Benutzerhandbuch zu Makros

Inhalt

Kapitel 1. Verwendung von Makros in Unica Campaign

Dieses Kapitel enthält Informationen zur Verwendung von Makros in Unica Campaign. Lesen Sie dieses Kapitel unbedingt, bevor Sie versuchen, Makros in Unica Campaign zu verwenden.

Zusammenfassung der Makrofunktionen für Unica Campaign

Die Tabellen in diesem Abschnitt fassen die Makrofunktionen nach Kategorien zusammen.

Ausführliche Referenzseiten zu den einzelnen Makrofunktionen in alphabetischer Reihenfolge werden in Makroreferenz (auf Seite 22) bereitgestellt.

⚠ **Wichtig:** Makros können für Unica Campaign und Unica Interact oder nur für eines der beiden Produkte Anwendung finden. In den Makrobeschreibungen werden die Produkte angegeben, in denen das jeweilige Makro verfügbar ist.

Weitere Informationen zu den Eingabeparametern der Makrofunktionen finden Sie unter Makrofunktionsparameter für Unica Campaign (auf Seite 12).

Statistische Funktionen

Makroname	Rückgabe	Syntax
AVG	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet das arithmetische Mittel oder den Durchschnitt eines Zellenbereichs
COUNT	Einzelwert in einer neuen Spalte	Zählt die Anzahl der Werte in einem

Makroname	Rückgabe	Syntax
		angegebenen Datenbereich
MAX	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet das Maximum eines Zellenbereichs
MEAN	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet das arithmetische Mittel oder den Durchschnitt eines Zellenbereichs
MIN	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet das Minimum eines Zellenbereichs
STDV oder STDEV	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet die Standardabweichung eines Zellenbereichs
VARIANCE	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet die Varianz eines Zellenbereichs

Mathematische und trigonometrische Funktionen

Makroname	Rückgabe	Syntax
ABS	Eine Spalte für jede Eingabespalte	Berechnet den absoluten Wert des Inhalts des angegebenen Datenbereichs.

Makroname	Rückgabe	Syntax
ACOS	Eine Spalte für jede Eingabespalte	Berechnet den Arkuskosinus des Inhalts des angegebenen Datenbereichs.
ACOT	Eine Spalte für jede Eingabespalte	Berechnet den Arkuskotangens des Inhalts des angegebenen Datenbereichs.
ASIN	Eine Spalte für jede Eingabespalte	Berechnet den Arkussinus des Inhalts des angegebenen Datenbereichs.
ATAN	Eine Spalte für jede Eingabespalte	Berechnet den Arkustangens des Inhalts des angegebenen Datenbereichs.
AVG	Eine Spalte für jede Eingabespalte	Berechnet das arithmetische Mittel oder den Durchschnitt der Zellen im angegebenen Datenbereich.
BETWEEN	Eine Spalte für jede Eingabespalte	Vergleicht zwei Werte, um festzustellen, ob der angegebene Wert zwischen zwei anderen Werten liegt.
CEILING	Eine Spalte für jede Eingabespalte	Berechnet die Obergrenze jedes Werts im angegebenen Datenbereich.
COLUMN	Eine Spalte für jede Eingabespalte	Erstellt neue Spalten, wobei die Eingabewerte in jeder

Makroname	Rückgabe	Syntax
		Spalte vertikal verkettet werden.
COS	Eine Spalte für jede Eingabespalte	Berechnet den Kosinus des Inhalts des angegebenen Datenbereichs.
COSH	Eine Spalte für jede Eingabespalte	Berechnet den Hyperbelkosinus des Inhalts des angegebenen Datenbereichs.
COT	Eine Spalte für jede Eingabespalte	Berechnet den Kotangens des Inhalts des angegebenen Datenbereichs.
COUNT	Eine Spalte mit einem Einzelwert	Zählt die Anzahl der Zellen, die Werte enthalten, im angegebenen Datenbereich.
EXP	Eine Spalte für jede Eingabespalte	Erhebt die natürliche Zahl (e) in die Potenz, die durch den Inhalt jeder Zelle im angegebenen Datenbereich angegeben ist.
FACTORIAL	Eine Spalte für jede Eingabespalte	Berechnet die Fakultät jedes Werts im angegebenen Datenbereich.
FLOOR	Eine Spalte für jede Eingabespalte	Berechnet die Untergrenze jedes Werts im angegebenen Datenbereich.

Makroname	Rückgabe	Syntax
FRACTION	Eine Spalte für jede Eingabespalte	Berechnet die Nachkommastellen jedes Werts im angegebenen Datenbereich.
INT	Eine Spalte für jede Eingabespalte	Berechnet den abgerundeten Ganzzahlwert des Inhalts des angegebenen Datenbereichs.
INVERSE	Eine Spalte für jede Eingabespalte	Berechnet den Negativwert des Inhalts des angegebenen Datenbereichs.
LN	Eine Spalte für jede Eingabespalte	Berechnet den natürlichen Logarithmus des Inhalts des angegebenen Datenbereichs.
LOG	Eine Spalte für jede Eingabespalte	Berechnet den natürlichen Logarithmus des Inhalts des angegebenen Datenbereichs.
LOG2	Eine Spalte für jede Eingabespalte	Berechnet die Protokollbasis2 des Inhalts des angegebenen Datenbereichs
LOG10	Eine Spalte für jede Eingabespalte	Berechnet die Protokollbasis10 des Inhalts des angegebenen Datenbereichs
MAX	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort	Berechnet das Maximum eines Zellenbereichs

Makroname	Rückgabe	Syntax
	COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	
MEAN	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet das arithmetische Mittel oder den Durchschnitt eines Zellenbereichs
MIN	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet das Minimum eines Zellenbereichs
RANDOM	Eine Spalte mit der angegebenen Anzahl von Werten	Gibt die angegebene Anzahl von Zufallszahlen zurück.
RANDOM_GAUSS	Eine Spalte mit der angegebenen Anzahl von Werten	Gibt die angegebene Anzahl von Zufallswerten aus einer gaußschen Verteilung zurück.
ROUND	Eine Spalte für jede Eingabespalte	Berechnet den gerundeten Wert des Inhalts des angegebenen Datenbereichs.
SIGN	Eine Spalte für jede Eingabespalte	Berechnet das Vorzeichen (positiv oder negativ) der Werte im angegebenen Datenbereich.
SIN	Eine Spalte für jede Eingabespalte	Berechnet den Sinus des Inhalts des angegebenen Datenbereichs.

Makroname	Rückgabe	Syntax
SINH	Eine Spalte für jede Eingabespalte	Berechnet den Hyperbelsinus des Inhalts des angegebenen Datenbereichs.
SQRT	Eine Spalte für jede Eingabespalte	Berechnet die Quadratwurzel des Inhalts des angegebenen Datenbereichs.
STDV oder STDEV	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet die Standardabweichung eines Zellenbereichs
SUM	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet die Summe eines Zellenbereichs.
TAN	Eine Spalte für jede Eingabespalte	Berechnet den Tangens des Inhalts des angegebenen Datenbereichs.
TANH	Eine Spalte für jede Eingabespalte	Berechnet den Hyperbeltangens des Inhalts des angegebenen Datenbereichs.
TOTAL	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet die Summe eines Zellenbereichs.

Makroname	Rückgabe	Syntax
TRUNCATE	Eine Spalte für jede Eingabespalte	Berechnet den Ganzzahlanteil jedes Werts im angegebenen Datenbereich.
VARIANCE	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet die Varianz eines Zellenbereichs

Zeichenfolgefunktionen

Makroname	Rückgabe	Syntax
FORMAT	Eine Spalte für jede Eingabespalte	Steuert die Ausgabeformatierung für Zahlen und Zeichenfolgen (z. B. Ausgabebreite, Ausrichtung, numerische Genauigkeit, Dezimalzeichen, Trennzeichen usw.). Gibt die formatierte Ausgabezeichenfolge zurück.
LIKE	Eine Spalte für jede Eingabespalte	Ermittelt, ob eine Zeichenfolge mit einem angegebenen Muster übereinstimmt
LOWER	Eine Spalte für jede Eingabespalte	Konvertiert Zeichenfolgewerte in Kleinschreibung
LTRIM	Eine Spalte für jede Eingabespalte	Entfernt führende Leerzeichen aus jedem Zeichenfolgewert
NUMBER	Eine Spalte für jede Eingabespalte	Konvertiert ASCII-Zeichenfolgen für Uhrzeit- und Datumsangaben in numerische Werte
POSITION	Eine Spalte für jede Eingabespalte	Gibt die Anfangsposition eines Musters in einer Zeichenfolge zurück

Makroname	Rückgabe	Syntax
RTRIM	Eine Spalte für jede Eingabespalte	Entfernt nachfolgende Leerzeichen aus jedem Zeichenfolgewert
STRING_CONCAT	Eine Spalte mit einem Wert für jede Zeile der kürzesten Eingabespalte	Verkettet Zeichenfolgen aus den angegebenen Datenbereichen
STRING_HEAD	Eine Spalte für jede Eingabespalte	Gibt die ersten *n* Zeichen jeder Zeichenfolge im angegebenen Datenbereich zurück.
STRING_LENGTH	Eine Spalte für jede Eingabespalte	Gibt die Länge jeder Zeichenfolge im angegebenen Datenbereich zurück
STRING_PROPER	Eine Spalte für jede Eingabespalte	Konvertiert jeden Zeichenfolgewert so, dass der erste Buchstabe oder irgendein Buchstabe, der auf ein Leerzeichen oder Symbol (außer dem Unterstrich) folgt, in einen Großbuchstaben und alle anderen Zeichen in Kleinbuchstaben geändert werden
STRING_SEG	Eine Spalte für jede Eingabespalte	Gibt das Zeichenfolgesegment zwischen zwei angegebenen Indizes zurück
STRING_TAIL	Eine Spalte für jede Eingabespalte	Gibt die letzten *n* Zeichen jeder Zeichenfolge im angegebenen Datenbereich zurück.
SUBSTR or SUBSTRING	Eine Spalte für jede Eingabespalte	Gibt Zeichen aus einer Zeichenfolge ab einer Anfangsposition zurück
UPPER	Eine Spalte für jede Eingabespalte	Konvertiert Zeichenfolgewerte in Großschreibung

Datums- und Zeitfunktionen

Makroname	Rückgabe	Syntax
CURRENT_DATE	Eine Spalte für jede Eingabespalte	Gibt das aktuelle Datum in zurück. format
CURRENT_DAY	Eine Spalte für jede Eingabespalte	Gibt den aktuellen Tag des Monats als Zahl zwischen 1 und 31 zurück
CURRENT_JULIAN	Eine Spalte für jede Eingabespalte	Gibt die julianische Zahl für das aktuelle Datum zurück
CURRENT_MONTH	Eine Spalte für jede Eingabespalte	Gibt den aktuellen Monat des Jahres als Zahl zwischen 1 und 12 zurück
CURRENT_TIME	Eine Spalte für jede Eingabespalte	Gibt die aktuelle Uhrzeit als Zeichenfolge zurück
CURRENT_WEEKDAY	Eine Spalte für jede Eingabespalte	Gibt den aktuellen Wochentag als Zahl zwischen 0 und 6 zurück
CURRENT_YEAR	Eine Spalte für jede Eingabespalte	Gibt das aktuelle Jahr als Zahl zurück
DATE	Eine Spalte für jede Eingabespalte	Konvertiert eine Datumszeichenfolge in ein julianisches Datum
DATE_FORMAT	Eine Spalte für jede Eingabespalte	Wandelt Datumsformate um
DATE_JULIAN	Eine Spalte für jede Eingabespalte	Gibt das julianische Datum zurück
DATE_STRING	Eine Spalte für jede Eingabespalte	Gibt die Datumszeichenfolge des julianischen Datums zurück
DAY_BETWEEN	Eine Spalte für jede Eingabespalte	Gibt die Anzahl der Tage zwischen zwei Terminen zurück
DAY_FROMNOW	Eine Spalte für jede Eingabespalte	Gibt die Anzahl der Tage vom aktuellen Datum bis zum angegebenen Datum zurück
DAY_INTERVAL	Eine Spalte für jede Eingabespalte	Gibt die Anzahl der Tage zwischen zwei Terminen zurück
DAYOF	Eine Spalte für jede Eingabespalte	Gibt den Tag des Monats als Zahl zurück

Makroname	Rückgabe	Syntax
MONTHOF	Eine Spalte für jede Eingabespalte	Gibt den Monat des Jahres als Zahl zurück
WEEKDAY	Eine Spalte für jede Eingabespalte	Konvertiert ASCII-Datumszeichenfolgen in Wochentage
WEEKDAYOF	Eine Spalte für jede Eingabespalte	Gibt den Wochentag der Woche als Zahl zurück
YEAROF	Eine Spalte für jede Eingabespalte	Gibt das Jahr als Zahl zurück

Gruppierungsfunktionen

Makroname	Rückgabe	Syntax
GROUPBY	Eine neue Spalte mit einem Wert für jede Zeile	Fasst mehrere Zeilen von Daten in einer Gruppe zusammen
GROUPBY_WHERE	Eine neue Spalte mit einem Wert für jede Zeile	Fasst mehrere Zeilen von Daten zusammen, die eine bestimmte Bedingung erfüllen und sich in einer Gruppe befinden

Verschiedene Funktionen

Makroname	Rückgabe	Syntax
IF	Eine Spalte mit einem Wert für jede Zeile der kürzesten Eingabespalte	Leitet eine If-then-else-Anweisung einer bedingten Anweisung ein
ISERROR	Eine Spalte mit einem Wert für jede Zeile der kürzesten Eingabespalte	Gibt eine Eins zurück, wenn ein Wert in der Eingabezeile eine fehlerhafte Zelle (???) enthält, andernfalls eine Null
ISEVEN	Eine Spalte für jede Eingabespalte	Prüft, ob die Eingabewerte gerade (d. h. durch zwei teilbar) sind
ISODD	Eine Spalte für jede Eingabespalte	Prüft, ob die Eingabewerte ungerade (d. h. nicht durch zwei teilbar) sind

Makroname	Rückgabe	Syntax
`ROWNUM`	Eine Spalte für jede Eingabespalte	Generiert fortlaufende Zahlen von eins bis zur Anzahl der Einträge

Makrofunktionsparameter für Unica Campaign

Dieser Abschnitt beschreibt die Parameter und ihre Verwendung für Makrofunktionen in Unica Campaign.

Formatspezifikationen

Dieser Abschnitt beschreibt das Format für einige häufig verwendete Parameter. Es gilt für alle Referenzen auf diese Parameter durch Makrofunktionsspezifikationen in diesem Kapitel.

Daten

Der Parameter `data` stellt eine Datenspalte dar, auf die eine Makrofunktion angewendet werden soll.

Es kann sich um eine Konstante oder um ein Feld handeln. Nähere Einzelheiten finden Sie im Abschnitt zur jeweiligen Makrofunktion.

Anmerkung: Unica Campaign unterstützt keine Berechnungen in mehreren Feldern gleichzeitig oder in einer Untergruppe von Zeilen.

Einige weitere Parameternamen verwenden dasselbe Format wie `data`. Die Beschreibungen dieser Parameter verweisen auf dieses Kapitel und Format.

Schlüsselwort

Der Parameter `keyword` steuert das Verhalten der Makrofunktion. Er zeigt an, dass ein Schlüsselwort angegeben werden kann (wenn es weggelassen wird, wird der Standardwert verwendet). Die Schlüsselwortoptionen werden für die jeweilige Makrofunktion in der folgenden Form aufgeführt:

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

Wählen Sie die Schlüsselwortoption aus, die zu dem gewünschten Verhalten führt. Die Standardoption wird in Fettschrift angezeigt. Es können zum Beispiel die folgenden Optionen angegeben sein:

```
{RADIANS | DEGREES}
```

In diesem Fall sind die beiden folgenden Makrofunktionen gültig:

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

Anmerkung: Viele Makrofunktionen nehmen die Schlüsselwortparameter {ALL | COL | ROW} an. Diese Schlüsselwörter gelten nicht für Unica Campaign, da es sich bei den Eingabedaten immer um eine Einzelspalte oder ein Einzelfeld handelt. Das Makro verhält sich immer so, als ob das Schlüsselwort COL angegeben würde. Deshalb brauchen Sie diese Schlüsselwörter nicht anzugeben, wenn Sie Unica Campaign verwenden.

Verwendung von Konstanten

Die meisten Makrofunktionsparameter verarbeiten numerische Konstanten oder Ausdrücke, deren Auswertung eine numerische Konstante ergibt (Makrofunktionen für Zeichenfolgen können Zeichenfolgekonstanten verarbeiten).

In Makrofunktionen, die alle Einträge einzeln durchgehen (etwa bei der Addition zweier numerischer Spalten), ist die Verwendung einer Konstante äquivalent zur Angabe einer Spalte, die in jeder Zeile den betreffenden konstanten Wert enthält. Im Wesentlichen wird eine Konstante, die als ein Eingabeparameter angegeben ist, bis zur Länge der Eingabe expandiert.

Einige Makrofunktionen können ASCII-Zeichenfolgen und numerische Konstanten verarbeiten. Bei Parametern, die sowohl numerische Konstanten als auch ASCII-Zeichenfolgen verarbeiten können, ist dies im Abschnitt "Parameter" der jeweiligen Makrofunktion vermerkt.

Die folgende Tabelle enthält einige Beispiele.

Funktionsdefinition	Interpretation der Konstante
`PERCENT_UTILIZ = (CURR_BAL*100)/ CREDIT_LIM`	Die Konstante `100` wird als Spalte interpretiert, die dieselbe Zeilenzahl wie die Spalte `CURR_BAL` enthält, wobei jede Zeile die Konstante `100` enthält. Das abgeleitete Feld `PERCENT_UTILIZ` enthält jeden Wert von `CURR_BAL`, multipliziert mit `100` und dividiert durch jeden Wert von `CREDIT_LIM`.
`NAME = STRING_CONCAT ("Mr. ", LAST_NAME)`	Die Konstante `"Mr."` wird als Spalte interpretiert, die dieselbe Zeilenzahl wie die Spalte `LAST_NAME` enthält, wobei jede Zeile die Konstante `"Mr."` enthält. Im abgeleiteten Feld `NAME` ist jeder der Zeichenfolgen in `LAST_NAME` die Zeichenfolge `"Mr."` vorangestellt.

Kapitel 2. Verwendung von Makros in Unica Interact

Dieses Kapitel enthält Informationen zur Verwendung von Makros in Unica Interact. Lesen Sie dieses Kapitel unbedingt, bevor Sie versuchen, Makros in Unica Interact zu verwenden.

Formelhilfe und Syntaxprüfung von Makroausdrücken

⚠ **Wichtig:** Das Dialogfeld der **Formelhilfe** sowie die darin enthaltene Funktion zur Syntaxprüfung prüfen Makroausdrücke derzeit gemäß den von Unica Campaign unterstützten Makros. Unica Interact unterstützt jedoch nur eine Teilmenge der Makrofunktionalität von Unica Campaign. Daher müssen Sie sicherstellen, dass die Makros und Schlüsselwörter (z. B. Schlüsselwörter für das Datumsformat) unterstützt werden, die in Unica Interact verwendet werden. Suchen Sie im Kapitel zur Makroreferenz in diesem Handbuch nach Hinweisen in Bezug auf Unica Interact.

Zusammenfassung der Makrofunktionen für Unica Interact

Die Tabellen in den folgenden Abschnitten enthalten ausführliche Beschreibungen der Makros, die für Unica Interact spezifisch sind.

Ausführliche Referenzseiten zu den einzelnen Makrofunktionen in alphabetischer Reihenfolge werden in Makroreferenz (auf Seite 22) bereitgestellt.

⚠ **Wichtig:** Makros können für Unica Campaign und Unica Interact oder nur für eines der beiden Produkte Anwendung finden. In den Makrobeschreibungen werden die Produkte angegeben, in denen das jeweilige Makro verfügbar ist.

Makrofunktionsparameter für Unica Interact (auf Seite 19) enthält Informationen zu den Eingabeparametern der Makrofunktionen für Unica Interact.

Statistische Funktionen

Makroname	Rückgabe	Syntax
AVG	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet das arithmetische Mittel oder den Durchschnitt eines Zellenbereichs
MAX	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet das Maximum eines Zellenbereichs
MEAN	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet das arithmetische Mittel oder den Durchschnitt eines Zellenbereichs
MIN	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet das Minimum eines Zellenbereichs
STDV oder STDEV	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet die Standardabweichung eines Zellenbereichs

Mathematische und trigonometrische Funktionen

Makroname	Rückgabe	Syntax
AVG	Eine Spalte für jede Eingabespalte	Berechnet das arithmetische Mittel oder den Durchschnitt der Zellen im angegebenen Datenbereich.

Makroname	Rückgabe	Syntax
MAX	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet das Maximum eines Zellenbereichs
MEAN	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet das arithmetische Mittel oder den Durchschnitt eines Zellenbereichs
MIN	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet das Minimum eines Zellenbereichs
STDV or STDEV	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet die Standardabweichung eines Zellenbereichs
SUM	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet die Summe eines Zellenbereichs.
TOTAL	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet die Summe eines Zellenbereichs.

Zeichenfolgefunktionen

Makroname	Rückgabe	Syntax
`LIKE`	Eine Spalte für jede Eingabespalte	Ermittelt, ob eine Zeichenfolge mit einem angegebenen Muster übereinstimmt
`LOWER`	Eine Spalte für jede Eingabespalte	Konvertiert Zeichenfolgewerte in Kleinschreibung
`LTRIM`	Eine Spalte für jede Eingabespalte	Entfernt führende Leerzeichen aus jedem Zeichenfolgewert
`NUMBER`	Eine Spalte für jede Eingabespalte	Konvertiert ASCII-Zeichenfolgen für Uhrzeit- und Datumsangaben in numerische Werte
`RTRIM`	Eine Spalte für jede Eingabespalte	Entfernt nachfolgende Leerzeichen aus jedem Zeichenfolgewert
`STRING_CONCAT`	Eine Spalte mit einem Wert für jede Zeile der kürzesten Eingabespalte	Verkettet Zeichenfolgen aus den angegebenen Datenbereichen
`SUBSTR or SUBSTRING`	Eine Spalte für jede Eingabespalte	Gibt Zeichen aus einer Zeichenfolge ab einer Anfangsposition zurück
`UPPER`	Eine Spalte für jede Eingabespalte	Konvertiert Zeichenfolgewerte in Großschreibung

Datums- und Zeitfunktionen

Makroname	Rückgabe	Syntax
`CURRENT_DATE`	Eine Spalte für jede Eingabespalte	Gibt das aktuelle Datum in zurück. `format`
`CURRENT_DAY`	Eine Spalte für jede Eingabespalte	Gibt den aktuellen Tag des Monats als Zahl zwischen 1 und 31 zurück
`CURRENT_MONTH`	Eine Spalte für jede Eingabespalte	Gibt den aktuellen Monat des Jahres als Zahl zwischen 1 und 12 zurück
`CURRENT_WEEKDAY`	Eine Spalte für jede Eingabespalte	Gibt den aktuellen Wochentag des Monats als Zahl zwischen 0 und 6 zurück

Makroname	Rückgabe	Syntax
CURRENT_YEAR	Eine Spalte für jede Eingabespalte	Gibt das aktuelle Jahr als Zahl zurück
DATE	Eine Spalte für jede Eingabespalte	Konvertiert eine Datumszeichenfolge in ein julianisches Datum
DATE_FORMAT	Eine Spalte für jede Eingabespalte	Wandelt Datumsformate um

Verschiedene Funktionen

Makroname	Rückgabe	Syntax
EXTERNALCALLOUT	Werte, die von der über die ExternalCallout-API erstellten kundenspezifischen Anwendung definiert worden sind	Ruft eine kundenspezifische Anwendung auf, die mithilfe der ExternalCallout-API geschrieben wurde. Weitere Informationen finden Sie im Unica Interact-Administratorhandbuch.
IF	Eine Spalte mit einem Wert für jede Zeile der kürzesten Eingabespalte	Leitet eine If-then-else-Anweisung einer bedingten Anweisung ein
INDEXOF	Liste der Werte für eine bestimmte Bedingung in einer Spalte	Entworfen als vordefiniertes Makro, das durch Übergabe als Parameter innerhalb des EXTERNALCALLOUT-Makros verwendet werden kann. Es wird eine Liste von Indizes zurückgegeben, die den angegebenen Ausdruck erfüllen.

Makrofunktionsparameter für Unica Interact

Dieser Abschnitt beschreibt die Parameter und ihre Verwendung für Makrofunktionen in Unica Interact.

Formatspezifikationen

Dieser Abschnitt beschreibt das Format für einige häufig verwendete Parameter. Es gilt für alle Referenzen auf diese Parameter durch Makrofunktionsspezifikationen in diesem Abschnitt.

Daten

Der Parameter `data` stellt eine Datenspalte dar, auf die eine Makrofunktion angewendet werden soll. Es kann sich um eine Konstante oder um ein Feld handeln. Nähere Einzelheiten finden Sie im Abschnitt zur jeweiligen Makrofunktion.

Anmerkung: Unica Interact unterstützt keine Berechnungen in mehreren Feldern gleichzeitig oder in einer Untergruppe von Zeilen.

Einige weitere Parameternamen verwenden dasselbe Format wie `data`. Die Beschreibungen dieser Parameter verweisen auf dieses Kapitel und Format.

Schlüsselwort

Der Parameter `keyword` steuert das Verhalten der Makrofunktion. Er zeigt an, dass ein Schlüsselwort angegeben werden kann (wenn es weggelassen wird, wird der Standardwert verwendet). Die Schlüsselwortoptionen werden für die jeweilige Makrofunktion in der folgenden Form aufgeführt:

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

Wählen Sie die Schlüsselwortoption aus, die zu dem gewünschten Verhalten führt. Die Standardoption wird in Fettschrift angezeigt. Es können zum Beispiel die folgenden Optionen angegeben sein:

```
{RADIANS | DEGREES}
```

In diesem Fall sind die beiden folgenden Makrofunktionen gültig:

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

Anmerkung: Viele Makrofunktionen nehmen die Schlüsselwortparameter `{ALL | COL | ROW}` an. Diese Schlüsselwörter gelten nicht für Unica Interact, da es sich bei den

Eingabedaten immer um eine Einzelspalte oder ein Einzelfeld handelt. Das Makro verhält sich immer so, als ob das Schlüsselwort `COL` angegeben würde. Deshalb brauchen Sie diese Schlüsselwörter nicht anzugeben, wenn Sie Unica Interact verwenden.

Verwendung von Konstanten

Die meisten Makrofunktionsparameter verarbeiten numerische Konstanten oder Ausdrücke, deren Auswertung eine numerische Konstante ergibt (Makrofunktionen für Zeichenfolgen können Zeichenfolgekonstanten verarbeiten).

In Makrofunktionen, die alle Einträge einzeln durchgehen (etwa bei der Addition zweier numerischer Spalten), ist die Verwendung einer Konstante äquivalent zur Angabe einer Spalte, die in jeder Zeile den betreffenden konstanten Wert enthält. Im Wesentlichen wird eine Konstante, die als ein Eingabeparameter angegeben ist, bis zur Länge der Eingabe expandiert.

Einige Makrofunktionen können ASCII-Zeichenfolgen und numerische Konstanten verarbeiten. Bei Parametern, die sowohl numerische Konstanten als auch ASCII-Zeichenfolgen verarbeiten können, ist dies im Abschnitt "Parameter" der jeweiligen Makrofunktion vermerkt.

Die folgende Tabelle enthält einige Beispiele.

Funktionsdefinition	**Interpretation der Konstante**
`PERCENT_UTILIZ = (CURR_BAL*100)/ CREDIT_LIM`	Die Konstante `100` wird als Spalte interpretiert, die dieselbe Zeilenzahl wie die Spalte `CURR_BAL` enthält, wobei jede Zeile die Konstante `100` enthält. Das abgeleitete Feld `PERCENT_UTILIZ` enthält jeden Wert von `CURR_BAL`, multipliziert mit `100` und dividiert durch jeden Wert von `CREDIT_LIM`.
`NAME = STRING_CONCAT ("Mr.", LAST_NAME)`	Die Konstante `"Mr."` wird als Spalte interpretiert, die dieselbe Zeilenzahl wie die Spalte `LAST_NAME` enthält, wobei jede Zeile die Konstante `"Mr."` enthält. Im abgeleiteten Feld `NAME` ist jeder der Zeichenfolgen in `LAST_NAME` die Zeichenfolge `"Mr."` vorangestellt.

Anmerkung: Konstanten wie DT_DELIM_M_D_Y erfordern einfache Anführungszeichen.

Kapitel 3. Makroreferenz

In diesem Abschnitt werden die einzelnen Makros beschrieben, die zur Verwendung in Campaign und/oder Interact verfügbar sind. Makros werden in alphabetischer Reihenfolge aufgelistet.

⚠ **Wichtig:** Verwenden Sie keine Funktionsnamen oder Schlüsselwörter aus der Makrosprache für Spaltenüberschriften von Benutzertabellen in Unica Campaign, unabhängig davon, ob deren Zuordnung auf einer Datenbank oder einer unstrukturierten Datei beruht. Diese reservierten Wörter können Fehler verursachen, wenn sie in Spaltenüberschriften zugeordneter Tabellen verwendet werden.

Gültige Schlüsselwörter für das Datumsformat

In der folgenden Tabelle sehen Sie die Schlüsselwörter für gültige Formate, einschließlich einer Beschreibung und eines Beispiels.

Suchbegriff	Syntax	Beispiel(e)
`MM`	2-stelliger Monat	01, 02, 03, ..., 12
`MMDD`	2-stelliger Monat und 2-stelliger Tag	Der 31. März ist 0331
`MMDDYY`	2-stelliger Monat, 2-stelliger Tag und 2-stelliges Jahr	Der 31. März 1970 ist 033170
`MMDDYYYY`	2-stelliger Monat, 2-stelliger Tag und 4-stelliges Jahr	Der 31. März 1970 ist 03311970
`DELIM_M_D`	Beliebiger Monat mit Begrenzer, gefolgt vom Tag	Der 31. März ist 3/31 oder 03-31
`DELIM_M_D_Y`	Belieb. Monat, Tag und Jahr, getrennt durch Begrenzer	März 31 1970 oder 3/31/70
`DELIM_M_D_YYYY`	Belieb. Monat, Tag und 4-stelliges Jahr, getrennt durch Begrenzer	März 31 1970 oder 3/31/1970

Suchbegriff	Syntax	Beispiel(e)
DELIM_Y_M	Beliebiges Jahr mit Begrenzer, gefolgt vom Monat	70 März, 70-3 oder 1970/3
DELIM_Y_M_D	Beliebiger Jahr, Monat und Tag, getrennt durch Begrenzer	1970 Mär 31 oder 70/3/31
YYMMM	2-stelliges Jahr und 3-stelliger Monat	70MÄR
YYMMMDD	2-stelliges Jahr, 3-stelliger Monat und 2-stelliger Tag	70MÄR31
YY	2-stelliges Jahr	70
YYMM	2-stelliges Jahr und 2-stelliger Monat	7003
YYMMDD	2-stelliges Jahr, 2-stelliger Monat und 2-stelliger Tag	700331
YYYYMMM	4-stelliges Jahr und 3-stelliger Monat	1970MÄR
YYYYMMMDD	4-stelliges Jahr, 3-stelliger Monat und 2-stelliger Tag	1970MÄR31
YYYY	4-stelliges Jahr	1970
YYYYMM	4-stelliges Jahr und 2-stelliger Monat	197003
YYYYMMDD	4-stelliges Jahr, 2-stelliger Monat und 2-stelliger Tag	19700331
DELIM_M_Y	Beliebiger Monat mit Begrenzer, gefolgt vom Jahr	3-70, 3/70, Mär 70, März 1970
DELIM_D_M	Beliebiger Tag mit Begrenzer, gefolgt vom Monat	31-3, 31/3, 31 März
DELIM_D_M_Y	Belieb. Tag, Monat und Jahr, getrennt durch Begrenzer	31-MÄR-70, 31/3/1970, 31 03 70
DD	2-stelliger Tag	31
DDMMM	2-stelliger Tag und 3-stelliger Monat	31MÄR

Suchbegriff	Syntax	Beispiel(e)
DDMMMYY	2-stelliger Tag, 3-stelliger Monat und 2-stelliges Jahr	31MÄR1970
DDMMMYYYY	2-stelliger Tag, 3-stelliger Monat und 4-stelliges Jahr	31MÄR1970
DDMM	2-stelliger Tag und 2-stelliger Monat	3103
DDMMYY	2-stelliger Tag, 2-stelliger Monat und 2-stelliges Jahr	310370
DDMMYYYY	2-stelliger Tag, 2-stelliger Monat und 4-stelliges Jahr	31031970
MMYY	2-stelliger Monat und 2-stelliges Jahr	0370
MMYYYY	2-stelliger Monat und 4-stelliges Jahr	031970
MMM	3-stelliger Monat	MÄR
MMMDD	3-stelliger Monat und 2-stelliger Tag	MÄR31
MMMDDYY	3-stelliger Monat, 2-stelliger Tag und 2-stelliges Jahr	MÄR3170
MMMDDYYYY	3-stelliger Monat, 2-stelliger Tag und 4-stelliges Jahr	MÄR311970
MMMYY	3-stelliger Monat und 2-stelliges Jahr	MÄR1970
MMMYYYY	3-stelliger Monat und 4-stelliges Jahr	MÄR1970
MONTH	Monat des Jahres	Januar, Februar, März usw. oder Jan, Feb, März usw.
WEEKDAY	Wochentag	Sonntag, Montag, Dienstag usw. (Sonntag = 0)
WKD	Abgekürzter Wochentag	Son, Mon, Die usw. (Son = 0)

ABS-Makro

Das ABS Makro ist nur in Unica Campaign verfügbar.

Syntax

```
ABS(data)
```

Parameter

`data`

Die numerischen Werte, deren absoluter Wert berechnet werden soll. Dieser Parameter kann ein konstanter Wert, eine Spalte, ein Zellenbereich oder ein Ausdruck sein, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

ABS berechnet den absoluten Wert der Zahlen im angegebenen Datenbereich. Der absolute Wert einer Zahl ist ihr Wert ohne ihr Vorzeichen (d.h. positive Zahlen bleiben unverändert; negative Zahlen werden als positive Zahlen zurückgegeben). ABS gibt für jede Eingabespalte eine neue Spalte zurück, die jeweils den absoluten Wert der Zahlen in der entsprechenden Eingabespalte enthält.

Beispiele

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

Erstellt eine Spalte TEMP, die den Wert 3 enthält.

```
TEMP = ABS(V1)
```

Erstellt eine Spalte TEMP, in der jeder Wert den absoluten Wert des Inhalts von Spalte V1 darstellt.

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

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Die Werte in der Spalte TEMP sind die absoluten Werte des Inhalts von Spalte V1, die Werte von Spalte VX sind die absoluten Werte des Inhalts von Spalte V2 und die Werte von Spalte VY sind die absoluten Werte des Inhalts von Spalte V3.

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

Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen jeweils die Vorzeichen der Werte in Zeile 10-20 von Spalte V1 enthalten. Die anderen Zellen in TEMP sind leer.

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

Erstellt zwei neue Spalten TEMP und VX, die jeweils Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind die gerundeten Werte der entsprechenden Zeilen von Spalte V1, die Werte in der Spalte VX sind die gerundeten Werte der entsprechenden Zeilen von Spalte V2.

Zugehörige Funktionen

Funktion	Syntax
SIGN	Berechnet das Vorzeichen (positiv oder negativ) der Werte im angegebenen Datenbereich.

ACOS-Makro

Die ACOS Makro ist nur in Unica Campaign verfügbar.

Syntax

ACOS(data [, units_keyword])

Parameter

data

Die numerischen Werte, deren Arkuskosinus berechnet werden soll. Dieser Parameter kann ein konstanter Wert, eine Spalte, ein Zellenbereich oder ein Ausdruck sein, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

units_keyword

Dieses optionale Schlüsselwort legt fest, ob die Eingabewerte und Ergebnisse als Grad oder als Radiant interpretiert werden. Wählen Sie einen der folgenden Werte aus:

RADIAN - Führt die Berechnungen in Radiant aus (Standard)

DEGREE - Führt die Berechnungen in Grad aus

Wird dieser Parameter nicht angegeben, ist der Standard Radiant. (Die Umrechnung von Radiant in Grad erfolgt durch Division durch PI und Multiplikation mit 180.)

Weitere Informationen zur Verwendung von Schlüsselwörtern in Unica Campaign finden Sie in Formatspezifikationen (auf Seite 12).

Syntax

ACOS berechnet den Arkuskosinus der Werte im angegebenen Datenbereich. Der Arccosinus ist der Winkel, dessen Kosinus den Inhalt jeder Zelle angibt. ACOSgibt für jede Eingabespalte eine neue Spalte zurück, die jeweils den Arccosin der Zahlen in der entsprechenden Eingabespalte enthält.

Wenn das Schlüsselwort RADIAN verwendet wird, gibt ACOS Werte im Bereich von 0 bis π zurück. Wenn das Schlüsselwort DEGREE verwendet wird, gibt ACOSWerte im Bereich von 0 bis 180 zurück.

Anmerkung: Der Zelleninhalt jeder angegebenen Spalte muss Werte zwischen -1,0 und 1,0 (einschließlich) aufweisen. Andernfalls wird für jede ungültige Eingabe eine leere Zelle zurückgegeben.

Beispiele

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

Erstellt eine Spalte TEMP, die den Wert `1.571` (π⁄2 Radiant) enthält.

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

Erstellt eine Spalte TEMP, die den Wert `90` (Grad) enthält.

```
TEMP = ACOS(V1)
```

Erstellt eine neue Spalte TEMP, in der jeder Wert den Kosinus (in Radianten) des Inhalts von Spalte V1 darstellt.

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

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Die Werte in der Spalte TEMP sind jeweils der Arkuskosinus des Inhalts von Spalte V1, die Werte von Spalte VX sind der Arkuskosinus des Inhalts von Spalte V2 und die Werte von Spalte VY sind der Arkuskosinus des Inhalts von Spalte V3. Alle Werte sind in Grad angegeben.

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

Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen jeweils den Arkuskosinus der Werte in Zeile 10-20 von Spalte V1 darstellen. (in Radianten). Die anderen Zellen in TEMP sind leer.

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

Erstellt zwei neue Spalten TEMP und VX, die jeweils Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind der Arkuskosinus der entsprechenden Zeilen von Spalte V1, die Werte in der Spalte VX sind der Arkuskosinus der entsprechenden Zeilen von Spalte V2. Alle Werte sind in Radiant angegeben.

Zugehörige Funktionen

Funktion	Syntax
ACOT	Berechnet den Arkuskotangens des Inhalts des angegebenen Datenbereichs.
ASIN	Berechnet den Arkussinus des Inhalts des angegebenen Datenbereichs.
ATAN	Berechnet den Arkustangens des Inhalts des angegebenen Datenbereichs.
COS	Berechnet den Kosinus des Inhalts des angegebenen Datenbereichs.

ACOT-Makro

Die ACOT Makro ist nur in Unica Campaign verfügbar.

Syntax

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

Parameter

```
data
```

Die numerischen Werte, deren Arkuskotangens berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`units_keyword`

Dieses optionale Schlüsselwort legt fest, ob die Eingabewerte und Ergebnisse als Grad oder als Radiant interpretiert werden. Eine der folgenden Optionen auswählen:

`RADIAN` - Führt die Berechnungen in Radiant aus (Standard)

`DEGREE` - Führt die Berechnungen in Grad aus

Wird dieser Parameter nicht angegeben, ist der Standard Radiant. (Die Umrechnung von Radiant in Grad erfolgt durch Division durch `PI` und Multiplikation mit 180.)

Weitere Informationen zur Verwendung von Schlüsselwörtern in Unica Campaign finden Sie in Formatspezifikationen (auf Seite 12).

Syntax

`ACOT` gibt den Winkel zurück, dessen Kotangens den Inhalt jeder Zelle darstellt. `ACOT` gibt für jede Eingabespalte eine neue Spalte zurück, die jeweils den Bikotangens der Zahlen in der entsprechenden Eingabespalte enthält. Es werden 64-Bit-Gleitkommazahlen verwendet.

Beispiele

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

Erstellt eine Spalte `TEMP`, die den Wert `2.157` (Radiant) enthält.

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

Erstellt eine Spalte `TEMP`, die den Wert `0.022` (1/45) Grad enthält.

```
TEMP = ACOT(0)
```

Erstellt eine Spalte `TEMP`, die den Wert `MAX32_Float` in Radiant enthält.

```
TEMP = ACOT(V1)
```

Erstellt eine neue Spalte TEMP, in der jeder Wert den Arkustangens (in Radiant) des Inhalts von Spalte V1 darstellt.

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

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Die Werte in der Spalte TEMP sind jeweils der Arkuskotangens des Inhalts von Spalte V1, die Werte von Spalte VX sind der Arkuskotangens des Inhalts von Spalte V2 und die Werte von Spalte VY sind der Arkuskotangens des Inhalts von Spalte V3. Alle Werte sind in Grad angegeben.

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

Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen jeweils den Arkuskotangens der Werte in Zeile 10-20 von Spalte V1 darstellen. Die anderen Zellen in TEMP sind leer.

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

Erstellt zwei neue Spalten TEMP und VX, die jeweils Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind der Arkuskotangens der entsprechenden Zeilen von Spalte V1, die Werte in der Spalte VX sind der Arkuskotangens der entsprechenden Zeilen von Spalte V2. Alle Werte sind in Radiant angegeben.

Zugehörige Funktionen

Funktion	Syntax
ACOS	Berechnet den Arkuskosinus des Inhalts des angegebenen Datenbereichs.
ASIN	Berechnet den Arkussinus des Inhalts des angegebenen Datenbereichs.
ATAN	Berechnet den Arkustangens des Inhalts des angegebenen Datenbereichs.
COT	Berechnet den Kotangens des Inhalts des angegebenen Datenbereichs.

