



PUBLIC (ÖFFENTLICH)

SAP BusinessObjects Business Intelligence Suite

Dokumentversion: 4.3 Support Package 4 – 2023-12-07

Handbuch zur Verwendung von Funktionen, Formeln und Berechnungen in Web Intelligence

Inhalt

1	Verwendung von Funktionen, Formeln und Berechnungen zur Datenanalyse.	3
1.1	Dokumentverlauf: Funktionen, Formeln und Berechnungen von Web Intelligence.	3
1.2	Informationen zu diesem Handbuch.	4
1.3	Verwenden von Standard- und benutzerdefinierten Berechnungen.	4
	Einführung in den Formeleditor.	4
	Verwenden von Standard- und benutzerdefinierten Berechnungen.	5
1.4	Berechnungskontexte.	13
	Berechnungskontexte.	13
1.5	Berechnen von Werten mit intelligenten Kennzahlen.	26
	Berechnen von Werten mit intelligenten Kennzahlen.	26
1.6	Funktionen, Operatoren und Schlüsselwörter.	35
	Funktionen.	35
	Funktions- und Formeloperatoren.	272
	Erweiterte Syntax und Schlüsselwörter.	290
	Runden und Abschneiden von Zahlen.	295
	Verweisen auf Elemente und Elementmengen in Hierarchien.	296
1.7	Definieren benutzerdefinierter Funktionen.	298
	Übersicht über externe Funktionen.	298
	Definieren von benutzerdefinierten Berechnungen.	300
	Beispiele.	307
	#EXTERNAL-Fehlermeldung.	308
	Ereignisprotokoll-Fehlermeldungen.	308
1.8	Fehlerbehebung bei Formeln.	309
	Mechanismus zum automatischen Umschreiben von Formeln.	309
	Formelfehler- und Informationsmeldungen.	310
1.9	Vergleichen von Werten mithilfe von Funktionen.	317
	Vergleichen von Werten mithilfe der Funktion "Vorherige".	317
	Vergleichen von Werten mithilfe der Funktion RelativerWert.. . . .	317

1 Verwendung von Funktionen, Formeln und Berechnungen zur Datenanalyse

1.1 Dokumentverlauf: Funktionen, Formeln und Berechnungen von Web Intelligence

Die folgende Tabelle enthält eine Übersicht über die wichtigsten Änderungen:

Version	Datum	Beschreibung
SAP BusinessObjects Web Intelligence 4.3 SP3	Dezember 2022	<p>Folgende Abschnitte wurden aktualisiert oder dem Handbuch hinzugefügt:</p> <ul style="list-style-type: none">Die neuen Funktionen Umkehren [Seite 103], ElementLinkingFilters [Seite 166], InputControlFilter [Seite 167], DescriptionOf [Seite 241], FormulaOf [Seite 243] und Next [Seite 252] wurden hinzugefügt.ReportFilter [Seite 172] und ToDate [Seite 130] wurden aktualisiert.
SAP BusinessObjects Web Intelligence 4.3 SP1	Dezember 2020	<p>Folgende Abschnitte wurden aktualisiert oder dem Handbuch hinzugefügt:</p> <ul style="list-style-type: none">Neue Funktion (RPos [Seite 107]) hinzugefügt:Abschnitt Pos [Seite 101] wurde aktualisiert.
SAP BusinessObjects Web Intelligence 4.3	Juni 2020	<p>Folgende Abschnitte wurden aktualisiert oder dem Handbuch hinzugefügt:</p> <ul style="list-style-type: none">Die neuen Funktionen DocumentDescription [Seite 160], DokumentÜbergeordnerOrdner [Seite 161], DokumentPfad [Seite 163] und AnzahlSpalten [Seite 148] wurden hinzugefügt.Sie können nun Kommentare im Quellcode einer Formel einfügen.Aktualisiert:<ul style="list-style-type: none">Löschen [Seite 109], LöschenLeerLinks [Seite 97]

Version	Datum	Beschreibung
		<p>und LöschenRechts [Seite 106]. Sie können nun die Zeichen festlegen, die Sie entfernen möchten.</p> <ul style="list-style-type: none"> • Abfragezusammenfassung [Seite 170] und DatenproviderTyp [Seite 138]. Beide Funktionen geben nun die neuen Datenprovider-Typen zurück.

1.2 Informationen zu diesem Handbuch

Detaillierte Informationen zu erweiterten Berechnungsmöglichkeiten, die Sie bei der Durchführung von Datenanalysen verwenden können, finden Sie im Handbuch zur Verwendung von Funktionen, Formeln und Berechnungen in Web Intelligence.

Hier finden Sie auch eine Syntaxreferenz für verfügbare Funktionen und Operatoren.

1.3 Verwenden von Standard- und benutzerdefinierten Berechnungen

1.3.1 Einführung in den Formeleditor

Der Formeleditor ist der zentrale Ort zum Erstellen erweiterter Berechnungen und Variablen.



Er wurde entwickelt, um die Bearbeitung von Objekten zu erleichtern und mithilfe des integrierten Code-Editors Formeln oder Variablen schnell zu erstellen.

Code-Editor

Der Code-Editor bietet mehrere Funktionen, die Sie beim Schreiben von Formeln unterstützen:

- Klammerabgleich
- Syntaxanalyse
- Farbcodierung
- Automatische Vervollständigung
- Tastaturbefehle (`Strg` + `C`, `Strg` + `Z` usw.)




- Zeilennummerierung

Verwenden Sie die entsprechende Umschaltfläche, um den Zeilenumbruch () sowie die Syntaxanalyse und die Farbcodierung zu aktivieren ()

Eine vereinfachte Version des Code-Editors ist in der Formelleiste und in den Seitenbereichen in Web Intelligence verfügbar. Sie finden ihn beispielsweise in Formeltextfeldern zu Zellen, Blöcken und Berichten,

wenn Sie  >  > *Anzeigeeinstellungen* > *Ausblenden* > *Ausblenden, wenn Formel wahr ist* wählen.

Die Bereiche "Objekte", "Funktionen" und "Operatoren"

In den drei Bereichen (*Objekte* ) , *Funktionen* ) und *Operatoren* ) neben dem Texteditor können Sie Objekte per Drag&Drop schnell innerhalb des Code-Editors verschieben. Jeder dieser drei Bereiche kann bei Bedarf in der Größe geändert oder ausgeblendet werden.

Sie können die Option *Alle aufklappen* (bzw. *Alle zuklappen*) oben in jedem Bereich verwenden, um den Inhalt des Seitenbereichs auf- bzw. zuzuklappen.

Um Hilfe zu einem der verfügbaren Objekte zu erhalten, bewegen Sie den Mauszeiger darüber, um eine QuickInfo mit detaillierten Informationen zu erhalten. Wenn Sie auf eine der verfügbaren Funktionen oder einen Operator klicken, wird unten rechts im Dialogfeld eine Verknüpfung angezeigt, die Sie zur detaillierten Dokumentation im Help Portal führt. Wenn Sie den Mauszeiger über ein Objekt bewegen, wird eine QuickInfo mit Informationen angezeigt, die aus dem Objektverzeichnis abgerufen werden.

1.3.2 Verwenden von Standard- und benutzerdefinierten Berechnungen

Sie können Standard-Berechnungsformeln verwenden, um Daten rasch zu berechnen.

Wenn Ihre Anforderungen durch Standardberechnungen nicht erfüllt werden, können Sie mithilfe der Formelsprache benutzerdefinierte Berechnungen erstellen.

1.3.2.1 Standardberechnungen

Sie können Standard-Berechnungsformeln verwenden, um Daten rasch zu berechnen.

Folgende Standardberechnungen stehen zur Verfügung:

Berechnung	Beschreibung
Summe	Berechnet die Summe der markierten Daten.
Anzahl	Zählt alle Zeilen für ein Objekt vom Typ Kennzahl oder zählt die eindeutigen Zeilen für ein Objekt vom Typ Dimension oder Information.

Berechnung	Beschreibung
Durchschnitt	Berechnet den Durchschnitt der Daten.
Min	Zeigt den Mindestwert der markierten Daten an.
Max	Zeigt den Höchstwert der markierten Daten an.
Prozent	Zeigt die markierten Daten als Prozentsatz des Ganzen an. Die Ergebnisse einer Prozentrechnung werden in einer zusätzlichen Tabellenspalte oder -zeile angezeigt.

ⓘ Hinweis

Prozentsätze werden für die ausgewählte Kennzahl berechnet und mit den Gesamtergebnissen für diese Kennzahl in der Tabelle oder in dem Gruppenwechsel verglichen. Um den Prozentsatz einer Kennzahl im Vergleich zu einer anderen Kennzahl zu berechnen, müssen Sie eine benutzerdefinierte Berechnung erstellen.

Wenn Sie Standardberechnungen auf Tabellenspalten anwenden, werden die Berechnungsergebnisse im Fuß angezeigt. Ein Fuß wird pro Berechnung hinzugefügt.

1.3.2.2 Verwenden von Formeln zum Erstellen benutzerdefinierter Berechnungen

Benutzerdefinierte Berechnungen ermöglichen das Hinzufügen von Berechnungen zu Berichten, die über die grundlegenden Objekte und Standardberechnungen hinaus gehen.

Sie fügen eine benutzerdefinierte Berechnung hinzu, indem Sie eine Formel schreiben. Eine Formel kann aus grundlegenden Berichtsvariablen, Funktionen, Operatoren und Berechnungskontexten bestehen.

Bei einer benutzerdefinierten Berechnung handelt es sich um eine Formel, die aus Berichtsobjekten, Funktionen und Operatoren besteht. Formeln verfügen über einen Berechnungskontext, den Sie ggf. direkt angeben können.

Beispiel: Anzeigen des pro Artikel erzielten Durchschnittsumsatzes

Wenn Sie einem Bericht mit den Objekten "Umsatz" und "Anzahl verkaufter Produkte" das Objekt "Umsatz pro Artikel" hinzufügen möchten, ermittelt die Formel `[Sales Revenue] / [Number Sold]` den Wert für "Umsatz pro Artikel", indem sie den Umsatz durch die Anzahl der verkauften Artikel dividiert.

1.3.2.2.1 Verwenden von Variablen zur Vereinfachung von Formeln

Variablen sind nützlich, um Formeln in überschaubare Teile aufzuteilen und sie leichter lesbar zu machen. Zudem lassen sich mit Variablen einige Fehler beim Erstellen einer Formel vermeiden.

Sie finden die Variablen im Bereich *Objekte* unter *Variablen*, zusammen mit anderen Objekten der Abfrage.

Im Feld *Beschreibung* können Sie Kontext und Details für eine bestimmte Variable angeben. Die Beschreibung wird im *Abfrageeditor* angezeigt, wenn Sie den Mauszeiger auf die Variable bewegen. Sie können die Beschreibung anpassen, wenn Sie eine Variable erstellen, bearbeiten oder umbenennen.

1.3.2.3 Arbeiten mit Funktionen

Eine benutzerdefinierte Berechnung enthält manchmal ausschließlich Berichtsobjekte, z.B. `[Umsatz] / [Anzahl verkaufter Produkte]`. Berechnungen können neben Berichtsobjekten auch Funktionen beinhalten.

Eine Funktion verwendet null oder mehr Werte als Eingabe und gibt ein auf diesen Werten basierendes Ergebnis zurück. Die Funktion `Summe` berechnet beispielsweise die Summe aller Werte einer Kennzahl und gibt das Ergebnis dieser Berechnung zurück. Die Formel `Summe ([Umsatz])` gibt den Gesamtbetrag aller Umsatzwerte zurück. In diesem Fall wird die Kennzahl `Umsatz` als Eingabewert für die Funktion verwendet. Der Ausgabewert entspricht der Summe aller Werte der Kennzahl `Umsatz`.

Weitere Informationen

[Funktions- und Formeloperatoren \[Seite 272\]](#)

[Funktionen \[Seite 35\]](#)

1.3.2.3.1 Einfügen von Funktionen in Zellen

In Berichtszellen beginnt der Text immer mit dem Zeichen '='.

Literaltext steht zwischen Anführungszeichen, während Formeln ohne Anführungszeichen angezeigt werden. Die Formel `Durchschnitt([Umsatz])` wird beispielsweise folgendermaßen in einer Zelle angezeigt: `=Durchschnitt([Umsatz])`. Der Text "Durchschnitt Umsatz?" wird als `"Durchschnitt Umsatz?"` angezeigt

Text kann auch allein in Zellen verwendet oder unter Zuhilfenahme des Operators "+" mit Formeln kombiniert werden. Wenn eine Zelle den durchschnittlichen Umsatz mit dem vorangestellten Text "Durchschnitt Umsatz:" anzeigen soll, lautet der Zellentext wie folgt: `"Durchschnitt Umsatz:" + Durchschnitt([Umsatz])`

Beachten Sie die Leerstelle am Ende der Zeichenfolge. Auf diese Weise werden Text und Wert in der Zelle durch eine Leerstelle voneinander getrennt.

1.3.2.3.2 Funktionssyntax

Der *Formel-Editor* zeigt die Funktionssyntax an, wenn Sie die Funktion auswählen.

Zur Verwendung einer Funktion ist es wichtig, ihren Namen und die Anzahl der für sie erforderlichen Eingabewerte zu kennen. Der von der Funktion ausgegebene Datentyp muss ebenfalls bekannt sein.

Die Funktion `Summe` z.B. erfordert ein numerisches Objekt als Eingabe (z.B. ein Kennzahlobjekt, das den Umsatz ausweist). Sie gibt numerische Daten zurück (die Summe aller Werte des Kennzahlobjekts).

Die Syntax der Funktion `Abs` lautet:

```
Num Abs ( Zahl )
```

Aus der Syntax ist ersichtlich, dass die Funktion `Abs` eine Zahl als Eingabe erfordert und eine Zahl als Ausgabe zurückgibt.

1.3.2.3 Beispiele für Funktionen

Dieses Thema gibt Beispiele von in Formeln verwendeten Funktionen.

Beispiel: Anzeigen der Eingabewerte einer Eingabeaufforderung mit der AntwortBenutzer-Funktion

Sie verfügen über einen Bericht, der die Objekte Jahr, Quartal und Umsatz enthält. Das Objekt Land ist ebenfalls in den Berichtsdaten enthalten, obwohl es nicht angezeigt wird. Bei der Ausführung des Berichts wird dieses Objekt durch eine Eingabeaufforderung dargestellt, die zur Auswahl eines Landes auffordert. Sie möchten das Land anzeigen, das ein Benutzer als Berichtstitel ausgewählt hat. Wenn Ihr Datenprovider "eModen" heißt und der Text in der Eingabeaufforderung "Wählen Sie bitte ein Land aus" entspricht, dann lautet die Formel für den Titel:

```
"Quartalsweise Umsätze für " + AntwortBenutzer([Abfrage 1];"Werte für Staat eingeben: ")
```

Der Bericht sieht folgendermaßen aus, wenn der Benutzer beim Regenerieren des Datenproviders "Illinois" als Bundesstaat ausgewählt hat:

Quarterly Revenues for Illinois

Year	Quarter	Sales revenue
2004	Q1	\$256,454
2004	Q2	\$241,149
2004	Q3	\$107,006
2004	Q4	\$133,306
2004	Total	\$737,914

Year	Quarter	Sales revenue
2005	Q1	\$334,297
2005	Q2	\$254,722
2005	Q3	\$230,573
2005	Q4	\$331,067
2005	Total	\$1,150,659

Year	Quarter	Sales revenue
2006	Q1	\$255,658
2006	Q2	\$354,724
2006	Q3	\$273,186
2006	Q4	\$250,517
2006	Total	\$1,134,085

Beispiel: Berechnen von Prozentsätzen unter Verwendung der Funktion "Prozent"

Mit der Funktion "Prozentsatz" werden Prozentsätze berechnet. Diese Funktion berechnet den Prozentsatz einer Zahl im Verhältnis zu ihrem Umgebungskontext. In der folgenden Tabelle wird der Umsatz pro Jahr und Quartal angezeigt. Die Spalte "Prozentsatz" enthält die Formel `Prozentsatz ([Umsatz])`.

Year	Quarter	Sales revenue	Percentage
2004	Q1	\$256,454	0.08
2004	Q2	\$241,149	0.08
2004	Q3	\$107,006	0.04
2004	Q4	\$133,306	0.04
2005	Q1	\$334,297	0.11
2005	Q2	\$254,722	0.08
2005	Q3	\$230,573	0.08
2005	Q4	\$331,067	0.11
2006	Q1	\$255,658	0.08
2006	Q2	\$354,724	0.12
2006	Q3	\$273,186	0.09
2006	Q4	\$250,517	0.08
		Sum	1

In diesem Fall berechnet die Funktion den Prozentsatz, den jeder einzelne Umsatzwert im Vergleich zum Gesamtumsatz darstellt. Der Umgebungskontext ist also der Gesamtumsatz. Diese Zahl ist die einzige Zahl, die außer der Aufteilung nach Jahr und Quartal für die Tabelle relevant ist.

