

# Macros for - Guide d'utilisation



# Table des matières

<b>Chapitre 1. Utilisation de macros dans Unica Campaign.....</b>	<b>1</b>
Récapitulatifs des fonctions macro pour Unica Campaign.....	1
Fonctions statistiques.....	1
fonctions mathématiques et trigonométriques.....	2
fonctions des chaînes de caractères.....	7
Fonctions d'horodatage.....	9
Fonctions de groupement.....	10
fonctions diverses.....	10
Paramètres des fonctions macro pour Unica Campaign.....	11
spécifications de format.....	11
utilisation des constantes.....	12
<b>Chapitre 2. Utilisation de macros dans Unica Interact.....</b>	<b>14</b>
Récapitulatifs des fonctions macro pour Unica Interact.....	14
Fonctions statistiques.....	14
fonctions mathématiques et trigonométriques.....	15
fonctions des chaînes de caractères.....	16
Fonctions d'horodatage.....	17
fonctions diverses.....	18
Paramètres des fonctions macro pour Unica Interact.....	18
spécifications de format.....	18
utilisation des constantes.....	20
<b>Chapitre 3. référence des macros.....</b>	<b>21</b>
Mots clés de formats de date valides.....	21

Macro ABS.....	24
Macro ACOS.....	25
Macro ACOT.....	27
Macro ADD_MONTHS.....	30
Macro AND.....	32
Macro ASIN.....	34
Macro ATAN.....	37
Macro AVG.....	39
Macro BETWEEN.....	42
Macro BIT_AND.....	42
Macro BIT_NOT.....	45
Macro BIT_OR.....	46
Macro BIT_XOR.....	48
Macro CEILING.....	51
Macro COLUMN.....	52
Macro COS.....	54
Macro COSH.....	56
Macro COT.....	58
Macro COUNT.....	60
Macro CURRENT_DATE.....	62
Macro CURRENT_DAY.....	63
Macro CURRENT_JULIAN.....	64
Macro CURRENT_MONTH.....	65
Macro CURRENT_TIME.....	66
configuration des dates sur votre application Web.....	66

Macro CURRENT_WEEKDAY.....	68
Macro CURRENT_YEAR.....	68
DATE.....	69
Macro DATE_FORMAT.....	72
Macro DATE_JULIAN.....	73
Macro DATE_STRING.....	74
Macro DAY_BETWEEN.....	76
Macro DAY_FROMNOW.....	77
Macro DAY_INTERVAL.....	78
Macro DAYOF.....	79
Macro DISTANCE.....	80
Macro DIV.....	82
Macro EQ.....	84
Macro EXP.....	86
Macro EXTERNALCALLOUT.....	88
Macro FACTORIAL.....	90
Macro FLOOR.....	92
Macro FORMAT.....	93
Macro FRACTION.....	97
Macro GE.....	99
Macro GET.....	101
Macro GROUPBY.....	102
Macro GROUPBY_WHERE.....	105
Macro GT.....	106
Macro IF.....	108

Macro IN.....	110
Macro INT.....	112
Macro INVERSE.....	113
Macro IS.....	115
Macro ISERROR.....	116
Macro ISODD.....	117
Macro ISEVEN.....	118
Macro ISODD.....	120
Macro LE.....	122
Macro LIKE.....	124
Macro LN ou LOG.....	126
Macro LOG2.....	128
Macro LOG10.....	130
Macro LOWER.....	131
Macro LT.....	132
Macro LTRIM.....	134
Macro MAX.....	135
Macro MEAN.....	137
Macro MIN.....	140
Macro MINUS.....	142
Macro MOD.....	144
Macro MONTHOF.....	147
Macro MULT.....	148
Macro NE.....	150
Macro NOT.....	152

Macro NUMBER.....	154
Macro OR.....	164
Macro POSITION.....	166
Macro PLUS.....	168
Macro POWER.....	170
Macro RANDOM.....	172
Macro RANDOM_GAUSS.....	174
Macro ROUND.....	176
Macro ROWNUM.....	177
Macro RTRIM.....	178
Macro SIGN.....	178
Macro SIN.....	180
Macro SINH.....	181
Macro COUNT_DIM.....	183
Macro SORT.....	184
Macro SQRT.....	185
Macro STDV ou STDEV.....	186
Macro STRING_CONCAT.....	189
Macro STRING_HEAD.....	191
Macro STRING_LENGTH.....	192
Macro STRING_PROPER.....	194
Macro STRING_SEG.....	195
Macro STRING_TAIL.....	197
Macro SUBSTR ou SUBSTRING.....	199
Macro SUM.....	200

Macro TAN.....	202
Macro TANH.....	204
Macro TOTAL.....	206
Macro TRUNCATE.....	208
Macro UPPER.....	210
Macro VARIANCE.....	211
Macro WEEKDAY.....	213
Macro WEEKDAYOF.....	215
Macro XOR.....	216
Macro YEAROF.....	218
<b>Chapitre 4. Index.....</b>	<b>a</b>

# Chapitre 1. Utilisation de macros dans Unica Campaign

Ce chapitre fournit des informations d'utilisation sur les macros contenues dans Unica Campaign. Veuillez lire ce chapitre avant d'utiliser des macros dans Unica Campaign.

## Récapitulatifs des fonctions macro pour Unica Campaign

Les tableaux de cette section récapitulent les fonctions macro par catégorie.

Des pages de référence détaillées sont fournies pour chaque macro, par ordre alphabétique, dans la rubrique [référence des macros \(à la page 21\)](#).

**⚠ Important :** Les macros peuvent s'appliquer à la fois à Unica Campaign Unica Interact et à , ou à un seul de ces produits. Les descriptions des macros identifient les produits dans lesquels elles sont disponibles.

Pour des informations sur les paramètres d'entrée de la fonction de macro, voir [Paramètres des fonctions macro pour Unica Campaign \(à la page 11\)](#).

## Fonctions statistiques

Nom de la macro	Renvoie	Description
AVG	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la moyenne arithmétique d'une plage de cibles
COUNT	Valeur unique dans une nouvelle colonne.	Compte le nombre de valeurs d'une plage de données spécifiée



Nom de la macro	Renvoi	Description
MAX	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la valeur maximale d'une plage de cibles
MEAN	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la moyenne arithmétique d'une plage de cibles
MIN	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la valeur minimale d'une plage de cibles
STDV ou STDEV	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule l'écart type d'une plage de cibles
VARIANCE	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la variance d'une plage de cibles

## fonctions mathématiques et trigonométriques

Nom de la macro	Renvoi	Description
ABS	Une colonne par colonne d'entrée	Calcule la valeur absolue du contenu de la plage de données spécifiée
ACOS	Une colonne par colonne d'entrée	Calcule le cosinus inverse du contenu de la plage de données spécifiée

<b>Nom de la macro</b>	<b>Renvoi</b>	<b>Description</b>
ACOT	Une colonne par colonne d'entrée	Calcule l'arc cotangente du contenu de la plage de données spécifiée
ASIN	Une colonne par colonne d'entrée	Calcule l'arc sinus du contenu de la plage de données spécifiée
ATAN	Une colonne par colonne d'entrée	Calcule l'arc tangente du contenu de la plage de données spécifiée
AVG	Une colonne par colonne d'entrée	Calcule la moyenne arithmétique des cibles de la plage de données spécifiée
BETWEEN	Une colonne par colonne d'entrée	Compare deux valeurs pour déterminer si la valeur fournie est comprise entre deux autres valeurs
CEILING	Une colonne par colonne d'entrée	Calcule le plafond de chaque valeur de la plage de données spécifiée
COLUMN	Une colonne par colonne d'entrée	Crée des colonnes, en concaténant verticalement les valeurs en entrée dans chaque colonne
COS	Une colonne par colonne d'entrée	Calcule le cosinus du contenu de la plage de données spécifiée
COSH	Une colonne par colonne d'entrée	Calcule le cosinus hyperbolique du contenu

<b>Nom de la macro</b>	<b>Renvoie</b>	<b>Description</b>
COT	Une colonne par colonne d'entrée	de la plage de données spécifiée Calcule la cotangente du contenu de la plage de données spécifiée
COUNT	Une colonne contenant une valeur unique	Compte le nombre de cibles contenant des valeurs dans la plage de données spécifiée
EXP	Une colonne par colonne d'entrée	Calcule le nombre naturel (e) élevé à la puissance du contenu de chaque cible de la plage de données spécifiée
FACTORIAL	Une colonne par colonne d'entrée	Calcule la factorielle de chaque valeur de la plage de données spécifiée
FLOOR	Une colonne par colonne d'entrée	Calcule la partie entière de chaque valeur de la plage de données spécifiée
FRACTION	Une colonne par colonne d'entrée	Renvoie la partie décimale de chaque valeur de la plage de données spécifiée
INT	Une colonne par colonne d'entrée	Calcule la valeur entière (arrondie) du contenu de la plage de données spécifiée
INVERSE	Une colonne par colonne d'entrée	Calcule la valeur négative du contenu de la plage de données spécifiée

Nom de la macro	Renvoi	Description
LN	Une colonne par colonne d'entrée	Calcule le logarithme naturel du contenu de la plage de données spécifiée
LOG	Une colonne par colonne d'entrée	Calcule le logarithme naturel du contenu de la plage de données spécifiée
LOG2	Une colonne par colonne d'entrée	Calcule le logarithme base 2 du contenu de la plage de données spécifiée
LOG10	Une colonne par colonne d'entrée	Calcule le logarithme base 10 du contenu de la plage de données spécifiée
MAX	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la valeur maximale d'une plage de cibles
MEAN	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la moyenne arithmétique d'une plage de cibles
MIN	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la valeur minimale d'une plage de cibles
RANDOM	Une colonne avec le nombre de valeurs spécifié	Renvoie le nombre spécifié de nombres aléatoires

Nom de la macro	Renvoi	Description
RANDOM_GAUSS	Une colonne avec le nombre de valeurs spécifié	Renvoie le nombre spécifié de valeurs aléatoires à partir d'une distribution de Gauss
ROUND	Une colonne par colonne d'entrée	Calcule la valeur arrondie du contenu de la plage de données spécifiée
SIGN	Une colonne par colonne d'entrée	Détermine le signe (positif ou négatif) des valeurs dans la plage de données spécifiée
SIN	Une colonne par colonne d'entrée	Calcule le sinus du contenu de la plage de données spécifiée
SINH	Une colonne par colonne d'entrée	Calcule le sinus hyperbolique du contenu de la plage de données spécifiée
SQRT	Une colonne par colonne d'entrée	Calcule la racine carrée du contenu de la plage de données spécifiée
STDV OU STDEV	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule l'écart type d'une plage de cibles
SUM	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une	Calcule la somme d'une plage de cibles

Nom de la macro	Renvoie	Description
	colonne avec une valeur par ligne pour le mot clé ROW.	
TAN	Une colonne par colonne d'entrée	Calcule la tangente du contenu de la plage de données spécifiée
TANH	Une colonne par colonne d'entrée	Calcule la tangente hyperbolique du contenu de la plage de données spécifiée
TOTAL	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la somme d'une plage de cibles
TRUNCATE	Une colonne par colonne d'entrée	Renvoie la partie non décimale de chaque valeur de la plage de données spécifiée
VARIANCE	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la variance d'une plage de cibles

## fonctions des chaînes de caractères

Nom de la macro	Renvoie	Description
FORMAT	Une colonne par colonne d'entrée	Fournit un contrôle de formatage en sortie pour les nombres et les chaînes (tel que la largeur de la sortie, l'alignement, la précision numérique, le symbole du

Nom de la macro	Renvoi	Description
		séparateur décimal, le symbole de groupement, etc.). Renvoie la chaîne en sortie formatée.
LIKE	Une colonne par colonne d'entrée	Détermine si une chaîne de texte correspond à un modèle spécifié
LOWER	Une colonne par colonne d'entrée	Convertit une valeur de chaîne en minuscules
LTRIM	Une colonne par colonne d'entrée	Supprime les espaces de début de chaque valeur de chaîne
NUMBER	Une colonne par colonne d'entrée	Convertit les chaînes de texte ASCII des heures et des dates en valeurs numériques
POSITION	Une colonne par colonne d'entrée	Renvoie la position initiale d'un modèle dans une chaîne de texte
RTRIM	Une colonne par colonne d'entrée	Supprime les espaces de fin de chaque valeur de chaîne
STRING_CONCAT	Une colonne avec une valeur pour chaque ligne de la colonne d'entrée la plus courte	Concatène les chaînes de texte des plages de données spécifiées
STRING_HEAD	Une colonne par colonne d'entrée	Renvoie les <i>n</i> premiers caractères de chaque chaîne de la plage de données spécifiée
STRING_LENGTH	Une colonne par colonne d'entrée	Renvoie la longueur de chaque chaîne de la plage de données spécifiée
STRING_PROPER	Une colonne par colonne d'entrée	Convertit chaque valeur de chaîne en mettant en majuscule la première lettre de chaque chaîne ou toute lettre suivant un espace blanc ou un symbole (autre que le trait de soulignement) et transforme en minuscules tous les autres caractères
STRING_SEG	Une colonne par colonne d'entrée	Renvoie le segment de chaîne entre deux index spécifiés

Nom de la macro	Renvoie	Description
STRING_TAIL	Une colonne par colonne d'entrée	Renvoie les $n$ derniers caractères de chaque chaîne de la plage de données spécifiée
SUBSTR <i>or</i> SUBSTRING	Une colonne par colonne d'entrée	Renvoie les caractères d'une chaîne à partir d'une position initiale
UPPER	Une colonne par colonne d'entrée	Convertit une valeur de chaîne en majuscules

## Fonctions d'horodatage

Nom de la macro	Renvoie	Description
CURRENT_DATE	Une colonne par colonne d'entrée	Renvoie la date du jour dans le format <code>format</code> .
CURRENT_DAY	Une colonne par colonne d'entrée	Renvoie le jour du mois en cours sous la forme d'un nombre compris entre 1 et 31
CURRENT_JULIAN	Une colonne par colonne d'entrée	Renvoie le nombre au format Julien de la date du jour
CURRENT_MONTH	Une colonne par colonne d'entrée	Renvoie le mois en cours de l'année sous la forme d'un nombre compris entre 1 et 12
CURRENT_TIME	Une colonne par colonne d'entrée	Renvoie l'heure actuelle sous la forme d'une chaîne
CURRENT_WEEKDAY	Une colonne par colonne d'entrée	Renvoie le jour de la semaine en cours du mois sous la forme d'un nombre compris entre 0 et 6
CURRENT_YEAR	Une colonne par colonne d'entrée	Renvoie l'année en cours sous la forme d'un nombre
DATE	Une colonne par colonne d'entrée	Convertit une chaîne de date en date du calendrier julien
DATE_FORMAT	Une colonne par colonne d'entrée	Convertit les formats de date
DATE_JULIAN	Une colonne par colonne d'entrée	Renvoie la date au format Julien



<b>Nom de la macro</b>	<b>Renvoie</b>	<b>Description</b>
DATE_STRING	Une colonne par colonne d'entrée	Renvoie la chaîne de date sous la forme d'une date du calendrier julien
DAY_BETWEEN	Une colonne par colonne d'entrée	Renvoie le nombre de jours entre deux dates
DAY_FROMNOW	Une colonne par colonne d'entrée	Renvoie le nombre de jours entre la date du jour et la date spécifiée
DAY_INTERVAL	Une colonne par colonne d'entrée	Renvoie le nombre de jours entre deux dates
DAYOF	Une colonne par colonne d'entrée	Renvoie le jour du mois sous la forme d'un nombre
MONTHOF	Une colonne par colonne d'entrée	Renvoie le mois de l'année sous la forme d'un nombre
WEEKDAY	Une colonne par colonne d'entrée	Convertit les chaînes de date textuelles ASCII en jour de la semaine
WEEKDAYOF	Une colonne par colonne d'entrée	Renvoie le jour de la semaine de la semaine sous forme de nombre
YEAROF	Une colonne par colonne d'entrée	Renvoie l'année sous la forme d'un nombre

## Fonctions de groupement

<b>Nom de la macro</b>	<b>Renvoie</b>	<b>Description</b>
GROUPBY	Une colonne avec une valeur pour chaque ligne	Récapitule plusieurs lignes de données dans un groupe
GROUPBY_WHERE	Une colonne avec une valeur pour chaque ligne	Récapitule plusieurs lignes de données répondant à une condition indiquée et se trouvant dans un groupe

## fonctions diverses

Nom de la macro	Renvoi	Description
IF	Une colonne avec une valeur pour chaque ligne de la colonne d'entrée la plus courte	Commence une instruction conditionnelle if-then-else
ISERROR	Une colonne avec une valeur pour chaque ligne de la colonne d'entrée la plus courte	Renvoie un un si une valeur de la ligne en entrée contient une cible erronée (???). Renvoie zéro dans le cas contraire
ISEVEN	Une colonne par colonne d'entrée	Teste si les valeurs en entrée sont paires (à savoir, divisible par deux)
ISODD	Une colonne par colonne d'entrée	Teste si les valeurs en entrée sont impaires (à savoir, non divisible par deux)
ROWNUM	Une colonne par colonne d'entrée	Génère des nombres séquentiels de un à n, où n correspond au nombre d'enregistrements

## Paramètres des fonctions macro pour Unica Campaign

Cette section décrit les paramètres et l'utilisation des fonctions macro dans Unica Campaign.


### spécifications de format

Cette section décrit le format de certains paramètres fréquemment utilisés. Elle s'applique à toutes les références à ces paramètres par les spécifications de fonction macro de ce chapitre.

#### données

Le paramètre `data` représente une colonne de données sur laquelle une fonction macro doit agir.

Il peut s'agir d'une constante ou d'une zone. Pour plus d'informations, reportez-vous à la fonction macro spécifique.

 **Remarque** : Unica Campaign ne prend pas en charge les calculs simultanés sur plusieurs zones ou sur un sous-ensemble de lignes.

Certains autres noms de paramètre utilisent également le même format que `data`. Les descriptions de ces paramètres font référence à cette section et ce format.

## mot clé

Le paramètre `keyword` contrôle le comportement de la fonction macro. Il indique qu'un mot clé peut être spécifié (s'il est omis, la valeur par défaut est utilisée). Les options de mot clé sont répertoriées pour chaque fonction macro sous la forme suivante :


```
{choice1 | choice2 | choice3}
```

Sélectionnez l'option de mot clé offrant le comportement souhaité. L'option par défaut est indiquée en gras. Par exemple, pour les options suivantes :

```
{RADIANS | DEGREES}
```

Les fonctions macro suivantes sont toutes deux valides :

```
COS(V1, RADIANS) COS(V1, DEGREES)
```

 **Remarque** : De nombreuses fonctions de macro utilisent les paramètres de mot clé {ALL | COL | ROW}. Ces mots clés ne s'appliquent pas dans Unica Campaign car les données d'entrées correspondent toujours à une colonne ou à une zone. La macro se comporte toujours comme si le mot clé COL a été spécifié. Par conséquent, vous n'avez pas besoin de spécifier ces mots clés lorsque vous utilisez Unica Campaign.

## utilisation des constantes

La plupart des paramètres de fonction macro acceptent des constantes numériques ou des expressions converties en constante numérique (les fonctions macro utilisant des chaînes acceptent les constantes de chaîne).

Dans les fonctions macro effectuant des opérations enregistrement par enregistrement (par exemple, l'ajout de deux colonnes numériques), l'utilisation d'une constante revient à spécifier une colonne contenant cette valeur de constante dans chaque ligne. En fait, si une constante est fournie comme paramètre d'entrée, cette constante est étendue à la même longueur que l'entrée.

Certaines fonctions macro acceptent des chaînes de texte ASCII et des constantes numériques. Les paramètres acceptant à la fois des constantes numériques et des chaînes de texte ASCII sont indiqués dans la section "Paramètres" de chaque fonction macro.


Des exemples sont fournis dans le tableau ci-après.

<b>Définition de la fonction</b>	<b>Interprétation de la constante</b>
PERCENT_UTILIZ = (CURR_BAL*100)/CREDIT_LIM	La constante 100 est interprétée comme une colonne contenant le même nombre de lignes que la colonne CURR_BAL, chaque ligne contenant la constante 100. La zone dérivée PERCENT_UTILIZ contiendra chaque valeur de CURR_BAL multipliée par 100 et divisée par chaque valeur de CREDIT_LIM.
NAME = STRING_CONCAT ("Mr. ", LAST_NAME)	La constante "Mr. " est interprétée comme une colonne contenant le même nombre de lignes que la colonne LAST_NAME, chaque ligne contenant la constante "Mr. ". La zone dérivée NAME contiendra chacune des chaînes de texte de LAST_NAME précédée de "Mr. ".

# Chapitre 2. Utilisation de macros dans Unica Interact

Ce chapitre fournit des informations d'utilisation sur les macros contenues dans Unica Interact. Veuillez lire ce chapitre avant d'utiliser des macros dans Unica Interact.


## Auxiliaire de formule et vérification de la syntaxe des expressions

 **Important** : La boîte de dialogue **Assistant de formule**, y compris la fonction de vérification de syntaxe associée, valide actuellement les expressions de macro en fonction de ce qui est pris en charge par Unica Campaign supports. Toutefois, Unica Interact ne prend en charge qu'un sous-ensemble de la fonctionnalité de macro de Unica Campaign. Vous devez donc vous assurer que les macros et les mots clés (tels que les mots clés de format de date) utilisés pour Unica Interact sont pris en charge. Voir les notes relatives à Unica Interact dans le chapitre Guide des macros du présent guide.

## Récapitulatifs des fonctions macro pour Unica Interact

Les tableaux dans les sections suivantes présentent en détail les macros spécifiques à Unica Interact .

Des pages de référence détaillées sont fournies pour chaque macro, par ordre alphabétique, dans la rubrique [référence des macros \(à la page 21\)](#).

 **Important** : Les macros peuvent s'appliquer à la fois à Unica Campaign Unica Interact et à , ou à un seul de ces produits. Les descriptions des macros identifient les produits dans lesquels elles sont disponibles.

La rubrique [Paramètres des fonctions macro pour Unica Interact \(à la page 18\)](#) fournit des informations sur les paramètres en entrée de la fonction macro pour Unica Interact.

## Fonctions statistiques

Nom de la macro	Renvoi	Description
AVG	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la moyenne arithmétique d'une plage de cibles
MAX	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la valeur maximale d'une plage de cibles
MEAN	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la moyenne arithmétique d'une plage de cibles
MIN	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la valeur minimale d'une plage de cibles
STDEV ou STDEV	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule l'écart type d'une plage de cibles

## fonctions mathématiques et trigonométriques

Nom de la macro	Renvoi	Description
AVG	Une colonne par colonne d'entrée	Calcule la moyenne arithmétique des cibles de la plage de données spécifiée

Nom	Renvoi	Description
<b>macro</b>		
MAX	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; Calcule la valeur maximale d'une plage de cibles pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	
MEAN	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; Calcule la moyenne arithmétique d'une plage de cibles pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	
MIN	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; Calcule la valeur minimale d'une plage de cibles pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	
STDEV	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; Calcule l'écart type d'une plage de cibles pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	
SUM	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; Calcule la somme d'une plage de cibles pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	
TOTAL	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; Calcule la somme d'une colonne avec une valeur unique par colonne d'entrée pour une plage de cibles pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	

## fonctions des chaînes de caractères

Nom de la macro	Renvoi	Description
LIKE	Une colonne par colonne d'entrée	Détermine si une chaîne de texte correspond à un modèle spécifié
LOWER	Une colonne par colonne d'entrée	Convertit une valeur de chaîne en minuscules
LTRIM	Une colonne par colonne d'entrée	Supprime les espaces de début de chaque valeur de chaîne
NUMBER	Une colonne par colonne d'entrée	Convertit les chaînes de texte ASCII des heures et des dates en valeurs numériques
RTRIM	Une colonne par colonne d'entrée	Supprime les espaces de fin de chaque valeur de chaîne
STRING_CONCAT	Une colonne avec une valeur pour chaque ligne de la colonne d'entrée la plus courte	Concatène les chaînes des plages de données spécifiées
SUBSTR or SUBSTRING	Une colonne par colonne d'entrée	Renvoie les caractères d'une chaîne à partir d'une position initiale
UPPER	Une colonne par colonne d'entrée	Convertit une valeur de chaîne en majuscules

## Fonctions d'horodatage

Nom de la macro	Renvoi	Description
CURRENT_DATE	Une colonne par colonne d'entrée	Renvoie la date du jour dans le format <i>format</i> .
CURRENT_DAY	Une colonne par colonne d'entrée	Renvoie le jour du mois en cours sous la forme d'un nombre compris entre 1 et 31
CURRENT_MONTH	Une colonne par colonne d'entrée	Renvoie le mois en cours de l'année sous la forme d'un nombre compris entre 1 et 12
CURRENT_WEEKDAY	Une colonne par colonne d'entrée	Renvoie le jour de la semaine en cours du mois sous la forme d'un nombre compris entre 0 et 6



Nom de la macro	Revoie	Description
CURRENT_YEAR	Une colonne par colonne d'entrée	Revoie l'année en cours sous la forme d'un nombre
DATE	Une colonne par colonne d'entrée	Convertit une chaîne de date en date du calendrier julien
DATE_FORMAT	Une colonne par colonne d'entrée	Convertit les formats de date

## fonctions diverses

Nom de la macro	Revoie	Description
EXTERNALCALLOUT	Valeurs telles que définies par l'application personnalisée écrite à l'aide de l'API ExternalCallout	Appelle une application personnalisée écrite à l'aide de l'API ExternalCallout. Pour plus d'informations, reportez-vous au document Unica Interact - Guide d'administration.
IF	Une colonne avec une valeur pour chaque ligne de la colonne d'entrée la plus courte	Commence une instruction conditionnelle if-then-else
INDEXOF	Liste des valeurs pour la condition donnée dans une colonne	Conçue comme une macro prédéfinie, qui peut être utilisée en la transmettant en tant que paramètre dans la macro EXTERNALCALLOUT. Elle renvoie une liste des index qui répondent à l'expression donnée.

## Paramètres des fonctions macro pour Unica Interact


Cette section décrit les paramètres et l'utilisation des fonctions macro dans Unica Interact.

## spécifications de format

Cette section décrit le format de certains paramètres fréquemment utilisés. Elle s'applique à toutes les références à ces paramètres par les spécifications de fonction macro de cette section.

### données

Le paramètre `data` représente une colonne de données sur laquelle une fonction macro doit agir. Il peut s'agir d'une constante ou d'une zone. Pour plus d'informations, reportez-vous à la fonction macro spécifique.

 **Remarque :** Unica Interact ne prend pas en charge les calculs simultanés sur plusieurs zones ou sur un sous-ensemble de lignes.

Certains autres noms de paramètre utilisent également le même format que `data`. La description de ces paramètres fait référence à cette section et ce format.

### mot clé

Le paramètre `keyword` contrôle le comportement de la fonction macro. Il indique qu'un mot clé peut être spécifié (s'il est omis, la valeur par défaut est utilisée). Les options de mot clé sont répertoriées pour chaque fonction macro sous la forme suivante :


```
{choice1 | choice2 | choice3}
```

Sélectionnez l'option de mot clé offrant le comportement souhaité. L'option par défaut est indiquée en gras. Par exemple, pour les options suivantes :

```
{RADIANS | DEGREES}
```

Les fonctions macro suivantes sont toutes deux valides :

```
COS(V1, RADIANS) COS(V1, DEGREES)
```

 **Remarque :** De nombreuses fonctions de macro utilisent les paramètres de mot clé `{ALL | COL | ROW}`. Ces mots clés ne s'appliquent pas dans Unica Interact car les données d'entrées correspondent toujours à une colonne ou à une zone. La macro se

comporte toujours comme si le mot clé `COL` a été spécifié. Par conséquent, vous n'avez pas besoin de spécifier ces mots clés lorsque vous utilisez Unica Interact.

## utilisation des constantes


La plupart des paramètres de fonction macro acceptent des constantes numériques ou des expressions converties en constante numérique (les fonctions macro utilisant des chaînes acceptent les constantes de chaîne).

Dans les fonctions macro effectuant des opérations enregistrement par enregistrement (par exemple, l'ajout de deux colonnes numériques), l'utilisation d'une constante revient à spécifier une colonne contenant cette valeur de constante dans chaque ligne. En fait, si une constante est fournie comme paramètre d'entrée, cette constante est étendue à la même longueur que l'entrée.

Certaines fonctions macro acceptent aussi bien des chaînes de texte ASCII que des constantes numériques. Les paramètres acceptant à la fois des constantes numériques et des chaînes de texte ASCII sont indiqués dans la section "Paramètres" de chaque fonction macro.

Des exemples sont fournis dans le tableau ci-après.

<b>Définition de la fonction</b>	<b>Interprétation de la constante</b>
<code>PERCENT_UTILIZ = (CURR_BAL*100)/CREDIT_LIM</code>	La constante <code>100</code> est interprétée comme une colonne contenant le même nombre de lignes que la colonne <code>CURR_BAL</code> , chaque ligne contenant la valeur de <code>CURR_BAL</code> multipliée par <code>100</code> et divisée par chaque valeur de <code>CREDIT_LIM</code> . La zone dérivée <code>PERCENT_UTILIZ</code> contiendra chaque valeur de <code>CURR_BAL</code> multipliée par <code>100</code> et divisée par chaque valeur de <code>CREDIT_LIM</code> .
<code>NAME = STRING_CONCAT ("Mr. ", LAST_NAME)</code>	La constante <code>"Mr. "</code> est interprétée comme une colonne contenant le même nombre de lignes que la colonne <code>LAST_NAME</code> , chaque ligne contenant la constante <code>"Mr. "</code> . La zone dérivée <code>NAME</code> contiendra chacune des chaînes de texte de <code>LAST_NAME</code> précédée de <code>"Mr. "</code> .

 **Remarque :** Les constantes, telles que `DT_DELIM_M_D_Y`, requièrent des apostrophes.

# Chapitre 3. référence des macros

Cette section décrit toutes les macros pouvant être utilisées dans Campaign et/ou IBM Interact. Les macros sont répertoriées dans l'ordre alphabétique.

**⚠ Important :** N'utilisez pas des noms de fonction ou des mots clés du langage des macros pour les en-têtes de colonne des tables utilisateur dans Unica Campaign, que vous effectuiez un mappage à partir d'une base de données ou d'un fichier à plat. Ces mots réservés peuvent provoquer des erreurs s'ils sont utilisés dans les en-têtes de colonne des tables mappées.

## Mots clés de formats de date valides

Le tableau suivant présente les mots clés pour les formats valides avec une description et un exemple de chaque.

Mot clé	Description	Exemple(s)
MM	Mois sur 2 chiffres	01, 02, 03, ..., 12
MMDD	Mois sur 2 chiffres et jour sur 2 chiffres	31 mars correspond à 0331
MMDDYY	Mois sur deux chiffres, jour sur deux chiffres et année sur deux chiffres	31 mars 1970 correspond à 033170
MMDDYYYY	Mois sur deux chiffres, jour sur deux chiffres et année sur quatre chiffres	31 mars 1970 correspond à 03311970
DELIM_M_D	Tout mois délimité suivi du jour	31 mars, 3/31 ou 03-31
DELIM_M_D_Y	Toute combinaison de mois, jour et année délimitée	31 mars 1970 ou 3/31/70
DELIM_M_D_YYYY	Toute combinaison de mois, jour et année sur 4 chiffres délimitée	Mars 31, 1970 ou 3/31/1970

<b>Mot clé</b>	<b>Description</b>	<b>Exemple(s)</b>
DELIM_Y_M	Toute année délimitée suivie du mois	Mars, 70 ; 3-70 ou 3/1970
DELIM_Y_M_D	Toute combinaison d'année, de mois et de jour délimitée	1970 Mar 31 ou 70/3/31
YYMMM	Année sur 2 chiffres et mois sur 3 lettres	70MAR
YYMMDD	Année sur 2 chiffres, mois sur 3 lettres et jour sur 2 chiffres	70MAR31
YY	Année sur 2 chiffres	70
YYMM	Année sur 2 chiffres et mois sur 2 chiffres	7003
YYMMDD	Année sur 2 chiffres, mois sur 2 chiffres et jour sur 2 chiffres	700331
YYYYMMM	Année sur 4 chiffres et mois sur 3 lettres	1970MAR
YYYYMMDD	Année sur 4 chiffres, mois sur 3 lettres et jour sur 2 chiffres	1970MAR31
YYYY	Année sur 4 chiffres	1970
YYYYMM	Année sur 4 chiffres et mois sur 2 chiffres	197003
YYYYMMDD	Année sur 4 chiffres, mois sur 2 chiffres et jour sur 2 chiffres	19700331
DELIM_M_Y	Tout mois délimité suivi de l'année	3-70, 3/70, Mar 70, Mars 1970
DELIM_D_M	Tout jour délimité suivi du mois	31-3, 31/3, 31 Mars
DELIM_D_M_Y	Toute combinaison de jour, de mois et d'année délimitée	31-MAR-70, 31/3/1970, 31 03 70
DD	Jour sur 2 chiffres	31
DDMMM	Jour sur 2 chiffres et mois sur 3 lettres	31MAR

<b>Mot clé</b>	<b>Description</b>	<b>Exemple(s)</b>
DDMMYY	Jour sur 2 chiffres, mois sur 2 chiffres et année sur 2 chiffres	31MAR70
DDMMYYYY	Jour sur 2 chiffres, mois sur 2 chiffres et année sur 4 chiffres	31MAR1970
DDMM	Jour sur 2 chiffres et mois sur 2 chiffres	3103
DDMMYY	Jour sur 2 chiffres, mois sur 2 chiffres et année sur 2 chiffres	310370
DDMMYYYY	Jour sur 2 chiffres, mois sur 2 chiffres et année sur 4 chiffres	31031970
MMYY	Mois sur deux chiffres et année sur deux chiffres	0370
MMYYYY	Mois sur deux chiffres et année sur 4 chiffres	031970
MMM	Mois sur 3 lettres	MAR
MMMDD	Mois sur 3 lettres et jour sur 2 chiffres	MAR31
MMMDDYY	Mois sur 3 lettres, jour sur 2 chiffres et année sur 2 chiffres	MAR3170
MMMDDYYYY	Mois sur 3 lettres, jour sur 2 chiffres et année sur 4 chiffres	MAR311970
MMYY	Mois sur 3 lettres et année sur 2 chiffres	MAR70
MMYYYY	Mois sur 3 lettres et année sur 4 chiffres	MAR1970

Mot clé	Description	Exemple(s)
MONTH	Mois de l'année	Janvier, Février, Mars et ainsi de suite ou Jan, Fév, Mar, etc.
WEEKDAY	Jour de la semaine	Dimanche, Lundi, Mardi, etc. (Dimanche = 0)
WKD	Jour abrégé de la semaine	Dim, Lun, Mar, etc. (Dim = 0)

## Macro ABS

La macro `ABS` est disponible uniquement dans Unica Campaign.

### Syntaxe

`ABS(data)`

### Paramètres

`data`

Valeurs numériques dont la valeur absolue doit être calculée. Ce paramètre peut correspondre à une valeur constante, une colonne, une plage de cibles ou une expression convertie en l'un de ces types. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

`ABS` calcule la valeur absolue des nombres dans la plage de données spécifiée. La valeur absolue d'un nombre correspond à sa valeur dépourvue de signe (les nombres positifs restent identiques, tandis que les nombres négatifs sont renvoyés sous la forme de nombres positifs). `ABS` renvoie une nouvelle colonne par colonne d'entrée, chacune contenant la valeur absolue des nombres de la colonne d'entrée correspondante.

### Exemples

`TEMP = ABS(-3) OU TEMP = ABS(3)`

Crée une colonne nommée `TEMP` contenant la valeur trois.

`TEMP = ABS(V1)`

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur correspond à la valeur absolue du contenu de la colonne `V1`.

```
TEMP = ABS(V1:V3)
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux valeurs absolues du contenu de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux valeurs absolues du contenu de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux valeurs absolues du contenu de la colonne `V3`.

```
TEMP = ABS(V1[10:20])
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les valeurs absolues des valeurs des lignes 10 à 20 de la colonne `V1`. Les autres cibles de `TEMP` sont vides.

```
TEMP = ABS(V1[1:5]:V2)
```

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` correspondent aux valeurs absolues des lignes correspondantes de la colonne `V1` et les valeurs de la colonne `VX` correspondent aux valeurs absolues des lignes correspondantes de la colonne `V2`.

## Fonctions associées

Fonction	Description
<code>SIGN</code>	Détermine le signe (positif ou négatif) des valeurs dans la plage de données spécifiée

## Macro ACOS

Le `ACOS` est disponible uniquement dans Unica Campaign.

### Syntaxe

```
ACOS( data [, units_keyword ] )
```

### Paramètres

`data`



Valeurs numériques dont l'arc cosinus doit être calculé. Ce paramètre peut correspondre à une valeur constante, une colonne, une plage de cibles ou une expression convertie en l'un de ces types. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

`units_keyword`

Ce mot clé facultatif détermine si les valeurs en entrée et les résultats sont interprétés comme des degrés ou des radians. Sélectionnez l'une des valeurs suivantes :

`RADIAN` - Effectue les calculs en radians (valeur par défaut)

`DEGREE` - Effectue les calculs en degrés

Si ce paramètre n'est pas spécifié, la valeur par défaut est radians. (Pour convertir des radians en degrés, divisez par `PI` et multipliez par 180.)


Pour plus de détails sur l'utilisation des mots clés dans Unica Campaign, voir [spécifications de format \(à la page 11\)](#).

## Description

`ACOS` calcule l'arc cosinus des valeurs de la plage de données spécifiée. L'arc cosinus correspond à l'angle dont le cosinus est le contenu de chacune des cibles. `ACOS` renvoie une nouvelle colonne par colonne d'entrée, chacune contenant l'arc cosinus des nombres de la colonne d'entrée correspondante.

Utilisée avec le mot clé `RADIAN`, la fonction `ACOS` renvoie des valeurs comprises entre 0 et  $\pi$ .

Utilisée avec le mot clé `DEGREE`, la fonction `ACOS` renvoie des valeurs comprises entre 0 et 180.

 **Remarque :** Le contenu des cibles de chaque colonne spécifiée doit avoir des valeurs comprises entre -1.0 et 1.0 inclus. Dans le cas contraire, une cible vide est renvoyée pour chaque entrée non valide.

## Exemples

`TEMP = ACOS(0), TEMP = ACOS(0, 0) OU TEMP = ACOS(0, RADIAN)`

Crée une colonne nommée `TEMP` contenant la valeur 1.571 ( $\pi/2$  radians)

`TEMP = ACOS(0, 1) OU TEMP = ACOS(0, DEGREE)`

Crée une colonne nommée `TEMP` contenant la valeur 90 (degrés)

`TEMP = ACOS(V1)`

Crée une colonne nommée `TEMP`, où chaque valeur correspond à l'arc cosinus (en radians) du contenu de la colonne `V1`.

`TEMP = ACOS(V1:V3, 1)`

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux arcs cosinus du contenu de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux arcs cosinus du contenu de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux arcs cosinus du contenu de la colonne `V3`. Toutes les valeurs sont en degrés.

`TEMP = ACOS(V1[10:20])`

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les arcs cosinus des valeurs des lignes 10 à 20 de la colonne `V1` (en radians). Les autres cibles de `TEMP` sont vides.

`TEMP = ACOS(V1[1:5]:V2)`

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` correspondent aux arcs cosinus des lignes correspondantes de la colonne `V1` et les valeurs de la colonne `VX` correspondent aux arcs cosinus des lignes correspondantes de la colonne `V2`. Toutes les valeurs sont en radians.