ADD_MONTHS-Makro

Das ADD_MONTHSMakro ist in Unica Campaign verfügbar. Bei ORACLE-, DB2- und MSSQLServer-Datenbanken wird die Ausführung auf der Datenbank statt auf dem Campaign Server bevorzugt.

Syntax

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

Parameter

`months`

Eine Ganzzahl, die eine Anzahl von Monaten darstellt, die zu [Datumszeichenfolge] addiert werden soll.

`date_string`

Eine Zeichenfolge, die ein gültiges Datum in dem Format "DELIM_M_D_Y" oder in dem durch das optionale Argument input_format angegebenen Format darstellt.

`input_format`

Das Format, das für das berechnete Datum verwendet werden soll. Eine Liste der unterstützten Datumsformate finden Sie unter der Funktion DATE_FORMAT. Beachten Sie, dass das `input_format` sowohl das Format der Eingabezeichenkette als auch das Format der Ausgabezeichenkette bestimmt.

Datenbank

Dieser Parameter ist optional. Bei ORACLE/DB2/MSSQLServer wird die Makroausführung auf der Datenbank bevorzugt, auch wenn der Parameter DB nicht angegeben ist. Für die übrigen Datenbanktypen bleibt das Verhalten gleich, d. h. die Ausführung auf dem Campagne-Server.

Die Ausführung wird implizit auf Datenbank ausgeführt, wenn der Ausdruck eine Datenbank enthält.

Die Ausführung erfolgt auf dem Campaign-Server, wenn der Ausdruck alle Nicht-Datenbankspalten enthält, z. B. UCGF oder Datumszeichenfolgen usw. Um die Ausführung in der Datenbank zu erzwingen, fügen Sie den Parameter DB ein. Bitte beachten Sie, dass Sie auch input_format verwenden müssen, um DB-Parameter angeben zu können.

Beschreibung

`ADD_MONTHS` gibt ein Datum zurück, das sich aus der Addition der angegebenen Anzahl von Monaten zu der angegebenen Datumszeichenfolge ergibt. Das Datum wird in dem Standardformat "DELIM_M_D_Y" oder in dem durch das optionale Argument input_format

angegebenen Format zurückgegeben. Mithilfe von DATE_FORMAT können Sie ein anderes Ausgabeformat festlegen.

Wenn die Erhöhung des Monats um die angegebene Anzahl von Monaten ein ungültiges Datum erzeugt, wird als Ergebnis der letzte Tag des Monats berechnet, wie im letzten der folgenden Beispiele zu sehen. Gegebenenfalls werden auch Schaltjahre berücksichtigt. Wenn beispielsweise zum 31. Januar 2012 ein Monat addiert wird, ergibt sich der 29. Februar 2012.

Beispiele

`ADD_MONTHS(12, '06-25-11')` addiert zu dem angegebenen Datum ein Jahr (12 Monate) und gibt das Datum 06-25-12 zurück.

`ADD_MONTHS(3, '2011-06-25', DT_DELIM_Y_M_D)` addiert zu dem angegebenen Datum drei Monate und gibt das Datum 2011-09-25 zurück.

`ADD_MONTHS(1, '02-28-2011')` gibt das Datum 03-28-2011 zurück.

`ADD_MONTHS(1, '03-31-2012')` gibt das Datum 04-30-2012 zurück.

Beispielausdrücke und deren Ausführung. DATE1 und DATE2 sind Datenbankspalten.

S.No	Ausdruck	Ausführung am
1	DATE1 < ADD_MONTHS(1,DATE2)	Datenbank
2	DATE1 < ADD_MONTHS(1,DATE2,DELIM_M_D_Y,DB)	Datenbank
3	ADD_MONTHS(1,'02-29-2016',DELIM_M_D_Y,DB) > DATE1	Datenbank
4	ADD_MONTHS(24,'2012-02-29',DT_DELIM_Y_M_D) > DATE2	Campaign-Server
5	ADD_MONTHS(24,'2012-02-29',DT_DELIM_Y_M_D,DB) < DATE2	Datenbank
6	DATE2 < ADD_MONTHS(1, DATE2)	Campaign-Server
7	DATE2 < ADD_MONTHS(1, DATE2,DELIM_M_D_Y,DB)	Datenbank
8	ADD_MONTHS(24,'2012-02-29', DELIM_Y_M_D, DB) > DATE2	Datenbank
9	ADD_MONTHS(24,'02-29-2020') > DATE2	Campaign-Server

S.No	Ausdruck	Ausführung am
10	DATE1 = ADD_MONTHS(1,DATE2)	Datenbank
11	DATE1 = ADD_MONTHS(1,DATE2,DELIM_M_D_Y,DB)	Datenbank
12	DATE1 != ADD_MONTHS(1,DATE2,DELIM_M_D_Y,DB)	Datenbank
13	DATE1 != ADD_MONTHS(1,DATE2)	Datenbank
14	ADD_MONTHS(3,'11NOV',DDMMM,DB) >DATE_FORMAT(DATE1,DT_DELIM_Y_M_D,DDMMM)	Campaign-Server
15	ADD_MONTHS(0,'2012-02-29',DT_DELIM_Y_M_D) < DATE1	Datenbank
16	ADD_MONTHS(-1, DATE1, DT_DELIM_Y_M_D, DB) < DATE2	Datenbank

Zugehörige Funktionen

Funktion	Beschreibung
`DATE`	Wandelt eine Datumszeichenfolge in ein julianisches Datum um.
`DATE_FORMAT`	Wandelt ein Datum von `input_format` in `output_format` um.

AND-Makro

Das `AND` Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

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

Parameter

`data1`

Die Zahlen, die durch logisches Und mit den Werten in `data2` verknüpft werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`data2`

Die Zahl(en), die durch logisches Und mit den Werten in `data1` verknüpft werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in `data2` muss mit der Anzahl der Spalten in `data1` übereinstimmen, es sei denn, bei `data2` handelt es sich um eine Konstante. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`AND` Berechnet das logische Und zwischen den beiden angegebenen Datenbereichen. Es gibt für jede Eingabespalte eine neue Spalte zurück, wobei jeweils die entsprechende Spalte von `data1` durch logisches Und mit der entsprechenden Spalte von `data2` verknüpft wird (d. h., die erste Spalte von `data1` wird durch logisches Und mit der ersten Spalte von `data` verknüpft, die zweite Spalte mit der zweiten Spalte usw.).

Wenn es sich bei `data2` um eine Konstante handelt, wird jeder Wert in `data1` durch logisches Oder mit dem betreffenden Wert verknüpft. Wenn `data2` eine oder mehrere Spalten enthält, werden die Berechnungen auf Zeilenbasis für eine Spalte aus `data2` und eine Spalte aus `data2` durchgeführt. Die erste Zeile aus `data1` wird mit logischem UND mit dem ersten Zeilenwert aus `data2` verknüpft, die zweite Zeile mit der zweiten Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

Anmerkung: Der Operator `AND` kann durch ein Et-Zeichen (`&&`) abgekürzt werden. Mit dem doppelten Et-Zeichen können die beiden Argumente getrennt werden (statt z. B. `V1 AND 3`anzugeben, können Sie einfach `V1&&3`eingeben).

Beispiele

`TEMP = 1 AND 8` oder `TEMP = 1 && 8`

Erstellt eine neue Spalte mit dem Name `TEMP`, die den Wert eins enthält (jede Zahl ungleich null wird als eins behandelt).

```
TEMP = V1 && 1
```

Erstellt eine neue Spalte TEMP, die für jeden Wert von Spalte V1 den Wert eins enthält.

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

Erstellt eine neue Spalte TEMP, die für jeden Wert ungleich null in Spalte V1 den Wert eins und für jede Null in Spalte V1 den Wert null enthält.

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

Erstellt eine neue Spalte TEMP, in der jeder Zeilenwert von Spalte V1 durch logisches Und mit dem entsprechenden Zeilenwert von Spalte V2 verknüpft ist.

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

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. In der Spalte TEMP sind die Werte von Spalte V1 durch logisches Und mit den entsprechenden Zeilenwerten von Spalte V4 verknüpft. In der Spalte VX sind die Werte von Spalte V2 durch logisches Und mit den Werten von Spalte V5 verknüpft. In der Spalte VY sind die Werte von Spalte V3 durch logisches Und mit den Werten von Spalte V6 verknüpft.

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

Erstellt eine neue Spalte TEMP, in deren ersten 11 Zellen die Werte in Zeile 10-20 von Spalte V1 durch logisches Und mit den Werten in Zeile 1-11 von Spalte V2 verknüpft sind. Die anderen Zellen in TEMP sind leer.

Zugehörige Funktionen

Funktion	Syntax
NOT	Berechnet das logische Nicht des Inhalts des angegebenen Datenbereichs.
OR	Berechnet das logische Oder zwischen den beiden angegebenen Datenbereichen.

ASIN-Makro

Die ASIN Makro ist nur in Unica Campaign verfügbar.

Syntax

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

Parameter

`data`

Die numerischen Werte, deren Arkussinus berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`units_keyword`

Dieses optionale Schlüsselwort legt fest, ob die Eingabewerte und Ergebnisse als Grad oder als Radiant interpretiert werden. Eine der folgenden Optionen auswählen:

`RADIAN` - Führt die Berechnungen in Radiant aus (Standard)

`DEGREE` - Führt die Berechnungen in Grad aus

Wird dieser Parameter nicht angegeben, ist der Standard Radiant. (Die Umrechnung von Radiant in Grad erfolgt durch Division durch `PI` und Multiplikation mit 180.)

Weitere Informationen zur Verwendung von Schlüsselwörtern in Unica Campaign finden Sie in Formatspezifikationen (auf Seite 12).

Syntax

`ASIN` berechnet den Arkussinus der Werte im angegebenen Datenbereich. Der Arkussinus ist der Winkel, dessen Sinus den Inhalt jeder Zelle angibt. `ASIN` gibt für jede Eingabespalte eine neue Spalte zurück, die jeweils den Arkussinus der Zahlen in der entsprechenden Eingabespalte enthält.

Wenn das Schlüsselwort `RADIAN` verwendet wird, gibt `ASIN` Werte im Bereich -π/2 bis π/2 zurück. Wenn das Schlüsselwort `DEGREE` verwendet wird, gibt `ASIN`Werte im Bereich von -90 bis 90 zurück.

Anmerkung: Der Zelleninhalt jeder angegebenen Spalte muss Werte zwischen -1,0 und 1,0 (einschließlich) aufweisen. Andernfalls wird für jede ungültige Eingabe `???` zurückgegeben.

Beispiele

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

Erstellt eine neue Spalte `TEMP`, die den Wert `0.524` ($\pi/6$ Radiant) enthält.

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

Erstellt eine Spalte `TEMP`, die den Wert `30` (Grad) enthält.

```
TEMP = ASIN(V1)
```

Erstellt eine neue Spalte `TEMP`, in der jeder Wert den Arkussinus (in Radiant) des Inhalts der Spalte `V1` darstellt.

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

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Werte in der Spalte `TEMP` sind jeweils der Arkussinus des Inhalts von Spalte `V1`, die Werte von Spalte `VX` sind der Arkussinus des Inhalts von Spalte `V2` und die Werte von Spalte `VY` sind der Arkussinus des Inhalts von Spalte `V3`. Alle Werte sind in Grad angegeben.

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

Erstellt eine neue Spalte `TEMP`, in der die ersten 11 Zellen jeweils den Arkussinus der Werte in Zeile 10-20 von Spalte `V1` darstellen. (in Radianten). Die anderen Zellen in `TEMP` sind leer.

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

Erstellt zwei neue Spalten `TEMP` und `VX`, die jeweils Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte `TEMP` sind der Arkussinus der entsprechenden Zeilen von Spalte `V1`, die Werte in der Spalte `VX` sind der Arkussinus der entsprechenden Zeilen von Spalte `V2`. Alle Werte sind in Radiant angegeben.

Zugehörige Funktionen

Funktion	Syntax
`ACOS`	Berechnet den Arkuskosinus des Inhalts des angegebenen Datenbereichs.
`ACOT`	Berechnet den Arkuskotangens des Inhalts des angegebenen Datenbereichs.
`ATAN`	Berechnet den Arkustangens des Inhalts des angegebenen Datenbereichs.
`SIN`	Berechnet den Sinus des Inhalts des angegebenen Datenbereichs.

ATAN-Makro

Die `ATAN` Makro ist nur in Unica Campaign verfügbar.

Syntax

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

Parameter

`data`

Die numerischen Werte, deren Arkustangens berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`units_keyword`

Dieses optionale Schlüsselwort legt fest, ob die Eingabewerte und Ergebnisse als Grad oder als Radiant interpretiert werden. Eine der folgenden Optionen auswählen:

`RADIAN` - Führt die Berechnungen in Radiant aus (Standard)

`DEGREE` - Führt die Berechnungen in Grad aus

Wird dieser Parameter nicht angegeben, ist der Standard Radiant. (Die Umrechnung von Radiant in Grad erfolgt durch Division durch `PI` und Multiplikation mit 180.)

Weitere Informationen zur Verwendung von Schlüsselwörtern in Unica Campaign finden Sie in Formatspezifikationen (auf Seite 12).

Syntax

`ATAN` berechnet den Arkustangens der Werte im angegebenen Datenbereich. Der Arktangens ist der Winkel, dessen Tangens den Inhalt jeder Zelle darstellt. `ATAN` gibt für jede Eingabespalte eine neue Spalte zurück, die jeweils den Arktangens der Zahlen in der entsprechenden Eingabespalte enthält.

Wenn das Schlüsselwort RADIAN verwendet wird, gibt ATAN Werte im Bereich -π/2 bis π/2 zurück. Wenn das Schlüsselwort DEGREE verwendet wird, gibt ATANWerte im Bereich von -90 bis 90 zurück.

Beispiele

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

Erstellt eine neue Spalte TEMP, die den Wert `0.785` ($\pi/4$ Radiant) enthält.

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

Erstellt eine Spalte TEMP, die den Wert `45` (Grad) enthält.

```
TEMP = ATAN(V1)
```

Erstellt eine neue Spalte TEMP, in der jeder Wert den Arktangens (in Radiant) des Inhalts von Spalte V1 darstellt.

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

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Die Werte in der Spalte TEMP sind jeweils der Arkustangens des Inhalts von Spalte V1, die Werte von Spalte VX sind der Arkustangens des Inhalts von Spalte V2 und die Werte von Spalte VY sind der Arkustangens des Inhalts von Spalte V3. Alle Werte sind in Grad angegeben.

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

Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen jeweils den Arkustangens der Werte in Zeile 10-20 von Spalte V1 darstellen. (in Radiant). Die anderen Zellen in TEMP sind leer.

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

Erstellt zwei neue Spalten TEMP und VX, die jeweils Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind der Arkustangens der entsprechenden Zeilen von Spalte V1, die Werte in der Spalte VX sind der Arkustangens der entsprechenden Zeilen von Spalte V2. Alle Werte sind in Radiant angegeben.

Zugehörige Funktionen

Funktion	Syntax
ACOS	Berechnet den Arkuskosinus des Inhalts des angegebenen Datenbereichs.

Funktion	Syntax
ASIN	Berechnet den Arkussinus des Inhalts des angegebenen Datenbereichs.
ATAN	Berechnet den Arkustangens des Inhalts des angegebenen Datenbereichs.
TAN	Berechnet den Tangens des Inhalts des angegebenen Datenbereichs.

AVG-Makro

Die AVG Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

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

Parameter

`data`

Die numerischen Werte, deren arithmetisches Mittel berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen der obigen Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`keyword`

Dieses optionale Schlüsselwort legt fest, wie die Berechnung auf den Eingabedatenbereich angewendet wird. Wählen Sie eines der folgenden Schlüsselwörter aus:

`ALL` - Wendet die Berechnung auf alle Zellen in `data` an (Standard)

`COL` - Führt die Berechnung für jede Zeile von data gesondert aus `data`

`ROW` - Führt die Berechnung für jede Spalte von data gesondert aus `data`

Weitere Informationen zur Verwendung von Schlüsselwörtern in Unica Campaign finden Sie in Formatspezifikationen (auf Seite 12).

Anmerkung: Viele Makrofunktionen nehmen die Schlüsselwortparameter `{ALL | COL | ROW}` an. Diese Schlüsselwörter gelten nicht für **Unica Campaign**, da es sich bei den

Eingabedaten immer um eine Einzelspalte oder ein Einzelfeld handelt. Das Makro verhält sich immer so, als ob das Schlüsselwort `COL` angegeben würde. Daher brauchen Sie diese Schlüsselwörter nicht anzugeben, wenn Sie **Unica Campaign** verwenden.

Syntax

`AVG` Berechnet das arithmetische Mittel oder den Durchschnitt der Zellen im angegebenen Datenbereich. Das arithmetische Mittel wird berechnet, indem der Inhalt aller Zellen addiert und das Ergebnis durch die Anzahl der Zellen geteilt wird. Die Anzahl der von `AVG` zurückgegebenen Spalten hängt von `keyword` ab.

- Wenn `keyword` den Wert `ALL` hat, gibt `AVG` eine neue Spalte zurück, die einen Einzelwert (den Durchschnitt aller Zellen in `data`) enthält.
- Wenn `keyword` den Wert `COL` hat, gibt `AVG` für jede Eingabespalte eine neue Spalte zurück. Jede neue Spalte enthält einen Wert (den Durchschnitt aller Zellen in der entsprechenden Eingabespalte).
- Wenn `keyword` den Wert `ROW` hat, gibt `AVG` eine neue Spalte zurück, die den Durchschnitt für jede Zeile von `data` enthält.

Anmerkung: Leere Zellen werden bei der Berechnung ignoriert.

Anmerkung: `AVG` ist mit der Makrofunktion `MEAN` identisch.

Beispiele

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

Erstellt eine Spalte namens TEMP, die einen Einzelwert enthält, der das arithmetische Mittel des Inhalts von Spalte V1 darstellt.

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

Erstellt eine Spalte namens TEMP, die einen Einzelwert enthält, der das arithmetische Mittel des Inhalts von Spalte V1, V2 und V3 darstellt.

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

Erstellt eine Spalte namens TEMP, die einen Einzelwert enthält, der das arithmetische Mittel der Zellen in den Zeilen 10 bis 20 von Spalte V1 darstellt.

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

Erstellt eine Spalte namens TEMP, die einen Einzelwert enthält, der das arithmetische Mittel der Zellen in den Zeilen 1 bis 5 der Spalten V1 bis V4 darstellt.

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

Erstellt drei neue Spalten namens TEMP, VX und VY. Der Einzelwert in der Spalte TEMP ist das arithmetische Mittel des Inhalts von Spalte V1. Der Einzelwert in der Spalte VX ist das arithmetische Mittel des Inhalts von Spalte V2. Der Einzelwert in der Spalte VY ist das arithmetische Mittel des Inhalts von Spalte V3.

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

Erstellt drei neue Spalten namens TEMP, VX und VY, die jeweils einen Einzelwert enthalten. Der Wert in der Spalte TEMP ist das arithmetische Mittel der Zellen in den Zeilen 1 bis 5 von Spalte V1. Der Wert in der Spalte VX ist das arithmetische Mittel der Zellen in den Zeilen 1 bis 5 von Spalte V2. Der Wert in der Spalte VY ist das arithmetische Mittel der Zellen in den Zeilen 1 bis 5 von Spalte V3.

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

Erstellt eine Spalte namens TEMP, die dieselben Werte wie Spalte V1 enthält (das arithmetische Mittel einer beliebigen Zahl ist die Zahl selbst).

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

Erstellt eine Spalte namens TEMP, in der jeder Zelleneintrag das arithmetische Mittel der entsprechenden Zeile in den Spalten V1, V2 und V3 darstellt.

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

Erstellt eine Spalte namens TEMP, in der die Zellen in den Zeilen 1 bis 5 das arithmetische Mittel der entsprechenden Zeile in den Spalten V1 bis V3 enthalten. Die anderen Zellen in TEMP sind leer.

Zugehörige Funktionen

Funktion	Syntax
SUM oder TOTAL	Berechnet die Summe eines Zellenbereichs.

BETWEEN-Makro

Die `BETWEEN` Makro ist nur in Unica Campaign verfügbar.

Syntax

```
value1 BETWEEN value2 AND value3
```

Parameter

```
Equivalent to value1 >= value2 AND < value3
```

Syntax

BETWEEN ist eine spezielle Variante des Vergleichsprädikats. Es kommt auf die Details dieses Prädikats an; die Reihenfolge der Operanden hat einige unerwartete Auswirkungen. Vergleichen Sie dazu die nachfolgenden Beispiele.

Anmerkung: FROM und FOR verwenden eine identische Syntax.

Beispiele

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

da der entsprechende Ausdruck mit BETWEEN und AND eine bestimmte Reihenfolge aufweist, die keine Rolle spielt, wenn Sie Literale verwenden, aber von großer Bedeutung ist, wenn Sie durch Variablen, Parameter oder sogar Unterabfragen `value2` und `value3` angeben.

BIT_AND-Makro

Die `BIT_AND` Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

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

Parameter

```
data1
```

Die nicht negativen Ganzzahlen, die durch bitweises Und mit den Werten in `data2` verknüpft werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`data2`

Die nicht negative(n) Ganzzahl(en), die durch bitweises Und mit den Werten in `data1` verknüpft werden soll(en). Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in `data2` muss mit der Anzahl der Spalten in `data1` übereinstimmen, es sei denn, bei `data2` handelt es sich um eine Konstante. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`BIT_AND` berechnet das bitweise Und zwischen den beiden angegebenen Datenbereichen. Es gibt für jede Eingabespalte eine neue Spalte zurück, wobei jeweils die entsprechende Spalte von `data1` durch bitweises Und mit der entsprechenden Spalte von `data2` verknüpft wird (d. h., die erste Spalte von `data1` wird durch logisches Und mit der ersten Spalte von `data` verknüpft, die zweite Spalte mit der zweiten Spalte usw.).

Wenn es sich bei `data2` um eine Konstante handelt, wird jeder Wert in `data1` durch bitweises Und mit dem betreffenden Wert verknüpft. Wenn `data2` eine oder mehrere Spalten enthält, werden die Berechnungen auf Zeilenbasis für eine Spalte aus `data2` und eine Spalte aus `data2` durchgeführt. Die erste Zeile von `data1` wird durch bitweises Und mit der ersten Zeile von `data2` verknüpft, die zweite Zeile mit der zweiten Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

Anmerkung: Bei dieser Makrofunktion ist die Genauigkeit auf Ganzzahlwerte kleiner als 2^{24} begrenzt. Negative Werte sind nicht zulässig.

Anmerkung: Der `BIT_AND` Operator kann durch ein Et-Zeichen (`&`) abgekürzt werden. Mit dem Et-Zeichen können die beiden Argumente getrennt werden (statt z. B. `BIT_AND(V1, 3)`anzugeben, können Sie einfach `V1&3`eingeben).

Beispiele

`TEMP = 3 BIT_AND 7` oder `TEMP = 3 & 7`

Erstellt eine neue Spalte `TEMP`, die den Wert drei enthält (das bitweise Und von `011` und `111` ergibt `011`).

`TEMP = V1 & 8`

Erstellt eine neue Spalte `TEMP`, in der der Inhalt von Spalte `V1` jeweils durch bitweises Und mit dem Binärwert `1000` verknüpft ist.

`TEMP = V1 & V1`

Erstellt eine neue Spalte `TEMP`, die denselben Inhalt wie die Spalte `V1` enthält (jeder Wert ergibt bei Verknüpfung durch UND mit sich selbst den Wert selbst).

`TEMP = V1 & V2`

Erstellt eine neue Spalte `TEMP`, in der jeder Zeilenwert von Spalte `V1` durch bitweises Und mit dem entsprechenden Zeilenwert von Spalte `V2` verknüpft ist.

`TEMP = V1:V3 & V4:V6`

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. In der Spalte `TEMP` sind die Werte von Spalte `V1` durch bitweises Und mit den entsprechenden Zeilenwerten von Spalte `V4` verknüpft. In der Spalte `VX` sind die Werte von Spalten `V2` und `V5` durch bitweises Und verknüpft. In der Spalte `VY` sind die Werte von Spalten `V3` und `V6` durch bitweises Und verknüpft.

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

Erstellt eine neue Spalte `TEMP`, in deren ersten 11 Zellen die Werte in Zeile 10-20 von Spalte `V1` durch bitweises Und mit den Werten in Zeile 1-11 von Spalte `V2` verknüpft sind. Die anderen Zellen in `TEMP` sind leer.

Zugehörige Funktionen

Funktion	Syntax
`BIT_NOT`	Berechnet das bitweise Nicht des Inhalts des angegebenen Datenbereichs.

Funktion	Syntax
`BIT_OR`	Berechnet das bitweise Oder zwischen den beiden angegebenen Datenbereichen.
`BIT_XOR` oder `XOR`	Berechnet das bitweise exklusive Oder zwischen den beiden angegebenen Datenbereichen.

BIT_NOT-Makro

Die `BIT_NOT` Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

```
BIT_NOT data ~ data
```

Parameter

`data`

Die nicht negativen Ganzzahlen, die durch bitweises Nicht verknüpft werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`BIT_NOT` gibt das bitweise Nicht der Werte im angegebenen Datenbereich zurück. gibt für jede Eingabespalte eine neue Spalte zurück, die jeweils das bitweise Nicht der Werte in der entsprechenden Eingabespalte von `data` enthält.

Anmerkung: Bei dieser Makrofunktion ist die Genauigkeit auf Ganzzahlwerte kleiner als 2^{24} begrenzt. Negative Werte sind nicht zulässig.

Anmerkung: Wenn eine Spalte in jeder Zeile dieselbe Zahl *x* wie `data` enthält, ist dies dasselbe, als wenn als *x* die Konstante `data` verwendet wird.

Anmerkung: Der Operator `BIT_NOT` kann durch eine Tilde (~) abgekürzt werden. Geben Sie die Tilde vor dem Datenwert an. Beispiel: Zur Angabe von `BIT_NOT(V1)` können Sie einfach `~V1` eingeben.

Beispiele

`TEMP = BIT_NOT 3` oder `TEMP = ~3`

Erstellt eine neue Spalte `TEMP`, die den Wert vier enthält (das bitweise Nicht von `011` ergibt `100`).

`TEMP = ~V1`

Erstellt eine neue Spalte `TEMP`, in der jeder Wert das bitweise NICHT des Inhalts der Spalte `V1` darstellt.

`TEMP = ~V1:V3`

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Werte in der Spalte `TEMP` sind das bitweise NICHT des Inhalts von Spalte `V1`, die Werte von Spalte `VX` sind das bitweise NICHT des Inhalts von Spalte `V2` und die Werte von Spalte `VY` sind das bitweise NICHT des Inhalts von Spalte `V3`.

`TEMP = ~V1[100:200]`

Erstellt eine neue Spalte `TEMP`, in der die ersten 101 Zellen das bitweise NICHT der Werte in den Zeilen 1-50 der Spalte `V1` enthalten.

Zugehörige Funktionen

Funktion	Syntax
`BIT_AND`	Berechnet das bitweise Und zwischen den beiden angegebenen Datenbereichen.
`BIT_OR`	Berechnet das bitweise Oder zwischen den beiden angegebenen Datenbereichen.
`BIT_XOR` oder `XOR`	Berechnet das bitweise exklusive Oder zwischen den beiden angegebenen Datenbereichen.

BIT_OR-Makro

Die `BIT_OR` Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

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

Parameter

`data1`

Die nicht negativen Ganzzahlen, die durch bitweises Oder mit den Werten in `data2` verknüpft werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`data2`

Die nicht negativen Ganzzahlen, die durch bitweises Oder mit den Werten in `data1` verknüpft werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in `data2` muss mit der Anzahl der Spalten in `data1` übereinstimmen, es sei denn, bei `data2` handelt es sich um eine Konstante. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`BIT_OR` berechnet das bitweise Oder zwischen den beiden angegebenen Datenbereichen. Es gibt für jede Eingabespalte eine neue Spalte zurück, wobei jeweils die entsprechende Spalte von `data1` durch bitweises Oder mit der entsprechenden Spalte von `data2` verknüpft wird (d. h., die erste Spalte von `data1` wird durch bitweises Oder mit der ersten Spalte von `data` verknüpft, die zweite Spalte mit der zweiten Spalte usw.).

Wenn es sich bei `data2` um eine Konstante handelt, wird jeder Wert in `data1` durch bitweises Und mit dem betreffenden Wert verknüpft. Wenn `data2` eine oder mehrere Spalten enthält, werden die Berechnungen auf Zeilenbasis für eine Spalte aus `data2` und eine Spalte

aus `data2` durchgeführt. Die erste Zeile aus `data1` wird mit bitweisem ODER mit dem ersten Zeilenwert aus `data2` verknüpft, die zweite Zeile mit der zweiten Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

Anmerkung: Bei dieser Makrofunktion ist die Genauigkeit auf Ganzzahlwerte kleiner als 2^{24} begrenzt. Negative Werte sind nicht zulässig.

Anmerkung: Der Operator `BIT_OR` kann durch einen vertikalen Strich (`|`) abgekürzt werden. Mit dem vertikalen Strich können die beiden Spalten getrennt werden (statt z. B. `BIT_OR(V1, 3)` anzugeben, können Sie einfach `V1|3` eingeben. Sie können auch `OR` verwenden.

Beispiele

`TEMP = 3 BIT_OR 7` oder `TEMP = 3 OR 7` oder `TEMP = 3 | 7`

Erstellt eine neue Spalte `TEMP`, die den Wert drei enthält (das bitweise Oder von `011` und `111` ergibt `111`).

```
TEMP = V1 | 8
```

Erstellt eine neue Spalte `TEMP`, in der der Inhalt von Spalte `V1` jeweils durch bitweises Oder mit dem Binärwert `1000` verknüpft ist.

```
TEMP = V1 | V1
```

Erstellt eine neue Spalte `TEMP`, die denselben Inhalt wie die Spalte `V1` enthält (jeder Wert ergibt bei Verknüpfung durch Oder mit sich selbst den Wert selbst).

```
TEMP = V1 | V2
```

Erstellt eine neue Spalte `TEMP`, in der jeder Zeilenwert von Spalte `V1` durch bitweises Oder mit dem entsprechenden Zeilenwert von Spalte `V2` verknüpft ist.

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

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. In der Spalte `TEMP` sind die Werte von Spalte `V1` durch logisches Oder mit den entsprechenden Zeilenwerten von Spalte `V4` verknüpft. In der Spalte `VX` sind die Werte von Spalten `V2` und `V5` durch logisches Oder verknüpft. In der Spalte `VY` sind die Werte von Spalten `V3` und `V6` durch logisches Oder verknüpft.

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

Erstellt eine neue Spalte `TEMP`, in deren ersten 11 Zellen die Werte in Zeile 10-20 von Spalte `V1` durch bitweises Oder mit den Werten in Zeile 1-11 von Spalte `V2` verknüpft sind. Die anderen Zellen in `TEMP` sind leer.

Zugehörige Funktionen

Funktion	Syntax
`BIT_AND`	Berechnet das bitweise Und zwischen den beiden angegebenen Datenbereichen.
`BIT_NOT`	Berechnet das bitweise Nicht des Inhalts des angegebenen Datenbereichs.
`BIT_XOR` oder `XOR`	Berechnet das bitweise exklusive Oder zwischen den beiden angegebenen Datenbereichen.

BIT_XOR-Makro

Die `BIT_XOR` Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

```
data1 BIT_XOR data2
```

Parameter

`data1`

Die nicht negativen Ganzzahlen, die durch bitweises exklusives Oder mit den Werten in `data2` verknüpft werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`data2`

Die nicht negativen Ganzzahlen, die durch bitweises exklusives Oder mit den Werten in `data1` verknüpft werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser

Typen ergibt. Die Anzahl der Spalten in `data2` muss mit der Anzahl der Spalten in `data1` übereinstimmen, es sei denn, bei `data2` handelt es sich um eine Konstante. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`BIT_XOR` berechnet das bitweise exklusives Oder zwischen den beiden angegebenen Datenbereichen. Es gibt für jede Eingabespalte eine neue Spalte zurück, wobei jeweils die entsprechende Spalte von `data1` durch bitweises exklusives Oder mit der entsprechenden Spalte von `data2` verknüpft wird (d. h., die erste Spalte von `data1` wird durch bitweises exklusives Oder mit der ersten Spalte von `data` verknüpft, die zweite Spalte mit der zweiten Spalte usw.).

Wenn es sich bei `data2` um eine Konstante handelt, wird jeder Wert in `data1` durch bitweises exklusives Oder mit dem betreffenden Wert verknüpft. Wenn `data2` eine oder mehrere Spalten enthält, werden die Berechnungen auf Zeilenbasis für eine Spalte aus `data2` und eine Spalte aus `data2` durchgeführt. Die erste Zeile von `data1` wird durch bitweises exklusives Oder mit der ersten Zeile von `data2` verknüpft, die zweite Zeile mit der zweiten Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

Anmerkung: Bei dieser Makrofunktion ist die Genauigkeit auf Ganzzahlwerte kleiner als 2^{24} begrenzt. Negative Werte sind nicht zulässig.

Beispiele

```
TEMP = 3 BIT_XOR 7
```

Erstellt eine neue Spalte `TEMP`, die den Wert vier enthält (das bitweise exklusive Oder von `011` und `111` ergibt `100`).

```
TEMP = V1 BIT_XOR 8
```

Erstellt eine neue Spalte `TEMP`, in der der Inhalt von Spalte `V1` jeweils durch bitweises Und mit dem Binärwert `1000` verknüpft ist.

```
TEMP = V1 BIT_XOR V1
```

Erstellt eine neue Spalte TEMP, die nur Nullen enthält (jeder Wert, der durch exklusives Oder mit sich selbst verknüpft wird, ergibt null).

```
TEMP = V1 BIT_XOR V2
```

Erstellt eine neue Spalte TEMP, in der jeder Zeilenwert von Spalte V1 durch bitweises exklusives Oder mit dem entsprechenden Zeilenwert von Spalte V2 verknüpft ist.

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

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. In der Spalte TEMP sind die Werte von Spalte V1 durch bitweises exklusives Oder mit den entsprechenden Zeilenwerten von Spalte V4 verknüpft. In der Spalte VX sind die Werte von Spalten V2 und V5 durch bitweises exklusives Oder verknüpft. In der Spalte VY sind die Werte von Spalten V3 und V6 durch bitweises exklusives Oder verknüpft.

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

Erstellt eine neue Spalte TEMP, in deren ersten 11 Zellen die Werte in Zeile 10-20 von Spalte V1 durch bitweises exklusives Oder mit den Werten in Zeile 1-11 von Spalte V2 verknüpft sind. Die anderen Zellen in TEMP sind leer.

Zugehörige Funktionen

Funktion	Syntax
BIT_AND	Berechnet das bitweise Und zwischen den beiden angegebenen Datenbereichen.
BIT_NOT	Berechnet das bitweise Nicht des Inhalts des angegebenen Datenbereichs.
BIT_OR	Berechnet das bitweise Oder zwischen den beiden angegebenen Datenbereichen.

CEILING-Makro

Die CEILING Makro ist nur in Unica Campaign verfügbar.

Syntax

```
CEILING(data)
```

Parameter

```
data
```

Die numerischen Werte, deren Obergrenze berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`CEILING` berechnet die Obergrenze der Werte im angegebenen Datenbereich. Die Obergrenze einer Zahl ist die kleinste Zahl, nicht kleiner als der Ganzzahl. `CEILING` gibt für jede Eingabespalte eine neue Spalte zurück, die jeweils die Obergrenze der Zahlen in der entsprechenden Eingabespalte enthält.

Beispiele

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

Erstellt eine neue Spalte `TEMP`, die den Wert `5` enthält.

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

Erstellt eine neue Spalte `TEMP`, die den Wert `-2` enthält.

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

Erstellt eine neue Spalte `TEMP`, in der jeder Wert die Obergrenze des Inhalts der Spalte `V1` darstellt.

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

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Werte in der Spalte `TEMP` sind Obergrenzen für den Inhalt der Spalte `V1`, die Werte von Spalte `VX` sind die Obergrenzen für den Inhalt der Spalte `V2` und die Werte von Spalte `VY` sind die Obergrenzen für den Inhalt der Spalte `V3`.

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

Erstellt eine neue Spalte `TEMP`, in der die ersten 11 Zellen die Obergrenzen der Werte den in den Zeilen 10-20 von Spalte `V1` enthalten. Die anderen Zellen in `TEMP` sind leer.

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

Erstellt zwei neue Spalten TEMP und VX, die jeweils Werte in Zeile 1-50 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind jeweils die Obergrenze der Zeilen von Spalte V1, die Werte in der Spalte VX sind die Obergrenze der Zeilen von Spalte V2.

Zugehörige Funktionen

Funktion	Syntax
FLOOR oder INT	Berechnet die Untergrenze jedes Werts im angegebenen Datenbereich.
FRACTION	Berechnet die Nachkommastellen jedes Werts im angegebenen Datenbereich.
TRUNCATE	Berechnet den Ganzzahlanteil jedes Werts im angegebenen Datenbereich.

COLUMN-Makro

Die COLUMN Makro ist nur in Unica Campaign verfügbar.

Syntax

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

Parameter

`data`

Ein Wert, der beim Erstellen einer Spalte verwendet werden soll. Dabei kann es sich um einen konstanten Wert (numerisch oder ASCII-Text in Anführungszeichen), eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Dieser Parameter kann mehrmals wiederholt werden, doch nachfolgende Parameter müssen dieselbe Dimensionalität (d. h. Spaltenbreite) wie der erste Parameter aufweisen. Alle Werte in allen `data`-Parametern müssen entweder numerisch oder ASCII-Text sein (d. h., Zahlen- und Textwerte dürfen nicht gemischt werden). Wenn mehrere `data`-Parameter angegeben werden, müssen alle dieselbe Anzahl von Spalten aufweisen. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

COLUMN verkettet seine Eingaben in Spalten einer Funktionsgruppe. Dabei wird dieselbe Anzahl neuer Spalten zurückgegeben, wie im jeweiligen Eingabeparameter angegeben. Es kann eine unbegrenzte Zahl von Argumenten angegeben werden. Alle Argumente müssen entweder numerisch oder ASCII-Zeichenfolgen sein (d. h., Zahlen- und Textwerte dürfen nicht gemischt werden).

Anmerkung: Die Makrofunktion COLUMN kann abgekürzt werden, indem die data-Argumente innerhalb runder Klammern durch Kommas getrennt aufgelistet werden (z. B. TEMP= MEAN((1,2,3,4), ALL)). Wenn es nicht um den Einsatz innerhalb einer anderen Makrofunktion geht, sind die Klammern nicht erforderlich; beispielsweise ist V1=1,2,3 äquivalent zu V1=COLUMN(1,2,3).

Beispiele

TEMP = COLUMN(3, 4, 5) oder TEMP = (3,4,5) oder TEMP = 3,4,5

Erstellt eine neue Spalte TEMP, deren erste drei Zellen die Werte 3, 4 und 5 enthalten.

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

Erstellt eine neue Spalte TEMP, deren erste drei Zellen die Werte "one", "two " und " three" enthalten.

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

Erstellt eine neue Spalte TEMP, die in der ersten Zelle den Durchschnitt von Spalte V1 und in der zweiten Zelle die Standardabweichung von Spalte V1 enthält.

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

Erstellt zwei neue Spalten TEMP und VX. Die Spalte TEMP enthält die Werte aus Spalte V1, gefolgt von den Werten aus Spalte V3. Die Spalte VX enthält die Werte aus Spalte V2, gefolgt von den Werten aus Spalte V4.

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

Erstellt zwei neue Spalten TEMPund VX. Die Spalte TEMP enthält die Werte aus Zelle 1-10 von Spalte V1, gefolgt von allen Werten aus Spalte V3. Die Spalte VXenthält die Werte aus Zelle 1-10 von Spalte V2, gefolgt von allen Werten aus Spalte V4.

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

Erstellt drei neue Spalten namens TEMP und VX, die jeweils einen Einzelwert enthalten. Die Spalte TEMP enthält den Durchschnitt der Spalten V1 und V2. Die Spalte VX enthält den Durchschnitt der Spalten V3 und V4.

COS-Makro

Die COS Makro ist nur in Unica Campaign verfügbar.

Syntax

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

Parameter

data

Die numerischen Werte, deren Kosinus berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

units_keyword

Dieses optionale Schlüsselwort legt fest, ob die Eingabewerte und Ergebnisse als Grad oder als Radiant interpretiert werden. Eine der folgenden Optionen auswählen:

RADIAN - Führt die Berechnungen in Radiant aus (Standard)

DEGREE - Führt die Berechnungen in Grad aus

Wird dieser Parameter nicht angegeben, ist der Standard Radiant. (Die Umrechnung von Radiant in Grad erfolgt durch Division durch PI und Multiplikation mit 180.)

Weitere Informationen zur Verwendung von Schlüsselwörtern in Unica Campaign finden Sie in Formatspezifikationen (auf Seite 12).

Syntax

COS berechnet den Kosinus der Werte im angegebenen Datenbereich. COS gibt für jede Eingabespalte eine neue Spalte zurück, die den Kosinus der Zahlen in der entsprechenden Eingabespalte enthält.

Beispiele

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

Gibt eine neue Spalte TEMP zurück, die den Einzelwert `-1` enthält.

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

Gibt eine neue Spalte TEMP zurück, die den Einzelwert null enthält.

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

Erstellt eine neue Spalte TEMP, in der jeder Wert der Kosinus (in Radiant) des Inhalts der Spalte V1 entspricht

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

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Die Werte in der Spalte TEMP sind jeweils die Kosinuswerte des Inhalts von Spalte V1, die Werte von Spalte VX sind die Kosinuswerte des Inhalts von Spalte V2 und die Werte von Spalte VY sind die Kosinuswerte des Inhalts von Spalte V3. Alle Werte sind in Grad angegeben.