Wenn der Bericht in Sektionen unterteilt wird, die sich jeweils auf ein bestimmtes Jahr beziehen, dann stellt der Umgebungskontext außerhalb der Tabelle den Gesamtumsatz für die Sektion dar.

2004

Year	Quarter	Sales revenue	Percentage
2004	Q1	\$256,454	0.35
2004	Q2	\$241,149	0.33
2004	Q3	\$107,006	0.15
2004	Q4	\$133,306	0.18
		Sum	1

Wenn die Zelle "Prozent" außerhalb der Tabelle, jedoch innerhalb der Sektion angeordnet wird, stellt der Umgebungskontext den Gesamtumsatz dar. In diesem Fall wird durch die Funktion "Prozent" den Gesamtumsatz für die Sektion als Prozentanteil des Gesamtumsatzes berechnet.

2004

0.22

Year	Quarter	Sales revenue
2004	Q1	\$256,454
2004	Q2	\$241,149
2004	Q3	\$107,006
2004	Q4	\$133,306

2005

0.38

Year	Quarter	Sales revenue
2005	Q1	\$334,297
2005	Q2	\$254,722
2005	Q3	\$230,573
2005	Q4	\$331,067

Beispiel: Berechnen von Prozentsätzen unter Verwendung der Funktion "Summe"

Der Kontext, in dem ein Prozentsatz berechnet wird, lässt sich besser steuern, indem Sie anstelle der Funktion "Prozent" die Funktion "Summe" verwenden. Wenn Sie eine Zahl aus einer Menge von Zahlen durch die Gesamtsumme aller Zahlen teilen, erhalten Sie ihren prozentualen Anteil am Gesamtbetrag. Mit der Formel $[\text{Umsatz}] / \text{Summe}([\text{Umsatz}])$ erhalten Sie beispielsweise den Umsatz in der Form eines Prozentanteils im Vergleich zum Gesamtumsatz.

In der folgenden Tabelle enthält die Spalte "Prozent von Gruppenergebnis" die Formel:

```
[Umsatz]/(Summe([Umsatz] In Bericht))
```

und die Spalte [Prozentanteil Jahr] enthält die Formel:

```
[Umsatz]/(Summe([Umsatz] In Sektion))
```

2004

Year	Quarter	Sales revenue	Percentage of Year	Percentage of Total
2004	Q1	\$256,454	0.08	0.35
2004	Q2	\$241,149	0.08	0.33
2004	Q3	\$107,006	0.04	0.15
2004	Q4	\$133,306	0.04	0.18

In diesen Formeln werden die Schlüsselwörter Bericht und Sektion der erweiterten Syntax verwendet. Sie weisen die Funktion "Summe" an, den Gesamtumsatz des Berichts bzw. den Jahresumsatz zu berechnen.

Weitere Informationen

[Ändern von Standardberechnungskontexten unter Verwendung der erweiterten Syntax \[Seite 22\]](#)

1.3.2.3.3.1 Vereinfachen einer Varianz-Formel unter Verwendung von Variablen

Varianz ist ein statistischer Begriff. Bei der Varianz wird die Abweichung bestimmter Werte von einem Durchschnitt ermittelt.

Obwohl mit der `var`-Funktion die Varianz in einem Schritt berechnet wird, bietet die manuelle Berechnung der Varianz ein gutes Beispiel dafür, wie eine komplexe Formel mithilfe von Variablen vereinfacht werden kann. So berechnen Sie die Varianz manuell:

- Berechnen Sie die durchschnittliche Anzahl verkaufter Artikel.
- Berechnen Sie die Differenz zwischen der Anzahl verkaufter Artikel und dem Durchschnitt, und erheben Sie diesen Wert dann ins Quadrat.
- Addieren Sie die ins Quadrat erhobenen Differenzen.
- Dividieren Sie die Summe durch die Anzahl der Werte - 1.

Angenommen, Sie verfügen über einen Bericht, der die Anzahl verkaufter Artikel bzw. Produkte nach Quartal anzeigt. Sie möchten die Varianz in den Bericht einfügen. Die komplexe Formel, in der keine Variablen zur Vereinfachung der Formel verwendet werden, sieht folgendermaßen aus:

```
Summe(((Anzahl verkaufter Produkte] - Durchschnitt([Anzahl verkaufter  
Produkte] FürJeden [Quartal]) In Bericht)*([Anzahl verkaufter Produkte] -  
Durchschnitt([Anzahl verkaufter Produkte] FürJeden [Quartal]) In Bericht)) In  
[Quartal])/(Anzahl ([Anzahl verkaufter Produkte] FürJeden [Quartal]) - 1)
```

Erstellen der Varianz-Formel

Die Erstellung einer Varianz-Formel umfasst mehrere Schritte. Jeder einzelne dieser Schritte wird in Form einer Variablen festgehalten. Die erstellten Variablen lauten:

- Durchschnitt "Verkaufte Artikel"
- Anzahl der Vorkommen (d. h. Anzahl der einzeln auftretenden Werte im Rahmen der Anzahl verkaufter Artikel)
- Differenz zwischen Vorkommen und Durchschnitt, ins Quadrat erhoben
- Summe der Differenzen dividiert durch die Anzahl der Vorkommen - 1

Die Variablenformeln lauten wie folgt:

Variable	Formel
Durchschnittliche Verkaufsmenge	Durchschnitt([Verkaufsmenge] In ([Quartal])) In Bericht
Anzahl der Vorkommen	Anzahl([Verkaufsmenge] In ([Quartal])) In Bericht
Differenz im Quadrat	Potenz(([Verkaufsmenge] - [Durchschnittliche Verkaufsmenge]);2)
Varianz	Summe([Differenz im Quadrat] In ([Quartal]))/([Anzahl der Vorkommen] - 1)

Die endgültige Formel sieht folgendermaßen aus:

```
Summe([Differenz im Quadrat]/[Anzahl der Vorkommen] - 1)
```

Diese Formel ist viel einfacher zu verstehen. Sie spiegelt die anhand der Formel ausgeführten Berechnungen auf übersichtliche Weise wider. Die verwirrenden Einzelheiten bleiben Ihnen erspart. Sie können die einzelnen Formeln der Variablen einer vereinfachten Formel ggf. zum besseren Verständnis näher betrachten.

In der vereinfachten Formel wird beispielsweise die Variable "Differenz im Quadrat" verwendet, die ihrerseits auf der Variablen "Durchschnittliche Verkaufsmenge" aufbaut. Indem Sie die einzelnen Formeln von "Differenz im Quadrat" und "Durchschnittliche Verkaufsmenge" näher betrachten, analysieren Sie sie ähnlich wie bei einem Drilldown und können sich somit die einzelnen Schritte bei der Formelberechnung vergegenwärtigen.

1.4 Berechnungskontexte

1.4.1 Berechnungskontexte

Beim Berechnungskontext handelt es sich um die Daten, die von der Berechnung berücksichtigt werden, um ein Ergebnis zu erzeugen.

Dies bedeutet, dass der durch eine Kennzahl angegebene Wert durch die Dimensionen bestimmt wird, die zur Berechnung der Kennzahl verwendet werden.

Ein Bericht enthält zwei Arten von Objekten:

- Dimensionen stehen für Geschäftsdaten, die Zahlen generieren. Geschäfte, Jahre oder Regionen sind Beispiele für Dimensionsdaten. Beispielsweise kann ein Geschäft, ein Jahr oder eine Region Umsatz

generieren: So können Umsätze nach Geschäft, Umsätze nach Jahr oder Umsätze nach Region unterschieden werden.

- Kennzahlen sind numerische Daten, die durch Dimensionsdaten generiert werden. Beispiele für Kennzahlen sind der Umsatz und die Anzahl verkaufter Produkte. Beispielsweise lässt sich die Anzahl verkaufter Produkte in einem bestimmten Geschäft ermitteln.

Kennzahlen können auch aus kombinierten Dimensionsdaten generiert werden. So kann der Umsatz ermittelt werden, der 2005 von einem bestimmten Geschäft erzielt wurde.

Der Berechnungskontext einer Kennzahl umfasst zwei Komponenten:

- die Dimension oder Liste von Dimensionen, die den Kennzahlwert bestimmen
- den Teil der Dimensionsdaten, die den Kennzahlwert bestimmen

Der Berechnungskontext umfasst zwei Komponenten:

- Der Eingabekontext
- Der Ausgabekontext

Weitere Informationen

[Der Eingabekontext \[Seite 14\]](#)

[Der Ausgabekontext \[Seite 15\]](#)

1.4.1.1 Der Eingabekontext

Der Eingabekontext einer Kennzahl oder Formel entspricht der Liste von Dimensionen, die in die Berechnung einfließen.

Die Liste der Dimensionen in einem Eingabekontext wird innerhalb der Klammern der Funktion angezeigt, durch die der Wert ausgegeben wird. Die Liste der Dimensionen muss außerdem in Klammern eingeschlossen werden (auch wenn sie nur eine Dimension enthält), und die Dimensionen müssen durch Semikolons getrennt werden.

Beispiel: Definieren des Eingabekontextes

In einem Bericht mit Jahressektionen und einem Block in jeder Sektion mit Kunden- und Umsatzspalten lauten die Eingabekontexte wie folgt:

Berichtbestandteil	Eingabekontext
Sektionszelle und Blockfüße	Jahr
Zeilen im Block	Jahr, Kunde

Anders ausgedrückt bedeutet dies, dass in den Sektionszellen und Blockfüßen der aggregierte Umsatz nach Jahr und in den einzelnen Zeilen im Block der aggregierte Umsatz nach Jahr und Kunde angezeigt wird (der vom jeweiligen Kunden im betreffenden Jahr generierte Umsatz).

Bei ausdrücklicher Angabe in einer Formel sehen diese Eingabekontexte folgendermaßen aus:

```
Summe ([Umsatz] In ([Jahr]))
```

```
Summe ([Umsatz] In ([Jahr];[Kunde]))
```

Die im Eingabekontext definierten Dimensionen werden innerhalb der Klammern der Funktion (in diesem Fall Summe) angezeigt, deren Eingabekontext angegeben wird.

1.4.1.2 Der Ausgabekontext

Der Ausgabekontext veranlasst die Formel zur Ausgabe eines Wertes, wenn sie sich im Fuß eines Blocks mit einem Gruppenwechsel befindet.

Beispiel: Definieren von Ausgabekontexten

Im unten abgebildeten Bericht wird der Umsatz pro Jahr und Quartal gezeigt. Dabei wurde ein Gruppenwechsel auf Jahr angewendet und der Minimalumsatz pro Jahr berechnet.

Year	Quarter	Sales revenue
2004	Q1	\$2,660,700
	Q2	\$2,278,693
	Q3	\$1,367,841
	Q4	\$1,788,580
2004		
	Min:	\$1,367,841

Year	Quarter	Sales revenue
2005	Q1	\$3,326,172
	Q2	\$2,840,651
	Q3	\$2,879,303
	Q4	\$4,186,120
2005		
	Min:	\$2,840,651

Year	Quarter	Sales revenue
2006	Q1	\$3,742,989
	Q2	\$4,006,718
	Q3	\$3,953,395
	Q4	\$3,356,041
2006		
	Min:	\$3,356,041

Auf welche Weise lässt sich der Minimalumsatz pro Jahr in einem Block ohne Gruppenwechsel anzeigen? Dies erzielen Sie, indem Sie den Ausgabekontext in einer Formel definieren. Die Formel dafür sieht folgendermaßen aus:

```
Min ([Umsatz]) In ([Jahr])
```

Der Ausgabekontext wird nach den Klammern der Funktion angegeben, deren Ausgabekontext Sie gerade definieren. In diesem Fall wird mit dem Ausgabekontext der Minimalumsatz nach Jahr berechnet.

Wenn Sie eine zusätzliche Spalte zu dem Block hinzufügen, die diese Formel enthält, erhalten Sie folgendes Ergebnis:

Year	Quarter	Sales revenue	Min By Year
2004	Q1	\$2,660,700	\$1,367,841
2004	Q2	\$2,278,693	\$1,367,841
2004	Q3	\$1,367,841	\$1,367,841
2004	Q4	\$1,788,580	\$1,367,841
2005	Q1	\$3,326,172	\$2,840,651
2005	Q2	\$2,840,651	\$2,840,651
2005	Q3	\$2,879,303	\$2,840,651
2005	Q4	\$4,186,120	\$2,840,651
2006	Q1	\$3,742,989	\$3,356,041
2006	Q2	\$4,006,718	\$3,356,041
2006	Q3	\$3,953,395	\$3,356,041
2006	Q4	\$3,356,041	\$3,356,041

Die Spalte "Min. pro Jahr" enthält Werte zum Minimalumsatz, die im vorherigen Bericht in den Gruppenwechselfüße angezeigt wurden.

In diesem Beispiel wurde kein Eingabekontext definiert, da es sich um den Standardkontext (Jahr, Quartal) des Blocks handelt. In anderen Worten bestimmt der Ausgabekontext, welcher Umsatz pro Jahr und Quartal ausgegeben werden soll. Formeln, in denen sowohl der Eingabe- als auch der Ausgabekontext definiert wurden, sehen folgendermaßen aus:

```
Min ([Umsatz] In([Jahr];[Quartal])) In ([Jahr])
```

Mit dieser Formel wird der Umsatz pro Jahr und Quartal berechnet und dann der niedrigste Umsatz des jeweiligen Jahres ausgegeben.

Was passiert, wenn der Ausgabekontext nicht in der Spalte "Min. pro Jahr" angegeben würde? In diesem Fall würden die Zahlen mit denen der Spalte "Umsatz" übereinstimmen. Warum? Weil der Standardkontext eines Blocks die in diesem Block eingefügten Dimensionen umfasst. Der Minimalumsatz pro Jahr und Quartal ist deshalb derselbe wie der Umsatz pro Jahr und Quartal, weil es nur einen Umsatz für jede Jahr/Quartal-Kombination gibt.

1.4.1.3 Standardberechnungskontexte

Eine Kennzahl hat einen Standard-Berechnungskontext, der sich nach ihrem Platz im Bericht richtet.

Die von einer Kennzahl zurückgegebenen Zahlen hängen von den Dimensionen ab, denen sie zugeordnet ist. Diese Kombination aus Dimensionen stellt den Berechnungskontext dar.

Sie können den Standardkontext mit erweiterter Syntax ändern. Anders ausgedrückt können Sie also die Gruppe von Dimensionen bestimmen, die zum Generieren einer Kennzahl verwendet wird. Dies wird als Definieren des Berechnungskontextes bezeichnet.

Beispiel: Standardkontexte in einem Bericht

In diesem Beispiel wird der Standardberechnungskontext der Kennzahlen in einem einfachen Bericht beschrieben. Der Bericht zeigt den durch Kunden generierten Umsatz an und ist in Sektionen nach Jahr unterteilt.

2005	Gesamt: 8000
Kunde	Umsatz
Meier	1000
Müller	3000
Schulz	4000
Gesamt:	8000
Gesamtumsatz Bericht: 8000	

In der folgenden Tabelle wird der Berechnungskontext der im Bericht enthaltenen Kennzahlen aufgeführt:

Kennzahl	Wert	Kontext
Gesamtumsatz Bericht	20000	Gesamtsumme aller Umsatzwerte des Berichts
Summe der Sektionszellen	8000	Jahr
Summe Kunden	1000, 3000, 4000	Jahr;Kunde
Summe Blockfuß	8000	Jahr

Weitere Informationen

[Berechnungskontexte \[Seite 13\]](#)

[Ändern von Standardberechnungskontexten unter Verwendung der erweiterten Syntax \[Seite 22\]](#)

1.4.1.3.1 Standardkontexte in vertikalen Tabellen

Bei einer vertikalen Tabellen handelt es sich um eine standardmäßige Berichtstabelle, die am oberen Ende Köpfe, anschließend von oben nach unten Daten und schließlich am Ende Füße aufweist.

Für vertikale Tabellen gibt es folgende Standardkontexte:

Einfügeposition der Berechnung	Eingabekontext	Ausgabekontext
Kopf	Die zur Erstellung des Blockkörpers verwendeten Dimensionen und Kennzahlen.	Die Daten werden verdichtet. Die Berechnungsfunktion gibt einen einzelnen Wert zurück.
Im Körper des Blocks	Die zur Erstellung der aktuellen Zeile verwendeten Dimensionen und Kennzahlen.	Identisch mit dem Eingabekontext.
Fuß	Die zur Erstellung des Blockkörpers verwendeten Dimensionen und Kennzahlen.	Die Daten werden verdichtet. Die Berechnungsfunktion gibt einen einzelnen Wert zurück.