## Fonctions associées

Fonction	Description
<code>ACOT</code>	Calcule l'arc cotangente du contenu de la plage de données spécifiée
<code>ASIN</code>	Calcule l'arc sinus du contenu de la plage de données spécifiée
<code>ATAN</code>	Calcule l'arc tangente du contenu de la plage de données spécifiée
<code>COS</code>	Calcule le cosinus du contenu de la plage de données spécifiée

# Macro ACOT

Le `ACOT` est disponible uniquement dans Unica Campaign.

## Syntaxe

```
ACOT(data [, units_keyword])
```

## Paramètres

`data`

Valeurs numériques dont l'arc cotangente doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

`units_keyword`

Ce mot clé facultatif détermine si les valeurs en entrée et les résultats sont interprétés comme des degrés ou des radians. Sélectionnez l'une des options suivantes :

`RADIAN` - Effectue les calculs en radians (valeur par défaut)

`DEGREE` - Effectue les calculs en degrés

Si ce paramètre n'est pas spécifié, la valeur par défaut est radians. (Pour convertir des radians en degrés, divisez par `PI` et multipliez par 180.)

Pour plus de détails sur l'utilisation des mots clés dans Unica Campaign, voir [spécifications de format \(à la page 11\)](#).

## Description

`ACOT` renvoie l'angle dont la cotangente correspond au contenu de chaque cible. `ACOT` renvoie une nouvelle colonne par colonne d'entrée, chacune contenant la chaîne convertie dans la colonne d'entrée correspondante. Des nombres à virgule flottante sur 64 bits sont utilisés.

## Exemples

```
TEMP = ACOT(0.5), TEMP = ACOT(0.5, 0) OU TEMP = ACOT(0.5, RADIAN)
```

Crée une colonne nommée `TEMP` contenant la valeur 2.157 (radians)

`TEMP = ACOT(1, 1)` OU `TEMP = ACOT(1, DEGREE)`

Crée une colonne nommée `TEMP` contenant la valeur 0.022 (1/45) degrés

`TEMP = ACOT(0)`

Crée une colonne nommée `TEMP` contenant la valeur `MAX32_Float` en radians

`TEMP = ACOT(V1)`

Crée une colonne nommée `TEMP`, où chaque valeur correspond à l'arc cotangente (en radians) du contenu de la colonne `V1`.

`TEMP = ACOT(V1:V3, 1)`

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux arcs cotangentes du contenu de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux arcs cotangentes du contenu de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux arcs cotangentes du contenu de la colonne `V3`. Toutes les valeurs sont en degrés.

`TEMP = ACOT(V1[10:20])`

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les arcs cotangentes des valeurs des lignes 10 à 20 de la colonne `V1` (en radians). Les autres cibles de `TEMP` sont vides.

`TEMP = ACOT(V1[1:5]:V2)`

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` correspondent aux arcs cotangentes des lignes correspondantes de la colonne `V1` et les valeurs de la colonne `VX` correspondent aux arcs cotangentes des lignes correspondantes de la colonne `V2`. Toutes les valeurs sont en radians.

## Fonctions associées

Fonction	Description
<code>ACOS</code>	Calcule le cosinus inverse du contenu de la plage de données spécifiée
<code>ASIN</code>	Calcule l'arc sinus du contenu de la plage de données spécifiée
<code>ATAN</code>	Calcule l'arc tangente du contenu de la plage de données spécifiée
<code>COT</code>	Calcule la cotangente du contenu de la plage de données spécifiée

# Macro ADD\_MONTHS

La macro `ADD_MONTHS` est disponible dans Unica Campaign. Pour les bases de données ORACLE, DB2 et MS SQL Server il est préférable d'effectuer l'exécution sur la base de données plutôt que sur le serveur Campaign.

## Syntaxe

```
ADD_MONTHS(months, date_string [, input_format[,DB]])
```

## Paramètres

`months`

Entier représentant un nombre de mois à ajouter à `date_string`.

`date_string`

Chaîne de texte représentant une date valide, au format `DELIM_M_D_Y` ou au format spécifié par l'argument `input_format` facultatif.

`input_format`

Format à utiliser pour la date calculée. Pour une liste des formats de date pris en charge, voir la fonction `DATE_FORMAT`. Notez que `input_format` détermine le format de la chaîne en entrée, mais aussi de la chaîne en sortie.

## DB

Ce paramètre est facultatif. L'exécution de macro est préférable sur la base de données pour les bases de données ORACLE/DB2/MS SQL Server, même si le paramètre de base de données n'est pas spécifié. Pour les autres types de base de données, le comportement reste le même, c'est-à-dire une exécution sur le serveur Campaign.

L'exécution se produira implicitement sur la base de données si l'expression contient une colonne de base de données.

L'exécution aura lieu sur le serveur Campaign si l'expression contient uniquement des colonnes hors colonnes de base de données. par exemple UCGF ou chaînes de date, etc. Pour forcer l'exécution sur la base de données, incluez le paramètre `DB`. Notez que

pour pouvoir spécifier le paramètre DB, vous devez également utiliser impérativement `input_format`.

## Description

`ADD_MONTHS` renvoie une date après l'ajout du nombre de mois spécifié à la chaîne `date_string` spécifiée. La date est renvoyée au format par défaut (`DELIM_M_D_Y`) ou au format spécifié par l'argument `input_format` facultatif. Si vous souhaitez un format différent en sortie, utilisez `DATE_FORMAT`.

Si l'augmentation du mois par le nombre de mois spécifié génère une date non valide, le résultat est calculé de sorte à correspondre au dernier jour du mois, comme illustré dans le dernier exemple ci-dessous. Si nécessaire, les années bissextiles sont prises en compte. Par exemple, l'ajout d'un mois à 31-Jan-2012 renvoie 29-Feb-2012.

## Exemples

`ADD_MONTHS(12, '06-25-11')` ajoute un an (12 mois) à la date spécifiée et renvoie la date 06-25-12.

`ADD_MONTHS(3, '2011-06-25', DT_DELIM_Y_M_D)` ajoute trois mois à la date spécifiée et renvoie la date 2011-09-25.

`ADD_MONTHS(1, '02-28-2011')` renvoie la date 03-28-2011.

`ADD_MONTHS(1, '03-31-2012')` renvoie la date 04-30-2012.

Exemples d'expressions et endroit où elles ont été exécutées. Les colonnes `DATE1`, `DATE2` sont des colonnes de la base de données.

Numéro de série	Expression	Exécution le
1	<code>DATE1 &lt; ADD_MONTHS(1,DATE2)</code>	Base de données
2	<code>DATE1 &lt; ADD_MONTHS(1,DATE2,DELIM_M_D_Y,DB)</code>	Base de données
3	<code>ADD_MONTHS(1,'02-29-2016',DELIM_M_D_Y,DB) &gt; DATE1</code>	Base de données
4	<code>ADD_MONTHS(24,'2012-02-29',DT_DELIM_Y_M_D) &gt; DATE2</code>	Serveur Campaign

Numéro de série	Expression	Exécution le
5	ADD_MONTHS(24,'2012-02-29',DT_DELIM_Y_M_D,DB) < DATE2	Base de données
6	DATE2 < ADD_MONTHS(1, DATE2)	Serveur Campaign
7	DATE2 < ADD_MONTHS(1, DATE2,DELIM_M_D_Y,DB)	Base de données
8	ADD_MONTHS(24,'2012-02-29', DELIM_Y_M_D, DB) > DATE2	Base de données
9	ADD_MONTHS(24,'02-29-2020') > DATE2	Serveur Campaign
10	DATE1 = ADD_MONTHS(1,DATE2)	Base de données
11	DATE1 = ADD_MONTHS(1,DATE2,DELIM_M_D_Y,DB)	Base de données
12	DATE1 != ADD_MONTHS(1,DATE2,DELIM_M_D_Y,DB)	Base de données
13	DATE1 != ADD_MONTHS(1,DATE2)	Base de données
14	ADD_MONTHS(3,'11NOV',DDMMM,DB) >DATE_FORMAT( DATE1,DT_DELIM_Y_M_D,DDMMM)	Serveur Campaign
15	ADD_MONTHS(0,'2012-02-29',DT_DELIM_Y_M_D) < DATE1	Base de données
16	ADD_MONTHS(-1, DATE1, DT_DELIM_Y_M_D, DB) < DATE2	Base de données

## Fonctions associées

Fonction	Description
DATE	Convertit une chaîne de date en date du calendrier julien.
DATE_FORMAT	Convertit une date de <code>input_format</code> en <code>output_format</code> .

## Macro AND

La macro `AND` est disponible dans Unica Campaign et Unica Interact.

### Syntaxe

```
data1 AND data2 data1 && data2
```

## Paramètres

`data1`

Nombres dont le AND logique doit être calculé avec les valeurs de `data2`. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .


`data2`

Le ou les nombres dont le AND logique doit être calculé avec les valeurs de `data1`. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans `data2` doit correspondre au nombre de colonnes dans `data1`, à moins que `data2` ne soit une constante. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

`AND` calcule le AND logique entre les deux plages de données spécifiées. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de `data1` dont le AND logique avec la colonne correspondante de `data2` a été calculé (à savoir, le AND logique de la première colonne de `data1` avec la première colonne de `data`, celui de la deuxième colonne avec la deuxième colonne et ainsi de suite).

Si `data2` est une constante, le AND logique de chaque valeur de `data1` avec cette valeur est calculé. Si `data2` contient une ou plusieurs colonnes, les calculs sont effectués ligne par ligne, entre une colonne de `data1` et une colonne de `data2`. Le AND logique de la première ligne de `data1` avec la première ligne de `data2` est calculé, la seconde ligne avec la seconde ligne, etc. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

 **Remarque :** L'opérateur `AND` peut être abrégé en double perluète (`&&`). Utilisez la double perluète pour séparer les deux arguments (par exemple, pour spécifier `V1 AND 3`, vous pouvez simplement entrer `V1&&3`).



## Exemples

`TEMP = 1 AND 8 OU TEMP = 1 && 8`

Crée une colonne nommée `TEMP` contenant la valeur un (tout nombre non nul prend la valeur un).

`TEMP = V1 && 1`

Crée une colonne nommée `TEMP` avec la valeur un pour chaque valeur de la colonne `V1`.

`TEMP = V1 && V1`

Crée une colonne nommée `TEMP` avec la valeur un pour chaque valeur non nulle de la colonne `V1` et la valeur zéro pour chaque zéro de la colonne `V1`.

`TEMP = V1 && V2`

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est la valeur de ligne de la colonne `V1` dont le AND logique est calculé avec la valeur de ligne correspondante de la colonne `V2`.

`TEMP = V1:V3 && V4:V6`

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. La colonne `TEMP` contient les valeurs de `V1` dont le AND logique avec les valeurs de ligne correspondantes de la colonne `V4` a été calculé. La colonne `VX` contient les valeurs résultant du AND logique des colonnes `V2` et `V5`. La colonne `VY` contient les valeurs résultant du AND logique des colonnes `V3` et `V6`.

`TEMP = V1[10:20] && V2 OU TEMP = V1[10:20] && V2[1:11]`

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent le résultat du AND logique des valeurs des lignes 10 à 20 de la colonne `V1` avec les valeurs des lignes 1 à 11 de la colonne `V2`. Les autres cibles de `TEMP` sont vides.

## Fonctions associées

Fonction	Description
NOT	Calcule le non logique (NON) du contenu de la plage de données spécifiée
OR	Calcule le OR logique entre deux plages de données spécifiées

# Macro ASIN

Le `ASIN` est disponible uniquement dans Unica Campaign.

## Syntaxe

```
ASIN(data [, units_keyword])
```

## Paramètres

`data`

Valeurs numériques dont l'arc sinus doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

`units_keyword`

Ce mot clé facultatif détermine si les valeurs en entrée et les résultats sont interprétés comme des degrés ou des radians. Sélectionnez l'une des options suivantes :

`RADIAN` - Effectue les calculs en radians (valeur par défaut)

`DEGREE` - Effectue les calculs en degrés


Si ce paramètre n'est pas spécifié, la valeur par défaut est radians. (Pour convertir des radians en degrés, divisez par `PI` et multipliez par 180.)

Pour plus de détails sur l'utilisation des mots clés dans Unica Campaign, voir [spécifications de format \(à la page 11\)](#).

## Description

`ASIN` calcule l'arc sinus des valeurs de la plage de données spécifiée. L'arc sinus correspond à l'angle dont le sinus est le contenu de chacune des cibles. `ASIN` renvoie une nouvelle colonne par colonne d'entrée, chacune contenant l'arc sinus des nombres de la colonne d'entrée correspondante.

Utilisée avec le mot clé `RADIAN`, la fonction `ASIN` renvoie des valeurs comprises entre  $-\pi/2$  et  $\pi/2$ . Utilisée avec le mot clé `DEGREE`, la fonction `ASIN` renvoie des valeurs comprises entre -90 et 90.

 **Remarque** : Le contenu des cibles de chaque colonne spécifiée doit avoir des valeurs comprises entre -1.0 et 1.0 inclus. Dans le cas contraire, ??? est renvoyé pour chaque entrée non valide.

## Exemples

`TEMP = ASIN(0.5), TEMP = ASIN(0.5, 0) OU TEMP = ASIN(0.5, RADIAN)`

Crée une nouvelle colonne nommée `TEMP` contenant la valeur 0.524. ( $\pi/6$  radians).

`TEMP = ASIN(0.5, 1) OU TEMP = ASIN(0.5, DEGREE)`

Crée une nouvelle colonne nommée `TEMP` contenant la valeur 30 (degrés).

`TEMP = ASIN(V1)`

Crée une colonne nommée `TEMP`, où chaque valeur correspond à l'arc sinus (en radians) du contenu de la colonne `V1`.

`TEMP = ASIN(V1:V3, 1)`

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux arcs sinus du contenu de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux arcs sinus du contenu de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux arcs sinus du contenu de la colonne `V3`. Toutes les valeurs sont en degrés.

`TEMP = ASIN(V1[10:20])`

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les arcs sinus des valeurs des lignes 10 à 20 de la colonne `V1` (en radians). Les autres cibles de `TEMP` sont vides.

`TEMP = ASIN(V1[1:5]:V2)`

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` correspondent aux arcs sinus des lignes correspondantes de la colonne `V1` et les valeurs de la colonne `VX` correspondent aux arcs sinus des lignes correspondantes de la colonne `V2`. Toutes les valeurs sont en radians.

## Fonctions associées

### Fonction

### Description

`ACOS` Calcule le cosinus inverse du contenu de la plage de données spécifiée

Fonction	Description
ACOT	Calcule l'arc cotangente du contenu de la plage de données spécifiée
ATAN	Calcule l'arc tangente du contenu de la plage de données spécifiée
SIN	Calcule le sinus du contenu de la plage de données spécifiée

## Macro ATAN

Le `ATAN` est disponible uniquement dans Unica Campaign.

### Syntaxe

```
ATAN(data [, units_keyword])
```

### Paramètres

`data`

Valeurs numériques dont l'arc tangente doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

`units_keyword`

Ce mot clé facultatif détermine si les valeurs en entrée et les résultats sont interprétés comme des degrés ou des radians. Sélectionnez l'une des options suivantes :

`RADIAN` - Effectue les calculs en radians (valeur par défaut)

`DEGREE` - Effectue les calculs en degrés

Si ce paramètre n'est pas spécifié, la valeur par défaut est radians. (Pour convertir des radians en degrés, divisez par `PI` et multipliez par 180.)

Pour plus de détails sur l'utilisation des mots clés dans Unica Campaign, voir [spécifications de format \(à la page 11\)](#).

## Description

`ATAN` calcule l'arc tangente des valeurs de la plage de données spécifiée. L'arc tangente correspond à l'angle dont la tangente est contenue dans chacune des cibles. Pour chaque nombre de la colonne d'entrée, `ATAN` renvoie l'arc tangente correspondant dans une nouvelle colonne.

Utilisée avec le mot clé `RADIAN`, la fonction `ATAN` renvoie des valeurs comprises entre  $-\pi/2$  et  $\pi/2$ . Utilisée avec le mot clé `DEGREE`, la fonction `ATAN` renvoie des valeurs comprises entre  $-90$  et  $90$ .

## Exemples

`TEMP = ATAN(1), TEMP = ATAN(1, 0) OU TEMP = ATAN(1, RADIAN)`

Crée une nouvelle colonne nommée `TEMP` contenant la valeur 0.785. ( $\pi/4$  radians).

`TEMP = ATAN(1, 1) OU TEMP = ATAN(1, DEGREE)`

Crée une nouvelle colonne nommée `TEMP` contenant la valeur 45 (degrés).

`TEMP = ATAN(V1)`

Crée une colonne nommée `TEMP`, où chaque valeur correspond à l'arc tangente (en radians) du contenu de la colonne `V1`.

`TEMP = ATAN(V1:V3, 1)`

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux arcs tangentes du contenu de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux arcs tangentes du contenu de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux arcs tangentes du contenu de la colonne `V3`. Toutes les valeurs sont en degrés.

`TEMP = ATAN(V1[10:20])`

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les arcs tangentes des valeurs des lignes 10 à 20 de la colonne `V1` (en radians). Les autres cibles de `TEMP` sont vides.

`TEMP = ATAN(V1[1:5]:V2)`

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` correspondent aux arcs tangentes

des lignes correspondantes de la colonne `v1` et les valeurs de la colonne `vX` correspondent aux arcs tangentes des lignes correspondantes de la colonne `v2`. Toutes les valeurs sont en radians.

## Fonctions associées

Fonction	Description
ACOS	Calcule l'arc cosinus du contenu de la plage de données spécifiée
ASIN	Calcule l'arc sinus du contenu de la plage de données spécifiée
ATAN	Calcule l'arc tangente du contenu de la plage de données spécifiée
TAN	Calcule la tangente du contenu de la plage de données spécifiée

## Macro AVG

Le `AVG` est disponible dans Unica Campaign et Unica Interact.

### Syntaxe

```
AVG(data [, keyword])
```

### Paramètres

`data`

Valeurs numériques dont la moyenne arithmétique doit être calculée. Ces valeurs peuvent être une valeur constante, une colonne, une plage de cibles ou une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

`keyword`


Ce mot clé facultatif détermine la manière dont le calcul est effectué sur la plage de données d'entrée. Sélectionnez l'un des mots clés suivants :

`ALL` - Effectue le calcul sur toutes les cibles de `data` (valeur par défaut)

`COL` - Effectue le calcul séparément pour chaque colonne de `data`

ROW - Effectue le calcul séparément pour chaque ligne de `data`

Pour plus de détails sur l'utilisation des mots clés dans Unica Campaign, voir [spécifications de format \(à la page 11\)](#).

 **Remarque** : De nombreuses fonctions de macro utilisent les paramètres de mot clé {ALL | COL | ROW}. Ces mots clés ne s'appliquent pas dans **Unica Campaign** car les données d'entrées correspondent toujours à une colonne ou à une zone. La macro se comporte toujours comme si le mot clé COL a été spécifié. Ainsi, il n'est pas nécessaire de spécifier ces mots clés lorsque vous utilisez **Unica Campaign**.

## Description

AVG calcule la moyenne arithmétique des cibles de la plage de données spécifiée. Pour calculer la moyenne arithmétique, le contenu de l'ensemble des cellules est additionné, puis le résultat est divisé par le nombre de cellules. Le nombre de colonnes renvoyé par AVG dépend de `keyword`.

- Si `keyword` a la valeur ALL, AVG renvoie une nouvelle colonne, contenant une valeur unique (la moyenne de toutes les cibles de `data`).
- Si `keyword` a la valeur COL, AVG renvoie une nouvelle colonne pour chaque colonne d'entrée. Chaque nouvelle colonne contient une valeur (moyenne de toutes les cibles de la colonne d'entrée correspondante).
- Si `keyword` a la valeur ROW, AVG renvoie une nouvelle colonne qui contient la moyenne de chaque ligne de `data`.

 **Remarque** : Les cibles vides sont exclues du calcul.

 **Remarque** : AVG est identique à la fonction macro MEAN.

## Exemples

```
TEMP = AVG(V1)
```

Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la moyenne arithmétique du contenu de la colonne V1.

```
TEMP = AVG(V1:V3)
```

Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la moyenne arithmétique du contenu des colonnes V1, V2 et V3.

```
TEMP = AVG(V1[10:20])
```

Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la moyenne arithmétique des cibles sur les lignes 10-20 de la colonne V1.

```
TEMP = AVG(V1[1:5]:V4)
```

Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la moyenne arithmétique des cibles sur les lignes 1-5 des colonnes V1 à V4.

```
TEMP = AVG(V1:V3, COL)
```

Crée trois colonnes nommées TEMP, VX et VY. La valeur unique dans la colonne TEMP est la moyenne arithmétique du contenu de la colonne V1. La valeur unique dans la colonne VX est la moyenne arithmétique du contenu de la colonne V2. La valeur unique dans la colonne VY est la moyenne arithmétique du contenu de la colonne V3.

```
TEMP = AVG(V1[1:5]:V3, COL)
```

Crée trois colonnes nommées TEMP, VX et VY contenant chacune une seule valeur. La valeur de la colonne TEMP est la moyenne arithmétique des cibles sur les lignes 1-5 de la colonne V1. La valeur de la colonne VX est la moyenne arithmétique des cibles sur les lignes 1-5 de la colonne V2. La valeur de la colonne VY est la moyenne arithmétique des cibles sur les lignes 1-5 de la colonne V3.

```
TEMP = AVG(V1, ROW)
```

Crée une colonne nommée TEMP contenant les mêmes valeurs que la colonne V1 (la moyenne arithmétique de tout nombre est lui-même).

```
TEMP = AVG(V1:V3, ROW)
```

Crée une colonne nommée TEMP dans laquelle chaque entrée de cible correspond à la moyenne arithmétique de la ligne correspondante dans les colonnes V1, V2 et V3.

```
TEMP = AVG(V1[1:5]:V3, ROW)
```

Crée une colonne nommée TEMP, dans laquelle les cibles des lignes 1 à 5 contiennent la moyenne arithmétique de la ligne correspondante dans les colonnes V1 à V3. Les autres cibles de TEMP sont vides.



## Fonctions associées

Fonction	Description
SUM OU TOTAL	Calcule la somme d'une plage de cibles

## Macro BETWEEN

Le `BETWEEN` est disponible uniquement dans Unica Campaign.

### Syntaxe


```
value1 BETWEEN value2 AND value3
```

### Paramètres

Equivalent to `value1 >= value2 AND < value3`

### Description

`BETWEEN` est une variante spéciale du prédicat de comparaison. Les détails de ce prédicat sont importants et l'ordre des opérandes a des conséquences inattendues. Voir la section des exemples.

 **Remarque** : `FROM` et `FOR` utilise une syntaxe identique.

### Exemples

```
10 BETWEEN 5 AND 15 Is true, but: 10 BETWEEN 15 AND 5 Is false:
```

car la manière équivalente d'exprimer `BETWEEN` (à l'aide de `AND`) a un ordre spécifique qui n'a pas d'importance lorsque vous utilisez des littéraux, mais qui en a une si vous spécifiez `value2` et `value3` à l'aide de variables hôte, de paramètres, voire de sous-requêtes.

## Macro BIT\_AND

Le `BIT_AND` est disponible dans Unica Campaign et Unica Interact.

## Syntaxe

```
data1 BIT_AND data2 data1 & data2
```

## Paramètres

`data1`

Entiers non négatifs dont le AND bit à bit doit être calculé avec les valeurs de `data2`. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .


`data2`


Entier(s) non négatif(s) dont le AND bit à bit doit être calculé avec les valeurs de `data1`. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans `data2` doit correspondre au nombre de colonnes dans `data1`, à moins que `data2` ne soit une constante. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

`BIT_AND` calcule le AND bit à bit entre les deux plages de données spécifiées. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de `data1` dont le AND bit à bit avec la colonne correspondante de `data2` a été calculé (à savoir, le AND bit à bit de la première colonne de `data1` avec la première colonne de `data`, celui de la deuxième colonne avec la deuxième colonne et ainsi de suite).

Si `data2` est une constante, le AND bit à bit de chaque valeur de `data1` avec cette valeur est calculé. Si `data2` contient une ou plusieurs colonnes, les calculs sont effectués ligne par ligne, entre une colonne de `data1` et une colonne de `data2`. Le AND bit à bit de la première ligne de `data1` avec la première ligne de `data2` est calculé, la seconde ligne avec la seconde ligne, etc. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

 **Remarque** : La précision de cette fonction macro est limitée aux valeurs entières inférieures à  $2^{24}$ . Les valeurs négatives sont interdites.

 **Remarque** : L'opérateur `BIT_AND` peut être abrégé en perluète (`&`). Utilisez la perluète pour séparer les deux arguments (par exemple, pour spécifier `BIT_AND(V1, 3)`, vous pouvez simplement entrer `V1&3`).

## Exemples

```
TEMP = 3 BIT_AND 7 OU TEMP = 3 & 7
```

Crée une colonne nommée `TEMP` contenant la valeur trois (le AND bit à bit de `011` et `111` est égal à `011`).

```
TEMP = V1 & 8
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur correspond au AND bit à bit du contenu de la colonne `V1` avec la valeur binaire `1000`.

```
TEMP = V1 & V1
```

Crée une colonne nommée `TEMP` avec le même contenu que la colonne `V1` (un AND entre toute valeur et elle-même génère cette même valeur).

```
TEMP = V1 & V2
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est la valeur de ligne de la colonne `V1` dont le AND bit à bit est calculé avec la valeur de ligne correspondante de la colonne `V2`.

```
TEMP = V1:V3 & V4:V6
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. La colonne `TEMP` contient les valeurs de `V1` dont le AND bit à bit avec les valeurs de ligne correspondantes de la colonne `V4` a été calculé. La colonne `VX` contient les valeurs résultant du AND bit à bit des colonnes `V2` et `V5`. La colonne `VY` contient les valeurs résultant du AND bit à bit des colonnes `V3` et `V6`.

```
TEMP = V1[10:20] & V2 OU TEMP = V1[10:20] & V2[1:11]
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent le résultat du AND bit à bit des valeurs des lignes 10 à 20 de la colonne `V1` avec les valeurs des lignes 1 à 11 de la colonne `V2`. Les autres cibles de `TEMP` sont vides.

## Fonctions associées

Fonction	Description
<code>BIT_NOT</code>	Calcule le NOT bit à bit du contenu de la plage de données spécifiée
<code>BIT_OR</code>	Calcule le OR bit à bit entre deux plages de données spécifiées
<code>BIT_XOR</code> ou <code>XOR</code>	Calcule le XOR bit à bit entre deux plages de données spécifiées

## Macro BIT\_NOT

Le `BIT_NOT` est disponible dans Unica Campaign et Unica Interact.

### Syntaxe

```
BIT_NOT data ~ data
```


### Paramètres


`data`


Entiers non négatifs dont le NOT bit à bit doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

### Description

`BIT_NOT` calcule le NOT bit à bit des valeurs de la plage de données spécifiée. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant le NOT bit à bit des valeurs des colonnes correspondantes de `data`.

 **Remarque** : La précision de cette fonction macro est limitée aux valeurs entières inférieures à  $2^{24}$ . Les valeurs négatives sont interdites.

 **Remarque** : L'utilisation d'une colonne contenant le même nombre `x` dans chaque ligne que `data` revient à utiliser la constante `x` pour `data`.

 **Remarque** : L'opérateur `BIT_NOT` peut être abrégé en tilde (`~`). Utilisez le tilde avant la valeur (par exemple, pour spécifier `BIT_NOT(V1)`, vous pouvez simplement saisir `~V1`).

## Exemples

```
TEMP = BIT_NOT 3 OU TEMP = ~3
```

Crée une colonne nommée `TEMP` contenant la valeur quatre (le NOT bit à bit de `011` est égal à `100`).

```
TEMP = ~V1
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur correspond au NOT bit à bit du contenu de la colonne `V1`.

```
TEMP = ~V1:V3
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent au NOT bit à bit du contenu de la colonne `V1`, les valeurs de la colonne `VX` correspondent au NOT bit à bit du contenu de la colonne `V2` et les valeurs de la colonne `VY` correspondent au NOT bit à bit du contenu de la colonne `V3`.

```
TEMP = ~V1[100:200]
```

Crée une colonne nommée `TEMP`, dans laquelle les 101 premières cibles contiennent le NOT bit à bit des valeurs des lignes 1 à 50 de la colonne `V1`.

## Fonctions associées

Fonction	Description
<code>BIT_AND</code>	Calcule le AND bit à bit entre deux plages de données spécifiées
<code>BIT_OR</code>	Calcule le OR bit à bit entre deux plages de données spécifiées
<code>BIT_XOR</code> ou <code>XOR</code>	Calcule le XOR bit à bit entre deux plages de données spécifiées

## Macro BIT\_OR

Le `BIT_OR` est disponible dans Unica Campaign et Unica Interact.

### Syntaxe

```
data1 BIT_OR data2 data1 OR data2 data1 | data2
```

### Paramètres

`data1`

Entiers non négatifs dont le OR bit à bit doit être calculé avec les valeurs de `data2`. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .


`data2`


Entier(s) non négatif(s) dont le OR bit à bit doit être calculé avec les valeurs de `data1`. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans `data2` doit correspondre au nombre de colonnes dans `data1`, à moins que `data2` ne soit une constante. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

`BIT_OR` calcule le OR bit à bit entre les deux plages de données spécifiées. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de `data1` dont le OR bit à bit avec la colonne correspondante de `data2` a été calculé (à savoir, le OR bit à bit de la première colonne de `data1` avec la première colonne de `data`, celui de la deuxième colonne avec la deuxième colonne et ainsi de suite).

Si `data2` est une constante, le OR bit à bit de chaque valeur de `data1` avec cette valeur est calculé. Si `data2` contient une ou plusieurs colonnes, les calculs sont effectués ligne par ligne, entre une colonne de `data1` et une colonne de `data2`. Le OR bit à bit de la première ligne de `data1` avec la première ligne de `data2` est calculé, la seconde ligne avec la seconde ligne, etc. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

 **Remarque** : La précision de cette fonction macro est limitée aux valeurs entières inférieures à  $2^{24}$ . Les valeurs négatives sont interdites.

 **Remarque** : L'opérateur `BIT_OR` peut être abrégé en barre verticale (`|`). Utilisez la barre verticale pour séparer les deux colonnes (par exemple, au lieu d'indiquer `BIT_OR(V1, 3)`, vous pouvez simplement taper `V1 | 3`. Vous pouvez également utiliser `OR`.

## Exemples

`TEMP = 3 BIT_OR 7, TEMP = 3 OR 7 OU TEMP = 3 | 7`

Crée une colonne nommée `TEMP` contenant la valeur sept (le OR bit à bit de `011` et `111` est égal à `111`).

`TEMP = V1 | 8`

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur correspond au OR bit à bit du contenu de la colonne `V1` avec la valeur binaire `1000`.

`TEMP = V1 | V1`

Crée une colonne nommée `TEMP` avec le même contenu que la colonne `V1` (un OR entre toute valeur et elle-même génère cette même valeur).

`TEMP = V1 | V2`

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est la valeur de ligne de la colonne `V1` dont le OR bit à bit est calculé avec la valeur de ligne correspondante de la colonne `V2`.

`TEMP = V1:V3 | V4:V6`

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. La colonne `TEMP` contient les valeurs de `V1` dont le OR logique avec les valeurs de ligne correspondantes de la colonne `V4` a été calculé. La colonne `VX` contient les valeurs résultant du OR logique des colonnes `V2` et `V5`. La colonne `VY` contient les valeurs résultant du OR logique des colonnes `V3` et `V6`.

`TEMP = V1[10:20] | V2 OU TEMP = V1[10:20] | V2[1:11]`

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent le résultat du OR bit à bit des valeurs des lignes 10 à 20 de la colonne `V1` avec les valeurs des lignes 1 à 11 de la colonne `V2`. Les autres cibles de `TEMP` sont vides.

## Fonctions associées

Fonction	Description
<code>BIT_AND</code>	Calcule le AND bit à bit entre deux plages de données spécifiées
<code>BIT_NOT</code>	Calcule le NOT bit à bit du contenu de la plage de données spécifiée
<code>BIT_XOR</code> ou <code>XOR</code>	Calcule le XOR bit à bit entre deux plages de données spécifiées

# Macro BIT\_XOR

Le `BIT_XOR` est disponible dans Unica Campaign et Unica Interact.

## Syntaxe

```
data1 BIT_XOR data2
```

## Paramètres

`data1`

Entiers non négatifs dont le XOR bit à bit doit être calculé avec les valeurs de `data2`. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

`data2`

Entier(s) non négatif(s) dont le XOR bit à bit doit être calculé avec les valeurs de `data1`. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans `data2` doit correspondre au nombre de colonnes dans `data1`, à moins que `data2` ne soit une constante. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .


## Description

`BIT_XOR` calcule le XOR bit à bit entre les deux plages de données spécifiées. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de `data1` dont le XOR bit à bit avec la colonne correspondante de `data2` a été calculé (à savoir, le XOR bit à bit de la première colonne de `data1` avec la première colonne de `data`, celui de la deuxième colonne avec la deuxième colonne et ainsi de suite).

Si `data2` est une constante, le XOR bit à bit de chaque valeur de `data1` avec cette valeur est calculé. Si `data2` contient une ou plusieurs colonnes, les calculs sont effectués ligne par ligne, entre une colonne de `data2` et une colonne de `data2`. Le XOR bit à bit de la première ligne de `data1` avec la première ligne de `data2` est calculé, la seconde ligne avec la seconde



ligne, etc. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

 **Remarque** : La précision de cette fonction macro est limitée aux valeurs entières inférieures à  $2^{24}$ . Les valeurs négatives sont interdites.

## Exemples

```
TEMP = 3 BIT_XOR 7
```

Crée une colonne nommée `TEMP` contenant la valeur quatre (le XOR bit à bit de 011 et 111 est égal à 100).

```
TEMP = V1 BIT_XOR 8
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur correspond au XOR bit à bit du contenu de la colonne `V1` avec la valeur binaire 1000.

```
TEMP = V1 BIT_XOR V1
```

Crée une colonne nommée `TEMP` ne contenant que des zéros (un XOR entre toute valeur et elle-même génère zéro).

```
TEMP = V1 BIT_XOR V2
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est la valeur de ligne de la colonne `V1` dont le XOR bit à bit est calculé avec la valeur de ligne correspondante de la colonne `V2`.

```
TEMP = V1:V3 BIT_XOR V4:V6
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. La colonne `TEMP` contient les valeurs de `V1` dont le XOR bit à bit avec les valeurs de ligne correspondantes de la colonne `V4` a été calculé. La colonne `VX` contient les valeurs résultant du XOR bit à bit des colonnes `V2` et `V5`. La colonne `VY` contient les valeurs résultant du XOR bit à bit des colonnes `V3` et `V6`.

```
TEMP = V1[10:20] BIT_XOR V2 OU TEMP = V1[10:20] BIT_XOR V2[1:11]
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent le résultat du XOR bit à bit des valeurs des lignes 10 à 20 de la colonne `V1` avec les valeurs des lignes 1 à 11 de la colonne `V2`. Les autres cibles de `TEMP` sont vides.

## Fonctions associées

Fonction	Description
<code>BIT_AND</code>	Calcule le AND bit à bit entre deux plages de données spécifiées
<code>BIT_NOT</code>	Calcule le NOT bit à bit du contenu de la plage de données spécifiée
<code>BIT_OR</code>	Calcule le OR bit à bit entre deux plages de données spécifiées

## Macro CEILING

Le `CEILING` est disponible uniquement dans Unica Campaign.

### Syntaxe

`CEILING(data)`

### Paramètres

`data`

Valeurs numériques dont le plafond doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

### Description

`CEILING` calcule le plafond des valeurs de la plage de données spécifiée. Le plafond d'un nombre est l'entier le plus petit non inférieur à ce nombre. `CEILING` renvoie une nouvelle colonne par colonne d'entrée, chacune contenant le plafond des nombres de la colonne d'entrée correspondante.

### Exemples

```
TEMP = CEILING(4.3)
```

Crée une nouvelle colonne nommée `TEMP` contenant la valeur 5.

```
TEMP = CEILING(2.9)
```

Crée une nouvelle colonne nommée `TEMP` contenant la valeur -2.

```
TEMP = CEILING(V1)
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur correspond au plafond du contenu de la colonne `V1`.

```
TEMP = CEILING(V1:V3)
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux plafonds du contenu de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux plafonds du contenu de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux plafonds du contenu de la colonne `V3`.

```
TEMP = CEILING(V1[10:20])
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les plafonds des valeurs des lignes 10 à 20 de la colonne `V1`. Les autres cibles de `TEMP` sont vides.

```
TEMP = CEILING(V1[50:99]:V2)
```

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 50 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` sont les plafonds des lignes de la colonne `V1` et les valeurs de la colonne `VX` sont les plafonds des valeurs de la colonne `V2`.

## Fonctions associées

Fonction	Description
FLOOR OU INT	Calcule la partie entière de chaque valeur de la plage de données spécifiée
FRACTION	Renvoie la partie décimale de chaque valeur de la plage de données spécifiée
TRUNCATE	Renvoie la partie non décimale de chaque valeur de la plage de données spécifiée

## Macro COLUMN

Le `COLUMN` est disponible uniquement dans Unica Campaign.

### Syntaxe

```
COLUMN(data [, data]...) or (data [, data]...)
```


## Paramètres

`data`

Valeur à utiliser lors de la création d'une colonne. Il peut s'agir d'une valeur constante (valeur numérique ou texte ASCII entre guillemets), d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Ce paramètre peut être répété plusieurs fois, mais les paramètres suivants doivent avoir la même dimensionnalité (largeur de colonne) que le premier paramètre. Toutes les valeurs de tous les paramètres `data` doivent être des valeurs numériques ou du texte ASCII (vous ne pouvez pas mélanger des valeurs numériques et du texte). Si plusieurs paramètres `data` sont fournis, ils doivent tous comporter le même nombre de colonnes. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

`COLUMN` concatène verticalement ses entrées dans les colonnes d'un groupe de fonctions. Le nombre de nouvelles colonnes renvoyé dépend de celui indiqué dans chaque paramètre d'entrée. Un nombre illimité d'arguments peuvent être fournis. Tous les arguments doivent être des valeurs numériques ou des chaînes de texte ASCII (vous ne pouvez pas mélanger des valeurs numériques et du texte).

 **Remarque** : La fonction macro `COLUMN` peut être abrégée en séparant les arguments `data` par des virgules, entre parenthèses (par exemple, `TEMP = MEAN((1, 2, 3, 4), ALL)`). En dehors d'une autre fonction macro, la paire de parenthèses n'est pas nécessaire (par exemple, `V1=1, 2, 3` est équivalent à `V1=COLUMN(1, 2, 3)`).

## Exemples

```
TEMP = COLUMN(3, 4, 5), TEMP = (3, 4, 5) OU TEMP = 3, 4, 5
```

Crée une colonne nommée `TEMP` dont les trois premières cibles contiennent les valeurs 3, 4 et 5.

```
TEMP = COLUMN("one", "two", "three")
```

Crée une colonne nommée `TEMP` dont les trois premières cibles contiennent les valeurs " one ", " two " et " three ".

```
TEMP = AVG(V1), STDV(V1)
```

Crée une colonne nommée `TEMP` avec la moyenne de la colonne `V1` dans la première cible et l'écart type de la colonne `V1` dans la deuxième cible.