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

Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen jeweils die Kosinusse der Werte in Zeile 10-20 von Spalte V1 darstellen. (in Radianten). Die anderen Zellen in TEMP sind leer.

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

Erstellt zwei neue Spalten TEMP und VX, die jeweils Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind die Kosinuswerte der entsprechenden Zeilen von Spalte V1, die Werte in der Spalte VX sind die Kosinuswerte der entsprechenden Zeilen von Spalte V2. Alle Werte sind in Radiant angegeben.

Zugehörige Funktionen

Funktion	Syntax
ACOS	Berechnet den Arkuskosinus des Inhalts des angegebenen Datenbereichs.

Funktion	Syntax
COSH	Berechnet den Hyperbelkosinus des Inhalts des angegebenen Datenbereichs.
SIN	Berechnet den Sinus des Inhalts des angegebenen Datenbereichs.
TAN	Berechnet den Tangens des Inhalts des angegebenen Datenbereichs.

COSH-Makro

Die COSH Makro ist nur in Unica Campaign verfügbar.

Syntax

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

Parameter

`data`

Die numerischen Werte, deren Hyperbelkosinus berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`units_keyword`

Dieses optionale Schlüsselwort legt fest, ob die Eingabewerte und Ergebnisse als Grad oder als Radiant interpretiert werden. Eine der folgenden Optionen auswählen:

`RADIAN` - Führt die Berechnungen in Radiant aus (Standard)

`DEGREE` - Führt die Berechnungen in Grad aus

Wird dieser Parameter nicht angegeben, ist der Standard Radiant. (Die Umrechnung von Radiant in Grad erfolgt durch Division durch `PI` und Multiplikation mit 180.)

Weitere Informationen zur Verwendung von Schlüsselwörtern in Unica Campaign finden Sie in Formatspezifikationen (auf Seite 12).

Syntax

COSH berechnet den hyperbolischen Kosinus der Werte im angegebenen Datenbereich. Bei *x* in Radiant ist der Hyperbelkosinus einer Zahl

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

.

wobei e die natürliche Zahl ist, 2,7182818. COSH gibt für jede Eingabespalte eine neue Spalte zurück, die jeweils den hyperbolischen Kosinus der Zahlen in der entsprechenden Eingabespalte enthält.

Anmerkung: Wenn der Wert *x* zu groß ist, wird ein Überlauffehler zurückgegeben. Dies geschieht, wenn *cosh* (*x*) den maximalen 32-Bit-Gleitkommawert überschreitet.

Beispiele

`TEMP = COSH(0)` oder `TEMP = COSH(0, 0)` oder `TEMP = COSH(0, RADIAN)`

Gibt eine neue Spalte TEMP zurück, die den Wert eins enthält.

```
TEMP = COSH(V1)
```

Erstellt eine neue Spalte TEMP, in der jeder Wert den hyperbolischen Kosinus (in Radiant) des Inhalts der Spalte V1 darstellt.

`TEMP = COSH(V1:V3, 1)` oder `TEMP = COSH(V1:V3, DEGREE)`

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Die Werte in der Spalte TEMP sind die hyperbolischen Kosinuswerte des Inhalts von Spalte V1, die Werte von Spalte VX sind die hyperbolischen Kosinuswerte des Inhalts von Spalte V2 und die Werte von Spalte VY sind die hyperbolischen Kosinuswerte des Inhalts von Spalte V3. Alle Werte sind in Grad angegeben.

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

Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen jeweils die hyperbolischen Kosinusse der Werte in Zeile 10-20 von Spalte V1 darstellen. (in Radianten). Die anderen Zellen in TEMP sind leer.

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

Erstellt zwei neue Spalten TEMP und VX, die jeweils Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind die hyperbolischen Kosinuswerte der

entsprechenden Zeilen von Spalte V1, die Werte in der Spalte VX sind die hyperbolischen Kosinuswerte der entsprechenden Zeilen von Spalte V2. Alle Werte sind in Radiant angegeben.

Zugehörige Funktionen

Funktion	Syntax
ACOS	Berechnet den Arkuskosinus des Inhalts des angegebenen Datenbereichs.
COS	Berechnet den Kosinus des Inhalts des angegebenen Datenbereichs.
SINH	Berechnet den Hyperbelsinus des Inhalts des angegebenen Datenbereichs.
TANH	Berechnet den Hyperbeltangens des Inhalts des angegebenen Datenbereichs.

COT-Makro

Die COT Makro ist nur in Unica Campaign verfügbar.

Syntax

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

Parameter

`data`

Die numerischen Werte, deren Kotangens berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`units_keyword`

Dieses optionale Schlüsselwort legt fest, ob die Eingabewerte und Ergebnisse als Grad oder als Radiant interpretiert werden. Eine der folgenden Optionen auswählen:

RADIAN - Führt die Berechnungen in Radiant aus (Standard)

DEGREE - Führt die Berechnungen in Grad aus

Wird dieser Parameter nicht angegeben, ist der Standard Radiant. (Die Umrechnung von Radiant in Grad erfolgt durch Division durch `PI` und Multiplikation mit 180.)

Weitere Informationen zur Verwendung von Schlüsselwörtern in Unica Campaign finden Sie in Formatspezifikationen (auf Seite 12).

Syntax

`COT` berechnet den Kotangens der Werte im angegebenen Datenbereich. Der Kotangens ist der Kehrwert des Tangens. `COT` gibt für jede Eingabespalte eine neue Spalte zurück, die jeweils den Kotangens der Zahlen in der entsprechenden Eingabespalte enthält.

Anmerkung: Wenn eine Zelle einen Wert enthält, dessen Tangens null ist, ist der Kotangens unendlich. In diesem Fall gibt `COT` die größte 32-Bit-Gleitkommazahl zurück.

Beispiele

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

Gibt eine neue Spalte `TEMP` zurück, die den Wert `-0.5` enthält.

```
TEMP = COT(0)
```

Gibt eine neue Spalte `TEMP` zurück, die den Wert `MAX_FLOAT_32` enthält.

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

Erstellt eine neue Spalte `TEMP`, in der jeder Wert den Kotangens des Inhalts (in Grad) der Spalte `V1` darstellt.

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

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Werte in der Spalte `TEMP` sind der Kotangens des Inhalts von Spalte `V1`, die Werte von Spalte `VX` sind der Kotangens des Inhalts von Spalte `V2` und die Werte von Spalte `VY` sind der Kotangens des Inhalts von Spalte `V3`. Alle Werte sind in Grad angegeben.

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

Erstellt eine neue Spalte `TEMP`, in der die ersten 11 Zellen jeweils den Kotangens der Werte in Zeile 10-20 von Spalte `V1` enthalten. (in Radianten). Die anderen Zellen in `TEMP` sind leer.

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

Erstellt zwei neue Spalten TEMP und VX, die jeweils Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind der Kotangens der entsprechenden Zeilen von Spalte V1, die Werte in der Spalte VX sind der Kotangens der entsprechenden Zeilen von Spalte V2. Alle Werte sind in Radiant angegeben.

Zugehörige Funktionen

Funktion	Syntax
ACOT	Berechnet den Arkuskotangens des Inhalts des angegebenen Datenbereichs.
COS	Berechnet den Kosinus des Inhalts des angegebenen Datenbereichs.
SIN	Berechnet den Sinus des Inhalts des angegebenen Datenbereichs.
TAN	Berechnet den Tangens des Inhalts des angegebenen Datenbereichs.

COUNT-Makro

Die COUNT Makro ist nur in Unica Campaign verfügbar.

Syntax

```
COUNT(data)
```

Parameter

`data`

Der Zellenbereich, in dem die Anzahl der Zellen gezählt werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

COUNT berechnet die Anzahl der Werte im angegebenen Datenbereich. COUNT gibt eine neue Spalte mit einem Einzelwert zurück, der die Anzahl der Zellen, die Werte enthalten, innerhalb des angegebenen Datenbereichs darstellt.

Anmerkung: Bei einer leeren Spalte wird null zurückgegeben.

Beispiele

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

Erstellt eine neue Spalte `TEMP`, die den Einzelwert eins enthält (die Funktion `AVG` gibt im Standardmodus eine einzelne Zelle zurück).

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

Erstellt eine neue Spalte `TEMP`, die einen Einzelwert enthält, der die Anzahl der Zellen mit Werten in Spalte `V1` angibt.

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

Erstellt eine neue Spalte `TEMP`, die einen Einzelwert enthält, der die Anzahl der Zellen mit Werten in den Spalten `V1`, `V2` und `V3`angibt.

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

Erstellt eine neue Spalte `TEMP`, die den Wert `11` enthält (Bereiche schließen die Grenzen ein), sofern sämtliche Zellen Werte enthalten.

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

Erstellt eine neue Spalte `TEMP`, die den Wert `20` enthält (5 Zellen in jeder Spalte bei 4 Spalten ergeben 20 Zellen), sofern sämtliche Zellen Werte enthalten.

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

Erstellt eine neue Spalte `TEMP`, die den Wert `3` enthält, sofern die Zeilen 1-3 von Spalte `V1` Werte enthalten und die Zeilen 4-10 leer sind.

Zugehörige Funktionen

Funktion	Syntax
`SUM` oder `TOTAL`	Berechnet die Summe eines Zellenbereichs.

CURRENT_DATE-Makro

Die `CURRENT_DATE` Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

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

Parameter

`format`

Eines der Schlüsselwörter in der folgenden Tabelle, das das Datumsformat von `date_string` angibt.

Anmerkung: Weitere Informationen zu den verfügbaren Datumsformaten finden Sie unter "Gültige Formatschlüsselwörter".

Syntax

`CURRENT_DATE` Gibt das aktuelle Datum in `format` zurück. Das Datum wird durch die Systemzeit auf dem Server festgelegt. Wenn kein Schlüsselwort `format` angegeben wird, wird standardmäßig das Format `DELIM_M_D_Y` verwendet.

Bei allen empfohlenen Datenbanken versucht , das Unica CampaignCURRENT_DATE mithilfe eines von der Datenbank unterstützten SQL-Aufrufs für die aktuelle Uhrzeit (z. B. SYSDATE, GETDATE, DATE oder TODAY) in der Datenbank auszuführen. In diesen Fällen werden von dieser Makrofunktion alle Parameter (einschließlich des Datumsformats) ignoriert und die Ausgabe schließt alle Angaben ein, die von der Datenbank zurückgegeben werden (unter Umständen kann die Ausgabe also eine Zeitkomponente enthalten).
Falls dies der Fall ist und nur das Datum oder das Datum in einem anderen Format zurückgegeben werden soll, können Sie mit direktem SQL ein benutzerdefiniertes Makro schreiben oder Makros verwenden. Beispiel:

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

In einigen Fällen wird auf dem Unica Campaign-Server das Makro CURRENT_DATE() ausgeführt (nämlich wenn eine unstrukturierte Datei oder eine nicht empfohlene Datenbank ohne entsprechende SQL-Unterstützung vorliegt oder wenn der Campaign-Makroausdruck in der Datenbank nicht aufgelöst werden kann). In diesen Fällen werden alle Parameter erkannt und die Ausgabe wird im ausgewählten Format zurückgegeben.

Anmerkung: Nicht alle Formate, die in Unica Campaign verfügbar sind, werden von Unica Interactunterstützt.

Beachten Sie, dass Sie möglicherweise das Makro DATE_FORMAT verwenden müssen, um das CURRENT_DATE für Ihren Datenbanktyp anzupassen. Zum Beispiel kann bei DB2 das folgende Makro verwendet werden:

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

Bei Oracle müssen Sie das Makro DATE_FORMAT jedoch wie folgt verwenden:

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

Beispiele

Wenn das aktuelle Datum der 13. September 2015 ist, `CURRENT_DATE()` gibt "09/13/15" zurück.

Zugehörige Funktionen

Funktion	Syntax
`DATE_FORMAT`	Konvertiert Datumsangaben von einem Format in ein anderes.
`DATE_JULIAN`	Gibt das julianische Datum der Eingabe zurück.
`DATE_STRING`	Gibt die Datumszeichenfolge des julianischen Datums zurück.
`DATE`	Konvertiert eine Datumszeichenfolge in ein julianisches Datum.

CURRENT_DAY-Makro

Die `CURRENT_DAY` Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

`CURRENT_DAY()`

Syntax

`CURRENT_DAY` gibt den aktuellen Tag des Monats als Zahl zwischen 1 und 31 zurück. Das Datum wird durch die Systemzeit auf dem Server festgelegt.

Beispiele

Wenn das aktuelle Datum der 19. Juni ist, gibt CURRENT_DAY() die Zahl 19 zurück.

Zugehörige Funktionen

Funktion	Syntax
CURRENT_JULIAN	Gibt die julianische Zahl für das aktuelle Datum zurück.
CURRENT_MONTH	Gibt den aktuellen Monat als Zahl zurück.
CURRENT_TIME	Gibt die aktuelle Uhrzeit als Zeichenfolge zurück.
CURRENT_WEEKDAY	Gibt den aktuellen Wochentag als Zahl zurück.
CURRENT_YEAR	Gibt das aktuelle Jahr als Zahl zurück.

Syntax

Die CURRENT_JULIAN Makro ist nur in Unica Campaign verfügbar.

Syntax

```
CURRENT_JULIAN()
```

Syntax

CURRENT_JULIAN() gibt die julianische Zahl für das aktuelle Datum (die Anzahl der seit dem 1. Januar 0000 vergangenen Tage) zurück. Dies ist äquivalent zu dem Makro DATE(CURRENT_DATE()).

Beispiele

Wenn das aktuelle Datum der 31. August 2000 ist, gibt CURRENT_JULIAN() die Zahl 730729 zurück.

Zugehörige Funktionen

Funktion	Syntax
CURRENT_DAY	Gibt den aktuellen Tag als Zahl zurück.
CURRENT_MONTH	Gibt den aktuellen Monat als Zahl zurück.
CURRENT_TIME	Gibt die aktuelle Uhrzeit als Zeichenfolge zurück.

Funktion	Syntax
CURRENT_WEEKDAY	Gibt den aktuellen Wochentag als Zahl zurück.
CURRENT_YEAR	Gibt das aktuelle Jahr als Zahl zurück.

CURRENT_MONTH-Makro

Die CURRENT_MONTH Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

```
CURRENT_MONTH()
```

Syntax

CURRENT_MONTH gibt den aktuellen Tag des Monats als Zahl zwischen 1 und 12 zurück.

Beispiele

Wenn das aktuelle Datum der 19. Juni ist, gibt CURRENT_MONTH() die Zahl 6 zurück.

Zugehörige Funktionen

Funktion	Syntax
CURRENT_DAY	Gibt den aktuellen Tag als Zahl zurück.
CURRENT_JULIAN	Gibt die aktuelle julianische Zahl zurück.
CURRENT_TIME	Gibt die aktuelle Uhrzeit als Zeichenfolge zurück.
CURRENT_WEEKDAY	Gibt den aktuellen Wochentag als Zahl zurück.
CURRENT_YEAR	Gibt das aktuelle Jahr als Zahl zurück.

CURRENT_TIME-Makro

Die CURRENT_TIME Makro ist nur in Unica Campaign verfügbar.

Syntax

```
CURRENT_TIME()
```

Syntax

`CURRENT_TIME` Gibt die aktuelle Uhrzeit als Zeichenfolge zurück. Die Uhrzeit wird durch die Systemzeit auf dem Server festgelegt.

Datumseinstellung in Ihrer Webanwendung

Damit Datumsangaben in Ihrer Webanwendung bei aktuellen Versionen von Unica Campaign korrekt angezeigt werden, muss die Konfigurationsdatei Ihres Back-End-Servers ordnungsgemäß konfiguriert werden. Dies ist besonders wichtig bei den Parametern `dDateFormat` und `DateOutputFormatString` für die Datenbank, die die Systemtabellen enthält. Wenn diese nicht ordnungsgemäß konfiguriert sind, werden Datumsangaben auch in Campaign falsch angezeigt. Diese Eigenschaften können mithilfe von Platform konfiguriert werden.

Sprachspezifische Datumsangaben in Ihrer Webanwendung

Anmerkung: Alle referenzierten Dateien werden vom Installationsprogramm der Webanwendung installiert, sofern nicht anders angegeben.

Wichtig: `webapphome` bezieht sich auf das Verzeichnis, in dem die Webanwendung von Campaign installiert wurde. `language_code` bezieht sich auf die Spracheinstellung(en), die Sie für Ihr System wählen.

1. Bearbeiten Sie die Datei `webapphome/conf/campaign_config.xml`, um sicherzustellen, dass `language_code` in der durch Kommas getrennten Liste im Tag `<supportedLocales>` enthalten ist, wie im Folgenden gezeigt:

   ```
   <supportedLocales>en_US, language_code</supportedLocales>
   ```
2. Kopieren Sie im Verzeichnis `webapphome/webapp` die gesamte Verzeichnisstruktur `en_US to language_code` (Groß-/Kleinschreibung muss beachtet werden).
3. In `webapphome/webapp/WEB-INF/classes/resources`, kopieren Sie `StaticMessages_en_US.properties` auf `StaticMessages_ language_code.properties`. Kopieren Sie auch `ErrorMessages_en_US.properties` auf `ErrorMessages_ language_code.properties`.

4. Bearbeiten Sie die Eigenschaften StaticMessages_ language_code.properties: Suchen Sie nach DatePattern und ändern Sie den Eintrag in DatePattern=dd/MM/yyyy (Groß-/Kleinschreibung muss beachtet werden).

   Anmerkung: Dieses Format wird von Java™ definiert. Vollständige Informationen zu dem Format finden Sie in der Java für `java.text.SimpleDateFormat` unter http://java.sun.com. Die Datei `StaticMessages.properties` braucht nicht geändert zu werden.
5. Für WebSphere®: Rufen Sie die Webanwendung erneut auf.
6. Bei WebLogic: Entfernen Sie das aktuelle Modul der Webanwendung.
   a. Fügen Sie das neue Modul hinzu.
   b. Führen Sie die Bereitstellung der Webanwendung erneut aus.
   c. Ein Neustart des Unica Campaign-Listeners ist nicht erforderlich.
7. Stellen Sie sicher, dass `language_code` in der Spracheinstellung des Web-Browsers die höchste Priorität hat. Weitere Informationen finden Sie in den folgenden Abschnitten, die die korrekte Spracheinstellung für Ihren Web-Browser und die Anzeige einer bestimmten Sprache auf Ihrem Computer behandeln.

   Anmerkung: Achten Sie darauf, in `language_code` einen Bindestrich und keinen Unterstrich zu verwenden. Die Konfiguration der Webanwendung ist der einzige Ort, an dem anstelle eines Unterstrichs ein Bindestrich verwendet wird.
8. Melden Sie sich bei Campaign an. Datumsangaben sollten in Campaign in dem Format angezeigt werden, das Sie in `StaticMessages_language_code.properties` angegeben haben.

Informationen zur Konfiguration der Uhrzeit für Unica Campaign finden Sie in der Dokumentation zu Unica Campaign.

Beispiele

Wenn die Uhrzeit 10:54 ist, gibt `CURRENT_TIME()` die Zeichenfolge "10:54:00" zurück.

Zugehörige Funktionen

Funktion	Syntax
CURRENT_DAY	Gibt den aktuellen Tag als Zahl zurück.
CURRENT_JULIAN	Gibt die aktuelle julianische Zahl zurück.
CURRENT_WEEKDAY	Gibt den aktuellen Wochentag als Zahl zurück.
CURRENT_YEAR	Gibt das aktuelle Jahr als Zahl zurück.

CURRENT_WEEKDAY-Makro

Die CURRENT_WEEKDAY Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

```
CURRENT_WEEKDAY()
```

Syntax

CURRENT_WEEKDAY gibt den aktuellen Wochentag als Zahl zwischen 0 und 6 zurück. Der Sonntag wird als 0 dargestellt, der Montag als 1 usw.

Beispiele

Wenn der aktuelle Wochentag ein Freitag ist, gibt CURRENT_WEEKDAY() die Zahl 5 zurück.

Zugehörige Funktionen

Funktion	Syntax
CURRENT_DAY	Gibt den aktuellen Tag als Zahl zurück.
CURRENT_JULIAN	Gibt die aktuelle julianische Zahl zurück.
CURRENT_MONTH	Gibt den aktuellen Monat als Zahl zurück.
CURRENT_TIME	Gibt die aktuelle Uhrzeit als Zeichenfolge zurück.
CURRENT_YEAR	Gibt das aktuelle Jahr als Zahl zurück.

CURRENT_YEAR-Makro

Die CURRENT_YEAR Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

```
CURRENT_YEAR()
```

Syntax

CURRENT_YEAR Gibt das aktuelle Jahr als Zahl zurück.

Beispiele

Wenn das aktuelle Jahr 2000 ist, gibt CURRENT_YEAR() die Zahl zurück: 2000

Zugehörige Funktionen

Funktion	Syntax
CURRENT_DAY	Gibt den aktuellen Tag als Zahl zurück.
CURRENT_JULIAN	Gibt die aktuelle julianische Zahl zurück.
CURRENT_MONTH	Gibt den aktuellen Monat als Zahl zurück.
CURRENT_TIME	Gibt die aktuelle Uhrzeit als Zeichenfolge zurück.
CURRENT_WEEKDAY	Gibt den aktuellen Wochentag als Zahl zurück.
MONTHOF	Gibt den Monat des Jahres als Zahl zurück.
WEEKDAYOF	Gibt den Wochentag der Woche als Zahl zurück.
YEAROF	Gibt das Jahr als Zahl zurück.

DATUM

Syntax

```
DATE(input_date, [input_date format])
```

Parameter

```
date_string
```

Eine Textzeichenfolge, die ein gültiges Datum darstellt.

`format`

Eines der Schlüsselwörter in der Tabelle unter "Gültige Schlüsselwörter für Datumsformat", das das Datumsformat von `date_string` (optional).

Syntax

Das -Makro Unica Interact `DATE` konvertiert ein eingegebenes Datum in einen formatneutralen Ganzzahlwert.

Das Makro `DATE` wird wie folgt berechnet: DATE (X) = **365 +** die Anzahl der vollen Tagen, die seit dem 1. Januar, 0001 n. Chr., 12:00 Uhr vergangen sind. Für DATE (X) kann ein optionales Schlüsselwort für das DATE-Eingabeformat bereitgestellt werden, um anzugeben, wie das eingegebene `DATE` geparst werden soll. Wenn kein Schlüsselwort `format` angegeben wird, wird standardmäßig das Format `DELIM_M_D_Y` verwendet. Weitere Informationen zu gültigen Datumsformaten finden Sie unter Gültige Schlüsselwörter für Datumsformat (auf Seite 22).

Datumsformate weisen entweder eine feste Breite (dann wird der 28. Februar 1970 im Format MMDDYYYY als 02281970 dargestellt) oder einen Begrenzer auf (dann kann der 28. Februar 1970 als 2-28-1970 oder 02/28/1970 im Format `DELIM_M_D_YY` dargestellt werden).

In begrenzten Formaten kann der Begrenzer ein Schrägstrich (/), ein Strich (-), ein Leerzeichen ( ), ein Komma (,) oder ein Doppelpunkt (:) sein; Jahre können 2- oder 4-stellig dargestellt werden; Monate können ausgeschrieben (z. B. Februar), abgekürzt (z. B. Feb) oder numerisch dargestellt werden (z. B. 2 oder 02).

Für alle zweistellig angegebenen Jahre gilt Folgendes:

- Unica Interact setzt standardmäßig voraus, dass die Datumsangaben in den Jahren 1920 bis 2020 begrenzt zweistellig angegeben werden
- Zweistellige Jahresangaben, die kleiner als der Grenzwert für das Millennium sind (standardmäßig 20, kann jedoch vom JVM-Parameter festgelegt werden) werden dem 21. Jahrhundert zugerechnet.
- Zweistellige Jahresangaben, die größer-gleich dem Grenzwert sind, werden dem 20. Jahrhundert zugerechnet.

Anmerkung:

- Nicht alle in `DATE Formats` verfügbaren Unica Campaign werden von Unica Interact unterstützt.
- Weitere Informationen zu zweistelligen Jahren finden Sie unter Gültige Schlüsselwörter für Datumsformat.
- Weitere Informationen zur Konfiguration des Grenzwerts für das Millennium bei zweistelligen Jahresangaben finden Sie im Abschnitt "JVM-Argumente" im Handbuch zur Optimierung von Unica Interact.

Dieses Makro ist in Unica Interact verfügbar.

In vielen Geschäftssystemen werden Grenzwerte des Julianischen Datums verwendet. Das Ergebnis des Unica Interact-Makros DATE() steht wie folgt in Bezug zum Julianischen Datum:

Julianisches Datum = DATE(...) + 1.721.059 + Bruchteil des Tages, der seit 12:00 Uhr des Vortages vergangen ist.

Zu den DATE()-Nutzungswerten für den Zeitraum n. Chr. gehören:

- Januar `1, 2050 A.D` gibt `748,749` zurück.
- Januar `1, 2000 A.D` gibt `730,486` zurück
- Januar `1, 1990 A.D` gibt `726,834` zurück
- Januar `1, 1900 A.D` gibt `693,962` zurück
- Januar `1, 0001 A.D` gibt `365` zurück

Anmerkung: Nach der Standard- und XML-Schemadefinition für die Objekte "Datum" und "Datum/Uhrzeit" aus der ISO 8601 wird der proleptische gregorianische Kalender für die Berechnung der Anzahl der vergangenen Tage verwendet. In diesem Kalendersystem ist das hypothetische Jahr 0000 n. Chr. gleichbedeutend mit 0001 v. Chr.

Beispiele

`DATE("8/31/2000")` gibt die Zahl `730,729` zurück.

`DATE("8/31/2000",DELIM_MM_DD_YYYY)` gibt die Zahl `730,729` zurück.

`DATE("2015-01-01",DELIM_Y_M_D)` gibt die Zahl `735,965` zurück.

`DATE("01",DD), DATE("0101",MMDDD)` und `DATE("1970-01-01",DELIM_Y_M_D)` geben die Zahl `719,529` zurück.

Zugehörige Funktionen

Funktion	Syntax
DATE_FORMAT	Konvertiert Datumsangaben von einem Format in ein anderes.
DATE_JULIAN	Gibt das julianische Datum der Eingabe zurück.
DATE_STRING	Gibt die Datumszeichenfolge des julianischen Datums zurück.
CURRENT_DATE	Gibt das aktuelle Datum in einem bestimmten Format zurück.

DATE_FORMAT-Makro

Die `DATE_FORMAT` Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

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

Parameter

`date_string`

Ein Text, der ein gültiges Datum darstellt.

`input_format`

Eines der Schlüsselwörter in der folgenden Tabelle, das das Datumsformat von `date_string` angibt.

`output_format`

Eines der Schlüsselwörter in der folgenden Tabelle, das das vorgesehene Ausgabedatumsformat angibt.

Syntax

`DATE_FORMAT()`Wandelt ein Datum von `input_format` in `output_format` um.

Wenn das Datum eine feste Breite aufweist, muss dem Format einer der folgenden Werte zugewiesen werden:

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

MM ist eine aus 2 Ziffern, MMM eine aus 3 Buchstaben bestehende Abkürzung des Monats.

Wenn das Datum Begrenzer aufweist (als Begrenzer können Leerzeichen, Gedankenstriche und Schrägstriche verwendet werden), muss dem Format einer der folgenden Werte zugewiesen werden:

- DELIM_D_M_Y
- DELIM_M_D_Y
- DELIM_Y_M_D

Anmerkung: Nicht alle Formate, die in Unica Campaign verfügbar sind, werden von Unica Interactunterstützt.

Beispiele

`DATE_FORMAT("012171", MMDDYY, MMDDYYYY)` gibt die Zeichenfolge "01211971" zurück.

Anmerkung: Weitere Informationen zu gültigen Datumsformaten finden Sie unter DATUM *(auf Seite 71)*.

Zugehörige Funktionen

Funktion	Syntax
`DATE`	Konvertiert eine Datumszeichenfolge in ein julianisches Datum.
`DATE_JULIAN`	Gibt das julianische Datum der Eingabe zurück.
`DATE_STRING`	Gibt die Datumszeichenfolge des julianischen Datums zurück.

DATE_JULIAN-Makro

Die DATE_JULIAN Makro ist nur in Unica Campaign verfügbar.

Syntax

```
DATE_JULIAN(year, month, day)
```

Parameter

year

Gültige 2-stellige oder 4-stellige Jahreszahl.

month

Gültige Monatszahl zwischen 1 und 12.

day

Gültige Tageszahl zwischen 1 und 31.

Syntax

DATE_JULIAN gibt das julianische Datum der Eingabe zurück. Das julianische Datum ist die Anzahl der seit dem 1. Januar 0000 vergangenen Tage.

Beispiele

DATE_JULIAN (2000,08,31) gibt die Zahl 730729 zurück.

Zugehörige Funktionen

Funktion	Syntax
DATE	Konvertiert eine Datumszeichenfolge in ein julianisches Datum.
DATE_FORMAT	Konvertiert Datumsangaben von einem Format in ein anderes.
DATE_STRING	Gibt die Datumszeichenfolge des julianischen Datums zurück.

DATE_STRING-Makro

Die DATE_STRING Makro ist nur in Unica Campaign verfügbar.

Syntax

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

Parameter

`julian_date`

Eine Zahl, die ein Julianisches Datum darstellt, d. h. die Anzahl der seit dem 1. Januar 0000 vergangenen Tage.

`output_format`

Zeichenfolge, gültiges Datumsformat.

`max_length`

`format_string`

Eine Formatierzeichenfolge, die optional eine beliebige Kombination der folgenden Formatcodes enthalten kann:

Code	Syntax
%a	Abgekürzter Wochentagsname
%A	Vollständiger Wochentagsname
%b	Abgekürzter Monatsname
%B	Vollständiger Monatsname
%c	Datums- und Uhrzeitdarstellung gemäß der Ländereinstellung
%d	Tag des Monats (01 - 31)
%H	Stunde im 24-Stunden-Format (00 - 23)
%I	Stunde im 12-Stunden-Format (01 - 12)
%j	Tag des Jahres (001 - 366)
%m	Monat (01 - 12)
%M	Minute (00 - 59)
%p	Current® AM/PM-Anzeiger für 12-Stunden-Format gemäß der Ländereinstellung
%S	Sekunde (00 - 59)
%U	Woche des Jahres, wobei Sonntag als erster Wochentag zählt (00 - 51)
%w	Wochentag (0 - 6; Sonntag ist 0)

Code	Syntax
%W	Woche des Jahres, wobei Montag als erster Wochentag zählt (00 - 51)
%x	Datumsdarstellung gemäß der aktuellen Ländereinstellung
%X	Uhrzeitdarstellung gemäß der aktuellen Ländereinstellung
%y	2-stelliges Jahr (00 - 99)
%Y	4-stelliges Jahr. Die führenden Nullen im Jahr werden nicht abgeschnitten. Beispiel: Das Jahr 0201 wird als 0201 angezeigt und das Jahr 0001 als 0001.
%4Y	4-stelliges Jahr. Die führenden Nullen im Jahr werden nicht abgeschnitten. Beispiel: Das Jahr 0201 wird als 0201 angezeigt und das Jahr 0001 als 0001.
%z, %Z	Name oder Abkürzung der Zeitzone; keine Ausgabe, falls Zeitzone unbekannt ist
%%	Prozentzeichen

Syntax

DATE_STRING gibt die Datumszeichenfolge des julianischen Datums zurück. Wenn output_format nicht angegeben ist, wird das Standardschlüsselwort DELIM_M_D_Y verwendet.

Beispiele

DATE_STRING(730729) gibt die Zeichenfolge "08/31/00" zurück.

Anmerkung: Weitere Informationen zu gültigen Datumsformaten finden Sie unter DATUM *(auf Seite 71)*.

Zugehörige Funktionen

Funktion	Syntax
DATE	Konvertiert eine Datumszeichenfolge in ein julianisches Datum.
DATE_JULIAN	Gibt das julianische Datum der Eingabe zurück.
DATE_FORMAT	Konvertiert Datumsangaben von einem Format in ein anderes.

DAY_BETWEEN-Makro

Die `DAY_BETWEEN` Makro ist nur in Unica Campaign verfügbar.

Syntax

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

Parameter

`from_date_string`

Ein Text, der ein gültiges Datum darstellt, von dem aus die Anzahl der verstrichenen Tage berechnet werden soll.

`to_date_string`

Ein Text, der ein gültiges Datum darstellt, bis zu dem die Anzahl der Tage berechnet werden soll. Dieses Datum muss im selben Format wie `from_date_string` sein.

`input_format`

Eines der Schlüsselwörter in der folgenden Tabelle, das das Datumsformat von `from_date_string` und `to_date_string` angibt.

Syntax

`DAY_BETWEEN` gibt die Anzahl der Tage zwischen `from_date_string` und `to_date_string` zurück. Wenn `input_format` nicht angegeben ist, wird das Standardschlüsselwort `DELIM_M_D_Y` verwendet.

Beispiele

`DAY_BETWEEN("08/25/00","08/31/00")` gibt die Zahl 6 zurück.

Anmerkung: Weitere Informationen zu gültigen Datumsformaten finden Sie unter DATUM *(auf Seite 71)*.

Zugehörige Funktionen

Funktion Syntax

`DAY_FROMNOW` Gibt die Anzahl der Tage vom aktuellen Datum bis zu einem angegebenen Datum zurück.

`DAY_INTERVAL` Gibt die Anzahl der Tage zwischen zwei angegebenen Terminen zurück.

DAY_FROMNOW-Makro

Die `DAY_FROMNOW` Makro ist nur in Unica Campaign verfügbar.

Syntax

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

Parameter

`to_year`

Gültige 2-stellige oder 4-stellige Jahreszahl.

`to_month`

Gültige Monatszahl zwischen 1 und 12.

`to_day`

Gültige Tageszahl zwischen 1 und 31.

Syntax

`DAY_FROMNOW` gibt die Anzahl der Tage vom aktuellen Datum bis zu einem angegebenen Datum zurück. `to_year/to_month/to_day.`

Anmerkung: Wenn das angegebene Datum in der Vergangenheit liegt, ist der Rückgabewert negativ.

Beispiele

Wenn das aktuelle Datum der 31. August 2000 ist, gibt `DAY_FROMNOW(2000,12,31)` die Zahl 122 zurück.

Zugehörige Funktionen

Funktion	Syntax
DAY_BETWEEN	Gibt die Anzahl der Tage zwischen zwei angegebenen Datumszeichenfolgen zurück.
DAY_INTERVAL	Gibt die Anzahl der Tage zwischen zwei angegebenen Terminen zurück.

DAY_INTERVAL-Makro

Die DAY_INTERVAL Makro ist nur in Unica Campaign verfügbar.

Syntax

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

Parameter

`from_year`

Gültige 2-stellige oder 4-stellige Jahreszahl.

`from_month`

Gültige Monatszahl zwischen 1 und 12.

`from_day`

Gültige Tageszahl zwischen 1 und 31.

`to_year`

Gültige 2-stellige oder 4-stellige Jahreszahl.

`to_month`

Gültige Monatszahl zwischen 1 und 12.

`to_day`

Gültige Tageszahl zwischen 1 und 31.

Syntax

DAY_INTERVAL gibt die Anzahl der Tage zwischen dem angegebenen Von-Datum from_year/from_month/from_day) und dem angegebenen Bis-Datum (to_year/to_month/to_day) zurück.

Beispiele

DAY_INTERVAL(2000,8,31,2000,12,31) gibt die Zahl 122 zurück.

Zugehörige Funktionen

Funktion	Syntax
DAY_BETWEEN	Gibt die Anzahl der Tage zwischen zwei angegebenen Datumszeichenfolgen zurück.
DAY_FROMNOW	Gibt die Anzahl der Tage vom aktuellen Datum bis zu einem angegebenen Datum zurück.

DAYOF-Makro

Die DAYOF Makro ist nur in Unica Campaign verfügbar.

Syntax

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

Parameter

date_string

Ein Text, der ein gültiges Datum darstellt.

input_format

Eines der Schlüsselwörter in der folgenden Tabelle, das das Datumsformat von date_string angibt.

Syntax

`DAYOF` gibt den Tag des Monats als Zahl für das durch das `date_string`dargestellte Datum zurück. Wenn `input_format` nicht angegeben ist, wird das Standardschlüsselwort `DELIM_M_D_Y` verwendet.

Beispiele

`DAYOF("08/31/00")` gibt die Zahl 31 zurück.

Anmerkung: Weitere Informationen zu gültigen Datumsformaten finden Sie unter DATUM *(auf Seite 71)*.

DISTANCE-Makro

Die `DISTANCE` Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

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

Parameter

`lat1`

Der Breitengrad des ersten Punkts als Dezimalwert.

`long1`

Der Längengrad des ersten Punkts als Dezimalwert.

`lat2`

Der Breitengrad des zweiten Punkts als Dezimalwert.

`long2`

Der Längengrad des zweiten Punkts als Dezimalwert.

`UNIT_OF_MEASURE`

Ein optionaler Parameter, der die Maßeinheit für den zurückgegebenen Abstand angibt. Mögliche Werte sind MILES (Meilen) oder KILOMETERS (Kilometer). Wenn Sie diesen Parameter nicht angeben, wird als Standardeinstellung MILES verwendet.

`PRECISION`

Ein optionaler Parameter, der die Genauigkeit nach dem Dezimalzeichen für den zurückgegebenen Abstand angibt. Wenn Sie einen Genauigkeitswert festlegen, wird der zurückgegebene Abstand auf die angegebene Anzahl von Dezimalstellen gekürzt. Der Maximalwert ist 5. Wenn Sie hierfür keinen Wert angeben, wird die Anzahl der Dezimalstellen nicht gekürzt.

Beschreibung

`DISTANCE` berechnet den Abstand zwischen zwei Punkten. Breitengrad und Längengrad werden in Dezimalzahleinheiten erwartet. Verwenden Sie zum Trennen numerischer Werte immer ein Komma und ein Leerzeichen. Das ist notwendig, um Sprachen zu berücksichtigen, in denen ein Komma als Dezimaltrennzeichen verwendet wird, wie im zweiten Beispiel unten gezeigt wird. Es wird unterstützt, den Abstand zwischen mehreren Punkten zu berechnen. Wenn (lat1, long1) eine Liste mit mehreren Werten ist und (lat2, long2) ebenfalls eine Liste mit mehreren Werten ist, wird der Abstand des ersten Punkts in Liste 1 und des ersten Punkts in Liste 2 berechnet und als erstes Element in der Ergebnisliste zurückgegeben, der Abstand des zweiten Punkts in Liste 1 und des zweiten Punkts in Liste 2 wird berechnet und als zweites Element in der Ergebnisliste zurückgegeben, und so weiter, bis alle Elemente in Liste 1 und Liste 2 berechnet wurden. Wenn Liste 1 nur ein Element enthält und Liste 2 mehrere Elemente enthält, werden die Entfernungen zwischen dem Element in Liste 1 und allen Elementen in Liste 2 berechnet.

Beispiele

`DISTANCE (18.529747, 73.839798, 18.533511, 73.8777995, MILES, 2)` gibt den Wert 2,50 Meilen zurück.

`DISTANCE (18,529747, 73,839798, 18,533511, 73,8777995, KILOMETERS, 1)` gibt den Wert 4,0 Kilometer zurück.

DIV-Makro

Die `DIV` Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

```
data DIV divisor data / divisor
```

Parameter

`data`

Die numerischen Werte, die dividiert werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`divisor`

Der Wert oder die Werte, durch den oder die die Werte im angegebenen Datenbereich dividiert werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in `data2` muss mit der Anzahl der Spalten in `data1` übereinstimmen, es sei denn, bei `data2` handelt es sich um eine Konstante. Informationen zur Formatdefinition von `divisor` (dasselbe wie bei `data`), siehe den Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`DIV` dividiert den angegebenen Datenbereich durch den Divisorwert. Er gibt für jede Eingabespalte eine neue Spalte zurück, die jeweils die entsprechende Spalte in `data1` dividiert durch die entsprechende Spalte von `data2` enthält (d.h. die erste Spalte von `data1` wird durch zur ersten Spalte von `data` dividiert, die zweite Spalte mit der zweiten Spalte usw.)

Wenn `data2` eine Konstante ist, wird jeder Wert in `data1` durch diesen Wert dividiert. Wenn `data2` eine oder mehrere Spalten enthält, werden die Berechnungen auf Zeilenbasis für eine

Spalte aus `data2` und eine Spalte aus `data2` durchgeführt. Die erste Zeile von `data1` wird durch den Wert der ersten Zeile von `data2` dividiert, die zweite Zeile durch die zweite Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

Anmerkung: Wenn eine Spalte in jeder Zeile dieselbe Zahl *x* wie `divisor` enthält, ist dies dasselbe, als wenn als `divisor` die Konstante *x* verwendet wird.

Anmerkung: Der Operator `DIV` kann durch einen Schrägstrich (`/`) abgekürzt werden.

Beispiele

`TEMP = 8 DIV 4` oder `TEMP = 8/4`

Erstellt eine neue Spalte `TEMP`, die den Wert zwei enthält.

```
TEMP = V1/8
```

Erstellt eine neue Spalte `TEMP`, in der jeder Wert der Inhalt der Spalte `V1` dividiert durch acht ist.

```
TEMP =V1:V3/2
```

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Werte in der Spalte `TEMP` entsprechen dem Inhalt der Spalte `V1` geteilt durch zwei, die Werte der Spalte `VX` entsprechen dem Inhalt der Spalte `V2` geteilt durch zwei und die Werte der Spalte `VY` entsprechen dem Inhalt der Spalte `V3` dividiert durch zwei.

```
TEMP = V1/V1
```

Erstellt eine neue Spalte `TEMP`, die nur Einsen enthält (jede Zahl ist gleich sich selbst).

```
TEMP = V1/V2
```

Erstellt eine neue Spalte `TEMP`, wobei jeder Zeilenwert von Spalte `V1` durch den entsprechenden Zeilenwert von Spalte `V2` dividiert ist.