1.4.1.3.2 Standardkontexte in horizontalen Tabellen

Für horizontale Tabellen stehen dieselben Standardkontexte wie für vertikale Tabellen zur Verfügung.

Eine horizontale Tabelle ähnelt einer vertikalen Tabelle, mit dem Unterschied, dass die Elemente horizontal, d.h. nebeneinander angeordnet sind.

Die Darstellung der horizontalen Tabelle hängt vom bevorzugten Anzeigegebietsschema ab, das Sie in den Einstellungen von BI-Launchpad ausgewählt haben. In manchen Gebietsschemas, wie beispielsweise dem englischen, werden Schnittstellenelemente von links nach rechts (so genanntes LTR-Gebietsschema) angeordnet, während in anderen Gebietsschemas, wie beispielsweise dem arabischen, die Anzeige von rechts nach links (so genanntes RTL-Gebietsschema) verläuft.

In einem LTR-Gebietsschema werden Köpfe links, Daten von links nach rechts und Füße rechts angezeigt. In einem RTL-Gebietsschema werden Köpfe rechts, Daten von rechts nach links und Füße links angezeigt.

1.4.1.3.3 Standardkontexte in Kreuztabellen

Eine Kreuztabelle zeigt Daten in einer Matrix an. Die Kennzahlen werden an den Schnittpunkten mit den Dimensionen angezeigt.

Für Kreuztabellen gibt es folgende Standardkontexte:

Einfügeposition der Berechnung	Eingabekontext	Ausgabekontext
Kopf	Die zur Erstellung des Blockkörpers verwendeten Dimensionen und Kennzahlen.	Die Daten werden verdichtet. Die Berechnungsfunktion gibt einen einzelnen Wert zurück.
Im Körper des Blocks	Die zur Erstellung des Blockkörpers verwendeten Dimensionen und Kennzahlen.	Identisch mit dem Eingabekontext.

Einfügeposition der Berechnung	Eingabekontext	Ausgabekontext
Fuß	Die zur Erstellung des Blockkörpers verwendeten Dimensionen und Kennzahlen.	Die Daten werden verdichtet. Die Berechnungsfunktion gibt einen einzelnen Wert zurück.
VKörper Fuß	Die zur Erstellung der aktuellen Spalte verwendeten Dimensionen und Kennzahlen.	Die Daten werden verdichtet. Die Berechnungsfunktion gibt einen einzelnen Wert zurück.
HKörper Fuß	Die zur Erstellung der aktuellen Zeile verwendeten Dimensionen und Kennzahlen.	Die Daten werden verdichtet. Die Berechnungsfunktion gibt einen einzelnen Wert zurück.
VFuß	Identisch mit Fuß.	Die Daten werden verdichtet. Die Berechnungsfunktion gibt einen einzelnen Wert zurück.
HFuß	Identisch mit Fuß.	Die Daten werden verdichtet. Die Berechnungsfunktion gibt einen einzelnen Wert zurück.

Beispiel: Standardkontexte in Kreuztabellen

Der folgende Bericht zeigt die Standardkontexte in einer Kreuztabelle:

		FY2000 Q1	FY2000 Q2	FY2000 Q3	FY2000 Q4	1.115.730
France	259.170	61.895	76.555	70.080	50.640	259.170
US	856.560	196.831	189.886	234.574	235.269	856.560
Summe:	1.115.730	258.726	266.441	304.654	285.909	1.115.730

1.4.1.3.4 Standardkontexte in Sektionen

Eine Sektion besteht aus einem Kopf, einem Körper und einem Fuß.

Für eine Sektion gibt es folgende Standardkontexte:

Einfügeposition der Berechnung	Eingabekontext	Ausgabekontext
Körper	Die Dimensionen und Kennzahlen des Berichts, die zur Einschränkung der in der Sektion angezeigten Daten gefiltert wurden.	Die Daten werden verdichtet. Die Berechnungsfunktion gibt einen einzelnen Wert zurück.

Beispiel: Standardkontexte in Sektionen

Der folgende Bericht zeigt die Standardkontexte in Sektionen:

2001	€8.095.814
------	------------

Quartal	Umsatz	Sektion
Q1	€2.660.700	8.095.814
Q2	€2.278.693	8.095.814
Q3	€1.367.841	8.095.814
Q4	€1.788.580	8.095.814
Summe:	€8.095.814	

2002	€13.232.246
------	-------------

Quartal	Umsatz	Sektion
Q1	€3.326.172	13.232.246
Q2	€2.840.651	13.232.246
Q3	€2.879.303	13.232.246
Q4	€4.186.120	13.232.246
Summe:	€13.232.246	

2003	€15.059.143
------	-------------

Quartal	Umsatz	Sektion
Q1	€3.742.989	15.059.143
Q2	€4.006.717	15.059.143
Q3	€3.953.395	15.059.143
Q4	€3.356.041	15.059.143
Summe:	€15.059.143	

1.4.1.3.5 Standardkontexte in Gruppenwechseln

Ein Gruppenwechsel besteht aus einem Kopf, einem Körper und einem Fuß.

Für einen Gruppenwechsel gibt es folgende Standardkontexte:

Einfügeposition der Berechnung	Eingabekontext	Ausgabekontext
Kopf	Aktuelle Instanz des Gruppenwechsels.	Die Daten werden verdichtet. Die Berechnungsfunktion gibt einen einzelnen Wert zurück.
Fuß	Aktuelle Instanz des Gruppenwechsels.	Die Daten werden verdichtet. Die Berechnungsfunktion gibt einen einzelnen Wert zurück.

Beispiel: Standardkontexte in Gruppenwechseln

Der folgende Bericht zeigt die Standardkontexte in einem Gruppenwechsel:

Jahr	Quartal	€8.096.124
	Q1	€2.660.700
	Q2	€2.279.003
	Q3	€1.367.841
	Q4	€1.788.580
2001		
	Summe:	€8.096.124

Jahr	Quartal	€13.232.246
	Q1	€3.326.172
	Q2	€2.840.651
	Q3	€2.879.303
	Q4	€4.186.120
2002		
	Summe:	€13.232.246

1.4.1.4 Ändern von Standardberechnungskontexten unter Verwendung der erweiterten Syntax

Die erweiterte Syntax ermöglicht das Hinzufügen von Kontextoperatoren zu einer Formel oder Kennzahl zur gezielteren Definition des Berechnungskontexts.

Der Kontext einer Kennzahl oder Formel besteht aus einem Eingabekontext und einem Ausgabekontext.

Weitere Informationen

[Erweiterte Syntax und Schlüsselwörter \[Seite 290\]](#)

[Operatoren der erweiterten Syntax \[Seite 23\]](#)

1.4.1.4.1 Operatoren der erweiterten Syntax

Sie geben Eingabe- und Ausgabekontexte explizit mit Kontextoperatoren an.

In der folgenden Tabelle sind die Kontextoperatoren aufgelistet:

Operator	Beschreibung
In	Gibt eine explizite Liste mit Dimensionen an, die im Kontext verwendet werden sollen.
FürJeden	Fügt dem Standardkontext Dimensionen hinzu.
FürAlle	Entfernt Dimensionen aus dem Standardkontext.

Die Operatoren "FürAlle" und "FürJeden" sind nützlich, wenn Sie über einen Standardkontext verfügen, in dem zahlreiche Dimensionen aufgeführt sind. Es ist meistens einfacher, den Kontext unter Verwendung von "FürAlle" und "FürJeden" zu ergänzen oder zu reduzieren, als die Liste mithilfe des Operators "In" ausdrücklich zu definieren.

1.4.1.4.1.1 Der Kontextoperator "In"

Der Kontextoperator In gibt Dimensionen ausdrücklich in einem Kontext an.

Beispiel: Verwenden des Kontextoperators "In" zur ausdrücklichen Angabe der Dimensionen eines Kontextes

In dem folgenden Beispiel wird ein Bericht gezeigt, der die Objekte "Jahr" und "Umsatz" enthält. Ihr Datenprovider enthält darüber hinaus das Objekt "Quartal", allerdings fügen Sie diese Dimension nicht in den Block ein. Stattdessen möchten Sie eine zusätzliche Spalte einfügen, in der der Maximalumsatz pro Quartal für jedes Jahr gezeigt wird. Der Bericht sieht folgendermaßen aus:

Jahr	Umsatz	Maximaler vierteljährlicher Umsatz
2001	\$8,096,123.60	\$2,660,699.50
2002	\$13,232,246.00	\$4,186,120.00
2003	\$15,059,142.80	\$4,006,717.50

Woher die Werte der Spalte "Max. pro Quartal" stammen, können Sie nachvollziehen, indem Sie diesen Block mit dem Block vergleichen, der die Dimension "Quartal" enthält:

Jahr	Quartal	Umsatz
2001	Q1	\$2,660,699.50
2001	Q2	\$2,279,003.00
2001	Q3	\$1,367,841.00
2001	Q4	\$1,788,580.00
	Max:	\$2,660,699.50

Jahr	Quartal	Umsatz
	Q1	\$3,326,172.00
	Q2	\$2,840,651.00
	Q3	\$2,879,303.00
	Q4	\$4,186,120.00
	Max:	\$4,186,120.00

Jahr	Quartal	Umsatz
	Q1	\$3,742,989.00
	Q2	\$4,006,717.50
	Q3	\$3,953,395.00
	Q4	\$3,356,041.00
	Max:	\$4,006,717.50

Die Spalte Max. pro Quartal zeigt den höchsten vierteljährlich in jedem Jahr erwirtschafteten Umsatz. Q4 weist beispielsweise den höchsten Umsatz im Jahr 2002 auf. Die Spalte "Max. pro Quartal" zeigt den Umsatz von Q4 in der Zeile für 2002.

Unter Verwendung des Operators In lautet die Formel für Max. pro Quartal:

```
Max ([Umsatz] In ([Jahr];[Quartal])) In ([Jahr])
```

Mit dieser Formel wird der maximale Umsatz für jede Kombination (Jahr,Quartal) berechnet, und danach wird die Zahl nach Jahr ausgegeben.

ⓘ Hinweis

Da der Standardausgabekontext des Blocks "Jahr" ist, brauchen Sie den Ausgabekontext in dieser Formel nicht ausdrücklich zu definieren.

1.4.1.4.1.2 Der Kontextoperator "Für Jeden"

Der Operator FürJeden fügt Dimensionen zu einem Kontext hinzu.

Beispiel: Verwenden des Operators "FürJeden" zum Hinzufügen von Dimensionen zu einem Kontext

Die folgende Tabelle zeigt den maximalen Umsatz für jedes Quartal in einem Bericht, in dem die Dimension "Quartal" zwar enthalten, aber nicht in den Block eingeschlossen ist:

Jahr	Umsatz	Maximaler vierteljährlicher Umsatz
2001	8096123.60	2660699.50
2002	13232246.00	4186120.00
2003	15059142.80	4006717.50

Es ist möglich, eine Formel für die Spalte "Max. pro Quartal" zu erstellen, in der der Operator "FürJeden" nicht enthalten ist:

```
Max ([Umsatz] In ([Jahr];[Quartal])) In ([Jahr])
```

Mithilfe des Kontextoperators "FürJeden" können Sie über die folgende Formel dasselbe Ergebnis erzielen:

```
Max ([Umsatz] FürJeden ([Quartal])) In ([Jahr])
```

Warum? Da die Dimension "Jahr" der Standardeingabekontext im Block ist. Indem Sie den Operator "FürJeden" verwenden, fügen Sie die Dimension "Quartal" zum Kontext hinzu und definieren dadurch den Eingabekontext von ([Jahr];[Quartal]).

1.4.1.4.1.3 Der Kontextoperator "FürAlle"

Mit dem Kontextoperator `ForAll` werden Dimensionen aus einem Kontext entfernt.

Beispiel: Verwenden des Operators "FürAlle" zum Entfernen von Dimensionen aus einem Kontext

Sie verfügen über einen Bericht, der die Objekte "Jahr", "Quartal" und "Umsatz" enthält. Sie möchten eine Spalte hinzufügen, in der der Gesamtumsatz pro Jahr angezeigt wird, wie im folgenden Block dargestellt:

Year	Quarter	Sales revenue	Yearly Revenue
2004	Q1	\$2,660,700	\$8,096,124
2004	Q2	\$2,279,003	\$8,096,124
2004	Q3	\$1,367,841	\$8,096,124
2004	Q4	\$1,788,580	\$8,096,124
2005	Q1	\$3,326,172	\$13,232,246
2005	Q2	\$2,840,651	\$13,232,246
2005	Q3	\$2,879,303	\$13,232,246
2005	Q4	\$4,186,120	\$13,232,246
2006	Q1	\$3,742,989	\$15,059,143
2006	Q2	\$4,006,718	\$15,059,143
2006	Q3	\$3,953,395	\$15,059,143
2006	Q4	\$3,356,041	\$15,059,143

Um den Umsatz mit Bezug auf das Jahr zu berechnen, muss der Eingabekontext gleich (Jahr) sein. Die Standardeinstellung ist (Jahr; Quartal). Sie können "Quartal" aus dem Eingabekontext entfernen, indem Sie in der Formel `FürAlle ([Quartal])` angeben, was folgende Formel ergibt:

```
Summe([Umsatz] FürAlle ([Quartal]))
```

Sie können auch den Operator "In" verwenden, um dasselbe Ergebnis zu erzielen. In diesem Fall würde die Formel folgendermaßen aussehen:

```
Summe([Umsatz] In ([Jahr]))
```

Anstatt "Quartal" zu entfernen und "Jahr" zu belassen, wird in dieser Version der Formel "Jahr" ausdrücklich als Kontext definiert.

1.5 Berechnen von Werten mit intelligenten Kennzahlen

1.5.1 Berechnen von Werten mit intelligenten Kennzahlen

Intelligente Kennzahlen sind Kennzahlen, deren Werte in der Datenbank (relational oder OLAP) berechnet werden, auf der ein Universum basiert.

Sie unterscheiden sich von herkömmlichen Kennzahlen, die auf der Grundlage der von der Datenbank zurückgegebenen detaillierten Werte berechnet werden. Die von intelligenten Kennzahlen zurückgegebenen Daten werden auf eine Art und Weise aggregiert, die ursprünglich nicht von der Web-Intelligence-Komponente von SAP BusinessObjects Business Intelligence unterstützt wird.

Abfragen, die intelligente Kennzahlen enthalten, berechnen die Kennzahlen in allen in einem Bericht erforderlichen Berechnungskontexten. Diese Kontexte können sich ändern, wenn sich der Bericht ändert. Folglich wird die Abfrage bei jeder Datenregenerierung geändert, nachdem sich die erforderlichen Kontexte geändert haben.

Wenn Sie einen solchen Bericht bearbeiten, wird automatisch die Meldung #ZUREGENERIEREN eingefügt, die Sie darauf hinweist, dass der Bericht regeneriert werden muss, damit die Änderungen berücksichtigt werden. Sie können den Bericht automatisch aktualisieren, indem Sie im Dialogfeld "Dokumenteigenschaften" die Option "Automatische Regenerierung aktivieren" auswählen.

Hinweis

Die Kennzahldelegation ist statisch und wird auf Basis der Berichtsdefinition zur Entwurfszeit definiert. In manchen Fällen (die Formel basiert auf "if [Auswahl]= 1 then [Dimension 1] else [Dimension 2]") ist der Dimensionskontext zur Laufzeit variabel. In diesem Fall kann das System die Kennzahlberechnung nicht delegieren und gibt einen leeren Wert zurück.

Intelligente Kennzahlen verhalten sich anders als klassische Kennzahlen, die eine Grundmenge von Aggregationsfunktionen (Max, Min, Anzahl, Summe, Durchschnitt) unterstützen, die ohne Hilfe der Datenbank in allen Kontexten berechnet werden können. Beispiel: Sie erstellen eine Abfrage, die die Dimensionen [Land] und [Region] und die Kennzahl [Umsatz] enthält (durch die die Umsatzsumme berechnet wird). Dann werden zunächst Land, Region und Umsatz in einem Block angezeigt. Wenn Sie "Region" aus dem Block entfernen, kann der Gesamtumsatz für jedes Land noch immer ohne Datenregenerierung berechnet werden, indem die Umsätze aller Regionen in dem Land summiert werden. In diesem Fall erfordert eine intelligente Kennzahl eine Datenregenerierung.

Berechnungskontexte werden in der generierten Abfrage durch Gruppierungsmengen dargestellt.

1.5.1.1 Gruppierungsmengen und intelligente Kennzahlen

Eine Gruppierungsmenge entspricht einer Gruppe von Dimensionen, die ein Ergebnis für eine Kennzahl generiert.

Die generierte SQL, die Daten in einer intelligenten Kennzahl zurückgibt, beinhaltet Gruppierungsmengen für alle Aggregationen der Kennzahl, die in dem Bericht berücksichtigt sind.

