ dans l'ordre croissant. Si plusieurs champs de type caractère sont indexés dans l'ordre croissant, seul le premier champ indiqué dans l'index est exploré.

La commande SEEK ne peut pas être utilisée pour rechercher dans des champs d'index qui ne soient pas de type caractère ni des champs caractère indexés dans l'ordre décroissant.

## Rapprochement partiel pris en charge

Le rapprochement partiel est pris en charge. L'expression de recherche peut être contenue par une valeur plus longue dans le champ indexé. Toutefois, l'expression recherchée doit apparaître au début du champ pour constituer un rapprochement.

La commande SEEK n'est pas affectée par l'option **Comparaisons exactes de caractères** (SET EXACT ON/OFF).

# Commande SEQUENCE

Détermine si un ou plusieurs champs d'une table Analytics sont classés par ordre séquentiel, et identifie les éléments hors séquence.

## Syntaxe

```
SEQUENCE <ON> {<FIELDS> champ <D> <...n> | <FIELDS> ALL} <UNFORMATTED>
<ERRORLIMIT n> <IF test> <WHILE test> <FIRST plage | NEXT plage> <TO {SCREEN | nom_
fichier | PRINT}> <APPEND> <HEADER texte_en_tête> <FOOTER texte_pied_page>
<PRESORT> <LOCAL> <ISOLOCALE code_paramètres_régionaux>
```

## Paramètres

Nom	Description
ON FIELDS <i>champ</i> D <...n>   FIELDS ALL	Expressions ou champs dans lesquels vérifier l'ordre séquentiel. Spécifiez ALL pour vérifier tous les champs de la table Analytics.  Incluez D pour trier le champ clé dans l'ordre décroissant. L'ordre de tri par défaut est croissant.
UNFORMATTED optionnel	Supprime les en-têtes et les sauts de page lorsque les résultats sont renvoyés dans un fichier.
ERRORLIMIT <i>n</i> optionnel	Le nombre d'erreurs toléré avant la fin de la commande. La valeur par défaut est 10.
IF <i>test</i> optionnel	Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.  <b>Remarque</b> La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).
WHILE <i>test</i> optionnel	Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.  <b>Remarque</b> Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.
FIRST <i>plage</i>   NEXT <i>plage</i>	Le nombre d'enregistrements à traiter :



Nom	Description
optionnel	<ul style="list-style-type: none"> <li>◦ <b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li>◦ <b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> </ul> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter.</p> <p>Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p>
TO SCREEN   <i>nom_fichier</i>   PRINT optionnel	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li>◦ <b>SCREEN</b> : affiche les résultats dans la zone d'affichage d'Analytics.</li> <li>◦ <b><i>nom_fichier</i></b> : enregistre les résultats dans un fichier</li> </ul> <p>Indiquez <i>nom_fichier</i> sous forme de chaîne entre guillemets avec l'extension de fichier appropriée. Par exemple : TO "Sortie.TXT"</p> <p>Par défaut, le fichier est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.TXT"</li> <li>• TO "Résultats\Sortie.TXT"</li> </ul> <li>◦ <b>PRINT</b> : envoie les résultats vers l'imprimante par défaut</li>
APPEND optionnel	<p>Ajoute la sortie de commande à la fin d'un fichier existant au lieu de remplacer ce fichier.</p> <p><b>Remarque</b></p> <p>Vous devez vous assurer que la structure de la sortie de la commande et du fichier existant est identique :</p> <ul style="list-style-type: none"> <li>• les mêmes champs</li> <li>• le même ordre des champs</li> <li>• les champs correspondants ont la même longueur</li> <li>• les champs correspondants ont le même type de données</li> </ul> <p>Analytics ajoute la sortie à un fichier existant sans tenir compte de sa structure, ce qui peut désordonner les données si la structure de la sortie et du fichier existant ne correspond pas.</p>
HEADER <i>texte_en-tête</i> optionnel	<p>Texte à insérer en haut de chaque page d'un rapport.</p> <p><i>texte_en-tête</i> doit être indiqué sous forme de chaîne entre guillemets. La valeur remplace la variable système Analytics HEADER.</p>
FOOTER <i>texte_pied_page</i> optionnel	<p>Texte à insérer en bas de chaque page d'un rapport.</p> <p><i>texte_pied_page</i> doit être indiqué sous forme de chaîne entre guillemets. La valeur remplace la variable système Analytics FOOTER.</p>
PRESORT optionnel	<p>Trie la table par le champ clé avant d'exécuter la commande.</p> <p><b>Remarque</b></p> <p>Vous ne pouvez pas utiliser PRESORT dans la commande GROUP.</p>
LOCAL	<p>Enregistre le fichier de sortie au même emplacement que le projet Analytics.</p>

Nom	Description
optionnel	<p><b>Remarque</b></p> <p>Applicable uniquement lorsque vous exécutez la commande sur une table de serveur avec un fichier de sortie qui est une table Analytics.</p>
ISOLOCALE <i>code_paramètres_régionaux</i> optionnel	<p><b>Remarque</b></p> <p>Applicable dans l'édition Unicode d'Analytics uniquement.</p> <p>Paramètre système local au format <i>langue_pays</i>. Par exemple, pour utiliser le français canadien, saisissez <i>fr_ca</i>.</p> <p>Utilisez les codes suivants :</p> <ul style="list-style-type: none"> <li>◦ <b>langue</b> : code linguistique de la norme ISO 639</li> <li>◦ <b>pays</b> : code pays de la norme ISO 3166</li> </ul> <p>Si vous n'indiquez pas de code pays, c'est le pays par défaut pour la langue qui est utilisé.</p> <p>Si vous n'utilisez pas ISOLOCALE, ce sont les paramètres système régionaux par défaut qui sont utilisés.</p>

## Variables de sortie Analytics

Nom	Contient
WRITE <i>n</i>	Nombre total des erreurs de séquence identifiées par la commande.

## Exemples

### Test de recherche des ID d'employés et des dates d'embauche hors séquence

Vous écrivez les erreurs de séquence identifiées dans les champs **IDEmployé** et **DateEmb** dans un fichier texte :

```
SEQUENCE ON IDEmployé DateEmb ERRORLIMIT 10 TO "ErreursSéquence.txt"
```

## Remarques

### Utilisation de SEQUENCE dans un GROUP

Si vous utilisez SEQUENCE dans une commande GROUP, la commande s'exécute pour éviter toute interférence avec le traitement du groupe, mais aucune autre erreur de séquence de données n'est signalée.

# Commande SET

Définit une option Analytics configurable.

## Remarque

La commande SET définit une option Analytics pour la durée de la session Analytics uniquement. Ce comportement s'applique que vous utilisiez la commande SET dans la ligne de commande d'Analytics ou dans un script Analytics.

Pour définir les options ACL afin qu'elles soient conservées entre différentes sessions Analytics, vous devez utiliser la boîte de dialogue **Options**. Pour plus d'informations, consultez la section [Configuration des options d'ACL](#).

## Syntaxe

Syntaxe	Exemples et remarques
SET BEEP <i>valeur</i>	<pre>SET BEEP 2</pre> <p>Indique le nombre de bips émis lorsque le traitement de la commande est terminé. Le paramètre <i>valeur</i> doit être compris entre 1 et 255.</p>
SET CENTURY <i>valeur</i>	<pre>SET CENTURY 40</pre> <p>Indique l'année de début du siècle pour les années à deux chiffres. Le paramètre <i>valeur</i> doit être entre 0 et 99. La définition de la valeur de début de siècle à 40 signifie que les années à deux chiffres 40 à 99 sont interprétées comme 1940 à 1999, et que les années à deux chiffres 00 à 39 sont interprétées comme 2000 à 2039.</p>
SET CLEAN {ON   OFF}	<pre>SET CLEAN ON</pre> <p>Lorsque cette option est activée, Analytics remplace les données de type caractère non valides par des espaces et les données numériques non valides par des zéros.</p>
SET DATE <TO> {0   1   2   chaîne}	<pre>SET DATE "AAAA/MM/JJ"</pre> <p>Indique le format d'affichage des dates dans Analytics, et la portion correspondant à la date des DateHeures, dans les vues, les rapports et les fichiers exportés.</p> <ul style="list-style-type: none"> <li>○ SET DATE 0 définit la date au format MM/JJ/AAAA</li> <li>○ SET DATE 1 définit la date au format MM/JJ/AA</li> <li>○ SET DATE 2 définit la date au format JJ/MM/AA</li> </ul>

Syntaxe	Exemples et remarques
	<ul style="list-style-type: none"> <li>◦ SET DATE "&lt;chaîne&gt;" définit la date au format personnalisé que vous indiquez</li> </ul> <p>Lorsque vous utilisez la commande SET DATE pour spécifier des formats de date personnalisés, vous devez utiliser J pour Jour, M pour Mois et A pour Année, même si vous avez spécifié des caractères de format de date différents dans la boîte de dialogue <b>Options</b>. Par exemple :</p> <pre style="border: 1px solid #ccc; padding: 5px; width: fit-content; margin: 10px auto;">SET DATE "JJ MMM AAAA"</pre>
SET DELETE_FILE {ON   OFF}	<pre style="border: 1px solid #ccc; padding: 5px; width: fit-content; margin-bottom: 10px;">SET DELETE_FILE ON</pre> <p>Paramètre par défaut : OFF</p> <p>Spécifiez ON pour supprimer automatiquement le fichier de données associé lorsque vous supprimez un format de table.</p> <p>Spécifiez OFF pour éviter la suppression du fichier de données associé lorsque vous supprimez un format de table.</p> <p>Vous devez inclure le caractère de soulignement ( _ ) dans DELETE_FILE.</p> <p>Indiquer SET DELETE_FILE, sans aucun paramètre, dans la ligne de commande permet d'obtenir un affichage indiquant si DELETE_FILE est actuellement activée ou désactivée.</p> <div style="border-left: 2px solid #c00; padding-left: 10px; margin-top: 10px;"> <p><b>Attention</b></p> <p>Redoublez de prudence lorsque vous activez cette option. Il se peut que ce soit le fichier de données d'origine qui soit supprimé avec la table !</p> <p>Les fichiers de données sont supprimés de manière définitive. Ils ne sont pas envoyés dans la Corbeille de Windows.</p> </div>
SET DESIGNATION <i>valeur</i>	<pre style="border: 1px solid #ccc; padding: 5px; width: fit-content; margin-bottom: 10px;">SET DESIGNATION "Produit par l'entreprise ABC"</pre> <p>Le paramètre <i>valeur</i> est une chaîne entre guillemets qui indique le libellé à afficher en haut de chaque page imprimée.</p>
SET ECHO {ON   NONE}	<pre style="border: 1px solid #ccc; padding: 5px; width: fit-content; margin-bottom: 10px;">SET ECHO NONE Commandes COM et résultats des scripts exclus de la trace. SET ECHO ON</pre> <p>Indiquez NONE pour arrêter l'écriture des commandes et des résultats dans les scripts vers la trace de commande Analytics. Indiquez ON pour reprendre l'écriture dans la trace.</p> <p>La commande SET ECHO s'applique uniquement à la journalisation des commandes et des résultats dans les scripts. Les commandes réalisées à travers l'interface utilisateur ou émises à partir de la ligne de commande, ainsi que tous résultats qu'elles produisent, sont toujours journalisées, quelle que soit la manière dont la commande ECHO est définie.</p> <p>Vous pouvez exécuter la commande SET ECHO NONE/ON dans un script ou à partir de la ligne de commande, mais quel que soit l'endroit où vous émettez la commande, cela affecte uniquement la journalisation des commandes et des résultats dans les scripts.</p>

Syntaxe	Exemples et remarques				
	<p>Indiquer SET ECHO, sans aucun paramètre, dans la ligne de commande permet d'obtenir un affichage indiquant si la journalisation des commandes et des résultats dans les scripts est actuellement activée ou désactivée.</p>				
<p>SET EXACT {ON   OFF}</p>	<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <p>SET EXACT ON</p> </div> <p>Paramètre par défaut : OFF</p> <p>Contrôle la façon dont Analytics compare les champs de type caractère, les expressions ou les valeurs littérales.</p> <p><b>Remarque</b></p> <p>Les espaces vides sont considérées comme des caractères.</p> <ul style="list-style-type: none"> <li>○ <b>SET EXACT est sur OFF</b> : Pour comparer deux chaînes de caractères de longueur inégale, Analytics utilise la plus petite chaîne de caractères. La comparaison commence par les caractères les plus à gauche et se déplace vers la droite. <ul style="list-style-type: none"> <li>Par exemple, la chaîne « AB » est égale à « AB » et elle aussi considérée comme égale à « ABC ».</li> </ul> </li> <li>○ <b>SET EXACT est sur ON</b> : les chaînes de comparaison doivent être exactement identiques pour constituer un rapprochement. Lors de la comparaison de deux chaînes de caractères de longueur inégale, Analytics complète la chaîne la plus courte avec des espaces de fin pour faire correspondre les deux chaînes avec la longueur de la chaîne la plus longue. <ul style="list-style-type: none"> <li>Par exemple, la chaîne « AB » est égale à « AB », mais elle n'est pas considérée comme égale à « ABC ».</li> </ul> </li> </ul> <p>Pour plus d'exemples illustrant SET EXACT, reportez-vous à « Comparaisons exactes de caractères » dans <a href="#">l'onglet Table (boîte de dialogue Options)</a>.</p> <p>Vous pouvez utiliser la fonction ALLTRIM( ) pour supprimer les espaces de début et de fin et vous assurer que seuls des caractères de type texte et des espaces internes sont comparés.</p> <p>Par exemple : ALLTRIM(" AB") = ALLTRIM("AB") est Vrai lorsque les valeurs sont renvoyées à la ligne avec ALLTRIM( ), mais est Faux sinon.</p> <p>Certaines commandes et fonctions d'Analytics sont affectées par SET EXACT et d'autres ne le sont pas :</p> <table border="1" data-bbox="483 1434 1443 1688" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">Affecté</th> <th style="width: 50%;">Non affecté</th> </tr> </thead> <tbody> <tr> <td style="vertical-align: top;"> <ul style="list-style-type: none"> <li>○ Commande LOCATE</li> <li>○ Fonction MATCH( )</li> <li>○ Fonction BETWEEN( )</li> </ul> </td> <td style="vertical-align: top;"> <ul style="list-style-type: none"> <li>○ Commande JOIN</li> <li>○ Commande DEFINE RELATION</li> <li>○ Fonction FIND( )</li> <li>○ Fonction FINDMULTI( )</li> </ul> </td> </tr> </tbody> </table>	Affecté	Non affecté	<ul style="list-style-type: none"> <li>○ Commande LOCATE</li> <li>○ Fonction MATCH( )</li> <li>○ Fonction BETWEEN( )</li> </ul>	<ul style="list-style-type: none"> <li>○ Commande JOIN</li> <li>○ Commande DEFINE RELATION</li> <li>○ Fonction FIND( )</li> <li>○ Fonction FINDMULTI( )</li> </ul>
Affecté	Non affecté				
<ul style="list-style-type: none"> <li>○ Commande LOCATE</li> <li>○ Fonction MATCH( )</li> <li>○ Fonction BETWEEN( )</li> </ul>	<ul style="list-style-type: none"> <li>○ Commande JOIN</li> <li>○ Commande DEFINE RELATION</li> <li>○ Fonction FIND( )</li> <li>○ Fonction FINDMULTI( )</li> </ul>				
<p>SET FILTER &lt;TO&gt; {test   nom_filtre}</p>	<div style="border: 1px solid #ccc; padding: 5px;"> <p>SET FILTER TO NumProd = "070104347"</p> </div>				

Syntaxe	Exemples et remarques
	<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;">SET FILTER TO FiltreNoProd</div> <p>Crée un filtre global (filtre de vue) sur la table ouverte et indique un test logique, ou le nom d'un filtre enregistré existant.</p> <p>Indiquer SET FILTER, sans paramètre, supprime des filtres de la table ouverte.</p>
SET FOLDER <i>chemin dossier</i>	<p>Indique le répertoire du projet Analytics dans l'onglet <b>Vue globale</b> pour la sortie de la commande. Le dossier de sortie par défaut est le dossier contenant la table active.</p> <p>Le chemin de type DOS indique le format <b>/nom de dossier/sous-dossier</b>, où la première barre oblique (/) indique le niveau racine de l'onglet <b>Vue globale</b>. Vous devez indiquer un chemin de fichier.</p> <ul style="list-style-type: none"> <li>◦ SET FOLDER /Tables/Résultats définit le dossier de sortie dans le sous-dossier Résultats. Si le sous-dossier Résultats n'existe pas, il est créé.</li> <li>◦ SET FOLDER / définit le dossier de sortie au niveau racine dans l'onglet <b>Vue globale</b></li> <li>◦ SET FOLDER définit le dossier de sortie sur les paramètres par défaut (le dossier de la table active)</li> </ul> <p>Le dossier de sortie reste tel que vous l'avez défini, jusqu'à sa redéfinition ou la fermeture du projet. À l'ouverture du projet, le dossier de sortie revient aux paramètres par défaut du dossier contenant la table active.</p>
SET FORMAT {ON   OFF}	<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;">SET FORMAT ON</div> <p>Paramètre par défaut : OFF</p> <p>Si vous utilisez le paramètre ON, Analytics affiche automatiquement les définitions de format de table et de champ calculé actuelles lorsque vous ouvrez une nouvelle table. Les résultats s'affichent dans la trace des commandes.</p>
SET FUZZYGROUPSIZE <TO> <i>num</i>	<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;">SET FUZZYGROUPSIZE TO 10</div> <p>Indique le nombre maximal d'éléments qui peuvent figurer dans un groupe de doublons approximatifs dans les résultats de sortie. Le paramètre <i>num</i> ne peut pas être inférieur à 2 ni supérieur à 100. La taille par défaut est 20. La taille indiquée demeure effective pendant la durée de la session Analytics.</p>
SET GRAPH <i>type</i>	<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;">SET GRAPH LINE</div> <p>Indique le type de graphe à utiliser pour tous les graphes générés ultérieurement. L'exécution des commandes doit être compatible avec le type de graphe indiqué. Par exemple, la commande BENFORD ne peut pas produire un graphique PIE2D ou PIE3D. Si un type de graphe incompatible est indiqué, le type par défaut est utilisé (BAR3D).</p> <p>Le paramètre <i>type</i> doit avoir l'une des valeurs suivantes :</p> <ul style="list-style-type: none"> <li>◦ PIE2D</li> </ul>

Syntaxe	Exemples et remarques
	<ul style="list-style-type: none"> <li>○ PIE3D</li> <li>○ BAR2D</li> <li>○ BAR3D : type de graphe par défaut.</li> <li>○ STACKED2D</li> <li>○ STACKED3D</li> <li>○ LAYERED</li> <li>○ LINE</li> <li>○ BENFORD : combine un graphe à barres 2D et un graphe linéaire 2D.</li> </ul>
SET HISTORY <TO> <i>valeur</i>	<pre>SET HISTORY TO 50</pre> <p>Indique le nombre maximal d'entrées historiques de table à conserver. Le paramètre <i>valeur</i> doit être compris entre 1 et 100.</p>
SET INDEX <TO> <i>valeur</i>	<pre>SET INDEX TO "CodeClient.INX"</pre> <p>Indique l'index à appliquer à la table active.</p>
SET LEARN <TO> <i>script</i>	<pre>SET LEARN TO EnrInventaire</pre> <p>Indique le nom du fichier de script utilisé par l'<b>enregistrement des scripts</b> pour enregistrer les commandes.</p>
SET LOG <TO> { <i>fichier</i>   OFF}	<pre>SET LOG TO "analyse.log"</pre> <pre>SET LOG OFF</pre> <p>La première commande permet de faire passer la journalisation sur la trace spécifiée. Si la trace spécifiée n'existe pas, elle est créée.</p> <p>La deuxième commande restaure la journalisation sur la trace des commandes Analytics d'origine.</p> <p><b>Remarque</b></p> <p>La longueur maximale du chemin d'accès et du nom de trace d'un projet Analytics est de 259 caractères. Cette longueur comprend le chemin d'accès du fichier, le nom de la trace et l'extension du fichier (.log).</p>
SET LOOP <TO> <i>num</i>	<pre>SET LOOP TO 20</pre> <p>Indique le nombre maximum de boucles qui peuvent être exécutées par la commande LOOP avant la fin de la commande.</p>

Syntaxe	Exemples et remarques
	La plage <i>num</i> est comprise entre 0 et 32 767 (0 désactive le test de boucle).
SET MARGIN <i>côté</i> <TO> <i>valeur</i>	<pre>SET MARGIN TOP TO 100</pre> <p>Indique LEFT, RIGHT, TOP ou BOTTOM pour le paramètre <i>côté</i>. Pour modifier la marge sur tous les côtés, indiquez chaque marge avec une commande SET MARGIN distincte. La spécification de la <i>valeur</i> 100 crée une marge de 1 pouce.</p>
SET MATH <TO> {FIRST   LAST   MIN   MAX}	<pre>SET MATH TO MIN</pre> <p>Paramètre par défaut : MAX</p> <p>Indique le fonctionnement de la précision décimale lorsque deux opérandes sont évalués dans une expression numérique.</p> <ul style="list-style-type: none"> <li>◦ <b>FIRST</b> : utilise le nombre de décimales du premier opérande d'une paire d'opérandes</li> <li>◦ <b>LAST</b> : utilise le nombre de décimales du dernier opérande d'une paire d'opérandes</li> <li>◦ <b>MIN</b> : utilise le plus petit nombre de décimales dans une paire d'opérandes</li> <li>◦ <b>MAX</b> : utilise le plus grand nombre de décimales dans une paire d'opérandes</li> </ul> <p>Dans les expressions contenant plusieurs opérandes, le paramètre SET MATH fonctionne par paires, en appliquant le paramètre spécifié à chaque paire d'opérandes, en l'arrondissant si nécessaire, à mesure qu'ils sont évalués dans l'ordre mathématique standard (BOMDAS).</p> <p>Si le paramètre SET MATH diminue le nombre de décimales dans un résultat, le résultat est arrondi et non tronqué.</p> <p>Pour plus d'informations, consultez la section <a href="#">Contrôle de l'arrondi dans les expressions numériques</a>.</p> <p><b>Remarque</b></p> <p>Vous ne pouvez pas utiliser SET MATH pendant qu'une table Analytics est ouverte.</p>
SET MONTHS <TO> <i>chaîne</i>	Indique les abréviations à trois caractères par défaut pour les noms de mois. Le paramètre <i>chaîne</i> correspond à la liste des abréviations de mois, séparées par des virgules.
SET NOTIFYFAILSTOP {ON   OFF}	<pre>SET NOTIFYFAILSTOP ON</pre> <p>Paramètre par défaut : OFF</p> <ul style="list-style-type: none"> <li>◦ <b>NOTIFYFAILSTOP est sur OFF</b> : Analytics autorise la poursuite du script même en cas d'échec d'une commande NOTIFY dans le script.</li> <li>◦ <b>NOTIFYFAILSTOP est sur ON</b> : Analytics arrête le traitement d'un script et écrit un message dans la trace si une commande NOTIFY échoue dans le script. Le script s'arrête après le premier échec ou après le nombre NOTIFYRETRYATTEMPTS spécifié, si aucune des tentatives n'aboutit.</li> </ul>
SET NOTIFYRETRYATTEMPTS	<pre>SET NOTIFYRETRYATTEMPTS TO 10</pre>



Syntaxe	Exemples et remarques
<TO> <i>num</i>	<p>Indique le nombre de fois où la commande NOTIFY tentera d'envoyer un message électronique en cas d'échec de la première tentative. Entrez un nombre compris entre 0 et 255. Si vous entrez 0, aucune autre tentative ne sera effectuée après un premier échec. La valeur par défaut est 5.</p> <p>Une raison possible d'échec d'envoi de message électronique par la commande NOTIFY est l'indisponibilité du serveur de messagerie.</p>
SET NOTIFYRETRYINTERVAL <TO> <i>secondes</i>	<pre>SET NOTIFYRETRYINTERVAL TO 30</pre> <p>Indique la durée en secondes entre les NOTIFYRETRYATTEMPTS. Entrez un nombre compris entre 1 et 255. La valeur par défaut est 10 secondes.</p>
SET ORDER <TO> <i>valeurs</i>	<p>Indique la séquence de tri des champs de type caractère. Le paramètre <i>valeurs</i> répertorie tous les caractères pour l'ordre de tri sélectionné.</p>
SET OVERFLOW {ON   OFF}	<pre>SET OVERFLOW OFF</pre> <p>Paramètre par défaut : ON</p> <p>Si OFF est indiqué, Analytics n'arrête pas le traitement en cas d'erreur de dépassement.</p>
SET PASSWORD <i>num</i> <TO> <i>chaîne</i>	<pre>SET PASSWORD 1 TO "motdepasse123"</pre> <p>Sert à créer une définition de mot de passe et à spécifier une valeur de mot de passe pour l'exécution de scripts autonome.</p> <p>Le paramètre <i>num</i> identifie de façon unique la définition de mot de passe et doit être une valeur comprise entre 1 et 10. Indique la valeur de mot de passe sous forme de chaîne entre guillemets.</p>
SET PERIODS <TO> <i>valeur &lt;;...n&gt;</i>	<pre>SET PERIODS TO "0;30;90;180;10000"</pre> <p>Indique les périodes d'antériorité par défaut utilisées par la commande AGE.</p>
SET PICTURE <i>format</i>	<pre>SET PICTURE "(9.999.999,99)"</pre> <p>Indique le format par défaut des valeurs numériques.</p>
SET READAHEAD <TO> <i>taille</i>	<p>Indique la taille de lecture de blocs de données. Modifiez cette valeur uniquement si l'Assistance vous le demande.</p>
SET RETRY <TO> <i>num</i> SET RETRYIMPORT <TO> <i>num</i>	<pre>SET RETRY TO 50</pre> <p>Indique le nombre de fois qu'Analytics tente d'importer ou d'exporter des données en cas d'échec de la première tentative. Entrez un nombre compris entre 0 et 255. Si vous entrez</p>

Syntaxe	Exemples et remarques
	<p>0, aucune autre tentative ne sera effectuée après un premier échec. La valeur par défaut est 0.</p> <p>Il n'y a pas de temps d'attente entre les différentes tentatives. Chaque tentative successive a lieu tout de suite après l'échec précédent.</p> <p>Il est utile de préciser le nombre de tentatives de connexion aux bases de données ou aux services de données cloud, ce qui peut être temporairement indisponible.</p> <p>S'applique aux commandes suivantes :</p> <ul style="list-style-type: none"> <li>○ ACCESSDATA</li> <li>○ IMPORT GRCPROJECT</li> <li>○ IMPORT GRCRESULTS</li> <li>○ IMPORT SAP</li> <li>○ RETRIEVE</li> <li>○ REFRESH</li> </ul> <p>(uniquement dans le cas des tables initialement créées à l'aide des options ACCESSDATA ou IMPORT SAP)</p> <ul style="list-style-type: none"> <li>○ EXPORT . . . ACLGRC</li> </ul> <p>(export vers HighBond Résultats)</p> <p><b>Remarque</b></p> <p>L'option SET RETRYIMPORT est conservée pour la compatibilité ascendante. Les options SET RETRYIMPORT et SET RETRY effectuent des actions similaires.</p>
SET SAFETY {ON   OFF}	<div style="border: 1px solid gray; padding: 5px; margin-bottom: 10px;">SET SAFETY OFF</div> <p>Indiquez ON pour afficher une boîte de dialogue de confirmation lors du remplacement de l'un des éléments suivants :</p> <ul style="list-style-type: none"> <li>○ champs d'un format de table</li> <li>○ tables Analytics</li> <li>○ fichiers, y compris les fichiers de données Analytics (.fil)</li> </ul> <p>Indiquez OFF pour empêcher l'affichage de la boîte de dialogue.</p> <p>Indiquer SET SAFETY, sans aucun paramètre, dans la ligne de commande permet d'obtenir un affichage indiquant si SAFETY est actuellement activée ou désactivée.</p>
SET SEPARATORS <TO> valeurs	<div style="border: 1px solid gray; padding: 5px; margin-bottom: 10px;">SET SEPARATORS TO ",,,"</div> <p>Indique les séparateurs par défaut utilisés par Analytics pour les décimales, les milliers et les listes. Les valeurs SET SEPARATORS doivent être trois caractères de séparateur valides dans l'ordre suivant :</p> <ul style="list-style-type: none"> <li>○ décimales (point, virgule ou espace)</li> <li>○ milliers (point, virgule ou espace)</li> <li>○ liste (point-virgule, virgule ou espace)</li> </ul>

Syntaxe	Exemples et remarques
	<p>Parmi les trois séparateurs, le séparateur décimal doit être unique. Vous devez indiquer les trois séparateurs lorsque vous utilisez la commande. Le séparateur de liste est utilisé principalement pour séparer des paramètres de fonction.</p>
<p>SET SESSION <i>nom_session</i></p>	<div data-bbox="485 394 1445 466" style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">SET SESSION</div> <div data-bbox="485 487 1445 558" style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">SET SESSION "Analyse"</div> <p>Crée une nouvelle session dans la trace des commandes Analytics. La session est identifiée par l'horodatage actif.</p> <p>Le <i>nom_session</i> facultatif permet d'ajouter jusqu'à 30 caractères d'informations d'identification supplémentaires. Les guillemets sont autorisés, mais ils ne sont pas obligatoires.</p>
<p>SET SORTMEMORY <i>num</i></p>	<div data-bbox="485 779 1445 850" style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">SET SORTMEMORY 800</div> <p>Indique la quantité maximale de mémoire allouée pour le tri et l'indexation des processus. Le paramètre <i>num</i> doit être une valeur comprise entre 0 et 2 000 mégaoctets (Mo), à saisir par incréments de 20 Mo. Si la mémoire de tri est définie sur 0, Analytics utilise la mémoire actuellement disponible.</p>
<p>SET SUPPRESSTIME {ON   OFF}</p>	<div data-bbox="485 1024 1445 1096" style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">SET SUPPRESSTIME ON</div> <p>Paramètre par défaut : OFF</p> <p>Uniquement pour une utilisation lorsque vous définissez une table Analytics qui utilise une source de données ODBC (commande IMPORT ODBC), ou l'accès direct à la base de données (commande DEFINE TABLE DB).</p> <p>Si vous utilisez le paramètre ON, lorsque vous définissez la table Analytics supprime la portion correspondant à l'heure des valeurs DateHeure. Par exemple, 20141231 235959 est lu, affiché dans les vues, et ensuite traité en tant que 20141231.</p> <p>Inclure cette commande dans un script Analytics pré-DateHeure (pre v.10.0) qui suppose que la portion correspondant à l'heure des données DateHeure sera tronquée permet au script de s'exécuter dans les versions avec DateHeure activée d'Analytics.</p> <p>Analytics supprime la portion correspondant à l'heure en utilisant uniquement la portion correspondant à la date du format DateHeure. Les données heure sont toujours présentes dans le fichier .fil ou dans la table de la base de données. Si nécessaire, vous pouvez redéfinir le champ ou définir un nouveau champ pour inclure la portion des données correspondant à l'heure.</p> <p>Si SET SUPPRESSTIME = OFF, les tables Analytics définies à l'aide d'ODBC ou de l'accès direct à la base de données incluent les valeurs de DateHeure complètes.</p> <p>Vous pouvez exécuter la commande SET SUPPRESSTIME ON/OFF dans un script ou à partir de la ligne de commande.</p> <p>Indiquer SET SUPPRESSTIME, sans aucun paramètre, dans la ligne de commande per-</p>

Syntaxe	Exemples et remarques
	met d'afficher si la suppression de la portion correspondant à l'heure des données DateHeure est actuellement activée ou désactivée.
SET SUPPRESSXML {ON   OFF}	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">SET SUPPRESSXML ON</div> Paramètre par défaut : OFF Indique que la sortie de la commande est en texte brut, et non en texte formaté.
SET TEST {ON   OFF}	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">SET TEST ON</div> Indique si les résultats des tests IF, WHILE, FOR et NEXT associés à des commandes GROUP doivent être enregistrés dans la trace.
SET TIME <TO> chaîne	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">SET TIME "hh:mm:ss PM"</div> Indique le format d'affichage de la portion correspondant à l'heure des DateHeures et les valeurs heure autonomes dans Analytics, dans les vues, les rapports et les fichiers exportés.  Lorsque vous utilisez la commande SET TIME pour indiquer des formats d'heures personnalisés, vous devez utiliser 'h' pour Heure, 'm' pour Minute, et 's' pour Seconde, même si vous avez indiqué des caractères de format d'heure différents dans la boîte de dialogue <b>Options</b> . Par exemple : <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">SET TIME TO "hh:mm"</div>
SET UTCZONE {ON   OFF}	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">SET UTCZONE OFF</div> Paramètre par défaut : ON <ul style="list-style-type: none"> <li>◦ <b>UTCZONE est sur ON</b> : Analytics modifie l'affichage des heures locales avec un décalage horaire UTC pour l'équivalent UTC de l'heure locale. (L'heure UTC est l'heure universelle coordonnée à zéro degré de longitude.)</li> <li>◦ <b>UTCZONE est sur OFF</b> : Analytics affiche les heures locales avec un décalage UTC sans les convertir en UTC.</li> </ul> Par exemple : <ul style="list-style-type: none"> <li>◦ 01 jan 2015 04:59:59 (SET UTCZONE ON)</li> <li>◦ 31 déc 2014 23:59:59-05:00 (SET UTCZONE OFF)</li> </ul> La conversion de l'heure locale en UTC est effectuée à des fins d'affichage uniquement, et elle n'affecte pas les données source. Vous pouvez faire des allées et venues entre les deux modes d'affichage à chaque fois que vous le souhaitez.
SET VERIFY {ON   OFF   BLANK}	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">SET VERIFY ON</div>

Syntaxe	Exemples et remarques
	<p>Lorsque ON est indiqué, à chaque ouverture d'une table Analytics vérifie automatiquement si le contenu d'un champ de données correspond au type de données du champ dans le format de table. Lorsque BLANK est indiqué, Analytics remplace des données de type caractère non valides par des espaces et les données numériques non valides par des zéros, en plus de la vérification décrite pour le paramètre ON.</p>
SET WIDTH <TO> <i>caractères</i>	<div data-bbox="485 457 1446 527" style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">SET WIDTH TO 20</div> <p>Indique la largeur d'affichage par défaut en caractères des champs calculés numériques ou des expressions numériques ad hoc lorsque Analytics ne peut pas déterminer la largeur maximale.</p>

# Commande SIZE

Calcule une taille d'échantillon statistiquement valide et un intervalle d'échantillonnage pour l'échantillonnage par enregistrement ou par unité monétaire.

Échantillonnage par enregistrement Échantillonnage par unité monétaire

## Syntaxe

```
SIZE RECORD CONFIDENCE niveau_confiance POPULATION taille_population PRECISION taux_acceptable <ERRORLIMIT taux_attendu> <TO {SCREEN|nom_fichier}>
```

## Paramètres

### Remarque

N'incluez pas les séparateurs des milliers ou les symboles de pourcentage lorsque vous saisissez des valeurs.

Nom	Description
RECORD	Calculez la taille de l'échantillon dans le cas d'un échantillon par enregistrement. ATTRIBUTE est un paramètre obsolète faisant la même chose que RECORD.
CONFIDENCE <i>niveau_confiance</i>	Le niveau de confiance que vous souhaitez attribuer au fait que l'échantillon créé est représentatif de la population totale.  Par exemple, saisir 95 signifie que vous souhaitez être sûr 95 % du temps que l'échantillon sera bien représentatif. La confiance est le complément du « risque d'échantillonnage ». Un niveau de confiance de 95 % équivaut à un risque d'échantillonnage de 5 %.
POPULATION <i>taille_population</i>	Nombre d'enregistrements dans la table que vous échantillonnez.
PRECISION <i>taux_acceptable</i>	Taux d'écart acceptable, qui correspond au taux d'écart maximal d'un contrôle prescrit par rapport auquel vous continuez de considérer le contrôle efficace.  Par exemple, saisir 5 implique que le taux d'écart doit être supérieur de 5 % pour que vous puissiez considérer le contrôle comme n'étant pas efficace.
ERRORLIMIT <i>taux_attendu</i> optionnel	Taux d'écart probable dans la population. Il s'agit du taux d'écart par rapport à un contrôle prescrit que vous vous attendez à trouver.  Par exemple, saisir 1 implique que vous prévoyez un taux d'écart de 1 %.  Si vous ignorez ce paramètre, c'est un taux d'écart attendu dans la population de 0 % qui est utilisé.

Nom	Description
TO SCREEN   <i>nom_fichier</i>	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li>◦ <b>SCREEN</b> : affiche les résultats dans la zone d'affichage d'Analytics.</li> <li>◦ <b>nom_fichier</b> : enregistre les résultats dans un fichier</li> </ul> <p>Indiquez <i>nom_fichier</i> sous forme de chaîne entre guillemets avec l'extension de fichier appropriée. Par exemple : TO "Sortie.TXT"</p> <p>Par défaut, le fichier est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.TXT"</li> <li>• TO "Résultats\Sortie.TXT"</li> </ul>

## Variables de sortie Analytics

Nom	Contient
SAMPINT $n$	Intervalle d'échantillonnage requis calculé par la commande.
SAMPSIZE $n$	Taille d'échantillon requise calculée par la commande.

## Exemples

### Calculer la taille et l'intervalle requis dans le cas d'un échantillon d'enregistrements

Vous avez décidé d'utiliser l'échantillonnage par enregistrement pour estimer le taux d'écart par rapport au contrôle prescrit dans un compte contenant des factures.

Avant d'extraire l'échantillon, vous devez d'abord calculer les taille et intervalle d'échantillon statistiquement valides.

Vous souhaitez être sûr 95 % du temps que l'échantillon extrait par Analytics sera bien représentatif de l'ensemble de la population.

À l'aide du niveau de confiance que vous avez spécifié, l'exemple ci-dessous calcule une taille d'échantillon de 95 et une valeur d'intervalle d'échantillon de 8,12 à utiliser pour extraire un échantillon d'enregistrements :

```
SIZE RECORD CONFIDENCE 95 POPULATION 772 PRECISION 5 ERRORLIMIT 1 TO SCREEN
```

# Remarques

## Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

# Syntaxe

```
SIZE MONETARY CONFIDENCE niveau_confiance POPULATION taille_population
MATERIALITY anomalie_acceptable <ERRORLIMIT anomalie_probable> <TO {SCREEN|nom_fichier}>
```

# Paramètres

## Remarque

N'incluez pas les séparateurs des milliers ou les symboles de pourcentage lorsque vous saisissez des valeurs.

Nom	Description
MONETARY	Calculer la taille de l'échantillon dans le cas d'un échantillon par unité monétaire.
CONFIDENCE <i>niveau_confiance</i>	<p>Le niveau de confiance que vous souhaitez attribuer au fait que l'échantillon créé est représentatif de la population totale.</p> <p>Par exemple, saisir 95 signifie que vous souhaitez être sûr 95 % du temps que l'échantillon sera bien représentatif. La confiance est le complément du « risque d'échantillonnage ». Un niveau de confiance de 95 % équivaut à un risque d'échantillonnage de 5 %.</p>
POPULATION <i>taille_population</i>	La valeur absolue totale du champ de l'échantillon numérique.
MATERIALITY <i>anomalie_acceptable</i>	<p>Anomalie acceptable qui correspond au montant total maximal de l'anomalie que peut présenter le champ de l'échantillon sans que cela soit considéré comme étant une anomalie significative.</p> <p>Par exemple, saisir 29000 signifie que le montant total de l'anomalie doit être supérieur à 29 000 \$ pour être considéré comme une anomalie pertinente.</p>
ERRORLIMIT <i>anomalie_probable</i> optionnel	<p>Anomalie probable. Il s'agit du montant total de l'anomalie que vous prévoyez de voir contenir dans le champ de l'échantillon.</p> <p>Par exemple, saisir 5800 signifie que vous vous attendez à ce que le montant total de l'anomalie soit de 5 800 \$.</p>



Nom	Description
	Si vous ignorez ce paramètre, une anomalie probable de 0,00 \$ est utilisée.
TO SCREEN   <i>nom_fichier</i>	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li>◦ <b>SCREEN</b> : affiche les résultats dans la zone d'affichage d'Analytics.</li> <li>◦ <b>nom_fichier</b> : enregistre les résultats dans un fichier</li> </ul> <p>Indiquez <i>nom_fichier</i> sous forme de chaîne entre guillemets avec l'extension de fichier appropriée. Par exemple : TO "Sortie.TXT"</p> <p>Par défaut, le fichier est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.TXT"</li> <li>• TO "Résultats\Sortie.TXT"</li> </ul>

## Variables de sortie Analytics

Nom	Contient
SAMPINT $n$	Intervalle d'échantillonnage requis calculé par la commande.
SAMPSIZE $n$	Taille d'échantillon requise calculée par la commande.

## Exemples

### Calculer la taille et l'intervalle requis dans le cas d'un échantillon par unité monétaire

Vous avez décidé d'utiliser un échantillonnage par unité monétaire pour estimer le montant total d'anomalie monétaire dans un compte contenant des factures.

Avant d'extraire l'échantillon, vous devez d'abord calculer les taille et intervalle d'échantillon statistiquement valides.

Vous souhaitez être sûr 95 % du temps que l'échantillon extrait par Analytics sera bien représentatif de l'ensemble de la population.

À l'aide du niveau de confiance que vous avez spécifié, l'exemple ci-dessous calcule une taille d'échantillon de 93 et une valeur d'intervalle d'échantillon de 6 283,33 à utiliser pour extraire un échantillon par unité monétaire :

SIZE MONETARY CONFIDENCE 95 POPULATION 585674,41 MATERIALITY 29000  
ERRORLIMIT 5800 TO SCREEN

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

# Commande SORT

Trie les enregistrements dans une table Analytics dans un ordre séquentiel croissant ou décroissant, en fonction d'un ou plusieurs champs clés indiqués. Les résultats sont renvoyés vers une nouvelle table Analytics qui a été réordonnée physiquement.

## Syntaxe

```
SORT ON {champ_clé <D> <...n>|ALL} <FIELDS nom_champ <AS nom_affichage> <...n>|FIELDS ALL> TO nom_table <IF test> <WHILE test> <FIRST plage|NEXT plage> <APPEND> <OPEN> <ISOLocale code_paramètres_régionaux>
```

## Paramètres

Nom	Description
ON <i>champ_clé</i> D <...n>   ALL	<p>Le ou les champs clés ou bien l'expression à utiliser pour le tri.</p> <p>Vous pouvez trier tout type de champ, y compris les champs calculés et les expressions ad hoc, quel que soit le type de données.</p> <ul style="list-style-type: none"> <li> <b><i>champ_clé</i></b> : utilisez le(s) champ(s) spécifié(s)           <p>Si vous triez en fonction de plusieurs champs, vous créez un tri imbriqué dans la table de sortie. L'ordre de l'imbrication suit l'ordre dans lequel vous indiquez les champs.</p> <p>Incluez D pour trier le champ clé dans l'ordre décroissant. L'ordre de tri par défaut est croissant.</p> </li> <li> <b>ALL</b> : utilisez tous les champs dans la table           <p>Si vous triez en fonction de tous les champs dans une table, vous créez un tri imbriqué dans la table de sortie. L'ordre de l'imbrication suit l'ordre d'apparition des champs dans le format de table.</p> <p>Un ordre de tri croissant est la seule option pour ALL.</p> </li> </ul>
FIELDS <i>nom_champ</i> <...n>   FIELDS ALL optionnel	<p><b>Remarque</b></p> <p>Les champs clés sont inclus automatiquement dans la table de sortie et ne doivent pas être spécifiés à l'aide de FIELDS.</p> <p>Les champs à inclure dans la sortie :</p> <ul style="list-style-type: none"> <li> <b>FIELDS <i>nom_champ</i></b> : utilisez les champs spécifiés           <p>Les champs sont utilisés dans l'ordre dans lequel vous les indiquez.</p> <p>Convertit les champs calculés en champs physiques du type de données approprié dans la table de destination : ASCII ou Unicode (selon l'édition d'Analytics), ACL (type de données numérique natif), DateHeure ou Logique. Remplit les champs physiques avec les valeurs calculées réelles.</p> </li> </ul>

Nom	Description
	<ul style="list-style-type: none"> <li>◦ <b>FIELDS ALL</b> : utilisez tous les champs dans la table Les champs sont utilisés dans leur ordre d'apparition dans le format de table. Convertit les champs calculés en champs physiques du type de données approprié dans la table de destination : ASCII ou Unicode (selon l'édition d'Analytics), ACL (type de données numérique natif), DateHeure ou Logique. Remplit les champs physiques avec les valeurs calculées réelles.</li> <li>◦ <b>ignorer FIELDS</b> : l'intégralité de l'enregistrement est incluse dans le fichier de sortie triée : tous les champs et toutes les parties indéfinies de l'enregistrement Les champs sont utilisés dans leur ordre d'apparition dans le format de table. Les champs calculés sont conservés.</li> </ul> <p><b>Astuce</b> S'il vous faut uniquement une partie des données contenues dans un enregistrement, n'incluez pas tous les champs ou l'intégralité de l'enregistrement dans la table de sortie triée. Sélectionnez uniquement les champs qu'il vous faut, ce qui, dans la plupart des cas, accélère le processus de tri.</p>
AS <i>nom_affichage</i> optionnel	<p>Utilisé uniquement pour le tri à l'aide de FIELDS.</p> <p>Le nom d'affichage (titre de la colonne) pour le champ dans la vue dans la nouvelle table Analytics. Si vous souhaitez que le nom d'affichage soit identique au nom du champ ou à un nom d'affichage existant dans la table source, n'utilisez pas AS.</p> <p>Indiquez le <i>nom_affichage</i> sous forme de chaîne entre guillemets. Utilisez un point-virgule (;) entre les mots si vous souhaitez insérer un saut de ligne dans le titre de la colonne.</p> <p><b>Remarque</b> AS fonctionne uniquement lors d'une sortie vers une nouvelle table. Si vous effectuez un ajout dans une table existante, les autres titres de colonne de la table existante ont la priorité.</p>
TO <i>nom_table</i>	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li>◦ <b>nom_table</b> : enregistre les résultats dans une table Analytics</li> </ul> <p>Indiquez <i>nom_table</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple : TO "Sortie.FIL"</p> <p>Par défaut, le fichier de données de table (.FIL) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.FIL"</li> <li>• TO "Résultats\Sortie.FIL"</li> </ul> <p><b>Remarque</b> La longueur des noms des tables est limitée à 64 caractères alphanumériques, sans l'extension .FIL. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>

Nom	Description
IF <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.</p> <p><b>Remarque</b></p> <p>La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p>
WHILE <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.</p> <p><b>Remarque</b></p> <p>Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.</p>
FIRST <i>plage</i>   NEXT <i>plage</i> optionnel	<p>Le nombre d'enregistrements à traiter :</p> <ul style="list-style-type: none"> <li>◦ <b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li>◦ <b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> </ul> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter.</p> <p>Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p>
APPEND optionnel	<p>Ajoute la sortie de commande à la fin d'un fichier existant au lieu de remplacer ce fichier.</p> <p><b>Remarque</b></p> <p>Vous devez vous assurer que la structure de la sortie de la commande et du fichier existant est identique :</p> <ul style="list-style-type: none"> <li>• les mêmes champs</li> <li>• le même ordre des champs</li> <li>• les champs correspondants ont la même longueur</li> <li>• les champs correspondants ont le même type de données</li> </ul> <p>Analytics ajoute la sortie à un fichier existant sans tenir compte de sa structure, ce qui peut désordonner les données si la structure de la sortie et du fichier existant ne correspond pas.</p>
OPEN optionnel	<p>Ouvrez la table et appliquez l'index à la table.</p>
ISOLOCALE <i>code_paramètres_régionaux</i> optionnel	<p><b>Remarque</b></p> <p>Applicable dans l'édition Unicode d'Analytics uniquement.</p> <p>Paramètre système local au format <i>langue_pays</i>. Par exemple, pour utiliser le français canadien, saisissez <i>fr_ca</i>.</p> <p>Utilisez les codes suivants :</p> <ul style="list-style-type: none"> <li>◦ <b>langue</b> : code linguistique de la norme ISO 639</li> <li>◦ <b>pays</b> : code pays de la norme ISO 3166</li> </ul>

Nom	Description
	<p>Si vous n'indiquez pas de code pays, c'est le pays par défaut pour la langue qui est utilisé.</p> <p>Si vous n'utilisez pas ISOCALE, ce sont les paramètres système régionaux par défaut qui sont utilisés.</p>

## Exemples

### Trier par un seul champ, sortir des enregistrements intégraux

Vous souhaitez trier les enregistrements dans l'exemple de table **Inventaire** par numéro de produit. Les enregistrements triés sont extraits dans une nouvelle table Analytics appelée **Inventaire\_Numéro\_Produit**.

Les enregistrements intégraux sont inclus dans la table de sortie :

```
SORT ON NumProd TO "Numéro_Produit_Inventaire"
```

Pour basculer de l'ordre de tri croissant par défaut à un ordre de tri décroissant, vous ajoutez D après le nom du champ clé :

```
SORT ON NumProd D TO "Numéro_Produit_Inventaire"
```

### Trier par un seul champ, sortir un sous-ensemble de champs

Vous souhaitez trier les enregistrements dans l'exemple de table **Inventaire** par numéro de produit. Seuls le champ clé et les champs spécifiés qui ne sont pas des champs clés sont extraits dans une table Analytics appelée **Quantité\_disponible\_Inventaire**.

Le troisième champ non clé, **QtyOH**, porte le nom d'affichage **Quantité disponible** dans la table de sortie :

```
SORT ON NumProd FIELDS DescProd StatProd QtyOH AS "Quantité disponible" TO "Quantité_dis-  
ponible_Inventaire"
```

### Trier par un seul champ, sortir tous les champs

Vous souhaitez trier les enregistrements dans l'exemple de table **Inventaire** par numéro de produit. Tous les champs sont extraits dans une nouvelle table Analytics appelée **Inventaire\_Numéro\_Produit**.

La différence entre utiliser **FIELDS ALL** et sortir l'enregistrement intégral, c'est que **FIELDS ALL** convertit les champs calculés de la table source en champs physiques dans la table de sortie, et remplit les champs avec les valeurs calculées réelles :

```
SORT ON NumProd FIELDS ALL TO "Numéro_Produit_Inventaire"
```

## Trier sur plusieurs champs (tri imbriqué)

Vous souhaitez trier les enregistrements dans l'exemple de table **Inventaire** par emplacement, puis par classe de produit, puis par numéro de produit. Les enregistrements triés sont extraits dans une nouvelle table Analytics appelée **Numéro\_Classe\_Emplacement\_Inventaire**.

```
SORT ON Emplacement ClsProd NumProd TO "Numéro_Classe_Emplacement_Inventaire"
```

## Tri à l'aide de champs associés

Vous souhaitez trier les enregistrements dans la table de l'échantillon **Trans\_Cf** en fonction des champs suivants :

- état du fournisseur (table **Fournisseur** associée)
- ville du fournisseur (table **Fournisseur** associée)
- numéro du fournisseur (table **Trans\_Cf**)

Les trois champs clés et les champs spécifiés qui ne sont pas des champs clés, y compris le champ associé **Fournisseur.Nom\_Fournisseur**, sont extraits dans une table Analytics appelée **Trans\_Cf\_Etat\_Ville** :

```
SORT ON Fournisseur.Etat_Fournisseur Fournisseur.Ville_Fournisseur No_Fournisseur FIELDS Four-
nisseur.Nom_Fournisseur Num_facture Date_facture Montant_Facture NumProd Quantité Coût_uni-
taire TO "Trans_Cf_Etat_Ville"
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Tri par champs associés

Vous pouvez trier par champs associés et les inclure sous forme de champs qui ne soient pas des champs clés dans la table de sortie triée. Pour faire référence à un champ associé dans la commande SORT, indiquez *nom table enfant.nom champ*.

## Fichiers de données à longueur fixe et à longueur variable

La commande SORT fonctionne à la fois sur des fichiers de données à longueur fixe et à longueur variable.

# Commande STATISTICS

Calcule des statistiques sur un ou plusieurs champs numériques ou DateHeure dans une table Analytics.

## Syntaxe

```
STATISTICS <ON> {champ<...n>|ALL} <STD> <MODMEDQ> <NUMBER n> <TO
{SCREEN|nom_fichier|PRINT}> <IF test> <WHILE test> <FIRST plage|NEXT plage> <APPEND>
```

## Paramètres

Nom	Description
ON champ <...n>   ALL	Indiquez un ou plusieurs champs numériques ou DateHeure sur lesquels générer des statistiques, ou indiquez ALL pour générer des statistiques sur tous les champs numérique ou DateHeure de la table Analytics.
STD optionnel	Calcule l'écart-type des champs spécifiés, en plus des autres statistiques.
MODMEDQ optionnel	Calcule le mode, la valeur médiane, les valeurs du premier quartile et du troisième quartile des champs spécifiés, en plus des autres statistiques.
NUMBER n optionnel	Nombre de valeurs faibles et élevées à conserver durant le traitement. La valeur par défaut est 5.
TO SCREEN   nom_fichier   PRINT optionnel	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li>◦ <b>SCREEN</b> : affiche les résultats dans la zone d'affichage d'Analytics.</li> <li>◦ <b>nom_fichier</b> : enregistre les résultats dans un fichier</li> </ul> <p>Indiquez <i>nom_fichier</i> sous forme de chaîne entre guillemets avec l'extension de fichier appropriée. Par exemple : TO "Sortie.TXT"</p> <p>Par défaut, le fichier est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.TXT"</li> <li>• TO "Résultats\Sortie.TXT"</li> </ul> <ul style="list-style-type: none"> <li>◦ <b>PRINT</b> : envoie les résultats vers l'imprimante par défaut</li> </ul>
IF test optionnel	Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.



Nom	Description
	<p><b>Remarque</b></p> <p>La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p>
WHILE <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.</p> <p><b>Remarque</b></p> <p>Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.</p>
FIRST <i>plage</i>   NEXT <i>plage</i> optionnel	<p>Le nombre d'enregistrements à traiter :</p> <ul style="list-style-type: none"> <li>◦ <b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li>◦ <b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> </ul> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter.</p> <p>Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p>
APPEND optionnel	<p>Ajoute la sortie de commande à la fin d'un fichier existant au lieu de remplacer ce fichier.</p> <p><b>Remarque</b></p> <p>Vous devez vous assurer que la structure de la sortie de la commande et du fichier existant est identique :</p> <ul style="list-style-type: none"> <li>• les mêmes champs</li> <li>• le même ordre des champs</li> <li>• les champs correspondants ont la même longueur</li> <li>• les champs correspondants ont le même type de données</li> </ul> <p>Analytics ajoute la sortie à un fichier existant sans tenir compte de sa structure, ce qui peut désordonner les données si la structure de la sortie et du fichier existant ne correspond pas.</p>

## Variables de sortie Analytics

### Remarque

Si vous générez des statistiques pour plusieurs champs dans une table, les variables de sortie générées par le système contiennent des valeurs pour le premier champ répertorié uniquement.

Nom	Contient
ABS <i>n</i>	Valeur absolue calculée par la commande.

Nom	Contient
AVERAGE $n$	Valeur moyenne calculée par la commande.
COUNT $n$	<p>Nombre d'enregistrements calculé par la commande.</p> <ul style="list-style-type: none"> <li>Si le nom de variable est COUNT1, elle stocke le nombre d'enregistrements de la dernière commande exécutée.</li> <li>Si le nom de variable est COUNT<math>n</math>, où <math>n</math> est supérieur à 1, la variable stocke le nombre d'enregistrements d'une commande exécutée dans une commande GROUP.</li> </ul> <p>La valeur <math>n</math> est attribuée en fonction du numéro de ligne de la commande dans GROUP. Par exemple, si la commande se situe une ligne sous de la commande GROUP, la valeur COUNT2 lui est attribuée. Si la commande se situe quatre lignes sous de la commande GROUP, la valeur COUNT5 lui est attribuée.</p>
HIGH $n$	<p>5e valeur la plus élevée identifiée par la commande.</p> <p>La 5e valeur la plus élevée est le paramètre par défaut. Il est possible de modifier le paramètre à l'aide du paramètre NUMBER. Par exemple, NUMBER 3 indique que la 3e valeur la plus élevée est stockée.</p> <p><b>Remarque</b></p> <p>Lorsqu'Analytics identifie la valeur la plus élevée, les valeurs dupliquées ne sont pas exclues. Par exemple, si dans l'ordre décroissant les valeurs sont 100, 100, 99, 98, la 3e valeur la plus élevée est 99 et non 98.</p>
LOW $n$	<p>5e valeur la plus basse identifiée par la commande.</p> <p>La 5e valeur la plus basse est le paramètre par défaut. Il est possible de modifier le paramètre à l'aide du paramètre NUMBER. Par exemple, NUMBER 3 indique que la 3e valeur la plus basse est stockée.</p> <p><b>Remarque</b></p> <p>Lorsqu'Analytics identifie la valeur la plus basse, les valeurs dupliquées ne sont pas exclues. Par exemple, si dans l'ordre croissant les valeurs sont 1, 1, 2, 3, la 3e valeur la plus basse est 2 et non 3.</p>
MAX $n$	Valeur maximale identifiée par la commande.
MEDIAN $n$	Valeur médiane identifiée par la commande.
MIN $n$	Valeur minimale identifiée par la commande.
MODE $n$	Valeur apparaissant le plus souvent qui a été identifiée par la commande.
Q25 $n$	Valeur du premier quartile (valeur du quartile inférieur) calculée par la commande.
Q75 $n$	Valeur du troisième quartile (valeur du quartile supérieur) calculée par la commande.
RANGE $n$	Différence entre les valeurs maximale et minimale calculée par la commande.
STDDEV $n$	Valeur d'écart-type calculée par la commande.
TOTAL $n$	Valeur totale calculée par la commande.

Nom	Contient
	La valeur $n$ est 1, sauf si la commande TOTAL est incluse dans une commande GROUP, auquel cas la valeur $n$ correspond au numéro de ligne de la commande TOTAL dans la commande GROUP.  Pour plus d'informations, consultez la section "Commande GROUP" Page 232.

## Exemples

### Génération de statistiques conditionnelles

Vous générez des statistiques sur le champ **Quantité** dans des enregistrements dont l'identifiant de classe de produit est 01 :

```
STATISTICS ON Quantité IF ClsProd = "01"
```

# Commande STRATIFY

Regroupe des enregistrements en intervalles numériques par valeurs dans un champ numérique. Compte le nombre d'enregistrements dans chaque intervalle et calcule également le sous-total des champs numériques spécifiés pour chaque intervalle.

## Syntaxe

```
STRATIFY <ON> champ_numérique MINIMUM valeur MAXIMUM valeur {<INTERVALS
nombre>|FREE valeur_intervalle <...n> dernier_intervalle} <SUPPRESS> <SUBTOTAL champ_
numérique <...n>|SUBTOTAL ALL> <KEY champ_fractionnement> <TO {SCREEN|nom_
table|nom_fichier|GRAPH|PRINT}> <IF test> <FIRST plage|NEXT plage> <WHILE test>
<APPEND> <OPEN> <HEADER texte_en_tête> <FOOTER texte_pied_page> <LOCAL>
<STATISTICS>
```

## Paramètres

Nom	Description
ON <i>champ_numérique</i>	Champ ou expression numérique à stratifier.
MINIMUM <i>valeur</i>	S'applique aux champs numériques uniquement. Valeur minimale du premier intervalle numérique. MINIMUM est facultatif si vous utilisez FREE, sinon il est obligatoire.
MAXIMUM <i>valeur</i>	S'applique aux champs numériques uniquement. Valeur maximale du dernier intervalle numérique. MAXIMUM est facultatif si vous utilisez FREE, sinon il est obligatoire.
INTERVALS <i>nombre</i> optionnel	S'applique aux champs numériques uniquement. Le nombre d'intervalles à taille égale générés par Analytics sur la plage spécifiée par les valeurs MINIMUM et MAXIMUM. Si vous n'indiquez pas de nombre d'intervalles, c'est la valeur par défaut qui est utilisée. La valeur par défaut est spécifiée par le nombre d' <b>Intervalles</b> dans l'onglet <b>Commande</b> de la boîte de dialogue <b>Options</b> .
FREE <i>valeur_intervalle</i> <i>&lt;...n&gt; dernier_intervalle</i> optionnel	S'applique aux champs numériques uniquement. Crée des intervalles de taille personnalisée en spécifiant le point de départ de chaque intervalle et le point de fin du dernier intervalle. Si vous spécifiez les valeurs MINIMUM et MAXIMUM, ces valeurs se trouvent au point de départ du premier intervalle et au point de fin du dernier intervalle ; chaque <i>valeur_</i>

Nom	Description
	<p><i>intervalle</i> crée un nouvel intervalle à l'intérieur de la plage. Les valeurs des intervalles que vous spécifiez doivent être supérieures à la valeur MINIMUM et inférieures ou égales à la valeur MAXIMUM.</p> <p>Les valeurs des intervalles doivent suivre l'ordre numérique et ne peuvent pas contenir de doublons :</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>FREE -1000; 0; 1000; 2000; 3000</p> </div> <p>Si vous indiquez à la fois FREE et INTERVALS, INTERVALS est ignoré.</p>
<p>SUPPRESS optionnel</p>	<p>Les valeurs supérieures à la valeur MAXIMUM et inférieures à la valeur MINIMUM sont exclues de la sortie de commande.</p>
<p>SUBTOTAL <i>champ_numérique</i> &lt;...n&gt;   SUBTOTAL ALL optionnel</p>	<p>Un ou plusieurs champs ou expressions numériques à sous-totaliser pour chaque groupe.</p> <p>Les champs multiples doivent être séparés par des espaces. Indiquez ALL pour calculer le sous-total de tous les champs numériques de la table.</p> <p>Si vous ne sélectionnez pas de champ de sous-total, le champ par lequel vous effectuez la stratification est automatiquement sous-totalisé.</p> <p>Vous devez indiquer explicitement le champ Stratifier si vous souhaitez le sous-totaliser avec un ou plusieurs autres champs ou si vous souhaitez inclure des statistiques pour le champ Stratifier sous-totalisé.</p>
<p>KEY <i>champ_fractionnement</i> optionnel</p>	<p>Champ ou expression regroupant les calculs de sous-totaux. Un sous-total est calculé chaque fois que la valeur de <i>champ_fractionnement</i> change.</p> <p><i>champ_fractionnement</i> doit être un champ ou une expression de type caractère. Vous ne pouvez indiquer qu'un seul champ, mais vous pouvez utiliser une expression contenant plusieurs champs.</p>
<p>TO SCREEN <i>nom_table</i>   <i>nom_fichier</i>   GRAPH   PRINT</p>	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li>◦ <b>SCREEN</b> : affiche les résultats dans la zone d'affichage d'Analytics.</li> <li>◦ <b><i>nom_table</i></b> : enregistre les résultats dans une table Analytics</li> </ul> <p>Indiquez <i>nom_table</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple : TO "Sortie.FIL"</p> <p>Par défaut, le fichier de données de table (.FIL) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.FIL"</li> <li>• TO "Résultats\Sortie.FIL"</li> </ul> <p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alphanumériques, sans l'extension .FIL. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>

Nom	Description
	<ul style="list-style-type: none"> <li>◦ <b>nom_fichier</b> : enregistre les résultats dans un fichier Indiquez <i>nom_fichier</i> sous forme de chaîne entre guillemets avec l'extension de fichier appropriée. Par exemple : TO "Sortie.TXT"</li> <li>Par défaut, le fichier est enregistré dans le dossier contenant le projet Analytics. Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier dans un autre dossier existant : <ul style="list-style-type: none"> <li>• TO "C:\Sortie.TXT"</li> <li>• TO "Résultats\Sortie.TXT"</li> </ul> </li> <li>◦ <b>GRAPH</b> : affiche les résultats dans un graphique dans la zone d'affichage d'Analytics</li> <li>◦ <b>PRINT</b> : envoie les résultats vers l'imprimante par défaut</li> </ul>
IF <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.</p> <p><b>Remarque</b></p> <p>La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p>
FIRST <i>plage</i>   NEXT <i>plage</i> optionnel	<p>Le nombre d'enregistrements à traiter :</p> <ul style="list-style-type: none"> <li>◦ <b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li>◦ <b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> </ul> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter. Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p>
WHILE <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.</p> <p><b>Remarque</b></p> <p>Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.</p>
APPEND optionnel	<p>Ajoute la sortie de commande à la fin d'un fichier existant au lieu de remplacer ce fichier.</p>

Nom	Description
	<p><b>Remarque</b></p> <p>Vous devez vous assurer que la structure de la sortie de la commande et du fichier existant est identique :</p> <ul style="list-style-type: none"> <li>• les mêmes champs</li> <li>• le même ordre des champs</li> <li>• les champs correspondants ont la même longueur</li> <li>• les champs correspondants ont le même type de données</li> </ul> <p>Analytics ajoute la sortie à un fichier existant sans tenir compte de sa structure, ce qui peut désordonner les données si la structure de la sortie et du fichier existant ne correspond pas.</p>
OPEN optionnel	Ouvre la table créée par la commande après l'exécution de la commande. Valide uniquement lorsque la commande crée une table de sortie.
HEADER <i>texte_en-tête</i> optionnel	Texte à insérer en haut de chaque page d'un rapport. <i>texte_en-tête</i> doit être indiqué sous forme de chaîne entre guillemets. La valeur remplace la variable système Analytics HEADER.
FOOTER <i>texte_pied_page</i> optionnel	Texte à insérer en bas de chaque page d'un rapport. <i>texte_pied_page</i> doit être indiqué sous forme de chaîne entre guillemets. La valeur remplace la variable système Analytics FOOTER.
LOCAL optionnel	Enregistre le fichier de sortie au même emplacement que le projet Analytics.  <b>Remarque</b> Applicable uniquement lorsque vous exécutez la commande sur une table de serveur avec un fichier de sortie qui est une table Analytics.
STATISTICS optionnel	<b>Remarque</b> Ne peut pas être utilisée sauf si SUBTOTAL est également spécifié.  Calcule les valeurs moyenne, minimale et maximale pour tous les champs SUBTOTAL.

## Exemples

### Stratifier par montant de facture

Vous devez stratifier une table Comptes clients sur le champ **Montant\_Facture**. Le montant des factures est généralement sous-totalisé automatiquement.

La sortie est regroupée par intervalles de 1000 \$ :

- de 0 \$ à 999,99 \$
- de 1 000 \$ à 1 999,99 \$
- etc.

Le montant total des factures est inclus pour chaque intervalle.

```
OPEN Comptes_clients
STRATIFY ON Montant_Facture MINIMUM 0 MAXIMUM 10000 INTERVALS 10 TO "Factures_Stratifiées.FIL"
```

## Remarques

### Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Fonctionnement

STRATIFY permet de regrouper les enregistrements par intervalles numériques à taille égale ou personnalisée à partir de valeurs d'un champ numérique.

La sortie contient un seul enregistrement pour chaque intervalle, avec le compte du nombre d'enregistrements de la table source inclus dans chaque intervalle.

## Remplir automatiquement les valeurs MINIMUM et MAXIMUM

Vous pouvez exécuter les commandes STATISTICS ou PROFILE sur un champ Stratifier avant l'exécution de la commande STRATIFY afin de remplir automatiquement les valeurs des paramètres MINIMUM et MAXIMUM avec la valeur la plus faible et la plus élevée dans le champ.

## Noms des champs des sous-totaux et des statistiques générés automatiquement

Si vous utilisez STATISTICS pour effectuer des calculs statistiques sur un ou plusieurs champs SUBTOTAL, et que vous produisez les résultats vers une table Analytics, les champs générés automatiquement par les paramètres ont les noms suivants :

Description du champ généré automatiquement	Nom du champ dans la table de sortie	Autre titre de la colonne (nom d'affichage) dans la table de sortie
Champ Sous-total	<i>nom du champ sous-totalisé dans la table source</i>	<b>Total</b> + autre titre de la colonne sous-totalisé dans la table source
Champ Moyenne	<b>a_</b> <i>nom du champ sous-totalisé dans la table source</i>	<b>Moyenne</b> + autre titre de la colonne sous-totalisé dans la table source



Description du champ généré automatiquement	Nom du champ dans la table de sortie	Autre titre de la colonne (nom d'affichage) dans la table de sortie
Champ Minimum	<i>m_nom du champ sous-totalisé dans la table source</i>	<b>Minimum</b> + <i>autre titre de la colonne sous-totalisé dans la table source</i>
Champ Maximum	<i>x_nom du champ sous-totalisé dans la table source</i>	<b>Maximum</b> + <i>autre titre de la colonne sous-totalisé dans la table source</i>

# Commande SUMMARIZE

Regroupe les enregistrements en fonction des valeurs identiques d'un ou plusieurs champs caractères, numériques ou DateHeure. Compte le nombre d'enregistrements dans chaque groupe et calcule également le sous-total des champs numériques spécifiés pour chaque groupe.

## Syntaxe

```
SUMMARIZE ON champ_clé <...n> <SUBTOTAL champ_numérique <...n>|SUBTOTAL ALL>
<OTHER champ <...n>|OTHER ALL> <TO {SCREEN|nom_table|PRINT}> <IF test> <WHILE test>
<FIRST plage|NEXT plage> <PRESORT> <APPEND> <OPEN> <LOCAL> <HEADER texte_en-
tête> <FOOTER texte_pied_page> <STATISTICS> <MODMEDQ> <STDEV> <CPERCENT>
<ISOLOCALE code_paramètres_régionaux>
```

## Paramètres

Nom	Description
ON <i>champ_clé</i> <...n>	Un ou plusieurs champs caractères, numériques ou DateHeure à totaliser. Les champs multiples doivent être séparés par des espaces et il peut s'agir de différents types de données.
SUBTOTAL <i>champ_numérique</i> <...n>   SUBTOTAL ALL optionnel	Un ou plusieurs champs ou expressions numériques à sous-totaliser pour chaque groupe.  Les champs multiples doivent être séparés par des espaces. Indiquez ALL pour calculer le sous-total de tous les champs numériques de la table.
OTHER <i>champ</i> <...n>   OTHER ALL optionnel	Un ou plusieurs champs supplémentaires à inclure dans la sortie. <ul style="list-style-type: none"> <li>◦ <b>champ &lt;...n&gt;</b> : incluez le(s) champ(s) spécifié(s)</li> <li>◦ <b>ALL</b> : incluez tous les champs de la table qui ne sont pas spécifiés comme champs clés ou comme champs de sous-total</li> </ul> <p>Utilisez OTHER uniquement avec des champs contenant la même valeur pour tous les enregistrements dans chaque groupe totalisé. Si vous spécifiez un champ qui contient des valeurs différentes pour un groupe totalisé, seule la valeur pour le premier enregistrement du groupe s'affiche, ce qui n'est pas judicieux.</p> <p>Par exemple :</p> <ul style="list-style-type: none"> <li>◦ <b>totalisez une table par numéro client</b> : un « autre champ » approprié peut être le <b>nom du client</b>. En général, le nom du client est identique pour tous les enregistrements avec le même numéro client.</li> <li>◦ <b>totalisez une table fournisseurs par état</b> : un « autre champ » inapproprié est <b>Ville</b>. Seule la première ville indiquée pour chaque état apparaît dans la sortie. Dans cet exemple, la meilleure méthode consiste à effectuer la totalisation avec les champs clés état et ville, dans cet ordre.</li> </ul>

Nom	Description
TO SCREEN <i>nom_table</i>   PRINT	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li>◦ <b>SCREEN</b> : affiche les résultats dans la zone d'affichage d'Analytics.</li> <li>◦ <b>nom_table</b> : enregistre les résultats dans une table Analytics</li> </ul> <p>Indiquez <i>nom_table</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple : TO "Sortie.FIL"</p> <p>Par défaut, le fichier de données de table (.FIL) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.FIL"</li> <li>• TO "Résultats\Sortie.FIL"</li> </ul> <p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alphanumériques, sans l'extension .FIL. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p> <ul style="list-style-type: none"> <li>◦ <b>PRINT</b> : envoie les résultats vers l'imprimante par défaut</li> </ul>
IF <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.</p> <p><b>Remarque</b></p> <p>La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p>
WHILE <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.</p> <p><b>Remarque</b></p> <p>Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.</p>
FIRST <i>plage</i>   NEXT <i>plage</i> optionnel	<p>Le nombre d'enregistrements à traiter :</p> <ul style="list-style-type: none"> <li>◦ <b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li>◦ <b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> </ul> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter.</p> <p>Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p>
PRESORT optionnel	<p>Trie la table par le champ clé avant d'exécuter la commande.</p> <p><b>Remarque</b></p> <p>Vous ne pouvez pas utiliser PRESORT dans la commande GROUP.</p>

Nom	Description
	<p><b>Si vous utilisez PRESORT</b></p> <p>Si vous utilisez l'option <b>PRESORT</b>, la sortie est triée et contient un seul groupe unique pour chaque jeu de valeurs identiques ou combinaison identique de valeurs dans le ou les champs clés.</p> <p><b>Astuce</b></p> <p>Si la table d'entrée est déjà triée, vous pouvez gagner du temps de traitement en évitant d'indiquer <b>PRESORT</b>.</p> <p><b>Si vous n'utilisez pas PRESORT</b></p> <p>Si vous n'utilisez pas la commande <b>PRESORT</b>, les résultats de sortie utilisent l'ordre de tri de la table d'entrée.</p> <p>Si le ou les champs clés contiennent des valeurs identiques non séquentielles, les résultats de sortie contiennent plusieurs groupes pour chaque jeu de valeurs identiques ou combinaison identique de valeurs.</p> <p><b>Remarque</b></p> <p>Plusieurs groupes pour chaque jeu de valeurs identiques ou combinaison identique de valeurs peuvent mettre en échec l'objet de la totalisation selon le contexte.</p>
APPEND optionnel	<p>Ajoute la sortie de commande à la fin d'un fichier existant au lieu de remplacer ce fichier.</p> <p><b>Remarque</b></p> <p>Vous devez vous assurer que la structure de la sortie de la commande et du fichier existant est identique :</p> <ul style="list-style-type: none"> <li>• les mêmes champs</li> <li>• le même ordre des champs</li> <li>• les champs correspondants ont la même longueur</li> <li>• les champs correspondants ont le même type de données</li> </ul> <p>Analytics ajoute la sortie à un fichier existant sans tenir compte de sa structure, ce qui peut désordonner les données si la structure de la sortie et du fichier existant ne correspond pas.</p>
OPEN optionnel	<p>Ouvre la table créée par la commande après l'exécution de la commande. Valide uniquement lorsque la commande crée une table de sortie.</p>
LOCAL optionnel	<p>Enregistre le fichier de sortie au même emplacement que le projet Analytics.</p> <p><b>Remarque</b></p> <p>Applicable uniquement lorsque vous exécutez la commande sur une table de serveur avec un fichier de sortie qui est une table Analytics.</p>
HEADER <i>texte_en-tête</i> optionnel	<p>Texte à insérer en haut de chaque page d'un rapport.</p> <p><i>texte_en-tête</i> doit être indiqué sous forme de chaîne entre guillemets. La valeur remplace la variable système Analytics HEADER.</p>

Nom	Description
FOOTER <i>texte_pied_page</i> optionnel	Texte à insérer en bas de chaque page d'un rapport. <i>texte_pied_page</i> doit être indiqué sous forme de chaîne entre guillemets. La valeur remplace la variable système Analytics FOOTER.
STATISTICS optionnel	<p><b>Remarque</b> Ne peut pas être utilisée sauf si SUBTOTAL est également spécifié.</p> <p>Calcule les valeurs moyenne, minimale et maximale pour tous les champs SUBTOTAL.</p>
MODMEDQ optionnel	<p><b>Remarque</b> Ne peut pas être utilisée sauf si SUBTOTAL est également spécifié.</p> <p>Calcule les valeurs du mode, de la médiane, du premier quartile et du troisième quartile pour tous les champs SUBTOTAL.</p>
STDEV optionnel	<p><b>Remarque</b> Ne peut pas être utilisée sauf si SUBTOTAL est également spécifié.</p> <p>Calcule l'écart-type et le pourcentage des valeurs totales pour tous les champs SUBTOTAL.</p>
CPERCENT optionnel	Calcule le pourcentage du compte d'enregistrements pour chaque groupe.
ISOLOCALE optionnel	<p><b>Remarque</b> Applicable dans l'édition Unicode d'Analytics uniquement.</p> <p>Paramètre système local au format <i>langue_pays</i>. Par exemple, pour utiliser le français canadien, saisissez <i>fr_ca</i>.</p> <p>Utilisez les codes suivants :</p> <ul style="list-style-type: none"> <li>○ <b>langue</b> : code linguistique de la norme ISO 639</li> <li>○ <b>pays</b> : code pays de la norme ISO 3166</li> </ul> <p>Si vous n'indiquez pas de code pays, c'est le pays par défaut pour la langue qui est utilisé.</p> <p>Si vous n'utilisez pas ISOLOCALE, ce sont les paramètres système régionaux par défaut qui sont utilisés.</p>

## Exemples

### Montant total des transactions par client

Vous totalisez une table Comptes clients en fonction du champ **Numéro\_client** et sous-totalisez le champ **Montant\_trans**. La sortie est regroupée par client et inclut le montant total des transactions pour chaque client :

```
OPEN Comptes_clients
SUMMARIZE ON Numéro_Client SUBTOTAL Montant_Trans TO "Total_Clients.FIL" PRESORT
```

## Montant total des transactions par client par date de transaction

Vous totalisez une table Comptes clients en fonction des champs **Numéro\_Client** et **Date\_Trans**. Vous sous-totalisez le champ **Montant\_Trans**.

La sortie est regroupée par client et, à l'intérieur de chaque client, par date. Elle inclut le montant total des transactions pour chaque client à chaque date à laquelle des transactions ont été effectuées.

```
OPEN Comptes_clients
SUMMARIZE ON Numéro_Client Date_Trans SUBTOTAL Montant_Trans TO "Total_client_par_date.FIL" PRESORT
```

## Montants total, moyen, minimum et maximum des transactions par client par date de transaction

Vous ajoutez STATISTICS à l'exemple précédent.

En plus du montant des transactions sous-totalisées pour chaque client pour chaque date à laquelle des transactions ont été effectuées par le client, vous calculez également les montants moyen, minimum et maximum des transactions pour chaque client pour chaque date :

```
OPEN Comptes_clients
SUMMARIZE ON Numéro_Client Date_Trans SUBTOTAL Montant_Trans TO "Stats_client_par_date.FIL" PRESORT STATISTICS
```

## Montants de transaction identiques, même date

Vous totalisez une table de transactions par carte de crédit sur les champs **Date\_trans** et **Montant\_Trans**.

La sortie est regroupée par date, et, à l'intérieur de chaque date, par montant. Vous pouvez utiliser le compte associé pour identifier les transactions ayant des montants identiques et des dates identiques :

```
OPEN Trans_CC
SUMMARIZE ON Date_Trans Montant_Trans TO "Transactions_par_date_montant.FIL" OPEN
PRESORT
SET FILTER TO COUNT > 1
```

# Remarques

## Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

## Fonctionnement

SUMMARIZE regroupe les enregistrements présentant la même valeur ou combinaison de valeurs dans un ou plusieurs champs caractères, numériques ou DateHeure. La sortie contient un seul enregistrement pour chaque groupe, avec le compte du nombre d'enregistrements de la table source inclus dans le groupe.

## Sous-total et statistiques : calculs et noms des champs dans les résultats de sortie

Vous pouvez utiliser un ou plusieurs paramètres facultatifs pour effectuer des calculs statistiques sur n'importe quel champ SUBTOTAL que vous spécifiez. Les calculs statistiques sont décomposés par groupe dans la sortie :

Paramètre facultatif	Autre titre de la colonne (nom d'affichage) dans la table de sortie	Nom du champ dans la table de sortie	Calcul effectué sur le champ sous-totalisé
SUBTOTAL	<b>Total</b> + autre titre de la colonne sous-totalisé	<i>nom du champ sous-totalisé</i>	Valeurs sous-totalisées pour chaque groupe
STATISTICS	<b>Moyenne</b> + autre titre de la colonne sous-totalisé	<i>a_nom du champ sous-totalisé</i>	La valeur moyenne pour chaque groupe
	<b>Minimum</b> + autre titre de la colonne sous-totalisé	<i>m_nom du champ sous-totalisé</i>	La valeur minimale pour chaque groupe
	<b>Maximum</b> + autre titre de la colonne sous-totalisé	<i>x_nom du champ sous-totalisé</i>	La valeur maximale pour chaque groupe
MODMEDQ	<b>Médiane</b> + autre titre de la colonne sous-totalisé	<i>c_nom du champ sous-totalisé</i>	La valeur médiane pour chaque groupe <ul style="list-style-type: none"> <li>◦ Jeux de valeurs impaires : la valeur du milieu</li> <li>◦ Jeux de valeurs paires : la moyenne des deux valeurs du milieu</li> </ul>
	<b>Mode</b> + autre titre de la colonne sous-totalisé	<i>o_nom du champ sous-totalisé</i>	La valeur la plus fréquente pour chaque groupe

Paramètre facultatif	Autre titre de la colonne (nom d'affichage) dans la table de sortie	Nom du champ dans la table de sortie	Calcul effectué sur le champ sous-totalisé
			<ul style="list-style-type: none"> <li>○ Affiche « N/A » si aucune valeur n'apparaît plusieurs fois</li> <li>○ En cas d'égalité, affiche la valeur la plus faible</li> </ul>
	<b>Q25</b> + autre titre de la colonne sous-totalisé	<b>q_nom</b> du champ sous-totalisé	<p>La valeur du premier quartile pour chaque groupe (valeur du quartile inférieur)</p> <ul style="list-style-type: none"> <li>○ Le résultat est une valeur interpolée calculée sur un algorithme Analytics</li> <li>○ Produit le même résultat que les fonctions QUARTILE et QUARTILE.INC dans Microsoft Excel</li> </ul>
	<b>Q75</b> + autre titre de la colonne sous-totalisé	<b>p_nom</b> du champ sous-totalisé	<p>La valeur du troisième quartile pour chaque groupe (valeur du quartile supérieur)</p> <ul style="list-style-type: none"> <li>○ Le résultat est une valeur interpolée calculée sur un algorithme Analytics</li> <li>○ Produit le même résultat que les fonctions QUARTILE et QUARTILE.INC dans Microsoft Excel</li> </ul>
STDEV	<b>STDDEV</b> + autre titre de la colonne sous-totalisé	<b>d_nom</b> du champ sous-totalisé	L'écart-type pour chaque groupe
	<b>% Champ</b> + autre titre de la colonne sous-totalisé	<b>f_nom</b> du champ sous-totalisé	Le sous-total de chaque groupe exprimé sous forme du pourcentage du total du champ
CPERCENT	<b>Pourcentage du compte</b>	<b>COUNT_PERCENTAGE</b>	Le pourcentage d'enregistrements de la table source qui appartiennent à chaque groupe



Paramètre facultatif	Autre titre de la colonne (nom d'affichage) dans la table de sortie	Nom du champ dans la table de sortie	Calcul effectué sur le champ sous-totalisé
			<b>Remarque</b> N'exige pas de champ de sous- total

# Commande TOP

Revient au premier enregistrement d'une table Analytics.

## Syntaxe

```
TOP
```

## Paramètres

Cette commande ne comporte pas de paramètres.

## Remarques

### Quand utiliser TOP

Utilisez TOP pour accéder au premier enregistrement dans une table si une commande précédente, telle que FIND, a sélectionné un autre enregistrement dans la table.

# Commande TOTAL

Calcule la valeur totale pour un ou plusieurs champs dans une table Analytics.

## Syntaxe

```
TOTAL {<FIELDS> champ_numérique<...n>|<FIELDS> ALL} <IF test> <WHILE test> <FIRST
plage|NEXT plage>
```

## Paramètres

Nom	Description
FIELDS <i>champ_numérique</i> <...n>   FIELDS ALL	Champs numériques à totaliser. Indiquez ALL pour totaliser chacun des champs numériques de la table.
IF <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.</p> <p><b>Remarque</b></p> <p>La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p>
WHILE <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.</p> <p><b>Remarque</b></p> <p>Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.</p>
FIRST <i>plage</i>   NEXT <i>plage</i> optionnel	<p>Le nombre d'enregistrements à traiter :</p> <ul style="list-style-type: none"> <li><b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li><b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> </ul> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter.</p> <p>Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p>

# Variables de sortie Analytics

## Remarque

Si vous totalisez pour plusieurs champs dans une table, la variable de sortie générée par le système contient le total pour le premier champ répertorié uniquement.

Nom	Contient
TOTAL $n$	<p>Valeur totale calculée par la commande.</p> <p>La valeur <math>n</math> est 1, sauf si la commande TOTAL est incluse dans une commande GROUP, auquel cas la valeur <math>n</math> correspond au numéro de ligne de la commande TOTAL dans la commande GROUP.</p> <p>Pour plus d'informations, consultez la section "Commande GROUP" Page 232.</p>

## Exemples

### Totalisation des 25 premiers enregistrements

Vous calculez le montant total du champ **MKTVAL** pour les 25 premiers enregistrements de la table.

```
TOTAL FIELDS MKTVAL FIRST 25
```

## Remarques

### Quand utiliser TOTAL

Utilisez TOTAL pour vérifier si les données sources sont complètes et précises, et pour générer des totaux de contrôle. La commande calcule la somme arithmétique des expressions ou des champs indiqués.

# Commande TRAIN

Utilise le Machine Learning automatisé pour créer un modèle prédictif optimal à l'aide d'un jeu de données d'entraînement.

## Syntaxe

```
TRAIN {CLASSIFIER|REGRESSOR} <ON> champ_clé <...n> TARGET champ_libellé SCORER
{ACCURACY|AUC|F1|LOGLOSS|PRECISION|RECALL|MAE|MSE|R2} SEARCHTIME minutes
MAXEVALTIME minutes MODEL nom_modèle TO nom_table <IF test> <WHILE test> <FIRST
plage|NEXT plage> FOLDS nombre_de_replis <SEED valeur_départ> <LINEAR> <NOFP>
```

### Remarque

La taille maximale prise en charge pour le jeu de données utilisé avec la commande TRAIN est de 1 Go.

## Paramètres

Nom	Description
CLASSIFIER   REGRESSOR	Type de prédiction à utiliser lors de l'entraînement d'apprentissage d'un modèle prédictif : <ul style="list-style-type: none"> <li>◦ <b>CLASSIFIER</b> : utiliser des algorithmes de classification pour entraîner un modèle Utilisez la classification si vous souhaitez prédire à quelle classe ou catégorie appartiennent les enregistrements.</li> <li>◦ <b>REGRESSOR</b> : utiliser des algorithmes de régression pour entraîner un modèle Utilisez la régression si vous souhaitez prédire les valeurs numériques associées aux enregistrements</li> </ul>
ON champ_clé <...n>	Un ou plusieurs champs d'entrée d'entraînement de l'apprentissage. Les champs peuvent être de type caractère, numérique ou logique. Les champs multiples doivent être séparés par des espaces. <p><b>Remarque</b></p> <p>Les champs de type caractère doivent être « catégoriques ». Cela signifie qu'ils doivent identifier des catégories ou des classes et contenir un nombre maximal de valeurs uniques.</p> <p>Ce maximum est spécifié par l'option <b>Valeurs max des catégories (Outils &gt; Options &gt; Commande)</b>.</p>
TARGET champ_libellé	Champ que le modèle est en train d'apprendre à prédire en se basant sur les champs d'entrée de l'entraînement de l'apprentissage.

Nom	Description				
	<p>Les différents types de prédiction (classification ou régression) fonctionnent avec différents types de données de champs :</p> <table border="1"> <tr> <td>Valide avec CLASSIFIER</td> <td>champ cible de type caractère ou logique</td> </tr> <tr> <td>Valide avec REGRESSOR</td> <td>champ cible numérique</td> </tr> </table>	Valide avec CLASSIFIER	champ cible de type caractère ou logique	Valide avec REGRESSOR	champ cible numérique
Valide avec CLASSIFIER	champ cible de type caractère ou logique				
Valide avec REGRESSOR	champ cible numérique				
SCORER ACCURACY   AUC   F1   LOGLOSS   PRECISION   RECALL   MAE   MSE   R2	<p>Mesure à utiliser lors de la notation (calibrage et classement) des modèles générés.</p> <p>Le modèle généré ayant la meilleure valeur pour cette mesure est conservé, et les autres modèles sont abandonnés.</p> <p>Un sous-ensemble différent de mesures est valide selon le type de prédiction que vous utilisez (classification ou régression) :</p> <table border="1"> <tr> <td>Valide avec CLASSIFIER</td> <td>ACCURACY   AUC   F1   LOGLOSS   PRECISION   RECALL</td> </tr> <tr> <td>Valide avec REGRESSOR</td> <td>MAE   MSE   R2</td> </tr> </table>	Valide avec CLASSIFIER	ACCURACY   AUC   F1   LOGLOSS   PRECISION   RECALL	Valide avec REGRESSOR	MAE   MSE   R2
Valide avec CLASSIFIER	ACCURACY   AUC   F1   LOGLOSS   PRECISION   RECALL				
Valide avec REGRESSOR	MAE   MSE   R2				
SEARCHTIME <i>minutes</i>	<p>Temps total en minutes à consacrer à l'entraînement de l'apprentissage et à l'optimisation d'un modèle prédictif.</p> <p>L'entraînement de l'apprentissage et l'optimisation impliquent de rechercher dans différentes configurations de pipeline (différentes combinaisons de modèles, de pré-processeurs et d'hyperparamètres).</p> <p><b>Remarque</b></p> <p>La durée totale d'exécution de la commande TRAIN est SEARCHTIME plus jusqu'à deux fois MAXEVALTIME.</p> <p><b>Astuce</b></p> <p>Spécifiez une valeur pour SEARCHTIME égale à 10 fois la valeur de MAXEVALTIME.</p> <p>Cette durée établit un équilibre raisonnable entre le temps de traitement et la possibilité d'évaluer une variété de types de modèles.</p>				
MAXEVALTIME <i>minutes</i>	<p>Durée d'exécution maximale en minutes par évaluation de modèle.</p> <p><b>Astuce</b></p> <p>Allouez 45 minutes pour 100 Mo de données d'entraînement de l'apprentissage.</p> <p>Cette durée établit un équilibre raisonnable entre le temps de traitement et la possibilité d'évaluer une variété de types de modèles.</p>				
MODEL <i>nom_modèle</i>	<p>Nom du fichier de modèle sorti par le processus d'entraînement de l'apprentissage.</p> <p>Le fichier de modèle contient le modèle le mieux adapté au jeu de données d'entraînement de l'apprentissage. Vous saisissez le modèle dans la commande PREDICT pour générer des prédictions sur un nouvel ensemble de données inédit.</p> <p>Indiquez le <i>nom_modèle</i> sous forme de chaîne entre guillemets. Par exemple : TO "Prédiction_par_défaut_prêt"</p>				

Nom	Description
	<p>Vous pouvez spécifier l'extension de fichier <code>*.model</code> ou laisser Analytics l'indiquer automatiquement.</p> <p>Par défaut, le fichier de modèle est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de modèle dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>○ TO "C:\Prédiction_par_défaut_prêt"</li> <li>○ TO "Sortie Entraînement ML\Prédiction_par_défaut_prêt.model"</li> </ul>
TO <i>nom_table</i>	<p>Nom de la table d'évaluation du modèle sortie par le processus d'entraînement de l'apprentissage.</p> <p>La table d'évaluation du modèle contient deux types d'informations différentes :</p> <ul style="list-style-type: none"> <li>○ <b>Indicateur de performance/Mesure</b> : pour les mesures de classification ou de régression, estimations quantitatives des performances prédictives du fichier de modèle sorti par le processus d'entraînement <ul style="list-style-type: none"> <li>Des mesures différentes fournissent différents types d'estimations. <b>Indicateur de performance</b> identifie la mesure que vous avez indiquée avec SCORER. <b>Mesure</b> identifie les mesures que vous n'avez pas indiquées.</li> </ul> </li> <li>○ <b>Importance/Coefficient</b> : dans l'ordre décroissant : valeurs indiquant dans quelle mesure chaque caractéristique (prédicteur) contribue aux prédictions effectuées par le modèle</li> </ul> <p>Indiquez <i>nom_table</i> sous forme de chaîne entre guillemets avec une extension de fichier .FIL. Par exemple : TO "Evaluation_modèle.FIL"</p> <p>Par défaut, le fichier de données de table (.FIL) est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier de données dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>○ TO "C:\Evaluation_modèle.FIL"</li> <li>○ TO "Sortie Entraînement ML\Evaluation_modèle.FIL"</li> </ul> <p><b>Remarque</b></p> <p>La longueur des noms des tables est limitée à 64 caractères alpha-numériques, sans l'extension .FIL. Le nom peut inclure le caractère de soulignement ( _ ), mais aucun autre caractère spécial ni espace. Le nom ne peut pas commencer par un chiffre.</p>
IF <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.</p> <p><b>Remarque</b></p> <p>La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).</p>
WHILE <i>test</i> optionnel	<p>Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.</p>

Nom	Description
	<p><b>Remarque</b></p> <p>Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.</p>
FIRST <i>plage</i>   NEXT <i>plage</i> optionnel	<p>Le nombre d'enregistrements à traiter :</p> <ul style="list-style-type: none"> <li>◦ <b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li>◦ <b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> </ul> <p>Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter.</p> <p>Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.</p>
FOLDS <i>nombre_de_replis</i>	<p>Nombre de replis de validation croisée à utiliser pour évaluer et optimiser le modèle.</p> <p>Les replis correspondent aux sous-divisions du jeu de données d'entraînement de l'apprentissage ; ils sont utilisés dans un processus de validation croisée.</p> <p>Généralement, l'utilisation de 5 à 10 replis donne de bons résultats lors de l'entraînement d'apprentissage d'un modèle. Le nombre minimum de replis autorisé est 2 ; le nombre maximum est 10.</p> <p><b>Astuce</b></p> <p>Avec des jeux de données d'entraînement de l'apprentissage moins volumineux, l'augmentation du nombre de replis peut produire une meilleure estimation des performances prédictives d'un modèle, mais elle augmente également la durée d'exécution globale.</p>
SEED <i>valeur_départ</i> optionnel	<p>Valeur de départ à utiliser pour lancer le générateur de nombres aléatoires dans Analytics.</p> <p>Si vous omettez SEED, Analytics sélectionne aléatoirement la valeur de départ.</p> <p>Spécifiez explicitement une valeur de départ et enregistrez-la si vous souhaitez reproduire le processus d'entraînement de l'apprentissage avec le même jeu de données à l'avenir.</p>
LINEAR optionnel	<p>Entraînez et notez seulement les modèles linéaires.</p> <p>Le fait de n'inclure que des modèles linéaires dans le processus d'entraînement de l'apprentissage garantit des constantes dans la sortie.</p> <p>Si LINEAR est omis, tous les types de modèles pertinents pour la classification ou la régression sont évalués.</p>
NOFP optionnel	<p>Excluez la sélection des caractéristiques et le prétraitement des données du processus d'entraînement de l'apprentissage.</p> <p>La sélection des caractéristiques est la sélection automatisée des champs du jeu de données de l'entraînement qui sont les plus utiles pour optimiser le modèle prédictif. La sélection automatisée peut améliorer les performances prédictives et réduire la quantité de données nécessaires à l'optimisation des modèles.</p> <p>Le prétraitement des données effectue des transformations telles que la mise à l'échelle et la normalisation du jeu de données d'entraînement afin de le rendre mieux adapté aux</p>



Nom	Description
	<p>algorithmes d'apprentissage.</p> <p><b>Attention</b></p> <p>Vous ne devez exclure la sélection de caractéristiques et le prétraitement des données que si vous avez une raison de le faire.</p>

## Exemples

### Entraîner un modèle de classification

Vous souhaitez entraîner un modèle de classification que vous pourrez utiliser dans un processus ultérieur pour prédire les demandeurs de prêt par défaut.

Vous entraînez le modèle sur un jeu de données historiques sur les prêts avec un résultat connu pour chaque prêt, y compris si le client a fait défaut.

Dans le processus de prévision ultérieur, vous utiliserez le modèle produit par la commande TRAIN pour traiter les données actuelles sur les demandeurs de prêts.