```
TEMP = V1:V2, V3:V4
```

Crée deux colonnes nommées `TEMP` et `VX` où la colonne `TEMP` contient les valeurs de la colonne `V1` suivies des valeurs de la colonne `V3`. La colonne `VX` contient les valeurs de la colonne `V2` suivies des valeurs de la colonne `V4`.

```
TEMP = V1:V2, V3:V4
```

Crée deux colonnes nommées `TEMP` et `VX` où la colonne `TEMP` contient les valeurs des cibles 1 à 10 de la colonne `V1` suivies de toutes les valeurs de la colonne `V3`. La colonne `VX` contient les valeurs des cibles 1 à 10 de la colonne `V2` suivies de toutes les valeurs de la colonne `V4`.

```
TEMP = V1:V2, V3:V4
```

Crée deux colonnes nommées `TEMP` et `VX`, contenant chacune une valeur unique. La colonne `TEMP` contient la moyenne des colonnes `V1` et `V2`. La colonne `VX` contient la moyenne des colonnes `V3` et `V4`.

## Macro COS

Le `COS` est disponible uniquement dans Unica Campaign.

### Syntaxe

```
COS(data [, units_keyword])
```

### Paramètres

`data`

Valeurs numériques dont le cosinus doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit.

`units_keyword`

Ce mot clé facultatif détermine si les valeurs en entrée et les résultats sont interprétés comme des degrés ou des radians. Sélectionnez l'une des options suivantes :

`RADIAN` - Effectue les calculs en radians (valeur par défaut)

`DEGREE` - Effectue les calculs en degrés

Si ce paramètre n'est pas spécifié, la valeur par défaut est radians. (Pour convertir des radians en degrés, divisez par `PI` et multipliez par 180.)

Pour plus de détails sur l'utilisation des mots clés dans Unica Campaign, voir [spécifications de format \(à la page 11\)](#).

## Description

`COS` calcule le cosinus des valeurs de la plage de données spécifiée. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant le cosinus des nombres de la colonne d'entrée correspondante.

## Exemples

`TEMP = COS(PI), TEMP = COS(PI, 0) OU TEMP = COS(PI, RADIAN)`

Renvoie une colonne nommée `TEMP` contenant la valeur unique -1.

`TEMP = COS(90, 1) OU TEMP = COS(90, DEGREE)`

Renvoie une colonne nommée `TEMP` contenant la valeur unique zéro.

`TEMP = COS(V1), TEMP = COS(V1, 0) OU TEMP = COS(V1, RADIAN)`

Crée une colonne nommée `TEMP`, où chaque valeur correspond au cosinus (en radians) du contenu de la colonne `V1`.

`TEMP = COS(V1:V3, 1)`

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux cosinus du contenu de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux cosinus du contenu de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux cosinus du contenu de la colonne `V3`. Toutes les valeurs sont en degrés.

`TEMP = COS(V1[10:20])`

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les cosinus des valeurs des lignes 10 à 20 de la colonne `V1` (en radians). Les autres cibles de `TEMP` sont vides.

```
TEMP = COS(V1[1:5]:V2)
```

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` correspondent aux cosinus des lignes correspondantes de la colonne `V1` et les valeurs de la colonne `VX` correspondent aux cosinus des lignes correspondantes de la colonne `V2`. Toutes les valeurs sont en radians.

## Fonctions associées

Fonction	Description
<code>ACOS</code>	Calcule l'arc cosinus du contenu de la plage de données spécifiée
<code>COSH</code>	Calcule le cosinus hyperbolique du contenu de la plage de données spécifiée
<code>SIN</code>	Calcule le sinus du contenu de la plage de données spécifiée
<code>TAN</code>	Calcule la tangente du contenu de la plage de données spécifiée

## Macro COSH

Le `COSH` est disponible uniquement dans Unica Campaign.

### Syntaxe

```
COSH(data [, units_keyword])
```

### Paramètres

`data`

Valeurs numériques dont le cosinus hyperbolique doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

`units_keyword`

Ce mot clé facultatif détermine si les valeurs en entrée et les résultats sont interprétés comme des degrés ou des radians. Sélectionnez l'une des options suivantes :

RADIAN - Effectue les calculs en radians (valeur par défaut)

DEGREE - Effectue les calculs en degrés

Si ce paramètre n'est pas spécifié, la valeur par défaut est radians. (Pour convertir des radians en degrés, divisez par  $\pi$  et multipliez par 180.)


Pour plus de détails sur l'utilisation des mots clés dans Unica Campaign, voir [spécifications de format \(à la page 11\)](#).

## Description

COSH calcule le cosinus hyperbolique des valeurs de la plage de données spécifiée. Pour  $x$  en radians, le cosinus hyperbolique d'un nombre est :

$$\cosh(x) = \frac{e^x + e^{-x}}{2}$$

où  $e$  représente le nombre naturel, 2.7182818. COSH renvoie une nouvelle colonne par colonne d'entrée, chacune contenant le cosinus hyperbolique des nombres de la colonne d'entrée correspondante.

 **Remarque** : Si la valeur  $x$  est trop élevée, une erreur de dépassement est renvoyée. Cela se produit si  $\cosh(x)$  dépasse la valeur maximale à virgule flottante sur 32 bits.

## Exemples

TEMP = COSH(0), TEMP = COSH(0, 0) OU TEMP = COSH(0, RADIAN)

Renvoie une colonne nommée TEMP contenant la valeur un.

TEMP = COSH(V1)

Crée une colonne nommée TEMP, où chaque valeur correspond au cosinus hyperbolique (en radians) du contenu de la colonne V1.

TEMP = COSH(V1:V3, 1) OU TEMP = COSH(V1:V3, DEGREE)

Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux cosinus hyperboliques du contenu de la colonne V1, les valeurs de la colonne VX



correspondent aux cosinus hyperboliques du contenu de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux cosinus hyperboliques du contenu de la colonne `V3`. Toutes les valeurs sont en degrés.

```
TEMP = COSH(V1[10:20])
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les cosinus hyperboliques des valeurs des lignes 10 à 20 de la colonne `V1` (en radians). Les autres cibles de `TEMP` sont vides.

```
TEMP = COSH(V1[1:5]:V2)
```

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` correspondent aux cosinus hyperboliques des lignes correspondantes de la colonne `V1` et les valeurs de la colonne `VX` correspondent aux cosinus hyperboliques des lignes correspondantes de la colonne `V2`. Toutes les valeurs sont en radians.

## Fonctions associées

Fonction	Description
<code>ACOS</code>	Calcule l'arc cosinus du contenu de la plage de données spécifiée
<code>COS</code>	Calcule le cosinus du contenu de la plage de données spécifiée
<code>SINH</code>	Calcule le sinus hyperbolique du contenu de la plage de données spécifiée
<code>TANH</code>	Calcule la tangente hyperbolique du contenu de la plage de données spécifiée

## Macro COT

Le `COT` est disponible uniquement dans Unica Campaign.

### Syntaxe

```
COT(data [, units_keyword])
```

### Paramètres

`data`

Valeurs numériques dont la cotangente doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

`units_keyword`

Ce mot clé facultatif détermine si les valeurs en entrée et les résultats sont interprétés comme des degrés ou des radians. Sélectionnez l'une des options suivantes :

`RADIAN` - Effectue les calculs en radians (valeur par défaut)


`DEGREE` - Effectue les calculs en degrés

Si ce paramètre n'est pas spécifié, la valeur par défaut est radians. (Pour convertir des radians en degrés, divisez par `PI` et multipliez par 180.)

Pour plus de détails sur l'utilisation des mots clés dans Unica Campaign, voir [spécifications de format \(à la page 11\)](#).

## Description

`COT` calcule la cotangente des valeurs de la plage de données spécifiée. La cotangente est la réciproque de la tangente. `COT` renvoie une nouvelle colonne par colonne d'entrée, chacune contenant la cotangente des nombres de la colonne d'entrée correspondante.

 **Remarque** : Si une cible contient une valeur dont la tangente est zéro, alors l'arc cotangente est infini. Dans ce cas, `COT` renvoie le plus grand nombre à virgule flottante de 32 bits.

## Exemples

`TEMP = COT(90), TEMP = COT(90, 0) OU TEMP = COT(90, RADIAN)`

Renvoie une colonne nommée `TEMP` contenant la valeur `-0.5`.

`TEMP = COT(0)`

Renvoie une colonne nommée `TEMP` contenant la valeur `MAX_FLOAT_32`.

`TEMP = COT(V1, 1) OU TEMP = COT(V1, DEGREE)`

Crée une colonne nommée `TEMP`, où chaque valeur correspond à la cotangente du contenu (en degrés) de la colonne `V1`.

```
TEMP = COT(V1:V3, 1)
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux cotangentes du contenu de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux cotangentes du contenu de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux cotangentes du contenu de la colonne `V3`. Toutes les valeurs sont en degrés.

```
TEMP = COT(V1[10:20])
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les cotangentes des valeurs des lignes 10 à 20 de la colonne `V1` (en radians). Les autres cibles de `TEMP` sont vides.

```
TEMP = COT(V1[1:5]:V2)
```

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` correspondent aux cotangentes des lignes correspondantes de la colonne `V1` et les valeurs de la colonne `VX` correspondent aux cotangentes des lignes correspondantes de la colonne `V2`. Toutes les valeurs sont en radians.

## Fonctions associées

Fonction	Description
<code>ACOT</code>	Calcule l'arc cotangente du contenu de la plage de données spécifiée
<code>COS</code>	Calcule le cosinus du contenu de la plage de données spécifiée
<code>SIN</code>	Calcule le sinus du contenu de la plage de données spécifiée
<code>TAN</code>	Calcule la tangente du contenu de la plage de données spécifiée

## Macro COUNT

Le `COUNT` est disponible uniquement dans Unica Campaign.

### Syntaxe

```
COUNT(data)
```


## Paramètres

`data`

Plage de cibles dont le nombre de cibles doit être compté. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

`COUNT` compte le nombre de valeurs dans la plage de données spécifiée. `COUNT` renvoie une nouvelle colonne avec une seule valeur qui représente le nombre de cibles contenant des valeurs à l'intérieur de la plage de données spécifiée.

 **Remarque** : Le comptage d'une colonne vide renvoie la valeur zéro.

## Exemples

```
TEMP = COUNT(AVG(V1:V5))
```

Crée une colonne nommée `TEMP` contenant une valeur unique égale à un (la fonction `AVG` renvoie une seule cible dans le mode par défaut).

```
TEMP = COUNT(V1)
```

Crée une colonne nommée `TEMP` contenant une valeur unique indiquant le nombre de cibles contenant des valeurs dans la colonne `V1`.

```
TEMP = COUNT(V1:V3)
```

Crée une colonne nommée `TEMP` contenant une valeur unique indiquant le nombre de cibles contenant des valeurs dans les colonnes `V1`, `V2` et `V3`.

```
TEMP = COUNT(V1[10:20])
```

Crée une colonne nommée `TEMP` contenant la valeur 11 (les plages sont inclusives), à condition que toutes les cibles contiennent des valeurs.

```
TEMP = COUNT(V1[1:5]:V4)
```

Crée une colonne nommée `TEMP` contenant la valeur 20 (5 cibles dans chaque colonne fois 4 colonnes = 20 cibles), à condition que toutes les cibles contiennent des valeurs.

```
TEMP = COUNT(V1[1:10])
```

Crée une colonne nommée `TEMP` contenant la valeur 3, à condition que les lignes 4 à 10 de la colonne `V1` contiennent des valeurs et que les lignes 4 à 10 sont vides.

## Fonctions associées

Fonction	Description
<code>SUM</code> ou <code>TOTAL</code>	Calcule la somme d'une plage de cibles

## Macro `CURRENT_DATE`

Le `CURRENT_DATE` est disponible dans Unica Campaign et Unica Interact.


### Syntaxe

```
CURRENT_DATE([format])
```

### Paramètres

`format`

Un des mots clés du tableau ci-après spécifiant le format de date de `date_string`.

 **Remarque** : Pour plus d'informations sur les formats de date disponibles, voir la section relative aux mots clés de format valide.

### Description


`CURRENT_DATE` renvoie la date du jour dans le format `format`. La date est déterminée par l'horloge sur le serveur . Si aucun mot clé `format` n'est spécifié, la valeur par défaut `DELIM_M_D_Y` est utilisée.

Pour toutes les bases de données recommandées, Unica Campaign tente d'exécuter la macro `CURRENT_DATE` dans la base de données à l'aide d'un appel SQL de l'heure actuelle pris en charge par la base de données (par exemple, `SYSDATE`, `GETDATE`, `DATE` ou `TODAY`). Dans ces cas, tous les paramètres (y compris le format de la date) de cette fonction macro sont ignorés et la sortie inclut tout ce qui est renvoyé par la base de données (par exemple, un composant de temps peut être inclus dans la sortie). Si cela se produit et que vous ne souhaitez renvoyer que la date ou la date dans un format différent, vous pouvez écrire votre

propre macro personnalisée à l'aide d'un script SQL brut ou utiliser d'autres macros . Par exemple :

```
DATE_STRING(CURRENT_JULIAN( ), ...)
```

Dans certains cas, la macro CURRENT\_DATE() est exécutée sur le serveur Unica Campaign (par exemple, lors d'une exécution sur un fichier à plat, sur une base de données non recommandée sans support SQL équivalent ou si l'expression de la macro Campaign ne peut pas être convertie dans la base de données). Dans ces cas, tous les paramètres sont reconnus et la sortie est renvoyée au format sélectionné.

 **Remarque** : Tous les formats disponibles dans Unica Campaign ne sont pas pris en charge par Unica Interact.

Notez que vous devrez peut-être utiliser la macro DATE\_FORMAT afin d'ajuster CURRENT\_DATE pour votre type de base de données. Par exemple, la macro suivante fonctionne avec DB2 :

```
table_name = CURRENT_DATE()-1
```

Toutefois, pour Oracle, vous devez utiliser la macro DATE\_FORMAT comme suit :

```
table_name = DATE_FORMAT(CURRENT_DATE()-1, DELIM_M_D_YYYY, '%Y-%m-%d')
```

## Exemples

Si la date du jour est le 13 septembre 2015, CURRENT\_DATE( ) renvoie "09/13/15".

## Fonctions associées

Fonction	Description
DATE_FORMAT	Convertit les dates d'un format à un autre.
DATE_JULIAN	Renvoie la date du calendrier julien de l'entrée.
DATE_STRING	Renvoie la chaîne de date sous la forme d'une date du calendrier julien.
DATE	Convertit une chaîne de date en date du calendrier julien.

## Macro CURRENT\_DAY

Le `CURRENT_DAY` est disponible dans Unica Campaign et Unica Interact.

### Syntaxe

`CURRENT_DAY( )`

### Description

`CURRENT_DAY` renvoie le jour du mois en cours sous la forme d'un nombre compris entre 1 et 31. La date est déterminée par l'horloge système sur le serveur .

### Exemples

Si la date du jour est le 19 juin, `CURRENT_DAY( )` renvoie le nombre 19.

### Fonctions associées

Fonction	Description
<code>CURRENT_JULIAN</code>	Renvoie le nombre au format Julien de la date du jour.
<code>CURRENT_MONTH</code>	Renvoie le mois actuel sous la forme d'un nombre.
<code>CURRENT_TIME</code>	Renvoie l'heure actuelle sous la forme d'une chaîne.
<code>CURRENT_WEEKDAY</code>	Renvoie le jour de la semaine sous la forme d'un nombre.
<code>CURRENT_YEAR</code>	Renvoie l'année en cours sous la forme d'un nombre.

## Macro CURRENT\_JULIAN

Le `CURRENT_JULIAN` est disponible uniquement dans Unica Campaign.

### Syntaxe

`CURRENT_JULIAN( )`

### Description

`CURRENT_JULIAN( )` renvoie le nombre Julien de la date du jour (nombre de jours écoulés depuis le 1er janvier 0000). Equivalent à la macro `DATE( CURRENT_DATE( ) )`.

## Exemples

Si la date du jour est le 31 août 2000, `CURRENT_JULIAN( )` renvoie le nombre 730729.

## Fonctions associées

Fonction	Description
<code>CURRENT_DAY</code>	Renvoie le jour actuel sous la forme d'un nombre.
<code>CURRENT_MONTH</code>	Renvoie le mois actuel sous la forme d'un nombre.
<code>CURRENT_TIME</code>	Renvoie l'heure actuelle sous la forme d'une chaîne.
<code>CURRENT_WEEKDAY</code>	Renvoie le jour de la semaine sous la forme d'un nombre.
<code>CURRENT_YEAR</code>	Renvoie l'année en cours sous la forme d'un nombre.

## Macro CURRENT\_MONTH

Le `CURRENT_MONTH` est disponible dans Unica Campaign et Unica Interact.

## Syntaxe

`CURRENT_MONTH( )`

## Description

`CURRENT_MONTH` renvoie le mois en cours de l'année sous la forme d'un nombre compris entre 1 et 12.

## Exemples

Si la date du jour est le 19 juin, `CURRENT_MONTH( )` renvoie le nombre 6.

## Fonctions associées

Fonction	Description
<code>CURRENT_DAY</code>	Renvoie le jour actuel sous la forme d'un nombre.
<code>CURRENT_JULIAN</code>	Renvoie le nombre au format Julien actuel.
<code>CURRENT_TIME</code>	Renvoie l'heure actuelle sous la forme d'une chaîne.
<code>CURRENT_WEEKDAY</code>	Renvoie le jour de la semaine sous la forme d'un nombre.
<code>CURRENT_YEAR</code>	Renvoie l'année en cours sous la forme d'un nombre.



## Macro CURRENT\_TIME

Le `CURRENT_TIME` est disponible uniquement dans Unica Campaign.

### Syntaxe

```
CURRENT_TIME( )
```


### Description


`CURRENT_TIME` renvoie l'heure actuelle sous la forme d'une chaîne. L'heure est déterminée par l'horloge système sur le serveur .

## configuration des dates sur votre application Web

Pour afficher correctement les dates sur votre application Web dans les versions actuelles d' Unica Campaign, le fichier de configuration de votre serveur dorsal doit être correctement configuré. Cela est particulièrement important pour les paramètres `dDateFormat` et `DateOutputFormatString` de la base de données qui contient les tables système. Si ces paramètres ne sont pas configurés correctement, les dates ne seront pas affichées non plus correctement dans Campaign. Vous configurez ces propriétés à l'aide d' Platform.

## Définition des dates d'une langue spécifique sur votre application Web


 **Remarque** : Tous les fichiers référencés sont installés par le programme d'installation de l'application Web, sauf indication contraire.

 **Important** : `webapphome` fait référence au répertoire dans lequel l'application Web de Campaign a été installée. `language_code` fait référence aux paramètres de langue que vous choisissez pour votre système.


1. Editez le fichier `webapphome/conf/campaign_config.xml` pour vous assurer que `language_code` est présent dans la liste séparée par des virgules de la balise `<supportedLocales>`, comme illustré ci-dessous :

<supportedLocales>en\_US, language\_code</supportedLocales>

2. Dans le répertoire `webapphome/webapp`, copiez intégralement l'arborescence de répertoire `en_US to language_code` (sensible à la casse).
3. Dans `webapphome/webapp/WEB-INF/classes/resources`, copiez `StaticMessages_en_US.properties` dans `StaticMessages_ language_code.properties`. Copiez également `ErrorMessages_en_US.properties` dans `ErrorMessages_ language_code.properties`.
4. Modifiez le fichier `StaticMessages_ language_code.properties` : recherchez `DatePattern` et changez-le en `DatePattern=dd/MM/yyyy` (la casse doit être respectée).

 **Remarque** : Ce format est défini par Java™. Pour des détails complets sur le format, reportez-vous à la documentation Java de `java.text.SimpleDateFormat`, à l'adresse <http://java.sun.com>. Il n'est pas nécessaire de modifier le fichier `StaticMessages.properties`.

5. Pour WebSphere® : Redémarrez l'application Web.
6. Pour WebLogic : supprimez le module actuel de l'application Web.
  - a. Ajoutez le nouveau module.
  - b. Redéployez l'application Web.
  - c. Le redémarrage du programme d'écoute de Unica Campaign n'est pas nécessaire.
7. Vérifiez que pour le paramètre de langue du navigateur Web, `language_code` est défini sur la première priorité. Pour plus de détails, voir les sections ci-dessous : Configuration de la langue appropriée pour votre navigateur Web et Configuration de votre ordinateur pour l'affichage d'un langage spécifique.

 **Remarque** : Utilisez bien un tiret et non un trait de soulignement dans `language_code`. La configuration de l'application Web est le seul endroit où un tiret est utilisé à la place d'un trait de soulignement.

8. Connectez-vous à Campaign. Dans Campaign, les dates doivent être affichées au format spécifié dans `StaticMessages_ language_code.properties`.

Pour plus d'informations sur la manière de configurer l'heure pour Unica Campaign, voir la documentation Unica Campaign

## Exemples

Si l'heure est 10:54 a.m., `CURRENT_TIME()` renvoie la chaîne "10:54:00 AM".

## Fonctions associées

Fonction	Description
<code>CURRENT_DAY</code>	Renvoie le jour actuel sous la forme d'un nombre.
<code>CURRENT_JULIAN</code>	Renvoie le nombre au format Julien actuel.
<code>CURRENT_WEEKDAY</code>	Renvoie le jour de la semaine sous la forme d'un nombre.
<code>CURRENT_YEAR</code>	Renvoie l'année en cours sous la forme d'un nombre.

## Macro `CURRENT_WEEKDAY`

Le `CURRENT_WEEKDAY` est disponible dans Unica Campaign et Unica Interact.

## Syntaxe

`CURRENT_WEEKDAY()`

## Description

`CURRENT_WEEKDAY` renvoie le jour de la semaine en cours sous la forme d'un nombre compris entre 0 et 6. Dimanche est représenté par 0, lundi par 1, etc.

## Exemples

Si nous sommes vendredi, `CURRENT_WEEKDAY()` renvoie le nombre 5.

## Fonctions associées

Fonction	Description
<code>CURRENT_DAY</code>	Renvoie le jour actuel sous la forme d'un nombre.
<code>CURRENT_JULIAN</code>	Renvoie le nombre au format Julien actuel.
<code>CURRENT_MONTH</code>	Renvoie le mois actuel sous la forme d'un nombre.
<code>CURRENT_TIME</code>	Renvoie l'heure actuelle sous la forme d'une chaîne.
<code>CURRENT_YEAR</code>	Renvoie l'année en cours sous la forme d'un nombre.

# Macro CURRENT\_YEAR

Le `CURRENT_YEAR` est disponible dans Unica Campaign et Unica Interact.

## Syntaxe

`CURRENT_YEAR( )`

## Description

`CURRENT_YEAR` renvoie l'année en cours sous la forme d'un nombre.

## Exemples

Si l'année en cours est 2000, `CURRENT_YEAR( )` renvoie le nombre : 2000.

## Fonctions associées

Fonction	Description
<code>CURRENT_DAY</code>	Renvoie le jour actuel sous la forme d'un nombre.
<code>CURRENT_JULIAN</code>	Renvoie le nombre au format Julien actuel.
<code>CURRENT_MONTH</code>	Renvoie le mois actuel sous la forme d'un nombre.
<code>CURRENT_TIME</code>	Renvoie l'heure actuelle sous la forme d'une chaîne.
<code>CURRENT_WEEKDAY</code>	Renvoie le jour de la semaine sous la forme d'un nombre.
<code>MONTHOF</code>	Renvoie le mois de l'année sous la forme d'un nombre.
<code>WEEKDAYOF</code>	Renvoie le jour de la semaine de la semaine sous forme de nombre.
<code>YEAROF</code>	Renvoie l'année sous la forme d'un nombre.

# DATE

## Syntaxe

`DATE(input_date, [input_date format])`

## Paramètres

`date_string`

Chaîne de texte représentant une date valide.

format

Facultatif ; un des mots clés figurant dans le tableau "Mots clés des formats de date valides", qui spécifie le format de date de `date_string`.

## Description

La macro d'Unica Interact `DATE` convertit une date en entrée en valeur entière de format neutre.

La macro `DATE` est calculée comme suit :  $DATE(X) = 365 +$  le nombre de jours entiers écoulés depuis midi le premier janvier 0001 ap. J.-C. Pour `DATE(X)`, vous pouvez fournir un mot clé facultatif de format `DATE` en entrée afin de spécifier le mode d'analyse de l'élément `DATE` en entrée. Si aucun mot clé `format` n'est spécifié, la valeur par défaut `DELIM_M_D_Y` est utilisée. Pour des informations supplémentaires sur les formats de date valides, voir [Mots clés des formats de date valides \(à la page 21\)](#).

Les formats de date sont de longueur fixe (par exemple, la date Février 28, 1970 est représentée sous la forme 02281970 au format `MMDDYYYY`) ou délimités (par exemple, Février 28, 1970, 2-28-1970, ou 02/28/1970 au format `DELIM_M_D_YY`).

Dans les formats délimités, les délimiteurs sont la barre oblique (/), le tiret (-), l'espace ( ), la virgule (,) et les deux-points (:). Les années peuvent être représentées sur 2 ou 4 chiffres et les mois peuvent être indiqués en toutes lettres (par exemple, Février), abrégés (par exemple, Fév) ou sous forme numérique (par exemple, 2 ou 02).

Pour toutes les années spécifiées sur deux chiffres :

- Par défaut, Unica Interact suppose que les dates à deux chiffres délimitées sont comprises entre 1920 et 2020.
- Les années sur deux chiffres sans le millénaire (la valeur par défaut est 20, mais vous pouvez la modifier à l'aide du paramètre `JVM`) sont considérées comme étant dans les années 2000.
- Les années sur deux chiffres postérieures ou égales au seuil sont considérées comme étant dans les années 1900.

 **Remarque :**

- Tous les `DATE` `Formats` disponibles dans Unica Campaign ne sont pas pris en charge par Unica Interact.
- Pour plus d'informations sur les années sur deux chiffres, voir la rubrique "Mots clés des formats de date valides" de ce guide.
- Pour plus d'informations sur la configuration de la suppression du millénaire pour les années sur deux chiffres, voir la section Arguments JVM dans le guide d'optimisation d'Unica Interact.


Cette macro est disponible dans Unica Interact.

De nombreux systèmes métier utilisent le nombre de jours écoulés au format julien. Le résultat de la macro `DATE()` d'Unica Interact est lié à une date au format julien comme suit :

Format julien = `DATE(...)` + 1,721,059 + fraction du jour écoulée depuis midi.

Voici les valeurs de `DATE()` utiles pour la période A.D. :

- Janvier 1, 2050 A.D renvoie 748,749.
- Janvier 1, 2000 A.D renvoie 730,486
- Janvier 1, 1990 A.D renvoie 726,834
- Janvier 1, 1900 A.D renvoie 693,962
- Janvier 1, 0001 A.D renvoie 365

 **Remarque** : Conformément à la norme ISO 8601 et à la définition de schéma XML pour les objets `Date` et `DateTime`, le calendrier grégorien proleptique est utilisé pour calculer le nombre de jours écoulés. Dans ce système de gestion d'agenda, l'année hypothétique 0000 ap. J.-C. est synonyme de l'année 0001 av. J.-C.

## Exemples

`DATE("8/31/2000")` renvoie le nombre 730,729.

`DATE("8/31/2000",DELIM_MM_DD_YYYY)` renvoie le nombre 730,729.

`DATE("2015-01-01",DELIM_Y_M_D)` renvoie le nombre 735,965.

`DATE("01",DD), DATE("0101",MMDDD)` et `DATE("1970-01-01",DELIM_Y_M_D)` renvoie le nombre 719,529.

## Fonctions associées

Fonction	Description
DATE_FORMAT	Convertit les dates d'un format à un autre.
DATE_JULIAN	Revoie la date du calendrier julien de l'entrée.
DATE_STRING	Revoie la chaîne de date sous la forme d'une date du calendrier julien.
CURRENT_DATE	Revoie la date du jour dans un format spécifié.

## Macro DATE\_FORMAT

Le `DATE_FORMAT` est disponible dans Unica Campaign et Unica Interact.

### Syntaxe

```
DATE_FORMAT(date_string, input_format, output_format)
```

### Paramètres

`date_string`

Texte représentant une date valide.

`input_format`

Un des mots clés du tableau ci-après spécifiant le format de date de `date_string`.

`output_format`

Un des mots clés du tableau ci-après spécifiant le format de date en sortie souhaité.

### Description

`DATE_FORMAT()` convertit une date de `input_format` en un autre format `output_format`.

Si la date est de largeur fixe, elle doit être définie dans l'un des formats suivants :


- DDMMYY[YY]
- DDMMMYY[YY]
- MMDDYY[YY]
- MMMDDYY[YY]

- YY[YY]MMDD
- YY[YY]MMMDD

MM correspond à un mois sur deux chiffres et MMM représente l'abréviation du mois sur trois caractères


Si la date est délimitée (tout délimiteur et notamment SPACE, DASH ou SLASH peut être utilisé), elle doit être définie dans l'un des formats suivants :

- DELIM\_D\_M\_Y
- DELIM\_M\_D\_Y
- DELIM\_Y\_M\_D

 **Remarque** : Tous les formats disponibles dans Unica Campaign ne sont pas pris en charge par Unica Interact.

## Exemples

`DATE_FORMAT("012171", MMDDYY, MMDDYYYY)` renvoie la chaîne "01211971".

 **Remarque** : Pour des informations supplémentaires sur les formats de date valides, voir [DATE \(à la page 69\)](#).

Fonctions associées

Fonction	Description
<code>DATE</code>	Convertit une chaîne de date en date du calendrier julien.
<code>DATE_JULIAN</code>	Renvoie la date du calendrier julien de l'entrée.
<code>DATE_STRING</code>	Renvoie la chaîne de date sous la forme d'une date du calendrier julien.

## Macro DATE\_JULIAN

Le `DATE_JULIAN` est disponible uniquement dans Unica Campaign.

### Syntaxe

`DATE_JULIAN(year, month, day)`



## Paramètres

year

Numéro d'année valide sur 2 ou 4 chiffres.

month

Numéro de mois valide compris entre 1 et 12.

day

Numéro de jour valide compris entre 1 et 31.

## Description

`DATE_JULIAN` envoie la date du calendrier Julien pour l'entrée spécifiée. La date au format julien correspond au nombre de jours écoulés depuis le 1er janvier 0000.

## Exemples

`DATE_JULIAN (2000,08,31)` renvoie le nombre 730729.

## Fonctions associées

Fonction	Description
<code>DATE</code>	Convertit une chaîne de date en date du calendrier julien.
<code>DATE_FORMAT</code>	Convertit les dates d'un format à un autre.
<code>DATE_STRING</code>	Renvoie la chaîne de date sous la forme d'une date du calendrier julien.

## Macro DATE\_STRING

Le `DATE_STRING` est disponible uniquement dans Unica Campaign.

## Syntaxe

```
DATE_STRING(julian_date [, 'output_format'[, max_length]])
```

```
DATE_STRING(julian_date [, 'format_string'[, max_length]])
```

## Paramètres

`julian_date`

Nombre représentant une date au format julien ; nombre de jours écoulés depuis le 1er janvier 0000.

`output_format`

Chaîne ; format de date valide.

`max_length`

`format_string`

Chaîne de format incluant éventuellement une combinaison des codes de format suivants :

Code	Description
%a	Nom abrégé du jour de la semaine
%A	Nom complet du jour de la semaine
%b	Nom abrégé du mois
%B	Nom complet du mois
%c	Représentation de la date et de l'heure conforme aux paramètres régionaux
%d	Jour du mois (01 - 31)
%H	Heure au format 24 heures (00 - 23)
%I	Heure au format 12 heures (01 - 12)
%j	Jour de l'année (001 - 366)
%m	Mois (01 - 12)
%M	Minute (00 - 59)
%p	Indicateur AM/PM des paramètres régionaux Current® pour une horloge au format 12 heures
%S	Secondes (00 - 59)
%U	Semaine de l'année, le dimanche étant considéré comme le premier jour de la semaine (00-51)
%w	Jour de la semaine (0 - 6 ; dimanche correspond à 0)
%W	Semaine de l'année, le lundi étant considéré comme le premier jour de la semaine (00-51)
%x	Représentation de la date pour les paramètres régionaux actuels


Code	Description
%X	Représentation de l'heure pour les paramètres régionaux actuels
%Y	Année sur 2 chiffres (00-99)
%Y	Année sur 4 chiffres. Les zéros de début dans l'année ne sont pas supprimés. Par exemple, l'année 0201 est affichée sous la forme 0201, et l'année 0001 sous la forme 0001.
%4Y	Année sur 4 chiffres. Les zéros de début dans l'année ne sont pas supprimés. Par exemple, l'année 0201 est affichée sous la forme 0201, et l'année 0001 sous la forme 0001.
%z ,	Nom ou abréviation du fuseau horaire ; aucune sortie si le fuseau horaire est inconnu
%Z	
%%	Symbole de pourcentage

## Description

`DATE_STRING` renvoie la chaîne de date sous la forme d'une date du calendrier Julien. Si `output_format` n'est pas fourni, le mot clé par défaut `DELIM_M_D_Y` est utilisé.

## Exemples

`DATE_STRING(730729)` renvoie la chaîne "08/31/00".

 **Remarque** : Pour des informations supplémentaires sur les formats de date valides, voir [DATE \(à la page 69\)](#).

## Fonctions associées

Fonction	Description
<code>DATE</code>	Convertit une chaîne de date en date du calendrier julien.
<code>DATE_JULIAN</code>	Renvoie la date du calendrier julien de l'entrée.
<code>DATE_FORMAT</code>	Convertit les dates d'un format à un autre.

## Macro DAY\_BETWEEN

Le `DAY_BETWEEN` est disponible uniquement dans Unica Campaign.

## Syntaxe

```
DAY_BETWEEN(from_date_string, to_date_string [, input_format])
```

## Paramètres

`from_date_string`

Texte représentant une date valide à partir de laquelle le nombre de jours écoulés doit être calculé.

`to_date_string`

Texte représentant une date valide jusqu'à laquelle le nombre de jours écoulés est calculé. Cette date doit être dans le même format que `from_date_string`.

`input_format`


Un des mots clés du tableau ci-après spécifiant le format de date de `from_date_string` et `to_date_string`.

## Description

`DAY_BETWEEN` renvoie le nombre de jours entre deux dates `from_date_string` et `to_date_string`. Si `input_format` n'est pas fourni, le mot clé par défaut `DELIM_M_D_Y` est utilisé.

## Exemples

`DAY_BETWEEN("08/25/00", "08/31/00")` renvoie le nombre 6.

 **Remarque** : Pour des informations supplémentaires sur les formats de date valides, voir [DATE \(à la page 69\)](#).

## Fonctions associées

Fonction	Description
<code>DAY_FROMNOW</code>	Renvoie le nombre de jours entre le jour actuel et une date spécifiée.
<code>DAY_INTERVAL</code>	Renvoie le nombre de jours entre deux dates spécifiées.

# Macro DAY\_FROMNOW

Le `DAY_FROMNOW` est disponible uniquement dans Unica Campaign.

## Syntaxe

```
DAY_FROMNOW(to_year, to_month, to_day)
```

## Paramètres

`to_year`

Numéro d'année valide sur 2 ou 4 chiffres.

`to_month`


Numéro de mois valide compris entre 1 et 12.

`to_day`

Numéro de jour valide compris entre 1 et 31.

## Description

`DAY_FROMNOW` renvoie le nombre de jours entre la date actuelle et la date spécifiée par `to_year/to_month/to_day..`

 **Remarque** : Si la date spécifiée est antérieure, la valeur renvoyée est négative.

## Exemples

Si la date du jour est le 31 août 2000, `DAY_FROMNOW(2000,12,31)` renvoie le nombre 122.

## Fonctions associées

Fonction	Description
<code>DAY_BETWEEN</code>	Renvoie le nombre de jours entre deux chaînes de date spécifiées.
<code>DAY_INTERVAL</code>	Renvoie le nombre de jours entre deux dates spécifiées.

# Macro DAY\_INTERVAL

Le `DAY_INTERVAL` est disponible uniquement dans Unica Campaign.

## Syntaxe

`DAY_INTERVAL(from_year, from_month, from_day, to_year, to_month, to_day)`

## Paramètres

`from_year`

Numéro d'année valide sur 2 ou 4 chiffres.

`from_month`

Numéro de mois valide compris entre 1 et 12.

`from_day`

Numéro de jour valide compris entre 1 et 31.

`to_year`

Numéro d'année valide sur 2 ou 4 chiffres.

`to_month`

Numéro de mois valide compris entre 1 et 12.

`to_day`

Numéro de jour valide compris entre 1 et 31.

## Description

`DAY_INTERVAL` renvoie le nombre de jours entre la date de début spécifiée (`from_year/`  
`from_month/from_day`) et la date de fin spécifiée (`to_year/to_month/to_day`).

## Exemples

`DAY_INTERVAL(2000,8,31,2000,12,31)` renvoie le nombre 122.

## Fonctions associées

Fonction	Description
<code>DAY_BETWEEN</code>	Renvoie le nombre de jours entre deux chaînes de date spécifiées.
<code>DAY_FROMNOW</code>	Renvoie le nombre de jours entre le jour actuel et une date spécifiée.

# Macro DAYOF

Le `DAYOF` est disponible uniquement dans Unica Campaign.

## Syntaxe

```
DAYOF(date_string [, input_format])
```

## Paramètres

`date_string`

Texte représentant une date valide.

`input_format`


Un des mots clés du tableau ci-après spécifiant le format de date de `date_string`.

## Description

`DAYOF` renvoie le jour du mois sous la forme d'un nombre pour la date représentée par `date_string`. Si `input_format` n'est pas fourni, le mot clé par défaut `DELIM_M_D_Y` est utilisé.

## Exemples

`DAYOF("08/31/00")` renvoie le nombre 31.

 **Remarque** : Pour des informations supplémentaires sur les formats de date valides, voir [DATE \(à la page 69\)](#).

# Macro DISTANCE

Le `DISTANCE` est disponible dans Unica Campaign et Unica Interact.

## Syntaxe

```
DISTANCE(lat1, long1, lat2, long2[, UNIT_OF_MEASURE][, PRECISION])
```

## Paramètres

`lat1`

Latitude du premier point, sous forme de valeur décimale.

`long1`

Longitude du premier point, sous forme de valeur décimale.

`lat2`

Latitude du deuxième point, sous forme de valeur décimale.

`long2`

Longitude du deuxième point, sous forme de valeur décimale.

`UNIT_OF_MEASURE`

Paramètre facultatif indiquant l'unité de mesure de la distance renvoyée. Les valeurs admises sont MILES et KILOMETERS. Si vous omettez ce paramètre, MILES est l'unité par défaut.

`PRECISION`

Paramètre facultatif indiquant le niveau de précision après le séparateur décimal pour la distance renvoyée. Si vous spécifiez une valeur de précision, la valeur de la distance renvoyée est tronquée en fonction du nombre de décimales indiqué. La valeur maximale est 5. Si vous omettez cette valeur, les décimales ne sont pas tronquées.

## Description

`DISTANCE` calcule la distance entre deux points. La latitude et la longitude devraient être représentées par des unités décimales. Utilisez toujours une virgule et un espace pour séparer les valeurs numériques. Cela est obligatoire pour les langages utilisant une virgule comme séparateur décimal, comme le montre le deuxième exemple ci-dessous. Elle est prise en charge pour calculer la distance entre plusieurs points. Si (`lat1`, `long1`) est une liste avec plusieurs valeurs et (`lat2`, `long2`) est également une liste avec plusieurs valeurs, la distance entre le premier point de la liste 1 et du premier point de la liste 2 est calculée comme étant renvoyée en tant que premier élément de la liste des résultats. La distance entre le deuxième point de la liste 1 et le deuxième point de la liste 2 est calculée comme étant renvoyée en tant que second élément de la liste des résultats, et ainsi de suite, jusqu'à ce que tous les éléments de la liste 1 et de la liste 2 soient calculés. Si la liste 1 ne possède



qu'un seul élément, alors que la liste 2 en a plusieurs, les distances entre l'élément de la liste 1 et tous les éléments de la liste 2 sont calculées.