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

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. In der Spalte `TEMP` sind die Werte von Spalte `V1` durch die entsprechenden Zeilenwerte von Spalte `V4` dividiert. Die Spalte `VX` enthält die Division von Spalte `V2` durch Spalte `V5`. Die Spalte `VY` enthält die Division von Spalte `V3` durch Spalte `V6`.

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

Erstellt eine neue Spalte `TEMP`, in der die ersten 11 Zellen das Ergebnis der Division der Werte in den Zeilen 10-20 der Spalte `V1` durch die Werte in den Zeilen 1-11 der Spalte `V2` enthalten. Die anderen Zellen in `TEMP` sind leer.

Zugehörige Funktionen

Funktion	Syntax
`MOD`	Berechnet den Modulo-Wert des Inhalts des angegebenen Datenbereichs.
`MULT`	Multipliziert die Inhalte zweier Datenbereiche.
`POW`	Erhebt einen Basiswert in die Potenz des angegebenen Exponenten.

EQ-Makro

Die `EQ` Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

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

Parameter

`data1`

Der Zellenbereich, der verglichen werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`data2`

Die Zahlen, mit denen alle Werte in der angegebenen Spalte verglichen werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in `data2` muss mit der Anzahl der Spalten in `data1` übereinstimmen, es sei denn, bei `data2` handelt es sich um eine Konstante. Informationen zur Formatdefinition von `data` finden Sie

im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`EQ` vergleicht die beiden angegebenen Datenbereiche und gibt eine Eins zurück, wenn die Werte ungleich sind, bzw. eine Null, wenn die Werte gleich sind. Sie gibt für jede Eingabespalte eine neue Spalte zurück, die jeweils die entsprechende Spalte in `data1` im Vergleich zur entsprechenden Spalte von `data2` enthält (d. h. die erste Spalte von `data1` wird mit der ersten Spalte von `data` verglichen, die zweite Spalte mit der zweiten Spalte usw.).

Wenn es sich bei `data2` um eine Konstante handelt, wird jeder Wert in `data` mit dem betreffenden Wert verglichen. Wenn es sich bei `data2` um eine Spalte handelt, werden die Berechnungen zeilenweise ausgeführt. Die erste Zeile von `data1` wird durch den Wert der ersten Zeile von `data2` dividiert, die zweite Zeile durch die zweite Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

Beim Vergleich von Zeichenfolgen spielt die Groß-/Kleinschreibung keine Rolle (d. h., "Ja", "ja", "JA" und "jA" werden gleich behandelt).

Anmerkung: Der Operator `EQ` kann mit einem doppelten Gleichheitszeichen (==) abgekürzt werden. In Klammern kann auch ein einzelnes Gleichheitszeichen (=) für die Makrofunktion `EQ` verwendet werden (außerhalb der Klammern wird das Gleichheitszeichen als Zuordnungsoperator interpretiert).

Beispiele

`TEMP = 3 EQ 4` oder `TEMP = 3==4` oder `TEMP = (3=4)`

Erstellt eine neue Spalte `TEMP`, die den Wert eins enthält (vier ist gleich sich selbst).

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

Erstellt eine neue Spalte `TEMP`, die den Wert eins enthält (beim Vergleich von Zeichenfolgen wird die Groß-/Kleinschreibung nicht beachtet).

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

Erstellt eine neue Spalte TEMP, in der jeder Wert eins ist, wenn der entsprechende Zeilenwert von Spalte V1 gleich der Zahl acht ist; andernfalls ist der Wert null.

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

Erstellt eine neue Spalte TEMP, die nur Einsen enthält (jede Zahl ist gleich sich selbst).

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

Erstellt eine neue Spalte TEMP, bei der jeder Wert der Zeilenwert der Spalte V1 im Vergleich zum entsprechenden Zeilenwert der Spalte V2 ist.

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

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Die Spalte TEMP enthält die Werte in V1 im Vergleich zu den entsprechenden Zeilenwerten der Spalte V4. Die Spalte VX enthält den Vergleich von Spalte V2 mit Spalte V5 Die Spalte VY enthält den Vergleich von Spalte V3 mit Spalte V6

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

Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen die Ergebnisse des Vergleichs der Werte in den Zeilen 10-20 der Spalte V1 mit den Zeilen 1-11 der Spalte V2 enthalten. Die anderen Zellen in TEMP sind leer.

Zugehörige Funktionen

Funktion	Syntax
EQ	Gibt TRUE zurück, wenn der eine Datenbereich dem anderen gleich ist.
GE	Gibt TRUE zurück, wenn der eine Datenbereich größer-gleich dem anderen ist.
GT	Gibt TRUE zurück, wenn der eine Datenbereich größer als der andere ist.
LE	Gibt TRUE zurück, wenn der eine Datenbereich kleiner-gleich dem anderen ist.
LT	Gibt TRUE zurück, wenn der eine Datenbereich kleiner als der andere ist.
NE	Gibt TRUE zurück, wenn der eine Datenbereich dem anderen ungleich ist.

EXP-Makro

Die EXP Makro ist nur in Unica Campaign verfügbar.

Syntax

```
EXP(data)
```

Parameter

`data`

Die numerischen Werte, die als Exponenten für die natürliche Zahl e verwendet werden. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

EXP erhebt die natürliche Zahl, e, in die Potenz, die jeweils durch den Wert im angegebenen Datenbereich angegeben wird (berechnet wird also e^x). Die Konstante e ist gleich 2,7182818. EXP gibt für jede Eingabespalte eine neue Spalte zurück, die jeweils das Ergebnis e^x für jeden Wert x in der/den entsprechenden Eingabespalte(n) enthält. EXP ist der Kehrwert der LN-Makrofunktion

Anmerkung: Wenn der Wert *x* zu groß oder zu klein ist, wird ein Überlauffehler zurückgegeben. Dies geschieht, wenn e^x den maximalen 32-Bit-Gleitkommawert überschreitet.

Beispiele

```
TEMP = EXP(2)
```

Erstellt eine neue Spalte TEMP, die den Wert `7.39` enthält.

```
TEMP = EXP(V1)
```

Erstellt eine neue Spalte TEMP, in der jeder Wert das Ergebnis der Anhebung von e auf den Inhalt der Spalte V1 ist.

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

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Die Werte in der Spalte TEMP sind das Ergebnis der Anhebung von e auf die Spalte V1, die Werte der Spalte VX sind das

Ergebnis der Anhebung von e auf den Inhalt der Spalte V2, und die Werte der Spalte VY sind das Ergebnis der Anhebung von e auf den Inhalt der Spalte V3.

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

Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen die Ergebnisse der Anhebung von e auf die Werte in den Zeilen 10-20 der Spalte V1 enthalten. Die anderen Zellen in TEMP sind leer.

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

Erstellt zwei neue Spalten TEMP und VX, die jeweils Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind das Ergebnis der Anhebung von e auf die entsprechenden Zeilenwerte der Spalte V1, und die Werte in der Spalte VX sind das Ergebnis der Anhebung von eauf die entsprechenden Zeilenwerte der Spalte V2.

Zugehörige Funktionen

Funktion	Syntax
LN oder LOG	Berechnet den natürlichen Logarithmus des Inhalts des angegebenen Datenbereichs.
LOG2	Berechnet die Protokollbasis2 des Inhalts des angegebenen Datenbereichs
LOG10	Berechnet die Protokollbasis10 des Inhalts des angegebenen Datenbereichs
POW	Erhebt einen Basiswert in die Potenz des angegebenen Exponenten.

EXTERNALCALLOUT-Makro

Das EXTERNALCALLOUT Makro ist nur in Unica Interact verfügbar.

Syntax

EXTERNALCALLOUT(*calloutName, arg1, ...)*

Parameter

calloutName

Der Name des mithilfe der ExternalCallout-API erstellten Aufrufs. Dieser Name muss mit dem Namen der ExternalCallout-Kategorie übereinstimmen, die Sie in Platform erstellt haben.

`arg1`

Ein für Ihren Aufruf gegebenenfalls erforderliches Argument.

Syntax

`EXTERNALCALLOUT` können Sie eine externe Anwendung aufrufen, um Ihrem interaktiven Ablaufdiagramm Daten hinzuzufügen. EXTERNALCALLOUT kann alle Daten zurückgeben, die von Ihrem Aufruf zurückgegeben werden. Sie müssen diesen Aufruf mithilfe der ExternalCallout-API in Java schreiben. Weitere Informationen finden Sie im Unica Interact Administratorhandbuch.

Beispiele

`EXTERNALCALLOUT(getStockPrice, UNCA)`

Ruft den Aufruf `getStockPrice` auf und übergibt den Namen der Aktie, UNCA, als Argument. Dieser benutzerdefinierte Aufruf gibt den Aktienkurs zurück, wie er durch den Aufruf definiert ist.

INDEXOF-Makro

Das INDEXOF-Makro ist ein internes Makro, das nur in Unica Interact verfügbar ist. Dieses Makro wird als Parameter im Makro EXTERNALCALLOUT übergeben. Das Makro wird mit dem Start des Servers zu EXTERNALCALLOUT hinzugefügt. Für die Verwendung dieses Makros ist keine externe Konfiguration erforderlich.

Syntax

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

Parameter

'indexOf'

indexOf wird im EXTERNALCALLOUT-Makro als vordefinierter Callout-Name übergeben. Dieser Parameter ist obligatorisch und unterscheidet nicht zwischen Groß- und Kleinschreibung.

Dimensionsfedausdruck

Ein Argument, das für das Callout "indexof" erforderlich ist. Die Benutzer müssen eine Bedingung übergeben, die mehrere Dimensionstabellenfelder umfassen kann.

Beschreibung

Das Makro "indexOf" bietet die Möglichkeit, Felder mit mehreren Dimensionstabellen abzufragen. Dieses Makro gibt die Liste der Indizes zurück, die die angegebene Bedingung für jeden Kunden erfüllen. Beim Erstellen eines interaktiven Flussdiagramms können Benutzer Datensätze basierend auf einem bestimmten Ausdruck abrufen. Das Makro generiert einen Fehler, wenn eine falsche Anzahl von Argumenten an das Makro übergeben wird. Bei Syntaxfehlern wird die Fehlermeldung angezeigt, während das Flussdiagramm ausgeführt wird.

Eine Syntaxprüfung überprüft diese Fehler nicht.

Beispiele

Für die folgende Dimensionstabelle **Account_details**, `EXTERNAL_CALLOUT('IndexOf', Account_details.AccountId>1000 AND account_details.Status=='G')`

AccountId	Saldo	Status
101	1100	G
102	800	G
103	1600	G
104	2100	G

Der obige Ausdruck unter Verwendung von "INDEXOF" gibt eine Liste mit den Indizes 1 und 4 zurück.

FACTORIAL-Makro

Die `FACTORIAL` Makro ist nur in Unica Campaign verfügbar.

Syntax

`FACTORIAL(data)`

Parameter

`data`

Die Ganzzahlwerte, deren Fakultät berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt; der Wert muss jedoch größer-gleich null sein. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`FACTORIAL` berechnet die Fakultät der Werte im angegebenen Datenbereich. Alle Eingabewerte müssen Ganzzahlen größer-gleich null sein. Die Fakultät einer Ganzzahl kleiner-gleich eins ist eins. Für ganze Zahlen `X # 2`, die Fakultät x! = x (x-1)(x-2)…(x - (x-1)). `FACTORIAL` liefert für jede Eingabespalte eine neue Spalte, die jeweils die Fakultät der Zahlen der entsprechenden Eingabespalte enthält.

Anmerkung: Jeder Wert größer als 34 erzeugt die Ausgabe `???` (Gleitkomma-Überlauffehler).

Beispiele

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

Erstellt eine neue Spalte `TEMP`, die den Wert `6` enthält.

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

Generiert einen Fehler 333, der darauf hinweist, dass das Argument größer-gleich 0 sein muss.

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

Erstellt eine neue Spalte `TEMP`, in der jeder Wert die Fakultät des Inhalts der Spalte `V1` ist.

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

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Die Werte in der TEMP-Spalte sind die Fakultäten des Inhalts der Spalte V1, die Werte der Spalte VX sind die Fakultäten des Inhalts der Spalte V2, und die Werte der Spalte VY sind die Fakultäten des Inhalts der Spalte V3.

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

Erstellt eine neue Spalte TEMP, wobei die ersten 11 Zellen die Fakultäten der Werte in den Zeilen 10-20 der Spalte V1 enthalten. Die anderen Zellen in TEMP sind leer.

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

Erstellt zwei neue Spalten TEMP und VX, die jeweils Werte in Zeile 1-50 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind jeweils die Fakultät der Zeilen von Spalte V1, und die Werte in der Spalte VX sind die Fakultät der Werte von Spalte V2.

FLOOR-Makro

Die FLOOR Makro ist nur in Unica Campaign verfügbar.

Syntax

```
FLOOR(data)
```

Parameter

`data`

Die numerischen Werte, deren Untergrenze berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

FLOOR berechnet die Untergrenze der Werte im angegebenen Datenbereich. Die Untergrenze einer Zahl ist die größte ganze Zahl, die kleiner als diese Zahl ist. FLOOR gibt für jede

Eingabespalte eine neue Spalte zurück, die jeweils die Untergrenze der Zahlen in der entsprechenden Eingabespalte enthält.

Anmerkung: Dies ist mit der `INT`-Makrofunktion identisch.

Beispiele

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

Erstellt eine neue Spalte `TEMP`, die den Wert `4` enthält.

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

Erstellt eine neue Spalte `TEMP`, die den Wert `-3` enthält.

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

Erstellt eine neue Spalte `TEMP`, in der jeder Wert die Untergrenze des Inhalts der Spalte `V1` darstellt.

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

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Werte in der `TEMP`-Spalte sind die Untergrenzen des Inhalts der Spalte `V1`, die Werte der Spalte `VX` sind die Untergrenzen des Inhalts der Spalte `V2`, und die Werte der Spalte `VY` sind die Untergrenzen des Inhalts der Spalte `V3`.

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

Erstellt eine neue Spalte `TEMP`, in der die ersten 11 Zellen die Untergrenzen der Werte den in den Zeilen 10-20 von Spalte `V1` enthalten. Die anderen Zellen in `TEMP` sind leer.

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

Erstellt zwei neue Spalten `TEMP` und `VX`, die jeweils Werte in Zeile 1-50 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte `TEMP` sind jeweils die Untergrenzen der Zeilen von Spalte `V1`, die Werte in der Spalte `VX` sind die Untergrenzen der Zeilen von Spalte `V2`.

Zugehörige Funktionen

Funktion	Syntax
`CEILING`	Berechnet die Obergrenze jedes Werts im angegebenen Datenbereich.
`FRACTION`	Berechnet die Nachkommastellen jedes Werts im angegebenen Datenbereich.
`TRUNCATE`	Berechnet den Ganzzahlanteil jedes Werts im angegebenen Datenbereich.

FORMAT-Makro

Die `FORMAT` Makro ist nur in Unica Campaign verfügbar.

Syntax

`Format` weist zwei Formen auf, eine für numerische Datentypen und eine für Text-/Zeichendatentypen.

Bei numerischen Datentypen:

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

Bei Text-/Zeichendatentypen:

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

Parameter

`colName`

Das Makro untersucht `colName` und ermittelt dessen Datentyp, um dementsprechend die jeweiligen Regeln für nachfolgende Parameter anzuwenden.

`width`

Die Breite muss groß genug sein, um das vollständige Ergebnis aufnehmen zu können; andernfalls wird das Ergebnis abgeschnitten. Die zulässigen Werte reichen von 1 bis 29, wenn `colName` numerisch ist, andernfalls von 1 bis 255.

`precision`

Die Genauigkeit ist die Anzahl der Stellen nach dem Dezimalzeichen. Die zulässigen Werte reichen von 0 bis 15. Wenn sie null ist, ist das Ergebnis eine Ganzzahl. Der Standardwert für die Genauigkeit ist 2.

`format_type`

Gültige Schlüsselwörter für `format_type` sind:

`PERIOD` Punkt(.) wird als Dezimalzeichen verwendet. Es wird kein Trennzeichen verwendet. Dies ist der Standardwert.

`COMMA` Als Dezimalzeichen wird ein Komma (,) verwendet. Es wird kein Trennzeichen verwendet.

`PERIOD_COMMA` Als Dezimalzeichen wird ein Punkt und als Trennzeichen ein Komma verwendet.

`COMMA_PERIOD` Als Dezimalzeichen wird ein Komma und als Trennzeichen ein Punkt verwendet.

`alignment`

Gültige Schlüsselwörter für alignment sind LEFT und RIGHT. Bei numerischen Datentypen ist der Standardwert RIGHT, bei Text-/Zeichendatentypen LEFT.

`padding`

Gültige Schlüsselwörter für padding sind SPACE und ZERO. Der Standardwert ist SPACE. Wenn alignment den Wert LEFT aufweist, wird ZERO ignoriert (und stattdessen SPACE verwendet).

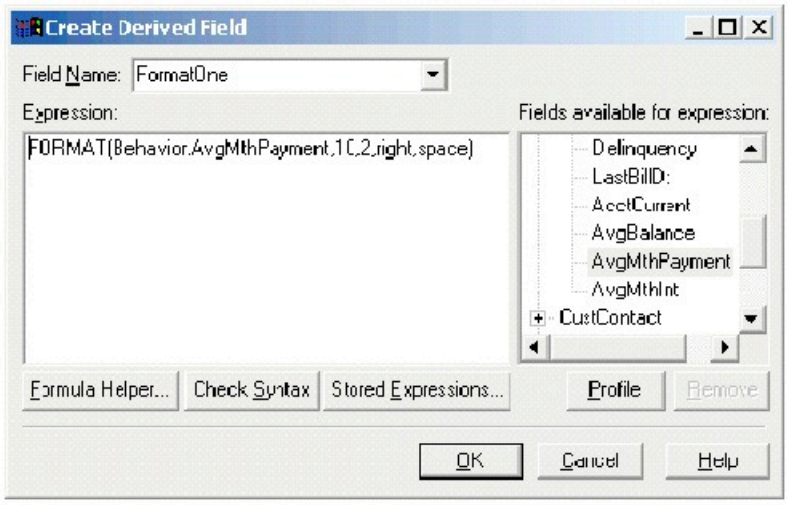
Numerische Zeichenfolgen innerhalb eines Text-/Zeichendatentyps werden als Text/Zeichen behandelt. Außerdem unterstützt die numerische Form mehrere optionale Schlüsselwörter, die jeweils einen Standardwert aufweisen. Um jedoch den Standardwert des zweiten oder eines nachfolgenden optionalen Schlüsselworts zu überschreiben, MÜSSEN Sie die Standardwerte für die vorhergehenden optionalen Schlüsselwörter angeben (sie werden somit obligatorisch). Beispiel: um die Ausrichtung auf LINKS zu überschreiben, müssen Sie kodieren: `FORMAT(myNumCol, 10, 2, PERIOD, LEFT)`.

Syntax

`FORMAT` konvertiert numerische Daten in eine Zeichenfolgeform, wobei die Ausgabezeichenfolge durch verschiedene Formatierungsoptionen gesteuert und definiert werden kann. Dies ist besonders nützlich, wenn Momentaufnahmedateien mit bestimmten Formaten für das Mailing von Dateien erstellt werden sollen.

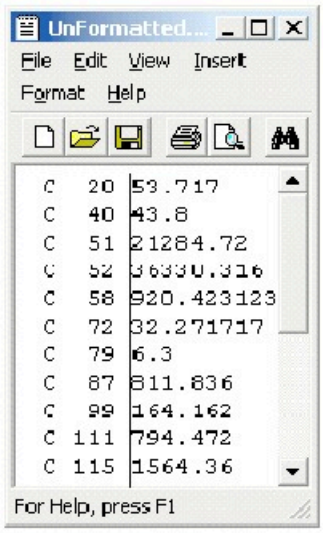
Beispiele

Das folgende Beispiel definiert mithilfe von `FORMAT` ein abgeleitetes Feld.

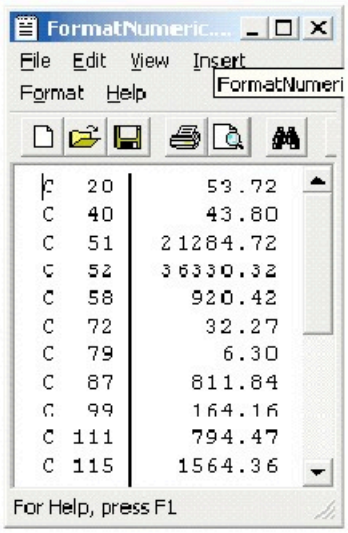


Die folgenden Beispiele zeigen dasselbe Feld, `AvgMthPayment`, in drei Formaten.

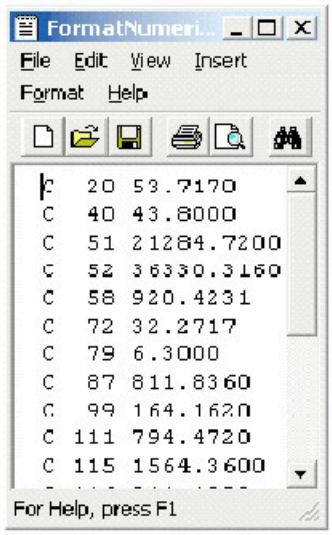
Unformatiert:



Mit `FORMAT(Behavior.AvgMthPayment,10,2,right,space)` formatiert:



Mit `FORMAT(Behavior.AvgMthPayment,10,4)` formatiert:



FRACTION-Makro

Die `FRACTION` Makro ist nur in Unica Campaign verfügbar.

Syntax

`FRACTION(data)`

Parameter

`data`

Die numerischen Werte, deren Nachkommastellen berechnet werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`FRACTION` berechnet den Bruchteil der Werte im angegebenen Datenbereich. Dabei wird für jede Eingabespalte eine neue Spalte zurückgegeben, die die Nachkommastellen der Zahlen in der entsprechenden Eingabespalte enthält.

Anmerkung: Die Makrofunktionen `FRACTION` und `TRUNCATE` sind komplementär, da sie in der Summe die ursprünglichen Werte ergeben.

Beispiele

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

Erstellt eine neue Spalte `TEMP`, die den Wert `0.3` enthält.

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

Erstellt eine neue Spalte `TEMP`, die den Wert `-0.9` enthält.

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

Erstellt eine neue Spalte `TEMP`, in der jeder Wert ein Bruchteil des Inhalts der Spalte `V1` ist.

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

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Werte in der `TEMP`-Spalte sind die Bruchteile des Inhalts der Spalte `V1`, die Werte der Spalte `VX` sind die Bruchteile des Inhalts der Spalte `V2`, und die Werte der Spalte `VY` sind die Bruchteile des Inhalts der Spalte `V3`.

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

Erstellt eine neue Spalte `TEMP`, wobei die ersten 11 Zellen die Bruchteile der Werte in den Zeilen 10-20 der Spalte `V1` enthalten. Die anderen Zellen in `TEMP` sind leer.

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

Erstellt zwei neue Spalten `TEMP` und `VX`, die jeweils Werte in Zeile 1-50 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte `TEMP` sind jeweils sind die Bruchteile der Zeilen von Spalte `V1`, und die Werte in der Spalte `VX` sind die Bruchteile der Werte von Spalte `V2`.

Zugehörige Funktionen

Funktion	Syntax
`CEILING`	Berechnet die Obergrenze jedes Werts im angegebenen Datenbereich.
`FLOOR`	Berechnet die Untergrenze jedes Werts im angegebenen Datenbereich.
`TRUNCATE`	Berechnet den Ganzzahlanteil jedes Werts im angegebenen Datenbereich.

GE-Makro

Das GE-Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

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

Parameter

`data1`

Der numerische Zellenbereich, der verglichen werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`data2`

Die Zahlen, mit denen alle Werte in der angegebenen Spalte verglichen werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in `data2` muss mit der Anzahl der Spalten in `data1` übereinstimmen, es sei denn, bei `data2` handelt es sich um eine Konstante. Informationen zur Formatdefinition von `data` finden Sie

im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`GE` vergleicht die beiden angegebenen Datenbereiche und gibt eine Eins zurück, wenn die Werte im ersten Dataset größer-gleich den Werten im zweiten Dataset sind, andernfalls eine Null. Sie gibt für jede Eingabespalte eine neue Spalte zurück, die jeweils die entsprechende Spalte in `data1` im Vergleich zur entsprechenden Spalte von `data2` enthält (d. h. die erste Spalte von `data1` wird mit der ersten Spalte von `data` verglichen, die zweite Spalte mit der zweiten Spalte usw.).

Wenn es sich bei `data2` um eine Konstante handelt, wird jeder Wert in `data` mit dem betreffenden Wert verglichen. Wenn es sich bei `data2` um eine Spalte handelt, werden die Berechnungen zeilenweise ausgeführt. Die erste Zeile von `data1` wird durch den Wert der ersten Zeile von `data2` dividiert, die zweite Zeile durch die zweite Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

Anmerkung: Der Operator `GE` kann durch ein Größer-als-Zeichen, auf das ein Gleichheitszeichen folgt, abgekürzt werden (>=).

Beispiele

`TEMP = 9 GE 4` oder `TEMP = 9 >= 4`

Erstellt eine neue Spalte `TEMP`, die den Wert eins enthält (neun ist größer als vier).

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

Erstellt eine neue Spalte `TEMP`, in der jeder Wert eins ist, wenn der entsprechende Zeilenwert von Spalte `V1` größer-gleich der Zahl acht ist; andernfalls ist der Wert null.

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

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Werte in der Spalte `TEMP` entsprechen dem Inhalt der Spalte `V1` im Vergleich zum Wert zwei, die Werte der Spalte `VX` entsprechen dem Inhalt der Spalte `V2` im Vergleich zum Wert zwei, und die Werte der Spalte `VY` entsprechen dem Inhalt der Spalte `V3` im Vergleich zum Wert zwei.

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

Erstellt eine neue Spalte TEMP, die nur Einsen enthält (jede Zahl ist gleich sich selbst).

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

Erstellt eine neue Spalte TEMP, bei der jeder Wert der Zeilenwert der Spalte V1 im Vergleich zum entsprechenden Zeilenwert der Spalte V2 ist.

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

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Die Spalte TEMP enthält die Werte in V1 im Vergleich zu den entsprechenden Zeilenwerten der Spalte V4. Die Spalte VX enthält den Vergleich von Spalte V2 mit Spalte V5 Die Spalte VY enthält den Vergleich von Spalte V3 mit Spalte V6

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

Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen die Ergebnisse des Vergleichs der Werte in den Zeilen 10-20 der Spalte V1 mit den Werten in den Zeilen 1-11 der Spalte V2 enthalten. Die anderen Zellen in TEMP sind leer.

Zugehörige Funktionen

NEGibt TRUE zurück, wenn der eine Datenbereich dem anderen ungleich ist.

GET macro

Das GET-Makro ist verfügbar in Unica Interact.

Syntax

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

Parameter

`index`

Das Element im angegebenen Index abrufen. Bei der Verwendung in Interact ist es erlaubt, mehrere Indizes als Ergebnis eines anderen Ausdrucks zu haben. In diesem Fall werden die in "index" angegebenen Elemente in <Dim-Feld> abgerufen. Ungültige(r) Index(e) werden übersprungen.

`dim field`

Das Element mit dem angegebenen Index wird aus dem Dimensionsfeld abgerufen.

Beschreibung

Dieses Makro ist unter "Alle eingebauten Makros" aufgelistet. Sie können dieses Makro beim Entwerfen eines interaktiven Ablaufdiagramms verwenden. Wenn der Index außerhalb des Bereichs liegt, wird ein Fehler mit dem erwarteten Indexbereich zurückgegeben.

Anmerkung: Dieses Makro unterstützt die 1-basierte Indizierung.

Beispiel

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

GROUPBY-Makro

Die `GROUPBY` Makro ist nur in Unica Campaign verfügbar.

Syntax

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

Parameter

- `group_field`

  Gibt die Variable an, nach der die Einträge gruppiert werden (d. h., alle identischen Werte der angegebenen Variable werden derselben Gruppe zugeordnet).
- `keyword`

  Gibt die zusammenfassende Rollup-Funktion an, die auf das Rollup-Feld angewendet werden soll.
- `rolled_field`

  Gibt die Variable an, die zusammengefasst werden soll.
- `output_field`

Gibt eine alternative Variable an, die für eine einzelne Zeile einer Gruppe zurückgegeben werden soll, und kann nur mit den Schlüsselwörtern `MinOf`, `MaxOf` und `MedianOf` eingesetzt werden.

Syntax

`GROUPBY` Fasst mehrere Zeilen von Daten in einer Gruppe zusammen Die Ausgabe dieser Funktion ist eine einzelne Spalte. Die Ausgabe ist das Ergebnis der durch `keyword` angegebenen Operation bezüglich `rolled_field` in der durch `group_field` angegebenen homogenen Gruppe. Wenn es mehrere Antworten gibt, die eine bestimmte Bedingung erfüllen, wird die erste gefundene Antwort zurückgegeben.

Wenn das optionale `output_field` nicht angegeben ist, ist die Ausgabe das Ergebnis der auf `rolled_field` angewendeten Operation. Wenn `output_field` angegeben ist, ist das Ergebnis das `output_field` der Zeile in der Gruppe.

Wenn mehrere Zeilen in einer Gruppe die angegebene Bedingung erfüllen (etwa wenn bezüglich des Maximalwerts ein Gleichstand vorliegt), wird das `output-field` zurückgegeben, das der ersten Zeile zugeordnet ist, die die Bedingung erfüllt.

Anmerkung: Wenn Sie eine mehrspaltige Gruppierung bearbeiten möchten, können Sie eine Liste von Feldnamen, die durch Kommas getrennt sind, in geschweifte Klammern "{ }" einschließen und im Aufruf des Makros `GROUPBY` als ersten Parameter verwenden.

Die folgenden Schlüsselwörter werden unterstützt (die Groß-/Kleinschreibung wird nicht beachtet):

Suchbegriff	Zeichenfolge? Ja/Nein	Syntax
`CountOf`	`Yes`	Gibt die Anzahl der Einträge in jeder Gruppe zurück (`rolled_field`kann numerisch oder eine Zeichenfolge sein; der Rückgabewert ist unabhängig vom Wert von `rolled_field` derselbe).
`MinOf`	`Yes`	Gibt den Minimalwert von `rolled_field` in jeder Gruppe zurück (`rolled_field` kann numerisch oder eine Zeichenfolge sein; wenn `rolled_field` eine Zeichenfolge ist, wird der Wert

Suchbegriff	Zeichenfolge? Ja/Nein	Syntax
		zurückgegeben, der bei alphabetischer Sortierung dem Anfang des Alphabets am nächsten liegt).
`MaxOf`	`Yes`	Gibt den Maximalwert von `rolled_field` in jeder Gruppe zurück (`rolled_field` kann numerisch oder eine Zeichenfolge sein; wenn `rolled_field` eine Zeichenfolge ist, wird der Wert zurückgegeben, der bei alphabetischer Sortierung dem Ende des Alphabets am nächsten liegt).
`DiffOf`	`Yes`	Gibt die Anzahl unterschiedlicher Werte von `rolled_field` in jeder Gruppe zurück (`rolled_field` kann numerisch oder eine Zeichenfolge sein).
`AvgOf`	`No`	Gibt den Durchschnittswert von `rolled_field` in jeder Gruppe zurück (`rolled_field` muss numerisch sein).
`ModeOf`	`Yes`	Gibt den Modalwert (d. h. den am häufigsten vorkommenden Wert) von `rolled_field` in jeder Gruppe zurück (`rolled_field` kann numerisch oder eine Zeichenfolge sein).
`MedianOf`	`Yes`	Gibt den gemittelten Wert (d. h. den Mittelwert bei Sortierung nach `rolled_field`) von `rolled_field` in jeder Gruppe zurück (`rolled_field` kann numerisch oder eine Zeichenfolge sein; wenn `rolled_field` eine Zeichenfolge ist, werden die Werte alphabetisch sortiert).
`OrderOf`	`Yes`	Gibt die Reihenfolge von `rolled_field` in jeder Gruppe zurück (`rolled_field` muss numerisch sein). Wenn mehrere Einträge denselben Wert haben, erhalten alle denselben Wert.
`SumOf`	`No`	Gibt die Summe von `rolled_field` in jeder Gruppe zurück (`rolled_field` muss numerisch sein).
`StdevOf`	`No`	Gibt die Standardabweichung von `rolled_field` in jeder Gruppe zurück (`rolled_field` muss numerisch sein).
`IndexOf`	`Yes`	Gibt den 1-basierten Index (nach rolled_field sortiert) jedes Datensatzes zurück (rolled_field kann numerisch oder eine Zeichenfolge sein). Die Sortierreihenfolge ist aufsteigend.

Suchbegriff	Zeichenfolge? Ja/Nein	Syntax
		Anmerkung: Bei numerischen Feldern kann die Sortierreihenfolge von RankOf und IndexOf absteigend gemacht werden, indem dem Sortierfeld ein Minuszeichen (-) vorangestellt wird.
`RankOf`	`Yes`	Gibt die 1-basierte Kategorie (nach rolled_field sortiert) zurück, in der jeder Datensatz liegt (rolled_field kann numerisch oder eine Zeichenfolge sein). Die Sortierreihenfolge ist aufsteigend. Anmerkung: Bei numerischen Feldern kann die Sortierreihenfolge von RankOf und IndexOf absteigend gemacht werden, indem dem Sortierfeld ein Minuszeichen (-) vorangestellt wird.

Beispiele

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

Berechnet die Summe aller Kontostände für den jeweiligen Haushalt.

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

Gibt die Kontonummer des ersten Kontos zurück, das vom Kunden eröffnet wurde.

GROUPBY_WHERE-Makro

Die `GROUPBY_WHERE` Makro ist nur in Unica Campaign verfügbar.

Syntax

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

Parameter

- `group_field`

  Gibt die Variable an, nach der die Einträge gruppiert werden (d. h., alle identischen Werte der angegebenen Variable werden derselben Gruppe zugeordnet).
- `keyword`

Gibt die zusammenfassende Rollup-Funktion an, die auf das Rollup-Feld angewendet werden soll.

- `rolled_field`

  Gibt die Variable an, die zusammengefasst werden soll.

- `where_value`

  Ein Ausdruck, dessen Auswertung den Wert eins oder null ergibt und der angibt, welche Zeilen in die Rollup-Operation einbezogen werden sollen.

- `output_field`

  Gibt eine alternative Variable an, die für eine einzelne Zeile einer Gruppe zurückgegeben werden soll, und kann nur mit den Schlüsselwörtern `MinOf`, `MaxOf` und `MedianOf` eingesetzt werden.

Syntax

`GROUPBY_WHERE` fasst mehrere Zeilen von Daten in einer Gruppe zusammen. Die Ausgabe dieser Funktion ist eine einzelne Spalte. Die Ausgabe ist das Ergebnis der durch `keyword` angegebenen Operation bezüglich `rolled_field` in der durch `group_field` angegebenen homogenen Gruppe, gefiltert durch `where_value`. Es werden nur Zeilen mit dem `where_value` eins in die Berechnung einbezogen.

Wenn das optionale `output_field` nicht angegeben ist, ist die Ausgabe das Ergebnis der auf `rolled_field` angewendeten Operation. Wenn `output_field` angegeben ist, ist das Ergebnis das `output_field` der Zeile in der Gruppe.

Anmerkung: Unter GROUPBY-Makro *(auf Seite 105)* finden Sie weitere Informationen zu gültigen Werten für `keyword`.

Beispiele

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

Berechnet die Summe aller Konten mit positivem Saldo für den jeweiligen Haushalt.

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

Berechnet den durchschnittlichen Kaufbetrag des jeweiligen Kunden bei Einkäufen in den vergangenen 90 Tagen.

GT-Makro

Die `GT` Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

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

Parameter

`data1`

Der numerische Zellenbereich, der verglichen werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`data2`

Die Zahlen, mit denen alle Werte in der angegebenen Spalte verglichen werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in `data2` muss mit der Anzahl der Spalten in `data1` übereinstimmen, es sei denn, bei `data2` handelt es sich um eine Konstante. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`GT` vergleicht die beiden angegebenen Datenbereiche und gibt eine Eins zurück, wenn die Werte im ersten Dataset größer als die Werte im zweiten Dataset sind, andernfalls eine Null. Sie gibt für jede Eingabespalte eine neue Spalte zurück, die jeweils die entsprechende Spalte in `data1` im Vergleich zur entsprechenden Spalte von `data2` enthält (d. h. die erste Spalte

von `data1` wird mit der ersten Spalte von `data` verglichen, die zweite Spalte mit der zweiten Spalte usw.).

Wenn es sich bei `data2` um eine Konstante handelt, wird jeder Wert in `data` mit dem betreffenden Wert verglichen. Wenn es sich bei `data2` um eine Spalte handelt, werden die Berechnungen zeilenweise ausgeführt. Die erste Zeile von `data1` wird durch den Wert der ersten Zeile von `data2` dividiert, die zweite Zeile durch die zweite Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

Anmerkung: Der Operator `GT` kann durch ein Größer-als-Zeichen abgekürzt werden (`>`).

Beispiele

`TEMP = 3 GT 4` oder `TEMP = 3 > 4`

Erstellt eine neue Spalte `TEMP`, die den Wert Null enthält (da drei nicht größer als vier ist).

```
TEMP = V1 > 8
```

Erstellt eine neue Spalte `TEMP`, in der jeder Wert Eins ist, wenn der entsprechende Zeilenwert der Spalte `V1` größer als die Zahl Acht ist, andernfalls ist der Wert Null.

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

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Werte in der Spalte `TEMP` entsprechen dem Inhalt der Spalte `V1` im Vergleich zum Wert zwei, die Werte der Spalte `VX` entsprechen dem Inhalt der Spalte `V2` im Vergleich zum Wert zwei, und die Werte der Spalte `VY` entsprechen dem Inhalt der Spalte `V3` im Vergleich zum Wert zwei.

```
TEMP = V1 > V1
```

Erstellt eine neue Spalte `TEMP`, die nur Nullen enthält (keine Zahl ist größer als sie selbst).

```
TEMP = V1 > V2
```

Erstellt eine neue Spalte `TEMP`, bei der jeder Wert der Zeilenwert der Spalte `V1` im Vergleich zum entsprechenden Zeilenwert der Spalte `V2` ist.

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

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Spalte `TEMP` enthält die Werte in `V1` im Vergleich zu den entsprechenden Zeilenwerten der Spalte `V4`. Die Spalte `VX` enthält

den Vergleich von Spalte V2 mit Spalte V5 Die Spalte VY enthält den Vergleich von Spalte V3 mit Spalte V6

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

Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen die Ergebnisse des Vergleichs der Werte in den Zeilen 10-20 der Spalte V1 mit den Werten in den Zeilen 1-11 der Spalte V2 enthalten. Die anderen Zellen in TEMP sind leer.

Zugehörige Funktionen

Funktion	Syntax
EQ	Gibt TRUE zurück, wenn der eine Datenbereich dem anderen gleich ist.
GE	Gibt TRUE zurück, wenn der eine Datenbereich größer-gleich dem anderen ist.
LE	Gibt TRUE zurück, wenn der eine Datenbereich kleiner-gleich dem anderen ist.
LT	Gibt TRUE zurück, wenn der eine Datenbereich kleiner als der andere ist.
NE	Gibt TRUE zurück, wenn der eine Datenbereich dem anderen ungleich ist.

IF-Makro

Die IF Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

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

Parameter

`predicate_col`

Eine Spalte mit booleschen Werten oder ein Ausdruck, dessen Auswertung eine einzelne Spalte mit booleschen Werten ergibt. Boolesche Werte werden als null oder ungleich null interpretiert. Diese Spalte sollte mindestens so viele Zeilen enthalten wie der Datenbereich, aus dem die data extrahiert werden.

`then_value`

Der oder die Werte, die zurückgegeben werden, wenn die entsprechende Zeile von `predicate_col` einen Wert ungleich null enthält. Dabei kann es sich um einen konstanten

Wert, eine Spalte oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Unter Makrofunktionsparameter für Unica Campaign (auf Seite 12) finden Sie weitere Informationen zur Formatdefinition von `then_value` (identisch mit `data`).

`else_value`

Wenn dieser optionale Parameter angegeben wird, wird dieser Wert zurückgegeben, falls die entsprechende Zeile von `predicate_col` eine Null enthält. Dabei kann es sich um einen konstanten Wert, eine Spalte oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Wenn `else_value` nicht angegeben wird, wird jedes Mal eine Null zurückgegeben, wenn die Auswertung von `predicate_col` 'falsh' ergibt. Unter Makrofunktionsparameter für Unica Campaign (auf Seite 12) finden Sie weitere Informationen zur Formatdefinition von `else_value` (identisch mit `data`).

Syntax

`IF` wertet den Ausdruck in `predicate_col` aus und gibt `then_value` zurück, wenn der Ausdruck wahr ist, bzw. `else_value`, wenn der Ausdruck "false" ist. IF gibt dieselbe Anzahl von Spalten im `then_value` und im `else_value` zurück. Die neuen Spalten enthalten die entsprechenden Zeilenwerte aus `then_value`, wenn der Wert von `predicate_col` ungleich null ist. Wenn der `else_value` angegeben wird, wird er zurückgegeben, wenn der Wert der `predicate_col` null ist. Wenn `else_value` nicht angegeben wird, wird null zurückgegeben.

Da `IF` zeilenweise vorgeht, wird für jede Zeile bis zum letzten Wert der kürzesten Spalte (d. h. der kürzesten Spalte von `predicate_col`, `then_value` und `else_value`) ein Ergebnis erzeugt.

Anmerkung: In der Regel empfiehlt es sich, eine Prädikatspalte zu erstellen, die eine der Vergleichsmakrofunktionen verwendet (==, >, <, `ISEVEN`, `ISODD` usw.).

Beispiele

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

Erstellt eine neue Spalte `TEMP`, die eine Kopie von Spalte `V1` enthält.