```
OPEN "Historique_demandeurs_prêt"
TRAIN CLASSIFIER ON Age Catégorie_emploi Salaire Solde_compte Montant_prêt Période_prêt
Refinancé Cote_crédit TARGET Valeur_par_défaut SCORER LOGLOSS SEARCHTIME 960
MAXEVALTIME 90 MODEL "Prédiction_prêt_par_défaut.model" TO "Evaluation_modèle.FIL" FOLDS
5
```

### Entraîner un modèle de régression

Vous souhaitez entraîner un modèle de régression que vous pouvez utiliser ultérieurement pour prédire le futur prix de vente des maisons.

Vous entraînez le modèle à l'aide de la commande TRAIN à l'aide d'un ensemble de données concernant les dernières ventes de maison, y compris le prix de vente.

Dans le processus de prévision ultérieur, vous utiliserez le modèle produit par la commande TRAIN pour générer les évaluations des prix des maisons.

```
OPEN "Ventes_maisons"
TRAIN REGRESSOR ON Taille_terrain Chambres SDB Etages Allée Salle_jeux Plainpied Chauff-
fage_gaz Climatisation Garage_Stationnement Quartier_préfééré TARGET Prix SCORER MSE
SEARCHTIME 960 MAXEVALTIME 90 MODEL "Prédiction_prix_maison.model" TO "Evaluation_
modèle.FIL" FOLDS 5
```

# Remarques

## Remarque

Pour plus d'informations sur le fonctionnement de cette commande, consultez le [Aide d'Analytics](#).

# Commande VERIFY

Recherche des erreurs de validité de données dans un ou plusieurs champs d'une table Analytics, en vérifiant que les données sont cohérentes avec les définitions de champ dans le format de table.

## Syntaxe

```
VERIFY {<FIELDS> champ <...n> | <FIELDS> ALL} <IF test> <WHILE test> <FIRST
plage | NEXT plage> <ERRORLIMIT n> <TO {SCREEN | nom_fichier | PRINT}> <APPEND>
```

## Paramètres

Nom	Description
FIELDS <i>champ</i> <...n>   FIELDS ALL	Champs ou expressions à vérifier. Spécifiez ALL pour vérifier tous les champs dans la table.  <b>Remarque</b> Par définition, les champs calculés, les expressions ad hoc et les champs binaires sont toujours valides.
IF <i>test</i> optionnel	Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée uniquement sur les enregistrements remplissant la condition.  <b>Remarque</b> La condition IF est évaluée uniquement par rapport aux enregistrements restant dans une table après application des options relevant du champ d'application (WHILE, FIRST, NEXT).
WHILE <i>test</i> optionnel	Expression conditionnelle qui doit être vraie afin de traiter chaque enregistrement. La commande est exécutée jusqu'à ce que la condition soit évaluée comme fausse ou lorsque la fin de la table est atteinte.  <b>Remarque</b> Si vous utilisez WHILE conjointement avec FIRST ou NEXT, le traitement des enregistrements s'arrête dès qu'une limite est atteinte.
FIRST <i>plage</i>   NEXT <i>plage</i> optionnel	Le nombre d'enregistrements à traiter : <ul style="list-style-type: none"> <li><b>FIRST</b> : pour commencer le traitement à partir du premier enregistrement jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> <li><b>NEXT</b> : pour commencer le traitement à partir de l'enregistrement actuellement sélectionné jusqu'à ce que le nombre d'enregistrements spécifié soit atteint</li> </ul> Utilisez <i>plage</i> pour indiquer le nombre d'enregistrements à traiter. Si vous ignorez FIRST et NEXT, tous les enregistrements sont traités par défaut.

Nom	Description
ERRORLIMIT <i>n</i> optionnel	Le nombre d'erreurs toléré avant la fin de la commande. La valeur par défaut est 10.
TO SCREEN   <i>nom_fichier</i>   PRINT optionnel	<p>Emplacement vers lequel envoyer les résultats de la commande :</p> <ul style="list-style-type: none"> <li>◦ <b>SCREEN</b> : affiche les résultats dans la zone d'affichage d'Analytics.</li> <li>◦ <b><i>nom_fichier</i></b> : enregistre les résultats dans un fichier</li> </ul> <p>Indiquez <i>nom_fichier</i> sous forme de chaîne entre guillemets avec l'extension de fichier appropriée. Par exemple : TO "Sortie.TXT"</p> <p>Par défaut, le fichier est enregistré dans le dossier contenant le projet Analytics.</p> <p>Utilisez un chemin de fichier absolu ou relatif pour enregistrer le fichier dans un autre dossier existant :</p> <ul style="list-style-type: none"> <li>• TO "C:\Sortie.TXT"</li> <li>• TO "Résultats\Sortie.TXT"</li> <li>◦ <b>PRINT</b> : envoie les résultats vers l'imprimante par défaut</li> </ul>
APPEND optionnel	<p>Ajoute la sortie de commande à la fin d'un fichier existant au lieu de remplacer ce fichier.</p> <p><b>Remarque</b></p> <p>Vous devez vous assurer que la structure de la sortie de la commande et du fichier existant est identique :</p> <ul style="list-style-type: none"> <li>• les mêmes champs</li> <li>• le même ordre des champs</li> <li>• les champs correspondants ont la même longueur</li> <li>• les champs correspondants ont le même type de données</li> </ul> <p>Analytics ajoute la sortie à un fichier existant sans tenir compte de sa structure, ce qui peut désordonner les données si la structure de la sortie et du fichier existant ne correspond pas.</p>

## Variables de sortie Analytics

Nom	Contient
WRITE <i>n</i>	Nombre total des erreurs de validité des données identifiées par la commande.

## Exemples

### Vérification des données et spécification d'une limite d'erreur

Vous vérifiez toutes les colonnes dans une table et définissez la limite d'erreur sur 10. La commande arrête le traitement si 10 erreurs de validité des données sont détectées :

```
VERIFY ALL ERRORLIMIT 10 TO "ImporterErreurs.txt"
```

## Remarques

### Fonctionnement

VERIFY compare les valeurs d'un ou plusieurs champs au type de données spécifié pour chaque champ dans le format de table, et renvoie toutes les erreurs. La commande vérifie les points suivants :

- Les **champs de type caractère** : contiennent uniquement des caractères valides et aucun caractère non imprimable
- Les **champs numériques** : contiennent uniquement des données numériques valides. En plus de chiffres, les champs numériques peuvent être précédées d'un signe plus ou d'un signe moins et contenir une virgule comme séparateur décimal
- Les **champs de type dateheure** : contiennent des valeurs de date, dateheure ou heure valides

Pour chaque erreur identifiée, le numéro d'enregistrement et le nom de champ sont renvoyés, avec la valeur erronée au format hexadécimal.

# Fonctions

# Fonction ABS ( )

Renvoie la valeur absolue d'une expression numérique. La valeur absolue d'un nombre correspond au nombre sans son signe.

## Syntaxe

```
ABS(nombre)
```

## Paramètres

Nom	Type	Description
<i>nombre</i>	numérique	Valeur pour laquelle il faut trouver la valeur absolue.

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 7,2 :

```
ABS(7,2)
```

Renvoie 7,2 :

```
ABS(-7,2)
```

# Fonction AGE( )

Renvoie l'antériorité, en jours, d'une date donnée par rapport à une date de référence spécifiée ou à la date actuelle du système d'exploitation.

## Syntaxe

```
AGE(date/DateHeure/chaîne_de_caractères <;date_référence>)
```

## Paramètres

Nom	Type	Description
<i>date/DateHeure/chaîne_de_caractères</i>	caractère DateHeure	Le champ, l'expression, ou la valeur littérale à classer par antériorité.
<i>date_référence</i> optionnel	caractère DateHeure	Le champ, l'expression, ou la valeur littérale par rapport auxquels <i>date/DateHeure/chaîne_de_caractères</i> est comparé. Si ce paramètre est ignoré, la date actuelle du système d'exploitation est utilisée comme date de référence.

### Remarque

*date/DateHeure/chaîne\_de\_caractères* et *date\_référence* peut accepter une valeur de type DateHeure, mais la portion de la valeur correspondant à l'heure sera ignorée. Vous ne pouvez pas utiliser AGE( ) avec les valeurs correspondant à l'heure uniquement.

## Sortie

Numérique.

## Exemples

### Exemples basiques

#### Aucune date de référence

Renvoie le nombre de jours entre le 31 décembre 2014 et la date actuelle :



- Si une valeur positive est renvoyée, elle est égale au nombre de jours dans le passé où le 31 décembre 2014 a eu lieu
- Si une valeur négative est renvoyée, elle est égale au nombre de jours dans le futur où le 31 décembre 2014 aura eu lieu
- Si 0 est renvoyé, la date actuelle est le 31 décembre 2014

```
AGE(`20141231`)
```

Renvoie le nombre de jours entre chaque date dans le champ **Date\_échéance** et la date actuelle :

```
AGE(Date_échéance)
```

## Mélanger les types de données

Renvoie 518, le nombre de jours entre les deux dates spécifiées.

```
AGE(`20130731`;`20141231`)
```

```
AGE("20130731";"20141231")
```

```
AGE(`20130731`; "20141231")
```

```
AGE(`20130731 235959`;`20141231`)
```

## À l'aide des dates de référence et des champs

Renvoie le nombre de jours entre chaque date dans le champ **Date\_échéance** et la date de référence du 31 décembre 2014 :

- Les dates antérieures à la date de référence renvoient une valeur positive égale au nombre de jours avant le jour de référence auquel elles ont lieu
- Les dates postérieures à la date de référence renvoient une valeur négative égale au nombre de jours après le jour de référence auquel elles ont lieu

```
AGE(Date_échéance ; `20141231`)
```

Renvoie le nombre de jours entre le 31 décembre 2014 et chaque date dans le champ **Date\_échéance**. Les résultats sont les mêmes que dans l'exemple immédiatement ci-dessus, mais le signe des valeurs renvoyées (positif ou négatif) est inversé :

```
AGE(`20141231`; Date_échéance)
```

## Comparaison de dates dans des champs

Renvoie le nombre de jours entre chaque date dans le champ **Date\_paiement** et une date correspondante dans le champ **Date\_échéance** :

- Les dates de paiement antérieures aux dates d'échéance renvoient une valeur positive, indiquant un paiement avant l'échéance
- Les dates de paiement postérieures aux dates d'échéance renvoient une valeur négative, indiquant un paiement après l'échéance

```
AGE(Date_paiement; Date_échéance)
```

Renvoie le nombre de jours entre chaque date dans le champ **Date\_paiement** et une date correspondante dans le champ **Date\_échéance** plus une période de grâce de 15 jours.

- Les dates de paiement antérieures aux dates d'échéance, ou jusqu'à 15 jours après la date d'échéance, renvoient une valeur positive
- Les dates de paiement postérieures de plus de 15 jours aux dates d'échéance renvoient une valeur négative, indiquant un paiement après l'échéance en dehors de la période de grâce

```
AGE(Date_paiement; Date_échéance+15)
```

## Exemples avancés

### Extraction des paiements en retard

Extraire le nom, le montant et la date de facture de chaque enregistrement dont la facture présente une antériorité supérieure à 180 jours, selon une date de référence placée au 31 décembre 2014 :

```
EXTRACT FIELDS Nom Montant Date_facture TO "En_retard" IF AGE(Date_facture;`20141231`) > 180
```

## Remarques

### Fonctionnement

La fonction AGE( ) calcule le nombre de jours séparant deux dates.

### Quand utiliser la fonction AGE( )

Utilisez AGE( ) pour comparer deux dates afin de déterminer les comptes en retard, effectuer des analyses de soldes cumulés ou effectuer n'importe quelle tâche demandant le nombre de jours écoulés entre deux dates.

## Valeurs renvoyées négatives

Une valeur négative est renvoyée si la date spécifiée pour *date/DateHeure/chaîne\_de\_caractères* est plus récente que la date indiquée comme *date\_référence* ou que la date du système d'exploitation, si aucune *date\_référence* n'est spécifiée.

## Utilisation d'un champ pour la date de référence

Contrairement à la commande AGE, qui requiert une valeur de date littérale pour la date de référence, la fonction AGE( ) vous permet d'utiliser également un champ pour la date de référence.

Par exemple :

```
AGE(Date_paiement; Date_échéance)
```

Utiliser la fonction AGE( ) de cette manière est équivalent à calculer la différence entre deux champs de date en les soustrayant dans une expression.

Par exemple :

```
Date_paiement - Date_échéance
```

## Informations sur les paramètres

Un champ de DateHeure indiqué pour *date/DateHeure/chaîne* ou *date\_référence* peut utiliser n'importe quel format de date ou DateHeure, tant que la définition de champ définit correctement le format.

## Spécification d'une valeur littérale de date ou de DateHeure

Lorsque vous indiquez une valeur littérale de date ou de DateHeure pour *date/DateHeure/chaîne* ou *date\_référence*, vous êtes limité aux formats de la table ci-dessous, et vous devez entourer la valeur de guillemets inversés, ou de guillemets simples ou doubles - par exemple, `20141231` ou "20141231".

N'utilisez aucun séparateur comme les barres obliques (/) ou les deux points (:) entre les composants individuels des dates ou des heures. Les deux-points sont autorisés dans les valeurs heure de type caractère.

- **Valeurs DateHeure** : vous pouvez utiliser n'importe laquelle des combinaisons de formats de date, de séparateur et d'heure répertoriées dans le tableau ci-dessous. La date doit précéder l'heure, et vous devez utiliser un séparateur entre les deux. Les séparateurs valides sont un espace unique, la lettre « t » ou la lettre « T ».

Exemples de formats	Exemples de valeurs littérales
AAAAMMJJ	`20141231` "20141231"

Exemples de formats	Exemples de valeurs littérales
AAMMJJ	`141231` "141231"
AAAAMMJJ hhmmss	`20141231 235959` "20141231 235959"
AAMMJJthhmm	`141231t2359` "141231t2359"
AAAAMMJJThh	`20141231T23` "20141231T23"
AAAAMMJJ hhmmss+/-hhmm (décalage UTC)	`20141231 235959-0500` "20141231 235959-0500"
AAMMJJ hhmm+/-hh (décalage UTC)	`141231 2359+01` "141231 2359+01"
<p><b>Remarque</b></p> <p>N'utilisez pas hh seul dans le format d'heure principal avec des données qui ont un décalage UTC. Par exemple, évitez : hh+hhmm. Il se peut que les résultats ne soient pas fiables.</p>	

# Fonction ALLTRIM( )

Renvoie une chaîne dont les espaces de début et de fin ont été supprimées de la chaîne d'entrée.

## Syntaxe

```
ALLTRIM(chaîne_de_caractères)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	Valeur dont il faut retirer les espaces de début et de fin.

## Sortie

Caractère.

## Exemples

### Exemples basiques

Renvoie "Vancouver" :

```
ALLTRIM(" Vancouver ")
```

Renvoie "New York" :

```
ALLTRIM(" New York ")
```

### Exemples avancés

#### Concaténation de champs de type caractère

Utilisez ALLTRIM( ) pour supprimer les espaces lorsque vous concaténez des champs de type caractère (par exemple, les champs Prénom et Nom de famille) afin que le champ créé ne contienne pas plusieurs

espaces vides entre les valeurs concaténées.

```
DEFINE FIELD Nom_complet COMPUTED ALLTRIM(Prénom) + " " + ALLTRIM(Nom)
```

## Suppression des espaces insécables

La fonction ALLTRIM( ) ne supprime pas les espaces insécables.

Si vous devez supprimer les espaces insécables de début ou de fin, créez un champ calculé à l'aide de l'expression suivante :

```
DEFINE FIELD Description_nettoyée COMPUTED ALLTRIM(REPLACE(Description; CHR(160);  
CHR(32)))
```

La fonction REPLACE( ) remplace les espaces insécables par des espaces classiques, puis la fonction ALLTRIM( ) supprime les espaces classiques de début ou de fin.

# Remarques

## Fonctionnement

La fonction ALLTRIM( ) supprime les espaces de début et de fin d'une chaîne. Les espaces se trouvant à l'intérieur de la chaîne ne sont pas supprimés.

## Fonctions associées

Si vous souhaitez supprimer uniquement les espaces de début d'une chaîne, utilisez la fonction LTRIM( ). Pour supprimer uniquement les espaces de fin, utilisez la fonction TRIM( ).

# Fonction ASCII()

Renvoie le code ASCII pour le caractère indiqué.

## Syntaxe

```
ASCII(caractère)
```

## Paramètres

Nom	Type	Description
<i>caractère</i>	caractère	Caractère dont le code ASCII doit être identifié. Vous pouvez indiquer un caractère entre guillemets ou une chaîne, un champ ou une expression à caractères multiples. Si vous indiquez des caractères multiples, seul le premier d'entre eux est évalué.

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 65 :

```
ASCII("A")
```

Renvoie 49 :

```
ASCII("1")
```

### Exemples avancés

**Extraction d'enregistrements qui commencent par un caractère de tabulation**

Extrayez des enregistrements comprenant un caractère de tabulation au début d'un champ appelé « Description ». Le code ASCII pour un caractère de tabulation est « 9 ».

```
EXTRACT RECORD TO "EntréesTab.acl" IF ASCII(Description) = 9
```

## Remarques

### Recherche des caractères non imprimables

Vous pouvez utiliser ASCII( ) pour rechercher les caractères non imprimables comme :

- **Nul** : ASCII "0"
- **Tabulation** : ASCII "9"
- **Saut de ligne (LF)** : ASCII "10"
- **Retour chariot (CR)** : ASCII "13"

### Fonctions associées

ASCII( ) est l'inverse de la fonction CHR( ).



# Fonction AT( )

Renvoie un nombre indiquant l'emplacement de début d'une occurrence spécifique d'une sous-chaîne dans une valeur de caractère.

## Syntaxe

```
AT(num_occurrence; chaîne_recherchée; dans_texte)
```

## Paramètres

Nom	Type	Description
<i>num_occurrence</i>	numérique	Occurrence (instance) de <i>chaîne_recherchée</i> dont renvoyer l'emplacement.  Par exemple, spécifiez 1 pour renvoyer l'emplacement de départ de la première occurrence de <i>chaîne_recherchée</i> .
<i>chaîne_recherchée</i>	caractère	Sous-chaîne à rechercher dans <i>dans_texte</i> . Cette valeur est sensible à la casse.  Si <i>chaîne_recherchée</i> inclut des guillemets doubles, il est nécessaire d'entourer la valeur de guillemets simples :  <div style="border: 1px solid #ccc; padding: 5px; width: fit-content; margin: 5px auto;">AT(1;"test"; Description)</div>
<i>dans_texte</i>	caractère	Valeur dans laquelle effectuer la recherche.  Vous pouvez concaténer deux champs ou plus dans le paramètre <i>dans_texte</i> si vous souhaitez que votre recherche porte sur plusieurs champs d'une table :  <div style="border: 1px solid #ccc; padding: 5px; width: fit-content; margin: 5px auto;">AT(1;"test"; Description+Résumé)</div>

## Sortie

Numérique. Renvoie la position d'octet de départ de l'occurrence indiquée pour la valeur *chaîne\_recherchée*. Renvoie 0 si aucune correspondance n'existe.

# Exemples

## Exemples basiques

### Occurrences trouvées

Renvoie 4 :

```
AT(1; "-"; "604-669-4225")
```

Renvoie 8 :

```
AT(2; "-"; "604-669-4225")
```

### Occurrences non trouvées

Renvoie 0 parce qu'il n'y a pas de troisième trait d'union dans la chaîne.

```
AT(3; "-"; "604-669-4225")
```

Renvoie 0 parce qu'il n'y a pas de quatrième lettre "a" en minuscule dans la chaîne.

```
AT(4; "a"; "Alabama")
```

## Groupes de caractères

Renvoie 5 :

```
AT(2; "iss"; "Mississippi")
```

## Rechercher dans un champ

Renvoie la position d'octet du premier trait d'union dans chaque valeur du champ **No\_Facture** :

```
AT(1; "-"; No_Facture)
```

## Exemples avancés

**Recherche de numéros de factures dans lesquels le deuxième trait d'union apparaît après la position du dixième octet**

Vous pouvez analyser la cohérence des numéros de facture dans une table à l'aide de la fonction `AT()` pour créer un filtre comme celui ci-dessous. Ce filtre isole tous les enregistrements dans lesquels le numéro de facture contient au moins 2 traits d'union et dont le deuxième trait d'union se place après la position du dixième octet :

```
SET FILTER TO AT(2; "-"; No_Facture) > 10
```

## Remarques

### Quand utiliser la fonction `AT()`

Utilisez cette fonction pour récupérer les positions de départ suivantes à l'intérieur d'une valeur de type caractère :

- position de départ d'une sous-chaîne
- position de départ d'une occurrence suivante de la sous-chaîne

Si vous souhaitez uniquement confirmer plusieurs occurrences de la même sous-chaîne dans un champ, il est conseillé d'utiliser la fonction `OCCURS()`. Pour plus d'informations, consultez la section "Fonction `OCCURS()`" Page 705.

### Valeur renvoyée lorsque *num\_occurrence* dépasse le nombre d'occurrences

Si *num\_occurrence* est supérieure au nombre réel d'occurrences dans la sous-chaîne dans *dans\_texte*, la fonction renvoie 0 parce que cette occurrence est introuvable dans la sous-chaîne.

### Champs concaténés et valeurs renvoyées

Lorsque vous effectuez une recherche dans plusieurs champs, la valeur renvoyée pour l'instance est l'emplacement de départ de *chaîne\_recherchée* dans tous les champs que vous spécifiez. Les champs concaténés sont traités comme un seul champ incluant les espaces de début et de fin des champs individuels sauf si vous utilisez la fonction `ALLTRIM()` pour supprimer les espaces.

Par exemple, si vous recherchez la première occurrence d'une chaîne dans deux champs d'une largeur de huit caractères chacun et que la chaîne est recherchée au début du deuxième champ, la valeur renvoyée est 9.

# Fonction BETWEEN( )

Renvoie une valeur logique indiquant si la valeur indiquée fait partie d'une plage.

## Syntaxe

```
BETWEEN(valeur, min; max)
```

## Paramètres

Nom	Type	Description
<i>valeur</i>	caractère numérique DateHeure	Le champ, l'expression, ou la valeur littérale à tester.
<i>min</i>	caractère numérique DateHeure	Valeur minimale de la plage.
<i>max</i>	caractère numérique DateHeure	Valeur maximale de la plage.

### Remarque

La plage d'évaluation par rapport à **T** (vrai) inclut les valeurs *min* et *max*.

Pour plus d'informations sur les plages de caractères, consultez la rubrique "Comportement de SET EXACT" Page opposée.

## Sortie

Logique. Renvoie **T** (vrai) si la *valeur* est supérieure ou égale à la valeur *min*, et inférieure ou égale à la valeur *max*. Renvoie **F** (faux) dans tous les autres cas.

# Exemples

## Exemples basiques

### Entrée de type numérique

Renvoie T :

```
BETWEEN(500;400;700)
```

Renvoie F :

```
BETWEEN(100;400;700)
```

### Entrée de type caractère

Renvoie T :

```
BETWEEN("B";"A";"C")
```

Renvoie « F » car la comparaison des caractères est sensible à la casse, c'est pourquoi il n'existe aucune correspondance pour la valeur « b » minuscule entre « A » et « C » majuscules.

```
BETWEEN("b";"A";"C")
```

### Entrée de DateHeure

Renvoie T :

```
BETWEEN(`141230`;`141229`;`141231`)
```

Renvoie « T » pour toutes les valeurs dans le champ **Heure\_connexion** de 07:00:00 AM à 09:00:00 AM inclus ; renvoie « F » dans tous les autres cas :

```
BETWEEN(Heure_connexion;`t070000`;`t090000`)
```

### Comportement de SET EXACT

Renvoie « T » pour toutes les valeurs dans le champ **Nom** qui commencent par les lettres allant de "C" à "K" inclus ; renvoie "F" dans tous les autres cas (SET EXACT doit être sur OFF) :

```
BETWEEN(Nom; "C"; "K")
```

Renvoie « T » pour toutes les valeurs dans le champ **Nom** qui commencent par les lettres allant de "C" à "J" inclus ; renvoie « F » dans tous les autres cas (SET EXACT doit être sur ON). Renvoie aussi « T » pour la seule lettre "K" :

```
BETWEEN(Nom; "C"; "K")
```

## Entrée de champ

Renvoie « T » pour toutes les valeurs du champ **Date\_facture** compris entre le 30 septembre 2014 et le 30 octobre 2014 y compris, et renvoie « F » dans tous les autres cas :

```
BETWEEN(Date_facture; `20140930`; `20141030`)
```

Renvoie « T » pour tous les enregistrements dans lesquels la date de facture n'est pas comprise entre la date du bon de commande et la date de paiement y compris, et renvoie « F » dans tous les autres cas :

```
NOT BETWEEN(Date_facture; Date_BdC; Date_Paiement)
```

Renvoie « T » pour toutes les valeurs du champ **Montant\_facture** comprises entre 1 000 \$ et 5 000 \$ y compris, et renvoie « F » dans tous les autres cas :

```
BETWEEN(Montant_Facture; 1000, 5000)
```

## Exemples avancés

### Création d'un filtre pour visualiser une tranche salariale

Dans l'exemple suivant, la table d'échantillonnage **Liste\_employés** est ouverte et applique un filtre limitant les enregistrements affichés pour inclure seulement les employés qui gagnent un salaire compris entre 40 000 et 50 000 €.

```
OPEN Liste_Employés
SET FILTER TO BETWEEN(Salaire; 40000,00; 50000,00)
```

# Remarques

## Types de données prises en charge

Les valeurs d'introduction de la fonction BETWEEN( ) peuvent être des données de type numérique, caractère ou DateHeure. Vous ne pouvez pas mélanger les types de données. Les trois valeurs d'introduction doivent toutes appartenir à la même catégorie de données.

## Utilisation de BETWEEN( ) au lieu de l'opérateur AND

Cette fonction permet de remplacer des expressions qui utilisent l'opérateur AND.

Par exemple :

```
BETWEEN(Montant_Facture; 1000, 5000)
```

est équivalent à

```
Montant_Facture >= 1000 AND Montant_Facture <= 5000
```

## Ordre de *min* et de *max*

Dans la fonction BETWEEN( ), l'ordre des paramètres *min* et *max* n'est pas important parce qu'Analytics identifie automatiquement quelles sont les valeurs minimum et maximum.

Les deux exemples ci-dessous renvoient T :

```
BETWEEN(2500; 1000; 5000)
```

```
BETWEEN(2500; 5000; 1000)
```

## Précision décimale des valeurs d'introduction numériques

Lorsque la précision décimale des valeurs d'introduction numériques comparées est différente, la comparaison utilise le niveau de précision le plus élevé.

Renvoie T, car 1,23 est égal à 1,23 :

```
BETWEEN(1,23; 1,23; 1,25)
```

Renvoie F, car 1,23 est inférieur à 1,234 si l'on tient compte de la troisième décimale :

```
BETWEEN(1,23; 1,234; 1,25)
```

## Données de type caractère

### Respect de la casse

La fonction BETWEEN( ) est sensible à la casse lorsque vous l'utilisez avec des données caractère. Lors de la comparaison des caractères, « a » n'est pas considéré comme étant équivalent à « A ».

Renvoie F :

```
BETWEEN("B";"a";"C")
```

Si vous travaillez avec des données qui incluent des incohérences de casse, vous pouvez utiliser la fonction UPPER( ) pour convertir les valeurs et remédier aux problèmes de casse avant de vous servir de la fonction BETWEEN( ).

Renvoie T :

```
BETWEEN(UPPER("B"); UPPER("a"); UPPER("C"))
```

### Rapprochement partiel

Le rapprochement partiel est pris en charge pour les comparaisons de caractère.

*valeur* peut être contenue par *min*.

Renvoie T, même si *valeur* "AB" apparaît comme étant inférieure à *min* "ABC" :

```
BETWEEN("AB"; "ABC"; "Z")
```

*max* peut être contenue par *valeur*.

Renvoie T, même si *valeur* "ZZ" apparaît comme étant inférieure à *max* "Z" :

```
BETWEEN("ZZ"; "ABC"; "Z")
```

#### Remarque

La plus petite valeur de la comparaison de caractères doit apparaître au début de la valeur plus longue pour constituer une correspondance.

### Rapprochement partiel et SET EXACT

Le rapprochement partiel est activé lorsque SET EXACT = OFF, qui est le paramètre par défaut d'Analytics. Si SET EXACT = ON, le rapprochement partiel est désactivé et les valeurs de comparaison doivent correspondre exactement pour constituer un rapprochement.



Les deux exemples précédents sont False (faux) lorsque SET EXACT est sur ON.

Pour plus d'informations sur SET EXACT (option **Comparaisons exactes de caractères**), consultez la rubrique "Commande SET" Page 435.

## Activation et désactivation de SET EXACT

Si vous souhaitez vous assurer que l'option **Comparaisons exactes de caractères** n'est pas utilisée avec la fonction BETWEEN( ), vérifiez si l'option est décochée dans l'onglet **Table** de la boîte de dialogue **Options (Outils > Options)**.

Si vous utilisez un script, vous pouvez ajouter la commande SET EXACT OFF avant l'apparition de la fonction BETWEEN( ). Si nécessaire, vous pouvez restaurer l'état antérieur grâce à la commande SET EXACT ON.

## Paramètres DateHeure

Un champ date, DateHeure ou heure indiqué comme entrée de fonction peut utiliser n'importe quel format de date, DateHeure ou heure, tant que la définition de champ définit correctement le format.

### Mélange d'entrées de type date, DateHeure et heure

Il n'est pas impossible de mélanger des valeurs de date, DateHeure et heure dans les trois entrées de la fonction BETWEEN( ), mais le fait de mélanger ces sous-types de DateHeure peut aboutir à des résultats qui n'ont pas de sens.

Analytics utilise des équivalents en chiffres sériels pour traiter les calculs de DateHeure, de sorte que même si vous êtes intéressé uniquement par la portion correspondant à la date d'une valeur DateHeure, la portion correspondant à l'heure fait toujours partie du calcul.

Prenons les exemples suivants :

Renvoie T, car PM le 31 décembre 2014 se situe dans la plage spécifiée par *min* et *max* :

```
BETWEEN(`20141231`;`20141229`;`20141231`)
```

Renvoie F, même si 12:00 PM le 31 décembre 2014 semble se situer dans la plage spécifiée par *min* et *max* :

```
BETWEEN(`20141231 120000`;`20141229`;`20141231`)
```

Si nous regardons l'équivalent en chiffre sériel de ces deux expressions, nous pouvons voir pourquoi le second génère une évaluation qui a pour résultat F (faux).

Renvoie T, car la *valeur* du nombre sériel est égale au nombre sériel *max* :

```
BETWEEN(42003,000000; 42001,000000; 42003,000000)
```

Renvoie F, car la *valeur* du nombre sériel est supérieure au nombre sériel *max* :

```
BETWEEN(42003,500000; 42001,000000; 42003,000000)
```

Le chiffre sériel 42003,500000 est plus grand que 42003,000000 et par conséquent à l'extérieur de la fourchette, même si les deux dates sont identiques. 0,500000 est le nombre sériel équivalent de 12:00 PM.

## Harmoniser les sous-types DateHeure

Pour éviter les problèmes pouvant découler du mélange de sous-types DateHeure, vous pouvez utiliser des fonctions pour harmoniser les sous-types.

Par exemple, cette expression, qui utilise les mêmes valeurs initiales que le deuxième exemple ci-dessus, renvoie T plutôt que F :

```
BETWEEN(CTOD(DATE(`20141231
120000`; "AAAAMMJJ")); "AAAAMMJJ"); `20141229`; `20141231`)
```

## Spécification d'une valeur littérale de date, de DateHeure ou d'heure

Lorsque vous indiquez une valeur littérale de date, DateHeure ou heure pour l'une des entrées de fonction, vous êtes limité aux formats de la table ci-dessous, et vous devez entourer la valeur de guillemets inversés - par exemple, `20141231`.

N'utilisez aucun séparateur comme les barres obliques (/) ou les deux points (:) entre les composants individuels des dates ou des heures.

- **Valeurs DateHeure** : vous pouvez utiliser n'importe laquelle des combinaisons de formats de date, de séparateur et d'heure répertoriées dans le tableau ci-dessous. La date doit précéder l'heure, et vous devez utiliser un séparateur entre les deux. Les séparateurs valides sont un espace unique, la lettre « t » ou la lettre « T ».
- **Valeurs heure** : vous devez indiquer l'heure à l'aide de l'horloge de 24 heures. Les décalages par rapport à l'heure universelle coordonnée (UTC) doivent être préfacés d'un signe plus (+) ou d'un signe moins (-).

Exemples de formats	Exemples de valeurs littérales
AAAAMMJJ	`20141231`
AAMMJJ	`141231`
AAAAMMJJ hhmmss	`20141231 235959`
AAMMJJthmm	`141231t2359`
AAAAMMJJThh	`20141231T23`

Exemples de formats	Exemples de valeurs littérales
AAAAMMJJ hhmmss+/-hhmm (décalage UTC)	`20141231 235959-0500`
AAMMJJ hhmm+/-hh (décalage UTC)	`141231 2359+01`
thmmss	`t235959`
Thhmm	`T2359`
<p><b>Remarque</b></p> <p>N'utilisez pas hh seul dans le format d'heure principal avec des données qui ont un décalage UTC. Par exemple, évitez : hh+hhmm. Il se peut que les résultats ne soient pas fiables.</p>	

# Fonction BINTOSTR( )

Renvoie des données caractère Unicode converties à partir de données caractère ZONED ou EBCDIC. Abréviation de "Binaire à Chaîne".

## Remarque :

Cette fonction est spécifique à l'édition Unicode d'Analytics. Elle n'est pas prise en charge dans l'édition non-Unicode.

## Syntaxe

```
BINTOSTR(chaîne_de_caractères; type_de_chaîne_de_caractères)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	Valeur ZONED ou EBCDIC que vous souhaitez convertir en codage de caractères Unicode.
<i>type_de_chaîne_de_caractères</i>	caractère	Format de départ de la conversion. Vous devez préciser une des valeurs suivantes : <ul style="list-style-type: none"> <li>"A" : conversion à partir de données ZONED (ASCII)</li> <li>"E" : conversion à partir de données EBCDIC</li> </ul>

## Sortie

Caractère.

## Exemples

### Exemples basiques

L'expression ZONED(-6448;4) convertit la valeur -6448 au format de type caractère "644Q". Toutefois, l'édition Unicode d'Analytics nécessite que vous convertissiez la sortie de ZONED( ) en caractères Unicode à l'aide de BINTOSTR( ).

Renvoie "644Q" au format Unicode :

```
BINTOSTR(ZONED(-6448;4); "A")
```

## Remarques

### Quand utiliser BINTOSTR( )

Cette fonction permet de convertir des valeurs renvoyées à partir des fonctions ZONED( ) et EBCDIC( ) en valeurs Unicode.

#### Remarque

Si cette fonction n'est pas appliquée aux valeurs renvoyées de ZONED( ) et EBCDIC( ) dans les éditions Unicode d'Analytics, le codage est mal interprété, entraînant ainsi un affichage incorrect de celles-ci.

# Fonction BIT()

Renvoie la représentation binaire de la position d'octet indiquée dans l'enregistrement actuel en tant que chaîne à huit caractères.

## Syntaxe

```
BIT(position_octet)
```

## Paramètres

Nom	Type	Description
<i>position_octet</i>	numérique	Position d'octet à renvoyer sous forme de valeur binaire.

## Sortie

Caractère.

## Exemples

### Exemples basiques

Renvoie « 00110001 » si le huitième octet contient la valeur « 1 » :

```
BIT(8)
```

Renvoie « 01000001 » si le neuvième octet contient la valeur « A » :

```
BIT(9)
```

Renvoie « 01100001 » si le dix-septième octet contient la valeur « a » :

```
BIT(17)
```

## Exemples avancés

### Utilisation de BIT ( ) et SUBSTRING ( ) pour extraire une valeur

Supposons que la position d'octet 17 contienne un ensemble de 8 indicateurs de crédit.

Pour extraire tous les enregistrements de clients dont le troisième bit est défini sur 1 (ce qui signifie « ne pas expédier »), spécifiez :

```
EXTRACT IF SUBSTRING(BIT(17); 3; 1) = "1"
```

Dans cet exemple, la fonction SUBSTRING( ) est utilisée pour extraire la valeur du troisième bit.

## Remarques

### Fonctionnement

La fonction BIT( ) convertit l'octet situé à la position d'octet indiquée en chaîne de huit caractères composée de zéros et de uns.

### Quand utiliser la fonction BIT( )

Utilisez BIT( ) pour examiner les bits individuels qui constituent un octet.

### Fonctions associées

Si vous souhaitez récupérer le caractère situé à l'emplacement d'octet indiqué, utilisez la fonction BIT( ).

# Fonction BLANKS()

Renvoie une chaîne contenant un nombre spécifique d'espaces.

## Syntaxe

```
BLANKS(compte)
```

## Paramètres

Nom	Type	Description
<i>compte</i>	numérique	Nombre d'espaces à insérer.

## Sortie

Caractère.

## Exemples

### Exemples basiques

Renvoie «    » :

```
BLANKS(5)
```

Renvoie "Entreprise ABC" :

```
"Entreprise" + BLANKS(1) + "ABC"
```



# Remarques

## Quand utiliser `BLANKS( )`

Utiliser la fonction `BLANKS( )` pour harmoniser les champs, initialiser des variables dans les scripts ou insérer des espaces lors de la mise en forme de champs ou de la concaténation de chaînes.

# Fonction BYTE( )

Renvoie le caractère stocké dans la position d'octet indiquée de l'enregistrement actuel.

## Syntaxe

```
BYTE(position_octet)
```

## Paramètres

Nom	Type	Description
<i>position_octet</i>	numérique	Position d'octet à renvoyer sous forme de valeur caractère. La valeur fait référence à la position dans l'enregistrement (en comptant à partir de 1), indépendamment des éventuelles définitions de champs.

## Sortie

Caractère.

## Exemples

### Exemples basiques

Renvoie « 1 » à partir d'un enregistrement commençant par un champ ID contenant « 1 » :

```
byte(112)
```

### Exemples avancés

#### Identifier les enregistrements dans des fichiers d'impression ou PDF à partir d'une mise en forme cohérente

Utilisez BYTE( ) pour identifier les enregistrements dans un fichier de données contenant un caractère

spécifique, à une position d'octet précise. C'est généralement le cas dans les fichiers Image d'impression (rapport) et Adobe Acrobat (PDF), pour lesquels les données sont formatées de manière cohérente dans l'ensemble du document.

Par exemple, pour trouver et extraire les enregistrements incluant un point en position d'octet 113 :

```
EXTRACT RECORD IF BYTE(113) = "." TO "Sortie.fil"
```

## Remarques

### Quand utiliser la fonction BYTE( )

Utilisez BYTE( ) pour examiner le contenu d'une position dans un enregistrement, sans avoir à définir un champ à cet effet.

### Utilisation de BYTE( ) sur des données EBCDIC

Si vous utilisez cette fonction sur des données EBCDIC, la valeur renvoyée sera également EBCDIC. Il est possible que vous ne puissiez pas les comparer à des valeurs caractère.

### Fonctions associées

Si vous souhaitez récupérer la représentation binaire de l'emplacement d'octet indiqué, utilisez la fonction BIT( ).

# Fonction CDOW()

Renvoie le nom du jour de la semaine de la date ou de la DateHeure indiquée. Abréviation de "Jour de la semaine au format caractère".

## Syntaxe

```
CDOW(date/DateHeure; longueur)
```

## Paramètres

Nom	Type	Description
<i>date/DateHeure</i>	DateHeure	Le champ, l'expression, ou la valeur littérale pour lesquels renvoyer le nom du jour.
<i>longueur</i>	numérique	Une valeur entre 1 et 9 indiquant la longueur de la chaîne de sortie. Pour afficher les noms raccourcis des jours, indiquez une valeur inférieure.

## Sortie

Caractère.

## Exemples

### Exemples basiques

Renvoie "Wednesday" car le 31 décembre 2014 tombe un mercredi et la *longueur* est de 9 :

```
CDOW('20141231'; 8)
```

Renvoie "Wed" car le 31 décembre 2014 tombe un mercredi et la *longueur* est de 3 :

```
CDOW('20141231 235959'; 3)
```

Renvoie le nom complet du jour pour chaque valeur dans le champ **Date\_facture** :

```
CROW(Date_facture; 9)
```

Renvoie le nom du jour abrégé pour chaque valeur dans le champ **Horodatage\_reçu** :

```
CROW(Horodatage_reçu; 3)
```

## Exemples avancés

### Ajout d'un champ identifiant les jours de la semaine pour les dates

Utilisez la fonction `CROW( )` pour créer un champ calculé qui identifie les jours de la semaine pour toutes les dates d'un champ de dates. Une fois que vous avez créé le champ calculé, vous pouvez l'ajouter à la vue à côté de la colonne de date.

```
DEFINE FIELD Nom_du_jour COMPUTED CROW(Date_Trans; 3)
```

### Création d'un filtre pour rechercher les transactions ayant eu lieu un week-end

Utilisez la fonction `CROW( )` pour créer un filtre qui isole les transactions ayant eu lieu pendant un week-end :

```
SET FILTER TO CROW(Date_Trans; 3) = "Sam" OR CROW(Date_Trans; 3) = "Dim"
```

## Remarques

### Informations sur les paramètres

Un champ indiqué pour *date/DateHeure* peut utiliser n'importe quel format de date ou de DateHeure, tant que la définition de champ définit correctement le format.

Si le paramètre *longueur* est inférieur au nom du jour, celui-ci est tronqué pour correspondre à la longueur indiquée. Si le paramètre *longueur* est plus long que le nom du jour, celui-ci est complété avec des espaces.

### Spécification d'une valeur littérale de date ou de DateHeure

Lorsque vous indiquez une valeur littérale de date ou de DateHeure pour *date/DateHeure*, vous êtes limité aux formats de la table ci-dessous, et vous devez entourer la valeur de guillemets inversés - par exemple, `20141231`.

N'utilisez aucun séparateur comme les barres obliques (/) ou les deux points (:) entre les composants individuels des dates ou des heures.

- **Valeurs DateHeure** : vous pouvez utiliser n'importe laquelle des combinaisons de formats de date, de séparateur et d'heure répertoriées dans le tableau ci-dessous. La date doit précéder l'heure, et vous devez utiliser un séparateur entre les deux. Les séparateurs valides sont un espace unique, la lettre « t » ou la lettre « T ».
- **Valeurs heure** : vous devez indiquer l'heure à l'aide de l'horloge de 24 heures. Les décalages par rapport à l'heure universelle coordonnée (UTC) doivent être préfacés d'un signe plus (+) ou d'un signe moins (-).

Exemples de formats	Exemples de valeurs littérales
AAAAMMJJ	`20141231`
AAMMJJ	`141231`
AAAAMMJJ hhmmss	`20141231 235959`
AAMMJJthhmm	`141231t2359`
AAAAMMJJThh	`20141231T23`
AAAAMMJJ hhmmss+/-hhmm (décalage UTC)	`20141231 235959-0500`
AAMMJJ hhmm+/-hh (décalage UTC)	`141231 2359+01`
<p><b>Remarque</b></p> <p>N'utilisez pas hh seul dans le format d'heure principal avec des données qui ont un décalage UTC. Par exemple, évitez : hh+hhmm. Il se peut que les résultats ne soient pas fiables.</p>	

## Fonctions associées

Si vous devez renvoyer le jour de la semaine comme un nombre (de 1 à 7), utilisez DOW() au lieu de CDOW().

# Fonction CHR( )

Renvoie le caractère associé au code ASCII indiqué.

## Syntaxe

```
CHR(nombre)
```

## Paramètres

Nom	Type	Description
<i>nombre</i>	numérique	Expression numérique valide comprise entre 1 et 255.

## Sortie

Caractère.

## Exemples

### Exemples basiques

Renvoie « A » :

```
CHR(65)
```

Renvoie « 1 » :

```
CHR(49)
```

### Exemples avancés

#### Ajouter le symbole de la livre sterling (£) à chaque valeur d'un champ de devise

Créez un champ calculé ajoutant le symbole de la livre (ASCII code 163) avant les montants dans le champ

**Montant\_Facture.** Le champ numérique **Montant\_Facture** est d'abord converti en champ de type caractère, puis les espaces vides de début et de fin sont coupés.

```
DEFINE FIELD Devise_RU COMPUTED CHR(163)+ALLTRIM(STRING(Montant_Facture; 12))
```

## Remarques

### Quand utiliser la fonction CHR( )

Utilisez la fonction CHR( ) pour renvoyer le caractère associé à un code ASCII, y compris les caractères qui ne peuvent pas être saisis directement avec un clavier ou affichés à l'écran. La fonction CHR( ) vous permet de rechercher ces caractères spécifiques dans les champs ou les enregistrements.

### Référencement des caractères NUL

Le référencement des caractères ASCII NUL (nuls), CHR(0), peut produire des résultats imprévisibles dans la mesure où elle est utilisée par Analytics comme qualificatif de texte. Il est préférable de ne pas l'utiliser.

### Fonctions associées

CHR( ) est l'inverse de la fonction ASCII( ).



# Fonction CLEAN( )

Remplace le premier caractère non valide, et tous les caractères suivants, d'une chaîne par des blancs.

## Syntaxe

```
CLEAN(chaîne <;caractères_invalides_supplémentaires>)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	Valeur à partir de laquelle supprimer les caractères non valides par défaut et les caractères non valides supplémentaires.
<i>caractères_invalides_supplémentaires</i> optionnel	caractère	<p>Les caractères non valides que vous souhaitez supprimer de <i>chaîne</i> en plus des caractères non valides par défaut. Vous pouvez indiquer plusieurs caractères supplémentaires non valides :</p> <pre>" ,\\"</pre> <p>Les caractères de tabulation, les caractères Null et les retours chariot ainsi que les caractères saut de ligne sont des caractères non valides par défaut qui sont automatiquement supprimés et qui ne doivent pas nécessairement être spécifiés.</p> <p>Pour signaler le caractère de guillemets doubles en tant que caractère non valide supplémentaire, entourez <i>caractères_invalides_supplémentaires</i> de guillemets simples :</p> <pre>""</pre>

## Sortie

Caractère.

# Exemples

## Exemples basiques

Renvoie "ABC " ("ABC" suivi de 4 espaces vides) :

```
CLEAN("ABC%DEF";"%")
```

Renvoie "1234,56 " ("1234,56" suivi de 6 espaces vides) :

```
CLEAN("1234,56;111;2"; ",")
```

## Remarques

### Quand utiliser la fonction CLEAN( )

Cette fonction permet de vérifier que les champs contenant des données incorrectes sont imprimés correctement. Vous pouvez également l'utiliser pour isoler des parties d'un champ, tel que le nom d'un champ de client qui inclut à la fois le prénom et le nom du client.

### Signaler des guillemets doubles et simples comme non valides

S'il vous faut spécifier les guillemets simples et doubles en tant que caractères non valides, vous devez imbriquer la fonction CLEAN( ) en elle-même :

```
CLEAN(CLEAN(chaine_de_caractères; ""); "")
```

### Application automatique de la fonction CLEAN( )

Dans un script Analytics, vous pouvez automatiquement appliquer la fonction CLEAN( ) à tous les champs caractère en ajoutant SET CLEAN ON au script. Il n'est pas possible d'indiquer des caractères individuels supplémentaires à l'aide de cette option.

# Fonction CMOY()

Renvoie le nom du mois de l'année de la date ou de la DateHeure indiquée. Abréviation de "Mois de l'année au format caractère".

## Syntaxe

```
CMOY(date/DateHeure; longueur)
```

## Paramètres

Nom	Type	Description
<i>date/DateHeure</i>	DateHeure	Le champ, l'expression, ou la valeur littérale pour lesquels renvoyer le nom du mois.
<i>longueur</i>	numérique	Une valeur entre 1 et 9 indiquant la longueur de la chaîne de sortie. Pour afficher les noms raccourcis des mois, indiquez une valeur inférieure.

## Sortie

Caractère.

## Exemples

### Exemples basiques

Renvoie « Décembre » :

```
CMOY(`20141231`; 9)
```

Renvoie « Dec » :

```
CMOY(`20141231 235959`; 3)
```

Renvoie le nom du mois abrégé pour chaque valeur dans le champ **Horodatage\_reçu** :

```
CMOY(Horodatage_reçu; 3)
```

Renvoie le nom complet du mois pour chaque valeur dans le champ **Date\_facture** :

```
CMOY(Date_facture; 9)
```

Renvoie le nom complet du mois 15 jours après chaque valeur dans le champ **Date\_facture** :

```
CMOY(Date_facture + 15; 9)
```

## Remarques

### Informations sur les paramètres

Un champ indiqué pour *date/DateHeure* peut utiliser n'importe quel format de date ou de DateHeure, tant que la définition de champ définit correctement le format.

Si le paramètre *longueur* est inférieur au nom du mois, celui-ci est tronqué pour correspondre à la longueur indiquée. Si le paramètre *longueur* est plus long que le nom du mois, celui-ci est complété avec des espaces.

### Spécification d'une valeur littérale de date ou de DateHeure

Lorsque vous indiquez une valeur littérale de date ou de DateHeure pour *date/DateHeure*, vous êtes limité aux formats de la table ci-dessous, et vous devez entourer la valeur de guillemets inversés - par exemple, `20141231`.

N'utilisez aucun séparateur comme les barres obliques (/) ou les deux points (:) entre les composants individuels des dates ou des heures.

- **Valeurs DateHeure** : vous pouvez utiliser n'importe laquelle des combinaisons de formats de date, de séparateur et d'heure répertoriées dans le tableau ci-dessous. La date doit précéder l'heure, et vous devez utiliser un séparateur entre les deux. Les séparateurs valides sont un espace unique, la lettre « t » ou la lettre « T ».
- **Valeurs heure** : vous devez indiquer l'heure à l'aide de l'horloge de 24 heures. Les décalages par rapport à l'heure universelle coordonnée (UTC) doivent être préfacés d'un signe plus (+) ou d'un signe moins (-).

Exemples de formats	Exemples de valeurs littérales
AAAAMMJJ	`20141231`

Exemples de formats	Exemples de valeurs littérales
AAMMJJ	`141231`
AAAAMMJJ hhmmss	`20141231 235959`
AAMMJJthhmm	`141231t2359`
AAAAMMJJThh	`20141231T23`
AAAAMMJJ hhmmss+/-hhmm (décalage UTC)	`20141231 235959-0500`
AAMMJJ hhmm+/-hh (décalage UTC)	`141231 2359+01`
<p><b>Remarque</b></p> <p>N'utilisez pas hh seul dans le format d'heure principal avec des données qui ont un décalage UTC. Par exemple, évitez : hh+hhmm. Il se peut que les résultats ne soient pas fiables.</p>	

## Fonctions associées

Si vous devez renvoyer le mois de l'année en tant que nombre (1 à 12), utilisez la fonction MONTH() au lieu de CMOY().

# Fonction COS( )

Renvoie le cosinus d'un angle exprimé en radians, avec une précision de 15 décimales.

## Syntaxe

```
COS(radians)
```

## Paramètres

Nom	Type	Description
<i>radians</i>	numérique	Mesure de l'angle en radians.

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 0,5000000000000000 (le nombre de radians spécifié) :

```
COS(1,047197551196598)
```

### Exemples avancés

#### Utilisation d'une valeur d'entrée en degrés

Renvoie 0,5000000000000000 (le cosinus de 60 degrés) :

```
COS(60 * PI( )/180)
```

### Arrondi à 3 décimales

Renvoie 0,500 (le cosinus de 60 degrés, arrondi à 3 décimales) :

```
DEC(COS(60 * PI()/180);3)
```

## Remarques

### Réalisation du test de l'arc mantisse

Les trois fonctions trigonométriques d'Analytics (SIN(), COS() et TAN()) prennent en charge le test de l'arc de mantisse associé à la loi de Benford.

### Conversion de degrés en radians

Si votre entrée est en degrés, vous pouvez utiliser la fonction PI() pour la convertir en radians :  $(\text{degrés} * \text{PI}() / 180) = \text{radians}$ . Si besoin, vous pouvez arrondir ou tronquer la valeur renvoyée à l'aide de la fonction DEC().

# Fonction CTOD( )

Convertit une valeur de date au format Caractère ou Numérique en date. Cette fonction peut également extraire la date d'une valeur de type DateHeure au format Caractère ou Numérique et la renvoyer sous forme de date. Abréviation de "Caractère à Date".

## Syntaxe

```
CTOD(chaîne/chiffre <;format>)
```

## Paramètres

Nom	Type	Description
<i>chaîne/chiffre</i>	caractère numérique	Le champ, l'expression, ou la valeur littérale à convertir en date, ou à partir desquels extraire la date.
<i>format</i> optionnel	caractère	Format de date de <i>chaîne/chiffre</i> . Le <i>format</i> est requis pour les valeurs qui utilisent tout format de date autre que AAAAMMJJ ou AAMMJJ, par exemple "JJ/MM/AAAA".  <div style="border-left: 2px solid #0056b3; padding-left: 10px;"> <p><b>Remarque</b></p> <p>Si vous utilisez la fonction CTOD avec une valeur de type DateHeure qui requiert le paramètre <i>format</i>, indiquez uniquement la partie du format correspondant à la date, et pas la partie correspondant à l'heure. Par exemple :</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <pre>CTOD("31/12/2014 23:59:59"; "JJ/MM/AAAA")</pre> </div> <p>Spécifier la partie heure empêche l'affichage des résultats.</p> </div>

## Sortie

DateHeure. La valeur de date est générée à l'aide du format d'affichage de date Analytics actuel.



# Exemples

## Exemples basiques

### Valeur d'entrée littérale de type caractère

Revoie `20141231` affiché au format 31 déc 2014 en supposant que le format d'affichage actuel de la date dans Analytics soit JJ MMM AAAA :

```
CTOD("20141231")
```

```
CTOD("31/12/2014"; "JJ/MM/AAAA")
```

```
CTOD("20141231 235959")
```

### Valeur d'entrée littérale numérique

Revoie `20141231` affiché au format 31 déc 2014 en supposant que le format d'affichage actuel de la date dans Analytics soit JJ MMM AAAA :

```
CTOD(20141231)
```

```
CTOD(31122014; "DDMMYYYY")
```

```
CTOD(20141231,235959)
```

### Valeur d'entrée de champ de type caractère

Revoie chaque valeur du champ caractère spécifié sous la forme d'une date, à l'aide du format d'affichage de la date d'Analytics actuel :

```
CTOD(Date_facture, "DD/MM/YYYY")
```

```
CTOD(Reçu_horodatage)
```

### Valeur d'entrée de champ numérique

Revoie chaque valeur du champ numérique spécifié sous forme de date à l'aide du format d'affichage de la date Analytics actuel :

```
CTOD(Date_échéance; "DDMMYYYY")
```

```
CTOD(Paiement_horodatage)
```

## Exemples avancés

### Comparaison d'un champ Caractère ou Numérique avec une date

Utilisez la fonction CTOD( ) pour comparer une date avec des champs de type caractère ou numérique, contenant des valeurs représentant des dates.

Le filtre ci-après compare deux valeurs :

- le champ numérique **Date\_échéance** stockant les dates sous forme de nombre au format JJMMAAAA
- la valeur date littérale au 1er juillet 2014

```
SET FILTER TO CTOD(Date_échéance; "JJMMAAAA") < `20140701`
```

## Remarques

### Formats date requis

Les champs de caractères et les champs numériques contenant des valeurs de date ou de DateHeure doivent correspondre aux formats de la table ci-dessous. Les valeurs de DateHeure peuvent utiliser n'importe quelle combinaison de formats de date, séparateur et heure valides pour leur type de données. La date doit précéder l'heure, et il doit y avoir un séparateur entre les deux.

Les dates, ou la portion correspondant à la date des valeurs DateHeure, peuvent utiliser n'importe quel format de date pris en charge par Analytics, et valide pour ce type de données, tant que les formats autres que AAAAMMJJ ou AAMMJJ sont correctement définis par le paramètre *format*.

Formats de date	Formats des séparateurs	Formats d'heure
<b>Champs de type caractère</b>		
AAAAMMJJ	espace unique	hhmmss hh:mm:ss
AAMMJJ	la lettre 't'	hhmm hh:mm
n'importe quel format de date pris en charge par Analytics, valide pour ce type de	la lettre 'T'	hh

Formats de date	Formats des séparateurs	Formats d'heure
données, s'il est défini par <i>format</i>		
		+/-hhmm +/-hh:mm (décalage UTC)
		+/-hh (décalage UTC)

Formats de date	Formats des séparateurs	Formats d'heure
		R- e- m- a- r- q- u- e  N- '- u- t- i- l- i- s- e- z  p- a- s  h- h  s- e- u- l  d- a- n- s  l- e  f- o- r- m- a- t  d- '- h- e- u- r- e  p- r- i-  n- c- i-

Formats de date	Formats des séparateurs	Formats d'heure
<b>Champs numériques</b>		
AAAAMMJJ	point de décimale	hhmmss
AAMMJJ		hhmm
n'importe quel format de date pris en charge par Analytics, valide pour ce type de données, s'il est défini par <i>format</i>		hh

## Autres fonctions de conversion DateHeure

### Conversion Caractère ou Numérique en DateHeure

Fonction	Description
<a href="#">CTODT()</a>	Convertit une valeur DateHeure au format Caractère ou Numérique en DateHeure. Abréviation de "Caractère à DateHeure".
<a href="#">CTOT()</a>	Convertit une valeur de type heure au format Caractère ou Numérique en heure. Cette fonction peut également extraire l'heure d'une valeur de type DateHeure au format Caractère ou Numérique et la renvoyer sous forme d'heure. Abréviation de "Caractère à Heure".

### Conversion DateHeure en Caractère

Fonction	Description
<a href="#">DATE()</a>	Extrait la date d'une date ou d'une DateHeure spécifiée et la renvoie sous forme de chaîne de caractères. Peut également renvoyer la date du système d'exploitation en cours d'utilisation.
<a href="#">DATETIME()</a>	Convertit une DateHeure en une chaîne de caractères. Peut également renvoyer la DateHeure du système d'exploitation en cours d'utilisation.
<a href="#">TIME()</a>	Extrait l'heure d'une heure ou d'une DateHeure spécifiée et la renvoie sous forme de chaîne de caractères. Peut également renvoyer l'heure du système d'exploitation en cours d'utilisation.

### Conversion Sérielle en DateHeure

Fonction	Description
<a href="#">STOD()</a>	Convertit une date sérielle - c'est-à-dire, une date exprimée sous forme d'intègre - en valeur de

Fonction	Description
	date. Abréviation de "Sérielle à Date".
<a href="#">STODT()</a>	Convertit une valeur DateHeure sérielle - c'est-à-dire, une DateHeure exprimée sous forme d'une intègre, et une portion fractionnelle de 24 heures - en une valeur correspondant à une DateHeure. Abréviation de "Sérielle à DateHeure".
<a href="#">STOT()</a>	Convertit une valeur horaire sérielle - c'est-à-dire une heure exprimée sous forme d'une portion fractionnelle de 24 heures, où 24 heures équivalent à 1 - en une valeur correspondant à une heure. Abréviation de "Sérielle à Heure".

# Fonction CTODT()

Convertit une valeur DateHeure au format Caractère ou Numérique en DateHeure. Abréviation de "Caractère à DateHeure".

## Syntaxe

```
CTODT(chaîne/chiffre <;format>)
```

## Paramètres

Nom	Type	Description
<i>chaîne/chiffre</i>	caractère numérique	Le champ, l'expression, ou la valeur littérale à convertir en DateHeure.
<i>format</i> optionnel	caractère	Format de date de <i>chaîne/chiffre</i> . Le <i>format</i> est requis pour les valeurs qui utilisent tout format de date autre que AAAAMMJJ ou AAMMJJ pour la partie date de la valeur, par exemple "JJ/MM/AAAA".

## Sortie

DateHeure. La valeur de DateHeure est générée à l'aide des formats d'affichage de date et d'heure Analytics actuels.

## Exemples

### Exemples basiques

#### Valeur d'entrée littérale de type caractère

Renvoie `20141231t235959` affiché au format 31 déc 2014 23:59:59 en supposant que les formats d'affichage actuels de la date et de l'heure dans Analytics soient JJ MMM AAAA et hh:mm:ss :

```
CTODT("20141231 235959")
```

```
CTODT("31/12/2014 23:59:59"; "JJ/MM/AAAA hh:mm:ss")
```

## Valeur d'entrée littérale numérique

Renvoie `20141231t235959` affiché au format 31 déc 2014 23:59:59 en supposant que les formats d'affichage actuels de la date et de l'heure dans Analytics soient JJ MMM AAAA et hh:mm:ss :

```
CTODT(20141231,235959)
```

```
CTODT(31122014,235959; "DDMMYYYY,hhmmss")
```

## Valeur d'entrée de champ de type caractère

Renvoie chaque valeur du champ de type caractère **Reçu\_horodatage** sous forme de DateHeure à l'aide du format d'affichage de la date Analytics actuel :

```
CTODT(Reçu_horodatage; "DD/MM/YYYY hh:mm:ss")
```

## Valeur d'entrée de champ numérique

Renvoie chaque valeur du champ numérique **Paiement\_horodatage** sous forme de DateHeure, à l'aide du format d'affichage de la date d'Analytics actuel :

```
CTODT(Paiement_horodatage; "JJ/MM/AAAA hh:mm:ss")
```

## Exemples avancés

### Comparaison d'un champ Caractère ou Numérique avec une DateHeure

Utilisez la fonction CTODT ( ) pour comparer une DateHeure avec des champs de type caractère ou numérique, contenant des valeurs représentant des DateHeures.

Le filtre ci-après compare deux valeurs :

- le champ de type caractère **Reçu\_horodatage** stockant les DateHeures sous forme de données de type caractère au format JJ/MM/AAAA hh:mm:ss
- la valeur DateHeure littérale au 1er juillet 2014 13:30:00

```
SET FILTER TO CTODT(Reçu_horodatage; "JJ/MM/AAAA hh:mm:ss") < `20140701t133000`
```



# Remarques

## Formats DateHeure requis

Les champs de caractères et les champs numériques contenant des valeurs de DateHeure doivent correspondre aux formats de la table ci-dessous. Les valeurs de DateHeure peuvent utiliser n'importe quelle combinaison de formats de date, séparateur et heure valides pour leur type de données. La date doit précéder l'heure, et il doit y avoir un séparateur entre les deux.

La portion correspondant à la date des valeurs peut utiliser n'importe quel format de date pris en charge par Analytics, et valide pour ce type de données, tant que les formats autres que AAAAMMJJ ou AAMMJJ sont correctement définis par *format*. Si vous utilisez *format* vous devez également indiquer le format de l'heure, qui doit être l'un des formats d'heure qui apparaissent dans la table ci-dessous.

Analytics reconnaît automatiquement le séparateur entre la date et les portions correspondant à l'heure des valeurs de DateHeure, c'est pourquoi il n'est pas nécessaire d'indiquer le séparateur dans *format*. Vous pouvez préciser le séparateur si vous le souhaitez.

Formats de date	Formats des séparateurs	Formats d'heure
<b>Champs de type caractère</b>		
AAAAMMJJ	espace unique	hhmmss hh:mm:ss
AAMMJJ	la lettre 't'	hhmm hh:mm
n'importe quel format de date pris en charge par Analytics, valide pour ce type de données, s'il est défini par <i>format</i>	la lettre 'T'	hh
		+/-hhmm +/-hh:mm (décalage UTC)
		+/-hh (décalage UTC)

Formats de date	Formats des séparateurs	Formats d'heure
-----------------	-------------------------	-----------------

--	--	--

R-  
e-  
m-  
a-  
r-  
q-  
u-  
e  
N-  
'-  
u-  
t-  
i-  
l-  
i-  
s-  
e-  
z  
p-  
a-  
s  
h-  
h  
s-  
e-  
u-  
l  
d-  
a-  
n-  
s  
l-  
e  
f-  
o-  
r-  
m-  
a-  
t  
d-  
'-  
h-  
e-  
u-  
r-  
e

p-  
r-  
i-  
n-  
c-  
i-

Formats de date	Formats des séparateurs	Formats d'heure
<b>Champs numériques</b>		
AAAAMMJJ	point de décimale	hhmmss
AAMMJJ		hhmm
n'importe quel format de date pris en charge par Analytics, valide pour ce type de données, s'il est défini par <i>format</i>		hh

## Autres fonctions de conversion DateHeure

### Conversion Caractère ou Numérique en DateHeure

Fonction	Description
<a href="#">CTOD()</a>	Convertit une valeur de date au format Caractère ou Numérique en date. Cette fonction peut également extraire la date d'une valeur de type DateHeure au format Caractère ou Numérique et la renvoyer sous forme de date. Abréviation de "Caractère à Date".
<a href="#">CTOT()</a>	Convertit une valeur de type heure au format Caractère ou Numérique en heure. Cette fonction peut également extraire l'heure d'une valeur de type DateHeure au format Caractère ou Numérique et la renvoyer sous forme d'heure. Abréviation de "Caractère à Heure".

### Conversion DateHeure en Caractère

Fonction	Description
<a href="#">DATE()</a>	Extrait la date d'une date ou d'une DateHeure spécifiée et la renvoie sous forme de chaîne de caractères. Peut également renvoyer la date du système d'exploitation en cours d'utilisation.
<a href="#">DATETIME()</a>	Convertit une DateHeure en une chaîne de caractères. Peut également renvoyer la DateHeure du système d'exploitation en cours d'utilisation.
<a href="#">TIME()</a>	Extrait l'heure d'une heure ou d'une DateHeure spécifiée et la renvoie sous forme de chaîne de caractères. Peut également renvoyer l'heure du système d'exploitation en cours d'utilisation.

### Conversion Sérielle en DateHeure

Fonction	Description
<a href="#">STOD()</a>	Convertit une date sérielle - c'est-à-dire, une date exprimée sous forme d'intègre - en valeur de

Fonction	Description
	date. Abréviation de "Sérielle à Date".
<a href="#">STODT()</a>	Convertit une valeur DateHeure sérielle - c'est-à-dire, une DateHeure exprimée sous forme d'une intègre, et une portion fractionnelle de 24 heures - en une valeur correspondant à une DateHeure. Abréviation de "Sérielle à DateHeure".
<a href="#">STOT()</a>	Convertit une valeur horaire sérielle - c'est-à-dire une heure exprimée sous forme d'une portion fractionnelle de 24 heures, où 24 heures équivalent à 1 - en une valeur correspondant à une heure. Abréviation de "Sérielle à Heure".

# Fonction CTOT()

Convertit une valeur de type heure au format Caractère ou Numérique en heure. Cette fonction peut également extraire l'heure d'une valeur de type DateHeure au format Caractère ou Numérique et la renvoyer sous forme d'heure. Abréviation de "Caractère à Heure".

## Syntaxe

```
CTOT(chaîne/chiffre)
```

## Paramètres

Nom	Type	Description
<i>chaîne/chiffre</i>	caractère numérique	Le champ, l'expression, ou la valeur littérale à convertir en heure, ou à partir desquels extraire l'heure.

## Sortie

DateHeure. La valeur de type heure est générée à l'aide du format d'affichage de l'heure Analytics actuel.

## Exemples

### Exemples basiques

#### Valeur d'entrée littérale de type caractère

Renvoie `t235959` affiché sous la forme 23:59:59 en supposant que le format d'affichage actuel de l'heure dans Analytics soit hh:mm:ss :

```
CTOT("t235959")
```

```
CTOT("23:59:59")
```

```
CTOT("20141231 235959")
```

## Valeur d'entrée littérale numérique

Renvoie `t235959` affiché sous la forme 23:59:59 en supposant que le format d'affichage actuel de l'heure dans Analytics soit hh:mm:ss :

```
CTOT(,235959)
```

```
CTOT(0,235959)
```

```
CTOT(20141231,235959)
```

## Valeur d'entrée de champ de type caractère

Renvoie chaque valeur du champ de type caractère **Heure\_connexion** sous la forme d'une heure, à l'aide du format d'affichage de l'heure d'Analytics actuel :

```
CTOT(Heure_connexion)
```

## Valeur d'entrée de champ numérique

Renvoie chaque valeur du champ numérique **Paiement\_dateheure** sous forme d'heure, sans partie date, à l'aide du format d'affichage de l'heure d'Analytics actuel :

```
CTOT(Paiement_dateheure)
```

## Exemples avancés

### Comparaison d'un champ Caractère ou Numérique avec une heure

Utilisez la fonction CTOT( ) pour comparer une heure avec des champs de type caractère ou numérique, contenant des valeurs représentant des heures.

Le filtre ci-après compare deux valeurs :

- le champ numérique **Heure\_connexion** stockant des heures sous forme de données numériques
- la valeur heure littérale 09:30:00

```
SET FILTER TO CTOT(Heure_connexion) > `t093000`
```

# Remarques

## Formats DateHeure requis

Les champs de caractères et les champs numériques contenant des valeurs d'heure ou de DateHeure doivent correspondre aux formats de la table ci-dessous.

Les valeurs d'heure peuvent utiliser n'importe quelle combinaison de format de séparateur et heure. Il doit y avoir un séparateur avant la valeur d'heure, ou des deux-points entre les composants de l'heure, pour que la fonction s'exécute correctement.

Les valeurs de DateHeure peuvent utiliser n'importe quelle combinaison de formats de date, séparateur et heure valides pour leur type de données. La date doit précéder l'heure, et il doit y avoir un séparateur entre les deux.

Utilisez la fonction CTOD( ) si vous souhaitez convertir une valeur de type date au format Caractère ou Numérique en date, ou extraire la date d'une valeur de type DateHeure au format Caractère ou Numérique et la renvoyer sous forme de date.

Utilisez la fonction CTODT( ) si vous souhaitez convertir une valeur de type DateHeure au format Caractère ou Numérique en DateHeure.

Formats de date	Formats des séparateurs	Formats d'heure
<b>Champs de type caractère</b>		
AAAAMMJJ	espace unique	hhmmss hh:mm:ss
AAMMJJ	la lettre 't'	hhmm hh:mm
	la lettre 'T'	hh
		+/-hhmm +/-hh:mm (décalage UTC)
		+/-hh (décalage UTC)

Formats de date	Formats des séparateurs	Formats d'heure
		<p><b>Remarque :</b></p> <p>N'utilisez pas hh seul dans le format d'heure principal avec des données qui ont un décalage UTC. Par exemple, évitez : hh+hhmm. Les résultats peuvent être non fiables.)</p>
Champs numériques		
AAAAMMJJ	point de décimale	hhmmss
AAMMJJ		hhmm
		hh

## Autres fonctions de conversion DateHeure

### Conversion Caractère ou Numérique en DateHeure

Fonction	Description
<a href="#">CTOD()</a>	Convertit une valeur de date au format Caractère ou Numérique en date. Cette fonction peut également extraire la date d'une valeur de type DateHeure au format Caractère ou Numérique et la renvoyer sous forme de date. Abréviation de "Caractère à Date".
<a href="#">CTODT()</a>	Convertit une valeur DateHeure au format Caractère ou Numérique en DateHeure. Abréviation de "Caractère à DateHeure".

### Conversion DateHeure en Caractère

Fonction	Description
<a href="#">DATE()</a>	Extrait la date d'une date ou d'une DateHeure spécifiée et la renvoie sous forme de chaîne de caractères. Peut également renvoyer la date du système d'exploitation en cours d'utilisation.
<a href="#">DATETIME()</a>	Convertit une DateHeure en une chaîne de caractères. Peut également renvoyer la DateHeure du système d'exploitation en cours d'utilisation.
<a href="#">TIME()</a>	Extrait l'heure d'une heure ou d'une DateHeure spécifiée et la renvoie sous forme de chaîne de caractères. Peut également renvoyer l'heure du système d'exploitation en cours



Fonction	Description
	d'utilisation.

## Conversion Sérielle en DateHeure

Fonction	Description
<a href="#">STOD()</a>	Convertit une date sérielle - c'est-à-dire, une date exprimée sous forme d'intègre - en valeur de date. Abréviation de "Sérielle à Date".
<a href="#">STODT()</a>	Convertit une valeur DateHeure sérielle - c'est-à-dire, une DateHeure exprimée sous forme d'une intègre, et une portion fractionnelle de 24 heures - en une valeur correspondant à une DateHeure. Abréviation de "Sérielle à DateHeure".
<a href="#">STOT()</a>	Convertit une valeur horaire sérielle - c'est-à-dire une heure exprimée sous forme d'une portion fractionnelle de 24 heures, où 24 heures équivalent à 1 - en une valeur correspondant à une heure. Abréviation de "Sérielle à Heure".

# Fonction CUMIPMT()

Renvoie le montant cumulé des intérêts payés sur un emprunt pendant une plage de périodes.

## Syntaxe

```
CUMIPMT(taux; périodes; quantité; période_début; période_fin <; type>)
```

## Paramètres

Nom	Type	Description
<i>taux</i>	numérique	Taux d'intérêt par période.
<i>périodes</i>	numérique	Nombre total de périodes de paiement.
<i>quantité</i>	numérique	Capital de l'emprunt.
<i>période_début</i>	numérique	Première période du calcul. <i>période_début</i> ne peut pas être 0.
<i>période_fin</i>	numérique	Dernière période du calcul. <i>période_fin</i> ne peut pas être supérieure au nombre total de périodes de paiement.
<i>type</i> optionnel	numérique	Date des paiements : <ul style="list-style-type: none"> <li>○ 0 : paiement en fin de période</li> <li>○ 1 : paiement en début de période</li> </ul> Si vous ignorez ce paramètre, c'est la valeur par défaut de 0 qui est utilisée.

### Remarque

Vous devez utiliser des périodes temporelles cohérentes lorsque vous spécifiez *taux* et *périodes* pour vous assurer que vous spécifiez un taux d'intérêt **par période**.

Par exemple :

- dans le cas de paiements mensuels, sur un emprunt ou un investissement d'une durée de deux ans avec un taux d'intérêt de 5 % par an, spécifiez 0,05/12 pour *taux* et 2 \* 12 pour *périodes*
- dans le cas de paiements annuels sur le même prêt ou investissement, indiquez 0,05 pour *taux* et 2 pour *périodes*

# Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 17437,23, le montant total des intérêts payés la deuxième année d'un emprunt de 275 000 \$ sur vingt-cinq ans à 6,5 % par an dont les paiements ont une échéance à la fin de chaque mois :

```
CUMIPMT(0,065/12; 12*25; 275000; 13; 24; 0)
```

Renvoie 17741,31, le montant total des intérêts payés sur le même emprunt durant la première année de l'emprunt :

```
CUMIPMT(0,065/12; 12*25; 275000; 1; 12; 0)
```

## Remarques

### Fonctions associées

La fonction CUMPRINC( ) est le complément de la fonction CUMIPMT( ).

La fonction IPMT( ) calcule les intérêts payés pour une seule période.

# Fonction CUMPRINC()

Renvoie le montant cumulé du remboursement du capital sur un emprunt pendant une plage de périodes.

## Syntaxe

```
CUMPRINC(taux; périodes; quantité; période_début; période_fin <;type>)
```

## Paramètres

Nom	Type	Description
<i>taux</i>	numérique	Taux d'intérêt par période.
<i>périodes</i>	numérique	Nombre total de périodes de paiement.
<i>quantité</i>	numérique	Capital de l'emprunt.
<i>période_début</i>	numérique	Première période du calcul. <i>période_début</i> ne peut pas être 0.
<i>période_fin</i>	numérique	Dernière période du calcul. <i>période_fin</i> ne peut pas être supérieure au nombre total de périodes de paiement.
<i>type</i> optionnel	numérique	Date des paiements : <ul style="list-style-type: none"> <li>○ 0 : paiement en fin de période</li> <li>○ 1 : paiement en début de période</li> </ul> Si vous ignorez ce paramètre, c'est la valeur par défaut de 0 qui est utilisée.

### Remarque

Vous devez utiliser des périodes temporelles cohérentes lorsque vous spécifiez *taux* et *périodes* pour vous assurer que vous spécifiez un taux d'intérêt **par période**.

Par exemple :

- dans le cas de paiements mensuels, sur un emprunt ou un investissement d'une durée de deux ans avec un taux d'intérêt de 5 % par an, spécifiez 0,05/12 pour *taux* et 2 \* 12 pour *périodes*
- dans le cas de paiements annuels sur le même prêt ou investissement, indiquez 0,05 pour *taux* et 2 pour *périodes*

# Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 4844,61, le montant total du capital payé la deuxième année d'un emprunt de 275 000 \$ sur vingt-cinq ans à 6,5 % par an dont les paiements ont une échéance à la fin de chaque mois :

```
CUMPRINC(0,065/12; 12*25; 275000; 13; 24; 0)
```

Renvoie 367,24, le montant du capital payé sur le même emprunt lors du dernier mois de celui-ci :

```
CUMPRINC(0,065/12; 12*25; 275000; 1; 1; 0)
```

## Remarques

### Fonctions associées

La fonction CUMIPMT( ) est le complément de la fonction CUMPRINC( ).

La fonction PPMT( ) calcule le capital payé pour une seule période.

# Fonction DATE()

Extrait la date d'une date ou d'une DateHeure spécifiée et la renvoie sous forme de chaîne de caractères. Peut également renvoyer la date du système d'exploitation en cours d'utilisation.

## Syntaxe

```
DATE(<date/DateHeure> <;format>)
```

## Paramètres

Nom	Type	Description
<i>date/DateHeure</i> optionnel	DateHeure	Le champ, l'expression, ou la valeur littérale à partir desquels extraire la date. Si ce paramètre est omis, c'est la date du système d'exploitation en cours d'utilisation qui est renvoyée.
<i>format</i> optionnel	caractère	Le format à appliquer à la chaîne de sortie, par exemple "JJ/MM/AAAA".. Si ce paramètre est ignoré, le format d'affichage de la date actuel d'Analytics est utilisé. Vous ne pouvez pas indiquer de <i>format</i> si vous avez ignoré <i>date/DateHeure</i> .

## Sortie

Caractère.

## Exemples

### Exemples basiques

Renvoie "20141231" dans le format d'affichage de la date actuelle d'Analytics :

```
DATE(`20141231 235959`)
```

Renvoie "31-Déc-2014" :

```
DATE(`20141231 235959`; "DD-MMM-YYYY")
```

Renvoie la date du système d'exploitation en cours d'utilisation sous forme de chaîne de caractères, au format d'affichage de la date actuelle d'Analytics.

```
DATE()
```

Renvoie chaque valeur du champ **Reçu\_horodatage** sous forme de chaîne de caractères à l'aide du format d'affichage de la date Analytics actuel :

```
DATE(Reçu_horodatage)
```

Renvoie chaque valeur du champ **Reçu\_horodatage** sous forme de chaîne de caractères à l'aide du format d'affichage de la date spécifié :

```
DATE(Reçu_horodatage; "DD/MM/YYYY")
```

## Remarques

### Longueur de la chaîne de sortie

La chaîne de sortie comporte toujours 12 caractères. Si le *format* indiqué pour générer les résultats, ou le format d'affichage de la date d'Analytics, comporte moins de 12 caractères, la chaîne générée est complétée par des espaces en fin de chaîne.

### Informations sur les paramètres

Un champ indiqué pour *date/DateHeure* peut utiliser n'importe quel format de date ou de DateHeure, tant que la définition de champ définit correctement le format.

Si vous utilisez *format* pour contrôler la manière dont la chaîne de sortie s'affiche, vous pouvez utiliser n'importe quel format d'affichage de la date pris en charge par Analytics. Par exemple :

- JJ/MM/AAAA
- MM-JJ-AA
- JJ MMM AAAA

*format* doit être indiqué à l'aide de guillemets simples ou doubles - par exemple, "JJ MMM AAAA".

### Spécification d'une valeur littérale de date ou de DateHeure

Lorsque vous indiquez une valeur littérale de date ou de DateHeure pour *date/DateHeure*, vous êtes limité aux formats de la table ci-dessous, et vous devez entourer la valeur de guillemets inversés - par exemple, `20141231`.

N'utilisez aucun séparateur comme les barres obliques (/) ou les deux points (:) entre les composants individuels des dates ou des heures.

- **Valeurs DateHeure** : vous pouvez utiliser n'importe laquelle des combinaisons de formats de date, de séparateur et d'heure répertoriées dans le tableau ci-dessous. La date doit précéder l'heure, et vous devez utiliser un séparateur entre les deux. Les séparateurs valides sont un espace unique, la lettre « t » ou la lettre « T ».
- **Valeurs heure** : vous devez indiquer l'heure à l'aide de l'horloge de 24 heures. Les décalages par rapport à l'heure universelle coordonnée (UTC) doivent être préfacés d'un signe plus (+) ou d'un signe moins (-).

Exemples de formats	Exemples de valeurs littérales
AAAAMMJJ	`20141231`
AAMMJJ	`141231`
AAAAMMJJ hhmmss	`20141231 235959`
AAMMJJthhmm	`141231t2359`
AAAAMMJJThh	`20141231T23`
AAAAMMJJ hhmmss+/-hhmm (décalage UTC)	`20141231 235959-0500`
AAMMJJ hhmm+/-hh (décalage UTC)	`141231 2359+01`
<p><b>Remarque</b></p> <p>N'utilisez pas hh seul dans le format d'heure principal avec des données qui ont un décalage UTC. Par exemple, évitez : hh+hhmm. Il se peut que les résultats ne soient pas fiables.</p>	

## Fonctions associées

Si vous devez renvoyer l'heure actuelle du système d'exploitation sous forme de valeur DateHeure, utilisez TODAY() plutôt que DATE().



# Autres fonctions de conversion DateHeure

## Conversion DateHeure en Caractère

Fonction	Description
<a href="#">DATETIME()</a>	Convertit une DateHeure en une chaîne de caractères. Peut également renvoyer la DateHeure du système d'exploitation en cours d'utilisation.
<a href="#">TIME()</a>	Extrait l'heure d'une heure ou d'une DateHeure spécifiée et la renvoie sous forme de chaîne de caractères. Peut également renvoyer l'heure du système d'exploitation en cours d'utilisation.

## Conversion Caractère ou Numérique en DateHeure

Fonction	Description
<a href="#">CTOD()</a>	Convertit une valeur de date au format Caractère ou Numérique en date. Cette fonction peut également extraire la date d'une valeur de type DateHeure au format Caractère ou Numérique et la renvoyer sous forme de date. Abréviation de "Caractère à Date".
<a href="#">CTODT()</a>	Convertit une valeur DateHeure au format Caractère ou Numérique en DateHeure. Abréviation de "Caractère à DateHeure".
<a href="#">CTOT()</a>	Convertit une valeur de type heure au format Caractère ou Numérique en heure. Cette fonction peut également extraire l'heure d'une valeur de type DateHeure au format Caractère ou Numérique et la renvoyer sous forme d'heure. Abréviation de "Caractère à Heure".

## Conversion Sérielle en DateHeure

Fonction	Description
<a href="#">STOD()</a>	Convertit une date sérielle - c'est-à-dire, une date exprimée sous forme d'intègre - en valeur de date. Abréviation de "Sérielle à Date".
<a href="#">STODT()</a>	Convertit une valeur DateHeure sérielle - c'est-à-dire, une DateHeure exprimée sous forme d'une intègre, et une portion fractionnelle de 24 heures - en une valeur correspondant à une DateHeure. Abréviation de "Sérielle à DateHeure".
<a href="#">STOT()</a>	Convertit une valeur horaire sérielle - c'est-à-dire une heure exprimée sous forme d'une portion fractionnelle de 24 heures, où 24 heures équivalent à 1 - en une valeur correspondant à une heure. Abréviation de "Sérielle à Heure".

# Fonction DATETIME()

Convertit une DateHeure en une chaîne de caractères. Peut également renvoyer la DateHeure du système d'exploitation en cours d'utilisation.

## Syntaxe

```
DATETIME(<DateHeure> <;format>)
```

## Paramètres

Nom	Type	Description
<i>DateHeure</i> optionnel	DateHeure	Le champ, l'expression, ou la valeur littérale à convertir. Si ce paramètre est omis, c'est la date du système d'exploitation en cours d'utilisation qui est renvoyée.
<i>format</i> optionnel	caractère	Le format à appliquer à la chaîne de sortie, par exemple "JJ/MM/AAAA".. Si ce paramètre est ignoré, le format d'affichage de la date actuel d'Analytics est utilisé. Vous ne pouvez pas indiquer de <i>format</i> si vous avez ignoré <i>date/DateHeure</i> .

## Sortie

Caractère.

## Exemples

### Exemples basiques

#### Entrée DateHeure littérale

Renvoie "20141231 235959" dans les formats d'affichage des date et heures actuelles d'Analytics :

```
DATETIME('20141231 235959')
```

Renvoie "31-Dec-2014 11:59 P" :

```
DATETIME(`20141231 235959`; "DD-MMM-YYYY hh:mm A")
```

Renvoie les date et heure du système d'exploitation en cours d'utilisation sous forme de chaîne de caractères, au format d'affichage des date et heures actuelles d'Analytics.

```
DATETIME()
```

## Entrée de champ

Renvoie chaque valeur du champ **Reçu\_horodatage** sous forme de chaîne de caractères à l'aide des formats d'affichage des date et heure Analytics actuelles :

```
DATETIME(Reçu_horodatage)
```

Renvoie chaque valeur du champ **Reçu\_horodatage** sous forme de chaîne de caractères à l'aide des formats d'affichage des date et heure spécifiés :

```
DATETIME(Reçu_horodatage; "DD/MM/YYYY hh:mm:ss")
```

## Remarques

### Longueur de la chaîne de sortie

La chaîne de sortie comporte toujours 27 caractères. Si le *format* indiqué pour générer les résultats ou que les formats d'affichage de la date et de l'heure d'Analytics comportent moins de 27 caractères, la chaîne générée est complétée par des espaces en fin de chaîne.

### Informations sur les paramètres

Un champ indiqué pour *DateHeure* peut utiliser n'importe quel format de DateHeure, tant que la définition de champ définit correctement le format.

Si vous utilisez *format* pour contrôler la manière dont la chaîne de sortie est affichée, vous êtes limité aux formats de la table ci-dessous.

- Vous pouvez utiliser n'importe quelle combinaison des formats de date, d'heure et formats AM/PM.
- La date doit précéder l'heure. Placer un séparateur entre les deux n'est pas requis car Analytics utilise automatiquement un espace unique en tant que séparateur dans la chaîne de sortie.
- Le format AM/PM est facultatif et il est placé en dernier.
- *format* doit être indiqué à l'aide de guillemets simples ou doubles.

Par exemple : "JJ MMM AAAA hh:mm:ss AM"

Formats de date	Formats d'heure	Formats AM/PM	Exemples
tous les formats d'affichage de la date pris en charge par Analytics	hh:mm:ss	aucun horloge de 24 heures	"JJ/MM/AAAA hh:mm:ss"
	hhmmss	AM, ou PM horloge de 12 heures	"MMJJAA hhmmss PM"
	hh:mm	A, ou P horloge de 12 heures	"JJ-MMM-AAAA hh:mm A"
	hhmm		
	hh		

### Spécification d'une valeur littérale de DateHeure

Lorsque vous indiquez une valeur littérale de DateHeure pour *DateHeure*, vous êtes limité aux formats de la table ci-dessous, et vous devez entourer la valeur de guillemets inversés - par exemple, `20141231 235959`.

N'utilisez aucun séparateur comme les barres obliques (/) ou les deux points (:) entre les composants individuels des dates ou des heures.

- **Valeurs DateHeure** : vous pouvez utiliser n'importe laquelle des combinaisons de formats de date, de séparateur et d'heure répertoriées dans le tableau ci-dessous. La date doit précéder l'heure, et vous devez utiliser un séparateur entre les deux. Les séparateurs valides sont un espace unique, la lettre « t » ou la lettre « T ».
- **Valeurs heure** : vous devez indiquer l'heure à l'aide de l'horloge de 24 heures. Les décalages par rapport à l'heure universelle coordonnée (UTC) doivent être préfacés d'un signe plus (+) ou d'un signe moins (-).

Exemples de formats	Exemples de valeurs littérales
AAAAMMJJ hhmmss	`20141231 235959`
AAMMJJthhmm	`141231t2359`
AAAAMMJJThh	`20141231T23`
AAAAMMJJ hhmmss+/-hhmm (décalage UTC)	`20141231 235959-0500`
AAMMJJ hhmm+/-hh	`141231 2359+01`

Exemples de formats	Exemples de valeurs littérales
(décalage UTC)	
<p><b>Remarque</b></p> <p>N'utilisez pas hh seul dans le format d'heure principal avec des données qui ont un décalage UTC. Par exemple, évitez : hh+hhmm. Il se peut que les résultats ne soient pas fiables.</p>	

## Autres fonctions de conversion DateHeure

### Conversion DateHeure en Caractère

Fonction	Description
<a href="#">DATE()</a>	Extrait la date d'une date ou d'une DateHeure spécifiée et la renvoie sous forme de chaîne de caractères. Peut également renvoyer la date du système d'exploitation en cours d'utilisation.
<a href="#">TIME()</a>	Extrait l'heure d'une heure ou d'une DateHeure spécifiée et la renvoie sous forme de chaîne de caractères. Peut également renvoyer l'heure du système d'exploitation en cours d'utilisation.

### Conversion Caractère ou Numérique en DateHeure

Fonction	Description
<a href="#">CTOD()</a>	Convertit une valeur de date au format Caractère ou Numérique en date. Cette fonction peut également extraire la date d'une valeur de type DateHeure au format Caractère ou Numérique et la renvoyer sous forme de date. Abréviations de "Caractère à Date".
<a href="#">CTODT()</a>	Convertit une valeur DateHeure au format Caractère ou Numérique en DateHeure. Abréviations de "Caractère à DateHeure".
<a href="#">CTOT()</a>	Convertit une valeur de type heure au format Caractère ou Numérique en heure. Cette fonction peut également extraire l'heure d'une valeur de type DateHeure au format Caractère ou Numérique et la renvoyer sous forme d'heure. Abréviations de "Caractère à Heure".

### Conversion Sérielle en DateHeure

Fonction	Description
<a href="#">STOD()</a>	Convertit une date sérielle - c'est-à-dire, une date exprimée sous forme d'intègre - en valeur de date. Abréviations de "Sérielle à Date".

Fonction	Description
<a href="#">STODT()</a>	Convertit une valeur DateHeure sérielle - c'est-à-dire, une DateHeure exprimée sous forme d'une intègre, et une portion fractionnelle de 24 heures - en une valeur correspondant à une DateHeure. Abréviation de "Sérielle à DateHeure".
<a href="#">STOT()</a>	Convertit une valeur horaire sérielle - c'est-à-dire une heure exprimée sous forme d'une portion fractionnelle de 24 heures, où 24 heures équivalent à 1 - en une valeur correspondant à une heure. Abréviation de "Sérielle à Heure".