## Exemples

`DISTANCE (18.529747, 73.839798, 18.533511, 73.8777995, MILES, 2)` renvoie la valeur de 2.50 miles.

`DISTANCE (18,529747, 73,839798, 18,533511, 73,8777995, KILOMETERS, 1)` renvoie la valeur de 4,0 kilomètres.

## Macro DIV

Le `DIV` est disponible dans Unica Campaign et Unica Interact.

### Syntaxe

`data DIV divisor data / divisor`

### Paramètres

`data`

Valeurs numériques dans lesquelles la division doit être effectuée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .


`divisor`

Valeurs par lesquelles les valeurs de la plage de données spécifiée doivent être divisées. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans `data2` doit correspondre au nombre de colonnes dans `data1`, à moins que `data2` ne soit une constante. Pour la définition du format de `divisor` (identique à `data`), voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

`DIV` divise la plage de données spécifiée par la valeur du diviseur. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de `data1` divisée par la colonne correspondante de `data2` (à savoir, la première colonne de `data1` est divisée par la première colonne de `data`, la deuxième colonne à la deuxième colonne et ainsi de suite).

Si `data2` est une constante, chaque valeur de `data1` est divisée par cette valeur. Si `data2` contient une ou plusieurs colonnes, les calculs sont effectués ligne par ligne, entre une colonne de `data1` et une colonne de `data2`. La première ligne de `data1` est divisée par la première ligne de `data2`, la seconde ligne, par la seconde ligne, etc. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

 **Remarque** : L'utilisation d'une colonne contenant le même nombre `x` dans chaque ligne que `divisor` revient à utiliser la constante `x` pour `divisor`.

 **Remarque** : L'opérateur `DIV` peut être abrégé en barre oblique (`/`).

## Exemples

`TEMP = 8 DIV 4` OU `TEMP = 8/4`

Crée une nouvelle colonne nommée `TEMP` contenant la valeur deux.

`TEMP = V1/8`

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur correspond au contenu de la colonne `V1` divisé par huit.

`TEMP =V1:V3/2`

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent au contenu de la colonne `V1` divisé par deux, les valeurs de la colonne `VX` correspondent au contenu de la colonne `V2` divisé par deux et les valeurs de la colonne `VY` correspondent au contenu de la colonne `V3` divisé par deux.

`TEMP = V1/V1`

Crée une colonne nommée `TEMP` ne contenant que des uns (tout nombre divisé par lui-même étant égal à un).

`TEMP = V1/V2`

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est la valeur de ligne de la colonne `V1` divisée par la valeur de ligne correspondante de la colonne `V2`.

`TEMP = V1:V3/V4:V6`

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. La colonne `TEMP` contient les valeurs de `V1` divisées par les valeurs de ligne correspondantes de la colonne `V4`. La colonne `VX` contient la division de la colonne `V2` par `V5`. La colonne `VY` contient la division de la colonne `V3` par `V6`.

`TEMP = V1[10:20] / V2` OU `TEMP = V1[10:20] / V2[1:11]`

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent le résultat de la division des valeurs des lignes 10 à 20 de la colonne `V1` par les valeurs des lignes 1 à 11 de la colonne `V2`. Les autres cibles de `TEMP` sont vides.

## Fonctions associées

Fonction	Description
<code>MOD</code>	Calcule le modulo du contenu de la plage de données spécifiée
<code>MULT</code>	Multiplie le contenu de deux plages de données
<code>POW</code>	Calcule une valeur de base élevée aux puissances exponentielles spécifiées

## Macro EQ

Le `EQ` est disponible dans Unica Campaign et Unica Interact.

### Syntaxe

`data1 EQ data2 data1 == data2 (data1 = data2)`

### Paramètres

`data1`

Plage de cibles à comparer. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du

format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

`data2`


Nombres auxquels toutes les valeurs de la colonne spécifiée doivent être comparées. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans `data2` doit correspondre au nombre de colonnes dans `data1`, à moins que `data2` ne soit une constante. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

`EQ` compare les deux plages de données spécifiées, puis renvoie un un si les valeurs sont égales ou un zéro si elles ne le sont pas. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de `data1` comparée à la colonne correspondante de `data2` (à savoir, la première colonne de `data1` est comparée à la première colonne de `data`, la deuxième colonne à la deuxième colonne et ainsi de suite).

Si `data2` est une constante, chaque valeur de `data` est comparée à cette valeur. Si `data2` est une colonne, les calculs sont effectués ligne par ligne. Les valeurs de `data1` sont comparées à la valeur de la première ligne de `data2`, la deuxième ligne à la deuxième ligne et ainsi de suite. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

Lors de la comparaison de chaînes, la casse n'est pas importante ("Oui", "OUI", "oui" et "oul" sont considérées comme étant égales).

 **Remarque** : L'opérateur `EQ` peut être abrégé en double signe égal (`==`). A l'intérieur des parenthèses, un signe égal unique (`=`) peut également être utilisé pour la fonction macro `EQ` (en dehors des parenthèses, e signe égal est interprété comme l'opérateur d'affectation).

## Exemples

`TEMP = 3 EQ 4, TEMP = 3==4 OU TEMP = (3=4)`

Crée une colonne nommée `TEMP` contenant la valeur zéro (car trois n'est pas égal à quatre).

```
TEMP = "No" == "NO"
```

Crée une colonne nommée `TEMP` contenant la valeur un (la casse n'est pas importante dans les comparaisons de chaînes).

```
TEMP = V1 == 8
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est égale à un si la ligne correspondante de la colonne `V1` est égale à huit. Sinon, la valeur est égale à zéro.

```
TEMP = V1==V1
```

Crée une colonne nommée `TEMP` ne contenant que des uns (car tout nombre est égal à lui-même).

```
TEMP = V1==V2
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est la valeur de ligne de la colonne `V1` comparée à la valeur de ligne correspondante de la colonne `V2`.

```
TEMP = V1:V3 == V4:V6
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. La colonne `TEMP` contient les valeurs de `V1` comparées aux valeurs de ligne correspondantes de la colonne `V4`. La colonne `VX` compare les colonnes `V2` et `V5`. La colonne `VY` compare les colonnes `V3` et `V6`.

```
TEMP = V1[10:20] == V2 OU TEMP = V1[10:20] == V2[1:11]
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les résultats de la comparaison des valeurs des lignes 10 à 20 de la colonne `V1` aux lignes 1 à 11 de la colonne `V2`. Les autres cibles de `TEMP` sont vides.

## Fonctions associées

Fonction	Description
EQ	Renvoie TRUE si une plage de données est égale à une autre
GE	Renvoie TRUE si une plage de données est supérieure ou égale à une autre
GT	Renvoie TRUE si une plage de données est supérieure à une autre
LE	Renvoie TRUE si une plage de données est inférieure ou égale à une autre
LT	Renvoie TRUE si une plage de données est inférieure à une autre
NE	Renvoie TRUE si une plage de données n'est pas égale à une autre

# Macro EXP

Le `EXP` est disponible uniquement dans Unica Campaign.

## Syntaxe

`EXP(data)`


## Paramètres

`data`

Valeurs numériques utilisées comme exposant du nombre naturel  $e$ . Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

`EXP` élève le nombre naturel  $e$  de chacune des valeurs de la plage de données spécifiée (il calcule  $e^x$ ). La constante  $e$  est égale à 2,7182818. `EXP` renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant le résultat  $e^x$  pour chaque valeur  $x$  des colonnes en entrée correspondantes. `EXP` est l'inverse de la fonction macro `LN`.

 **Remarque** : Si la valeur  $x$  est trop élevée ou trop faible, une erreur de dépassement est renvoyée. Cela se produit si  $e^x$  dépasse la valeur à virgule flottante sur 32 bits maximale ou minimale.

## Exemples

`TEMP = EXP(2)`

Crée une nouvelle colonne nommée `TEMP` contenant la valeur 7.39.

`TEMP = EXP(V1)`

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur correspond à l'élévation de  $e$  au contenu de la colonne `V1`.

`TEMP = EXP(V1:V3)`

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent à l'élévation de `e` au contenu de la colonne `V1`, les valeurs de la colonne `VX` correspondent à l'élévation de `e` au contenu de la colonne `V2` et les valeurs de la colonne `VY` correspondent à l'élévation de `e` au contenu de la colonne `V3`.

```
TEMP = EXP(V1[10:20])
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les résultats de l'élévation de `e` aux valeurs des lignes 10 à 20 de la colonne `V1`. Les autres cibles de `TEMP` sont vides.

```
TEMP = EXP(V1[1:5]:V2)
```

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` correspondent aux résultats de l'élévation de `e` aux valeurs de ligne correspondantes de la colonne `V1` et les valeurs de la colonne `VX` correspondent aux résultats de l'élévation de `e` aux valeurs de ligne correspondantes de la colonne `V2`.

## Fonctions associées

Fonction	Description
<code>LN</code> ou <code>LOG</code>	Calcule le logarithme naturel du contenu de la plage de données spécifiée
<code>LOG2</code>	Calcule le logarithme base 2 du contenu de la plage de données spécifiée
<code>LOG10</code>	Calcule le logarithme base 10 du contenu de la plage de données spécifiée
<code>POW</code>	Puissance exponentielle

## Macro EXTERNALCALLOUT

La macro `EXTERNALCALLOUT` est disponible uniquement dans Unica Interact.

### Syntaxe

```
EXTERNALCALLOUT( calloutName, arg1, ...)
```

### Paramètres

`calloutName`

Nom de l'appel que vous avez créé à l'aide de l'API ExternalCallout. Ce nom doit correspondre au nom de la catégorie d'appels externes que vous avez créée dans Platform.

`arg1`

Argument requis par votre appel, si nécessaire.

## Description

`EXTERNALCALLOUT` permet d'appeler une application externe pour ajouter des données à votre diagramme temps réel. `EXTERNALCALLOUT` peut renvoyer tout ce pour quoi l'appel a été créé. Vous devez écrire cet appel dans Java à l'aide de l'API ExternalCallout. Pour plus d'informations, reportez-vous au document [Unica Interact - Guide d'administration](#).

## Exemples

```
EXTERNALCALLOUT(getStockPrice, UNCA)
```

Appelle l'appel `getStockPrice` en transmettant le nom de l'action, `UNCA`, comme argument. Cet appel défini par l'utilisateur renvoie le cours de l'action tel qu'il est défini par l'appel.

## Macro INDEXOF

La macro `INDEXOF` est une macro interne, disponible uniquement dans Unica Interact. Cette macro est transmise en tant que paramètre dans la macro `EXTERNALCALLOUT`. La macro est ajoutée à `EXTERNALCALLOUT` au démarrage du serveur. Aucune configuration externe n'est requise pour utiliser cette macro.

## Syntaxe

```
EXTERNALCALLOUT('indexOf', dimension field expression)
```

## Paramètres

`'indexOf'`

`indexOf` est transmis en tant que nom d'appel prédéfini dans la macro `EXTERNALCALLOUT`. Ce paramètre est requis et n'est pas sensible à la casse.

## Expression de champ de dimension



Un argument requis par l'appel 'indexOf'. Les utilisateurs doivent transmettre une condition, qui peut impliquer plusieurs champs de table de dimensions.

### Description

La macro 'indexOf' permet d'interroger plusieurs champs de table de dimensions. Cette macro renvoie la liste des index qui répondent à la condition donnée pour chaque client. Lors de la création d'un diagramme interactif, les utilisateurs peuvent obtenir des enregistrements d'après une expression donnée. Si un nombre incorrect d'arguments lui sont transmis, la macro génère une erreur. Dans le cas d'erreurs de syntaxe, le message d'erreur s'affiche lors de l'exécution du diagramme.

Une vérification de syntaxe ne valide pas ces erreurs.

### Exemples

Pour la table de dimensions **Account\_details**, `EXTERNAL_CALLOUT('IndexOf', Account_details.AccountId>1000 AND account_details.Status=='G')`

AccountId	Equilibrage	Statut
101	1100	G
102	800	G
103	1600	G
104	2100	G

L'expression ci-dessus utilisant 'INDEXOF' renvoie une liste contenant les index 1 et 4.

## Macro FACTORIAL

Le `FACTORIAL` est disponible uniquement dans Unica Campaign.

### Syntaxe

`FACTORIAL(data)`


### Paramètres

data

Valeurs entières dont la factorielle doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs, mais la valeur doit être supérieure ou égale à zéro. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

`FACTORIAL` calcule la factorielle des valeurs de la plage de données spécifiée. Toutes les entrées doivent être des entiers supérieurs ou égaux à zéro. La factorielle d'un nombre entier inférieur ou égal à un est un. Pour les entiers  $x \geq 2$ , la factorielle  $x! = x(x-1)(x-2)\dots(x-1)$ . `FACTORIAL` renvoie une nouvelle colonne par colonne en entrée, chacune contenant la factorielle des nombres de la colonne d'entrée correspondante.

 **Remarque** : Toute valeur supérieure à 34 génère ??? (erreur de dépassement des valeurs à virgule flottante).

## Exemples

```
TEMP = FACTORIAL(3)
```

Crée une nouvelle colonne nommée `TEMP` contenant la valeur 6.

```
TEMP = FACTORIAL(-2)
```

Génère une erreur 333, indiquant que l'argument doit être supérieur ou égal à 0.

```
TEMP = FACTORIAL(V1)
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur correspond à la factorielle du contenu de la colonne `V1`.

```
TEMP = FACTORIAL(V1:V3)
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux factorielles du contenu de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux factorielles du contenu de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux factorielles du contenu de la colonne `V3`.

```
TEMP = FACTORIAL(V1[10:20])
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les factorielles des valeurs des lignes 10 à 20 de la colonne `V1`. Les autres cibles de `TEMP` sont vides.

```
TEMP = FACTORIAL(V1[50:99]:V2)
```

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 50 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` sont les factorielles des lignes de la colonne `V1` et les valeurs de la colonne `VX` sont les factorielles des valeurs de la colonne `V2`.

## Macro FLOOR

Le `FLOOR` est disponible uniquement dans Unica Campaign.

### Syntaxe

```
FLOOR(data)
```


### Paramètres

`data`

Valeurs numériques dont la partie entière doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

### Description

`FLOOR` calcule la partie entière des valeurs de la plage de données spécifiée. La partie entière d'un nombre correspond au plus grand entier inférieur à ce nombre. `FLOOR` renvoie une nouvelle colonne par colonne d'entrée, chacune contenant la partie entière des nombres de la colonne d'entrée correspondante.

 **Remarque** : Cela est équivalent à la fonction macro `INT`.

## Exemples

```
TEMP = FLOOR(4.3)
```

Crée une nouvelle colonne nommée `TEMP` contenant la valeur 4.

```
TEMP = FLOOR(2.9)
```

Crée une nouvelle colonne nommée `TEMP` contenant la valeur -3.

```
TEMP = FLOOR(V1)
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur correspond à la partie entière du contenu de la colonne `V1`.

```
TEMP = FLOOR(V1:V3)
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux parties entières du contenu de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux parties entières du contenu de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux parties entières du contenu de la colonne `V3`.

```
TEMP = FLOOR(V1[10:20])
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les parties entières des valeurs des lignes 10 à 20 de la colonne `V1`. Les autres cibles de `TEMP` sont vides.

```
TEMP = FLOOR(V1[50:99]:V2)
```

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 50 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` sont les parties entières des lignes de la colonne `V1` et les valeurs de la colonne `VX` sont les parties entières des valeurs de la colonne `V2`.

## Fonctions associées

Fonction	Description
<code>CEILING</code>	Calcule le plafond de chaque valeur de la plage de données spécifiée
<code>FRACTION</code>	Renvoie la partie décimale de chaque valeur de la plage de données spécifiée
<code>TRUNCATE</code>	Renvoie la partie non décimale de chaque valeur de la plage de données spécifiée

# Macro FORMAT

Le `FORMAT` est disponible uniquement dans Unica Campaign.

## Syntaxe

`Format` possède deux formes ; l'une pour les données de type numérique et l'autre pour celles de type texte/caractère.

Pour les données de type numérique :

```
FORMAT(colName, width [, precision [, format_type [, alignment [, padding]]]])
```

Pour les données de type texte/caractère :

```
FORMAT(colName, width [, alignment])
```

## Paramètres

`colName`

La macro examine `colName`, détermine son type de données, puis impose les règles appropriées pour les paramètres suivants en conséquence.

`width`

La largeur doit être assez importante pour contenir le résultat complet. Sinon le résultat est tronqué. Les valeurs admises sont comprises entre 1 et 29 si `colName` est numérique et entre 1 et 255 dans le cas contraire.

`precision`

La précision est le nombre de chiffres après le séparateur décimal. Les valeurs admises sont comprises entre 0 et 15. Si la valeur est égale à zéro, le résultat est un entier. La précision par défaut est de 2.

`format_type`

Les mots clés valides pour `format_type` sont :

`PERIOD` Le point (.) est utilisé comme symbole décimal. Aucun séparateur de milliers n'est utilisé. Il s'agit de la valeur par défaut.

`COMMA` La virgule (,) est utilisée comme symbole décimal. Aucun séparateur de milliers n'est utilisé.

`PERIOD_COMMA` Le point est utilisé comme symbole décimal et la virgule comme symbole de groupement des chiffres.

`COMMA_PERIOD` Virgule comme symbole décimal et point comme symbole de groupement des chiffres.

`alignment`

Les mots clés valides pour l'alignement sont `LEFT` et `RIGHT`. La valeur par défaut est `RIGHT` pour les données de type numérique et `LEFT` pour les données de type texte/caractère.

`padding`

Les mots clés valides pour le remplissage sont `SPACE` et `ZERO`. La valeur par défaut est `SPACE`. `ZERO` est ignoré (`SPACE` est utilisé à la place) si l'alignement est `LEFT`.

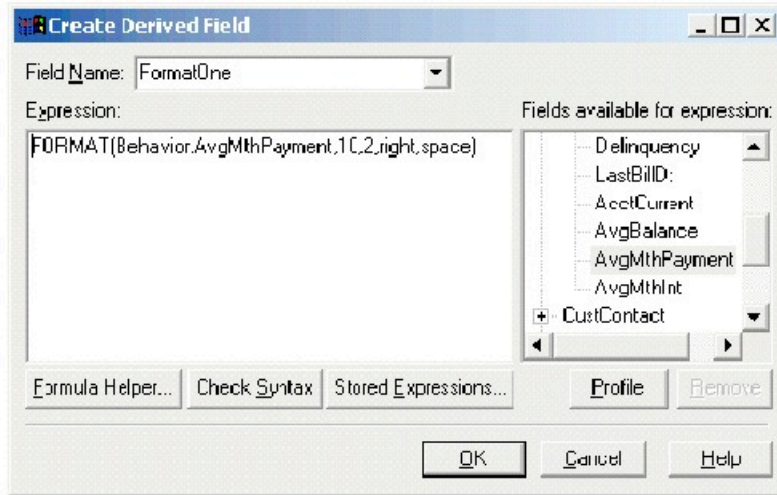
Notez que les chaînes numériques contenues dans des données de type texte/caractère sont traitées comme texte/caractère. Notez également que la forme numérique accepte plusieurs mots clés facultatifs possédant chacun une valeur par défaut. Cependant, pour remplacer la valeur par défaut du deuxième mot clé et des mots clés facultatifs ultérieurs, vous devez obligatoirement codifier les valeurs par défaut pour les mots clés facultatifs précédents (qui deviennent obligatoires). Par exemple : pour remplacer l'alignement par un alignement à gauche (`LEFT`), vous devez coder : `FORMAT(myNumCol, 10, 2, PERIOD, LEFT)`.

## Description

`FORMAT` convertit des données numériques en chaîne avec diverses options de formatage pour contrôler et définir la chaîne en sortie. Cela est particulièrement utile pour créer des fichiers d'instantané avec des formats spécifiques à des fins de publipostage.

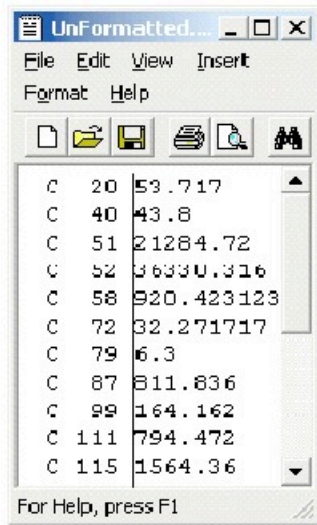
## Exemples

L'exemple ci-après définit une zone dérivée à l'aide de `FORMAT`.

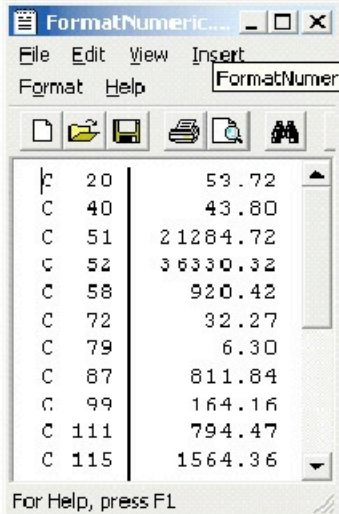


Les exemples ci-après montrent la même zone, `AvgMthPayment`, dans trois formats.

Non formaté :

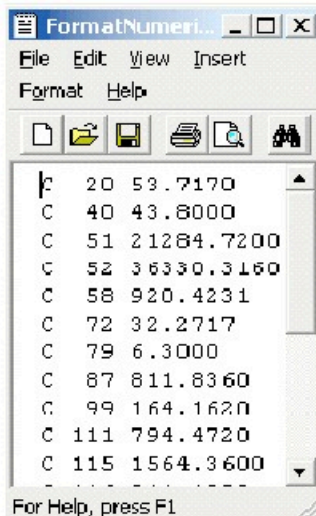


Formaté à l'aide de `FORMAT(Behavior.AvgMthPayment,10,2,right,space)` :



Code	Valeur
f 20	53.72
C 40	43.80
C 51	21284.72
C 52	36330.32
C 58	920.42
C 72	32.27
C 79	6.30
C 87	811.84
C 99	164.16
C 111	794.47
C 115	1564.36

Formaté à l'aide de `FORMAT(Behavior.AvgMthPayment, 10, 4)` :



Code	Valeur
f 20	53.7170
C 40	43.8000
C 51	21284.7200
C 52	36330.3160
C 58	920.4231
C 72	32.2717
C 79	6.3000
C 87	811.8360
C 99	164.1620
C 111	794.4720
C 115	1564.3600

## Macro FRACTION

Le `FRACTION` est disponible uniquement dans Unica Campaign.

### Syntaxe

`FRACTION(data)`




## Paramètres

`data`

Valeurs numériques dont la fraction doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

`FRACTION` calcule la partie décimale des valeurs de la plage de données spécifiée. Pour chaque colonne d'entrée, `FRACTION` renvoie une nouvelle colonne qui contient la partie fractionnaire des nombres de la colonne d'entrée correspondante.

 **Remarque** : Les fonctions macro `FRACTION` et `TRUNCATE` sont complémentaires car leur somme correspond aux valeurs d'origine.

## Exemples

```
TEMP = FRACTION(4.3)
```

Crée une nouvelle colonne nommée `TEMP` contenant la valeur 0.3.

```
TEMP = FRACTION(2.9)
```

Crée une nouvelle colonne nommée `TEMP` contenant la valeur -0.9.

```
TEMP = FRACTION(V1)
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur correspond à la partie décimale du contenu de la colonne `V1`.

```
TEMP = FRACTION(V1:V3)
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux parties décimales du contenu de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux parties décimales du contenu de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux parties décimales du contenu de la colonne `V3`.

```
TEMP = FRACTION(V1[10:20])
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les parties décimales des valeurs des lignes 10 à 20 de la colonne `V1`. Les autres cibles de `TEMP` sont vides.

```
TEMP = FRACTION(V1[50:99]:V2)
```

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 50 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` correspondent aux parties décimales des lignes de la colonne `V1` et les valeurs de la colonne `VX` correspondent aux parties décimales des valeurs de la colonne `V2`.

## Fonctions associées

Fonction	Description
<code>CEILING</code>	Calcule le plafond de chaque valeur de la plage de données spécifiée
<code>FLOOR</code>	Calcule la partie entière de chaque valeur de la plage de données spécifiée
<code>TRUNCATE</code>	Renvoie la partie non décimale de chaque valeur de la plage de données spécifiée

## Macro GE

La macro GE est disponible dans Unica Campaign and Unica Interact.

### Syntaxe

```
data1 GE data2 data1 >= data2
```

### Paramètres

`data1`

Plage de cibles numériques à comparer. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

`data2`

Nombres auxquels toutes les valeurs de la colonne spécifiée doivent être comparées. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une

expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans `data2` doit correspondre au nombre de colonnes dans `data1`, à moins que `data2` ne soit une constante. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

`GE` compare les deux plages de données spécifiées, puis renvoie un un si les valeurs du premier jeu de données sont supérieures ou égales à celles du deuxième jeu de données ou un zéro dans le cas contraire. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de `data1` comparée à la colonne correspondante de `data2` (à savoir, la première colonne de `data1` est comparée à la première colonne de `data`, la deuxième colonne à la deuxième colonne et ainsi de suite).

Si `data2` est une constante, chaque valeur de `data` est comparée à cette valeur. Si `data2` est une colonne, les calculs sont effectués ligne par ligne. Les valeurs de `data1` sont comparées à la valeur de la première ligne de `data2`, la deuxième ligne à la deuxième ligne et ainsi de suite. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

 **Remarque :** L'opérateur `GE` peut être abrégé en signe supérieur suivi d'un signe égal (`>=`).

## Exemples

```
TEMP = 9 GE 4 OU TEMP = 9 >= 4
```

Crée une colonne nommée `TEMP` contenant la valeur un (car neuf est supérieur à quatre).

```
TEMP = V1 >= 8
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est égale à un si la ligne correspondante de la colonne `V1` est supérieure ou égale à huit. Sinon, la valeur est égale à zéro.

```
TEMP = V1:V3 >= 2
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent au contenu de la colonne `V1` comparé à la valeur deux, les valeurs de la colonne `VX` correspondent au contenu de la colonne `V2` comparé à la valeur deux, et les valeurs de la colonne `VY` correspondent au contenu de la colonne `V3` comparé à la valeur deux.

```
TEMP = V1 >= V1
```

Crée une colonne nommée `TEMP` ne contenant que des uns (car tout nombre est égal à lui-même).

```
TEMP = V1 >= V2
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est la valeur de ligne de la colonne `V1` comparée à la valeur de ligne correspondante de la colonne `V2`.

```
TEMP = V1:V3 >= V4:V6
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. La colonne `TEMP` contient les valeurs de `V1` comparées aux valeurs de ligne correspondantes de la colonne `V4`. La colonne `VX` compare les colonnes `V2` et `V5`. La colonne `VY` compare les colonnes `V3` et `V6`.

```
TEMP = V1[10:20] >= V2 OU TEMP = V1[10:20] >= V2[1:11]
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les résultats de la comparaison des valeurs des lignes 10 à 20 de la colonne `V1` aux valeurs des lignes 1 à 11 de la colonne `V2`. Les autres cibles de `TEMP` sont vides.

## Fonctions associées

`NE` Renvoie TRUE si une plage de données n'est pas égale à une autre

## Macro GET

La macro GET est disponible dans Unica Interact.

### Syntaxe

```
GET(<dim field>, index)
```

### Paramètres

`index`

Récupérez l'élément dans l'index spécifié. Lorsqu'il est utilisé dans Interact, il est autorisé à avoir plusieurs index, suite à une autre expression. Dans ce cas, les éléments de `<zone_dim>` spécifiés dans « index » sont extraits. Le ou les index non valides seront ignorés.

```
dim field
```

Vous récupérez l'élément dans l'index spécifié depuis le champ dimensionnel.

## Description

Cette macro est répertoriée sous Toutes les macros prédéfinies Vous pouvez utiliser cette macro pendant que vous concevez un diagramme interactif. Si l'index se situe hors plage, une erreur est renvoyée avec la plage d'index attendue.

 **Remarque** : Cette macro prend en charge l'indexation en base 1.

## Exemple

```
GET(inttest183_interact_pftbl_null.rank,3)
```

# Macro GROUPBY

Le `GROUPBY` est disponible uniquement dans Unica Campaign.

## Syntaxe

```
GROUPBY(group_field, keyword, rolled_field [,output_field])
```

## Paramètres

- `group_field`

Indique la variable sur laquelle les enregistrements sont regroupés (toutes les valeurs identiques de la variable spécifiée sont regroupées ensemble).

- `keyword`

Indique la fonction de cumul récapitulatif à effectuer sur `rolled-field`.

- `rolled_field`

Indique la variable à récapituler ou cumuler.

- `output_field`


Identifie une variable de remplacement à renvoyer pour une ligne d'un groupe et ne pouvant être utilisée qu'avec les mots clés `MinOf`, `MaxOf` et `MedianOf`.

## Description

`GROUPBY` récapitule plusieurs lignes de données dans un groupe La sortie de cette fonction est une colonne unique. La sortie est le résultat de l'opération spécifiée par `keyword` sur la zone `rolled_field` du groupe homogène spécifié par `group_field`. S'il existe plusieurs réponses correspondant à une condition spécifiée, la première détectée est renvoyée.

Si le paramètre facultatif `output_field` n'est pas fourni, la sortie est le résultat de l'opération sur `rolled_field`. Si `output_field` est fourni, le résultat est la zone `output_field` de la ligne dans le groupe.

Si plusieurs lignes d'un groupe répondent à la condition spécifiée (par exemple, il existe des liens pour la valeur max), la zone `output-field` associée à la première ligne remplissant la condition est renvoyée.

 **Remarque** : Pour effectuer un groupement sur plusieurs colonnes, vous pouvez placer entre accolades "{}" une liste de noms de zone, séparés par des virgules, et utiliser cette liste comme premier paramètre de l'appel de macro `GROUPBY`.

Les mots clés pris en charge sont les suivants (casse indifférente) :

### Chaîne ?

Mot clé	Oui/		Description
	Non		
CountOf	Yes		Renvoie le nombre d'enregistrements de chaque groupe ( <code>rolled_field</code> peut être de type numérique ou chaîne ; la valeur renvoyée est la même quelle que soit la valeur de <code>rolled_field</code> ).
MinOf	Yes		Renvoie la valeur minimale de <code>rolled_field</code> dans chaque groupe ( <code>rolled_field</code> peut être de type numérique ou chaîne. Si <code>rolled_field</code> est une chaîne, la valeur la plus proche du début de l'alphabet (pour un tri alphabétique) est renvoyée).
MaxOf	Yes		Renvoie la valeur maximale de <code>rolled_field</code> dans chaque groupe ( <code>rolled_field</code> peut être de type numérique ou chaîne. Si <code>rolled_field</code> est une chaîne, la valeur la plus proche de la fin de l'alphabet (pour un tri alphabétique) est renvoyée).

**Chaîne ?**

Mot clé	Chaîne ?		Description
	Oui/	Non	
DiffOf	Yes		Renvoie le nombre de valeurs distinctes de <code>rolled_field</code> dans chaque groupe ( <code>rolled_field</code> peut être de type numérique ou chaîne).
AvgOf	No		Renvoie la valeur moyenne de <code>rolled_field</code> dans chaque groupe ( <code>rolled_field</code> doit être numérique).
ModeOf	Yes		Renvoie la valeur modale (à savoir, la valeur la plus fréquente) de <code>rolled_field</code> dans chaque groupe ( <code>rolled_field</code> peut être de type numérique ou chaîne).
MedianOf	Yes		Renvoie la valeur médiane (à savoir, la valeur du milieu lors d'un tri en fonction de <code>rolled_field</code> ) de <code>rolled_field</code> dans chaque groupe ( <code>rolled_field</code> peut être de type numérique ou chaîne. Si <code>rolled_field</code> est une chaîne, les valeurs sont triées par ordre alphabétique).
OrderOf	Yes		Renvoie l'ordre de <code>rolled_field</code> dans chaque groupe ( <code>rolled_field</code> doit être numérique). Si plusieurs enregistrements ont la même valeur, ils reçoivent tous la même valeur.
SumOf	No		Renvoie la somme de <code>rolled_field</code> dans chaque groupe ( <code>rolled_field</code> doit être numérique).
StdevOf	No		Renvoie l'écart type de <code>rolled_field</code> dans chaque groupe ( <code>rolled_field</code> doit être numérique).
IndexOf	Yes		Renvoie l'index de base 1 (trié par <code>rolled_field</code> ) de chaque enregistrement ( <code>rolled_field</code> peut être de type numérique ou chaîne). L'ordre de tri est croissant.
RankOf	Yes		Renvoie la catégorie de base 1 (triée par <code>rolled_field</code> ) dans laquelle se trouve chaque enregistrement ( <code>rolled_field</code> peut être de type numérique ou chaîne). L'ordre de tri est croissant.
			Remarque : pour les zones numériques, l'ordre de tri de RankOf et IndexOf peut être décroissant, si l'on place un signe moins (-) devant la zone de tri.
			Remarque : pour les zones numériques, l'ordre de tri de RankOf et IndexOf peut être décroissant, si l'on place un signe moins (-) devant la zone de tri.

## Exemples

```
GROUPBY (Household_ID, SumOf, Account_Balance)
```

Calcule la somme de tous les soldes de compte par ménage.

```
GROUPBY (Cust_ID, MinOf, Date(Account_Open_Date), Acc_Num)
```

Renvoie le numéro de compte du premier compte ouvert par le client.

## Macro GROUPBY\_WHERE

Le `GROUPBY_WHERE` est disponible uniquement dans Unica Campaign.

### Syntaxe

```
GROUPBY_WHERE(group_field, keyword, rolled_field, where_value [,output_field])
```

### Paramètres

- `group_field`

Indique la variable sur laquelle les enregistrements sont regroupés (toutes les valeurs identiques de la variable spécifiée sont regroupées ensemble).

- `keyword`

Indique la fonction de cumul récapitulatif à effectuer.

- `rolled_field`

Indique la variable à récapituler ou cumuler.

- `where_value`

Expression convertie en un ou zéro qui indique les lignes à inclure dans l'opération de cumul.

- `output_field`


Identifie une variable de remplacement à renvoyer pour une ligne d'un groupe et ne pouvant être utilisée qu'avec les mots clés `MinOf`, `MaxOf` et `MedianOf`.



## Description

`GROUPBY_WHERE` récapitule des lignes de données spécifiques dans un groupe. La sortie de cette fonction est une colonne unique. La sortie est le résultat de l'opération spécifiée par `keyword` sur la zone `rolled_field` du groupe homogène spécifié par `group_field` et filtré par `where_value`. Seules les lignes pour lesquelles `where_value` contient la valeur un sont incluses dans le calcul.

Si le paramètre facultatif `output_field` n'est pas fourni, la sortie est le résultat de l'opération sur `rolled_field`. Si `output_field` est fourni, le résultat est la zone `output_field` de la ligne dans le groupe.

 **Remarque :** Pour plus d'informations sur les valeurs valides de [Macro GROUPBY \(à la page 102\)](#), voir `keyword`.

## Exemples

```
GROUPBY_WHERE (Household_ID, SumOf, Account_Balance, Account_Balance>0)
```

Calcule la somme de tous les comptes dont les soldes sont positifs pour chaque ménage.

```
GROUPBY_WHERE (Cust_ID, AvgOf, Purchase_Amt, Date(Current_Date) -
Date(Purchase_Date)<90)
```

Calcule le montant des achats moyen de chaque client pour les achats au cours des 90 derniers jours.

## Macro GT

Le `GT` est disponible dans Unica Campaign et Unica Interact.

### Syntaxe

```
data1 GT data2 data1 > data2
```

### Paramètres

```
data1
```

Plage de cibles numériques à comparer. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .


`data2`

Nombres auxquels toutes les valeurs de la colonne spécifiée doivent être comparées. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans `data2` doit correspondre au nombre de colonnes dans `data1`, à moins que `data2` ne soit une constante. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

`GT` compare les deux plages de données spécifiées, puis renvoie un un si les valeurs du premier jeu de données sont supérieures à celles du deuxième jeu de données ou un zéro dans le cas contraire. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de `data1` comparée à la colonne correspondante de `data2` (à savoir, la première colonne de `data1` est comparée à la première colonne de `data`, la deuxième colonne à la deuxième colonne et ainsi de suite).

Si `data2` est une constante, chaque valeur de `data` est comparée à cette valeur. Si `data2` est une colonne, les calculs sont effectués ligne par ligne. Les valeurs de `data1` sont comparées à la valeur de la première ligne de `data2`, la deuxième ligne à la deuxième ligne et ainsi de suite. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

 **Remarque** : L'opérateur `GT` peut être abrégé en signe supérieur (`>`).

## Exemples

`TEMP = 3 GT 4 OU TEMP = 3 > 4`

Crée une colonne nommée `TEMP` contenant la valeur zéro (car trois n'est pas supérieur à quatre).

`TEMP = V1 > 8`

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est égale à un si la ligne correspondante de la colonne `V1` est supérieure à huit. Sinon, la valeur est égale à zéro.

```
TEMP = V1:V3 > 2
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent au contenu de la colonne `V1` comparé à la valeur deux, les valeurs de la colonne `VX` correspondent au contenu de la colonne `V2` comparé à la valeur deux, et les valeurs de la colonne `VY` correspondent au contenu de la colonne `V3` comparé à la valeur deux.

```
TEMP = V1 > V1
```

Crée une colonne nommée `TEMP` ne contenant que des zéros (car aucun nombre n'est supérieur à lui-même).

```
TEMP = V1 > V2
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est la valeur de ligne de la colonne `V1` comparée à la valeur de ligne correspondante de la colonne `V2`.