Beispiel: Gruppierungsmengen in einer Abfrage

Eine Abfrage enthält die Dimensionen [Land], [Region], [Stadt] und die intelligente Kennzahl [Umsatz]. Diese Objekte beinhalten die folgenden Gruppierungsmengen, um den Umsatz in allen möglichen Kontexten zu berechnen:

- Gesamtwert der intelligenten Kennzahl
- Wert der intelligenten Kennzahl nach (Land, Region, Stadt)
- Wert der intelligenten Kennzahl nach (Land, Stadt)
- Wert der intelligenten Kennzahl nach (Stadt)
- Wert der intelligenten Kennzahl nach (Region, Stadt)

- Wert der intelligenten Kennzahl nach (Region)
- Wert der intelligenten Kennzahl nach (Land, Region)
- Wert der intelligenten Kennzahl nach (Land)

Wenn die Datenbank `UNION` unterstützt, werden die einzelnen Gruppierungsmengen in der generierten SQL in einer `UNION`-Bedingung dargestellt.

Die Gruppierungsmengen werden entsprechend den vom Bericht benötigten Berechnungskontexten aktualisiert, die sich in Anpassung an Änderungen in der Berichtsstruktur ändern können.

1.5.1.1.1 Verwalten von Gruppierungsmengen

Wenn Sie erstmalig eine Abfrage erstellen und ausführen, die intelligente Kennzahlen enthält, umfasst die generierte SQL die Gruppierungsmenge, die zur Berechnung der intelligenten Kennzahlen auf der höchsten von den Abfrageobjekten unterstützten Detailebene erforderlich ist.

Beispiel: Sie erstellen eine Abfrage, die die Dimensionen [Land], [Region] und [Stadt] und die intelligente Kennzahl [Umsatz] enthält. Die Gruppierungsmenge (Stadt, Region, Land) wird dann in die generierte SQL eingefügt. Die detaillierteste Gruppierungsmenge wird stets in die SQL eingefügt. Andere Gruppierungsmengen werden als Reaktion auf Änderungen im Bericht hinzugefügt und entfernt.

Wenn Sie die Dimension [Stadt] aus dem Block entfernen, wird die Gruppierungsmenge (Land, Region) benötigt, um die Umsatzwerte zurückzugeben. Diese Gruppierungsmenge ist noch nicht in der Abfrage-SQL verfügbar, sodass `#ZUREGENERIEREN` in den Zellen [Umsatz] angezeigt wird. Wenn Sie die Daten regenerieren, wird `#ZUREGENERIEREN` durch die Umsatzwerte ersetzt.

Wenn Sie anschließend die Dimension [Stadt] im Block ersetzen, wird die Gruppierungsmenge (Land, Region) nicht mehr benötigt. Sie wird aus der Abfrage-SQL entfernt und ihre Werte werden bei der nächsten Regenerierung der Daten verworfen.

Bei jeder Regenerierung von Berichtsdaten werden Gruppierungsmengen entsprechend den von dem Bericht benötigten Berechnungskontexten einbezogen oder verworfen.

Unter bestimmten Umständen kann der Wert einer intelligenten Kennzahl nicht angezeigt werden. In diesem Fall wird in den Kennzahlzellen `#NICHTVERFÜGBAR` angezeigt.

1.5.1.2 Intelligente Kennzahlen und Analysetiefe

Wenn Sie eine Abfrage mit einer Analysetiefe erstellen, enthält die anfängliche Gruppierungsmenge die Ergebnisobjekte, aber keine Tiefenbereichsobjekte.

Von der Abfrage werden nicht alle möglichen Gruppierungsmengen aus der Kombination von Ergebnisobjekten plus Tiefenbereichsobjekten generiert.

Beispiel: Eine Abfrage mit einer Analysetiefe und einer intelligenten Kennzahl

Eine Abfrage verfügt über die Ergebnisobjekte [Land] und [Umsatz]. Die Analysetiefe enthält die Dimensionen [Region] und [Stadt]. Wenn Sie die Abfrage ausführen, enthält ihre SQL die Gruppierungsmenge (Land), sie und zeigt [Land] und [Umsatz] in einem Block an.

1.5.1.3 Intelligente Kennzahlen und SQL

1.5.1.3.1 Gruppierungsmengen und der UNION-Operator

Einige Datenbanken unterstützen Gruppierungsmengen ausdrücklich mit dem Operator `GROUPING SETS`.

Wenn Sie eine Abfrage erstellen, die intelligente Kennzahlen enthält, verwendet die generierte SQL mehrere Ergebnismengen und den `UNION`-Operator, um die Auswirkungen von `GROUPING SETS` zu simulieren.

Beispiel: Mit dem UNION-Operator abgerufene Gruppierungsmengen

In diesem Beispiel wird eine Abfrage beschrieben, die die Dimensionen [Land], [Region], [Stadt] und die intelligente Kennzahl [Umsatz] enthält.

📌 Hinweis

Aus Gründen der Einfachheit berechnet die intelligente Kennzahl eine Summe. In der Praxis ist für diese Aggregation keine intelligente Kennzahl erforderlich, da Web Intelligence systemeigene Unterstützung von Summen bietet.

Wenn die Abfrage erstmalig ausgeführt wird, lautet die Gruppierungsmenge (Land, Region, Stadt). Die gesamte SQL-Abfrage gibt diese Gruppierungsmenge zurück, sodass kein `UNION`-Operator in der SQL benötigt wird.

Wenn Sie die Dimension [Stadt] aus der Tabelle entfernen, wird die Gruppierungsmenge (Land, Region) benötigt, um den Umsatz anzuzeigen (der durch #ZUREGENERIEREN dargestellt wird). Nach der Datenregenerierung lautet die SQL wie folgt:

```
SELECT
  SELECT
    0 AS GID,
    country.country_name,
    region.region_name,
    NULL,
    sum(city.revenue)
  FROM
    country,
    region,
    city
  WHERE
    ( country.country_id=region.country_id )
    AND ( region.region_id=city.region_id )
  GROUP BY
```

```

country.country_name,
region.region_name
UNION
SELECT
1 AS GID,
country.country_name,
region.region_name,
city.city_name,
sum(city.revenue)
FROM
country,
region,
city
WHERE
( country.country_id=region.country_id )
AND ( region.region_id=city.region_id )
GROUP BY
country.country_name,
region.region_name,
city.city_name

```

Jede Gruppierungsmenge wird durch eine SELECT-Anweisung dargestellt und verfügt jeweils über eine eigene ID (die GID-Spalte). Gruppierungsmengen, die keinen vollständigen Satz von Dimensionen enthalten, verfügen über leere Spalten (SELECT ' '), da jede SELECT-Anweisung in einer Abfrage, die den UNION-Operator enthält, über dieselbe Anzahl von Spalten verfügen muss.

Wenn Sie dem Bericht einen neuen Block hinzufügen, der [Land] und [Umsatz] enthält, wird die Gruppierungsmenge (Land) benötigt. Die generierte SQL enthält jetzt die folgenden drei Gruppierungsmengen:

```

SELECT
0 AS GID,
country.country_name,
region.region_name,
NULL,
sum(city.revenue)
FROM
country,
region,
city
WHERE
( country.country_id=region.country_id )
AND ( region.region_id=city.region_id )
GROUP BY
country.country_name,
region.region_name
UNION
SELECT
1 AS GID,
country.country_name,
NULL,
NULL,
sum(city.revenue)
FROM
country,
city,
region
WHERE
( country.country_id=region.country_id )
AND ( region.region_id=city.region_id )
GROUP BY
country.country_name
UNION
SELECT
2 AS GID,
country.country_name,
region.region_name,

```

```

    city.city_name,
    sum(city.revenue)
FROM
    country,
    region,
    city
WHERE
    ( country.country_id=region.country_id )
    AND ( region.region_id=city.region_id )
GROUP BY
    country.country_name,
    region.region_name,
    city.city_name

```

1.5.1.4 Intelligente Kennzahlen und Formeln

1.5.1.4.1 Intelligente Kennzahlen und Dimensionen mit Formeln

Wenn eine Formel oder Variable im Berechnungskontext einer intelligenten Kennzahl als Dimension angezeigt wird und die Formel bestimmt, welche Gruppierungsmenge von der Kennzahl benötigt wird, können die Werte für die intelligente Kennzahl angezeigt werden.

Intelligente Kennzahlen und Dimensionen geben jetzt beispielsweise Werte für folgende Elemente zurück:

- Eine mit dem Hyperlink-Assistenten erstellte URL
- Einfache Verkettung für eine Dimension (oder Entfernung von leeren Zellen)
- Bei Verwendung von FormatDatum für [Datum]

ⓘ Hinweis

Die Meldung #NICHTVERFÜGBAR wird noch immer für folgende Funktionen zurückgegeben: FürJeden, FürAlle, In, Wo, Rangfolge, Vorherige, RelativerWert, DatumRelativ, ZeitDim und in Aggregationsfunktionen, wenn Min, Max, Letzte oder Erste in der Formel verwendet werden: if ([Auswahl] =1) then [Dim1] else [Dim2]

1.5.1.4.2 Intelligente Kennzahlen in Formeln

Intelligente Kennzahlen, die in Formeln eingeschlossen sind, können auch dann Werte zurückgeben, wenn die Formel einen anderen Berechnungskontext als den von der Position der Formel vorausgesetzten erfordert.

Ein Bericht enthält beispielsweise folgenden Block:

Land	Region	Umsatz
USA	Norden	10000
USA	Süden	15000

Land	Region	Umsatz
USA	Ost	14000
USA	West	12000

Angenommen, Sie nehmen eine zusätzliche Tabellenspalte mit folgender Formel auf:

```
[Umsatz] FürAlle ([Region])
```

Der anfängliche Wert der Spalte ist #ZUREGENERIEREN, da die Formel, die Regionen aus der Berechnung ausschließt, die Gruppierungsmenge (Land) erfordert. Durch Regenerieren der Daten wird die Gruppierungsmenge (Land) zu der Abfrage hinzugefügt, und die Werte der Kennzahl werden angezeigt.

Weitere Informationen

[Der Kontextoperator "FürAlle" \[Seite 25\]](#)

1.5.1.5 Intelligente Kennzahlen und Filter

1.5.1.5.1 Einschränkungen bei intelligenten Kennzahlen und Filtern

Eine intelligente Kennzahl kann im Körper einer Tabelle ausgewertet werden, wenn kein Filter in der Tabelle oder im übergeordneten Kontext (Berichtsfilter) vorliegt.

In der folgenden Tabelle wird beschrieben, wie intelligente Kennzahlen ausgewertet werden, wenn Filter vorhanden sind.

Auswertung von intelligenten Kennzahlen bei Berichten mit Filtern

Filterziel	Auswertung der intelligenten Kennzahl
Kennzahl	Die intelligente Kennzahl wird korrekt ausgewertet, allerdings werden einige Zeilen aus der Tabelle entfernt.
Dimension, die bereits Teil der Tabellenachse ist	Die intelligente Kennzahl wird korrekt ausgewertet, allerdings werden einige Zeilen aus der Tabelle entfernt. Die intelligente Kennzahl kann ausgewertet werden, da auf die Filterung keine Aggregation folgt.
Dimension, die nicht Teil der Tabellenachse ist, und einwertiger Filteroperand (der Filter gibt genau einen Wert/eine Zeile zurück)	Die intelligente Kennzahl wird korrekt ausgewertet. Die intelligente Kennzahl kann ausgewertet werden, da auf die Filterung keine Aggregation folgt.
Dimension, die nicht Teil der Tabellenachse ist, und mehrwertiger Filteroperand (der Filter kann viele Werte/Zeilen zurückgeben)	Die intelligente Kennzahl kann nicht ausgewertet werden (#NICHTVERFÜGBAR wird angezeigt), da in diesem Fall die Filterung vor der Aggregation erfolgt, und für eine Zeile der Tabelle die Aggregation erforderlich ist.

1.5.1.5.2 Intelligente Kennzahlen und Filter für Dimensionen

Wenn ein Filter mit mehreren Werten auf eine Dimension angewendet wird, von der der Wert einer intelligenten Kennzahl abhängig ist, die Dimension jedoch nicht explizit im Berechnungskontext der Kennzahl angezeigt wird, kann für die intelligente Kennzahl kein Wert zurückgegeben werden, und für die Zelle wird #NICHTVERFÜGBAR angezeigt.

Dies gilt auch, wenn ein Berichtsfilter aus einem Eingabesteuerelement stammt.

#NICHTVERFÜGBAR wird angezeigt, da die Kennzahl im Bericht gefiltert und dann aggregiert werden muss, aber eine intelligente Kennzahl kann nicht aggregiert werden, nachdem ein Filter auf Berichtsebene angewendet wurde. Die Kennzahl könnte berechnet werden, indem ein Abfragefilter auf die generierte SQL angewendet wird. Diese Lösung kann jedoch Auswirkungen auf andere Berichte haben, die auf der gleichen Abfrage basieren.

📘 Hinweis

Ein Filter mit mehreren Werten filtert mehrere Werte mit Operatoren wie "Größer als", "In Liste" oder "Kleiner als". Sie können Filter mit einem Wert wie "Gleich" anwenden, ohne dass der Fehler #NICHTVERFÜGBAR generiert wird.

📘 Hinweis

Es gibt eine Umgehungslösung für Fälle, bei denen keine Aggregation erforderlich ist: Definieren Sie die Formel als Variable, deren Qualifikation eine Kennzahl ist, und stellen Sie sicher, dass die verwendete Dimension im Block mit der Variablen enthalten ist (Sie können diese Spalte ausblenden, um eine bessere Anzeige zu erhalten).

Beispiel: Eine intelligente Kennzahl und ein Filter für eine Dimension

Eine Abfrage enthält die Dimensionen "Land" und "Produkt" und die intelligente Kennzahl "Umsatz". "Land" und "Umsatz" werden in einem Block angezeigt. Wenn Sie einen Filter anwenden, der die Werte von "Produkt" auf "Kleider" oder "Jacken" beschränkt, wird in den "Umsatz"-Zellen #NICHTVERFÜGBAR angezeigt.

Land	Umsatz
Frankreich	#NICHTVERFÜGBAR
USA	#NICHTVERFÜGBAR
Summe:	#NICHTVERFÜGBAR

Wenn Sie "Produkt" nur auf "Jacken" beschränken, werden die Werte angezeigt.

Land	Umsatz
USA	971.444
Summe:	971.444

Hinweis

Ein Mehrfachwert-Filter für "Land" gibt die Ergebnisse zurück, da sich der Filter in der Tabelle befindet.

1.5.1.5.3 Filtern intelligenter Kennzahlen

Der Wert im Tabellenfuß muss die Aggregation dessen darstellen, was dem Benutzer in der Tabelle angezeigt wird.

Wenn das, was dem Benutzer angezeigt wird, lokal gefiltert wird, kann das System nicht die delegierte Aggregation der lokal gefilterten Inhalte zurückgeben.

Beispiel: Filtern einer intelligenten Kennzahl

Land	BestellwertDel
Brasilien	28.833,36
China	51.384,33
Frankreich	68.630,22
USA	3.529.511,14
Gesamt:	3.678.359,05
Summe:	3.678.359,05

Wenn die Daten in der folgenden Tabelle nach "BestellwertDel" > 60.000 gefiltert werden, ergibt sich Folgendes:

In der Tabelle werden die Zeilen angezeigt, für die "BestellwertDel" im Kontext der Tabelle (pro Land) größer als 60.000 ist:

Die in der Fußzeile berechnete Summe ist die Summe der sichtbaren Zeilen.

"Gesamt" gibt #NICHTVERFÜGBAR zurück, weil die Berechnung die Aggregation ans Backend übergibt. Da Web Intelligence jedoch eine lokale Filterung ausgeführt hat, kann die Aggregation nicht delegiert werden.

Land	BestellwertDel
Frankreich	68.630,22
USA	3.529.511,14
Gesamt:	#NICHTVERFÜGBAR
Summe:	3.598.141,36

1.5.1.5.4 Intelligente Kennzahlen und Drillfilter

Ein Drill-Filter ist ein Filter mit einem einzigen Wert.

Über die Drill-Leiste können Sie Drill-Vorgänge direkt durchführen.

1.5.1.5.5 Intelligente Kennzahlen und verschachtelte ODER-Filter

Verschachtelte ODER-Filter, in denen mindestens eine der gefilterten Dimensionen nicht in einem Block angezeigt wird, lösen die Fehlermeldung #NICHTVERFÜGBAR für eine intelligente Kennzahl im Block aus.

Dies ist darauf zurückzuführen, dass die intelligente Kennzahl nach einigen lokalen Verarbeitungsschritten (zum Beispiel Filterung, eine bestimmte Web-Intelligence-Formel) lokal aggregiert werden muss und dies nicht delegiert wird.