# Fonction DAY( )

Extrait le jour du mois d'une date ou d'une DateHeure spécifiée et le renvoie sous forme de valeur numérique (1 à 31).

## Syntaxe

```
DAY(date/DateHeure)
```

## Paramètres

Nom	Type	Description
<i>date/DateHeure</i>	DateHeure	Le champ, l'expression, ou la valeur littérale à partir desquels extraire le jour.

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 31 :

```
DAY('20141231')
```

```
DAY('20141231 235959')
```

Renvoie le jour du mois pour chaque valeur dans le champ **Date\_facture** :

```
DAY(Date_facture)
```

# Remarques

## Informations sur les paramètres

Un champ indiqué pour *date/DateHeure* peut utiliser n'importe quel format de date ou de DateHeure, tant que la définition de champ définit correctement le format.

## Spécification d'une valeur littérale de date ou de DateHeure

Lorsque vous indiquez une valeur littérale de date ou de DateHeure pour *date/DateHeure*, vous êtes limité aux formats de la table ci-dessous, et vous devez entourer la valeur de guillemets inversés - par exemple, `20141231`.

N'utilisez aucun séparateur comme les barres obliques (/) ou les deux points (:) entre les composants individuels des dates ou des heures.

- **Valeurs DateHeure** : vous pouvez utiliser n'importe laquelle des combinaisons de formats de date, de séparateur et d'heure répertoriées dans le tableau ci-dessous. La date doit précéder l'heure, et vous devez utiliser un séparateur entre les deux. Les séparateurs valides sont un espace unique, la lettre « t » ou la lettre « T ».
- **Valeurs heure** : vous devez indiquer l'heure à l'aide de l'horloge de 24 heures. Les décalages par rapport à l'heure universelle coordonnée (UTC) doivent être préfacés d'un signe plus (+) ou d'un signe moins (-).

Exemples de formats	Exemples de valeurs littérales
AAAAMMJJ	`20141231`
AAMMJJ	`141231`
AAAAMMJJ hhmmss	`20141231 235959`
AAMMJJthhmm	`141231t2359`
AAAAMMJJThh	`20141231T23`
AAAAMMJJ hhmmss+/-hhmm (décalage UTC)	`20141231 235959-0500`
AAMMJJ hhmm+/-hh (décalage UTC)	`141231 2359+01`



Exemples de formats	Exemples de valeurs littérales
<p><b>Remarque</b></p> <p>N'utilisez pas hh seul dans le format d'heure principal avec des données qui ont un décalage UTC. Par exemple, évitez : hh+hhmm. Il se peut que les résultats ne soient pas fiables.</p>	

## Fonctions associées

Si vous devez renvoyer :

- le jour de la semaine comme un nombre (de 1 à 7), utilisez DOW( ) au lieu de DAY( )
- le nom du jour de la semaine, utilisez CDOW( ) au lieu de DAY( )

# Fonction DBYTE( )

Renvoie le caractère Unicode situé à la position d'octet indiquée dans un enregistrement.

## Remarque

Cette fonction est spécifique à l'édition Unicode d'Analytics. Elle n'est pas prise en charge dans l'édition non-Unicode.

## Syntaxe

```
DBYTE(octet_emplacement)
```

## Paramètres

Nom	Type	Description
<i>position_octet</i>	numérique	Position d'octet à renvoyer sous forme de valeur caractère. Pour renvoyer une valeur significative, vous devez indiquer le point de départ du caractère à deux octets. En d'autres termes, vous devez indiquer uniquement des nombres impairs dans le paramètre <i>octet_emplacement</i> .

## Sortie

Caractère.

## Exemples

### Exemples basiques

Les exemples illustrent le comportement de la fonction lorsqu'elle est appliquée au champ Unicode ci-dessous (dont la valeur fait 11 caractères, soit 22 octets) 美丽 10072DOE :

Renvoie "丽" :

```
DBYTE(3)
```

Renvoie "D" :

```
DBYTE(17)
```

Renvoie "E" :

```
DBYTE(21)
```

## Remarques

### Quand utiliser DBYTE( )

Utilisez DBYTE( ) pour examiner le contenu d'une position dans un enregistrement, sans avoir à définir un champ à cet effet.

# Fonction DEC( )

Renvoie une valeur ou le résultat d'une expression numérique avec le nombre de décimales indiqué.

## Syntaxe

```
DEC(nombre; décimales)
```

## Paramètres

Nom	Type	Description
<i>nombre</i>	numérique	Valeur ou résultat pour lequel il faut ajuster le nombre de décimales. <ul style="list-style-type: none"> <li>o <b>entiers</b> : des décimales sont ajoutées à la fin du <i>nombre</i> sous forme de zéros de fin.</li> <li>o <b>nombres décimaux</b> : Si le nombre de décimales est réduit, <i>nombre</i> est arrondi, et non tronqué. Si le nombre de décimales augmente, des zéros de fin sont ajoutés à la fin du <i>nombre</i>.</li> </ul>
<i>décimales</i>	numérique	Nombre de décimales à utiliser dans la valeur renvoyée. <div style="border-left: 2px solid #0056b3; padding-left: 10px; margin-top: 10px;"> <p><b>Remarque</b></p> <p>Vous ne pouvez pas utiliser DEC( ) pour augmenter la précision décimale des résultats.</p> <p>Pour obtenir des informations sur la façon d'augmenter la précision décimale, consultez la section <a href="#">Contrôle de l'arrondi dans les expressions numériques</a>.</p> </div>

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 7,00 :

```
DEC(7; 2)
```

Renvoie 7,565 :

```
DEC(7,5647; 3)
```

Renvoie 7,56470 :

```
DEC(7,5647; 5)
```

## Exemples avancés

### Calcul de l'intérêt journalier

Calcule l'intérêt journalier à six décimales pour le champ **Taux\_annuel** :

```
DEC(Taux_annuel; 6) / 365
```

## Remarques

### Quand utiliser DEC( )

Utilisez cette fonction pour ajuster le nombre de décimales dans un champ ou pour arrondir une valeur ou un résultat avec un nombre spécifique de décimales.

### DEC( ) ne peut pas inverser l'arrondi à virgule fixe

Vous ne pouvez pas utiliser la fonction DEC( ) pour inverser l'arrondi standard effectué par les opérations à virgule fixe dans les expressions numériques.

### Exemple

Prenez cette série d'expressions dans Analytics :

```
1,1 * 1,1 = 1,2
1,1 * 1,10 = 1,21
DEC(1,1 * 1,1; 2) = 1,20
```

L'arrondi à virgule fixe signifie que le résultat de  $1,1 * 1,1$  est 1,2, et non 1,21, qui correspond au résultat non arrondi. L'utilisation de DEC( ) pour spécifier un résultat à deux décimales ne crée pas une précision à deux décimales. À la place, elle ajoute un zéro de fin pour créer le nombre de décimales spécifié, sans augmenter la précision.

Pour obtenir des informations sur la façon d'augmenter la précision décimale, consultez la section [Contrôle de l'arrondi dans les expressions numériques](#).

## Fonctions associées

Si vous souhaitez arrondir une valeur au nombre entier le plus proche, utilisez la fonction "Fonction ROUND()" Page 797.

# Fonction DHEX()

Convertit une chaîne Unicode en chaîne hexadécimale.

## Remarque :

Cette fonction est spécifique à l'édition Unicode d'Analytics. Elle n'est pas prise en charge dans l'édition non-Unicode.

## Syntaxe

```
DHEX(champ)
```

## Paramètres

Nom	Type	Description
<i>champ</i>	caractère	La chaîne Unicode à convertir en chaîne hexadécimale.

## Sortie

Caractère.

## Exemples

### Exemples basiques

Renvoie « 004100420043003100320033 » :

```
DHEX("ABC123")
```

## Remarques

### Fonctionnement

DHEX() affiche chaque caractère de deux octets au format gros-boutiste, le plus important de ces caractères étant stocké en premier.

Chaque caractère est représenté par un code de quatre caractères. La chaîne de sortie est quatre fois plus longue que la valeur *champ* et inclut des chiffres compris entre 0 et 9, ainsi que des lettres comprises entre A et F composant les valeurs hexadécimales.

## Fonctions associées

DHEX ( ) est l'inverse de la fonction HTOU( ), qui convertit une chaîne hexadécimale en chaîne Unicode.



# Fonction DICECOEFFICIENT()

Renvoie le coefficient de Dice de deux chaînes de caractères spécifiées : il s'agit de la mesure de la similarité entre deux chaînes.

## Syntaxe

```
DICECOEFFICIENT(chaîne_de_caractères1; chaîne_de_caractères2 <;ngram>)
```

## Paramètres

Nom	Type	Description
<i>chaîne_de_caractères1</i>	caractère	Première chaîne de la comparaison.
<i>chaîne_de_caractères2</i>	caractère	Deuxième chaîne de la comparaison.
<i>ngram</i> optionnel	numérique	<p>Longueur en <i>n</i>-gramme à utiliser.</p> <p>Spécifiez un nombre entier, 1 ou supérieur. Augmenter la longueur <i>ngram</i> rend encore plus strict le critère de similarité entre deux chaînes de caractères.</p> <p>Si vous ne spécifiez pas de longueur, c'est la longueur par défaut de 2 qui est utilisée.</p> <p>Les <i>N</i>-grammes sont les sous-chaînes (blocs de caractères) qui se chevauchent et parmi lesquelles les chaînes de comparaison sont divisées dans le cadre du calcul du coefficient de Dice.</p> <p>Pour des informations détaillées, consultez la section "Remarques" Page 571.</p>

## Sortie

Numérique. La valeur est le coefficient de Dice des deux chaînes de caractères, qui représente le pourcentage du nombre total de *n*-grammes dans les deux chaînes de caractères qui sont identiques. La plage s'étend de 0,0000 à 1,0000 compris.

# Exemples

## Exemples basiques

### Incidence de la longueur en $n$ -grammes sur le résultat

Les trois exemples ci-après comparent les mêmes deux chaînes. Le degré de similarité renvoyé varie selon la longueur en  $n$ -grammes spécifiée.

Renvoie 0,9167 (en utilisant la longueur en  $n$ -grammes par défaut (2), les  $n$ -grammes dans les deux chaînes de caractères sont identiques à 92 %) :

```
DICECOEFFICIENT("125 SW 39TH ST, Suite 100";"Suite 100, 125 SW 39TH ST")
```

Renvoie 1,0000 (en utilisant une longueur en  $n$ -grammes de 1, les  $n$ -grammes dans les deux chaînes de caractères sont identiques à 100 %) :

```
DICECOEFFICIENT("125 SW 39TH ST, Suite 100";"Suite 100, 125 SW 39TH ST"; 1)
```

Renvoie 0,8261 (en utilisant une longueur en  $n$ -grammes de 3, les  $n$ -grammes dans les deux chaînes de caractères sont identiques à 83 %) :

```
DICECOEFFICIENT("125 SW 39TH ST, Suite 100";"Suite 100, 125 SW 39TH ST"; 3)
```

### Entrée de champ

Renvoie le coefficient de Dice de chaque valeur du champ **Adresse** lors de la comparaison avec la chaîne de caractères "125 SW 39TH ST, Suite 100" (basé sur la longueur en  $n$ -grammes par défaut de 2) :

```
DICECOEFFICIENT(Adresse;"125 SW 39TH ST, Suite 100")
```

## Exemples avancés

### Utilisation d'éléments transposés

En diminuant la longueur en  $n$ -grammes et en supprimant les caractères superflus, vous pouvez optimiser DICECOEFFICIENT( ) lorsque vous recherchez des éléments transposés.

Renvoie 0,7368 (en utilisant la longueur en  $n$ -grammes par défaut (2), les  $n$ -grammes dans les deux chaînes de caractères sont identiques à 74 %) :

```
DICECOEFFICIENT("John Smith";"Smith, John")
```

Renvoie 1,0000 (en excluant la virgule entre le nom de famille et le prénom et en utilisant une longueur en  $n$ -grammes de 1, les  $n$ -grammes dans les deux chaînes de caractères sont identiques à 100 %) :

```
DICECOEFFICIENT("John Smith"; EXCLUDE("Smith, John"; ","); 1)
```

## Classer des valeurs en fonction de "125 SW 39TH ST, Suite 100"

Créez le champ calculé **Co\_Dice** pour afficher le coefficient de Dice situé entre "125 SW 39TH ST, Suite 100" et chaque valeur dans le champ **Adresse** :

```
DEFINE FIELD Co_Dice COMPUTED DICECOEFFICIENT(Adresse;"125 SW 39TH ST; Suite 100")
```

Ajoutez le champ calculé **Co\_Dice** à la vue, puis effectuez un tri express dans l'ordre décroissant pour classer toutes les valeurs du champ **Adresse** en fonction de leur similarité avec "125 SW 39TH ST, Suite 100".

## Isoler les doublons approximatifs de "125 SW 39TH ST, Suite 100"

Créez un filtre isolant toutes les valeurs du champ **Adresse** comprises dans un certain degré de similarité avec "125 SW 39TH ST, Suite 100" :

```
SET FILTER TO DICECOEFFICIENT(Adresse;"125 SW 39TH ST, Suite 100") > 0,5
```

Modifier le nombre dans l'expression vous permet de régler le degré de similarité dans les valeurs filtrées.

# Remarques

## Quand utiliser DICECOEFFICIENT( )

Utilisez la fonction DICECOEFFICIENT( ) pour rechercher des valeurs presque identiques (des doublons approximatifs). DICECOEFFICIENT( ) sert aussi à rechercher les valeurs ayant des contenus identiques ou presque identiques, à l'exception des éléments transposés. Par exemple :

- des numéros de téléphone ou des numéros de sécurité sociale, avec des chiffres transposés
- des versions de la même adresse, mise en forme différemment

## Fonctionnement

DICECOEFFICIENT( ) renvoie le coefficient de Dice des deux chaînes de caractères évaluées. Le coefficient de Dice mesure le degré de similarité entre les chaînes, sur une échelle de 0,0000 à 1,0000. Plus la valeur renvoyée est grande, plus les deux chaînes sont similaires :

- **1,0000** : signifie que chaque chaîne de caractères est composée d'un jeu de caractères identique, bien que les caractères puissent être présentés dans un ordre différent et utiliser une casse différente.
- **0,7500** : signifie que les  $n$ -grammes dans les deux chaînes de caractères sont identiques à 75 %.

- **0,0000** : signifie que les deux chaînes de caractères n'ont aucun  $n$ -gramme (expliqué ci-dessous) en commun ou que la longueur spécifiée du  $n$ -gramme utilisé dans le calcul est plus longue que la plus courte des deux chaînes comparées.

## Conseils d'utilisation

- **Filtre ou tri** : Filtrer ou trier les valeurs d'un champ en fonction de leur coefficient de Dice identifie les valeurs ressemblant le plus à la chaîne de comparaison.
- **Sensibilité à la casse** : Cette fonction n'est pas sensible à la casse (par ex., « MARTIN » est équivalent à « martin »).
- **Espaces de début et de fin** : La fonction tronque également les espaces de début et de fin dans les champs de manière automatique. Vous n'avez donc pas besoin d'utiliser la fonction TRIM( ) ou ALLTRIM( ) lorsque vous renseignez un champ comme paramètre.
- **Suppression des éléments génériques** : Les fonctions OMIT( ) et EXCLUDE( ) peuvent améliorer l'efficacité de la fonction DICECOEFFICIENT( ) en supprimant des éléments génériques tels que « Corporation » ou « Inc. » ou des caractères comme les virgules, les points et les esperluettes (&) dans les valeurs de champ.

La suppression d'éléments génériques et de ponctuation recentre la comparaison de chaînes DICECOEFFICIENT( ) uniquement sur la portion des chaînes dans lesquelles une différence significative peut se présenter.

## Mode de calcul du coefficient de Dice

Le coefficient de Dice représente le pourcentage du nombre total de  $n$ -grammes dans les deux chaînes de caractères qui sont identiques.

Le coefficient de Dice est calculé en divisant d'abord les chaînes de caractères à comparer en  $n$ -grammes. Les  $N$ -grammes (aussi appelés  $q$ -grammes) sont des sous-chaînes ou des blocs de caractères qui se chevauchent, d'une longueur de  $n$ . Il est possible de spécifier la longueur de  $n$  à l'aide du paramètre *ngram* ou d'accepter la longueur par défaut de 2.

### Deux noms divisés en $n$ -grammes

Ci-dessous les noms « John Smith » et « John D. Smith » sont divisés en  $n$ -grammes d'une longueur de 2 et en  $n$ -grammes d'une longueur de 3. Les caractères de soulignement indiquent les espaces. Les espaces internes et la ponctuation comptent comme des caractères.

Longueur en $n$ -grammes	"John Smith" $n$ -grammes	"Smith, John D." $n$ -grammes
2	Jo   oh   hn   n_   _S   Sm   mi   it   th	Sm   mi   it   th   h,   ,_   _J   Jo   oh   hn   n_   _D   D.
3	Joh   ohn   hn_   n_S   _Sm   Smi   mit   ith	Smi   mit   ith   th,   h,_   ,_J   _Jo   Joh   ohn   hn_   n_D   _D.

## Formule du coefficient de Dice

Une fois que les  $n$ -grammes ont été définis pour les deux chaînes de caractères à comparer, le calcul est effectué à l'aide de la formule suivante :

- $2 \times \text{nombre de } n\text{-grammes en commun} / \text{nombre total des } n\text{-grammes dans les deux chaînes de caractères}$

Les  $n$ -grammes en commun sont les  $n$ -grammes qui apparaissent dans les deux chaînes de caractères. Par exemple, « ABC » et « BCD » ont en commun le  $n$ -gramme « BC », en supposant une longueur en  $n$ -gramme de 2 (AB | BC et BC | CD).

## Exemples de calcul du coefficient de Dice

Le tableau ci-dessous illustre le calcul du coefficient de Dice pour les deux chaînes de caractères, « John Smith » et « John D. Smith », à l'aide de différentes longueurs en  $n$ -grammes.

Notez que lorsque la longueur en  $n$ -grammes augmente pour la même paire de chaînes de caractères, la valeur du coefficient de Dice diminue, ce qui indique une similarité moindre. Bien que les chaînes de caractères restent identiques, le critère de similarité est plus strict car diviser les chaînes en  $n$ -grammes plus longs implique que les séquences de caractères plus longues doivent correspondre à un  $n$ -gramme pour être considérées comme  $n$ -gramme en commun.

Une autre façon de voir les choses à ce stade serait que la position relative des caractères pèse davantage à mesure que vous augmentez la valeur des  $n$ -grammes. Par contraste, la position relative des caractères n'est pas prise en compte lors de l'utilisation d'une longueur en  $n$ -grammes de 1. La position relative des caractères renvoie à la position des caractères par rapport aux autres, plutôt qu'à leur position absolue au sein d'une chaîne.

### Astuce

Si vous recherchez spécifiquement une transposition, utilisez une longueur en  $n$ -gramme de 1.

Longueur en $n$ -grammes	"John Smith" $n$ -grammes	"Smith, John D." $n$ -grammes	$n$ -grammes en commun	coefficient de Dice
1	J   o   h   n   _   S   m   i   t   h (10 $n$ -grammes)	S   m   i   t   h   ,   _   J   o   h   n   _   D   . (14 $n$ -grammes)	10	$2 \times 10 / (10 + 14) =$ <b>0,8333</b>
2 (par défaut)	Jo   oh   hn   n_   _S   Sm   mi   it   th (9 $n$ -grammes)	Sm   mi   it   th   h,   ,_   _J   Jo   oh   hn   n_   _D   D. (13 $n$ -grammes)	8	$2 \times 8 / (9 + 13) =$ <b>0,7273</b>
3	Joh   ohn   hn_   n_S   _ Sm   Smi   mit   ith (8 $n$ -grammes)	Smi   mit   ith   th,   h,_   ,_J   _Jo   Joh   ohn   hn_   n_D   _D. (12 $n$ -grammes)	6	$2 \times 6 / (8 + 12) =$ <b>0,6000</b>
4	John   ohn_   hn_S   n_Sm   _Smi   Smit   mith	Smit   mith   ith,   th,_   h,_J   ,_Jo   _Joh   John   ohn_   hn_D   n_D.	4	$2 \times 4 / (7 + 11) =$ <b>0,4444</b>

Longueur en $n$ -grammes	"John Smith" $n$ -grammes	"Smith, John D." $n$ -grammes	$n$ -grammes en commun	coefficient de Dice
	(7 $n$ -grammes)	(11 $n$ -grammes)		

## DICECOEFFICIENT( ) comparée à ISFUZZYDUP( ) et LEVDIST( )

L'une des principales différences entre la fonction DICECOEFFICIENT( ), la fonction ISFUZZYDUP( ) et la fonction LEVDIST( ), qui utilisent la distance Levenshtein, c'est que la fonction DICECOEFFICIENT( ) ne met pas l'accent sur la position relative des caractères ou des blocs de caractères, voire les ignore, dans les deux chaînes comparées. La position relative est un critère important dans les fonctions basées sur la distance Levenshtein.

### Comparaison des valeurs avec transposition

Si vous comparez des chaînes de caractères comme des adresses, dans lesquelles des éléments entiers peuvent être transposés, il se peut que la fonction DICECOEFFICIENT( ) soit un meilleur choix. Par exemple, la même adresse avec la transposition de l'élément "Suite" est identifiée comme très similaire par la fonction DICECOEFFICIENT( ), mais très différente avec la fonction LEVDIST( ) :

Paire d'adresses	coefficient de Dice ( $n$ -gramme de 2 par défaut)	Distance Levenshtein
<ul style="list-style-type: none"> <li>o 125 SW 39TH ST, Suite 100</li> <li>o Suite 100, 125 SW 39TH ST</li> </ul>	0,9167	22  (plus la distance Levenshtein est grande, plus les deux chaînes sont différentes)

### Comparaison des valeurs sans transposition

Si la transposition ne présente pas de problème, il se peut que la fonction LEVDIST( ) donne des résultats plus utiles. Par exemple, le même nom d'entreprise avec une ponctuation différente est identifié comme très différent par la fonction DICECOEFFICIENT( ), mais très similaire avec la fonction LEVDIST( ) :

Paire de noms d'entreprise	coefficient de Dice ( $n$ -gramme de 2 par défaut)	Distance Levenshtein
<ul style="list-style-type: none"> <li>o AVS, Inc</li> <li>o A.V.S. Inc</li> </ul>	0,3750	3

# Fonction DIGIT()

Renvoie le chiffre supérieur ou inférieur d'un octet de type Données compactées indiqué.

## Syntaxe

```
DIGIT(position_octet, position)
```

## Paramètres

Nom	Type	Description
<i>position_octet</i>	numérique	Position de l'octet dans l'enregistrement.
<i>position</i>	numérique	Chiffre à renvoyer : <ul style="list-style-type: none"> <li>○ indiquez 1 pour renvoyer la moitié supérieure de l'octet</li> <li>○ indiquez 2 pour renvoyer la moitié inférieure de l'octet</li> </ul>

## Sortie

Numérique.

## Exemples

### Exemples basiques

Un champ compacté avec la valeur 123,45 (00 12 34 5C), contenant deux décimales et commençant à la position 10 d'octet, apparaît dans l'enregistrement de données sous le format suivant :

	Octet 10	Octet 11	Octet 12	Octet 13
UPPER(1)	0	1	3	5
LOWER(2)	0	2	4	C

Renvoie 3 (recherche le chiffre apparaissant en 12e position dans la moitié supérieure de l'octet) :

```
DIGIT(12; 1)
```

Renvoie 4 (recherche le chiffre apparaissant en 12e position dans la moitié inférieure de l'octet) :

```
DIGIT(12; 2)
```

## Remarques

### Fonctionnement

DIGIT ( ) sépare les différentes moitiés d'un octet et renvoie la valeur de l'octet indiqué dans le paramètre de position en tant que chiffre compris entre 0 et 15.

### Quand utiliser la fonction DIGIT( )

Utilisez DIGIT ( ) pour accéder aux demi-octets individuels. Ceci est nécessaire si vous travaillez avec des applications utilisant des champs compactés et alignés en fonction des demi-octets (par exemple, les applications Unisys).



# Fonction DOW( )

Renvoie une valeur numérique (1 à 7) représentant le jour de la semaine de la date ou de la DateHeure indiquée. Abréviations de "Jour de la semaine".

## Syntaxe

```
DOW(date/DateHeure)
```

## Paramètres

Nom	Type	Description
<i>date/DateHeure</i>	DateHeure	Le champ, l'expression ou la valeur littérale à partir desquels extraire le jour de la semaine au format numérique.

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 4, car le 31 décembre 2014 correspond à un mercredi, le 4ème jour de la semaine :

```
DOW(`20141231`)
```

```
DOW(`20141231 235959`)
```

Renvoie le jour de la semaine numérique pour chaque valeur dans le champ **Date\_facture** :

```
DOW(Date_facture)
```

## Exemples avancés

### Identification des transactions ayant lieu un week-end

Utilisez la fonction `DOW()` pour identifier les transactions ayant lieu un week-end. Le filtre ci-dessous isole les dates dans le champ `Date_Trans` qui correspondent à un samedi ou à un dimanche :

```
SET FILTER TO DOW(Date_Trans) = 7 OR DOW(Date_Trans) = 1
```

## Remarques

### Informations sur les paramètres

Un champ indiqué pour `date/DateHeure` peut utiliser n'importe quel format de date ou de `DateHeure`, tant que la définition de champ définit correctement le format.

### Spécification d'une valeur littérale de date ou de `DateHeure`

Lorsque vous indiquez une valeur littérale de date ou de `DateHeure` pour `date/DateHeure`, vous êtes limité aux formats de la table ci-dessous, et vous devez entourer la valeur de guillemets inversés - par exemple, ``20141231``.

N'utilisez aucun séparateur comme les barres obliques (/) ou les deux points (:) entre les composants individuels des dates ou des heures.

- **Valeurs `DateHeure`** : vous pouvez utiliser n'importe laquelle des combinaisons de formats de date, de séparateur et d'heure répertoriées dans le tableau ci-dessous. La date doit précéder l'heure, et vous devez utiliser un séparateur entre les deux. Les séparateurs valides sont un espace unique, la lettre « t » ou la lettre « T ».
- **Valeurs `heure`** : vous devez indiquer l'heure à l'aide de l'horloge de 24 heures. Les décalages par rapport à l'heure universelle coordonnée (UTC) doivent être préfacés d'un signe plus (+) ou d'un signe moins (-).

Exemples de formats	Exemples de valeurs littérales
AAAAMMJJ	<code>`20141231`</code>
AAMMJJ	<code>`141231`</code>
AAAAMMJJ hhmmss	<code>`20141231 235959`</code>
AAMMJJthhmm	<code>`141231t2359`</code>

Exemples de formats	Exemples de valeurs littérales
AAAAMMJJThh	`20141231T23`
AAAAMMJJ hhmmss+/-hhmm (décalage UTC)	`20141231 235959-0500`
AAMMJJ hhmm+/-hh (décalage UTC)	`141231 2359+01`
<p><b>Remarque</b></p> <p>N'utilisez pas hh seul dans le format d'heure principal avec des données qui ont un décalage UTC. Par exemple, évitez : hh+hhmm. Il se peut que les résultats ne soient pas fiables.</p>	

## Fonctions associées

Si vous devez renvoyer :

- le nom du jour de la semaine, utilisez CDOW( ) au lieu de DOW( )
- le jour du mois sous forme de chiffre (de 1 à 31), utilisez DAY( ) au lieu de DOW( )

# Fonction DTOU( )

Convertit une valeur date Analytics en chaîne de caractères Unicode dans la langue et le format régional spécifiés. Abréviation de "Date en Unicode".

## Remarque

Cette fonction est spécifique à l'édition Unicode d'Analytics. Elle n'est pas prise en charge dans l'édition non-Unicode.

## Syntaxe

```
DTOU(<date> <;paramètre> <;style>)
```

## Paramètres

Nom	Type	Description
<i>date</i> optionnel	DateHeure	<p>Le champ, l'expression, ou la valeur littérale à convertir en chaîne de caractères Unicode. Si ce paramètre est ignoré, la date actuelle du système d'exploitation est utilisée.</p> <p>La <i>date</i> peut contenir une valeur de type DateHeure, mais la portion de la valeur correspondant à l'heure sera ignorée. Les valeurs de temps seules ne sont pas prises en charge.</p> <p>Vous pouvez indiquer une valeur de champ ou de date littérale :</p> <ul style="list-style-type: none"> <li>◦ <b>Champ</b> : peut utiliser n'importe quel format de date, tant que la définition de champ définit correctement le format</li> <li>◦ <b>Littéral</b> : doit utiliser un des formats AAAAMMJJ ou AAMMJJ, par exemple `20141231`</li> </ul> <p>La valeur <i>date</i> minimale prise en charge est le 31 décembre 1969.</p>
<i>paramètre</i> optionnel	caractère	<p>Code des paramètres régionaux indiquant la langue de la chaîne de sortie, et, de façon facultative, la version de la langue associée à un pays ou territoire particulier.</p> <p>Par exemple, "zh" indique chinois, et "pt_BR" indique portugais brésilien.</p> <p>Si vous ignorez ce paramètre, c'est la langue locale par défaut de votre ordinateur qui est utilisée. Si une langue est spécifiée mais qu'aucun pays n'est spécifié, le pays par défaut de la langue concernée est utilisé.</p> <p>Vous ne pouvez pas indiquer de <i>paramètres régionaux</i> si vous n'avez pas indiqué de <i>date</i>.</p> <p>Pour obtenir des informations sur les codes locaux, consultez le site</p>

Nom	Type	Description
		<a href="http://www.unicode.org">www.unicode.org</a> .
<i>style</i> optionnel	numérique	<p>Style de format de date à utiliser pour la chaîne de caractères Unicode. Le style de format correspond à la norme relative aux paramètres régionaux spécifiés :</p> <ul style="list-style-type: none"> <li>○ 0 : format complet, comme « Dimanche 18 septembre 2016 »</li> <li>○ 1 : format long, comme « 18 septembre 2016 »</li> <li>○ 2 : format moyen, comme « 18 sept. 2016 »</li> <li>○ 3 : format numérique court, comme "18/09/16"</li> </ul> <p>Si vous ignorez ce paramètre, c'est la valeur par défaut de 2 qui est utilisée. Vous ne pouvez pas indiquer de <i>style</i> si vous n'avez pas indiqué de <i>date</i> et les <i>paramètres régionaux</i>.</p>

## Sortie

Caractère.

## Exemples

### Exemples basiques

#### Valeurs d'introduction littérales

Renvoie « 31 de dezembro de 2014 » :

```
DTOU(`20141231`; "pt_BR"; 1)
```

Renvoie « 31 grudnia 2014 » :

```
DTOU(`20141231`; "pl"; 1)
```

#### Valeurs d'introduction champ

Renvoie chaque date numérique dans le champ **Date\_facture** sous forme d'une chaîne de caractères Unicode :

```
DTOU(Date_facture; "zh"; 1)
```

#### La sortie utilise un style de date complet

Renvoie « 星期三, 2014 十二月 31 » (sans identifiant de région spécifié) :

```
DTOU(`20141231`; "zh"; 0)
```

Renvoie « 2014年12月31日星期三 » (identifiant de région spécifié) :

```
DTOU(`20141231`; "zh_CN"; 0)
```

## La sortie utilise un style de date long

Renvoie « 2014 十二月 31 » (sans identifiant de région spécifié) :

```
DTOU(`20141231`; "zh"; 1)
```

Renvoie « 2014年12月31日 » (identifiant de région spécifié) :

```
DTOU(`20141231`; "zh_CN"; 1)
```

# Remarques

## Fonctions associées

DTOU ( ) est l'inverse de la fonction UTOD( ), qui convertit une chaîne Unicode en date.

# Fonction EBCDIC()

Renvoie une chaîne convertie en codage de caractère EBCDIC.

## Syntaxe

```
EBCDIC(chaîne_de_caractères)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	Valeur à convertir en EBCDIC.

## Sortie

Caractère.

## Exemples

### Exemples basiques

Renvoie « ñòó@Æ '...@â£K » :

```
EBCDIC("123 Fake St.")
```

### Exemples avancés

#### Création d'un champ encodé EBCDIC à exporter

Pour créer un champ contenant la valeur encodée EBCDIC d'un champ **Nom** en vue de l'exporter vers une application qui requiert le codage EBCDIC, vous devez indiquer :

```
DEFINE FIELD Nom_Exp COMPUTED EBCDIC(Nom)
```

# Remarques

## Quand utiliser la fonction EBCDIC( )

Utilisez cette fonction pour convertir des données en codage de caractère EBCDIC (Extended Binary Coded Decimal Interchange Code). Le codage de caractère EBCDIC est utilisé essentiellement sur les systèmes d'exploitation centraux IBM, tels que z/OS.



# Fonction EFFECTIVE( )

Renvoie le taux d'intérêt annuel réel d'un emprunt.

## Syntaxe

```
EFFECTIVE(taux_nominal; périodes)
```

## Paramètres

Nom	Type	Description
<i>taux_nominal</i>	numérique	Taux d'intérêt nominal annuel.
<i>périodes</i>	numérique	Nombre de périodes composées par an. <b>Remarque</b> Indiquez un entier. Si vous indiquez une partie décimale, elle est tronquée.

## Sortie

Numérique. Le taux est calculé jusqu'à la huitième décimale.

## Exemples

### Exemples basiques

Renvoie 0,19561817 (19,56 %), soit le taux d'intérêt annuel effectif sur le solde non payé d'une carte de crédit dont les frais sont des intérêts composés mensuels de 18 % par an :

```
EFFECTIVE(0,18; 12)
```

# Remarques

## Qu'est-ce que le taux d'intérêt annuel effectif ?

Le taux d'intérêt annuel effectif d'un emprunt est le taux annuel réel des intérêts payés, en prenant en compte les intérêts composés mensuels ou journaliers sur le solde restant.

## Fonctions associées

La fonction `NOMINAL()` est l'inverse de la fonction `EFFECTIVE()`.

# Fonction EOMONTH( )

Renvoie la date du dernier jour du mois qui correspond au nombre de mois indiqué avant ou après une date indiquée.

## Syntaxe

```
EOMONTH(<date/DateHeure> <;mois>)
```

## Paramètres

Nom	Type	Description
<i>date/DateHeure</i> optionnel	DateHeure	Le champ, l'expression, ou la valeur littérale à partir desquels calculer la date de fin de mois. Si ce paramètre est ignoré, la date de fin de mois est calculée à partir de la date actuelle du système d'exploitation.  <b>Remarque</b> Vous pouvez indiquer une valeur de type DateHeure pour <i>date/DateHeure</i> , mais la portion de la valeur correspondant à l'heure sera ignorée.
<i>mois</i> optionnel	numérique	Le nombre de mois avant ou après <i>date/DateHeure</i> . Si ce paramètre est ignoré, la valeur par défaut 0 (zéro) est utilisée.  Vous ne pouvez pas indiquer des <i>mois</i> si vous avez ignoré <i>date/DateHeure</i> .

## Sortie

DateHeure. La valeur de date est générée à l'aide du format d'affichage de date Analytics actuel.

## Exemples

### Exemples basiques

#### Aucune entrée

Renvoie le dernier jour du mois correspondant à la date du système d'exploitation actuel :

```
EOMONTH()
```

## Valeurs d'introduction littérales

Renvoie `20140131` affiché au format 31 jan 2014 en supposant que le format d'affichage actuel de la date dans Analytics soit JJ MMM AAAA :

```
EOMONTH(`20140115`)
```

Renvoie `20140430` affiché au format 30 avr 2014 en supposant que le format d'affichage actuel de la date dans Analytics soit JJ MMM AAAA :

```
EOMONTH(`20140115`; 3)
```

Renvoie `20131031` affiché au format 31 oct 2013 en supposant que le format d'affichage actuel de la date dans Analytics soit JJ MMM AAAA :

```
EOMONTH(`20140115`; -3)
```

## Valeurs d'introduction champ

Renvoie le dernier jour du mois qui tombe trois mois après chaque date dans le champ **Date\_facture** :

```
EOMONTH(Date_facture; 3)
```

Renvoie le dernier jour du mois qui tombe trois mois après chaque date dans le champ **Date\_facture** plus une période de grâce de 15 jours :

```
EOMONTH(Date_facture + 15; 3)
```

Renvoie le premier jour du mois correspondant à la date de la facture :

```
EOMONTH(Date_facture; -1) + 1
```

# Remarques

## Formats DateHeure

Un champ indiqué pour *date/DateHeure* peut utiliser n'importe quel format de date ou de DateHeure, tant que la définition de champ définit correctement le format.

Une valeur de date littérale doit utiliser l'un des formats suivants :

- AAAAMMJJ
- AAMMJJ

Vous devez saisir les valeurs de dates littérales entre guillemets inversés. Par exemple : `20141231`

## Fonctionnement de la valeur *mois*

- **Valeur positive** : la date générée est plus récente que la *date/DateHeure* indiquée
- **Valeur négative** : la date générée est antérieure à la *date/DateHeure* indiquée.
- **Valeur omise ou '0' (zéro)** : la date générée est le dernier jour du mois en cours de la *date/DateHeure*

## Renvoi de la date du premier jour d'un mois

Ajoutez 1 jour au résultat de la fonction EOMONTH( ) pour renvoyer la date du premier jour d'un mois.

Renvoie `20140201` affiché au format 01 fév 2014 en supposant que le format d'affichage actuel de la date dans Analytics soit JJ MMM AAAA :

```
EOMONTH(`20140115`) + 1
```

## Fonctions associées

Utilisez la fonction GOMONTH( ) si vous souhaitez renvoyer la date du dernier jour du mois, plutôt que la date exacte, qui correspond au nombre de mois indiqué avant ou après une date indiquée.

# Fonction EXCLUDE()

Renvoie une chaîne qui exclut les caractères spécifiés.

## Syntaxe

```
EXCLUDE(chaîne_de_caractères; caractères_à_exclure)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	Valeur dans laquelle exclure les caractères.
<i>caractères_à_exclure</i>	caractère	Liste des caractères à exclure. Si vous indiquez des guillemets doubles dans les <i>caractères_à_exclure</i> , vous devez placer la liste de caractères à l'intérieur de guillemets simples. Par exemple : "'-/'

## Sortie

Caractère.

## Exemples

### Exemples basiques

Renvoie " Alberni Street", ce qui correspond à la chaîne d'entrée à l'exclusion des chiffres :

```
EXCLUDE("1550 Alberni Street"; "0123456789")
```

Renvoie toutes les valeurs du champ **Numéro\_produit** à l'exclusion de la barre oblique et du signe dièse :

```
EXCLUDE(Numéro_produit; "/#")
```

# Remarques

## Fonctionnement

La fonction EXCLUDE( ) compare chaque caractère de la *chaîne\_de\_caractères* avec les caractères de la liste *caractères\_à\_exclure*. En cas de correspondance, le caractère est exclu de la chaîne de sortie.

Par exemple, la sortie pour EXCLUDE("123-45-4536"; "-") est « 123454536 ».

## Aucun caractère correspondant

S'il n'existe aucune correspondance entre *chaîne\_de\_caractères* et *caractères\_à\_exclure*, la *chaîne\_de\_caractères* et la sortie de la fonction sont identiques.

Par exemple, la sortie pour EXCLUDE("ABC"; "D") est "ABC".

## Respect de la casse

La fonction EXCLUDE( ) est sensible à la casse. Si vous indiquez « ID » dans *caractères\_à\_exclure*, ces caractères ne seront pas exclus de "id#94022". Si vous pensez que la casse est incohérente dans la *chaîne*, utilisez la fonction UPPER( ) pour convertir tous les caractères en majuscules.

Par exemple :

```
EXCLUDE(UPPER("id#94022"); "ID#")
```

## Conseil d'utilisation

Utilisez EXCLUDE( ) si le jeu de caractères à exclure est petit et que le jeu à inclure est grand.

## Exclure les guillemets doubles et les guillemets simples

Les guillemets servent de délimiteurs de chaîne. Par conséquent, pour exclure les guillemets doubles et les guillemets simples en même temps, vous devez imbriquer la fonction EXCLUDE( ) pour qu'il y a une seule fonction pour chaque type de guillemet :

```
EXCLUDE(EXCLUDE(champ_à_traiter; """); """)
```

## Fonctions associées

La fonction EXCLUDE( ) est l'opposé de la fonction INCLUDE( ).

# Fonction EXP()

Renvoie la valeur exponentielle (base 10) d'une expression numérique avec un nombre de décimales indiqué.

## Syntaxe

```
EXP(nombre; décimales)
```

## Paramètres

Nom	Type	Description
<i>nombre</i>	numérique	Champ numérique, expression ou valeur dont il faut renvoyer la valeur exponentielle.
<i>décimales</i>	numérique	Nombre de décimales à inclure dans la valeur renvoyée.

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 1000,00 :

```
EXP(3; 2)
```

Renvoie 72443,596007 :

```
EXP(4,86; 6)
```



## Exemples avancés

### Rechercher la racine cubique

Crée un champ qui est la racine cubique du champ X à deux décimales :

```
DEFINE FIELD racine_cubique COMPUTED EXP(LOG(X; 6) / 3; 2)
```

#### Astuce

Pour déterminer la *N*ième racine, divisez la trace de la valeur par *n* et relevez l'exponentiel du résultat.

## Remarques

### Fonctionnement

Cette fonction renvoie la valeur exponentielle (base 10) d'une expression numérique, définie comme 10 élevé à la puissance *n*. Par exemple, la valeur exponentielle de 3 est  $10^3$ , soit 1 000.

### Quand utiliser la fonction EXP( )

Utilisez la fonction EXP( ) pour les applications financières nécessitant des calculs mathématiques complexes. La fonction EXP( ) exécute la même opération que l'opérateur d'élevation à une puissance (^). Elle peut également être utile aux applications utilisant la fonction LOG( ).

### Fonctions associées

L'inverse d'un exposant est son logarithme. Ainsi, EXP( ) est l'opposé de la fonction LOG( ).

# Fonction FILESIZE()

Renvoie la taille du fichier spécifié en octets ou -1 si le fichier n'existe pas.

## Syntaxe

```
FILESIZE(nom_de_fichier)
```

## Paramètres

Nom	Type	Description
<i>nom_fichier</i>	caractère	<p>Le nom du fichier.</p> <p>Si le fichier se trouve dans le même dossier que le projet Analytics, vous n'avez pas besoin d'indiquer le chemin du fichier.</p> <p>Pour les fichiers dans d'autres dossiers, indiquez un chemin d'accès absolu ou relatif. Par exemple :</p> <ul style="list-style-type: none"><li>"résultats\sortie_test.fil"</li><li>"c:\résultats\sortie_test.fil"</li></ul> <p><b>Remarque</b></p> <p>Vous devez indiquer le nom de fichier de données physique (.fil) pour les tables Analytics et non le nom de la table.</p>

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 14 744 :

```
FILESIZE("Inventaire.fil")
```

Si le fichier que vous vérifiez n'est pas dans le même dossier que le projet Analytics, vous devez indiquer le chemin d'accès relatif ou le chemin d'accès absolu au fichier.

Renvoie 6018 :

```
FILESIZE("C:\ACL Data\Exemples de fichiers de données\Sauvegarder\Ap_Trans.fil")
```

## Exemples avancés

### Exécution d'un script si un fichier n'existe pas

Exécute uniquement le script `import_données` si le fichier `Inventaire_Metaphor_2002.fil` n'existe pas :

```
DO SCRIPT import_données IF FILESIZE("Inventaire_Metaphor_2002.fil") = -1
```

### Enregistrement de la taille d'un fichier dans la trace des commandes Analytics

Utilisez la commande `CALCULATE` pour enregistrer la taille de `Inventaire_Metaphor_2002.fil` dans la trace des commandes Analytics :

```
CALCULATE FILESIZE("Inventaire_Metaphor_2002.fil")
```

# Fonction FIND( )

Renvoie une valeur logique indiquant si la chaîne indiquée est présente dans un champ particulier ou n'importe où dans un enregistrement entier.

## Remarque

La fonction FIND( ) et la "Commande FIND" Page 214 sont deux fonctionnalités Analytics distinctes présentant des différences significatives.

## Syntaxe

```
FIND(chaîne_de_caractères<;champ_à_explorer>)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	Chaîne de caractères à rechercher. Cette recherche n'est pas sensible à la casse.
<i>champ_à_explorer</i> optionnel	caractère	Le champ ou la variable à explorer. Si ce paramètre est ignoré, l'enregistrement entier est exploré, y compris les parties non définies de l'enregistrement.

## Sortie

Logique. Renvoie T (« true », vrai) si la valeur *chaîne\_de\_caractères* indiquée est trouvée ; renvoie F (« false », faux) dans les autres cas.

## Exemples

### Exemples basiques

#### Recherche dans un enregistrement entier

Renvoie « T » pour tous les enregistrements contenant la chaîne « New York » dans n'importe quel champ, dans toutes les limites de champs et dans n'importe quelle partie non définie de l'enregistrement. Renvoie F autrement :

```
FIND("New York")
```

## Rechercher dans un champ unique

Renvoie « T » pour tous les enregistrements contenant la chaîne « New York » dans le champ **Ville**. Renvoie F autrement.

```
FIND("New York"; Ville)
```

Renvoie T pour tous les enregistrements contenant la chaîne « Ne » dans le champ **Ville**. Renvoie F autrement :

```
FIND("Ne"; Ville)
```

Renvoie T pour tous les enregistrements contenant la chaîne « New York » précédée d'une ou plusieurs espaces dans le champ **Ville**. Renvoie F autrement :

```
FIND(" New York"; Ville)
```

Renvoie T pour l'ensemble des enregistrements présentant une valeur dans le champ **Description** correspondant ou contenant la valeur de la variable `v_terme_recherché`. Renvoie F autrement :

```
FIND(v_terme_recherché; Description)
```

## Rechercher dans plusieurs champs

Renvoie T pour tous les enregistrements contenant la chaîne « New York » dans le champ **Ville** ou dans le champ **Ville\_2**. Renvoie F autrement :

```
FIND("New York"; Ville+Ville_2)
```

Renvoie T pour tous les enregistrements contenant la chaîne « New York » dans le champ **Ville** ou dans le champ **Ville\_2**. Renvoie F autrement :

```
FIND("New York"; Ville) OR FIND("New York"; Ville_2)
```

## Combinaison avec d'autres fonctions

Renvoie T pour l'ensemble des enregistrements présentant une valeur dans le champ **Nom\_2** correspondant ou contenant la valeur tronquée du champ **Nom**. Renvoie F autrement :

```
FIND(ALLTRIM(Nom); Nom_2)
```

# Remarques

## Quand utiliser FIND( )

Utilisez la fonction FIND( ) pour vérifier la présence de la *chaîne\_de\_caractères* indiquée dans un ou plusieurs champs ou bien dans l'intégralité d'un enregistrement.

## Fonctionnement des correspondances

La valeur *chaîne\_de\_caractères* peut correspondre exactement ou bien figurer dans une chaîne plus longue. Les espaces de début dans les champs n'affectent pas la recherche sauf si vous incluez un ou plusieurs espaces de début dans la valeur *chaîne\_de\_caractères*.

## Recherche dans un enregistrement entier

Si le paramètre facultatif *champ\_à\_explorer* n'est pas indiqué, l'enregistrement entier est exploré, y compris les parties non définies de l'enregistrement. Les limites de champ sont ignorées lorsque l'intégralité de l'enregistrement est explorée et lorsque les espaces de fin dans les champs sont traités comme des caractères.

### Remarque

Lorsque vous recherchez dans un enregistrement entier, c'est dans l'enregistrement physique que se passe la recherche. La recherche n'est pas effectuée dans les champs calculés et les champs associés.

## Recherche dans un sous-ensemble de champs

Vous pouvez concaténer deux champs ou plus dans le paramètre *champ\_à\_explorer* si vous souhaitez que votre recherche porte sur un sous-ensemble de champs d'une table. Par exemple, pour rechercher la chaîne « New York » à la fois dans les champs **Ville** et **Ville\_2** :

```
FIND("New York"; Ville+Ville_2)
```

Les champs concaténés sont traités comme un seul champ incluant les espaces de début et de fin des champs individuels sauf si vous utilisez la fonction ALLTRIM( ) pour supprimer les espaces.

Vous pouvez aussi créer une expression qui explore chaque champ individuellement :

```
FIND("New York"; Ville) OR FIND("New York"; Ville_2)
```

Si *chaîne\_de\_caractères* inclut une espace de début, les résultats de recherche des deux méthodes peuvent être différents.

## Sensibilité de la casse et comparaison exacte des caractères

La fonction FIND( ) n'est pas sensible à la casse et permet de rechercher à la fois les caractères ASCII et EBCDIC. La fonction n'est pas affectée par l'option **Comparaisons exactes de caractères** (SET EXACT ON/OFF).

### Recherche dans un champ calculé

Pour effectuer une recherche dans un champ calculé, vous devez indiquer le nom du champ dans *champ\_à\_explorer*. Par exemple, si **Ville\_fournisseur** est un champ calculé qui isole la ville dans une adresse :

```
FIND("New York"; Ville_fournisseur)
```

### Recherche dans un champ associé

Pour effectuer une recherche dans un champ associé, vous devez indiquer le nom complet du champ (c'est-à-dire *table.nom du champ*) dans la valeur *champ\_à\_explorer* :

```
FIND("New York"; Fournisseur.Ville_Fournisseur)
```

## Recherche parmi des données DateHeure ou numériques

Il est possible d'utiliser la fonction FIND( ) pour effectuer des recherches parmi des données DateHeure ou numériques au niveau des enregistrements. La spécification du *champ\_à\_explorer* n'est pas prise en charge pour la recherche parmi des données DateHeure ou numériques.

La *chaîne* numérique ou DateHeure doit être entre guillemets et doit correspondre exactement à la mise en forme des données sources plutôt qu'à la mise en forme de la vue.

L'utilisation de la fonction FIND( ) pour effectuer une recherche parmi des données DateHeure ou numériques dans des champs calculés ou associés n'est pas prise en charge.

#### Remarque

Il n'est pas recommandé d'utiliser la fonction FIND( ) pour effectuer une recherche parmi des données DateHeure ou numériques, car sa manipulation peut être difficile.

# Fonction FINDMULTI( )

Renvoie une valeur logique indiquant si une chaîne de caractères dans un ensemble d'une ou plusieurs chaînes de caractères spécifiée est présente dans un champ particulier ou quelque part dans un enregistrement entier.

## Syntaxe

```
FINDMULTI({explorer|RECORD}; chaîne_1 <;...n>)
```

## Paramètres

Nom	Type	Description
<i>explorer</i>   RECORD	caractère	<p>Le champ ou la variable à explorer.</p> <p>Indiquez le mot-clé RECORD pour explorer l'enregistrement entier, y compris les parties non définies de l'enregistrement.</p> <p>Vous pouvez aussi indiquer une liste de mots-clés en concaténant des noms de champs :</p> <pre>Champ_1+Champ_2+Champ_3</pre>
<i>chaîne_1 &lt;;...n&gt;</i>	caractère	<p>Une ou plusieurs chaînes de caractères à rechercher. Séparez plusieurs chaînes de recherche par des virgules :</p> <pre>FINDMULTI(RECORD; "Joa"; "Jim"; "Joh")</pre> <p>La recherche n'est pas sensible à la casse.</p>

## Sortie

Logique. Renvoie T (« true », vrai) si une ou plusieurs des valeurs *chaîne\_de\_caractères* indiquées sont trouvées ; renvoie F (« false », faux) dans les autres cas.



# Exemples

## Exemples basiques

### Recherche dans un enregistrement entier

Renvoie T pour tous les enregistrements contenant « New York » ou « Chicago » dans n'importe quel champ, dans toutes les limites de champs et dans n'importe quelle partie non définie de l'enregistrement. Renvoie F autrement :

```
FINDMULTI(RECORD; "New York"; "Chicago")
```

### Rechercher dans un champ unique

Renvoie T pour tous les enregistrements contenant « New York » ou « Chicago » dans le champ **Ville**. Renvoie F autrement :

```
FINDMULTI(Ville; "New York"; "Chicago")
```

Renvoie T pour tous les enregistrements contenant la chaîne « Ne » ou « Chi » dans le champ **Ville**. Renvoie F autrement :

```
FINDMULTI(Ville; "Ne"; "Chi")
```

Renvoie T pour tous les enregistrements contenant « New York » ou « Chicago » précédée d'une ou plusieurs espaces dans le champ **Ville**. Renvoie F autrement :

```
FINDMULTI(Ville; " New York"; " Chicago")
```

Renvoie T pour l'ensemble des enregistrements présentant une valeur dans le champ **Description** correspondant ou contenant une valeur des variables *v\_terme\_recherché*. Renvoie F dans les autres cas :

```
FINDMULTI(Description; v_terme_recherché_1; v_terme_recherché_2; v_terme_recherché_3)
```

### Rechercher dans plusieurs champs

Renvoie T pour tous les enregistrements contenant la chaîne « New York » ou « Chicago » dans le champ **Ville** ou dans le champ **Ville\_2**. Renvoie F autrement :

```
FINDMULTI(Ville+Ville_2; "New York"; "Chicago")
```

Renvoie T pour tous les enregistrements contenant la chaîne « New York » ou « Chicago » dans le champ **Ville** ou dans le champ **Ville\_2**. Renvoie F autrement :

```
FINDMULTI(Ville; "New York"; "Chicago") OR FINDMULTI(Ville_2; "New York"; "Chicago")
```

## Combinaison avec d'autres fonctions

Renvoie T pour l'ensemble des enregistrements présentant une valeur dans le champ **Nom\_1** correspondant ou contenant la valeur tronquée du champ **Nom\_2** ou **Nom\_3**. Renvoie F autrement :

```
FINDMULTI(Nom_1; ALLTRIM(Nom_2); ALLTRIM(Nom_3))
```

# Remarques

## Quand utiliser FINDMULTI( )

Utilisez la fonction FINDMULTI( ) pour vérifier la présence d'une ou plusieurs chaînes de caractères indiquées dans un ou plusieurs champs ou bien dans l'intégralité d'un enregistrement.

## Fonctionnement des correspondances

La valeur *chaîne\_de\_caractères* peut correspondre exactement ou bien figurer dans une chaîne plus longue. Les espaces de début dans les champs n'affectent pas la recherche sauf si vous incluez un ou plusieurs espaces de début dans la valeur *chaîne\_de\_caractères*.

## Recherche dans un enregistrement entier

Si vous indiquez RECORD plutôt qu'un champ *explorer*, l'enregistrement entier est exploré, y compris les parties non définies de l'enregistrement. Les limites de champ sont ignorées lorsque l'intégralité de l'enregistrement est explorée et lorsque les espaces de fin dans les champs sont traités comme des caractères.

### Remarque

Lorsque vous recherchez dans un enregistrement entier, c'est dans l'enregistrement physique que se passe la recherche. La recherche n'est pas effectuée dans les champs calculés et les champs associés.

## Recherche dans un sous-ensemble de champs

Vous pouvez concaténer deux champs ou plus dans le paramètre *explorer* si vous souhaitez que votre recherche porte sur un sous-ensemble de champs d'une table. Par exemple, pour rechercher les chaînes « New York » ou « Chicago » à la fois dans les champs **Ville** et **Ville\_2** :

```
FINDMULTI(Ville+Ville_2; "New York"; "Chicago")
```

Les champs concaténés sont traités comme un seul champ incluant les espaces de début et de fin des champs individuels sauf si vous utilisez la fonction ALLTRIM( ) pour supprimer les espaces.

Vous pouvez aussi créer une expression qui explore chaque champ individuellement :

```
FINDMULTI(Ville; "New York"; "Chicago") OR FINDMULTI(Ville_2; "New York"; "Chicago")
```

Si une valeur *chaîne\_de\_caractères* inclut une espace de début, les résultats de recherche des deux méthodes peuvent être différents.

## Sensibilité de la casse et comparaison exacte des caractères

La fonction FINDMULTI( ) n'est pas sensible à la casse et permet de rechercher à la fois les caractères ASCII et EBCDIC. La fonction n'est pas affectée par l'option **Comparaisons exactes de caractères** (SET EXACT ON/OFF).

## Recherche dans un champ calculé

Pour effectuer une recherche dans un champ calculé, vous devez indiquer le nom du champ dans *explorer*. Par exemple, si **Ville\_fournisseur** est un champ calculé qui isole la ville dans une adresse :

```
FINDMULTI(Ville_Fournisseur; "New York"; "Chicago")
```

## Recherche dans un champ associé

Pour effectuer une recherche dans un champ associé, vous devez indiquer le nom complet du champ (c'est-à-dire *table.nom du champ*) dans la valeur *explorer* :

```
FINDMULTI(Fournisseur.Ville_Fournisseur; "New York"; "Chicago")
```

## Recherche parmi des données DateHeure ou numériques

Il est possible d'utiliser la fonction FINDMULTI( ) pour effectuer des recherches parmi des données DateHeure ou numériques au niveau des enregistrements, lorsque vous indiquez RECORD. La spécification d'un champ *explorer* n'est pas prise en charge pour la recherche parmi des données DateHeure ou numériques.

Les valeurs de la *chaîne* numérique ou DateHeure doivent être entre guillemets et doivent correspondre exactement à la mise en forme des données sources plutôt qu'à la mise en forme de la vue.

L'utilisation de la fonction FINDMULTI( ) pour effectuer une recherche parmi des données DateHeure ou numériques dans des champs calculés ou associés n'est pas prise en charge.

**Remarque**

Il n'est pas recommandé d'utiliser la fonction FINDMULTI( ) pour effectuer une recherche parmi des données DateHeure ou numériques, car sa manipulation peut être difficile.

# Fonction FREQUENCY()

Renvoie la fréquence Benford attendue pour des chiffres numériques positifs et séquentiels de tête avec une précision de huit décimales.

## Syntaxe

```
FREQUENCY(nombre)
```

## Paramètres

Nom	Type	Description
<i>nombre</i>	caractère	Chaîne de caractères contenant des chiffres (entre 0 et 9) dont il faut identifier la fréquence. <i>nombre</i> doit être un nombre positif. Notez que les zéros de tête sont ignorés.

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 0,00998422 :

```
FREQUENCY("43")
```

Renvoie 0,00000000 :

```
FREQUENCY("87654321")
```

#### Remarque

Le résultat est 0,00000000495. Cependant, étant donné qu'Analytics calcule avec une précision de huit décimales, une valeur zéro est renvoyée.

# Remarques

## Fonctionnement

FREQUENCY() renvoie la fréquence Benford attendue de chiffres numériques positifs et séquentiels de tête avec une précision de huit chiffres. Elle permet d'exécuter des tests Benford limités pour des situations spécifiques.

## Utilisation de cette fonction pour ces combinaisons de chiffres spécifiques

Vous pouvez utiliser cette fonction à la place de la commande BENFORD si vous souhaitez vous concentrer sur des combinaisons de chiffres spécifiques. Par exemple, lors de l'audit de réclamations d'assurance ayant des limites d'agrément avec des valeurs de demande spécifiées, vous pouvez utiliser la fonction FREQUENCY() pour la recherche de valeurs immédiatement inférieures à la limite agréée.

Pour trouver les réclamations dont le montant approche la limite d'agrément de 5 000 €, vous pouvez sélectionner la plage 4 900 € à 4 999 €. Tout d'abord, comptez le nombre total d'enregistrements, puis utilisez un filtre pour compter les enregistrements pour lesquels LEADING() renvoie 49 et comparez le quotient des deux comptes avec la valeur obtenue pour FREQUENCY("49").

Ceci va plus vite que l'exécution d'une analyse complète d'une table comportant un million d'enregistrements et ne génère pas une table volumineuse ou de longues entrées dans la trace des commandes.

## Spécification de chaînes excédant six chiffres

La spécification de chaînes excédant six chiffres peut engendrer des valeurs zéro. Les calculs de chaînes supérieures à six chiffres peuvent nécessiter une précision dépassant la limite d'Analytics, à savoir huit décimales.

# Fonction FTYPE( )

Renvoie un caractère identifiant la catégorie de données d'un champ ou d'une variable, ou le type d'un élément de projet Analytics.

## Syntaxe

```
FTYPE(nom_champ_chaine_de_caractères)
```

## Paramètres

Nom	Type	Description
<i>nom_champ_chaine_de_caractères</i>	caractère	Nom de champ, nom de variable ou nom d'élément de projet Analytics. Entourez <i>nom_champ_chaine_de_caractères</i> de guillemets : <pre>FTYPE("Montant")</pre>

## Sortie

Caractère. Cette fonction renvoie un des caractères suivants. Ce dernier indique le champ, la variable ou le type d'élément de projet Analytics :

- "C" : Champ caractère
- "N" : Champ numérique
- "D" : Champ DateHeure
- "L" : Champ logique
- "c" : Variable de caractère
- "n" : Variable numérique
- "d" : Variable DateHeure
- "l" : Variable logique
- "b" : Script Analytics
- "y" : Format de table Analytics
- "w" : Environnement de travail Analytics
- "i" : Index Analytics
- "r" : Rapport Analytics
- "a" : Fichier journal Analytics
- "U" : Non défini

# Exemples

## Exemples basiques

Dans l'exemple suivant, une valeur de 4 est attribuée à la variable *num* et le type est ensuite vérifié.

Renvoie « n » :

```
ASSIGN num = 4
FTYPE("num")
```

## Exemples avancés

### Test du type de données d'un champ

Vous travaillez avec un script ou un outil d'analyse requérant un champ numérique **Montant** et vous devez vérifier que le type du champ est correct avant d'exécuter le script.

La commande suivante exécute uniquement Script\_1 si **Montant** est un champ numérique :

```
OPEN Factures
DO Script_1 IF FTYPE("Montant") = "N"
```

### Test de l'existence d'une table ou d'un élément de projet Analytics

La commande suivante exécute uniquement Script\_1 si une table nommée Factures se trouve dans le projet :

```
DO Script_1 IF FTYPE("Factures") <> "U"
```

### Test de l'environnement d'exécution

Vous pouvez utiliser la fonction FTYPE pour déterminer si un outil d'analyse s'exécute dans Analytics, sur Analytics Exchange ou dans la fenêtre App Analyse.

Si un outil d'analyse s'exécute sur Analytics Exchange ou dans la fenêtre App Analyse, 'ax\_main' est égal à 'b' :

```
IF FTYPE('ax_main') = 'b' v_running_in_ax_or_analysis_app = T
```

Si un outil d'analyse s'exécute dans Analytics, 'ax\_main' n'est pas égal à 'b' :

```
IF FTYPE('ax_main') <> 'b' v_running_in_ax_or_analysis_app = F
```



La possibilité de détecter l'environnement d'exécution vous permet de concevoir un seul script exécutant différents blocs de codes en fonction de l'application il s'exécute.

# Fonction FVANNUITY()

Renvoie la valeur capitalisée d'une série de paiements calculés à l'aide d'un taux d'intérêt constant. La valeur capitalisée correspond à la somme des paiements augmentée des intérêts composés cumulés.

## Syntaxe

```
FVANNUITY(taux; périodes; règlement <;type>)
```

## Paramètres

Nom	Type	Description
<i>taux</i>	numérique	Taux d'intérêt par période.
<i>périodes</i>	numérique	Nombre total de périodes de paiement.
<i>règlement</i>	numérique	Paiement par période. Le montant du paiement doit demeurer constant pendant toute la durée de l'annuité.
<i>type</i> optionnel	numérique	Date des paiements : <ul style="list-style-type: none"> <li>○ 0 : paiement en fin de période</li> <li>○ 1 : paiement en début de période</li> </ul> Si vous ignorez ce paramètre, c'est la valeur par défaut de 0 qui est utilisée.

### Remarque

Vous devez utiliser des périodes temporelles cohérentes lorsque vous spécifiez *taux*, *périodes* et *paiement* pour vous assurer que vous spécifiez un taux d'intérêt **par période**.

Par exemple :

- dans le cas d'un *paiement* mensuel, sur un emprunt ou un investissement d'une durée de deux ans avec un taux d'intérêt de 5 % par an, spécifiez 0,05/12 pour *taux* et 2 \* 12 pour *périodes*
- dans le cas d'un *paiement* annuel sur le même prêt ou investissement, indiquez 0,05 pour *taux* et 2 pour *périodes*

## Sortie

Numérique. Le résultat est calculé avec deux décimales.

# Exemples

## Exemples basiques

### Paiements mensuels

Renvoie 27243,20, la valeur capitalisée de 1 000 \$ payée au début de chaque mois pendant 2 ans au taux composé mensuel de 1 % par mois :

```
FVANNUIITY(0,01; 2*12; 1000; 1)
```

Renvoie 12809,33, la valeur capitalisée de la même annuité au bout de la première année :

```
FVANNUIITY(0,01; 12; 1000; 1)
```

### Paiements annuels

Renvoie 25440,00, la valeur capitalisée de 12 000 \$ payée à la fin de chaque année pendant 2 ans à un taux composé annuel de 12 % par an :

```
FVANNUIITY(0,12; 2; 12000; 0)
```

## Exemples avancés

### Calculs d'annuité

Les calculs d'annuité impliquent quatre variables :

- **valeur actualisée ou valeur capitalisée** : \$21243,39 et \$ 26973,46 dans les exemples ci-après
- **montant de paiement par période** : \$1000,00 dans les exemples ci-après
- **taux d'intérêt par période** : 1 % par mois dans les exemples ci-après
- **nombre de périodes** : 24 mois dans les exemples ci-après

Si vous connaissez la valeur de trois des variables, vous pouvez utiliser une fonction Analytics pour calculer la quatrième.

Je souhaite trouver :	Fonction Analytics à utiliser :
Valeur actualisée	PVANNUIITY() Renvoie 21243,39 : <pre>PVANNUIITY(0,01; 24; 1000)</pre>

Je souhaite trouver :	Fonction Analytics à utiliser :
Valeur capitalisée	FVANNUIITY( ) Renvoie 26973,46 : <div style="border: 1px solid black; padding: 2px; width: fit-content;">FVANNUIITY(0,01; 24; 1000)</div>
Montant du paiement par période	PMT( ) Renvoie 1000 : <div style="border: 1px solid black; padding: 2px; width: fit-content;">PMT(0,01; 24; 21243,39)</div>
Taux d'intérêt par période	RATE( ) Renvoie 0,00999999 (1 %) : <div style="border: 1px solid black; padding: 2px; width: fit-content;">RATE(24; 1000; 21243,39)</div>
Nombre de périodes	NPER( ) Renvoie 24,00 : <div style="border: 1px solid black; padding: 2px; width: fit-content;">NPER(0,01; 1000; 21243,39)</div>

### Formules de calcul des rentes

La formule permettant de calculer la **valeur actualisée** d'une rente ordinaire (paiement à la fin d'une période) :

La formule permettant de calculer la **valeur capitalisée** d'une rente ordinaire (paiement à la fin d'une période) :

## Remarques

### Fonctions associées

La fonction PVANNUIITY( ) est l'inverse de la fonction FVANNUIITY( ).

# Fonction FVLUMPSUM( )

Renvoie la valeur capitalisée d'un montant forfaitaire en cours calculé à l'aide d'un taux d'intérêt constant.

## Syntaxe

```
FVLUMPSUM(taux; périodes; quantité)
```

## Paramètres

Nom	Type	Description
<i>taux</i>	numérique	Taux d'intérêt par période.
<i>périodes</i>	numérique	Nombre total de périodes.
<i>quantité</i>	numérique	Investissement effectué au début de la première période.

### Remarque

Vous devez utiliser des périodes temporelles cohérentes lorsque vous spécifiez *taux* et *périodes* pour vous assurer que vous spécifiez un taux d'intérêt **par période**.

Par exemple :

- dans le cas de paiements mensuels, sur un emprunt ou un investissement d'une durée de deux ans avec un taux d'intérêt de 5 % par an, spécifiez 0,05/12 pour *taux* et 2 \* 12 pour *périodes*
- dans le cas de paiements annuels sur le même prêt ou investissement, indiquez 0,05 pour *taux* et 2 pour *périodes*

## Sortie

Numérique. Le résultat est calculé avec deux décimales.

# Exemples

## Exemples basiques

### Intérêts composés mensuels

Renvoie 1269,73, la valeur capitalisée d'un montant forfaitaire de 1 000 \$ investi pendant 2 ans au taux composé mensuel de 2 % par an :

```
FVLUMPSUM(0,01; 2*12; 1000)
```

Renvoie 1126,83, la valeur capitalisée du même investissement au bout de la première année :

```
FVLUMPSUM(0,01; 12; 1000)
```

Renvoie 27243,20, la valeur capitalisée de 21 455,82 \$ investi pendant 2 ans au taux composé mensuel de 1 % par mois :

```
FVLUMPSUM(0,01; 2*12; 21455,82)
```

### Intérêts composés semestriels

Renvoie 1262,48, la valeur capitalisée d'un montant forfaitaire de 1 000 \$ investi pendant 2 ans au taux composé semestriel de 12 % par an :

```
FVLUMPSUM(0,12/2; 2*2; 1000)
```

### Intérêts composés annuels

Renvoie 1254,40, la valeur capitalisée d'un montant forfaitaire de 1 000 \$ investi pendant 2 ans au taux composé annuel de 12 % par an :

```
FVLUMPSUM(0,12; 2; 1000)
```

# Remarques

## Qu'est-ce que la valeur capitalisée ?

La valeur capitalisée d'un montant forfaitaire investi équivaut au capital de l'investissement initial augmenté des intérêts composés accumulés.

## Fonctions associées

La fonction PVLUMPSUM( ) est l'inverse de la fonction FVLUMPSUM( ).

# Fonction FVSCCHEDULE()

Renvoie la valeur capitalisée d'un montant forfaitaire en cours calculé à l'aide d'une série de taux d'intérêt.

## Syntaxe

```
FVSCCHEDULE(capital; taux1 <;taux2...>)
```

## Paramètres

Nom	Type	Description
<i>capital</i>	numérique	Montant de l'investissement initial.
<i>taux1, taux2...</i>	numérique	Une série de taux d'intérêt pour des périodes de durée égale. <div style="border-left: 2px solid #0056b3; padding-left: 10px; margin-left: 20px;"> <p><b>Remarque</b></p> <p>Ces périodes peuvent représenter des mois ou des années, ou une tout autre durée, du moment que le type de période est cohérent.</p> <p>Vous devez préciser les taux d'intérêt <b>par période</b>. Par exemple, si l'un des taux d'intérêt est de 5 % par an et que les périodes sont des mois, vous devez préciser 0,05 / 12.</p> </div>

## Sortie

Numérique. Le résultat est calculé avec deux décimales.

## Exemples

### Exemples basiques

Renvoie 1282,93, la valeur actualisée d'un montant forfaitaire de 1000 \$ investi sur 3 ans à un taux d'intérêt composé annuel de 10 % la première année, 9 % la deuxième et 7 % la troisième :

```
FVSCCHEDULE(1000; 0,1; 0,09; 0,07)
```



# Remarques

La valeur capitalisée d'un montant forfaitaire investi équivaut au capital de l'investissement initial augmenté des intérêts composés accumulés.

# Fonction GETOPTIONS()

Renvoie le paramètre actuel pour l'option Analytics spécifiée (paramètre de la boîte de dialogue **Options**).

## Syntaxe

```
GETOPTIONS(option)
```

## Paramètres

Nom	Type	Description
<i>option</i>	caractère	<p>L'option Analytics pour laquelle renvoyer un paramètre.</p> <p>Le nom de l'option doit être spécifié exactement tel qu'il apparaît dans la liste ci-dessous et il doit être entouré de guillemets :</p> <ul style="list-style-type: none"><li>◦ <b>separators</b> : renvoie les paramètres actuels pour les trois caractères séparateurs Analytics dans l'ordre suivant :<ul style="list-style-type: none"><li>• séparateur décimal</li><li>• séparateur des milliers</li><li>• séparateur de liste</li></ul></li></ul> <p><b>Remarque</b></p> <p>Actuellement, « SEPARATORS » est la seule <i>option</i> qui peut être spécifiée pour la fonction GETOPTIONS().</p>

## Sortie

Caractère.

## Exemples

### Exemples basiques

Renvoie les paramètres actuels pour les trois caractères de séparation Analytics. Par exemple, ".,,":

```
GETOPTIONS("SEPARATORS")
```

## Exemples avancés

### Utilisation de GETOPTIONS() dans un script

Si un script doit modifier un ou plusieurs caractères séparateurs Analytics, la fonction GETOPTIONS() fournit une méthode permettant de découvrir les paramètres actuels. Les paramètres actuels peuvent être stockés dans une variable, puis rétablis à la fin du script.

```
ASSIGN v_SeparatorsSetting = GETOPTIONS("separators")
SET SEPARATORS ",,;"
<contenu du script>
SET SEPARATORS "%v_SeparatorsSetting%"
```

## Remarques

Les trois caractères de séparation Analytics sont spécifiés pour les options suivantes dans la boîte de dialogue **Options** :

- Séparateur décimal
- Séparateur des milliers
- Séparateur de liste

# Fonction GOMONTH( )

Renvoie la date qui correspond au nombre de mois indiqué avant ou après une date indiquée.

## Syntaxe

```
GOMONTH(date/DateHeure; mois)
```

## Paramètres

Nom	Type	Description
<i>date/DateHeure</i>	DateHeure	Le champ, l'expression, ou la valeur littérale à partir desquels calculer la date.
<i>mois</i>	numérique	Le nombre de mois avant ou après <i>date/DateHeure</i> .  <div style="border-left: 2px solid #0056b3; padding-left: 10px; margin-left: 20px;"> <p><b>Remarque</b></p> <p>Vous pouvez indiquer une valeur de type DateHeure pour <i>date/DateHeure</i>, mais la portion de la valeur correspondant à l'heure sera ignorée.</p> </div>

## Sortie

DateHeure. La valeur de date est générée à l'aide du format d'affichage de date Analytics actuel.

## Exemples

### Exemples basiques

#### Valeurs d'introduction littérales

Renvoie `20140415` affiché au format 15 avr 2014 en supposant que le format d'affichage actuel de la date dans Analytics soit JJ MMM AAAA :

```
GOMONTH(`20140115`; 3)
```

Renvoie `20131015` affiché au format 15 oct 2013 en supposant que le format d'affichage actuel de la date dans Analytics soit JJ MMM AAAA :

```
GOMONTH(`20140115`; -3)
```

Renvoie `20140430` affiché au format 30 avr 2014 en supposant que le format d'affichage actuel de la date dans Analytics soit JJ MMM AAAA (l'arrondi de date évite de renvoyer le 31 avr 2014, qui est une date incorrecte) :

```
GOMONTH(`20140330`; 1)
```

```
GOMONTH(`20140331`; 1)
```

Renvoie `20140501` affiché au format 01 mai 2014 en supposant que le format d'affichage de la date actuel dans Analytics soit JJ MMM AAAA :

```
GOMONTH(`20140401`; 1)
```

## Valeurs d'introduction champ

Renvoie la date trois mois après chaque date dans le champ **Date\_facture** :

```
GOMONTH(Date_facture; 3)
```

Renvoie la date trois mois après chaque date dans le champ **Date\_facture** plus une période de grâce de 15 jours :

```
GOMONTH(Date_facture + 15; 3)
```

# Remarques

## Formats DateHeure

Un champ indiqué pour *date/DateHeure* peut utiliser n'importe quel format de date ou de DateHeure, tant que la définition de champ définit correctement le format.

Une valeur de date littérale doit utiliser l'un des formats suivants :

- AAAAMMJJ
- AAMMJJ

Vous devez saisir les valeurs de dates littérales entre guillemets inversés. Par exemple : `20141231`

## Fonctionnement de la valeur *mois*

- **Valeur positive** : la date générée est plus récente que la *date/DateHeure* indiquée
- **Valeur négative** : la date générée est antérieure à la *date/DateHeure* indiquée.
- **Valeur omise ou '0' (zéro)** : la date générée est la même que la *date/DateHeure*

## Arrondi de dates pour éviter les dates inexistantes

Si la combinaison de *date/dateheure* et *mois* génère une date inexistante, la fonction GOMONTH( ) utilise l'arrondi de date pour renvoyer la date correcte la plus proche à l'intérieur du même mois.

Renvoie `20140430` (30 avr 2014) car le 31 avr 2014 est une date incorrecte :

```
GOMONTH(`20140331`;1)
```

## Fonctions associées

Utilisez la fonction EOMONTH( ) si vous souhaitez renvoyer la date du dernier jour du mois, plutôt que la date exacte, qui correspond au nombre de mois indiqué avant ou après une date indiquée.

# Fonction HASH( )

Renvoie une valeur de hachage cryptographique salted à partir de la valeur d'introduction.

## Syntaxe

```
HASH(champ <;valeur_salt>)
```

## Paramètres

Nom	Type	Description
<i>champ</i>	caractère numérique DateHeure logique	Valeur à hacher.
<i>valeur_salt</i> optionnel	caractère numérique	<p>La valeur salt à utiliser. Vous pouvez spécifier un numéro d'identificateur PASSWORD compris entre 1 et 10 ou une chaîne de caractères.</p> <p>Si vous ignorez ce paramètre, la valeur salt par défaut Analytics est utilisée.</p> <p>La valeur salt est limitée à 128 caractères et est tronquée automatiquement à 128 caractères si vous spécifiez une valeur salt plus longue.</p> <p>Pour plus d'informations, consultez la section "La valeur salt" Page 626.</p>

## Sortie

Caractère.

# Exemples

## Exemples basiques

### Avec la valeur salt par défaut Analytics

Renvoie "819A974BB91215D58E7753FD5A42226150100A0763087CA7DECD93F3C3090405" :

```
HASH("555-44-3322")
```

Renvoie la valeur de hachage pour chaque nombre dans le champ **Num\_carte\_crédit** :

```
HASH(Num_carte_crédit)
```

### Avec une valeur salt spécifiée par l'utilisateur

Renvoie "AD1E7D9B97B6F6B5345AB13471A74C31EBE6630CA2622BB7E8C280E9FBEE1F17" :

```
HASH("555-44-3322"; "ma valeur salt 123")
```

## Exemples avancés

### S'assurer que les valeurs de hachage sont identiques

Utilisez d'autres fonctions conjointement avec la fonction HASH( ) pour standardiser les valeurs en texte clair qui devraient produire des valeurs de hachage identiques.

Prenez l'ensemble d'exemples suivant. Remarquez en quoi la casse des valeurs en texte clair modifie complètement la valeur de hachage de sortie dans les deux premiers exemples.

Renvoie "DF6789E1EC65055CD9CA17DD5B0BEA5892504DFE7661D258737AF7CB9DC46462" :

```
HASH("John Smith")
```

Renvoie "3E12EABB5940B7A2AD90A6B0710237B935FAB68E629907927A65B3AA7BE6781D" :

```
HASH("JOHN SMITH")
```

En utilisant la fonction UPPER( ) pour standardiser la casse, une valeur de hachage identique est créée.

Renvoie "3E12EABB5940B7A2AD90A6B0710237B935FAB68E629907927A65B3AA7BE6781D" :



```
HASH(UPPER("John Smith"))
```

## Utilisation de HASH( ) pour comparer les grands blocs de texte

Utilisez HASH( ) pour tester si les blocs de texte dans deux champs de commentaires sont identiques.

Pour effectuer ce test, créez deux champs calculés similaires à ceux qui sont illustrés ci-dessous, puis créez un filtre pour rechercher des blocs de texte qui ne sont pas identiques.

```
DEFINE FIELD Hachage_1 COMPUTED HASH(Champ_Commentaire_1)  
DEFINE FIELD Hachage_2 COMPUTED HASH(Champ_Commentaire_2)  
SET FILTER TO Hachage_1 <> Hachage_2
```

Si les champs de commentaires se trouvent dans des tables distinctes, créez un champ calculé HASH( ) dans chaque table, puis utilisez les champs calculés comme champ clé commun pour effectuer une jointure dissociée des deux tables. Les enregistrements dans la table de sortie jointe représentent des blocs de texte non identiques.

## Remarques

### Quand utiliser la fonction HASH( )

Utilisez la fonction HASH( ) pour protéger les données sensibles, comme les numéros de carte de crédit, les informations de salaire ou les numéros de sécurité sociale.

### Fonctionnement

HASH( ) est une fonction d'encodage en une seule passe. Des données dans le texte en clair peuvent être utilisées pour générer une valeur de hachage. Toutefois, il ne sera pas possible d'enlever l'encodage à la valeur de hachage ni de la décrypter par la suite.

Une valeur de texte en clair spécifique génère toujours la même valeur de hachage. Vous pouvez donc rechercher dans un champ de numéros de carte de crédit hachés des doublons ou joindre deux champs de numéros de carte de crédit hachés, et les résultats seront les mêmes que si vous aviez effectué l'opération sur des champs de texte en clair équivalents.

### Protection des données sensibles

Pour éviter de stocker des données sensibles sur un serveur, vous pouvez créer un champ calculé en local à l'aide de la fonction HASH( ), puis créer une nouvelle table en extrayant le champ haché et tous les autres champs requis, tout en excluant le champ de texte en clair. Vous pouvez utiliser la nouvelle table sur le serveur pour votre analyse, puis une fois que vous avez les résultats, vous pouvez vous reporter à la table d'origine si vous avez besoin de consulter la version du texte en clair de certaines données hachées.

S'il est interdit de stocker en local les données sensibles après leur utilisation initiale, vous pouvez supprimer la table d'origine après avoir créé la nouvelle table avec les valeurs hachées et vous reporter au système source d'origine pour les valeurs du texte en clair.

## Les valeurs en texte clair doivent être exactement identiques

Pour générer des valeurs de hachage identiques, deux valeurs de texte en clair doivent être identiques exactement. Par exemple, différentes valeurs de hachage sont produites à partir du même numéro de carte de crédit avec ou sans trait d'union, du même nom avec la première lettre en majuscule ou toutes les lettres en majuscules.

Il se peut que vous deviez intégrer des fonctions comme `INCLUDE()`, `EXCLUDE()` ou `UPPER()` dans la fonction `HASH()` pour standardiser les valeurs de texte en clair.

Les espaces de début et de fin sont automatiquement coupés par la fonction `HASH()` ; il est donc inutile d'utiliser les fonctions `TRIM()` ou `ALLTRIM()`.

## Que se passe-t-il si les espaces de début ou de fin sont significatives ?

Si vous disposez de données dans lesquelles les espaces de début et de fin représentent des différences significatives entre les valeurs, vous devez remplacer les valeurs vides par un autre caractère avant de hacher les valeurs.

Remplace les espaces dans les valeurs des champs par le caractère de soulignement (`_`) avant le hachage :

```
HASH(REPLACE(nom_champ," ","_"))
```

## Algorithme de chiffrement utilisé par `HASH()`

La fonction `HASH()` utilise un algorithme de hachage cryptographique SHA-2 produisant une sortie hachée à longueur fixe de 64 octets, indépendamment de la longueur de la valeur d'introduction. La valeur d'introduction du texte en clair peut avoir une longueur supérieure à 64 octets.

## La valeur salt

### Fonctionnement

La protection offerte par la fonction `HASH()` est renforcée par l'ajout automatique d'une valeur salt avant le hachage. La valeur salt est une chaîne alphanumérique concaténée avec la valeur de données source. L'intégralité de la chaîne de caractères concaténée est ensuite utilisée pour produire la valeur hachée salted. Cette approche rend les valeurs hachées plus résistantes aux techniques de décodage.

## Facultatif : préciser votre propre valeur salt

Une valeur salt fixe est utilisée automatiquement sauf si vous spécifiez une valeur salt. Vous pouvez aussi utiliser l'une des méthodes suivantes pour spécifier une valeur salt :

- **Valeur salt comme chaîne de texte clair**

Spécifier une chaîne de caractères alphanumériques. Par exemple :

```
HASH(Num_carte_crédit; "ma valeur salt")
```

- **Valeur salt comme mot de passe**

Utiliser la commande PASSWORD conjointement avec la fonction HASH( ) et spécifier un numéro d'identificateur PASSWORD compris entre 1 et 10. Par exemple :

```
PASSWORD 3 "Saisir une valeur salt"  
EXTRACT FIELDS HASH(Num_carte_crédit; 3) TO "Table_protégée"
```

### Remarque

La valeur salt PASSWORD doit être saisie avant que le champ dans la fonction HASH( ) peut être extrait.

L'intérêt d'utiliser un numéro d'identificateur PASSWORD avec HASH( ) est que vous n'avez pas à exposer une valeur salt du texte en clair.

Pour plus d'informations, consultez la section "Commande PASSWORD" Page 373.

## Instructions relatives à la méthode du mot de passe

La méthode du mot de passe est destinée à être utilisée dans les scripts qui invitent à saisir le mot de passe au début du script ou avant que la fonction HASH( ) apparaisse dans le script.

La méthode du mot de passe ne convient pas à une utilisation dans des champs calculés, car les affectations PASSWORD sont supprimées lorsque vous fermez Analytics.

En outre, les champs calculés qui utilisent une valeur salt basée sur un mot de passe sont supprimés automatiquement des vues lorsque vous rouvrez Analytics. Cette suppression est nécessaire pour éviter le recalcul des valeurs hachées à l'aide de la valeur salt par défaut. Les valeurs recalculées diffèrent des valeurs hachées d'origine calculées avec une valeur salt fournie par l'utilisateur.

# Fonction HEX( )

Convertit une chaîne ASCII en chaîne hexadécimale.

## Syntaxe

```
HEX(champ)
```

## Paramètres

Nom	Type	Description
<i>champ</i>	caractère	La chaîne ASCII à convertir en chaîne hexadécimale.

## Sortie

Caractère.

## Exemples

### Exemples basiques

Renvoie "3132333435" :

```
HEX("12345")
```

Renvoie les valeurs dans le champ **Compte** sous forme de chaînes de caractères hexadécimaux :

```
HEX(Compte)
```

# Remarques

## Fonctionnement

Cette fonction renvoie la chaîne hexadécimale équivalente à la valeur de champ ou à l'expression que vous indiquez. Elle permet d'identifier le contenu exact d'un champ, y compris les caractères ne pouvant pas s'afficher à l'écran, comme les retours chariot, les sauts de ligne et NUL (valeur null).

## Longueur de la valeur renvoyée

La valeur renvoyée est une chaîne deux fois plus longue que la valeur *champ*. Les chiffres de 0 à 9 et les lettres de A à F (pour les chiffres de 10 à 15) représentent les 16 valeurs hexadécimales.

## Utiliser comme valeurs d'entrée des champs plutôt que des expressions

En général, il est recommandé d'utiliser cette fonction avec des champs plutôt qu'avec des expressions. En effet, HEX() affiche une représentation du format de stockage interne des expressions, qui n'est pas pertinente dans la plupart des cas.

# Fonction HOUR()

Extrait l'heure d'une heure ou d'une DateHeure spécifiée et la renvoie sous forme de valeur numérique à l'aide de l'horloge de 24 heures.

## Syntaxe

```
HOUR(heure/DateHeure)
```

## Paramètres

Nom	Type	Description
<i>heure/DateHeure</i>	DateHeure	Le champ, l'expression, ou la valeur littérale à partir desquels extraire l'heure.

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 23 :

```
HOUR('t235959')
```

```
HOUR('20141231 235959')
```

Renvoie l'heure pour chaque valeur dans le champ **Appel\_heure\_début** :

```
HOUR(Appel_heure_début)
```

# Remarques

## Informations sur les paramètres

Un champ indiqué pour *heure/DateHeure* peut utiliser n'importe quel format d'heure ou de *DateHeure*, tant que la définition de champ définit correctement le format.

### Spécification d'une valeur littérale d'heure ou de *DateHeure*

Lorsque vous indiquez une valeur littérale d'heure ou de *DateHeure* pour *heure/DateHeure*, vous êtes limité aux formats de la table ci-dessous, et vous devez entourer la valeur de guillemets inversés - par exemple, `20141231 235959`.

N'utilisez aucun séparateur comme les barres obliques (/) ou les deux points (:) entre les composants individuels des dates ou des heures.

- **Valeurs heure** : vous pouvez utiliser n'importe quel format d'heure répertorié dans le tableau ci-dessous. Vous devez utiliser un séparateur avant une valeur d'heure autonome pour que la fonction s'exécute correctement. Les séparateurs valides sont la lettre « t » ou la lettre « T ». Vous devez indiquer les heures à l'aide de l'horloge de 24 heures. Les décalages par rapport à l'heure universelle coordonnée (UTC) doivent être préfacés d'un signe plus (+) ou d'un signe moins (-).
- **Valeurs *DateHeure*** : vous pouvez utiliser n'importe laquelle des combinaisons de formats de date, de séparateur et d'heure répertoriées dans le tableau ci-dessous. La date doit précéder l'heure, et vous devez utiliser un séparateur entre les deux. Les séparateurs valides sont un espace unique, la lettre « t » ou la lettre « T ».

Exemples de formats	Exemples de valeurs littérales
thhmmss	`t235959`
Thhmm	`T2359`
AAAAMMJJ hhmmss	`20141231 235959`
AAMMJJthhmm	`141231t2359`
AAAAMMJJThh	`20141231T23`
AAAAMMJJ hhmmss+/-hhmm (décalage UTC)	`20141231 235959-0500`
AAMMJJ hhmm+/-hh (décalage UTC)	`141231 2359+01`

Exemples de formats	Exemples de valeurs littérales
<p><b>Remarque</b></p> <p>N'utilisez pas hh seul dans le format d'heure principal avec des données qui ont un décalage UTC. Par exemple, évitez : hh+hhmm. Il se peut que les résultats ne soient pas fiables.</p>	



# Fonction HTOU( )

Convertit une chaîne hexadécimale en chaîne de caractères Unicode. Abréviation de "Hexadécimal à Unicode".

## Remarque

Cette fonction est spécifique à l'édition Unicode d'Analytics. Elle n'est pas prise en charge dans l'édition non-Unicode.

## Syntaxe

```
HTOU(hex_chaine_de_caractères)
```

## Paramètres

Nom	Type	Description
<i>hex_chaine_de_caractères</i>	caractère	La chaîne hexadécimale à convertir en chaîne Unicode. La chaîne peut uniquement contenir des valeurs hexadécimales, par exemple « 20AC ».

## Sortie

Caractère.

## Exemples

### Exemples basiques

Renvoie "ABC123" :

```
HTOU("004100420043003100320033")
```

### Exemples avancés

#### Ajout d'un symbole de devise à une valeur

Vous devez extraire un champ extraire vers une nouvelle table. Le champ doit afficher la valeur du champ numérique **Montant** et y ajouter comme préfixe un symbole de devise euro (€) :

```
EXTRACT HTOU("20AC") + STRING(Montant; 10) AS "Montant_Devis" TO Table_Affichage
```

Lorsque la commande EXTRACT s'exécute, HTOU( ) renvoie le symbole Euro "« € »" et le concatène avec la valeur **Montant** que STRING( ) convertit en caractère. Si la valeur d'origine de **Montant** était 2000, la valeur de **Montant\_Devis** est « €2000 ».

## Remarques

### Fonctions associées

HTOU( ) est l'inverse de la fonction DHEX( ), qui convertit une chaîne Unicode en chaîne hexadécimale.

# Fonction INCLUDE()

Renvoie une chaîne qui inclut uniquement les caractères spécifiés.

## Syntaxe

```
INCLUDE(chaîne_de_caractères; caractères_à_inclure)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	Valeur à limiter aux caractères inclus.
<i>caractères_à_inclure</i>	caractère	Liste des caractères à inclure. Si vous indiquez des guillemets doubles dans les <i>caractères_à_inclure</i> , vous devez placer la liste de caractères à l'intérieur de guillemets simples. Par exemple : "'-/'

**Remarque**

Si un caractère que vous spécifiez à inclure n'apparaît pas dans *chaîne de caractères*, il n'est pas inclus dans la valeur renvoyée.

## Sortie

Caractère.

## Exemples

### Exemples basiques

Renvoie "123", ce qui correspond à la chaîne d'entrée ne contenant que des chiffres :

```
INCLUDE("123 Main St."; "0123456789")
```

Renvoie "1231234", ce qui correspond à la chaîne d'entrée ne contenant que des chiffres :

```
INCLUDE("123-123-4"; "1243")
```

Renvoie "" (rien), car la chaîne d'entrée ne contient pas de "D" :

```
INCLUDE("ABC"; "D")
```

## Remarques

### Fonctionnement

La fonction INCLUDE() compare chaque caractère de la *chaîne\_de\_caractères* avec les caractères de la liste *caractères\_à\_inclure*. En cas de correspondance, le caractère est inclus dans la chaîne de sortie.

### Aucun caractère correspondant

S'il n'existe aucune correspondance entre *chaîne\_de\_caractères* et *caractères\_à\_inclure*, la sortie de la fonction est vide.

### Respect de la casse

La fonction INCLUDE() est sensible à la casse. Si vous indiquez « ID » dans *caractères\_à\_inclure*, ces caractères ne seront pas inclus dans « id#94022 ». Si vous pensez que la casse est incohérente dans la *chaîne*, utilisez la fonction UPPER() pour convertir tous les caractères en majuscules.

Par exemple :

```
INCLUDE(UPPER("id#94022"), "ID0123456789")
```

### Conseil d'utilisation

Utilisez INCLUDE() si le jeu de caractères à inclure est petit et que le jeu à exclure est grand.

### Fonctions associées

La fonction INCLUDE() est l'opposé de la fonction EXCLUDE().

# Fonction INSERT()

Renvoie la chaîne d'origine avec le texte spécifié inséré à un emplacement d'octet spécifique.