```
TEMP = V1:V3 > V4:V6
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. La colonne `TEMP` contient les valeurs de `V1` comparées aux valeurs de ligne correspondantes de la colonne `V4`. La colonne `VX` compare les colonnes `V2` et `V5`. La colonne `VY` compare les colonnes `V3` et `V6`.

```
TEMP = V1[10:20] > V2 OU TEMP = V1[10:20] > V2[1:11]
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les résultats de la comparaison des valeurs des lignes 10 à 20 de la colonne `V1` aux valeurs des lignes 1 à 11 de la colonne `V2`. Les autres cibles de `TEMP` sont vides.

## Fonctions associées

Fonction	Description
<code>EQ</code>	Renvoie TRUE si une plage de données est égale à une autre
<code>GE</code>	Renvoie TRUE si une plage de données est supérieure ou égale à une autre
<code>LE</code>	Renvoie TRUE si une plage de données est inférieure ou égale à une autre
<code>LT</code>	Renvoie TRUE si une plage de données est inférieure à une autre
<code>NE</code>	Renvoie TRUE si une plage de données n'est pas égale à une autre

# Macro IF

Le `IF` est disponible dans Unica Campaign et Unica Interact.

## Syntaxe

```
IF(predicate_col, then_value) IF(predicate_col, then_value, else_value)
```

## Paramètres

`predicate_col`

Colonne de valeurs booléennes ou expression convertie en une colonne de valeurs booléennes. Les valeurs booléennes sont interprétées comme des nulles ou non nulles. Cette colonne doit contenir au moins autant de lignes que la plage de données de laquelle les données sont extraites.

`then_value`

Valeurs à renvoyer si la ligne de `predicate_col` correspondante contient une valeur non nulle. Il peut s'agir d'une valeur constante, d'une colonne ou d'une expression convertie comme l'une de ces valeurs. Pour la définition de format de `then_value` (identique à `data`), voir [Paramètres des fonctions macro pour Unica Campaign \(à la page 11\)](#).

`else_value`


Si ce paramètre facultatif est fourni, il est renvoyé si la ligne correspondante de `predicate_col` contient un zéro. Il peut s'agir d'une valeur constante, d'une colonne ou d'une expression convertie comme l'une de ces valeurs. Si `else_value` n'est pas fourni, un zéro est renvoyé chaque fois que `predicate_col` a la valeur `false`. Pour la définition de format de `else_value` (identique à `data`), voir [Paramètres des fonctions macro pour Unica Campaign \(à la page 11\)](#).

## Description

`IF` convertit l'expression de `predicate_col` et renvoie `then_value` si l'expression est `true` ou `else_value` si l'expression est `false`. Il renvoie le même nombre de colonnes dans `then_value` et `else_value`. Les nouvelles colonnes contiendront les valeurs de ligne correspondantes de `then_value` si la valeur de `predicate_col` est non nulle. Si `else_value`

est fourni, il est renvoyé lorsque la valeur de `predicate_col` est zéro. Si `else_value` n'est pas fourni, zéro est renvoyé.

`IF` fonctionnant ligne par ligne, il génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte (c'est-à-dire la colonne la plus courte entre `predicate_col`, `then_value` et `else_value`).

 **Remarque** : En général, une colonne de prédicat est créée à l'aide de l'une des fonctions macro de comparaison (par exemple, `==`, `>`, `<`, `ISEVEN`, `ISODD`, etc.).

## Exemples

```
TEMP = IF(1, V1)
```

Crée une colonne nommée `TEMP` contenant une copie de la colonne `V1`.

```
TEMP = IF(V1, 1, 0)
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est égale à un si la valeur correspondante de la colonne `V1` est non nulle. Sinon, la valeur est égale à zéro.

```
TEMP = IF(V3, V1, V2)
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est copiée à partir de la colonne `V1` si la valeur correspondante de la colonne `V3` est non nulle. Sinon, la valeur est copiée à partir de la colonne `V2`.

```
TEMP = IF(ABS(V1-AVG(V1)) < STDV(V1), V1)
```

Crée une colonne nommée `TEMP` contenant chaque valeur de la colonne `V1` qui est inférieure d'un écart type de la moyenne.

```
TEMP = IF(V3[20:30], V1[30:40], V2)
```

Crée une colonne nommée `TEMP` contenant les valeurs des lignes 10 à 20. Chaque valeur est copiée à partir de la colonne `V1` (cibles 10 à 20) si la valeur correspondante de la colonne `V3` (cibles 30 à 40) est non nulle. Sinon, la valeur est copiée à partir de la colonne `V2` (cibles 1 à 11).

# Macro IN

Le `IN` est disponible dans Unica Campaign et Unica Interact.

## Syntaxe

```
valuet IN (value1 AND value2 . . . .) or valuet IN subquery
```


## Paramètres


La première forme autorise l'utilisation d'une liste de valeurs à la place d'une sous-requête.

La seconde forme utilise une sous-requête qui est convertie pour générer un résultat intermédiaire, sur lequel un traitement supplémentaire peut être effectué.

## Description

Le prédicat `IN` permet d'utiliser une liste de valeurs au lieu d'une sous-requête ou lance une sous-requête.

 **Remarque** : Le prédicat `IN` possède une version négative : `NOT IN`. Son format est identique à celui de `IN`. `NOT IN` est vrai uniquement si la valeur fournie est introuvable dans les valeurs renvoyées par la sous-requête.

 **Important** : Si vous utilisez `IN` dans Unica Interact, vous ne pouvez utiliser que la syntaxe `value IN (value1 AND value2 . . . .)`.

## Exemples

```
TEMP = IN(25, COLUMN(1...10))
```

Renvoie les colonnes spécifiées d'une plage de données

```
TEMP = IN("cat", COLUMN("cat", "dog", "bird"))
```

Crée une nouvelle colonne nommée `TEMP` contenant la valeur un.

```
TEMP = IN(V1, V1)
```

Crée une colonne nommée `TEMP` ne contenant que des uns.

```
TEMP = IN(V1, V2)
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est égale à un si la ligne correspondante de la colonne `V1` contient une valeur dans la colonne `V2` et à zéro dans le cas contraire.

## Macro INT

Le `INT` est disponible uniquement dans Unica Campaign.

### Syntaxe

```
INT(data)
```

### Paramètres

`data`

Valeurs numériques à arrondir en valeurs entières. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

### Description

`INT` calcule l'entier le plus grand inférieur aux valeurs (également appelé seuil) de la plage de données spécifiée. `INT` renvoie une nouvelle colonne par colonne d'entrée, chacune contenant la partie entière des nombres de la colonne d'entrée correspondante.

 **Remarque** : Cela est équivalent à la fonction macro `FLOOR`.

### Exemples

```
TEMP = INT(4.7)
```

Crée une nouvelle colonne nommée `TEMP` contenant la valeur 4.

```
TEMP = INT(-1.5)
```

Crée une nouvelle colonne nommée `TEMP` contenant la valeur -2.

```
TEMP = INT(V1)
```

Crée une colonne nommée `TEMP`, où chaque valeur correspond à l'entier le plus grand inférieur ou égal au contenu de la colonne `V1`.

```
TEMP = V1 - INT(V1)
```

Crée une colonne nommée `TEMP` contenant la partie décimale de chaque valeur de la colonne `V1`.

```
TEMP = INT(V1:V3)
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux entiers les plus grands inférieurs ou égaux au contenu de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux entiers les plus grands inférieurs ou égaux au contenu de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux entiers les plus grands inférieurs ou égaux au contenu de la colonne `V3`.

```
TEMP = INT(V1[10:20])
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les entiers les plus grands inférieurs ou égaux aux valeurs correspondantes des lignes 10 à 20 de la colonne `V1`. Les autres cibles de `TEMP` sont vides.

```
TEMP = INT(V1[1:5]:V2)
```

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` correspondent aux entiers les plus grands inférieurs ou égaux aux valeurs de ligne correspondantes de la colonne `V1` et les valeurs de la colonne `VX` correspondent aux entiers les plus grands inférieurs ou égaux aux valeurs de ligne correspondantes de la colonne `V2`.

## Fonctions associées

Fonction	Description
<code>ROUND</code>	Calcule la valeur arrondie du contenu de la plage de données spécifiée
<code>TRUNCATE</code>	Renvoie la partie non décimale de chaque valeur de la plage de données spécifiée

## Macro INVERSE

Le `INVERSE` est disponible uniquement dans Unica Campaign.



## Syntaxe

`INVERSE(data)`


## Paramètres

`data`

Valeurs numériques dont l'inverse doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

`INVERSE` calcule la valeur négative des valeurs de la plage de données spécifiée. Il renvoie -x (les valeurs négatives sont renvoyées comme valeurs positives et les valeurs positives sont renvoyées comme valeurs négatives). Pour chaque colonne d'entrée, `INVERSE` renvoie une nouvelle colonne qui contient l'opposé des valeurs de la colonne d'entrée correspondante.

 **Remarque** : Pour inverser une valeur ou une colonne, précédez-la d'un signe moins (-). Par exemple,  $v2 = -v1$  est identique à  $v2 = \text{INVERSE}(v1)$ .

## Exemples

`TEMP = INVERSE(3.2)`

Crée une nouvelle colonne nommée `TEMP` contenant la valeur -3.2.

`TEMP = INVERSE(V1)`

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur correspond à la valeur négative des valeurs de la colonne `V1`.

`TEMP = INVERSE(V1:V3)`

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux valeurs négatives des valeurs de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux valeurs négatives des valeurs de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux valeurs négatives des valeurs de la colonne `V3`.

`TEMP = INVERSE(V1[10:20])`

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les valeurs négatives des valeurs des lignes 10 à 20 de la colonne `V1`. Les autres cibles de `TEMP` sont vides.

```
TEMP = INVERSE(V1[1:5]:V2)
```

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` correspondent aux valeurs négatives des valeurs des lignes correspondantes de la colonne `V1` et les valeurs de la colonne `VX` correspondent aux valeurs négatives des valeurs des lignes correspondantes de la colonne `V2`.

## Fonctions associées

Fonction	Description
<code>ABS</code>	Calcule la valeur absolue du contenu de la plage de données spécifiée
<code>NOT</code>	Calcule le non logique (NON) du contenu de la plage de données spécifiée
<code>SIGN</code>	Détermine le signe (positif ou négatif) des valeurs dans la plage de données spécifiée

## Macro IS

Le `IS` est disponible dans Unica Campaign et Unica Interact.

### Syntaxe

```
IS <keyword>
```

### Paramètres

`keyword`

Condition de recherche, généralement "NULL," "TRUE," "UNKNOWN" et "FALSE."

### Description

`IS` est utilisé dans les conditions de recherche complexes. Plus la recherche est complexe, plus la condition `IS` peut s'avérer utile. Les conditions de recherche booléenne de ce type permettent d'améliorer de simples conditions de recherche.

IS renvoie des résultats différents dans Unica Interact et Unica Campaign. NULL renvoie 1 s'il existe au moins une valeur NULL pour un ID audience. UNKNOWN renvoie 1 pour un ID audience s'il ne possède pas de valeur.

## Macro ISERROR

Le `ISERROR` est disponible uniquement dans Unica Campaign.

### Syntaxe

```
ISERROR(data)
```

### Paramètres

`data`

Valeurs à tester pour vérifier si l'une des lignes contient une erreur (à savoir, une cible ???). Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

### Description

`ISERROR` vérifie si l'une des cibles de chaque ligne de la plage de données spécifiée contient une erreur (à savoir, une cible ???). Il renvoie une nouvelle colonne, chaque ligne contenant un un si la ligne correspondante de `data` contient une erreur. Sinon, la ligne contient un zéro. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus longue.

### Exemples

```
TEMP = ISERROR(-3)
```

Crée une nouvelle colonne nommée `TEMP` contenant la valeur zéro.

```
TEMP = ISERROR(V1)
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est égale à un si la ligne correspondante de la colonne `V1` contient `???` et à zéro dans le cas contraire.

```
TEMP = ISERROR(V1:V3)
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est égale à un si l'une des cibles des lignes correspondantes des colonnes `V1 - V3` contient `???` et à zéro dans le cas contraire.

```
TEMP = ISERROR(V1[50:100]:V10)
```

Crée une colonne nommée `TEMP`, avec les valeurs des lignes 1 à 50. Chaque valeur est égale à un si l'une des cibles des lignes 50 à 100 des colonnes `V1` à `V10` contient `???` et à zéro dans le cas contraire.

## Macro ISODD

Le `ISODD` est disponible uniquement dans Unica Campaign.

### Syntaxe

```
ISODD(data)
```


### Paramètres

`data`

Valeurs numériques à tester pour vérifier si elles sont impaires. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

### Description

`ISODD` teste chaque valeur du jeu de données spécifié pour vérifier si elle est impaire. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant un un pour toutes les valeurs impaires (à savoir, le modulo deux de la valeur est un) ou un zéro pour toutes les valeurs non impaires (paires).

 **Remarque :** Pour les valeurs autres que des entiers, la fonction macro `INT` est appliquée au préalable. Par exemple, `ISODD(2.5) = 0`, car 2 est impair.

## Exemples

```
TEMP = ISODD(-3)
```

Crée une nouvelle colonne nommée `TEMP` contenant la valeur un.

```
TEMP = ISODD(V1)
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur correspond au résultat du test de valeur impaire du contenu de la colonne `V1`.

```
TEMP = ISODD(V1:V3)
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux résultats du test de valeur impaire du contenu de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux résultats du test de valeur impaire du contenu de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux résultats du test de valeur impaire du contenu de la colonne `V3`.

```
TEMP = ISODD(V1[10:20])
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les résultats du test de valeur impaire des valeurs des lignes 10 à 20 de la colonne `V1`. Les autres cibles de `TEMP` sont vides.

```
TEMP = ISODD(V1[1:5]:V2)
```

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` correspondent aux résultats du test de valeur impaire des lignes correspondantes de la colonne `V1` et les valeurs de la colonne `VX` correspondent aux résultats du test de valeur impaire des lignes correspondantes de la colonne `V2`.

## Fonctions associées

### Fonction

### Description

`ISEVEN` Teste si les valeurs en entrée sont paires (à savoir, divisible par deux)

# Macro ISEVEN

Le `ISEVEN` est disponible uniquement dans Unica Campaign.

## Syntaxe

`ISEVEN(data)`


## Paramètres

`data`

Valeurs numériques à tester pour vérifier si elles sont paires. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

`ISEVEN` teste chaque valeur du jeu de données spécifié pour vérifier si elle est paire. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant un un pour toutes les valeurs paires (à savoir, le modulo deux de la valeur est zéro) ou un zéro pour toutes les valeurs non paires (impaires).

 **Remarque** : Pour les valeurs autres que des entiers, la fonction macro `INT` est appliquée au préalable. Par exemple, `ISEVEN(2.5) = 1`, car 2 est pair.

## Exemples

`TEMP = ISEVEN(-3)`

Crée une nouvelle colonne nommée `TEMP` contenant la valeur zéro.

`TEMP = ISEVEN(V1)`

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur correspond au résultat du test de valeur paire du contenu de la colonne `V1`.

`TEMP = ISEVEN(V1:V3)`

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux résultats du test de valeur paire du contenu de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux résultats du test de valeur paire du contenu de la colonne `V2` et les

valeurs de la colonne `VY` correspondent aux résultats du test de valeur paire du contenu de la colonne `V3`.

```
TEMP = ISEVEN(V1[10:20])
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les résultats du test de valeur paire des valeurs des lignes 10 à 20 de la colonne `V1`. Les autres cibles de `TEMP` sont vides.

```
TEMP = ISEVEN(V1[1:5]:V2)
```

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` correspondent aux résultats du test de valeur paire des lignes correspondantes de la colonne `V1` et les valeurs de la colonne `VX` correspondent aux résultats du test de valeur paire des lignes correspondantes de la colonne `V2`.

## Fonctions associées

Fonction	Description
<code>ISODD</code>	Teste si les valeurs en entrée sont impaires (à savoir, non divisible par deux)

## Macro ISODD

Le `ISODD` est disponible uniquement dans Unica Campaign.

### Syntaxe

```
ISODD(data)
```


### Paramètres

`data`

Valeurs numériques à tester pour vérifier si elles sont impaires. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

`ISODD` teste chaque valeur du jeu de données spécifié pour vérifier si elle est impaire. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant un un pour toutes les valeurs impaires (à savoir, le modulo deux de la valeur est un) ou un zéro pour toutes les valeurs non impaires (paires).

 **Remarque** : Pour les valeurs autres que des entiers, la fonction macro `INT` est appliquée au préalable. Par exemple, `ISODD(2.5) = 0`, car 2 est impair.

## Exemples

```
TEMP = ISODD(-3)
```

Crée une nouvelle colonne nommée `TEMP` contenant la valeur un.

```
TEMP = ISODD(V1)
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur correspond au résultat du test de valeur impaire du contenu de la colonne `V1`.

```
TEMP = ISODD(V1:V3)
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux résultats du test de valeur impaire du contenu de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux résultats du test de valeur impaire du contenu de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux résultats du test de valeur impaire du contenu de la colonne `V3`.

```
TEMP = ISODD(V1[10:20])
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les résultats du test de valeur impaire des valeurs des lignes 10 à 20 de la colonne `V1`. Les autres cibles de `TEMP` sont vides.

```
TEMP = ISODD(V1[1:5]:V2)
```

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` correspondent aux résultats du test de valeur impaire des lignes correspondantes de la colonne `V1` et les valeurs de la colonne `VX` correspondent aux résultats du test de valeur impaire des lignes correspondantes de la colonne `V2`.



## Fonctions associées

### Fonction

### Description

`ISEVEN` Teste si les valeurs en entrée sont paires (à savoir, divisible par deux)

## Macro LE

Le `LE` est disponible dans Unica Campaign et Unica Interact.

### Syntaxe

```
data1 LE data2 data1 <= data2
```

### Paramètres

`data1`

Plage de cibles numériques à comparer. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

`data2`

Nombres auxquels toutes les valeurs de la colonne spécifiée doivent être comparées. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

### Description

`LE` compare les deux plages de données spécifiées, puis renvoie un un si les valeurs du premier jeu de données sont inférieures ou égales à celles du deuxième jeu de données ou un zéro dans le cas contraire. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de `data1` comparée à la colonne correspondante de `data2` (à savoir, la première colonne de `data1` est comparée à la première colonne de `data`, la deuxième colonne à la deuxième colonne et ainsi de suite).

Si `data2` est une constante, chaque valeur de `data` est comparée à cette valeur. Si `data2` est une colonne, les calculs sont effectués ligne par ligne. Les valeurs de `data1` sont comparées à la valeur de la première ligne de `data2`, la deuxième ligne à la deuxième ligne et ainsi de suite. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

 **Remarque :** L'opérateur `LE` peut être abrégé en signe inférieur suivi d'un signe égal (`<=`).

## Exemples

```
TEMP = 4 LE 4 OU TEMP = 4 <= 4
```

Crée une colonne nommée `TEMP` contenant la valeur un (car quatre est égal à lui-même).

```
TEMP = V1 <= 8
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est égale à un si la ligne correspondante de la colonne `V1` est inférieure ou égale à huit. Sinon, la valeur est égale à zéro.

```
TEMP = V1:V3 <= 2
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent au contenu de la colonne `V1` comparé à la valeur deux, les valeurs de la colonne `VX` correspondent au contenu de la colonne `V2` comparé à la valeur deux, et les valeurs de la colonne `VY` correspondent au contenu de la colonne `V3` comparé à la valeur deux.

```
TEMP = V1 <= V1
```

Crée une colonne nommée `TEMP` ne contenant que des uns (car tout nombre est égal à lui-même).

```
TEMP = V1 <= V2
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est la valeur de ligne de la colonne `V1` comparée à la valeur de ligne correspondante de la colonne `V2`.

```
TEMP = V1[10:20] <= V2 OU TEMP = V1[10:20] <= V2[1:11]
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les résultats de la comparaison des valeurs des lignes 10 à 20 de la colonne `V1` aux valeurs des lignes 1 à 11 de la colonne `V2`. Les autres cibles de `TEMP` sont vides.

## Fonctions associées

Fonction	Description
EQ	Renvoie TRUE si une plage de données est égale à une autre
GE	Renvoie TRUE si une plage de données est supérieure ou égale à une autre
GT	Renvoie TRUE si une plage de données est supérieure à une autre
LT	Renvoie TRUE si une plage de données est inférieure à une autre
NE	Renvoie TRUE si une plage de données n'est pas égale à une autre

## Macro LIKE

Le `LIKE` est disponible dans Unica Campaign et Unica Interact.

### Syntaxe

```
data1 [NOT] LIKE data2
```

### Paramètres

`data1`

Plage de cibles à comparer. Il peut s'agir d'une chaîne de texte ou d'une expression convertie en chaîne de texte. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

`data2`

Masque textuel auquel toutes les valeurs de la colonne spécifiée doivent être comparées. Il peut s'agir d'une chaîne de texte ou d'une expression convertie en chaîne de texte. Le nombre de colonnes dans `data2` doit correspondre au nombre de colonnes dans `data1`, à moins que `data2` ne soit une constante. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .


Un trait de soulignement (`_`) dans `data2` représente un caractère générique correspondant à tout caractère de `data1`. Un signe pourcentage (`%`) correspond à zéro, un ou plusieurs caractères de `data1`.

## Description

`LIKE` compare les deux plages de données spécifiées, puis renvoie un un si les chaînes correspondent ou un zéro dans le cas contraire. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de `data1` comparée à la colonne correspondante de `data2` (à savoir, la première colonne de `data1` est comparée à la première colonne de `data2`, la deuxième colonne à la deuxième colonne et ainsi de suite).

Si `data2` est une constante de chaîne, chaque chaîne de `data1` est comparée à cette chaîne. Si `data2` est une colonne, les calculs sont effectués ligne par ligne. La chaîne de la première ligne de `data1` est comparée à la chaîne de la première ligne de `data2`, la deuxième ligne à la deuxième ligne et ainsi de suite. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière chaîne de la colonne la plus courte.

Lors de la comparaison de chaînes, la casse n'est pas importante ("Oui", "OUI", "oui" et "oul" sont considérées comme étant égales).

 **Remarque** : La macro `LIKE` possède une version négative, `NOT LIKE`. Son format est identique à celui de `LIKE`. `NOT LIKE` renvoie un un si la chaîne de `data1` ne correspond pas au modèle défini par `data2`.

## Exemples

```
TEMP = "gold" LIKE "gold"
```

Crée une colonne nommée `TEMP` contenant la valeur un (car les deux chaînes correspondent).

```
TEMP = "No" LIKE "NO"
```

Crée une colonne nommée `TEMP` contenant la valeur un (la casse n'est pas importante dans les comparaisons de chaînes).

```
TEMP = V1 LIKE "gold%"
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est égale à un si la ligne correspondante de la colonne `V1` est égale à la chaîne "gold" suivie d'un ou de plusieurs caractères. Sinon, chaque valeur est égale à zéro.

```
TEMP = V1 LIKE "g_ld"
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est égale à un si la ligne correspondante de la colonne `V1` est égale à la chaîne "g" suivie de tout caractère, lui-même suivi de "ld". Sinon, chaque valeur est égale à zéro.

```
TEMP = V1 LIKE V1
```

Crée une colonne nommée `TEMP` ne contenant que des uns (car tout nombre est égal à lui-même).

```
TEMP = V1 LIKE V2
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est la valeur de ligne de la colonne `V1` comparée à la valeur de ligne correspondante de la colonne `V2`.

```
TEMP = V1:V3 LIKE V4:V6
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. La colonne `TEMP` contient les chaînes de `V1` comparées aux chaînes de ligne correspondantes de la colonne `V4`. La colonne `VX` compare les colonnes `V2` et `V5`. La colonne `VY` compare les colonnes `V3` et `V6`.

```
TEMP = V1[10:20] LIKE V2 OU TEMP = V1[10:20] LIKE V2[1:11]
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les résultats de la comparaison des chaînes des lignes 10 à 20 de la colonne `V1` aux lignes 1 à 11 de la colonne `V2`. Les autres cibles de `TEMP` sont vides.

## Fonctions associées

Fonction	Description
<code>EQ</code>	Renvoie TRUE si une plage de données est égale à une autre
<code>GE</code>	Renvoie TRUE si une plage de données est supérieure ou égale à une autre
<code>GT</code>	Renvoie TRUE si une plage de données est supérieure à une autre
<code>LE</code>	Renvoie TRUE si une plage de données est inférieure ou égale à une autre
<code>LT</code>	Renvoie TRUE si une plage de données est inférieure à une autre
<code>NE</code>	Renvoie TRUE si une plage de données n'est pas égale à une autre

## Macro LN ou LOG

`LN` ou `LOG` est disponible uniquement dans Unica Campaign.

## Syntaxe

`LN(data)` OU `LOG(data)`


## Paramètres

`data`

Valeurs numériques dont le logarithme naturel doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

`LN` OU `LOG` calcule le logarithme naturel de chaque valeur de la plage de données spécifiée. Pour chaque colonne d'entrée, `LN` renvoie une nouvelle colonne qui contient le logarithme naturel des nombres de la colonne d'entrée correspondante. Les logarithmes naturels sont basés sur la constante  $e = 2.7182818$ . `LN` est la fonction macro inverse de `EXP`.

 **Remarque** : Toutes les valeurs de la plage de données spécifiée doivent être supérieures à zéro. Dans le cas contraire, une cible vide est renvoyée pour chaque entrée non valide.

## Exemples

`TEMP = LN(3)` OU `TEMP = LOG(3)`

Crée une nouvelle colonne nommée `TEMP` contenant la valeur 1.099.

`TEMP = LN(V1)`

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur correspond au logarithme naturel du contenu de la colonne `V1`.

`TEMP = LN(V1:V3)`

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux logarithmes naturels du contenu de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux naturels du contenu de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux logarithmes naturels du contenu de la colonne `V3`.

`TEMP = LN(V1[10:20])`

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les logarithmes naturels des valeurs des lignes 10 à 20 de la colonne `V1`. Les autres cibles de `TEMP` sont vides.

```
TEMP = LN(V1[1:5]:V2)
```

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` correspondent aux logarithmes naturels des lignes correspondantes de la colonne `V1` et les valeurs de la colonne `VX` correspondent aux logarithmes naturels des lignes correspondantes de la colonne `V2`.

## Fonctions associées

Fonction	Description
<code>EXP</code>	Calcule le nombre naturel (e) élevé à la puissance du contenu de chaque cible de la plage de données spécifiée
<code>LOG2</code>	Calcule le logarithme base 2 du contenu de la plage de données spécifiée
<code>LOG10</code>	Calcule le logarithme base 10 du contenu de la plage de données spécifiée
<code>POW</code>	Calcule une valeur de base élevée aux puissances exponentielles spécifiées

## Macro LOG2

Le `LOG2` est disponible uniquement dans Unica Campaign.

### Syntaxe

```
LOG2(data)
```


### Paramètres

`data`

Valeurs numériques dont le logarithme base 2 doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

`LOG2` calcule le logarithme base 2 des valeurs de la plage de données spécifiée. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant le logarithme base 2 des nombres de la colonne d'entrée correspondante.

 **Remarque** : Toutes les valeurs de la plage de données spécifiée doivent être supérieures à zéro. Dans le cas contraire, une cible vide est renvoyée pour chaque entrée non valide.

## Exemples

```
TEMP = LOG2(8)
```

Crée une nouvelle colonne nommée `TEMP` contenant la valeur trois.

```
TEMP = LOG2(V1)
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur correspond au logarithme base 2 du contenu de la colonne `V1`.

```
TEMP = LOG2(V1:V3)
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux logarithmes base 2 du contenu de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux logarithmes base 2 du contenu de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux logarithmes base 2 du contenu de la colonne `V3`.

```
TEMP = LOG2(V1[10:20])
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les logarithmes base 2 des valeurs des lignes 10 à 20 de la colonne `V1`. Les autres cibles de `TEMP` sont vides.

```
TEMP = LOG2(V1[1:5]:V2)
```

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` correspondent aux logarithmes base 2 des lignes correspondantes de la colonne `V1` et les valeurs de la colonne `VX` correspondent aux logarithmes base 2 des lignes correspondantes de la colonne `V2`.



## Fonctions associées

Fonction	Description
LN ou LOG	Calcule le logarithme naturel du contenu de la plage de données spécifiée
LOG10	Calcule le logarithme base 10 du contenu de la plage de données spécifiée
POW	Puissance exponentielle

## Macro LOG10

Le LOG10 est disponible uniquement dans Unica Campaign.

### Syntaxe

LOG10(data)


### Paramètres

data

Valeurs numériques dont le logarithme base 10 doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

### Description

LOG10 calcule le logarithme base 10 des valeurs de la plage de données spécifiée. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant le logarithme base 10 des nombres de la colonne d'entrée correspondante.

 **Remarque** : Toutes les valeurs de la plage de données spécifiée doivent être supérieures à zéro. Dans le cas contraire, une cible vide est renvoyée pour chaque entrée non valide.

### Exemples

TEMP = LOG10(100)

Crée une nouvelle colonne nommée TEMP contenant la valeur deux.

```
TEMP = LOG10(V1)
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur correspond au logarithme base 10 du contenu de la colonne `V1`.

```
TEMP = LOG10(V1:V3)
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux logarithmes base 10 du contenu de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux logarithmes base 10 du contenu de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux logarithmes base 10 du contenu de la colonne `V3`.

```
TEMP = LOG10(V1[10:20])
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les logarithmes base 10 des valeurs des lignes 10 à 20 de la colonne `V1`. Les autres cibles de `TEMP` sont vides.

```
TEMP = LOG10(V1[1:5]:V2)
```

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` correspondent aux logarithmes base 10 des lignes correspondantes de la colonne `V1` et les valeurs de la colonne `VX` correspondent aux logarithmes base 10 des lignes correspondantes de la colonne `V2`.

## Fonctions associées

Fonction	Description
<code>LN</code> ou <code>LOG</code>	Calcule le logarithme naturel du contenu de la plage de données spécifiée
<code>LOG2</code>	Calcule le logarithme base 2 du contenu de la plage de données spécifiée
<code>POW</code>	Puissance exponentielle

## Macro LOWER

Le `LOWER` est disponible dans Unica Campaign et Unica Interact.

### Syntaxe

```
LOWER(data)
```

## Paramètres

data

Valeur de chaîne à convertir en minuscules.

## Description

LOWER convertit chaque valeur de chaîne de la plage de données spécifiée en minuscules. Il renvoie une nouvelle colonne dont chaque cible contient la chaîne en minuscules de la cible source correspondante.

## Exemples

```
Temp = LOWER "GOLD"
```

Crée une colonne nommée Temp contenant "gold".

```
TEMP = LOWER( "JAN 15, 1997")
```

Crée une colonne nommée TEMP, qui contient la chaîne de texte ASCII " jan 15, 1997 ".

```
TEMP = LOWER( "Pressure")
```

Crée une colonne nommée TEMP, qui contient la chaîne de texte ASCII "pressure".

```
TEMP = LOWER(V1)
```

Crée une colonne nommée TEMP contenant les caractères en minuscules de chaque chaîne de la colonne v1.

## Macro LT

Le LT est disponible dans Unica Campaign et Unica Interact.

## Syntaxe

```
data1 LT data2 data1 < data2
```

## Paramètres

data1

Plage de cibles numériques à comparer. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .


`data2`

Nombres auxquels toutes les valeurs de la colonne spécifiée doivent être comparées. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

`LT` compare les deux plages de données spécifiées, puis renvoie un un si les valeurs du premier jeu de données sont inférieures à celles du deuxième jeu de données ou un zéro dans le cas contraire. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de `data1` comparée à la colonne correspondante de `data2` (à savoir, la première colonne de `data1` est comparée à la première colonne de `data`, la deuxième colonne à la deuxième colonne et ainsi de suite).

Si `data2` est une constante, chaque valeur de `data` est comparée à cette valeur. Si `data2` est une colonne, les calculs sont effectués ligne par ligne. Les valeurs de `data1` sont comparées à la valeur de la première ligne de `data2`, la deuxième ligne à la deuxième ligne et ainsi de suite. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

 **Remarque** : L'opérateur `LT` peut être abrégé en signe inférieur (`<`).

## Exemples

`TEMP = 3 LT 4 OU TEMP = 3 < 4`

Crée une colonne nommée `TEMP` contenant la valeur un (car trois est inférieur à quatre).

`TEMP = V1 < 8`

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est égale à un si la ligne correspondante de la colonne `V1` est inférieure à huit. Sinon, la valeur est égale à zéro.

```
TEMP = V1:V3 < 2
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent au contenu de la colonne `V1` comparé à la valeur deux, les valeurs de la colonne `VX` correspondent au contenu de la colonne `V2` comparé à la valeur deux, et les valeurs de la colonne `VY` correspondent au contenu de la colonne `V3` comparé à la valeur deux.

```
TEMP = V1 < V1
```

Crée une colonne nommée `TEMP` ne contenant que des zéros (car aucun nombre n'est inférieur à lui-même).

```
TEMP = V1 < V2
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est la valeur de ligne de la colonne `V1` comparée à la valeur de ligne correspondante de la colonne `V2`.

```
TEMP = V1[10:20] < V2 OU TEMP = V1[10:20] < V2[1:11]
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les résultats de la comparaison des valeurs des lignes 10 à 20 de la colonne `V1` aux lignes 1 à 11 de la colonne `V2`. Les autres cibles de `TEMP` sont vides.

## Fonctions associées

Fonction	Description
EQ	Renvoie TRUE si une plage de données est égale à une autre
GE	Renvoie TRUE si une plage de données est supérieure ou égale à une autre
GT	Renvoie TRUE si une plage de données est supérieure à une autre
LE	Renvoie TRUE si une plage de données est inférieure ou égale à une autre

## Macro LTRIM

Le `LTRIM` est disponible dans Unica Campaign et Unica Interact.

### Syntaxe

```
LTRIM(data)
```

## Paramètres

`data`

Chaîne de laquelle l'espace à gauche sera supprimé.

## Description

`LTRIM` supprime les premiers caractères d'espacement de chaque valeur de chaîne contenue dans la plage de données spécifiée, puis renvoie la chaîne convertie. Il renvoie une nouvelle colonne pour chaque colonne d'entrée.

## Exemples

```
Temp = LTRIM " gold"
```

Crée une chaîne nommée `Temp` qui contient "gold".

# Macro MAX

Le `MAX` est disponible dans Unica Campaign et Unica Interact.

## Syntaxe

```
MAX(data [, keyword])
```

## Paramètres

`data`

Valeurs numériques dont la valeur maximale doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

`keyword`


Ce mot clé facultatif détermine la manière dont le calcul est effectué sur la plage de données d'entrée. Sélectionnez l'une des options suivantes :

`ALL` - Effectue le calcul sur toutes les cibles de `data` (valeur par défaut)

`COL` - Effectue le calcul séparément pour chaque colonne de `data`

`ROW` - Effectue le calcul séparément pour chaque ligne de `data`

Pour plus de détails sur l'utilisation des mots clés dans Unica Campaign, voir [spécifications de format \(à la page 11\)](#).

 **Remarque** : De nombreuses fonctions de macro utilisent les paramètres de mot clé {`ALL` | `COL` | `ROW`}. Ces mots clés ne s'appliquent pas dans Unica Campaign car les données d'entrées correspondent toujours à une colonne ou à une zone. La macro se comporte toujours comme si le mot clé `COL` a été spécifié. Par conséquent, vous n'avez pas besoin de spécifier ces mots clés lorsque vous utilisez Unica Campaign.

## Description

`MAX` calcule la valeur maximale des valeurs de la plage de données spécifiée. Il renvoie une nouvelle colonne unique contenant la valeur maximale.

## Exemples

`TEMP = MAX(3)` OU `TEMP = MAX(3, ALL)`

Crée une nouvelle colonne nommée `TEMP` contenant la valeur trois.

`TEMP = MAX(V1)`

Crée une colonne nommée `TEMP` contenant une valeur unique qui correspond à la valeur maximale du contenu de la colonne `V1`.

`TEMP = MAX(V1:V3)`

Crée une colonne nommée `TEMP` contenant une valeur unique qui correspond à la valeur maximale des colonnes `V1`, `V2` et `V3`.

`TEMP = MAX(V1[10:20])`

Crée une colonne nommée `TEMP` contenant une valeur unique qui correspond à la valeur maximale des cibles des lignes 10 à 20 de la colonne `V1`.

`TEMP = MAX(V1[1:5]:V4)`

Crée une colonne nommée `TEMP` contenant une valeur unique qui correspond à la valeur maximale des cibles des lignes 1 à 5 des colonnes `V1` à `V4`.

`TEMP = MAX(V1:V3, COL)`

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. La valeur unique de la colonne `TEMP` correspond à la valeur maximale du contenu de la colonne `V1`, la valeur unique de la colonne `VX` correspond à la valeur maximale du contenu de la colonne `V2` et la valeur unique de la colonne `VY` correspond à la valeur maximale du contenu de la colonne `V3`.

```
TEMP = MAX(V1[1:5]:V3, COL)
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY` contenant chacune une seule valeur. La valeur de la colonne `TEMP` correspond à la valeur maximale des cibles des lignes 1 à 5 de la colonne `V1`, la valeur de la colonne `VX` correspond à la valeur maximale des cibles des lignes 1 à 5 de la colonne `V2`, et la valeur de la colonne `VY` correspond à la valeur maximale des cibles des lignes 1 à 5 de la colonne `V3`.

```
TEMP = MAX(V1:V3, ROW)
```

Crée une colonne nommée `TEMP` dans laquelle chaque entrée de cible correspond à la valeur maximale de la ligne correspondante dans les colonnes `V1`, `V2` et `V3`.

```
TEMP = MAX(V1[10:20]:V3, ROW)
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent la valeur maximale des valeurs des lignes 10 à 20 dans les colonnes `V1` à `V3`. Les autres cibles de `TEMP` sont vides.

## Fonctions associées

Fonction	Description
<code>MIN</code>	Calcule la valeur minimale d'une plage de cibles

## Macro MEAN

Le `MEAN` est disponible dans Unica Campaign et Unica Interact.

### Syntaxe

```
MEAN(data [, keyword])
```

### Paramètres

`data`



Valeurs numériques dont la moyenne arithmétique doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

`keyword`


Ce mot clé facultatif détermine la manière dont le calcul est effectué sur la plage de données d'entrée. Sélectionnez l'une des options suivantes :

`ALL` - Effectue le calcul sur toutes les cibles de `data` (valeur par défaut)

`COL` - Effectue le calcul séparément pour chaque colonne de `data`

`ROW` - Effectue le calcul séparément pour chaque ligne de `data`

Pour plus de détails sur l'utilisation des mots clé, voir [DATE \(à la page 69\)](#).

 **Remarque** : De nombreuses fonctions de macro utilisent les paramètres de mot clé {`ALL` | `COL` | `ROW`}. Ces mots clés ne s'appliquent pas dans Unica Campaign car les données d'entrées correspondent toujours à une colonne ou à une zone. La macro se comporte toujours comme si le mot clé `COL` a été spécifié. Par conséquent, vous n'avez pas besoin de spécifier ces mots clés lorsque vous utilisez Unica Campaign.

## Description

`MEAN` calcule la moyenne arithmétique des cibles de la plage de données spécifiée. Pour calculer la moyenne arithmétique, le contenu de l'ensemble des cibles est additionné, puis le résultat est divisé par le nombre de cibles. Le nombre de colonnes renvoyé par `MEAN` dépend de `keyword`.

- Si `keyword` a la valeur `ALL`, `MEAN` renvoie une nouvelle colonne, contenant une valeur unique (la moyenne de toutes les cibles de `data`).
- Si `keyword` a la valeur `COL`, `MEAN` renvoie une nouvelle colonne pour chaque colonne d'entrée. Chaque nouvelle colonne contient une valeur (moyenne de toutes les cibles de la colonne d'entrée correspondante).
- Si `keyword` a la valeur `ROW`, `MEAN` renvoie une nouvelle colonne contenant la moyenne de chaque ligne de `data`.

 **Remarque :** Les cibles vides sont exclues de la moyenne.

 **Remarque :** `MEAN` est identique à la fonction macro `AVG`.

## Exemples

```
TEMP = MEAN(V1)
```

Crée une colonne nommée `TEMP` contenant une valeur unique qui correspond à la moyenne arithmétique du contenu de la colonne `V1`.

```
TEMP = MEAN(V1:V3)
```

Crée une colonne nommée `TEMP` contenant une valeur unique qui correspond à la moyenne arithmétique du contenu des colonnes `V1`, `V2` et `V3`.

```
TEMP = MEAN(V1[10:20])
```

Crée une colonne nommée `TEMP` contenant une valeur unique qui correspond à la moyenne arithmétique des cibles des lignes 10 à 20 de la colonne `V1`.

```
TEMP = MEAN(V1[1:5]:V4)
```

Crée une colonne nommée `TEMP` contenant une valeur unique qui correspond à la moyenne arithmétique des cibles des lignes 1 à 5 des colonnes `V1` à `V4`.