1.6 Funktionen, Operatoren und Schlüsselwörter

1.6.1 Funktionen

Formelfunktionen werden in mehrere Kategorien unterteilt.

📌 Hinweis

In den folgenden Sprachen werden die Funktionen nicht übersetzt: Chinesisch, Japanisch, Ungarisch, Polnisch, Türkisch, Thai und Russisch. Sie werden auf der Oberfläche auf Englisch angezeigt.

Kategorie	Beschreibung
Aggregation	Aggregierte Daten (z.B. summierte oder gemittelte Werte).
Zeichen	Bearbeiten von Zeichenfolgen
Datum und Uhrzeit	Gibt Daten mit Bezug auf Datum und Uhrzeit zurück.
Dokument	Gibt Daten mit Bezug auf ein Dokument zurück.
Datenprovider	Gibt Daten mit Bezug auf den Datenprovider eines Dokuments zurück.
Logisch	Gibt WAHR oder FALSCH zurück.
Numerisch	Gibt numerische Daten zurück.
Sonstige	Funktionen, die nicht einer der oben genannten Kategorien angehören.
Menge	Gibt Mengen von Elementen von Hierarchien zurück

1.6.1.1 Benutzerdefinierte Formate

Sie können festlegen, wie numerische Werte oder Datums-/Zeitwerte über benutzerdefinierte Formate angezeigt werden sollen.

Die nachstehend abgebildete Tabelle enthält eine Liste der Token, die Sie bei der Erstellung benutzerdefinierter Formate verwenden können:

Format-Token	Beschreibung	Beispiel
#	Die entsprechende Ziffer. Wenn die Zahl weniger Ziffern umfasst, als #-Zeichen für die Formatangabe verwendet werden, werden keine führenden Nullen eingefügt.	'12345' im Format #, ##0 ergibt '12,345' (wenn in Ihrem Gebietsschema ein Komma als Gruppierungstrennzeichen definiert ist) oder '12 345' (wenn in Ihrem Gebietsschema ein Leerzeichen als Gruppierungstrennzeichen definiert ist).
0	Die entsprechende Ziffer. Wenn die Zahl weniger Ziffern umfasst, als 0-Zeichen für die Formatangabe verwendet werden, werden eine oder mehrere führende Nullen vor der Zahl eingefügt.	'123' im Format #0, 000 ergibt '0.123'.
,	Gruppierungstrennzeichen wie in Ihrem Gebietsschema definiert.	'1234567' im Format #, ##0 ergibt '1,234,567' (wenn in Ihrem Gebietsschema ein Komma als Gruppierungstrennzeichen definiert ist) oder '1 234 567' (wenn in Ihrem Gebietsschema ein geschütztes Leerzeichen als Gruppierungstrennzeichen definiert ist).
.	Dezimaltrennzeichen wie in Ihrem Gebietsschema definiert.	'12.34' im Format #. #0 ergibt '12.34' (wenn in Ihrem Gebietsschema ein Punkt als Dezimaltrennzeichen definiert ist) oder '12,34' (wenn in Ihrem Gebietsschema ein Komma als Dezimaltrennzeichen definiert ist).
[%] %	Zeigt ein Prozentzeichen (%) nach dem Ergebnis an und multipliziert das Ergebnis mit 100.	0,50 wird zu 50 %.
%	Zeigt ein Prozentzeichen (%) nach dem Ergebnis an, multipliziert das Ergebnis aber nicht mit 100.	0,50 wird zu 0,50 %
	Geschütztes Leerzeichen ()	'1234567' im Format # ##0 ergibt '1234 567'.

Format-Token	Beschreibung	Beispiel
1, 2, 3, a, b, c, \$, £, €(usw.)	Alphanumerische Zeichen.	'705.15' ergibt '\$705.15' im Format \$#. #0 oder '705,15 €' im Format #, #0 €.
<div> <div>📌 Hinweis</div> <p>Alphanumerische Zeichen sollten durch einfache Anführungszeichen getrennt werden. Andernfalls können sie als Formatierungszeichen interpretiert werden. Das Format # # ergibt beispielsweise '123 4', während '# #' den Wert '# 1234' ergibt.</p> </div>		
COMPACT	Runden eines numerischen Werts und Anzeigen des Werts mit einem abgekürzten Suffix. Das Suffix ist vom Gebietsschema abhängig.	-1234 im Format COMPACT ergibt -1K im Gebietsschema „Englisch (USA)“.
LONG_COMPACT	Runden eines numerischen Werts und Anzeigen des Werts mit einem Suffix. Das Suffix wird im vollständigen Namen angezeigt und ist vom Gebietsschema abhängig.	-1234 im Format COMPACT ergibt -1 Tausend im Gebietsschema „Englisch (USA)“.
CURRENCY	Anzeigen des Werts als Währung und Anwenden der Regeln des bevorzugten Anzeigegebietsschemas für Geldwerte.	-1234 im Format CURRENCY ergibt -¤1,234.00 im Gebietsschema „Englisch (USA)“.
ACCOUNTING_CURRENCY	Anzeigen des Werts als Währung und Anwenden der Regeln des bevorzugten Anzeigegebietsschemas für Kontierungswerte.	-1234 im Format ACCOUNTING_CURRENCY ergibt (¤1,234.00) im Gebietsschema „Englisch (USA)“.
CURRENCY_COMPACT	Anzeigen des Werts als Währung mit einem abgekürzten Suffix.	-1234 im Format CURRENCY_COMPACT ergibt -¤1K im Gebietsschema „Englisch (USA)“.
[MIN_DEC : n]	Wird mit COMPACT, LONG_COMPACT und CURRENCY_COMPACT verwendet, um die Mindestanzahl der anzuzeigenden Dezimalstellen zu definieren. Sein Wert ist standardmäßig 0.	-1234 im Format COMPACT [MIN_DEC : 5] ergibt -1.23400K im Gebietsschema „Englisch (USA)“.
[MAX_DEC : n]	Wird mit COMPACT, LONG_COMPACT und CURRENCY_COMPACT verwendet, um die Höchstzahl der anzuzeigenden Dezimalstellen zu definieren. Sein Wert ist standardmäßig gleich MIN_DEC.	-1234 im Format COMPACT [MAX_DEC : 2] ergibt -1.23K im Gebietsschema „Englisch (USA)“.
[CURRENCY : c]	Wird mit CURRENCY, ACCOUNTING_CURRENCY und CURRENCY_COMPACT verwendet, um das Währungssymbol zu definieren. Sein Wert ist standardmäßig ¤.	-1234 im Format CURRENCY ergibt -1,234.00\$ im Gebietsschema „Englisch (USA)“.

Format-Token	Beschreibung	Beispiel
[Red], [Blue], [Green], [Yellow], [Gray], [White], [Dark Red], [Dark Blue], [Dark Green]	Der Wert in der angegebenen Farbe.	'150' wird im Format #,##0[Red] in Rot angegeben, im Format #,##0[Blue] in Blau.
Tages-/Datumstoken (day, date)		
d	Zahl des Tages im Monat ohne führende Null. Wenn das Datum für den Tag weniger als zwei Zeichen umfasst, wird das Datum ohne vorangestellte Null angezeigt.	Der erste Tag des Monats im Format d ergibt '1'.
dd	Zahl des Tages mit führender Null. Wenn das Datum für den Tag weniger als zwei Zeichen umfasst, wird das Datum mit einer vorangestellten Null angezeigt.	Der erste Tag des Monats im Format dd ergibt '01'.
ddd	Abgekürzter Name des Tages. Der erste Buchstabe wird großgeschrieben, wenn im ausgewählten Gebietschema großgeschriebene Tagesnamen verwendet werden.	'Montag' im Format ddd ergibt 'Mon' auf Englisch und 'lun' auf Französisch.
Dddd	Die Großschreibung des Tagesnamens wird in jedem Gebietsschema erzwungen.	'Montag' im Format Dddd ergibt 'Mon' auf Englisch und 'Lun' auf Französisch.
dddd	Vollständiger Name des Tages. Der erste Buchstabe wird großgeschrieben, wenn im ausgewählten Gebietschema großgeschriebene Tagesnamen verwendet werden.	'Montag' im Format dddd ergibt 'Monday' auf Englisch. Auf Französisch lautet der Tag 'lundi'.
DDDD	Vollständiger Name des Tages in Großbuchstaben.	'Montag' im Format DDDD ergibt 'MONDAY' auf Englisch. Auf Französisch lautet der Tag 'LUNDI'.
dddd dd	Wochentag gefolgt von einem Leerzeichen und der Zahl des Tages.	'Montag' im Format dddd dd ergibt 'Montag 01'.
Kalender-Token (week, month, year)		
M	Zahl des Monats ohne führende Null. Wenn die Zahl für den Monat weniger als zwei Zeichen umfasst, wird die Zahl ohne vorangestellte Null angezeigt.	'Januar' im Format M ergibt '1'.
MM	Zahl des Monats mit führender Null. Wenn die Zahl für den Monat weniger als zwei Zeichen umfasst, wird die Zahl mit vorangestellter Null angezeigt.	'Januar' im Format MM ergibt '01'.

Format-Token	Beschreibung	Beispiel
mmm	Abgekürzter Name des Monats. Der erste Buchstabe wird großgeschrieben, wenn im ausgewählten Gebietsschema Großschreibung verwendet wird.	'Januar' im Format mmm ergibt 'Jan' auf Englisch. Auf Französisch lautet der Monat 'jan'.
Mmmm	Abgekürzter Name des Monats. Der erste Buchstabe wird in allen Gebietsschemas großgeschrieben.	'Januar' im Format mmmm ergibt 'Jan' auf Englisch. Auf Französisch lautet der Monat 'Jan'.
mmmm	Vollständiger Name des Monats. Der erste Buchstabe wird großgeschrieben, wenn im ausgewählten Gebietsschema Großschreibung verwendet wird.	'Januar' im Format mmmm ergibt 'January' auf Englisch und 'janvier' auf Französisch.
MMMM	Vollständiger Name des Monats in Großbuchstaben.	'Januar' im Format MMMM ergibt 'JANUARY' auf Englisch und 'JANVIER' auf Französisch.
ww	Nummer der Kalenderwoche innerhalb des Jahres	Das Format ww ergibt '02' für den 9. Februar 2015, da dieser in die siebte Woche des Jahres 2015 fällt.
w	Nummer der Kalenderwoche des Jahres ohne führende Null	Das Format w ergibt '2' für den 9. Februar 2015, da dieser in die siebte Woche des Jahres 2015 fällt.
W	Nummer der Woche innerhalb des Monats	Das Format W ergibt '2' für den 9. Januar 2015, da dieser in die zweite Januarwoche fällt.
yy	Die letzten beiden Ziffern für das Jahr.	'2003' im Format yy ergibt '03'.
yyyy	Alle vier Ziffern für das Jahr.	'2003' im Format yyyy ergibt '2003'.
Zeit-Token	(hours, minutes, seconds, am/pm)	
hh:mm:ss a	Stundenangabe ohne führende Null, Minuten und Sekunden mit führender Null. Das Zeichen a steht für die im englischsprachigen Raum gebräuchliche Angabe von AM (vormittags) oder PM (nachmittags) nach der Zeitanzeige.	'21:05:03' im Format hh:mm:ss a ergibt '9:05:03 PM' im englischen Gebietsschema.
H	Stundenangabe gemäß der 24-Stundenuhr, beginnend bei 0. Aus einer Ziffer bestehenden Stunden wird keine Null vorangestellt.	'21:00' im Format H ergibt '21'. Mögliche Werte sind 0 - 23.
HH	Stundenangabe gemäß der 24-Stundenuhr, beginnend bei 0.	'21:00' im Format HH ergibt '21'. Mögliche Werte sind 00 - 23.
k	Stundenangabe gemäß der 24-Stundenuhr, beginnend bei 1. Aus einer Ziffer bestehenden Stunden wird keine Null vorangestellt.	'21:00' im Format k ergibt '21'. Mögliche Werte sind 1 - 24.
kk	Stundenangabe gemäß der 24-Stundenuhr, beginnend bei 01.	'21:00' im Format kk ergibt '21'. Mögliche Werte sind 01 - 24.

Format-Token	Beschreibung	Beispiel
hh	Stundenangabe gemäß der 12-Stunden- de Uhr.	'21:00' im Format hh ergibt '09'.
HH:mm	Stunden- und Minutenabgabe mit einer Null vor einer einstelligigen Stundenan- gabe.	'07:15' im Format HH:mm ergibt '07:15'.
HH:mm:ss	Stunden-, Minuten- und Sekundenab- gabe mit einer Null vor einer einstelligen Stundenangabe.	'07:15' im Format HH:mm:ss ergibt '07:15:00'.
mm:ss	Minuten- und Sekundenabgabe mit ei- ner Null vor einer einstelligigen Stunden- angabe.	'07:15:03' im Format mm:ss ergibt '15:03'.
x	Zeitzone in Stunden.	-08, +0530, +00
xx	Zeitzone in Stunden, Minuten.	-0800, +0530, +0000
xxx	Zeitzone in Stunden:Minuten.	-08:00, +05:30, +00:00
xxxx	Zeitzone in Stunden, Minuten, Sekun- den.	-0800, +075228, +0000
xxxxx	Zeitzone in Stunden:Minuten:Sekun- den.	-08:00, +07:52:28, +00:00
X	Wie im Fall von x, außer dass "Z" ange- zeigt wird, wenn die Zeitzone UTC ist.	-08, +0530, Z
XX	Wie im Fall von xx, außer dass "Z" an- gezeigt wird, wenn die Zeitzone UTC ist.	-0800, +0530, Z
XXX	Wie im Fall von xxx, außer dass "Z" angezeigt wird, wenn die Zeitzone UTC ist.	-08:00, +05:30, Z
XXXX	Wie im Fall von xxxx, außer dass "Z" angezeigt wird, wenn die Zeitzone UTC ist.	-0800, +075228, Z
XXXXX	Wie im Fall von xxxxx, außer dass "Z" angezeigt wird, wenn die Zeitzone UTC ist.	-08:00, +07:52:28, Z
VV	Zeitzone-ID.	Amerika/Los_Angeles
O	Zeitzone in Stunden von GMT.	GMT-8
OOOO	Zeitzone in Stunden und Minuten von GMT (ersetzt das bisherige Format "Z").	GMT-08:00

Format-Token	Beschreibung	Beispiel
z	Bezeichnung der Zeitzone. Wenn die Zeitzone keine Bezeichnung hat, zeigt z den Zeitunterschied an.	CEST oder PST, wenn die Zeitzone eine Bezeichnung hat. Wenn der Zeitzone keine Bezeichnung zugewiesen ist, zeigt z den Zeitunterschied wie folgt an: +02, +530 usw.
[TIMEZONE:t]	Wird verwendet, um die Zeitzone eines Datums/Uhrzeit-Werts anzugeben (standardmäßig wird das Datum und die Uhrzeit in Web Intelligence in der UTC-Zeitzone angegeben). Die unterstützten Zeitzonen sind unten aufgeführt.	Für den 1. Januar 2015 um 12:00:00 Uhr HH' : 'mm' : 'ss [TIMEZONE : US / Eastern] gibt z 19:00:00 EST an.

ⓘ Hinweis

Vor Release 4.3 erstellte Dokumente, in denen das vorherige Format z verwendet wird, werden automatisch konvertiert, damit das in 4.3 angezeigte Ergebnis gleich bleibt. Das vorherige Format z wird als 0000 aus der Tabelle interpretiert.