## Syntaxe

```
INSERT(chaîne_de_caractères; caractères_à_insérer; emplacement)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	Valeur dans laquelle il faut insérer le texte.
<i>caractères_à_insérer</i>	caractère	Texte à insérer.
<i>emplacement</i>	numérique	Position du caractère où insérer <i>caractères_à_insérer</i> dans la <i>chaîne_de_caractères</i> .

## Sortie

Caractère.

## Exemples

### Exemples basiques

Renvoie "aXXXbcde" :

```
INSERT("abcde"; "XXX"; 2)
```

Renvoie "XXXabcde" :

```
INSERT("abcde"; "XXX"; 0)
```

Renvoie "abcdeXXX", avec "XXX" inséré en position d'octet 6 plutôt que 8, car "abcde" a seulement une longueur de 5 octets ::

```
INSERT("abcde"; "XXX"; 8)
```

## Remarques

### Fonctionnement

La fonction INSERT() insère des caractères spécifiés ou des espaces dans une chaîne de caractères, commençant à une position spécifiée dans la chaîne.

### Quand utiliser INSERT()

Utilisez INSERT() pour normaliser les données pour la mise en forme, la correspondance des doublons et pour les commandes JOIN et DEFINE RELATION, qui nécessitent que les champs soient identiques.

Par exemple, les numéros de parties dans un fichier peuvent être au format « 12345 », et dans un autre fichier, au format « 12-345 ». Dans ce premier fichier, vous pouvez utiliser INSERT() pour insérer un trait d'union (-) à la position 3.

### Instructions concernant l'emplacement

- Si la valeur d'*emplacement* est supérieure à la longueur de *chaîne\_de\_caractères*, la valeur de *caractères\_à\_insérer* est insérée à la fin de la chaîne.
- Si le paramètre *emplacement* est égal à 0 ou à 1, la valeur de *caractères\_à\_insérer* est insérée au début de la chaîne.

### Insertion de guillemets doubles

Si vous indiquez des guillemets doubles dans les *caractères\_à\_insérer*, vous devez les placer à l'intérieur de guillemets simples.

Par exemple : ""

# Fonction INT( )

Renvoie la valeur entière d'une expression numérique ou d'une valeur de champ.

## Syntaxe

```
INT(nombre)
```

## Paramètres

Nom	Type	Description
<i>nombre</i>	numérique	Champ ou expression numérique à convertir en nombre entier. Si la valeur indiquée inclut des décimales, celles-ci sont tronquées et la valeur n'est pas arrondie.

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 7 :

```
INT(7,9)
```

Renvoie -7 :

```
INT(-7,9)
```

# Fonction IPMT()

Renvoie les intérêts payés sur un emprunt pour une seule période.

## Syntaxe

```
IPMT(taux; période_spécifiée; périodes; quantité <; type>)
```

## Paramètres

Nom	Type	Description
<i>taux</i>	numérique	Taux d'intérêt par période.
<i>période_spécifiée</i>	numérique	Période pour laquelle il faut trouver les intérêts payés.
<i>périodes</i>	numérique	Nombre total de périodes de paiement.
<i>quantité</i>	numérique	Capital de l'emprunt.
<i>type</i> optionnel	numérique	Date des paiements : <ul style="list-style-type: none"> <li>○ 0 : paiement en fin de période</li> <li>○ 1 : paiement en début de période</li> </ul> Si vous ignorez ce paramètre, c'est la valeur par défaut de 0 qui est utilisée.

### Remarque

Vous devez utiliser des périodes temporelles cohérentes lorsque vous spécifiez *taux* et *périodes* pour vous assurer que vous spécifiez un taux d'intérêt **par période**.

Par exemple :

- dans le cas de paiements mensuels, sur un emprunt ou un investissement d'une durée de deux ans avec un taux d'intérêt de 5 % par an, spécifiez 0,05/12 pour *taux* et 2 \* 12 pour *périodes*
- dans le cas de paiements annuels sur le même prêt ou investissement, indiquez 0,05 pour *taux* et 2 pour *périodes*

## Sortie

Numérique.



# Exemples

## Exemples basiques

Renvoie 1489,58, le montant total des intérêts payés le premier mois d'un emprunt de 275 000 \$ sur vingt-cinq ans à 6,5 % par an dont les paiements ont une échéance à la fin de chaque mois :

```
IPMT(0,065/12; 1; 12*25; 275000; 0)
```

Renvoie 10,00, les intérêts payés sur le même emprunt lors du dernier mois de celui-ci :

```
IPMT(0,065/12; 300; 12*25; 275000; 0)
```

## Remarques

### Fonctions associées

La fonction PPMT ( ) est le complément de la fonction IPMT ( ).

La fonction CUMIPMT ( ) calcule les intérêts payés sur une plage de périodes.

# Fonction ISBLANK()

Renvoie une valeur logique indiquant si la valeur d'entrée est vide.

## Syntaxe

```
ISBLANK(chaîne_de_caractères)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	Valeur pour laquelle il faut déterminer la présence de données vides.

## Sortie

Logique. Renvoie T (« true », vrai) si la valeur du paramètre *chaîne\_de\_caractères* est vide ; renvoie F (« false », faux) dans les autres cas.

## Exemples

### Exemples basiques

Renvoie F :

```
ISBLANK(" A")
```

Renvoie T :

```
ISBLANK(" ")
```

```
ISBLANK("")
```

Renvoie T pour toutes les valeurs vides du champ **Adresse** ; renvoie « F » sinon :

```
ISBLANK(Adresse)
```

## Remarques

### Quand utiliser la fonction ISBLANK( )

Utilisez ISBLANK() pendant la phase d'intégrité des données d'un projet d'analyse pour identifier les champs contenant des données manquantes (ceci peut mettre en évidence la présence de problèmes au niveau des données source).

### Qu'est-ce qu'une entrée vierge ?

Pour que la fonction soit évaluée sur vrai, la valeur d'introduction doit respecter l'un des critères suivants :

- entièrement vide (c'est-à-dire qu'elle ne contient que des espaces)
- chaîne d'une longueur de zéro

Cette fonction n'identifie que des espaces blancs réels dans des données source, et non des caractères invalides qui apparaissent comme des espaces blancs dans une vue.

### Caractères de type Null

Il est possible que les résultats renvoyés par ISBLANK() ne soient pas concluants quand la fonction est utilisée avec des champs caractère contenant des caractères nuls. Analytics utilise les caractères nuls pour indiquer la fin d'une chaîne. Ainsi, la fonction ISBLANK( ) ne lit pas les données de caractère suivant un caractère nul (y compris les espaces).

# Fonction ISDEFINED( )

Renvoie **T** (« true », vrai) si le champ ou la variable indiqué est défini ; renvoie **F** (« false », faux) dans les autres cas.

## Syntaxe

```
ISDEFINED(chaîne_de_caractères)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	Nom du champ ou de la variable dont il faut vérifier l'existence. La valeur doit être entrée en tant que chaîne entre guillemets : <pre>ISDEFINED("v_limite_numérique")</pre>

## Sortie

Logique.

## Exemples

### Exemples basiques

Renvoie **T** si `v_limite_numérique` est défini comme variable ou comme champ, dans les autres cas renvoie **F** :

```
ISDEFINED("v_limite_numérique")
```

### Exemples avancés

#### Utilisation de ISDEFINED( ) pour tester un champ

Dans l'exemple suivant, la fonction ISDEFINED( ) est utilisée pour vérifier si le champ **Limite** est défini

dans la table avant l'extraction des enregistrements en fonction de la valeur indiquée dans le champ :

```
OPEN Metaphor_Employees_US  
IF ISDEFINED("Limite") EXTRACT RECORD IF Limite > 50000 TO "LimiteSup.fil"
```

# Fonction ISFUZZYDUP()

Renvoie une valeur logique indiquant si une chaîne est un doublon approximatif d'une chaîne de comparaison.

## Syntaxe

```
ISFUZZYDUP(chaîne_de_caractères1; chaîne_de_caractères2; levdist <; diffpct >)
```

## Paramètres

Nom	Type	Description
<i>chaîne_de_caractères1</i>	caractère	Première chaîne de la comparaison.
<i>chaîne_de_caractères2</i>	caractère	Deuxième chaîne de la comparaison.
<i>levdist</i>	numérique	Distance Levenshtein maximale autorisée entre les deux chaînes pour qu'elles soient identifiées comme doublons approximatifs. La valeur <i>levdist</i> ne peut pas être inférieure à 1 ou supérieure à 10. Le fait d'augmenter la valeur <i>levdist</i> augmente le nombre de résultats en incluant des valeurs d'un niveau d'approximation supérieur, c'est-à-dire des valeurs qui diffèrent davantage les unes des autres.
<i>diffpct</i> optionnel	numérique	Seuil supérieur du « pourcentage de différence ». Le concept de pourcentage de différence est expliqué en "Fonctionnement" Page 648. La valeur <i>diffpct</i> ne peut pas être inférieure à 1 ni supérieure à 99. Le fait d'augmenter la valeur <i>diffpct</i> augmente le nombre de résultats en incluant des valeurs ayant une proportion de différence supérieure relativement à leur longueur. Si le paramètre est ignoré, le pourcentage de différence n'est pas pris en compte lors du traitement de la fonction FUZZYDUP( ).

# Sortie

Logique. Renvoie **T** (« true », vrai) si les valeurs du paramètre *chaîne\_de\_caractères* sont des doublons approximatifs ; renvoie **F** (« false », faux) dans les autres cas.

## Exemples

### Exemples basiques

Renvoie **F**, car deux modifications sont nécessaires pour transformer « Smith » en « Smythe » ; cependant, la valeur *levdist* est 1 uniquement :

```
ISFUZZYDUP("Smith";"Smythe"; 1; 99)
```

Renvoie **T**, car deux modifications sont nécessaires pour transformer « Smith » en « Smythe », et la valeur *levdist* est 2 :

```
ISFUZZYDUP("Smith";"Smythe"; 2; 99)
```

Renvoie **T**, car aucune modification n'est nécessaire pour transformer « SMITH » en « smith », et la valeur *levdist* est 1 (la fonction ISFUZZYDUP() n'est pas sensible à la casse) :

```
ISFUZZYDUP("SMITH";"smith"; 1; 99)
```

Renvoie une valeur logique (T ou F) indiquant si les valeurs individuelles du champ **Nom** sont des doublons approximatifs pour la chaîne « Smith » :

```
ISFUZZYDUP(Nom;"Smith"; 3; 99)
```

## Exemples avancés

### Utilisation du pourcentage de différence

Le pourcentage de différence vous donne un outil permettant de réduire le nombre de faux positifs renvoyés par ISFUZZYDUP().

#### Aucun *diffpct* spécifié

Renvoie **T**, car cinq modifications sont nécessaires pour transformer « abc » en « Smith », et la valeur *levdist* est 5 :

```
ISFUZZYDUP("abc"; "Smith"; 5)
```

### ***diffpct* spécifié**

Renvoie F, même si "abc" se trouve dans la distance Levenshtein spécifiée de « Smith », car *5 modifications/une longueur de chaîne de 3* entraîne un pourcentage de différence de 167 %, ce qui dépasse le *diffpct* spécifié de 99 % :

```
ISFUZZYDUP("abc"; "Smith"; 5; 99)
```

Le concept de pourcentage de différence est entièrement expliqué en "Fonctionnement" bas.

### **Isoler les doublons approximatifs de « Smith »**

Créez un filtre isolant toutes les valeurs du champ **Nom** qui sont des doublons approximatifs de « Smith » :

```
SET FILTER TO ISFUZZYDUP(Nom; "Smith"; 3; 99)
```

Modifier les valeurs *levdist* ou *diffpct* vous permet d'ajuster la différence des valeurs filtrées.

## Remarques

### Quand utiliser la fonction ISFUZZYDUP( )

Utilisez la fonction ISFUZZYDUP( ) pour rechercher des valeurs presque identiques (des doublons approximatifs) ou pour repérer les incohérences orthographiques dans les données saisies manuellement.

## Fonctionnement

La fonction ISFUZZYDUP( ) calcule la distance Levenshtein séparant deux chaînes, ainsi que le pourcentage de différence.

ISFUZZYDUP( ) est évaluée sur T (true, vrai) si :

- La distance Levenshtein est inférieure ou égale à la valeur *levdist*.
- Le pourcentage de différence est inférieur ou égal à la valeur *diffpct* (le cas échéant).

### **Distance Levenshtein**

La distance Levenshtein représente le nombre minimal de modifications d'un caractère unique requises pour rendre une chaîne identique à une autre chaîne.

Pour plus d'informations, consultez la section "Fonction LEVDIST( )" Page 657.



## Pourcentage de différence

Le pourcentage de différence correspond au pourcentage de la plus courte des deux chaînes évaluées et qui présente des dissemblances.

Le pourcentage de différence est le résultat du calcul Analytics interne suivant (ce calcul se base sur la distance Levenshtein séparant les deux chaînes) :

*Distance Levenshtein / nombre de caractères dans la chaîne plus courte × 100 = pourcentage de différence*

Le recours au pourcentage de différence permet de réduire le nombre de faux positifs renvoyés par ISFUZZYDUP() :

- Le seuil supérieur de *diffpct* est 99 %, ce qui évite de remplacer entièrement une chaîne afin de la rendre identique.
- Les chaînes de caractères nécessitant un grand nombre de modifications par rapport à leur longueur sont exclues.

## Conseils d'utilisation

- **Sensibilité à la casse** : Cette fonction n'est pas sensible à la casse (par ex., « MARTIN » est équivalent à « martin »).
- **Espaces de fin** : Elle tronque les espaces de fin dans les champs de manière automatique. Vous n'avez donc pas besoin d'utiliser la fonction TRIM() lorsque vous renseignez un champ comme paramètre.
- **Suppression des éléments génériques** : La fonction OMIT() peut améliorer l'efficacité de la fonction ISFUZZYDUP() en supprimant des éléments génériques tels que « Corporation » ou « Inc. » des valeurs de champ.

La suppression d'éléments génériques recentre la comparaison de chaînes ISFUZZYDUP() uniquement sur la portion des chaînes dans lesquelles une différence significative peut se produire.

## Différences entre la commande FUZZYDUP et la fonction ISFUZZYDUP()

La commande FUZZYDUP identifie tous les doublons approximatifs d'un champ, les organise en groupes et renvoie des résultats non exhaustifs.

La fonction ISFUZZYDUP() identifie une liste exhaustive de doublons approximatifs pour une valeur de caractère unique.

La commande et la fonction identifient tous deux les doublons exacts. Contrairement à la commande, cette fonction ne vous permet pas d'exclure les doublons exacts.

### Qu'entend-on « exhaustif » ?

Exhaustif signifie que toutes les valeurs du degré de différence spécifié sont renvoyées, quelle que soit leur position dans le champ du test par rapport à la valeur du test.

La fonction ISFUZZYDUP( ) est utile si les résultats non-exhaustifs produits par la commande FUZZYDUP ne sont pas suffisants dans le cadre de votre analyse et si vous devez examiner directement chaque doublon approximatif pour rechercher une valeur de caractère spécifique.

## Fonctions associées

- **La fonction LEVDIST( )** : fournit une méthode alternative pour comparer des chaînes en fonction de la distance Levenshtein.

Contrairement à la fonction ISFUZZYDUP( ), la fonction LEVDIST( ) est sensible à la casse par défaut.

- La fonction **DICECOEFFICIENT( )** : ne met pas l'accent sur la position des caractères ou des blocs de caractères voire les ignore complètement lors de la comparaison de chaînes de caractères.
- **Les fonctions SOUNDSLIKE( ) et SOUNDEX( )** : comparent des chaînes de façon phonétique (son) plutôt qu'orthographique.

# Fonction LAST()

Renvoie un nombre spécifique de caractères à partir de la fin d'une chaîne.

## Syntaxe

```
LAST(chaîne_de_caractères; longueur)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	Valeur à partir de laquelle il faut renvoyer les caractères.
<i>longueur</i>	numérique	Nombre de caractères à renvoyer.

## Sortie

Caractère.

## Exemples

### Exemples basiques

Renvoie "Savings" :

```
LAST("Account Type: Savings"; 7)
```

Renvoie "efghi" :

```
LAST("abcdefghi"; 5)
```

Renvoie "fghi" :

```
LAST("abcdefghi "; 5)
```

Renvoie " abc", car la valeur *chaîne* est plus courte que la longueur de 6 indiquée, c'est pourquoi des espaces de tête sont ajoutées à la sortie :

```
LAST("abc"; 6)
```

## Remarques

### Résultats vides dus aux espaces de fin

La présence d'espaces de fin dans la *chaîne* peut entraîner des résultats vides produits par la fonction LAST().

Par exemple, la sortie pour LAST("6483-30384 "; 3) est " ".

Vous pouvez utiliser la fonction ALLTRIM( ) conjointement avec LAST( ) pour supprimer les espaces en fin de *chaîne*.

Par exemple, LAST(ALLTRIM("6483-30384 "); 3) renvoie "384".

### Caractères de retour en début de chaîne

Si vous souhaitez renvoyer un nombre spécifié de caractères à partir du début d'une chaîne, utilisez la fonction SUBSTR(). Pour plus d'informations, consultez la section "Fonction SUBSTR( )" Page 839.

# Fonction LEADING()

Renvoie une chaîne contenant un nombre spécifique de chiffres dominants.

## Syntaxe

```
LEADING(nombre; chiffres_dominants)
```

## Paramètres

Nom	Type	Description
<i>nombre</i>	numérique	Valeur à partir de laquelle il faut renvoyer les chiffres dominants.
<i>chiffres_dominants</i>	numérique	Nombre de chiffres dominants à renvoyer.

## Sortie

Caractère.

## Exemples

### Exemples basiques

#### Entrée numérique littérale

Renvoie 623 :

```
LEADING(6234,56; 3)
```

Renvoie 62345 :

```
LEADING(-6234,56; 5)
```

#### Remplissage avec des zéros non significatifs

Renvoie 000 :

```
LEADING(0,00; 3)
```

Renvoie 00000 :

```
LEADING(0,00; 5)
```

Renvoie 35500 :

```
LEADING(3,55; 5)
```

## Remarques

Utilisez LEADING( ) pour extraire des chiffres d'un champ numérique sous forme de chaîne, et pour filtrer les éléments qui ne sont pas des chiffres, comme les décimales ou les symboles dollar.

# Fonction LENGTH( )

Renvoie le nombre de caractères dans une chaîne.

## Syntaxe

```
LENGTH(chaîne_de_caractères)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	Valeur dont il faut trouver la longueur.

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 15 :

```
LENGTH("Entreprise ABC")
```

Renvoie la longueur en caractères du champ **Description** dans le format de table :

```
LENGTH(Description)
```

### Exemples avancés

#### Affichage de la longueur de chaque adresse dans un champ adresse

Créez un champ calculé qui affiche la longueur en caractères de chaque adresse dans le champ **Rue\_Fournisseur**. Les espaces vides de début et de fin sont d'abord coupés dans les valeurs adresse pour ne pas

être comptabilisées dans la longueur.

```
DEFINE FIELD Longueur_Adresse COMPUTED LENGTH(ALLTRIM(Rue_Fournisseur))
```

## Remarques

### Fonctionnement

La fonction LENGTH( ) compte le nombre de caractères dans *chaîne\_de\_caractères*, y compris les espaces, et renvoie ce nombre.

### Espaces de fin

Les espaces de fin comptent comme des caractères. Si vous ne souhaitez pas comptabiliser les espaces de fin, utilisez la fonction TRIM( ) ou ALLTRIM( ) pour les supprimer. Par exemple :

```
LENGTH(TRIM(Rue_Fournisseur))
```

Si vous créez un champ calculé pour afficher la longueur des valeurs dans un champ et que vous ne supprimez pas les espaces de fin, la longueur maximale du champ s'affiche pour chaque valeur.



# Fonction LEVDIST()

Renvoie la distance Levenshtein entre deux chaînes données. Il s'agit d'une mesure du niveau de différence entre les deux chaînes.

## Syntaxe

```
LEVDIST(chaîne_de_caractères1; chaîne_de_caractères2 <;sensible_casse>)
```

## Paramètres

Nom	Type	Description
<i>chaîne_de_caractères1</i>	caractère	Première chaîne de la comparaison.
<i>chaîne_de_caractères2</i>	caractère	Deuxième chaîne de la comparaison.
<i>sensible_casse</i> optionnel	logique	Indiquez T pour obtenir une comparaison de chaînes sensible à la casse, ou F pour ignorer la casse.  Si vous ignorez ce paramètre, c'est la valeur par défaut de T qui est utilisée.

## Sortie

Numérique. La valeur est la distance Levenshtein entre deux chaînes.

## Exemples

### Exemples basiques

Renvoie 3, car deux substitutions et une insertion sont requises pour transformer « smith » en « Smythe » :

```
LEVDIST("smith";"Smythe")
```

Renvoie 2, car la casse étant ignorée, seules deux substitutions sont nécessaires pour transformer « smith's » en « Smythes » :

```
LEVDIST("smith's";"Smythes"; F)
```

Renvoie la distance Levenshtein entre chaque valeur dans le champ **Nom** et la chaîne « Smith » :

```
LEVDIST(TRIM(Nom);"Smith")
```

## Exemples avancés

### Classement des valeurs vis-à-vis de « Smith »

Créez le champ calculé **dist\_Lev** pour afficher la distance Levenshtein entre « Smith » et chaque valeur du champ **Nom** :

```
DEFINE FIELD dist_Lev COMPUTED LEVDIST(TRIM(Nom);"Smith"; F)
```

Ajoutez le champ calculé **dist\_Lev** à la vue, puis effectuez un tri express dans l'ordre croissant pour classer toutes les valeurs du champ **Nom** en fonction du nombre de différences qu'elles présentent par rapport à « Smith ».

### Isoler les doublons approximatifs de « Smith »

Créez un filtre isolant toutes les valeurs du champ **Nom** se trouvant dans la distance Levenshtein spécifiée par rapport à « Smith » :

```
SET FILTER TO LEVDIST(TRIM(Nom);"Smith"; F) < 3
```

Modifier le nombre dans l'expression vous permet de régler l'importance de la distance Levenshtein dans les valeurs filtrées.

## Remarques

### Quand utiliser la fonction LEVDIST( )

Utilisez la fonction LEVDIST( ) pour rechercher des valeurs presque identiques (des doublons approximatifs) ou pour repérer les incohérences orthographiques dans les données saisies manuellement. LEVDIST( ) identifie également les doublons exacts.

### Fonctionnement

La fonction LEVDIST( ) renvoie la distance Levenshtein entre les deux chaînes évaluées. Cela correspond à une valeur représentant le nombre minimal de modifications de caractères simples requises pour rendre une chaîne identique à une autre.

Chaque modification requise incrémente la valeur de la distance Levenshtein de 1. Plus la distance Levenshtein est grande, plus la différence entre les deux chaînes est importante. Une distance égale à zéro (0) signifie que les chaînes sont identiques.

## Types de modifications

Il existe trois types de modifications :

- insertion
- suppression
- substitution

Les transpositions (deux lettres adjacentes inversées) ne sont pas reconnues par l'algorithme Levenshtein, et comptent comme deux modifications - en particulier, deux substitutions.

## Caractères non alphanumériques

Les signes de ponctuation, les caractères spéciaux et les espaces vides sont considérés comme des caractères simples, à l'instar des lettres et des chiffres.

## Casse des caractères

Modifier la casse d'un caractère est considéré comme étant une substitution, à moins que vous ne désactiviez la sensibilité à la casse à l'aide du paramètre *sensible\_casse*.

## Position des caractères

La distance Levenshtein tient compte de la position des caractères. Les mêmes caractères ordonnés différemment peuvent engendrer une distance Levenshtein différente.

Renvoie 2 :

```
LEVDIST("abc"; "dec")
```

Renvoie 3 :

```
LEVDIST("abc"; "cde")
```

## Utilisation de TRIM( ) avec LEVDIST( )

Pour obtenir des résultats exacts lors de la comparaison d'une chaîne littérale (par ex., « Smith ») et d'un champ caractère à l'aide la fonction LEVDIST( ), vous devez d'abord utiliser TRIM( ) pour supprimer les espaces de fin du champ.

Si vous comparez deux champs, vous devez utiliser la fonction TRIM( ) avec chaque champ.

L'algorithme Levenshtein comptabilise les espaces en tant que caractères. De ce fait, les espaces de fin sont inclus dans le calcul du nombre de modifications requises pour rendre deux chaînes identiques.

## Utilisation de OMIT( ) avec LEVDIST( )

La fonction OMIT( ) peut améliorer l'efficacité de la fonction LEVDIST( ) en supprimant des éléments génériques tels que « Corporation » ou « Inc. » des valeurs de champ. La suppression d'éléments génériques recentre la comparaison de chaînes LEVDIST( ) uniquement sur la portion des chaînes dans lesquelles une différence significative peut se produire.

## Fonctions associées

- **La fonction ISFUZZYDUP( )** : fournit une méthode alternative pour comparer des chaînes en fonction de la distance Levenshtein.

Contrairement au comportement par défaut de la fonction LEVDIST( ), la fonction ISFUZZYDUP( ) n'est pas sensible à la casse.

- La fonction **DICECOEFFICIENT( )** : ne met pas l'accent sur la position des caractères ou des blocs de caractères voire les ignore complètement lors de la comparaison de chaînes de caractères.
- **Les fonctions SOUNDSLIKE( ) et SOUNDEX( )** : comparent des chaînes de façon phonétique (son) plutôt qu'orthographique.

# Fonction LOG( )

Renvoie le logarithme (base 10) d'une expression numérique ou d'une valeur de champ comportant un nombre spécifique de décimales.

## Syntaxe

```
LOG(nombre; décimales)
```

## Paramètres

Nom	Type	Description
<i>nombre</i>	numérique	Valeur dont il faut trouver le logarithme.
<i>décimales</i>	numérique	Nombre de décimales pour la valeur renvoyée.

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 3,0000 :

```
LOG(1000; 4)
```

Renvoie 4,86 :

```
LOG(72443; 2)
```

### Exemples avancés

#### Rechercher la racine cubique

Crée un champ qui est la racine cubique du champ X à deux décimales :

```
DEFINE FIELD Racine_cubique COMPUTED EXP(LOG(X; 6) / 3; 2)
```

### Remarque

Pour déterminer la *N*ème racine, divisez la trace de la valeur par *n* et relevez la valeur exponentielle du résultat.

## Remarques

### Fonctionnement

Le logarithme d'un nombre est l'exposant (ou puissance) 10 nécessaire pour générer ce nombre. Ainsi, le logarithme de 1 000 est 3.

### Fonctions associées

La fonction LOG( ) est l'inverse de la fonction EXP( ).

# Fonction LOWER( )

Renvoie une chaîne dont les caractères alphabétiques ont été convertis en minuscules.

## Syntaxe

```
LOWER(chaîne_de_caractères)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	Valeur à convertir en minuscules.

## Sortie

Caractère.

## Exemples

### Exemples basiques

Renvoie « abc » :

```
LOWER("ABC")
```

Renvoie « abc 123 def » :

```
LOWER("abc 123 DEF")
```

Renvoie « abcd 12 » :

```
LOWER("AbCd 12")
```

Renvoie toutes les valeurs du champ **Nom** converties en minuscules :

```
LOWER(Nom)
```

## Remarques

### Fonctionnement

La fonction LOWER( ) convertit tous les caractères alphabétiques de la *chaîne\_de\_caractères* en minuscules. Tous les caractères non alphabétiques restent inchangés.

### Quand utiliser la fonction LOWER( )

Utilisez la fonction LOWER( ) pour rechercher les données dont la casse est mélangée ou inconnue, ou bien pour mettre en forme des données en minuscules.



# Fonction LTRIM( )

Renvoie une chaîne dont les espaces de début ont été supprimées de la chaîne d'entrée.

## Syntaxe

```
LTRIM(chaîne_de_caractères)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	Valeur dont il faut supprimer les espaces de début.

## Sortie

Caractère.

## Exemples

### Exemples basiques

Vous remarquerez que dans ces deux exemples, la fonction LTRIM( ) ne supprime pas les espaces de fin.

Renvoie "Vancouver " :

```
LTRIM(" Vancouver ")
```

Renvoie "New York " :

```
LTRIM(" New York ")
```

### Exemples avancés

#### Suppression des espaces insécables

La fonction LTRIM( ) ne supprime pas les espaces insécables.

Si vous devez supprimer les espaces insécables de début, créez un champ calculé à l'aide de l'expression suivante :

```
DEFINE FIELD Description_nettoyée COMPUTED LTRIM(REPLACE(Description; CHR(160);  
CHR(32)))
```

La fonction REPLACE( ) remplace les espaces insécables par des espaces classiques, puis la fonction LTRIM( ) supprime les espaces classiques de début.

## Remarques

### Fonctionnement

La fonction LTRIM( ) supprime uniquement les espaces de début. Les espaces se trouvant à l'intérieur de la chaîne et les espaces de fin ne sont pas supprimés.

### Fonctions associées

La fonction LTRIM( ) est liée à la fonction TRIM( ), qui supprime les espaces de fin d'une chaîne, et à la fonction ALLTRIM( ), qui supprime les espaces de début et de fin.

# Fonction MAP( )

Renvoie une valeur logique indiquant si une chaîne de caractères correspond à une chaîne au format spécifié avec des caractères génériques, des caractères littéraux ou les deux.

## Syntaxe

```
MAP(chaîne_de_caractères; format)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	Le champ, l'expression, ou la valeur littérale à tester pour les rapprochements.
<i>format</i>	caractère	<p>Le modèle de données, ou la chaîne de caractères, que vous souhaitez comparer avec la <i>chaîne</i>.</p> <p>Le <i>format</i> peut contenir des caractères génériques, des caractères littéraux ou une combinaison des deux.</p> <pre>"\9\9-999-9999"</pre> <p>Les caractères génériques suivants sont pris en charge :</p> <ul style="list-style-type: none"> <li>○ "X" : permet de comparer tous les caractères alphabétiques (a-z, A-Z, caractères européens). Ce caractère générique n'est pas sensible à la casse. Vous pouvez utiliser "X" ou "x"</li> <li>○ "9" : permet de comparer tous les nombres (0-9).</li> <li>○ "!" : permet de comparer tous les caractères non vides.</li> <li>○ "?" : permet de comparer tous les caractères, y compris les caractères vides.</li> <li>○ "\" : caractère d'échappement permettant d'indiquer que le caractère suivant immédiatement est un caractère littéral. Utilisez le caractère d'échappement si vous souhaitez faire correspondre littéralement l'un des caractères génériques (X, x, 9, !, ?)</li> <li>○ "\\\" : indique un antislash littéral</li> </ul>

## Sortie

Logique. Renvoie **T** (« true », vrai) si une correspondance est trouvée ; renvoie **F** (« false », faux) dans les autres cas.

# Exemples

## Exemples basiques

### Modèles de recherche simples

Renvoie T :

```
MAP("ABC Plumbing"; "xxx")
```

Renvoie F (*chaîne* comporte uniquement 3 chiffres alors qu'un minimum de 4 chiffres est requis) :

```
MAP("045"; "9999")
```

### Échappement d'un caractère générique

Si l'objectif consiste à renvoyer T uniquement pour les valeurs commençant par le caractère littéral « X » suivi d'une deuxième lettre, le paramètre *format* "XX" s'assure que le premier « X » du paramètre est interprété littéralement et non comme un caractère générique.

Renvoie T :

```
MAP("XA-123"; "XX")
```

```
MAP("GC-123"; "XX")
```

```
MAP("XA-123"; "\XX")
```

Renvoie F :

```
MAP("GC-123"; "\XX")
```

### Champs et modèles

Renvoie T pour tous les enregistrements dont les numéros de factures se composent de deux lettres suivies de cinq chiffres ou bien commencent par deux lettres suivies de cinq chiffres. Renvoie F autrement :

```
MAP(No_Facture; "XX99999")
```

Renvoie T pour tous les enregistrements comportant les numéros de facture "AB12345" ou pour tous les enregistrements comportant des numéros commençant par "AB12345". Renvoie F autrement :

```
MAP(No_Facture; "AB12345")
```

Renvoie T pour tous les enregistrements dont les numéros de facture se composent de "AB" suivis de cinq chiffres ou commencent par "AB" suivies de cinq chiffres. Renvoie F autrement :

```
MAP(No_Facture; "AB99999")
```

Renvoie T pour tous les enregistrements dont le format ne correspond pas au format standard des numéros de sécurité sociale dans le champ SSN. Renvoie F autrement :

```
NOT MAP(SSN; "999-99-9999")
```

## Exemples avancés

### Extraction des enregistrements avec des codes de produits de 10 caractères et avec les caractères de tête "859-"

Utilisez une instruction IF et la fonction MAP( ) pour extraire uniquement les enregistrements dont les codes produits présentent une longueur de 10 caractères et dont les premiers chiffres sont "859-":

```
EXTRACT RECORD IF MAP(Code_Produit; "85\9-999999") TO "Codes_Longs_859"
```

## Remarques

### Quand utiliser la fonction MAP( )

Utilisez la fonction MAP( ) pour rechercher des modèles ou des formats particuliers parmi les données alphanumériques. Les modèles ou formats peuvent se composer de caractères génériques, de caractères littéraux ou d'une combinaison des deux.

### Respect de la casse

La fonction MAP( ) respecte la casse lorsqu'elle compare deux caractères littéraux. Par exemple, « a » n'est pas équivalent à « A ».

Si *chaîne\_de\_caractères* inclut des données de caractères avec des incohérences de casse, vous pouvez utiliser la fonction UPPER( ) pour convertir les valeurs et remédier aux problèmes de casse avant de vous servir de la fonction MAP( ).

Par exemple :

```
MAP(UPPER(Numéro_Facture); "AB99999")
```

## Rapprochement partiel

MAP() prend en charge le rapprochement partiel dans une situation mais pas dans l'autre.

Le rapprochement partiel dans la fonction MAP() n'est pas affecté par l'option **Comparaisons exactes de caractères** (SET EXACT ON/OFF).

### Rapprochement partiel pris en charge

Le rapprochement partiel est pris en charge si la valeur *format* est plus courte que la valeur *chaîne\_de\_caractères*.

Renvoie T, car *format* fait 7 caractères et *chaîne\_de\_caractères* en fait 9 :

```
MAP("AB1234567"; "AB99999")
```

#### Remarque

Pour renvoyer vrai, la valeur *format* doit apparaître au début de la valeur *chaîne\_de\_caractères*.

### Rapprochement partiel non pris en charge

Le rapprochement partiel n'est pas pris en charge si la valeur *format* est plus longue que la valeur *chaîne\_de\_caractères*.

Renvoie F, car *format* fait 7 caractères et *chaîne\_de\_caractères* en fait 6 :

```
MAP("AB1234"; "AB99999")
```

Si *format* est plus long que *chaîne\_de\_caractères*, le résultat est toujours Faux.

## Traitement des espaces

Les espaces vides sont traitées comme des caractères ; elles peuvent être traitées de l'une de ces deux façons :

- comparer des valeurs vides littéralement, en incluant des valeurs vides dans le paramètre *format* à la place appropriée
- utiliser le caractère générique « ? », qui permet de comparer tous les caractères, y compris les caractères vides.

Si besoin, vous pouvez utiliser les fonctions TRIM(), LTRIM() et ALLTRIM() pour supprimer les espaces de fin ou de début de *chaîne\_de\_caractères*, les fonctions LTRIM() ou ALLTRIM() pour supprimer les espaces de début, dans le paramètre *chaîne\_de\_caractères*, et vous assurer que seuls les caractères de texte et les vides internes sont comparés.

## Concaténation de champs

Vous pouvez concaténer deux champs ou plus dans *chaîne\_de\_caractères* si vous souhaitez que votre recherche porte sur plusieurs champs d'une table. Les champs concaténés sont traités comme un seul champ incluant les espaces de début et de fin des champs individuels sauf si vous utilisez la fonction ALLTRIM( ) pour supprimer les espaces.

# Fonction MASK()

Effectue une opération au niveau du bit AND sur les premiers octets des chaînes de deux caractères.

## Syntaxe

```
MASK(caractère; caractère_masque)
```

## Paramètres

Nom	Type	Description
<i>caractère</i>	caractère	Chaîne de caractères comportant l'octet à tester.
<i>caractère_masque</i>	caractère	Chaîne de caractères comportant l'octet par rapport auquel effectuer le test (valeur de masque).

## Sortie

Caractère. La sortie est la représentation de type caractère du résultat binaire de l'opération AND au niveau du bit.

## Exemples

### Exemples basiques

Renvoie "2" (00110010), le résultat d'une opération AND au niveau du bit de 3 (00110011) et 6 (00110110) :

```
MASK("3"; "6")
```



# Remarques

## Quand utiliser la fonction MASK( )

Utilisez MASK( ) pour identifier des modèles de bits spécifiques dans un octet de données, même si un bit particulier est défini sur 1 ou pas.

## Fonctionnement

La fonction MASK( ) effectue une opération AND au niveau du bit sur les représentations binaires des premiers caractères de *caractère* et *caractère\_masque*. Les deux octets de comparaison sont comparés un bit à la fois, ce qui crée une troisième valeur binaire.

Le résultat de chaque comparaison des bits correspondants est 1 ou 0 :

bit <i>caractère</i>	bit <i>caractère_masque</i>	Résultat
0	0	0
0	1	0
1	0	0
1	1	1

## Longueur des chaînes de comparaison supérieure à 1 octet

Si l'une des chaînes de comparaison est plus longue qu'un octet, les caractères suivants sont ignorés.

# Fonction MATCH( )

Renvoie une valeur logique indiquant si la valeur spécifiée correspond à l'une des valeurs à laquelle celle-ci est comparée.

## Syntaxe

```
MATCH(valeur_à_comparer, test <;...n>)
```

## Paramètres

Nom	Type	Description
<i>valeur_à_comparer</i>	caractère numérique DateHeure	Le champ, l'expression, ou la valeur littérale à tester pour les rapprochements.
<i>test</i> <;... <i>n</i> >	caractère numérique DateHeure	<p>Tout champ, expression ou valeur littérale que vous souhaitez comparer à <i>valeur_à_comparer</i>.</p> <p>Vous pouvez indiquer autant de valeurs de test que nécessaire, cependant, toutes les valeurs indiquées doivent appartenir au même type de données :</p> <pre>MATCH(<i>valeur_à_comparer</i>, `20140930`; `20141030`)</pre>

### Remarque

Les valeurs d'introduction de la fonction MATCH( ) peuvent être des données de type caractère, numérique ou DateHeure. Vous ne pouvez pas mélanger les types de données. Toutes les valeurs d'introduction doivent appartenir à la même catégorie de données.

## Sortie

Logique. Renvoie **T** (« true », vrai) si au moins une correspondance est trouvée ; renvoie **F** (« false », faux) dans les autres cas.

# Exemples

## Exemples basiques

### Remarque

Les valeurs de retour pour les comparaisons de caractères supposent que SET EXACT est sur OFF (paramètre par défaut), sauf indication contraire.

## Test de valeurs littérales

Renvoie T :

```
MATCH("ABC"; "BCD"; "CDE"; "AB")
```

Renvoie F :

```
MATCH(98; 99; 100; 101)
```

## Test d'un champ

Renvoie T pour tous les enregistrements dont le champ **Ville\_fournisseur** contient « Phoenix », « Austin » ou « Los Angeles ». Renvoie F autrement :

```
MATCH(Ville_fournisseur; "Phoenix"; "Austin"; "Los Angeles")
```

Renvoie T pour tous les enregistrements dont le champ **Ville\_fournisseur** ne contient pas « Phoenix », « Austin » ou « Los Angeles ». Renvoie F autrement :

```
NOT MATCH(Ville_fournisseur; "Phoenix"; "Austin"; "Los Angeles")
```

Renvoie T pour tous les enregistrements dont le champ **Ville\_fournisseur** contient « PHOENIX », « AUSTIN » ou « LOS ANGELES », indépendamment de la casse des caractères dans le champ. Renvoie F autrement :

Les valeurs du champ **Ville\_fournisseur** sont converties en majuscules avant d'être comparées avec les noms de villes (déjà en majuscules).

```
MATCH(UPPER(Ville_fournisseur); "PHOENIX"; "AUSTIN"; "LOS ANGELES")
```

## Test de plusieurs champs

Renvoie T pour tous les enregistrements dont le champ **Ville\_fournisseur**, **Ville** ou **Ville\_2** contient « Phoenix ». Renvoie F autrement :

```
MATCH("Phoenix"; Ville_fournisseur; Ville; Ville_2)
```

## Comportement de SET EXACT

Renvoie T pour tous les enregistrements dont le champ **Code\_Produit** comporte les codes produit "A", "D" ou "F" ou des codes produit commençant par "A", "D" ou "F". Renvoie F autrement :

```
MATCH(Code_Produit; "A"; "D"; "F")
```

Renvoie T pour tous tous les enregistrements dont le champ **Code\_Produit** comporte un code produit à un seul caractère "A", "D" ou "F". Renvoie F dans les autres cas (SET EXACT doit être ON) :

```
MATCH(Code_Produit; "A"; "D"; "F")
```

## Comparaison de deux champs

Renvoie T pour tous les enregistrements qui contiennent des adresses de fournisseur et d'employé identiques. Renvoie F autrement :

Vous devrez peut-être utiliser des fonctions supplémentaires pour normaliser le format des adresses fournisseur et employé.

```
MATCH(Adresse_fournisseur; Adresse_employé)
```

## Comparaison de dates

Renvoie T pour tous les enregistrements dont la date de facture est le 30 septembre 2014 ou le 30 octobre 2014. Renvoie F autrement :

```
MATCH(Date_facture; `20140930`; `20141030`)
```

## Exemples avancés

### Extraction des enregistrements de stock irréguliers

Utilisez une instruction IF et la fonction MATCH( ) pour extraire tous les enregistrements contenant différents montants dans le champ **Valeur\_de\_stock\_au\_coût** et dans le champ calculé **Coût\_x\_Quantité** :

```
EXTRACT RECORD IF NOT MATCH(Valeur_de_stock_au_coût; Coût_x_Quantité) TO "Montants_
non_correspondants"
```

### Extraction d'enregistrements pour les départements 101, 103 et 107

Utilisez une instruction IF et la fonction MATCH( ) pour extraire uniquement les enregistrements associés aux départements 101, 103 ou 107 :

```
EXTRACT RECORD IF MATCH(Dept; "101"; "103"; "107") TO "Trois_Départements"
```

## Remarques

### Utiliser MATCH( ) plutôt que l'opérateur OR

La fonction MATCH( ) permet de remplacer des expressions qui utilisent l'opérateur OR.

Par exemple :

```
MATCH(Ville; "Phoenix"; "Austin"; "Los Angeles")
```

est équivalent à

```
Ville="Phoenix" OR Ville="Austin" OR Ville="Los Angeles"
```

### Précision décimale des valeurs d'introduction numériques

Lorsque la précision décimale des valeurs d'introduction numériques comparées est différente, la comparaison utilise le niveau de précision le plus élevé.

Renvoie T, car 1,23 est égal à 1,23 :

```
MATCH(1,23; 1,23; 1,25)
```

Renvoie F, car 1,23 n'est pas égal à 1,234 si l'on tient compte de la troisième décimale :

```
MATCH(1,23; 1,234; 1,25)
```

### Paramètres de type caractère

#### Respect de la casse

La fonction MATCH( ) est sensible à la casse lorsque vous l'utilisez avec des données caractères. Lors de la comparaison des caractères, « a » n'est pas considéré comme étant équivalent à « A ».

Renvoie F :

```
MATCH("a";"A";"B";"C")
```

Si vous travaillez avec des données qui incluent des incohérences de casse, vous pouvez utiliser la fonction UPPER( ) pour convertir les valeurs et remédier aux problèmes de casse avant de vous servir de la fonction MATCH( ).

Renvoie T :

```
MATCH(UPPER("a"); UPPER("A"); UPPER("B"); UPPER("C"))
```

## Rapprochement partiel

Le rapprochement partiel est pris en charge pour les comparaisons de caractère. Chaque valeur comparée peut être contenue par l'autre valeur et être considérée comme une correspondance.

Ces deux exemples renvoient T :

```
MATCH("AB"; "ABC")
```

```
MATCH("ABC"; "AB")
```

### Remarque

La plus petite valeur doit apparaître au début de la valeur plus longue pour constituer une correspondance.

## Rapprochement partiel et SET EXACT

Le rapprochement partiel est activé lorsque SET EXACT = OFF, qui est le paramètre par défaut d'Analytics. Si SET EXACT = ON, le rapprochement partiel est désactivé et les valeurs de comparaison doivent correspondre exactement pour constituer un rapprochement.

Les deux exemples précédents sont False (faux) lorsque SET EXACT est sur ON.

Pour plus d'informations sur SET EXACT (option **Comparaisons exactes de caractères**), consultez la rubrique "Commande SET" Page 435.

## Activation et désactivation de SET EXACT

Si vous souhaitez vous assurer que l'option **Comparaisons exactes de caractères** n'est pas utilisée avec la fonction MATCH( ), vérifiez que l'option est décochée dans l'onglet **Table** de la boîte de dialogue **Options (Outils > Options)**.

Si vous utilisez un script, vous pouvez ajouter la commande SET EXACT OFF avant l'apparition de la fonction MATCH( ). Si nécessaire, vous pouvez restaurer l'état antérieur grâce à la commande SET EXACT ON.

## Paramètres DateHeure

Un champ date, DateHeure ou heure indiqué comme entrée de fonction peut utiliser n'importe quel format de date, DateHeure ou heure, tant que la définition de champ définit correctement le format.

### Mélange d'entrées de type date, DateHeure et heure

Il n'est pas impossible de mélanger des valeurs de date, DateHeure et heure dans les entrées de la fonction MATCH(), mais le fait de mélanger ces sous-types de DateHeure peut aboutir à des résultats qui n'ont pas de sens.

Analytics utilise des équivalents en chiffres sériels pour traiter les calculs de DateHeure, de sorte que même si vous êtes intéressé uniquement par la portion correspondant à la date d'une valeur DateHeure, la portion correspondant à l'heure fait toujours partie du calcul.

Prenons les exemples suivants :

Renvoie T, car le 31 décembre 2014 correspond à la deuxième valeur *test* :

```
MATCH(`20141231`;`20141229`;`20141231`)
```

Renvoie F, même si la *valeur\_à\_comparer* et la deuxième valeur *test* ont une date identique au 31 décembre 2014 :

```
MATCH(`20141231 120000`;`20141229`;`20141231`)
```

Si nous regardons l'équivalent en chiffre sériel de ces deux expressions, nous pouvons voir pourquoi le second génère une évaluation qui a pour résultat F (faux).

Renvoie T, car la *valeur\_à\_comparer* du nombre sériel est égale au *test* du deuxième nombre sériel :

```
MATCH(42003,000000; 42001,000000; 42003,000000)
```

Renvoie F, car le nombre sériel *valeur\_à\_comparer* n'est égal à aucune valeur de *test* :

```
MATCH(42003,500000; 42001,000000; 42003,000000)
```

La partie date des nombres sériels 42003,500000 et 42003,000000 correspond, mais ce n'est pas le cas des parties heure. 0,500000 est le nombre sériel équivalent de 12:00 PM.

### Harmoniser les sous-types DateHeure

Pour éviter les problèmes pouvant découler du mélange de sous-types DateHeure, vous pouvez utiliser des fonctions pour harmoniser les sous-types.

Par exemple, cette expression, qui utilise les mêmes valeurs initiales que le deuxième exemple ci-dessus, renvoie T plutôt que F :

```
MATCH(CTOD( DATE(`20141231 120000`; "AAAAMMJJ"); "AAAAMMJJ"); `20141229`; `20141231`)
```

## Spécification d'une valeur littérale de date, de DateHeure ou d'heure

Lorsque vous indiquez une valeur littérale de date, DateHeure ou heure pour l'une des entrées de fonction, vous êtes limité aux formats de la table ci-dessous, et vous devez entourer la valeur de guillemets inversés – par exemple, `20141231`.

N'utilisez aucun séparateur comme les barres obliques (/) ou les deux points (:) entre les composants individuels des dates ou des heures.

- **Valeurs DateHeure** : vous pouvez utiliser n'importe laquelle des combinaisons de formats de date, de séparateur et d'heure répertoriées dans le tableau ci-dessous. La date doit précéder l'heure, et vous devez utiliser un séparateur entre les deux. Les séparateurs valides sont un espace unique, la lettre « t » ou la lettre « T ».
- **Valeurs heure** : vous devez indiquer l'heure à l'aide de l'horloge de 24 heures. Les décalages par rapport à l'heure universelle coordonnée (UTC) doivent être préfacés d'un signe plus (+) ou d'un signe moins (-).

Exemples de formats	Exemples de valeurs littérales
AAAAMMJJ	`20141231`
AAMMJJ	`141231`
AAAAMMJJ hhmmss	`20141231 235959`
AAMMJJthhmm	`141231t2359`
AAAAMMJJThh	`20141231T23`
AAAAMMJJ hhmmss+/-hhmm (décalage UTC)	`20141231 235959-0500`
AAMMJJ hhmm+/-hh (décalage UTC)	`141231 2359+01`
thhmmss	`t235959`
Thhmm	`T2359`



Exemples de formats	Exemples de valeurs littérales
<p><b>Remarque</b></p> <p>N'utilisez pas hh seul dans le format d'heure principal avec des données qui ont un décalage UTC. Par exemple, évitez : hh+hhmm. Il se peut que les résultats ne soient pas fiables.</p>	

# Fonction MAXIMUM( )

Renvoie la valeur maximale d'un jeu de valeurs numériques ou bien la valeur la plus récente d'un jeu de valeurs DateHeure.

## Syntaxe

```
MAXIMUM(valeur_1; valeur_2 <;...n>)
```

## Paramètres

Nom	Type	Description
<i>valeur_1; valeur_2</i> <;...n>	numérique DateHeure	Les valeurs à comparer, séparées par des virgules. Toutes les valeurs doivent être du même type de données. De plus, les valeurs DateHeure doivent être du même sous-type. Vous ne pouvez pas mélanger des valeurs de type date, DateHeure ou heure dans une seule exécution de la fonction.

## Sortie

Numérique ou DateHeure.

## Exemples

### Exemples basiques

#### Entrée numérique littérale

Renvoie 7 :

```
MAXIMUM(4; 7)
```

Renvoie 8 :

```
MAXIMUM(4; 7; 3; 8)
```

Renvoie 8,00 :

```
MAXIMUM(4; 7,25; 3; 8)
```

## Entrée DateHeure littérale

Renvoie `20161231` :

```
MAXIMUM(`20161231`; `20161229`; `20161230`)
```

Renvoie `20161231 23:59:59` :

```
MAXIMUM(`20161231 235959`; `20161229 235959`)
```

Renvoie `23:59:59` :

```
MAXIMUM(`0,235957`; `0,235959`; `0,235958`)
```

## Entrée de champ

Renvoie la date la plus récente parmi les trois champs pour chaque enregistrement :

```
MAXIMUM(Date_BdC; Date_facture; Date_paiement)
```

## Exemples avancés

### Création d'un champ calculé avec une valeur minimale par défaut

Si vous avez une table des comptes impayés, créez un champ calculé **Intérêt\_dû** d'une valeur minimale par défaut de 1,00 € :

```
DEFINE FIELD Intérêt_dû COMPUTED MAXIMUM(BALANCE * ANNUAL_RATE; 1)
```

Si le solde multiplié par le taux d'intérêt est inférieur à 1 €, `MAXIMUM()` renvoie 1. Sinon, `MAXIMUM()` renvoie le montant des intérêts calculés.

### Découverte de dates après la fin d'un trimestre

Pour découvrir si des dates de plusieurs champs se trouvent après la fin d'un trimestre, créez un champ calculé avec une expression ressemblant à celle-ci :

```
DEFINE FIELD Past_Qtr COMPUTED MAXIMUM(Date_BdC, Date_facture; Date_paiement;  
`20160331`)
```

- Les enregistrements contenant toutes les dates avant le 31 mars 2016 inclus renvoient `20160331`.
- Les enregistrements contenant une ou plusieurs dates ultérieures au 31 mars 2016 renvoient la date la plus récente parmi les trois champs.

## Remarques

### Fonctionnement des décimales dans les jeux de valeurs numériques

Si les valeurs numériques comparées n'ont pas le même nombre de décimales, le résultat est ajusté pour correspondre au plus grand nombre de décimales.

Renvoie 20,400 :

```
MAXIMUM(3,682; 10,88; 20,4)
```

La fonction DECIMALS( ) permet d'adapter le nombre de décimales attribué aux paramètres *valeur*.

Renvoie 20,40 :

```
MAXIMUM(DECIMALS(3,682; 2); 10,88; 20,4)
```

# Fonction MINIMUM( )

Renvoie la valeur minimale d'un jeu de valeurs numériques ou bien la valeur la plus ancienne d'un jeu de valeurs DateHeure.

## Syntaxe

```
MINIMUM(valeur_1; valeur_2<;...n>)
```

## Paramètres

Nom	Type	Description
<i>valeur_1; valeur_2&lt;;...n&gt;</i>	numérique DateHeure	Les valeurs à comparer, séparées par des virgules. Toutes les valeurs doivent être du même type de données. De plus, les valeurs DateHeure doivent être du même sous-type. Vous ne pouvez pas mélanger des valeurs de type date, DateHeure ou heure dans une seule exécution de la fonction.

## Sortie

Numérique ou DateHeure.

## Exemples

### Exemples basiques

#### Entrée numérique littérale

Renvoie 4 :

```
MINIMUM(4; 7)
```

Renvoie 3 :

```
MINIMUM(4; 7; 3; 8)
```

Renvoie 3,00 :

```
MINIMUM(4; 7,25; 3; 8)
```

## Entrée DateHeure littérale

Renvoie `20161229` :

```
MINIMUM(`20161231`; `20161229`; `20161230`)
```

Renvoie `20161229 23:59:59` :

```
MINIMUM(`20161231 235959`; `20161229 235959`)
```

Renvoie `23:59:57` :

```
MINIMUM(`0,235957`; `0,235959`; `0,235958`)
```

## Entrée de champ

Renvoie la date la plus ancienne parmi les trois champs pour chaque enregistrement :

```
MINIMUM(Date_BdC; Date_facture; Date_paiement)
```

## Exemples avancés

### Identification de la valeur la plus basse parmi plusieurs champs

Créez un champ calculé pour identifier la valeur la plus basse parmi les champs **Coût**, **Prix\_vente**, et **Prix\_réduction** :

```
DEFINE FIELD Valeur_faible COMPUTED MINIMUM(Coût; Prix_vente; Prix_réduction)
```

### Découverte de dates avant le début d'un trimestre

Pour découvrir si des dates de plusieurs champs se trouvent avant le début d'un trimestre, créez un champ calculé avec une expression ressemblant à celle ci-dessous :

```
DEFINE FIELD Pre_Qtr COMPUTED MINIMUM(Date_BdC; Date_facture; Date_paiement;  
`20160101`)
```

- Les enregistrements contenant toutes les dates après le 1er janvier 2016 inclus renvoient `20160101`.
- Les enregistrements contenant une ou plusieurs dates antérieures au 1er janvier 2016 renvoient la date la plus ancienne parmi les trois champs.

## Remarques

### Fonctionnement des décimales dans les jeux de valeurs numériques

Si les valeurs numériques comparées n'ont pas le même nombre de décimales, le résultat est ajusté pour correspondre au plus grand nombre de décimales.

Renvoie 3,600 :

```
MINIMUM(3,6;10,88; 20,482)
```

La fonction DECIMALS( ) permet d'adapter le nombre de décimales attribué aux paramètres *valeur*.

Renvoie 3,60 :

```
MINIMUM(3,6;10,88; DECIMALS(20,482; 2))
```

### L'abréviation MIN( )

Dans ACLScript, vous pouvez utiliser l'abréviation MIN( ) pour la fonction MINIMUM( ) même si elle n'identifie pas de façon unique la fonction, ce qui est la condition préalable normale pour les abréviations des noms de fonctions.

MIN pourrait également être l'abréviation de MINUTE( ), mais Analytics réserve l'abréviation MIN( ) pour la fonction MINIMUM( ).

# Fonction MINUTE()

Extrait les minutes d'une heure ou d'une DateHeure spécifiée et la renvoie sous forme de valeur numérique.

## Syntaxe

```
MINUTE(heure/DateHeure)
```

## Paramètres

Nom	Type	Description
<i>heure/DateHeure</i>	DateHeure	Le champ, l'expression, ou la valeur littérale à partir desquels extraire les minutes.

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 59 :

```
MINUTE(`t235930`)
```

```
MINUTE(`20141231 235930`)
```

Renvoie les minutes pour chaque valeur dans le champ **Appel\_heure\_début** :

```
MINUTE(Appel_heure_début)
```



# Remarques

## Abréviation de MINUTE( ) dans des scripts

Dans ACLScript, si vous abrégez la fonction MINUTE( ), vous devez utiliser au moins les quatre premières lettres (MINU). Analytics réserve l'abréviation MIN pour la fonction MINIMUM( ).

## Informations sur les paramètres

Un champ indiqué pour *heure/DateHeure* peut utiliser n'importe quel format d'heure ou de DateHeure, tant que la définition de champ définit correctement le format.

### Spécification d'une valeur littérale d'heure ou de DateHeure

Lorsque vous indiquez une valeur littérale d'heure ou de DateHeure pour *heure/DateHeure*, vous êtes limité aux formats de la table ci-dessous, et vous devez entourer la valeur de guillemets inversés - par exemple, `20141231 235959`.

N'utilisez aucun séparateur comme les barres obliques (/) ou les deux points (:) entre les composants individuels des dates ou des heures.

- **Valeurs heure** : vous pouvez utiliser n'importe quel format d'heure répertorié dans le tableau ci-dessous. Vous devez utiliser un séparateur avant une valeur d'heure autonome pour que la fonction s'exécute correctement. Les séparateurs valides sont la lettre « t » ou la lettre « T ». Vous devez indiquer les heures à l'aide de l'horloge de 24 heures. Les décalages par rapport à l'heure universelle coordonnée (UTC) doivent être préfacés d'un signe plus (+) ou d'un signe moins (-).
- **Valeurs DateHeure** : vous pouvez utiliser n'importe laquelle des combinaisons de formats de date, de séparateur et d'heure répertoriées dans le tableau ci-dessous. La date doit précéder l'heure, et vous devez utiliser un séparateur entre les deux. Les séparateurs valides sont un espace unique, la lettre « t » ou la lettre « T ».

Exemples de formats	Exemples de valeurs littérales
thhmmss	`t235959`
Thhmm	`T2359`
AAAAMMJJ hhmmss	`20141231 235959`
AAMMJJthhmm	`141231t2359`
AAAAMMJJThh	`20141231T23`
AAAAMMJJ hhmmss+/-hhmm (décalage UTC)	`20141231 235959-0500`
AAMMJJ hhmm+/-hh	`141231 2359+01`

Exemples de formats	Exemples de valeurs littérales
(décalage UTC)	
<p><b>Remarque</b></p> <p>N'utilisez pas hh seul dans le format d'heure principal avec des données qui ont un décalage UTC. Par exemple, évitez : hh+hhmm. Il se peut que les résultats ne soient pas fiables.</p>	

# Fonction MOD( )

Renvoie le reste de l'opération de division de deux nombres.

## Syntaxe

```
MOD(nombre; diviseur)
```

## Paramètres

Nom	Type	Description
<i>nombre</i>	numérique	Nombre à diviser.
<i>diviseur</i>	numérique	Nombre à utiliser pour diviser le paramètre <i>nombre</i> . Si <i>nombre</i> ou <i>diviseur</i> ou les deux inclut des décimales, la sortie a la même précision décimale que le paramètre d'entrée présentant le plus grand nombre de décimales. Far exemple, le résultat pour MOD (45,35; 5,3) est "2,95".

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 3 :

```
MOD(93; 10)
```

Renvoie 2,0 :

```
MOD(66; 16,00)
```

Renvoie 3,45 :

```
MOD(53,45; 10)
```

## Exemples avancés

### Calcul d'une date d'anniversaire

Définit un champ affichant le nombre de mois depuis le dernier anniversaire :

```
DEFINE FIELD Mois_depuis_dernier_anniversaire COMPUTED MOD(Mois_de_service; 12)
```

## Remarques

### Quand utiliser la fonction MOD( )

Utilisez la fonction MOD( ) pour vérifier si deux nombres se divisent de manière égale, ou isoler le reste de la division. Elle divise un nombre par un autre et renvoie le reste.

# Fonction MONTH()

Extrait le mois d'une date ou d'une DateHeure spécifiée et le renvoie sous forme de valeur numérique (1 à 12).

## Syntaxe

```
MONTH(date/DateHeure)
```

## Paramètres

Nom	Type	Description
<i>date/DateHeure</i>	DateHeure	Le champ, l'expression, ou la valeur littérale à partir desquels extraire le mois.

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 12 :

```
MONTH('20141231')
```

```
MONTH('20141231 235959')
```

Renvoie le mois pour chaque valeur dans le champ **Date\_facture** :

```
MONTH(Date_facture)
```

# Remarques

## Informations sur les paramètres

Un champ indiqué pour *date/DateHeure* peut utiliser n'importe quel format de date ou de DateHeure, tant que la définition de champ définit correctement le format.

## Spécification d'une valeur littérale de date ou de DateHeure

Lorsque vous indiquez une valeur littérale de date ou de DateHeure pour *date/DateHeure*, vous êtes limité aux formats de la table ci-dessous, et vous devez entourer la valeur de guillemets inversés - par exemple, `20141231`.

N'utilisez aucun séparateur comme les barres obliques (/) ou les deux points (:) entre les composants individuels des dates ou des heures.

- **Valeurs DateHeure** : vous pouvez utiliser n'importe laquelle des combinaisons de formats de date, de séparateur et d'heure répertoriées dans le tableau ci-dessous. La date doit précéder l'heure, et vous devez utiliser un séparateur entre les deux. Les séparateurs valides sont un espace unique, la lettre « t » ou la lettre « T ».
- **Valeurs heure** : vous devez indiquer l'heure à l'aide de l'horloge de 24 heures. Les décalages par rapport à l'heure universelle coordonnée (UTC) doivent être préfacés d'un signe plus (+) ou d'un signe moins (-).

Exemples de formats	Exemples de valeurs littérales
AAAAMMJJ	`20141231`
AAMMJJ	`141231`
AAAAMMJJ hhmmss	`20141231 235959`
AAMMJJthhmm	`141231t2359`
AAAAMMJJThh	`20141231T23`
AAAAMMJJ hhmmss+/-hhmm (décalage UTC)	`20141231 235959-0500`
AAMMJJ hhmm+/-hh (décalage UTC)	`141231 2359+01`

Exemples de formats	Exemples de valeurs littérales
<p><b>Remarque</b></p> <p>N'utilisez pas hh seul dans le format d'heure principal avec des données qui ont un décalage UTC. Par exemple, évitez : hh+hhmm. Il se peut que les résultats ne soient pas fiables.</p>	

## Fonctions associées

Si vous devez renvoyer le nom du mois de l'année, utilisez la fonction CMOY( ) au lieu de MONTH( ).

# Fonction NOMINAL( )

Renvoie le taux d'intérêt annuel nominal d'un emprunt.

## Syntaxe

```
NOMINAL(taux_réel; périodes)
```

## Paramètres

Nom	Type	Description
<i>taux_réel</i>	numérique	Taux d'intérêt annuel réel.
<i>périodes</i>	numérique	Nombre de périodes composées par an.  <div style="border-left: 2px solid #4a7ebb; padding-left: 10px; margin-left: 20px;"> <p><b>Remarque</b> Indiquez un entier. Si vous indiquez une partie décimale, elle est tronquée.</p> </div>

## Sortie

Numérique. Le taux est calculé jusqu'à la huitième décimale.

## Exemples

### Exemples basiques

Renvoie 0,17998457 (18 %), soit le taux d'intérêt annuel nominal sur le solde non payé d'une carte de crédit dont les frais sont un taux annuel effectif de 19,56 % :

```
NOMINAL(0,1956; 12)
```



# Remarques

## Qu'est-ce que le taux d'intérêt nominal ?

Le taux d'intérêt annuel nominal d'un emprunt est le taux des intérêts payés établis ou publiés, sans tenir compte des intérêts composés mensuels ou journaliers sur le solde restant.

## Fonctions associées

La fonction EFFECTIVE( ) est l'inverse de la fonction NOMINAL( ).

# Fonction NORMDIST()

Renvoie la probabilité qu'une variable aléatoire d'un jeu de données d'une loi normale soit inférieure ou égale à une valeur spécifiée ou exactement égale à une valeur spécifiée.

## Syntaxe

```
NORMDIST(x; moyenne; écart_type; cumulatif)
```

## Paramètres

Nom	Type	Description
<i>x</i>	numérique	Valeur pour laquelle vous souhaitez calculer la probabilité.
<i>moyenne</i>	numérique	Valeur moyenne du jeu de données.
<i>écart_type</i>	numérique	Écart-type du jeu de données. La valeur <i>écart_type</i> doit être supérieure à 0.
<i>cumulatif</i>	logique	Spécifiez T pour calculer la probabilité qu'une variable aléatoire soit inférieure ou égale à <i>x</i> (probabilité cumulée) ou F pour calculer la probabilité qu'une variable aléatoire soit exactement égale à <i>x</i> (probabilité simple).

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 0,908788780274132 :

```
NORMDIST(42; 40; 1,5; T)
```

Renvoie 0,109340049783996 :

```
NORMDIST(42; 40; 1,5; F)
```

# Fonction NORMSINV( )

Renvoie la variable centrée réduite associée à une probabilité spécifiée dans une loi normale centrée réduite. La variable centrée réduite est le nombre d'écart-types qui séparent une valeur de la moyenne d'une loi normale centrée réduite.

## Syntaxe

```
NORMSINV(probabilité)
```

## Paramètres

Nom	Type	Description
<i>probabilité</i>	numérique	Probabilité pour laquelle vous souhaitez calculer la variable centrée réduite.

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 1,333401745213610 :

```
NORMSINV(0,9088)
```

# Fonction NOW( )

Renvoie l'heure du système d'exploitation en cours d'utilisation sous forme de données de type DateHeure.

## Syntaxe

```
NOW()
```

## Paramètres

Cette fonction ne comporte pas de paramètres.

## Sortie

DateHeure.

## Exemples

### Exemples basiques

Renvoie l'heure du système d'exploitation actuelle sous forme de valeur DateHeure, par exemple `t235959`, affichée au format d'affichage de l'heure actuelle d'Analytics :

```
NOW()
```

## Remarques

### Fonctions associées

Si vous devez renvoyer l'heure actuelle du système d'exploitation sous forme de chaîne de caractères, utilisez TIME( ) plutôt que NOW( ).

# Fonction NPER()

Renvoie les périodes nécessaires pour rembourser un emprunt.

## Syntaxe

```
NPER(taux; règlement; quantité <; type>)
```

## Paramètres

Nom	Type	Description
<i>taux</i>	numérique	Taux d'intérêt par période.
<i>règlement</i>	numérique	Paiement par période.
<i>quantité</i>	numérique	Capital de l'emprunt.
<i>type</i> optionnel	numérique	Date des paiements : <ul style="list-style-type: none"> <li>○ 0 : paiement en fin de période</li> <li>○ 1 : paiement en début de période</li> </ul> Si vous ignorez ce paramètre, c'est la valeur par défaut de 0 qui est utilisée.

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 300,00, le nombre de mois requis pour rembourser un emprunt de 275 000 \$, avec des paiements de 1856,82 \$ arrivant à échéance à la fin de chaque mois :

```
NPER(0,065/12; 1856,82; 275000; 0)
```

Renvoie 252,81, le nombre de mois requis pour rembourser le même emprunt, avec des paiements de 2000 \$ arrivant à échéance à la fin de chaque mois :

```
NPER(0,065/12; 2000; 275000; 0)
```

Renvoie 249,92, le nombre de mois requis pour rembourser le même emprunt, avec des paiements de 2000 \$ arrivant à échéance au début de chaque mois :

```
NPER(0,065/12; 2000; 275000; 1)
```

## Exemples avancés

### Calculs d'annuité

Les calculs d'annuité impliquent quatre variables :

- **valeur actualisée ou valeur capitalisée** : \$21243,39 et \$ 26973,46 dans les exemples ci-après
- **montant de paiement par période** : \$1000,00 dans les exemples ci-après
- **taux d'intérêt par période** : 1 % par mois dans les exemples ci-après
- **nombre de périodes** : 24 mois dans les exemples ci-après

Si vous connaissez la valeur de trois des variables, vous pouvez utiliser une fonction Analytics pour calculer la quatrième.

Je souhaite trouver :	Fonction Analytics à utiliser :
Valeur actualisée	PVANNUIITY() Renvoie 21243,39 : <pre>PVANNUIITY(0,01; 24; 1000)</pre>
Valeur capitalisée	FVANNUIITY() Renvoie 26973,46 : <pre>FVANNUIITY(0,01; 24; 1000)</pre>
Montant du paiement par période	PMT() Renvoie 1000 : <pre>PMT(0,01; 24; 21243,39)</pre>
Taux d'intérêt par période	RATE() Renvoie 0,00999999 (1 %) :

Je souhaite trouver :	Fonction Analytics à utiliser :
	<div style="border: 1px solid black; padding: 5px; width: fit-content;">RATE(24; 1000; 21243,39)</div>
Nombre de périodes	NPER() Renvoie 24,00 : <div style="border: 1px solid black; padding: 5px; width: fit-content;">NPER(0,01; 1000; 21243,39)</div>

### Formules de calcul des rentes

La formule permettant de calculer la **valeur actualisée** d'une rente ordinaire (paiement à la fin d'une période) :

La formule permettant de calculer la **valeur capitalisée** d'une rente ordinaire (paiement à la fin d'une période) :



# Fonction OCCURS()

Renvoie un décompte du nombre de fois qu'une sous-chaîne se produit dans une valeur de caractères indiquée.

## Syntaxe

```
OCCURS(chaîne_de caractères; chaîne_recherchée)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	Valeur dans laquelle effectuer la recherche. Vous pouvez concaténer deux champs ou plus si vous souhaitez que votre recherche porte sur plusieurs champs d'une table : <pre>OCCURS(Prénom+Nom;"John")</pre>
<i>chaîne_recherchée</i>	caractère	Valeur à rechercher. La recherche est sensible à la casse.

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 2 :

```
OCCURS("abc/abc/a";"ab")
```

Renvoie 3 :

```
OCCURS("abc/abc/a";"ab")
```

Renvoie le nombre de fois qu'un trait d'union apparaît dans chaque valeur du champ **No\_Facture** :

```
OCCURS(No_Facture; "-")
```

## Exemples avancés

### Recherche de numéros de facture comportant plusieurs traits d'union

Si les numéros de facture d'une table ne doivent avoir qu'un trait d'union, utilisez la fonction OCCURS() pour créer un filtre qui isole les numéros de facture ayant au moins deux traits d'union :

```
SET FILTER TO OCCURS(No_Facture; "-") > 1
```

### Recherche des occurrences d'une valeur de champ dans un autre champ

Utilisez OCCURS() pour rechercher les occurrences d'une valeur de champ dans un autre champ. Par exemple, vous pouvez créer un filtre isolant les enregistrements dans lesquels les valeurs **Nom** apparaissent dans le champ **Nom\_complet** :

```
SET FILTER TO OCCURS(Nom_complet; ALLTRIM(Nom)) = 1
```

Inclure la fonction ALLTRIM() dans l'expression retire les espaces de début ou de fin dans le champ **Nom**, ce qui permet de s'assurer que seules des valeurs de texte sont comparées.

### Recherches sensibles à la casse

À la différence de la fonction FIND(), la fonction OCCURS() est sensible à la casse, ce qui vous permet d'effectuer des recherches sensibles à la casse.

L'expression suivante isole tous les enregistrements contenant le nom « ÉQUIPEMENT UNI », en majuscules, dans le champ **Nom\_Fournisseur**, tout en ignorant les occurrences de « Équipement uni ».

```
SET FILTER TO OCCURS(Nom_Fournisseur; "ÉQUIPEMENT UNI") > 0
```

Si vous souhaitez rechercher toutes les occurrences d'"Équipement uni" quelle que soit la casse, utilisez la fonction UPPER() pour convertir les valeurs du champ de recherche en majuscules :

```
SET FILTER TO OCCURS(UPPER(Nom_Fournisseur); "ÉQUIPEMENT UNI") > 0
```

# Fonction OFFSET()

Renvoie la valeur d'un champ dont la position de départ a été décalée par un certain nombre d'octets.

## Syntaxe

```
OFFSET(champ; nombre_d'_octets)
```

## Paramètres

Nom	Type	Description
<i>champ</i>	caractère numérique DateHeure	Un nom de champ.
<i>nombre_d'_octets</i>	numérique	N'importe quelle expression numérique positive.

## Sortie

La valeur renvoyée appartient au même type de données que le paramètre d'entrée *champ*.

## Exemples

### Exemples basiques

Si un champ appelé « Nombre » contient la valeur « 1234567890 » et que vous définissez un champ chevauchant appelé « Nombre\_décalage » dont la position de début est 1, la longueur 3 et qui ne comporte aucune décimale, vous pouvez utiliser la fonction OFFSET() pour déplacer les nombres dans le champ.

Renvoie 123 :

```
OFFSET(Nombre_décalage;0)
```

Renvoie 234 :

```
OFFSET(Nombre_décalage;1)
```

Renvoie 789 :

```
OFFSET(Nombre_décalage;6)
```

## Remarques

Cette fonction permet de décaler temporairement la position de tête d'un champ. Ceci est utile si vous traitez des données dont la position de tête de champ est variable.

Si vous utilisez la fonction OFFSET( ) avec des champs calculés conditionnels, tout champ référencé dans le test IF sera également décalé.

# Fonction OMIT()

Renvoie une chaîne dont une ou plusieurs sous-chaînes spécifiques ont été supprimées.

## Syntaxe

```
OMIT(chaîne_de_caractères1; chaîne_de_caractères2 <;sensible_casse>)
```

## Paramètres

Nom	Type	Description
<i>chaîne_de_caractères1</i>	caractère	Chaîne dont il faut supprimer une ou plusieurs sous-chaînes.
<i>chaîne_de_caractères2</i>	caractère	<p>Une ou plusieurs sous-chaînes à supprimer.</p> <ul style="list-style-type: none"> <li>Utilisez des virgules pour séparer différentes sous-chaînes.</li> <li>Utilisez un espace après une virgule uniquement si cela fait partie de la sous-chaîne que vous souhaitez supprimer.</li> <li>Si les guillemets doubles sont présents dans au moins une des sous-chaînes, ajoutez des guillemets simples - plutôt que des guillemets doubles - de part et d'autre du paramètre <i>chaîne_de_caractères2</i> (c'est-à-dire, ' ').</li> <li>Pour ignorer une virgule, placez une seule virgule en dernier dans la liste des sous-chaînes, immédiatement suivie par le guillemet de fin (voir l'exemple final ci-dessous).</li> </ul>
<i>sensible_casse</i> optionnel	logique	<p>Indiquez T pour que les sous-chaînes soient sensibles à la casse ou F pour ignorer la casse.</p> <p>Si <i>sensible_casse</i> est ignoré, la valeur par défaut T est utilisée.</p>

## Sortie

Caractère.

# Exemples

## Exemples basiques

### Valeur d'entrée littérale de type caractère

Renvoie "Intercity Couriers" :

```
OMIT("Intercity Couriers Corporation"; " Corporation, Corp.")
```

Renvoie "Inter-city Couriers" :

```
OMIT("Inter-city Couriers Corp."; " Corporation, Corp.")
```

#### Remarque

La distance Levenshtein entre les valeurs renvoyées dans les deux premiers exemples est 1. Si les éléments génériques ne sont pas supprimés, la distance entre les deux exemples est 8, ce qui pourrait permettre aux valeurs d'échapper à une détection comme doublons approximatifs.

### Entrée de champ

Renvoie toutes les valeurs du champ **Nom\_Fournisseur** comportant des éléments génériques comme "Corporation" et "Inc." supprimé :

```
OMIT(Nom_Fournisseur;" Corporation, Corp., Corp, Inc., Inc, Ltd., Ltd")
```

Renvoie toutes les valeurs du champ **Nom\_Fournisseur** comportant des éléments génériques comme "Corporation" et "Inc." supprimé :

```
OMIT(Nom_Fournisseur;" ,.,Corporation,Corp,Inc,Ltd")
```

#### Remarque

Les deux exemples précédents renvoient les mêmes résultats, mais la syntaxe utilisée pour le deuxième est plus efficace.

Renvoie toutes les valeurs dans le champ **Nom\_Fournisseur** sans "Corporation", "Corp" ni virgule :

```
OMIT(Nom_Fournisseur;" Corporation, Corp,")
```

## Remarques

### OMIT( ) peut supprimer des sous-chaînes comme des unités

La fonction OMIT( ) supprime une ou plusieurs sous-chaînes d'une chaîne. Elle n'est pas équivalente aux fonctions CLEAN( ), EXCLUDE( ), INCLUDE( ) et REMOVE( ) en cela que les caractères sont supprimés en fonction des sous-chaînes et non selon un modèle caractère par caractère. La suppression des sous-chaînes vous permet de supprimer des mots spécifiques, des abréviations, ou des séquences de caractères répétées dans une chaîne sans affecter le reste de celle-ci.

### Une fonction d'aide pour les comparaisons approximatives

OMIT( ) peut améliorer l'efficacité des fonctions LEVDIST( ) et ISFUZZYDUP( ), ou des commandes FUZZYDUP ou FUZZYJOIN, en supprimant des éléments génériques tels que « Corporation » ou « Inc. » des valeurs de champ. La suppression des éléments génériques permet d'axer les comparaisons effectuées par ces fonctions et commandes sur la portion des chaînes qui peut inclure une différence significative.

### Comment l'ordre des sous-chaînes affecte les résultats

Si vous indiquez plusieurs sous-chaînes pour la suppression, l'ordre dans lequel vous les répertoriez dans *chaîne\_de\_caractères2* peut affecter les résultats de sortie.

Lorsque vous utilisez la fonction OMIT( ), la première sous-chaîne est supprimée de toutes les valeurs qui la contiennent. La deuxième est ensuite supprimée de la même manière, et ainsi de suite. Si une sous-chaîne fait partie d'une autre sous-chaîne - par exemple, « Corp » et « Corporation » - supprimer la plus courte entraîne des modifications dans l'autre sous-chaîne (« Corporation » devient « oration ») et empêche la plus longue d'être trouvée.

Pour éviter cela, indiquez les sous-chaînes longues avant les sous-chaînes plus courtes qu'elles peuvent contenir. Par exemple :

```
OMIT(Nom_Fournisseur;" Corporation, Corp., Corp")
```

### D'abord tenter de supprimer les caractères spéciaux

Vous pouvez indiquer des sous-chaînes à caractère unique, comme les signes de ponctuation, les caractères spéciaux et les espaces, ce qui permet de réduire les données génériques contenues dans les chaînes. Il peut s'avérer plus efficace de supprimer un caractère unique en premier lieu (un point ou un espace, par exemple), ce qui réduit le nombre de variations de sous-chaînes que vous devrez ensuite indiquer. Comparez les exemples numéro 3 et 4 ci-dessus. Ils renvoient tous deux les mêmes résultats, cependant, le quatrième exemple se révèle être plus efficace.

## Traitement des espaces

Les espaces dans les sous-chaînes sont traitées comme n'importe quel autre caractère. Vous devez préciser explicitement chacune des espaces que vous souhaitez supprimer dans une sous-chaîne. Par exemple, si vous indiquez une esperluette sans espace ("&"), "Ricoh Sales & Service" devient "Ricoh Sales Service". Si vous incluez les espaces (" & "), "Ricoh Sales & Service" devient "Ricoh SalesService".

Si vous indiquez une espace vide ne faisant pas partie de la sous-chaîne, cette dernière ne sera pas détectée. Par exemple, si vous indiquez une esperluette avec des espaces vides (" & "), "Ricoh Sales&Service" reste inchangé.

Lorsque vous utilisez des virgules pour séparer plusieurs sous-chaînes, faites suivre la virgule d'une espace uniquement si elle correspond à la sous-chaîne réelle que vous souhaitez supprimer.

Une approche permettant de gérer les espaces consiste à supprimer tous les espaces d'un champ, en indiquant un espace comme une sous-chaîne à caractère unique avant de préciser tout autre sous-chaîne.

## Examen des résultats suite à l'utilisation de la fonction OMIT( )

Après avoir utilisé la fonction OMIT( ) pour créer un champ calculé, passez le contenu du champ en revue et vérifiez que vous n'avez pas involontairement ignoré des portions de chaînes significatives. Par exemple, ignorer "Co" supprime une abréviation commune pour le mot "Company", mais cela résulte également en la suppression des lettres "Co" dans "Coca-Cola".



# Fonction PACKED()

Renvoie des données numériques converties en type de données Packed.

## Syntaxe

```
PACKED(nombre; longueur_résultat)
```

## Paramètres

Nom	Type	Description
<i>nombre</i>	numérique	Valeur numérique ou champ à convertir.
<i>longueur_résultat</i>	numérique	Nombre d'octets à utiliser dans la chaîne de sortie.

## Sortie

Numérique.

## Exemples

### Exemples basiques

#### Valeurs d'entrée entière et décimale

Renvoie 00075C :

```
PACKED(75; 3)
```

```
PACKED(7,5; 3)
```

#### Chiffres tronqués dans la sortie

Renvoie 00000012456D :

```
PACKED(-12,456; 6)
```

Renvoie 456D :

```
PACKED(-12,456; 2)
```

## Exemples avancés

### Création d'un champ de 8 octets pour actualiser un ordinateur central

Vous devez créer un champ de 8 octets contenant le salaire de chaque employé en tant que nombre PACKED en vue de le télécharger sur un ordinateur central :

```
EXTRACT PACKED(SALARY; 8) AS "Export_Salaire" TO "export"
```

## Remarques

### Que sont les données Packed ?

Le type de données Packed est utilisé par les systèmes d'exploitation d'ordinateur central pour stocker des valeurs numériques dans un format qui utilise un minimum d'espace de stockage. Le type de données Packed stocke deux chiffres par octet. Le dernier octet indique si la valeur est positive ou négative.

### Quand utiliser la fonction PACKED( )

Utilisez la fonction PACKED( ) pour convertir des données numériques au format Packed pour une exportation vers les systèmes centraux.

### Valeurs renvoyées tronquées

Si la valeur *longueur\_résultat* est inférieure à la longueur de la valeur *nombre*, les chiffres supplémentaires sont tronqués.

# Fonction PI( )

Renvoie la valeur de pi à 15 décimales.

## Syntaxe

```
PI( )
```

## Paramètres

Cette fonction ne comporte pas de paramètres.

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 3,141592653589793 (la valeur de pi à 15 décimales) :

```
PI( )
```

Renvoie 1,047197551196598 (l'équivalent en radians de 60 degrés) :

```
60 * PI( )/180
```

### Exemples avancés

#### Utilisation d'une valeur d'entrée en degrés

Renvoie 0,866025403784439 (le sinus de 60 degrés) :

```
SIN(60 * PI( )/180)
```

# Remarques

## Quand utiliser PI( )

Utilisez PI( ) pour convertir des degrés en radians :  $(\text{degrés} * \text{PI}()) / 180 = \text{radians}$ . Des radians doivent être entrés pour trois fonctions mathématiques d'Analytics : SIN( ), COS( ) et TAN( ).

# Fonction PMT( )

Renvoie le montant du paiement périodique (capital + intérêt) requis pour rembourser un emprunt.

## Syntaxe

```
PMT(taux; périodes; quantité <; type>)
```

## Paramètres

Nom	Type	Description
<i>taux</i>	numérique	Taux d'intérêt par période.
<i>périodes</i>	numérique	Nombre total de périodes de paiement.
<i>quantité</i>	numérique	Capital de l'emprunt.
<i>type</i> optionnel	numérique	Date des paiements : <ul style="list-style-type: none"> <li>◦ 0 : paiement en fin de période</li> <li>◦ 1 : paiement en début de période</li> </ul> Si vous ignorez ce paramètre, c'est la valeur par défaut de 0 qui est utilisée.

### Remarque

Vous devez utiliser des périodes temporelles cohérentes lorsque vous spécifiez *taux* et *périodes* pour vous assurer que vous spécifiez un taux d'intérêt **par période**.

Par exemple :

- dans le cas de paiements mensuels, sur un emprunt ou un investissement d'une durée de deux ans avec un taux d'intérêt de 5 % par an, spécifiez 0,05/12 pour *taux* et 2 \* 12 pour *périodes*
- dans le cas de paiements annuels sur le même prêt ou investissement, indiquez 0,05 pour *taux* et 2 pour *périodes*

## Sortie

Numérique.

# Exemples

## Exemples basiques

Renvoie 1856.82, le paiement mensuel (capital + intérêt) requis pour rembourser un emprunt 275 000 \$ sur 25 ans à 6,5 % par an, avec des paiements dus à la fin du mois :

```
PMT(0,065/12; 12*25; 275000; 0)
```

Renvoie 1856,82, le paiement mensuel (capital + intérêt) requis pour rembourser le même emprunt, avec des paiements dus au début du mois :

```
PMT(0,065/12; 12*25; 275000)
```

## Exemples avancés

### Calculs d'annuité

Les calculs d'annuité impliquent quatre variables :

- **valeur actualisée ou valeur capitalisée** : \$21243,39 et \$ 26973,46 dans les exemples ci-après
- **montant de paiement par période** : \$1000,00 dans les exemples ci-après
- **taux d'intérêt par période** : 1 % par mois dans les exemples ci-après
- **nombre de périodes** : 24 mois dans les exemples ci-après

Si vous connaissez la valeur de trois des variables, vous pouvez utiliser une fonction Analytics pour calculer la quatrième.

Je souhaite trouver :	Fonction Analytics à utiliser :
Valeur actualisée	PVANNUITY() Renvoie 21243,39 : <pre>PVANNUITY(0,01; 24; 1000)</pre>
Valeur capitalisée	FVANNUITY() Renvoie 26973,46 : <pre>FVANNUITY(0,01; 24; 1000)</pre>
Montant du paiement par période	PMT() Renvoie 1000 :

Je souhaite trouver :	Fonction Analytics à utiliser :
	PMT(0,01; 24; 21243,39)
Taux d'intérêt par période	RATE() Renvoie 0,00999999 (1 %) : RATE(24; 1000; 21243,39)
Nombre de périodes	NPER() Renvoie 24,00 : NPER(0,01; 1000; 21243,39)

### Formules de calcul des rentes

La formule permettant de calculer la **valeur actualisée** d'une rente ordinaire (paiement à la fin d'une période) :

La formule permettant de calculer la **valeur capitalisée** d'une rente ordinaire (paiement à la fin d'une période) :

# Fonction PPMT()

Renvoie le capital payé sur un emprunt pour une seule période.

## Syntaxe

```
PPMT(taux; période_spécifiée; périodes; quantité <; type>)
```

## Paramètres

Nom	Type	Description
<i>taux</i>	numérique	Taux d'intérêt par période.
<i>période_spécifiée</i>	numérique	Période pour laquelle il faut trouver le capital payé.
<i>périodes</i>	numérique	Nombre total de périodes de paiement.
<i>quantité</i>	numérique	Capital de l'emprunt.
<i>type</i> optionnel	numérique	Date des paiements : <ul style="list-style-type: none"> <li>○ 0 : paiement en fin de période</li> <li>○ 1 : paiement en début de période</li> </ul> Si vous ignorez ce paramètre, c'est la valeur par défaut de 0 qui est utilisée.

### Remarque

Vous devez utiliser des périodes temporelles cohérentes lorsque vous spécifiez *taux* et *périodes* pour vous assurer que vous spécifiez un taux d'intérêt **par période**.

Par exemple :

- dans le cas de paiements mensuels, sur un emprunt ou un investissement d'une durée de deux ans avec un taux d'intérêt de 5 % par an, spécifiez 0,05/12 pour *taux* et 2 \* 12 pour *périodes*
- dans le cas de paiements annuels sur le même prêt ou investissement, indiquez 0,05 pour *taux* et 2 pour *périodes*

## Sortie

Numérique.



# Exemples

## Exemples basiques

Renvoie 367,24, le capital payé lors du premier mois d'un emprunt de 275,000 \$ sur vingt-cinq ans à 6,5 % avec des paiements à la fin de chaque mois.

```
PPMT(0,065/12; 1; 12*25; 275000; 0)
```

Renvoie 1846,82 ; le capital payé sur le même emprunt lors du dernier mois de celui-ci :

```
PPMT(0,065/12; 300; 12*25; 275000; 0)
```

## Remarques

### Fonctions associées

La fonction IPMT ( ) est le complément de la fonction PPMT ( ).

La fonction CUMPRINC ( ) calcule le capital payé sur une plage de périodes.

# Fonction PROPER()

Renvoie une chaîne dans laquelle le premier caractère de chaque mot est en majuscule, tandis que les caractères restants sont définis en minuscules.

## Syntaxe

```
PROPER(chaîne_de_caractères)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	Valeur à convertir avec la bonne casse.

## Sortie

Caractère.

## Exemples

### Exemples basiques

Renvoie "John Doe" :

```
PROPER("JOHN DOE")
```

Renvoie "John Doe" :

```
PROPER("john doe")
```

Renvoie "1550 Alberni St." :

```
PROPER("1550 ALBERNI st.")
```

Renvoie "Bill O'Hara" :

```
PROPER("BILL O'HARA")
```

Renvoie toutes les valeurs du champ **Nom\_société** converties dans la casse appropriée :

```
PROPER(Nom_société)
```

## Remarques

### Fonctionnement

La fonction PROPER( ) convertit le premier caractère de *chaîne\_de\_caractères*, et tout caractère ultérieur précédé d'une espace, en majuscules.

Les caractères ultérieurs précédés d'un trait d'union, d'une apostrophe, d'un et commercial (&) ainsi que de plusieurs autres signes de ponctuation et caractères spéciaux sont également convertis en majuscules. Tous les autres caractères alphabétiques sont convertis en minuscules.

### Quand utiliser PROPER( )

Généralement, la fonction PROPER permet de convertir les noms stockés tout en majuscules ou tout en minuscules dans les données source dans le format de casse approprié afin que les noms s'affichent correctement dans un modèle de lettre ou un rapport.

# Fonction PROPERTIES()

Renvoie les informations de propriétés pour l'élément de projet Analytics spécifié.

## Syntaxe

```
PROPERTIES(nom; type_obj; type_info)
```

## Paramètres

Nom	Type	Description
<i>nom</i>	caractère	<p>Nom de l'élément de projet Analytics pour lequel vous souhaitez des informations. <i>nom</i> n'est pas sensible à la casse.</p> <p>Si l'élément de projet est une table Analytics, indiquez le nom du format de table, et non le nom du fichier de données. Par exemple : "Factures", et non "factures_janvier.fil"</p> <p>Si vous utilisez la fonction PROPERTIES() pour renvoyer le nom de la table active, spécifiez le <i>nom</i> "activetable".</p>
<i>type_obj</i>	caractère	<p>Le type d'élément de projet Analytics appelé <i>nom</i>.</p> <p><b>Remarque</b> À l'heure actuelle, "table" est le seul type d'élément de projet pris en charge.</p>
<i>type_info</i>	caractère	<p>Type d'information que vous souhaitez sur l'élément de projet Analytics</p> <p>Pour plus d'informations, consultez la section "Types d'informations de propriétés" Page 726.</p>

## Sortie

Caractère. La chaîne de sortie comporte 260 caractères au maximum. Si les informations de propriétés sont introuvables, une chaîne vide est renvoyée.

# Exemples

## Exemples basiques

### Informations sur le fichier de données Analytics (.fil)

Renvoie "Ap\_Trans.fil" :

```
PROPERTIES("Ap_Trans"; "table"; "filename")
```

Renvoie "C:\ACL DATA\Exemples de fichiers de données" :

```
PROPERTIES("Ap_Trans"; "table"; "filepath")
```

### Informations sur la table Analytics ouverte

Renvoie "Ap\_Trans" :

```
PROPERTIES("activetable"; "table"; "open")
```

### Informations sur une source de données externe

Renvoie "Trans\_Mai.xls" :

```
PROPERTIES("Trans_Mai"; "table"; "sourcename")
```

Renvoie "C:\Project Data\Factures mensuelles\_Excel" :

```
PROPERTIES("Trans_Mai"; "table"; "sourcepath")
```

Renvoie "EXCEL" :

```
PROPERTIES("Trans_Mai"; "table"; "sourcetype")
```

## Remarques

### Informations sur le fichier :

Les types d'informations fournissent des informations sur le fichier de données Analytics (.fil) associé à une table Analytics.