```
TEMP = MEAN(V1:V3, COL)
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. La valeur unique de la colonne `TEMP` correspond à la moyenne arithmétique du contenu de la colonne `V1`, la valeur unique de la colonne `VX` correspond à la moyenne arithmétique du contenu de la colonne `V2` et la valeur unique de la colonne `VY` correspond à la moyenne arithmétique du contenu de la colonne `V3`.

```
TEMP = MEAN(V1[10:20]:V3, COL)
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY` contenant chacune une seule valeur. La valeur de la colonne `TEMP` correspond à la moyenne arithmétique des cibles des lignes 10 à 20 de la colonne `V1`, la valeur de la colonne `VX` correspond à la moyenne arithmétique des cibles des lignes 10 à 20 de la colonne `V2`, et la valeur de la colonne `VY` correspond à la moyenne arithmétique des cibles des lignes 10 à 20 de la colonne `V3`.

```
TEMP = MEAN(V1:V3, ROW)
```

Crée une colonne nommée `TEMP` dans laquelle chaque entrée de cible correspond à la moyenne arithmétique de la ligne correspondante dans les colonnes `V1`, `V2` et `V3`.

```
TEMP = MEAN(V1[1:5]:V3,ROW)
```

Crée une colonne nommée `TEMP`, dans laquelle les cibles des lignes 1 à 5 contiennent la moyenne arithmétique de la ligne correspondante dans les colonnes `V1` à `V3`. Les autres cibles de `TEMP` sont vides.

## Fonctions associées

Fonction	Description
<code>SUM</code> ou <code>TOTAL</code>	Calcule la somme d'une plage de cibles

## Macro MIN

La macro `MIN` est disponible dans Unica Campaign Unica Interact et .

### Syntaxe

```
MIN(data [, keyword])
```

### Paramètres

`data`

Valeurs numériques dont la valeur minimale doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

`keyword`


Ce mot clé facultatif détermine la manière dont le calcul est effectué sur la plage de données d'entrée. Sélectionnez l'une des options suivantes :

`ALL` - Effectue le calcul sur toutes les cibles de `data` (valeur par défaut)

`COL` - Effectue le calcul séparément pour chaque colonne de `data`

`ROW` - Effectue le calcul séparément pour chaque ligne de `data`

Pour plus de détails sur l'utilisation des mots clé, voir [DATE \(à la page 69\)](#).

 **Remarque** : De nombreuses fonctions de macro utilisent les paramètres de mot clé {ALL | COL | ROW}. Ces mots clés ne s'appliquent pas dans **Unica Campaign** car les données d'entrées correspondent toujours à une colonne ou à une zone. La macro se comporte toujours comme si le mot clé COL a été spécifié. Par conséquent, vous n'avez pas besoin de spécifier ces mots clés lorsque vous utilisez **Unica Campaign**.

## Description

MIN calcule la valeur minimale de toutes les cibles de la plage de données spécifiée. Il renvoie une colonne unique contenant la valeur minimale.

## Exemples

```
TEMP = MIN(V1)
```

Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la valeur minimale de la colonne V1.

```
TEMP = MIN(V1:V3)
```

Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la valeur minimale des colonnes V1, V2 et V3.

```
TEMP = MIN(V1[10:20])
```

Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la valeur minimale des cibles des lignes 10 à 20 de la colonne V1.

```
TEMP = MIN(V1[1:5]:V4)
```

Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la valeur minimale des cibles des lignes 1 à 5 des colonnes V1 à V4.

```
TEMP = MIN(V1:V3, COL)
```

Crée trois colonnes nommées TEMP, VX et VY. La valeur unique de la colonne TEMP correspond à la valeur minimale de la colonne V1 la valeur unique de la colonne VX correspond à la valeur minimale de la colonne V2 et la valeur unique de la colonne VY correspond à la valeur minimale de la colonne V3.

```
TEMP = MIN(V1[1:5]:V3, COL)
```

Crée trois colonnes nommées TEMP, VX et VY contenant chacune une seule valeur. La valeur de la colonne TEMP correspond à la valeur minimale des cibles des lignes 1 à 5 de la colonne

V1, la valeur de la colonne VX correspond à la valeur minimale des cibles des lignes 1 à 5 de la colonne V2, et la valeur de la colonne VY correspond à la valeur minimale des cibles des lignes 1 à 5 de la colonne V3.

```
TEMP = MIN(V1:V3, ROW)
```

Crée une colonne nommée TEMP dans laquelle chaque entrée de cible correspond à la valeur minimale de la ligne correspondante dans les colonnes V1, V2 et V3.

```
TEMP = MIN(V1[10:20]:V3, ROW)
```

Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent la valeur maximale des valeurs des lignes 1 à 5 dans les colonnes V1 à V3. Les autres cibles de TEMP sont vides.

## Fonctions associées

Fonction	Description
MAX	Calcule la valeur maximale d'une plage de cibles
MAX_TO_INDEX	Renvoie l'index de colonne de la valeur maximale pour chaque ligne de la colonne spécifiée

## Macro MINUS

Le MINUS est disponible dans Unica Campaign et Unica Interact.

### Syntaxe

```
data MINUS subtrahend data - subtrahend
```

### Paramètres

data

Plage de cibles contenant les nombres desquels une valeur doit être soustraite. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

subtrahend


Nombres à soustraire de toutes les valeurs de la colonne spécifiée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans `subtrahend` doit correspondre au nombre de colonnes dans `data`, à moins que `subtrahend` ne soit une constante. Pour la définition du format de `subtrahend` (identique à `data`), voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

`MINUS` ôte `subtrahend` à soustraire de la plage de données spécifiée `data`. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de `data` moins la colonne correspondante de `subtrahend` (à savoir, la première colonne de `data` est soustraite de la première colonne de `subtrahend`, la deuxième colonne à la deuxième colonne et ainsi de suite).

Si `subtrahend` est une constante, cette valeur est soustraite de chaque valeur de `data`.

Si `subtrahend` contient une ou plusieurs colonnes, les calculs sont effectués ligne par ligne, entre une colonne de `data` et une colonne de `subtrahend`. La première ligne de `data` soustrait la valeur de la première ligne de `subtrahend`, la seconde ligne soustrait la valeur de la deuxième ligne, etc. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

 **Remarque** : L'opérateur `MINUS` peut être abrégé en signe moins ou tiret (-).

## Exemples

`TEMP = 7 MINUS 4` OU `TEMP = 7 - 4`

Crée une nouvelle colonne nommée `TEMP` contenant la valeur trois.

`TEMP = V1 - 8`

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur correspond au contenu de la colonne `V1` moins huit.

`TEMP = V1:V3 - 2`

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent au contenu de la colonne `V1` moins deux, les valeurs de la colonne `VX` correspondent au contenu de la colonne `V2` moins deux et les valeurs de la colonne `VY` correspondent au contenu de la colonne `V3` moins deux.

```
TEMP = V1 - V1
```

Crée une colonne nommée `TEMP` ne contenant que des zéros (car tout nombre moins lui-même est égal à zéro).

```
TEMP = V1 - V2
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est la valeur de ligne de la colonne `V1` moins la valeur de ligne correspondante de la colonne `V2`.

```
TEMP = V1:V3 -V4:V6
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. La colonne `TEMP` contient les valeurs de `V1` moins les valeurs de ligne correspondantes de la colonne `V4`. La colonne `VX` soustrait la colonne `V5` de `V2`. La colonne `VY` soustrait la colonne `V6` de `V3`.

```
TEMP = V1[10:20] - V2 OU TEMP = V1[10:20] - V2[1:11]
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les valeurs des lignes 10 à 20 de la colonne `V1` moins les valeurs des lignes 1 à 11 de la colonne `V2`. Les autres cibles de `TEMP` sont vides.

## Fonctions associées

Fonction	Description
<code>PLUS</code>	Ajoute le contenu de deux plages de données
<code>SUM</code> ou <code>TOTAL</code>	Calcule la somme d'une plage de cibles

## Macro MOD

Le `MOD` est disponible dans Unica Campaign et Unica Interact.

### Syntaxe

```
data MOD divisor data % divisor
```

## Paramètres

`data`

Valeurs entières dont le modulo doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .


`divisor`


Entier de base non nulle par rapport auquel le modulo doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans `divisor` doit correspondre au nombre de colonnes dans `data`, à moins que `divisor` ne soit une constante. Pour la définition du format de `divisor` (identique à `data`), voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

`MOD` calcule le reste de la division de la plage de données spécifiée par une valeur spécifiée. Ce reste est calculé en divisant par `divisor` chaque valeur et en renvoyant le reste. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant les nombres de `data` modulo `divisor`. Le reste possède le même signe (positif ou négatif) que `data`.

Si `divisor` est une constante, chaque valeur de la colonne spécifiée est calculée modulo cette valeur. Si `divisor` est une colonne, les calculs sont effectués ligne par ligne. Les valeurs de `data` sont calculées modulo la valeur de la première ligne de `divisor` et le même calcul est effectué entre les deuxièmes lignes, et ainsi de suite. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

 **Remarque** : Si `divisor` est égal à zéro, une erreur de division par zéro est renvoyée.

 **Remarque** : L'opérateur `MOD` peut être abrégé en signe pourcentage (%). Par exemple, `TEMP = 5 % 3` est identique à `TEMP = 5 MOD 3`.

## Exemples

`TEMP = 10 MOD 8` OU `TEMP = 10 % 8`



Crée une nouvelle colonne nommée `TEMP` contenant la valeur 2.

```
TEMP = -10 % 8
```

Crée une nouvelle colonne nommée `TEMP` contenant la valeur -2.

```
TEMP = V1 % 8
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur correspond au contenu de la colonne `V1` modulo huit.

```
TEMP = V1:V3 % 2
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux valeurs modulo deux du contenu de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux valeurs modulo deux du contenu de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux valeurs modulo deux du contenu de la colonne `V3`.

```
TEMP = V1 % V1
```

Crée une colonne nommée `TEMP` contenant un zéro pour chaque entrée de la colonne `V1`. En effet, tout nombre modulo lui-même renvoie la valeur zéro.

```
TEMP = V1 % V2
```

Crée une colonne nommée `TEMP` dans laquelle chaque valeur est la valeur de ligne de la colonne `V1` modulo la valeur de ligne correspondante de la colonne `V2`. Notez que si `V2=V1`, des zéros sont renvoyés, comme dans l'exemple précédent.

```
TEMP = V1:V3 % V4:V6
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. La colonne `TEMP` contient les valeurs de `V1` modulo les valeurs de ligne correspondantes de la colonne `V4`. La colonne `VX` contient les résultats de la colonne `V2` modulo `V5`. La colonne `VY` contient les résultats de la colonne `V3` modulo `V6`.

```
TEMP = V1[10:20] % V2 OU TEMP = V1[10:20] % V2[1:11]
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles correspondent aux valeurs des lignes 10 à 20 de la colonne `V1` modulo les valeurs des lignes 1 à 11 de la colonne `V2`. Les autres cibles de `TEMP` sont vides.

## Fonctions associées

Fonction	Description
DIV	Divise une plage de données spécifiée par une autre
MOD	Calcule le modulo du contenu de la plage de données spécifiée

## Macro MONTHOF

Le `MONTHOF` est disponible uniquement dans Unica Campaign.

### Syntaxe

```
MONTHOF(date_string [, input_format])
```

### Paramètres

`date_string`

Texte représentant une date valide.

`input_format`

Un des mots clés du tableau ci-après spécifiant le format de date de `date_string`.


### Description

`MONTHOF` renvoie le mois sous la forme d'un nombre pour la date indiquée par `date_string`.

Si `input_format` n'est pas fourni, le mot clé par défaut `DELIM_M_D_Y` est utilisé.

### Exemples

`MONTHOF("012171", MMDDYY)` renvoie le nombre 1.

 **Remarque** : Pour des informations supplémentaires sur les formats de date valides, voir [DATE \(à la page 69\)](#).

## Fonctions associées

Fonction	Description
DAYOF	Renvoie le jour de la semaine sous la forme d'un nombre.

Fonction	Description
WEEKDAYOF	Renvoie le jour de la semaine de la semaine sous forme de nombre.
YEAROF	Renvoie l'année sous la forme d'un nombre.

## Macro MULT

Le `MULT` est disponible dans Unica Campaign et Unica Interact.

### Syntaxe

```
data MULT multiplicier data * multiplicier
```

### Paramètres

`data`

Valeurs numériques à multiplier. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .


`multiplicier`


Nombre par lequel toutes les valeurs de la colonne spécifiée doivent être multipliées. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans `multiplicier` doit correspondre au nombre de colonnes dans `data`, à moins que `multiplicier` ne soit une constante. Pour la définition du format de `multiplicier` (identique à `data`), voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

### Description

`MULT` multiplie les valeurs des deux plages de données spécifiées. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant les nombres de `data` multipliés par `multiplicier`. Si `multiplicier` est une constante, chaque valeur de `data` est multipliée par cette valeur. Si `multiplicier` est une colonne, les calculs sont effectués ligne par ligne.

Les valeurs de `data` sont multipliées par la valeur de la première ligne de `multiplier`, la seconde ligne, par la seconde ligne, etc. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

 **Remarque** : L'utilisation d'une colonne contenant le même nombre `x` dans chaque ligne que `multiplier` revient à utiliser la constante `x` pour `multiplier`.

 **Remarque** : L'opérateur `MULT` peut être abrégé en astérisque (`*`).

## Exemples

```
TEMP = 8 MULT 4 OU TEMP = 8 * 4
```

Crée une nouvelle colonne nommée `TEMP` contenant la valeur 32.

```
TEMP = V1 * 8
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur correspond au contenu de la colonne `V1` multiplié par huit.

```
TEMP = V1:V3 * 2
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent à deux fois le contenu de la colonne `V1`, les valeurs de la colonne `VX` correspondent à deux fois le contenu de la colonne `V2` et les valeurs de la colonne `VY` correspondent à deux fois le contenu de la colonne `V3`.

```
TEMP = V1 * V1
```

Crée une colonne nommée `TEMP` contenant le carré de chaque valeur de la colonne `V1`.

```
TEMP = V1 * V2
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est la valeur de ligne de la colonne `V1` multipliée par la valeur de ligne correspondante de la colonne `V2`.

```
TEMP = V1:V3 * V4:V6
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. La colonne `TEMP` contient les valeurs de `V1` multipliées par les valeurs de ligne correspondantes de la colonne `V4`. La colonne `VX` multiplie la colonne `V2` par `V5`. La colonne `VY` multiplie la colonne `V3` par `V6`.

```
TEMP = V1[10:20] * V2 OU TEMP = V1[10:20] * V2[1:11]
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les valeurs des lignes 10 à 20 de la colonne `V1` fois les valeurs des lignes 1 à 11 de la colonne `V2`. Les autres cibles de `TEMP` sont vides.

## Fonctions associées

Fonction	Description
<code>DIV</code>	Divise une plage de données spécifiée par une autre
<code>EXP</code>	Calcule le nombre naturel (e) élevé à la puissance du contenu de chaque cible de la plage de données spécifiée
<code>POW</code>	Calcule une valeur de base élevée aux puissances exponentielles spécifiées

## Macro NE

Le `NE` est disponible dans Unica Campaign et Unica Interact.

### Syntaxe

```
data1 NE data2 data1 != data2 data1 <> data2
```

### Paramètres

`data1`

Plage de cibles à comparer. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .


`data2`


Nombres auxquels toutes les valeurs de la colonne spécifiée doivent être comparées. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans `data2` doit correspondre au nombre de colonnes dans `data1`, à moins que `data 2` ne soit une constante. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

`NE` compare les deux plages de données spécifiées, puis renvoie un un si les valeurs ne sont pas égales ou un zéro si elles le sont. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de `data1` comparée à la colonne correspondante de `data2` (à savoir, la première colonne de `data1` est comparée à la première colonne de `data`, la deuxième colonne à la deuxième colonne et ainsi de suite).

Si `data2` est une constante, chaque valeur de `data1` est comparée à cette valeur. Si `data2` est une colonne, les calculs sont effectués ligne par ligne. Les valeurs de la première ligne de `data1` sont comparées à la valeur de la première ligne de `data2`, la deuxième ligne à la deuxième ligne et ainsi de suite. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

 **Remarque** : L'utilisation d'une colonne contenant le même nombre `x` dans chaque ligne que `data2` revient à utiliser la constante `x` pour `data2`.

 **Remarque** : L'opérateur `NE` peut être abrégé en point d'exclamation suivi d'un signe égal (`!=`) ou en signe inférieur suivi d'un signe supérieur (`<>`).

## Exemples

```
TEMP = 3 NE 4 OU TEMP = 3 != 4 TEMP = 3 <> 4
```

Crée une colonne nommée `TEMP` contenant la valeur un (car trois n'est pas égal à quatre).

```
TEMP = V1 != 8
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est égale à un si la ligne correspondante de la colonne `V1` n'est pas égale à huit. Sinon, la valeur est égale à zéro.

```
TEMP = V1:V3 != 2
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent au contenu de la colonne `V1` comparé à la valeur deux, les valeurs de la colonne `VX` correspondent au contenu de la colonne `V2` comparé à la valeur deux, et les valeurs de la colonne `VY` correspondent au contenu de la colonne `V3` comparé à la valeur deux.

```
TEMP = V1 != V1
```

Crée une colonne nommée `TEMP` ne contenant que des zéros (car tout nombre est égal à lui-même).

```
TEMP = V1 != V2
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est la valeur de ligne de la colonne `V1` comparée à la valeur de ligne correspondante de la colonne `V2`.

```
TEMP = V1:V3 != V4:V6
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. La colonne `TEMP` contient les valeurs de `V1` comparées aux valeurs de ligne correspondantes de la colonne `V4`. La colonne `VX` compare les colonnes `V2` et `V5`. La colonne `VY` compare les colonnes `V3` et `V6`.

```
TEMP = V1[10:20] != V2 OU TEMP = V1[10:20] != V2[1:11]
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les résultats de la comparaison des valeurs des lignes 10 à 20 de la colonne `V1` et des lignes 1 à 11 de la colonne `V2`. Les autres cibles de `TEMP` sont vides.

## Fonctions associées

Fonction	Description
<code>EQ</code>	Renvoie TRUE si une plage de données est égale à une autre
<code>GE</code>	Renvoie TRUE si une plage de données est supérieure ou égale à une autre
<code>GT</code>	Renvoie TRUE si une plage de données est supérieure à une autre
<code>LE</code>	Renvoie TRUE si une plage de données est inférieure ou égale à une autre
<code>LT</code>	Renvoie TRUE si une plage de données est inférieure à une autre

## Macro NOT

Le `NOT` est disponible dans Unica Campaign et Unica Interact.

### Syntaxe

```
NOT(data) ! data
```


### Paramètres

`data`

Valeurs numériques dont le NOT logique doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

`NOT` renvoie le NOT logique des valeurs de la plage de données spécifiée. Pour chaque colonne d'entrée, `NOT` renvoie une nouvelle colonne contenant le NOT logique des valeurs de la colonne d'entrée correspondante. Cette fonction renvoie la valeur zéro pour les valeurs non nulles et la valeur un pour les valeurs nulles.

 **Remarque :** L'opérateur `NOT` peut être abrégé en point d'exclamation (!) Utilisez le point d'exclamation avant la valeur (par exemple, pour spécifier `NOT(V1)`, vous pouvez simplement saisir `!V1`).

## Exemples

```
TEMP = NOT(3.2) OU TEMP = !1
```

Crée une nouvelle colonne nommée `TEMP` contenant la valeur zéro.

```
TEMP = !0 OU TEMP = !(2+2=3)
```

Crée une nouvelle colonne nommée `TEMP` contenant la valeur un.

```
TEMP = !V1
```

Crée une colonne nommée `TEMP` dans laquelle chaque valeur correspond au NOT logique des valeurs de la colonne `V1`.

```
TEMP = !V1:V3
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux valeurs NOT logiques des valeurs de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux valeurs NOT logiques des valeurs de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux valeurs NOT logiques des valeurs de la colonne `V3`.

```
TEMP = !V1[10:20]
```



Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les valeurs NOT logiques des valeurs des lignes 10 à 20 de la colonne `V1`. Les autres cibles de `TEMP` sont vides.

```
TEMP = !V1[1:5]:V2
```

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` correspondent aux valeurs NOT logiques des valeurs des lignes correspondantes de la colonne `V1` et les valeurs de la colonne `VX` correspondent aux valeurs NOT logiques des valeurs des lignes correspondantes de la colonne `V2`.

## Fonctions associées

Fonction	Description
<code>AND</code>	Calcule le et logique (AND) entre deux plages de données spécifiées
<code>INVERSE</code>	Calcule la valeur négative du contenu de la plage de données spécifiée
<code>OR</code>	Calcule le OR logique entre deux plages de données spécifiées
<code>SIGN</code>	Détermine le signe (positif ou négatif) des valeurs dans la plage de données spécifiée

## Macro NUMBER

La macro `NUMBER` est disponible dans Unica Campaign Unica Interact et .

### Syntaxe

```
NUMBER(data [, conversion_keyword])
```


### Paramètres

`data`

Données de texte ASCII à convertir en valeurs numériques. Il peut s'agir de texte ASCII entre guillemets, d'une colonne de texte, d'une plage de cibles contenant du texte ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

`conversion_keyword`

Ce mot clé facultatif indique comment interpréter les formats de texte des dates et des heures. Sélectionnez l'un des mots clés du tableau ci-après.

 **Remarque** : Si ce paramètre n'est pas spécifié, la valeur par défaut est 1.

Mot clé de conversion	Format	Description
0	#####	Convertit les 5 premiers caractères de chaque chaîne de texte en un numéro unique
1	\$ ( default )	Convertit les valeurs en dollars en valeurs numériques (par exemple, "\$123.45" en 123.45)
2	%	Convertit un pourcentage en valeur numérique (par exemple, "50%" en 0.5)
3	mm/dd/yy hh:mm	Convertit une date et une heure en nombre de jours écoulés depuis le 1 janvier 0000 (1900 est automatiquement ajouté à l'année yy)
4	dd-mmm-yy	Convertit une date en nombre de jours écoulés depuis le 1 janvier 0000 (1900 est automatiquement ajouté à l'année yy)
5	mm/dd/yy	Convertit une date en nombre de jours écoulés depuis le 1 janvier 0000 (1900 est automatiquement ajouté à l'année yy)
6	mmm-yy	Convertit une date en nombre de jours écoulés entre le premier jour du mois spécifié et le 1er janvier 0000 (1900 est automatiquement ajouté à l'année yy)
7	dd-mmm	Convertit une date en nombre de jours écoulés depuis le début de l'année (par exemple, "01-FEB" en 32)

Mot clé de conversion	Format	Description
8	mmm	Convertit une abréviation de mois en trois lettres en une valeur comprise entre 1 et 12 (par exemple, "DEC" en 12)
9	{January   February   March ... }	Convertit un nom de mois en toutes lettres en valeur comprise entre 1 et 12 (par exemple, "March" en 3)
10	{Sun   Mon   Tue ... }	Convertit une abréviation de jour de la semaine sur trois caractères en valeur comprise entre 0 et 6, dimanche correspondant au début de la semaine (par exemple, "Sun" en 0)
11	{Sunday   Monday   Tuesday ... }	Convertit un nom de jour de la semaine en toutes lettres en valeur comprise entre 0 et 6, dimanche correspondant au début de la semaine (par exemple, "Monday" en 1)
12	hh:mm:ss {AM   PM}	Convertit l'heure en nombre de secondes écoulées depuis 00:00:00 AM (minuit) (par exemple, "01:00:00 AM" en 3600)
13	hh:mm:ss	Convertit l'heure en nombre de secondes écoulées depuis 00:00:00 AM (minuit) (par exemple, "01:00:00" en 3600)
14	hh:mm {AM   PM}	Convertit l'heure en nombre de minutes écoulées depuis 00:00:00 AM (minuit) (par exemple, "01:00 AM" en 60)
15	hh:mm	Convertit l'heure en nombre de minutes écoulées depuis 00:00:00 AM (minuit) (par exemple, "01:00" en 60)
16	mm:ss	Convertit l'heure en nombre de secondes écoulées depuis 00:00:00 AM (minuit) (par exemple, "30:00" en 1800)

Mot clé de conversion	Format	Description
17	ddmm	Convertit une date en nombre de jours écoulés depuis le début de l'année (par exemple, "3101" en 31)
18	ddmmm	Convertit une date en nombre de jours écoulés depuis le début de l'année (par exemple, "31JAN" en 31)
19	ddmmyy	Convertit une date en nombre de jours écoulés depuis le 1 janvier 0000 (1900 est automatiquement ajouté à l'année si yy est inférieur ou égal à 20. Sinon, 2000 est ajouté)
20	ddmmyyyy	Convertit une date en nombre de jours écoulés depuis le 1er janvier 0000 (par exemple, "31JAN0000" en 31)
21	ddmmyy	Convertit une date en nombre de jours écoulés depuis le 1 janvier 0000 (1900 est automatiquement ajouté à l'année si yy est inférieur ou égal à 20. Sinon, 2000 est ajouté)
22	ddmmyyyy	Convertit une date en nombre de jours écoulés depuis le 1er janvier 0000 (par exemple, "31010000" en 31)
23	mmdd	Convertit une date en nombre de jours écoulés depuis le début de l'année (par exemple, "0131" en 31)
24	mddy	Convertit une date en nombre de jours écoulés depuis le 1 janvier 0000 (1900 est automatiquement ajouté à l'année si yy est inférieur ou égal à 20. Sinon, 2000 est ajouté)
25	mddy	Convertit une date en nombre de jours écoulés depuis le 1er janvier 0000 (par exemple, "01010001" en 366)

Mot clé de conversion	Format	Description
26	mmm	Convertit une abréviation de mois en trois lettres en une valeur comprise entre 1 et 12 (par exemple, "MAR" en 3) [Remarque : identique au mot clé de conversion 8]
27	mmdd	Convertit une date en nombre de jours écoulés depuis le début de l'année (par exemple, "JAN31" en 31)
28	mmddyy	Convertit une date en nombre de jours écoulés depuis le 1 janvier 0000 (1900 est automatiquement ajouté à l'année si yy est inférieur ou égal à 20. Sinon, 2000 est ajouté)
29	mmddyyyy	Convertit une date en nombre de jours écoulés depuis le 1er janvier 0000 (par exemple, "FEB010001" en 32)
30	mmyy	Convertit une date en nombre de jours écoulés entre le 1er du mois spécifié et le 1er janvier 0000 (1900 est automatiquement ajouté à l'année si yy est inférieur ou égal à 20 ; sinon, 2000 est ajouté)
31	mmyyyy	Convertit une date en nombre de jours écoulés entre le 1er du mois spécifié et le 1er janvier 0000 (par exemple, "FEB0001" en 32)
32	mmyy	Convertit une date en nombre de jours écoulés entre le 1er du mois spécifié et le 1er janvier 0000 (1900 est automatiquement ajouté à l'année si yy est inférieur ou égal à 20 ; sinon, 2000 est ajouté)
33	mmyyyy	Convertit une date en nombre de jours écoulés entre le 1er du mois spécifié et le 1er janvier 0000 (par exemple, "020001" en 32)

Mot clé de conversion	Format	Description
34	yyymm	Convertit une date en nombre de jours écoulés entre le 1er du mois spécifié et le 1er janvier 0000 (1900 est automatiquement ajouté à l'année si yy est inférieur ou égal à 20 ; sinon, 2000 est ajouté)
35	yyymmdd	Convertit une date en nombre de jours écoulés depuis le 1 janvier 0000 (1900 est automatiquement ajouté à l'année si yy est inférieur ou égal à 20. Sinon, 2000 est ajouté)
36	yymmm	Convertit une date en nombre de jours écoulés entre le 1er du mois spécifié et le 1er janvier 0000 (1900 est automatiquement ajouté à l'année si yy est inférieur ou égal à 20 ; sinon, 2000 est ajouté)
37	yymmdd	Convertit une date en nombre de jours écoulés depuis le 1 janvier 0000 (1900 est automatiquement ajouté à l'année si yy est inférieur ou égal à 20. Sinon, 2000 est ajouté)
38	YYYY	Convertit le nombre d'années écoulées depuis l'an 0000 (par exemple, "1998" en 1998)
39	YYYYmm	Convertit une date en nombre de jours écoulés entre le 1er du mois spécifié et le 1er janvier 0000 (par exemple, "000102" en 32)
40	YYYYmdd	Convertit une date en nombre de jours écoulés depuis le 1er janvier 0000 (par exemple, "00010201" en 32)
41	YYYYmmm	Convertit une date en nombre de jours écoulés entre le 1er du mois spécifié et le 1er janvier 0000 (par exemple, "000102" en 32)

<b>Mot clé de conversion</b>	<b>Format</b>	<b>Description</b>
42	yyyymmdd	Convertit une date en nombre de jours écoulés depuis le 1er janvier 0000 (par exemple, "0001FEB01" en 32)
43	<day>* <month>	Convertit toute date délimitée avec le jour suivi du mois en nombre de jours écoulés depuis le début de l'année (par exemple, "15-JAN" en 15)
44	<day>* <month>* <year>	Convertit toute date délimitée avec le jour précédant le mois et suivi de l'année en nombre de jours écoulés depuis le 1er janvier 0000 (par exemple, "1/1/0001" en 366)
45	<month>* <day>	Convertit toute date délimitée avec le mois suivi du jour en nombre de jours écoulés depuis le début de l'année (par exemple, "JAN 31" en 31)
46	<month>* <day>* <year>	Convertit toute date délimitée avec le mois suivi du jour, suivi de l'année en nombre de jours écoulés depuis le 1er janvier 0000 (par exemple, "JAN 1, 0001" en 366)
47	<month>* <year>	Convertit toute date délimitée avec le mois suivi de l'année en nombre de jours écoulés entre le premier jour du mois spécifié et le 1er janvier 0000
48	<year>* <month>	Convertit toute date délimitée avec l'année suivi du mois en nombre de jours écoulés entre le premier jour du mois spécifié et le 1er janvier 0000
49	<year>* <month>* <day>	Convertit toute date délimitée avec le mois suivi du jour, suivi de l'année en nombre de

	<b>Mot clé de conversion</b>	<b>Format</b>	<b>Description</b>
			jours écoulés depuis le 1er janvier 0000 (par exemple, "0001/01/01" en 366)
50	YY		Convertit l'année en nombre d'années écoulées depuis l'an 0000 (par exemple, "97" en 97)
51		mm	Convertit le mois en valeur comprise entre 1 et 12 (par exemple, "SEP" en 9)
52		dd	Convertit le jour en valeur comprise entre 1 et 31 (par exemple, "28" en 28)
53		{January   February   March ... }	Convertit un nom de mois en toutes lettres en valeur comprise entre 1 et 12 (par exemple, "March" en 3) [Remarque : identique au mot clé de conversion 9]
54		{Sunday   Monday   Tuesday ... }	Convertit un nom de jour de la semaine en toutes lettres en valeur comprise entre 1 et 7, dimanche correspondant au début de la semaine (par exemple, "Sunday" en 1)
55		{Sun   Mon   Tue ... }	Convertit une abréviation de jour de la semaine sur trois caractères en valeur comprise entre 1 et 7, dimanche correspondant au début de la semaine (par exemple, "Sun" en 1)

## Description


NUMBER convertit les valeurs de texte dans la plage de données spécifiée en valeurs numériques, à l'aide du format spécifié pour convertir les dates et les heures. Si une chaîne de texte ne peut pas être analysée à l'aide du mot clé `conversion_keyword` spécifié, NUMBER génère une erreur. Le format 0 convertit les cinq premiers caractères de chaque chaîne de texte en un nombre différent pour chaque chaîne de texte unique. C'est un moyen simple de changer une colonne de texte en classes uniques pour les sorties dans un discriminant.





Les formats délimités (mots clés de conversion 43 à 49) acceptent les caractères suivants comme délimiteurs :


- / (barre oblique)
- - (tiret)
- , (virgule)
- " " (espace)
- : (deux-points)

Les mois peuvent être représentés sous la forme `mm` ou `mmm` ; les jours peuvent être représentés sous la forme `d` ou `dd` ; les années peuvent être représentées sous la forme `yy` ou `yyyy`.

 **Remarque** : Dans le cadre de la conformité à l'an 2000, toutes les années des dates peuvent être désignées sous la forme `yyyy` au lieu de `yy`. Pour la compatibilité amont, dans les mots clés de conversion 1 à 16, 1900 est automatiquement ajouté à `yy` (années sur deux chiffres). Pour les mots clés de conversion 17 à 55, 2000 a été ajouté automatiquement au seuil `yy < threshold` ; 1900 automatiquement ajouté au seuil `yy # threshold`.

 **Remarque** : La valeur `threshold` de l'an 2000 est définie dans l'onglet **Nettoyage des données** de la fenêtre **Paramètres avancés** (appelez-la en sélectionnant **Options > Paramètres > Paramètres avancés**).

 **Remarque** : Si vous changez la valeur de seuil an 2000, vous devez mettre à jour toutes les fonctions macro à l'aide de la fonction macro `NUMBER` pour manipuler les valeurs de date avec des années sur 2 chiffres. Pour forcer une mise à jour d'une fonction macro, vous pouvez effectuer tout changement (par exemple, ajouter un espace et le supprimer) et cliquer sur l'icône en forme de coche pour accepter le changement.

 **Remarque** : Lorsque vous utilisez le format 0, seuls les cinq premiers caractères de chaque chaîne de texte sont utilisés pour générer un nombre unique. Toutes les chaînes contenant ces cinq premiers caractères seront converties dans la même valeur numérique. La même chaîne de texte générera la même valeur numérique à chaque fois, même entre des feuilles de calcul différentes. Si nécessaire, utilisez des macros de chaîne pour

manipuler les chaînes de sorte que les cinq premiers caractères définissent une classe de manière unique. Notez que les valeurs numériques résultantes peuvent être très petites. Utilisez la fenêtre **Formats d'affichage** pour augmenter le nombre de décimales à afficher ou changer le format en mode exponentiel (00E+00).

## Exemples

TEMP = NUMBER("\$1.23") OU TEMP = NUMBER("123%", 2)

Crée une colonne nommée TEMP contenant le nombre 1.23.

TEMP = NUMBER(column("Jan", "Mar", "Dec", 8))

Crée une colonne nommée TEMP contenant les nombres 1, 3 et 12.

TEMP = NUMBER("1:52 PM", 14)

Crée une colonne nommée TEMP contenant le nombre 832.

TEMP = NUMBER("1/1/95", 5)

Crée une colonne nommée TEMP contenant le nombre 728660.

TEMP = NUMBER(V1)

Crée une colonne nommée TEMP contenant les valeurs numériques des chaînes de texte de la colonne V1. Les valeurs en dollars sont correctement converties en valeurs numériques. ??? est renvoyé pour les chaînes de texte qui ne peuvent pas être analysées à l'aide du format \$.

TEMP = NUMBER(V1:V3, 4)

Crée trois colonnes nommées TEMP, VX et VY. La colonne TEMP contient les valeurs numériques des chaînes de texte de la colonne V1. La colonne VX contient les valeurs numériques des chaînes de texte de la colonne V2. La colonne VY contient les valeurs numériques des chaînes de texte de la colonne V3. Les dates au format dd-mmm-yy sont converties en nombre de jours de décalage à partir du 1er janvier 0000. ??? sont renvoyées pour les chaînes de texte qui ne peuvent pas être analysées à l'aide du format \$.

TEMP = NUMBER(V1[10:20]:V2, 10)

Crée deux colonnes nommées TEMP et VX. La colonne TEMP contient les valeurs numériques des chaînes de texte des lignes 10 à 20 de la colonne V1. La colonne VX contient les valeurs numériques des chaînes de texte des lignes 10 à 20 de la colonne V2. Toutes

les représentations standard de jour de la semaine à trois caractères sont converties en nombres de 0 à 6 (0 = dimanche, 6 = samedi). En l'absence de correspondance pour un nom de jour de la semaine, ??? est renvoyé.

```
TEMP = NUMBER(V1, 0)
```

Si la colonne `V1` ne contient que des chaînes de texte à cinq chiffres, crée une colonne nommée `TEMP` contenant une valeur numérique différente par chaîne unique.

## Fonctions associées

### Fonction

### Description

`WEEKDAY` Convertit les chaînes de date textuelles ASCII en jour de la semaine

## Macro OR

Le `OR` est disponible dans Unica Campaign et Unica Interact.

### Syntaxe

```
data1 OR data2 data1 || data2
```

### Paramètres

`data1`

Nombres dont le OR logique doit être calculé avec les valeurs de `data2`. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .


`data2`


Nombre(s) dont le OR logique doit être calculé avec les valeurs de `data1`. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans `data2` doit correspondre au nombre de colonnes dans `data1`, à moins que `data2` ne soit une constante. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

OR calcule le OR logique entre les deux plages de données spécifiées. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de `data1` dont le OR logique avec la colonne correspondante de `data2` a été calculé (à savoir, le OR logique de la première colonne de `data1` avec la première colonne de `data`, celui de la deuxième colonne avec la deuxième colonne et ainsi de suite).

Si `data2` est une constante, le OR logique de chaque valeur de `data1` avec cette valeur est calculé. Si `data2` contient une ou plusieurs colonnes, les calculs sont effectués ligne par ligne, entre une colonne de `data1` et une colonne de `data2`. Le OR logique de la première ligne de `data1` avec la première ligne de `data2` est calculé, la seconde ligne avec la seconde ligne, etc. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

 **Remarque** : L'utilisation d'une colonne contenant le même nombre `x` dans chaque ligne que `data2` revient à utiliser la constante `x` pour `data2`.

 **Remarque** : L'opérateur OR peut être abrégé en double barre verticale (`||`). Utilisez la double barre verticale pour séparer les deux arguments (par exemple, pour spécifier `V1 OR 3`, vous pouvez simplement entrer `V1 || 3`).

## Exemples

```
TEMP = 1 OR 8 OU TEMP = 1 || 8
```

Crée une colonne nommée `TEMP` contenant la valeur un (tout nombre non nul prend la valeur un).

```
TEMP = V1 || 1
```

Crée une colonne nommée `TEMP` ne contenant que des uns (un OR entre toute valeur et le nombre un génère un).

```
TEMP = V1 || V2
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est la valeur de ligne de la colonne `V1` dont le OR logique est calculé avec la valeur de ligne correspondante de la colonne `V2`.

```
TEMP = V1:V3 || V4:V6
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. La colonne `TEMP` contient les valeurs de `V1` dont le OR logique avec les valeurs de ligne correspondantes de la colonne `V4` a été calculé. La colonne `VX` contient les valeurs résultant du OR logique des colonnes `V2` et `V5`. La colonne `VY` contient les valeurs résultant du OR logique des colonnes `V3` et `V6`.

```
TEMP = V1[10:20] || V2
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent le résultat du OR logique entre les valeurs des lignes 10 à 20 des colonnes `V1` et `V2`. Les autres cibles de `TEMP` sont vides.

## Fonctions associées

Fonction	Description
<code>AND</code>	Calcule le et logique (AND) entre deux plages de données spécifiées
<code>NOT</code>	Calcule le non logique (NON) du contenu de la plage de données spécifiée

## Macro POSITION

Le `POSITION` est disponible uniquement dans Unica Campaign.

### Syntaxe

```
POSITION(colName, pattern [, start [, occurrence]])
```

### Paramètres

`colName`

Valeur d'une colonne (doit être de type `string`).

`pattern`

Modèle ou chaîne que vous recherchez.

`start`


Octet à partir duquel la recherche commence.