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

Erstellt eine neue Spalte `TEMP`, in der jeder Wert eins ist, wenn der entsprechende Zeilenwert von Spalte `V1` ungleich null ist; andernfalls ist der Wert null.

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

Erstellt eine neue Spalte `TEMP`, in die jeder Wert aus der Spalte `V1` kopiert wird, wenn der entsprechende Wert der Spalte `V3` ungleich Null ist; andernfalls wird der Wert aus der Spalte `V2` kopiert.

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

Erstellt eine neue Spalte `TEMP`, die jeden Wert in Spalte `V1` enthält, der weniger als eine Standardabweichung vom Mittelwert entfernt ist.

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

Erstellt eine neue Spalte `TEMP`, die Werte für Zeile 10-20 enthält. Jeder Wert wird aus Spalte `V1` (Zelle 10-20) kopiert, wenn der entsprechende Wert von Spalte `V3` (Zelle 30-40) ungleich null ist; andernfalls wird der Wert aus Spalte `V2` (Zelle 1-11) kopiert.

IN-Makro

Die `IN` Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

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

Parameter

Die erste Form ermöglicht die Verwendung einer Werteliste statt einer Unterabfrage.

Die zweite Form verwendet eine Unterabfrage, deren Auswertung ein Zwischenergebnis erzeugt, auf das weitere Verarbeitungsschritte angewendet werden können.

Syntax

Das Prädikat IN ermöglicht die Verwendung einer Liste von Werten statt einer Unterabfrage oder es leitet eine Unterabfrage ein.

Anmerkung: Das Prädikat IN verfügt über eine negative Version, NOT IN. Deren Format ist mit dem von IN identisch. NOT IN ist nur wahr, wenn der angegebene Wert nicht in den von der Unterabfrage zurückgegebenen Werten gefunden wird.

Wichtig: Bei der Verwendung von `IN` in Unica Interact können Sie nur die `value IN (value1 AND value2 . . . .)`-Syntax verwenden.

Beispiele

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

Gibt die angegebene(n) Spalte(n) aus einem Datenbereich zurück.

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

Erstellt eine neue Spalte `TEMP`, die den Wert Eins enthält.

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

Erstellt eine neue Spalte `TEMP`, die alle Einsen enthält.

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

Erstellt eine neue Spalte `TEMP`, in der jeder Wert eine Eins ist, wenn die entsprechende Zeile der Spalte `V1` einen Wert in der Spalte `V2`enthält, andernfalls eine Null.

INT-Makro

Die `INT` Makro ist nur in Unica Campaign verfügbar.

Syntax

```
INT(data)
```

Parameter

`data`

Die numerischen Werte, die auf einen Ganzzahlwert abgerundet werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition

von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`INT` berechnet die größte Ganzzahl, die kleiner als die Werte (auch als Untergrenze bezeichnet) im angegebenen Datenbereich ist. `INT` gibt für jede Eingabespalte eine neue Spalte zurück, die jeweils die Untergrenze der Zahlen in der entsprechenden Eingabespalte enthält.

Anmerkung: Dies ist mit der `FLOOR`-Makrofunktion identisch.

Beispiele

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

Erstellt eine neue Spalte `TEMP`, die den Wert `4` enthält.

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

Erstellt eine neue Spalte `TEMP`, die den Wert `-2` enthält.

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

Erstellt eine neue Spalte `TEMP`, in der jeder Wert die größte Ganzzahl ist, die kleiner oder gleich dem Inhalt der Spalte `V1` ist.

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

Erstellt eine neue Spalte `TEMP`, die den Dezimalanteil jedes Werts in Spalte `V1` enthält.

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

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Werte in der Spalte `TEMP` sind jeweils die größten Ganzzahlen kleiner-gleich dem Inhalt von Spalte `V1`, die Werte von Spalte `VX`sind die größten Ganzzahlen kleiner-gleich dem Inhalt von Spalte `V2` und die Werte von Spalte `VY` sind die größten Ganzzahlen kleiner-gleich dem Inhalt von Spalte `V3`.

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

Erstellt eine neue Spalte `TEMP`, in der die ersten 11 Zellen die größten Ganzzahlen enthalten, die kleiner oder gleich den entsprechenden Werten in den Zeilen 10-20 der Spalte `V1` sind. Die anderen Zellen in `TEMP` sind leer.

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

Erstellt zwei neue Spalten `TEMP` und `VX`, die jeweils Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte `TEMP` sind die größten Ganzzahlen kleiner-gleich den entsprechenden Zeilenwerten von Spalte `V1`, die Werte in der Spalte `VX` sind die größten Ganzzahlen kleiner-gleich den entsprechenden Zeilenwerten von Spalte `V2`.

Zugehörige Funktionen

Funktion	Syntax
`ROUND`	Berechnet den gerundeten Wert des Inhalts des angegebenen Datenbereichs.
`TRUNCATE`	Berechnet den Ganzzahlanteil jedes Werts im angegebenen Datenbereich.

INVERSE-Makro

Die `INVERSE` Makro ist nur in Unica Campaign verfügbar.

Syntax

```
INVERSE(data)
```

Parameter

`data`

Die numerischen Werte, deren Umkehrfunktion berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`INVERSE` berechnet den Negativwert der Werte im angegebenen Datenbereich. Gibt -x zurück (d.h. negative Werte werden als positive Werte und positive Werte als negative Werte zurückgegeben). `INVERSE`gibt für jede Eingabespalte eine neue Spalte zurück, die jeweils den Kehrwert der Werte in der entsprechenden Eingabespalte enthält.

Anmerkung: Stellen Sie zum Umkehren eines Werts oder einer Spalte ein Minuszeichen (`-`) voran. Zum Beispiel ist `V2 = -V1` gleich `V2 = INVERSE(V1)`.

Beispiele

```
TEMP = INVERSE(3.2)
```

Erstellt eine neue Spalte `TEMP`, die den Wert `-3.2` enthält.

```
TEMP = INVERSE(V1)
```

Erstellt eine neue Spalte `TEMP`, in der jeder Wert den Negativwert der Werte in Spalte `V1` darstellt.

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

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Werte in der Spalte `TEMP` sind jeweils der Negativwert der Werte in Spalte `V1`, die Werte von Spalte `VX` sind der Negativwert der Werte in Spalte `V2` und die Werte von Spalte `VY` sind der Negativwert der Werte in Spalte `V3`.

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

Erstellt eine neue Spalte `TEMP`, in der die ersten 11 Zellen die Negativwerte der Werte in den Zeilen 10-20 der Spalte `V1` enthalten. Die anderen Zellen in `TEMP` sind leer.

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

Erstellt zwei neue Spalten `TEMP` und `VX`, die jeweils Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte `TEMP` sind die Negativwerte der Werte der entsprechenden Zeilen der Spalte `V1`, und die Werte in der Spalte `VX` sind die Negativwerte der Werte der entsprechenden Zeilen der Spalte `V2`.

Zugehörige Funktionen

Funktion	Syntax
`ABS`	Berechnet den absoluten Wert des Inhalts des angegebenen Datenbereichs.
`NOT`	Berechnet das logische Nicht des Inhalts des angegebenen Datenbereichs.
`SIGN`	Berechnet das Vorzeichen (positiv oder negativ) der Werte im angegebenen Datenbereich.

IS-Makro

Die `IS` Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

```
IS <keyword>
```

Parameter

`keyword`

Suchbedingung, üblicherweise "NULL," "TRUE," "UNKNOWN" oder "FALSE".

Syntax

`IS` wird in komplexen Suchbedingungen eingesetzt. Je komplexer die Suche, desto nützlicher kann die IS-Bedingung sein. Diese booleschen Suchbedingungen stellen eine alternative Möglichkeit dar, einfache Suchbedingungen zu formulieren.

gibt in Unica Interact andere Ergebnisse zurück als in Unica Campaign. Bei NULL wird 1 zurückgegeben, wenn mindestens ein Nullwert für eine Zielgruppen-ID vorliegt. Bei UNKNOWN wird für eine Zielgruppen-ID 1 zurückgegeben, wenn sie keinen Wert aufweist.

ISERROR-Makro

Die `ISERROR` Makro ist nur in Unica Campaign verfügbar.

Syntax

```
ISERROR(data)
```

Parameter

`data`

Die zu prüfenden Werte, wenn eine der Zeilen einen Fehler enthält (d. h. eine Zelle mit `???`). Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur

Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`ISERROR` prüft, ob eine der Zellen in jeder Zeile des angegebenen Datenbereichs einen Fehler enthält (d. h. eine `???`-Zelle). gibt eine neue Spalte zurück, in der jede Zeile eine Eins enthält, wenn die entsprechende Zeile von `data` einen Fehler enthält. Andernfalls enthält die Zeile eine Null. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der längsten Spalte.

Beispiele

```
TEMP = ISERROR(-3)
```

Erstellt eine neue Spalte `TEMP`, die den Wert Null enthält.

```
TEMP = ISERROR(V1)
```

Erstellt eine neue Spalte `TEMP`, in der jeder Wert eins ist, wenn die entsprechende Zeile von Spalte `V1`die Angabe `???` enthält; andernfalls ist der Wert null.

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

Erstellt eine neue Spalte `TEMP`, in der jeder Wert eine Eins ist, wenn eine der Zellen in den entsprechenden Zeilen der Spalte `V1` - `V3` `???` enthält, andernfalls ist der Wert null.

```
TEMP = ISERROR(V1[50:100]:V10)
```

Erstellt eine neue Spalte `TEMP` mit Werten in Zeile 1-50. Jeder Wert ist eins, wenn eine der Zellen in Zeile 50-100 von Spalte `V1` - `V10` die Angabe `???` enthält; andernfalls ist der Wert null.

ISODD-Makro

Die `ISODD` Makro ist nur in Unica Campaign verfügbar.

Syntax

```
ISODD(data)
```

Parameter

`data`

Die numerischen Werte, bei denen geprüft werden soll, ob sie ungerade sind. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`ISODD` testet jeden Wert im angegebenen Dataset auf Geradheit. ISODD gibt für jede Eingabespalte eine neue Spalte zurück, die bei allen ungeraden Werten (d. h., der Wert modulo zwei ist eins) eine Eins und bei allen geraden Werten eine Null enthält.

Anmerkung: Bei Werten, die keine Ganzzahlen sind, wird zunächst die Makrofunktion `INT` angewendet. Beispiel: `ISODD(2.5) = 0`, da `2` nicht ungerade ist.

Beispiele

```
TEMP = ISODD(-3)
```

Erstellt eine neue Spalte `TEMP`, die den Wert Eins enthält.

```
TEMP = ISODD(V1)
```

Erstellt eine neue Spalte `TEMP`, in der jeder Wert das Ergebnis der Prüfung des Inhalts der Spalte `V1` auf Ungeradheit ist.

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

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Werte in der Spalte `TEMP` sind die Ergebnisse der Prüfung des Inhalts der Spalte `V1` auf Ungeradheit, die Werte der Spalte `VX` sind die Ergebnisse der Prüfung des Inhalts der Spalte `V2` auf Ungeradheit, und die Werte der Spalte `VY` sind die Ergebnisse der Prüfung des Inhalts der Spalte `V3` auf Ungeradheit.

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

Erstellt eine neue Spalte `TEMP`, in der die ersten 11 Zellen die Ergebnisse der Prüfung der Werte in den Zeilen 10-20 der Spalte `V1`auf Ungeradheit enthalten. Die anderen Zellen in `TEMP` sind leer.

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

Erstellt zwei neue Spalten `TEMP` und `VX`, die jeweils Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte `TEMP` sind die Ergebnisse der Prüfung der entsprechenden Zeilen der Spalte `V1` auf Ungeradheit, und die Werte in der Spalte `VX` sind die Ergebnisse der Prüfung der entsprechenden Zeilen der Spalte `V2` auf Ungeradheit.

Zugehörige Funktionen

Funktion	Syntax
`ISEVEN`	Prüft, ob die Eingabewerte gerade (d. h. durch zwei teilbar) sind

ISEVEN-Makro

Die `ISEVEN` Makro ist nur in Unica Campaign verfügbar.

Syntax

```
ISEVEN(data)
```

Parameter

`data`

Die numerischen Werte, bei denen geprüft werden soll, ob sie gerade sind. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`ISEVEN` testet jeden Wert im angegebenen Dataset auf Geradheit. ISEVEN gibt für jede Eingabespalte eine neue Spalte zurück, die bei allen geraden Werten (d. h., der Wert modulo zwei ist null) eine Eins und bei allen ungeraden Werten eine Null enthält.

Anmerkung: Bei Werten, die keine Ganzzahlen sind, wird zunächst die Makrofunktion `INT` angewendet. Zum Beispiel, `ISEVEN(2.5) = 1`, da `2` gleich groß ist.

Beispiele

```
TEMP = ISEVEN(-3)
```

Erstellt eine neue Spalte TEMP, die den Wert Null enthält.

```
TEMP = ISEVEN(V1)
```

Erstellt eine neue Spalte TEMP, in der jeder Wert das Ergebnis der Prüfung des Inhalts der Spalte V1 auf Gleichheit ist.

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

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Die Werte in der Spalte TEMP sind die Ergebnisse der Prüfung des Inhalts der Spalte V1 auf Gleichheit, die Werte der Spalte VX sind die Ergebnisse der Prüfung des Inhalts der Spalte V2 auf Gleichheit, und die Werte der Spalte VY sind die Ergebnisse der Prüfung des Inhalts der Spalte V3auf Gleichheit.

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

Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen die Ergebnisse der Prüfung der Werte in den Zeilen 10-20 der Spalte V1 auf Gleichheit enthalten. Die anderen Zellen in TEMP sind leer.

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

Erstellt zwei neue Spalten TEMP und VX, die jeweils Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind die Ergebnisse der Prüfung der entsprechenden Zeilen der Spalte V1 auf Gleichheit, und die Werte in der Spalte VX sind die Ergebnisse der Prüfung der entsprechenden Zeilen der Spalte V2 auf Gleichheit.

Zugehörige Funktionen

Funktion	Syntax
ISODD	Prüft, ob die Eingabewerte ungerade (d. h. nicht durch zwei teilbar) sind

ISODD-Makro

Die ISODD Makro ist nur in Unica Campaign verfügbar.

Syntax

```
ISODD(data)
```

Parameter

`data`

Die numerischen Werte, bei denen geprüft werden soll, ob sie ungerade sind. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`ISODD` testet jeden Wert im angegebenen Dataset auf Geradheit. ISODD gibt für jede Eingabespalte eine neue Spalte zurück, die bei allen ungeraden Werten (d. h., der Wert modulo zwei ist eins) eine Eins und bei allen geraden Werten eine Null enthält.

Anmerkung: Bei Werten, die keine Ganzzahlen sind, wird zunächst die Makrofunktion `INT` angewendet. Beispiel: `ISODD(2.5) = 0`, da `2` nicht ungerade ist.

Beispiele

```
TEMP = ISODD(-3)
```

Erstellt eine neue Spalte `TEMP`, die den Wert Eins enthält.

```
TEMP = ISODD(V1)
```

Erstellt eine neue Spalte `TEMP`, in der jeder Wert das Ergebnis der Prüfung des Inhalts der Spalte `V1` auf Ungeradheit ist.

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

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Werte in der Spalte `TEMP` sind die Ergebnisse der Prüfung des Inhalts der Spalte `V1` auf Ungeradheit, die Werte der Spalte `VX` sind die Ergebnisse der Prüfung des Inhalts der Spalte `V2` auf Ungeradheit, und die Werte der Spalte `VY` sind die Ergebnisse der Prüfung des Inhalts der Spalte `V3` auf Ungeradheit.

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

Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen die Ergebnisse der Prüfung der Werte in den Zeilen 10-20 der Spalte V1auf Ungeradheit enthalten. Die anderen Zellen in TEMP sind leer.

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

Erstellt zwei neue Spalten TEMP und VX, die jeweils Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind die Ergebnisse der Prüfung der entsprechenden Zeilen der Spalte V1 auf Ungeradheit, und die Werte in der Spalte VX sind die Ergebnisse der Prüfung der entsprechenden Zeilen der Spalte V2 auf Ungeradheit.

Zugehörige Funktionen

Funktion	Syntax
ISEVEN	Prüft, ob die Eingabewerte gerade (d. h. durch zwei teilbar) sind

LE-Makro

Die LE Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

```
data1 LE data2 data1 <= data2
```

Parameter

`data1`

Der numerische Zellenbereich, der verglichen werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`data2`

Die Zahlen, mit denen alle Werte in der angegebenen Spalte verglichen werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur

Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`LE` vergleicht die beiden angegebenen Datenbereiche und gibt eine Eins zurück, wenn die Werte im ersten Dataset kleiner-gleich den Werten im zweiten Dataset sind, andernfalls eine Null. Sie gibt für jede Eingabespalte eine neue Spalte zurück, die jeweils die entsprechende Spalte in `data1` im Vergleich zur entsprechenden Spalte von `data2` enthält (d. h. die erste Spalte von `data1` wird mit der ersten Spalte von `data` verglichen, die zweite Spalte mit der zweiten Spalte usw.).

Wenn es sich bei `data2` um eine Konstante handelt, wird jeder Wert in `data` mit dem betreffenden Wert verglichen. Wenn es sich bei `data2` um eine Spalte handelt, werden die Berechnungen zeilenweise ausgeführt. Die erste Zeile von `data1` wird durch den Wert der ersten Zeile von `data2` dividiert, die zweite Zeile durch die zweite Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

Anmerkung: Der Operator `LE` kann durch ein Kleiner-als-Zeichen, auf das ein Gleichheitszeichen folgt, abgekürzt werden (<=).

Beispiele

`TEMP = 4 LE 4` oder `TEMP = 4 <= 4`

Erstellt eine neue Spalte `TEMP`, die den Wert eins enthält (vier ist gleich sich selbst).

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

Erstellt eine neue Spalte `TEMP`, in der jeder Wert eins ist, wenn der entsprechende Zeilenwert von Spalte `V1` kleiner-gleich der Zahl acht ist; andernfalls ist der Wert null.

```
TEMP = V1:V3 <= 2
```

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Werte in der Spalte `TEMP` entsprechen dem Inhalt der Spalte `V1` im Vergleich zum Wert zwei, die Werte der Spalte `VX` entsprechen dem Inhalt der Spalte `V2` im Vergleich zum Wert zwei, und die Werte der Spalte `VY` entsprechen dem Inhalt der Spalte `V3` im Vergleich zum Wert zwei.

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

Erstellt eine neue Spalte TEMP, die nur Einsen enthält (jede Zahl ist gleich sich selbst).

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

Erstellt eine neue Spalte TEMP, bei der jeder Wert der Zeilenwert der Spalte V1 im Vergleich zum entsprechenden Zeilenwert der Spalte V2 ist.

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

Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen die Ergebnisse des Vergleichs der Werte in den Zeilen 10-20 der Spalte V1 mit den Werten in den Zeilen 1-11 der Spalte V2 enthalten. Die anderen Zellen in TEMP sind leer.

Zugehörige Funktionen

Funktion	Syntax
EQ	Gibt TRUE zurück, wenn der eine Datenbereich dem anderen gleich ist.
GE	Gibt TRUE zurück, wenn der eine Datenbereich größer-gleich dem anderen ist.
GT	Gibt TRUE zurück, wenn der eine Datenbereich größer als der andere ist.
LT	Gibt TRUE zurück, wenn der eine Datenbereich kleiner als der andere ist.
NE	Gibt TRUE zurück, wenn der eine Datenbereich dem anderen ungleich ist.

LIKE-Makro

Die LIKE Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

```
data1 [NOT] LIKE data2
```

Parameter

```
data1
```

Der Zellenbereich, der verglichen werden soll. Dabei kann es sich um eine Zeichenfolge oder um einen Ausdruck handeln, dessen Auswertung eine Zeichenfolge ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

```
data2
```

Das Textmuster, mit dem alle Werte in der angegebenen Spalte verglichen werden sollen. Dabei kann es sich um eine Zeichenfolge oder um einen Ausdruck handeln, dessen Auswertung eine Zeichenfolge ergibt. Die Anzahl der Spalten in `data2` muss mit der Anzahl der Spalten in `data1` übereinstimmen, es sei denn, bei `data2` handelt es sich um eine Konstante. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Ein Unterstrich (_) in `data2` stellt ein Platzhalterzeichen dar, das mit einem beliebigen einzelnen Zeichen in `data1` übereinstimmt. Ein Prozentzeichen (%) stimmt mit null oder mehr Zeichen in `data1`überein.

Syntax

`LIKE` vergleicht die beiden angegebenen Datenbereiche und gibt eine Eins zurück, wenn die Zeichenfolgen übereinstimmen, bzw. eine Null, wenn die Zeichenfolgen nicht übereinstimmen. Sie gibt für jede Eingabespalte eine neue Spalte zurück, die jeweils die entsprechende Spalte in `data1` im Vergleich zur entsprechenden Spalte von `data2` enthält (d. h. die erste Spalte von `data1` wird mit der ersten Spalte von `data2` verglichen, die zweite Spalte mit der zweiten Spalte usw.).

Wenn es sich bei `data2` um eine Zeichenfolgekonstante handelt, wird jede Zeichenfolge in `data1` mit der betreffenden Zeichenfolge verglichen. Wenn es sich bei `data2` um eine Spalte handelt, werden die Berechnungen zeilenweise ausgeführt. Die Zeichenfolge der ersten Zeile von `data1` wird mit der Zeichenfolge der ersten Zeile von `data2` verglichen, die Zeichenfolge der zweiten Zeile mit der Zeichenfolge der zweiten Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zur letzten Zeichenfolge der kürzesten Spalte.

Beim Vergleich von Zeichenfolgen spielt die Groß-/Kleinschreibung keine Rolle (d. h., "Ja", "ja", "JA" und "jA" werden gleich behandelt).

Anmerkung: Das Makro LIKE verfügt über eine negative Version, NOT LIKE. Deren Format ist mit dem von LIKE identisch. NOT LIKE gibt eine Eins zurück, wenn die Zeichenfolge in `data1` nicht mit der durch `data2` definierten Schablone übereinstimmt.

Beispiele

```
TEMP = "gold" LIKE "gold"
```

Erstellt eine neue Spalte TEMP, die den Wert eins enthält (die beiden Zeichenfolgen stimmen überein).

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

Erstellt eine neue Spalte TEMP, die den Wert eins enthält (beim Vergleich von Zeichenfolgen wird die Groß-/Kleinschreibung nicht beachtet).

```
TEMP = V1 LIKE "gold%"
```

Erstellt eine neue Spalte TEMP, in der jeder Wert eins ist, wenn der entsprechende Zeilenwert von Spalte V1 mit der Zeichenfolge "Gold", auf die eine beliebige Anzahl von Zeichen folgt, übereinstimmt. Andernfalls ist jeder Wert null.

```
TEMP = V1 LIKE "g_ld"
```

Erstellt eine neue Spalte TEMP, in der jeder Wert eins ist, wenn der entsprechende Zeilenwert von Spalte V1 mit der Zeichenfolge aus "G", einem beliebigen Zeichen und den Zeichen "ld" übereinstimmt. Andernfalls ist jeder Wert null.

```
TEMP = V1 LIKE V1
```

Erstellt eine neue Spalte TEMP, die nur Einsen enthält (jede Zahl ist gleich sich selbst).

```
TEMP = V1 LIKE V2
```

Erstellt eine neue Spalte TEMP, bei der jeder Wert der Zeilenwert der Spalte V1 im Vergleich zum entsprechenden Zeilenwert der Spalte V2 ist.

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

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. In der Spalte TEMP sind die Zeichenfolgen von Spalte V1 mit den entsprechenden Zeichenfolgen von Spalte V4 verglichen. Die Spalte VX enthält den Vergleich von Spalte V2 mit Spalte V5 Die Spalte VY enthält den Vergleich von Spalte V3 mit Spalte V6

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

Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen die Ergebnisse des Vergleichs der Zeichenfolgen in den Zeilen 10-20 der Spalte V1 mit den Zeilen 1-11 der Spalte V2 enthalten. Die anderen Zellen in TEMP sind leer.

Zugehörige Funktionen

Funktion	Syntax
EQ	Gibt TRUE zurück, wenn der eine Datenbereich dem anderen gleich ist.
GE	Gibt TRUE zurück, wenn der eine Datenbereich größer-gleich dem anderen ist.
GT	Gibt TRUE zurück, wenn der eine Datenbereich größer als der andere ist.
LE	Gibt TRUE zurück, wenn der eine Datenbereich kleiner-gleich dem anderen ist.
LT	Gibt TRUE zurück, wenn der eine Datenbereich kleiner als der andere ist.
NE	Gibt TRUE zurück, wenn der eine Datenbereich dem anderen ungleich ist.

LN- oder LOG-Makro

LN oder LOG Makro ist nur in Unica Campaign verfügbar.

Syntax

LN(data) oder LOG(data)

Parameter

data

Die numerischen Werte, deren natürlicher Logarithmus berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

LN oder LOG berechnet den natürlichen Logarithmus jedes Wertes im angegebenen Datenbereich. Dabei wird für jede Eingabespalte eine neue Spalte zurückgegeben, die den natürlichen Logarithmus der Zahlen in der entsprechenden Eingabespalte enthält. Natürliche Logarithmen basieren auf der Konstante e = 2,7182818. LN ist der Kehrwert der EXP-Makrofunktion.

Anmerkung: Alle Werte im angegebenen Datenbereich müssen größer als null sein. Andernfalls wird für jede ungültige Eingabe eine leere Zelle zurückgegeben.

Beispiele

`TEMP = LN(3)` oder `TEMP = LOG(3)`

Erstellt eine neue Spalte `TEMP`, die den Wert `1.099` enthält.

```
TEMP = LN(V1)
```

Erstellt eine neue Spalte `TEMP`, in der jeder Wert der natürliche Logarithmus des Inhalts der Spalte `V1` ist.

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

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Werte in der `TEMP`-Spalte sind die natürlichen Logarithmen des Inhalts der Spalte `V1`, die Werte in der Spalte `VX` sind die natürlichen Logarithmen des Inhalts der Spalte `V2`, und die Werte in der Spalte `VY` sind die natürlichen Logarithmen des Inhalts der Spalte `V3`.

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

Erstellt eine neue Spalte `TEMP`, in der die ersten 11 Zellen die natürlichen Logarithmen der Werte in den Zeilen 10-20 von Spalte `V1` enthalten. Die anderen Zellen in `TEMP` sind leer.

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

Erstellt zwei neue Spalten `TEMP` und `VX`, die jeweils Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte `TEMP` sind die natürlichen Logarithmen der entsprechenden Zeilen von Spalte `V1`, die Werte in der Spalte `VX` sind die natürlichen Logarithmen der entsprechenden Zeilen von Spalte `V2`.

Zugehörige Funktionen

Funktion	Syntax
`EXP`	Erhebt die natürliche Zahl (e) in die Potenz, die durch den Inhalt jeder Zelle im angegebenen Datenbereich angegeben ist.
`LOG2`	Berechnet die Protokollbasis2 des Inhalts des angegebenen Datenbereichs
`LOG10`	Berechnet die Protokollbasis10 des Inhalts des angegebenen Datenbereichs
`POW`	Erhebt einen Basiswert in die Potenz des oder der angegebenen Exponenten.

LOG2-Makro

Die LOG2 Makro ist nur in Unica Campaign verfügbar.

Syntax

```
LOG2(data)
```

Parameter

data

Die numerischen Werte, deren binärer Logarithmus berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

LOG2 berechnet den binären Logarithmus der Werte im angegebenen Datenbereich. Dabei wird für jede Eingabespalte eine neue Spalte zurückgegeben, die den binären Logarithmus der Zahlen in der entsprechenden Eingabespalte enthält.

Anmerkung: Alle Werte im angegebenen Datenbereich müssen größer als null sein. Andernfalls wird für jede ungültige Eingabe eine leere Zelle zurückgegeben.

Beispiele

```
TEMP = LOG2(8)
```

Erstellt eine neue Spalte TEMP, die den Wert drei enthält.

```
TEMP = LOG2(V1)
```

Erstellt eine neue Spalte TEMP, in der jeder Wert der binäre Algorithmus des Inhalts der Spalte V1 ist.

```
TEMP = LOG2(V1:V3)
```

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Die Werte in der TEMP-Spalte sind die binären Algorithmen des Inhalts der Spalte V1, die Werte der Spalte VX sind die die

binären Algorithmen des Inhalts der Spalte V2, und die Werte der Spalte VY sind die binären Algorithmen des Inhalts der Spalte V3.

```
TEMP = LOG2(V1[10:20])
```

Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen die binären Algorithmen der Werte in den Zeilen 10-20 von Spalte V1 enthalten. Die anderen Zellen in TEMP sind leer.

```
TEMP = LOG2(V1[1:5]:V2)
```

Erstellt zwei neue Spalten TEMP und VX, die jeweils Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind die natürlichen Logarithmen der entsprechenden Zeilen von Spalte V1, die Werte in der Spalte VX sind die binären Logarithmen der entsprechenden Zeilen von Spalte V2.

Zugehörige Funktionen

Funktion	Syntax
LN oder LOG	Berechnet den natürlichen Logarithmus des Inhalts des angegebenen Datenbereichs.
LOG10	Berechnet die Protokollbasis10 des Inhalts des angegebenen Datenbereichs
POW	Erhebt einen Basiswert in die Potenz des angegebenen Exponenten.

LOG10-Makro

Die LOG10 Makro ist nur in Unica Campaign verfügbar.

Syntax

```
LOG10(data)
```

Parameter

```
data
```

Die numerischen Werte, deren dekadischer Logarithmus berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition

von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`LOG10` berechnet den dekadischen Logarithmus der Werte im angegebenen Datenbereich. Dabei wird für jede Eingabespalte eine neue Spalte zurückgegeben, die den dekadischen Logarithmus der Zahlen in der entsprechenden Eingabespalte enthält.

Anmerkung: Alle Werte im angegebenen Datenbereich müssen größer als null sein. Andernfalls wird für jede ungültige Eingabe eine leere Zelle zurückgegeben.

Beispiele

```
TEMP = LOG10(100)
```

Erstellt eine neue Spalte `TEMP`, die den Wert zwei enthält.

```
TEMP = LOG10(V1)
```

Erstellt eine neue Spalte `TEMP`, in der jeder Wert der dekadische Logarithmus des Inhalts der Spalte `V1` darstellt.

```
TEMP = LOG10(V1:V3)
```

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Werte in der `TEMP`-Spalte sind die dekadischen Logarithmen des Inhalts der Spalte `V1`, die Werte der Spalte `VX` sind die dekadischen Logarithmen des Inhalts der Spalte `V2`, und die Werte der Spalte `VY` sind die dekadischen Logarithmen des Inhalts der Spalte `V3`.

```
TEMP = LOG10(V1[10:20])
```

Erstellt eine neue Spalte `TEMP`, in der die ersten 11 Zellen die dekadischen Logarithmen der Werte den in den Zeilen 10-20 von Spalte `V1` enthalten. Die anderen Zellen in `TEMP` sind leer.

```
TEMP = LOG10(V1[1:5]:V2)
```

Erstellt zwei neue Spalten `TEMP` und `VX`, die jeweils Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte `TEMP` sind die dekadischen Logarithmen der entsprechenden Zeilen von Spalte `V1`, die Werte in der Spalte `VX` sind die dekadischen Logarithmen der entsprechenden Zeilen von Spalte `V2`.

Zugehörige Funktionen

Funktion	Syntax
LN oder LOG	Berechnet den natürlichen Logarithmus des Inhalts des angegebenen Datenbereichs.
LOG2	Berechnet die Protokollbasis2 des Inhalts des angegebenen Datenbereichs
POW	Erhebt einen Basiswert in die Potenz des angegebenen Exponenten.

LOWER-Makro

Die LOWER Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

```
LOWER(data)
```

Parameter

`data`

Der Zeichenfolgewert, der in Kleinbuchstaben konvertiert werden soll.

Syntax

LOWER konvertiert jeden Zeichenfolgewert im angegebenen Datenbereich in Kleinbuchstaben. Dabei wird eine neue Spalte zurückgegeben, in der jede Zelle die in Kleinbuchstaben umgewandelte Zeichenfolge der entsprechenden Eingangszelle enthält.

Beispiele

```
Temp = LOWER "GOLD"
```

Erstellt eine neue Spalte TEMP, die den Wert "gold" enthält.

```
TEMP = LOWER( "JAN 15, 1997")
```

Erstellt eine neue Spalte TEMP, die die ASCII-Zeichenfolge "jan 15, 1997" enthält.

```
TEMP = LOWER( "Pressure")
```

Erstellt eine neue Spalte TEMP, die die ASCII-Zeichenfolge "pressure" enthält.

```
TEMP = LOWER(V1)
```

Erstellt eine neue Spalte `TEMP`, die die Kleinbuchstaben jeder Zeichenfolge in der Spalte `V1` enthält.

LT-Makro

Die `LT` Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

```
data1 LT data2 data1 < data2
```

Parameter

`data1`

Der numerische Zellenbereich, der verglichen werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`data2`

Die Zahlen, mit denen alle Werte in der angegebenen Spalte verglichen werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`LT` vergleicht die beiden angegebenen Datenbereiche und gibt eine Eins zurück, wenn die Werte im ersten Dataset kleiner als die Werte im zweiten Dataset sind, andernfalls eine Null. Sie gibt für jede Eingabespalte eine neue Spalte zurück, die jeweils die entsprechende Spalte in `data1` im Vergleich zur entsprechenden Spalte von `data2` enthält (d. h. die erste Spalte von `data1` wird mit der ersten Spalte von `data` verglichen, die zweite Spalte mit der zweiten Spalte usw.).

Wenn es sich bei `data2` um eine Konstante handelt, wird jeder Wert in `data` mit dem betreffenden Wert verglichen. Wenn es sich bei `data2` um eine Spalte handelt, werden die Berechnungen zeilenweise ausgeführt. Die erste Zeile von `data1` wird durch den Wert der ersten Zeile von `data2` dividiert, die zweite Zeile durch die zweite Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

Anmerkung: Der Operator `LT` kann durch ein Kleiner-als-Zeichen abgekürzt werden (<).

Beispiele

`TEMP = 3 LT 4` oder `TEMP = 3 < 4`

Erstellt eine neue Spalte `TEMP`, die den Wert Null enthält (da drei kleiner als vier ist).

`TEMP = V1 < 8`

Erstellt eine neue Spalte `TEMP`, in der jeder Wert Eins ist, wenn der entsprechende Zeilenwert der Spalte `V1` kleiner als die Zahl Acht ist, andernfalls ist der Wert Null.

`TEMP = V1:V3 < 2`

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Werte in der Spalte `TEMP` entsprechen dem Inhalt der Spalte `V1` im Vergleich zum Wert zwei, die Werte der Spalte `VX` entsprechen dem Inhalt der Spalte `V2` im Vergleich zum Wert zwei, und die Werte der Spalte `VY` entsprechen dem Inhalt der Spalte `V3` im Vergleich zum Wert zwei.

`TEMP = V1 < V1`

Erstellt eine neue Spalte `TEMP`, die nur Nullen enthält (da keine Zahl kleiner ist als sie selbst).

`TEMP = V1 < V2`

Erstellt eine neue Spalte `TEMP`, bei der jeder Wert der Zeilenwert der Spalte `V1` im Vergleich zum entsprechenden Zeilenwert der Spalte `V2` ist.

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

Erstellt eine neue Spalte `TEMP`, in der die ersten 11 Zellen die Ergebnisse des Vergleichs der Werte in den Zeilen 10-20 der Spalte `V1` mit den Zeilen 1-11 der Spalte `V2` enthalten. Die anderen Zellen in `TEMP` sind leer.

Zugehörige Funktionen

Funktion	Syntax
EQ	Gibt TRUE zurück, wenn der eine Datenbereich dem anderen gleich ist.
GE	Gibt TRUE zurück, wenn der eine Datenbereich größer-gleich dem anderen ist.
GT	Gibt TRUE zurück, wenn der eine Datenbereich größer als der andere ist.
LE	Gibt TRUE zurück, wenn der eine Datenbereich kleiner-gleich dem anderen ist.

LTRIM-Makro

Die `LTRIM` Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

```
LTRIM(data)
```

Parameter

`data`

Die Zeichenfolge, aus der führende Leerzeichen entfernt werden sollen.

Syntax

`LTRIM` entfernt führende Leerzeichen aus jedem Zeichenfolgewert im angegebenen Datenbereich und gibt die konvertierte Zeichenfolge zurück. RTRIM gibt für jede Eingabespalte eine neue Spalte zurück.

Beispiele

```
Temp = LTRIM " gold"
```

Erstellt eine neue Zeichenfolge `Temp`, die `"gold"` enthält.

MAX-Makro

Die `MAX` Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

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

Parameter

`data`

Die numerischen Werte, deren Maximum berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`keyword`

Dieses optionale Schlüsselwort legt fest, wie die Berechnung auf den Eingabedatenbereich angewendet wird. Eine der folgenden Optionen auswählen:

`ALL` - Wendet die Berechnung auf alle Zellen in `data` an (Standard)

`COL` - Führt die Berechnung für jede Zeile von data gesondert aus `data`

`ROW` - Führt die Berechnung für jede Spalte von data gesondert aus `data`

Weitere Informationen zur Verwendung von Schlüsselwörtern in Unica Campaign finden Sie in Formatspezifikationen (auf Seite 12).

Anmerkung: Viele Makrofunktionen nehmen die Schlüsselwortparameter `{ALL | COL | ROW}` an. Diese Schlüsselwörter gelten nicht für Unica Campaign, da es sich bei den Eingabedaten immer um eine Einzelspalte oder ein Einzelfeld handelt. Das Makro verhält sich immer so, als ob das Schlüsselwort `COL`angegeben würde. Deshalb brauchen Sie diese Schlüsselwörter nicht anzugeben, wenn Sie Unica Campaign verwenden.

Syntax

`MAX` berechnet das Maximum der Werte im angegebenen Datenbereich. Es wird eine einzelne neue Spalte zurückgegeben, die den Maximalwert enthält.

Beispiele

`TEMP = MAX(3)` oder `TEMP = MAX(3, ALL)`

Erstellt eine neue Spalte TEMP, die den Wert drei enthält.

```
TEMP = MAX(V1)
```

Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der den Maximalwert des Inhalts von Spalte V1 darstellt.

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

Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der das Maximum des Inhalts der Spalten V1, V2 und V3 darstellt.

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

Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der das Maximum der Zellen in Zeile 10-20 von Spalte V1 darstellt.

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

Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der das Maximum der Zellen in Zeile 1-5 von Spalte V1 bis V4 darstellt.

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

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Der Einzelwert in der Spalte TEMP ist das Maximum des Inhalts von Spalte V1, der Einzelwert in der Spalte VX ist das Maximum des Inhalts von Spalte V2 und der Einzelwert in der Spalte VY ist das Maximum des Inhalts von Spalte V3.

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

Erstellt drei neue Spalten TEMP, VX und VY, die jeweils einen Einzelwert enthalten. Der Wert in der Spalte TEMP ist das Maximum der Zellen in Zeile 1-5 von Spalte V1, der Wert in der Spalte VX ist das Maximum der Zellen in Zeile 1-5 von Spalte V2, und der Wert in der Spalte VY ist das Maximum der Zellen in Zeile 1-5 von Spalte V3.

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

Erstellt eine neue Spalte TEMP, in der jeder Zelleneintrag das Maximum der entsprechenden Zeile in Spalte V1, V2 und V3darstellt.

```
TEMP = MAX(V1[10:20]:V3, ROW)
```

Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen das Maximum der Werte in den Zeilen 10-20 über die Spalten V1 bis V3 enthalten. Die anderen Zellen in TEMP sind leer.

Zugehörige Funktionen

Funktion	Syntax
MIN	Berechnet das Minimum eines Zellenbereichs

MEAN-Makro

Die MEAN Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

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

Parameter

data

Die numerischen Werte, deren arithmetisches Mittel berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

keyword

Dieses optionale Schlüsselwort legt fest, wie die Berechnung auf den Eingabedatenbereich angewendet wird. Eine der folgenden Optionen auswählen:

ALL - Wendet die Berechnung auf alle Zellen in data an (Standard)

COL - Führt die Berechnung für jede Zeile von data gesondert aus data

ROW - Führt die Berechnung für jede Spalte von data gesondert aus data

Weitere Informationen zur Verwendung von Schlüsselwörtern finden Sie unter DATUM (auf Seite 71).

Anmerkung: Viele Makrofunktionen nehmen die Schlüsselwortparameter {ALL | COL | ROW} an. Diese Schlüsselwörter gelten nicht für Unica Campaign, da es sich bei den Eingabedaten immer um eine Einzelspalte oder ein Einzelfeld handelt. Das Makro verhält

sich immer so, als ob das Schlüsselwort `COL`angegeben würde. Deshalb brauchen Sie diese Schlüsselwörter nicht anzugeben, wenn Sie Unica Campaign verwenden.

Syntax

`MEAN` Berechnet das arithmetische Mittel oder den Durchschnitt der Zellen im angegebenen Datenbereich. Das arithmetische Mittel wird berechnet, indem der Inhalt aller Zellen addiert und durch die Anzahl der Zellen geteilt wird. Die Anzahl der von `MEAN` zurückgegebenen Spalten hängt von `keyword` ab.

- Wenn `keyword` den Wert `ALL` hat, gibt `MEAN` eine neue Spalte zurück, die einen Einzelwert (den Durchschnitt aller Zellen in `data`) enthält.
- Wenn `keyword` den Wert `COL` hat, gibt `MEAN` für jede Eingabespalte eine neue Spalte zurück. Jede neue Spalte enthält einen Wert (den Durchschnitt aller Zellen in der entsprechenden Eingabespalte).
- Wenn `keyword` den Wert `ROW` hat, gibt `MEAN` eine neue Spalte zurück, die den Durchschnitt für jede Zeile von `data` enthält.

Anmerkung: Leere Zellen werden bei der Berechnung des Durchschnitts ignoriert.

Anmerkung: `MEAN` ist mit der Makrofunktion `AVG` identisch.