## Informations source

Les types d'informations commençant par « source » fournissent des informations sur les sources de données externes pouvant être associées à une table Analytics. Il est uniquement possible de créer des rapports à partir des sources de données externes prenant en charge l'actualisation d'une table Analytics à l'aide de la fonction `PROPERTIES()` :

- Microsoft Excel
- Microsoft Access
- Texte délimité
- Adobe Acrobat (PDF)
- Image d'impression (rapport)
- Format de fichier SAP privé / DART
- XML
- XBRL
- ODBC, sources de données

## Types d'informations de propriétés

Le tableau ci-dessous répertorie les types d'informations de propriétés pouvant être renvoyées par la fonction `PROPERTIES()`. Les tables Analytics sont les seuls éléments de projet Analytics pouvant être actuellement utilisés avec la fonction `PROPERTIES()` :

type_obj	type_info	Renvoi :
"table"	"filename"	Le nom du fichier de données associé à la table Analytics.
	"filepath"	Le chemin d'accès du fichier de données associé à la table Analytics.
	"filesize"	La taille, en Ko, du fichier de données associé à la table Analytics.
	"filemodifiedat"	L'heure et la date de la dernière modification du fichier de données associé à la table Analytics.
	"sourcename"	Le nom de la source de données associée à la table Analytics. Les sources de données peuvent être des fichiers externes comme des fichiers Excel, Access, PDF, XML ou des fichiers texte délimité, ou bien des sources de données ODBC.
	"sourcepath"	Le chemin d'accès de la source de données associée à la table Analytics. Non pris en charge pour les sources de données ODBC.
	"sourcetype"	Le type de la source de données associée à la table Analytics.
	"sourcesize"	La taille, en Ko, de la source de données associé à la table Analytics. Non pris en charge pour les sources de données ODBC.
	"sourcemodifiedat"	L'heure et la date de la dernière modification de la source de données associée à la table Analytics. Non pris en charge pour les sources de données ODBC.
	"open"	Le nom de la table Analytics actuellement active.  <div style="border-left: 2px solid blue; padding-left: 10px; margin-left: 20px;"> <p><b>Remarque</b></p> <p>Il est possible d'ouvrir plusieurs tables Analytics en même temps, mais seule une table à la fois peut être active dans l'interface utilisateur.</p> </div>

# Fonction PVANNUITY()

Renvoie la valeur actualisée d'une série de paiements futurs calculés à l'aide d'un taux d'intérêt constant. La valeur actualisée correspond à la valeur d'un montant forfaitaire en cours.

## Syntaxe

```
PVANNUITY(taux; périodes; règlement <;type>)
```

## Paramètres

Nom	Type	Description
<i>taux</i>	numérique	Taux d'intérêt par période.
<i>périodes</i>	numérique	Nombre total de périodes de paiement.
<i>règlement</i>	numérique	Paiement par période. Le montant du paiement doit demeurer constant pendant toute la durée de l'annuité.
<i>type</i> optionnel	numérique	Date des paiements : <ul style="list-style-type: none"> <li>○ 0 : paiement en fin de période</li> <li>○ 1 : paiement en début de période</li> </ul> Si vous ignorez ce paramètre, c'est la valeur par défaut de 0 qui est utilisée.

### Remarque

Vous devez utiliser des périodes temporelles cohérentes lorsque vous spécifiez *taux*, *périodes* et *paiement* pour vous assurer que vous spécifiez un taux d'intérêt **par période**.

Par exemple :

- dans le cas d'un *paiement* mensuel, sur un emprunt ou un investissement d'une durée de deux ans avec un taux d'intérêt de 5 % par an, spécifiez 0,05/12 pour *taux* et 2 \* 12 pour *périodes*
- dans le cas d'un *paiement* annuel sur le même prêt ou investissement, indiquez 0,05 pour *taux* et 2 pour *périodes*

## Sortie

Numérique. Le résultat est calculé avec deux décimales.



# Exemples

## Exemples basiques

### Paiements mensuels

Renvoie 21455,82, la valeur actualisée de 1 000 \$ payée au début de chaque mois pendant 2 ans au taux composé mensuel de 1 % par mois :

```
PVANNUIITY(0,01; 2*12; 1000; 1)
```

### Paiements annuels

Renvoie 20280,00, la valeur actualisée de 12 000 \$ payée à la fin de chaque année pendant 2 ans à un taux composé annuel de 12 % par an :

```
PVANNUIITY(0,12; 2; 12000; 0)
```

## Exemples avancés

### Calculs d'annuité

Les calculs d'annuité impliquent quatre variables :

- **valeur actualisée ou valeur capitalisée** : \$21243,39 et \$ 26973,46 dans les exemples ci-après
- **montant de paiement par période** : \$1000,00 dans les exemples ci-après
- **taux d'intérêt par période** : 1 % par mois dans les exemples ci-après
- **nombre de périodes** : 24 mois dans les exemples ci-après

Si vous connaissez la valeur de trois des variables, vous pouvez utiliser une fonction Analytics pour calculer la quatrième.

Je souhaite trouver :	Fonction Analytics à utiliser :
Valeur actualisée	PVANNUIITY() Renvoie 21243,39 : <pre>PVANNUIITY(0,01; 24; 1000)</pre>
Valeur capitalisée	FVANNUIITY() Renvoie 26973,46 :

Je souhaite trouver :	Fonction Analytics à utiliser :
	FVANNUIITY(0,01; 24; 1000)
Montant du paiement par période	PMT() Renvoie 1000 : PMT(0,01; 24; 21243,39)
Taux d'intérêt par période	RATE() Renvoie 0,00999999 (1 %) : RATE(24; 1000; 21243,39)
Nombre de périodes	NPER() Renvoie 24,00 : NPER(0,01; 1000; 21243,39)

### Formules de calcul des rentes

La formule permettant de calculer la **valeur actualisée** d'une rente ordinaire (paiement à la fin d'une période) :

La formule permettant de calculer la **valeur capitalisée** d'une rente ordinaire (paiement à la fin d'une période) :

## Remarques

### Fonctions associées

La fonction FVANNUIITY( ) est l'inverse de la fonction PVANNUIITY( ).

# Fonction PVLUMPSUM( )

Renvoie la valeur actualisée requise pour générer un montant forfaitaire futur spécifique calculé à l'aide d'un taux d'intérêt constant. La valeur actualisée correspond à la valeur d'un montant forfaitaire en cours.

## Syntaxe

```
PVLUMPSUM(taux; périodes; quantité)
```

## Paramètres

Nom	Type	Description
<i>taux</i>	numérique	Taux d'intérêt par période.
<i>périodes</i>	numérique	Nombre total de périodes.
<i>quantité</i>	numérique	La valeur du montant forfaitaire actualisé à la fin de la dernière période.

### Remarque

Vous devez utiliser des périodes temporelles cohérentes lorsque vous spécifiez *taux* et *périodes* pour vous assurer que vous spécifiez un taux d'intérêt **par période**.

Par exemple :

- dans le cas de paiements mensuels, sur un emprunt ou un investissement d'une durée de deux ans avec un taux d'intérêt de 5 % par an, spécifiez 0,05/12 pour *taux* et 2 \* 12 pour *périodes*
- dans le cas de paiements annuels sur le même prêt ou investissement, indiquez 0,05 pour *taux* et 2 pour *périodes*

## Sortie

Numérique. Le résultat est calculé avec deux décimales.

# Exemples

## Exemples basiques

### Intérêts composés mensuels

Renvoie 1000,00, le capital d'investissement initial nécessaire pour générer un montant forfaitaire capitalisé de 1 269,73 \$, lorsqu'il est investi pour 2 ans au taux d'intérêt composé mensuel de 1 % par mois :

```
PVLUMPSUM(0,01; 2*12; 1269,73)
```

Renvoie 787,57, le capital d'investissement initial nécessaire pour générer un montant forfaitaire capitalisé de 1 000 \$, lorsqu'il est investi pour 2 ans au taux d'intérêt composé mensuel de 1 % par mois :

```
PVLUMPSUM(0,01; 2*12; 1000)
```

Renvoie 21455,82, le capital d'investissement initial nécessaire pour générer un montant forfaitaire capitalisé de 27 243,20 \$, lorsqu'il est investi pour 2 ans au taux d'intérêt composé mensuel de 1 % par mois :

```
PVLUMPSUM(0,01; 2*12; 27243,20)
```

### Intérêts composés semestriels

Renvoie 792,09, le capital d'investissement initial nécessaire pour générer un montant forfaitaire capitalisé de 1 000 \$, lorsqu'il est investi pour 2 ans au taux d'intérêt composé semestriel de 12 % par an :

```
PVLUMPSUM(0,12/2; 2*2; 1000)
```

### Intérêts composés annuels

Renvoie 797,19, le capital d'investissement initial nécessaire pour générer un montant forfaitaire capitalisé de 1 000 \$, lorsqu'il est investi pour 2 ans au taux d'intérêt composé annuel de 12 % par an :

```
PVLUMPSUM(0,12; 2; 1000)
```

# Remarques

## Qu'est-ce-que la valeur actualisée ?

La valeur actualisée d'un montant forfaitaire est le capital initial nécessaire pour générer un montant forfaitaire spécifique capitalisé, dans un délai précis. La valeur capitalisée correspond au capital augmenté des intérêts composés cumulés.

## Fonctions associées

La fonction FVLUMPSUM( ) est l'inverse de la fonction PVLUMPSUM( ).

# Fonction PYDATE()

Renvoie une valeur date calculée par une fonction dans un script Python externe. Le traitement des données dans Python est externe à Analytics.

## Syntaxe

```
PYDATE("fichier py;fonction py" <; champ|valeur <;...n>>)
```

## Paramètres

Nom	Type	Description
<i>fichier py;fonction py</i>	caractère	<p>Le nom du script Python à exécuter suivi d'une virgule, puis le nom de la fonction renvoyant la valeur :</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;">"monScript;maFonction"</div> <p>Lorsque vous indiquez le script Python, ignorez l'extension du fichier. La fonction que vous appelez peut appeler d'autres fonctions dans le script ou à l'intérieur d'autres scripts ; toutefois, tous les scripts qui s'exécutent doivent être placés à l'intérieur d'un dossier dans la variable d'environnement système PYTHONPATH avant l'exécution.</p> <p>Pour plus d'informations, consultez la section "Configuration de Python pour être utilisé avec Analytics" Page 953.</p> <p><b>Remarque</b> Votre <i>fonction py</i> doit renvoyer un objet DateHeure.Date Python.</p>
<i>champ  valeur &lt;;...n&gt;</i> optionnel	caractère numérique DateHeure logique	<p>Cette liste de champs, d'expressions ou de valeurs littérales à utiliser comme arguments pour la fonction Python. Les valeurs sont transmises dans la fonction que vous appelez dans l'ordre dans lequel vous les spécifiez.</p> <p>Vous pouvez inclure autant d'arguments que nécessaires pour répondre à la définition de la fonction dans le script Python.</p> <p><b>Remarque</b> La fonction ALLTRIM() sert à supprimer les espaces de début ou de fin de l'entrée de caractères : ALLTRIM(str). Pour plus d'informations, consultez la section "Fonction ALLTRIM()" Page 493.</p>

# Sortie

DateHeure.

## Exemples

### Exemples basiques

Renvoie `20160630` :

```
PYDATE('hello,due_date'; `20160531`; 30)
```

Script Python externe qui accepte une date et un délai de grâce sous forme de nombres de jours et calcule la date d'échéance de la facture : Pour une date de facture au **2016-05-31** et une période de 30 jours : "2016-06-30"

```
#!/python
from datetime import timedelta

def due_date(inv_date, period):
    return(inv_date + timedelta(period))
```

### Exemples avancés

#### Définition d'un champ calculé

Définit un champ calculé dans la table `Trans_Cf` à l'aide du script Python qui calcule la date d'échéance :

```
OPEN Trans_Cf
DEFINE FIELD date_échéance COMPUTED
WIDTH 27
PYDATE("hello;due_date"; date_facture; période_paie)
```

# Fonction PYDATETIME()

Renvoie une valeur DateHeure calculée par une fonction dans un script Python externe. Le traitement des données dans Python est externe à Analytics.

## Syntaxe

```
PYDATETIME("fichier py;fonction py" <; champ|valeur <;...n>>)
```

## Paramètres

Nom	Type	Description
<i>fichier py;fonction py</i>	caractère	<p>Le nom du script Python à exécuter suivi d'une virgule, puis le nom de la fonction renvoyant la valeur :</p> <pre>"monScript;maFonction"</pre> <p>Lorsque vous indiquez le script Python, ignorez l'extension du fichier. La fonction que vous appelez peut appeler d'autres fonctions dans le script ou à l'intérieur d'autres scripts ; toutefois, tous les scripts qui s'exécutent doivent être placés à l'intérieur d'un dossier dans la variable d'environnement système PYTHONPATH avant l'exécution.</p> <p>Pour plus d'informations, consultez la section "Configuration de Python pour être utilisé avec Analytics" Page 953.</p> <p><b>Remarque</b> Votre <i>fonction py</i> doit renvoyer un objet DateHeure Python.</p>
<i>champ  valeur &lt;;...n&gt;</i> optionnel	caractère numérique DateHeure logique	<p>Cette liste de champs, d'expressions ou de valeurs littérales à utiliser comme arguments pour la fonction Python. Les valeurs sont transmises dans la fonction que vous appelez dans l'ordre dans lequel vous les spécifiez.</p> <p>Vous pouvez inclure autant d'arguments que nécessaires pour répondre à la définition de la fonction dans le script Python.</p> <p><b>Remarque</b> La fonction ALLTRIM() sert à supprimer les espaces de début ou de fin de l'entrée de caractères : ALLTRIM(str). Pour plus d'informations, consultez la section "Fonction ALLTRIM()" Page 493.</p>



# Sortie

DateHeure.

## Exemples

### Exemples basiques

Renvoie `20170101t0500` :

```
PYDATETIME("hello, combine_date_time"; `20170101`; `t0500`)
```

Le script Python externe accepte un argument date et un argument heure, puis renvoie un objet DateHeure combiné :

```
# hello.py content
from datetime import datetime

def combine_date_time(d,t):
    return datetime.combine(d,t)
```

### Exemples avancés

#### Ajout d'une heure à une DateHeure

Renvoie `20160101t2230` :

```
PYDATETIME("hello,add_time"; `20160101 150000`; `t073000`)
```

Script Python externe acceptant une DateHeure et une heure, puis qui ajoute l'heure à la DateHeure : 2016-01-01 15:00:00 + 7 heures, 30 minutes, 00 seconde = 2016-01-01 22:30:00.

```
# hello.py content
from datetime import timedelta
from datetime import datetime
from datetime import time
def add_time(start, time_to_add):
    return start + timedelta(hours=time_to_add.hour, minutes=time_to_add.minute, seconds=time_to_add.second)
```

# Fonction PYLOGICAL()

Renvoie une valeur logique calculée par une fonction dans un script Python externe. Le traitement des données dans Python est externe à Analytics.

## Syntaxe

```
PYLOGICAL("fichier py;fonction py" <; champ|valeur <;...n>>)
```

## Paramètres

Nom	Type	Description
<i>fichier py;fonction py</i>	caractère	<p>Le nom du script Python à exécuter suivi d'une virgule, puis le nom de la fonction renvoyant la valeur :</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;">"monScript;maFonction"</div> <p>Lorsque vous indiquez le script Python, ignorez l'extension du fichier. La fonction que vous appelez peut appeler d'autres fonctions dans le script ou à l'intérieur d'autres scripts ; toutefois, tous les scripts qui s'exécutent doivent être placés à l'intérieur d'un dossier dans la variable d'environnement système PYTHONPATH avant l'exécution.</p> <p>Pour plus d'informations, consultez la section "Configuration de Python pour être utilisé avec Analytics" Page 953.</p> <p><b>Remarque</b> Votre <i>fonction py</i> doit renvoyer une valeur de vérité Python.</p>
<i>champ  valeur &lt;;...n&gt;</i> optionnel	caractère numérique DateHeure logique	<p>Cette liste de champs, d'expressions ou de valeurs littérales à utiliser comme arguments pour la fonction Python. Les valeurs sont transmises dans la fonction que vous appelez dans l'ordre dans lequel vous les spécifiez.</p> <p>Vous pouvez inclure autant d'arguments que nécessaires pour répondre à la définition de la fonction dans le script Python.</p> <p><b>Remarque</b> La fonction ALLTRIM() sert à supprimer les espaces de début ou de fin de l'entrée de caractères : ALLTRIM(str). Pour plus d'informations, consultez la section "Fonction ALLTRIM()" Page 493.</p>

# Sortie

Logique.

## Exemples

### Exemples basiques

Renvoie F :

```
PYLOGICAL( "hello,str_compare"; "basketball"; "baseball"; "b" )
```

Script Python externe qui compare *str1* et *str2* à l'aide du nombre de caractères transmis sous forme de *char* :

```
# hello.py content
def str_compare(str1, str2, char):
    return str1.count(char) > str2.count(char)
```

### Exemples avancés

#### Utilisation des champs

Renvoie une valeur de vérité après la comparaison de *Nom\_Fournisseur* et *Ville\_Fournisseur* :

```
PYLOGICAL( "hello;str_compare"; Nom_Fournisseur; Ville_Fournisseur; "b" )
```

Script Python externe qui compare *str1* et *str2* à l'aide du nombre de caractères transmis sous forme de *char* :

```
# hello.py content
def str_compare(str1, str2, char):
    return str1.count(char) > str2.count(char)
```

# Fonction PYNUMERIC()

Renvoie une valeur numérique calculée par une fonction dans un script Python externe. Le traitement des données dans Python est externe à Analytics.

## Syntaxe

```
PYNUMERIC(fichier py;fonction py, décimale <; champ|valeur <;...n>>)
```

## Paramètres

Nom	Type	Description
<i>fichier py;fonction py</i>	caractère	<p>Le nom du script Python à exécuter suivi d'une virgule, puis le nom de la fonction renvoyant la valeur :</p> <pre>"monScript;maFonction"</pre> <p>Lorsque vous indiquez le script Python, ignorez l'extension du fichier. La fonction que vous appelez peut appeler d'autres fonctions dans le script ou à l'intérieur d'autres scripts ; toutefois, tous les scripts qui s'exécutent doivent être placés à l'intérieur d'un dossier dans la variable d'environnement système PYTHONPATH avant l'exécution.</p> <p>Pour plus d'informations, consultez la section "Configuration de Python pour être utilisé avec Analytics" Page 953.</p> <p><b>Remarque</b> Votre <i>fonction py</i> doit renvoyer un type numérique Python.</p>
<i>décimale</i>	numérique	<p>Nombre de décimales à inclure dans la valeur renvoyée. Doit être un nombre entier positif.</p>
<i>champ  valeur &lt;;...n&gt;</i> optionnel	caractère numérique DateHeure logique	<p>Cette liste de champs, d'expressions ou de valeurs littérales à utiliser comme arguments pour la fonction Python. Les valeurs sont transmises dans la fonction que vous appelez dans l'ordre dans lequel vous les spécifiez.</p> <p>Vous pouvez inclure autant d'arguments que nécessaires pour répondre à la définition de la fonction dans le script Python.</p> <p><b>Remarque</b> La fonction ALLTRIM() sert à supprimer les espaces de début ou de fin de l'entrée de caractères : ALLTRIM(str). Pour plus d'informations, consultez la section "Fonction ALLTRIM()" Page 493.</p>

# Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 35,00 :

```
PYNUMERIC("hello;get_nth_percent"; 2; 80; 120; 30; 45; 30; 100; 35; 45)
```

Script Python externe qui renvoie la valeur au centile demandé à partir d'une liste de valeurs à taille dynamique :

```
# hello.py content
from math import ceil
def get_nth_percent(percentage, *values):
    input_length = len(values)
    position = ceil((percentage/100.00) * input_length)
    return values[position-1]
```

# Fonction PYSTRING()

Renvoie une valeur de type caractère calculée par une fonction dans un script Python externe. Le traitement des données dans Python est externe à Analytics.

## Syntaxe

```
PYSTRING("fichier py;fonction py"; longueur <;champ|valeur <;...n>>)
```

Nom	Type	Description
<i>fichier py;fonction py</i>	caractère	<p>Le nom du script Python à exécuter suivi d'une virgule, puis le nom de la fonction renvoyant la valeur :</p> <pre>"monScript;maFonction"</pre> <p>Lorsque vous indiquez le script Python, ignorez l'extension du fichier. La fonction que vous appelez peut appeler d'autres fonctions dans le script ou à l'intérieur d'autres scripts ; toutefois, tous les scripts qui s'exécutent doivent être placés à l'intérieur d'un dossier dans la variable d'environnement système PYTHONPATH avant l'exécution.</p> <p>Pour plus d'informations, consultez la section "Configuration de Python pour être utilisé avec Analytics" Page 953.</p> <p><b>Remarque</b> Votre <i>fonction py</i> doit renvoyer un objet chaîne Python.</p>
<i>longueur</i>	numérique	La longueur à allouer pour la chaîne de retour.
<i>champ  valeur &lt;;...n&gt;</i> optionnel	caractère numérique DateHeure logique	<p>Cette liste de champs, d'expressions ou de valeurs littérales à utiliser comme arguments pour la fonction Python. Les valeurs sont transmises dans la fonction que vous appelez dans l'ordre dans lequel vous les spécifiez.</p> <p>Vous pouvez inclure autant d'arguments que nécessaires pour répondre à la définition de la fonction dans le script Python.</p> <p><b>Remarque</b> La fonction ALLTRIM() sert à supprimer les espaces de début ou de fin de l'entrée de caractères : ALLTRIM(str). Pour plus d'informations, consultez la section "Fonction ALLTRIM()" Page 493.</p>

# Sortie

Caractère.

## Exemples

### Exemples basiques

Renvoie "my test" :

```
PYSTRING('hello,main'; 20; "my")
```

Script Python externe qui accepte une chaîne et concatène " test" à la chaîne :

```
#!/python
# hello.py content
def main(str):
    str2 = str + ' test'
    return(str2)
```

### Exemples avancés

#### Renvoi d'une sous-chaîne

Cet exemple supprime les deux derniers caractères du champ Nom du fournisseur et renvoie la sous-chaîne :

```
PYSTRING( "hello;sub_set"; LENGTH(Nom_Fournisseur); ALLTRIM(Nom_Fournisseur); LENGTH
(ALLTRIM(Nom_Fournisseur)); 0; LENGTH(ALLTRIM(Nom_Fournisseur)) - 2)
```

Script Python externe qui accepte une chaîne de caractères, une longueur de chaîne et deux positions de caractères. La fonction renvoie une sous-chaîne entre la position un et la position deux :

```
#!/hello.py content
def sub_set(str, length, p1, p2):
    if p1 >= 0 and p2 < length and p1 < p2:
        str2 = str[p1:p2]
    else:
        str2 = str
    return str2
```

# Fonction PYTIME( )

Renvoie une valeur heure calculée par une fonction dans un script Python externe. Le traitement des données dans Python est externe à Analytics.

## Syntaxe

```
PYTIME("fichier py;fonction py" <; champ|valeur <;...n>>)
```

## Paramètres

Nom	Type	Description
<i>fichier py;fonction py</i>	caractère	<p>Le nom du script Python à exécuter suivi d'une virgule, puis le nom de la fonction renvoyant la valeur :</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;">"monScript;maFonction"</div> <p>Lorsque vous indiquez le script Python, ignorez l'extension du fichier. La fonction que vous appelez peut appeler d'autres fonctions dans le script ou à l'intérieur d'autres scripts ; toutefois, tous les scripts qui s'exécutent doivent être placés à l'intérieur d'un dossier dans la variable d'environnement système PYTHONPATH avant l'exécution.</p> <p>Pour plus d'informations, consultez la section "Configuration de Python pour être utilisé avec Analytics" Page 953.</p> <p><b>Remarque</b> Votre <i>fonction py</i> doit renvoyer un objet DateHeure.Heure Python.</p>
<i>champ  valeur &lt;;...n&gt;</i> optionnel	caractère numérique DateHeure logique	<p>Cette liste de champs, d'expressions ou de valeurs littérales à utiliser comme arguments pour la fonction Python. Les valeurs sont transmises dans la fonction que vous appelez dans l'ordre dans lequel vous les spécifiez.</p> <p>Vous pouvez inclure autant d'arguments que nécessaires pour répondre à la définition de la fonction dans le script Python.</p> <p><b>Remarque</b> La fonction ALLTRIM() sert à supprimer les espaces de début ou de fin de l'entrée de caractères : ALLTRIM(str). Pour plus d'informations, consultez la section "Fonction ALLTRIM()" Page 493.</p>



# Sortie

DateHeure.

## Exemples

### Exemples basiques

Renvoie `t2122` :

```
ASSIGN v_time_part = PYTIME("hello;get_time"; `20160101 212223`)
```

Script Python externe :

```
# hello.py content
from datetime import time
from datetime import date

def get_time(timestamp):
    return timestamp.time();
```

# Fonction RAND( )

Renvoie un nombre aléatoire situé dans les limites spécifiées.

## Syntaxe

```
RAND(nombre)
```

## Paramètres

Nom	Type	Description
<i>nombre</i>	numérique	<p>La limite numérique du nombre aléatoire.</p> <p>Si vous indiquez un nombre avec des décimales, le nombre aléatoire généré a le même nombre de décimales.</p> <ul style="list-style-type: none"><li>◦ <b>Si vous saisissez un nombre positif</b> : le nombre aléatoire renvoyé est supérieur ou égal à zéro et inférieur au nombre que vous avez spécifié.</li></ul> <p>Renvoie un nombre compris entre 0 et 99 :</p> <pre>RAND(100)</pre> <ul style="list-style-type: none"><li>◦ <b>Si vous saisissez un nombre négatif</b> : le nombre aléatoire renvoyé est inférieur à zéro et supérieur ou égal au nombre que vous avez spécifié.</li></ul> <p>Renvoie un nombre compris entre -1 et -100 :</p> <pre>RAND(-100)</pre>

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 278,61 :

```
RAND(1000,00)
```

Renvoie 3781 :

```
RAND(10000)
```

### Remarque

La valeur renvoyée différera avec chaque exécution de la fonction.

## Remarques

### RAND( ) ne peut pas répliquer les résultats

Si vous utilisez la fonction RAND( ) deux fois de suite avec la même valeur *nombre*, vous obtiendrez des résultats différents. Contrairement à la commande RANDOM, la fonction RAND( ) n'a pas de valeur de départ.

### Nombres aléatoires en doublon possibles

Si vous utilisez RAND( ) pour créer un champ calculé qui attribue un nombre aléatoire à chaque enregistrement dans une table, il est possible que des nombres aléatoires en doublon soient générés. Il n'y a aucune garantie que les nombres aléatoires soient uniques.

Plus la valeur *nombre* est grande par rapport au nombre d'enregistrements dans la table, plus la chance que les nombres générés soient uniques est grande.

### Mise à jour dynamique des nombres aléatoires

Un champ calculé avec la fonction RAND( ) génère un nouveau jeu de nombres aléatoires chaque fois que vous effectuez des actions comme le tri express, l'application d'un filtre, le réarrangement des colonnes ou le défilement de la vue.

Si vous souhaitez fixer un ensemble de nombres aléatoires, extrayez les données vers une nouvelle table à l'aide de l'option **Vue** ou **Champs** située dans la boîte de dialogue **Extraire**.

# Fonction RATE()

Renvoie le taux d'intérêt par période.

## Syntaxe

```
RATE(périodes; règlement; quantité)
```

## Paramètres

Nom	Type	Description
<i>périodes</i>	numérique	Nombre total de périodes de paiement.
<i>règlement</i>	numérique	Paiement par période.
<i>quantité</i>	numérique	Capital de l'emprunt.

### Remarque

La fonction RATE() part du principe que les paiements sont effectués à la fin de chaque période.

## Sortie

Numérique. Le taux est calculé jusqu'à la huitième décimale.

## Exemples

### Exemples basiques

Renvoie 0,00541667 (0,54 %), le taux d'intérêt mensuel découlant d'un emprunt de 275 000 \$ sur vingt-cinq ans avec des paiements mensuels de 1 856,82 \$ :

```
RATE(12*25; 1856,82; 275000)
```

Renvoie 0,06500004 (6,5 %), le taux d'intérêt annuel découlant du même emprunt :

```
RATE(12*25; 1856,82; 275000)*12
```

## Exemples avancés

### Conversion du taux nominal en taux effectif

La fonction RATE() calcule le taux d'intérêt nominal. Vous pouvez vous servir de la fonction EFFECTIVE() pour convertir le résultat de RATE() au taux d'intérêt effectif.

Renvoie 0,06715155 (6,7 %), le taux d'intérêt effectif annuel découlant du même emprunt dans les exemples ci-dessus :

```
EFFECTIVE((RATE(12*25; 1856,82; 275000)*12); 12*25)
```

### Calculs d'annuité

Les calculs d'annuité impliquent quatre variables :

- **valeur actualisée ou valeur capitalisée** : \$21243,39 et \$ 26973,46 dans les exemples ci-après
- **montant de paiement par période** : \$1000,00 dans les exemples ci-après
- **taux d'intérêt par période** : 1 % par mois dans les exemples ci-après
- **nombre de périodes** : 24 mois dans les exemples ci-après

Si vous connaissez la valeur de trois des variables, vous pouvez utiliser une fonction Analytics pour calculer la quatrième.

Je souhaite trouver :	Fonction Analytics à utiliser :
Valeur actualisée	PVANNUITY() Renvoie 21243,39 : <pre>PVANNUITY(0,01; 24; 1000)</pre>
Valeur capitalisée	FVANNUITY() Renvoie 26973,46 : <pre>FVANNUITY(0,01; 24; 1000)</pre>
Montant du paiement par période	PMT() Renvoie 1000 : <pre>PMT(0,01; 24; 21243,39)</pre>
Taux d'intérêt par période	RATE() Renvoie 0,00999999 (1 %) :

Je souhaite trouver :	Fonction Analytics à utiliser :
	<div style="border: 1px solid black; padding: 5px; width: fit-content;">RATE(24; 1000; 21243,39)</div>
Nombre de périodes	NPER() Renvoie 24,00 : <div style="border: 1px solid black; padding: 5px; width: fit-content;">NPER(0,01; 1000; 21243,39)</div>

### Formules de calcul des rentes

La formule permettant de calculer la **valeur actualisée** d'une rente ordinaire (paiement à la fin d'une période) :

La formule permettant de calculer la **valeur capitalisée** d'une rente ordinaire (paiement à la fin d'une période) :

# Fonction RDATE()

Renvoie une valeur date calculée par un script ou une fonction de Rvachev. Le traitement des données dans Rvachev est externe à Analytics.

## Syntaxe

```
RDATE(rScript|rCode <;champ/valeur <;...n>>)
```

## Paramètres

Nom	Type	Description
<i>rScript</i>   <i>rCode</i>	caractère	<p>Le chemin d'accès complet ou relatif au script R ou à un snippet de code R à exécuter.</p> <p>Si vous saisissez le code R directement plutôt que d'utiliser un fichier externe, vous ne pouvez pas utiliser le caractère de guillemet fermant, même si vous l'échappez :</p> <ul style="list-style-type: none"> <li>o <b>valide</b> : 'var &lt;- "\test"'</li> <li>o <b>non valide</b> : 'var &lt;- "\test"'</li> </ul>
<i>champ</i>   <i>valeur</i> <;... <i>n</i> > optionnel	caractère numérique DateHeure logique	<p>La liste des champs, des expressions ou des valeurs littérales à utiliser comme arguments pour le snippet de code ou le script R.</p> <p>Les valeurs sont transmises dans la fonction que vous appelez dans l'ordre dans lequel vous les indiquez, puis vous y référez à l'aide de valeur1; valeur2 ... valeurN dans le code R.</p> <p>Vous pouvez inclure autant d'arguments que nécessaires pour répondre à la définition de la fonction dans le code R.</p> <p><b>Remarque</b></p> <p>La fonction ALLTRIM() sert à supprimer les espaces de début ou de fin de l'entrée de caractères : ALLTRIM(str). Pour plus d'informations, consultez la section "Fonction ALLTRIM()" Page 493.</p>

## Sortie

DateHeure.

# Exemples

## Exemples basiques

Renvoie « 20160530 » :

```
RDATE("as.Date(valeur1,'%m-%d-%Y'); "05-30-16")
```

## Exemples avancés

### Utilisation d'un script R externe

Convertit une chaîne de caractères en date, puis la renvoie :

```
RDATE("a<-source('c:\\scripts\\r_scripts\\sample.r');a[[1]]"; dateText)
```

Script R externe (sample.r) :

```
dateForm <- function(dateText) {
  return(as.Date(dateText,format='%y%m%d'))
}
dateForm(value1)
```

## Remarques

### Renvoi de données à partir de R

Lorsque vous appelez des scripts R, utilisez la fonction `source` et affectez l'objet renvoyé à une variable. Vous pouvez alors accéder à la valeur renvoyée depuis votre fonction R à partir de l'objet renvoyé :

```
# 'a' contient l'objet de réponse et a[[1]] accède à la valeur des données
"a<-source('c:\\scripts\\r_scripts\\sample.r');a[[1]]"
```

### Fichier trace R

Analytics enregistre les messages en langage R dans un fichier `aclrlang.log` dans le dossier de projet. Utilisez ce fichier trace pour le débogage des erreurs R



### Astuce

Ce fichier trace se trouve dans le dossier de résultats des tâches analytiques d'Analytics Exchange.

## Exécution de scripts R externes sur AX Serveur

Si vous rédigez une App Analyse à exécuter sur AX Serveur et que vous souhaitez travailler sur des scripts R externes :

1. Chargez le fichier sous forme de fichier associé à l'App Analyse.
2. Utilisez la balise d'analyse FILE pour identifier le(s) fichier(s).
3. Faites référence au(x) fichier(s) à l'aide du chemin d'accès relatif `./nom_fichier.r`.

### Remarque

L'utilisation d'un fichier associé garantit que le compte du serveur d'application TomEE dispose des autorisations suffisantes pour accéder au fichier lors de l'exécution de R avec Analytics Exchange.

# Fonction RDATETIME()

Renvoie une valeur DateHeure calculée par un script ou une fonction de Rvachev. Le traitement des données dans Rvachev est externe à Analytics.

## Syntaxe

```
RDATETIME(rScript|rCode <;champ|valeur <;...n>>)
```

## Paramètres

Nom	Type	Description
<i>rScript</i>   <i>rCode</i>	caractère	<p>Le chemin d'accès complet ou relatif au script R ou à un snippet de code R à exécuter.</p> <p>Si vous saisissez le code R directement plutôt que d'utiliser un fichier externe, vous ne pouvez pas utiliser le caractère de guillemet fermant, même si vous l'échappez :</p> <ul style="list-style-type: none"> <li>o <b>valide</b> : 'var &lt;- "\test\''</li> <li>o <b>non valide</b> : 'var &lt;- "\test''</li> </ul>
<i>champ</i>   <i>valeur</i> <;... <i>n</i> > optionnel	caractère numérique DateHeure logique	<p>La liste des champs, des expressions ou des valeurs littérales à utiliser comme arguments pour le snippet de code ou le script R.</p> <p>Les valeurs sont transmises dans la fonction que vous appelez dans l'ordre dans lequel vous les indiquez, puis vous y référez à l'aide de valeur1; valeur2 ... valeurN dans le code R.</p> <p>Vous pouvez inclure autant d'arguments que nécessaires pour répondre à la définition de la fonction dans le code R.</p> <p><b>Remarque</b></p> <p>La fonction ALLTRIM() sert à supprimer les espaces de début ou de fin de l'entrée de caractères : ALLTRIM(str). Pour plus d'informations, consultez la section "Fonction ALLTRIM()" Page 493.</p>

## Sortie

DateHeure.

# Exemples

## Exemples basiques

Ajoute 45 minutes à la date et à l'heure actuelles :

```
RDATETIME("Sys.time() + valeur1";2700)
```

## Exemples avancés

### Utilisation d'un script R externe

Ajoute 45 minutes à un champ DateHeure en transmettant un champ et une valeur littérale à une fonction R externe :

```
RDATETIME("a<-'c:\\scripts\\sample.r'); a[[1]]"; start_date; 2700)
```

Script R externe (sample.r) :

```
add_time <- function(start, sec) {  
  return(start + sec)  
}  
add_time(value1, value2)
```

## Remarques

### Renvoi de données à partir de R

Lorsque vous appelez des scripts R, utilisez la fonction `source` et affectez l'objet renvoyé à une variable. Vous pouvez alors accéder à la valeur renvoyée depuis votre fonction R à partir de l'objet renvoyé :

```
# 'a' contient l'objet de réponse et a[[1]] accède à la valeur des données  
"a<-source('c:\\scripts\\r_scripts\\sample.r');a[[1]]"
```

### Fichier trace R

Analytics enregistre les messages en langage R dans un fichier `aclr_lang.log` dans le dossier de projet. Utilisez ce fichier trace pour le débogage des erreurs R

### Astuce

Ce fichier trace se trouve dans le dossier de résultats des tâches analytiques d'Analytics Exchange.

## Exécution de scripts R externes sur AX Serveur

Si vous rédigez une App Analyse à exécuter sur AX Serveur et que vous souhaitez travailler sur des scripts R externes :

1. Chargez le fichier sous forme de fichier associé à l'App Analyse.
2. Utilisez la balise d'analyse FILE pour identifier le(s) fichier(s).
3. Faites référence au(x) fichier(s) à l'aide du chemin d'accès relatif `./nom_fichier.r`.

### Remarque

L'utilisation d'un fichier associé garantit que le compte du serveur d'application TomEE dispose des autorisations suffisantes pour accéder au fichier lors de l'exécution de R avec Analytics Exchange.

## Fuseau horaire du système

L'heure moyenne de Greenwich (GMT) est le fuseau horaire actif par défaut dans l'environnement R utilisé par Analytics.

# Fonction RECLLEN( )

Renvoie la longueur de l'enregistrement en cours.

## Syntaxe

```
RECLLEN( )
```

## Paramètres

Cette fonction ne comporte pas de paramètres.

## Sortie

Numérique.

## Exemples

### Exemples basiques

Dans l'exemple suivant, tous les enregistrements dont la longueur est exactement 110 sont extraits :

```
EXTRACT RECORD IF RECLLEN( ) = 110 TO "Extrait.fil"
```

## Remarques

La fonction RECLLEN( ) permet d'identifier les enregistrements d'une longueur particulière, ou de rechercher les enregistrements plus courts que prévu. Cela est utile si vous travaillez avec des fichiers image d'impression (rapport). En effet, cette fonction fournit un moyen facile d'examiner les longueurs d'enregistrement :

- Pour les enregistrements de longueur fixe, la valeur renvoyée est une constante (la longueur d'enregistrement).
- Pour ceux de longueur variable, la longueur renvoyée varie d'un enregistrement à l'autre.

# Fonction RECNO( )

Renvoie le numéro de l'enregistrement actuel.

## Syntaxe

```
RECNO( )
```

## Paramètres

Cette fonction ne comporte pas de paramètres.

## Sortie

Numérique.

## Exemples

### Exemples basiques

Dans les exemples suivants, les enregistrements numérotés de 10 à 20 sont extraits vers une nouvelle table Analytics :

```
EXTRACT RECORD IF BETWEEN(RECNO( );10;20) TO "Sous-ensemble.fil"
```

## Remarques

La fonction RECNO( ) permet d'exporter les numéros des enregistrements dans une table ou de déterminer l'emplacement relatif d'un enregistrement particulier dans une table.

### Tables indexées et tables non indexées

Cette fonction renvoie le numéro de l'enregistrement logique actuel :

- Si la table n'est pas indexée, RECNO( ) démarre à 1 et augmente d'une unité pour chaque enregistrement dans la table. Les numéros d'enregistrements logiques et physiques sont identiques.

- Si la table est indexée, RECNO( ) se comporte de manière similaire, à ceci près qu'il compte les enregistrements par ordre logique, et non physique.

## Utilisation de la commande SEEK ou FIND

Si les commandes SEEK ou FIND sont utilisées, le numéro de l'enregistrement est réinitialisé à 1 après l'exécution de ces commandes.

## Réordonnement des enregistrements

Réordonner les enregistrements dans une table n'entraîne pas de réarrangement des numéros d'enregistrement générés par RECNO( ). Si vous souhaitez garder les numéros d'enregistrement avec les enregistrements auxquels ils ont été associés à l'origine, vous devez extraire les données vers une nouvelle table à l'aide de l'option **Champs** avant de réordonner les enregistrements.

# Fonction RECOFFSET()

Renvoie une valeur de champ à partir d'un enregistrement qui correspond à un nombre défini d'enregistrements issus de l'enregistrement actuel.

## Syntaxe

```
RECOFFSET(champ; nombre_d'enregistrements)
```

## Paramètres

Nom	Type	Description
<i>champ</i>	caractère numérique DateHeure	Nom du champ dans lequel il faut récupérer la valeur.
<i>nombre_d'enregistrements</i>	numérique	Nombre d'enregistrements issus de l'enregistrement actuel. Un nombre positif indique qu'il existe un enregistrement après l'enregistrement actuel, tandis qu'un nombre négatif indique qu'il existe un enregistrement antérieur à l'enregistrement actuel.

## Sortie

Caractère, Numérique ou DateHeure. La valeur renvoyée appartient à la même catégorie de données que le paramètre d'entrée *champ*.

## Exemples

### Exemples basiques

Renvoie une valeur *Montant* à partir de l'enregistrement suivant :

```
RECOFFSET(Montant;1)
```

Renvoie une valeur *Montant* à partir de l'enregistrement précédent :



```
RECOFFSET(Montant; -1)
```

## Exemples avancés

### Utilisation de RECOFFSET dans un champ calculé

Le champ calculé *Champ\_Suivant* affiche la valeur du champ Montant dans l'enregistrement suivant uniquement si ce dernier possède le même numéro de client.

Pour définir ce champ calculé dans un script, utilisez la syntaxe suivante :

```
DEFINE FIELD Montant_Suivant COMPUTED
RECOFFSET(Montant;1) IF RECOFFSET(Client;1) = Client
0
```

*Montant\_Suivant* correspond à la valeur du champ Montant de l'enregistrement suivant uniquement si le numéro client de l'enregistrement suivant est le même que celui de l'enregistrement actuel. Sinon, la valeur zéro est attribuée à *Montant\_Suivant*.

## Remarques

La fonction RECOFFSET( ) renvoie une valeur de champ issue d'un enregistrement et qui correspond à un nombre spécifique d'enregistrements antérieurs ou postérieurs à l'enregistrement actuel.

### Quand utiliser la fonction RECOFFSET( )

Cette fonction est généralement utilisée pour entreprendre des tests de comparaison avancés.

Cette fonction permet de comparer des valeurs de champ se trouvant dans l'enregistrement actuel avec un champ issu d'un autre enregistrement. Vous pouvez, par exemple, ajouter un champ calculé, chargé de calculer l'écart entre un montant de l'enregistrement actuel et un montant de l'enregistrement précédent.

### Début ou fin d'une table

Si elle arrive à la fin ou au début de la table, la fonction renvoie la valeur zéro pour les champs numériques, une chaîne vide pour les champs de type caractère ou 1900/01/01 pour les champs de date. RECOFFSET( ) renvoie alors une sortie vide parce qu'il n'existe aucun enregistrement postérieur auquel comparer l'enregistrement actuel.

# Fonction REGEXFIND( )

Renvoie une valeur logique indiquant si le modèle spécifié par une expression ordinaire se produit dans une chaîne de caractères.

## Syntaxe

```
REGEXFIND(chaîne_de_caractères; modèle)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	Le champ, l'expression ou la valeur littérale à tester pour un modèle correspondant.
<i>modèle</i>	caractère	La chaîne de modèle (expression ordinaire) à rechercher. <i>modèle</i> peut contenir des caractères littéraux, des métacaractères ou une association des deux. Les caractères littéraux incluent tous les caractères alphanumériques, certains caractères de ponctuation et des caractères vides. La recherche respecte la casse, ce qui signifie que les caractères alphanumériques en majuscules et en minuscules doivent être explicitement spécifiés.

## Sortie

Logique. Renvoie T (« true », vrai) si la valeur *modèle* indiquée est trouvée ; renvoie F (« false », faux) dans les autres cas.

## Exemples

### Exemples basiques

#### Modèles de caractères alphabétiques

Renvoie T pour tous les enregistrements dont le champ **Ville\_fournisseur** contient « Phoenix », « Austin » ou « Los Angeles ». Renvoie F autrement :

```
REGEXFIND(Ville_fournisseur; "Phoenix|Austin|Los Angeles")
```

Renvoie T pour tous les noms qui commencent par « John » ou par « Jon ». Par exemple : John, Jon, Johnson, Johnston, Jonson, Jonston, Jones, etc. Renvoie F autrement :

```
REGEXFIND(Nom;"^Joh?n")
```

Renvoie T pour uniquement les noms qui sont « John » ou « Jon ». Renvoie F autrement :

```
REGEXFIND(Nom;"^Joh?n\b")
```

## Modèles de caractères numériques

Renvoie « T » pour tous les enregistrements dont les numéros de facture contiennent « 98 » : Renvoie F autrement :

```
REGEXFIND(No_Facture; "98")
```

Renvoie T pour tous les enregistrements dont les numéros de facture commencent par « 98 ». Renvoie F autrement :

```
REGEXFIND(No_Facture; "\b98")
```

Renvoie T pour tous les enregistrements dont les numéros de facture se terminent par « 98 ». Renvoie F autrement :

```
REGEXFIND(No_Facture; "98\b")
```

Renvoie T pour tous les enregistrements dont les numéros de facture contiennent « 98 » aux 5e et 6e places. Renvoie F autrement :

```
REGEXFIND(No_Facture; "\b\d\d\d\d98")
```

```
REGEXFIND(No_Facture; "\b\d{4}98")
```

## Modèles de caractères mélangés

Renvoie T pour tous les enregistrements dont les codes produits commencent par 3 chiffres, suivis d'un trait d'union et de 6 lettres. Renvoie F autrement :

```
REGEXFIND(Code_Produit; "\b\d{3}-[a-zA-Z]{6}\b")
```

Renvoie T pour tous les enregistrements dont les codes produits commencent par 3 chiffres, suivis d'un trait d'union et d'au moins 6 lettres. Renvoie F autrement :

```
REGEXFIND(Code_Produit; "\bd{3,}-[a-zA-Z]{6}")
```

Renvoie T pour tous les enregistrements dont les identificateurs de factures alphanumériques contiennent « 98 » aux 5e et 6e places. Renvoie F autrement :

```
REGEXFIND(No_Facture; "\bw{4}98")
```

Renvoie T pour tous les enregistrements dont les identificateurs de factures contiennent les deux éléments suivants, renvoie F dans le cas contraire :

- n'importe quel caractère dans les quatre premières places
- « 98 » aux 5e et 6e places

```
REGEXFIND(No_Facture; "\b.{4}98")
```

Renvoie T pour tous les enregistrements dont les identificateurs de factures contiennent « 98 » précédé de 1 à 4 caractères de début. Renvoie F autrement :

```
REGEXFIND(No_Facture; "\b.{1,4}98")
```

Renvoie « T » pour tous les enregistrements dont les identificateurs de factures contiennent tous les éléments suivants, renvoie F dans le cas contraire :

- n'importe quel caractère dans les trois premières places
- « 5 » ou « 6 » à la 4e place et
- « 98 » aux 5e et 6e places

```
REGEXFIND(No_Facture; "\b.{3}[56]98")
```

Renvoie « T » pour tous les enregistrements dont les identificateurs de factures contiennent tous les éléments suivants, renvoie F dans le cas contraire :

- n'importe quel caractère dans les deux premières places
- « 55 » ou « 56 » aux 3e et 4e places
- « 98 » aux 5e et 6e places

```
REGEXFIND(No_Facture; "\b.{2}(55|56)98")
```

# Remarques

## Fonctionnement

La fonction REGEXFIND( ) utilise une expression ordinaire pour rechercher des données dans Analytics.

Les expressions ordinaires sont des chaînes de caractères de recherche puissantes et souples qui associent caractères littéraux et métacaractères, qui sont des caractères spéciaux effectuant une grande variété d'opérations de recherche.

Par exemple :

```
REGEXFIND(Nom;"Sm(i|y)the{0,1}")
```

utilise les métacaractères de groupe ( ), de changement | et de quantificateur { } pour créer une expression ordinaire qui trouve « Smith », « Smyth », « Smithe » ou « Smythe » dans le champ **Nom**.

## Correspondances séquentielles

La correspondance entre les valeurs *chaîne\_de\_caractères* et *modèle* s'effectue de manière séquentielle. Dans l'exemple précédent :

- "S" a une correspondance en première position dans le champ **Nom**
- "m" a une correspondance en deuxième position
- "i" et "y" ont une correspondance en troisième position
- "t" a une correspondance en quatrième position
- "h" a une correspondance en cinquième position
- "e" a une correspondance en sixième position, si une sixième position existe dans la valeur source

## Quand utiliser la fonction REGEXFIND( )

Utilisez la fonction REGEXFIND( ) pour rechercher des données à l'aide de mise en correspondance de modèles simples ou complexes.

La construction d'expressions ordinaires peut s'avérer être une tâche délicate, notamment si vous débutez dans la syntaxe. Il se peut que vous atteigniez vos objectifs de recherche à l'aide de fonctions de recherche Analytics plus simples comme FIND( ), MATCH( ) ou MAP( ).

Si vos critères de recherche dépassent les capacités de ces fonctions plus simples, les expressions ordinaires vous donnent presque une flexibilité illimitée dans la construction de chaînes de recherche.

## Comment REGEXFIND( ) gère les espaces

Les espaces (valeurs vides) sont traitées comme des caractères dans les *chaîne\_de\_caractères* et *modèle* ; vous devez donc redoubler de vigilance lorsque vous traitez les espaces.

Dans *modèle*, il est possible d'indiquer un espace soit littéralement, en saisissant un espace, soit à l'aide du métacaractère \s. L'utilisation du métacaractère facilite la lecture des espaces dans *modèle*, qui risquent moins d'être négligées, notamment lorsque vous construisez des modèles plus complexes.

## Concaténation de champs

Vous pouvez concaténer deux champs ou plus dans *chaîne\_de\_caractères* si vous souhaitez que votre recherche porte sur plusieurs champs en même temps.

Par exemple :

```
REGEXFIND(Nom_Fournisseur+Rue_Fournisseur;"Matériel.*Grand")
```

recherche les mots « Matériel » et « Grand » séparés par zéro ou plusieurs caractères dans les champs **Nom\_Fournisseur** et **Rue\_Fournisseur**.

Une entreprise dont le nom comporte le mot « Matériel », située dans une rue intitulée « Grand », correspond à cette expression ordinaire. Par exemple, une entreprise nommée « Matériel Grand-Rue » peut correspondre.

Les champs concaténés sont traités comme un seul champ incluant les espaces de début et de fin des champs individuels sauf si vous utilisez la fonction ALLTRIM( ) pour supprimer les espaces.

### L'ordre des champs concaténés a de l'importance

REGEXFIND( ) recherchant les caractères dans *modèle* dans l'ordre que vous indiquez, l'ordre dans lequel vous concaténez les champs exerce une incidence. Si vous inversiez **Nom\_Fournisseur** et **Rue\_Fournisseur** dans l'expression ci-dessus, vous auriez moins de chances d'obtenir des résultats.

## Métacaractères des expressions ordinaires

Le tableau suivant répertorie les métacaractères que vous pouvez utiliser avec REGEXFIND( ) et REGEXREPLACE( ) et décrit l'opération effectuée par chacun.

Une autre syntaxe, plus complexe et prise en charge par Analytics, existe pour les expressions ordinaires. Une explication complète d'une syntaxe supplémentaire sort de la portée du présent guide. De nombreuses ressources expliquant les expressions ordinaires sont disponibles sur Internet.

Analytics utilise l'implémentation **ECMAScript** des expressions ordinaires. La plupart des implémentations des expressions ordinaires utilisent une syntaxe principale courante.

#### Remarque

L'implémentation actuelle d'expressions ordinaires dans Analytics ne prend pas complètement en charge les langues de recherche différente de l'anglais.

Métacaractère	Description
.	Correspond à n'importe quel caractère (à l'exception d'un caractère de nouvelle ligne)

Métacaractère	Description
?	Correspond à 0 ou 1 occurrence de la valeur littérale, du métacaractère ou de l'élément qui précède immédiatement
*	Correspond à 0 ou plusieurs occurrences de la valeur littérale, du métacaractère ou de l'élément qui précède immédiatement
+	Correspond à 1 ou plusieurs occurrences de la valeur littérale, du métacaractère ou de l'élément qui précède immédiatement
{ }	Correspond au nombre d'occurrences spécifié de la valeur littérale, du métacaractère ou de l'élément qui précède immédiatement. Il est possible de spécifier un nombre exact, une plage ou une plage non finie. Par exemple : <ul style="list-style-type: none"> <li>○ <b>a{3}</b> correspond à « aaa »</li> <li>○ <b>X{0,2}L</b> correspond à « L », « XL » et « XXL »</li> <li>○ <b>AB-ld{2,-}YZ</b> correspond à n'importe quel identificateur alphanumérique dont le corps de l'identificateur contient le préfixe « AB- », le suffixe « -YZ » et au moins deux chiffres</li> </ul>
[ ]	Correspond à n'importe quel caractère unique entre les crochets Par exemple : <ul style="list-style-type: none"> <li>○ <b>[aeiou]</b> correspond à a ou e ou i ou o ou u</li> <li>○ <b>[^aeiou]</b> correspond à n'importe quel caractère qui n'est ni un a ni un e ni un i ni un o ni un u</li> <li>○ <b>[A-G]</b> correspond à n'importe quelle lettre en majuscule comprise entre A et G</li> <li>○ <b>[A-Ga-g]</b> correspond à n'importe quelle lettre en majuscules comprise entre A et G ou en minuscules entre a et g</li> <li>○ <b>[5-9]</b> correspond à n'importe quel nombre compris entre 5 et 9</li> </ul>
()	Crée un groupe qui définit une séquence ou un bloc de caractères, qui peut ensuite être traité comme une seule unité. Par exemple : <ul style="list-style-type: none"> <li>○ <b>S(ch)?mid?th?</b> correspond à « Smith » ou « Schmidt »</li> <li>○ <b>(56A.*){2}</b> correspond à n'importe quel identificateur alphanumérique dans lequel la séquence « 56A » est présente au moins deux fois</li> <li>○ <b>(56A).*.*\1</b> correspond à n'importe quel identificateur alphanumérique dans lequel la séquence « 56A » est présente au moins deux fois, avec un trait d'union entre les deux occurrences</li> </ul>
\	Caractère d'échappement permettant d'indiquer que le caractère suivant immédiatement est un caractère littéral. Utilisez le caractère d'échappement si vous souhaitez faire correspondre littéralement des métacaractères. Par exemple, \ <code>(</code> trouve une parenthèse gauche et \ <code>\)</code> trouve une barre oblique inverse. Utilisez le caractère d'échappement si vous souhaitez faire correspondre littéralement l'un des caractères suivants : <b>^ \$ . * + ? = ! :   \ ( ) [ ] { }</b>

Métacaractère	Description
	Les autres caractères de ponctuation comme l'esperluette (&) ou le signe arobase (@) n'exigent pas de caractère d'échappement.
<b>\int</b>	<p>Indique qu'un groupe, défini auparavant avec des parenthèses ( ), se répète. <i>int</i> est un entier qui identifie la position séquentielle du groupe défini précédemment vis-à-vis des autres groupes. Ce métacaractère peut être utilisé dans le paramètre <i>modèle</i> à la fois dans both REGEXFIND ( ) et dans REGEXREPLACE ( ).</p> <p>Par exemple :</p> <ul style="list-style-type: none"> <li>o <b>(123).*\1</b> correspond à n'importe quel identificateur dans lequel le groupe de chiffres « 123 » apparaît au moins deux fois</li> <li>o <b>^(d{3}).*\1</b> correspond à n'importe quel identificateur dans lequel les trois premiers chiffres se répètent</li> <li>o <b>^(d{3}).*\1.*\1</b> correspond à n'importe quel identificateur dans lequel les trois premiers chiffres se répètent au moins deux fois</li> <li>o <b>^(D)(d)-.*\2\1</b> correspond à n'importe quel identificateur dans lequel le préfixe alphanumérique se répète avec l'inversion des caractères alphabétiques et numériques</li> </ul>
<b>\$int</b>	<p>Indique qu'un groupe trouvé dans une chaîne de caractères cible est utilisé dans une chaîne de remplacement. <i>int</i> est un entier qui identifie la position séquentielle du groupe dans la chaîne de caractères cible vis-à-vis des autres groupes. Ce métacaractère peut être uniquement utilisé dans le paramètre <i>nouvelle_chaine_de_caractères</i> dans REGEXREPLACE ( ).</p> <p>Par exemple :</p> <ul style="list-style-type: none"> <li>o Si le modèle <b>(d{3})[-]?(\d{3})[-]?(\d{4})</b> est utilisé pour correspondre à différents formats de numéros de téléphone, la <i>nouvelle_chaine_de_caractères</i> <b>(\$1)-\$2-\$3</b> peut être utilisée pour remplacer les numéros par eux-mêmes et standardiser la mise en forme. 999 123-4567 et 9991234567 deviennent tous les deux (999)-123-4567.</li> </ul>
<b> </b>	<p>Correspond au caractère, aux blocs de caractères ou à l'expression avant ou après la barre verticale ( )</p> <p>Par exemple :</p> <ul style="list-style-type: none"> <li>o <b>a b</b> correspond à a ou à b</li> <li>o <b>abc def</b> correspond à « abc » ou à « def »</li> <li>o <b>Sm(i y)th</b> correspond à Smith ou à Smyth</li> <li>o <b>[a-c][[Q-S]][[x-z]</b> correspond à l'une des lettres suivantes : a, b, c, Q, R, S, x, y, z</li> <li>o <b>\s </b> correspond à un espace ou à un trait d'union</li> </ul>
<b>\w</b>	Correspond à un caractère de mot (a à z, A à Z, 0 à 9 et le caractère de soulignement _)
<b>\W</b>	Correspond à un caractère de non-mot (non compris entre a et z ni entre A et Z ni entre 0 et 9 ni le caractère de soulignement _)
<b>\d</b>	Correspond à n'importe quel nombre (décimale)
<b>\D</b>	Correspond à n'importe quel nombre (tout caractère qui n'est pas une décimale)



Métacaractère	Description
<code>\s</code>	Correspond à un espace (valeur vide)
<code>\S</code>	Correspond à tout non-espace (caractère non vierge)
<code>\b</code>	<p>Correspond à une limite de mots (entre des caractères <code>\w</code> et <code>\W</code>)</p> <p>Les limites de mots n'utilisent pas d'espace eux-mêmes. Par exemple :</p> <ul style="list-style-type: none"> <li>◦ "Équipement uni" contient 4 limites de mots : une de chaque côté de l'espace et une au début et à la fin de la chaîne de caractères. "Équipement uni" est associé à l'expression ordinaire <code>\b\w*\b\W\b\w*\b</code></li> </ul> <p><b>Astuce</b></p> <p>Outre les espaces, les limites de mots peuvent émaner de virgules, points et autres caractères de non-mot.</p> <p>Par exemple, l'expression suivante juge le résultat comme T (« true », vrai) :</p> <pre>REGEXFIND("jsmith@example.net"; "\bexample\b")</pre>
<code>^</code>	<p>Correspond au début d'une chaîne de caractères</p> <p>Entre crochets <code>[ ]</code>, <code>^</code> inverse le contenu</p>
<code>\$</code>	Correspond à la fin d'une chaîne de caractères

## Fonctions associées

Si vous souhaitez rechercher et remplacer des modèles de mise en correspondance, utilisez la "Fonction `REGEXREPLACE()`" Page suivante.

# Fonction REGEXREPLACE()

Remplace toutes les instances de chaînes de caractères correspondant à une expression ordinaire par une nouvelle chaîne de caractères.

## Syntaxe

```
REGEXREPLACE(chaîne_de_caractères; modèle; nouvelle_chaîne_de_caractères)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	Le champ, l'expression ou la valeur littérale à tester pour un modèle correspondant.
<i>modèle</i>	caractère	La chaîne de modèle (expression ordinaire) à rechercher. <i>modèle</i> peut contenir des caractères littéraux, des métacaractères ou une association des deux. Les caractères littéraux incluent tous les caractères alphanumériques, certains caractères de ponctuation et des caractères vides. La recherche respecte la casse, ce qui signifie que les caractères alphanumériques en majuscules et en minuscules doivent être explicitement spécifiés.
<i>nouvelle_chaîne_de_caractères</i>	caractère	La chaîne de caractères à utiliser pour remplacer toutes les valeurs correspondant au <i>modèle</i> . La chaîne de remplacement peut contenir des caractères littéraux, des groupes de caractères à partir de la chaîne de caractères d'origine (à l'aide de l'élément <i>\$int</i> ) ou une combinaison des deux.

## Sortie

Caractère.

# Exemples

## Exemples basiques

### Utilisation des espaces

Renvoie "AB CD EF", en remplaçant plusieurs espaces entre les caractères textuels par une seule espace :

```
REGEXREPLACE("AB CD EF"; "\s+"; " ")
```

Renvoie les données des champs caractères avec l'espacement entre les mots standardisé sur une seule espace :

```
REGEXREPLACE(champ_caractère; "\s+"; " ")
```

Renvoie les données des champs caractères avec l'espacement entre les mots standardisé sur une seule espace. L'utilisation de la fonction `BLANKS()` dans *nouvelle\_chaine\_de\_caractères* plutôt qu'une espace littérale facilite la lecture des espaces ; elles sont donc moins susceptibles d'être oubliées :

```
REGEXREPLACE(champ_caractère; "\s+"; BLANKS(1))
```

### Standardisation des numéros de téléphone

Renvoie "(123) 456-7890". La mise en forme du numéro de téléphone « 1234567890 » est standardisée.

```
REGEXREPLACE(SUBSTR("1234567890";1;14); "(\\d{3})[\\s-]*(\\d{3})[\\s-]*(\\d{4})"; "(\\$1) \\$2-\\$3")
```

Renvoie les nombres dans le champ **Numéro\_Téléphone** et standardise leur mise en forme :

```
REGEXREPLACE(Numéro_Téléphone; ".*(\\d{3})[\\s-\\.\\)]*(\\d{3})[\\s-\\.\\)]*(\\d{4})"; "(\\$1) \\$2-\\$3")
```

Extrait "123-456-7890" dans le texte environnant :

```
REGEXREPLACE("N° tél : 123-456-7890 (bureau)"; "(.*)((\\d{3})[\\s-\\.\\)]*(\\d{3})[\\s-\\.\\)]*(\\d{4})).*"; "\\$2")
```

Extrait les numéros de téléphone du texte environnant dans le champ **Commentaire** et standardise leur mise en forme :

```
REGEXREPLACE(Commentaire; "(.*)((\\d{3})[\\s-\\.\\)]*(\\d{3})[\\s-\\.\\)]*(\\d{4})).*"; "(\\$2) \\$3-\\$4")
```

## Identification de formats génériques

Renvoie "9XXX-999xx", qui représente le format générique de la valeur spécifiée par *chaîne\_de\_caractères* ("1ABC-123aa") :

```
REGEXREPLACE(REGEXREPLACE(REGEXREPLACE("1ABC-123aa";"d";"9");"[a-z]";"x");"[A-Z]";"X")
```

Renvoie le format générique de tous les identificateurs dans le champ **No\_Facture** :

```
REGEXREPLACE(REGEXREPLACE(REGEXREPLACE(No_Facture;"d";"9");"[a-z]";"x");"[A-Z]";"X")
```

## Standardisation du format des noms

Renvoie "John David Smith" :

```
REGEXREPLACE("Smith, John David";"^(\\w+);(\\s\\w+)(\\s\\w+)?(\\s\\w+)?";"$2$3$4 $1")
```

Renvoie les noms dans le champ **Nom\_complet** dans leur ordre ordinaire : *Prénom (Deuxième prénom) (Troisième prénom) Nom* :

```
REGEXREPLACE(Nom_complet;"^(\\w+);(\\s\\w+)(\\s\\w+)?(\\s\\w+)?";"$2$3$4 $1")
```

### Remarque

Les données des noms peuvent présenter diverses complications, comme des apostrophes dans des noms. La représentation des variations des données de noms requiert généralement des expressions ordinaires plus complexes que celle donnée dans l'exemple précédent.

## Remarques

### Fonctionnement

La fonction REGEXREPLACE() utilise une expression ordinaire pour rechercher des modèles correspondants dans les données et remplace les valeurs correspondantes par une nouvelle chaîne de caractères.

Par exemple :

```
REGEXREPLACE(champ_caractère,"\\s+";" ")
```

standardise l'espacement dans les données caractère en remplaçant une ou plusieurs espaces entre les caractères texte par une espace unique.

La portion de recherche de REGEXREPLACE() est identique à la fonction REGEXFIND(). Pour des informations détaillées sur la fonctionnalité de recherche commune aux deux fonctions, consultez la section "Fonction REGEXFIND()" Page 762.

## Quand utiliser la fonction REGEXREPLACE()

Utilisez la fonction REGEXREPLACE() à chaque fois que vous souhaitez rechercher et remplacer des données dans Analytics à l'aide d'une correspondance de modèles simple ou complexe.

## Remplacement de caractères par eux-mêmes

Il est possible d'utiliser l'élément *\$int* pour remplacer des caractères par eux-mêmes, ce qui permet de conserver les parties significatives des données, tout en standardisant ou en omettant les données environnantes ou fusionnées.

Vous trouverez ci-dessus plusieurs exemples d'utilisation de numéros de téléphone et de noms.

Pour utiliser l'élément *\$int*, vous devez d'abord créer des groupes à l'aide de parenthèses ( ) dans la valeur *modèle*. Pour plus d'informations, consultez la section "Fonction REGEXFIND()" Page 762.

## Éviter la correspondance de caractères séquentiels

Il est possible d'éviter une correspondance de caractères séquentielle et de remplacer les sous-chaînes indépendamment de leur position l'une vis-à-vis de l'autre, en imbriquant les fonctions REGEXREPLACE().

Dans les deux exemples ci-dessous, le problème consiste à tirer un format générique des données source alphanumériques dans lesquelles des nombres et des lettres peuvent apparaître dans n'importe quel ordre. Sans connaître l'ordre des nombres et des lettres, comment créer la chaîne de caractères *modèle* ?

La solution consiste tout d'abord à rechercher-remplacer des nombres à l'aide de la fonction REGEXREPLACE() interne, puis à rechercher-remplacer des lettres à l'aide de la fonction REGEXREPLACE() externe.

Renvoie "999XXX" :

```
REGEXREPLACE(REGEXREPLACE("123ABC";"d","9");"[A-Z]";"X")
```

Renvoie "9X9X9X" :

```
REGEXREPLACE(REGEXREPLACE("1A2B3C";"d","9");"[A-Z]";"X")
```

## Longueur et troncation des chaînes de remplacement

Lorsque vous utilisez REGEXREPLACE() pour créer un champ calculé, la longueur du champ calculé est identique à la longueur du champ d'origine.

Si la longueur de la chaîne de remplacement dépasse la longueur de la chaîne cible, la longueur globale de la chaîne augmente, ce qui aboutit à une troncation si la longueur du champ calculé ne peut pas adapter la longueur de chaîne augmentée.

Les caractères qui tracent la chaîne cible sont tronqués en premier ; ils sont ensuite suivis des caractères de fin de la chaîne de remplacement. Les exemples ci-dessous illustrent la troncation :

chaîne	modèle	nouvelle_chaîne_de_caractères	Longueur de champ	Résultat	Caractères tronqués
x123x	123	A	5	"xAx"	aucun
x123x	123	ABC	5	"xABCx"	aucun
x123x	123	ABCD	5	"xABCD"	"x"
x123x	123	ABCDE	5	"xABCD"	"x", "E"
x123x	123	ABCDE	6	"xABCDE"	"x"
x123x	123	ABCDE	7	"xABCDEx"	aucun

## Comment éviter la troncation

Pour éviter la troncation, utilisez la fonction SUBSTR() pour augmenter la longueur du champ, comme montré dans le deuxième exemple ci-dessous.

Renvoie "xABCD", qui tronque le caractère de remplacement "E" et le caractère existant "x" :

```
REGEXREPLACE("x123x","123","ABCDE")
```

Renvoie "xABCDEx", qui inclut tous les caractères de remplacement et les caractères existants non remplacés :

```
REGEXREPLACE(SUBSTR("x123x";1;10);"123";"ABCDE")
```

## Métacaractères des expressions ordinaires

Le tableau suivant répertorie les métacaractères que vous pouvez utiliser avec REGEXFIND() et REGEXREPLACE() et décrit l'opération effectuée par chacun.

Une autre syntaxe, plus complexe et prise en charge par Analytics, existe pour les expressions ordinaires. Une explication complète d'une syntaxe supplémentaire sort de la portée du présent guide. De nombreuses ressources expliquant les expressions ordinaires sont disponibles sur Internet.

Analytics utilise l'implémentation **ECMAScript** des expressions ordinaires. La plupart des implémentations des expressions ordinaires utilisent une syntaxe principale courante.

### Remarque

L'implémentation actuelle d'expressions ordinaires dans Analytics ne prend pas complètement en charge les langues de recherche différente de l'anglais.

Métacaractère	Description
.	Correspond à n'importe quel caractère (à l'exception d'un caractère de nouvelle ligne)
?	Correspond à 0 ou 1 occurrence de la valeur littérale, du métacaractère ou de l'élément qui précède immédiatement
*	Correspond à 0 ou plusieurs occurrences de la valeur littérale, du métacaractère ou de l'élément qui précède immédiatement
+	Correspond à 1 ou plusieurs occurrences de la valeur littérale, du métacaractère ou de l'élément qui précède immédiatement
{ }	Correspond au nombre d'occurrences spécifié de la valeur littérale, du métacaractère ou de l'élément qui précède immédiatement. Il est possible de spécifier un nombre exact, une plage ou une plage non finie.  Par exemple : <ul style="list-style-type: none"> <li>○ <b>a{3}</b> correspond à « aaa »</li> <li>○ <b>X{0,2}L</b> correspond à « L », « XL » et « XXL »</li> <li>○ <b>AB-ld{2,-}YZ</b> correspond à n'importe quel identificateur alphanumérique dont le corps de l'identificateur contient le préfixe « AB- », le suffixe « -YZ » et au moins deux chiffres</li> </ul>
[ ]	Correspond à n'importe quel caractère unique entre les crochets  Par exemple : <ul style="list-style-type: none"> <li>○ <b>[aeiou]</b> correspond à a ou e ou i ou o ou u</li> <li>○ <b>[^aeiou]</b> correspond à n'importe quel caractère qui n'est ni un a ni un e ni un i ni un o ni un u</li> <li>○ <b>[A-G]</b> correspond à n'importe quelle lettre en majuscule comprise entre A et G</li> <li>○ <b>[A-Ga-g]</b> correspond à n'importe quelle lettre en majuscules comprise entre A et G ou en minuscules entre a et g</li> <li>○ <b>[5-9]</b> correspond à n'importe quel nombre compris entre 5 et 9</li> </ul>
()	Crée un groupe qui définit une séquence ou un bloc de caractères, qui peut ensuite être traité comme une seule unité.  Par exemple : <ul style="list-style-type: none"> <li>○ <b>S(ch)?mid?th?</b> correspond à « Smith » ou « Schmidt »</li> <li>○ <b>(56A.*){2}</b> correspond à n'importe quel identificateur alphanumérique dans lequel la séquence</li> </ul>

Métacaractère	Description
	<p>« 56A » est présente au moins deux fois</p> <ul style="list-style-type: none"> <li>◦ <b>(56A).*-.*\1</b> correspond à n'importe quel identificateur alphanumérique dans lequel la séquence « 56A » est présente au moins deux fois, avec un trait d'union entre les deux occurrences</li> </ul>
\	<p>Caractère d'échappement permettant d'indiquer que le caractère suivant immédiatement est un caractère littéral. Utilisez le caractère d'échappement si vous souhaitez faire correspondre littéralement des métacaractères. Par exemple, \<code>(</code> trouve une parenthèse gauche et \<code>\</code> trouve une barre oblique inverse.</p> <p>Utilisez le caractère d'échappement si vous souhaitez faire correspondre littéralement l'un des caractères suivants :</p> <p><b>^\$. * + ? = ! :   \ ( ) [ ] { }</b></p> <p>Les autres caractères de ponctuation comme l'esperluette (&amp;) ou le signe arobase (@) n'exigent pas de caractère d'échappement.</p>
\int	<p>Indique qu'un groupe, défini auparavant avec des parenthèses ( ), se répète. <i>int</i> est un entier qui identifie la position séquentielle du groupe défini précédemment vis-à-vis des autres groupes. Ce métacaractère peut être utilisé dans le paramètre <i>modèle</i> à la fois dans both REGEXFIND( ) et dans REGEXREPLACE( ).</p> <p>Par exemple :</p> <ul style="list-style-type: none"> <li>◦ <b>(123).*\1</b> correspond à n'importe quel identificateur dans lequel le groupe de chiffres « 123 » apparaît au moins deux fois</li> <li>◦ <b>^(d{3}).*\1</b> correspond à n'importe quel identificateur dans lequel les trois premiers chiffres se répètent</li> <li>◦ <b>^(d{3}).*\1.*\1</b> correspond à n'importe quel identificateur dans lequel les trois premiers chiffres se répètent au moins deux fois</li> <li>◦ <b>^(D)(d)-.*\2\1</b> correspond à n'importe quel identificateur dans lequel le préfixe alphanumérique se répète avec l'inversion des caractères alphabétiques et numériques</li> </ul>
\$int	<p>Indique qu'un groupe trouvé dans une chaîne de caractères cible est utilisé dans une chaîne de remplacement. <i>int</i> est un entier qui identifie la position séquentielle du groupe dans la chaîne de caractères cible vis-à-vis des autres groupes. Ce métacaractère peut être uniquement utilisé dans le paramètre <i>nouvelle_chaine_de_caractères</i> dans REGEXREPLACE( ).</p> <p>Par exemple :</p> <ul style="list-style-type: none"> <li>◦ Si le modèle <b>(d{3})[-]?(d{3})[-]?(d{4})</b> est utilisé pour correspondre à différents formats de numéros de téléphone, la <i>nouvelle_chaine_de_caractères</i> <b>(\$1)-\$2-\$3</b> peut être utilisée pour remplacer les numéros par eux-mêmes et standardiser la mise en forme. 999 123-4567 et 9991234567 deviennent tous les deux (999)-123-4567.</li> </ul>
	<p>Correspond au caractère, aux blocs de caractères ou à l'expression avant ou après la barre verticale ( )</p> <p>Par exemple :</p> <ul style="list-style-type: none"> <li>◦ <b>a b</b> correspond à a ou à b</li> <li>◦ <b>abc def</b> correspond à « abc » ou à « def »</li> <li>◦ <b>Sm(ily)th</b> correspond à Smith ou à Smyth</li> <li>◦ <b>[a-c][Q-S][x-z]</b> correspond à l'une des lettres suivantes : a, b, c, Q, R, S, x, y, z</li> </ul>



Métacaractère	Description
	<ul style="list-style-type: none"> <li>◦ <code>\s </code> correspond à un espace ou à un trait d'union</li> </ul>
<code>\w</code>	Correspond à un caractère de mot (a à z, A à Z, 0 à 9 et le caractère de soulignement <code>_</code> )
<code>\W</code>	Correspond à un caractère de non-mot (non compris entre a et z ni entre A et Z ni entre 0 et 9 ni le caractère de soulignement <code>_</code> )
<code>\d</code>	Correspond à n'importe quel nombre (décimale)
<code>\D</code>	Correspond à n'importe quel nombre (tout caractère qui n'est pas une décimale)
<code>\s</code>	Correspond à un espace (valeur vide)
<code>\S</code>	Correspond à tout non-espace (caractère non vierge)
<code>\b</code>	<p>Correspond à une limite de mots (entre des caractères <code>\w</code> et <code>\W</code>)</p> <p>Les limites de mots n'utilisent pas d'espace eux-mêmes. Par exemple :</p> <ul style="list-style-type: none"> <li>◦ "Équipement uni" contient 4 limites de mots : une de chaque côté de l'espace et une au début et à la fin de la chaîne de caractères. "Équipement uni" est associé à l'expression ordinaire <code>\b\w*\b\W\b\w*\b</code></li> </ul> <p><b>Astuce</b></p> <p>Outre les espaces, les limites de mots peuvent émaner de virgules, points et autres caractères de non-mot.</p> <p>Par exemple, l'expression suivante juge le résultat comme T (« true », vrai) :</p> <pre>REGEXFIND("jsmith@example.net"; "\bexample\b")</pre>
<code>^</code>	<p>Correspond au début d'une chaîne de caractères</p> <p>Entre crochets <code>[ ]</code>, <code>^</code> inverse le contenu</p>
<code>\$</code>	Correspond à la fin d'une chaîne de caractères

## Fonctions associées

Si vous souhaitez rechercher des modèles de mise en correspondance sans les remplacer, utilisez la "Fonction REGEXFIND()" Page 762.

# Fonction REMOVE()

Renvoie une chaîne qui inclut uniquement les caractères spécifiés.

## Syntaxe

```
REMOVE(chaîne_de_caractères; caractères_valides)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	Chaîne dont il faut supprimer les caractères.
<i>caractères_valides</i>	caractère	Caractères à conserver dans <i>chaîne_de_caractères</i> . Si vous indiquez des guillemets doubles dans les <i>caractères_valides</i> , vous devez placer la liste de caractères à l'intérieur de guillemets simples. Par exemple : "'-/' <b>Remarque</b> Si un caractère que vous spécifiez n'apparaît pas dans <i>chaîne de caractères</i> , il n'est pas inclus dans la valeur renvoyée.

## Sortie

Caractère.

## Exemples

### Exemples basiques

Renvoie "ABC123 " :

```
REMOVE("ABC 123 XX4","ABC123")
```

Renvoie "ABC123XX " :

```
REMOVE("zABC 123 XX4";"ABCX123")
```

Renvoie "1234 " :

```
REMOVE("ABC 123 XX4"; "1234567890")
```

Renvoie toutes les valeurs du champ **Numéro\_Produit** dans lesquelles tous les caractères non numériques ont été supprimés :

```
REMOVE(Numéro_Produit;"0123456789")
```

## Remarques

### Remarque

La fonction REMOVE( ) a été remplacée par les fonctions INCLUDE( ) et EXCLUDE( ).

REMOVE( ) est toujours disponible dans la version actuelle d'Analytics pour rétro-compatibilité avec des versions antérieures.

## Fonctionnement

La fonction REMOVE( ) supprime les caractères indésirables des données de caractère et renvoie une chaîne de longueur fixe.

## Quand utiliser la fonction REMOVE( )

Utilisez REMOVE( ) pour normaliser les champs de données dont le format n'est pas cohérent (par exemple, les champs d'adresse). Vous pouvez également utiliser REMOVE( ) pour supprimer la ponctuation ou d'autres informations incorrectes dans des champs peu édités.

Vous pouvez aussi utiliser la fonction pour nettoyer les données des champs avant d'utiliser les commandes SORT ou JOIN, pour la correspondance de doublons ou la sortie de rapport.

## Respect de la casse

La fonction REMOVE( ) respecte la casse. Si vous indiquez « ID » dans *caractères\_valides*, ces caractères ne seront pas inclus dans « id#94022 ». Si vous pensez que la casse est incohérente dans la *chaîne*, utilisez la fonction UPPER( ) pour convertir tous les caractères en majuscules.

Par exemple :

```
REMOVE(UPPER("id#94022"); "ID0123456789")
```

## Fonctions associées

REMOVE() est similaire à la fonction INCLUDE(), à ceci près que

- REMOVE() ajoute des espaces à la fin de la sortie pour remplacer les caractères supprimés. La longueur d'origine de la *chaîne* est conservée.
- INCLUDE() n'ajoute pas d'espace.

# Fonction REPEAT()

Renvoie une chaîne répétant un certain nombre de fois une sous-chaîne.

## Syntaxe

```
REPEAT(chaîne_de_caractères; compte)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	La chaîne de caractères à répéter.
<i>compte</i>	numérique	Nombre de fois que la valeur de <i>chaîne_de_caractères</i> doit être répétée.

## Sortie

Caractère.

## Exemples

### Exemples basiques

Renvoie "ABCABCABC":

```
REPEAT("ABC";3)
```

Renvoie "000000000" :

```
REPEAT("0";9)
```

# Remarques

## Quand utiliser REPEAT( )

Utilisez la fonction REPEAT( ) pour initialiser une variable avec des valeurs constantes ou des espaces, ou paramétrer une valeur par défaut pour un champ calculé.

# Fonction REPLACE( )

Remplace toutes les instances d'une chaîne de caractères indiquée par une nouvelle chaîne de caractères.

## Syntaxe

```
REPLACE(chaîne_de_caractères; ancien_texte; nouveau_texte)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	Valeur dans laquelle remplacer les caractères.
<i>ancien_texte</i>	caractère	Chaîne de caractères à remplacer. La recherche est sensible à la casse.
<i>nouveau_texte</i>	caractère	Texte avec lequel remplacer la valeur se trouvant dans <i>ancien_texte</i> .

## Sortie

Caractère.

## Exemples

### Exemples basiques

Renvoie "a12345efg" :

```
REPLACE("abcdefg";"bcd";"12345")
```

Renvoie "Rd." :

```
REPLACE("Road";"Road";"Rd.")
```

Renvoie "ac" :

```
REPLACE("abc";"b";"")
```

## Exemples avancés

### Suppression des caractères spécifiés

Utilisez REPLACE() pour supprimer une chaîne de caractères spécifiés d'une chaîne source en la remplaçant par une chaîne de caractères vides ( "" ).

Renvoie "1234 Scott" :

```
REPLACE("1234 Scott rd."; "rd."; "")
```

### Adaptation de la longueur du champ

Si le *nouveau\_texte* ("ABC") est plus long que *ancien\_texte* ("X"), la longueur de champ de la chaîne créée est automatiquement augmentée pour adapter le premier remplacement :

Renvoie "9ABC9", avec une longueur de champ qui passe de 3 à 5 caractères :

```
REPLACE("9X9"; "X"; "ABC")
```

La longueur de champ n'est pas automatiquement augmentée pour les remplacements suivants et une coupure peut survenir si le champ est trop court pour tenir compte de tous les nouveaux caractères.

Renvoie "9ABC9A" :

```
REPLACE("9X9X"; "X"; "ABC")
```

Pour éviter les coupures, vous pouvez augmenter la longueur de la *chaîne* à l'aide de la fonction BLANKS() ou d'espaces vides littéraux.

Renvoie "9ABC9ABC" :

```
REPLACE("9X9X" + BLANKS(2); "X"; "ABC")
```

```
REPLACE("9X9X" + " "; "X"; "ABC")
```

Si la chaîne créée est plus courte que *chaîne*, celle-ci est remplie avec des espaces pour conserver la même longueur de champ.

Renvoie "9X9" :

```
REPLACE("9ABC9"; "ABC"; "X")
```



# Remarques

## Fonctionnement

La fonction REPLACE( ) remplace toutes les instances d'une chaîne existante par une nouvelle chaîne.

Renvoie "1234 Scott Road" :

```
REPLACE("1234 Scott rd."; "rd."; "Road")
```

## Quand utiliser la fonction REPLACE( )

Utilisez REPLACE( ) pour normaliser les champs de données dont les formats sont incohérents, comme les champs d'adresse, ou pour remplacer des informations non valides dans des champs mal édités. Pour s'exécuter correctement, des opérations comme le test des doublons, ou la liaison ou mise en relation de tables, requièrent des données dans un format normalisé ou standardisé.

## Respect de la casse

La fonction REPLACE( ) respecte la casse. Si vous indiquez "RD." dans le paramètre *ancien\_texte* et que les valeurs de la *chaîne* sont en minuscules, la valeur du *nouveau\_texte* ne sera pas remplacée car aucune correspondance n'existe.

Si vous pensez que la casse est incohérente dans la *chaîne*, commencez par utiliser la fonction UPPER( ) pour convertir tous les caractères en majuscules.

Renvoie "1234 SCOTT ROAD" :

```
REPLACE(UPPER("1234 Scott rd."); "RD."; "ROAD")
```

## Se protéger des remplacements commis par inadvertance

Lors de la création d'une expression REPLACE( ), il est impératif que vous connaissiez l'existence de toutes les instances du paramètre *ancien\_texte* dans la *chaîne\_de\_caractères* afin d'éviter tout remplacement involontaire.

Renvoie "645 RichaRoad Road", car les deux dernières lettres de "Richard" sont "rd" :

```
REPLACE("645 Richard rd "; "rd"; "Road")
```

Pour empêcher la fonction de remplacer les instances dans lesquelles "rd" apparaît dans un nom (car ces instances ne comportent pas d'espace de début), ajoutez une espace de début et une espace de fin à la valeur dans *ancien\_texte* (" rd ").

Renvoie "645 Richard Road" :

```
REPLACE("645 Richard rd "; " rd "; " Road")
```

# Fonction REVERSE( )

Renvoie une chaîne dont l'ordre des caractères est inversé.

## Syntaxe

```
REVERSE(chaîne_de_caractères)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	Valeur dont il faut inverser l'ordre.

## Sortie

Caractère.

## Exemples

### Exemples basiques

Renvoie « E DCBA » :

```
REVERSE("ABCD E")
```

# Fonction RJUSTIFY()

Renvoie une chaîne justifiée à droite dont la longueur est égale à celle d'une chaîne indiquée et dont les espaces de fin ont été déplacés vers la gauche de la chaîne.

## Syntaxe

```
RJUSTIFY(chaîne_de_caractères)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	Valeur qu'il faut justifier à droite.

## Sortie

Caractère.

## Exemples

### Exemples basiques

Renvoie « ABC » :

```
RJUSTIFY("ABC ")
```

## Remarques

### Quand utiliser RJUSTIFY()

Utilisez la fonction RJUSTIFY() pour justifier un champ caractère à droite.

# Fonction RLOGICAL()

Renvoie une valeur logique calculée par un script ou une fonction de Rvachev. Le traitement des données dans Rvachev est externe à Analytics.

## Syntaxe

```
RLOGICAL(rScript/rCode <;champ/valeur <;...n>>)
```

## Paramètres

Nom	Type	Description
<i>rScript</i> / <i>rCode</i>	caractère	<p>Le chemin d'accès complet ou relatif au script R ou à un snippet de code R à exécuter.</p> <p>Si vous saisissez le code R directement plutôt que d'utiliser un fichier externe, vous ne pouvez pas utiliser le caractère de guillemet fermant, même si vous l'échappez :</p> <ul style="list-style-type: none"> <li>◦ <b>valide</b> : 'var &lt;- "\test"'</li> <li>◦ <b>non valide</b> : 'var &lt;- "\test"'</li> </ul>
<i>champ</i> / <i>valeur</i> <;... <i>n</i> > optionnel	caractère numérique DateHeure logique	<p>La liste des champs, des expressions ou des valeurs littérales à utiliser comme arguments pour le snippet de code ou le script R.</p> <p>Les valeurs sont transmises dans la fonction que vous appelez dans l'ordre dans lequel vous les indiquez, puis vous y référez à l'aide de <i>valeur1</i>; <i>valeur2</i> ... <i>valeurN</i> dans le code R.</p> <p>Vous pouvez inclure autant d'arguments que nécessaires pour répondre à la définition de la fonction dans le code R.</p> <p><b>Remarque</b></p> <p>La fonction ALLTRIM() sert à supprimer les espaces de début ou de fin de l'entrée de caractères : ALLTRIM(str). Pour plus d'informations, consultez la section "Fonction ALLTRIM()" Page 493.</p>

## Sortie

Logique.

# Exemples

## Exemples basiques

Renvoie T :

```
RLOGICAL("(valeur1>0.6) & (valeur2>0.7) & (valeur3>0.5)"; 0.8; 0.9; 0.55)
```

## Exemples avancés

### Utilisation d'un script R externe

Accepte un montant et une valeur de seuil supérieure et inférieure. La fonction renvoie une valeur de vérité selon une série de comparaisons logiques :

```
RLOGICAL("a<-'c:\scripts\sample.r');a[[1]]"; montant_dépense; seuil_bas; seuil_élevé)
```

Script R externe (sample.r) :

```
test_truth <- function(montant; bas; élevé) {
  return(((montant > bas) & (montant < élevé)) | ((montant==bas) | (montant==élevé)))
}
test_truth(valeur1; valeur2; valeur3)
```

### Utilisation du code R stocké dans une variable

Effectue un test logique des trois champs à l'aide de la logique AND :

```
v_rcode = "(valeur1>0,6) & (valeur2>0,7) & (valeur3>0,5)"
RLOGICAL(v_rcode; PACKED; MICRO_LONG; ACCPAC)
```

## Remarques

### Renvoi de données à partir de R

Lorsque vous appelez des scripts R, utilisez la fonction `source` et affectez l'objet renvoyé à une variable. Vous pouvez alors accéder à la valeur renvoyée depuis votre fonction R à partir de l'objet renvoyé :

```
# 'a' contient l'objet de réponse et a[[1]] accède à la valeur des données
"a<-source('c:\scripts\r_scripts\sample.r');a[[1]]"
```

## Fichier trace R

Analytics enregistre les messages en langage R dans un fichier `aclrlang.log` dans le dossier de projet. Utilisez ce fichier trace pour le débogage des erreurs R

### Astuce

Ce fichier trace se trouve dans le dossier de résultats des tâches analytiques d'Analytics Exchange.

## Exécution de scripts R externes sur AX Serveur

Si vous rédigez une App Analyse à exécuter sur AX Serveur et que vous souhaitez travailler sur des scripts R externes :

1. Chargez le fichier sous forme de fichier associé à l'App Analyse.
2. Utilisez la balise d'analyse FILE pour identifier le(s) fichier(s).
3. Faites référence au(x) fichier(s) à l'aide du chemin d'accès relatif `./nom_fichier.r`.

### Remarque

L'utilisation d'un fichier associé garantit que le compte du serveur d'application TomEE dispose des autorisations suffisantes pour accéder au fichier lors de l'exécution de R avec Analytics Exchange.

# Fonction RNUMERIC()

Renvoie une valeur numérique calculée par un script ou une fonction de Rvachev. Le traitement des données dans Rvachev est externe à Analytics.

## Syntaxe

```
RNUMERIC(rScript|rCode; décimales <;champ|valeur <;...n>>)
```

## Paramètres

Nom	Type	Description
<i>rScript   rCode</i>	caractère	<p>Le chemin d'accès complet ou relatif au script R ou à un snippet de code R à exécuter.</p> <p>Si vous saisissez le code R directement plutôt que d'utiliser un fichier externe, vous ne pouvez pas utiliser le caractère de guillemet fermant, même si vous l'échappez :</p> <ul style="list-style-type: none"> <li>o <b>valide</b> : 'var &lt;- "\"test\""'</li> <li>o <b>non valide</b> : 'var &lt;- "\"test\"'</li> </ul>
<i>décimales</i>	numérique	<p>Nombre de décimales à inclure dans la valeur renvoyée. Doit être un nombre entier positif.</p>
<i>champ   valeur</i> <;... <i>n</i> > optionnel	caractère numérique DateHeure logique	<p>La liste des champs, des expressions ou des valeurs littérales à utiliser comme arguments pour le snippet de code ou le script R.</p> <p>Les valeurs sont transmises dans la fonction que vous appelez dans l'ordre dans lequel vous les indiquez, puis vous y référez à l'aide de valeur1; valeur2 ... valeurN dans le code R.</p> <p>Vous pouvez inclure autant d'arguments que nécessaires pour répondre à la définition de la fonction dans le code R.</p> <p><b>Remarque</b></p> <p>La fonction ALLTRIM() sert à supprimer les espaces de début ou de fin de l'entrée de caractères : ALLTRIM(str). Pour plus d'informations, consultez la section "Fonction ALLTRIM()" Page 493.</p>

## Sortie

Numérique.



# Exemples

## Exemples basiques

Renvoie 100 avec 10 décimales (100.0000000000) :

```
RNUMERIC("print(value1)"; 10; 100)
```

## Exemples avancés

### Stockage d'un code R en tant que variable

Renvoie 100 avec 10 décimales (100,0000000000) :

```
ASSIGN v_rcode = "print(value1)"
RNUMERIC(v_rcode; 10; 100)
```

### Écriture dans un fichier externe

Effectue une simple addition et écrit le commentaire joint à la fonction au fichier à l'aide de la fonction `sink` dans R :

```
RNUMERIC("foo<-function(x,y){x+y};attr(foo, 'comment') <- 'foo performs simple addition';sink('c:/-temp/result.txt');attributes(foo);sink(NULL);foo(value1,value2)"; 0; amt; gross)
```

# Remarques

## Renvoi de données à partir de R

Lorsque vous appelez des scripts R, utilisez la fonction `source` et affectez l'objet renvoyé à une variable. Vous pouvez alors accéder à la valeur renvoyée depuis votre fonction R à partir de l'objet renvoyé :

```
# 'a' contient l'objet de réponse et a[[1]] accède à la valeur des données
"a<-source('c:\\scripts\\r_scripts\\sample.r');a[[1]]"
```

## Fichier trace R

Analytics enregistre les messages en langage R dans un fichier `aclrlang.log` dans le dossier de projet. Utilisez ce fichier trace pour le débogage des erreurs R

**Astuce**

Ce fichier trace se trouve dans le dossier de résultats des tâches analytiques d'Analytics Exchange.

## Exécution de scripts R externes sur AX Serveur

Si vous rédigez une App Analyse à exécuter sur AX Serveur et que vous souhaitez travailler sur des scripts R externes :

1. Chargez le fichier sous forme de fichier associé à l'App Analyse.
2. Utilisez la balise d'analyse FILE pour identifier le(s) fichier(s).
3. Faites référence au(x) fichier(s) à l'aide du chemin d'accès relatif `./nom_fichier.r`.

**Remarque**

L'utilisation d'un fichier associé garantit que le compte du serveur d'application TomEE dispose des autorisations suffisantes pour accéder au fichier lors de l'exécution de R avec Analytics Exchange.

# Fonction ROOT()

Renvoie la racine carrée d'une expression numérique.

## Syntaxe

```
ROOT(nombre; décimales)
```

## Paramètres

Nom	Type	Description
<i>nombre</i>	numérique	Expression numérique dont il faut trouver la racine carrée. Cette fonction renvoie zéro si <i>nombre</i> est un nombre négatif.
<i>décimales</i>	numérique	Nombre de décimales à utiliser dans la sortie.

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 10,00 :

```
ROOT(100; 2)
```

Renvoie 31,6228 :

```
ROOT(1000; 4)
```

# Remarques

## Fonctionnement

La fonction `ROOT()` renvoie la racine carrée de l'expression numérique ou valeur de champ avec le nombre indiqué de décimales. Le résultat est arrondi de façon appropriée.

## Quand utiliser la fonction `ROOT()`

Utilisez `LOG()` pour exécuter d'autres fonctions racine, telles que la racine cubique.

# Fonction ROUND()

Renvoie un nombre entier arrondi pour une valeur numérique.

## Syntaxe

```
ROUND(nombre)
```

## Paramètres

Nom	Type	Description
<i>nombre</i>	numérique	Valeur à arrondir au nombre entier le plus proche.

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 7 :

```
ROUND(7,2)
```

Renvoie 8 :

```
ROUND(7,5)
```

Renvoie -8 :

```
ROUND(-7,5)
```

## Exemples avancés

### Arrondir des valeurs monétaires

Crée un champ qui correspond au solde arrondi à la valeur euro la plus proche :

```
DEFINE FIELD Valeur_euro_la_plus_proche COMPUTED ROUND(Solde)
```

## Remarques

### Fonctionnement

ROUND( ) renvoie un nombre égal à la valeur *nombre* arrondie au nombre entier le plus proche.

	Valeurs positives	Valeurs négatives
Arrondi à l'entier suivant supérieur	$\geq 0,5$	$< 0,5$
Arrondi à l'entier suivant inférieur	$< 0,5$	$\geq 0,5$

### Arrondir avec un nombre précis de décimales

Pour arrondir un nombre avec un nombre précis de décimales, utilisez la fonction "Fonction DEC( )" Page 564. La fonction ROUND( ) est équivalente à la fonction DEC( ), à ceci près qu'elle n'indique aucune décimale.