`occurrence`

Indiquez une valeur pour `n`, où vous recherchez la `n` ème occurrence du modèle à renvoyer.

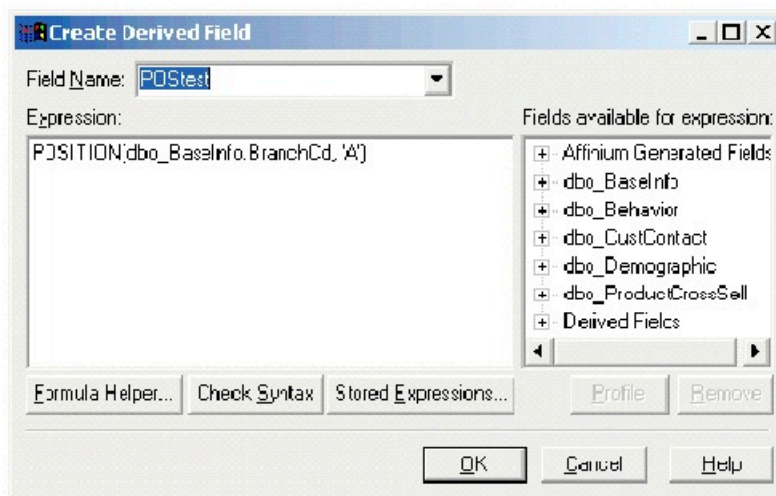
## Description

`POSITION` renvoie la position de l'octet de départ du modèle ou de la chaîne, dans la valeur d'une colonne (`colName`) qui doit être de type chaîne. Si vous donnez une valeur au paramètre début, la recherche commence à ce point. Le paramètre occurrence correspond à la n-ième apparition du modèle à renvoyer.

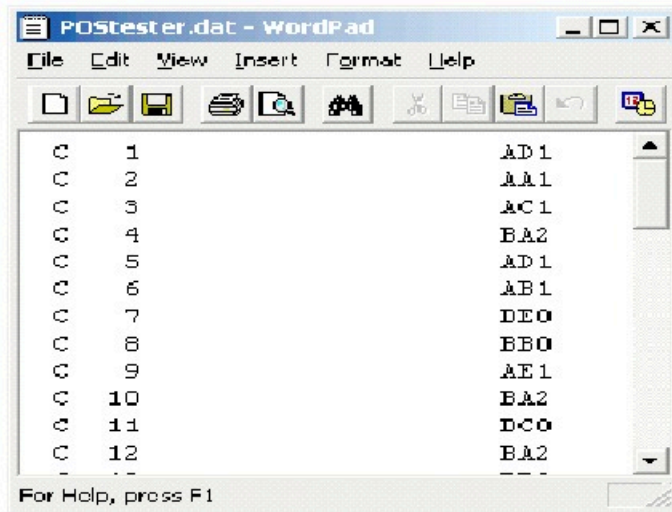
 **Remarque** : La recherche n'est pas sensible à la casse.

## Exemples

Dans l'exemple ci-dessous, nous recherchons le modèle ou la chaîne 'A' dans la valeur de la colonne `dbo_BaseInfo.BranchCd` et affectons la valeur renvoyée au fichier dérivé `POSTest`.



L'exemple suivant illustre quelques lignes de la table contenant les valeurs de `dbo_BaseInfo.BranchCd` et `POSTest`.



Exemple plus complexe :

```
STRING_SEG(POSITION(CellCode, "X", 1, 2)+1,
STRING_LENGTH(CellCode), CellCode) = "AAA"
```

Cette commande renvoie les lignes pour lesquelles les valeurs de `CellCode` se terminent par "AAA" après la deuxième occurrence de "X".

## Macro PLUS

Le `PLUS` est disponible dans Unica Campaign et Unica Interact.

### Syntaxe

```
data PLUS addend data + addend
```

### Paramètres

`data`

Plage de cibles contenant les nombres à ajouter. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs.

Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .


addend

Nombres à ajouter à toutes les valeurs de la colonne spécifiée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans `data2` doit correspondre au nombre de colonnes dans `data1`, à moins que `data2` ne soit une constante. Pour la définition du format de `addend` (identique à `data`), voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

`PLUS` ajoute les valeurs des deux plages de données spécifiées. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de `data1` ajoutée à la colonne correspondante de `data2` (à savoir, la première colonne de `data1` est ajoutée à la première colonne de `data`, la deuxième colonne à la deuxième colonne et ainsi de suite).

Si `data2` est une constante, chaque valeur de `data1` est augmentée de cette valeur. Si `data2` contient une ou plusieurs colonnes, les calculs sont effectués ligne par ligne, entre une colonne de `data1` et une colonne de `data2`. La première ligne de `data1` est ajoutée à la première ligne de `data2`, la seconde ligne, par la seconde ligne, etc. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

 **Remarque** : L'opérateur `PLUS` peut être abrégé en signe plus (+).

## Exemples

`TEMP = 3 PLUS 4` OU `TEMP = 3 + 4`

Crée une nouvelle colonne nommée `TEMP` contenant la valeur sept.

`TEMP = V1 + 8`

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur correspond au contenu de la colonne `V1` plus huit.

`TEMP = V1 + V1`

Crée une colonne nommée `TEMP` contenant deux fois le contenu de la colonne `V1`.

`TEMP = V1 + V2`



Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est la valeur de ligne de la colonne `V1` plus la valeur de ligne correspondante de la colonne `V2`.

`TEMP = V1:V3 + V4:V6`

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. La colonne `TEMP` contient les valeurs de `V1` plus les valeurs de ligne correspondantes de la colonne `V4`. La colonne `VX` ajoute les colonnes `V2` et `V5`. La colonne `VY` ajoute les colonnes `V5` et `V6`.

`TEMP = V1[10:20] + V2` OU `TEMP = V1[10:20] + V2[1:11]`

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les sommes des valeurs des lignes 10 à 20 de la colonne `V1` et les valeurs des lignes 1 à 11 de la colonne `V2`. Les autres cibles de `TEMP` sont vides.

## Fonctions associées

Fonction	Description
<code>MINUS</code>	Soustrait une plage de données spécifiée d'une autre
<code>SUM</code> OU <code>TOTAL</code>	Calcule la somme d'une plage de cibles

## Macro POWER

Le `POWER` est disponible dans Unica Campaign et Unica Interact.

### Syntaxe

`base POWER exponent base ^ exponent`

### Paramètres

`base`

Valeurs numériques à élever à une puissance exponentielle. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `base` (identique à `data`), voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .


`exponent`


Nombres exponentiels desquels les valeurs de `data` doivent être élevées. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans `exponent` doit correspondre au nombre de colonnes dans `base`, à moins que `base` ne soit une constante. Pour la définition du format de `exponent` (identique à `data`), voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

`POWER` élève les valeurs de la première plage de données à la puissance spécifiée dans la deuxième plage de données (il calcule  $base^{exponent}$  ). Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant le résultat de l'élévation de `base` à la puissance `exponent` (à savoir, la première colonne de `data1` est élevée à la première colonne de `data`, la deuxième colonne à la deuxième colonne et ainsi de suite).

Si `exponent` est une constante, chaque valeur de `base` est élevée à cette valeur. Si `exponent` contient une ou plusieurs colonnes, les calculs sont effectués ligne par ligne, entre une colonne de `base` et une colonne de `exponent`. La première ligne de `base` est élevée à la première ligne de `exponent`, la seconde ligne, par la seconde ligne, etc. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

 **Remarque** : L'opérateur `POWER` peut être abrégé en accent circonflexe (^). Par exemple, `TEMP = 2^8` est identique à `TEMP = 2 POWER 8`.

 **Remarque** : Si la valeur `x` est trop faible, une erreur de dépassement est renvoyée. Cela se produit si `base^exponent` dépasse la valeur à virgule flottante sur 32 bits maximale ou minimale.

## Exemples

`TEMP = 2 POWER 3` OU `TEMP = 2^3`

Crée une nouvelle colonne nommée `TEMP` contenant la valeur huit.

`TEMP = V1 ^ 0.5`

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur correspond à la racine carrée du contenu de la colonne `V1` (équivalent à `SQRT(V1)`).

```
TEMP = V1 ^ V3
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est la valeur de ligne de la colonne `V1` élevée à la valeur de ligne correspondante de la colonne `V2`.

```
TEMP = V1:V3 ^ V4:V6
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. La colonne `TEMP` contient les valeurs de `V1` élevées aux valeurs de ligne correspondantes de la colonne `V4`. La colonne `VX` contient le résultat de la colonne `V2` élevée aux valeurs correspondantes dans la colonne `V5`. La colonne `VY` contient le résultat de la colonne `V3` élevée aux valeurs correspondantes dans la colonne `V6`.

```
TEMP = V1[10:20] POWER V2 OU TEMP = V1[10:20] POWER V2[1:11]
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent le résultat de l'élevation des valeurs des lignes 10 à 20 de la colonne `V1` par les valeurs des lignes 1 à 10 de la colonne `V2`. Les autres cibles de `TEMP` sont vides.

## Fonctions associées

Fonction	Description
<code>EXP</code>	Calcule le nombre naturel (e) élevé à la puissance du contenu de chaque cible de la plage de données spécifiée
<code>LN</code> OU <code>LOG</code>	Calcule le logarithme naturel du contenu de la plage de données spécifiée
<code>LN2</code>	Calcule le logarithme base 2 du contenu de la plage de données spécifiée
<code>LN10</code>	Calcule le logarithme base 10 du contenu de la plage de données spécifiée

## Macro RANDOM

Le `RANDOM` est disponible uniquement dans Unica Campaign.

### Syntaxe

```
RANDOM(num [, seed]) RANDOM(num, value1, value2 [, seed])
```

## Paramètres

`num`

Nombre de nombres aléatoires à générer. Il doit s'agir d'un entier positif supérieur à zéro.

`value1`

Limite sur les nombres aléatoires à générer. Il peut s'agir de toute valeur constante ou d'une expression convertie en constante. Si ce paramètre n'est pas fourni, la valeur par défaut est zéro.

`value2`


L'autre limite sur les nombres aléatoires à générer. Il peut s'agir de toute valeur constante ou d'une expression convertie en constante. Si ce paramètre n'est pas fourni, la valeur par défaut est un.

`seed`

Valeur de départ facultative à utiliser pour la génération de nombres aléatoires. Vous devez indiquer un nombre entier.

## Description

`RANDOM` génère une colonne de nombres aléatoires. Il renvoie une nouvelle colonne contenant `num` nombre aléatoires. Si `value1` et `value2` sont spécifiés, les nombres aléatoires sont générés entre ces limites (incluses). Si vous n'en indiquez pas, les valeurs sont générées par défaut entre zéro et un. Si `seed` est fourni, il est utilisé comme valeur de départ pour le générateur de nombres aléatoires.

 **Remarque** : Si `seed` est supérieur ou égal à  $2^{32}$ , la valeur est remplacée par  $2^{32} - 1$ . Les valeurs de `seed` supérieures à  $2^{24}$  sont arrondies (la précision est perdue). Par conséquent, plusieurs valeurs peuvent posséder la même valeur `seed`.

## Exemples

```
TEMP = RANDOM()
```

Crée une colonne nommée `TEMP` contenant des nombres aléatoires de longueur illimitée.

```
TEMP = RANDOM(100)
```

Crée une colonne nommée `TEMP` contenant 100 nombres aléatoires compris entre 0,0 et 1,0.

```
TEMP = RANDOM(100, 5943049)
```

Crée une colonne nommée `TEMP` contenant 100 nombres aléatoires générés à partir du numéro de valeur de départ 5943049.

```
TEMP = RANDOM(100, 0, 100)
```

Crée une colonne nommée `TEMP` contenant 100 nombres aléatoires compris entre 0,0 et 100,0.

```
TEMP = RANDOM(100, 0, 100, 5943049)
```

Crée une colonne nommée `TEMP` contenant 100 nombres aléatoires compris entre 0 et 100 générés à partir du numéro de valeur de départ 5943049.

## Fonctions associées

Fonction	Description
<code>RANDOM_GAUSS</code>	Renvoie le nombre spécifié de valeurs aléatoires à partir d'une distribution de Gauss

## Macro RANDOM\_GAUSS

Le `RANDOM_GAUSS` est disponible uniquement dans Unica Campaign.

### Syntaxe

```
RANDOM_GAUSS(num [, seed]) RANDOM_GAUSS(num, mean, std [, seed])
```

### Paramètres

`num`

Nombre de nombres aléatoires à générer. Il doit s'agir d'un entier positif supérieur à zéro.

`mean`

Moyenne de la valeur de Gauss. Il peut s'agir de toute valeur constante ou d'une expression convertie en constante. Si ce paramètre n'est pas fourni, la valeur par défaut est zéro.

`std`

Ecart type de la valeur de Gauss. Il peut s'agir de toute valeur constante ou d'une expression convertie en constante. Si ce paramètre n'est pas fourni, la valeur par défaut est un.

`seed`

Valeur de départ facultative à utiliser pour la génération de nombres aléatoires. Vous devez indiquer un nombre entier. (Si une valeur non entière est fournie, la partir entière de cette valeur est automatiquement utilisée à la place.)

## Description

`RANDOM_GAUSS` génère une colonne de nombres aléatoires basée sur une distribution de Gauss. Il renvoie une nouvelle colonne contenant `num` nombre aléatoires. Si `mean` et `std` sont spécifiés, les nombres aléatoires sont générés à l'aide d'une distribution de Gauss avec la moyenne et l'écart type spécifiés. Si ces arguments ne sont pas indiqués, par défaut, la distribution gaussienne a une valeur de zéro et son écart type est égal à un. Si `seed` est fourni, il est utilisé comme valeur de départ pour le générateur de nombres aléatoires.

## Exemples

```
TEMP = RANDOM_GAUSS(100)
```

Crée une colonne nommée `TEMP` contenant 100 valeurs échantillonnées de manière aléatoire à partir d'une distribution de Gauss de moyenne nulle et d'écart type égal à un.

```
TEMP = RANDOM_GAUSS(500, 3)
```

Crée une colonne nommée `TEMP` contenant 100 valeurs échantillonnées de manière aléatoire à partir d'une distribution de Gauss de moyenne nulle et d'écart type égal à un. Le nombre 3 est utilisé comme valeur de départ pour le générateur de nombre aléatoire.

```
TEMP = RANDOM_GAUSS(5000, 100, 32)
```

Crée une colonne nommée `TEMP` contenant 5000 valeurs échantillonnées de manière aléatoire à partir d'une distribution de Gauss de moyenne égale à 100 et d'écart type égal à 32.

```
TEMP = RANDOM_GAUSS(500, -1, 2, 3)
```

Crée une colonne nommée `TEMP` contenant 500 valeurs échantillonnées de manière aléatoire à partir d'une distribution de Gauss de moyenne égale à -1 et d'écart type égal à 2. Le nombre 3 est utilisé comme valeur de départ pour le générateur de nombre aléatoire.

## Fonctions associées

Fonction	Description
RANDOM	Renvoie le nombre spécifié de nombres aléatoires

## Macro ROUND

Le `ROUND` est disponible uniquement dans Unica Campaign.

### Syntaxe

`ROUND(data)`

### Paramètres

`data`

Valeurs numériques à arrondir. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

### Description

`ROUND` arrondit les valeurs de la plage de données spécifiée à l'entier le plus proche. Pour chaque colonne d'entrée, `ROUND` renvoie une nouvelle colonne qui contient la valeur arrondie des nombres de la colonne d'entrée correspondante. Les nombres à égale distance de deux entiers sont arrondis vers le haut (par exemple, 2.5 est arrondi à 3.0 et 2.5 est arrondi à -2.0).

### Exemples

```
TEMP = ROUND(3.2)
```

Crée une nouvelle colonne nommée `TEMP` contenant la valeur trois.

```
TEMP = ROUND(V1)
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur correspond à la valeur arrondie du contenu de la colonne `V1`.

```
TEMP = ROUND(V1:V3)
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux valeurs arrondies du contenu de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux valeurs arrondies du contenu de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux valeurs arrondies du contenu de la colonne `V3`.

```
TEMP = ROUND(V1[10:20])
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les valeurs arrondies des lignes 10 à 20 de la colonne `V1`. Les autres cibles de `TEMP` sont vides.

```
TEMP = ROUND(V1[1:5]:V2)
```

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` correspondent aux valeurs arrondies des lignes correspondantes de la colonne `V1` et les valeurs de la colonne `VX` correspondent aux valeurs arrondies des lignes correspondantes de la colonne `V2`.

## Fonctions associées

Fonction	Description
<code>INT</code>	Calcule la valeur entière (arrondie) du contenu de la plage de données spécifiée
<code>MOD</code>	Calcule le modulo du contenu de la plage de données spécifiée
<code>TRUNCATE</code>	Renvoie la partie non décimale de chaque valeur de la plage de données spécifiée

## Macro ROWNUM

Le `ROWNUM` est disponible uniquement dans Unica Campaign.


### Syntaxe

```
ROWNUM( )
```

### Description

`ROWNUM` génère des nombres séquentiels de un à n, où n correspond au nombre d'enregistrements. Le nombre pour le premier enregistrement est 1, celui du deuxième 2 et ainsi de suite.



 **Remarque** : Le nombre maximal d'enregistrements que `ROWNUM` peut gérer est de deux milliards.

## Macro RTRIM

La macro RTRIM est disponible dans Unica Campaign Unica Interact et .

### Syntaxe

```
RTRIM(data)
```

### Paramètres

data

### Description

`RTRIM` supprime les derniers caractères d'espacement de chaque valeur de chaîne contenue dans la plage de données spécifiée, puis renvoie la chaîne convertie. Il renvoie une nouvelle colonne pour chaque colonne d'entrée.

### Exemples

```
Temp = RTRIM "gold "
```

Crée une chaîne nommée Temp qui contient "gold".

## Macro SIGN

Le `SIGN` est disponible uniquement dans Unica Campaign.

### Syntaxe

```
SIGN(data)
```

### Paramètres

data

Valeurs numériques dont la signe doit être déterminé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

`SIGN` teste le signe des valeurs de la plage de données spécifiée. Pour chaque colonne d'entrée, `SIGN` renvoie une nouvelle colonne qui contient le signe des nombres de la colonne d'entrée correspondante. Un un positif est renvoyé pour toutes les valeurs supérieures à zéro. Un un négatif est renvoyé pour toutes les valeurs inférieures à zéro. Zéro est renvoyé pour les valeurs égales à zéro.

## Exemples

```
TEMP = SIGN(-3)
```

Crée une nouvelle colonne nommée `TEMP` contenant la valeur `-1`.

```
TEMP = SIGN(V1)
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur correspond au signe du contenu de la colonne `V1`.

```
TEMP = SIGN(V1:V3)
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux signes du contenu de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux signes du contenu de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux signes du contenu de la colonne `V3`.

```
TEMP = SIGN(V1[10:20])
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les signes des valeurs des lignes 10 à 20 de la colonne `V1`. Les autres cibles de `TEMP` sont vides.

```
TEMP = SIGN(V1[10:50]:V2)
```

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 41 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` correspondent aux signes des valeurs des lignes 10 à 50 de la colonne `V1` et les valeurs de la colonne `VX` correspondent aux signes des valeurs des lignes 10 à 50 de la colonne `V2`.

## Fonctions associées

Fonction	Description
<code>ABS</code>	Calcule la valeur absolue du contenu de la plage de données spécifiée
<code>INVERSE</code>	Calcule la valeur négative du contenu de la plage de données spécifiée

## Macro SIN

Le `SIN` est disponible uniquement dans Unica Campaign.

### Syntaxe

```
SIN(data [, units_keyword])
```

### Paramètres

`data`

Valeurs numériques dont le sinus doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

`units_keyword`

Ce mot clé facultatif détermine si les valeurs en entrée et les résultats sont interprétés comme des degrés ou des radians. Sélectionnez l'une des options suivantes :

`RADIAN` - Effectue les calculs en radians (valeur par défaut)

`DEGREE` - Effectue les calculs en degrés

Si ce paramètre n'est pas spécifié, la valeur par défaut est radians. (Pour convertir des radians en degrés, divisez par `PI` et multipliez par 180.)

Pour plus de détails sur l'utilisation des mots clés dans Unica Campaign, voir [spécifications de format \(à la page 11\)](#).

## Description

`SIN` calcule le sinus des valeurs de la plage de données spécifiée. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant le sinus des nombres de la colonne d'entrée correspondante.

## Exemples

```
TEMP = SIN(PI/2), TEMP = SIN(PI/2, 0) OU TEMP = SIGN(PI/2, RADIAN)
```

Crée une nouvelle colonne nommée `TEMP` contenant la valeur un.

```
TEMP = SIN(V1)
```

Crée une colonne nommée `TEMP`, où chaque valeur correspond au sinus (en radians) du contenu de la colonne `V1`.

```
TEMP = SIN(V1:V3, 1) OU TEMP = SIN(V1:V3, DEGREE)
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux sinus du contenu de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux sinus du contenu de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux sinus du contenu de la colonne `V3`. Toutes les valeurs sont en degrés.

```
TEMP = SIN(V1[10:50]:V2)
```

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 41 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` correspondent aux sinus des valeurs des lignes 10 à 50 de la colonne `V1` et les valeurs de la colonne `VX` correspondent aux sinus des valeurs des lignes 10 à 50 de la colonne `V2`. Toutes les valeurs sont en radians.

## Fonctions associées

Fonction	Description
<code>ASIN</code>	Calcule l'arc sinus du contenu de la plage de données spécifiée
<code>COS</code>	Calcule le cosinus du contenu de la plage de données spécifiée
<code>SINH</code>	Calcule le sinus hyperbolique du contenu de la plage de données spécifiée
<code>TAN</code>	Calcule la tangente du contenu de la plage de données spécifiée

# Macro SINH

Le `SINH` est disponible uniquement dans Unica Campaign.

## Syntaxe

```
SINH(data [, units_keyword])
```

## Paramètres

`data`

Valeurs numériques dont le sinus hyperbolique doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

`units_keyword`

Ce mot clé facultatif détermine si les valeurs en entrée et les résultats sont interprétés comme des degrés ou des radians. Sélectionnez l'une des options suivantes :

`RADIAN` - Effectue les calculs en radians (valeur par défaut)

`DEGREE` - Effectue les calculs en degrés

Si ce paramètre n'est pas spécifié, la valeur par défaut est radians. (Pour convertir des radians en degrés, divisez par `PI` et multipliez par 180.)


Pour plus de détails sur l'utilisation des mots clés dans Unica Campaign, voir [spécifications de format \(à la page 11\)](#).

## Description

`SINH` calcule le sinus hyperbolique des valeurs de la plage de données spécifiée. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant le sinus hyperbolique des nombres de la colonne d'entrée correspondante. Pour  $x$  en radians, le sinus hyperbolique d'un nombre est :

$$\sinh(x) = \frac{e^x - e^{-x}}{2}$$

où  $e$  représente le nombre naturel, 2.7182818.

 **Remarque** : Si la valeur  $x$  est trop élevée, une erreur de dépassement est renvoyée. Cela se produit si  $\sinh(x)$  dépasse la valeur maximale à virgule flottante sur 32 bits.

## Exemples

`TEMP = SINH(1), TEMP = SINH(1, 0) OU TEMP = SINH(1, RADIAN)`

Crée une nouvelle colonne nommée `TEMP` contenant la valeur 1.18.

`TEMP = SINH(V1)`

Crée une colonne nommée `TEMP`, où chaque valeur correspond au sinus hyperbolique (en radians) du contenu de la colonne `V1`.

`TEMP = SINH(V1:V3, 1) OU TEMP = SINH(V1:V3, DEGREE)`

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux sinus hyperboliques du contenu de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux sinus hyperboliques du contenu de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux sinus hyperboliques du contenu de la colonne `V3`. Toutes les valeurs sont en degrés.

`TEMP = SINH(V1[10:50]:V2)`

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 41 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` correspondent aux sinus hyperboliques des valeurs des lignes 10 à 50 de la colonne `V1` et les valeurs de la colonne `VX` correspondent aux sinus hyperboliques des valeurs des lignes 10 à 50 de la colonne `V2`. Toutes les valeurs sont en radians.

## Fonctions associées

Fonction	Description
<code>COSH</code>	Calcule le cosinus hyperbolique du contenu de la plage de données spécifiée
<code>SIN</code>	Calcule le sinus du contenu de la plage de données spécifiée
<code>TANH</code>	Calcule la tangente hyperbolique du contenu de la plage de données spécifiée

## Macro COUNT\_DIM

La macro `COUNT_DIM` est disponible dans Unica Interact.

## Syntaxe

```
COUNT_DIM(<dim field>)
```

## Paramètre

```
dim field
```

Champ dimensionnel.

## Description

Cette macro est répertoriée sous Toutes les macros prédéfinies Utilisez cette macro lorsque vous créez un diagramme interactif afin d'obtenir la taille du champ dimensionnel.Elle peut être utilisée sous les options de stratégie interactive suivantes :

- Considérer cette règle comme éligible si l'expression suivante est vraie
- Utiliser l'expression suivante comme score marketing

D'autres macros peuvent être utilisées de la même manière sous Stratégie interactive.

## Exemple

```
COUNT_DIM(inttest183_interact_pftbl_null.creditScore)
```

# Macro SORT

La macro SORT est disponible dans Unica Interact.

## Syntaxe

```
SORT(<dim field>[, <sort order>])
```

## Paramètre

```
dim field
```

Champ dimensionnel.

```
sort order
```

S'il faut trier par ordre croissant ou décroissant. Les valeurs valides sont ASC et DESC.

## Description

Cette macro est répertoriée sous Toutes les macros prédéfinies Utilisez cette macro pour trier les données du champ dimensionnel dans l'ordre naturel lors de la création d'un diagramme interactif.

## Exemple

```
SORT(inttest183_interact_pftbl_null.rank)
```

# Macro SQRT

Le `SQRT` est disponible uniquement dans Unica Campaign.

## Syntaxe

```
SQRT(data)
```


## Paramètres

`data`

Valeurs numériques dont la racine carrée doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

`SQRT` calcule la racine carrée des valeurs de la plage de données spécifiée. Pour chaque colonne d'entrée, `SQRT` renvoie une nouvelle colonne qui contient la racine carrée des nombres de la colonne d'entrée correspondante.

 **Remarque** : Si l'une des valeurs de la plage de données définie est négative, le symbole ??? est renvoyé pour cette cible.

## Exemples

```
TEMP = SQRT(2)
```



Crée une nouvelle colonne nommée `TEMP` contenant la valeur 1.41.

```
TEMP = SQRT(V1)
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur correspond à la racine carrée du contenu de la colonne `V1`.

```
TEMP = SQRT(V1:V3)
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux racines carrées du contenu de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux racines carrées du contenu de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux racines carrées du contenu de la colonne `V3`.

```
TEMP = SQRT(V1[10:20])
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les racines carrées des valeurs des lignes 10 à 20 de la colonne `V1`. Les autres cibles de `TEMP` sont vides.

```
TEMP = SQRT(V1[10:50]:V2)
```

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 41 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` correspondent aux racines carrées des valeurs des lignes 10 à 50 de la colonne `V1` et les valeurs de la colonne `VX` correspondent aux racines carrées des valeurs des lignes 10 à 50 de la colonne `V2`.

## Fonctions associées

Function	Description
DIV	Divise une plage de données spécifiée par une autre
MULT	Multiplie le contenu de deux plages de données
POW	Calcule une valeur de base élevée aux puissances exponentielles spécifiées

## Macro STDV ou STDEV

`STDV` ou `STDEV` est disponible dans Unica Campaign et Unica Interact.

### Syntaxe

```
STDV(data [, keyword]) STDEV(data [, keyword])
```

## Paramètres

`data`

Valeurs numériques dont l'écart type doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

`keyword`


Ce mot clé facultatif détermine la manière dont le calcul est effectué sur la plage de données d'entrée. Sélectionnez l'une des options suivantes :

`ALL` - Effectue le calcul sur toutes les cibles de `data` (valeur par défaut)

`COL` - Effectue le calcul séparément pour chaque colonne de `data`

`ROW` - Effectue le calcul séparément pour chaque ligne de `data`

Pour plus de détails sur l'utilisation des mots clés dans **Unica Campaign**, voir [spécifications de format \(à la page 11\)](#).


 **Remarque** : De nombreuses fonctions de macro utilisent les paramètres de mot clé {`ALL` | `COL` | `ROW`}. Ces mots clés ne s'appliquent pas dans **Unica Campaign** car les données d'entrées correspondent toujours à une colonne ou à une zone. La macro se comporte toujours comme si le mot clé `COL` a été spécifié. Par conséquent, vous n'avez pas besoin de spécifier ces mots clés lorsque vous utilisez **Unica Campaign** .

## Description

`STDV` calcule l'écart type de toutes les cibles de la plage de données spécifiée. L'écart type d'une distribution est la racine carrée de la variance. L'écart type est calculé comme suit :

$$\sqrt{\frac{1}{n-1} \sum_{j=1}^n (x_j - \text{mean})^2}$$

où les  $x$  représentent les échantillons,  $n$  représente le nombre d'échantillons et `mean` est la moyenne de la distribution.

 **Remarque** : Si le nombre d'échantillons  $n = 1$ , `STDV` renvoie une erreur.

## Exemples

```
TEMP = STDV(V1)
```

Crée une colonne nommée `TEMP` contenant une valeur unique qui correspond à l'écart type du contenu de la colonne `V1`.

```
TEMP = STDV(V1:V3)
```

Crée une colonne nommée `TEMP` contenant une valeur unique qui correspond à l'écart type du contenu des colonnes `V1`, `V2` et `V3`.

```
TEMP = STDV(V1[1:5]:V4)
```

Crée une colonne nommée `TEMP` contenant une valeur unique qui correspond à l'écart type des cibles des lignes 1 à 5 des colonnes `V1` à `V4`.

```
TEMP = STDV(V1:V3, COL)
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. La valeur unique de la colonne `TEMP` correspond à l'écart type du contenu de la colonne `V1`, la valeur unique de la colonne `VX` correspond à l'écart type du contenu de la colonne `V2` et la valeur unique de la colonne `VY` correspond à l'écart type du contenu de la colonne `V3`.

```
TEMP = STDV(V1[10:50]:V3, COL)
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY` contenant chacune une seule valeur. La valeur de la colonne `TEMP` correspond à l'écart type des cibles des lignes 10 à 50 de la colonne `V1`, la valeur de la colonne `VX` correspond à l'écart type des cibles des lignes 10 à 50 de la colonne `V2` et la valeur de la colonne `VY` correspond à l'écart type des cibles des lignes 10 à 50 de la colonne `V3`.

```
TEMP = STDV(V1:V3, ROW)
```

Crée une colonne nommée `TEMP` dans laquelle chaque entrée de cible correspond à l'écart type de la ligne correspondante dans les colonnes `V1`, `V2` et `V3`.

```
TEMP = STDV(V1[1:5]:V3, ROW)
```

Crée une colonne nommée `TEMP` dans laquelle les cibles des lignes 1 à 5 contiennent l'écart type de la ligne correspondante dans les colonnes `V1` à `V3`. Les autres cibles de `TEMP` sont vides.

## Fonctions associées

Fonction	Description
VAR	Calcule la variance d'une plage de cibles

## Macro STRING\_CONCAT

Le `STRING_CONCAT` est disponible dans Unica Campaign et Unica Interact.


### Syntaxe

```
STRING_CONCAT(string1, string2, ... stringN)
```

### Paramètres


`string`

Chaîne de texte ASCII à concaténer. Il peut s'agir de texte ASCII entre guillemets, d'une colonne de texte, d'une plage de cibles contenant du texte ou d'une expression convertie comme l'une de ces valeurs. Si ce paramètre est une valeur numérique ou de date et heure, il est converti en chaîne à l'aide du format par défaut dans le serveur d'exécution Interact. Pour la définition du format de chaîne (identique à `data`), voir la section « Paramètres des fonctions macro ».

 **Remarque** : Les valeurs numériques et date/heure peuvent être directement transmises à cette macro dans toutes les zones Interact. Toutefois, la validation échoue si elle est utilisée dans un diagramme en temps réel.

### Description

`STRING_CONCAT` concatène les valeurs de texte ASCII de la plage de données spécifiée. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant les chaînes concaténées des lignes correspondantes de `strings`. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

 **Remarque** : La longueur totale de chaque chaîne résultante ne peut pas dépasser 255 caractères.

Unica Interact prend également en charge la syntaxe suivante :

```
STRING_CONCAT( string1 , string2 , ... stringN )
```

Par exemple `STRING_CONCAT('a', 'b', 'c', 'd')` est valide.

## Exemples

```
TEMP = STRING_CONCAT("house", "boat")
```

Crée une colonne nommée `TEMP`, qui contient la chaîne de texte ASCII "houseboat".

```
TEMP = STRING_CONCAT(V1, ".")
```

Crée une colonne nommée `TEMP`, chacune de ses lignes contenant la chaîne de texte ASCII de la ligne correspondante de la colonne `V1` à laquelle un point a été ajouté.

```
TEMP = STRING_CONCAT(V1, V2)
```

Crée une colonne nommée `TEMP` chacune de ses lignes contenant la chaîne de texte ASCII de la colonne `V1` concaténée avec la chaîne de texte de la colonne `V2`.

```
TEMP = STRING_CONCAT(V1:V3, V4:V6)
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux chaînes concaténées des lignes correspondantes des colonnes `V1` et `V4`, les valeurs de la colonne `VX` correspondent aux chaînes concaténées des lignes correspondantes des colonnes `V2` et `V5` et les valeurs de la colonne `VY` correspondent aux chaînes concaténées des lignes correspondantes `V3` et `V6`.

```
TEMP = STRING_CONCAT(V1[5:10]:V2, V3:V4)
```

Crée deux colonnes nommées `TEMP` et `VX`. Les valeurs de la colonne `TEMP` correspondent aux chaînes des lignes 5 à 10 de la colonne `V1` concaténées avec les lignes 1 à 6 de la colonne `V3`. Les valeurs de `VX` correspondent aux chaînes des lignes 5 à 10 de la colonne `V2` concaténées avec les lignes 1 à 6 de la colonne `V4`.

```
TEMP = STRING_CONCAT('a', 'b', 'c', 'd')
```

Crée une colonne nommée `TEMP`, qui contient la chaîne de texte ASCII "abcd".

## Fonctions associées

Fonction	Description
<code>STRING_HEAD</code>	Renvoie les n premiers caractères de chaque chaîne de la plage de données spécifiée
<code>STRING_LENGTH</code>	Renvoie la longueur de chaque chaîne de la plage de données spécifiée
<code>STRING_SEG</code>	Renvoie le segment de chaîne entre deux index spécifiés
<code>STRING_TAIL</code>	Renvoie les n derniers caractères de chaque chaîne de la plage de données spécifiée

## Macro `STRING_HEAD`

Le `STRING_HEAD` est disponible uniquement dans Unica Campaign.

### Syntaxe

```
STRING_HEAD(num_chars, data)
```

### Paramètres

`num_chars`

Nombre de caractères à renvoyer à partir du début de chaque chaîne de `data`. Il doit s'agir d'un entier positif supérieur à zéro.

`data`

Valeurs de chaîne de texte ASCII. Il peut s'agir de texte ASCII entre guillemets, d'une colonne de texte, d'une plage de cibles contenant du texte ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

### Description

`STRING_HEAD` renvoie les `num_chars` premiers caractères de chaque valeur de chaîne de la plage de données spécifiée. Si `num_chars` est supérieur au nombre de caractères d'une chaîne de texte, les caractères restants sont remplacés par le caractère null " \0 ".

## Exemples

```
TEMP = STRING_HEAD(3, "JAN 15, 1997")
```

Crée une colonne nommée `TEMP`, qui contient la chaîne de texte ASCII " JAN ".

```
TEMP = STRING_HEAD(10, "Pressure")
```

Crée une colonne nommée `TEMP`, qui contient la chaîne de texte ASCII " Pressure ".

```
TEMP = STRING_HEAD(5, V1)
```

Crée une colonne nommée `TEMP` contenant les cinq premiers caractères de chaque chaîne de la colonne `V1`.

```
TEMP = STRING_HEAD(1, V1:V3)
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux premiers caractères des chaînes des lignes correspondantes de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux premiers caractères des chaînes des lignes correspondantes de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux premiers caractères des chaînes des lignes correspondantes de la colonne `V3`.

```
TEMP = STRING_HEAD(12, V4[1:50]:V6]
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux 12 premiers caractères des chaînes des lignes 1 à 50 de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux 12 premiers caractères des chaînes des lignes 1 à 50 de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux 12 premiers caractères des chaînes des lignes 1 à 50 de la colonne `V3`.

## Fonctions associées

Fonction	Description
<code>STRING_CONCAT</code>	Concatène deux chaînes de texte des plages de données spécifiées
<code>STRING_LENGTH</code>	Renvoie la longueur de chaque chaîne de la plage de données spécifiée
<code>STRING_SEG</code>	Renvoie le segment de chaîne entre deux index spécifiés
<code>STRING_TAIL</code>	Renvoie les n derniers caractères de chaque chaîne de la plage de données spécifiée

# Macro STRING\_LENGTH

Le `STRING_LENGTH` est disponible uniquement dans Unica Campaign.

## Syntaxe

```
STRING_LENGTH(data)
```


## Paramètres

`data`

Valeurs de chaîne de texte ASCII dont la longueur doit être calculée. Il peut s'agir de texte ASCII entre guillemets, d'une colonne de texte, d'une plage de cibles contenant du texte ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

`STRING_LENGTH` renvoie la longueur de chaque valeur de chaîne dans la plage de données spécifiée. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant la longueur de la chaîne de texte correspondante.

 **Remarque** : Si `STRING_LENGTH` est appliqué à des colonnes contenant des données numériques, il renvoie des zéros.

## Exemples

```
TEMP = STRING_LENGTH("four")
```

Crée une nouvelle colonne nommée `TEMP` contenant la valeur 4.

```
TEMP = STRING_LENGTH(4)
```

Crée une nouvelle colonne nommée `TEMP` contenant la valeur 0.

```
TEMP = STRING_LENGTH(V1)
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur correspond à la longueur de la chaîne dans la ligne correspondante de la colonne `V1`.

```
TEMP = STRING_LENGTH(V1:V3)
```



Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux longueurs des chaînes des lignes correspondantes de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux longueurs des chaînes des lignes correspondantes de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux longueurs des chaînes des lignes correspondantes de la colonne `V3`.

```
TEMP = STRING_LENGTH(V4[1:50]:V6]
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux longueurs des chaînes des lignes 1 à 50 de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux longueurs des chaînes des lignes 1 à 50 de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux longueurs des chaînes des lignes 1 à 50 de la colonne `V3`.

## Fonctions associées

Fonction	Description
<code>STRING_CONCAT</code>	Concatène deux chaînes de texte des plages de données spécifiées
<code>STRING_HEAD</code>	Renvoie les n premiers caractères de chaque chaîne de la plage de données spécifiée
<code>STRING_SEG</code>	Renvoie le segment de chaîne entre deux index spécifiés
<code>STRING_TAIL</code>	Renvoie les n derniers caractères de chaque chaîne de la plage de données spécifiée

## Macro `STRING_PROPER`

Le `STRING_PROPER` est disponible uniquement dans Unica Campaign.

### Syntaxe

```
STRING_PROPER(data)
```

### Paramètres

`data`

Valeur de chaîne à convertir.