Beispiele

```
TEMP = MEAN(V1)
```

Erstellt eine Spalte namens `TEMP`, die einen Einzelwert enthält, der das arithmetische Mittel des Inhalts von Spalte `V1` darstellt.

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

Erstellt eine Spalte namens `TEMP`, die einen Einzelwert enthält, der das arithmetische Mittel des Inhalts von Spalte `V1`, `V2`und `V3`darstellt.

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

Erstellt eine Spalte namens `TEMP`, die einen Einzelwert enthält, der das arithmetische Mittel der Zellen in den Zeilen 10 bis 20 von Spalte `V1` darstellt.

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

Erstellt eine Spalte namens TEMP, die einen Einzelwert enthält, der das arithmetische Mittel der Zellen in den Zeilen 1 bis 5 der Spalten V1 bis V4 darstellt.

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

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Der Einzelwert in der Spalte TEMP ist das arithmetische Mittel des Inhalts von Spalte V1, der Einzelwert in der Spalte VX ist das arithmetische Mittel des Inhalts von Spalte V2 und der Einzelwert in der Spalte VY ist das arithmetische Mittel des Inhalts von Spalte V3.

```
TEMP = MEAN(V1[10:20]:V3, COL)
```

Erstellt drei neue Spalten TEMP, VX und VY, die jeweils einen Einzelwert enthalten. Der Wert in der Spalte TEMP ist das arithmetische Mittel der Zellen in Zeile 10-20 von Spalte V1, der Wert in der Spalte VX ist das arithmetische Mittel der Zellen in Zeile 10-20 von Spalte V2, und der Wert in der Spalte VY ist das arithmetische Mittel der Zellen in Zeile 10-20 von Spalte V3.

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

Erstellt eine Spalte namens TEMP, in der jeder Zelleneintrag das arithmetische Mittel der entsprechenden Zeile in den Spalten V1, V2 und V3 darstellt.

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

Erstellt eine Spalte namens TEMP, in der die Zellen in den Zeilen 1 bis 5 das arithmetische Mittel der entsprechenden Zeile in den Spalten V1 bis V3 enthalten. Die anderen Zellen in TEMP sind leer.

Zugehörige Funktionen

Funktion	Syntax
SUM oder TOTAL	Berechnet die Summe eines Zellenbereichs.

MIN-Makro

Das MIN-Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

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

Parameter

`data`

Die numerischen Werte, deren Minimum berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`keyword`

Dieses optionale Schlüsselwort legt fest, wie die Berechnung auf den Eingabedatenbereich angewendet wird. Eine der folgenden Optionen auswählen:

`ALL` - Wendet die Berechnung auf alle Zellen in `data` an (Standard)

`COL` - Führt die Berechnung für jede Zeile von data gesondert aus `data`

`ROW` - Führt die Berechnung für jede Spalte von data gesondert aus `data`

Weitere Informationen zur Verwendung von Schlüsselwörtern finden Sie unter DATUM (auf Seite 71).

Anmerkung: Viele Makrofunktionen nehmen die Schlüsselwortparameter `{ALL | COL | ROW}` an. Diese Schlüsselwörter gelten nicht für **Unica Campaign** , da es sich bei den Eingabedaten immer um eine Einzelspalte oder ein Einzelfeld handelt. Das Makro verhält sich immer so, als ob das Schlüsselwort `COL`angegeben würde. Deshalb brauchen Sie diese Schlüsselwörter nicht anzugeben, wenn Sie **Unica Campaign** verwenden.

Syntax

`MIN` berechnet das Minimum aller Zellen im angegebenen Datenbereich. Es wird eine einzelne neue Spalte zurückgegeben, die den Minimalwert enthält.

Beispiele

```
TEMP = MIN(V1)
```

Erstellt eine neue Spalte `TEMP`, die einen Einzelwert enthält, der den Minimalwert des Inhalts von Spalte `V1` darstellt.

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

Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der das Minimum des Inhalts der Spalten V1, V2 und V3 darstellt.

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

Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der das Minimum der Zellen in Zeile 10-20 von Spalte V1 darstellt.

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

Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der das Minimum der Zellen in Zeile 1-5 von Spalte V1 bis V4 darstellt.

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

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Der Einzelwert in der Spalte TEMP ist das Minimum des Inhalts von Spalte V1, der Einzelwert in der Spalte VX ist das Minimum des Inhalts von Spalte V2 und der Einzelwert in der Spalte VY ist das Minimum des Inhalts von Spalte V3.

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

Erstellt drei neue Spalten TEMP, VX und VY, die jeweils einen Einzelwert enthalten. Der Wert in der Spalte TEMP ist das Minimum der Zellen in Zeile 1-5 von Spalte V1, der Wert in der Spalte VX ist das Minimum der Zellen in Zeile 1-5 von Spalte V2, und der Wert in der Spalte VY ist das Minimum der Zellen in Zeile 1-5 von Spalte V3.

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

Erstellt eine neue Spalte TEMP, in der jeder Zelleneintrag das Minimum der entsprechenden Zeile in Spalte V1, V2 und V3darstellt.

```
TEMP = MIN(V1[10:20]:V3, ROW)
```

Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen das Minimum der Werte in den Zeilen 10-20 über die Spalten V1 bis V3 enthalten. Die anderen Zellen in TEMP sind leer.

Zugehörige Funktionen

Funktion	Syntax
MAX	Berechnet das Maximum eines Zellenbereichs

Funktion	Syntax
MAX_TO_INDEX	Gibt für jede Zeile der angegebenen Spalte den Spaltenindex des Maximalwerts zurück.

MINUS-Makro

Die MINUS Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

```
data MINUS subtrahend data - subtrahend
```

Parameter

data

Der Zellenbereich mit Zahlen, von denen etwas subtrahiert werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

subtrahend

Die Zahl(en), die von allen Werten in der angegebenen Spalte subtrahiert werden soll(en). Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in subtrahend muss mit der Anzahl der Spalten in data übereinstimmen, es sei denn, bei subtrahend handelt es sich um eine Konstante. Informationen zur Formatdefinition von subtrahend (dasselbe wie bei data), siehe den Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

MINUS subtrahiert subtrahend vom angegebenen Datenbereich data. MINUS gibt für jede Eingabespalte eine neue Spalte zurück, wobei jeweils die entsprechende Spalte von data subtrahend von der entsprechenden Spalte von subtrahend subtrahiert wird (d. h., von der

ersten Spalte von `data` wird die erste Spalte von `subtrahend` subtrahiert, von der zweiten Spalte die zweite Spalte usw.).

Wenn `subtrahend` eine Konstante ist, wird er von jedem Wert in `data` subtrahiert. Wenn `subtrahend` eine oder mehrere Spalten enthält, werden die Berechnungen auf Zeilenbasis für eine Spalte aus `data` und eine Spalte aus `subtrahend` durchgeführt. Die erste Zeile von `data` subtrahiert den Wert der ersten Zeile von `subtrahend`, die zweite Zeile mit der zweiten Zeile und so weiter. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

Anmerkung: Der `MINUS`-Operator kann mit einem Minuszeichen oder Bindestrich (`-`) abgekürzt werden.

Beispiele

`TEMP = 7 MINUS 4` oder `TEMP = 7 - 4`

Erstellt eine neue Spalte `TEMP`, die den Wert drei enthält.

`TEMP = V1 - 8`

Erstellt eine neue Spalte `TEMP`, in der jeder Wert den Inhalt von Spalte `V1` minus acht darstellt.

`TEMP = V1:V3 - 2`

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Werte in der Spalte `TEMP` entsprechen dem Inhalt der Spalte `V1` minus zwei, die Werte der Spalte `VX` entsprechen dem Inhalt der Spalte `V2` minus zwei und die Werte der Spalte `VY` entsprechen dem Inhalt der Spalte `V3` minus zwei.

`TEMP = V1 - V1`

Erstellt eine neue Spalte `TEMP`, die nur Nullen enthält (jede Zahl minus dieselbe Zahl ergibt null).

`TEMP = V1 - V2`

Erstellt eine neue Spalte `TEMP`, in der jeder Wert der Zeilenwert der Spalte `V1` minus dem entsprechenden Zeilenwert der Spalte `V2`ist.

`TEMP = V1:V3 -V4:V6`

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Die Spalte TEMP enthält die Werte in V1 minus die entsprechenden Zeilenwerte der Spalte V4. Die Spalte VX subtrahiert Spalte V5 von Spalte V2. Die Spalte VY subtrahiert Spalte V6 von Spalte V3.

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

Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen die Werte in den Zeilen 10-20 der Spalte V1 minus die Werte in den Zeilen 1-11 der Spalte V2 enthalten. Die anderen Zellen in TEMP sind leer.

Zugehörige Funktionen

Funktion	Syntax
PLUS	Addiert die Inhalte zweier Datenbereiche.
SUM oder TOTAL	Berechnet die Summe eines Zellenbereichs.

MOD-Makro

Die MOD Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

```
data MOD divisor data % divisor
```

Parameter

`data`

Die Ganzzahlwerte, deren Modulo-Wert berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`divisor`

Die Ganzzahl ungleich null, bezüglich deren der Modulo-Wert berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in

`divisor` muss mit der Anzahl der Spalten in `data` übereinstimmen, es sei denn, bei `divisor` handelt es sich um eine Konstante. Informationen zur Formatdefinition von `divisor` (dasselbe wie bei `data`), siehe den Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`MOD` MOD berechnet den Rest der Division des angegebenen Datenbereichs durch einen angegebenen Wert. MOD wird berechnet, indem der `divisor` durch den jeweils angegebenen Wert dividiert wird, und gibt den Rest zurück. Dabei wird für jede Eingabespalte eine neue Spalte zurückgegeben, die jeweils die Zahlen in `data` Modulo `divisor` enthält. Der Rest hat dasselbe Vorzeichen (positiv oder negativ) wie `data`.

Wenn es sich bei `divisor` um eine Konstante handelt, wird für jeden Wert in der angegebenen Spalte der betreffende Modulo-Wert berechnet. Wenn es sich bei `divisor` um eine Spalte handelt, werden die Berechnungen zeilenweise ausgeführt. Bei der Modulo-Berechnung wird die erste Zeile in `data` dem ersten Zeilenwert von `divisor` zugeordnet, die zweite Zeile dem zweiten Zeilenwert usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

Anmerkung: Wenn `divisor` null ist, wird der Fehler 'Division durch null' zurückgegeben.

Anmerkung: Der `MOD`-Operator kann mit dem Prozentzeichen (`%`) abgekürzt werden. Zum Beispiel ist `TEMP = 5 % 3` gleich `TEMP = 5 MOD 3`.

Beispiele

`TEMP = 10 MOD 8` oder `TEMP = 10 % 8`

Erstellt eine neue Spalte `TEMP`, die den Wert `2` enthält.

```
TEMP = -10 % 8
```

Erstellt eine neue Spalte `TEMP`, die den Wert `-2` enthält.

```
TEMP = V1 % 8
```

Erstellt eine neue Spalte `TEMP`, in der jeder Wert der Inhalt der Spalte `V1`, Modulo-Wert acht ist.

```
TEMP = V1:V3 % 2
```

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Die Werte in der Spalte TEMP sind die Werte modulo zwei des Inhalts der Spalte V1, die Werte der Spalte VX sind die Werte modulo zwei des Inhalts der Spalte V2, und die Werte der Spalte VY sind die Werte modulo zwei des Inhalts der Spalte V3

```
TEMP = V1 % V1
```

Erstellt eine neue Spalte TEMP, die für jeden Eintrag in Spalte V1eine Null enthält. Das liegt daran, dass jede Zahl modulo selbst Null ist.

```
TEMP = V1 % V2
```

Erstellt eine neue Spalte TEMP, in der jeder Wert der Zeilenwert der Spalte V1 modulo der entsprechende Zeilenwert der Spalte V2 ist. Wenn V2=V1, werden nur Nullen zurückgegeben, wie im vorherigen Beispiel.

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

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Die Spalte TEMP enthält die Werte in V1 Modulo die entsprechenden Zeilenwerte der Spalte V4. Die Spalte VX enthält die Ergebnisse von Spalte V2modulo V5. Die Spalte VY enthält die Ergebnisse von Spalte V3modulo V6.

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

Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen die Werte in den Zeilen 10-20 der Spalte V1 modulo die Werte in den Zeilen 1-11 der Spalte V2 enthalten. Die anderen Zellen in TEMP sind leer.

Zugehörige Funktionen

Funktion	Syntax
DIV	Dividiert einen angegebenen Datenbereich durch einen anderen.
MOD	Berechnet den Modulo-Wert des Inhalts des angegebenen Datenbereichs.

MONTHOF-Makro

Die MONTHOF Makro ist nur in Unica Campaign verfügbar.

Syntax

`MONTHOF(date_string [, input_format])`

Parameter

`date_string`

Ein Text, der ein gültiges Datum darstellt.

`input_format`

Eines der Schlüsselwörter in der folgenden Tabelle, das das Datumsformat von `date_string` angibt.

Syntax

`MONTHOF` gibt für das durch `date_string` angegebene Datum den Monat als Zahl zurück. Wenn `input_format` nicht angegeben ist, wird das Standardschlüsselwort `DELIM_M_D_Y` verwendet.

Beispiele

`MONTHOF("012171",MMDDYY)` gibt die Zahl 1 zurück.

Anmerkung: Weitere Informationen zu gültigen Datumsformaten finden Sie unter DATUM *(auf Seite 71)*.

Zugehörige Funktionen

Funktion	Syntax
`DAYOF`	Gibt den Wochentag als Zahl zurück.
`WEEKDAYOF`	Gibt den Wochentag der Woche als Zahl zurück.
`YEAROF`	Gibt das Jahr als Zahl zurück.

MULT-Makro

Die `MULT` Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

```
data MULT multiplier data * multiplier
```

Parameter

`data`

Die numerischen Werte, die multipliziert werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`multiplier`

Die Zahl, mit der alle Werte in der angegebenen Spalte multipliziert werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in `multiplier` muss mit der Anzahl der Spalten in `data` übereinstimmen, es sei denn, bei `multiplier` handelt es sich um eine Konstante. Informationen zur Formatdefinition von `multiplier` (dasselbe wie bei `data`), siehe den Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`MULT` MULT multipliziert die Werte in den beiden angegebenen Datenbereichen. MULT gibt für jede Eingabespalte eine neue Spalte zurück, die die mit `data`multiplier`multiplier` multiplizierten Zahlen in data enthält. Wenn es sich bei `multiplier` um eine Konstante handelt, wird jeder Wert in `data` mit dem betreffenden Wert multipliziert. Wenn es sich bei `multiplier` um eine Spalte handelt, werden die Berechnungen zeilenweise ausgeführt. Die erste Zeile von `data` wird mit der ersten Zeile von `multiplier`multipliziert, die zweite Zeile mit der zweiten Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

Anmerkung: Wenn eine Spalte in jeder Zeile dieselbe Zahl *x* wie `multiplier` enthält, ist dies dasselbe, als wenn als *x* die Konstante `multiplier` verwendet wird.

Anmerkung: Der `MULT`-Operator kann mit einem Sternchen (`*`) abgekürzt werden.

Beispiele

`TEMP = 8 MULT 4` oder `TEMP = 8 * 4`

Erstellt eine neue Spalte `TEMP`, die den Wert `32` enthält.

`TEMP = V1 * 8`

Erstellt eine neue Spalte `TEMP`, in der jeder Wert der Inhalt der Spalte `V1` multipliziert mit acht ist.

`TEMP = V1:V3 * 2`

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Werte in der `TEMP`-Spalte sind das Zweifache des Inhalts der Spalte `V1`, die Werte der `VX`-Spalte sind das Zweifache des Inhalts der Spalte `V2`, und die Werte der `VY`-Spalte sind das Zweifache des Inhalts der Spalte `V3`.

`TEMP = V1 * V1`

Erstellt eine neue Spalte mit dem Namen `TEMP`, die das Quadrat jedes Wertes in der Spalte `V1` enthält.

`TEMP = V1 * V2`

Erstellt eine neue Spalte `TEMP`, in der jeder Wert der Zeilenwert der Spalte `V1` multipliziert mit dem entsprechenden Zeilenwert der Spalte `V2` ist.

`TEMP = V1:V3 * V4:V6`

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Spalte `TEMP` enthält die Werte in `V1` multipliziert mit den entsprechenden Zeilenwerten der Spalte `V4`. Die Spalte `VX` enthält die Multiplikation von Spalte `V2` mit Spalte `V5`. Die Spalte `VY` enthält die Multiplikation von Spalte `V3` mit Spalte `V6`.

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

Erstellt eine neue Spalte `TEMP`, in der die ersten 11 Zellen die Werte in den Zeilen 10-20 der Spalte `V1` multipliziert mit den Werten in den Zeilen 1-11 der Spalte `V2` enthalten. Die anderen Zellen in `TEMP` sind leer.

Zugehörige Funktionen

Funktion	Syntax
DIV	Dividiert einen angegebenen Datenbereich durch einen anderen.
EXP	Erhebt die natürliche Zahl (e) in die Potenz, die durch den Inhalt jeder Zelle im angegebenen Datenbereich angegeben ist.
POW	Erhebt einen Basiswert in die Potenz des oder der angegebenen Exponenten.

NE-Makro

Die NE Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

```
data1 NE data2 data1 != data2 data1 <> data2
```

Parameter

data1

Der Zellenbereich, der verglichen werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

data2

Die Zahlen, mit denen alle Werte in der angegebenen Spalte verglichen werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in data2 muss mit der Anzahl der Spalten in data1 übereinstimmen, es sei denn, bei data 2 handelt es sich um eine Konstante. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

NE NE vergleicht die beiden angegebenen Datenbereiche und gibt eine Eins zurück, wenn die Werte ungleich sind, bzw. eine Null, wenn die Werte gleich sind. Sie gibt für jede Eingabespalte eine neue Spalte zurück, die jeweils die entsprechende Spalte in `data1` im Vergleich zur entsprechenden Spalte von `data2` enthält (d. h. die erste Spalte von `data1` wird mit der ersten Spalte von `data` verglichen, die zweite Spalte mit der zweiten Spalte usw.).

Wenn es sich bei `data2` um eine Konstante handelt, wird jeder Wert in `data1` mit dem betreffenden Wert verglichen. Wenn es sich bei `data2` um eine Spalte handelt, werden die Berechnungen zeilenweise ausgeführt. Die erste Zeile von `data1` wird mit der ersten Zeile von `data2` verglichen, die zweite Zeile mit der zweiten Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

Anmerkung: Wenn eine Spalte in jeder Zeile dieselbe Zahl *x* wie `data2` enthält, ist dies dasselbe, als wenn als *x* die Konstante `data2` verwendet wird.

Anmerkung: Der Operator `NE` kann durch die Folge aus Ausrufezeichen und Gleichheitszeichen (`!=`) oder durch die Folge aus Kleiner-als-Zeichen und Größer-als-Zeichen (<>) abgekürzt werden.

Beispiele

`TEMP = 3 NE 4` oder `TEMP = 3 != 4` `TEMP = 3 <> 4`

Erstellt eine neue Spalte `TEMP`, die den Wert Eins enthält (drei ist ungleich vier).

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

Erstellt eine neue Spalte `TEMP`, in der jeder Wert eins ist, wenn der entsprechende Zeilenwert der Spalte `V1` nicht gleich der Zahl acht ist, andernfalls ist der Wert null.

```
TEMP = V1:V3 != 2
```

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Werte in der Spalte `TEMP` entsprechen dem Inhalt der Spalte `V1` im Vergleich zum Wert zwei, die Werte der Spalte `VX` entsprechen dem Inhalt der Spalte `V2` im Vergleich zum Wert zwei, und die Werte der Spalte `VY` entsprechen dem Inhalt der Spalte `V3` im Vergleich zum Wert zwei.

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

Erstellt eine neue Spalte TEMP, die nur Nullen enthält (jede Zahl ist gleich sich selbst).

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

Erstellt eine neue Spalte TEMP, bei der jeder Wert der Zeilenwert der Spalte V1 im Vergleich zum entsprechenden Zeilenwert der Spalte V2 ist.

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

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Die Spalte TEMP enthält die Werte in V1 im Vergleich zu den entsprechenden Zeilenwerten der Spalte V4. Die Spalte VX enthält den Vergleich von Spalte V2 mit Spalte V5 Die Spalte VY enthält den Vergleich von Spalte V3 mit Spalte V6

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

Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen die Ergebnisse des Vergleichs der Werte in den Zeilen 10-20 der Spalte V1 mit den Zeilen 1-11 der Spalte V2 enthalten. Die anderen Zellen in TEMP sind leer.

Zugehörige Funktionen

Funktion	Syntax
EQ	Gibt TRUE zurück, wenn der eine Datenbereich dem anderen gleich ist.
GE	Gibt TRUE zurück, wenn der eine Datenbereich größer-gleich dem anderen ist.
GT	Gibt TRUE zurück, wenn der eine Datenbereich größer als der andere ist.
LE	Gibt TRUE zurück, wenn der eine Datenbereich kleiner-gleich dem anderen ist.
LT	Gibt TRUE zurück, wenn der eine Datenbereich kleiner als der andere ist.

NOT-Makro

Die NOT Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

```
NOT(data) ! data
```

Parameter

```
data
```

Die numerischen Werte, deren logisches Nicht berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`NOT` NOT gibt das logische Nicht der Werte im angegebenen Datenbereich zurück. NOT gibt für jede Eingabespalte eine neue Spalte zurück, die das logische Nicht der Werte in der entsprechenden Eingabespalte enthält. Diese Funktion gibt bei Werten ungleich null eine Null und bei Werten gleich null eine Eins zurück.

Anmerkung: Der Operator `NOT` kann mit einem Ausrufezeichen (`!`) abgekürzt werden. Geben Sie das Ausrufezeichen vor dem Datenwert an. Beispiel: Zur Angabe von `NOT(V1)` können Sie einfach `!V1` eingeben.

Beispiele

`TEMP = NOT(3.2)` oder `TEMP = !1`

Erstellt eine neue Spalte `TEMP`, die den Wert Null enthält.

`TEMP = !0` oder `TEMP = !(2+2=3)`

Erstellt eine neue Spalte `TEMP`, die den Wert Eins enthält.

```
TEMP = !V1
```

Erstellt eine neue Spalte `TEMP`, in der jeder Wert das logische Nicht der Werte in Spalte `V1` darstellt.

```
TEMP = !V1:V3
```

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Werte in der Spalte `TEMP` sind jeweils das logische Nicht der Werte in Spalte `V1`, die Werte von Spalte `VX` sind das logische Nicht der Werte in Spalte `V2` und die Werte von Spalte `VY` sind das logische Nicht der Werte in Spalte `V3`.

```
TEMP = !V1[10:20]
```

Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen die logischen NICHT-Werte der Werte in den Zeilen 10-20 von Spalte V1 enthalten. Die anderen Zellen in TEMP sind leer.

```
TEMP = !V1[1:5]:V2
```

Erstellt zwei neue Spalten TEMP und VX, die jeweils Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind die logischen NICHT-Werte der entsprechenden Zeilen von Spalte V1, die Werte in der Spalte VX sind die logischen NICHT-Werte der entsprechenden Zeilen von Spalte V2.

Zugehörige Funktionen

Funktion	Syntax
AND	Berechnet das logische Und zwischen den beiden angegebenen Datenbereichen.
INVERSE	Berechnet den Negativwert des Inhalts des angegebenen Datenbereichs.
OR	Berechnet das logische Oder zwischen den beiden angegebenen Datenbereichen.
SIGN	Berechnet das Vorzeichen (positiv oder negativ) der Werte im angegebenen Datenbereich.

NUMBER-Makro

Das NUMBER-Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

```
NUMBER(data [, conversion_keyword])
```

Parameter

`data`

Die ASCII-Textdaten, die in numerische Werte konvertiert werden sollen. Dabei kann es sich um ASCII-Text in Anführungszeichen, eine Textspalte, einen Zellenbereich mit Text oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`conversion_keyword`

Dieses optionale Schlüsselwort legt fest, wie Textformate für Datums- und Uhrzeitangaben interpretiert werden sollen. Wählen Sie eines der Schlüsselwörter in der folgenden Tabelle aus.

Anmerkung: Wird dieser Parameter nicht angegeben, ist der Standard `1`.

Schlüsselwort	Format	Syntax
`0`	`#####`	Konvertiert die ersten 5 Zeichen jeder Zeichenfolge in eine eindeutige Zahl
`1`	`$ ( default)`	Konvertiert Dollarwerte in numerische Werte (z. B. `"$123.45"` in `123.45`)
`2`	`%`	Konvertiert Prozentwerte in numerische Werte (z. B. `"50%"` in `0.5`)
`3`	`mm/dd/yy hh:mm`	Konvertiert ein Datum und eine Uhrzeit in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (zum Jahr `yy` wird automatisch 1900 addiert)
`4`	`dd-mmm-yy`	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (zum Jahr `yy` wird automatisch 1900 addiert)
`5`	`mm/dd/yy`	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (zum Jahr `yy` wird automatisch 1900 addiert)
`6`	`mmm-yy`	Konvertiert ein Datum in die Anzahl der Tage, die zwischen dem ersten Tag des angegebenen Monats und dem 1. Januar 0000 vergangen sind (zum Jahr `yy` wird automatisch 1900 addiert)
`7`	`dd-mmm`	Konvertiert ein Datum in die Anzahl der Tage, die seit dem Anfang des Jahres vergangen sind (z. B. `"01-FEB"` in `32`)

Schlüsselwort	Format	Syntax
8	mmm	Konvertiert eine Monatsabkürzung aus 3 Buchstaben in einen Wert zwischen 1 und 12 (z. B. "DEC" in 12)
9	{January | February | March … }	Konvertiert einen vollständigen Monatsnamen in einen Wert zwischen 1 und 12 (z. B. "March" in 3)
10	{Sun | Mon | Tue … }	Konvertiert eine Wochentagsabkürzung aus 3 Buchstaben in einen Wert zwischen 0 und 6, wobei die Woche am Sonntag beginnt (z. B. "Sun" in 0)
11	{Sunday | Monday | Tuesday … }	Konvertiert einen vollständigen Wochentagsnamen in einen Wert zwischen 0 und 6, wobei die Woche am Sonntag beginnt (z. B. "Monday" in 1)
12	hh:mm:ss {AM | PM}	Konvertiert die Uhrzeit in die Anzahl der Sekunden, die seit Mitternacht vergangen sind (z. B. "01:00:00 AM" in 3600),
13	hh:mm:ss	Konvertiert die Uhrzeit in die Anzahl der Sekunden, die seit Mitternacht vergangen sind (z. B. "01:00:00" in 3600),
14	hh:mm {AM | PM}	Konvertiert die Uhrzeit in die Anzahl der Minuten, die seit Mitternacht vergangen sind (z. B. "01:00 AM" in 60)
15	hh:mm	Konvertiert die Uhrzeit in die Anzahl der Minuten, die seit Mitternacht vergangen sind (z. B. "01:00" in 60)
16	mm:ss	Konvertiert die Uhrzeit in die Anzahl der Sekunden, die seit Mitternacht vergangen sind (z. B. "30:00" in 1800),

Schlüsselwort	Format	Syntax
17	ddmm	Konvertiert ein Datum in die Anzahl der Tage, die seit dem Anfang des Jahres vergangen sind (z. B. "3101" in 31)
18	ddmmm	Konvertiert ein Datum in die Anzahl der Tage, die seit dem Anfang des Jahres vergangen sind (z. B. "31JAN" in 31)
19	ddmmmyy	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (wenn das Jahr yy kleiner-gleich 20 ist, wird automatisch 1900 addiert; andernfalls wird 2000 addiert)
20	ddmmmyyyy	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (z. B. "31JAN0000" in 31)
21	ddmmyy	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (wenn das Jahr yy kleiner-gleich 20 ist, wird automatisch 1900 addiert; andernfalls wird 2000 addiert)
22	ddmmyyyy	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (z. B. "31010000" in 31)
23	mmdd	Konvertiert ein Datum in die Anzahl der Tage, die seit dem Anfang des Jahres vergangen sind (z. B. "0131" in 31)
24	mmddyy	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (wenn das Jahr yy kleiner-gleich 20 ist, wird automatisch 1900 addiert; andernfalls wird 2000 addiert)

Schlüsselwort	Format	Syntax
25	mmddyyyy	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (z. B. "01010001" in 366)
26	mmm	Konvertiert eine Monatsabkürzung aus 3 Buchstaben in einen Wert zwischen 1 und 12 (z. B. "MAR" in 3) [Beachten Sie, dass dies dasselbe ist wie das Konvertierungsschlüsselwort 8]
27	mmmdd	Konvertiert ein Datum in die Anzahl der Tage, die seit dem Anfang des Jahres vergangen sind (z. B. "JAN31" in 31)
28	mmmddyy	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (wenn das Jahr yy kleiner-gleich 20 ist, wird automatisch 1900 addiert; andernfalls wird 2000 addiert)
29	mmmddyyyy	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (z. B. "FEB010001" in 32)
30	mmmyy	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (wenn das Jahr yy kleiner-gleich 20 ist, wird automatisch 1900 addiert; andernfalls wird 2000 addiert)
31	mmmyyyy	Konvertiert ein Datum in die Anzahl der Tage, die zwischen dem ersten Tag des angegebenen Monats und dem 1. Januar 0000 vergangen sind (z. B. "FEB0001" in 32)
32	mmyy	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (wenn das Jahr yy kleiner-gleich 20 ist, wird

Schlüsselwort	Format	Syntax
		automatisch 1900 addiert; andernfalls wird 2000 addiert)
33	mmyyyy	Konvertiert ein Datum in die Anzahl der Tage, die zwischen dem ersten Tag des angegebenen Monats und dem 1. Januar 0000 vergangen sind (z. B. "'020001'" in 32)
34	yymm	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (wenn das Jahr yy kleiner-gleich 20 ist, wird automatisch 1900 addiert; andernfalls wird 2000 addiert)
35	yymmdd	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (wenn das Jahr yy kleiner-gleich 20 ist, wird automatisch 1900 addiert; andernfalls wird 2000 addiert)
36	yymmm	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (wenn das Jahr yy kleiner-gleich 20 ist, wird automatisch 1900 addiert; andernfalls wird 2000 addiert)
37	yymmmdd	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (wenn das Jahr yy kleiner-gleich 20 ist, wird automatisch 1900 addiert; andernfalls wird 2000 addiert)
38	yyyy	Konvertiert das Jahr in die Anzahl der Jahre, die seit dem Jahr 0000 vergangen sind (z. B. "'1998'" in 1998)
39	yyyymm	Konvertiert ein Datum in die Anzahl der Tage, die zwischen dem ersten Tag des

Schlüsselwort	Format	Syntax
		angegebenen Monats und dem 1. Januar 0000 vergangen sind (z. B. "'000102'" in 32)
40	yyyymmdd	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (z. B. "'00010201'" in 32)
41	yyyymmm	Konvertiert ein Datum in die Anzahl der Tage, die zwischen dem ersten Tag des angegebenen Monats und dem 1. Januar 0000 vergangen sind (z. B. "'000102'" in 32)
42	yyyymmmdd	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (z. B. "'0001FEB01'" in 32)
43	<day>* <month>	Konvertiert jedes Datum mit Begrenzer, gefolgt von Tag und Monat, in die Anzahl der Tage, die seit dem Beginn des Jahres vergangen sind (z.B. "'15-JAN'" in 15)
44	<day>* <month>* <year>	Konvertiert jedes Datum mit Begrenzer, bei dem der Tag vor dem Monat gefolgt vom Jahr erscheint, in die Anzahl der seit dem 1. Januar vergangenen Tage, 0000 (z. B. "'1/1/0001'" in 366)
45	<month>* <day>	Konvertiert ein Datum mit Begrenzer, das aus Monat und Tag besteht, in die Anzahl der Tage, die seit dem Anfang des Jahres vergangen sind (z. B. "'JAN 31'" in 31)
46	<month>* <day>* <year>	Konvertiert ein Datum mit Begrenzer, das aus Jahr, Monat und Tag besteht, in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (z. B. "'JAN 1, 0001'" in 366)
47	<month>* <year>	Konvertiert ein Datum mit Begrenzer, das aus Monat und Jahr besteht, in die Anzahl

Schlüsselwort	Format	Syntax
		der Tage, die zwischen dem ersten Tag des angegebenen Monats und dem 1. Januar 0000 vergangen sind
48	`<year>* <month>`	Konvertiert ein Datum mit Begrenzer, das aus Jahr und Monat besteht, in die Anzahl der Tage, die zwischen dem ersten Tag des angegebenen Monats und dem 1. Januar 0000 vergangen sind
49	`<year>* <month>* <day>`	Konvertiert ein Datum mit Begrenzer, das aus Jahr, Monat und Tag besteht, in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (z. B. `"0001/01/01"` in `366`)
50	`yy`	Konvertiert das Jahr in die Anzahl der Jahre, die seit dem Jahr 0000 vergangen sind (z. B. `"97"` in `97`)
51	`mm`	Konvertiert den Monat in einen Wert zwischen 1 und 12 (z. B. `"SEP"` in `9`)
52	`dd`	Konvertiert den Tag in einen Wert zwischen 1 und 31 (z. B. `"28"` in `28`)
53	`{January | February | March … }`	Konvertiert einen vollständigen Monatsnamen in einen Wert zwischen 1 und 12 (z. B. `"March"` in `3`) [Beachten Sie, dass dies dasselbe ist wie das Konvertierungsschlüsselwort 9]
54	`{Sunday | Monday | Tuesday … }`	Konvertiert einen vollständigen Wochentagsnamen in einen Wert zwischen 1 und 7, wobei die Woche am Sonntag beginnt (z. B. `"Sunday"` in `1`)
55	`{Sun | Mon | Tue … }`	Konvertiert eine 3-Tage-Wochentagsabkürzung in einen Wert zwischen 1 und 7, wobei die Woche am Sonntag beginnt (z. B. `"Sun"` in `1`)

Syntax

NUMBER konvertiert Textwerte im angegebenen Datenbereich anhand des angegebenen Umwandlungsformats für Datums- und Uhrzeitangaben in numerische Werte. Wenn eine Zeichenfolge mit dem angegebenen conversion_keyword nicht geparst werden kann, gibt NUMBER einen Fehler zurück. Format 0 konvertiert die ersten fünf Zeichen jeder Zeichenfolge für jede eindeutige Zeichenfolge in eine unterschiedliche Zahl. Dies ist ein einfacher Weg, eine Textspalte in eindeutige Klassen umzuwandeln, die als Ausgaben für einen Klassifikator dienen können.

Die begrenzten Formate (Schlüsselwort 43-49) unterstützen die folgenden Begrenzer:

- / (Schrägstrich)
- - (Bindestrich)
- , (Komma)
- " " (Leerzeichen)
- : (Doppelpunkt)

Monate können als mm oder mmm dargestellt werden; Tage können als d oder dd dargestellt werden; Jahre können als yy oder yyyy dargestellt werden.

Anmerkung: Alle Jahre in Datumsangaben können als yyyy statt als yy angegeben werden, um das Jahr 2000 zu unterstützen. Im Interesse der Abwärtskompatibilität wird bei Schlüsselwort 1-16 zu 2-stelligen Jahresangaben yy automatisch 1900 addiert. Bei den Umwandlungsschlüsselwörtern 17-55 wird bei yy < thresholdautomatisch 2000 hinzugefügt. Bei yy # threshold wird automatisch 1900 hinzugefügt.

Anmerkung: Der Jahr-2000-Grenzwert threshold wird auf der Registerkarte **Datenbereinigung** des Fensters **Erweiterte Einstellungen** festgelegt (Aufruf über **Optionen > Einstellungen > Erweiterte Einstellungen**).

Anmerkung: Wenn Sie den Jahr-2000-Grenzwert ändern, müssen Sie alle Makrofunktionen aktualisieren, die mithilfe der Makrofunktion NUMBERDatumswerte mit 2-stelligen Jahren bearbeiten. Um eine Aktualisierung einer Makrofunktion zu erzwingen,

können Sie eine beliebige Bearbeitung vornehmen (z. B. ein Leerzeichen hinzufügen und wieder löschen) und auf das Hakensymbol klicken, damit die Änderung übernommen wird.

Anmerkung: Bei Format 0 werden nur die ersten fünf Zeichen jeder Zeichenfolge für die Generierung einer eindeutigen Zahl herangezogen. Alle Zeichenfolgen mit denselben ersten fünf Zeichen ergeben denselben numerischen Wert. Dieselbe Zeichenfolge ergibt jedes Mal denselben numerischen Wert, sogar in verschiedenen Arbeitsblättern. Bearbeiten Sie die Zeichenfolgen bei Bedarf mit Zeichenfolgemakros, sodass die ersten fünf Zeichen eine Klasse eindeutig definieren. Beachten Sie, dass die resultierenden numerischen Werte unter Umständen sehr klein sind. Mithilfe des Fensters **Anzeigeformate** können Sie entweder die Anzahl der angezeigten Dezimalstellen erhöhen oder das Format in den exponentiellen Modus (`00E+00`) ändern.

Beispiele

`TEMP = NUMBER("$1.23")` oder `TEMP = NUMBER("123%", 2)`

Erstellt eine neue Spalte `TEMP`, die die Zahl `1.23` enthält.

```
TEMP = NUMBER(column("Jan", "Mar", "Dec", 8)
```

Erstellt eine neue Spalte `TEMP`, die die Zahlen Wert `1`, `3` und `12`enthält.

```
TEMP = NUMBER("1:52 PM", 14)
```

Erstellt eine neue Spalte `TEMP`, die die Zahl `832` enthält.

```
TEMP = NUMBER("1/1/95", 5)
```

Erstellt eine neue Spalte `TEMP`, die die Zahl `728660` enthält.

```
TEMP = NUMBER(V1)
```

Erstellt eine neue Spalte `TEMP`, die die numerischen Werte der Zeichenfolgen in Spalte `V1` enthält. Alle Dollarwerte werden korrekt in numerische Werte konvertiert. `???` wird bei Textzeichenfolgen zurückgegeben, die nicht mit dem `$`-Format geparst werden können.

```
TEMP = NUMBER(V1:V3, 4)
```

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Spalte `TEMP` enthält die numerischen Werte der Zeichenfolgen in Spalte `V1`. Die Spalte `VX` enthält die numerischen Werte der Zeichenfolgen in Spalte `V2`. Die Spalte `VY` enthält die numerischen Werte der Zeichenfolgen in Spalte `V3`. Alle Daten im Format `dd-mmm-yy` werden in die Anzahl der Tage

ab dem 1. Januar 0000 konvertiert. ??? werden für Textzeichenfolgen zurückgegeben, die nicht mit dem $-Format geparst werden können.

```
TEMP = NUMBER(V1[10:20]:V2, 10)
```

Erstellt zwei neue Spalten mit den Namen TEMP und VX. Die Spalte TEMP enthält die numerischen Werte der Zeichenfolgen in Zeile 10-20 von Spalte V1. Die Spalte VX enthält die numerischen Werte der Zeichenfolgen in Zeile 10-20 von Spalte V2. Alle standardmäßigen Darstellungen der Wochentage durch drei Zeichen werden in die Zahlen 0 bis 6 konvertiert (0 = Sonntag, 6 = Samstag). Wenn es für einen Wochentag keine Übereinstimmung gibt, wird ??? zurückgegeben.

```
TEMP = NUMBER(V1, 0)
```

Sofern Spalte V1 nur 5-stellige Zeichenfolgen enthält, wird eine neue Spalte TEMP erstellt, die für jede eindeutige Zeichenfolge einen unterschiedlichen numerischen Wert enthält.

Zugehörige Funktionen

Funktion	Syntax
WEEKDAY	Konvertiert ASCII-Datumszeichenfolgen in Wochentage

OR-Makro

Die OR Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

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

Parameter

`data1`

Die Zahlen, die durch logisches Oder mit den Werten in data2verknüpft werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`data2`

Die Zahl(en), die durch logisches Oder mit den Werten in `data1`verknüpft werden soll(en). Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in `data2` muss mit der Anzahl der Spalten in `data1` übereinstimmen, es sei denn, bei `data2` handelt es sich um eine Konstante. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`OR` OR berechnet das logische Oder zwischen den beiden angegebenen Datenbereichen. OR gibt für jede Eingabespalte eine neue Spalte zurück, wobei jeweils die entsprechende Spalte von `data1` durch logisches Oder mit der entsprechenden Spalte von `data2` verknüpft wird (d. h., die erste Spalte von `data1` wird durch logisches Oder mit der ersten Spalte von `data` verknüpft, die zweite Spalte mit der zweiten Spalte usw.).

Wenn es sich bei `data2` um eine Konstante handelt, wird jeder Wert in `data1` durch logisches Und mit dem betreffenden Wert verknüpft. Wenn `data2` eine oder mehrere Spalten enthält, werden die Berechnungen auf Zeilenbasis für eine Spalte aus `data2` und eine Spalte aus `data2` durchgeführt. Die erste Zeile aus `data1` wird mit logischem ODER mit dem ersten Zeilenwert aus `data2` verknüpft, die zweite Zeile mit der zweiten Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

Anmerkung: Wenn eine Spalte in jeder Zeile dieselbe Zahl *x* wie `data2` enthält, ist dies dasselbe, als wenn als *x* die Konstante `data2` verwendet wird.

Anmerkung: Der Operator `OR` kann mit einem doppelten senkrechten Strich (`||`) abgekürzt werden. Mit dem doppelten senkrechten Strich können Sie die beiden Argumente trennen (z.B. staat `V1 OR 3`, einfach `V1||3` eingeben).

Beispiele

`TEMP = 1 OR 8` oder `TEMP = 1 || 8`

Erstellt eine neue Spalte mit dem Name TEMP, die den Wert eins enthält (jede Zahl ungleich null wird als eins behandelt).

```
TEMP = V1 || 1
```

Erstellt eine neue Spalte TEMP, die nur Einsen enthält (jeder Wert, der durch logisches Oder mit der Zahl eins verknüpft wird, ergibt eins).