```
ROUND(nombre)
```

est équivalent à :

```
DEC(number, 0)
```

# Fonction RSTRING( )

Renvoie une valeur de chaîne de caractères calculée par un script ou une fonction R. Le traitement des données dans Rvachev est externe à Analytics.

## Syntaxe

```
RSTRING(rScript|rCode; longueur <;champ|valeur <;...n>>)
```

## Paramètres

Nom	Type	Description
<i>rScript   rCode</i>	caractère	<p>Le chemin d'accès complet ou relatif au script R ou à un snippet de code R à exécuter.</p> <p>Si vous saisissez le code R directement plutôt que d'utiliser un fichier externe, vous ne pouvez pas utiliser le caractère de guillemet fermant, même si vous l'échappez :</p> <ul style="list-style-type: none"> <li>o <b>valide</b> : 'var &lt;- "\test\''</li> <li>o <b>non valide</b> : 'var &lt;- "\test\''</li> </ul>
<i>longueur</i>	numérique	La longueur à allouer pour la chaîne de retour.
<i>champ   valeur</i> <;... <i>n</i> > optionnel	caractère numérique DateHeure logique	<p>La liste des champs, des expressions ou des valeurs littérales à utiliser comme arguments pour le snippet de code ou le script R.</p> <p>Les valeurs sont transmises dans la fonction que vous appelez dans l'ordre dans lequel vous les indiquez, puis vous y référez à l'aide de valeur1; valeur2 ... valeurN dans le code R.</p> <p>Vous pouvez inclure autant d'arguments que nécessaires pour répondre à la définition de la fonction dans le code R.</p> <p><b>Remarque</b></p> <p>La fonction ALLTRIM() sert à supprimer les espaces de début ou de fin de l'entrée de caractères : ALLTRIM(str). Pour plus d'informations, consultez la section "Fonction ALLTRIM( )" Page 493.</p>

## Sortie

Caractère.

# Exemples

## Exemples basiques

Renvoie "abc123" :

```
RSTRING("print(paste(valeur1;valeur2;sep=''));6;"abc";"123")
```

## Exemples avancés

### Utilisation d'un script R externe

Concatène x et y dans une seule chaîne délimitée par un caractère espace :

```
RSTRING("a<-source('./sample.r');a[[1]]"; 50; Prénom; Nom)
```

Script R externe (sample.r) :

```
conc <- function(x, y) {
  paste(x, y, sep=" ")
}
print(conc(value1, value2))
```

### Utilisation du code R stocké dans une variable

Concatène x et y dans une seule chaîne délimitée par un caractère espace :

```
ASSIGN v_script = "conc <- function(x, y){paste(x, y, sep=' ')};conc(value1, value2)"
RSTRING(v_script; 50; Prénom; Nom)
```

### Utilisation de R pour générer un UUID pour une table

Vous préparez une table d'exceptions à charger dans Résultats et il vous faut un identifiant unique garanti pour chaque enregistrement. Pour générer ce champ, vous utilisez le package **uuid** dans R afin de créer une valeur de clé primaire unique pour chaque enregistrement :

```
EXTRACT RSTRING("uuid::UUIDgenerate()"; 36) AS "id"; prénom; nom; date_anniversaire TO
table_exportation
```



### Astuce

Pour installer le package `uuid`, ouvrez `R.exe` et exécutez la commande suivante :

```
install.packages("uuid")
```

## Remarques

### Renvoi de données à partir de R

Lorsque vous appelez des scripts R, utilisez la fonction `source` et affectez l'objet renvoyé à une variable. Vous pouvez alors accéder à la valeur renvoyée depuis votre fonction R à partir de l'objet renvoyé :

```
# 'a' contient l'objet de réponse et a[[1]] accède à la valeur des données  
"a<-source('c:\\scripts\\r_scripts\\sample.r');a[[1]]"
```

### Fichier trace R

Analytics enregistre les messages en langage R dans un fichier `aclrlang.log` dans le dossier de projet. Utilisez ce fichier trace pour le débogage des erreurs R

#### Astuce

Ce fichier trace se trouve dans le dossier de résultats des tâches analytiques d'Analytics Exchange.

## Exécution de scripts R externes sur AX Serveur

Si vous rédigez une App Analyse à exécuter sur AX Serveur et que vous souhaitez travailler sur des scripts R externes :

1. Chargez le fichier sous forme de fichier associé à l'App Analyse.
2. Utilisez la balise d'analyse `FILE` pour identifier le(s) fichier(s).
3. Faites référence au(x) fichier(s) à l'aide du chemin d'accès relatif `./nom_fichier.r`.

#### Remarque

L'utilisation d'un fichier associé garantit que le compte du serveur d'application TomEE dispose des autorisations suffisantes pour accéder au fichier lors de l'exécution de R avec Analytics Exchange.

# Fonction RTIME( )

Renvoie une valeur heure calculée par un script ou une fonction de Rvachev. Le traitement des données dans Rvachev est externe à Analytics.

## Syntaxe

```
RTIME(rScript|rCode <;champ|valeur <;...n>>)
```

## Paramètres

Nom	Type	Description
<i>rScript</i>   <i>rCode</i>	caractère	<p>Le chemin d'accès complet ou relatif au script R ou à un snippet de code R à exécuter.</p> <p>Si vous saisissez le code R directement plutôt que d'utiliser un fichier externe, vous ne pouvez pas utiliser le caractère de guillemet fermant, même si vous l'échappez :</p> <ul style="list-style-type: none"> <li>◦ <b>valide</b> : 'var &lt;- "\"test\""'</li> <li>◦ <b>non valide</b> : 'var &lt;- "\"test\"'</li> </ul>
<i>champ</i>   <i>valeur</i> <;... <i>n</i> > optionnel	caractère numérique DateHeure logique	<p>La liste des champs, des expressions ou des valeurs littérales à utiliser comme arguments pour le snippet de code ou le script R.</p> <p>Les valeurs sont transmises dans la fonction que vous appelez dans l'ordre dans lequel vous les indiquez, puis vous y référez à l'aide de valeur1; valeur2 ... valeurN dans le code R.</p> <p>Vous pouvez inclure autant d'arguments que nécessaires pour répondre à la définition de la fonction dans le code R.</p> <p><b>Remarque</b></p> <p>La fonction ALLTRIM() sert à supprimer les espaces de début ou de fin de l'entrée de caractères : ALLTRIM(str). Pour plus d'informations, consultez la section "Fonction ALLTRIM( )" Page 493.</p>

## Sortie

DateHeure.

# Exemples

## Exemples basiques

Renvoie `t0545` :

```
RTIME("valeur1+2700";`t0500`)
```

## Exemples avancés

### Utilisation d'un script R externe

Ajoute 45 minutes à un champ heure en transmettant un champ et une valeur littérale à une fonction R externe :

```
RTIME("a<-source('c:\\scripts\\sample.r');a[[1]]"; end_time; 2700)
```

Script R externe (sample.r) :

```
add_time <- function(start, sec) {  
  return(start + sec)  
}  
add_time(value1, value2)
```

## Remarques

### Renvoi de données à partir de R

Lorsque vous appelez des scripts R, utilisez la fonction `source` et affectez l'objet renvoyé à une variable. Vous pouvez alors accéder à la valeur renvoyée depuis votre fonction R à partir de l'objet renvoyé :

```
# 'a' contient l'objet de réponse et a[[1]] accède à la valeur des données  
"a<-source('c:\\scripts\\r_scripts\\sample.r');a[[1]]"
```

### Fichier trace R

Analytics enregistre les messages en langage R dans un fichier `aclr_lang.log` dans le dossier de projet. Utilisez ce fichier trace pour le débogage des erreurs R

### Astuce

Ce fichier trace se trouve dans le dossier de résultats des tâches analytiques d'Analytics Exchange.

## Exécution de scripts R externes sur AX Serveur

Si vous rédigez une App Analyse à exécuter sur AX Serveur et que vous souhaitez travailler sur des scripts R externes :

1. Chargez le fichier sous forme de fichier associé à l'App Analyse.
2. Utilisez la balise d'analyse FILE pour identifier le(s) fichier(s).
3. Faites référence au(x) fichier(s) à l'aide du chemin d'accès relatif `./nom_fichier.r`.

### Remarque

L'utilisation d'un fichier associé garantit que le compte du serveur d'application TomEE dispose des autorisations suffisantes pour accéder au fichier lors de l'exécution de R avec Analytics Exchange.

## Fuseau horaire du système

L'heure moyenne de Greenwich (GMT) est le fuseau horaire actif par défaut dans l'environnement R utilisé par Analytics.

# Fonction SECOND()

Extrait les secondes d'une heure ou d'une DateHeure spécifiée et les renvoie sous forme de valeur numérique.

## Syntaxe

```
SECOND(heure/DateHeure)
```

## Paramètres

Nom	Type	Description
<i>heure/DateHeure</i>	DateHeure	Le champ, l'expression, ou la valeur littérale à partir desquels extraire les secondes.

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 30 :

```
SECOND('t235930')
```

```
SECOND('20141231 235930')
```

Renvoie les secondes pour chaque valeur dans le champ **Appel\_heure\_début** :

```
SECOND(Appel_heure_début)
```

# Remarques

## Informations sur les paramètres

Un champ indiqué pour *heure/DateHeure* peut utiliser n'importe quel format d'heure ou de DateHeure, tant que la définition de champ définit correctement le format.

### Spécification d'une valeur littérale d'heure ou de DateHeure

Lorsque vous indiquez une valeur littérale d'heure ou de DateHeure pour *heure/DateHeure*, vous êtes limité aux formats de la table ci-dessous, et vous devez entourer la valeur de guillemets inversés - par exemple, `20141231 235959`.

N'utilisez aucun séparateur comme les barres obliques (/) ou les deux points (:) entre les composants individuels des dates ou des heures.

- **Valeurs heure** : vous pouvez utiliser n'importe quel format d'heure répertorié dans le tableau ci-dessous. Vous devez utiliser un séparateur avant une valeur d'heure autonome pour que la fonction s'exécute correctement. Les séparateurs valides sont la lettre « t » ou la lettre « T ». Vous devez indiquer les heures à l'aide de l'horloge de 24 heures. Les décalages par rapport à l'heure universelle coordonnée (UTC) doivent être préfacés d'un signe plus (+) ou d'un signe moins (-).
- **Valeurs DateHeure** : vous pouvez utiliser n'importe laquelle des combinaisons de formats de date, de séparateur et d'heure répertoriées dans le tableau ci-dessous. La date doit précéder l'heure, et vous devez utiliser un séparateur entre les deux. Les séparateurs valides sont un espace unique, la lettre « t » ou la lettre « T ».

Exemples de formats	Exemples de valeurs littérales
thhmmss	`t235959`
Thhmm	`T2359`
AAAAMMJJ hhmmss	`20141231 235959`
AAMMJJthhmm	`141231t2359`
AAAAMMJJThh	`20141231T23`
AAAAMMJJ hhmmss+/-hhmm (décalage UTC)	`20141231 235959-0500`
AAMMJJ hhmm+/-hh (décalage UTC)	`141231 2359+01`

Exemples de formats	Exemples de valeurs littérales
<p><b>Remarque</b></p> <p>N'utilisez pas hh seul dans le format d'heure principal avec des données qui ont un décalage UTC. Par exemple, évitez : hh+hhmm. Il se peut que les résultats ne soient pas fiables.</p>	

# Fonction SHIFT()

Renvoie une chaîne de caractères unique dont les bits issus du premier caractère de la valeur d'entrée ont été déplacés vers la gauche ou vers la droite.

## Syntaxe

```
SHIFT(caractère; nombre_de_bits_à_gauche)
```

## Paramètres

Nom	Type	Description
<i>caractère</i>	caractère	Valeur dont les bits doivent être déplacés.
<i>nombre_de_bits_à_gauche</i>	numérique	Indique le nombre de bits dont la valeur <i>caractère</i> sera déplacée. <ul style="list-style-type: none"> <li>Si la valeur est positive : le <i>caractère</i> est déplacé à gauche</li> <li>Si la valeur est négative : le <i>caractère</i> est déplacé à droite</li> </ul> Si la valeur spécifiée est supérieure à 15 ou inférieure à -15, le résultat est un zéro binaire, CHR(0).

## Sortie

Caractère.

## Exemples

### Exemples basiques

Renvoie la lettre « X » ou CHR(88) (00010110 devient 01011000) :

```
SHIFT(CHR(22); 2)
```

Renvoie le caractère retour arrière ou CHR(8) (00010000 devient 00001000) :

```
SHIFT(CHR(16); -1)
```

Renvoie le caractère accent grave ou CHR(96) (10011011 devient 01100000) :



```
SHIFT(CHR(155); 5)
```

## Remarques

### Quand utiliser SHIFT( )

Utilisez la fonction SHIFT( ) conjointement aux fonctions BYTE( ), CHR( ) et MASK( ) pour isoler et déplacer des bits individuels dans un enregistrement.

# Fonction SIN( )

Renvoie le sinus d'un angle exprimé en radians, avec une précision de 15 décimales.

## Syntaxe

```
SIN(radians)
```

## Paramètres

Nom	Type	Description
<i>radians</i>	numérique	Mesure de l'angle en radians.

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 0,5000000000000000 (le sinus du nombre de *radians* spécifié, équivalent à 30 degrés) :

```
SIN(0,523598775598299)
```

Renvoie 0,5000000000000000 (le sinus de 30 degrés) :

```
SIN(30 * PI( )/180)
```

### Exemples avancés

#### Utilisation d'une valeur d'entrée en degrés

Renvoie 0,500 (le sinus de 30 degrés, arrondi à 3 décimales) :

```
DEC(SIN(30 * PI()/180);3)
```

## Remarques

### Réalisation du test de l'arc mantisse

Les trois fonctions trigonométriques d'Analytics (SIN(), COS() et TAN()) prennent en charge le test de l'arc de mantisse associé à la loi de Benford.

### Conversion de degrés en radians

Si votre entrée est en degrés, vous pouvez utiliser la fonction PI() pour la convertir en radians :  $(\text{degrés} * \text{PI}() / 180) = \text{radians}$ . Si besoin, vous pouvez arrondir ou tronquer la valeur renvoyée à l'aide de la fonction DEC().

# Fonction SOUNDEX( )

Renvoie le code soundex pour la chaîne indiquée. Ce dernier peut être utilisé pour effectuer des comparaisons phonétiques avec d'autres chaînes.

## Syntaxe

```
SOUNDEX(chaîne_de_caractères)
```

## Paramètres

Nom	Type	Description
<i>nom</i>	caractère	Expression de caractère à évaluer.

## Sortie

Caractère. Renvoie un code soundex à quatre caractères.

## Exemples

### Exemples basiques

#### Mots ayant la même prononciation mais une orthographe différente

Les deux exemples ci-dessous renvoient le même code soundex car ces deux mots se prononcent pareil même si leur orthographe est différente.

Renvoie F634 :

```
SOUNDEX("Fairdale")
```

Renvoie F634 :

```
SOUNDEX("Faredale")
```

## Mots ayant une prononciation similaire

Les deux exemples ci-dessous renvoient des codes soundex différents, mais très proches, car les deux mots ont une prononciation similaire.

Renvoie J525 :

```
SOUNDEX("Jonson")
```

Renvoie J523 :

```
SOUNDEX("Jonston")
```

## Mots ayant une prononciation différente

Les deux exemples ci-dessous renvoient des codes soundex assez différents, car les deux mots ont une prononciation différente.

Renvoie S530 :

```
SOUNDEX("Smith")
```

Renvoie M235 :

```
SOUNDEX("MacDonald")
```

## Entrée de champ

Renvoie le code soundex de chaque valeur du champ **Nom** :

```
SOUNDEX(Nom)
```

## Exemples avancés

### Identifier les codes soundex correspondants

Créez le champ calculé **Code\_soundex** pour afficher le code soundex de chaque valeur dans le champ **Nom** :

```
DEFINE FIELD Code_soundex COMPUTED SOUNDEX(Nom)
```

Ajoutez le champ calculé **Code\_soundex** à la vue, puis vérifiez la présence de doublons dans ce même champ en effectuant une recherche de doublons afin d'identifier toute correspondance de codes soundex :

```
DUPLICATES ON Code_soundex OTHER Nom PRESORT OPEN TO "Doublons_possibles.fil"
```

Si vous trouvez des correspondances de codes soundex, cela indique que les valeurs de caractère associées du champ **Nom** sont peut-être des doublons.

## Remarques

### Quand utiliser SOUNDEX( )

Utilisez la fonction SOUNDEX( ) pour rechercher des valeurs qui peuvent sembler similaires. La ressemblance phonétique constitue une façon de repérer les valeurs en double ou de trouver des fautes d'orthographe dans les données manuellement saisies.

### Fonctionnement

La fonction SOUNDEX( ) renvoie le code American Soundex correspondant à la chaîne évaluée. Tous les codes sont composés d'une lettre suivie de trois chiffres. Par exemple : "F634".

### Construction du code soundex

- Le premier caractère du code représente la première lettre de la chaîne évaluée.
- Chaque nombre du code représente l'un des six groupes American Soundex. Ces derniers sont composés des consonnes phonétiquement similaires.

En fonction de ces groupes, le processus soundex encode les trois premières consonnes de la chaîne évaluée, se trouvant après la première lettre.

### Ce que le processus soundex ignore

Le processus soundex ignore :

- les lettres majuscules
- les voyelles
- les consonnes "H", "W" et "Y"
- les consonnes qui s'affichent après les trois consonnes codées

Si le code renvoyé présente un ou plusieurs zéros (0) de fin, cela indique que la chaîne évaluée contient moins de trois consonnes après la première lettre.

### Limites du processus soundex

Les fonctions SOUNDSLIKE( ) et SOUNDEX( ) ont certaines limitations :

- L'algorithme soundex est conçu pour fonctionner avec des mots prononcés en anglais, et son efficacité est variable lorsqu'il est utilisé avec d'autres langues.

- Bien que le processus soundex réalise une correspondance phonétique, les mots correspondants doivent tous commencer par la même lettre, ce qui signifie que certains mots qui produisent le même son ne sont pas considérés comme correspondants.

Par exemple, un mot commençant par un "F" et un mot commençant par "Ph" pourraient produire le même son mais ils ne seront jamais considérés comme correspondants.

## Fonctions associées

- La fonction **SOUNDSLIKE( )** : fournit une autre méthode de comparaison phonétique de chaînes.
- **Les fonctions ISFUZZYDUP( ) et LEVDIST( )** : comparent des chaînes de façon orthographique plutôt que phonétique (son).
- La fonction **DICECOEFFICIENT( )** : ne met pas l'accent sur la position des caractères ou des blocs de caractères voire les ignore complètement lors de la comparaison de chaînes de caractères.

# Fonction SOUNDSLIKE()

Renvoie une valeur logique indiquant si une chaîne correspond phonétiquement à une chaîne de comparaison.

## Syntaxe

```
SOUNDSLIKE(chaîne_de_caractères; chaîne_à_comparer)
```

## Paramètres

Nom	Type	Description
<i>nom</i>	caractère	Première chaîne de la comparaison.
<i>chaîne_à_comparer</i>	caractère	Deuxième chaîne de la comparaison.

## Sortie

Logique. Renvoie **T** (« true », vrai) si les valeurs comparées présentent une correspondance phonétique et **F** (« false », faux) dans les autres cas.

## Exemples

### Exemples basiques

Renvoie T, car « Fairdale » et « Faredale » partagent tous deux le code soundex F634 :

```
SOUNDSLIKE("Fairdale";"Faredale")
```

Renvoie F, car le code soundex de « Jonson » est J525 et celui de « Jonston » est J523 :

```
SOUNDSLIKE("Jonson";"Jonston")
```

Renvoie une valeur logique (T ou F) indiquant si le code soundex de chaque valeur du champ **Nom** correspond au code soundex de la chaîne « Smith » :



```
SOUNDSLIKE(Nom;"Smith")
```

## Exemples avancés

### Isoler des valeurs dont la transcription phonétique est semblable à "Smith"

Créez un filtre isolant toutes les valeurs du champ **Nom** dont la transcription phonétique est semblable à celle de « Smith » :

```
SET FILTER TO SOUNDSLIKE(Nom;"Smith")
```

## Remarques

### Quand utiliser la fonction SOUNDSLIKE( )

Utilisez la fonction SOUNDSLIKE( ) pour rechercher des valeurs qui peuvent sembler similaires. La ressemblance phonétique constitue une façon de repérer les valeurs en double ou de trouver des fautes d'orthographe dans les données manuellement saisies.

### Fonctionnement

La fonction SOUNDSLIKE( ) convertit d'abord les chaînes de la comparaison en codes American Soundex à quatre caractères. Ces derniers sont basés sur la première lettre et les trois premières consonnes lui faisant suite dans chaque chaîne de caractères.

La fonction compare ensuite chaque code de chaîne de caractères et renvoie une valeur logique indiquant s'ils présentent des correspondances.

Pour obtenir des informations supplémentaires sur les codes soundex, consultez "Fonction SOUNDEX( )" Page 812.

### Respect de la casse

Cette fonction n'est pas sensible à la casse (par ex., « MARTIN » est équivalent à « martin »).

### Limites du processus soundex

Les fonctions SOUNDSLIKE( ) et SOUNDEX( ) ont certaines limitations :

- L'algorithme soundex est conçu pour fonctionner avec des mots prononcés en anglais, et son efficacité est variable lorsqu'il est utilisé avec d'autres langues.
- Bien que le processus soundex réalise une correspondance phonétique, les mots correspondants doivent tous commencer par la même lettre, ce qui signifie que certains mots qui produisent le même son ne sont pas considérés comme correspondants.

Par exemple, un mot commençant par un "F" et un mot commençant par "Ph" pourraient produire le même son mais ils ne seront jamais considérés comme correspondants.

## Fonctions associées

- La fonction **SOUNDEX()** : fournit une autre méthode de comparaison phonétique de chaînes.
- **Les fonctions ISFUZZYDUP() et LEVDIST()** : comparent des chaînes de façon orthographique plutôt que phonétique (son).
- La fonction **DICECOEFFICIENT()** : ne met pas l'accent sur la position des caractères ou des blocs de caractères voire les ignore complètement lors de la comparaison de chaînes de caractères.

# Fonction SPLIT()

Renvoie un segment spécifié d'une chaîne de caractères.

## Syntaxe

```
SPLIT(chaîne_de_caractères; séparateur; segment <;qualificatif_texte>)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	Valeur depuis laquelle extraire le segment.
<i>séparateur</i>	caractère	Caractère(s) délimitant les segments. Pour plus d'informations, consultez la section "Fonctionnement du caractère de séparation" Page 821.
<i>segment</i>	numérique	Segment à extraire. Utilisez un nombre pour préciser le segment à extraire. Par exemple, pour extraire le troisième segment, précisez 3.
<i>qualificatif_texte</i> optionnel	caractère	Caractère(s) indiquant le début et la fin des segments du texte. Si le caractère <i>séparateur</i> se trouve à l'intérieur d'une paire de qualificatifs de texte, il sera lu comme texte et non comme séparateur. Le <i>qualificatif de texte</i> doit être placé entre guillemets. Les <i>qualificatifs de texte</i> guillemets simples doivent être inclus entre guillemets doubles et les <i>qualificatifs de texte</i> guillemets doubles doivent être inclus entre guillemets simples.  <div style="border-left: 2px solid green; padding-left: 10px; margin-left: 20px;"> <p><b>Astuce</b></p> <p>Le paramètre optionnel peut être pratique lorsque vous travaillez avec des données source importées contenant des séparateurs et des qualificatifs de texte.</p> </div>

## Sortie

Caractère.

# Exemples

## Exemples basiques

### Segments délimités par des virgules

Renvoie "seg1" :

```
SPLIT("seg1 ; seg2 ;seg3" ; " ; " ; 1)
```

Renvoie "seg3" :

```
SPLIT("seg1,seg2,seg3" ; " ; " ; 3)
```

Renvoie "" (le troisième segment est vide) :

```
SPLIT("seg1,seg2,,seg4" ; " ; " ; 3)
```

### Délimiteurs d'espace et à caractères multiples

Renvoie "seg3" :

```
SPLIT("seg1/*seg2/*seg3" ; "/*" ; 3)
```

Renvoie "Doe" :

```
SPLIT("Jane Doe" ; " " ; 2)
```

### Échappement des délimiteurs avec un qualificatif de texte

Renvoie "Doe, Jane", incluant une virgule qui est lue comme partie intégrante du texte non comme un séparateur :

```
SPLIT("'Doe, Jane';'Smith, John'" ; " ; " ; 1 ; "'")
```

## Exemples avancés

### Extraire des chiffres à partir d'un numéro de carte de crédit

Utilisez la commande SPLIT() pour supprimer les tirets d'un numéro de carte de crédit.

Les variables sont utilisées pour capturer chaque segment du numéro de carte de crédit, et les segments sont ensuite concaténés dans une variable supplémentaire.

```
ASSIGN seg1 = SPLIT("4150-2222-3333-4444"; "-"; 1)
ASSIGN seg2 = SPLIT("4150-2222-3333-4444"; "-"; 2)
ASSIGN seg3 = SPLIT("4150-2222-3333-4444"; "-"; 3)
ASSIGN seg4 = SPLIT("4150-2222-3333-4444"; "-"; 4)
ASSIGN ccNum = seg1 + seg2 + seg3 + seg4
```

La valeur de ccNum est "4150222233334444".

L'exemple illustre l'usage de la fonction SPLIT(), mais remarquez que les tirets peuvent être plus efficacement supprimés à l'aide de la fonction EXCLUDE().

## Remarques

### Fonctionnement

La fonction SPLIT() divise les données caractère en segments en fonction des séparateurs (par exemple, les espaces ou les virgules), et renvoie un segment indiqué.

### Quand utiliser la fonction SPLIT()

Utilisez la fonction SPLIT() pour extraire un segment particulier de données d'un enregistrement ou d'un champ. Le segment doit apparaître à la même position dans chaque enregistrement ou champ.

### Fonctionnement du caractère de séparation

Le caractère de séparation délimite, ou indique, les segments de données dans une chaîne source.

Dans une chaîne contenant beaucoup de segments, ceux-ci apparaissent entre deux séparateurs. Cependant, il est possible que le premier segment ne soit pas précédé d'un caractère de séparation, et que le dernier segment n'en soit pas suivi d'un.

Par exemple :

```
SPLIT("seg1 ; seg2 ; seg3" ; " ;" ; 1)
```

Lorsque la chaîne source commence par un séparateur, le segment qui suit le séparateur est considéré comme étant le deuxième segment.

Renvoie "seg1" :

```
SPLIT(";seg1;seg2;seg3"; ";"; 2)
```

## Respect de la casse

Si les *séparateurs* ou les *qualificatifs de texte* précisent les caractères ayant une version majuscule et une minuscule, la casse utilisée doit correspondre à la casse du séparateur ou du qualificatif de texte dans les données.

## Fonctions associées

SPLIT() et SUBSTR() renvoient tous deux un segment de données d'une chaîne source plus longue.

- SPLIT() identifie le segment à partir d'un caractère de séparation.
- SUBSTR() identifie le segment à partir de la position d'un caractère numérique.

# Fonction STOD()

Convertit une date sérielle - c'est-à-dire, une date exprimée sous forme d'intègre - en valeur de date. Abréviatiion de "Sérielle à Date".

## Syntaxe

```
STOD(date_sérielle <;date_début>)
```

## Paramètres

Nom	Type	Description
<i>date_sérielle</i>	numérique	Le champ, l'expression, ou la valeur littérale à convertir. <i>date_sérielle</i> peut être une date sérielle ou une DateHeure sérielle. Seule la portion de la DateHeure sérielle correspondant à la date est prise en compte. La portion heure est ignorée.
<i>date_début</i> optionnel	DateHeure	La date de début à partir de laquelle les dates sérielles sont calculées. Si ce paramètre est ignoré, la date de début par défaut du 01 janvier 1900 est utilisée.

## Sortie

DateHeure. La valeur de date est générée à l'aide du format d'affichage de date Analytics actuel.

## Exemples

### Exemples basiques

Renvoie `20141231` affiché au format 31 déc 2014 en supposant que le format d'affichage actuel de la date dans Analytics soit JJ MMM AAAA :

```
STOD(42003)
```

Renvoie `20181231` affiché au format 31 déc 2018 en supposant que le format d'affichage actuel de la date dans Analytics soit JJ MMM AAAA :

```
STOD(42003; `19040101`)
```

Renvoie la date équivalente pour chaque valeur de date sérielle dans le champ **Date\_facture** :

```
STOD(Date_facture)
```

## Exemples avancés

### Ajustement d'une date de début antérieure au 01-01-1900

Utilisez une date arithmétique pour remplacer la date de début par une valeur antérieure à la date minimale d'Analytics du 1er janvier 1900 :

1. Convertissez la date sérielle à l'aide de la date de début par défaut.
2. Soustrayez le nombre de jours s'écoulant entre la date de début réelle et le 01-01-1900.

Pour utiliser 01-01-1899 comme date de début (évaluée sur `20131231`) :

```
STOD(42003) - 365
```

## Remarques

### Fonctionnement

La fonction STOD() vous permet de convertir des dates sérielles en dates ordinaires. Les dates sérielles d'Analytics représentent le nombre de jours qui se sont écoulés depuis le 01 janvier 1900.

Date sérielle	Équivalent en dates ordinaires
1	02 janvier 1900
365	31 décembre 1900
42003	31 décembre 2014
0	non valide

Pour plus d'informations sur les dates sérielles, consultez la section [DateHeures sérielles](#).



# Comparaisons entre les dates sérielles Analytics et les dates sérielles Excel

Les dates sérielles d'Analytics sont similaires aux dates sérielles de Microsoft Excel. Vous devez avoir conscience d'un point clé de similitude et d'un point clé de différence. Les deux points ne sont pas liés.

## Points communs

Analytics et Excel traitent tous les deux l'année 1900 comme une année bissextile, avec 366 jours. Bien que 1900 n'ait pas été en réalité une année bissextile, Excel l'a traitée ainsi afin de maintenir la compatibilité avec Lotus 1-2-3.

## Différences

Les dates sérielles d'Analytics sont calculées à partir des dates sérielles d'Excel par une journée. Dans Excel, le 01 janvier 1900 a une date sérielle de '1'. Dans Analytics, le 01 janvier 1900 n'est pas compté, et le 02 janvier 1900 a une date sérielle de '1'.

## *date\_début*

Certains fichiers de données source peuvent utiliser une date de début autre que le 01 janvier 1900. *date\_début* vous permet de faire correspondre la date de départ dans un fichier de données source. La date de début est la date à partir de laquelle les dates sérielles sont calculées.

Date de début dans le fichier de données source	Indiquez :	Détails
01 janvier 1900	STOD( <i>champ_date</i> )	Il n'est pas nécessaire d'indiquer une <i>date_début</i> , car 01 janvier 1900 est la date de début par défaut.
01 janvier 1901	STOD( <i>champ_date</i> ; `19010101`)	Vous spécifiez une <i>date_début</i> de `19010101` pour faire correspondre la date de début du 01 janvier 1901 utilisée dans le fichier de données source.
01 janvier 1899	STOD( <i>champ_date</i> ) - 365	Vous ne pouvez pas indiquer une <i>date_début</i> antérieure au 01 janvier 1900. Si un fichier de données source utilise une date de début antérieure au 01 janvier 1900, vous pouvez créer une expression DateHeure qui soustrait un nombre approprié de jours des résultats de sortie de la fonction STOD().

# Autres fonctions de conversion DateHeure

## Conversion Sérielle en DateHeure

Fonction	Description
<a href="#">STODT()</a>	Convertit une valeur DateHeure sérielle - c'est-à-dire, une DateHeure exprimée sous forme d'une intègre, et une portion fractionnelle de 24 heures - en une valeur correspondant à une DateHeure. Abréviation de "Sérielle à DateHeure".
<a href="#">STOT()</a>	Convertit une valeur horaire sérielle - c'est-à-dire une heure exprimée sous forme d'une portion fractionnelle de 24 heures, où 24 heures équivalent à 1 - en une valeur correspondant à une heure. Abréviation de "Sérielle à Heure".

## Conversion Caractère ou Numérique en DateHeure

Fonction	Description
<a href="#">CTOD()</a>	Convertit une valeur de date au format Caractère ou Numérique en date. Cette fonction peut également extraire la date d'une valeur de type DateHeure au format Caractère ou Numérique et la renvoyer sous forme de date. Abréviation de "Caractère à Date".
<a href="#">CTODT()</a>	Convertit une valeur DateHeure au format Caractère ou Numérique en DateHeure. Abréviation de "Caractère à DateHeure".
<a href="#">CTOT()</a>	Convertit une valeur de type heure au format Caractère ou Numérique en heure. Cette fonction peut également extraire l'heure d'une valeur de type DateHeure au format Caractère ou Numérique et la renvoyer sous forme d'heure. Abréviation de "Caractère à Heure".

## Conversion DateHeure en Caractère

Fonction	Description
<a href="#">DATE()</a>	Extrait la date d'une date ou d'une DateHeure spécifiée et la renvoie sous forme de chaîne de caractères. Peut également renvoyer la date du système d'exploitation en cours d'utilisation.
<a href="#">DATETIME()</a>	Convertit une DateHeure en une chaîne de caractères. Peut également renvoyer la DateHeure du système d'exploitation en cours d'utilisation.
<a href="#">TIME()</a>	Extrait l'heure d'une heure ou d'une DateHeure spécifiée et la renvoie sous forme de chaîne de caractères. Peut également renvoyer l'heure du système d'exploitation en cours d'utilisation.

# Fonction STODT()

Convertit une valeur DateHeure sérielle - c'est-à-dire, une DateHeure exprimée sous forme d'une intègre, et une portion fractionnelle de 24 heures - en une valeur correspondant à une DateHeure. Abréviation de "Sérielle à DateHeure".

## Syntaxe

```
STODT(DateHeure_sérielle <;date_début>)
```

## Paramètres

Nom	Type	Description
<i>DateHeure_sérielle</i>	numérique	Le champ, l'expression, ou la valeur littérale à convertir. Des valeurs de DateHeure sérielle avec les portions correspondant à la date et à l'heure séparées par une virgule de décimale sont requises. Par exemple, 42003,75000
<i>date_début</i> optionnel	DateHeure	La date de début à partir de laquelle les dates sérielles sont calculées. Si ce paramètre est ignoré, la date de début par défaut du 01 janvier 1900 est utilisée.

## Sortie

DateHeure. La valeur de DateHeure est générée à l'aide des formats d'affichage de date et d'heure Analytics actuels.

## Exemples

### Exemples basiques

#### Dates de début non ajustées

Renvoie `20141231t060000` affiché au format 31 déc 2014 06:00:00 AM en supposant que les formats d'affichage actuels de la date et de l'heure dans Analytics soient JJ MMM AAAA et hh:mm:ss PM :

```
STODT(42003,25000)
```

Renvoie `20141231t191530` affiché au format 31 déc 2014 07:15:30 AM en supposant que les formats d'affichage actuels de la date et de l'heure dans Analytics soient JJ MMM AAAA et hh:mm:ss PM :

```
STODT(42003,802431)
```

## Dates de début ajustées

Renvoie `20181231t120000` affiché au format 31 déc 2018 12:00:00 AM en supposant que les formats d'affichage actuels de la date et de l'heure dans Analytics soient JJ MMM AAAA et hh:mm:ss PM :

```
STODT(42003,50000; `19040101`)
```

## Champs comme valeurs d'entrée

Renvoie la DateHeure équivalente pour chaque valeur de DateHeure sérielle du champ **Reçu\_DateHeure** :

```
STODT(Reçu_DateHeure)
```

## Exemples avancés

### Ajustement d'une date de début antérieure au 01-01-1900

Utilisez une date arithmétique pour remplacer la date de début par une valeur antérieure à la date minimale d'Analytics du 1er janvier 1900 :

1. Convertissez la DateHeure sérielle à l'aide de la date de début par défaut.
2. Soustrayez le nombre de jours s'écoulant entre la date de début réelle et le 01-01-1900.

Pour utiliser 01-01-1899 comme date de début (évaluée sur `20131231t180000`) :

```
STODT(42003,75000) - 365
```

## Remarques

### Fonctionnement

La fonction STODT( ) vous permet de convertir des DateHeures sérielles en DateHeures ordinaires. Les DateHeures sérielles d'Analytics représentent le nombre de jours qui se sont écoulés depuis le 01 janvier 1900, et après la virgule de décimale, représentent une portion fractionnelle de 24 heures, où 24 heures équivalent à 1.

DateHeure sérielle	Équivalent en DateHeures ordinaires
1,25	02 janvier 1900 06:00:00 AM
365,75000	31 décembre 1900 06:00:00 PM
42003,79167	31 décembre 2014 07:00:00 PM
42003,802431	31 décembre 2014 07:15:30 PM
42003,00000	31 décembre 2014 12:00:00 AM
42003,50000	31 décembre 2014 12:00:00 PM
0,0	non valide

Pour plus d'informations sur les DateHeures sérielles, consultez la section [DateHeures sérielles](#).

## Comparaisons entre les dates sérielles Analytics et les dates sérielles Excel

Les dates sérielles d'Analytics sont similaires aux dates sérielles de Microsoft Excel. Vous devez avoir conscience d'un point clé de similitude et d'un point clé de différence. Les deux points ne sont pas liés.

### Points communs

Analytics et Excel traitent tous les deux l'année 1900 comme une année bissextile, avec 366 jours. Bien que 1900 n'ait pas été en réalité une année bissextile, Excel l'a traitée ainsi afin de maintenir la compatibilité avec Lotus 1-2-3.

### Différences

Les dates sérielles d'Analytics sont calculées à partir des dates sérielles d'Excel par une journée. Dans Excel, le 01 janvier 1900 a une date sérielle de '1'. Dans Analytics, le 01 janvier 1900 n'est pas compté, et le 02 janvier 1900 a une date sérielle de '1'.

### *date\_début*

Certains fichiers de données source peuvent utiliser une date de début autre que le 01 janvier 1900. *date\_début* vous permet de faire correspondre la date de départ dans un fichier de données source. La date de début est la date à partir de laquelle les DateHeures sérielles sont calculées.

Date de début dans le fichier de données source	Indiquez :	Détails
01 janvier 1900	STODT( <i>champ_dateheure</i> )	Il n'est pas nécessaire d'indiquer une <i>date_début</i> , car 01 janvier 1900 est la date de début par défaut.
01 janvier 1901	STODT( <i>champ_dateheure</i> ; `19010101`)	Vous spécifiez une <i>date_début</i> de `19010101` pour faire correspondre la date de début du 01 janvier 1901 utilisée dans le fichier de données source.
01 janvier 1899	STODT( <i>champ_dateheure</i> ) - 365	Vous ne pouvez pas indiquer une <i>date_début</i> antérieure au 01 janvier 1900. Si un fichier de données source utilise une date de début antérieure au 01 janvier 1900, vous pouvez créer une expression DateHeure qui soustrait un nombre approprié de jours des résultats de sortie de la fonction STODT( ).

## Autres fonctions de conversion DateHeure

### Conversion Sérielle en DateHeure

Fonction	Description
<a href="#">STOD()</a>	Convertit une date sérielle - c'est-à-dire, une date exprimée sous forme d'intègre - en valeur de date. Abréviation de "Sérielle à Date".
<a href="#">STOT()</a>	Convertit une valeur horaire sérielle - c'est-à-dire une heure exprimée sous forme d'une portion fractionnelle de 24 heures, où 24 heures équivalent à 1 - en une valeur correspondant à une heure. Abréviation de "Sérielle à Heure".

### Conversion Caractère ou Numérique en DateHeure

Fonction	Description
<a href="#">CTOD()</a>	Convertit une valeur de date au format Caractère ou Numérique en date. Cette fonction peut également extraire la date d'une valeur de type DateHeure au format Caractère ou Numérique et la renvoyer sous forme de date. Abréviation de "Caractère à Date".
<a href="#">CTODT()</a>	Convertit une valeur DateHeure au format Caractère ou Numérique en DateHeure. Abréviation de "Caractère à DateHeure".
<a href="#">CTOT()</a>	Convertit une valeur de type heure au format Caractère ou Numérique en heure. Cette fonction peut également extraire l'heure d'une valeur de type DateHeure au format Caractère ou Numérique et la renvoyer sous forme d'heure. Abréviation de "Caractère à Heure".

## Conversion DateHeure en Caractère

Fonction	Description
<a href="#">DATE()</a>	Extrait la date d'une date ou d'une DateHeure spécifiée et la renvoie sous forme de chaîne de caractères. Peut également renvoyer la date du système d'exploitation en cours d'utilisation.
<a href="#">DATETIME()</a>	Convertit une DateHeure en une chaîne de caractères. Peut également renvoyer la DateHeure du système d'exploitation en cours d'utilisation.
<a href="#">TIME()</a>	Extrait l'heure d'une heure ou d'une DateHeure spécifiée et la renvoie sous forme de chaîne de caractères. Peut également renvoyer l'heure du système d'exploitation en cours d'utilisation.

# Fonction STOT()

Convertit une valeur horaire sérielle - c'est-à-dire une heure exprimée sous forme d'une portion fractionnelle de 24 heures, où 24 heures équivalent à 1 - en une valeur correspondant à une heure. Abréviation de "Sérielle à Heure".

## Syntaxe

```
STOT(heure_sérielle)
```

## Paramètres

Nom	Type	Description
<i>heure_sérielle</i>	numérique	Le champ, l'expression, ou la valeur littérale à convertir. <i>heure_sérielle</i> peut être une heure sérielle ou une DateHeure sérielle. Seule la portion de la DateHeure sérielle correspondant à l'heure est prise en compte. La portion correspondant à la date est ignorée.

## Sortie

DateHeure. La valeur de type heure est générée à l'aide du format d'affichage de l'heure Analytics actuel.

## Exemples

### Exemples basiques

Renvoie `t060000` affiché sous la forme 06:00:00 AM en supposant que le format d'affichage actuel de l'heure dans Analytics soit hh:mm:ss PM :

```
STOT(0,25000)
```

Renvoie `t191530` affiché sous la forme 07:15:30 PM en supposant que le format d'affichage actuel de l'heure dans Analytics soit hh:mm:ss PM :

```
STOT(0,802431)
```



Renvoie l'heure ordinaire équivalente à chaque valeur d'heure sérielle du champ **Heure\_connexion** :

```
STOT(Heure_connexion)
```

## Remarques

### Quand utiliser la fonction STOT( )

Utilisez la fonction STOT( ) pour convertir des heures sérielles en heures ordinaires.

### Qu'est-ce qu'une heure sérielle ?

Les heures sérielles d'Analytics représentent une fraction de 24 heures, où 24 heures équivalent à 1.

Par exemple :

- l'équivalent d'1 heure en heure sérielle est 1/24, ou 0,04167.
- l'équivalent en heure sérielle d'1 minute est 1/1440, ou 0,0006945

Les heures sérielles peuvent être préfacées d'un '0' (zéro) et d'une virgule de décimale, ou juste d'une virgule de décimale.

### 1,000000 n'est pas une heure sérielle valide

Bien que 24 heures équivalent à 1 à des fins de calcul de sheures sérielles, 1,000000 n'est pas une heure sérielle valide. Les heures sérielles valides sont toutes des fractions décimales inférieures à 1. Par exemple : 0,75000 (06:00:00 PM).

Analytics traie le chiffre sériel 1,000000 comme l'équivalent DateHeure sériel de 02 jan 1900 12:00:00 AM. La fonction STOT( ) ignorant la portion correspondant à la date des DateHeures, STOT(1,000000) équivaut à STOT(0,000000) et les deux équivalent à l'heure ordinaire 12:00:00 AM.

### Heures sérielles et équivalents en heure ordinaire

Heure sérielle	Équivalent en heures ordinaires
0,00	12:00:00 AM
0,0006945	12:01:00 AM
0,04167	01:00:00 AM
0,0423645	01:01:00 AM
0,042998	01:01:55 AM

Heure sérielle	Équivalent en heures ordinaires
0,25	06:00:00 AM
0,50	12:00:00 PM
0,75	06:00:00 PM
0,79167	07:00:00 PM
0,802431	07:15:30 PM
1,00	12:00:00 AM

## Autres fonctions de conversion DateHeure

### Conversion Sérielle en DateHeure

Fonction	Description
<a href="#">STOD()</a>	Convertit une date sérielle - c'est-à-dire, une date exprimée sous forme d'intègre - en valeur de date. Abréviation de "Sérielle à Date".
<a href="#">STODT()</a>	Convertit une valeur DateHeure sérielle - c'est-à-dire, une DateHeure exprimée sous forme d'une intègre, et une portion fractionnelle de 24 heures - en une valeur correspondant à une DateHeure. Abréviation de "Sérielle à DateHeure".

### Conversion Caractère ou Numérique en DateHeure

Fonction	Description
<a href="#">CTOD()</a>	Convertit une valeur de date au format Caractère ou Numérique en date. Cette fonction peut également extraire la date d'une valeur de type DateHeure au format Caractère ou Numérique et la renvoyer sous forme de date. Abréviation de "Caractère à Date".
<a href="#">CTODT()</a>	Convertit une valeur DateHeure au format Caractère ou Numérique en DateHeure. Abréviation de "Caractère à DateHeure".
<a href="#">CTOT()</a>	Convertit une valeur de type heure au format Caractère ou Numérique en heure. Cette fonction peut également extraire l'heure d'une valeur de type DateHeure au format Caractère ou Numérique et la renvoyer sous forme d'heure. Abréviation de "Caractère à Heure".

## Conversion DateHeure en Caractère

Fonction	Description
<a href="#">DATE()</a>	Extrait la date d'une date ou d'une DateHeure spécifiée et la renvoie sous forme de chaîne de caractères. Peut également renvoyer la date du système d'exploitation en cours d'utilisation.
<a href="#">DATETIME()</a>	Convertit une DateHeure en une chaîne de caractères. Peut également renvoyer la DateHeure du système d'exploitation en cours d'utilisation.
<a href="#">TIME()</a>	Extrait l'heure d'une heure ou d'une DateHeure spécifiée et la renvoie sous forme de chaîne de caractères. Peut également renvoyer l'heure du système d'exploitation en cours d'utilisation.

# Fonction STRING( )

Convertit une valeur numérique en chaîne de caractères.

## Syntaxe

```
STRING(numéro; longueur <;format>)
```

## Paramètres

Nom	Type	Description
<i>nombre</i>	numérique	Valeur numérique à convertir en chaîne.
<i>longueur</i>	numérique	Nombre de caractères dans la chaîne de sortie.
<i>format</i> optionnel	caractère	Format à appliquer à la chaîne de sortie. Par exemple, "(9999,99)"

## Sortie

Caractère.

## Exemples

### Exemples basiques

#### Chaînes sans mise en forme

Renvoie " 125,2" :

```
STRING(125,2; 6)
```

Renvoie "25,2" (-1 est tronqué car le paramètre *longueur* est inférieur au nombre de chiffres et de caractères de formatage dans le paramètre *numéro*) :

```
STRING(-125,2; 4)
```

Renvoie " 125,2" :

```
STRING(-125,2; 7)
```

## Chaînes mises en forme

Renvoie " (125,20)" :

```
STRING(-125,2; 10; "(9999,99)")
```

Renvoie "25,20" (1 est tronqué car le paramètre *longueur* est inférieur au nombre de chiffres et de caractères de formatage dans le paramètre *numéro*) :

```
STRING(125,2; 6; "(9999,99)")
```

## Entrée de champ

Renvoie des valeurs numériques dans le champ **Numéro\_employé** sous forme de chaînes de caractères d'une longueur de 10 caractères. Si besoin, la valeur renvoyée est complétée ou tronquée :

```
STRING(Numéro_employé; 10)
```

# Remarques

## Valeurs renvoyées remplacées et tronquées

La fonction `STRING( )` convertit *numéro* en chaîne de caractères en adoptant la longueur indiquée dans *longueur* :

- Si *numéro* est inférieur à *longueur*, des espaces de début sont ajoutés à la valeur renvoyée
- Si *numéro* est supérieur à *longueur*, la valeur renvoyée est tronquée à partir de la gauche

## Mise en forme de la valeur renvoyée

Le paramètre *format* facultatif formate la valeur renvoyée en ajoutant le symbole du dollar, le symbole de pourcentage, des décimales, des virgules, des indicateurs négatifs ou des parenthèses. Le *format* doit être placé entre guillemets doubles.

Le chiffre 9 joue le rôle de paramètre fictif pour les chiffres à formater. Assurez-vous que le nombre correct de 9 s'affiche. Vous devez également prendre en compte les décimales et les caractères de formatage (par exemple, le signe du dollar et les parenthèses pour les nombres négatifs) lorsque vous indiquez la valeur de la *longueur*.

## Fonctions associées

La fonction `STRING()` est l'opposé de la fonction `VALUE()`, qui convertit des données caractère en données numériques.

# Fonction SUBSTR()

Renvoie une sous-chaîne indiquée d'une chaîne de caractères.

## Syntaxe

```
SUBSTR(chaîne_de_caractères; début; longueur)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	Valeur à partir de laquelle extraire la sous-chaîne.
<i>début</i>	numérique	Position de départ des caractères de la sous-chaîne.
<i>longueur</i>	numérique	Nombre de caractères dans la sous-chaîne. Si la <i>longueur</i> équivaut à 0, le résultat est nul.

## Sortie

Caractère.

## Exemples

### Exemples basiques

#### Valeur d'entrée littérale de type caractère

Renvoie "BCD" :

```
SUBSTR("ABCDEF"; 2; 3)
```

Renvoie "EF" :

```
SUBSTR("ABCDEF"; 5; 10)
```

## Analyse des données de type caractère structurées

Renvoie "189543" :

```
SUBSTR("****189543****"; 4; 6)
```

Renvoie l'année à quatre chiffres d'un champ de type caractère contenant des dates ayant le format "JJ/MM/AAAA" :

```
SUBSTR(DATE; 7; 4)
```

## Exemples avancés

### Augmentation de la longueur du champ

Utilisez SUBSTR( ) pour augmenter la longueur d'un champ de type caractère. Augmenter la longueur d'un champ est une tâche d'harmonisation courante à laquelle vous devrez peut-être vous livrer avant de joindre ou concaténer deux champs ensemble.

L'exemple ci-dessous remplit le champ **Description\_Produit** avec des espaces vides pour créer le champ calculé **Description\_Produit\_Long** d'une longueur de 50 caractères.

```
DEFINE FIELD Description_Produit_Long COMPUTED SUBSTR(Description_Produit; 1; 50)
```

## Remarques

### Fonctionnement

La fonction SUBSTR( ) renvoie des caractères de la valeur de la *chaîne* à partir de la position du caractère précisé par *start*. Le nombre de caractères renvoyé est spécifié par la *longueur*.

### Gestion des espaces par SUBSTR( )

Les espaces de début, de fin ou internes figurant dans la valeur de la *chaîne de caractères* sont traitées comme des caractères. Les espaces capturées par *début* et *longueur* sont incluses dans la chaîne de sortie.

### Fonctionnement du remplissage

Si la valeur *longueur* dépasse le nombre de caractères, y compris les espaces de fin, à partir de la position de *début* jusqu'à la fin de la valeur de la *chaîne de caractères*, le résultat peut être complété ou pas par des espaces.



## Rempli avec des espaces

Si vous utilisez SUBSTR() au sein d'une commande créant un champ, la sortie est remplie avec des espaces.

### Remplissage lors de la création du champ calculé

Crée le champ calculé **Description\_Produit\_Long**, avec une longueur de 50 caractères, basé sur le champ physique **Description\_Produit**, d'une longueur de 24 caractères :

```
DEFINE FIELD Description_Produit_Long COMPUTED SUBSTR(Description_Produit; 1; 50)
```

### Remplissage lors de l'extraction d'un champ physique

Extrait le champ **Description\_Produit\_Long**, avec une longueur de 50 caractères, dans une nouvelle table, basé sur le champ physique **Description\_Produit**, d'une longueur de 24 caractères :

```
EXTRACT FIELDS SUBSTR(Description_Produit; 1; 50) AS "Description_Produit_Long" TO Nouvelle_Table
```

## Pas rempli avec des espaces

Si vous utilisez SUBSTR() dans une définition variable ou dans une expression, la sortie n'est pas remplie avec des espaces.

### Pas rempli lors de la définition d'une variable

Crée la variable *v\_desc\_prod*, avec une longueur de 24 caractères, basée sur la longueur du champ de **Description\_Produit** :

```
ASSIGN v_prod_desc = SUBSTR(Description_Produit; 1; 50)
```

#### Remarque

Bien que SUBSTR() définisse une *longueur* de 50 caractères, la sortie est limitée à la longueur du champ **Description\_Produit**.

## Fonctions associées

SUBSTR() et SPLIT() renvoient tous deux un segment de données d'une chaîne source plus longue.

- SUBSTR() identifie le segment à partir de la position d'un caractère numérique.
- SPLIT() identifie le segment à partir d'un caractère de séparation.

# Fonction TAN( )

Renvoie la tangente d'un angle exprimée en radians, avec une précision de 15 décimales.

## Syntaxe

```
TAN(radians)
```

## Paramètres

Nom	Type	Description
<i>radians</i>	numérique	Mesure de l'angle en radians.

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 0,999999999999999 (la tangente du nombre spécifié de *radians*, équivalent à 45 degrés) :

```
TAN(0,785398163397448)
```

Renvoie 0,999999999999999 (la tangente de 45 degrés) :

```
TAN(45 * PI()/180)
```

### Exemples avancés

#### Utilisation d'une valeur d'entrée en degrés

Renvoie 1,000 (la tangente de 45 degrés, arrondie à 3 décimales) :

```
DEC(TAN(45 * PI()/180);3)
```

## Remarques

### Réalisation du test de l'arc mantisse

Les trois fonctions trigonométriques d'Analytics (SIN(), COS() et TAN()) prennent en charge le test de l'arc de mantisse associé à la loi de Benford.

### Conversion de degrés en radians

Si votre entrée est en degrés, vous pouvez utiliser la fonction PI() pour la convertir en radians :  $(\text{degrés} * \text{PI}() / 180) = \text{radians}$ . Si besoin, vous pouvez arrondir ou tronquer la valeur renvoyée à l'aide de la fonction DEC().

# Fonction TEST()

Renvoie une valeur logique précisant si une chaîne indiquée se trouve à une position d'octet spécifique dans un enregistrement.

## Syntaxe

```
TEST(position_octet, chaîne_de_caractères)
```

## Paramètres

Nom	Type	Description
<i>position_octet</i>	numérique	Nombre séquentiel recherché à partir de la gauche dans le format de table, qui identifie l'emplacement du premier caractère de la <i>chaîne_de_caractères</i> .  La fonction juge le résultat comme faux (F) si le début de la <i>chaîne</i> n'est pas identifié à l'emplacement indiqué, même si <i>chaîne_de_caractères</i> apparaît à un autre emplacement dans l'enregistrement.
<i>chaîne</i>	caractère	Chaîne de caractères à rechercher.  La recherche est sensible à la casse. Si vous pensez que la casse est incohérente, utilisez la fonction UPPER() pour convertir tous les caractères en majuscules.

## Sortie

Logique. Renvoie T (« true », vrai) si la chaîne indiquée commence à l'emplacement d'octet indiqué dans un enregistrement et F (« false », faux) dans les autres cas.

## Exemples

### Exemples basiques

Considérons un enregistrement contenant :

```
Département : Marketing
...|...|...|...|...|
```

Renvoi T :

```
TEST(5; "Département")
```

Renvoi F, car, dans l'enregistrement, "Département" commence à la position du cinquième octet, et non du sixième :

```
TEST(6; "Département")
```

Renvoi F, car la fonction est sensible à la casse.

```
TEST(5; "DÉPARTEMENT")
```

## Exemples avancés

### Isolement des enregistrements qui sont des en-têtes de page

Utilisez TEST( ) pour créer un filtre isolant tous les enregistrements commençant par "Page :"

```
SET FILTER TO TEST(1; "Page :")
```

# Fonction TIME( )

Extrait l'heure d'une heure ou d'une DateHeure spécifiée et la renvoie sous forme de chaîne de caractères. Peut également renvoyer l'heure du système d'exploitation en cours d'utilisation.

## Syntaxe

```
TIME(<heure/DateHeure> <;format>)
```

## Paramètres

Nom	Type	Description
<i>heure/DateHeure</i> optionnel	DateHeure	Le champ, l'expression, ou la valeur littérale à partir desquels extraire l'heure. Si ce paramètre est omis, c'est l'heure actuelle du système d'exploitation qui est renvoyée au format hh:mm:ss.
<i>format</i> optionnel	caractère	Format à appliquer à la chaîne de sortie, par exemple "hh:mm:ss". Si ce paramètre est ignoré, le format d'affichage de l'heure actuel d'Analytics est utilisé. Vous ne pouvez pas indiquer de <i>format</i> si vous avez ignoré <i>heure/DateHeure</i> .

## Sortie

Caractère.

## Exemples

### Exemples basiques

#### Valeurs d'introduction littérales

Renvoie "23:59:59" en supposant que le format d'affichage de l'heure Analytics soit hh:mm:ss:

```
TIME(`20141231 235959`)
```

Renvoie "11:59 P":

```
TIME(`20141231 235959`; "hh:mm A")
```

Renvoie l'heure actuelle du système d'exploitation renvoyée sous forme de chaîne de caractères au format hh:mm:ss format (horloge de 24 heures) :

```
TIME()
```

## Champ comme valeurs d'introduction

Renvoie une chaîne de caractères pour chaque valeur du champ **Reçu\_horodatage**, à l'aide du format d'affichage de l'heure actuelle d'Analytics :

```
TIME(Reçu_horodatage)
```

Renvoie une chaîne de caractères pour chaque valeur du champ **Reçu\_horodatage**, à l'aide du format d'affichage de l'heure spécifié :

```
TIME(Reçu_horodatage; "hh:mm:ss")
```

## Exemples avancés

### Calcul du temps écoulé pendant l'exécution d'une commande ou d'un script

Utilisez la fonction `TIME()` pour vous aider à calculer la durée d'exécution d'une commande Analytics en particulier ou d'un script tout entier.

Juste avant la commande sur laquelle vous souhaitez appliquer la fonction `TIME`, ou au début du script, indiquez cette ligne pour créer une variable qui stocke l'heure du système d'exploitation en cours d'utilisation :

```
ASSIGN Heure_début = TIME()
```

Juste après la commande, ou à la fin du script, indiquez les deux lignes ci-dessous.

La première ligne crée une variable qui stocke l'heure du système d'exploitation à la fin de l'exécution de la commande ou du script. La deuxième ligne calcule la différence entre les heures de début et de fin, puis convertit le résultat dans un format facilement lisible.

#### Astuce

Vous pouvez double-cliquer sur l'entrée de la trace `CALCULATE` afin de voir le temps écoulé pendant l'exécution de la commande ou du script.

```
ASSIGN Heure_fin = TIME()
CALCULATE STOT(CTOT(Heure_fin) - CTOT(Heure_début))
```

Si l'exécution de la commande ou du script franchit la ligne de changement de date, utilisez plutôt cette deuxième ligne :

```
CALCULATE `T000000` - (CTOT(Heure_début) - CTOT(Heure_fin))
```

## Remarques

### Longueur de la chaîne de sortie

La chaîne de sortie comporte toujours 14 caractères. Si le *format* indiqué pour générer les résultats ou que le format d'affichage de l'heure d'Analytics comporte moins de 14 caractères, la chaîne générée est complétée par des espaces en fin de chaîne.

### Informations sur les paramètres

Un champ indiqué pour *heure/DateHeure* peut utiliser n'importe quel format d'heure ou de DateHeure, tant que la définition de champ définit correctement le format.

Si vous utilisez *format* pour contrôler la manière dont la chaîne de sortie est affichée, vous êtes limité aux formats de la table ci-dessous. Vous pouvez utiliser n'importe quelle combinaison des formats d'heure et formats AM/PM. Le format AM/PM est facultatif et il est placé en dernier.

Indiquez *format* à l'aide de guillemets simples ou doubles. Par exemple : "hh:mm:ss AM"

Formats d'heure	Formats AM/PM	Exemples
hh:mm:ss	aucun horloge de 24 heures	"hh:mm:ss"
hhmmss	AM, ou PM horloge de 12 heures	"hhmmss PM"
hh:mm	A, ou P horloge de 12 heures	"hh:mm A"
hhmm		
hh		

### Spécification d'une valeur littérale d'heure ou de DateHeure



Lorsque vous indiquez une valeur littérale d'heure ou de DateHeure pour *heure/DateHeure*, vous êtes limité aux formats de la table ci-dessous, et vous devez entourer la valeur de guillemets inversés - par exemple, `20141231 235959`.

N'utilisez aucun séparateur comme les barres obliques (/) ou les deux points (:) entre les composants individuels des dates ou des heures.

- **Valeurs heure** : vous pouvez utiliser n'importe quel format d'heure répertorié dans le tableau ci-dessous. Vous devez utiliser un séparateur avant une valeur d'heure autonome pour que la fonction s'exécute correctement. Les séparateurs valides sont la lettre « t » ou la lettre « T ». Vous devez indiquer les heures à l'aide de l'horloge de 24 heures. Les décalages par rapport à l'heure universelle coordonnée (UTC) doivent être préfacés d'un signe plus (+) ou d'un signe moins (-).
- **Valeurs DateHeure** : vous pouvez utiliser n'importe laquelle des combinaisons de formats de date, de séparateur et d'heure répertoriées dans le tableau ci-dessous. La date doit précéder l'heure, et vous devez utiliser un séparateur entre les deux. Les séparateurs valides sont un espace unique, la lettre « t » ou la lettre « T ».

Exemples de formats	Exemples de valeurs littérales
thhmmss	`t235959`
Thhmm	`T2359`
AAAAMMJJ hhmss	`20141231 235959`
AAMMJJthhmm	`141231t2359`
AAAAMMJJThh	`20141231T23`
AAAAMMJJ hhmss+/-hhmm (décalage UTC)	`20141231 235959-0500`
AAMMJJ hhmm+/-hh (décalage UTC)	`141231 2359+01`
<p><b>Remarque</b></p> <p>N'utilisez pas hh seul dans le format d'heure principal avec des données qui ont un décalage UTC. Par exemple, évitez : hh+hhmm. Il se peut que les résultats ne soient pas fiables.</p>	

## Fonctions associées

Si vous devez renvoyer l'heure actuelle du système d'exploitation sous forme de valeur DateHeure, utilisez NOW() plutôt que TIME().

# Autres fonctions de conversion DateHeure

## Conversion DateHeure en Caractère

Fonction	Description
<a href="#">DATE()</a>	Extrait la date d'une date ou d'une DateHeure spécifiée et la renvoie sous forme de chaîne de caractères. Peut également renvoyer la date du système d'exploitation en cours d'utilisation.
<a href="#">DATETIME()</a>	Convertit une DateHeure en une chaîne de caractères. Peut également renvoyer la DateHeure du système d'exploitation en cours d'utilisation.

## Conversion Caractère ou Numérique en DateHeure

Fonction	Description
<a href="#">CTOD()</a>	Convertit une valeur de date au format Caractère ou Numérique en date. Cette fonction peut également extraire la date d'une valeur de type DateHeure au format Caractère ou Numérique et la renvoyer sous forme de date. Abréviation de "Caractère à Date".
<a href="#">CTODT()</a>	Convertit une valeur DateHeure au format Caractère ou Numérique en DateHeure. Abréviation de "Caractère à DateHeure".
<a href="#">CTOT()</a>	Convertit une valeur de type heure au format Caractère ou Numérique en heure. Cette fonction peut également extraire l'heure d'une valeur de type DateHeure au format Caractère ou Numérique et la renvoyer sous forme d'heure. Abréviation de "Caractère à Heure".

## Conversion Sérielle en DateHeure

Fonction	Description
<a href="#">STOD()</a>	Convertit une date sérielle - c'est-à-dire, une date exprimée sous forme d'intègre - en valeur de date. Abréviation de "Sérielle à Date".
<a href="#">STODT()</a>	Convertit une valeur DateHeure sérielle - c'est-à-dire, une DateHeure exprimée sous forme d'une intègre, et une portion fractionnelle de 24 heures - en une valeur correspondant à une DateHeure. Abréviation de "Sérielle à DateHeure".
<a href="#">STOT()</a>	Convertit une valeur horaire sérielle - c'est-à-dire une heure exprimée sous forme d'une portion fractionnelle de 24 heures, où 24 heures équivalent à 1 - en une valeur correspondant à une heure. Abréviation de "Sérielle à Heure".

# Fonction TODAY()

Renvoie la date du système d'exploitation en cours d'utilisation sous forme de données de type DateHeure.

## Syntaxe

```
TODAY()
```

## Paramètres

Cette fonction ne comporte pas de paramètres.

## Sortie

DateHeure.

## Exemples

### Exemples basiques

Renvoie la date du système d'exploitation en cours d'utilisation sous forme de valeur DateHeure, par exemple `20141231`, affichée au format d'affichage de la date actuelle d'Analytics :

```
TODAY()
```

## Remarques

### Fonctions associées

Si vous devez renvoyer la date actuelle du système d'exploitation sous forme de chaîne de caractères, utilisez DATE() plutôt que TODAY().

# Fonction TRANSFORM()

Inverse l'ordre d'affichage du texte bidirectionnel dans une chaîne indiquée.

## Syntaxe

```
TRANSFORM(chaîne_de_caractères_originale)
```

## Paramètres

Nom	Type	Description
<i>chaîne_de_caractères_originale</i>	caractère	Chaîne contenant le texte bidirectionnel.

## Sortie

Caractère.

## Exemples

### Exemples basiques

Dans la chaîne d'introduction, les caractères « XZQB » représente les caractères hébreux/bidirectionnels d'une chaîne d'introduction qui contient normalement des caractères réguliers.

Dans la chaîne de sortie, la direction de « XZQB » est inversée et la chaîne renvoie « BQZX ». Les autres caractères restent non modifiés.

Renvoie "ABC BQZX 123" :

```
TRANSFORM("ABC XZQB 123")
```

# Remarques

## Fonctionnement

La fonction TRANSFORMS( ) identifie les données bidirectionnelles et les affiche correctement dans la vue, de droite à gauche.

Tous les autres caractères traités par la fonction restent inchangés et continuent de s'afficher de gauche à droite.

## Quand utiliser TRANSFORMS( )

Utilisez TRANSFORMS( ) pour modifier l'ordre d'affichage des caractères arabes ou hébreux, afin qu'ils s'affichent correctement.

# Fonction TRIM( )

Renvoie une chaîne dont les espaces de fin ont été supprimées de la chaîne d'entrée.

## Syntaxe

```
TRIM(chaîne_de_caractères)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	La valeur dont il faut supprimer les espaces de fin.

## Sortie

Caractère.

## Exemples

### Exemples basiques

Notez que, dans les deux exemples, les espaces de début et les espaces entre les mots ne sont pas supprimés par la fonction TRIM( ).

Renvoie " Vancouver" :

```
TRIM(" Vancouver ")
```

Renvoie " New York" :

```
TRIM(" New York")
```

### Exemples avancés

#### Suppression des espaces insécables

La fonction TRIM( ) ne supprime pas les espaces insécables.

Si vous devez supprimer les espaces insécables de fin, créez un champ calculé à l'aide de l'expression suivante :

```
DEFINE FIELD Description_nettoyée COMPUTED TRIM(REPLACE(Description; CHR(160); CHR(32)))
```

La fonction REPLACE( ) remplace les espaces insécables par des espaces classiques, puis la fonction TRIM( ) supprime les espaces classiques de fin.

## Remarques

### Fonctionnement

La fonction TRIM( ) supprime uniquement les espaces de fin. Les espaces se trouvant à l'intérieur de la chaîne et les espaces de début ne sont pas supprimés.

### Fonctions associées

TRIM( ) est liée à la fonction LTRIM( ) qui supprime tous les espaces de début d'une chaîne, ainsi qu'à la fonction ALLTRIM( ) qui supprime à la fois les espaces de début et les espaces de fin.

# Fonction UNSIGNED( )

Renvoie des données numériques converties en type de données Unsigned.

## Syntaxe

```
UNSIGNED(nombre; longueur_résultat)
```

## Paramètres

Nom	Type	Description
<i>nombre</i>	numérique	Valeur à convertir.
<i>longueur_résultat</i>	numérique	Nombre d'octets à utiliser dans la chaîne de sortie.

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 000075 :

```
UNSIGNED(75; 3)
```

```
UNSIGNED(-75; 3)
```

```
UNSIGNED(7,5; 3)
```

Renvoie 2456 (le 1 est tronqué car seuls 4 chiffres peuvent être stockés lorsque la valeur *longueur\_résultat* équivaut à 2) :

```
UNSIGNED(12456; 2)
```



Renvoie 000000012456 :

```
UNSIGNED(-12,456; 6)
```

## Remarques

### Qu'est-ce qu'une donnée Unsigned ?

Le type de données Unsigned est utilisé par les systèmes d'exploitation centraux pour stocker des valeurs numériques dans un format qui utilise un minimum d'espace de stockage, tout en stockant deux chiffres dans chaque octet. Le type de données Unsigned est identique au type de données Packed, mais il n'utilise pas le dernier octet pour indiquer si la valeur est positive ou négative.

### Quand utiliser UNSIGNED( )

Utilisez la fonction UNSIGNED( ) permet de convertir des données numériques au format Unsigned pour une exportation vers les systèmes centraux.

### Valeurs renvoyées tronquées

Si la valeur *longueur\_résultat* est inférieure à la longueur de la valeur *nombre*, les chiffres supplémentaires sont tronqués.

# Fonction UPPER()

Renvoie une chaîne dont les caractères alphabétiques ont été convertis en majuscules.

## Syntaxe

```
UPPER(chaîne_de_caractères)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	Valeur à convertir en majuscules.

## Sortie

Caractère.

## Exemples

### Exemples basiques

Renvoie "ABC" :

```
UPPER("abc")
```

Renvoie "ABC 123 DEF" :

```
UPPER("abc 123 DEF")
```

Renvoie "ABCD 12" :

```
UPPER("AbCd 12")
```

Renvoie toutes les valeurs du champ **Nom** converties en majuscules :

```
UPPER(Nom)
```

## Remarques

### Fonctionnement

La fonction UPPER() convertit tous les caractères alphabétiques de la *chaîne\_de\_caractères* en majuscules. Tous les caractères non alphabétiques restent inchangés.

### Quand utiliser UPPER( )

Utilisez UPPER() pour vous assurer que la casse de tous les caractères inclus dans un champ, une variable ou une expression est identique. Le respect de la casse s'avère particulièrement important lorsque vous comparez des valeurs.

La fonction UPPER() peut aussi être utilisée pour mettre en majuscules des valeurs incluses dans des rapports.

# Fonction UTOD( )

Convertit une chaîne de caractères Unicode contenant une date mise en forme en valeur date Analytics. Abréviation de "Unicode en Date".

## Remarque

Cette fonction est spécifique à l'édition Unicode d'Analytics. Elle n'est pas prise en charge dans l'édition non-Unicode.

Utilisez cette fonction pour utiliser des dates dans des langues et formats différents de votre installation par défaut. Si la chaîne de caractères à convertir se trouve dans votre langue par défaut, utilisez CTOD( ) à la place.

## Syntaxe

```
UTOD(chaîne_de_caractères <;paramètre> <;style>)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	<p>La chaîne Unicode à convertir en date.</p> <p>La chaîne Unicode peut contenir une valeur de type DateHeure, mais la portion de la valeur correspondant à l'heure sera ignorée. Les valeurs de temps seules ne sont pas prises en charge.</p> <p><i>chaîne</i> doit correspondre au format d'entrée requis par la valeur <i>style</i> pour les paramètres régionaux de la date.</p>
<i>paramètre</i> optionnel	caractère	<p>Code indiquant la langue et les paramètres régionaux de la chaîne de sortie, et, de façon facultative, la version de la langue associée à un pays ou territoire particulier.</p> <p>Par exemple, "zh" indique chinois, et "pt_BR" indique portugais brésilien.</p> <p>Si vous ignorez ce paramètre, c'est la langue locale par défaut de votre ordinateur qui est utilisée. Si une langue est spécifiée mais qu'aucun pays n'est spécifié, le pays par défaut de la langue concernée est utilisé.</p> <p>Vous ne pouvez pas indiquer de <i>paramètres régionaux</i> si vous n'avez pas indiqué de <i>date</i>.</p> <p>Pour obtenir des informations sur les codes ISO locaux, consultez le site <a href="http://www.unicode.org">www.unicode.org</a>.</p>

Nom	Type	Description
<i>style</i> optionnel	numérique	<p>Style de format de date à utiliser pour la chaîne de caractères Unicode. Le style de format correspond à la norme relative aux paramètres régionaux spécifiés :</p> <ul style="list-style-type: none"> <li>○ 0 : format complet, comme « Dimanche 18 septembre 2016 »</li> <li>○ 1 : format long, comme « 18 septembre 2016 »</li> <li>○ 2 : format moyen, comme « 18 sept. 2016 »</li> <li>○ 3 : format numérique court, comme "18/09/16"</li> </ul> <p>Si vous ignorez ce paramètre, c'est la valeur par défaut de <b>2</b> qui est utilisée. Vous ne pouvez pas indiquer de <i>style</i> si vous n'avez pas indiqué les <i>paramètres régionaux</i> .</p> <p><b>Astuce</b></p> <p>Pour obtenir de l'aide afin de déterminer le format prévu pour votre chaîne d'entrée, effectuez l'une des opérations suivantes :</p> <ul style="list-style-type: none"> <li>• Utilisez la fonction <code>DTOU( )</code> pour générer un exemple de valeur à l'aide du style et des paramètres régionaux.</li> </ul> <p>Dans la ligne de commande, utilisez la commande <code>DISPLAY</code> pour imprimer la valeur :</p> <pre>DISPLAY DTOU('20160909'; "es_MX"; 3)</pre> <ul style="list-style-type: none"> <li>• Consultez une source officielle sur le format de date standard pour connaître le style des paramètres locaux spécifiques.</li> </ul>

## Sortie

DateHeure. La valeur de date est générée à l'aide du format d'affichage de date Analytics actuel.

# Exemples

## Exemples basiques

### Remarque

Tous les exemples supposent un format d'affichage de la date Analytics actuelle correspondant à JJ MMM AAAA.

Dans les exemples ci-dessous, le code régional pour le chinois ( "zh" ) et celui du chinois simplifié ( "zh\_CN" ) correspondent à des chaînes d'entrée différentes et ils ne sont pas interchangeables.

Vous devez également indiquer le *style* correct. Une longue chaîne de date Unicode (c'est-à-dire une chaîne dont le *style* est 1) ne renvoie pas une date Analytics si vous indiquez un *style* de 2.

### Valeurs d'introduction littérales

Renvoie `20141231` affiché au format 31 déc 2014 :

```
UTOD("31 de dezembro de 2014"; "pt_BR"; 1)
```

Renvoie `20141231` affiché au format 31 déc 2014 :

```
UTOD("31 grudnia 2014"; "pl"; 1)
```

### Valeurs d'introduction champ

Renvoie la date équivalente pour chaque chaîne de caractères Unicode dans le champ **Date\_facture** :

```
UTOD(Date_facture; "zh"; 1)
```

### La valeur d'entrée utilise le style de date complet

Renvoie `20141231` affiché au format 31 déc 2014 (sans identifiant de région spécifié) :

```
UTOD("星期三; 2014 十二月 31"; "zh"; 0)
```

Renvoie `20141231` affiché au format 31 déc 2014 (identifiant de région spécifié) :

```
UTOD("2014年12月31日星期三"; "zh_CN"; 0)
```

## La valeur d'entrée utilise le style de date long

Renvoie `20141231` affiché au format 31 déc 2014 (sans identifiant de région spécifié) :

```
UTOD("2014 十二月 31"; "zh"; 1)
```

Renvoie `20141231` affiché au format 31 déc 2014 (identifiant de région spécifié) :

```
UTOD("2014年12月31日"; "zh_CN"; 1)
```

## Remarques

### Réussite de la conversion de chaînes Unicode

Pour convertir avec succès des chaînes Unicode contenant des dates en dates Analytics, vous devez indiquer les paramètres *local* et *style* qui correspondent à la langue, au pays/région (le cas échéant), et le style de la date dans la chaîne Unicode.

### Fonctions associées

UTOD ( ) est l'inverse de la fonction DTOU( ), qui convertit une date en chaîne Unicode. Si vous n'êtes pas certain du pays/région et du style à indiquer pour la fonction UTOD( ), vous pouvez utiliser DTOU( ) et expérimenter avec différents paramètres pour produire une chaîne Unicode de sortie correspondant à la forme des chaînes Unicode en entrée que vous souhaitez convertir avec UTOD( ).

# Fonction VALUE()

Convertit une chaîne de caractères en valeur numérique.

## Syntaxe

```
VALUE(chaîne; décimales)
```

## Paramètres

Nom	Type	Description
<i>chaîne</i>	caractère	Le champ, la valeur littérale ou l'expression à convertir.
<i>décimales</i>	numérique	Nombre de positions décimales à inclure dans la sortie.

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie -123,400 :

```
VALUE("123,4-"; 3)
```

Renvoie 123456,00 :

```
VALUE("$123.456"; 2)
```

Renvoie -77,45 :

```
VALUE("77,45CR"; 2)
```

Renvoie -123457 :



```
VALUE(" (123.456,78)"; 0)
```

## Entrée de champ

Renvoie des valeurs de type caractère dans le champ **Salaire** sous forme de nombre sans décimales :

```
VALUE(Salaire; 0)
```

# Remarques

## Fonctionnement

Cette fonction convertit les données caractère en données numériques. La fonction VALUE() permet de convertir des expressions caractère ou des valeurs de champ en valeurs numériques pour les utiliser dans les commandes Analytics.

## Mise en forme des entrées numériques

VALUE() accepte les nombres de tout format. Vous pouvez utiliser en tant qu'entrée tout type de format numérique accepté par le type de données d'Impression, tels que la ponctuation, les signes de début ou de fin et les parenthèses.

## Valeurs négatives

La fonction VALUE() peut interpréter différents indicateurs de valeurs négatives telles que les parenthèses ou le signe moins. Elle peut également interpréter CR (crédit) et DR (débit). Par exemple :

Renvoie -1000,00 :

```
VALUE("(1000)"; 2)
```

```
VALUE("1000CR"; 2)
```

## Valeurs décimales et valeurs entières

Si la valeur *chaîne* n'inclut pas de décimales, Analytics traite le nombre comme un nombre entier. Par exemple :

Renvoie 123,00 :

```
VALUE("123"; 2)
```

Si le nombre de décimales indiqué dans le paramètre *décimales* est inférieur au nombre indiqué dans le champ ou l'expression, le résultat est arrondi. Par exemple :

Renvoie "10,6" :

```
VALUE("10,56"; 1)
```

## Fonctions associées

La fonction VALUE() est l'opposé de la fonction STRING(), qui convertit des données numériques en données caractère.

# Fonction VERIFY()

Renvoie une valeur logique indiquant si les données incluses dans un champ de données physique sont valides.

## Syntaxe

```
VERIFY(champ)
```

## Paramètres

Nom	Type	Description
<i>champ</i>	caractère numérique DateHeure	Doit être un champ de données physique.

## Sortie

Logique. Renvoie T (« true », vrai) si les données indiquées dans le champ sont valides ; renvoie F (« false », faux) dans les autres cas.

## Exemples

### Exemples basiques

Extrait les enregistrements pour lesquels la fonction VERIFY() indique un résultat jugé faux sont extraits dans une nouvelle table Analytics :

```
EXTRACT RECORD IF NOT VERIFY(Adresse) TO "Entrées Nonvalides.fil"
```

## Remarques

La fonction VERIFY() détermine si les données incluses dans un champ sont cohérentes avec le type de données indiqué pour ce champ.

## Quand utiliser VERIFY( )

La fonction VERIFY( ) permet de profiter d'un contrôle plus précis sur les champs à vérifier que la commande VERIFY ou l'option **Vérifier les données**, dans l'onglet **Numérique** de la boîte de dialogue **Options (Outils >Options)**. Cette fonction permet de détecter des erreurs dans les champs individuels et d'effectuer l'action appropriée à la situation.

## Lorsque la fonction est évaluée sur vrai

Pour que la fonction renvoie T :

- les champs de type caractère doivent contenir uniquement des caractères valides imprimables, comme des lettres, des chiffres et des symboles
- les champs numériques doivent contenir uniquement des caractères numériques valides, comme des chiffres, des décimales, et des symboles de devises
- les champs de type DateHeure doivent contenir uniquement des valeurs de date, DateHeure ou heure valides.

## Champs calculés et expressions

Les champs et les expressions calculés obtiennent toujours un résultat T (« true », vrai). De ce fait, cette fonction ne peut pas être utilisée avec les champs ou les expressions calculés, sauf s'ils sont d'abord convertis en champs physiques. Utilisez l'option **Champs** dans la boîte de dialogue **Extraire** qui permet d'extraire les champs ou expressions calculés pour les convertir en champs physiques.

# Fonction WORKDAY()

Renvoie le nombre de jours ouvrables entre deux dates.

## Syntaxe

```
WORKDAY(date_début, date_fin <;joursnonouvr>)
```

## Paramètres

Nom	Type	Description
<i>date_début</i>	DateHeure	Date de début de la période pour laquelle les jours ouvrables sont calculés. La date de début est incluse dans la période.
<i>date_fin</i>	DateHeure	Date de fin de la période pour laquelle les jours ouvrables sont calculés. La date de fin est incluse dans la période.
<i>joursnonouvr</i> optionnel	caractère	<p>Jours de la semaine qui sont les jours de week-end, ou jours non ouvrables, et exclus du calcul. Si vous ignorez le paramètre <i>joursnonouvr</i>, samedi et dimanche servent de jours non ouvrables par défaut.</p> <p>Saisissez <i>joursnonouvr</i> à l'aide des abréviations suivantes, séparées par une espace ou une virgule :</p> <ul style="list-style-type: none"> <li>◦ Mon</li> <li>◦ Tue</li> <li>◦ Wed</li> <li>◦ Thu</li> <li>◦ Fri</li> <li>◦ Sat</li> <li>◦ Sun</li> </ul> <p><i>joursnonouvr</i> ne respectent pas la casse. Les abréviations doivent être saisies en anglais même si vous utilisez une version d'Analytics qui n'est pas en anglais :</p> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-top: 10px;">"Fri, Sat, Sun"</div>

### Remarque

Vous pouvez indiquer une valeur de type DateHeure pour *date\_début* ou *date\_fin*, mais la portion de la valeur correspondant à l'heure sera ignorée.

Si *date\_début* est plus récente que *date\_fin*, une valeur négative est renvoyée.

# Sortie

Numérique. Nombre de jours ouvrables dans la période pour laquelle les jours ouvrables sont calculés.

## Exemples

### Exemples basiques

#### Valeurs d'introduction littérales

Renvoie 5 (nombre de jours ouvrables s'écoulant entre le lundi 2 mars 2015 et le dimanche 8 mars 2015 inclus) :

```
WORKDAY(`20150302`; `20150308`)
```

Renvoie 6 (nombre de jours ouvrables s'écoulant entre le lundi 2 mars 2015 et le dimanche 8 mars 2015 inclus), lorsque le dimanche est le seul jour non ouvrable) :

```
WORKDAY(`20150302`; `20150308`; "Sun")
```

Renvoie 5 (nombre de jours ouvrables s'écoulant entre le dimanche 1er mars 2015 et le samedi 7 mars 2015 inclus), lorsque le vendredi et le samedi sont les jours non ouvrables) :

```
WORKDAY(`20150301`; `20150307`; "Fri, Sat")
```

#### Valeurs d'introduction champ

Renvoie le nombre de jours ouvrables s'écoulant entre chaque date dans le champ **Date\_début** et le 31 décembre 2015 inclus :

```
WORKDAY(Date_début; `20151231`)
```

Renvoie le nombre de jours ouvrables s'écoulant entre chaque date dans le champ **Date\_début** et une date correspondante dans le champ **Date\_fin** inclus :

- Les jours fériés légaux sont inclus dans le total des jours ouvrables et il se peut qu'ils doivent être exclus à l'aide d'un calcul distinct
- Une valeur renvoyée négative indique que la date de début est plus récente que la date de fin

```
WORKDAY(Date_début; Date_fin)
```

# Remarques

## Formats de date

Un champ spécifié pour *date\_début* et *date\_fin* peut utiliser n'importe quel format de date, tant que la définition de champ définit correctement le format.

Lorsque vous indiquez une valeur littérale de date pour *date\_début* et *date\_fin*, vous êtes limité aux formats AAAAMMJJ et AAMMJJ, et vous devez entourer cette valeur de guillemets inversés, par exemple, `20141231`.

## Jours non ouvrables différents de samedi et de dimanche

La possibilité de spécifier des jours non ouvrables différents de samedi et de dimanche vous permet d'utiliser la fonction `WORKDAY()` avec des données qui ne dépendent pas de la semaine de travail habituelle du lundi au vendredi ou d'une semaine de travail de cinq jours.

Par exemple, si vous spécifiez "Sun" tout seul comme jour non ouvrable, vous créez une semaine de travail de six jours du lundi au samedi.

## Prise en compte des jours fériés légaux

La fonction `WORKDAY()` ne prend pas en compte les jours fériés légaux, ce qui signifie que la valeur renvoyée ne reflète pas forcément le nombre réel de jours ouvrables dans une période si la période contient un ou plusieurs jours fériés légaux.

### Script « Calculer les jours ouvrables » dans ScriptHub

Si vous devez prendre en compte les jours fériés légaux, une possibilité consiste à utiliser le script [Calculer les jours ouvrables](#) dans ScriptHub, qui accepte une liste de jours fériés définis par l'utilisateur.

Pour les données qui englobent des périodes plus longues et qui incluent de nombreux jours fériés, l'utilisation du script est probablement l'approche la plus simple. Pour plus d'informations, consultez la rubrique « Importation de scripts à partir de ScriptHub » dans l'aide d'Analytics.

Pour des périodes plus courtes ne comportant que trois ou quatre jours fériés, comme un trimestre, vous trouverez peut-être que la création du champ calculé conditionnel décrit ci-dessous ne demande pas trop de travail.

### Champ calculé conditionnel permettant d'ôter les jours fériés légaux

Si besoin, vous pouvez créer un champ calculé conditionnel permettant d'ôter les jours fériés légaux de la valeur renvoyée par la fonction `WORKDAY()`.

Par exemple, pour les données du 1er trimestre 2015, vous pourriez décrémenter la valeur renvoyée `WORKDAY()` de 1 pour chaque jour férié tombant dans une période spécifiée :

- 1er janvier 2015
- 19 janvier 2015
- 16 février 2015

L'exemple qui suit contient des périodes qui présentent n'importe quelle date de début et n'importe quelle date de fin pendant le trimestre.

Vous pourriez commencer par créer un champ calculé, par exemple **Jours\_ouvr**, qui calcule les jours ouvrables pour toutes les périodes du trimestre :

```
DEFINE FIELD Jours_ouvr COMPUTED WORKDAY(Date_début; Date_fin)
```

Vous créez ensuite un champ calculé conditionnel, par exemple **Jours\_ouvr\_sans\_jours\_fériés**, qui ajuste la valeur renvoyée par le premier champ calculé (**Jours\_ouvr**) :

```
DEFINE FIELD Jours_ouvr_sans_jours_fériés COMPUTED

Jours_ouvr-1 IF Date_début = `20150101` AND Date_fin < `20150119`
Jours_ouvr-2 IF Date_début = `20150101` AND Date_fin < `20150216`
Jours_ouvr-3 IF Date_début = `20150101` AND Date_fin <= `20150331`
Jours_ouvr IF Date_début < `20150119` AND Date_fin < `20150119`
Jours_ouvr-1 IF Date_début < `20150119` AND Date_fin < `20150216`
Jours_ouvr-2 IF Date_début < `20150119` AND Date_fin <= `20150331`
Jours_ouvr-1 IF Date_début = `20150119` AND Date_fin < `20150216`
Jours_ouvr-2 IF Date_début = `20150119` AND Date_fin <= `20150331`
Jours_ouvr IF Date_début < `20150216` AND Date_fin < `20150216`
Jours_ouvr-1 IF Date_début < `20150216` AND Date_fin <= `20150331`
Jours_ouvr-1 IF Date_début = `20150216` AND Date_fin <= `20150331`
Jours_ouvr IF Date_début < `20150331` AND Date_fin <= `20150331`
Jours ouvrables
```

### Remarque

L'ordre des conditions dans le champ calculé conditionnel est important.

Analytics évalue plusieurs conditions à partir du haut. La première condition évaluée sur Vrai pour un enregistrement affecte la valeur du champ calculé conditionnel pour cet enregistrement. Une condition ultérieure évaluée sur Vrai ne modifie pas la valeur affectée.



# Fonction YEAR()

Extrait l'année d'une date ou d'une DateHeure spécifiée et la renvoie sous forme de valeur numérique à l'aide du format AAAA.

## Syntaxe

```
YEAR(date/DateHeure)
```

## Paramètres

Nom	Type	Description
<i>date/DateHeure</i>	DateHeure	Le champ, l'expression, ou la valeur littérale à partir desquels extraire l'année.

## Sortie

Numérique.

## Exemples

### Exemples basiques

Renvoie 2014 :

```
YEAR('20141231')
```

```
YEAR('141231 235959')
```

Renvoie l'année pour chaque valeur dans le champ **Date\_facture** :

```
YEAR(Date_facture)
```

# Remarques

## Informations sur les paramètres

Un champ indiqué pour *date/DateHeure* peut utiliser n'importe quel format de date ou de DateHeure, tant que la définition de champ définit correctement le format.

## Spécification d'une valeur littérale de date ou de DateHeure

Lorsque vous indiquez une valeur littérale de date ou de DateHeure pour *date/DateHeure*, vous êtes limité aux formats de la table ci-dessous, et vous devez entourer la valeur de guillemets inversés - par exemple, `20141231`.

N'utilisez aucun séparateur comme les barres obliques (/) ou les deux points (:) entre les composants individuels des dates ou des heures.

- **Valeurs DateHeure** : vous pouvez utiliser n'importe laquelle des combinaisons de formats de date, de séparateur et d'heure répertoriées dans le tableau ci-dessous. La date doit précéder l'heure, et vous devez utiliser un séparateur entre les deux. Les séparateurs valides sont un espace unique, la lettre « t » ou la lettre « T ».
- **Valeurs heure** : vous devez indiquer l'heure à l'aide de l'horloge de 24 heures. Les décalages par rapport à l'heure universelle coordonnée (UTC) doivent être préfacés d'un signe plus (+) ou d'un signe moins (-).

Exemples de formats	Exemples de valeurs littérales
AAAAMMJJ	`20141231`
AAMMJJ	`141231`
AAAAMMJJ hhmmss	`20141231 235959`
AAMMJJthhmm	`141231t2359`
AAAAMMJJThh	`20141231T23`
AAAAMMJJ hhmmss+/-hhmm (décalage UTC)	`20141231 235959-0500`
AAMMJJ hhmm+/-hh (décalage UTC)	`141231 2359+01`

Exemples de formats	Exemples de valeurs littérales
<p><b>Remarque</b></p> <p>N'utilisez pas hh seul dans le format d'heure principal avec des données qui ont un décalage UTC. Par exemple, évitez : hh+hhmm. Il se peut que les résultats ne soient pas fiables.</p>	

# Fonction ZONED( )

Convertit les données numériques en caractères et ajoute des zéros de tête à la sortie.

## Syntaxe

```
ZONED(nombre; longueur)
```

## Paramètres

Nom	Type	Description
<i>nombre</i>	numérique	Valeur numérique à convertir en chaîne. <b>Remarque</b> Si vous souhaitez ajouter des zéros de début à une valeur de type caractère contenant une chaîne de caractères numériques, vous devez utiliser la fonction VALUE( ) pour convertir le type de données caractère en type de données numérique avant d'utiliser cette valeur comme entrée de ZONED( ). Pour plus d'informations, consultez la section "Fonction VALUE( )" Page 864.
<i>longueur</i>	numérique	Longueur de la chaîne de sortie.