## Description

`STRING_PROPER` convertit chaque valeur de chaîne contenue dans la plage de données spécifiée en mettant en majuscule la première lettre de chaque chaîne ou toute lettre suivant un espace blanc ou un symbole (autre que le trait de soulignement) et transforme en minuscules tous les autres caractères. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant la chaîne convertie dans la colonne d'entrée correspondante.

## Exemples

```
Temp = STRING_PROPER
```

## Macro STRING\_SEG

Le `STRING_SEG` est disponible uniquement dans Unica Campaign.

## Syntaxe

```
STRING_SEG(from, to, data)
```

## Paramètres

*from*

Nombre de caractères depuis le début de la chaîne à partir duquel le segment de chaîne doit commencer à être extrait. Il doit s'agir d'un entier positif supérieur à zéro et inférieur à *to* ou `STRING_SEG` renvoie une chaîne vide.

*to*

Nombre de caractères depuis le début de la chaîne à partir duquel le segment de chaîne ne doit plus être extrait. Il doit s'agir d'un entier positif supérieur ou égal à *from*. Si *to* est égal à *from* (et *to* est inférieur ou égal à la longueur de la chaîne), un caractère est renvoyé.

*data*

Valeurs de chaîne de texte ASCII. Il peut s'agir de texte ASCII entre guillemets, d'une colonne de texte, d'une plage de cibles contenant du texte ou d'une expression convertie comme

l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

`STRING_SEG` renvoie le segment de chaîne entre deux index de chaque valeur de chaîne dans la plage de données spécifiée. Si `from` est supérieur à la longueur d'une chaîne, aucune valeur n'est renvoyée. Si `to` est supérieur à la longueur d'une chaîne, tous les caractères de `from` sont renvoyés.

## Exemples

```
TEMP = STRING_SEG(1, 6, "JAN 15, 1997")
```

Crée une colonne nommée `TEMP`, qui contient la chaîne de texte ASCII " Jan 15 ".

```
TEMP = STRING_SEG(5, 20, "Pressure")
```

Crée une colonne nommée `TEMP`, qui contient la chaîne de texte ASCII " sure ".

```
TEMP = STRING_SEG(5, 6, V1)
```

Crée une colonne nommée `TEMP` contenant les cinquième et sixième caractères de chaque chaîne de la colonne `V1`.

```
TEMP = STRING_SEG(10, 20, V1:V3)
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux caractères 10 à 20 des chaînes des lignes correspondantes de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux caractères 10 à 20 des chaînes des lignes correspondantes de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux caractères 10 à 20 des chaînes des lignes correspondantes de la colonne `V3`.

```
TEMP = STRING_SEG(5, 10, V4[1:50]:V6]
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux caractères 5 à 10 des chaînes des lignes 1 à 50 de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux caractères 5 à 10 des chaînes des lignes 1 à 50 de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux caractères 5 à 10 des chaînes des lignes 1 à 50 de la colonne `V3`.

## Fonctions associées

Fonction	Description
<code>STRING_CONCAT</code>	Concatène deux chaînes de texte des plages de données spécifiées
<code>STRING_HEAD</code>	Renvoie les n premiers caractères de chaque chaîne de la plage de données spécifiée
<code>STRING_LENGTH</code>	Renvoie la longueur de chaque chaîne de la plage de données spécifiée
<code>STRING_TAIL</code>	Renvoie les n derniers caractères de chaque chaîne de la plage de données spécifiée

## Macro `STRING_TAIL`

Le `STRING_TAIL` est disponible uniquement dans Unica Campaign.

### Syntaxe

```
STRING_TAIL(num_chars, data)
```

### Paramètres

`num_chars`

Nombre de caractères à renvoyer à partir de la fin de chaque chaîne de `data`. Il doit s'agir d'un entier positif supérieur à zéro.

`data`

Valeurs de chaîne de texte ASCII. Il peut s'agir de texte ASCII entre guillemets, d'une colonne de texte, d'une plage de cibles contenant du texte ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

### Description

`STRING_TAIL` renvoie les `num_chars` derniers caractères de chaque valeur de chaîne de la plage de données spécifiée. Toutes les valeurs de chaîne sont remplies jusqu'à la longueur de la plus longue chaîne avec des caractères nuls "`\0`". Ensuite, les `num_chars` derniers

caractères sont renvoyés de chaque chaîne. Si `num_chars` est supérieur au nombre de caractères d'une chaîne de texte, l'intégralité de la chaîne de texte est renvoyée.

## Exemples

```
TEMP = STRING_TAIL(3, "JAN 15, 1997")
```

Crée une colonne nommée `TEMP`, qui contient la chaîne de texte ASCII " 997 ".

```
TEMP = STRING_TAIL(10, "Pressure")
```

Crée une colonne nommée `TEMP`, qui contient la chaîne de texte ASCII " Pressure ".

```
TEMP = STRING_TAIL(5, V1)
```

Crée une colonne nommée `TEMP` contenant les cinq derniers caractères de chaque chaîne de la colonne `V1`.

```
TEMP = STRING_TAIL(1, V1:V3)
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux derniers caractères des chaînes des lignes correspondantes de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux derniers caractères des chaînes des lignes correspondantes de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux derniers caractères des chaînes des lignes correspondantes de la colonne `V3`.

```
TEMP = STRING_TAIL(12, V4[1:50]:V6]
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux 12 derniers caractères des chaînes des lignes 1 à 50 de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux 12 derniers caractères des chaînes des lignes 1 à 50 de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux 12 derniers caractères des chaînes des lignes 1 à 50 de la colonne `V3`.

## Fonctions associées

Fonction	Description
<code>STRING_CONCAT</code>	Concatène deux chaînes de texte des plages de données spécifiées
<code>STRING_HEAD</code>	Renvoie les n premiers caractères de chaque chaîne de la plage de données spécifiée
<code>STRING_LENGTH</code>	Renvoie la longueur de chaque chaîne de la plage de données spécifiée
<code>STRING_SEG</code>	Renvoie le segment de chaîne entre deux index spécifiés

# Macro SUBSTR ou SUBSTRING

SUBSTR OU SUBSTRING est disponible dans Unica Campaign et Unica Interact.

## Syntaxe

```
SUBSTR(string_value, start_pos[, nchars]) or SUBSTR(string_value FROM
start_pos[ FOR nchars]) SUBSTRING(string_value, start_pos[, nchars]) or
SUBSTRING(string_value FROM start_pos[ FOR nchars])
```

## Paramètres

*string\_value*

Chaîne à partir de laquelle une sous-chaîne est extraite.

*start\_pos*

Caractère de début à partir duquel la sous-chaîne doit être extraite.

*nchars*

Nombre de caractères à extraire (doit être supérieur ou égal à 0). Si cette valeur n'est pas fournie, tous les caractères restants de *string\_value* sont extraits.

## Description

SUBSTR OU SUBSTRING extrait *nchars* caractères de la chaîne, à partir de *start\_pos*. Si *nchars* est omis, SUBSTR et SUBSTRING extraient les caractères de *start\_pos* jusqu'à la fin de la chaîne. Les espaces de fin sont automatiquement tronqués. Pour éviter les erreurs de syntaxe, veillez à séparer les valeurs numériques par une virgule et un espace, comme le montrent les exemples.

**⚠ Important :** Unica Interact prend uniquement en charge les types de fichier suivants :

```
SUBSTR(string_value, start_pos[, nchars]) OU SUBSTRING(string_value,
start_pos[, nchars])
```

## Exemples

```
SUBSTR SUBSTR Returns ("abcdef" FROM 1 FOR 2) ("abcdef", 1, 2) 'ab'
```

SUBSTR SUBSTR Returns ("abcdef" FROM -2 FOR 4) ("abcdef", -2, 4) 'a'

SUBSTR SUBSTR Returns ("abcdef" FROM 3) ("abcdef", 3) 'cdef'

## Macro SUM

Le `SUM` est disponible dans Unica Campaign et Unica Interact.

### Syntaxe

`SUM(data [, keyword])`

### Paramètres

`data`

Valeurs numériques dont la somme doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

`keyword`


Ce mot clé facultatif détermine la manière dont le calcul est effectué sur la plage de données d'entrée. Sélectionnez l'une des options suivantes :

`ALL` - Effectue le calcul sur toutes les cibles de `data` (valeur par défaut)

`COL` - Effectue le calcul séparément pour chaque colonne de `data`

`ROW` - Effectue le calcul séparément pour chaque ligne de `data`

Pour plus de détails sur l'utilisation des mots clés dans Unica Campaign, voir [spécifications de format \(à la page 11\)](#).

 **Remarque** : De nombreuses fonctions de macro utilisent les paramètres de mot clé {`ALL` | `COL` | `ROW`}. Ces mots clés ne s'appliquent pas dans Unica Campaign car les données d'entrées correspondent toujours à une colonne ou à une zone. La macro se comporte toujours comme si le mot clé `COL` a été spécifié. Par conséquent, vous n'avez pas besoin de spécifier ces mots clés lorsque vous utilisez Unica Campaign.

## Description

`SUM` calcule la somme de toutes les cibles de la plage de données spécifiée. Il renvoie une colonne unique.

 **Remarque** : `SUM` est identique à la fonction macro `TOTAL`.

## Exemples

`TEMP = SUM(3)`

Crée une nouvelle colonne nommée `TEMP` contenant la valeur trois.

`TEMP = SUM((COLUMN(3, 5, 1)))`

Crée une nouvelle colonne nommée `TEMP` contenant la valeur neuf.

`TEMP = SUM(V1)`

Crée une colonne nommée `TEMP` contenant une valeur unique qui correspond à la somme du contenu de la colonne `V1`.

`TEMP = SUM(V1:V3)`

Crée une colonne nommée `TEMP` contenant une valeur unique qui correspond à la somme du contenu des colonnes `V1`, `V2` et `V3`.

`TEMP = SUM(V1[1:5]:V4)`

Crée une colonne nommée `TEMP` contenant une valeur unique qui correspond à la somme des cibles des lignes 10 à 20 des colonnes `V1` à `V4`.

`TEMP = SUM(V1:V3, COL)`

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. La valeur unique de la colonne `TEMP` correspond à la valeur minimale du contenu de la colonne `V1`, la valeur unique de la colonne `VX` correspond à la valeur minimale du contenu de la colonne `V2` et la valeur unique de la colonne `VY` correspond à la valeur minimale du contenu de la colonne `V3`.

`TEMP = SUM(V1[1:5]:V3, COL)`

Crée trois colonnes nommées `TEMP`, `VX` et `VY` contenant chacune une seule valeur. La valeur de la colonne `TEMP` correspond à la somme des cibles des lignes 1 à 5 de la colonne `V1`, la valeur de la colonne `VX` correspond à la somme des cibles des lignes 1 à 5 de la colonne



V2, et la valeur de la colonne `VY` correspond à la somme des cibles des lignes 1 à 5 de la colonne `V3`.

```
TEMP = SUM(V1:V3, ROW)
```

Crée une colonne nommée `TEMP` dans laquelle chaque entrée de cible correspond à la somme de la ligne correspondante dans les colonnes `V1`, `V2` et `V3`.

```
TEMP = SUM(V1[1:5]:V3, ROW)
```

Crée une colonne nommée `TEMP`, dans laquelle les cibles des lignes 1 à 5 contiennent la somme de la ligne correspondante dans les colonnes `V1` à `V3`. Les autres cibles de `TEMP` sont vides.

## Fonctions associées

Fonction	Description
<code>AVG</code> ou <code>MEAN</code>	Calcule la moyenne arithmétique d'une plage de cibles

## Macro TAN

Le `TAN` est disponible uniquement dans Unica Campaign.

### Syntaxe

```
TAN(data [, units_keyword])
```

### Paramètres

`data`

Valeurs numériques dont la tangente doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

`units_keyword`

Ce mot clé facultatif détermine si les valeurs en entrée et les résultats sont interprétés comme des degrés ou des radians. Sélectionnez l'une des options suivantes :

**RADIAN** - Effectue les calculs en radians (valeur par défaut)

**DEGREE** - Effectue les calculs en degrés

Si ce paramètre n'est pas spécifié, la valeur par défaut est radians. (Pour convertir des radians en degrés, divisez par `PI` et multipliez par 180.)

Pour plus de détails sur l'utilisation des mots clés dans Unica Campaign, voir [spécifications de format \(à la page 11\)](#).

## Description

`TAN` calcule la tangente des valeurs de la plage de données spécifiée. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant la tangente des nombres de la colonne d'entrée correspondante.

## Exemples

`TEMP = TAN(PI/4), TEMP = TAN(PI/4, 0) OU TEMP = TAN(PI/4, RADIAN)`

Crée une nouvelle colonne nommée `TEMP` contenant la valeur un.

`TEMP = TAN(V1)`

Crée une colonne nommée `TEMP`, où chaque valeur correspond à la tangente (en radians) du contenu de la colonne `V1`.

`TEMP = TAN(V1:V3, 1) OU TEMP = TAN(V1:V3, DEGREE)`

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux tangentes du contenu de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux tangentes du contenu de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux tangentes du contenu de la colonne `V3`. Toutes les valeurs sont en degrés.

`TEMP = TAN(V1[1:5]:V2)`

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` correspondent aux tangentes des lignes correspondantes de la colonne `V1` et les valeurs de la colonne `VX` correspondent aux tangentes des lignes correspondantes de la colonne `V2`. Toutes les valeurs sont en radians.

## Fonctions associées

Fonction	Description
ATAN	Calcule l'arc tangente du contenu de la plage de données spécifiée
COS	Calcule le cosinus du contenu de la plage de données spécifiée
COT	Calcule la cotangente du contenu de la plage de données spécifiée
SIN	Calcule le sinus du contenu de la plage de données spécifiée
TANH	Calcule la tangente hyperbolique du contenu de la plage de données spécifiée

## Macro TANH

La macro TANH est disponible uniquement dans Unica Campaign .

### Syntaxe

```
TANH(data [, units_keyword])
```

### Paramètres

`data`

Valeurs numériques dont la tangente hyperbolique doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

`units_keyword`

Ce mot clé facultatif détermine si les valeurs en entrée et les résultats sont interprétés comme des degrés ou des radians. Sélectionnez l'une des options suivantes :

`RADIAN` - Effectue les calculs en radians (valeur par défaut)

`DEGREE` - Effectue les calculs en degrés


Si ce paramètre n'est pas spécifié, la valeur par défaut est radians. (Pour convertir des radians en degrés, divisez par  $\pi$  et multipliez par 180.)

Pour plus de détails sur l'utilisation des mots clés dans Unica Campaign, voir [spécifications de format \(à la page 11\)](#).

## Description

`TANH` calcule la tangente hyperbolique des valeurs de la plage de données spécifiée. Pour chaque colonne d'entrée, `TANH` renvoie une nouvelle colonne qui contient la tangente hyperbolique des nombres de la colonne d'entrée correspondante. La tangente hyperbolique d'un nombre est calculée comme suit :

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)}$$

 **Remarque** : Si la valeur  $x$  est trop élevée, une erreur de dépassement est renvoyée. Cela se produit si  $\tanh(x)$  dépasse la valeur maximale à virgule flottante sur 32 bits. Si  $\cosh(x)$  est égal à zéro, `TANH` renvoie la valeur maximale à virgule flottante sur 32 bits.

## Exemples

`TEMP = TANH(PI), TEMP = TANH(PI, 0) OU TEMP = TANH(PI, RADIAN)`

Crée une nouvelle colonne nommée `TEMP` contenant la valeur un.

`TEMP = TANH(V1)`

Crée une colonne nommée `TEMP`, où chaque valeur correspond à la tangente hyperbolique (en radians) du contenu de la colonne `V1`.

`TEMP = TANH(V1:V3, 1) OU TEMP = TANH(V1:V3, DEGREE)`

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux tangentes hyperboliques du contenu de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux tangentes hyperboliques du contenu de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux tangentes hyperboliques du contenu de la colonne `V3`.

Toutes les valeurs sont en degrés.

`TEMP = TANH(V1[1:5]:V2)`

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` correspondent aux tangentes hyperboliques des lignes correspondantes de la colonne `V1` et les valeurs de la colonne `VX` correspondent aux tangentes hyperboliques des lignes correspondantes de la colonne `V2`. Toutes les valeurs sont en radians.

## Fonctions associées

Fonction	Description
ATAN	Calcule l'arc tangente du contenu de la plage de données spécifiée
COSH	Calcule le cosinus hyperbolique du contenu de la plage de données spécifiée
COT	Calcule la cotangente du contenu de la plage de données spécifiée
SINH	Calcule le sinus hyperbolique du contenu de la plage de données spécifiée
TAN	Calcule la tangente du contenu de la plage de données spécifiée

## Macro TOTAL

Le `TOTAL` est disponible dans Unica Campaign et Unica Interact.

### Syntaxe

`TOTAL(data [, keyword])`

### Paramètres

`data`

Valeurs numériques dont la somme doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

`keyword`


Ce mot clé facultatif détermine la manière dont le calcul est effectué sur la plage de données d'entrée. Sélectionnez l'une des options suivantes :

`ALL` - Effectue le calcul sur toutes les cibles de `data` (valeur par défaut)

`COL` - Effectue le calcul séparément pour chaque colonne de `data`

`ROW` - Effectue le calcul séparément pour chaque ligne de `data`

Pour plus de détails sur l'utilisation des mots clés dans Unica Campaign, voir [spécifications de format \(à la page 11\)](#).

 **Remarque** : De nombreuses fonctions de macro utilisent les paramètres de mot clé {ALL | COL | ROW}. Ces mots clés ne s'appliquent pas dans Unica Campaign car les données d'entrées correspondent toujours à une colonne ou à une zone. La macro se comporte toujours comme si le mot clé COL a été spécifié. Par conséquent, vous n'avez pas besoin de spécifier ces mots clés lorsque vous utilisez Unica Campaign.

## Description

TOTAL calcule la somme de toutes les cibles de la plage de données spécifiée.

 **Remarque** : TOTAL est identique à la fonction macro SUM.

## Exemples

TEMP = TOTAL(3)

Crée une nouvelle colonne nommée TEMP contenant la valeur trois.

TEMP = TOTAL((COLUMN(3, 5, 1)))

Crée une nouvelle colonne nommée TEMP contenant la valeur neuf.

TEMP = TOTAL(V1)

Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la somme du contenu de la colonne V1.

TEMP = TOTAL(V1:V3)

Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la somme du contenu des colonnes V1, V2 et V3.

TEMP = TOTAL(V1[1:5]:V4)

Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la somme des cibles des lignes 10 à 20 des colonnes V1 à V4.

TEMP = TOTAL(V1:V3, COL)

Crée trois colonnes nommées TEMP, VX et VY. La valeur unique de la colonne TEMP correspond à la valeur minimale du contenu de la colonne V1, la valeur unique de la colonne VX correspond à la valeur minimale du contenu de la colonne V2 et la valeur unique de la colonne VY correspond à la valeur minimale du contenu de la colonne V3.

TEMP = TOTAL(V1[1:5]:V3, COL)

Crée trois colonnes nommées `TEMP`, `VX` et `VY` contenant chacune une seule valeur. La valeur de la colonne `TEMP` correspond à la somme des cibles des lignes 1 à 5 de la colonne `V1`, la valeur de la colonne `VX` correspond à la somme des cibles des lignes 1 à 5 de la colonne `V2`, et la valeur de la colonne `VY` correspond à la somme des cibles des lignes 1 à 5 de la colonne `V3`.

```
TEMP = TOTAL(V1:V3, ROW)
```

Crée une colonne nommée `TEMP` dans laquelle chaque entrée de cible correspond à la somme de la ligne correspondante dans les colonnes `V1`, `V2` et `V3`.

```
TEMP = TOTAL(V1[1:5]:V3, ROW)
```

Crée une colonne nommée `TEMP`, dans laquelle les cibles des lignes 1 à 5 contiennent la somme de la ligne correspondante dans les colonnes `V1` à `V3`. Les autres cibles de `TEMP` sont vides.

## Fonctions associées

Fonction	Description
<code>AVG</code> ou <code>MEAN</code>	Calcule la moyenne arithmétique d'une plage de cibles

## Macro TRUNCATE

Le `TRUNCATE` est disponible uniquement dans Unica Campaign.

### Syntaxe

```
TRUNCATE(data)
```


### Paramètres

`data`

Valeurs numériques à tronquer. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

## Description

`TRUNCATE` calcule la partie entière de chaque valeur dans la plage de données spécifiée. Pour chaque colonne d'entrée, `TRUNCATE` renvoie une nouvelle colonne qui contient la partie entière (non fractionnaire) des nombres de la colonne d'entrée correspondante.

 **Remarque** : Les fonctions macro `FRACTION` et `TRUNCATE` sont complémentaires car leur somme correspond aux valeurs d'origine.

## Exemples

```
TEMP = TRUNCATE(4.3)
```

Crée une nouvelle colonne nommée `TEMP` contenant la valeur 4.

```
TEMP = TRUNCATE(2.9)
```

Crée une nouvelle colonne nommée `TEMP` contenant la valeur -2.

```
TEMP = TRUNCATE(V1)
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur correspond à la partie décimale du contenu de la colonne `V1`.

```
TEMP = TRUNCATE(V1:V3)
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. Les valeurs de la colonne `TEMP` correspondent aux parties tronquées de la colonne `V1`, les valeurs de la colonne `VX` correspondent aux parties tronquées de la colonne `V2` et les valeurs de la colonne `VY` correspondent aux parties tronquées de la colonne `V3`.

```
TEMP = TRUNCATE(V1[10:20])
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent les parties tronquées des valeurs des lignes 10 à 20 de la colonne `V1`. Les autres cibles de `TEMP` sont vides.

```
TEMP = TRUNCATE(V1[50:99]:V2)
```

Crée deux colonnes nommées `TEMP` et `VX`, contenant des valeurs dans les lignes 1 à 50 (les autres cibles sont vides). Les valeurs de la colonne `TEMP` correspondent aux parties tronquées des lignes de la colonne `V1` et les valeurs de la colonne `VX` correspondent aux parties tronquées des valeurs de la colonne `V2`.



## Fonctions associées

Fonction	Description
CEILING	Calcule le plafond de chaque valeur de la plage de données spécifiée
FLOOR	Calcule la partie entière de chaque valeur de la plage de données spécifiée
FRACTION	Renvoie la partie décimale de chaque valeur de la plage de données spécifiée

## Macro UPPER

Le `UPPER` est disponible dans Unica Campaign et Unica Interact.

### Syntaxe

`UPPER(data)`

### Paramètres

`data`

Valeur de chaîne à convertir en majuscules.

### Description

`UPPER` convertit chaque valeur de chaîne de la plage de données spécifiée en majuscules. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant la chaîne en majuscules dans la colonne d'entrée correspondante.

### Exemples

```
Temp = UPPER "gold"
```

Crée une colonne nommée `Temp` contenant "GOLD".

```
TEMP = UPPER( "jan 15, 1997")
```

Crée une colonne nommée `TEMP`, qui contient la chaîne de texte ASCII "JAN 15, 1997".

```
TEMP = UPPER( "Pressure")
```

Crée une colonne nommée `TEMP`, qui contient la chaîne de texte ASCII "PRESSURE".

```
TEMP = UPPER(V1)
```

Crée une colonne nommée `TEMP` contenant les caractères en majuscules de chaque chaîne de la colonne `v1`.

## Macro VARIANCE

Le `VARIANCE` est disponible uniquement dans Unica Campaign.

### Syntaxe

```
VARIANCE(data [, keyword])
```

### Paramètres

`data`

Valeurs numériques dont la variance doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

`keyword`


Ce mot clé facultatif détermine la manière dont le calcul est effectué sur la plage de données d'entrée. Sélectionnez l'une des options suivantes :

`ALL` - Effectue le calcul sur toutes les cibles de `data` (valeur par défaut)

`COL` - Effectue le calcul séparément pour chaque colonne de `data`

`ROW` - Effectue le calcul séparément pour chaque ligne de `data`

Pour plus de détails sur l'utilisation des mots clés dans Unica Campaign, voir [spécifications de format \(à la page 11\)](#).

 **Remarque** : De nombreuses fonctions de macro utilisent les paramètres de mot clé {`ALL` | `COL` | `ROW`}. Ces mots clés ne s'appliquent pas dans **Unica Campaign** car les données d'entrées correspondent toujours à une colonne ou à une zone. La macro se comporte toujours comme si le mot clé `COL` a été spécifié. Par conséquent, vous n'avez pas besoin de spécifier ces mots clés lorsque vous utilisez **Unica Campaign** .

## Description

VARIANCE calcule la variance de toutes les valeurs de la plage de données spécifiée. La variance est égale au carré de l'écart type. La variance est calculée comme suit :

$$\frac{1}{n-1} \sum_{j=1}^n (x_j - \text{mean})^2$$

où les  $x$  représentent les échantillons,  $n$  représente le nombre d'échantillons et  $\text{mean}$  est la moyenne de la distribution.

 **Remarque :** Si le nombre d'échantillons  $n = 1$ , VARIANCE renvoie une erreur.

## Exemples

```
TEMP = VARIANCE(V1)
```

Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la variance du contenu de la colonne V1.

```
TEMP = VARIANCE(V1:V3)
```

Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la variance du contenu des colonnes V1, V2 et V3.

```
TEMP = VARIANCE(V1[10:20])
```

Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la variance des cibles des lignes 10 à 20 de la colonne V1.

```
TEMP = VARIANCE(V1[1:5]:V4)
```

Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la variance des cibles des lignes 1 à 5 des colonnes V1 à V4.

```
TEMP = VARIANCE(V1:V3, COL)
```

Crée trois colonnes nommées TEMP, VX et VY. La valeur unique de la colonne TEMP correspond à la variance du contenu de la colonne V1, la valeur unique de la colonne VX correspond à la variance du contenu de la colonne V2 et la valeur unique de la colonne VY correspond à la variance du contenu de la colonne V3.

```
TEMP = VARIANCE_(V1[1:5]:V3, COL) OU TEMP = VARIANCE(V1[1:5]:V3[1:5], COL)
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY` contenant chacune une seule valeur. La valeur de la colonne `TEMP` correspond à la variance des cibles des lignes 1 à 5 de la colonne `V1`, la valeur de la colonne `VX` correspond à la variance des cibles des lignes 1 à 5 de la colonne `V2`, et la valeur de la colonne `VY` correspond à la variance des cibles des lignes 1 à 5 de la colonne `V3`.

```
TEMP = VARIANCE(V1:V3, ROW)
```

Crée une colonne nommée `TEMP` dans laquelle chaque entrée de cible correspond à la variance de la ligne correspondante dans les colonnes `V1`, `V2` et `V3`.

```
TEMP = VARIANCE(V1[1:5]:V3,ROW) OU TEMP = VARIANCE(V1[1:5]:V3[1:5], ROW)
```

Crée une colonne nommée `TEMP`, dans laquelle les cibles des lignes 1 à 5 contiennent la variance de la ligne correspondante dans les colonnes `V1` à `V3`. Les autres cibles de `TEMP` sont vides.

## Macro WEEKDAY

La macro `WEEKDAY` est disponible uniquement dans Unica Campaign .

### Syntaxe

```
WEEKDAY(data [, conversion_keyword])
```

### Paramètres

`data`

Dates de texte ASCII à convertir en valeurs numériques représentant les jours de la semaine (1-7). Il peut s'agir de texte ASCII entre guillemets, d'une colonne de texte, d'une plage de cibles contenant du texte ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

`conversion_keyword`

Ce mot clé facultatif indique comment interpréter les formats de texte des dates et des heures. Sélectionnez l'une des options suivantes :

1 - mm/dd/yy (valeur par défaut).

2 - dd-mmm-yy

3 - mm/dd/yy hh:mm

Si ce paramètre n'est pas spécifié, la valeur par défaut est 1.

## Description

WEEKDAY convertit les valeurs de texte dans la plage de données spécifiée en valeurs numériques représentant les jours de la semaine, à l'aide du format spécifié pour convertir les dates et les heures. Le nombre 0 correspond à dimanche, le 1 à lundi et ainsi de suite jusqu'à 6 qui représente le samedi. Si une chaîne de texte ne peut pas être analysée à l'aide du mot clé `conversion_keyword`, WEEKDAY renvoie une erreur.

## Exemples

```
TEMP = WEEKDAY("1/1/95")
```

Crée une colonne nommée TEMP contenant le nombre 0 (1er janvier 1995 est un dimanche).

```
TEMP = WEEKDAY(V1, 2)
```

Crée une colonne nommée TEMP contenant les nombres des jours de la semaine pour les chaînes de texte de la colonne V1. Toutes les chaînes de texte de la colonne V1 doivent être au format dd-mmm-yy (sinon des ??? sont renvoyés).

```
TEMP = WEEKDAY(V1:V3, 3)
```

Crée trois colonnes nommées TEMP, VX et VY. La colonne TEMP contient des nombres représentant les jours de la semaine des chaînes de texte de la colonne V1. La colonne VX contient des nombres représentant les jours de la semaine des chaînes de texte de la colonne V2. La colonne VY contient des nombres représentant les jours de la semaine des chaînes de texte de la colonne V3. Toutes les chaînes de texte de la colonne V1 - V3 doivent être au format mm/dd/yy hh:mm (sinon des ??? sont renvoyés).

```
TEMP = WEEKDAY(V1[10:20]:V2, 10)
```

Crée deux colonnes nommées TEMP et VX. La colonne TEMP contient des nombres représentant les jours de la semaine des chaînes de texte des lignes 10 à 20 de la colonne V1. La colonne VX contient des nombres représentant les jours de la semaine des chaînes

de texte des lignes 10 à 20 de la colonne v2. Toutes les chaînes de texte doivent se présenter sous la forme mm/dd/yy (sinon, des ??? sont renvoyés).

## Fonctions associées

Fonction	Description
NUMBER	Convertit les chaînes de texte ASCII des heures et des dates en valeurs numériques

## Macro WEEKDAYOF

La macro WEEKDAYOF est disponible uniquement dans Unica Campaign .

### Syntaxe

```
WEEKDAYOF(date_string [, input_format])
```

### Paramètres

`date_string`

Texte représentant une date valide.

`input_format`


Un des mots clés du tableau ci-après spécifiant le format de date de `date_string`.

### Description

WEEKDAYOF renvoie le jour de la semaine sous la forme d'un nombre compris entre 0 et 6 (dimanche = 0, lundi = 1, et ainsi de suite) pour la date indiquée par `date_string`. Si `input_format` n'est pas fourni, le mot clé par défaut DELIM\_M\_D\_Y est utilisé.

### Exemples

WEEKDAYOF("08312000", MMDDYYYY) renvoie le nombre 4 car jeudi est le quatrième jour de la semaine.

 **Remarque** : Pour des informations supplémentaires sur les formats de date valides, voir [DATE \(à la page 69\)](#).

## Fonctions associées

Fonction	Description
DAYOF	Renvoie le jour du mois sous la forme d'un nombre.
MONTHOF	Renvoie le mois de l'année sous la forme d'un nombre.
YEAROF	Renvoie l'année sous la forme d'un nombre.

## Macro XOR

La macro XOR est disponible dans Unica Campaign Unica Interact et .

### Syntaxe

```
data1 XOR data2
```

### Paramètres

`data1`

Entiers non négatifs dont le XOR bit à bit doit être calculé avec les valeurs de `data2`. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

`data2`


Entier(s) non négatif(s) dont le XOR bit à bit doit être calculé avec les valeurs de `data1`. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans `data2` doit correspondre au nombre de colonnes dans `data1`, à moins que `data2` ne soit une constante. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit .

### Description

`XOR` calcule le XOR bit à bit entre les deux plages de données spécifiées. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne

correspondante de `data1` dont le XOR bit à bit avec la colonne correspondante de `data2` a été calculé (à savoir, le XOR bit à bit de la première colonne de `data1` avec la première colonne de `data`, celui de la deuxième colonne avec la deuxième colonne et ainsi de suite).

Si `data2` est une constante, le XOR bit à bit de chaque valeur de `data1` avec cette valeur est calculé. Si `data2` contient une ou plusieurs colonnes, les calculs sont effectués ligne par ligne, entre une colonne de `data2` et une colonne de `data2`. Le XOR bit à bit de la première ligne de `data1` avec la première ligne de `data2` est calculé, la seconde ligne avec la seconde ligne, etc. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

 **Remarque** : La précision de cette fonction macro est limitée aux valeurs entières inférieures à  $2^{24}$ . Les valeurs négatives sont interdites.

## Exemples

```
TEMP = 3 XOR 7
```

Crée une colonne nommée `TEMP` contenant la valeur quatre (le XOR bit à bit de `011` et `111` est égal à `100`).

```
TEMP = V1 XOR 8
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur correspond au XOR bit à bit du contenu de la colonne `V1` avec la valeur binaire `1000`.

```
TEMP = V1 XOR V1
```

Crée une colonne nommée `TEMP` ne contenant que des zéros (un XOR entre toute valeur et elle-même génère zéro).

```
TEMP = V1 XOR V2
```

Crée une colonne nommée `TEMP`, dans laquelle chaque valeur est la valeur de ligne de la colonne `V1` dont le XOR bit à bit est calculé avec la valeur de ligne correspondante de la colonne `V2`.

```
TEMP = V1:V3 XOR V4:V6
```

Crée trois colonnes nommées `TEMP`, `VX` et `VY`. La colonne `TEMP` contient les valeurs de `V1` dont le XOR bit à bit avec les valeurs de ligne correspondantes de la colonne `V4` a été



calculé. La colonne `VX` contient les valeurs résultant du XOR bit à bit des colonnes `V2` et `V5`. La colonne `VY` contient les valeurs résultant du XOR bit à bit des colonnes `V3` et `V6`.

```
TEMP = V1[10:20] XOR V2 OU TEMP = V1[10:20] XOR V2[1:11]
```

Crée une colonne nommée `TEMP`, dans laquelle les 11 premières cibles contiennent le résultat du XOR bit à bit des valeurs des lignes 10 à 20 de la colonne `V1` avec les valeurs des lignes 1 à 11 de la colonne `V2`. Les autres cibles de `TEMP` sont vides.

## Fonctions associées

Fonction	Description
<code>BIT_AND</code>	Calcule le AND bit à bit entre deux plages de données spécifiées
<code>BIT_NOT</code>	Calcule le NOT bit à bit du contenu de la plage de données spécifiée
<code>BIT_OR</code>	Calcule le OR bit à bit entre deux plages de données spécifiées

## Macro YEAROF

La macro `YEAROF` est disponible uniquement dans Unica Campaign .

### Syntaxe

```
YEAROF(date_string [, input_format])
```

### Paramètres

`date_string`

Texte représentant une date valide.

`input_format`

Un des mots clés du tableau ci-après spécifiant le format de date de `date_string`.

### Description

`YEAROF` renvoie l'année sous la forme d'un nombre pour la date indiquée par `date_string`. Si `input_format` n'est pas fourni, le mot clé par défaut `DELIM_M_D_Y` est utilisé.

## Exemples

`YEAROF("31082000",DDMMYYYY)` renvoie le nombre 2000.

Pour des informations supplémentaires sur les formats de date valides, voir [DATE \(à la page 69\)](#).

## Fonctions associées

<b>Fonction</b>	<b>Description</b>
<code>DAYOF</code>	Renvoie le jour du mois sous la forme d'un nombre.
<code>MONTHOF</code>	Renvoie le mois de l'année sous la forme d'un nombre.
<code>WEEKDAYOF</code>	Renvoie le jour de la semaine sous la forme d'un nombre.

# Index

## A

ABS  
24  
ACOS  
25  
ACOT  
27  
ADD\_MONTHS  
30  
AND  
32  
ASIN  
34  
ATAN  
37  
AVG  
39

## B

BETWEEN  
42  
BIT\_AND  
42  
BIT\_NOT  
45  
BIT\_OR  
46  
BIT\_XOR  
48

## C

CEILING  
51

## COLUMN

52  
configuration des dates sur votre  
application Web  
66  
Constantes  
utilisation dans les fonctions macro  
12,20

## COS

54

## COSH

56

## COT

58

## COUNT

60

## CURRENT\_DATE

62

## CURRENT\_DAY

63

## CURRENT\_JULIAN

9,64

## CURRENT\_MONTH

65

## CURRENT\_TIME

66

## CURRENT\_WEEKDAY

68

## CURRENT\_YEAR

68

## D

DATE	82
69	données
DATE_FORMAT	11,19
9,72	<b>E</b>
DATE_JULIAN	EQ
73	84
DATE_STRING	EST
74	115
DAY_BETWEEN	EXP
76	86
DAY_FROMNOW	EXTERNALCALLOUT
77	88
DAY_INTERVAL	<b>F</b>
78	FACTORIAL
DAYOF	90
79	FLOOR
définition des dates d'une langue	92
spécifique sur votre application Web	Fonctions d'horodatage
66	9,17
définitions de fonction	fonctions de chaîne
pages de cibles et	7
10,18	fonctions de groupement
pages de colonnes et	10
10,18	Fonctions des chaînes de caractères
Définitions de fonction	16
pages de cibles et	fonctions diverses
10	10,18
pages de colonnes et	fonctions macro
10	fonctions mathématiques et
DISTANCE	trigonométriques ; fonctions macro
80	trigonométriques ; fonctions macro
DIV	mathématiques

2	GROUPBY
fonctions statistiques ; fonctions	102,105
macro statistiques	GROUPBY_WHERE
2	105
plages de cibles dans	GT
10, 10,18	106
plages de colonnes dans	<b>H</b>
10, 10,18	having_value
Fonctions macro	105
fonctions macro de chaîne	<b>I</b>
7	IF
fonctions statistiques ; fonctions	108
macro statistiques	IN
1	110
fonctions mathématiques et	INT
trigonométriques	112
15	INVERSE
Fonctions statistiques	113
14	ISERROR
FORMAT	116
93	ISEVEN
Formats de date	118
69	ISODD
FRACTION	117,120
97	<b>K</b>
<b>G</b>	keyword
GE	12,19
99	Keyword
GET	102
101	<b>L</b>
Group_field	LE
102,105	122

LIKE	124	NE	150
LN ou LOG	126	NOT	152
LN;LOG	124	NUMBER	154
LOG10	130	<b>O</b>	
LOG2	128	OR	164
LOWER	131	Output_field	102,105
LT	132	<b>P</b>	
LTRIM	134	Paramètres des fonctions macro	11,18
<b>M</b>		PLUS	168
MAX	135	POSITION	166
MEAN	137	POW	170
MIN	140	<b>R</b>	
MINUS	142	RANDOM	172
MOD	144	RANDOM_GAUSS	174
MONTHOF	147, 215, 218	récapitulatif des fonctions macro	1,14
MULT	148	référence des macros	21
<b>N</b>		Rolled_field	102,105
		ROUND	

176	STRING_SEG
ROWNUM	195
177	STRING_TAIL
RTRIM	197
178	SUBSTR ou SUBSTRING
<b>S</b>	199
SIGN	SUBSTR;SUBSTRING
178	199
SIN	SUM
180	200
SINH	<b>T</b>
181	TAN
SIZE	202
183	TANH
SORT	204
184	TOTAL
spécifications de format	206
11,18	TRUNCATE
SQRT	208
185	<b>U</b>
STDV ou STDEV	UPPER
186	210
STDV;STDEV	Utilisation de macros dans
186	Unica Campaign
STRING_CONCAT	1
189	Utilisation de macros dans
STRING_HEAD	Unica Interact
191	14
STRING_LENGTH	utilisation des constantes
192	12,20
STRING_PROPER	<b>V</b>
194	Valeurs booléennes

108

VARIANCE

211

**W**

WEEKDAY

213

WEEKDAYOF

215

**X**

XOR

216

**Y**

YEAROF

218