Liste der Zeitzonen mit dem Token [TIMEZONE:t]:

Afrika/Abidjan	Amerika/Grand_Turk	Asien/Bagdad	Australien/Perth	Europa/Uljanowsk
Afrika/Accra	Amerika/Grenada	Asien/Bahrain	Australien/Queensland	Europa/Uzhgorod
Afrika/Addis_Abeba	Amerika/Guadeloupe	Asien/Baku	Australien/Süd	Europa/Vaduz
Afrika/Algier	Amerika/Guatemala	Asien/Bangkok	Australien/Sydney	Europa/Vatikan
Afrika/Asmara	Amerika/Guayaquil	Asien/Barnaul	Australien/Tasmanien	Europa/Wien
Afrika/Asmera	Amerika/Guyana	Asien/Beirut	Australien/Victoria	Europa/Vilnius
Afrika/Bamako	Amerika/Halifax	Asien/Bischkek	Australien/West	Europa/Wolgograd
Afrika/Bangui	Amerika/Havanna	Asien/Brunei	Australien/Yancowinna	Europa/Warschau
Afrika/Banjul	Amerika/Hermosillo	Asien/Kalkutta	Brasilien/Acre	Europa/Zagreb
Afrika/Bissau	Amerika/Indiana/Indianapolis	Asien/Chita	Brasilien/DeNoronha	Europa/Saporozhye
Afrika/Blantyre	Amerika/Indiana/Knox	Asien/Choibalsan	Brasilien/Ost	Europa/Zürich
Afrika/Brazzaville	Amerika/Indiana/Marengo	Asien/Chongqing	Brasilien/West	GB
Afrika/Bujumbura	Amerika/Indiana/Petersburg	Asien/Chungking	MEZ	GB-Irland
Afrika/Kairo	Amerika/Indiana/Tell_City	Asien/Colombo	CST6CDT	GMT

Afrika/Casablanca	Amerika/Indiana/Vevay	Asien/Dacca	Kanada/Atlantik	GMT+0
Afrika/Ceuta	Amerika/Indiana/Vincennes	Asien/Damaskus	Kanada/Central	GMT-0
Afrika/Conakry	Amerika/Indiana/Winamac	Asien/Dhaka	Kanada/Eastern	GMT0
Afrika/Dakar	Amerika/Indianapolis	Asien/Dili	Kanada/Mountain	Greenwich
Afrika/Dar_es_Salaam	Amerika/Inuvik	Asien/Dubai	Kanada/Neufundland	HST
Afrika/Dschibuti	Amerika/Iqaluit	Asien/Duschanbe	Kanada/Pazifik	Hongkong
Afrika/Douala	Amerika/Jamaika	Asien/Famagusta	Kanada/Saskatchewan	Island
Afrika/El_Aaiun	Amerika/Jujuy	Asien/Gaza	Kanada/Yukon	Indisch/Antananarivo
Afrika/Freetown	Amerika/Juneau	Asien/Harbin	Chile/Kontinental	Indisch/Chagos
Afrika/Gaborone	Amerika/Kentucky/Louisville	Asien/Hebron	Chile/Osterinsel	Indisch/Weihnachten
Afrika/Harare	Amerika/Kentucky/Monticello	Asien/Ho_Chi_Minh	Kuba	Indisch/Cocos
Afrika/Johannesburg	Amerika/Knox_IN	Asien/Hong_Kong	EET	Indisch/Komoren
Afrika/Juba	Amerika/Kralendijk	Asien/Hovd	EST	Indisch/Mahe
Afrika/Kampala	Amerika/La_Paz	Asien/Irkutsk	EST5EDT	Indisch/Malediven
Afrika/Khartum	Amerika/Lima	Asien/Istanbul	Ägypten	Indisch/Mauritius
Afrika/Kigali	Amerika/Los_Angeles	Asien/Jakarta	Irland	Indisch/Mayotte
Afrika/Kinshasa	Amerika/Louisville	Asien/Jayapura	Etc/GMT	Indisch/Réunion
Afrika/Lagos	Amerika/Lower_Princes	Asien/Jerusalem	Etc/GMT+0	Iran
Afrika/Libreville	Amerika/Maceio	Asien/Kabul	Etc/GMT+1	Israel
Afrika/Lome	Amerika/Managua	Asien/Kamtschatka	Etc/GMT+10	Jamaika
Afrika/Luanda	Amerika/Manaus	Asien/Karatschi	Etc/GMT+11	Japan
Afrika/Lubumbashi	Amerika/Marigot	Asien/Kashgar	Etc/GMT+12	Kwajalein
Afrika/Lusaka	Amerika/Martinique	Asien/Kathmandu	Etc/GMT+2	Libyen
Afrika/Malabo	Amerika/Matamoros	Asien/Katmandu	Etc/GMT+3	MET
Afrika/Maputo	Amerika/Mazatlan	Asien/Khandyga	Etc/GMT+4	MST
Afrika/Maseru	Amerika/Mendoza	Asien/Kolkata	Etc/GMT+5	MST7MDT
Afrika/Mbabane	Amerika/Menominee	Asien/Krasnojarsk	Etc/GMT+6	Mexiko/BajaNorte
Afrika/Mogadischu	Amerika/Merida	Asien/Kuala_Lumpur	Etc/GMT+7	Mexiko/BajaSur
Afrika/Monrovia	Amerika/Metlakatla	Asien/Kuching	Etc/GMT+8	Mexiko/Allgemein
Afrika/Nairobi	Amerika/Mexico_City	Asien/Kuwait	Etc/GMT+9	NZ
Afrika/Ndjamena	Amerika/Miquelon	Asien/Macau	Etc/GMT-0	NZ-CHAT

Afrika/Niamey	Amerika/Moncton	Asien/Macau	Etc/GMT-1	Navajo
Afrika/Nouakchott	Amerika/Monterrey	Asien/Magadan	Etc/GMT-10	VR China
Afrika/Ouagadougou	Amerika/Montevideo	Asien/Makassar	Etc/GMT-11	PST8PDT
Afrika/Porto-Novo	Amerika/Montreal	Asien/Manila	Etc/GMT-12	Pazifik/Apia
Afrika/Sao_Tome	Amerika/Montserrat	Asien/Maskat	Etc/GMT-13	Pazifik/Auckland
Afrika/Timbuktu	Amerika/Nassau	Asien/Nikosia	Etc/GMT-14	Pazifik/Bougainville
Afrika/Tripolis	Amerika/New_York	Asien/Novokuznetsk	Etc/GMT-2	Pazifik/Chatham
Afrika/Tunis	Amerika/Nipigon	Asien/Nowosibirsk	Etc/GMT-3	Pazifik/Chuuk
Afrika/Windhoek	Amerika/Nome	Asien/Omsk	Etc/GMT-4	Pazifik/Ostern
Amerika/Adak	Amerika/Noronha	Asien/Oral	Etc/GMT-5	Pazifik/Efate
Amerika/Anchorage	Amerika/North_Da- kota/Beulah	Asien/Phnom_Penh	Etc/GMT-6	Pazifik/Enderbury
Amerika/Anguilla	Amerika/North_Da- kota/Center	Asien/Pontianak	Etc/GMT-7	Pazifik/Fakaofu
Amerika/Antigua	Amerika/North_Da- kota/New_Salem	Asien/Pjöngjang	Etc/GMT-8	Pazifik/Fidschi
Amerika/Araguaina	Amerika/Ojinaga	Asien/Katar	Etc/GMT-9	Pazifik/Funafuti
Amerika/Argenti- nien/Buenos_Aires	Amerika/Panama	Asien/Qostanay	Etc/GMT0	Pazifik/Galapagos
Amerika/Argenti- nien/Catamarca	Amerika/Pangnirtung	Asien/Qyzylorda	Etc/Greenwich	Pazifik/Gambier
Amerika/Argenti- nien/ComodRivadavia	Amerika/Paramaribo	Asien/Rangoon30m	Etc/UCT	Pazifik/Guadalcanal
Amerika/Argenti- nien/Cordoba	Amerika/Phoenix	Asien/Riad	Etc/UTC	Pazifik/Guam
Amerika/Argenti- nien/Jujuy	Amerika/Port-au- Prince	Asien/Saigon	Etc/Universal	Pazifik/Honolulu
Amerika/Argenti- nien/La_Rioja	Amerika/ Port_of_Spain	Asien/Sachalin	Etc/Zulu	Pazifik/Johnston
Amerika/Argenti- nien/Mendoza	Amerika/Porto_Acre	Asien/Samarkand	Europa/Amsterdam	Pazifik/Kiritimati
Amerika/Argenti- nien/Rio_Gallegos	Amerika/Porto_Velho	Asien/Seoul	Europa/Andorra	Pazifik/Kosrae
Amerika/Argenti- nien/Salta	Amerika/Puerto_Rico	Asien/Shanghai	Europa/Astrachan	Pazifik/Kwajalein
Amerika/Argenti- nien/San_Juan	Amerika/Punta_Are- nas	Asien/Singapur	Europa/Athen	Pazifik/Majuro
Amerika/Argenti- nien/San_Luis	Amerika/Rainy_River	Asien/Srednekolymysk	Europa/Belfast	Pazifik/Marquesas
Amerika/Argenti- nien/Tucuman	Amerika/Rankin_Inlet	Asien/Taipeh	Europa/Belgrad	Pazifik/Midway

Amerika/Argentinien/Ushuaia	Amerika/Recife	Asien/Taschkent	Europa/Berlin	Pazifik/Nauru
Amerika/Aruba	Amerika/Regina	Asien/Tiflis	Europa/Bratislava	Pazifik/Niue
Amerika/Asuncion	Amerika/Resolute	Asien/Teheran	Europa/Brüssel	Pazifik/Norfolk
Amerika/Atikokan	Amerika/Rio_Branco	Asien/Tel_Aviv	Europa/Bukarest	Pazifik/Noumea
Amerika/Atka	Amerika/Rosario	Asien/Thimbu	Europa/Budapest	Pazifik/Pago_Pago
Amerika/Bahia	Amerika/Santa_Isabel	Asien/Thimphu	Europa/Busingen	Pazifik/Palau
Amerika/Bahia_Banderas	Amerika/Santarem	Asien/Tokio	Europa/Chisinau	Pazifik/Pitcairn
Amerika/Barbados	Amerika/Santiago	Asien/Tomsk	Europa/Kopenhagen	Pazifik/Pohnpei
Amerika/Belem	Amerika/Santo_Domingo	Asien/Ujung_Pandang	Europa/Dublin	Pazifik/Ponape
Amerika/Belize	Amerika/Sao_Paulo	Asien/Ulaanbaatar	Europa/Gibraltar	Pazifik/Port_Moresby
Amerika/Blanc-Sablon	Amerika/Scoresbysund	Asien/Ulan_Bator	Europa/Guernsey	Pazifik/Rarotonga
Amerika/Boa_Vista	Amerika/Shiprock	Asien/Urumqi	Europa/Helsinki	Pazifik/Saipan
Amerika/Bogota	Amerika/Sitka	Asien/Ust-Nera	Europa/Isle_of_Man	Pazifik/Samoa
Amerika/Boise	Amerika/St_Barthelemy	Asien/Vientiane	Europa/Istanbul	Pazifik/Tahiti
Amerika/Buenos_Aires	Amerika/St_Johns	Asien/Wladiwostok	Europa/Jersey	Pazifik/Tarawa
Amerika/Cambridge_Bay	Amerika/St_Kitts	Asien/Jakutsk	Europa/Kaliningrad	Pazifik/Tongatapu
Amerika/Campo_Grande	Amerika/St_Lucia	Asien/Yangon	Europa/Kiew	Pazifik/Truk
Amerika/Cancun	Amerika/St_Thomas	Asien/Jekaterinburg	Europa/Kirow	Pazifik/Wake
Amerika/Caracas	Amerika/St_Vincent	Asien/Eriwan	Europa/Lissabon	Pazifik/Wallis
Amerika/Catamarca	Amerika/Swift_Current	Atlantik/Azoren	Europa/Ljubljana	Pazifik/Yap
Amerika/Cayenne	Amerika/Tegucigalpa	Atlantik/Bermuda	Europa/London	Polen
Amerika/Cayman	Amerika/Thule	Atlantik/Kanarisch	Europa/Luxemburg	Portugal
Amerika/Chicago	Amerika/Thunder_Bay	Atlantik/Cape_Verde	Europa/Madrid	ROC
Amerika/Chihuahua	Amerika/Tijuana	Atlantik/Färöer	Europa/Malta	ROK
Amerika/Coral_Harbour	Amerika/Toronto	Atlantik/Färöer	Europa/Mariehamn	Singapur
Amerika/Cordoba	Amerika/Tortola	Atlantik/Jan_Mayen	Europa/Minsk	Türkei
Amerika/Costa_Rica	Amerika/Vancouver	Atlantik/Madeira	Europa/Monaco	UCT
Amerika/Creston	Amerika/Virgin	Atlantik/Reykjavik	Europa/Moskau	USA/Alaska
Amerika/Cuiaba	Amerika/Whitehorse	Atlantik/Südgeorgien	Europa/Nikosia	USA/Aleuten

Amerika/Curacao	Amerika/Winnipeg	Atlantik/St_Helena	Europa/Oslo	USA/Arizona
Amerika/Danmarks-havn	Amerika/Yakutat	Atlantik/Stanley	Europa/Paris	USA/Central
Amerika/Dawson	Amerika/Yellowknife	Australien/ACT	Europa/Podgorica	USA/Ost-Indiana
Amerika/Dawson_Creek	Antarktis/Macquarie	Australien/Adelaide	Europa/Prag	USA/Ost
Amerika/Denver	Antarktis/McMurdo	Australien/Brisbane	Europa/Riga	USA/Hawaii
Amerika/Detroit	Antarktis/Palmer	Australien/Broken_Hill	Europa/Rom	USA/Indiana-Starke
Amerika/Dominica	Antarktis/Südpol	Australien/Canberra	Europa/Samara	USA/Michigan
Amerika/Edmonton	Arktis/Longyearbyen	Australien/Currie	Europa/San_Marino	USA/Mountain
Amerika/Eirunepe	Asien/Aden	Australien/Darwin	Europa/Sarajevo	USA/Pazifik
Amerika/El_Salvador	Asien/Almaty	Australien/Eucla	Europa/Saratow	USA/Pazifik-Neu
Amerika/Ensenada	Asien/Amman	Australien/Hobart	Europa/Simferopol	USA/Samoa
Amerika/Fort_Nelson	Asien/Anadyr	Australien/LHI	Europa/Skopje	UTC
Amerika/Fort_Wayne	Asien/Aqtau	Australien/Lindeman	Europa/Sofia	Universal
Amerika/Fortaleza	Asien/Aqtobe	Australien/Lord_Howe	Europa/Stockholm	W-SU
Amerika/Glace_Bay	Asien/Aschgabat	Australien/Melbourne	Europa/Tallinn	WET
Amerika/Godthab	Asien/Ashkhabad	Australien/NSW	Europa/Tirane	Zulu
Amerika/Goose_Bay	Asien/Atyrau	Australien/Nord	Europa/Tiraspol	

1.6.1.2 Aggregatfunktionen

1.6.1.2.1 Aggregation

Beschreibung

Gibt die Standardaggregation einer bestimmten Elementmenge zurück

Funktionsgruppe

Aggregation

Syntax

```
Num Aggregation(Kennzahl[;Elementmenge])
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Kennzahl	Beliebige Kennzahl	Kennzahl	Ja
Elementmenge	Die zur Berechnung der Aggregation verwendete Elementmenge	Elementmenge	Nein

Hinweise

- Sie können mit der Funktion `Aggregation` Kontextoperatoren der erweiterten Syntax verwenden.
- Wenn Sie `Elementmenge` einschließen, gibt `Aggregation` den Aggregatwert der Kennzahl für alle Elemente der Elementmenge zurück.
- `Elementmenge` kann mehrere, durch Semikolon (;) getrennte Mengen enthalten.
- Die Liste der Elementmengen muss in {} eingeschlossen werden.
- Wenn im Ausdruck "Elementmenge" kein bestimmtes Element oder kein bestimmter Knoten angegeben ist, muss die referenzierte Hierarchie in der Tabelle vorhanden sein. Dann referenziert der Ausdruck "Elementmenge" das aktuelle Element in der Hierarchie in der Tabelle. Befindet sich die Hierarchie nicht in der Tabelle, gibt die Funktion die Meldung `#MEHRFACHWERT` zurück.
- Bei der delegierten Kennzahlaggregation wird `#ZUREGENERIEREN` zurückgegeben, wenn die erforderliche Aggregation in der Abfrage nicht verfügbar ist. Der Benutzer muss das Dokument regenerieren, um die neue Aggregationsebene abzurufen. Dies ist beispielsweise bei Verwendung der Filterleiste der Fall, wenn der Benutzer einen Wert auswählt, bevor er "Alle Werte" auswählt, und umgekehrt bei Auswahl von "Alle Werte" vor Auswahl eines Werts.

Beispiele

Wenn die Standardaggregation der Kennzahl [Umsatz] "Summe" und [Kalifornien] ein Element in der Hierarchie [Geografie] ist (Land > Bundesstaat > Stadt), gibt `Aggregation ([Umsatz]; {Nachfolger([Geografie]&[USA].[Kalifornien];1})` den Gesamtumsatz aller Städte in Kalifornien zurück.

Weitere Informationen

[Verweisen auf Elemente und Elementmengen in Hierarchien \[Seite 296\]](#)

1.6.1.2.2 Elementauswahl in Aggregatfunktionen

Beschreibung

Für bestimmte Aggregatfunktionen können Sie eine Elementauswahl festlegen, um den Aggregationskontext zu definieren, wenn der Block eine Hierarchie enthält.

Funktionsgruppe

Aggregation

Syntax

```
=AggregationFunction([my object];{memberselection})
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
AggregationFunction	Muss einer der folgenden Typen sein: <ul style="list-style-type: none">AggregateAverageCountMaxMinSum	Aggregation	Ja
my object	Dimension oder Kennzahl	Dimension oder Kennzahl	Ja
memberselection	Ein definiertes Element oder eine berechnete Menge von Elementen, die Set-Funktionen verwenden. Die Elementauswahl muss in geschweifte Klammern eingeschlossen sein. Jeder	Element oder berechnete Elementmenge, die Set-Funktionen verwenden.	Ja

Parameter	Beschreibung	Typ	Erforderlich
	<p>Bestandteil der Elementmenge wird durch ein Semikolon abgetrennt.</p> <pre>{[member one];[member two];CalculatedMemberSet() }</pre> <p>Während CalculatedMemberSet eine der folgenden Set-Funktionen verwendet:</p> <ul style="list-style-type: none"> • Ancestor • Descendant • Lag • Children • Parent • Siblings 		

Beschreibung

Diese Set-Funktionen verwenden Objekt, Ebene oder Element als Parameter. Wenn Sie nur das Objekt angeben und es sich bei dem Objekt um ein im Block vorhandenes hierarchisches Objekt handelt, wird das aktuelle Element verwendet. Über die folgende Syntax können Sie ein bestimmtes Element angeben:

```
[HierarchicalObject]&[RootMember].[ChildMember].[ChildMember]
```

Für Microsoft- und Essbase-UNX-Quellen können Sie eine Ebene auswählen:

```
[HierarchicalObject].[LevelName]
```

Beispiele

Die folgenden Beispiele stammen alle aus einer englischsprachigen Datenquelle.