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

Erstellt eine neue Spalte TEMP, in der jeder Zeilenwert von Spalte V1 durch logisches Oder mit dem entsprechenden Zeilenwert von Spalte V2 verknüpft ist.

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

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. In der Spalte TEMP sind die Werte von Spalte V1 durch logisches Oder mit den entsprechenden Zeilenwerten von Spalte V4 verknüpft. In der Spalte VX sind die Werte von Spalten V2 und V5 durch logisches Oder verknüpft. In der Spalte VY sind die Werte von Spalten V3 und V6 durch logisches Oder verknüpft.

```
TEMP = V1[10:20] || V2
```

Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen das Ergebnis der logischen ODER-verknüpften Werte der Zeilen 10-20 der Spalten V1 und V2 enthalten. Die anderen Zellen in TEMP sind leer.

Zugehörige Funktionen

Funktion	Syntax
AND	Berechnet das logische Und zwischen den beiden angegebenen Datenbereichen.
NOT	Berechnet das logische Nicht des Inhalts des angegebenen Datenbereichs.

POSITION-Makro

Die POSITION Makro ist nur in Unica Campaign verfügbar.

Syntax

```
POSITION(colName, pattern [, start [, occurrence]])
```

Parameter

`colName`

Der Wert einer Spalte (muss dem Datentyp `string` angehören).

`pattern`

Das Muster (die Zeichenfolge), nach dem gesucht wird.

`start`

Das Byte, bei dem die Suche beginnen soll.

`occurrence`

Wenn Sie einen Wert für `n` angeben, wird nach dem `n`-ten Vorkommen des Musters gesucht.

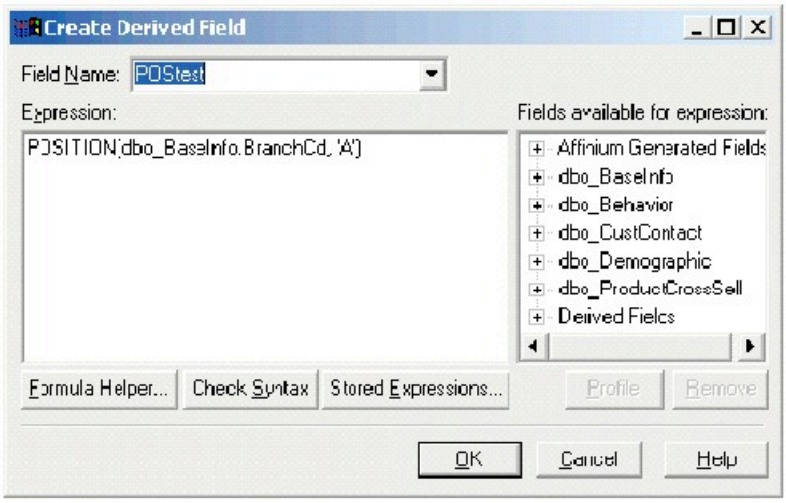
Syntax

`POSITION` gibt die Startbyteposition eines Musters oder einer Zeichenfolge innerhalb des Werts einer Spalte (`colName`), die dem Zeichenfolgedatentyp angehören muss, zurück. Wenn start angegeben ist, beginnt die Suche dort. Occurrence ist das n-te Vorkommen des Musters, das zurückgegeben werden soll.

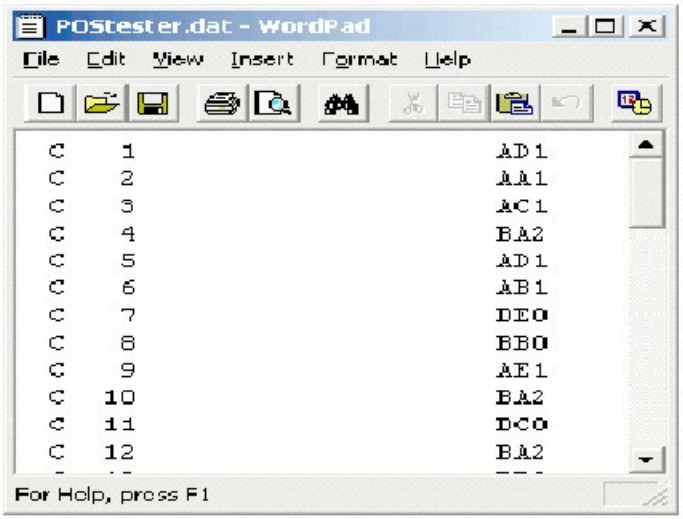
Anmerkung: Bei der Suche wird die Groß-/Kleinschreibung nicht beachtet.

Beispiele

Im folgenden Beispiel wird nach dem Muster oder der Zeichenfolge '`A`' innerhalb des Werts der Spalte `dbo_BaseInfo.BranchCd` gesucht, der Rückgabewert wird dem abgeleiteten Feld `POStest` zugeordnet.



Das folgende Beispiel zeigt einige Zeilen aus der Tabelle, wobei die Werte von `dbo_BaseInfo.BranchCd` und `POStest` nebeneinander gezeigt sind.



Ein komplexeres Beispiel:

```
STRING_SEG(POSITION(CellCode,"X",1,2)+1,

STRING_LENGTH(CellCode),CellCode) = "AAA"
```

Hierdurch werden Zeilen zurückgegeben, in denen die Werte von `CellCode` nach dem zweiten Vorkommen von 'AAA' am Ende 'X' enthalten.

PLUS-Makro

Die `PLUS` Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

```
data PLUS addend data + addend
```

Parameter

`data`

Der Zellenbereich mit Zahlen, zu denen etwas addiert werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`addend`

Die Zahl(en), die zu allen Werten in der angegebenen Spalte addiert werden soll(en). Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in `data2` muss mit der Anzahl der Spalten in `data1` übereinstimmen, es sei denn, bei `data2` handelt es sich um eine Konstante. Informationen zur Formatdefinition von `addend` (dasselbe wie bei `data`), siehe den Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`PLUS` PLUS addiert die Werte in den beiden angegebenen Datenbereichen. PLUS gibt für jede Eingabespalte eine neue Spalte zurück, wobei jeweils zu der entsprechenden Spalte von `data1` die entsprechende Spalte von `data2` addiert wird (d. h., zur ersten Spalte von `data1` wird die erste Spalte von `data` addiert, zur zweiten Spalte die zweite Spalte usw.).

Wenn es sich bei `data2` um eine Konstante handelt, wird jeder Wert in `data1` um den betreffenden Wert erhöht. Wenn `data2` eine oder mehrere Spalten enthält, werden die Berechnungen auf Zeilenbasis für eine Spalte aus `data2` und eine Spalte aus `data2` durchgeführt. Die erste Zeile von `data1` wird zum Wert der ersten Zeile mit `data2`addiert, die

zweite Zeile mit der zweiten Zeile und so weiter. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

Anmerkung: Der Operator PLUS kann mit einem Pluszeichen (+) abgekürzt werden.

Beispiele

`TEMP = 3 PLUS 4` oder `TEMP = 3 + 4`

Erstellt eine neue Spalte TEMP, die den Wert sieben enthält.

```
TEMP = V1 + 8
```

Erstellt eine neue Spalte TEMP, in der jeder Wert den Inhalt von Spalte V1 plus acht darstellt.

```
TEMP = V1 + V1
```

Erstellt eine neue Spalte TEMP, die jeweils den zweifachen Inhalt von Spalte V1 enthält.

```
TEMP = V1 + V2
```

Erstellt eine neue Spalte TEMP, in der jeder Wert der Zeilenwert der Spalte V1 plus der entsprechende Zeilenwert der Spalte V2ist.

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

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Die Spalte TEMP enthält die Werte in V1 plus die entsprechenden Zeilenwerte der Spalte V4. Die Spalte VX addiert Spalten V2 und V5. Die Spalte VY addiert Spalte V5 und Spalte V6.

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

Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen die Summen der Werte in den Zeilen 10-20 der Spalte v1 und die Werte in den Zeilen 1-11 der Spalte V2 enthalten. Die anderen Zellen in TEMP sind leer.

Zugehörige Funktionen

Funktion	Syntax
MINUS	Subtrahiert einen angegebenen Datenbereich von einem anderen.
SUM oder TOTAL	Berechnet die Summe eines Zellenbereichs.

POWER - Makro

Die POWER Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

```
base POWER exponent base ^ exponent
```

Parameter

base

Die numerischen Werte, die in die Potenz eines Exponenten erhoben werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von base (dasselbe wie bei data), siehe den Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

exponent

Die exponentielle(n) Zahl(en), in deren Potenz die Werte in data erhoben werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in exponent muss mit der Anzahl der Spalten in base übereinstimmen, es sei denn, bei base handelt es sich um eine Konstante. Informationen zur Formatdefinition von exponent (dasselbe wie bei data), siehe den Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

POWER erhebt die Werte im ersten Datenbereich in die im zweiten Datenbereich angegebene Potenz (berechnet wird also $base^{exponent}$). Dabei wird für jede Eingabespalte eine neue Spalte zurückgegeben, die jeweils das Ergebnis der Anhebung von base auf die exponent-Potenz enthält (d.h. die erste Spalte von data1 wird auf die erste Spalte von data angehoben, die zweite Spalte mit der zweiten Spalte, usw.).

Wenn es sich bei exponent um eine Konstante handelt, wird jeder Wert in base in die betreffende Potenz erhoben. Wenn exponent eine oder mehrere Spalten enthält, werden die Berechnungen auf Zeilenbasis für eine Spalte aus base und eine Spalte aus

`exponent` durchgeführt. Die erste Zeile von `base` wird auf den Wert der ersten Zeile von `exponent`angehoben, die zweite Zeile mit der zweiten Zeile und so weiter. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

Anmerkung: Der Operator `POWER` kann mit einem Zirkumflex (`^`) abgekürzt werden. Zum Beispiel ist `TEMP = 2^8` gleich `TEMP = 2 POWER 8`.

Anmerkung: Wenn der Wert *x* zu groß oder zu klein ist, wird ein Überlauffehler zurückgegeben. Dies geschieht, wenn `base^exponent` den maximalen 32-Bit-Gleitkommawert überschreitet.

Beispiele

`TEMP = 2 POWER 3` oder `TEMP = 2^3`

Erstellt eine neue Spalte `TEMP`, die den Wert acht enthält.

```
TEMP = V1 ^ 0.5
```

Erstellt eine neue Spalte `TEMP`, in der jeder Wert die Quadratwurzel des Inhalts der Spalte `V1` darstellt. (dies ist äquivalent zu `SQRT(V1)`).

```
TEMP = V1 ^ V3
```

Erstellt eine neue Spalte `TEMP`, in der jeder Wert der Zeilenwert der Spalte `V1` ist, der auf den entsprechenden Zeilenwert der Spalte `V2` angehoben wird.

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

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Erstellt eine neue Spalte `TEMP` , in der jeder Wert der Zeilenwert der Spalte `V1` ist, der auf den entsprechenden Zeilenwert der Spalte `V4` angehoben wird. Die Spalte `VX` enthält das Ergebnis der Spalte `V2` das auf die entsprechenden Werte in Spalte `V5` angehoben wurde. Die Spalte `VY` enthält das Ergebnis der Spalte `V3` das auf die entsprechenden Werte in Spalte `V6` angehoben wurde.

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

Erstellt eine neue Spalte `TEMP`, in der die ersten 11 Zellen das Ergebnis der Anhebung der Werte in den Zeilen 10-20 der Spalte `V1` um die Werte in den Zeilen 1-10 der Spalte `V2` enthalten. Die anderen Zellen in `TEMP` sind leer.

Zugehörige Funktionen

Funktion	Syntax
EXP	Erhebt die natürliche Zahl (e) in die Potenz, die durch den Inhalt jeder Zelle im angegebenen Datenbereich angegeben ist.
LN oder LOG	Berechnet den natürlichen Logarithmus des Inhalts des angegebenen Datenbereichs.
LN2	Berechnet die Protokollbasis2 des Inhalts des angegebenen Datenbereichs
LN10	Berechnet die Protokollbasis10 des Inhalts des angegebenen Datenbereichs

RANDOM-Makro

Die RANDOM Makro ist nur in Unica Campaign verfügbar.

Syntax

```
RANDOM(num [, seed]) RANDOM(num, value1, value2 [, seed])
```

Parameter

num

Die Anzahl der Zufallszahlen, die generiert werden sollen. Dieser Wert muss eine positive Ganzzahl größer als null sein.

value1

Eine Grenze für die Zufallszahlen, die generiert werden sollen. Dabei kann es sich um einen konstanten Wert oder um einen Ausdruck handeln, dessen Auswertung eine Konstante ergibt. Wird dieser Parameter nicht angegeben, ist der Standard null.

value2

Die andere Grenze für die Zufallszahlen, die generiert werden sollen. Dabei kann es sich um einen konstanten Wert oder um einen Ausdruck handeln, dessen Auswertung eine Konstante ergibt. Wird dieser Parameter nicht angegeben, ist der Standard eins.

seed

Ein optionaler Seed für die Generierung von Zufallszahlen. Der Wert muss eine ganze Zahl sein.

Syntax

`RANDOM` RANDOM generiert eine Spalte von Zufallszahlen. Dabei wird eine neue Spalte mit `num` Zufallszahlen zurückgegeben. Wenn `value1` und `value2` angegeben sind, werden die Zufallszahlen zwischen diesen Grenzen (inklusive) generiert. Wenn sie nicht angegeben sind, werden standardmäßig Werte zwischen null und eins generiert. Wenn `seed` angegeben ist, wird dieser Wert als Seed für die Generierung von Zufallszahlen verwendet.

Anmerkung: Wenn `seed` größer-gleich 2^{32} ist, wird der Wert durch 2^{32}-1 ersetzt. Werte von `seed` über 2^{24} werden gerundet (d. h., die Genauigkeit geht verloren). Deshalb können sich bei demselben Wert von `seed` mehrere Werte ergeben.

Beispiele

```
TEMP = RANDOM()
```

Erstellt eine neue Spalte `TEMP`, die Zufallszahlen mit unbegrenzter Länge enthält.

```
TEMP = RANDOM(100)
```

Erstellt eine neue Spalte `TEMP`, die 100 Zufallszahlen zwischen 0.0 und 1.0 enthält.

```
TEMP = RANDOM(100, 5943049)
```

Erstellt eine neue Spalte `TEMP` mit 100 Zufallszahlen zwischen 0 und 100, die ausgehend von dem Seedwert 5943049 generiert werden.

```
TEMP = RANDOM(100, 0, 100)
```

Erstellt eine neue Spalte `TEMP`, die 100 Zufallszahlen zwischen 0 und 100,0 enthält.

```
TEMP = RANDOM(100, 0, 100, 5943049)
```

Erstellt eine neue Spalte `TEMP` mit 100 Zufallszahlen zwischen -0 und 100, die ausgehend von dem Seedwert 5943049 generiert werden.

Zugehörige Funktionen

Funktion	Syntax
RANDOM_GAUSS	Gibt die angegebene Anzahl von Zufallswerten aus einer gaußschen Verteilung zurück.

RANDOM_GAUSS-Makro

Die RANDOM_GAUSS Makro ist nur in Unica Campaign verfügbar.

Syntax

```
RANDOM_GAUSS(num [, seed]) RANDOM_GAUSS(num, mean, std [, seed])
```

Parameter

`num`

Die Anzahl der Zufallszahlen, die generiert werden sollen. Dieser Wert muss eine positive Ganzzahl größer als null sein.

`mean`

Der Mittelwert der gaußschen Kurve. Dabei kann es sich um einen konstanten Wert oder um einen Ausdruck handeln, dessen Auswertung eine Konstante ergibt. Wird dieser Parameter nicht angegeben, ist der Standard null.

`std`

Die Standardabweichung der gaußschen Kurve. Dabei kann es sich um einen konstanten Wert oder um einen Ausdruck handeln, dessen Auswertung eine Konstante ergibt. Wird dieser Parameter nicht angegeben, ist der Standard eins.

`seed`

Ein optionaler Seed für die Generierung von Zufallszahlen. Der Wert muss eine ganze Zahl sein. (Wenn der angegebene Wert keine Ganzzahl ist, wird er automatisch durch die Untergrenze des Werts ersetzt.)

Syntax

`RANDOM_GAUSS` RANDOM_GAUSS generiert eine Spalte von Zufallszahlen, der eine gaußsche Verteilung zugrunde liegt. Dabei wird eine neue Spalte mit `num` Zufallszahlen zurückgegeben. Wenn `mean` und `std` angegeben sind, wird bei der Generierung der Zufallszahlen eine gaußsche Verteilung mit diesem Mittelwert und dieser Standardabweichung zugrunde gelegt. Wenn sie nicht angegeben sind, hat die gaußsche Verteilung standardmäßig den Mittelwert null und die Standardabweichung eins. Wenn `seed` angegeben ist, wird dieser Wert als Seed für die Generierung von Zufallszahlen verwendet.

Beispiele

```
TEMP = RANDOM_GAUSS(100)
```

Erstellt eine neue Spalte `TEMP` mit 100 Werten, die zufällig aus einer gaußschen Verteilung mit Mittelwert null und Einheitsstandardabweichung gewonnen wurden.

```
TEMP = RANDOM_GAUSS(500, 3)
```

Erstellt eine neue Spalte `TEMP` mit 100 Werten, die zufällig aus einer gaußschen Verteilung mit Mittelwert null und Einheitsstandardabweichung gewonnen wurden. Die Zahl `3` wird als Generierungswert für den Zufallsgenerator verwendet.

```
TEMP = RANDOM_GAUSS(5000, 100, 32)
```

Erstellt eine neue Spalte `TEMP` mit 5000 Werten, die zufällig aus einer gaußschen Verteilung mit Mittelwert 100 und Standardabweichung 32 gewonnen wurden.

```
TEMP = RANDOM_GAUSS(500, -1, 2, 3)
```

Erstellt eine neue Spalte `TEMP` mit 500 Werten, die zufällig aus einer gaußschen Verteilung mit Mittelwert `-1` und Standardabweichung `2` gewonnen wurden. Die Zahl `3` wird als Generierungswert für den Zufallsgenerator verwendet.

Zugehörige Funktionen

Funktion	Syntax
`RANDOM`	Gibt die angegebene Anzahl von Zufallszahlen zurück.

ROUND-Makro

Die ROUND Makro ist nur in Unica Campaign verfügbar.

Syntax

```
ROUND(data)
```

Parameter

data

Die numerischen Werte, die gerundet werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

ROUND rundet die Werte im angegebenen Datenbereich auf die nächste Ganzzahl. Dabei wird für jede Eingabespalte eine neue Spalte zurückgegeben, die den gerundeten Wert der Zahlen in der entsprechenden Eingabespalte enthält. Genau in der Mitte liegende Zahlen werden aufgerundet (so wird 2.5 zu 3.0 und 2.5 zu -2.0 gerundet).

Beispiele

```
TEMP = ROUND(3.2)
```

Erstellt eine neue Spalte TEMP, die den Wert drei enthält.

```
TEMP = ROUND(V1)
```

Erstellt eine neue Spalte TEMP, in der jeder Wert der gerundete Wert des Inhalts der Spalte V1 ist.

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

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Die Werte in der Spalte TEMP sind die gerundeten Werte des Inhalts von Spalte V1, die Werte von Spalte VX sind die gerundeten Werte des Inhalts von Spalte V2 und die Werte von Spalte VY sind die gerundeten Werte des Inhalts von Spalte V3.

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

Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen die gerundeten Werte den in den Zeilen 10-20 von Spalte V1 enthalten. Die anderen Zellen in TEMP sind leer.

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

Erstellt zwei neue Spalten TEMP und VX, die jeweils Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind die gerundeten Werte der entsprechenden Zeilen von Spalte V1, die Werte in der Spalte VX sind die gerundeten Werte der entsprechenden Zeilen von Spalte V2.

Zugehörige Funktionen

Funktion	Syntax
INT	Berechnet den abgerundeten Ganzzahlwert des Inhalts des angegebenen Datenbereichs.
MOD	Berechnet den Modulo-Wert des Inhalts des angegebenen Datenbereichs.
TRUNCATE	Berechnet den Ganzzahlanteil jedes Werts im angegebenen Datenbereich.

ROWNUM-Makro

Die ROWNUM Makro ist nur in Unica Campaign verfügbar.

Syntax

```
ROWNUM()
```

Syntax

ROWNUM generiert fortlaufende Zahlen von eins bis zur Anzahl der Einträge Die Zahl für den ersten Datensatz ist eins, die Zahl für den zweiten Datensatz ist zwei usw.

Anmerkung: ROWNUM kann maximal zwei Milliarden Einträge verarbeiten.

RTRIM-Makro

Das RTRIM-Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

```
RTRIM(data)
```

Parameter

`data`

Syntax

`RTRIM` entfernt nachfolgende Leerzeichen aus jedem Zeichenfolgewert im angegebenen Datenbereich und gibt die konvertierte Zeichenfolge zurück. RTRIM gibt für jede Eingabespalte eine neue Spalte zurück.

Beispiele

```
Temp = RTRIM "gold "
```

Erstellt eine neue Zeichenfolge Temp, die "gold" enthält.

SIGN-Makro

Die `SIGN` Makro ist nur in Unica Campaign verfügbar.

Syntax

```
SIGN(data)
```

Parameter

`data`

Die numerischen Werte, deren Vorzeichen berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

SIGN testet das Vorzeichen der Werte im angegebenen Datenbereich. SIGN gibt für jede Eingabespalte eine neue Spalte zurück, die das Vorzeichen der Zahlen in der entsprechenden Eingabespalte enthält. Bei allen Werten größer als null wird eine positive Eins zurückgegeben; bei allen Werten kleiner als null wird eine negative Eins zurückgegeben; bei allen Werten gleich null wird eine Null zurückgegeben.

Beispiele

```
TEMP = SIGN(-3)
```

Erstellt eine neue Spalte TEMP, die den Wert -1 enthält.

```
TEMP = SIGN(V1)
```

Erstellt eine neue Spalte TEMP, in der jeder Wert das Vorzeichen des Inhalts der Spalte V1 darstellt.

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

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Die Werte in der TEMP-Spalte sind die Vorzeichen des Inhalts der Spalte V1, die Werte der Spalte VX sind die Vorzeichen des Inhalts der Spalte V2, und die Werte der Spalte VY sind die Vorzeichen des Inhalts der Spalte V3.

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

Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen die Vorzeichen der Werte den in den Zeilen 10-20 von Spalte V1 enthalten. Die anderen Zellen in TEMP sind leer.

```
TEMP = SIGN(V1[10:50]:V2)
```

Erstellt zwei neue Spalten TEMP und VX, die jeweils die Werte in Zeile 1-41 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind jeweils die Vorzeichen der Werte in den Zeilen 10-50 der Spalte V1, und die Werte in der Spalte VX sind die Vorzeichen der Werte in den Zeilen 10-50 der Spalte V2.

Zugehörige Funktionen

Funktion	Syntax
ABS	Berechnet den absoluten Wert des Inhalts des angegebenen Datenbereichs.
INVERSE	Berechnet den Negativwert des Inhalts des angegebenen Datenbereichs.

SIN-Makro

Die `SIN` Makro ist nur in Unica Campaign verfügbar.

Syntax

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

Parameter

`data`

Die numerischen Werte, deren Sinus berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`units_keyword`

Dieses optionale Schlüsselwort legt fest, ob die Eingabewerte und Ergebnisse als Grad oder als Radiant interpretiert werden. Eine der folgenden Optionen auswählen:

`RADIAN` - Führt die Berechnungen in Radiant aus (Standard)

`DEGREE` - Führt die Berechnungen in Grad aus

Wird dieser Parameter nicht angegeben, ist der Standard Radiant. (Die Umrechnung von Radiant in Grad erfolgt durch Division durch `PI` und Multiplikation mit 180.)

Weitere Informationen zur Verwendung von Schlüsselwörtern in Unica Campaign finden Sie in Formatspezifikationen (auf Seite 12).

Syntax

`SIN` berechnet den Sinus der Werte im angegebenen Datenbereich. SIN gibt für jede Eingabespalte eine neue Spalte zurück, die den Sinus der Zahlen in der entsprechenden Eingabespalte enthält.

Beispiele

`TEMP = SIN(PI/2)` oder `TEMP = SIN(PI/2, 0)` oder `TEMP = SIGN(PI/2, RADIAN)`

Erstellt eine neue Spalte `TEMP`, die den Wert Eins enthält.

`TEMP = SIN(V1)`

Erstellt eine neue Spalte `TEMP`, in der jeder Wert der Sinus (in Radiant) des Inhalts der Spalte `V1` entspricht

`TEMP = SIN(V1:V3, 1)` oder `TEMP = SIN(V1:V3, DEGREE)`

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Werte in der Spalte `TEMP` sind jeweils die Sinuswerte des Inhalts von Spalte `V1`, die Werte von Spalte `VX` sind die Sinuswerte des Inhalts von Spalte `V2` und die Werte von Spalte `VY` sind die Sinuswerte des Inhalts von Spalte `V3`. Alle Werte sind in Grad angegeben.

`TEMP = SIN(V1[10:50]:V2)`

Erstellt zwei neue Spalten `TEMP` und `VX`, die jeweils die Werte in Zeile 1-41 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte `TEMP` sind jeweils die Sinuswerte der Zeilen 10-50 von Spalte `V1`, die Werte in der Spalte `VX` sind die Sinuswerte der Zeilen 10-50 von Spalte `V2`. Alle Werte sind in Radiant angegeben.

Zugehörige Funktionen

Funktion	Syntax
`ASIN`	Berechnet den Arkussinus des Inhalts des angegebenen Datenbereichs.
`COS`	Berechnet den Kosinus des Inhalts des angegebenen Datenbereichs.
`SINH`	Berechnet den Hyperbelsinus des Inhalts des angegebenen Datenbereichs.
`TAN`	Berechnet den Tangens des Inhalts des angegebenen Datenbereichs.

SINH-Makro

Die `SINH` Makro ist nur in Unica Campaign verfügbar.

Syntax

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

Parameter

`data`

Die numerischen Werte, deren Hyperbelsinus berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`units_keyword`

Dieses optionale Schlüsselwort legt fest, ob die Eingabewerte und Ergebnisse als Grad oder als Radiant interpretiert werden. Eine der folgenden Optionen auswählen:

`RADIAN` - Führt die Berechnungen in Radiant aus (Standard)

`DEGREE` - Führt die Berechnungen in Grad aus

Wird dieser Parameter nicht angegeben, ist der Standard Radiant. (Die Umrechnung von Radiant in Grad erfolgt durch Division durch `PI` und Multiplikation mit 180.)

Weitere Informationen zur Verwendung von Schlüsselwörtern in Unica Campaign finden Sie in Formatspezifikationen (auf Seite 12).

Syntax

`SINH` berechnet den Sinus der Werte im angegebenen Datenbereich. SINH gibt für jede Eingabespalte eine neue Spalte zurück, die den Hyperbelsinus der Zahlen in der entsprechenden Eingabespalte enthält. Bei *x* in Radiant ist der Hyperbelsinus einer Zahl

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

.

Dabei ist e die natürliche Zahl, 2,7182818.

Anmerkung: Wenn der Wert *x* zu groß ist, wird ein Überlauffehler zurückgegeben. Dies geschieht, wenn *sinh* (*x*) den maximalen 32-Bit-Gleitkommawert überschreitet.

Beispiele

`TEMP = SINH(1)` oder `TEMP = SINH(1, 0)` oder `TEMP = SINH(1, RADIAN)`

Erstellt eine neue Spalte `TEMP`, die den Wert `1.18` enthält.

`TEMP = SINH(V1)`

Erstellt eine neue Spalte `TEMP`, in der jeder Wert den hyperbolischen Sinus (in Radiant) des Inhalts der Spalte `V1` darstellt.

`TEMP = SINH(V1:V3, 1)` oder `TEMP = SINH(V1:V3, DEGREE)`

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Werte in der Spalte `TEMP` sind die hyperbolischen Sinuswerte des Inhalts von Spalte `V1`, die Werte von Spalte `VX` sind die hyperbolischen Sinuswerte des Inhalts von Spalte `V2` und die Werte von Spalte `VY` sind die hyperbolischen Sinuswerte des Inhalts von Spalte `V3`. Alle Werte sind in Grad angegeben.

`TEMP = SINH(V1[10:50]:V2)`

Erstellt zwei neue Spalten `TEMP` und `VX`, die jeweils die Werte in Zeile 1-41 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte `TEMP` sind jeweils die hyperbolischen Sinuswerte der Zeilen 10-50 von Spalte `V1`, die Werte in der Spalte `VX` sind die hyperbolischen Sinuswerte der Zeilen 10-50 von Spalte `V2`. Alle Werte sind in Radiant angegeben.

Zugehörige Funktionen

Funktion	Syntax
`COSH`	Berechnet den Hyperbelkosinus des Inhalts des angegebenen Datenbereichs.
`SIN`	Berechnet den Sinus des Inhalts des angegebenen Datenbereichs.
`TANH`	Berechnet den Hyperbeltangens des Inhalts des angegebenen Datenbereichs.

COUNT_DIM Makro

Das COUNT_DIM Makro ist in Unica Interact verfügbar.

Syntax

`COUNT_DIM(<dim field>)`

Parameter

`dim field`

Dimensionsfeld.

Beschreibung

Dieses Makro ist unter "Alle eingebauten Makros" aufgelistet. Verwenden Sie dieses Makro beim Erstellen eines interaktiven Ablaufdiagramms, um die Größe des Dimensionsfeldes zu ermitteln.Es kann unter den folgenden Optionen der Interaktiven Strategie verwendet werden:

- Diese Regel als berechtigt betrachten, wenn der folgende Ausdruck "true" ist:
- Folgenden Ausdruck als Marketing-Score verwenden:

Andere Makros können auf ähnliche Weise unter Interaktive Strategie verwendet werden.

Beispiel

```
COUNT_DIM(inttest183_interact_pftbl_null.creditScore)
```

SORT - Makro

Das SORT-Makro ist verfügbar in Unica Interact.

Syntax

```
SORT(<dim field>[, <sort order>])
```

Parameter

`dim field`

Dimensionsfeld.

`sort order`

Gibt an, ob nach aufsteigender oder absteigender Reihenfolge sortiert werden soll. Gültige Werte sind ASC und DESC.

Beschreibung

Dieses Makro ist unter "Alle eingebauten Makros" aufgelistet. Verwenden Sie dieses Makro, um die Dimensionsfelddaten in natürlicher Reihenfolge zu sortieren und gleichzeitig ein interaktives Ablaufdiagramm zu erstellen.

Beispiel

```
SORT(inttest183_interact_pftbl_null.rank)
```

SQRT-Makro

Die `SQRT` Makro ist nur in Unica Campaign verfügbar.

Syntax

```
SQRT(data)
```

Parameter

`data`

Die numerischen Werte, deren Quadratwurzel berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`SQRT` berechnet die Quadratwurzel der Werte im angegebenen Datenbereich. SQRT gibt für jede Eingabespalte eine neue Spalte zurück, die die positive Quadratwurzel der Zahlen in der entsprechenden Eingabespalte enthält.

Anmerkung: Wenn ein Wert im definierten Datenbereich negativ ist, wird der Wert `???` für diese Zelle zurückgegeben.

Beispiele

```
TEMP = SQRT(2)
```

Erstellt eine neue Spalte TEMP, die den Wert 1.41 enthält.

```
TEMP = SQRT(V1)
```

Erstellt eine neue Spalte TEMP, in der jeder Wert die Quadratwurzel des Inhalts der Spalte V1 darstellt.

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

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Die Werte in der Spalte TEMP sind jeweils die Quadratwurzel des Inhalts von Spalte V1, die Werte von Spalte VX sind die Quadratwurzel des Inhalts von Spalte V2 und die Werte von Spalte VY sind die Quadratwurzel des Inhalts von Spalte V3.

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

Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen die Quadratwurzel der Werte den in den Zeilen 10-20 von Spalte V1 enthalten. Die anderen Zellen in TEMP sind leer.

```
TEMP = SQRT(V1[10:50]:V2)
```

Erstellt zwei neue Spalten TEMP und VX, die jeweils die Werte in Zeile 1-41 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind jeweils die Quadratwurzel der Zeilen 10-50 von Spalte V1, die Werte in der Spalte VX sind die Quadratwurzel der Zeilen 10-50 von Spalte V2.

Zugehörige Funktionen

Function	Syntax
DIV	Dividiert einen angegebenen Datenbereich durch einen anderen.
MULT	Multipliziert die Inhalte zweier Datenbereiche.
POW	Erhebt einen Basiswert in die Potenz des oder der angegebenen Exponenten.

STDV- oder STDEV-Makro

STDV oder STDEV Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

`STDV(data [, keyword]) STDEV(data [, keyword])`

Parameter

`data`

Die numerischen Werte, deren Standardabweichung berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`keyword`

Dieses optionale Schlüsselwort legt fest, wie die Berechnung auf den Eingabedatenbereich angewendet wird. Eine der folgenden Optionen auswählen:

`ALL` - Wendet die Berechnung auf alle Zellen in `data` an (Standard)

`COL` - Führt die Berechnung für jede Zeile von data gesondert aus `data`

`ROW` - Führt die Berechnung für jede Spalte von data gesondert aus `data`

Weitere Informationen zur Verwendung von Schlüsselwörtern in Unica Campaign finden Sie in Formatspezifikationen (auf Seite 12).

Anmerkung: Viele Makrofunktionen nehmen die Schlüsselwortparameter `{ALL | COL | ROW}` an. Diese Schlüsselwörter gelten nicht für **Unica Campaign** , da es sich bei den Eingabedaten immer um eine Einzelspalte oder ein Einzelfeld handelt. Das Makro verhält sich immer so, als ob das Schlüsselwort `COL`angegeben würde. Deshalb brauchen Sie diese Schlüsselwörter nicht anzugeben, wenn Sie **Unica Campaign** verwenden.

Syntax

`STDV` berechnet die Standardabweichung aller Zellen im angegebenen Datenbereich.
Die Standardabweichung einer Verteilung ist die Quadratwurzel der Varianz. Die Standardabweichung wird wie folgt berechnet:

$$\sqrt{\frac{1}{n-1}\sum_{j=1}^{n}(x_j - \text{mean})^2}$$

Dabei steht x für eine Stichprobe, n für die Anzahl der Stichproben und Mittelwert für den Durchschnitt der Verteilung.

Anmerkung: Wenn die Anzahl der Stichproben *n* = 1 ist, gibt `STDV` einen Fehler zurück.

Beispiele

```
TEMP = STDV(V1)
```

Erstellt eine neue Spalte mit dem Namen `TEMP`, die einen einzelnen Wert enthält, der die Standardabweichung des Inhalts der Spalte `V1` darstellt.

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

Erstellt eine Spalte namens `TEMP`, die einen Einzelwert enthält, der die Standardabweichung des Inhalts von Spalte `V1`, `V2`und `V3`darstellt.

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

Erstellt eine neue Spalte `TEMP`, die einen Einzelwert enthält, der die Standardabweichung der Zellen in Zeile 1-5 von Spalte `V1` bis `V4` darstellt.

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

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Der Einzelwert in der Spalte `TEMP` ist die Standardabweichung des Inhalts von Spalte `V1`, der Einzelwert in der Spalte `VX` ist die Standardabweichung des Inhalts von Spalte `V2` und der Einzelwert in der Spalte `VY` ist die Standardabweichung des Inhalts von Spalte `V3`.

```
TEMP = STDV(V1[10:50]:V3,COL)
```

Erstellt drei neue Spalten `TEMP`, `VX` und `VY`, die jeweils einen Einzelwert enthalten. Der Wert in der Spalte `TEMP` ist die Standardabweichung der Zellen in Zeile 10-20 von Spalte `V1`, der Wert in der Spalte `VX` ist die Standardabweichung der Zellen in Zeile 10-50 von Spalte `V2` und der Wert in der Spalte `VY` ist die Standardabweichung der Zellen in Zeile 10-50 von Spalte `V3`.

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

Erstellt eine neue Spalte `TEMP`, in der jeder Zelleneintrag die Standardabweichung der entsprechenden Zeile in Spalte `V1`, `V2` und `V3` darstellt.

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

Erstellt eine neue Spalte `TEMP`, in der die Zellen in Zeile 1-5 die Standardabweichung der entsprechenden Zeile in Spalte `V1` bis `V3` enthalten. Die anderen Zellen in `TEMP` sind leer.

Zugehörige Funktionen

Funktion	Syntax
VAR	Berechnet die Varianz eines Zellenbereichs

STRING_CONCAT-Makro

Die `STRING_CONCAT` Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

```
STRING_CONCAT(string1, string2, ... stringN)
```

Parameter

`string`

Eine ASCII-Zeichenfolge, die verkettet werden soll. Dabei kann es sich um ASCII-Text in Anführungszeichen, eine Textspalte, einen Zellenbereich mit Text oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Wenn es sich bei diesem Parameter um einen numerischen Wert oder einen Datumszeitwert handelt, wird er unter Verwendung des Standardformats im Interact-RunTime-Server in einen String umgewandelt. Die Formatdefinition von string (wie data) finden Sie im Abschnitt "Makrofunktionsparameter" r für Ihr Produkt.

Anmerkung: Numerische und Datum/Zeitwerte können in allen Interact-Bereichen direkt an dieses Makro übergeben werden. Die Validierung schlägt jedoch fehl, wenn es in einem interaktiven Ablaufdiagramm verwendet wird.

Beschreibung

`STRING_CONCAT` verkettet die ASCII-Textwerte in den angegebenen Datenbereichen. Es gibt für jede Eingabespalte eine neue Spalte zurück, die jeweils die verketteten Zeichenfolgen aus den entsprechenden Zeilen von `strings` enthält. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

Anmerkung: Die Gesamtbreite jeder Ergebniszeichenfolge darf 255 Zeichen nicht überschreiten.

Unica Interact unterstützt auch die folgende Syntax:

`STRING_CONCAT(` *string1* `,` *string2* `, ...` *stringN* `)`

Beispiel: `STRING_CONCAT('a', 'b', 'c', 'd')` ist gültig.

Beispiele

```
TEMP = STRING_CONCAT("house", "boat")
```

Erstellt eine neue Spalte `TEMP`, die die ASCII-Zeichenfolge "`houseboat`" enthält.

```
TEMP = STRING_CONCAT(V1, ".")
```

Erstellt eine neue Spalte `TEMP`, in der jede Zeile die ASCII-Zeichenfolge in der entsprechenden Zeile von Spalte `V1` sowie einen angehängten Punkt enthält.

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

Erstellt eine neue Spalte `TEMP`, wobei in jeder Zeile die ASCII-Zeichenfolge in Spalte `V1` mit der Zeichenfolge in Spalte `V2`verkettet ist.

```
TEMP = STRING_CONCAT(V1:V3, V4:V6)
```

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Werte in der Spalte `TEMP` sind die verketteten Zeichenfolgen aus den entsprechenden Zeilen der Spalten `V1` und `V4`, die Werte der Spalte `VX` sind die verketteten Zeichenfolgen aus den entsprechenden Zeilen der Spalten `V2` und `V5` und die Werte der Spalte `VY` sind die verketteten Zeichenfolgen aus den entsprechenden Zeilen der Spalten `V3` und `V6`.

```
TEMP = STRING_CONCAT(V1[5:10]:V2, V3:V4)
```

Erstellt zwei neue Spalten mit den Namen `TEMP` und `VX`. Die Werte in der Spalte `TEMP` sind Zeichenfolgen aus den Zeilen 5-10 von Spalte `V1`, die mit den Zeichenfolgen aus den Zeilen 1-6 von Spalte `V3` verkettet sind. Die Werte in der Spalte `VX` sind Zeichenfolgen aus den Zeilen 5-10 von Spalte `V2`, die mit den Zeichenfolgen aus den Zeilen 1-6 von Spalte `V4` verkettet sind.

```
TEMP = STRING_CONCAT('a', 'b', 'c', 'd')
```

Erstellt eine neue Spalte `TEMP`, die die ASCII-Zeichenfolge "`abcd`" enthält.

Zugehörige Funktionen

Funktion	Beschreibung
`STRING_HEAD`	Gibt die ersten n Zeichen jeder Zeichenfolge im angegebenen Datenbereich zurück.
`STRING_LENGTH`	Gibt die Länge jeder Zeichenfolge im angegebenen Datenbereich zurück
`STRING_SEG`	Gibt das Zeichenfolgesegment zwischen zwei angegebenen Indizes zurück
`STRING_TAIL`	Gibt die letzten n Zeichen jeder Zeichenfolge im angegebenen Datenbereich zurück.

STRING_HEAD-Makro

Die `STRING_HEAD` Makro ist nur in Unica Campaign verfügbar.

Syntax

```
STRING_HEAD(num_chars, data)
```

Parameter

`num_chars`

Die Anzahl der Zeichen, die vom Ende jeder Zeichenfolge in `data`zurückgegeben werden sollen. Dieser Wert muss eine positive Ganzzahl größer als null sein.

`data`

ASCII-Zeichenfolgewerte. Dabei kann es sich um ASCII-Text in Anführungszeichen, eine Textspalte, einen Zellenbereich mit Text oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`STRING_HEAD` gibt die ersten `num_chars`-Zeichen jedes Zeichenfolgewerts im angegebenen Datenbereich zurück. Wenn `num_chars` größer als die Anzahl der Zeichen in einer Zeichenfolge ist, werden die verbleibenden Zeichen mit dem Nullzeichen " `\0` " aufgefüllt.

Beispiele

```
TEMP = STRING_HEAD(3, "JAN 15, 1997")
```

Erstellt eine neue Spalte TEMP, die die ASCII-Zeichenfolge "JAN" enthält.

```
TEMP = STRING_HEAD(10, "Pressure")
```

Erstellt eine neue Spalte TEMP, die die ASCII-Zeichenfolge "Pressure" enthält.

```
TEMP = STRING_HEAD(5, V1)
```

Erstellt eine neue Spalte TEMP, die die ersten fünf Zeichen jeder Zeichenfolge in der Spalte V1 enthält.