## Sortie

Caractère.

## Exemples

### Exemples basiques

#### Valeur d'entrée nombre entier

Renvoie "235" :

```
ZONED(235; 3)
```

Renvoie "00235" car le paramètre *longueur* est supérieur au nombre de chiffres dans *nombre* ; par conséquent, deux zéros de tête sont ajoutés au résultat :

```
ZONED(235; 5)
```

Renvoie "35" car le paramètre *longueur* est inférieur au nombre de chiffres dans *nombre* ; par conséquent, le chiffre le plus à gauche est tronqué dans le résultat :

```
ZONED(235; 2)
```

## Valeur d'entrée décimale

Renvoie "23585", car le format de données segmenté ne prend pas en charge de virgule décimale :

```
ZONED(235,85; 5)
```

## Valeur d'entrée négative

Renvoie "64489M", car le nombre est négatif et "M" représente le dernier chiffre 4 :

```
ZONED(-6448,94; 6)
```

Renvoie "489J" car le paramètre *longueur* est inférieur au nombre de chiffres dans *nombre* ; par conséquent, les deux chiffres les plus à gauche sont tronqués dans le résultat et le nombre est négatif et "J" représente le dernier chiffre 1 :

```
ZONED(-6448,91; 4)
```

## Exemples avancés

### Ajout de zéros de début à un champ de type caractère contenant des chiffres

Le champ Numéro\_employé contient la valeur "254879". Vous devez convertir la valeur en chaînes de 10 chiffres avec des zéros de début.

#### Astuce

Vous devez utiliser la fonction VALUE( ) pour convertir les données caractère en données numériques avant d'utiliser ces dernières comme entrée de ZONED( ).

```
COMMENT renvoie "0000254879"
ASSIGN v_longueur_chaine = 10
ASSIGN v_nombre_decimales = 0
ZONED(VALUE(Numero_employe; v_nombre_decimales); v_longueur_chaine)
```

## Harmonisation d'un champ clé lors de la jointure de tables

Vous avez deux tables, **Ar** et **Client**, et vous devez les joindre sur le champ **NumClient** pour une analyse ultérieure. Les deux tables présentent chacune un champ **NumClient**, mais le format des données est différent :

- **Ar** : champ numérique (par exemple, 235)
- **Client** : champ de 5 caractères qui remplit les numéros avec des zéros de tête (par exemple, "00235")

Pour harmoniser les champs lors de la jointure afin que les types de données et les longueurs soient égaux, vous utilisez la fonction **ZONED( )** pour convertir le champ clé **Ar NumClient** en champ de type caractère d'une longueur de 5 pour qu'il corresponde au format du champ clé dans **Client** :

```
OPEN Ar PRIMARY
OPEN Client SECONDARY
JOIN PKEY ZONED(NumClient;5) FIELDS NumClient Echéance Montant SKEY NumClient
UNMATCHED TO Ar_Client OPEN PRESORT SECSORT
```

## Remarques

### Fonctionnement

Cette fonction convertit des données numériques en caractères et ajoute des zéros de tête à la sortie. La fonction est communément utilisée pour harmoniser des champs qui requièrent des zéros de tête (par exemple, un numéro de chèque, un numéro de bon de commande et des champs de numéro de facture).

### Quand utiliser ZONED( )

Utilisez cette fonction pour convertir une valeur numérique positive en une valeur de caractère contenant des zéros de tête. Ceci s'avère utile pour la normalisation de données dans des champs destinés à être utilisés comme champs clés.

Par exemple, si une table contient des numéros de facture au format 100 dans un champ numérique, et si une autre table contient des numéros de facture au format "00100" dans un champ de caractères, la fonction **ZONED( )** permet de convertir la valeur numérique 100 au format "00100". Ceci permet de comparer des numéros de factures similaires.

## Valeurs des longueurs de chaînes et valeurs renvoyées

Des zéros de tête sont ajoutés à la valeur de sortie quand la valeur *longueur* est supérieure au nombre de chiffres dans *nombre*. Quand le paramètre *longueur* est inférieur au nombre de chiffres dans *nombre*, la sortie est tronquée à partir du côté gauche. Si la valeur *nombre* est de la même longueur que *longueur*, aucun zéro n'est ajouté.

## Nombres décimaux

Le format de données Zoned n'inclut pas de virgule décimale explicite.

## Nombres négatifs

Si le *nombre* d'entrée est négatif, le chiffre le plus à droite s'affiche sous forme de caractère dans le résultat :

- "}" pour 0
- une lettre comprise entre "J" et "R" pour les chiffres 1 à 9

## ZONED( ) et l'édition Unicode d'Analytics

Si vous utilisez l'édition Unicode d'Analytics, vous devez utiliser la fonction BINTOSTR( ) pour afficher correctement la valeur renvoyée de la fonction ZONED( ). Vous devez également utiliser la fonction BINTOSTR( ) si vous souhaitez utiliser la valeur renvoyée de la fonction ZONED( ) en tant que paramètre dans une autre fonction.

# Fonction ZSTAT()

Renvoie la statistique Z standard.

## Syntaxe

```
ZSTAT(proportion_réelle; proportion_probable; population)
```

## Paramètres

Nom	Type	Description
<i>proportion_réelle</i>	numérique	<ul style="list-style-type: none"> <li>○ <b>Lors de la spécification de paramètres en tant que nombres :</b> représente le compte réel, tel qu'un chiffre de tête ou une combinaison de chiffres de tête.</li> <li>○ <b>Lors de la spécification de paramètres en tant que proportions :</b> représente la proportion probable de la valeur en cours d'évaluation, qui doit être comprise entre 0 et 1 (c'est-à-dire, supérieure ou égale à 0 et inférieure ou égale à 1).</li> </ul>
<i>proportion_probable</i>	numérique	<ul style="list-style-type: none"> <li>○ <b>Lors de la spécification de paramètres en tant que nombres :</b> représente le compte probable, tel qu'un chiffre de tête ou une combinaison de chiffres de tête.</li> <li>○ <b>Lors de la spécification de paramètres en tant que proportions :</b> représente la proportion probable de la valeur en cours de test, qui doit être comprise entre 0 et 1 (c'est-à-dire, supérieure à 0 et inférieure à 1).</li> </ul>
<i>population</i>	numérique	Le nombre total d'éléments en cours d'évaluation. Ce paramètre doit être un nombre entier positif supérieur à 0.

## Sortie

Numérique.

## Exemples

### Exemples avancés

#### Paramètres exprimés sous forme de nombres



Sur la base des données des 10 années précédentes, on sait que la distribution mensuelle des sinistres invalidité du personnel est normalement très uniforme. En avril, mai et juin de cette année, les déclarations de sinistre ont augmenté d'environ 10 %, atteignant une moyenne mensuelle de 220 au lieu de 200. En juillet et août, elles ont été légèrement plus basses, atteignant 193 et 197. Au total, il y a eu 2 450 déclarations dans l'année. Pour déterminer si ces résultats élevés et bas ont été significatifs, utilisez la statistique Z.

Le nombre réel de déclarations d'avril à juin est plus élevé que prévu et atteint 660. Le nombre prévu de déclarations pour cette période correspond à 25 % des 2 450 déclarations annuelles, soit 612,5. La statistique Z pour ces comptes est de 2,193 :

$$ZSTAT(660; 612,5; 2450)$$

Une statistique Z de 1,96 a une importance de 0,05. Une statistique de 2,57 a une importance de 0,01. Par conséquent, la probabilité que les taux plus élevés de déclarations de sinistre soient dus au hasard est comprise entre 1 chance sur 20 et 1 chance sur 100.

Le nombre réel de déclarations pour juillet et août est plus bas que prévu et s'élève à 390. Le nombre prévu de déclarations pour cette période correspond à un sixième des 2 450 réclamations annuelles, soit 408,33. La statistique Z pour ces proportions est de 0,967 :

$$ZSTAT(390; 408,33; 2450)$$

Ce résultat n'est pas très significatif. Des statistiques Z de 1 000 ou moins sont très courantes et peuvent généralement être ignorées.

## Paramètres exprimés sous forme de proportions

Sur la base des données des 10 années précédentes, on sait que la distribution mensuelle des sinistres invalidité du personnel est normalement très uniforme. En avril, mai et juin de cette année, les déclarations de sinistre ont augmenté d'environ 10 %, atteignant une moyenne mensuelle de 220 au lieu de 200. En juillet et août, elles ont été légèrement plus basses, atteignant 193 et 197. Au total, il y a eu 2 450 déclarations dans l'année. Pour déterminer si ces résultats élevés et bas ont été significatifs, utilisez la statistique Z.

Le nombre réel de demandes pour la période allant d'avril à juin est représenté par la proportion 660/2450, qui est plus élevée que prévue. Le nombre probable de demandes pour cette période devrait correspondre à 25 pour cent des 2 450 demandes annuelles. La statistique Z pour ces proportions est de 2,193 :

$$ZSTAT((1,00000000 * 660 / 2450); 0,25; 2450)$$

Une statistique Z de 1,96 a une importance de 0,05. Une statistique de 2,57 a une importance de 0,01. Par conséquent, la probabilité que les taux plus élevés de déclarations de sinistre soient dus au hasard est comprise entre 1 chance sur 20 et 1 chance sur 100.

Le nombre réel de déclarations pour juillet et août est bas et s'élève à 390. Le nombre prévu de déclarations pour cette période doit correspondre à un sixième ou 16,6667 % des 2 450 déclarations annuelles. La statistique Z pour ces proportions est de 0,967 :

```
ZSTAT((1,00000000 * 390 / 2450); 0,16667; 2450)
```

Ce résultat n'est pas très significatif. Des statistiques Z de 1 000 ou moins sont très courantes et peuvent généralement être ignorées.

## Remarques

### Fonctionnement

La fonction ZSTAT() calcule la statistique Z standard utilisée dans plusieurs tâches de résolution des problèmes, y compris l'analyse numérique. Elle génère un résultat à trois décimales.

### Utilisation de ZSTAT( )

Utilisez ZSTAT() pour évaluer la fréquence probable d'occurrence d'un résultat donné dans une période indiquée ou une catégorie. Plus la statistique Z qui en résulte est élevée, plus la probabilité d'occurrence sera réduite.

Par exemple, une statistique Z de 1,96 a une importance de 0,05, à savoir la possibilité d'une occurrence sur 20, alors qu'une statistique Z de 2,57 a une importance de 0,01, représentant la possibilité d'une occurrence sur 100. Pour obtenir des informations sur la statistique Z, consultez un ouvrage de statistique.

### Indiquer une entrée pour ZSTAT( )

Vous pouvez spécifier les paramètres de ZSTAT() en tant que nombres ou proportions :

- Lors de la spécification des deux valeurs d'introduction en tant que nombres, la fonction calcule la statistique Z à l'aide d'opérations en virgule flottante
- Lors de la spécification des deux valeurs d'introduction en tant que proportions, la fonction calcule la statistique Z à l'aide d'opérations à virgule fixe et vous devez utiliser un multiplicateur décimal pour contrôler l'arrondi
- Lorsque vous utilisez une expression au sein d'une expression pour calculer la valeur *proportion\_réelle* ou *proportion\_probable*, vous devez spécifier le niveau de précision souhaité dans le résultat à l'aide d'un multiplicateur décimal. Analytics a une précision de 8 chiffres, c'est pourquoi un multiplicateur de 1.00000000 permettra de bénéficier de la plus grande précision possible.

# Outils d'analyse

# Scripts d'outil d'analyse

L'exécution de scripts ne se limite pas à Analytics uniquement. En convertissant des scripts classiques en **script d'outil d'analyse**, vous pouvez planifier et exécuter des scripts dans le module Robots d'HighBond ou dans Analytics Exchange. Vous pouvez aussi exécuter des scripts d'outil d'analyse dans la fenêtre App Analyse, un composant amovible d'Analytics.

## Que sont des scripts d'outils d'analyse ?

Un "outil d'analyse" ou script d'outil d'analyse correspond à un script comportant un en-tête d'outil d'analyse. L'en-tête d'outil d'analyse est une série de balises déclaratives qui permet au script de s'exécuter dans Robots, sur AX Serveur ou dans la fenêtre App Analyse. L'en-tête de l'outil d'analyse comprend des paramètres d'entrée que l'utilisateur renseigne au préalable permettant ainsi d'exécuter le script d'outil d'analyse de manière autonome, immédiatement ou à une heure planifiée.

### Astuce

Les script d'outils d'analyse sont généralement exclusivement développés et testés dans Analytics, qui permet un développement plus aisé. Utilisez le client AX pour effectuer des mises à jour simples au niveau des scripts d'outils d'analyse existants qui sont stockés dans le serveur AX.

## Qu'est-ce qu'une App Analyse ?

Une **App Analyse** correspond à un projet Analytics créé pour une utilisation dans la fenêtre Analytics Exchange ou App Analyse. Les apps Analyse contiennent un ou plusieurs scripts d'outils d'analyse, mais elles peuvent aussi contenir des tables de données et des interprétations.

### Remarque

Les Apps Analyse sont généralement créées ou développées par les experts en script internes d'une organisation ou par des accords avec des consultants de Galvanize.

## Passer des scripts classiques en scripts d'outil d'analyse

Des scripts d'outil d'analyse commencent par des scripts classiques. Pour exécuter un script classique dans Robots, sur AX Serveur ou dans la fenêtre App Analyse, vous devez convertir le script classique en script d'outil d'analyse :

1. Créez et testez un script dans Analytics.
2. Ajoutez les balises d'en-tête d'outils d'analyse qui conviennent pour transformer le script en script d'outil d'analyse.

3. Compressez le script d'outil d'analyse pour l'exécuter sur AX Serveur ou dans la fenêtre App Analyse. Ne compressez pas les scripts d'outil d'analyse exécutés dans Robots.

## Ajouter des en-têtes d'outils d'analyse

Les en-têtes d'outils d'analyse sont définis dans un bloc de commentaire à la première ligne du script. Au minimum, un en-tête d'outil d'analyse déclare que le script est un script d'outil d'analyse :

```
COMMENT
//ANALYTIC Identifier les chèques manquants
Ce script d'outil d'analyse identifie les numéros de chèque manquants
END
```

Pour plus d'informations, consultez la section "Ajouter des en-têtes d'outils d'analyse" Page 929.

## Diffuser et exécuter des scripts d'outil d'analyse

Plusieurs options permettent de diffuser et d'exécuter des script d'outil d'analyse, en fonction des produits et composants Galvanize utilisés par votre organisation.

Produit/composant	Méthode de diffusion et d'exécution d'un script d'outil d'analyse
Module Robots d'HighBond	<ul style="list-style-type: none"> <li>transmettre un ou plusieurs scripts d'outils d'analyse sous forme de version en mode développement dans Robots ainsi que planifier et exécuter une version activée en mode production</li> </ul>
Serveur AX	<p>Une de ces méthodes :</p> <ul style="list-style-type: none"> <li>importer le projet Analytics (fichier <code>.acl</code>) directement dans AX Serveur, et planifier et exécuter un script d'outil d'analyse à l'aide d'AX Client</li> <li>compresser le projet dans un fichier App Analyse compressé (fichier <code>.aclapp</code>), l'importer dans AX Serveur, et exécuter un script d'outil d'analyse à l'aide du Client Web d'AX</li> </ul> <p>Pour plus d'informations, consultez la section "Compresser des Apps Analyse" Page 939.</p>
Fenêtre App Analyse	<ul style="list-style-type: none"> <li>compresser le projet dans un fichier App Analyse compressé (fichier <code>.aclapp</code>), ouvrir le projet sous forme d'App Analyse (fichier <code>.aclx</code>), et exécuter le script d'outil d'analyse dans la fenêtre App Analyse</li> </ul> <p>Pour plus d'informations, consultez la section "Compresser des Apps Analyse" Page 939.</p>

## Déterminer l'environnement d'exécution d'un script d'outil d'analyse

Si vous souhaitez créer un script d'outil d'analyse capable de s'exécuter à la fois dans Analytics, dans Analytics Exchange ou dans la fenêtre App Analyse, il convient de déterminer l'environnement d'exécution pendant l'exécution du script. Vous pouvez utiliser ces informations pour prendre des décisions quant aux commandes à exécuter à partir de l'emplacement d'exécution du script.

Utilisez la fonction `FTYPE()` pour déterminer l'emplacement d'exécution du script :

```
FTYPE("ax_main") = "b"
```

Si le script s'exécute dans Analytics Exchange ou dans la fenêtre App Analyse, l'expression est évaluée sur vrai (T). Pour les scripts s'exécutant dans Analytics, l'expression est évaluée sur faux (F). Pour plus d'informations, consultez la section "Fonction `FTYPE()`" Page 607.

## Identifier l'utilisateur exécutant le script sur AX Serveur

Pour les outils d'analyse exécutés sur AX Serveur, vous pouvez utiliser la variable système `AXRunByUser` afin d'identifier le nom de l'utilisateur qui exécute actuellement le script au format `domaine\nom_utilisateur` :

```
EXTRACT FIELDS TIME() AS "Heure"; DATE() AS "Date"; AXRunByUser AS "Utilisateur actuel"  
TO R_RunRecord APPEND
```

### Remarque

`AXRunByUser` est uniquement disponible lors de l'exécution des scripts d'outil d'analyse sur AX Serveur. La variable n'est pas reconnue lors de l'exécution de scripts dans Analytics.

# En-têtes des outils d'analyse et balises

Un en-tête d'outil d'analyse est une série de balises insérée dans un bloc de commentaire au début d'un script Analytics. Les balises des outils d'analyse indiquent des paramètres d'entrée renseignés par un utilisateur avant la planification ou l'exécution d'un outil d'analyse ainsi que des paramètres de sortie.

Un en-tête d'outil d'analyse est requis pour tout script d'outil d'analyse que vous prévoyez d'exécuter dans Robots, sur AX Serveur ou dans la fenêtre App Analyse.

## Définition des en-têtes des outils d'analyse

Les en-têtes d'outils d'analyse doivent être définis dans un bloc de commentaire qui commence à la première ligne du script. Dans l'en-tête des outils d'analyse, les balises peuvent se trouver dans n'importe quel ordre, à l'exception des balises suivantes :

- la balise `ANALYTIC`, qui doit se trouver en première position ;
- les balises `FIELD`, qui doivent suivre immédiatement la balise `TABLE` associée

### Exemple

Cet en-tête d'outil d'analyse identifie une table et un champ à utiliser dans le script, ainsi qu'un paramètre de date de début :

```
COMMENT
//ANALYTIC Identifier les chèques manquants
  Cet outil d'analyse identifie les numéros de chèque manquants
//TABLE v_table_paiements Table paiements
  Sélectionner une table qui répertorie les paiements et qui inclut une colonne Numéro de chèque
//FIELD v_chèque_num CN Numéro de chèque
  Sélectionnez le champ qui contient les numéros de chèque
//PARAM v_date_début D OPTIONAL Date de début (facultatif)
  Entrez la date de début de l'analyse
END
```

## Format des balises

Chaque balise de l'en-tête utilise le format suivant :

```
//NomBalise attributs TexteDescriptif
```

L'indicateur de balise // doit être le premier caractère non espace blanc sur la ligne du script. La balise doit suivre immédiatement l'indicateur de balise, sans espace ni caractère entre les deux.

## Conventions relatives aux balises

Composant	Convention
Nom des balises	Les noms des balises ne sont pas sensibles à la casse. À la différence des noms de fonctions et de commandes Analytics, les noms des balises ne peuvent pas être abrégés.
Attributs des balises	Lorsque vous spécifiez des valeurs pour une balise, vous pouvez inclure des espaces et éventuellement mettre la valeur entre guillemets.
Descriptions des balises	Les descriptions sont facultatives. Si une description est indiquée, elle peut être contenue sur plusieurs lignes, mais les sauts de ligne ne sont pas conservés dans les applications clientes.

## Spécifier les valeurs d'introduction de test dans Analytics

Vous pouvez utiliser un opérateur d'affectation spécifique ( := ) pour spécifier des valeurs de test pour des balises d'outil d'analyse nécessitant une définition :

- FILE
- TABLE
- FIELD
- PARAM

Utilisez cette syntaxe pour tester des scripts d'analyse dans Analytics :

```
//TABLE v_TableAnalyse "Table à classer" := "Trans_Mai"
```

Lorsque le script s'exécute dans Analytics, le paramètre prend la valeur spécifiée dans l'affectation. Lorsque l'outil d'analyse s'exécute dans une application cliente, la valeur de test est ignorée et les paramètres d'entrée définis par l'utilisateur sont utilisés.

Vous devez laisser une espace entre l'opérateur d'affectation et le nom de la balise qui le précède. Les valeurs d'affectation doivent utiliser le bon qualificatif pour le type de données requis comme partout dans Analytics. Pour plus d'informations, consultez la section "Types de données" Page 21.



# Liste complète des balises d'outils d'analyse disponibles

Balise	Description
"ANALYTIC" Page 891	Désigne un script Analytics comme outil d'analyse pouvant s'exécuter dans Robots, sur AX Serveur ou dans la fenêtre App Analyse.
<b>Balises d'entrée</b>	
"FILE" Page 893	Indique un fichier non-Analytics, comme un fichier Excel ou un fichier délimité, qui fournit une valeur d'introduction pour un outil d'analyse s'exécutant dans Robots ou sur AX Serveur. <ul style="list-style-type: none"> <li>Robots le fichier doit être situé dans l'onglet <b>Entrée/Sortie</b> du même robot que l'outil d'analyse</li> <li>AX Serveur le fichier doit être situé dans le sous-répertoire <b>Fichiers associés</b> dans le dossier à l'emplacement de l'outil d'analyse</li> </ul>
"TABLE" Page 895	Définit une table Analytics que l'utilisateur sélectionne comme valeur d'introduction pour un outil d'analyse.  La balise TABLE peut être suivie de zéro ou de plusieurs balises FIELD entrées sur des lignes qui se suivent.
"FIELD" Page 897	Définit un champ que l'utilisateur sélectionne comme valeur d'introduction pour l'outil d'analyse.  Le champ doit faire partie de la table définie dans la balise TABLE qui précède. La première balise FIELD doit immédiatement suivre une balise TABLE et peut être suivie d'autres balises FIELD entrées sur des lignes qui se suivent.
"PARAM" Page 899	Crée un paramètre d'entrée pour un outil d'analyse et définit les conditions requises pour la valeur d'introduction.  Un paramètre d'entrée est un espace réservé qui permet à l'utilisateur de spécifier la valeur réelle lors de la planification ou de l'exécution d'un outil d'analyse.
"PASSWORD" Page 912	Crée un paramètre d'entrée de mot de passe pour un outil d'analyse. Le paramètre fournit le stockage crypté d'un mot de passe pour une utilisation ultérieure dans une commande ACLS-crypt.  L'utilisateur est invité à spécifier la valeur de mot de passe requise lorsqu'il planifie ou lance un outil d'analyse pour qu'aucune intervention de l'utilisateur ne soit requise pendant l'exécution du script.
<b>Balises de sortie</b>	
"DATA" Page 915	Indique qu'une table de sortie Analytics par un outil d'analyse est copiée dans le sous-répertoire de données (un emplacement de stockage) dans l'environnement de déploiement  En général, vous stockez les tables Analytics pour qu'elles puissent être utilisées comme tables d'entrée pour des outils d'analyse ultérieurs.
"RESULT" Page 919	Indique les résultats de sortie des outils d'analyse que vous souhaitez mettre à disposition des

Balise	Description
	<p>utilisateurs finaux dans les applications clientes.</p> <p>Les résultats de sortie, s'ils existent, ne sont pas automatiquement mis à disposition. Vous devez utiliser une balise RESULT distincte pour chaque élément de résultat que vous souhaitez mettre à disposition.</p>
"PUBLISH" Page 924	Indique un fichier contenant des métadonnées définissant les tables Analytics à publier dans AX Exception à la fin du traitement d'un outil d'analyse.

# ANALYTIC

Désigne un script Analytics comme outil d'analyse pouvant s'exécuter dans Robots, sur AX Serveur ou dans la fenêtre App Analyse.

## Syntaxe

```
//ANALYTIC <TYPE IMPORT|PREPARE|ANALYSIS> nom
<description>
```

## Paramètres

Nom	Description
TYPE optionnel	<p>Identifie le type de script d'outil d'analyse comme l'un des trois types suivants :</p> <ul style="list-style-type: none"> <li>◦ IMPORT : récupère les données d'une source de données. La sortie d'un outil d'analyse d'importation est une table de données brute.</li> <li>◦ PREPARE : transforme des données brutes de façon à les adapter à l'analyse. La sortie d'un outil d'analyse de préparation est une table d'analyse.</li> <li>◦ ANALYSIS : effectue des tests sur les données des tables d'analyse. La sortie d'un outil d'analyse d'analyse est une ou plusieurs tables de résultats.</li> </ul> <p>Les outils d'analyse ayant un TYPE spécifié sont organisés dans la zone <b>Importation</b>, <b>Préparation</b> ou <b>Analyse</b> correspondante dans Robots, dans le Client Web d'AX et dans la fenêtre App Analyse. Ce placement guide l'utilisateur dans la séquence appropriée pour l'exécution de l'outil d'analyse. Ni la séquence ni le type de fonctionnalité au sein de l'outil d'analyse n'est forcé.</p> <p>Si vous ignorez TYPE, l'outil d'analyse apparaît dans la section <b>Analyse</b>.</p>
<i>nom</i>	<p>Nom de l'outil d'analyse.</p> <p>Le nom identifie l'outil d'analyse dans les applications clientes. Le nom de l'outil d'analyse est distinct du nom du script que vous spécifiez dans Analytics lorsque vous créez le script à l'origine.</p> <p><b>Remarque</b></p> <p>Les outils d'analyse du même projet ou de la même App Analyse doivent porter un nom unique. Si le même <i>nom</i> est utilisé dans au moins deux outils d'analyse, une erreur se produira lorsque vous essaieriez de transmettre les scripts d'outil d'analyse ou bien d'importer ou d'ouvrir l'App Analyse.</p> <p>Les caractères interdits dans les noms de fichiers Windows (&lt;&gt; : " / \   ? * ) ne doivent pas être utilisés dans les noms des outils d'analyse, car ils entraîneraient une erreur empêchant l'exportation des résultats analytiques. N'utilisez pas la valeur TYPE comme nom.</p> <p>Dans les applications clientes, les noms apparaissent dans l'ordre alphanumérique. Pour</p>

Nom	Description
	guider les utilisateurs dans la séquence qui convient afin d'exécuter plusieurs outils d'analyse dans un seul robot ou dans une seule App Analyse, il est possible d'ajouter un préfixe pour classer les noms des outils d'analyse au sein de chaque zone. Par exemple : 01_analyse_BDC, 02_analyse_factures, etc. La séquence suggérée par l'ordre des noms n'est pas forcée.
<i>description</i> optionnel	Description de l'outil d'analyse ou toute autre information susceptible d'être nécessaire à l'utilisateur pour exécuter correctement l'outil d'analyse.  La description apparaît avec l'outil d'analyse dans les applications clientes. La description peut être contenue sur plusieurs lignes, mais aucune ligne ne peut être sautée. La description doit être entrée sur la ligne située en dessous de la balise ANALYTIC associée.

## Exemples

### En-tête de l'outil d'analyse

L'en-tête d'outil d'analyse suivant contient un nom et une description de l'outil d'analyse :

```
COMMENT
//ANALYTIC Identifier les chèques manquants
Cet outil d'analyse identifie les numéros de chèque manquants.
END
```

### En-tête d'outil d'analyse avec type

L'en-tête d'outil d'analyse suivant spécifie un outil d'analyse de préparation avec une description de ce que le script fait :

```
COMMENT
//ANALYTIC TYPE PREPARE Standardiser données adresse
Cet outil d'analyse nettoie et standardise le champ d'adresse en préparation pour une analyse de doublons.
END
```

## Remarques

Une commande ACLScript `COMMENT` doit être entrée sur la première ligne du script, suivie de la balise `ANALYTIC` sur la deuxième ligne. Si la balise `ANALYTIC` est utilisée à un autre emplacement, elle est ignorée.

Un ou plusieurs scripts d'un projet Analytics peuvent inclure une balise `ANALYTIC`.

# FILE

Indique un fichier non-Analytics, comme un fichier Excel ou un fichier délimité, qui fournit une valeur d'introduction pour un outil d'analyse s'exécutant dans Robots ou sur AX Serveur.

- Robots le fichier doit être situé dans l'onglet **Entrée/Sortie** du même robot que l'outil d'analyse
- AX Serveur le fichier doit être situé dans le sous-répertoire **Fichiers associés** dans le dossier à l'emplacement de l'outil d'analyse

## Remarque

Pour spécifier un fichier d'entrée non-Analytics pour un outil d'analyse exécuté dans la fenêtre App Analyse, consultez la section "PARAM" Page 899.

## Syntaxe

```
//FILE nom_fichier
```

## Paramètres

Nom	Description
<i>nom_fichier</i>	<p>Le nom de l'élément dans le robot ou dans le sous-répertoire <b>Fichiers associés</b> à utiliser comme fichier d'entrée pour un outil d'analyse. Le <i>nom de fichier</i> ne peut pas contenir de chemin.</p> <p>Les caractères génériques sont pris en charge dans le nom de fichier. Utilisez un astérisque simple (*) pour remplacer par zéro ou plusieurs caractères.</p> <p>Par exemple :</p> <ul style="list-style-type: none"> <li>◦ Fac12* correspond aux éléments suivants : Fac12, Fac123 et Fac1234</li> <li>◦ *.* correspond à tous les fichiers de toutes les extensions dans le robot ou dans le dossier <b>Fichiers associés</b></li> <li>◦ Fac_*. * correspond à Fac_Jan.pdf et à Fac_Fév.xls</li> </ul> <p><b>Astuce</b></p> <p>Vous pouvez utiliser la balise //FILE pour référencer un fichier de préférences Analytics .prf. Lorsque vous faites cela, le fichier de préférences dans le sous-répertoire <b>Fichiers associés</b> est utilisé pour définir les paramètres de l'environnement d'exécution plutôt que le fichier de préférences global sur AX Serveur. Le fichier de préférences doit provenir de la dernière version d'Analytics compatible avec votre installation Analytics Exchange.</p>

# Exemples

## Exemples basiques

Indique un fichier spécifique :

```
//FILE ComptesContrôlés.csv
```

Indique tous les fichiers CSV commençant par "Comptes" :

```
//FILE Comptes*.csv
```

Indique tous les fichiers :

```
//FILE *.*
```

## Exemples avancés

### Importer des données à partir d'un fichier inclus

Vous exécutez une analyse mensuelle des données des employés sur AX Serveur. L'un des outils d'analyse de l'App Analyse importe les données à analyser à partir d'un fichier délimité placé dans le dossier Fichiers associés chaque mois :

```
COMMENT
//ANALYTIC TYPE IMPORT employé_import
  Importe les enregistrements des employés à partir d'un fichier délimité stocké dans le dossier Fichiers
  associés.
//FILE Employés.csv
END
IMPORT DELIMITED TO Employés "Employés.fil" FROM "Employés.csv" 0 SEPARATOR ","
QUALIFIER "" CONSECUTIVE STARTLINE 1 KEPTITLE FIELD "Prénom" C AT 1 DEC 0 WID 11
PIC "" AS "Prénom" FIELD "Nom" C AT 12 DEC 0 WID 12 PIC "" AS "Nom"
```

## Remarques

La balise FILE n'est pas prise en charge dans l'utilisation des outils d'analyse exécutés dans la fenêtre App Analyse. Pour spécifier un fichier d'entrée pour les outils d'analyse exécutés dans la fenêtre App Analyse, utilisez la balise PARAM. Pour plus d'informations, consultez la section "PARAM" Page 899.

# TABLE

Définit une table Analytics que l'utilisateur sélectionne comme valeur d'introduction pour un outil d'analyse.

La balise `TABLE` peut être suivie de zéro ou de plusieurs balises `FIELD` entrées sur des lignes qui se suivent.

## Remarque

La balise `TABLE` nécessite qu'une table préexiste dans l'emplacement de stockage afin d'être disponible pour être sélectionnée. Pour plus d'informations, consultez la section "DATA" Page 915.

## Syntaxe

```
//TABLE id nom
<description>
```

## Paramètres

Nom	Description
<i>id</i>	Variable qui stocke le nom de la table d'entrée sélectionnée par l'utilisateur. Utilisez cette valeur dans le script de l'outil d'analyse pour faire référence à la table.
<i>nom</i>	Nom à associer à la table. La valeur est affichée dans les applications clientes quand l'utilisateur exécutant l'outil d'analyse est invité à sélectionner le champ.
<i>description</i> optionnel	Texte descriptif qui spécifie l'objectif de la table. La description peut être contenue sur plusieurs lignes, mais aucune ligne ne peut être sautée. La valeur est affichée dans les applications clientes quand l'utilisateur est invité à sélectionner la table. La description peut inviter l'utilisateur à sélectionner la table qui convient. Par exemple, « Sélectionner la table qui contient les informations relatives à la paie ». La description doit être entrée sur la ligne située en dessous de la balise <code>TABLE</code> associée.

# Exemples

## Exemples basiques

Balise TABLE avec la description permettant d'aider l'utilisateur à sélectionner la table d'entrée qui convient :

```
//TABLE v_table_paiements Table paiements  
Sélectionnez une table qui répertorie les paiements et qui inclut une colonne Numéro de chèque.
```

## Exemples avancés

### Utilisation d'une table définie dans une balise TABLE dans le script

Le script suivant exécute une commande AGE sur une table sélectionnée par l'utilisateur à partir des tables de données du projet :

```
COMMENT  
//ANALYTIC exemple_script  
//TABLE v_table_paiements Table paiements  
Sélectionnez une table qui répertorie les paiements et qui inclut une colonne Numéro de chèque.  
END  
  
OPEN %v_table_paiements%  
AGE ON date_paiement CUTOFF 20141231 INTERVAL 0;30;60;90;120;10000 SUBTOTAL Mon-  
tant_Paiement TO r_sortie  
CLOSE %v_table_paiements%
```



# FIELD

Définit un champ que l'utilisateur sélectionne comme valeur d'introduction pour l'outil d'analyse.

Le champ doit faire partie de la table définie dans la balise TABLE qui précède. La première balise FIELD doit immédiatement suivre une balise TABLE et peut être suivie d'autres balises FIELD entrées sur des lignes qui se suivent.

## Remarque

La balise TABLE nécessite qu'une table préexiste dans l'emplacement de stockage afin d'être disponible pour être sélectionnée. Pour plus d'informations, consultez la section "DATA" Page 915.

## Syntaxe

```
//FIELD id type nom
<description>
```

## Paramètres

Nom	Description
<i>id</i>	Variable qui stocke le nom du champ d'entrée sélectionné par l'utilisateur. Utilisez cette valeur dans le script de l'outil d'analyse pour faire référence au champ.
<i>type</i>	Types de champs pouvant être sélectionnés. N'importe quel type, ou association de types, de la liste suivante peut être spécifié : <ul style="list-style-type: none"> <li>○ C : données de type caractère</li> <li>○ N : données numériques</li> <li>○ D : sous-type date, DateHeure ou heure du type de données DateHeure</li> <li>○ L : données logiques</li> </ul> Les champs calculés dans une table peuvent être sélectionnés indépendamment du <i>type</i> spécifié.
<i>nom</i>	Nom à associer au champ. La valeur est affichée dans les applications clientes quand l'utilisateur est invité à sélectionner le champ.
<i>description</i> optionnel	Texte descriptif qui spécifie l'objectif du champ. La description peut être contenue sur plusieurs lignes, mais aucune ligne ne peut être sautée. La valeur est affichée dans les applications clientes quand l'utilisateur est invité à sélectionner le champ. La description peut inviter l'utilisateur à sélectionner le champ qui convient. Par exemple, « Sélectionner le champ qui contient le montant du paiement ».

Nom	Description
	La description doit être entrée sur la ligne située en dessous de la balise FIELD associée.

## Exemples

### Exemples basiques

Indique un champ de type caractère :

```
//FIELD v_nom C Champ nom
```

Indique un champ caractère ou numérique :

```
//FIELD v_num_fac CN Numéro de facture
```

### Exemples avancés

#### TABLE avec deux balises FIELD pour l'accompagner.

L'en-tête d'outil d'analyse suivant autorise l'utilisateur à indiquer deux champs d'entrée à partir de la table `v_table_paiements` lorsque le script s'exécute :

```
COMMENT
//ANALYTIC outil d'analyse test
//TABLE v_table_paiements Table paiements
  Sélectionner une table qui répertorie les paiements et qui inclut une colonne Numéro de chèque.
//FIELD v_chèque_num CN Champ Numéro de chèque
//FIELD v_date_paiement D Champ Date de paiement
  Sélectionner la colonne qui contient la date de paiement du chèque.
END

OPEN %v_table_paiements%
EXTRACT FIELDS %v_chèque_num%, %v_date_paiement% TO t_analyse
```

# PARAM

Crée un paramètre d'entrée pour un outil d'analyse et définit les conditions requises pour la valeur d'introduction.

Un paramètre d'entrée est un espace réservé qui permet à l'utilisateur de spécifier la valeur réelle lors de la planification ou de l'exécution d'un outil d'analyse.

## Syntaxe

```
//PARAM id type <OPTIONAL> <MULTI> <SEPARATOR valeur> <QUALIFIER valeur> <VALUES
liste_valeur> libellé
<description>
```

## Paramètres

Nom	Description
<i>id</i>	<p>Variable qui stocke la ou les valeurs d'introduction de l'outil d'analyse sélectionnées ou spécifiées par l'utilisateur.</p> <p>Par exemple :</p> <ul style="list-style-type: none"> <li>○ v_date_début</li> <li>○ v_régions</li> <li>○ v_fichier_entrée</li> </ul> <p>Sert également d'identifiant unique du paramètre.</p> <p><b>Remarque</b></p> <p>Lorsqu'un outil d'analyse est exécuté, la variable est créée uniquement si l'utilisateur fournit une valeur d'introduction. Si un paramètre est facultatif et que l'utilisateur l'ignore, la variable n'est pas créée.</p> <p>Si la logique ultérieure de l'outil d'analyse requiert l'existence de la variable, vous pouvez tester la présence de son existence et si elle n'existe pas, la créer et l'initialiser. Pour plus d'informations, consultez la section "Conception de paramètres d'entrée facultatifs" Page 906.</p>
<i>type</i>	<p>Type de données du paramètre, qui contrôle le type de valeurs d'introduction pouvant être saisies.</p> <p>Les types suivants peuvent être spécifiés à l'aide de lettres en majuscules :</p> <ul style="list-style-type: none"> <li>○ C : données de type caractère</li> <li>○ N : données numériques</li> <li>○ D : sous-type date du type de données DateHeure</li> <li>○ DT : sous-type DateHeure du type de données DateHeure</li> <li>○ T : sous-type heure du type de données DateHeure</li> <li>○ L : données logiques</li> </ul>

Nom	Description						
	<p><b>Remarque</b></p> <p>Pour qu'un outil d'analyse s'exécute correctement, il est nécessaire de qualifier des valeurs d'introduction de caractère.</p> <p><b>Fonctionnement de PARAM... F</b></p> <p>Vous pouvez aussi spécifier qu'un utilitaire de chargement de fichiers ou qu'un explorateur de fichiers Windows s'ouvre.</p> <ul style="list-style-type: none"> <li>o F : ouvre un utilitaire de chargement de fichiers ou un explorateur de fichiers Windows et autorise un utilisateur à sélectionner un fichier d'entrée non-Analytics pour l'outil d'analyse lors d'une exécution dans le Client Web d'AX ou la fenêtre App Analyse</li> </ul> <p>À la sélection, le nom du fichier est automatiquement entré sous forme de valeur d'introduction Caractère. Indiquez F uniquement. N'indiquez pas F C.</p> <p>Par exemple :</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <pre>//PARAM v_fichier_entrée F...</pre> </div> <p>Pour plus d'informations, consultez la section "Spécifier ou sélectionner un fichier d'entrée non-Analytics pour un outil d'analyse" Page 910.</p> <p><b>Remarque</b></p> <p>Un <i>type</i> de F n'est pas pris en charge dans l'utilisation des outils d'analyse exécutés dans Robots ou dans AX Client. Pour spécifier un fichier d'entrée pour ces environnements, utilisez la balise FILE. Pour plus d'informations, consultez la section "FILE" Page 893.</p>						
OPTIONAL optionnel	<p>Indique que le paramètre est facultatif et qu'il n'est pas nécessaire que l'utilisateur entre une valeur.</p> <p>Pour plus d'informations, consultez la section "Conception de paramètres d'entrée facultatifs" Page 906.</p>						
MULTI optionnel	<p>Indique que le paramètre accepte plusieurs valeurs d'introduction.</p> <p>MULTI peut être utilisée avec ou sans l'option VALUES :</p> <table border="1" data-bbox="505 1354 1463 1577"> <tbody> <tr> <td data-bbox="505 1354 695 1417">MULTI ✓</td> <td data-bbox="695 1354 1463 1417" rowspan="2">L'utilisateur peut sélectionner une ou plusieurs valeurs dans une liste de valeurs.</td> </tr> <tr> <td data-bbox="505 1417 695 1470">VALUES ✓</td> </tr> <tr> <td data-bbox="505 1470 695 1533">MULTI ✓</td> <td data-bbox="695 1470 1463 1533" rowspan="2">L'utilisateur peut saisir manuellement une ou plusieurs valeurs.</td> </tr> <tr> <td data-bbox="505 1533 695 1577">VALUES ✗</td> </tr> </tbody> </table> <p>Pour plus d'informations, consultez la section "Résumé des options MULTI et VALUES" Page 907.</p> <p>MULTI ne peut pas être utilisé si le <i>type</i> est L (logique).</p> <p><b>Plusieurs valeurs d'introduction caractères</b></p> <p>Si vous spécifiez MULTI et que le <i>type</i> est C (Caractère), vous pouvez aussi spécifier les options SEPARATOR et QUALIFIER pour insérer automatiquement des séparateurs (délimiteurs) et des identificateurs de texte dans une chaîne de valeurs d'introduction.</p>	MULTI ✓	L'utilisateur peut sélectionner une ou plusieurs valeurs dans une liste de valeurs.	VALUES ✓	MULTI ✓	L'utilisateur peut saisir manuellement une ou plusieurs valeurs.	VALUES ✗
MULTI ✓	L'utilisateur peut sélectionner une ou plusieurs valeurs dans une liste de valeurs.						
VALUES ✓							
MULTI ✓	L'utilisateur peut saisir manuellement une ou plusieurs valeurs.						
VALUES ✗							

Nom	Description
	<p><b>Remarque</b></p> <p>Pour qu'un outil d'analyse s'exécute correctement, il est nécessaire de délimiter et de qualifier plusieurs valeurs d'introduction caractères. Les séparateurs et identificateurs peuvent être insérés automatiquement ou manuellement par l'utilisateur.</p>
SEPARATOR <i>valeur</i> optionnel	<p>SEPARATOR peut être uniquement utilisé lorsque MULTI est spécifié et que le <i>type</i> est C (caractère).</p> <p>Indique qu'un séparateur caractère est inséré automatiquement entre plusieurs valeurs d'introduction caractères, créant une liste délimitée transmise à l'outil d'analyse pour son traitement.</p> <p><i>valeur</i> spécifie le séparateur caractère à utiliser. Un séparateur, ou délimiteur, couramment utilisé est la virgule ,.</p> <p>Si SEPARATOR est omis, une seule espace est utilisée comme séparateur par défaut. Le caractère espace ne peut pas être spécifié comme <i>valeur</i>.</p> <p>Pour plus d'informations, consultez la section "Délimitation et identification des valeurs d'introduction de caractères" Page 907.</p>
QUALIFIER <i>valeur</i> optionnel	<p>QUALIFIER peut uniquement être utilisé lorsque MULTI est spécifié et que le <i>type</i> est C (caractère).</p> <p>Indique qu'un caractère identificateur de texte est inséré automatiquement au début et à la fin de chaque valeur d'introduction caractère dans une liste délimitée transmise à l'outil d'analyse pour son traitement. Tout texte inclus dans les caractères d'identificateur de texte est traité comme du texte brut.</p> <p>La <i>valeur</i> spécifie le caractère d'identificateur à utiliser. Un identificateur fréquemment utilisé est le guillemet simple '.</p> <p>Si QUALIFIER est ignoré, aucun identificateur par défaut n'est utilisé. Le caractère espace ne peut pas être spécifié comme <i>valeur</i>.</p> <p>Pour plus d'informations, consultez la section "Délimitation et identification des valeurs d'introduction de caractères" Page 907.</p> <p><b>Remarque</b></p> <p>Les paramètres d'entrée des outils d'analyse ne prennent actuellement pas en charge l'utilisation du double guillemet (") comme identificateur de texte. À la place, vous pouvez utiliser le guillemet simple ('). Spécifier un identificateur guillemet double entraînera le dysfonctionnement de la balise PARAM.</p>
VALUES <i>liste_valeurs</i> optionnel	<p>Liste de valeurs dans laquelle l'utilisateur peut faire sa sélection lors de l'exécution de l'outil d'analyse.</p> <p>Utilisez la syntaxe suivante pour spécifier les valeurs :</p> <div data-bbox="483 1665 1446 1738" style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <p>VALUES  Valeur 1 Valeur 2 Valeur 3 Valeur n </p> </div> <p>VALUES peut être utilisée avec ou sans l'option MULTI :</p>

Nom	Description												
	<table border="1" data-bbox="505 270 1464 495"> <tr> <td data-bbox="505 270 695 384">VALUES ✓ MULTI ✓</td> <td data-bbox="695 270 1464 384">L'utilisateur peut sélectionner une ou plusieurs valeurs dans la liste de valeurs.</td> </tr> <tr> <td data-bbox="505 384 695 495">VALUES ✓ MULTI ✗</td> <td data-bbox="695 384 1464 495">L'utilisateur peut sélectionner une seule valeur dans la liste de valeurs.</td> </tr> </table> <p data-bbox="505 512 1442 569">Pour plus d'informations, consultez la section "Résumé des options MULTI et VALUES" Page 907.</p> <p data-bbox="505 585 922 613"><b>Format des valeurs dans <i>liste_valeurs</i></b></p> <table border="1" data-bbox="505 630 1464 1392"> <tr> <td data-bbox="505 630 695 753">Valeurs de type caractère</td> <td data-bbox="695 630 1464 753"> <ul style="list-style-type: none"> <li>peuvent contenir des espaces et des signes de ponctuation</li> </ul> </td> </tr> <tr> <td data-bbox="505 753 695 928">Valeurs numériques</td> <td data-bbox="695 753 1464 928"> <ul style="list-style-type: none"> <li>peuvent être positives ou négatives.</li> <li>doivent être précisées à l'aide de la notation décimale, et sans séparateur pour les milliers</li> </ul>           Par exemple, 1500,00 ou -1500,00         </td> </tr> <tr> <td data-bbox="505 928 695 1297">Valeurs DateHeure</td> <td data-bbox="695 928 1464 1297"> <ul style="list-style-type: none"> <li>Les valeurs de type <b>Date</b> : doivent être indiquées à l'aide du format MM/JJ/AAAA Par exemple, 12/31/2014</li> <li>Les valeurs de type <b>DateHeure</b> : doivent être indiquées à l'aide du format MM/JJ/AAAA hh:mm:ss Par exemple, 12/31/2014 23:59:59</li> <li>Les valeurs de type <b>heure</b> : doivent être indiquées à l'aide du format hh:mm:ss Par exemple, 23:59:59</li> </ul> </td> </tr> <tr> <td data-bbox="505 1297 695 1392">Valeurs logiques</td> <td data-bbox="695 1297 1464 1392">VALUES ne peuvent pas être utilisées si le <i>type</i> est L (Logique)</td> </tr> </table>	VALUES ✓ MULTI ✓	L'utilisateur peut sélectionner une ou plusieurs valeurs dans la liste de valeurs.	VALUES ✓ MULTI ✗	L'utilisateur peut sélectionner une seule valeur dans la liste de valeurs.	Valeurs de type caractère	<ul style="list-style-type: none"> <li>peuvent contenir des espaces et des signes de ponctuation</li> </ul>	Valeurs numériques	<ul style="list-style-type: none"> <li>peuvent être positives ou négatives.</li> <li>doivent être précisées à l'aide de la notation décimale, et sans séparateur pour les milliers</li> </ul> Par exemple, 1500,00 ou -1500,00	Valeurs DateHeure	<ul style="list-style-type: none"> <li>Les valeurs de type <b>Date</b> : doivent être indiquées à l'aide du format MM/JJ/AAAA Par exemple, 12/31/2014</li> <li>Les valeurs de type <b>DateHeure</b> : doivent être indiquées à l'aide du format MM/JJ/AAAA hh:mm:ss Par exemple, 12/31/2014 23:59:59</li> <li>Les valeurs de type <b>heure</b> : doivent être indiquées à l'aide du format hh:mm:ss Par exemple, 23:59:59</li> </ul>	Valeurs logiques	VALUES ne peuvent pas être utilisées si le <i>type</i> est L (Logique)
VALUES ✓ MULTI ✓	L'utilisateur peut sélectionner une ou plusieurs valeurs dans la liste de valeurs.												
VALUES ✓ MULTI ✗	L'utilisateur peut sélectionner une seule valeur dans la liste de valeurs.												
Valeurs de type caractère	<ul style="list-style-type: none"> <li>peuvent contenir des espaces et des signes de ponctuation</li> </ul>												
Valeurs numériques	<ul style="list-style-type: none"> <li>peuvent être positives ou négatives.</li> <li>doivent être précisées à l'aide de la notation décimale, et sans séparateur pour les milliers</li> </ul> Par exemple, 1500,00 ou -1500,00												
Valeurs DateHeure	<ul style="list-style-type: none"> <li>Les valeurs de type <b>Date</b> : doivent être indiquées à l'aide du format MM/JJ/AAAA Par exemple, 12/31/2014</li> <li>Les valeurs de type <b>DateHeure</b> : doivent être indiquées à l'aide du format MM/JJ/AAAA hh:mm:ss Par exemple, 12/31/2014 23:59:59</li> <li>Les valeurs de type <b>heure</b> : doivent être indiquées à l'aide du format hh:mm:ss Par exemple, 23:59:59</li> </ul>												
Valeurs logiques	VALUES ne peuvent pas être utilisées si le <i>type</i> est L (Logique)												
<i>libellé</i>	<p data-bbox="505 1430 1154 1457">Libellé de l'interface utilisateur correspondant au paramètre.</p> <p data-bbox="505 1474 1284 1501">Dans les applications clientes, <i>libellé</i> est affiché avec le champ d'entrée.</p>												
<i>description</i> optionnel	<p data-bbox="505 1541 1344 1568">Texte descriptif qui fournit des informations supplémentaires sur le paramètre.</p> <p data-bbox="505 1585 1377 1612">Dans les applications clientes, la <i>description</i> est affichée avec le champ d'entrée.</p> <p data-bbox="505 1629 1442 1686"><i>Description</i> peut fournir des instructions qui aident l'utilisateur. Par exemple, « Entrer la date de référence pour la période de paie ».</p> <p data-bbox="505 1703 1458 1787">La <i>description</i> doit être entrée sur la ligne suivant la balise PARAM associée. Le texte peut être contenu sur plusieurs lignes, mais aucune ligne ne peut être sautée. Les sauts de ligne ne sont pas conservés dans le cas de l'affichage dans des applications clientes.</p>												

# Exemples

## Exemples basiques

Autorise l'utilisateur à éventuellement spécifier une plage de dates :

```
//PARAM v_date_début D OPTIONAL Date de début (facultatif)
  Entrez la date de début de l'analyse
//PARAM v_date_fin D OPTIONAL Date de fin (facultatif)
  Entrez la date de fin de l'analyse
```

Demande à l'utilisateur de sélectionner le nombre maximal de transactions à traiter :

```
//PARAM v_maxTrans N VALUES |250|500|750|1000| Transactions maximales à traiter
```

Demande à l'utilisateur de spécifier un ou plusieurs codes de catégories de marchands :

```
//PARAM v_codes C MULTI SEPARATOR , QUALIFIER ' MC Code(s) à inclure
  Spécifier un ou plusieurs codes de catégories de marchands. Appuyez sur « Entrée » après chaque
  code pour que chacun d'entre eux se trouve sur une ligne distincte. N'entourez pas les codes de guillemets.
```

Demande à l'utilisateur de sélectionner un ou plusieurs codes de catégories de marchands :

```
//PARAM v_codes C MULTI SEPARATOR , QUALIFIER ' VALUES |4121 Taxis/Limousines|5812 Restaurants|5813 Cafés - Boissons alcoolisées|5814 Fast foods| MC Codes à inclure
  Sélectionner un ou plusieurs codes de catégories de marchands.
```

## Exemples avancés

### Demander à un utilisateur de spécifier une plage de montants

Vous devez classer les enregistrements d'une table qui se trouvent compris entre une plage de montants minimums et maximums. Cette plage change occasionnellement ; vous fournissez donc les paramètres permettant à l'utilisateur exécutant l'outil d'analyse de définir la plage lors de la planification ou lors de l'exécution du script :

```
COMMENT
//ANALYTIC test_outil_analyse
//PARAM v_montant_min N Montant minimum
```

```

Saisissez un montant minimum
//PARAM v_montant_max N Montant maximum
Saisissez un montant maximum
END

CLASSIFY ON %v_ChampA% IF BETWEEN(AMOUNT; v_montant_min; v_montant_max)
SUBTOTAL AMOUNT TO "Classée_%v_TableAnalyse%.FIL"

```

## Autoriser l'utilisateur à éventuellement exclure un ou plusieurs numéros client

Vous devez classer les enregistrements d'une table, mais vous souhaitez donner à l'utilisateur la possibilité d'exclure certains clients de l'analyse.

Pour ce faire, vous fournissez un paramètre de type caractère facultatif. Votre script teste si oui ou non la valeur est fournie, et, si tel est le cas, les numéros de clients correspondants sont exclus de la commande Classifier :

```

COMMENT
//ANALYTIC test_outil_analyse
//PARAM v_num_client C OPTIONAL MULTI SEPARATOR , QUALIFIER ' Numéro(s) client à
exclure (facultatif)
Spécifier un ou plusieurs numéros clients. Appuyez sur « Entrée » après chaque numéro pour que
chaque numéro se trouve sur une ligne distincte. N'entourez pas les numéros de guillemets.
END

IF FTYPE("v_num_client") = "U" v_num_client = ""
GROUP IF v_num_client = ""
CLASSIFY ON %v_ChampA% SUBTOTAL AMOUNT TO "Classée_%v_TableAnalyse%.FIL"
ELSE
CLASSIFY ON %v_ChampA% IF NOT MATCH(CUSTNO; %v_num_client%) SUBTOTAL
AMOUNT TO "Classée_%v_TableAnalyse%.FIL"
END

```

## Autoriser l'utilisateur à sélectionner un fichier d'entrée (Client Web d'AX ou la fenêtre de l'App Analyse uniquement)

Vous distribuez une App Analyse à des collègues qui l'exécuteront dans la fenêtre App Analyse. Lorsqu'ils exécuteront le script d'outil d'analyse dans l'application, vous aimeriez leur fournir un navigateur de fichiers Windows pour sélectionner un fichier Microsoft Excel à partir duquel importer des données :

```

COMMENT
//ANALYTIC test_outil_analyse

```



```
//PARAM v_fichier_entrée F Fichier d'entrée
  Sélectionner un fichier d'entrée
END

IMPORT EXCEL TO Trans_Mai_brut Trans_Mai_brut.fil FROM "%v_fichier_entrée%" TABLE
"Trans2_Mai$" CHARMAX 100 KEEPTITLE
```

## Demander à l'utilisateur de spécifier un chemin de fichier d'entrée et un nom de fichier (la fenêtre App Analyse uniquement)

Vous distribuez une App Analyse à des collègues qui l'exécuteront dans la fenêtre App Analyse. Lorsqu'ils exécuteront le script d'outil d'analyse dans l'application, vous aimeriez qu'ils spécifient un chemin d'accès au fichier et un nom de fichier à utiliser comme fichier d'importation :

```
COMMENT
//ANALYTIC test_outil_analyse
//PARAM v_fichier_entrée C Chemin et nom du fichier d'entrée
  Entrer un chemin de fichier absolu et un nom de fichier, par exemple : C:\U-
  sers\username\Documents\ACL Data\Exemples de fichiers de données\ Trans_Mai.xls
END

IMPORT EXCEL TO Trans_Mai_brut Trans_Mai_brut.fil FROM "%v_fichier_entrée%" TABLE
"Trans2_Mai$" CHARMAX 100 KEEPTITLE
```

## Utilisation des valeurs par défaut pour des paramètres facultatifs

Vous créez un outil d'analyse extrayant les enregistrements des transactions vers une table de résultats. Vous souhaitez donner à l'utilisateur exécutant le script la possibilité de fournir une plage de dates ainsi qu'une liste d'entités pour filtrer les enregistrements à extraire.

Pour ce faire, créez trois paramètres facultatifs :

- v\_date\_début
- v\_date\_fin
- v\_liste\_entités

Dans les lignes d'ouverture du script, vous testez si ces valeurs sont définies. Si elles ne sont pas définies, vous définissez les valeurs par défaut des dates minimale et maximale ainsi qu'un indicateur par défaut à rechercher par test avec v\_liste\_entités.

Dans la commande EXTRACT, utilisez les valeurs pour filtrer les enregistrements :

```
COMMENT
//ANALYTIC test
  Cet outil d'analyse teste PARAM
```

```
//RESULT TABLE t_résultats
//PARAM v_date_début D OPTIONAL Entrez la date de début
//PARAM v_date_fin D OPTIONAL Entrez la date de fin
//PARAM v_liste_entités C MULTI OPTIONAL |entité1|entité2|
END

IF NOT ISDEFINED("v_date_début") v_date_début = `19000101`
IF NOT ISDEFINED("v_date_fin") v_date_fin = `99991231`
IF NOT ISDEFINED("v_liste_entités") v_liste_entités = ""tous""

EXTRACT FIELDS ALL TO t_résultats IF BETWEEN(date_transaction v_date_début v_date_fin)
AND (MATCH(champ_entité %v_liste_entités%) OR v_liste_entités = ""tous"")
```

## Remarques

### Conception de paramètres d'entrée facultatifs

Si vous utilisez OPTIONAL avec la balise PARAM, la variable associée au paramètre d'entrée d'outil d'analyse peut ou non être créée au moment de l'exécution de l'outil d'analyse :

- **variable créée automatiquement** : si l'utilisateur spécifie une valeur d'introduction
- **variable non créée** : si l'utilisateur ignore le paramètre facultatif et qu'il ne précise pas de valeur d'introduction

### Test de l'existence de la variable paramètre

Si la logique ultérieure de l'outil d'analyse dépend de sa capacité à évaluer le contenu de la variable paramètre, y compris l'évaluation d'un état vide ou null, vous devez tester l'existence de la variable du paramètre. Si la variable paramètre n'existe pas, vous devez la créer et l'initialiser sur null.

Utilisez la commande IF avec la fonction FTYPE( ) ou la fonction ISDEFINED( ) pour effectuer le test et créer la variable si elle n'existe pas :

```
IF FTYPE("nom_var") = "U" nom_var = ""
```

```
IF NOT ISDEFINED("nom_var") nom_var = ""
```

### Quand effectuer le test

Effectuez le test après l'en-tête de l'outil d'analyse et avant toute logique AnalyticsScript qui dépend de l'existence de la variable du paramètre.

## Résumé des options MULTI et VALUES

Le tableau ci-dessous résume l'effet des options MULTI et VALUES sur la commande de saisie de l'utilisateur dans l'interface utilisateur.

Commande de saisie de l'utilisateur (Robots)	Conception des paramètres	MULTI	VALUES
	Une valeur d'introduction unique saisie manuellement dans un champ	✗	✗
	Une ou plusieurs valeurs d'introduction saisies manuellement dans un champ	✓	✗
	Une valeur d'introduction unique sélectionnée dans une liste déroulante de valeurs	✗	✓
	Une ou plusieurs valeurs d'introduction sélectionnées dans une liste de contrôle de valeurs	✓	✓

## Délimitation et identification des valeurs d'introduction de caractères

Pour la bonne exécution de l'outil d'analyse, s'il y a plusieurs valeurs d'introduction de caractère, elles doivent être délimitées par un séparateur et identifiées.

## Éviter les identificateurs de texte imbriqués

Lorsque vous créez des paramètres d'entrée de caractère, et lorsque vous expliquez aux utilisateurs de l'outil d'analyse comment saisir des valeurs d'introduction de caractère, vous devez veiller à éviter de créer des identificateurs de texte redondants et imbriqués (des identificateurs à l'intérieur d'identificateurs). Des identificateurs au texte redondant entraîneront un dysfonctionnement du paramètre d'entrée.

## Méthodes d'insertion d'identificateurs de texte

Il existe quatre méthodes différentes permettant d'insérer des identificateurs de texte autour de valeurs d'introduction de caractère. En fonction de la méthode, un séparateur est également inséré entre les valeurs d'introduction.

Au fur et à mesure que vous développez un outil d'analyse, vous devrez peut-être tester différentes méthodes pour trouver ce qui fonctionne le mieux pour les valeurs de type caractère que les utilisateurs saisiront.

### Remarque

Il se peut qu'une ou plusieurs méthodes ne soient pas applicables, en fonction de votre utilisation des options MULTI et VALUES.

Chaque paramètre d'entrée ne doit utiliser **qu'une seule** de ces méthodes.

1	Utiliser SEPARATOR et QUALIFIER	<p>Incluez les options SEPARATOR et QUALIFIER dans la balise PARAM.</p> <p>Par exemple :</p> <pre>//PARAM v_régions C MULTI SEPARATOR ; QUALIFIER '</pre> <p>Non applicable si vous utilisez VALUES sans MULTI.</p> <p><b>Astuce</b></p> <p>Utilisez cette méthode à chaque fois que cela est possible. C'est celle qui demande le moins de travail et qui générera le moins d'erreurs.</p>
2	Spécifier manuellement les séparateurs et les identificateurs	<p>Nécessite que l'utilisateur de l'outil d'analyse spécifie manuellement des séparateurs et des identificateurs en plus des valeurs d'introduction réelles.</p> <p>Par exemple :</p> <pre>'Amérique du Nord';'Europe';'Asie'</pre> <p>Non applicable si vous utilisez VALUES avec ou sans MULTI.</p>
3	Inclure les identificateurs dans la liste_valeurs	<p>Incluez les identificateurs avec chaque valeur dans la <i>liste_valeurs</i> spécifiée avec l'option VALUES.</p> <p>Par exemple :</p> <pre>VALUES ['Asie']['Europe']['Moyen Orient']['Amérique du Nord']</pre>

		Non applicable si vous utilisez MULTI sans VALUES.
4	Insérer la variable du paramètre dans les identificateurs	<p>Dans la syntaxe du script Analytics, insérez la variable paramètre entre des identificateurs de texte</p> <p>Par exemple :</p> <pre>IF MATCH(REGIONS; "%v_régions%")</pre> <p>Utilisez cette méthode uniquement si vous utilisez VALUES sans MULTI.</p>
	<p><b>Remarque</b></p> <p>Les paramètres d'entrée des outils d'analyse ne prennent actuellement pas en charge l'utilisation du double guillemet (") comme identificateur de texte. Vous pouvez utiliser le guillemet simple (') plutôt que l'option QUALIFIER, dans la <i>liste_valeurs</i>, ou lorsque vous indiquez manuellement des identificateurs autour de valeurs d'introduction. Des guillemets doubles peuvent être utilisés en tant qu'identificateurs de texte dans le corps d'un script Analytics.</p>	

## Cas d'utilisation des différentes méthodes

Le tableau ci-dessous résume quand utiliser les différentes méthodes pour insérer des identificateurs de texte.

	MULTI ✓ VALUES ✗	MULTI ✗ VALUES ✓	MULTI ✓ VALUES ✓
<b>Méthode 1</b> Utiliser les options SEPARATOR et QUALIFIER	Si elles sont utilisées, ne pas utiliser la méthode 2	Non applicable	Si elles sont utilisées, ne pas utiliser la méthode 3
<b>Méthode 2</b> Spécifier manuellement les séparateurs et les identificateurs	Si elles sont utilisées, ne pas utiliser la méthode 1	Non applicable	Non applicable
<b>Méthode 3</b> Inclure les identificateurs dans la <i>liste_valeurs</i>	Non applicable	Si elles sont utilisées, ne pas utiliser la méthode 4	Si elles sont utilisées, ne pas utiliser la méthode 1
<b>Méthode 4</b> Insérer la variable du paramètre dans les identificateurs	Ne pas utiliser	Si elles sont utilisées, ne pas utiliser la méthode 3	Ne pas utiliser

## Spécifier ou sélectionner un fichier d'entrée non-Analytics pour un outil d'analyse

Le tableau ci-dessous résume les différentes méthodes permettant de spécifier ou de sélectionner un fichier d'entrée non-Analytics pour un outil d'analyse. La méthode que vous choisissez dépend en partie de l'application qui sera utilisée pour exécuter l'outil d'analyse.

Méthode	Détails	Robots	Client AX	Client Web d'AX	La fenêtre App Analyse
Balise PARAM avec type de « F »	<ul style="list-style-type: none"> <li>L'utilisateur du Client Web d'AX sélectionne le fichier d'entrée à l'aide d'un utilitaire de chargement de fichier</li> </ul> <p>Le nom du fichier est automatiquement spécifié comme valeur d'introduction analytique. Le fichier est chargé automatiquement dans le sous-répertoire <b>Fichiers associés</b> approprié dans AX Serveur.</p> <ul style="list-style-type: none"> <li>L'utilisateur de la fenêtre App Analyse sélectionne le fichier d'entrée à l'aide du navigateur de fichiers Windows</li> </ul> <p>Le chemin d'accès au fichier et le nom du fichier sont automatiquement spécifiés comme valeur d'introduction analytique</p> <p>Cette méthode est la meilleure option disponible, car elle associe flexibilité, facilité d'utilisation et précision.</p>			✓	✓
Balise PARAM avec type de « C »	<p>L'utilisateur spécifie manuellement un chemin de fichier d'entrée et un nom de fichier comme valeur d'entrée d'analyse.</p> <p>Cette méthode offre de la flexibilité, car le chemin et le nom du fichier ne sont pas spécifiés à l'avance. Elle est toutefois laborieuse et sujette à l'erreur, car elle demande à l'utilisateur d'entrer ces valeurs manuellement.</p>				✓
balise FILE (Pour plus d'informations, consultez la section "FILE" Page 893)	<ul style="list-style-type: none"> <li>Robots le fichier d'entrée doit être situé dans l'onglet <b>Entrée/Sortie</b> du robot</li> <li><b>AX Client, Client Web d'AX</b> : le fichier d'entrée doit se trouver dans le sous-répertoire <b>Fichiers associés</b> approprié dans AX Serveur</li> </ul>	✓	✓	✓	
Chemin d'accès au fichier	<p>Cette méthode évite d'utiliser la balise PARAM ; il s'agit toutefois de la moins flexible. Sur chaque ordinateur sur lequel l'outil d'analyse est exé-</p>				✓

Méthode	Détails	Robots	Client AX	Client Web d'AX	La fenêtre App Analyse
d'entrée et nom de fichier codés en dur dans l'outil d'analyse	cuté, l'utilisateur doit s'assurer que le fichier d'entrée a un chemin de fichier et un nom de fichier identiques à ceux spécifiés dans l'outil d'analyse.				

# PASSWORD

Crée un paramètre d'entrée de mot de passe pour un outil d'analyse. Le paramètre fournit le stockage crypté d'un mot de passe pour une utilisation ultérieure dans une commande ACLScript.

L'utilisateur est invité à spécifier la valeur de mot de passe requise lorsqu'il planifie ou lance un outil d'analyse pour qu'aucune intervention de l'utilisateur ne soit requise pendant l'exécution du script.

## Syntaxe

```
//PASSWORD index nom
<description>
```

## Paramètres

Nom	Description
<i>index</i>	Identificateur numérique associé au mot de passe. La valeur doit être comprise entre 1 et 10.
<i>nom</i>	Libellé correspondant à l'invite de mot de passe. <i>nom</i> est affiché dans l'application cliente lorsque l'application invite l'utilisateur à entrer le mot de passe.
<i>description</i> optionnel	Texte descriptif sur le mot de passe requis ou d'autres informations. La description peut être contenue sur plusieurs lignes, mais aucune ligne ne peut être sautée. La description doit être entrée sur la ligne située en dessous de la balise PASSWORD associée.

## Exemples

### Créer un paramètre d'entrée de mot de passe pour une requête Direct Link SAP

L'en-tête d'outil d'analyse indique un paramètre d'entrée de mot de passe invitant l'utilisateur à saisir un mot de passe SAP. Le mot de passe stocké est utilisé dans la commande RETRIEVE ultérieure dans le corps du script.

```
COMMENT
//ANALYTIC SAP Exemple de mot de passe
```



```
//PASSWORD 1 Mot de passe SAP :
//DATA RSADMIN
END
SET SAFETY OFF
RETRIEVE RSADMIN PASSWORD 1
OPEN RSADMIN
SET SAFETY ON
```

### Remarque

Le paramètre d'entrée de mot de passe et le paramètre du mot de passe dans la commande RETRIEVE sont liés par le même identificateur numérique :

```
//PASSWORD 1 Mot de passe SAP :
,
,
,
RETRIEVE RSADMIN PASSWORD 1
```

## Créer un paramètre d'entrée de mot de passe pour une exportation dans Résultats

L'en-tête d'outil d'analyse indique un paramètre d'entrée de mot de passe invitant l'utilisateur à saisir un mot de passe HighBond. Le mot de passe stocké est utilisé dans la commande EXPORT ultérieure dans le corps du script.

```
COMMENT
//ANALYTIC HighBond Exemple de mot de passe
//PASSWORD 3 HighBond Mot de passe :
END
SET SAFETY OFF
OPEN Exceptions_Ar
EXPORT FIELDS Aucune date d'échéance Réf Montant Type ACLGRC PASSWORD 3 TO
"10926@us"
SET SAFETY ON
```

## Remarques

### Stockage des mots de passe et cryptage

Les valeurs des mots de passe sont associées aux utilisateurs individuels et sont cryptées au repos. Les mots de passe restent sécurisés tout au long du traitement de l'outil d'analyse et sont cryptés dans les fichiers temporaires créés dans l'environnement de déploiement.

## Test dans Analytics

Si vous testez un outil d'analyse contenant une ou plusieurs balises PASSWORD dans Analytics, Analytics génère automatiquement une commande PASSWORD et vous invite à saisir le mot de passe approprié. Cette commande générée automatiquement vous permet d'éviter le travail d'insertion des commandes PASSWORD dans la partie script d'un outil d'analyse à des fins de test, puis de devoir le supprimer à nouveau avant de livrer l'outil d'analyse aux utilisateurs.

La commande PASSWORD générée automatiquement est enregistrée dans la trace, sans la valeur du mot de passe.

Les valeurs des mots de passe ne sont pas enregistrées lorsque vous exécutez un outil d'analyse dans Analytics. Vous devez donc spécifier le(s) mot(s) de passe à chaque fois que vous exécutez l'outil d'analyse, y compris lors de l'exécution ou de l'exécution pas à pas de l'outil d'analyse à partir de la position du curseur.

# DATA

Indique qu'une table de sortie Analytics par un outil d'analyse est copiée dans le sous-répertoire de données (un emplacement de stockage) dans l'environnement de déploiement

En général, vous stockez les tables Analytics pour qu'elles puissent être utilisées comme tables d'entrée pour des outils d'analyse ultérieurs.

## Remarque

L'édition professionnelle d'ACL Robotics ne comprend pas d'emplacement de stockage pour les tables Analytics. La balise //DATA est ignorée lors de l'exécution des outils d'analyse dans l'édition professionnelle.

Vous pouvez passer à l'Enterprise Edition si vous avez besoin de la capacité de stockage.

## Syntaxe

```
//DATA nom
```

## Paramètres

Nom	Description
<i>nom</i>	<p>Le nom de la table Analytics à stocker. La valeur <i>nom</i> ne peut contenir aucun espace.</p> <p><b>Remarque</b></p> <p>La valeur <i>nom</i> doit correspondre exactement au nom de la table de sortie Analytics dans le script d'outil d'analyse. Vous ne nommez pas une table avec un <i>nom</i>, vous faites correspondre un nom spécifié dans le script.</p> <p>Vous devez spécifier le nom de la table, et non le nom du fichier de données source.</p> <p>Correct :</p> <pre>//DATA Chèques_Manquants</pre> <p>Incorrect :</p> <pre>//DATA Chèques_Manquants.fil</pre> <p><b>Remarque</b></p> <p>Une table Analytics existante portant le même nom que la valeur spécifiée est écrasée.</p> <p><b>Caractères génériques</b></p>

Nom	Description
	<p>Vous pouvez utiliser des caractères génériques dans <i>nom</i> s'il est possible qu'une partie du nom de la table change. Par exemple, si le nom de la table est basé sur le mois (factures-jan, factures-fév, etc.), spécifier factures-* garantit que la table est copiée dans le sous-répertoire de données indépendamment du suffixe représentant le mois.</p> <p>Vous pouvez indiquer un seul caractère générique pour copier toutes les tables de sorties Analytics dans le script d'outil d'analyse vers le sous-répertoire de données :</p> <div data-bbox="505 480 1463 552" style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <pre>//DATA *</pre> </div> <p><b>Attention</b></p> <p>Soyez vigilant lorsque vous utilisez des caractères génériques. Vous pouvez écraser par inadvertance des tables de données existantes si le schéma de caractères génériques que vous spécifiez correspond à des tables non souhaitées.</p> <p>Il est recommandé de s'arranger pour que la valeur <i>nom</i> soit aussi précise que possible. N'utilisez les caractères génériques que là où ils sont requis.</p> <p><b>Chargements vers Robots</b></p> <p>Pour plus d'informations sur les chargements vers Robots, consultez la section "Chargements vers le module cloud Robots" Page 918.</p>

## Exemples

Copie d'une table Analytics vers l'emplacement de stockage

L'en-tête d'outil d'analyse suivant indique que la table Factures, qui est créée dans le script associé, est copiée dans l'emplacement de stockage :

```
COMMENT
//ANALYTIC Table d'importation
//DATA Factures
END
IMPORT DELIMITED TO Factures "Factures.fil" FROM "Factures.csv" 0 SEPARATOR ";"
QUALIFIER "" CONSECUTIVE STARTLINE 1 KEPTITLE ALLCHAR ALLFIELDS
```

## Remarques

### Stockage des tables de sortie

Les tables de sortie ne sont pas copiées automatiquement dans l'emplacement de stockage. Vous devez utiliser une balise `DATA` pour chaque table à stocker. Vous pouvez inclure plusieurs balises `DATA` dans

un en-tête d'outil d'analyse si besoin.

## Dans quel(s) cas utiliser la balise DATA ?

Vous devez utiliser la balise `DATA` et le stockage des tables Analytics dans deux situations :

- des tables de sortie sont utilisées comme entrée pour des scripts d'outils d'analyse qui suivent
- les utilisateurs peuvent sélectionner des tables ou des champs d'entrée lorsqu'ils planifient un outil d'analyse ou l'exécutent de manière ponctuelle

### Remarque

Si tout un processus d'analyse de données est effectué à l'aide d'un seul script d'outil d'analyse, l'utilisation de la balise `DATA` n'est pas obligatoire.

La balise `DATA` n'est pas censée être utilisée pour spécifier des tables de résultats. Utilisez plutôt la balise `RESULT`. Pour plus d'informations, consultez la section "RESULT" Page 919.

## Des tables de sortie sont utilisées comme entrée pour des scripts d'outils d'analyse qui suivent.

Selon l'environnement de déploiement et la structure des scripts associés, il se peut que vous deviez utiliser la balise `DATA` pour stocker une table de sortie Analytics à utiliser dans un outil d'analyse qui suit.

Pendant le traitement de l'outil d'analyse, Robots et AX Serveur utilisent un répertoire temporaire pour stocker les tables de sortie Analytics et y accéder, il n'est donc peut-être pas utile d'utiliser la balise `DATA`.

Le tableau ci-après donne quelques indications.

Environnement de déploiement	Utilisez la balise <code>DATA</code> si...	La balise <code>DATA</code> n'est pas nécessaire si...
Robots (Enterprise Edition uniquement)	<ul style="list-style-type: none"> <li>○ une sortie de table Analytics dans une tâche du robot est requise comme entrée dans une autre tâche du robot</li> </ul>	<ul style="list-style-type: none"> <li>○ une table Analytics est sortie et, par conséquent, est entrée pendant l'exécution d'une séquence de scripts d'outils d'analyse dans une seule tâche du robot</li> <li>○ tout un processus d'analyse de données est effectué à l'aide d'un seul script d'outil d'analyse</li> </ul>
Serveur AX	<ul style="list-style-type: none"> <li>○ une sortie de table Analytics par un seul script d'outil d'analyse est requise comme entrée pour un autre script d'outil d'analyse</li> </ul>	<ul style="list-style-type: none"> <li>○ tout un processus d'analyse de données est effectué à l'aide d'un seul script d'outil d'analyse</li> </ul>
Fenêtre App Analyse	<ul style="list-style-type: none"> <li>○ une sortie de table Analytics par un seul script d'outil d'analyse est requise comme entrée pour un autre script d'outil d'analyse</li> </ul>	<ul style="list-style-type: none"> <li>○ tout un processus d'analyse de données est effectué à l'aide d'un seul script d'outil d'analyse</li> </ul>

## Les utilisateurs peuvent sélectionner des tables ou des champs d'entrée

Les balises d'outils d'analyse TABLE et FIELD créent des paramètres d'entrée permettant à un utilisateur de sélectionner une table Analytics et de sélectionner les champs de la table à utiliser comme entrée vers un script d'outil d'analyse. Toutefois, une table doit préexister dans l'emplacement de stockage afin d'être disponible pour être sélectionnée.

Si vous développez un outil d'analyse permettant à un utilisateur de choisir un(e) ou plusieurs tables et champs d'entrée, un outil d'analyse préalable contenant la balise DATA doit exécuter et enregistrer la ou les tables qui conviennent à l'emplacement de stockage.

## Localisez les tables de sortie dans la section Tables source dans Robots

Vous pouvez éventuellement ajouter le préfixe src\_ à un nom de table de sortie pour localiser la table de sortie dans la section **Tables source** de l'onglet **Entrée/Sortie** dans Robots.

```
//DATA src_Factures
```

Vous devez ajouter le préfixe au nom de la table dans la balise //DATA et dans le script qui l'accompagne.

La section **Tables source** vous permet de séparer visuellement les tables fournissant une entrée pour des scripts ultérieurs. Si aucun nom de table de sortie ne possède le préfixe src\_, la section **Tables source** n'apparaît pas dans l'onglet **Entrée/Sortie** et toutes les tables se trouvent par défaut dans la section **Autres tables**.

## Chargements vers le module cloud Robots

Avec les scripts d'outils d'analyse exécutés dans les installations de Robots, spécifier DATA charge uniquement le format de table de la table (nom des champs, type de données, longueur des champs) de l'Agent Robots local vers le module cloud Robots dans HighBond. Les données de la table restent sur le réseau de votre organisation, dans le répertoire Agent Robots.

Toutes les informations sont chiffrées en transit.

## Écrasement de tables liées ou partagées dans AX Serveur

Si une table de sortie écrase une table liée ou partagée dans AX Serveur, la table est remplacée par une table autonome.

# RESULT

Indique les résultats de sortie des outils d'analyse que vous souhaitez mettre à disposition des utilisateurs finaux dans les applications clientes.

Les résultats de sortie, s'ils existent, ne sont pas automatiquement mis à disposition. Vous devez utiliser une balise RESULT distincte pour chaque élément de résultat que vous souhaitez mettre à disposition.

## Syntaxe

```
//RESULT type nom
<description>
```

## Paramètres

Nom	Description
<i>type</i>	<p>Type de l'élément de résultat :</p> <ul style="list-style-type: none"> <li>TABLE : une table Analytics et un fichier de données associé (.fil)</li> <li>LOG : un fichier trace de l'outil d'analyse</li> <li>FILE : un fichier non-Analytics</li> </ul> <p>Pour plus d'informations sur les chargements vers Robots, consultez la section "Chargements vers le module cloud Robots" Page 921.</p>
<i>nom</i>	<p>Nom de l'élément de résultat. La valeur <i>nom</i> ne peut contenir aucun espace.</p> <p><b>Remarque</b></p> <p>La valeur <i>nom</i> doit correspondre exactement au nom de l'élément de résultat dans le script d'outil d'analyse. Vous ne donnez pas de <i>nom</i> à un élément, vous faites correspondre un nom spécifié dans le script.</p> <h3>Nom de table</h3> <p>La valeur <i>nom</i> indique un nom de table Analytics. Vous devez spécifier le nom de la table, et non le nom du fichier de données source.</p> <p>Correct :</p> <pre>//RESULT TABLE Chèques_Manquants</pre> <p>Incorrect :</p> <pre>//RESULT TABLE Chèques_Manquants.fil</pre>

Nom	Description
	<p><b>Caractères génériques</b></p> <p>Vous pouvez utiliser des caractères génériques dans <i>nom</i> s'il est possible qu'une partie du nom de la table change. Par exemple, si le nom de la table est basé sur le mois (factures-jan, factures-fév, etc.), spécifier <code>factures-*</code> garantit que la table est rendue disponible dans les résultats indépendamment du suffixe représentant le mois.</p> <p><b>Nom de la trace</b></p> <p>Facultatif. La valeur <i>nom</i> spécifie un nom de fichier trace d'outil d'analyse. Si vous ne spécifiez pas de <i>nom</i>, c'est le nom de trace par défaut qui est utilisé : <code>nom_outil_analyse.log</code>.</p> <p><b>Remarque</b></p> <p>Si vous spécifiez un nom de trace, <code>SET LOG TO nom_trace</code> doit apparaître dans le script.</p> <p><b>Nom de fichier</b></p> <p>La valeur <i>nom</i> spécifie un nom de fichier non-Analytics.</p> <p>Vous devez spécifier l'extension de fichier appropriée pour le type de fichier non-Analytics en sortie.</p> <p>Correct :</p> <pre data-bbox="505 984 1463 1052">//RESULT FILE Chèques_Manquants.xlsx</pre> <p>Incorrect :</p> <pre data-bbox="505 1123 1463 1190">//RESULT FILE Chèques_Manquants</pre> <p><b>Caractères génériques</b></p> <p>Vous pouvez utiliser des caractères génériques dans l'intégralité ou dans une partie de la valeur <i>nom</i> pour spécifier l'ensemble des fichiers portant une extension spécifique ( <code>*.xlsx</code> ) ou pour spécifier des fichiers pour lesquels une partie du nom peut changer.</p> <p>Par exemple, si le nom du fichier est basé sur le mois (factures-jan.xlsx, factures-fév.xlsx, etc.), spécifier <code>factures-*.xlsx</code> garantit que le fichier est disponible dans les résultats indépendamment du suffixe représentant le mois.</p>
<i>description</i> optionnel	Texte descriptif sur le résultat ou d'autres informations. La description peut être contenue sur plusieurs lignes, mais aucune ligne ne peut être sautée.

## Exemples

Balise RESULT pour une table Analytics :



```
//RESULT TABLE Chèques_Manquants
```

Balise RESULT pour la trace des outils d'analyse avec le nom par défaut :

```
//RESULT LOG
```

Balise RESULT pour la trace des outils d'analyse avec un nom spécifié :

```
//RESULT LOG Mon_nom_de_trace
,
,
,
SET LOG TO Mon_nom_de_trace
```

Balise RESULT pour un fichier Excel spécifique :

```
//RESULT FILE Chèques_Manquants.xlsx
```

Balise RESULT pour tous les fichiers Excel :

```
//RESULT FILE *.xlsx
```

## Remarques

### Chargements vers le module cloud Robots

Lorsque des scripts d'outils d'analyse sont exécutés dans des installations Robots, indiquer RESULT LOG ou RESULT FILE permet de charger des fichiers traces d'outil d'analyse des fichiers non-Analytics depuis l'Agent Robots local vers le module cloud Robots dans HighBond.

Pour plus d'informations sur les traces, consultez la section "Sortie des fichiers traces" bas.

En indiquant RESULT TABLE, seul le format de table est chargé (nom des champs, type de données, longueur des champs). Les données de la table de résultats restent sur le réseau de votre organisation, dans le répertoire Agent Robots.

Toutes les informations sont chiffrées en transit.

### Sortie des fichiers traces

La façon dont les fichiers traces des outils d'analyse sortent dépend de la réussite ou de l'échec d'un script et de l'environnement sur lequel le script est exécuté.

Script d'outil d'analyse	Agent Robots	Serveur AX	Fenêtre App Analyse
Réussite	<ul style="list-style-type: none"> <li>◦ <b>RESULT LOG spécifié</b> fichier trace chargé dans le module cloud Robots</li> <li>◦ <b>RESULT LOG non spécifié</b> aucun fichier trace</li> </ul>	<ul style="list-style-type: none"> <li>◦ <b>RESULT LOG spécifié</b> sortie du fichier trace dans AX Serveur (disponible dans les applications clientes)</li> <li>◦ <b>RESULT LOG non spécifié</b> aucun fichier trace</li> </ul>	<ul style="list-style-type: none"> <li>◦ <b>RESULT LOG spécifié</b> sortie du fichier trace dans l'onglet Résultats</li> <li>◦ <b>RESULT LOG non spécifié</b> aucun fichier trace</li> </ul>
Échec	<ul style="list-style-type: none"> <li>◦ <b>Balise RESULT LOG non prise en compte</b> <ul style="list-style-type: none"> <li>• sortie automatique du fichier trace dans le dossier de données de base de l'Agent Robots (paramètre par défaut = "false")</li> <li>• Chargement automatique du fichier trace dans le module cloud Robots (paramètre de configuration = "true" (par défaut))</li> </ul> </li> </ul> <p>Consultez le paramètre de configuration <b>UploadLog-sWhenFailed</b> dans <a href="#">Configuration d'un agent Robots</a>.</p>	<ul style="list-style-type: none"> <li>◦ <b>Balise RESULT LOG non prise en compte</b> sortie automatique du fichier trace dans AX Serveur (disponible dans les applications clientes)</li> </ul>	<ul style="list-style-type: none"> <li>◦ <b>Balise RESULT LOG non prise en compte</b> sortie automatique du fichier trace dans l'onglet Résultats</li> </ul>

## Limitation de la taille des fichiers de résultats sur AX Serveur

Les fichiers de résultats sont limités à un maximum de 2 Go pour les scripts d'outils d'analyse s'exécutant sur AX Serveur. Si le fichier dépasse cette taille, les résultats ne sont pas enregistrés.

## Stockage et disponibilité du fichier Résultat pendant l'exécution du script sur AX Serveur

Lorsque vous utilisez la balise //RESULT FILE, le fichier créé est disponible au téléchargement à partir du Client Web d'AX et d'AX Client à la fin du script. Ce fichier est stocké dans la base de données AX et n'est pas disponible sur le système de fichiers d'AX Serveur lorsque le script ne s'exécute pas.

Pendant l'exécution de votre script, le fichier est disponible temporairement sur le système de fichiers d'AX Serveur et vous pouvez l'utiliser avec des processus externes, comme ceux que vous appelez à

l'aide de la commande EXECUTE. Pendant l'exécution du script, des processus externes peuvent accéder au fichier à partir du sous-répertoire de la tâche d'analyse.

#### Remarque

Par défaut, les sous-répertoires des tâches d'analyse se trouvent dans `ACL\Data\jobs`. Une fois le script terminé, le sous-répertoire de la tâche d'analyse est supprimé et le fichier est stocké dans la base de données.

# PUBLISH

Indique un fichier contenant des métadonnées définissant les tables Analytics à publier dans AX Exception à la fin du traitement d'un outil d'analyse.

## Syntaxe

```
//PUBLISH nom_fichier
```

## Paramètres

Nom	Description
<i>nom_fichier</i>	Le nom du fichier contenant les métadonnées de publication d'AX Exception.

## Exemples

### La définition de l'outil d'analyse et le fichier texte qui spécifient les détails de publications de l'outil d'analyse

La balise FILE est nécessaire si le fichier à publier est stocké dans le dossier AX afin qu'il puisse être récupéré au moment du traitement de l'outil d'analyse.

```
COMMENT
//ANALYTIC Publier les résultats
//RESULT TABLE Résultats
//FILE ex_publication.txt
//PUBLISH ex_publication.txt
END
EXTRACT RECORD TO Résultats
```

Le fichier `ex_publication.txt` envoyé dans le sous-répertoire **Fichiers associés** de la collection contient la ligne de texte suivante. Les valeurs doivent être insérées entre guillemets et elles doivent présenter la syntaxe suivante : "nom de la table"; "nom de l'entité"; "nom de l'outil d'analyse".

```
"Résultats";"MonEntité";"MonOutildanalyse"
```

# Développement d'outils d'analyse

Pour faciliter le débogage et isoler les erreurs, développez le corps du script avant d'ajouter l'en-tête d'outil d'analyse. Une fois que vous ajoutez l'en-tête d'outil d'analyse, utilisez le fichier trace et des valeurs de test temporaires pour tester l'exécution de l'outil d'analyse. Enfin, déployez l'outil d'analyse dans l'environnement cible.

## Remarque

Le flux de travail suivant n'est qu'une méthode suggérée pour le développement d'outils d'analyse ; cependant, vous êtes libre de développer des outils d'analyse de la façon qui vous convient le mieux.

## Flux d'activités pour la création et le test du script d'un outil d'analyse

### Créer le script Analytics

Créez un script dans Analytics sans utiliser de boîtes de dialogue personnalisées pour les entrées de l'utilisateur ni toute autre fonctionnalité demandant des interactions avec l'utilisateur pendant l'exécution du script. Les outils d'analyse autorisent l'intervention des utilisateurs avant leur exécution, mais à la différence des scripts, ils ne prennent pas en charge l'interaction des utilisateurs pendant leur exécution.

Pour stocker des valeurs d'introduction dans le script Analytics, créez temporairement des variables dans la partie supérieure du script :

```
ASSIGN v_TableAnalyse = "Trans_Mai"
```

Testez et déboguez le script jusqu'à ce qu'il s'exécute sans erreur.

### Ajouter l'en-tête de l'outil d'analyse et les balises

Ajoutez un en-tête d'outil d'analyse au script. Copiez les noms des variables dans la partie supérieure du script dans les balises correspondantes de l'en-tête de l'outil d'analyse :

```
//TABLE v_TableAnalyse "Table à classer"
```

Pour plus d'informations, consultez la section "Ajouter des en-têtes d'outils d'analyse" Page 929.

## Inclure la trace dans les résultats des outils d'analyse

La trace est un outil essentiel permettant de diagnostiquer l'origine des échecs des outils d'analyse, mais elle peut aussi avoir de l'importance lorsque des outils d'analyse s'exécutent correctement, mais qu'ils donnent des résultats inattendus. La trace est sortie automatiquement lorsqu'un outil d'analyse échoue, mais si vous spécifiez la balise d'analyse `RESULT` suivante, la trace sort uniquement en cas de bonne exécution.

Incluez la ligne suivante dans l'en-tête de l'outil d'analyse pour vous assurer qu'une trace sera disponible à chaque exécution de l'outil d'analyse :

```
//RESULT LOG
```

## Valider l'en-tête d'un outil d'analyse

Validez l'en-tête de l'outil d'analyse. Vous pouvez valider l'en-tête de l'outil d'analyse aussi souvent que vous le souhaitez.

Pour plus d'informations, consultez la section "Valider les en-têtes des outils d'analyse" Page 930.

## Affecter des valeurs de test temporaires aux balises d'outils d'analyse

À l'aide de l'opérateur d'affectation spécifique ( `:=` ), affectez des valeurs de tests temporaires à toutes les balises d'analyse nécessitant une intervention de l'utilisateur. Vous pouvez copier les valeurs de test à partir des affectations de variables temporaires dans la partie supérieure du script :

```
//TABLE v_TableAnalyse "Table à classer" := "Trans_Mai"
```

Pour plus d'informations sur l'affectation de valeurs de test temporaires, consultez la rubrique "Spécifier les valeurs d'introduction de test dans Analytics" Page 888.

## Supprimer les variables temporaires

Supprimez les variables temporaires dans la partie supérieure du script ou mettez-les en commentaire si vous pensez encore vouloir les utiliser.

## Suivre l'outil d'analyse par étape

Suivez l'outil d'analyse par étape en cliquant sur **Étape**  ou en appuyant sur **F10** de façon répétitive. Examinez le contenu de l'onglet **Variables** dans le **Navigateur** pour vous assurer que toutes les variables contenues dans l'en-tête de l'outil d'analyse sont créées correctement, avec la bonne attribution des valeurs de test.

Testez et déboguez l'outil d'analyse jusqu'à ce qu'il s'exécute sans erreur.

Une fois les tests terminés, supprimez les valeurs de test temporaires et l'opérateur d'affectation spécifique dans toutes les balises d'analyse.

#### Remarque

Pour quitter l'outil d'analyse avant la fin, appuyez sur **Esc**, puis cliquez sur **Oui** dans l'invite de confirmation.

#### Astuce

Il est possible de supprimer toutes les variables stockées et toutes les attributions de variables du projet Analytics en saisissant DELETE ALL OK dans la ligne de commande. Effacez le contenu de l'onglet **Variables** avant de suivre un outil d'analyse par étape afin de commencer dans de bonnes conditions.