❖ Beispiel

Im folgenden Beispiel möchten Sie den Unterschied zwischen den Internetumsätzen der Jahre 2002 und 2001 ermitteln:

```
=Sum([Internet Sales].[Internet Sales Amount];{[Calendar].[Date.Calendar]&[All Periods].[CY 2002]}) + Sum([Internet Sales].[Internet Sales Amount];{[Calendar].[Date.Calendar]&[All Periods].[CY 2001]})
```

Oder Sie wählen zwei Elemente in der Elementauswahl aus:

```
=Sum([Internet Sales].[Internet Sales Amount];{[Calendar].[Date.Calendar]&[All Periods].[CY 2002]};[Calendar].[Date.Calendar]&[All Periods].[CY 2001]})
```


Date.Calendar	Internet Sales Amount	{CY 2001;CY 2002}
[-] All Periods	29,358,677.22	9,796,717.18
[+] CY 2001	3,266,373.66	
[+] CY 2002	6,530,343.53	
[+] CY 2003	9,791,060.3	
[+] CY 2004	9,770,899.74	

♣ Beispiel

Im folgenden Beispiel haben Sie eine Produkthierarchie und möchten die Internetumsätze für alle Produkte, die mit Fahrrädern zu tun haben, wissen. Zwei davon befinden sich jedoch in einem anderen Zweig:

```
=Sum([Query 3].[Internet Sales].[Internet Sales Amount];{[Product Model Categories]&[All Products].[Accessories].[Bike Racks];[Product Model Categories]&[All Products].[Accessories].[Bike Stands];[Product Model Categories]&[All Products].[Bikes]})
```

Bikes Amount	28,397,095.65
[-] All Products	29,358,677.22
[-] Accessories	700,759.96
+ Bike Racks	39,360
+ Bike Stands	39,591
+ Bottles and Cages	56,798.19
+ Cleaners	7,218.6
+ Fenders	46,619.58
+ Helmets	225,335.6
+ Hydration Packs	40,307.67
+ Tires and Tubes	245,529.32
[-] Bikes	28,318,144.65
+ Mountain Bikes	9,952,759.56
+ Road Bikes	14,520,584.04
+ Touring Bikes	3,844,801.05
[-] Clothing	339,772.61
+ Caps	19,688.1
+ Gloves	35,020.7
+ Jerseys	172,950.68
+ Shorts	71,319.81
+ Socks	5,106.32
+ Vests	35,687

❖ Beispiel

Im folgenden Beispiel möchten Sie die Internet-Umsatzbeträge der nordamerikanischen Länder vergleichen, zuerst Kanada mit den USA, dann die nordamerikanischen Länder mit anderen Ländern weltweit:

Zuerst bilden Sie die Summe für die nordamerikanischen Länder; in diesem Beispiel sind Sie nur an Kanada und den USA interessiert:

```
=Sum([Query 2].[Internet Sales].[Internet Sales Amount];{[Customer Geography]&[All Customers].[Canada];[Customer Geography]&[All Customers].[United States]})
```

Customer Geography		Internet Sales Amount
[-] All Customers	11,367,634.37	29,358,677.22
[+] Australia	11,367,634.37	9,061,000.58
[+] Canada	11,367,634.37	1,977,844.86
[+] France	11,367,634.37	2,644,017.71
[+] Germany	11,367,634.37	2,894,312.34
[+] United Kingdom	11,367,634.37	3,391,712.21
[+] United States	11,367,634.37	9,389,789.51

Dann möchten Sie alle Länder mit Nordamerika vergleichen:

```
=([Query 2].[Internet Sales].[Internet Sales Amount] / Sum([Query 2].[Internet Sales].[Internet Sales Amount];{[Customer Geography]&[All Customers].[Canada];[Customer Geography]&[All Customers].[United States]}))
```

Customer Geography		Internet Sales Amount
[-] All Customers	258.27%	29,358,677.22
[+] Australia	79.71%	9,061,000.58
[+] Canada	17.40%	1,977,844.86
[+] France	23.26%	2,644,017.71
[+] Germany	25.46%	2,894,312.34
[+] United Kingdom	29.84%	3,391,712.21
[+] United States	82.60%	9,389,789.51

Wir können sehen, dass die Gesamtzahl der Kunden weltweit zweieinhalb Mal so hoch ist wie die Nordamerikas und die Kundenzahl Australiens 80 % der Kundenzahl Nordamerikas beträgt.

Weitere Informationen

[Aggregation \[Seite 45\]](#)

1.6.1.2.3 Durchschnitt

Beschreibung

Gibt den Durchschnittswert einer Kennzahl zurück.

Funktionsgruppe

Aggregation

Syntax

```
Num Durchschnitt(Kennzahl[;Elementmenge][;EinschließlichLeer])
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Kennzahl	Beliebige Kennzahl	Kennzahl	Ja
Elementmenge	Eine Menge von Elementen	Elementmenge	Nein
EinschließlichLeer	Fügt leere Zeilen in die Berechnung ein	Schlüsselwort	Nein (Leere Zeilen werden standardmäßig ausgeschlossen)

Hinweise

- Sie können mit der Funktion `Durchschnitt` Kontextoperatoren der erweiterten Syntax verwenden.
- Wenn Sie `Elementmenge` einschließen, gibt `Durchschnitt` den Durchschnittswert der Kennzahl für alle Elemente der Elementmenge zurück.
- `Elementmenge` kann mehrere, durch Semikolon (;) getrennte Mengen enthalten.
- Die Liste der Elementmengen muss in {} eingeschlossen werden.
- Wenn im Ausdruck "Elementmenge" kein bestimmtes Element oder kein bestimmter Knoten angegeben ist, muss die referenzierte Hierarchie in der Tabelle vorhanden sein. Dann referenziert der Ausdruck "Elementmenge" das aktuelle Element in der Hierarchie in der Tabelle. Befindet sich die Hierarchie nicht in der Tabelle, gibt die Funktion die Meldung `#MEHRFACHWERT` zurück.

- Bei der delegierten Kennzahlaggregation wird #ZUREGENERIEREN zurückgegeben, wenn die erforderliche Aggregation in der Abfrage nicht verfügbar ist. Der Benutzer muss das Dokument regenerieren, um die neue Aggregationsebene abzurufen. Dies ist beispielsweise bei Verwendung der Filterleiste der Fall, wenn der Benutzer einen Wert auswählt, bevor er "Alle Werte" auswählt, und umgekehrt bei Auswahl von "Alle Werte" vor Auswahl eines Werts.
- Eine für eine Gruppe bereitgestellte delegierte Kennzahl gibt #NICHT VERFÜGBAR zurück, da eine lokale Aggregation erforderlich ist (Aggregation des Kennzahlwerts der gruppierten Werte). Selbst wenn Sie die lokale Aggregation für eine delegierte Kennzahl für eine If-then-Else-Formel oder einen Gruppenwert erzwingen, wird die Meldung #MEHRFACHWERT zurückgegeben.

Beispiele

Wenn die Kennzahl [Umsatz] die Werte 41569, 30500, 40000 und 50138 umfasst, dann gibt `Durchschnitt([Umsatz])` 40552 zurück.

Wenn [Kalifornien] ein Element in der Hierarchie [Geografie] ist (Land > Bundesstaat > Stadt), gibt `Durchschnitt([Umsatz]; {[Geografie]&[USA].[Kalifornien].Untergeordnete Elemente})` den durchschnittlichen Umsatz aller Städte in Kalifornien zurück.

Weitere Informationen

[Verweisen auf Elemente und Elementmengen in Hierarchien \[Seite 296\]](#)

[EinschließlichLeer \(Operator\) \[Seite 280\]](#)

1.6.1.2.4 Anzahl

Beschreibung

Gibt die Anzahl der Werte in einer Wertemenge zurück.

Funktionsgruppe

Aggregation

Syntax

```
Ganzzahl Anzahl(Aggregierte Daten[;Ergebnismenge][;EinschließlichLeer][;Einmalig|Alle])
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Aggregierte Daten	Beliebige Dimension, Kennzahl, Hierarchie, Ebene oder Elementmenge	Dimension, Kennzahl, Hierarchie, Elementmenge	Ja
Elementmenge	Die zur Berechnung der Anzahl verwendete Elementmenge	Elementmenge	Nein
EinschließlichLeer	Fügt leere Werte in die Berechnung ein	Schlüsselwort	Nein
Einmalig Alle	Schließt entweder nur eindeutige Werte (standardmäßig bei Dimensionen) oder alle Werte (standardmäßig bei Kennzahlen) in die Berechnung ein	Schlüsselwort	Nein

Hinweise

- Sie können erweiterte Syntax-Kontextoperatoren mit der Funktion `Count` verwenden.
- Wenn Sie `IncludeEmpty` als zweites Argument angeben, werden von der Funktion bei der Berechnung leere Werte (Null) berücksichtigt.
- Wenn Sie den Parameter `DistinctAll` nicht angeben, lautet der Standardwert für Dimensionen `Distinct` und für Kennzahlen `All`.
- Wenn Sie `member_set` einschließen, beschränkt `Count` die Anzahl auf die Anzahl der Werte in `member_set`.
- `member_set` kann mehrere, durch Semikolon (;) getrennte Mengen enthalten.
- Die Liste der Elementmengen muss in `{ }` eingeschlossen werden.
- Wenn im Ausdruck `Elementmenge` kein bestimmtes Element oder kein bestimmter Knoten angegeben ist, muss die referenzierte Hierarchie in der Tabelle vorhanden sein, und der Ausdruck `Elementmenge` referenziert das aktuelle Element in der Hierarchie in der Tabelle. Befindet sich die Hierarchie nicht in der Tabelle, gibt die Funktion die Meldung `#MEHRFACHWERT` zurück.
- Bei der delegierten Kennzahlaggregation wird `#ZUREGENERIEREN` zurückgegeben, wenn die erforderliche Aggregation in der Abfrage nicht verfügbar ist. Der Benutzer muss das Dokument regenerieren, um die neue Aggregationsebene abzurufen. Dies ist beispielsweise bei Verwendung der Filterleiste der Fall, wenn

der Benutzer einen Wert auswählt, bevor er *Alle Werte* auswählt, und umgekehrt bei Auswahl von *Alle Werte* vor Auswahl eines Werts.

- Eine für eine Gruppe bereitgestellte delegierte Kennzahl gibt #NICHT VERFÜGBAR zurück, da eine lokale Aggregation erforderlich ist (Aggregation des Kennzahlwerts der gruppierten Werte). Selbst wenn Sie die lokale Aggregation für eine delegierte Kennzahl für eine If-then-Else-Formel oder einen Gruppenwert erzwingen, wird die Meldung #MEHRFACHWERT zurückgegeben.

Hinweis

In sehr spezifischen Workflows weicht die Aggregation zusammengeführter Objektwerte in XI 3.x und 4.x voneinander ab.

In XI 3.x ergibt die Aggregation der Werte von zusammengeführten Objektelementen im Kontext des zusammengeführten Objekts (mithilfe der ForEach()-Formel) eine gefilterte Liste der aggregierten Werte: diejenigen, die mit den zusammengeführten Werten übereinstimmen.

In 4.x ergibt derselbe Workflow die vollständige Liste der aggregierten Werte: es wird kein Filter angewendet.

Beispiele

`Anzahl ("Test ")` gibt 1 zurück.

`Anzahl ([Stadt];Einmalig)` gibt 5 zurück, wenn 5 verschiedene Städte in einer Städteliste enthalten sind, selbst wenn die Liste aufgrund von Duplikaten mehr als 5 Zeilen enthält.

`Anzahl ([Stadt];Alle)` gibt 10 zurück, wenn in einer Städteliste 10 Städte enthalten sind, auch wenn einige doppelt vorkommen.

`Anzahl ([Stadt];EinschließlichLeer)` gibt 6 zurück, wenn 5 Städte und eine Leerzeile in einer Städteliste enthalten sind.

`Count ([Product]; { [Geography]&[State] })` gibt die Gesamtzahl der Produkte der Ebene [Bundesstaat] aus der Hierarchie [Geografie] zurück.

Weitere Informationen

[EinschließlichLeer \(Operator\) \[Seite 280\]](#)

[Einmalig/Alle \(Operatoren\) \[Seite 279\]](#)

1.6.1.2.5 Erste

Beschreibung

Gibt den ersten Wert eines Datensatzes zurück.

Funktionsgruppe

Aggregation

Syntax

```
Eingabetyp Erste(Dimension|Kennzahl)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Dimension Kennzahl	Beliebige Dimension oder Kennzahl	Dimension oder Kennzahl	Ja

Hinweise

- Wenn `First` in einem Gruppenwechselfuß platziert ist, wird der erste Wert unter [Umsatz] im Gruppenwechsel zurückgegeben.
- Wenn `First` in einer Tabellenfußzeile platziert ist, wird der erste Wert in der Tabelle zurückgegeben.
- Wenn `First` im Tabellentext platziert ist, ist das Ergebnis unvorhersehbar und ist abhängig von der Reihenfolge im Datensatz in der Datenquelle.

Beispiele

Wenn `Erste([Umsatz])` in einem Tabellenfuß platziert ist, wird der erste Wert unter [Umsatz] in der Tabelle zurückgegeben.

1.6.1.2.6 Interpolation

Beschreibung

Berechnet leere Kennzahlwerte durch Interpolation

Funktionsgruppe

Aggregation

Syntax

```
num Interpolation(Kennzahl[;PunktzuPunkt|Linear] [;NichtInGruppenwechsel|  
(Dimensionen_zurücksetzen)][;Zeile|Spalte])
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Kennzahl	Beliebige Kennzahl	Kennzahl	Ja
PunktzuPunkt Linear	Interpolationsmethode: <ul style="list-style-type: none">PunktzuPunkt: Punkt-zu-Punkt-InterpolationLinear – lineare Regression mit geringster Quadratinterpolation	Schlüsselwort	Nein (PunktzuPunkt ist Standard)
NichtInGruppenwechsel Dimensionen_zurücksetzen	<ul style="list-style-type: none">NichtInGruppenwechsel – Verhindert, dass die Funktion die Berechnung bei Block- und Sektionswechseln zurücksetztDimensionen_zurücksetzen – Die Liste der zum Zurücksetzen der Interpolation verwendeten Dimensionen	Schlüsselwort Dimensionsliste	Nein
Zeile Spalte	Legt die Berechnungsrichtung fest	Schlüsselwort	(Zeile ist Standard)

Hinweise

- Interpolation ist besonders nützlich, wenn Sie ein Liniendiagramm aus einer Kennzahl mit fehlenden Werten erstellen. Mit Hilfe der Funktion können Sie sicherstellen, dass im Diagramm eine kontinuierliche Linie statt unterbrochener Linien und Punkten dargestellt wird.

- Lineare Regression mit der Interpolation nach kleinsten Quadraten berechnet fehlende Werte mit einer linearen Gleichung in Form von " $f(x) = ax + b$ ", die alle verfügbaren Werte der Kennzahl so genau wie möglich weitergibt.
- Mit der Punkt-zu-Punkt-Interpolation werden die Werte durch Berechnung einer linearen Gleichung in Form von " $f(x) = ax + b$ " berechnet, die durch die beiden neben dem fehlenden Wert liegenden Werte verläuft.
- Diese Sortierreihenfolge der Kennzahl wirkt sich auf die von der `Interpolation` zurückgegebenen Werte aus.
- Auf eine Formel, die eine `Interpolation` enthält, können Sie keine Sortierung und keine Rangfolge anwenden.
- Wenn sich in der Liste der Werte nur ein Wert befindet, verwendet die `Interpolation` diesen Wert, um alle fehlenden Werte bereitzustellen.
- Durch Filter, die auf eine interpolierte Kennzahl angewendet werden, können die von der `Interpolation` zurückgegebenen Werte geändert werden. Dies hängt von den Werten ab, auf die sich der Filter auswirkt.