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

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Die Werte in der Spalte TEMP sind jeweils das erste Zeichen der Zeichenfolgen in den entsprechenden Zeilen von Spalte V1, die Werte von Spalte VX sind das erste Zeichen der Zeichenfolgen in den entsprechenden Zeilen von Spalte V2 und die Werte von Spalte VY sind das erste Zeichen der Zeichenfolgen in den entsprechenden Zeilen von Spalte V3.

```
TEMP = STRING_HEAD(12, V4[1:50]:V6]
```

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Die Werte in der Spalte TEMP sind die ersten 12 Zeichen der Zeichenfolgen in Zeile 1-50 von Spalte V1, die Werte von Spalte VX sind die ersten 12 Zeichen der Zeichenfolgen in Zeile 1-50 von Spalte V2 und die Werte von Spalte VY sind die ersten 12 Zeichen der Zeichenfolgen in Zeile 1-50 von Spalte V3.

Zugehörige Funktionen

Funktion	Syntax
STRING_CONCAT	Verkettet zwei Zeichenfolgen aus den angegebenen Datenbereichen.
STRING_LENGTH	Gibt die Länge jeder Zeichenfolge im angegebenen Datenbereich zurück
STRING_SEG	Gibt das Zeichenfolgesegment zwischen zwei angegebenen Indizes zurück
STRING_TAIL	Gibt die letzten n Zeichen jeder Zeichenfolge im angegebenen Datenbereich zurück.

STRING_LENGTH-Makro

Die `STRING_LENGTH` Makro ist nur in Unica Campaign verfügbar.

Syntax

```
STRING_LENGTH(data)
```

Parameter

`data`

ASCII-Zeichenfolgewerte, deren Länge berechnet werden soll. Dabei kann es sich um ASCII-Text in Anführungszeichen, eine Textspalte, einen Zellenbereich mit Text oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`STRING_LENGTH` Gibt die Länge jeder Zeichenfolge im angegebenen Datenbereich zurück Dabei wird für jede Eingabespalte eine neue Spalte zurückgegeben, die die Länge der entsprechenden Zeichenfolge enthält.

Anmerkung: Wenn `STRING_LENGTH` auf Spalten mit numerischen Daten angewendet wird, gibt die Funktion Nullen zurück.

Beispiele

```
TEMP = STRING_LENGTH("four")
```

Erstellt eine neue Spalte `TEMP`, die den Wert `4` enthält.

```
TEMP = STRING_LENGTH(4)
```

Erstellt eine neue Spalte `TEMP`, die den Wert `0` enthält.

```
TEMP = STRING_LENGTH(V1)
```

Erstellt eine neue Spalte `TEMP`, in der jeder Wert die Länge der Zeichenfolge in der entsprechenden Zeile von Spalte `V1` darstellt.

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

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Die Werte in der Spalte TEMP sind die Längen der Zeichenfolgen in den entsprechenden Zeilen von Spalte V1, die Werte von Spalte VX sind die Längen der Zeichenfolgen in den entsprechenden Zeilen von Spalte V2 und die Werte von Spalte VY sind die Längen der Zeichenfolgen in den entsprechenden Zeilen von Spalte V3.

```
TEMP = STRING_LENGTH(V4[1:50]:V6]
```

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Die Werte in der Spalte TEMP sind die Längen der Zeichenfolgen in Zeile 1-50 von Spalte V1, die Werte von Spalte VX sind die Längen der Zeichenfolgen in Zeile 1-50 von Spalte V2 und die Werte von Spalte VY sind die Längen der Zeichenfolgen in Zeile 1-50 von Spalte V3.

Zugehörige Funktionen

Funktion	Syntax
STRING_CONCAT	Verkettet zwei Zeichenfolgen aus den angegebenen Datenbereichen.
STRING_HEAD	Gibt die ersten n Zeichen jeder Zeichenfolge im angegebenen Datenbereich zurück.
STRING_SEG	Gibt das Zeichenfolgesegment zwischen zwei angegebenen Indizes zurück
STRING_TAIL	Gibt die letzten n Zeichen jeder Zeichenfolge im angegebenen Datenbereich zurück.

STRING_PROPER-Makro

Die STRING_PROPER Makro ist nur in Unica Campaign verfügbar.

Syntax

```
STRING_PROPER(data)
```

Parameter

`data`

Der Zeichenfolgewert, der konvertiert werden soll.

Syntax

`STRING_PROPER` Konvertiert jeden Zeichenfolgewert so, dass der erste Buchstabe oder irgendein Buchstabe, der auf ein Leerzeichen oder Symbol (außer dem Unterstrich) folgt, in einen Großbuchstaben und alle anderen Zeichen in Kleinbuchstaben geändert werden STRING_PROPER gibt für jede Eingabespalte eine neue Spalte zurück, die die konvertierte Zeichenfolge der entsprechenden Eingabespalte enthält.

Beispiele

```
Temp = STRING_PROPER
```

STRING_SEG-Makro

Die `STRING_SEG` Makro ist nur in Unica Campaign verfügbar.

Syntax

```
STRING_SEG(from, to, data)
```

Parameter

`from`

Die Zeichenzahl ab dem Anfang der Zeichenfolge, bei der die Extraktion des Zeichenfolgesegments beginnen soll. Es muss sich um eine positive Ganzzahl größer als null und kleiner als `STRING_SEG` handeln; andernfalls gibt `to` eine leere Zeichenfolge zurück.

`to`

Die Zeichenzahl ab dem Anfang der Zeichenfolge, bei der die Extraktion des Zeichenfolgesegments enden soll. Dabei muss es sich um eine positive Ganzzahl größer-gleich `from` handeln. Wenn `to` gleich `from` ist (und `to` kleiner-gleich der Länge der Zeichenfolge ist), wird ein Zeichen zurückgegeben.

`data`

ASCII-Zeichenfolgewerte. Dabei kann es sich um ASCII-Text in Anführungszeichen, eine Textspalte, einen Zellenbereich mit Text oder einen Ausdruck handeln, dessen Auswertung

einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`STRING_SEG` gibt aus jedem Zeichenfolgewert im angegebenen Datenbereich das Zeichenfolgesegment zwischen zwei Indizes zurück. Wenn `from` größer als die Länge einer Zeichenfolge ist, wird nichts zurückgegeben. Wenn `to` größer als die Länge einer Zeichenfolge ist, werden alle Zeichen ab `from` zurückgegeben.

Beispiele

```
TEMP = STRING_SEG(1, 6, "JAN 15, 1997")
```

Erstellt eine neue Spalte `TEMP`, die die ASCII-Zeichenfolge "`Jan 15`" enthält.

```
TEMP = STRING_SEG(5, 20, "Pressure")
```

Erstellt eine neue Spalte `TEMP`, die die ASCII-Zeichenfolge "`sure`" enthält.

```
TEMP = STRING_SEG(5, 6, V1)
```

Erstellt eine neue Spalte `TEMP`, die das fünfte und das sechste Zeichen jeder Zeichenfolge in der Spalte `V1` enthält.

```
TEMP = STRING_SEG(10, 20, V1:V3)
```

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Werte in der Spalte `TEMP` sind die Zeichen 10-20 der Zeichenketten in den entsprechenden Zeilen der Spalte `V1`, die Werte der Spalte `VX` sind die Zeichen 10-20 der Zeichenketten in den entsprechenden Zeilen der Spalte `V2`, und die Werte der Spalte `VY` sind die Zeichen 10-20 der Zeichenketten in den entsprechenden Zeilen der Spalte `V3`.

```
TEMP = STRING_SEG(5, 10, V4[1:50]:V6]
```

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Werte in der Spalte `TEMP` sind die Zeichen 5-10 der Zeichenfolgen in Zeile 1-50 von Spalte `V1`, die Werte von Spalte `VX` sind die Zeichen 5-10 der Zeichenfolgen in Zeile 1-50 von Spalte `V2` und die Werte von Spalte `VY` sind die Zeichen 5-10 der Zeichenfolgen in Zeile 1-50 von Spalte `V3`.

Zugehörige Funktionen

Funktion	Syntax
STRING_CONCAT	Verkettet zwei Zeichenfolgen aus den angegebenen Datenbereichen.
STRING_HEAD	Gibt die ersten n Zeichen jeder Zeichenfolge im angegebenen Datenbereich zurück.
STRING_LENGTH	Gibt die Länge jeder Zeichenfolge im angegebenen Datenbereich zurück
STRING_TAIL	Gibt die letzten n Zeichen jeder Zeichenfolge im angegebenen Datenbereich zurück.

STRING_TAIL-Makro

Die STRING_TAIL Makro ist nur in Unica Campaign verfügbar.

Syntax

```
STRING_TAIL(num_chars, data)
```

Parameter

num_chars

Die Anzahl der Zeichen, die vom Anfang jeder Zeichenfolge in datazurückgegeben werden sollen. Dieser Wert muss eine positive Ganzzahl größer als null sein.

data

ASCII-Zeichenfolgewerte. Dabei kann es sich um ASCII-Text in Anführungszeichen, eine Textspalte, einen Zellenbereich mit Text oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

STRING_TAIL Gibt die letzten num_chars Zeichen jeder Zeichenfolge im angegebenen Datenbereich zurück. Alle Zeichenfolgewerte werden bis zur Länge der längsten Zeichenfolge mit Nullzeichen "\0" aufgefüllt. Dann werden die letzten num_chars aus jeder

Zeichenfolge zurückgegeben. Wenn num_chars größer als die Anzahl der Zeichen in einer Zeichenfolge ist, wird die gesamte Zeichenfolge zurückgegeben.

Beispiele

```
TEMP = STRING_TAIL(3, "JAN 15, 1997")
```

Erstellt eine neue Spalte TEMP, die die ASCII-Zeichenfolge "997" enthält.

```
TEMP = STRING_TAIL(10, "Pressure")
```

Erstellt eine neue Spalte TEMP, die die ASCII-Zeichenfolge "Pressure" enthält.

```
TEMP = STRING_TAIL(5, V1)
```

Erstellt eine neue Spalte TEMP, die die letzten fünf Zeichen jeder Zeichenfolge in der Spalte V1 enthält.

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

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Die Werte in der Spalte TEMP sind jeweils die letzten Zeichen der Zeichenfolgen in den entsprechenden Zeilen von Spalte V1, die Werte von Spalte VX sind die letzten Zeichen der Zeichenfolgen in den entsprechenden Zeilen von Spalte V2 und die Werte von Spalte VY sind die letzten Zeichen der Zeichenfolgen in den entsprechenden Zeilen von Spalte V3.

```
TEMP = STRING_TAIL(12, V4[1:50]:V6]
```

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Die Werte in der Spalte TEMP sind die letzten 12 Zeichen der Zeichenfolgen in Zeile 1-50 von Spalte V1, die Werte von Spalte VX sind die letzten 12 Zeichen der Zeichenfolgen in Zeile 1-50 von Spalte V2 und die Werte von Spalte VY sind die letzten 12 Zeichen der Zeichenfolgen in Zeile 1-50 von Spalte V3.

Zugehörige Funktionen

Funktion	Syntax
STRING_CONCAT	Verkettet zwei Zeichenfolgen aus den angegebenen Datenbereichen.
STRING_HEAD	Gibt die ersten n Zeichen jeder Zeichenfolge im angegebenen Datenbereich zurück.
STRING_LENGTH	Gibt die Länge jeder Zeichenfolge im angegebenen Datenbereich zurück
STRING_SEG	Gibt das Zeichenfolgesegment zwischen zwei angegebenen Indizes zurück

SUBSTR- oder SUBSTRING-Makro

SUBSTR oder SUBSTRING Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

```
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])
```

Parameter

`string_value`

Die Zeichenfolge, der eine Unterzeichenfolge entnommen werden soll.

`start_pos`

Das erste Zeichen jeder Unterzeichenfolge wird extrahiert.

`nchars`

Die Anzahl der zu extrahierenden Zeichen (muss größer-gleich 0 sein). Wenn dieser Wert nicht angegeben ist, werden alle restlichen Zeichen in `string_value` extrahiert.

Syntax

SUBSTR oder SUBSTRING extrahiert `nchars` Zeichen aus der Zeichenkette, beginnend bei `start_pos`. Wenn `nchars` nicht angegeben ist, werden durch SUBSTR und SUBSTRING die Zeichen von `start_pos` bis zum Ende der Zeichenfolge extrahiert. Nachfolgende Leerzeichen werden automatisch abgeschnitten. Um Syntaxfehler zu vermeiden, stellen Sie sicher, dass numerische Werte durch ein Komma und ein Leerzeichen getrennt werden, wie in den Beispielen gezeigt.

⚠ **Wichtig:** Unica Interact unterstützt nur die folgenden Formate: `SUBSTR(string_value, start_pos[, nchars])` oder `SUBSTRING(string_value, start_pos[, nchars])`

Beispiele

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'

SUM-Makro

Die SUM Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

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

Parameter

data

Die numerischen Werte, deren Summe berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

keyword

Dieses optionale Schlüsselwort legt fest, wie die Berechnung auf den Eingabedatenbereich angewendet wird. Eine der folgenden Optionen auswählen:

ALL - Wendet die Berechnung auf alle Zellen in data an (Standard)

COL - Führt die Berechnung für jede Zeile von data gesondert aus data

ROW - Führt die Berechnung für jede Spalte von data gesondert aus data

Weitere Informationen zur Verwendung von Schlüsselwörtern in Unica Campaign finden Sie in Formatspezifikationen (auf Seite 12).

Anmerkung: Viele Makrofunktionen nehmen die Schlüsselwortparameter {ALL | COL | ROW} an. Diese Schlüsselwörter gelten nicht für Unica Campaign, da es sich bei den Eingabedaten immer um eine Einzelspalte oder ein Einzelfeld handelt. Das Makro verhält sich immer so, als ob das Schlüsselwort COLangegeben würde. Deshalb brauchen Sie diese Schlüsselwörter nicht anzugeben, wenn Sie Unica Campaign verwenden.

Syntax

SUM berechnet die Summe aller Zellen im angegebenen Datenbereich. SUM gibt eine einzelne Spalte zurück.

Anmerkung: SUM ist mit der Makrofunktion TOTAL identisch.

Beispiele

```
TEMP = SUM(3)
```

Erstellt eine neue Spalte TEMP, die den Wert drei enthält.

```
TEMP = SUM((COLUMN(3, 5, 1))
```

Erstellt eine neue Spalte TEMP, die den Wert neun enthält.

```
TEMP = SUM(V1)
```

Erstellt eine neue Spalte namens TEMP, die einen einzelnen Wert enthält, der die Summe des Inhalts der Spalte V1 darstellt.

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

Erstellt eine Spalte namens TEMP, die einen Einzelwert enthält, der die Summe des Inhalts von Spalte V1, V2und V3darstellt.

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

Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der die Summe der Zellen in Zeile 10-20 von Spalte V1 bis V4 darstellt.

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

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Der Einzelwert in der Spalte TEMP ist die Summe des Inhalts von Spalte V1, der Einzelwert in der Spalte VX ist die die Summe des Inhalts von Spalte V2 und der Einzelwert in der Spalte VY ist die Summe des Inhalts von Spalte V3.

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

Erstellt drei neue Spalten TEMP, VX und VY, die jeweils einen Einzelwert enthalten. Der Wert in der Spalte TEMP ist die Summe der Zellen in Zeile 1-5 von Spalte V1, der Wert in der Spalte VX ist die Summe der Zellen in Zeile 1-5 von Spalte V2 und der Wert in der Spalte VY ist die Summe der Zellen in Zeile 1-5 von Spalte V3.

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

Erstellt eine neue Spalte `TEMP`, in der jeder Zelleneintrag die Summe der entsprechenden Zeile in Spalte `V1`, `V2` und `V3`darstellt.

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

Erstellt eine neue Spalte `TEMP`, in der die Zellen in Zeile 1-5 die Summe der entsprechenden Zeile in Spalte `V1` bis `V3` enthalten. Die anderen Zellen in `TEMP` sind leer.

Zugehörige Funktionen

Funktion	Syntax
`AVG` oder `MEAN`	Berechnet das arithmetische Mittel oder den Durchschnitt eines Zellenbereichs

TAN-Makro

Die `TAN` Makro ist nur in Unica Campaign verfügbar.

Syntax

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

Parameter

`data`

Die numerischen Werte, deren Tangens berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`units_keyword`

Dieses optionale Schlüsselwort legt fest, ob die Eingabewerte und Ergebnisse als Grad oder als Radiant interpretiert werden. Eine der folgenden Optionen auswählen:

`RADIAN` - Führt die Berechnungen in Radiant aus (Standard)

DEGREE - Führt die Berechnungen in Grad aus

Wird dieser Parameter nicht angegeben, ist der Standard Radiant. (Die Umrechnung von Radiant in Grad erfolgt durch Division durch PI und Multiplikation mit 180.)

Weitere Informationen zur Verwendung von Schlüsselwörtern in Unica Campaign finden Sie in Formatspezifikationen (auf Seite 12).

Syntax

TAN berechnet den Tangens der Werte im angegebenen Datenbereich. TAN gibt für jede Eingabespalte eine neue Spalte zurück, die den Tangens der Zahlen in der entsprechenden Eingabespalte enthält.

Beispiele

`TEMP = TAN(PI/4)` oder `TEMP = TAN(PI/4, 0)` oder `TEMP = TAN(PI/4, RADIAN)`

Erstellt eine neue Spalte TEMP, die den Wert Eins enthält.

```
TEMP = TAN(V1)
```

Erstellt eine neue Spalte TEMP, in der jeder Wert den Tangens (in Radiant) des Inhalts von Spalte V1 darstellt.

`TEMP = TAN(V1:V3, 1)` oder `TEMP = TAN(V1:V3, DEGREE)`

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Die Werte in der Spalte TEMP sind jeweils der Tangens des Inhalts von Spalte V1, die Werte von Spalte VX sind der Tangens des Inhalts von Spalte V2 und die Werte von Spalte VY sind der Tangens des Inhalts von Spalte V3. Alle Werte sind in Grad angegeben.

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

Erstellt zwei neue Spalten TEMP und VX, die jeweils Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind der Tangens der entsprechenden Zeilen von Spalte V1, die Werte in der Spalte VX sind der Tangens der entsprechenden Zeilen von Spalte V2. Alle Werte sind in Radiant angegeben.

Zugehörige Funktionen

Funktion	Syntax
ATAN	Berechnet den Arkustangens des Inhalts des angegebenen Datenbereichs.

Funktion	Syntax
COS	Berechnet den Kosinus des Inhalts des angegebenen Datenbereichs.
COT	Berechnet den Kotangens des Inhalts des angegebenen Datenbereichs.
SIN	Berechnet den Sinus des Inhalts des angegebenen Datenbereichs.
TANH	Berechnet den Hyperbeltangens des Inhalts des angegebenen Datenbereichs.

TANH-Makro

Das TANH-Makro ist nur in Unica Campaign verfügbar.

Syntax

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

Parameter

`data`

Die numerischen Werte, deren Hyperbeltangens berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`units_keyword`

Dieses optionale Schlüsselwort legt fest, ob die Eingabewerte und Ergebnisse als Grad oder als Radiant interpretiert werden. Eine der folgenden Optionen auswählen:

`RADIAN` - Führt die Berechnungen in Radiant aus (Standard)

`DEGREE` - Führt die Berechnungen in Grad aus

Wird dieser Parameter nicht angegeben, ist der Standard Radiant. (Die Umrechnung von Radiant in Grad erfolgt durch Division durch `PI` und Multiplikation mit 180.)

Weitere Informationen zur Verwendung von Schlüsselwörtern in Unica Campaign finden Sie in Formatspezifikationen (auf Seite 12).

Syntax

TANH berechnet den Hyperbeltangens der Werte im angegebenen Datenbereich. TANH gibt für jede Eingabespalte eine neue Spalte zurück, die den Hyperbeltangens der Zahlen in der entsprechenden Eingabespalte enthält. Der Hyperbeltangens einer Zahl wird wie folgt berechnet:

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)}$$

Anmerkung: Wenn der Wert *x* zu groß ist, wird ein Überlauffehler zurückgegeben. Dies geschieht, wenn tahn (*x*) den maximalen 32-Bit-Gleitkommawert überschreitet. Wenn cosh (*x*) null ist, gibt TANH den maximalen 32-Bit-Gleitkommawert zurück.

Beispiele

`TEMP = TANH(PI)` oder `TEMP = TANH(PI, 0)` oder `TEMP = TANH(PI, RADIAN)`

Erstellt eine neue Spalte TEMP, die den Wert Eins enthält.

```
TEMP = TANH(V1)
```

Erstellt eine neue Spalte TEMP, in der jeder Wert den Hyperbeltangens (in Radiant) des Inhalts von Spalte V1 darstellt.

`TEMP = TANH(V1:V3, 1)` oder `TEMP = TANH(V1:V3, DEGREE)`

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Die Werte in der Spalte TEMP sind jeweils der Hyperbeltangens des Inhalts von Spalte V1, die Werte von Spalte VX sind der Hyperbeltangens des Inhalts von Spalte V2 und die Werte von Spalte VY sind der Hyperbeltangens des Inhalts von Spalte V3. Alle Werte sind in Grad angegeben.

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

Erstellt zwei neue Spalten TEMP und VX, die jeweils Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind der Hyperbeltangens der entsprechenden Zeilen von Spalte V1, die Werte in der Spalte VX sind der Hyperbeltangens der entsprechenden Zeilen von Spalte V2. Alle Werte sind in Radiant angegeben.

Zugehörige Funktionen

Funktion	Syntax
ATAN	Berechnet den Arkustangens des Inhalts des angegebenen Datenbereichs.

Funktion	Syntax
COSH	Berechnet den Hyperbelkosinus des Inhalts des angegebenen Datenbereichs.
COT	Berechnet den Kotangens des Inhalts des angegebenen Datenbereichs.
SINH	Berechnet den Hyperbelsinus des Inhalts des angegebenen Datenbereichs.
TAN	Berechnet den Tangens des Inhalts des angegebenen Datenbereichs.

TOTAL-Makro

Die TOTAL Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

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

Parameter

data

Die numerischen Werte, deren Summe berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

keyword

Dieses optionale Schlüsselwort legt fest, wie die Berechnung auf den Eingabedatenbereich angewendet wird. Eine der folgenden Optionen auswählen:

ALL - Wendet die Berechnung auf alle Zellen in data an (Standard)

COL - Führt die Berechnung für jede Zeile von data gesondert aus data

ROW - Führt die Berechnung für jede Spalte von data gesondert aus data

Weitere Informationen zur Verwendung von Schlüsselwörtern in Unica Campaign finden Sie in Formatspezifikationen (auf Seite 12).

Anmerkung: Viele Makrofunktionen nehmen die Schlüsselwortparameter `{ALL | COL | ROW}` an. Diese Schlüsselwörter gelten nicht für Unica Campaign, da es sich bei den Eingabedaten immer um eine Einzelspalte oder ein Einzelfeld handelt. Das Makro verhält sich immer so, als ob das Schlüsselwort `COL`angegeben würde. Deshalb brauchen Sie diese Schlüsselwörter nicht anzugeben, wenn Sie Unica Campaign verwenden.

Syntax

`TOTAL` berechnet die Summe aller Zellen im angegebenen Datenbereich.

Anmerkung: `TOTAL` ist mit der Makrofunktion `SUM` identisch.

Beispiele

```
TEMP = TOTAL(3)
```

Erstellt eine neue Spalte `TEMP`, die den Wert drei enthält.

```
TEMP = TOTAL((COLUMN(3, 5, 1))
```

Erstellt eine neue Spalte `TEMP`, die den Wert neun enthält.

```
TEMP = TOTAL(V1)
```

Erstellt eine neue Spalte namens `TEMP`, die einen einzelnen Wert enthält, der die Summe des Inhalts der Spalte `V1` darstellt.

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

Erstellt eine Spalte namens `TEMP`, die einen Einzelwert enthält, der die Summe des Inhalts von Spalte `V1`, `V2`und `V3`darstellt.

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

Erstellt eine neue Spalte `TEMP`, die einen Einzelwert enthält, der die Summe der Zellen in Zeile 10-20 von Spalte `V1` bis `V4` darstellt.

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

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Der Einzelwert in der Spalte `TEMP` ist die Summe des Inhalts von Spalte `V1`, der Einzelwert in der Spalte `VX` ist die die Summe des Inhalts von Spalte `V2` und der Einzelwert in der Spalte `VY` ist die Summe des Inhalts von Spalte `V3`.

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

Erstellt drei neue Spalten TEMP, VX und VY, die jeweils einen Einzelwert enthalten. Der Wert in der Spalte TEMP ist die Summe der Zellen in Zeile 1-5 von Spalte V1, der Wert in der Spalte VX ist die Summe der Zellen in Zeile 1-5 von Spalte V2, und der Wert in der Spalte VY ist die Summe der Zellen in Zeile 1-5 von Spalte V3.

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

Erstellt eine neue Spalte TEMP, in der jeder Zelleneintrag die Summe der entsprechenden Zeile in Spalte V1, V2 und V3darstellt.

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

Erstellt eine neue Spalte TEMP, in der die Zellen in Zeile 1-5 die Summe der entsprechenden Zeile in Spalte V1 bis V3 enthalten. Die anderen Zellen in TEMP sind leer.

Zugehörige Funktionen

Funktion	Syntax
AVG oder MEAN	Berechnet das arithmetische Mittel oder den Durchschnitt eines Zellenbereichs

TRUNCATE-Makro

Die TRUNCATE Makro ist nur in Unica Campaign verfügbar.

Syntax

```
TRUNCATE(data)
```

Parameter

`data`

Die numerischen Werte, die abgeschnitten werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

TRUNCATE berechnet den Gesamtteil jedes Wertes im angegebenen Datenbereich. Dabei wird für jede Eingabespalte eine neue Spalte zurückgegeben, die den ganzzahligen Anteil (den Vorkommaanteil) der Zahlen in der entsprechenden Eingabespalte enthält.

Anmerkung: Die Makrofunktionen FRACTION und TRUNCATE sind komplementär, da sie in der Summe die ursprünglichen Werte ergeben.

Beispiele

```
TEMP = TRUNCATE(4.3)
```

Erstellt eine neue Spalte TEMP, die den Wert 4 enthält.

```
TEMP = TRUNCATE(2.9)
```

Erstellt eine neue Spalte TEMP, die den Wert -2 enthält.

```
TEMP = TRUNCATE(V1)
```

Erstellt eine neue Spalte TEMP, in der jeder Wert ein Bruchteil des Inhalts der Spalte V1 ist.

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

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. Die Werte in der Spalte V1 sind die abgeschnittenen Teile von Spalte TEMP, die Werte von Spalte VX sind die abgeschnittenen Teile von Spalte V2 und die Werte von Spalte VY sind die abgeschnittenen Teile von Spalte V3.

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

Erstellt eine neue Spalte TEMP, wobei die ersten 11 Zellen die abgeschnittenen Teile der Werte in den Zeilen 10-20 der Spalte V1 enthalten. Die anderen Zellen in TEMP sind leer.

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

Erstellt zwei neue Spalten TEMP und VX, die jeweils Werte in Zeile 1-50 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind jeweils die abgeschnittenen Teile der Zeilen von Spalte V1, die Werte in der Spalte VX sind die Untergrenzen der Zeilen von Spalte V2.

Zugehörige Funktionen

Funktion	Syntax
CEILING	Berechnet die Obergrenze jedes Werts im angegebenen Datenbereich.
FLOOR	Berechnet die Untergrenze jedes Werts im angegebenen Datenbereich.
FRACTION	Berechnet die Nachkommastellen jedes Werts im angegebenen Datenbereich.

UPPER-Makro

Die UPPER Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

```
UPPER(data)
```

Parameter

`data`

Der Zeichenfolgewert, der in Großbuchstaben konvertiert werden soll.

Syntax

UPPER konvertiert jeden Zeichenfolgewert im angegebenen Datenbereich in Großbuchstaben. UPPER gibt für jede Eingabespalte eine neue Spalte zurück, die die in Großbuchstaben konvertierte Zeichenfolge der entsprechenden Eingabespalte enthält.

Beispiele

```
Temp = UPPER "gold"
```

Erstellt eine neue Spalte TEMP, die den Wert "GOLD" enthält.

```
TEMP = UPPER( "jan 15, 1997")
```

Erstellt eine neue Spalte TEMP, die die ASCII-Zeichenfolge "JAN 15, 1997" enthält.

```
TEMP = UPPER( "Pressure")
```

Erstellt eine neue Spalte TEMP, die die ASCII-Zeichenfolge "PRESSURE" enthält.

```
TEMP = UPPER(V1)
```

Erstellt eine neue Spalte `TEMP`, die die Großbuchstaben jeder Zeichenfolge in der Spalte `V1` enthält.

VARIANCE-Makro

Die `VARIANCE` Makro ist nur in Unica Campaign verfügbar.

Syntax

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

Parameter

`data`

Die numerischen Werte, deren Varianz berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`keyword`

Dieses optionale Schlüsselwort legt fest, wie die Berechnung auf den Eingabedatenbereich angewendet wird. Eine der folgenden Optionen auswählen:

`ALL` - Wendet die Berechnung auf alle Zellen in `data` an (Standard)

`COL` - Führt die Berechnung für jede Zeile von data gesondert aus `data`

`ROW` - Führt die Berechnung für jede Spalte von data gesondert aus `data`

Weitere Informationen zur Verwendung von Schlüsselwörtern in Unica Campaign finden Sie in Formatspezifikationen (auf Seite 12).

Anmerkung: Viele Makrofunktionen nehmen die Schlüsselwortparameter `{ALL | COL | ROW}` an. Diese Schlüsselwörter gelten nicht für **Unica Campaign** , da es sich bei den Eingabedaten immer um eine Einzelspalte oder ein Einzelfeld handelt. Das Makro verhält

sich immer so, als ob das Schlüsselwort `COL`angegeben würde. Deshalb brauchen Sie diese Schlüsselwörter nicht anzugeben, wenn Sie **Unica Campaign** verwenden.

Syntax

`VARIANCE` berechnet die Varianz aller Werte im angegebenen Datenbereich. Varianz ist die Standardabweichung im Quadrat. Die Varianz wird wie folgt berechnet:

$$\frac{1}{n-1}\sum_{j=1}^{n}(x_j - \text{mean})^2$$

Dabei steht x für eine Stichprobe, n für die Anzahl der Stichproben und Mittelwert für den Durchschnitt der Verteilung.

Anmerkung: Wenn die Anzahl der Stichproben *n* = 1 ist, gibt `VARIANCE` einen Fehler zurück.

Beispiele

```
TEMP = VARIANCE(V1)
```

Erstellt eine neue Spalte `TEMP`, die einen Einzelwert enthält, der die Varianz des Inhalts von Spalte `V1` darstellt.

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

Erstellt eine Spalte namens `TEMP`, die einen Einzelwert enthält, der die Varianz des Inhalts von Spalte `V1`, `V2`und `V3`darstellt.

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

Erstellt eine neue Spalte `TEMP`, die einen Einzelwert enthält, der die Varianz der Zellen in Zeile 10-20 von Spalte `V1` darstellt.

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

Erstellt eine neue Spalte `TEMP`, die einen Einzelwert enthält, der die Varianz der Zellen in Zeile 1-5 von Spalte `V1` bis `V4` darstellt.

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

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Der Einzelwert in der Spalte `TEMP` ist die Varianz des Inhalts von Spalte `V1`, der Einzelwert in der Spalte `VX` ist die Varianz des

Inhalts von Spalte V2 und der Einzelwert in der Spalte VY ist die Varianz des Inhalts von Spalte V3.

`TEMP = VARIANCE_(V1[1:5]:V3, COL)` oder `TEMP = VARIANCE(V1[1:5]:V3[1:5], COL)`

Erstellt drei neue Spalten TEMP, VX und VY, die jeweils einen Einzelwert enthalten. Der Wert in der Spalte TEMP ist die Varianz der Zellen in Zeile 1-5 von Spalte V1, der Wert in der Spalte VX ist die Varianz der Zellen in Zeile 1-5 von Spalte V2, und der Wert in der Spalte VY ist die Varianz der Zellen in Zeile 1-5 von Spalte V3.

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

Erstellt eine neue Spalte TEMP, in der jeder Zelleneintrag die Varianz der entsprechenden Zeile in Spalte V1, V2 und V3darstellt.

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

Erstellt eine Spalte namens TEMP, in der die Zellen in den Zeilen 1 bis 5 die Varianz der entsprechenden Zeile in den Spalten V1 bis V3 enthalten. Die anderen Zellen in TEMP sind leer.

WEEKDAY-Makro

Das WEEKDAY-Makro ist nur in Unica Campaign verfügbar.

Syntax

```
WEEKDAY(data [, conversion_keyword])
```

Parameter

`data`

Die ASCII-Datumsangaben, die in numerische Werte konvertiert werden sollen, die Wochentage darstellen (1-7). Dabei kann es sich um ASCII-Text in Anführungszeichen, eine Textspalte, einen Zellenbereich mit Text oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`conversion_keyword`

Dieses optionale Schlüsselwort legt fest, wie Textformate für Datums- und Uhrzeitangaben interpretiert werden sollen. Eine der folgenden Optionen auswählen:

`1 - mm/dd/yy` (Standardwert)

`2 - dd-mmm-yy`

`3 - mm/dd/yy hh:mm`

Wird dieser Parameter nicht angegeben, ist der Standard `1`.

Syntax

`WEEKDAY` WEEKDAY konvertiert Textwerte im angegebenen Datenbereich anhand des angegebenen Umwandlungsformats für Datums- und Uhrzeitangaben in numerische Werte, die Wochentage darstellen. Die Zahl `0` für Sonntag, die `1` für Montag usw. bis zur `6` für Samstag. Wenn eine Zeichenfolge mit dem angegebenen `conversion_keyword` nicht geparst werden kann, gibt `WEEKDAY` einen Fehler zurück.

Beispiele

```
TEMP = WEEKDAY("1/1/95")
```

Erstellt eine neue Spalte `TEMP`, die die Zahl `0` enthält (der 1. Januar 1995 ist ein Sonntag).

```
TEMP = WEEKDAY(V1, 2)
```

Erstellt eine neue Spalte `TEMP`, die für die Zeichenfolgen in Spalte `V1` Zahlen für die Wochentage enthält. Für alle Zeichenfolgen in Spalte `V1` wird das Format `dd-mmm-yy` erwartet (andernfalls wird jeweils `???` zurückgegeben).

```
TEMP = WEEKDAY(V1:V3, 3)
```

Erstellt drei neue Spalten mit den Namen `TEMP`, `VX`, und `VY`. Die Spalte `TEMP` enthält Zahlen, die die Wochentage der Zeichenfolgen in Spalte `V1` darstellen. Die Spalte `VX` enthält Zahlen, die die Wochentage der Zeichenfolgen in Spalte `V2` darstellen. Die Spalte `VY` enthält Zahlen, die die Wochentage der Zeichenfolgen in Spalte `V3` darstellen. Für alle Zeichenfolgen in den Spalten `V1` bis `V3` wird das Format `mm/dd/yy hh:mm` (andernfalls wird jeweils `???` zurückgegeben).

```
TEMP = WEEKDAY(V1[10:20]:V2, 10)
```

Erstellt zwei neue Spalten mit den Namen `TEMP` und `VX`. Die Spalte `TEMP` enthält Zahlen, die die Wochentage der Zeichenfolgen in Zeile 10-20 von Spalte `V1` darstellen. Die Spalte `VX` enthält Zahlen, die die Wochentage der Zeichenfolgen in Zeile 10-20 von Spalte `V2` darstellen. Für alle Zeichenfolgen wird das Format `mm/dd/yy` erwartet (andernfalls wird jeweils `???` zurückgegeben).

Zugehörige Funktionen

Funktion	Syntax
`NUMBER`	Konvertiert ASCII-Zeichenfolgen für Uhrzeit- und Datumsangaben in numerische Werte

WEEKDAYOF-Makro

Das WEEKDAYOF-Makro ist nur in Unica Campaign verfügbar.

Syntax

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

Parameter

`date_string`

Ein Text, der ein gültiges Datum darstellt.

`input_format`

Eines der Schlüsselwörter in der folgenden Tabelle, das das Datumsformat von `date_string` angibt.

Syntax

`WEEKDAYOF` gibt für das durch `date_string` angegebene Datum den Wochentag als Zahl zwischen 0 und 6 zurück (Sonntag 0, Montag 1 usw.). Wenn `input_format` nicht angegeben ist, wird das Standardschlüsselwort `DELIM_M_D_Y` verwendet.

Beispiele

`WEEKDAYOF("08312000", MMDDYYYY)` gibt die Zahl 4 zurück, da der Donnerstag als 4. Wochentag behandelt wird.

Anmerkung: Weitere Informationen zu gültigen Datumsformaten finden Sie unter DATUM *(auf Seite 71)*.

Zugehörige Funktionen

Funktion	Syntax
`DAYOF`	Gibt den Tag des Monats als Zahl zurück.
`MONTHOF`	Gibt den Monat des Jahres als Zahl zurück.
`YEAROF`	Gibt das Jahr als Zahl zurück.

XOR-Makro

Das XOR-Makro ist in Unica Campaign und Unica Interact verfügbar.

Syntax

```
data1 XOR data2
```

Parameter

`data1`

Die nicht negativen Ganzzahlen, die durch bitweises exklusives Oder mit den Werten in `data2` verknüpft werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

`data2`

Die nicht negativen Ganzzahlen, die durch bitweises exklusives Oder mit den Werten in `data1` verknüpft werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser

Typen ergibt. Die Anzahl der Spalten in `data2` muss mit der Anzahl der Spalten in `data1` übereinstimmen, es sei denn, bei `data2` handelt es sich um eine Konstante. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr Produkt.

Syntax

`XOR` berechnet das bitweise exklusives Oder zwischen den beiden angegebenen Datenbereichen. Es gibt für jede Eingabespalte eine neue Spalte zurück, wobei jeweils die entsprechende Spalte von `data1` durch bitweises exklusives Oder mit der entsprechenden Spalte von `data2` verknüpft wird (d. h., die erste Spalte von `data1` wird durch bitweises exklusives Oder mit der ersten Spalte von `data` verknüpft, die zweite Spalte mit der zweiten Spalte usw.).

Wenn es sich bei `data2` um eine Konstante handelt, wird jeder Wert in `data1` durch bitweises exklusives Oder mit dem betreffenden Wert verknüpft. Wenn `data2` eine oder mehrere Spalten enthält, werden die Berechnungen auf Zeilenbasis für eine Spalte aus `data2` und eine Spalte aus `data2` durchgeführt. Die erste Zeile von `data1` wird durch bitweises exklusives Oder mit der ersten Zeile von `data2` verknüpft, die zweite Zeile mit der zweiten Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

Anmerkung: Bei dieser Makrofunktion ist die Genauigkeit auf Ganzzahlwerte kleiner als 2^{24} begrenzt. Negative Werte sind nicht zulässig.

Beispiele

```
TEMP = 3 XOR 7
```

Erstellt eine neue Spalte `TEMP`, die den Wert vier enthält (das bitweise exklusive Oder von `011` und `111` ergibt `100`).

```
TEMP = V1 XOR 8
```

Erstellt eine neue Spalte `TEMP`, in der der Inhalt von Spalte `V1` jeweils durch bitweises Und mit dem Binärwert `1000` verknüpft ist.

```
TEMP = V1 XOR V1
```

Erstellt eine neue Spalte TEMP, die nur Nullen enthält (jeder Wert, der durch exklusives Oder mit sich selbst verknüpft wird, ergibt null).

```
TEMP = V1 XOR V2
```

Erstellt eine neue Spalte TEMP, in der jeder Zeilenwert von Spalte V1 durch bitweises exklusives Oder mit dem entsprechenden Zeilenwert von Spalte V2 verknüpft ist.

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

Erstellt drei neue Spalten mit den Namen TEMP, VX, und VY. In der Spalte TEMP sind die Werte von Spalte V1 durch bitweises exklusives Oder mit den entsprechenden Zeilenwerten von Spalte V4 verknüpft. In der Spalte VX sind die Werte von Spalten V2 und V5 durch bitweises exklusives Oder verknüpft. In der Spalte VY sind die Werte von Spalten V3 und V6 durch bitweises exklusives Oder verknüpft.

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

Erstellt eine neue Spalte TEMP, in deren ersten 11 Zellen die Werte in Zeile 10-20 von Spalte V1 durch bitweises exklusives Oder mit den Werten in Zeile 1-11 von Spalte V2 verknüpft sind. Die anderen Zellen in TEMP sind leer.

Zugehörige Funktionen

Funktion	Syntax
BIT_AND	Berechnet das bitweise Und zwischen den beiden angegebenen Datenbereichen.
BIT_NOT	Berechnet das bitweise Nicht des Inhalts des angegebenen Datenbereichs.
BIT_OR	Berechnet das bitweise Oder zwischen den beiden angegebenen Datenbereichen.

YEAROF-Makro

Das YEAROF-Makro ist nur in Unica Campaign verfügbar.

Syntax

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

Parameter

`date_string`

Ein Text, der ein gültiges Datum darstellt.

`input_format`

Eines der Schlüsselwörter in der folgenden Tabelle, das das Datumsformat von `date_string` angibt.

Syntax

`YEAROF` gibt für das durch `date_string` angegebene Datum das Jahr als Zahl zurück. Wenn `input_format` nicht angegeben ist, wird das Standardschlüsselwort `DELIM_M_D_Y` verwendet.

Beispiele

`YEAROF("31082000", DDMMYYYY)` gibt die Zahl 2000 zurück.

Weitere Informationen zu gültigen Datumsformaten finden Sie unter DATUM (auf Seite 71).

Zugehörige Funktionen

Funktion	Syntax
`DAYOF`	Gibt den Tag des Monats als Zahl zurück.
`MONTHOF`	Gibt den Monat des Jahres als Zahl zurück.
`WEEKDAYOF`	Gibt den Wochentag als Zahl zurück.

Index

E

F

G

H

I

K

L

M

N

O

P

R

S

T

U

V

W

X

Y

Z