## Flux d'activités pour le test d'une App Analyse

Pour les outils d'analyse qui seront exécutés dans le Client Web d'AX ou dans la fenêtre App Analyse, vous devez également tester l'App Analyse.

### Supprimer les formats de table redondants

Une fois que tous les outils d'analyse et scripts secondaires de l'App Analyse sont testés et débogués et qu'ils s'exécutent correctement, supprimez les formats de table du projet Analytics que vous n'allez pas inclure dans l'App Analyse.

Les formats de table redondants vont encombrer l'App Analyse dans AX Client, dans le Client Web d'AX et dans la fenêtre App Analyse, et pourraient induire en erreur les utilisateurs finaux.

### Ouvrir l'App Analyse dans la fenêtre App Analyse

Ouvrez l'App Analyse terminée dans la fenêtre App Analyse en cliquant avec le bouton droit de la souris sur l'entrée du projet Analytics dans l'onglet **Vue globale** et en sélectionnant **Ouvrir comme App Analyse**.

#### Remarque

Si l'App Analyse ne parvient pas à s'ouvrir et que vous obtenez un message d'erreur stipulant que les outils d'analyse portent des noms identiques, vérifiez la valeur *nom* dans la balise ANALYTIC pour chaque outil d'analyse spécifié dans le message d'erreur. Les valeurs *nom* des outils d'analyse doivent être uniques dans un projet Analytics.

### Exécuter les outils d'analyse

Exécutez tous les outils d'analyse dans l'App Analyse pour confirmer qu'ils fonctionnent correctement.

Respectez le bon ordre pour l'exécution des outils d'analyse si vous utilisez l'option TYPE avec la balise ANALYTIC et que vous créez des outils d'analyse d'importation, de préparation et d'analyse.

## Vérifier la trace

Si l'outil d'analyse échoue, ouvrez et examinez le fichier trace (*Nom\_Outil\_Analyse.log*). La trace doit inclure une entrée marquée d'une croix rouge signalant la raison de l'échec de l'outil d'analyse :

- si cet échec est dû à des valeurs d'introduction mal saisies, réexécutez l'outil d'analyse immédiatement avec une valeur d'introduction saisie correctement.
- si cet échec est dû à des erreurs de syntaxe ou de logique dans le corps du script, corrigez l'erreur dans Analytics, puis rouvrez l'App Analyse dans la fenêtre App Analyse

Il se peut qu'un outil d'analyse s'exécute correctement, mais que la table de résultats ne contienne pas les résultats attendus. Dans cette situation, examinez les entrées de la trace dans la séquence, puis vérifiez les valeurs d'introduction transmises à l'outil d'analyse pour vous assurer que l'outil d'analyse fonctionne de la manière prévue.

## Compression et validation d'une App Analyse

### Compresser ou importer l'App Analyse

Une fois que vous êtes satisfait du fonctionnement de l'App Analyse, compressez-la dans un package pour qu'elle soit diffusée et utilisée dans la fenêtre App Analyse ou importez-la dans AX Serveur pour une utilisation dans AX Client ou dans le Client Web d'AX. Pour plus d'informations, consultez la section "Compresser des Apps Analyse" Page 939.

### Exécuter des Apps Analyse AX Serveur

Si vous développez des outils d'analyse à utiliser dans AX Serveur, exécutez tous les outils d'analyse à l'aide d'AX Client et du Client Web d'AX pour vous assurer de leur bonne exécution.



# Ajouter des en-têtes d'outils d'analyse

Un **en-tête d'outil d'analyse** est une série de balises déclaratives incluses dans un commentaire au début d'un script. L'en-tête comprend des paramètres d'entrée que l'utilisateur renseigne au préalable permettant ainsi d'exécuter l'outil d'analyse de manière autonome, immédiatement ou à une heure planifiée.

Après avoir développé des scripts dans un projet Analytics, vous devez ajouter un en-tête d'outil d'analyse avant de pouvoir transmettre les scripts à Robots ou utiliser le projet en tant qu'App Analyse dans AX Serveur ou dans la fenêtre App Analyse.

## Conditions à respecter par un en-tête d'outil d'analyse

Un en-tête d'outil d'analyse doit respecter certaines conditions. S'il n'est pas conforme à ces exigences, le script d'outil d'analyse échoue lors de son exécution.

Pour obtenir des informations détaillées sur la syntaxe des en-têtes d'outils d'analyse et pour obtenir la liste complète des balises d'outils d'analyse, consultez la section "En-têtes des outils d'analyse et balises" Page 887.

## Incluez les en-têtes d'outil d'analyse dans un bloc de commentaires

Les en-têtes d'outils d'analyse doivent être entièrement définis dans un bloc de commentaire qui commence à la première ligne du script.

```
COMMENT
Les balises des outils d'analyse se placent ici.
END
```

## Déclaration des entrées et sorties

Dans l'en-tête d'outil d'analyse, vous devez déclarer les entrées requises par l'outil d'analyse et les sorties copiées dans Robots ou dans AX Serveur ou écrites sous forme de résultats dans la fenêtre App Analyse :

Entrées	Sorties
<ul style="list-style-type: none"> <li>○ des tables</li> <li>○ champs</li> <li>○ paramètres</li> <li>○ fichiers d'entrée</li> <li>○ invites de mot de passe</li> </ul>	<ul style="list-style-type: none"> <li>○ fichiers de données</li> <li>○ tables de résultats</li> <li>○ fichiers traces</li> </ul>

## Déclaration des valeurs d'entrée

Les en-têtes d'outils d'analyse doivent aussi déclarer les valeurs d'entrée que les utilisateurs spécifieront lorsqu'ils exécuteront ou planifieront l'outil d'analyse.

Utilisez la balise `PARAM` pour ajouter des paramètres d'entrée qui acceptent ces valeurs d'introduction spécifiées par l'utilisateur et les stockent dans des variables. Par exemple, si vous souhaitez que l'outil d'analyse sélectionne des données basées sur une période donnée, vous devez ajouter les paramètres d'entrée `Date de début` et `Date de fin` permettant aux utilisateurs de spécifier ces dates.

## Valider les en-têtes des outils d'analyse

Après avoir ajouté un en-tête d'outil d'analyse à un ou plusieurs scripts, utilisez des outils dans Analytics pour valider sa syntaxe afin de vous assurer qu'elle est correcte. Effectuez la validation avant de transmettre des scripts à Robots ou de compresser les Apps Analyse pour éviter tout échec des outils d'analyse lors de leur exécution.

Un outil valide les en-têtes d'outil d'analyse individuels au niveau du script. L'autre outil valide tous les en-têtes d'outils d'analyse d'un projet à la fois. Les deux types de validation se concentrent sur différentes choses.

### Valider un en-tête d'outil d'analyse individuel

La validation au niveau du script d'un en-tête d'outil d'analyse se concentre sur la syntaxe des balises d'outils d'analyse individuelles et signale les erreurs avec les numéros de lignes correspondants.

1. Ouvrez le script contenant l'en-tête de l'outil d'analyse.
2. Dans la barre d'outils Éditeur de script, cliquez sur **Valider un en-tête d'outil d'analyse**  .  
Un message s'affiche pour vous indiquer si l'en-tête de l'outil d'analyse est valide ou bien pour signaler une erreur et le numéro de la ligne où elle se trouve.
3. Si l'en-tête de l'outil d'analyse contient une erreur, corrigez-la, puis cliquez à nouveau sur **Valider l'en-tête de l'outil d'analyse**  pour vous assurer qu'il n'y a pas d'autre erreur.

#### Astuce

Si la nature de l'erreur n'est pas apparente, en fonction du message d'erreur, examinez la rubrique d'aide relative à la balise d'analyse associée. Comparez attentivement la syntaxe indiquée dans la rubrique à la syntaxe se trouvant à la ligne de l'en-tête de l'outil d'analyse. Des erreurs peuvent provenir de différences secondaires dans la syntaxe de l'en-tête de l'outil d'analyse.

### Valider tous les en-têtes d'outils d'analyse dans un projet

La validation au niveau d'un projet des en-têtes d'outils d'analyse contrôle les deux éléments suivants :

- l'existence d'au moins un en-tête d'outil d'analyse dans le projet
- l'unicité des noms de plusieurs outils d'analyse

### Remarque


Renvoie au nom indiqué dans la balise ANALYTIC, et non au nom du script dans l'onglet **Aperçu** du **Navigateur**.


La validation au niveau du projet s'effectue automatiquement lorsque vous transmettez des scripts à Robots. Vous pouvez aussi effectuer la validation manuellement si vous ajoutez le bouton **Vérifier les**

**scripts**  à la barre d'outils Analytics.

1. Si besoin, ajoutez le bouton **Vérifier les scripts** à la barre d'outils Analytics :
  - a. Dans la barre d'outils, double-cliquez sur un endroit vide pour ouvrir la boîte de dialogue **Personnaliser la barre d'outils**.
  - b. Dans la liste **Boutons de barre d'outils disponibles**, sélectionnez le bouton **Vérifier les scripts**, puis cliquez sur **Ajouter**.
  - c. Dans la liste **Boutons actuels de la barre d'outils**, sélectionnez le bouton **Vérifier les scripts** et cliquez sur **Haut** ou **Bas** pour modifier son emplacement.

L'ordre des boutons de haut en bas correspond à leurs positions de gauche à droite, dans la barre d'outils.

- d. Cliquez sur **Fermer** afin d'enregistrer vos modifications.
2. Dans la barre d'outils, cliquez sur **Vérifier les scripts** .
 

Un message s'affiche pour vous indiquer si l'en-tête de l'outil d'analyse est valide ou bien pour signaler une ou plusieurs erreurs.
  3. Si les en-têtes des outils d'analyse contiennent une erreur, corrigez-la, puis cliquez à nouveau sur **Vérifier les scripts**  pour vous assurer qu'il n'y a pas d'autre erreur.

## Exemple d'en-tête d'outil d'analyse

```
COMMENT
//ANALYTIC Identifier les chèques manquants
  Cet outil d'analyse identifie les numéros de chèque manquants
//TABLE v_table_paiements Table paiements
  Sélectionner une table qui répertorie les paiements et qui inclut une colonne Numéro de chèque
//FIELD v_chèque_num CN Numéro de chèque
  Sélectionnez le champ qui contient les numéros de chèque
//PARAM v_date_début D OPTIONAL Date de début (facultatif)
  Entrez la date de début de l'analyse
//PARAM v_date_fin D OPTIONAL Date de fin (facultatif)
  Entrez la date de fin de l'analyse
//PARAM v_région C MULTI SEPARATOR , QUALIFIER ' Région(s)
```

```
Entrez une ou plusieurs régions à inclure dans l'analyse
//RESULT LOG
//RESULT TABLE ChèquesManquants
//RESULT FILE DetailsChèquesManquants.xls
END
```

# Bonnes pratiques en termes de développement d'outils d'analyse

Les outils d'analyse prennent en charge la plupart des commandes que vous pouvez utiliser dans un script Analytics ordinaire. Toutefois, vous devez vous assurer que les outils d'analyse s'exécutent sans interaction avec l'utilisateur et qu'ils n'incluent pas de commandes non prises en charge par le moteur traitant les outils d'analyse dans l'environnement de déploiement.

Les outils d'analyse prennent en charge toutes les fonctions Analytics.

## Bonnes pratiques générales

### Utiliser un projet Analytics par robot ou par App Analyse

Créez un nouveau projet Analytics dans Analytics pour chaque robot ou App Analyse. Le projet doit contenir tous les outils d'analyse constituant le robot ou l'App Analyse et tous les sous-scripts requis. Dans le cas d'une App Analyse, le projet doit aussi contenir les fichiers de données requis par les outils d'analyse.

### Test en local

Testez tous les outils d'analyse en local avant de les déployer dans l'environnement cible. Assurez-vous que les outils d'analyse s'exécutent comme prévu et qu'ils ne nécessitent pas d'interaction de l'utilisateur.

Pour plus d'informations, consultez la section "Développement d'outils d'analyse" Page 925.

### Utiliser des connexions de données cohérentes à des fins de test

Pour tester en local si un outil d'analyse utilise une source de données ODBC, vous devez configurer une connexion ODBC à l'identique sur votre ordinateur local identique à la connexion dans l'environnement où sera exécuté l'outil d'analyse.

Pour les outils d'analyse diffusés afin d'être utilisés dans la fenêtre App Analyse, les utilisateurs finaux doivent configurer une connexion ODBC identique sur leurs ordinateurs.

### Éviter les chemins d'accès absolu à un fichier

Évitez d'utiliser des chemins d'accès absolu à un fichier dans des outils d'analyse (par exemple, `C:\résultats`) sauf si vous êtes sûr que des chemins de fichiers identiques existent dans l'environnement d'exécution de l'outil d'analyse.

L'utilisation de chemins d'accès relatifs au fichier comme `\résultats` vous permet de développer et de tester des outils d'analyse localement, puis de les déployer dans un autre environnement sans exiger que l'autre environnement présente une structure de répertoire identique.

## Utiliser SET pour les paramètres de préférence

Utilisez la commande SET pour spécifier les paramètres de préférence requis par l'outil d'analyse. Si vous ne spécifiez pas de préférences dans l'outil d'analyse, ce seront les préférences Analytics par défaut qui seront utilisées. Placez la commande SET après l'en-tête de l'outil d'analyse, mais avant toute logique analytique.

## Ne pas utiliser de champs calculés dans les tables de données ou de résultats

N'utilisez pas de champs calculés dans les tables que vous prévoyez de conserver après la session d'exécution du script d'outil d'analyse.

Les tables de résultats et les tables de données conservées pour être utilisées dans des interprétations ou comme valeurs d'introduction pour des scripts ultérieurs peuvent afficher des valeurs inattendues si elles contiennent des champs calculés. Les valeurs calculées dépendent des paramètres définis dans le fichier de préférences (`.prf`) ou à l'aide de la commande SET et, par conséquent, des environnements différents peuvent produire des valeurs différentes.

Si vous devez conserver ces valeurs dans un champ calculé, utilisez la commande EXTRACT avec l'option FIELDS ou ALL pour convertir ce champ en champ physique dans une table de résultats ou de données. Pour plus d'informations, consultez la section "Commande EXTRACT" Page 206.

## Chiffrer les mots de passe de connexion de données

Pour éviter d'avoir un mot de passe de source de données en texte brut dans un outil d'analyse, utilisez la balise d'outil d'analyse PASSWORD. Cette balise invite l'utilisateur à saisir un mot de passe avant l'exécution de l'outil d'analyse, puis crypte la valeur saisie.

## Utiliser un mot de passe lors de l'importation à partir d'HighBond ou lors de l'exportation vers HighBond

Le paramètre PASSWORD est requis dans les commandes qui effectuent des importations à partir d'ACL GRC ou des exportations vers HighBond :

- IMPORT GRCRESULTS
- IMPORT GRCPROJECT
- EXPORT... ACLGRC

Sans le paramètre PASSWORD, les commandes échouent dans Robots, Analytics Exchange ou la fenêtre App Analyse.

Lorsque vous utilisez le paramètre `PASSWORD` dans un script d'outil d'analyse, vous devez également préciser un paramètre d'entrée de mot de passe associé dans l'en-tête d'outil d'analyse. Pour plus d'informations, consultez la section "PASSWORD" Page 912.

### Remarque

Le paramètre `PASSWORD` n'est pas requis lors de l'exécution des commandes d'importation et d'exportation dans Analytics car le jeton d'accès à HighBond de l'utilisateur actif est utilisé automatiquement.

## Éviter les interactions avec les utilisateurs

Les outils d'analyse doivent pouvoir s'exécuter sans interaction avec l'utilisateur. Si une commande dans un outil d'analyse tente de créer une boîte de dialogue, le moteur de l'environnement de déploiement interrompt le traitement de l'outil d'analyse et un message d'erreur est entré dans le fichier journal.

## Remplacer les commandes d'interaction avec les utilisateurs par des balises d'outils d'analyse

N'utilisez pas de commandes Analytics nécessitant d'interaction avec l'utilisateur. Remplacez-les par des balises d'outils d'analyse équivalentes dans l'en-tête d'outil d'analyse. Les balises des outils d'analyse permettent aux utilisateurs d'indiquer des valeurs d'introduction avant l'exécution de l'outil d'analyse.

Ne pas utiliser	Remplacer par
DIALOG	//TABLE, //FIELD, //PARAM
ACCEPT	//TABLE, //FIELD, //PARAM
PASSWORD	//PASSWORD
PAUSE	aucun équivalent

## Instructions

- pour éviter les échecs de traitement des outils d'analyse, supprimez toutes les commandes interactives
- pour vous assurer que les fichiers peuvent être, au besoin, écrasés sans afficher de boîte de dialogue de confirmation, ajoutez la commande `SET SAFETY OFF` au début d'un outil d'analyse, puis ajoutez la commande `SET SAFETY ON` à la fin de l'outil d'analyse pour restaurer le comportement par défaut
- pour empêcher les boîtes de dialogue de confirmation de faire échouer l'outil d'analyse, ajoutez le paramètre `OK` après les commandes qui affichent normalement une boîte de dialogue de confirmation :
  - `RENAME`
  - `DELETE`

# Vérification de la syntaxe du script

Analytics fournit un outil permettant de détecter une syntaxe de script entraînant l'échec d'un outil d'analyse ou nécessitant une adaptation entre votre environnement local et l'environnement de déploiement des outils d'analyse. L'outil fournit un avertissement uniquement. Vous êtes toujours libre de transmettre ou d'importer des scripts d'outils d'analyse comportant des avertissements.

## Ce que l'outil vérifie

L'outil vérifie tous les scripts d'un projet et détecte trois choses :

- une commande nécessitant une interaction avec l'utilisateur
- un chemin de fichier absolu
- l'appel d'un script externe

L'outil vérifie les éléments suivants dans tous les scripts d'un projet :

- une commande nécessitant une interaction avec l'utilisateur
- un chemin de fichier absolu
- l'appel d'un script externe
- les commandes n'ayant pas de paramètre PASSWORD qui effectuent des importations à partir d'HighBond ou des exportations vers HighBond

## Dans quels cas la vérification est effectuée

La syntaxe du script est vérifiée automatiquement lorsque vous transmettez des scripts à Robots.

La vérification automatique de la syntaxe est activée par défaut. Si vous voulez la désactiver, sélectionnez **Désactiver la vérification de la syntaxe des scripts avant de transmettre des scripts** dans la boîte de dialogue **Options (Outils > Options > Interface)**.

## Effectuer manuellement la vérification

Vous pouvez effectuer manuellement la vérification de la syntaxe des scripts. Vous devrez peut-être


d'abord ajouter le bouton **Vérifier les scripts**  dans la barre d'outils Analytics.


1. Si besoin, ajoutez le bouton **Vérifier les scripts** à la barre d'outils Analytics :
  - a. Dans la barre d'outils, double-cliquez sur un endroit vide pour ouvrir la boîte de dialogue **Personnaliser la barre d'outils**.
  - b. Dans la liste **Boutons de barre d'outils disponibles**, sélectionnez le bouton **Vérifier les scripts**, puis cliquez sur **Ajouter**.
  - c. Dans la liste **Boutons actuels de la barre d'outils**, sélectionnez le bouton **Vérifier les scripts** et cliquez sur **Haut** ou **Bas** pour modifier son emplacement.

L'ordre des boutons de haut en bas correspond à leurs positions de gauche à droite, dans la barre d'outils.

- d. Cliquez sur **Fermer** afin d'enregistrer vos modifications.



2. Dans la barre d'outils, cliquez sur **Vérifier les scripts** .
 

Un message s'affiche pour vous indiquer si la syntaxe du script du projet est valide ou bien pour signaler un ou plusieurs avertissements.
3. Effectuez l'une des actions suivantes :
  - Corrigez les syntaxes de script générant un avertissement, puis recliquez sur **Vérifier les scripts**  pour vous assurer que les avertissements n'apparaissent plus.
  - Assurez-vous que l'environnement de déploiement contient une structure de répertoire ou des scripts externes qui correspondent aux chemins ou scripts externes indiqués dans l'outil d'analyse.

## Bonnes pratiques à adopter pour l'exécution d'outils d'analyse sur AX Serveur

### Développer dans Analytics

Développez avant tout les outils d'analyse et les scripts les accompagnant dans Analytics avant de les importer dans AX Serveur.

Par commodité, l'éditeur de script d'AX Client vous permet d'ajouter de nouveaux outils d'analyse ou de nouveaux sous-scripts ou bien de modifier des outils d'analyse ou des sous-scripts existants. Cette fonctionnalité s'avère utile pour ajuster le comportement d'un outil d'analyse sans avoir à l'exporter dans Analytics, puis à le réimporter dans AX Serveur. Cependant, en dehors des ajustements mineurs, il est plus simple d'effectuer le travail de développement d'outil d'analyse dans Analytics.

### Stocker les fichiers associés avec le projet Analytics

Les fichiers associés, comme les fichiers des profils de base de données, doivent être stockés dans le même dossier que le projet Analytics, mais doivent être importés séparément dans AX Serveur.

### Éviter les commandes non prises en charge sur AX Serveur

- tables de serveur de bases de données directes liées à Analytics Édition Serveur pour z/OS
- la commande NOTIFY prend uniquement en charge la messagerie SMTP Les protocoles de messagerie MAPI et VIM ne sont pas pris en charge
- pour utiliser la commande PRINT ou TO PRINT, une imprimante par défaut doit être configurée sur le serveur
- les commandes SAVE GRAPH et PRINT GRAPH ne sont pas prises en charge
- n'utilisez pas la commande SET LEARN dans les outils d'analyse.

### Minimiser les transactions des tables d'AX Serveur

Optimisez la performance des outils d'analyse en minimisant le nombre de fois où l'on accède à des tables

sur AX Serveur :

1. Utilisez la commande `FILTER` pour sélectionner les enregistrements dont vous avez besoin.
2. Utilisez la commande `EXTRACT` pour extraire uniquement les champs requis.

Le jeu de données réduit sera traité localement sur le serveur d'exécution de l'outil d'analyse par le Moteur AX.

Il est important d'optimiser vos outils d'analyse de cette façon lorsque les fichiers de données ne se trouvent pas sur le même serveur qu'AX Serveur ou que le Nœud moteur d'AX traitant l'outil d'analyse et lorsque l'option **Copier les données d'analyse** n'est pas sélectionnée dans l'application Web Configuration d'AX Serveur.

## Exemple d'outil d'analyse inefficace

```
OPEN GrandeTable
SET FILTER TO date_trans >= `20091201` AND date_trans < `20100101`
COUNT
TOTAL montant
CLASSIFY ON compte ACCUMULATE montant TO CompteClasseTrans
```

## Exemple d'outil d'analyse efficace

```
OPEN GrandeTable
SET FILTER TO date_trans >= `20091201` AND date_trans < `20100101`
EXTRACT FIELDS date_trans desc compte type montant TO TableAnalyse
OPEN TableAnalyse
COUNT
TOTAL montant
CLASSIFY ON compte ACCUMULATE montant TO CompteClasseTrans
```

## Accéder aux données SAP en mode Fond

Utilisez le mode Fond pour accéder aux données issues de systèmes ERP SAP à l'aide de Direct Link.

# Compresser des Apps Analyse

Pour exécuter des scripts d'outils d'analyse dans la fenêtre App Analyse ou sur AX Serveur, créez une App Analyse packagée (fichier `.aclapp`). Vous pouvez compresser un projet Analytics dans un nouveau fichier `.aclapp` ou fusionner un projet Analytics avec une App Analyse existante (fichier `.aclx`) pour inclure les interprétations existantes.

## Remarque

Si au moins un script d'un projet Analytics contient un en-tête d'outil d'analyse, le projet peut être compressé en tant qu'App Analyse. Avant de compresser une App Analyse, assurez-vous d'avoir validé les en-têtes des outils d'analyse de chaque outil d'analyse dans l'App Analyse.

## Pourquoi utiliser une App Analyse packagée ?

### Distribution auprès des utilisateurs de la fenêtre App Analyse

Utilisez des Apps Analyse packagées pour distribuer un projet aux utilisateurs pouvant extraire le fichier `.aclx` et ouvrir son contenu dans la fenêtre App Analyse.

Dans la fenêtre App Analyse, vous pouvez exécuter les scripts d'outils d'analyse et créer des interprétations à partir des tables et des résultats d'outils d'analyse.

### Importation dans AX Serveur

Utilisez des Apps Analyse packagées pour préparer un projet Analytics à importer dans AX Serveur. Vous pouvez choisir quels tables et fichiers de données importer avec les scripts d'outils d'analyse dans le projet.

Vous pouvez aussi utiliser une App Analyse packagée (fichier `.aclapp`) pour importer des interprétations existantes. Pour inclure les interprétations d'une App Analyse existante (`.aclx`), vous devez recompresser votre projet Analytics et fusionner cette App Analyse packagée (fichier `.aclapp`) avec le fichier `.aclx` existant dans votre dossier de projet.

## Remarque

Lorsque vous utilisez une App Analyse existante (fichier `.aclx`), le contenu du projet Analytics est prioritaire. Par conséquent, si des scripts ou des tables de `.aclx` n'existent plus dans le fichier `.acl`, ils ne sont pas inclus dans l'App Analyse packagée alors créée (fichier `.aclapp`).

## Limitation des tailles de fichiers

Pour bien utiliser une App Analyse compressée, vous devez vous assurer que la somme de toutes les tailles de fichiers inclus dans le package ne dépasse pas les 800 Mo avant de compresser l'App Analyse. Si vos fichiers précompressés dépassent cette taille totale, les fichiers de données peuvent être corrompus au moment de la décompression de l'App Analyse.

## Compresser une nouvelle App Analyse

1. Dans Analytics, cliquez avec le bouton droit de la souris sur l'entrée du projet dans l'onglet **Vue globale** du **Navigateur**, puis sélectionnez **Package App Analyse**.

Le projet Analytics est le dossier de premier niveau dans l'arborescence.

2. Dans la boîte de dialogue **Sélectionner des tables**, procédez comme suit :
  - a. Si vous souhaitez inclure une ou plusieurs tables du projet et des fichiers de données associés dans l'App Analyse, sélectionnez la ou les tables et fichiers de données à inclure.

### Remarque

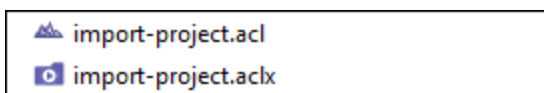
En général, il est conseillé d'inclure uniquement les fichiers de données et les tables statiques requis par un ou plusieurs outils d'analyse dans l'App Analyse, comme une table maîtresse de fournisseurs ou une liste de codes de catégories de marchands.

- b. Cliquez sur **A**, puis accédez à l'emplacement où vous souhaitez enregistrer l'App Analyse compressée.
- c. Dans la boîte de dialogue **Enregistrer sous**, saisissez un **Nom de fichier** portant l'extension **.aclapp**, puis cliquez sur **Enregistrer**.
- d. Cliquez sur **OK**.

**Résultat** : L'App Analyse compressée est enregistrée à l'emplacement que vous avez spécifié. Les autres utilisateurs peuvent récupérer l'App Analyse compressée à partir de cet emplacement ou vous pouvez la distribuer par e-mail ou par tout autre méthode appropriée. Vous pouvez aussi importer ce fichier dans AX Serveur.

## Compresser une App Analyse avec des interprétations

1. Assurez-vous que l'App Analyse (fichier .aclx) qui contient les interprétations que vous souhaitez importer existe dans le dossier de projet Analytics sur votre ordinateur et qu'elle utilise le même nom de fichier que le projet Analytics (fichier **.acl**).



2. Dans Analytics, cliquez avec le bouton droit de la souris sur l'entrée du projet dans l'onglet **Vue globale** du **Navigateur**, puis sélectionnez **Package App Analyse**.

Le projet Analytics est le dossier de premier niveau dans l'arborescence.

3. Dans la boîte de dialogue **Sélectionner des tables**, procédez comme suit :
  - a. Si vous souhaitez inclure une ou plusieurs tables du projet et des fichiers de données associés dans l'App Analyse, sélectionnez la ou les tables et fichiers de données à inclure.

#### Remarque

En général, il est conseillé d'inclure uniquement les fichiers de données et les tables statiques requis par un ou plusieurs outils d'analyse dans l'App Analyse, comme une table maîtresse de fournisseurs ou une liste de codes de catégories de marchands.

- b. Facultatif. Pour inclure les interprétations à partir de l'App Analyse existante, sélectionnez **Inclure les interprétations**.

Les interprétations associées à des tables ou à des scripts qui n'existent pas dans le nouveau package ne sont pas incluses.

- c. Cliquez sur **À**, puis accédez à l'emplacement où vous souhaitez enregistrer l'App Analyse compressée.
  - d. Dans la boîte de dialogue **Enregistrer sous**, saisissez un **Nom de fichier** portant l'extension **.aclapp**, puis cliquez sur **Enregistrer**.
  - e. Cliquez sur **OK**.

**Résultat** : L'App Analyse compressée est enregistrée à l'emplacement que vous avez spécifié. Les autres utilisateurs peuvent récupérer l'App Analyse compressée à partir de cet emplacement ou vous pouvez la distribuer par e-mail ou par tout autre méthode appropriée. Vous pouvez aussi importer ce fichier dans AX Serveur.

# Exemples de scripts d'outils d'analyse (App Analyse)

Les exemples de scripts d'outils d'analyse contiennent un outil d'analyse d'importation (deux versions), un outil d'analyse de préparation et un outil d'analyse d'analyse. Les scripts d'outils d'analyse peuvent être exécutés dans les environnements ou applications clientes qui suivent :

- Robots
- Serveur AX:
  - Client AX
  - Client Web d'AX
- la fenêtre App Analyse

## Ordre d'exécution des scripts d'outils d'analyse

Les trois outils d'analyse sont destinés à être utilisés conjointement et doivent être exécutés dans l'ordre suivant :

Séquence	ANALYTIC TYPE	Nom de l'outil d'analyse
1	IMPORT	Exemple d'importation d'outil d'analyse Robots_AX ou Exemple d'outil d'analyse d'importation Web_AA_Window
2	PREPARE	Exemple d'outil d'analyse de préparation
3	ANALYSE	Exemple d'outil d'analyse d'analyse

## Exemple d'outil d'analyse d'importation

Importe des données de l'exemple de fichier Excel `Trans_Mai.xls` et les enregistre dans la nouvelle table Analytics `Trans_Mai_brute` (la table de données brute).

Deux versions de cet outil d'analyse sont fournies :

Nom de l'outil d'analyse	À utiliser dans	Condition requise pour le fichier d'importation
Exemple d'importation d'outil d'analyse Robots_AX	<ul style="list-style-type: none"> <li>◦ Robots</li> <li>◦ Client AX</li> </ul>	<ul style="list-style-type: none"> <li>◦ Robots <code>Trans_Mai.xls</code> doit être situé dans l'onglet <b>Entrée/Sortie</b> du même robot que l'outil d'analyse</li> <li>◦ AX Client <code>Trans_Mai.xls</code> doit être situé dans le</li> </ul>

Nom de l'outil d'analyse	À utiliser dans	Condition requise pour le fichier d'importation
		sous-répertoire <b>Fichiers associés</b> du dossier AX à l'emplacement de l'outil d'analyse
Exemple d'outil d'analyse d'importation Web_AA_Window	<ul style="list-style-type: none"> <li>○ Client Web d'AX</li> <li>○ Fenêtre App Analyse</li> </ul>	

## Exemple d'outil d'analyse d'importation à utiliser dans Robots ou AX Client

```
COMMENT
//ANALYTIC TYPE IMPORT Exemple d'outil d'analyse d'importation Robots_AX
  Cet outil d'analyse importe des données de l'exemple de fichier Excel Trans_Mai.xls et l'enregistre
dans la nouvelle table Analytics « Trans_Mai_brute » (la table de données brute).
//FILE Trans_Mai.xls
//DATA Trans_Mai_brut
//RESULT LOG
END

SET SAFETY OFF
IMPORT EXCEL TO Trans_Mai_brut Trans_Mai_brut.fil FROM "Trans_Mai.xls" TABLE "Trans2_
Mai$" KEeptitle FIELD "NUMCARTE" C WID 22 AS "" FIELD "CODES" C WID 4 AS "" FIELD
"DATE" D WID 10 PIC "AAAA-MM-JJ" AS "" FIELD "NUMCLIENT" C WID 6 AS "" FIELD
"DESCRIPTION" C WID 95 AS "" FIELD "MONTANT" N WID 9 DEC 2 AS ""
SET SAFETY ON
```

## Exemple d'outil d'analyse d'importation à utiliser dans le Client Web d'AX ou dans la fenêtre App Analyse

```
COMMENT
//ANALYTIC TYPE IMPORT Exemple d'outil d'analyse d'importation Web_AA_Window
  Cet outil d'analyse importe des données de l'exemple de fichier Excel Trans_Mai.xls et l'enregistre
dans la nouvelle table Analytics « Trans_Mai_brute » (la table de données brute).
//PARAM v_fichier_entrée F Fichier d'entrée
  Sélectionner un fichier d'entrée
//DATA Trans_Mai_brut
//RESULT LOG
END

SET SAFETY OFF
```

```
IMPORT EXCEL TO Trans_Mai_brut Trans_Mai_brut.fil FROM "%v_fichier_entrée%" TABLE
"Trans2_Mai$" KEEPTITLE FIELD "NUMCARTE" C WID 22 AS "" FIELD "CODES" C WID 4 AS ""
FIELD "DATE" D WID 10 PIC "AAAA-MM-JJ" AS "" FIELD "NUMCLIENT" C WID 6 AS "" FIELD
"DESCRIPTION" C WID 95 AS "" FIELD "MONTANT" N WID 9 DEC 2 AS ""
SET SAFETY ON
```

## Exemple d'outil d'analyse de préparation

Prépare la table de données brute pour l'analyse et l'enregistre dans la nouvelle table Analytics `Trans_Mai_préparée` (la table d'analyse). L'outil d'analyse définit une version plus courte du champ « Description », car le classement prend uniquement en charge des longueurs de champ de 64 caractères au maximum.

```
COMMENT
//ANALYTIC TYPE PREPARE Exemple d'outil d'analyse de préparation
  Cet outil d'analyse prépare la table de données brutes pour analyse et l'enregistre dans la nouvelle
  table Analytics « Trans_Mai_préparée » (la table d'analyse). L'outil d'analyse définit une version plus
  courte du champ « Description », car le classement prend uniquement en charge des longueurs de
  champ de 64 caractères au maximum.
//TABLE v_TableBrute Table à préparer
  Sélectionnez la table de données brutes que vous souhaitez préparer
//RESULT TABLE Trans_*_préparée
//DATA Trans_*_préparée
//RESULT LOG
END

SET SAFETY OFF
OPEN %v_TableBrute%
DEFINE FIELD DESC_SHORT  ASCII  43 64
EXTRACT RECORD TO "Trans_Mai_préparée"
SET SAFETY ON
```

## Exemple d'outil d'analyse d'analyse

Classe la table d'analyse et sort les résultats dans une nouvelle table Analytics `Classée_Trans_Mai_préparée` (la table de résultats). Les utilisateurs peuvent spécifier le champ à utiliser pour classer la table ainsi que les codes de catégories de marchands, les numéros de clients, les périodes et les plages des montants de transactions pour limiter le nombre d'enregistrements à traiter.

```
COMMENT
//ANALYTIC TYPE ANALYSIS Exemple d'outil d'analyse d'analyse
```



Cet outil d'analyse classe la table d'analyse et sort les résultats dans une nouvelle table Analytics « Classée\_Trans\_Mai\_préparée » (la table de résultats). Vous pouvez spécifier les codes de catégories de marchands, les numéros de clients, les périodes et les plages des montants de transactions pour limiter le nombre d'enregistrements à traiter.

```
//TABLE v_TableAnalyse Table à classer
```

Sélectionnez la table d'analyse que vous souhaitez classer

```
//FIELD v_ChampA C Champ par rapport auquel classer
```

Sélectionnez le champ selon lequel classer

```
//PARAM v_codes C MULTI SEPARATOR , QUALIFIER ' VALUES |4112 Transport ferroviaire de pas-
sagers|4121 Taxis/Limousines|4131 Voyage en autobus|4215 Services de coursier - Air ou sol|4411
Croisières|4457 Locations de bateau|4722 Agences de voyages et tour-opérateurs|4814 Appels
locaux/longue distance|5812 Restaurants|5813 Cafés (boissons alcoolisées)|5814 Fast foods|5921
Détaillants d'alcool, bière, vin, liqueur|5993 Marchands de cigares et comptoirs à tabac|5994 Bura-
liste|7216 Nettoyage à sec| MC Code(s) à inclure
```

Spécifiez un ou plusieurs codes de catégories de marchands à inclure

```
//PARAM v_num_client C OPTIONAL MULTI SEPARATOR , QUALIFIER ' Numéro(s) client à exclure
(facultatif)
```

Spécifier un ou plusieurs numéros clients à exclure. Appuyez sur « Entrée » après chaque numéro pour que chaque numéro se trouve sur une ligne distincte. N'entourez pas les numéros de guillemets.

```
//PARAM v_date_début D
```

```
VALUES|05/01/2003|05/02/2003|05/03/2003|05/04/2003|05/05/2003|05/06/2003|05/07/2003|05/08/2-
003|05/09/2003|05/10/2003|05/11/2003|05/12/2003|05/13/2003|05/14/2003|05/15/2003|05/16/2003|0-
5/17/2003|05/18/2003|05/19/2003|05/20/2003|05/21/2003|05/22/2003|05/23/2003|05/24/2003|05/25/-
2003|05/26/2003|05/27/2003|05/28/2003|05/29/2003|05/30/2003|05/31/2003|Date de début
```

Sélectionnez une date de début

```
//PARAM v_date_fin D Date de fin
```

Saisissez une date de fin ou sélectionnez-en une dans le calendrier

```
//PARAM v_montant_min N Montant minimum
```

Saisissez un montant minimum

```
//PARAM v_montant_max N Montant maximum
```

Saisissez un montant maximum

```
//RESULT TABLE Classée_*
```

```
//RESULT LOG
```

```
END
```

```
SET SAFETY OFF
```

```
OPEN %v_TableAnalyse%
```

```
IF NOT ISDEFINED("v_num_client") v_num_client = ""
```

```
GROUP IF v_num_client = ""
```

```
CLASSIFY ON %ChampA% IF MATCH(CODES; %v_codes%) AND BETWEEN(DATE; v_date_
début; v_date_fin) AND BETWEEN(MONTANT; v_montant_min; v_montant_max)SUBTOTAL
MONTANT TO "Classée_%v_TableAnalyse%.FIL" OPEN
```

```
ELSE
```

```
CLASSIFY ON %v_ChampA% IF MATCH(CODES; %v_codes%) AND NOT MATCH(NUMCLIENT;
```

```
%v_num_client%) AND BETWEEN(DATE; v_date_début; v_date_fin) AND BETWEEN  
(MONTANT; v_montant_min; v_montant_max) SUBTOTAL MONTANT TO "Classée_%v_TableA-  
nalyse%.FIL" OPEN  
END  
SET SAFETY ON
```

# Annexe

# Configuration minimale requise

Avant d'installer Analytics, assurez-vous que votre ordinateur présente la configuration matérielle et logicielle minimale requise.

## Configuration logicielle requise

### Remarque

Certains prérequis logiciels sont installés automatiquement s'ils ne sont pas déjà présents sur votre ordinateur. Pour obtenir la liste complète des prérequis installés automatiquement, consultez la [documentation en ligne](#).

Condition préalable	Autres informations
<p>Un des systèmes d'exploitation suivants :</p> <ul style="list-style-type: none"> <li>○ Microsoft Windows 10 (64-bit)</li> <li>○ Microsoft Windows 8.1 (64 bits)</li> <li>○ Microsoft Windows 7 Service Pack 1 (SP1) (32 bits / 64 bits)</li> </ul>	<p>ACL pour Windows est une application 32 bits qui peut être exécutée sur les versions 64 bits de Windows.</p> <p><b>Remarque</b></p> <p>Pour installer ACL pour Windows sur Windows 7, le Service Pack 1 doit être installé. ACL pour Windows a besoin de Microsoft .NET 4.6, qui ne peut pas être installé sur les versions de Windows 7 antérieures à SP1.</p> <p>Le système d'exploitation Windows XP n'est plus pris en charge.</p>
<p>Pour les utilisateurs de Microsoft Windows 8.1 :</p> <p><a href="#">Mise à jour Windows 8.1 KB2919355</a></p>	<p><b>Important</b></p> <p>La mise à jour Windows 8.1 KB2919355 est requise par Microsoft .NET Framework 4.6.x, qui est aussi requis par ACL pour Windows 14.</p> <p>Si vous utilisez Windows 8.1, que .NET 4.6.x n'est pas installé et que vous n'avez pas exécuté la mise à jour KB2919355, un message d'erreur interrompt le programme d'installation d'ACL pour Windows pendant l'installation du composant prérequis .NET 4.6.2.</p> <p>Vous devez télécharger et installer la mise à jour KB2919355 avant de pouvoir poursuivre l'installation d'ACL pour Windows.</p> <p>Autrement, vous pouvez installer la mise à jour KB2919355 avant de lancer l'installation d'ACL pour Windows et éviter ainsi le message d'erreur.</p> <p><b>Attention</b></p> <p>Si vous êtes invité à redémarrer votre ordinateur à un moment pendant l'installation, faites-le aussitôt. <b>N'ignorez pas les messages concernant le redémarrage de votre ordinateur.</b></p> <p>Si vous ne redémarrez pas votre ordinateur lorsque vous y êtes invité, il se peut que vous génériez des problèmes avec l'installation de .NET, d'autres composants requis ou d'ACL pour Windows.</p>

Condition préalable	Autres informations
<p>Pour utiliser les fonctions Analytics intégrant le langage de programmation R, vous devez installer et configurer R (32 bits/64 bits) :</p> <p>Les versions suivantes de R ont été testées avec Analytics et fonctionnent :</p> <ul style="list-style-type: none"> <li>○ 3.3.2</li> <li>○ 3.3.1</li> <li>○ 3.2.5</li> <li>○ 3.2.3</li> </ul> <p>Vous pouvez utiliser CRAN R (32 bits/64 bits) ou Microsoft R (64 bits seulement).</p> <p><b>Remarque</b></p> <p>D'autres versions de R fonctionnent probablement aussi. Toutefois, leur fonctionnement n'est pas garanti.</p> <p>À l'heure actuelle, R 3.5.0 et les versions supérieures ne sont pas prises en charge pour être utilisées avec Analytics.</p>	<p>Le nombre de bits de la version installée de R doit correspondre au nombre de bits du système d'exploitation.</p> <p>Si vous utilisez l'un des packages CRAN R, il se peut que vous deviez ajouter le chemin d'accès au dossier binaire R à la variable d'environnement PATH sur votre ordinateur.</p> <p>Par exemple :</p> <ul style="list-style-type: none"> <li>○ C:\Program Files\R\R-&lt;version&gt;\bin\i386 (32 bits)</li> <li>○ C:\Program Files\R\R-&lt;version&gt;\bin\x64 (64 bits)</li> </ul> <p><b>Remarque</b></p> <p>Vous n'avez pas besoin d'installer R si vous n'avez pas l'intention d'utiliser les fonctions Analytics intégrées à ce langage.</p>
<p>Pour utiliser les fonctions Analytics s'intégrant au langage de programmation Python, vous devez installer et configurer :</p> <ul style="list-style-type: none"> <li>○ Python version 3.3.x ou versions ultérieures (32 bits)</li> <li>○ Variable d'environnement PYTHONPATH</li> <li>○ Variable d'environnement ACLPYTHONDLL</li> </ul>	<p>Python version 3.5.x est entièrement testé et pris en charge. Vous pouvez utiliser une version différente comme 3.3.x ou 3.6.x, toutefois, ces versions n'offrent pas la même garantie de test et de prise en charge avec Analytics que la version 3.5.x.</p> <p>Lors de l'installation de Python, vous devez également le configurer pour l'exécuter sur votre système. Pour plus d'informations, consultez la section "Configuration de Python pour être utilisé avec Analytics" Page 953.</p> <p><b>Remarque</b></p> <p>Vous n'avez pas besoin d'installer Python si vous n'avez pas l'intention d'utiliser les fonctions Analytics intégrées à ce langage.</p> <p>La copie locale du moteur Python contenue dans le répertoire d'installation d'Analytics n'est pas destinée à être utilisée avec les fonctions Python d'Analytics ou à l'utilisation générale de Python. A cette fin, vous devez installer une instance distincte de Python.</p>
<p>Pour utiliser le connecteur ACL pour Oracle, vous devez installer :</p>	<ul style="list-style-type: none"> <li>○ Vous n'avez pas besoin d'installer Oracle Instant Client si vous n'avez pas l'intention d'utiliser le Connecteur ACL pour Oracle</li> <li>○ Le nombre de bits d'Oracle Instant Client doit correspondre au nombre de bits de votre système d'exploitation. Si Oracle Instant Client 32 bits est installé sur</li> </ul>

Condition préalable	Autres informations
<ul style="list-style-type: none"> <li>Oracle Instant Client 11g ou 12c</li> </ul>	<ul style="list-style-type: none"> <li>une machine de 64 bits, la connexion échoue.</li> <li>Si vous utilisez le connecteur avec Analytics Exchange et installez Oracle Instant Client après le serveur AX, vous devez relancer le service Analytics Exchange avant de pouvoir utiliser le connecteur.</li> </ul>

## Configuration matérielle requise

### Remarque

Il est possible que des ressources supérieures à la configuration minimale soient nécessaires afin de fournir les meilleures performances Analytics possible dans un environnement de production.

Composant	Minimum	Recommandation
Processeur	1,8 GHz	
Mémoire (RAM)	2 Go	<ul style="list-style-type: none"> <li>Systèmes d'exploitation 64 bits : 8 Go ou plus, en particulier pour trier des fichiers volumineux</li> <li>Systèmes d'exploitation 32 bits : 4 Go, en particulier pour trier des fichiers volumineux</li> </ul>
Espace disque dur (fichiers d'application Analytics)	1,1 Go	
Espace disque dur (prérequis logiciels)	8 Go	
Espace disque dur (stockage des données)		<p>100 Go ou plus</p> <p>Outre l'espace disque dur requis pour installer les fichiers d'application Analytics et les conditions préalables, un espace disque dur considérablement plus volumineux peut être nécessaire si un ordinateur est utilisé pour stocker des extraits de données, des fichiers plats et des résultats.</p>

# Installer ACL pour Windows

Suivez les instructions d'installation d'ACL pour Windows pour installer ou mettre à niveau votre copie d'Analytics. Une fois l'installation terminée, vous devez activer votre licence.

## Remarque

Lorsque vous installez ou mettez à niveau Analytics, les exemples de fichiers de données Analytics existants sont écrasés s'ils se trouvent dans le répertoire de travail Analytics spécifié pendant l'installation ou la mise à niveau.

Si vous avez apporté des modifications à des exemples de projets ou à des fichiers de données que vous souhaitez conserver, enregistrez les fichiers ailleurs avant l'installation ou la mise à niveau ou bien renommez le dossier qui les contient. Faites de même pour les fichiers trace des commandes associées que vous souhaitez conserver.

## Obtenir le programme d'installation

Obtenez le programme d'installation auprès du contact primaire de votre organisation pour Galvanize. Si vous êtes le contact primaire :

- **Accès à Internet** : obtenez le programme d'installation à partir de la page des téléchargements dans la Barre de lancement (<http://www.aclgrc.com/>)
- **Aucun accès à Internet** : demandez à votre représentant Galvanize de vous aider à obtenir le programme d'installation à partir de la Barre de lancement

## Extraire le programme d'installation

1. Double-cliquez sur le package d'installation ACL pour Windows (`ACLforWindows141.exe`).
2. Si une boîte de dialogue d'avertissement de sécurité s'affiche, vérifiez les informations indiquées et cliquez sur **Oui**.
3. Sélectionnez la langue à utiliser pour votre installation, puis cliquez sur **OK**.
4. Sur la page **Emplacement d'extraction des fichiers d'installation**, cliquez sur **Extraire**.

Le programme d'installation se lance automatiquement après l'extraction des fichiers.

## Installer les prérequis

Si vous êtes invité à installer des composants requis, cliquez sur **Installer**.

Après leur installation, le programme d'installation se poursuit automatiquement.

## Installer ACL pour Windows

Suivez les instructions à l'écran pour effectuer l'installation d'ACL pour Windows.

Sur la page **Sélection de l'édition ACL**, sélectionnez l'édition à installer.

- **Non Unicode**
- **Unicode**

Pour plus d'informations sur les éditions Unicode et non Unicode, consultez la rubrique "Éditions Unicode et non Unicode" Page 956.

## Activer votre licence

Si vous avez accès à Internet, vous pouvez activer votre licence la première fois que vous lancez ACL pour Windows et que vous vous connectez à votre organisation. Pour activer votre licence hors connexion, contactez l'Assistance Galvanize.

## Démarrer Analytics

Pour lancer Analytics, effectuez l'une des actions suivantes :

Pour créer un nouveau projet Analytics vide	Sous <b>Créer</b> , cliquez sur <b>Projet d'outil d'analyse</b>
Pour ouvrir un projet Analytics existant	Sous <b>Ouvrir</b> , cliquez sur <b>Projet d'outil d'analyse</b>
Pour ouvrir un exemple de projet ou un projet Analytics récent (.acl)	Sous <b>Fichiers d'outils d'analyse récents</b> ou <b>Exemples de fichiers</b> , cliquez sur le nom d'un projet



# Configuration de Python pour être utilisé avec Analytics

Pour configurer Python afin de l'utiliser avec Analytics, vous devez installer la bonne version de Python et ajouter l'exécutable à la variable d'environnement PATH de votre système. Vous devez également définir les variables d'environnement ACLPYTHONDLL et PYTHONPATH de votre système.

## Fonctionnement

Pour exécuter des scripts Python, Analytics doit pouvoir appeler l'exécutable Python et trouver les scripts qu'on lui ordonne d'exécuter. Analytics utilise la variable d'environnement PATH pour localiser Python et la variable d'environnement PYTHONPATH pour localiser les scripts.

## Installer Python version 3.5.x ou versions ultérieures (32 bits)

1. À partir de la [page des téléchargements Python](#), téléchargez la dernière version de Python 3.5 sur votre poste.
2. Sur votre poste, double-cliquez sur le programme d'installation.
3. Dans le programme d'installation, sélectionnez **Ajouter Python *versionNumber* à PATH**.
4. Cliquez sur **Installer** et suivez les instructions à l'écran.
5. Redémarrez la machine avant d'exécuter des scripts Python depuis Analytics.

## Définir les variables d'environnement ACLPYTHONDLL et PYTHONPATH

1. Dans votre système d'exploitation Windows, créez un ou plusieurs dossiers qui hébergeront vos scripts Python.  
**Exemple** : `C:\python_scripts`.
2. Dans le système d'exploitation Windows, ouvrez la boîte de dialogue **Propriétés système**, puis cliquez sur **Variables d'environnement**.
3. Dans la section **Variables système**, cliquez sur **Nouveau**, puis saisissez les variables suivantes :

Nom de la variable	Valeur de la variable	Exemple
PYTHONPATH	Le chemin d'accès complet au(x) dossier(s) que vous avez créé(s)	C:\python_scripts;C:\dev;C:\tmp

Nom de la variable	Valeur de la variable	Exemple
	pour héberger les scripts Python. Séparez plusieurs chemins d'accès au dossier par un point-virgule.	
ACLPYTHONDLL	<p>Nom du chemin d'accès complet et nom du fichier Python DLL dans le dossier d'installation Python à utiliser avec Analytics.</p> <p>Si vous utilisez Python 3.3.x, les restrictions suivantes s'appliquent :</p> <ul style="list-style-type: none"> <li>Les caractères Unicode ne sont pas pris en charge dans le chemin d'accès pour les plateformes européennes</li> <li>Les caractères étendus ne sont pas pris en charge dans le chemin d'accès pour les plateformes asiatiques</li> </ul> <p><b>Remarque</b></p> <p>Python ajoute la DLL au dossier système (<code>c:\windows\system32\python33.dll</code>) plutôt qu'au dossier d'installation. Vous devez copier le fichier DLL du dossier système vers le dossier d'installation et l'utiliser comme valeur de variable afin qu'Analytics puisse accéder à la DLL.</p> <p>Il se peut que vous deviez aussi supprimer les paramètres de lecture seule du dossier d'installation.</p> <p>Si vous ne définissez pas cette valeur, Analytics tente d'utiliser la DLL version 3.5.x prise en charge par défaut <code>python35.dll</code>.</p>	c:\python_install\python33.dll

- Pour enregistrer la variable, cliquez sur **OK**, puis, dans la boîte de dialogue **Propriétés système**, cliquez à nouveau sur **OK**.

# Utiliser Python dans les fonctions Analytics Python

Dans Analytics, utilisez les fonctions Analytics Python pour appeler des fonctions dans les scripts existant dans PYTHONPATH.

## Remarque

Si vous apportez des modifications à un script Python, vous devez actualiser la vue dans votre projet Analytics pour utiliser la dernière version du script Python. La manière la plus simple d'actualiser la vue consiste à fermer la table avec laquelle vous travaillez et à la rouvrir.

# Éditions Unicode et non Unicode

Les éditions Unicode des produits Analytics vous permettent d'afficher des fichiers contenant des données Unicode et de les utiliser. Unicode est un standard reconnu de codage des caractères prenant en charge la plupart des langages du monde.

## Dois-je installer l'édition non Unicode ou Unicode ?

Analytics est disponible dans des éditions non Unicode et Unicode. Les deux éditions se trouvent dans le même package d'installation ; c'est pendant l'installation que vous indiquez quelle édition installer.

Vous devez installer l'édition non-Unicode, même si vous devez afficher ou analyser des données Unicode. Les données Unicode peuvent être ouvertes uniquement dans l'édition Unicode d'Analytics.

Vous devriez trouver plus de données Unicode si vous évoluez dans un environnement disposant de systèmes d'information globaux, ou si vous analysez des données contenant plusieurs langages.

## Lorsque l'édition Unicode est requise

Vous devez installer l'édition Unicode pour afficher ou analyser des données avec les caractères suivants :

- Caractères asiatiques
- une combinaison de codages de caractères non Unicode ou traditionnels

Par exemple, certaines combinaisons de langues issues d'au moins deux de ces codages de caractères :

- Latin 1 (anglais et langues d'Europe de l'Ouest)
- Latin 2 (langues d'Europe centrale)
- Cyrillique
- Grec
- Arabe

### Remarque

Si vous souhaitez utiliser l'interface utilisateur Analytics en chinois, en japonais ou en polonais, la seule solution consiste à installer l'édition Unicode. Cette condition préalable est liée à la langue de l'interface utilisateur et non à la langue des données.

# Convertir des outils d'analyse au format Unicode

Si vous migrez de l'édition non Unicode à l'édition Unicode d'Analytics, les outils d'analyse et les scripts existants sont convertis automatiquement au format Unicode. Cependant, vous devez vérifier que la logique des scripts reste identique lorsqu'elle est appliquée aux données Unicode à deux octets.

## Qu'est-ce qu'Unicode ?

Unicode est une norme standard de texte codé qui utilise deux octets ou plus pour représenter chaque caractère. Les caractères de tous les langages sont contenus dans un jeu de caractères unique. Les éditions au format Unicode des produits Galvanize vous permettent d'afficher et d'utiliser les fichiers et les bases de données contenant des données codées en Unicode dans toutes les langues vivantes.

### Remarque

Analytics et le Moteur AX prennent en charge les données Unicode codées en tant que petit-boutiste (LE). Ces produits ne peuvent pas être utilisés pour analyser des données codées en tant que gros-boutiste (BE).

## Migration vers Analytics Exchange édition Unicode

- le cryptage des scripts Unicode n'est pas pris en charge actuellement.
- les fichiers de projet et les fichiers traces Analytics sont codés sous forme de données Unicode (UTF-16 LE) et ne peuvent pas être utilisés avec l'édition non Unicode d'Analytics.
- lorsque vous utilisez Analytics pour définir des fichiers image d'impression et des fichiers délimités qui contiennent du texte au format ASCII ou EBCDIC, les champs de la table Analytics contenant ces données sont de type Unicode par défaut.

## Modifications requises sur les outils d'analyse

### Mettre à jour les paramètres indiquant une valeur en octets

Les caractères de l'édition non Unicode d'Analytics présentent une longueur d'un octet. Les caractères de l'édition Unicode - s'il s'agit de données Unicode - présentent une longueur de deux octets. Lorsque vous indiquez une longueur de champ ou une position de départ en octets dans l'édition non Unicode d'Analytics, le nombre d'octets est égal au nombre de caractères. Cela ne se vérifie pas dans les données Unicode de l'édition Unicode d'Analytics.

Pour convertir des outils d'analyse afin de les utiliser dans Analytics Unicode, vous devez ajuster la valeur numérique des paramètres indiquant la longueur de champ ou la position de départ en octets. Par exemple, dans le cas d'une commande IMPORT DELIMITED indiquant une valeur WID de 7 dans Analytics non Unicode, vous devez doubler la valeur WID sur 14 pour avoir le même résultat dans Analytics Unicode.

En outre, dans le cas de données Unicode, indiquez une position d'octet de départ impair pour les champs et un nombre d'octets pair pour les longueurs de champ. Indiquer une position de départ avec un chiffre pair ou une longueur de champ impaire peut entraîner l'affichage incorrect des caractères.

## Recréer toutes les instances de IMPORT PRINT et de IMPORT DELIMITED

Vous devez recréer toutes les instances des commandes IMPORT PRINT et IMPORT DELIMITED en important le fichier de données source à l'aide de l'Assistant de définition des données dans la version Unicode d'Analytics et en réimportant les projets dans AX Serveur. L'utilisation de l'Assistant de définition de données garantit la validité de l'ensemble de la syntaxe.

## Modifier toutes les instances des fonctions ZONED( ) et EBCDIC( )

Vous devez modifier toutes les instances des fonctions ZONED() et EBCDIC() comme suit pour que les valeurs ASCII renvoyées par les fonctions soient correctement converties en données Unicode :

- Les **champs calculés** : entourent la fonction BINTOSTR() autour des instances ZONED() ou EBCDIC()
- Les **expressions statiques** : entourent la fonction BINTOSTR() autour des instances ZONED()

```
BINTOSTR(ZONED(%résultat%; 5); 'A')
```

## Modifier toutes les instances de la commande OPEN FORMAT

Vous devez modifier toutes les instances de la commande OPEN FORMAT. Vous devez utiliser le paramètre SKIP pour ignorer les deux premiers octets du fichier Unicode que vous ouvrez. Cela est nécessaire car les deux premiers octets des fichiers codés UTF-16 sont réservés sous forme de marques d'ordre d'octet et sont distincts du texte dans le fichier.

### Non Unicode

```
OPEN "ascii_test.txt" FORMAT table_modèle CRLF
DEFINE FIELD enreg_complet ASCII 1 10
```

## Unicode

```
OPEN "utf-16_test.txt" FORMAT table_modèle CRLF SKIP 2  
DEFINE FIELD enreg_complet UNICODE 1 20
```

## Vérifier les outils d'analyse convertis

Vérifiez que les versions Unicode des outils d'analyse produisent des résultats identiques aux résultats générés par les outils d'analyse qui ne sont pas au format Unicode. Le meilleur moyen pour le faire consiste à utiliser un outil Diff qui compare les fichiers traces générés pendant l'analyse. L'outil Diff identifie les différences entre les fichiers.

## Que faire si les résultats ne sont pas identiques ?

Si vous ne pouvez pas obtenir les mêmes résultats avec la version Unicode d'un outil d'analyse qu'avec sa version non Unicode, vous devrez peut-être isoler le problème en comparant les sorties de la trace des scripts à chaque étape de l'analyse.

# Vérifier la compatibilité Unicode

Lorsque vous vous mettez à niveau sur une édition Unicode, vous devez vérifier que toute logique personnalisée que vous avez ajoutée aux scripts donnera les mêmes résultats lorsqu'elle sera exécutée avec des données Unicode. Des scripts peuvent être touchés à certains endroits prévisibles lorsqu'ils sont exécutés avec des données Unicode.

## Fonctions de caractères et de bit

Chacune des fonctions indiquées ci-dessous renvoie des valeurs basées sur les emplacements d'octets ou le nombre d'octets. Vous devez vérifier que ces fonctions sont toujours utilisées correctement lorsque vous passez de la représentation des caractères à un octet de l'édition qui n'est pas Unicode au codage à deux octets utilisé pour les données Unicode :

- ASCII()
- BIT()
- BYTE()
- CHR()
- DIGIT()
- HEX()
- MASK()
- SHIFT()

## La longueur d'octet n'équivaut pas à la longueur de caractère

Vous devez vérifier l'utilisation des fonctions suivantes dans vos scripts pour vous assurer qu'elles ne supposent pas de correspondance un-à-un entre le nombre de caractères dans les données et le nombre d'octets.

Si vous trouvez des instances où la logique suppose une correspondance un-à-un entre les caractères et les octets, vous devez adapter la logique en question pour qu'elle fonctionne correctement avec les données Unicode, qui utilisent deux octets pour représenter chaque caractère. Les nombres spécifiés en tant que paramètres de fonctions de chaîne, comme 4 dans `STRING(1000, 4)`, font référence au nombre de caractères. L'utilisation standard de ces fonctions n'entraînera pas de problèmes.



## Fonctions de conversion

- PACKED()
- STRING()
- UNSIGNED()
- VALUE()
- ZONED()

## Fonctions de chaîne

- AT()
- BLANKS()
- INSERT()
- LAST()
- LENGTH()
- REPEAT()
- SUBSTRING()

## Fonctions diverses

- FILESIZE()
- LEADING()
- OFFSET()
- RECLLEN()

# Substituer des fonctions spécifiques à Unicode

Les produits Galvanize Unicode prennent en charge six fonctions spécifiques à Unicode prenant en charge les conversions entre des données Unicode et des données qui ne le sont pas. Les fonctions suivantes sont disponibles dans les produits Galvanize Unicode :

- **BINTOSTR( )** : convertit les données ZONED ou EBCDIC dans sa chaîne Unicode correspondante. Cette opération garantit le bon affichage des valeurs codées en tant que données ZONED ou EBCDIC.
- **DHEX( )** : renvoie l'équivalent hexadécimal d'une valeur de champ Unicode spécifiée. Cette fonction est l'inverse de la fonction HTOU( ).

- **DBYTE()** : renvoie l'interprétation des caractères Unicode de deux octets à la position spécifiée dans un enregistrement.
- **DTOU()** : convertit une valeur de date au bon affichage de la chaîne Unicode basé sur le paramètre régional spécifié.
- **HTOU()** : renvoie l'équivalent Unicode d'une chaîne hexadécimale spécifiée. Cette fonction est l'inverse de la fonction DHEX().
- **UTOD()** : convertit une chaîne Unicode propre aux paramètres régionaux en une valeur de date Analytics.

# Exécution de scripts R sur AX Serveur

Importez les scripts R externes sous forme de fichiers associés avec une App Analyse, puis appelez les scripts R à partir de vos outils d'analyse pour tirer parti des fonctionnalités d'analyse statistiques du langage de script R sur le serveur. Pour préparer l'environnement AX Serveur à exécuter des scripts R, vous devez d'abord installer R, puis ajouter l'extension `.r` à la liste blanche des extensions de fichiers.

## Conditions préalables

Pour exécuter des scripts R sur AX Serveur, vous devez :

1. Installer une version prise en charge du langage de script R sur votre ordinateur AX Serveur.
2. Ajouter l'extension `.r` à la liste blanche des extensions de fichiers sur AX Serveur.
3. Dans Analytics, créer un projet à utiliser et l'importer dans AX Serveur.

### Remarque

Pour vous aider à remplir ces conditions préalables, contactez votre administrateur Analytics Exchange et reportez-vous aux rubriques suivantes :

- [Conditions requises pour AX Serveur](#)
- [Créer une liste blanche des extensions de fichiers](#)

## Ajouter des scripts R au répertoire de projet Analytics

Une fois le projet Analytics créé dans Analytics, copiez les scripts R que vous souhaitez utiliser, puis collez-les dans le dossier du projet afin de pouvoir tester votre script localement dans Analytics avant de l'importer dans Analytics Exchange.

### Exemple de fichiers R

Les exemples de fichiers R qui suivent contiennent des scripts banals qui concatènent deux chaînes et renvoient une seule chaîne jointe par une espace. Ces exemples sont destinés à montrer comment les scripts R s'exécutent sur AX Serveur, et non à analyser les données avec R.

#### `analyse_a.r`

```
conc<-function(x, y) {  
  paste(x, y, sep=" ")  
}  
print(conc(value1, value2))
```

## analyse\_b.r

```
conc<-function(x, y) {
  paste(y, x, sep=" ")
}
print(conc(value1, value2))
```

## Créer un script Analytics

Dans votre projet Analytics, créez un nouveau script à utiliser comme l'outil d'analyse que vous exécutez sur AX Serveur. Ce script fait ce qui suit :

1. Ouvrez une table temporaire nommée `t_tmp` avec un enregistrement.  
Vous devez ouvrir une table pour exécuter la commande `EXTRACT` dans Analytics, ici la table `t_tmp` est utilisée uniquement à cette fin.
2. Utilisez la commande `EXTRACT` pour exécuter chaque script R et écrivez les résultats dans une table.

## Ajouter l'en-tête de l'outil d'analyse

Ajoutez les en-têtes d'outil d'analyse appropriés au début du script pour que le script Analytics puisse s'exécuter sur AX Serveur après l'importation de votre App Analyse. Vous devez ajouter une balise `FILE` pour tout script R que vous prévoyez d'exécuter à partir de l'outil d'analyse :

```
COMMENT
//ANALYTIC intégration R test
  vérifier l'intégration R sur AX Serveur
//DATA t_tmp
//FILE analyse_a.r
//FILE analyse_b.r
//RESULT TABLE résultats
END
```

## Ajouter la logique du script

```
SET SAFETY OFF
DEL ALL OK
CLOSE PRIMARY SECONDARY

OPEN t_tmp
```

```

COM **** exécuter les scripts R et écrire les résultats dans la table
EXTRACT FIELDS RSTRING("a<-source('./analysis_a.r');a[[1]]";50;"test";"valeur") AS "valeur" TO
"results.fil"
EXTRACT FIELDS RSTRING("a<-source('./analysis_b.r');a[[1]]";50;"test";"valeur") AS "valeur" TO
"results.fil" APPEND

CLOSE t_tmp

```

## Le script complet de l'outil d'analyse

Le script complet de l'outil d'analyse que vous exécutez sur AX Serveur ressemble à celui-ci :

```

COMMENT
//ANALYTIC intégration R test
vérifier l'intégration R sur AX Serveur
//DATA t_tmp
//FILE analyse_a.r
//FILE analyse_b.r
//RESULT TABLE résultats
END

SET SAFETY OFF
DEL ALL OK
CLOSE PRIMARY SECONDARY

OPEN t_tmp

COM **** exécuter les scripts R et écrire les résultats dans la table
EXTRACT FIELDS RSTRING("a<-source('./analysis_a.r');a[[1]]";50;"test";"valeur") AS "valeur" TO
"results.fil"
EXTRACT FIELDS RSTRING("a<-source('./analyse_b.r');a[[1]]";50;"test";"valeur") AS "valeur" TO
"results.fil" APPEND

CLOSE t_tmp

```

## Importer le projet Analytics et les fichiers R associés

Une fois que vous avez créé le script de l'outil d'analyse :

1. Dans AX Client, créez une collection et un dossier pour héberger le projet Analytics.
2. Pour importer le projet et les fichiers R :
  - a. Cliquez avec le bouton droit de la souris sur le dossier et sélectionnez **Importer**.
  - b. Accédez à votre projet Analytics sur votre ordinateur local, puis sélectionnez le fichier de projet `.acl` et les scripts R `.r`.

#### Remarque

Assurez-vous que vous sélectionnez les fichiers R dans le dossier de projet ainsi que dans le projet Analytics à l'aide de la commande **Ctrl+clic** pour qu'ils soient importés dans AX Serveur. Vous devez également importer les fichiers de données sources de la table `t_tmp`.

- c. Cliquez sur **Ouvrir**.

## Explorateur de serveur après l'importation

- *NomCollection*
  - *NomDossier*
    - **Apps Analyse**
      - *NomProjetACL*
      - *NomScriptOutildanalyse*
    - **Données**
      - `t_tmp`
    - **Fichiers associés**
      - `analyse_a.r`
      - `analyse_b.r`

## Exécuter l'outil d'analyse

Dans l'**Explorateur de serveur** d'AX Client, cliquez avec le bouton droit de la souris sur l'outil d'analyse, puis sélectionnez **Exécuter**. Les scripts R sont exécutés dans le cadre de l'outil d'analyse et vous pouvez accéder aux résultats à la table de résultats à partir du Client Web d'AX.

### Résultats

## Explorateur de serveur après l'exécution de l'outil d'analyse

- *NomCollection*
  - *NomDossier*
    - **Apps Analyse**
      - *NomProjetACL*
      - *NomScriptOutildanalyse*
    - **Données**
      - résultats

- **Fichiers associés**
  - analyse\_a.r
  - analyse\_b.r

## **Table de résultats**

- **valeur**
- tester la valeur
- test de valeur

# Exécution de scripts Python sur AX Serveur

Demandez à un administrateur Analytics Exchange de charger des scripts Python externes dans le répertoire PYTHONPATH d'AX Serveur, puis appelez les scripts à partir de vos outils d'analyse pour exploiter les fonctionnalités orientées objet du langage de programmation Python sur le serveur. Pour préparer l'environnement AX Serveur à exécuter des scripts Python, vous devez d'abord installer Python, puis définir la variable d'environnement PYTHONPATH.

## Conditions préalables

Pour exécuter des scripts Python sur AX Serveur, vous devez :

1. Installer une version prise en charge du langage de script Python sur votre ordinateur AX Serveur.
2. Définir la variable d'environnement PYTHONPATH sur AX Serveur.
3. Dans Analytics, créer un projet à utiliser et l'importer dans AX Serveur.

### Remarque

Pour vous aider à remplir ces conditions préalables, contactez votre administrateur Analytics Exchange et reportez-vous aux rubriques suivantes :

- [Conditions requises pour AX Serveur](#)
- [Configuration de Python pour être utilisé avec AX Serveur](#)

## Créer un script Python

Une fois votre projet Analytics créé dans Analytics, créez un script Python que vous pouvez appeler à partir d'un outil d'analyse.

Ensuite, donnez à votre administrateur Analytics Exchange ce fichier de script à charger dans le répertoire PYTHONPATH du système hébergeant AX Serveur avant que vous appeliez le script à partir d'un outil d'analyse. Lorsque l'outil d'analyse s'exécute sur AX Serveur, l'exécutable Python recherche le script dans le répertoire PYTHONPATH, il doit donc être présent.

### Exemple de fichier Python

L'exemple de fichier Python qui suit contient un script banal utilisant une expression lambda permettant d'élever un nombre à la puissance de lui-même. Cet exemple est destiné à montrer comment les scripts Python s'exécutent sur AX Serveur, et non à analyser les données avec Python.



## Nom du fichier : lambda\_example.py

```
# myFunc au carré de la valeur1 et retourne la valeur
myFunc = lambda valeur1: valeur1**2
```

## Créer un script Analytics

Dans votre projet Analytics, créez un nouveau script à utiliser comme l'outil d'analyse que vous exécutez sur AX Serveur. Ce script fait ce qui suit :

1. Il ouvre une table simple nommée py avec un enregistrement.  
Vous devez ouvrir une table pour exécuter la commande GROUP dans Analytics, ici la table py est utilisée uniquement à cette fin.
2. Il s'exécute en boucle 10 fois et à chaque boucle, il exécute le script Python en transmettant le compteur d'incrémentation sous forme d'argument et en extrayant la sortie vers une table de résultats.

## Ajouter l'en-tête de l'outil d'analyse

Ajoutez les en-têtes d'outil d'analyse appropriés au début du script pour que le script Analytics puisse s'exécuter sur AX Serveur après l'importation de votre App Analyse :

```
COMMENT
//ANALYTIC Intégration Python Test
Vérifier l'intégration Python sur AX Serveur
//DATA py
//DATA résultats
//RESULT TABLE résultats
END
```

## Ajouter la logique du script

```
SET SAFETY OFF
DEL ALL OK
CLOSE

OPEN py

GROUP
ASSIGN v_max = 11
ASSIGN v_counter = 1
LOOP WHILE v_counter < v_max
```

```

EXTRACT PYNUMERIC("lambda_example,myFunc",0,v_counter) AS "Valeur Résultats" TO
"results.fil"
  v_counter = v_counter + 1
END
END
CLOSE py

```

## Le script complet de l'outil d'analyse

Le script complet de l'outil d'analyse que vous exécutez sur AX Serveur ressemble à celui-ci :

```

COMMENT
//ANALYTIC Intégration Python Test
Vérifier l'intégration Python sur AX Serveur
//DATA py
//DATA résultats
//RESULT TABLE résultats
END

SET SAFETY OFF
DEL ALL OK
CLOSE

OPEN py

GROUP
  ASSIGN v_max = 11
  ASSIGN v_counter = 1
  LOOP WHILE v_counter < v_max
    EXTRACT PYNUMERIC("lambda_example,myFunc",0,v_counter) AS "Valeur Résultats" TO
    "results.fil"
    v_counter = v_counter + 1
  END
END
CLOSE py

```

## Importer le projet Analytics

Une fois que vous avez créé le script de l'outil d'analyse :

1. Dans AX Client, créez une collection et un dossier pour héberger le projet Analytics.
2. Pour importer le projet :
  - a. Cliquez avec le bouton droit de la souris sur le dossier et sélectionnez **Importer**.
  - b. Accédez à votre projet Analytics sur votre ordinateur local, sélectionnez le fichier de projet `.acl`, puis cliquez sur **Ouvrir**.

### Remarque

Assurez-vous d'importer les fichiers de données sources pour importer la table `py` avec votre projet Analytics.

## Explorateur de serveur après l'importation

- *NomCollection*
- *NomDossier*
  - **Apps Analyse**
    - *NomProjetACL*
    - *NomScriptOutildanalyse*
  - **Données**
    - `py`
  - **Fichiers associés**

## Exécuter l'outil d'analyse

Dans l'**Explorateur de serveur** d'AX Client, cliquez avec le bouton droit de la souris sur l'outil d'analyse, puis sélectionnez **Exécuter**. Le script Python est exécuté dans le cadre de l'outil d'analyse et vous pouvez accéder aux résultats à la table de résultats à partir du Client Web d'AX.

### Remarque

Lorsque le script s'exécute, l'exécutable Python recherche le fichier du script dans le répertoire PYTHONPATH du serveur hébergeant AX Serveur. Si votre administrateur Analytics Exchange n'a pas chargé le fichier dans ce répertoire, l'outil d'analyse échoue.

## Résultats

### Explorateur de serveur après l'exécution de l'outil d'analyse

- *NomCollection*
- *NomDossier*
  - **Apps Analyse**
    - *NomProjetACL*
    - *NomScriptOutildanalyse*
  - **Données**
    - `py`
    - résultats
  - **Fichiers associés**

## Table de résultats

- Valeur des résultats
- 1
- 4
- 9
- 16
- 25
- 36
- 49
- 64
- 81
- 100

# Codes d'erreur du moteur d'outil d'analyse

Le tableau suivant répertorie les codes d'erreur que vous êtes susceptible de rencontrer lors de l'exécution d'outils d'analyse.

## Erreurs de démarrage du moteur d'outil d'analyse

Code d'erreur	Description
202	Erreur système.
203	La période d'évaluation est arrivée à expiration.
204	La période d'évaluation est arrivée à expiration.
205	Échec de l'activation.
206	Nombre maximum de sessions atteintes.
207	Problème(s) d'initialisation de mémoire.
209	Erreur de script inconnue.
210	Mot de passe du profil de base de données manquant.
211	La connexion au serveur a échoué.
212	Une commande non prise en charge a été détectée.
213	Une boîte de dialogue a été générée par le script.
256	Le moteur AX n'a pas réussi à démarrer.

# Codes d'erreur du moteur d'outil d'analyse

Code d'erreur	Description
1000	Aucun fichier de préférence n'a été spécifié. Un nouveau fichier de préférence par défaut a été créé.
1001	Un problème est survenu avec le fichier de préférence. Un nouveau fichier de préférence par défaut a été créé.
1002	Le projet a été mis à niveau depuis une version antérieure. Une copie a été enregistrée avec une extension .old.
1003	Le fichier du projet n'a pas pu être traité. Le dernier projet enregistré a été utilisé.
1004	Aucun fichier de projet spécifié.
1005	Le fichier de projet spécifié n'existe pas.
1006	Le fichier de projet spécifié est en lecture seule.
1007	Le projet spécifié est actuellement utilisé par une autre application.
1008	Le fichier de projet .old spécifié ne peut pas être utilisé. Vous devez spécifier un fichier de projet portant l'extension ACL.
1009	Le fichier de projet spécifié n'est pas un fichier de projet Analytics.
1011	Le fichier de projet spécifié ne peut pas être enregistré sous une version antérieure.
1012	Impossible d'ouvrir le fichier trace pour écriture.
1013	Aucun script n'a été spécifié.
1014	Le script spécifié n'existe pas.
1015	La licence Analytics est introuvable ou n'est pas valide.
1016	Un fichier bibliothèque requis (.dll) est introuvable.
1017	Une erreur inconnue est survenue.

## Erreurs de commande

Le tableau suivant répertorie les codes d'erreur renvoyés lors de l'échec d'un outil d'analyse en raison d'une commande ACLScript incorrecte. Le numéro de code d'erreur renvoyé identifie la commande ayant échoué.

Code d'erreur	Commande
1	SAMPLE
2	EXTRACT
3	LIST
4	TOTAL
5	DEFINE
6	COMMENT
7	QUIT
8	STOP
9	BYE
10	USE
11	OPEN
12	SAVE
13	DISPLAY
14	ACTIVATE
15	CLOSE
16	HELP
17	COUNT
18	STATISTICS
19	HISTOGRAM
20	STRATIFY
21	SUMMARIZE
22	EXPLAIN
23	GROUP

Code d'erreur	Commande
24	ELSE
25	END
26	CANCEL
27	SUBMIT
28	DELETE
29	RANDOM
30	SORT
31	FIND
32	DIRECTORY
33	TYPE
34	DUMP
35	INDEX
37	SET
40	DO
41	TOP
42	EXPORT
43	VERIFY
44	SEEK
45	JOIN
46	MERGE
47	SEQUENCE
48	CALCULATE
49	PRINT



Code d'erreur	Commande
50	LOCATE
51	RENAME
54	COPY
55	REPORT
56	EJECT
58	LET
59	ACCUMULATE
63	ACCEPT
64	ASSIGN
65	AGE
66	CLASSIFY
67	PROFILE
68	DO REPORT
69	LOOP
70	PAUSE
71	SIZE
72	EVALUATE
73	DIALOG
74	IF
75	GAPS
76	DUPS
77	SQLOPEN
78	PASSWORD

Code d'erreur	Commande
79	IMPORT
80	REFRESH
81	NOTIFY
82	CONNECT
83	RETRIEVE
84	FIELDSHIFT
85	BENFORD
86	CROSSTAB
87	(non utilisé)
88	ESCAPE
89	NOTES
90	FUZZY DUPLICATE
91	EXECUTE

## Erreurs de traitement des tâches analytiques

Code d'erreur	Message d'erreur
-10	Les résultats d'analyse n'ont pas pu être enregistrés car le dossier de résultats de destination a été supprimé après le lancement de l'outil d'analyse.
-11	La tâche a été interrompue.
-12	Arrêté en raison de la fermeture du serveur.
-13	Échec lors de la création des résultats.
-16	Exécution impossible en raison d'une erreur de configuration des propriétés du serveur.
-17	Impossible de créer un répertoire de résultats au nom unique.

Code d'erreur	Message d'erreur
-19	La tâche a été ignorée.
-20	Impossible de préparer les tables des résultats publiés.
-21	Impossible de publier les résultats sur AXException.
-22	Échec de la publication. Nom de table non valide.
-23	Échec de la publication. Un ou plusieurs noms de colonnes de la table sont trop longs.
-24	Échec de la publication. Données incorrectes détectées dans les cellules de données à l'intérieur d'une table Analytics.
-25	Échec de la publication. Types de données contenues dans des champs de la table non pris en charge.
-26	Échec de la publication. Impossible de se connecter au serveur AXException.
-27	Non-exécution de la tâche. L'utilisateur a été supprimé ou ne dispose pas de l'autorisation nécessaire.
-28	Non-exécution de la tâche. Erreur inattendue. Voir les détails dans la trace du serveur et dans la trace Analytics.
-29	Impossible de copier les fichiers de données. L'outil d'analyse a échoué car les fichiers de données requis n'ont pas pu être copiés dans le répertoire des tâches.
-30	Non-exécution de la tâche. Le lien de l'outil d'analyse est rompu.
-31	Échec de la publication. Le fichier de mappage d'exception n'a pas pu être localisé.
-32	Échec de la publication. Échec d'analyse du fichier de mappage des exceptions.
-34	Échec du stockage des résultats de la tâche. Vérifiez que l'espace du lecteur qui stocke le dossier des tâches soit suffisant et qu'aucun fichier de données ne soit verrouillé.

# Variables créées via des commandes Analytics

Lorsque certaines commandes sont exécutées, via la saisie d'informations dans les boîtes de dialogue Analytics ou via l'exécution de scripts, des variables système sont automatiquement créées par Analytics. Vous pouvez utiliser ces variables, et les valeurs qu'elles contiennent, lors de l'exécution des commandes Analytics subséquentes.

La valeur d'une variable système est remplacée par une valeur mise à jour si vous exécutez à nouveau la même commande.


"Variables système Analytics" Page opposée répertorie les variables système créées par Analytics.

## Remarque

Les variables système, et les valeurs qu'elles contiennent, sont conservées uniquement pour la durée de la session Analytics active.

## Affichage de la valeur actuelle des variables

Vous pouvez afficher les valeurs actuelles de toutes les variables définies par le système et par l'utilisateur dans un projet Analytics à l'aide de l'une des méthodes suivantes :

- Sélectionner l'onglet **Variables** dans le **Navigateur**
- Saisir `DISPLAY VARIABLES` dans la ligne de commande
- Cliquer sur **Afficher les variables**  dans la barre d'outils (vous devez ajouter ce bouton au préalable)

Les deuxième et troisième méthodes affichent également la mémoire restante disponible pour stocker les variables.

## Variables système numérotées de façon incrémentielle

Pour les variables système incluant  $n$  dans "Variables système Analytics" Page opposée,  $n$  est toujours 1 si les commandes sont exécutées en dehors d'un groupe, par exemple, `TOTAL1`.

Si vous utilisez un groupe pour exécuter plusieurs commandes, toutes les variables système qui en résultent sont numérotées en fonction du numéro de ligne de la commande créant la variable. La première commande d'un groupe est considérée comme étant la ligne numéro 2.

Par exemple :

- si la commande Total est la troisième commande d'un groupe, les résultats sont contenus dans la variable **TOTAL4**.
- Si la seconde commande Total est la cinquième commande du groupe, les résultats sont contenus dans la variable **TOTAL6**.

## Variables système Analytics

La table suivante répertorie les variables système créées par Analytics, les commandes par lesquelles elles sont générées et les valeurs qu'elles contiennent.

### Remarque

Si vous exécutez la commande **Statistics** sur plusieurs champs simultanément, les variables système contiennent des valeurs pour le premier champ que vous définissez.

Variable système	Commande	Valeur
<b>WRITE<math>n</math></b>	<ul style="list-style-type: none"> <li>○ Toute commande qui extrait une table</li> <li>○ Vérifier</li> <li>○ Séquence</li> </ul>	<ul style="list-style-type: none"> <li>○ Le nombre d'enregistrements de la table de sortie</li> <li>○ Le nombre d'erreurs de validité des données (Vérifier)</li> <li>○ Le nombre d'erreurs de séquence (Séquence)</li> </ul>
<b>OUTPUTFOLDER</b>	<ul style="list-style-type: none"> <li>○ Toute commande qui extrait une table Analytics</li> </ul>	<p>Le chemin d'accès au dossier de projet Analytics dans le <b>Navigateur</b> contenant la table de sortie.</p> <p>Le chemin de type DOS indique le format <b>/fol-demame/subfoldemame</b>, où la première barre oblique (/) indique le niveau racine de l'onglet <b>Vue globale</b>.</p> <p><b>Astuce</b></p> <p>Utilisez la commande <b>SET FOLDER</b> pour spécifier un autre répertoire de sortie ou pour créer un nouveau répertoire de sortie.</p>
<b>COUNT<math>n</math></b>	<ul style="list-style-type: none"> <li>○ Compte</li> <li>○ Statistiques</li> </ul>	Le nombre d'enregistrements recensés
<b>ACL_Ver_Major</b>	<ul style="list-style-type: none"> <li>○ Afficher la version (les numéros de version d'Analytics se présentent sous la forme <i>principale.seconde.patch</i>)</li> </ul>	La version principale d'Analytics fonctionnant actuellement.
<b>ACL_Ver_Minor</b>		La version secondaire d'Analytics fonctionnant actuellement.
<b>ACL_Ver_Patch</b>		La version patch d'Analytics fonctionnant actuellement.
<b>ACL_Ver_Type</b>		<p>L'édition d'Analytics fonctionnant actuellement :</p> <ul style="list-style-type: none"> <li>○ La valeur '0' correspond à l'édition non-Unicode</li> <li>○ La valeur '1' correspond à l'édition Unicode</li> </ul>
<b>MLE<math>n</math></b>	<ul style="list-style-type: none"> <li>○ Evaluer</li> </ul>	<p><b>Échantillonnage par unité monétaire</b></p> <p>Nombre d'erreurs le plus probable (inexactitude estimée)</p> <p><b>Échantillonnage par enregistrement</b></p>

Variable système	Commande	Valeur
		Taux de fréquence limite des erreurs supérieure (taux d'écart supérieur calculé)
UEL $n$		<p><b>Échantillonnage par unité monétaire</b></p> <p>Nombre limite des erreurs supérieure (limite d'inexactitude supérieure)</p>
RETURN_CODE	<ul style="list-style-type: none"> <li>Exécuter</li> </ul>	<p>Code renvoyé par une application externe ou une exécution de processus à l'aide de la commande Exécuter.</p> <p>Les codes de retour sont des nombres générés par l'application ou le processus externe et ils sont renvoyés à Analytics pour indiquer l'issue du processus externe. Analytics ne génère pas le code de retour.</p> <p>Les codes de retour types sont des valeurs entières qui mappent vers des notifications ou des messages d'erreurs particuliers. Par exemple, le code de retour « 0 » pourrait signifier « L'opération a été effectuée avec succès ». Le code de retour « 2 » pourrait signifier « Le système ne peut pas trouver le fichier spécifié ».</p> <p>Les codes de retour spécifiques et leurs significations varient en fonction de l'application ou du processus externe. Les listes des codes de retour, aussi appelés « codes d'erreur » ou « codes de sortie », et leurs significations, figurent souvent dans la documentation de l'application externe associée. Vous trouverez également les listes des codes de retour sur Internet.</p> <p>La variable RETURN_CODE est créée lorsque la commande EXECUTE est utilisée de façon synchrone, mais pas lorsque la commande est utilisée de façon asynchrone.</p>
GAPDUP $n$	<ul style="list-style-type: none"> <li>Omissions</li> <li>Doublons</li> <li>Doublons approximatifs</li> </ul>	Le nombre total d'omissions, de doublons, ou de groupes de doublons approximatifs.
SAMPINT $n$	<ul style="list-style-type: none"> <li>Taille</li> </ul>	L'intervalle de l'échantillon requis.
SAMPSIZE $n$		La taille de l'échantillon requis.
ABS $n$	<ul style="list-style-type: none"> <li>Statistiques</li> </ul>	La valeur absolue du premier champ défini.
AVERAGE $n$		La valeur moyenne du premier champ défini.
HIGH $n$		<p>La 5e valeur la plus élevée du premier champ défini.</p> <p>La 5e valeur la plus élevée est le paramètre par défaut. Le paramètre peut être modifié à l'aide de l'option <b>Nombre de valeurs extrêmes</b> dans la boîte de dialogue <b>Statistiques</b>.</p>

Variable système	Commande	Valeur
		<p><b>Remarque</b></p> <p>Lorsqu'Analytics identifie la valeur la plus élevée, les valeurs dupliquées ne sont pas exclues. Par exemple, si dans l'ordre décroissant les valeurs sont 100, 100, 99, 98, la 3e valeur la plus élevée est 99 et non 98.</p>
		<p>La 5e valeur la plus basse du premier champ défini.</p> <p>La 5e valeur la plus basse est le paramètre par défaut. Le paramètre peut être modifié à l'aide de l'option <b>Nombre de valeurs extrêmes</b> dans la boîte de dialogue <b>Statistiques</b>.</p> <p><b>Remarque</b></p> <p>Lorsqu'Analytics identifie la valeur la plus basse, les valeurs dupliquées ne sont pas exclues. Par exemple, si dans l'ordre croissant les valeurs sont 1, 1, 2, 3, la 3e valeur la plus basse est 2 et non 3.</p>
LOW $n$		
MAX $n$		La valeur maximale du premier champ défini.
MEDIAN $n$		La valeur médiane du premier champ défini.
MIN $n$		La valeur minimale du premier champ défini.
MODE $n$		La valeur s'affichant le plus souvent dans le premier champ défini.
Q25 $n$		La première valeur quartile (valeur quartile la plus basse) du premier champ défini.
Q75 $n$		La troisième valeur quartile (valeur quartile la plus élevée) du premier champ défini.
RANGE $n$		La différence entre les valeurs maximale et minimale du premier champ défini.
STDDEV $n$		L'écart standard du premier champ défini.
TOTAL $n$	<ul style="list-style-type: none"> <li>○ Total</li> <li>○ Statistiques</li> </ul>	Le total des valeurs du premier champ défini.

## Autres variables système

Les variables suivantes sont générées par le système, mais ne sont pas créées par des commandes :

- **AXRunByUser** : disponible dans les scripts s'exécutant sur AX Serveur, cette variable stocke le nom d'utilisateur de l'utilisateur exécutant l'outil d'analyse au format « *domaine\nom\_utilisateur* ».

- **OUTPUTFOLDER** : dossier de sortie du projet Analytics actuel



# Mots-clés réservés

Analytics réserve certains mots-clés dans un but particulier. Il est impossible de nommer des champs ou des variables avec les valeurs portées par ces mots-clés réservés.

Si vous ajoutez un suffixe à un mot-clé réservé, vous pouvez l'utiliser en tant que nom de champ ou nom de variable. Par exemple, le nom « Field » n'est pas autorisé, mais « Field\_1 » ou « Field\_2 » l'est.

## Remarque

Dans certains cas, il vous est également empêché d'utiliser des abréviations de mots-clés réservés, comme « Can » (CANCEL), « Form » (FORMAT) ou « Rec » (RECORD).

Mots-clés réservés	Objet dans Analytics
ALL	Fait référence à l'ensemble des champs définis précédemment.
AND	Opérateur logique AND.
AS	Affecte un nom d'affichage au champ ou à l'expression de sortie.
AXRunByUser	Variable système stockant le nom d'utilisateur de l'utilisateur exécutant un script d'outil d'analyse sur AX Serveur au format « <i>domaine\nom_utilisateur</i> ».
CANCEL	Annule la commande actuelle.
D	Spécifie un tri dans l'ordre décroissant pour le nom de l'expression ou de champ précédent.
END	Conclut le flux d'entrée et se comporte comme une ligne nulle.
EXPR	Préfixe d'un nom d'un champ de sortie par défaut.
F	Fait référence à la valeur <i>false</i> (faux) d'une expression logique.
FIELD/FIELDS	Partie des commandes EXPORT, EXTRACT, JOIN et SAMPLE.
FORMAT	Ancien nom pour format de table Analytics. Impossible à utiliser en tant que nom de table Analytics.
IF	Spécifie une condition.
LINE	Utilisée par la commande DEFINE COLUMN pour indiquer si un champ se divise en un certain nombre de lignes spécifié.
NODUPS	Supprime les valeurs d'affichage dupliquées dans le champ de fractionnement d'un rapport Analytics.
NOT	Opérateur logique NOT.

Mots-clés réservés	Objet dans Analytics
NOZEROS	Affiche ou imprime sous forme de cellules vides des valeurs zéro dans un champ numérique ou dans un rapport.
ON	Précède une liste de champs.
OR	Opérateur logique OR.
OTHER	Indique les champs ou les expressions à inclure, mais pas à sous-totaliser, dans la sortie de la commande SUMMARIZE.
PAGE	Utilisé par la commande REPORT pour créer des sauts de page.
PICTURE	Indique le format d'un champ numérique.
PRIMARY	Spécifie un certain type de jointure.
RECORD	Fait référence à l'intégralité de l'enregistrement d'entrée tel qu'il existe.
RECORD_LENGTH	Stocke les valeurs de longueurs d'enregistrement à utiliser dans les opérations de traitement d'enregistrements.
SECONDARY	Spécifie un certain type de jointure.
SUPPRESS	Supprime la sortie des totaux numériques.
T	Fait référence à la valeur <i>true</i> (vrai) d'une expression logique.
TAPE	Fait référence à une ancienne méthode d'accès aux données avec Analytics. Impossible à utiliser en tant que nom de table Analytics.
TO	Désigne un fichier de sortie pour n'importe quelle commande.
WIDTH	Modifie la largeur d'impression par défaut d'un champ ou d'une expression spécifié.