Beispiele

`Interpolation([Wert])` stellt die folgenden fehlenden Werte mithilfe der Punkt-zu-Punkt-Interpolationsmethode bereit:

Tag	Wert	Interpolation([Wert])
Montag	12	12
Dienstag	14	14
Mittwoch		15
Donnerstag	16	16
Freitag		17
Samstag		18
Sonntag	19	19

Weitere Informationen

[Linear \(Operator\) \[Seite 281\]](#)

[PunktzuPunkt \(Operator\) \[Seite 282\]](#)

1.6.1.2.7 Letzte

Beschreibung

Gibt den letzten Wert eines Datensatzes zurück.

Funktionsgruppe

Aggregation

Syntax

```
Eingabetyp Letzte(Dimension|Kennzahl)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Dimension Kennzahl	Beliebige Dimension oder Kennzahl	Dimension oder Kennzahl	Ja

Hinweise

- Wenn `Last` in einer Tabellenfußzeile platziert ist, wird der letzte Wert im Gruppenwechsel zurückgegeben.
- Wenn `Last` in einer Tabellenfußzeile platziert ist, wird der letzte Wert in der Tabelle zurückgegeben.
- Wenn `Last` im Tabellentext platziert ist, ist das Ergebnis unvorhersehbar und ist abhängig von der Reihenfolge im Datensatz in der Datenquelle.
- Aus technischen Gründen kann `Last` einen Nullwert zurückgeben, wenn der Eingabeparameter ein zusammengeführtes Objekt ist.

Beispiele

Wenn `Last ([Revenue])` in einer Tabellenfußzeile platziert ist, wird der letzte Wert für [Revenue] (Umsatz) in der Tabelle zurückgegeben.

1.6.1.2.8 Max

Beschreibung

Der größte Wert einer Menge von Werten wird zurückgegeben

Funktionsgruppe

Aggregation

Syntax

```
Eingabetyp Max(Aggregierte Daten[;Elementmenge])
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Aggregierte Daten	Beliebige Dimension, Kennzahl, Hierarchie, Ebene oder Elementmenge	Dimension, Kennzahl, Hierarchie, Ebene oder Elementmenge	Ja
Elementmenge	Eine Menge von Elementen	Elementmenge	Nein

Hinweise

- Sie können erweiterte Syntax-Kontextoperatoren mit der Funktion `Max` verwenden.
- Wenn Sie `Elementmenge` einschließen, gibt `Max` den maximalen Wert der aggregierten Daten für alle Elemente der Elementmenge zurück.
- `Elementmenge` kann mehrere, durch Semikolon (;) getrennte Mengen enthalten.
- Die Liste der Elementmengen muss in {} eingeschlossen werden.
- Wenn im Ausdruck "Elementmenge" kein bestimmtes Element oder kein bestimmter Knoten angegeben ist, muss die referenzierte Hierarchie in der Tabelle vorhanden sein. Dann referenziert der Ausdruck "Elementmenge" das aktuelle Element in der Hierarchie in der Tabelle. Befindet sich die Hierarchie nicht in der Tabelle, gibt die Funktion die Meldung `#MEHRFACHWERT` zurück.
- Bei der delegierten Kennzahlaggregation wird `#ZUREGENERIEREN` zurückgegeben, wenn die erforderliche Aggregation in der Abfrage nicht verfügbar ist. Der Benutzer muss das Dokument regenerieren, um die neue Aggregationsebene abzurufen. Dies ist beispielsweise bei Verwendung der Filterleiste der Fall, wenn der Benutzer einen Wert auswählt, bevor er "Alle Werte" auswählt, und umgekehrt bei Auswahl von "Alle Werte" vor Auswahl eines Werts.
- Eine für eine Gruppe bereitgestellte delegierte Kennzahl gibt `#NICHT VERFÜGBAR` zurück, da eine lokale Aggregation erforderlich ist (Aggregation des Kennzahlwerts der gruppierten Werte). Selbst wenn Sie die lokale Aggregation für eine delegierte Kennzahl für eine If-then-Else-Formel oder einen Gruppenwert erzwingen, wird die Meldung `#MEHRFACHWERT` zurückgegeben.

Beispiele

Wenn die Kennzahl [Umsatz] die Werte 3000, 60034 und 901234 umfasst, dann gibt `Max([Umsatz])` den Wert 901234 zurück.

Wenn die Dimension [Stadt] die Werte "Freiburg" und "Berlin" umfasst, dann gibt `Max([Stadt])` den Wert "Berlin" zurück.

Wenn [USA] ein Element der Hierarchie [Geografie] (Land > Bundesstaat > Stadt) ist, gibt `Max([Umsatz]; {[Geografie].[USA].Untergeordnete Elemente})` den höchsten Umsatz für einen US-Bundesstaat zurück.

1.6.1.2.9 Median

Beschreibung

Gibt den Median (Mittelwert) einer Kennzahl zurück.

Funktionsgruppe

Aggregation

Syntax

```
Num Median(Kennzahl)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Kennzahl	Beliebige Kennzahl	Kennzahl	Ja

Hinweise

Wenn die Zahlenmenge aus einer geraden Anzahl von Werten besteht, ermittelt `Median` den Durchschnitt der beiden mittleren Werte.

Beispiele

`Median([Umsatz])` gibt 971.444 zurück, wenn `[Umsatz]` die Werte 835420, 971444 und 1479660 aufweist.

1.6.1.2.10 Min

Beschreibung

Der kleinste Wert in einer Menge von Werten wird zurückgegeben.

Funktionsgruppe

Aggregation

Syntax

```
Eingabetyp Min(Aggregierte Daten[;Elementmenge])
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Aggregierte Daten	Beliebige Dimension, Kennzahl, Hierarchie, Ebene oder Elementmenge	Dimension, Kennzahl, Hierarchie, Ebene oder Elementmenge	Ja
Elementmenge	Eine Menge von Elementen	Elementmenge	Nein

Hinweise

- Sie können mit der Funktion `Min` Kontextoperatoren der erweiterten Syntax verwenden.
- Wenn Sie `Elementmenge` einschließen, gibt `Min` den minimalen Wert der aggregierten Daten für alle Elemente der Elementmenge zurück.
- `Elementmenge` kann mehrere, durch Semikolon (;) getrennte Mengen enthalten.
- Die Liste der Elementmengen muss in `{ }` eingeschlossen werden.

- Wenn im Ausdruck "Elementmenge" kein bestimmtes Element oder kein bestimmter Knoten angegeben ist, muss die referenzierte Hierarchie in der Tabelle vorhanden sein. Dann referenziert der Ausdruck "Elementmenge" das aktuelle Element in der Hierarchie in der Tabelle. Befindet sich die Hierarchie nicht in der Tabelle, gibt die Funktion die Meldung #MEHRFACHWERT zurück.
- Bei der delegierten Kennzahlaggregation wird #ZUREGENERIEREN zurückgegeben, wenn die erforderliche Aggregation in der Abfrage nicht verfügbar ist. Der Benutzer muss das Dokument regenerieren, um die neue Aggregationsebene abzurufen. Dies ist beispielsweise bei Verwendung der Filterleiste der Fall, wenn der Benutzer einen Wert auswählt, bevor er "Alle Werte" auswählt, und umgekehrt bei Auswahl von "Alle Werte" vor Auswahl eines Werts.
- Eine für eine Gruppe bereitgestellte delegierte Kennzahl gibt #NICHT VERFÜGBAR zurück, da eine lokale Aggregation erforderlich ist (Aggregation des Kennzahlwerts der gruppierten Werte). Selbst wenn Sie die lokale Aggregation für eine delegierte Kennzahl für eine If-then-Else-Formel oder einen Gruppenwert erzwingen, wird die Meldung #MEHRFACHWERT zurückgegeben.

Beispiele

Wenn die Kennzahl [Umsatz] die Werte 3000, 60034 und 901234 umfasst, dann gibt `Min([Umsatz])` den Wert 3000 zurück.

Wenn die Dimension [Stadt] die Werte Freiburg und Berlin umfasst, dann gibt `Min(<Stadt>)` den Wert "Freiburg" zurück.

`Min([Umsatz];{[Geografie]&[USA].Untergeordnete Elemente})` gibt den niedrigsten Umsatz für einen US-Bundesstaat zurück, wenn [USA] ein Element in der Hierarchie [Geografie] mit den Ebenen [Land] > [Bundesstaat] > [Stadt] ist.

1.6.1.2.11 Modus

Beschreibung

Gibt den am häufigsten auftretenden Wert eines Datensatzes zurück.

Funktionsgruppe

Aggregation

Syntax

```
Eingabetyp Modus(Dimension|Kennzahl)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Dimension Kennzahl	Beliebige Dimension oder Kennzahl	Kennzahl	Ja

Hinweise

- Modus gibt Null zurück, wenn der Datensatz keinen Wert enthält, der häufiger als alle anderen Werte auftritt.

Beispiele

`Modus([Umsatz])` gibt 200 zurück, wenn [Umsatz] die Werte 100, 200, 300, 200 aufweist.

`Modus([Land])` gibt den am häufigsten auftretenden Wert für [Land] zurück.

1.6.1.2.12 Prozent

Beschreibung

Drückt einen Kennzahlwert als Prozent des Berechnungskontextes aus.

Funktionsgruppe

Aggregation

Syntax

```
Num Prozent(Kennzahl[;Gruppenwechsel][;Zeile|Spalte])
```


Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Kennzahl	Beliebige Kennzahl	Kennzahl	Ja
Gruppenwechsel	Berücksichtigt Gruppenwechsel in Tabellen	Schlüsselwort	Nein
Zeile Spalte	Legt die Berechnungsrichtung fest	Schlüsselwort	Nein

Beispiele

In der folgenden Tabelle wurde in der Spalte "Prozent" die Formel `Prozent ([Umsatz])` verwendet.

<i>Jahr</i>	<i>Umsatz</i>	<i>Prozent</i>
2001	1000	10
2002	5000	50
2003	4000	40
<i>Summe:</i>	<i>10000</i>	<i>100</i>

Standardmäßig entspricht der Berechnungskontext dem Gesamtbetrag der Kennzahl in der Tabelle. Sie können die Funktion so definieren, dass dabei ein in der Tabelle eingefügter Gruppenwechsel berücksichtigt wird. Verwenden Sie dazu das optionale Argument `Gruppenwechsel`. In diesem Fall entspricht der standardmäßige Berechnungskontext dem Tablespace.

In der folgenden Tabelle wurde in der Spalte "Prozent" die Formel `Prozent ([Umsatz];Gruppenwechsel)` verwendet.

<i>Jahr</i>	<i>Quartal</i>	<i>Umsatz</i>	<i>Prozent</i>
2001	Q1	1000	10
	Q2	2000	20
	Q3	5000	50
	Q4	2000	20
<i>2001</i>	<i>Summe:</i>	<i>10000</i>	<i>100</i>

<i>Jahr</i>	<i>Quartal</i>	<i>Umsatz</i>	<i>Prozent</i>
2002	Q1	2000	20
	Q2	2000	20
	Q3	5000	50
	Q4	1000	10

2002	Summe:	10000	100
------	--------	-------	-----

Sie können die Funktion `Prozent` in Spalten oder Zeilen verwenden; geben Sie dazu das optionale Argument `Zeile|Spalte` an. In der folgenden Kreuztabelle wurde beispielsweise in der Spalte "%" die Formel `Prozent([Umsatz];Zeile)` verwendet.

	Q1	%	Q2	%	Q3	%	Q4	%
2001	1000	10	2000	20	5000	50	2000	20
2002	2000	20	2000	20	5000	50	1000	10

1.6.1.2.13 Perzentil

Beschreibung

Gibt die das n-te Perzentil einer Kennzahl zurück.

Funktionsgruppe

Numerisch

Syntax

```
Num Perzentil(Kennzahl;Perzentil)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Kennzahl	Beliebige Kennzahl	Kennzahl	Ja
Perzentil	Ein als Dezimalzahl ausgedrückter Prozentsatz.	Numerisch	Ja

Hinweise

Das n-te Perzentil entspricht einer Zahl, die größer oder gleich n % der Zahlen in einer Menge ist. n % wird im Format 0.n ausgedrückt.

Beispiele

Wenn [Kennzahl] über die Zahlenmenge (10;20;30;40;50) verfügt, gibt `Perzentil([Kennzahl];0.3)` 22 zurück. Dieser Wert ist größer oder gleich 30 % der Zahlen in der Menge.

1.6.1.2.14 Produkt

Beschreibung

Multipliziert die Werte einer Kennzahl.

Funktionsgruppe

Aggregation

Syntax

```
Num Produkt(Kennzahl)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Kennzahl	Beliebige Kennzahl	Kennzahl	Ja

Beispiele

`Produkt([Kennzahl])` gibt 30 zurück, wenn [Kennzahl] die Werte 2, 3, 5 aufweist.

1.6.1.2.15 RunningAverage

Beschreibung

Gibt den laufenden Durchschnitt einer Kennzahl zurück

Funktionsgruppe

Aggregation

Syntax

```
num RunningAverage(measure[;Row|Col][;IncludeEmpty][;(reset_dims)])
```

Um den laufenden Durchschnitt bei jeder Sektion zurückzusetzen, wird die folgende Syntax empfohlen:

```
num RunningAverage(measure;section)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Kennzahl	Beliebige Kennzahl	Kennzahl	Ja
Row Col	Legt die Berechnungsrichtung fest	Schlüsselwort	Nein
IncludeEmpty	Fügt leere Werte in die Berechnung ein	Schlüsselwort	Nein
reset_dims	Setzt die Berechnung für die angegebenen Dimensionen zurück	Dimensionsliste	Nein
Sektion	Dimension, für die die Sektion festgelegt ist	Schlüsselwort	Ja, wenn eine Sektion zurückgesetzt wird

Hinweise

- Sie können erweiterte Syntax-Kontextoperatoren mit `RunningAverage` verwenden.
- Mithilfe der Operatoren `Row` und `Col` können Sie die Berechnungsrichtung festlegen.

- Wenn Sie eine Sortierung auf die Kennzahl anwenden, auf die `RunningAverage`, verweist, wird der laufende Durchschnitt berechnet, nachdem die Kennzahl sortiert wurde.
- Dimensionen müssen selbst dann immer in Klammern eingeschlossen werden, wenn die Liste der zurückgesetzten Dimensionen nur eine Dimension enthält.
- Beim Festlegen einer Reihe zurückgesetzter Dimensionen müssen Sie diese durch Semikolons trennen.
- Durch `RunningAverage` wird der Durchschnitt nach einem Gruppenwechsel für einen Block oder nach einer neuen Sektion nicht automatisch zurückgesetzt.

Beispiele

`RunningAverage([Umsatz])` gibt die folgenden Ergebnisse zurück:

<i>Land</i>	<i>Erholungsort</i>	Umsatz	Laufender Durchschnitt
USA	Hawaiian Club	1.479.660	1.479.660
USA	Bahamas Beach	971.444	1.225.552
Frankreich	Französische Riviera	835.420	1.095.508

`RunningAverage([Umsatz]);([Land])` gibt die folgenden Ergebnisse zurück:

<i>Land</i>	<i>Erholungsort</i>	Umsatz	Laufender Durchschnitt
USA	Hawaiian Club	1.479.660	1.479.660
USA	Bahamas Beach	971.444	1.225.552
Frankreich	Französische Riviera	835.420	835.420

In einem Beispiel, in dem Sie die Funktion `RunningAverage` in einer Sektion für [Quartal] mit Hilfe der Formel `RunningAverage([Sales revenue];([Quarter]))`, verwenden, erhalten Sie die folgenden Ergebnisse:

Q1

Stadt	Umsatz	Laufender Durchschnitt
New York	1.987.114,70 \$	1.987.114,70 \$
Houston	1.544.627,80\$	1.765.871,25\$
Los Angeles	1.129.177,60\$	1.553.640,03\$

Q2

Stadt	Umsatz	Laufender Durchschnitt
New York	2.028.090,70\$	2.028.090,70\$
Houston	1.380.838,20\$	1.704.464,45\$
Los Angeles	980.405,30\$	1.463.111,40\$

Weitere Informationen

[EinschließlichLeer \(Operator\) \[Seite 280\]](#)

[Zeile/Spalte \(Operatoren\) \[Seite 283\]](#)

1.6.1.2.16 LaufendeAnzahl

Beschreibung

Gibt die laufende Anzahl einer Menge von Zahlen zurück.

Funktionsgruppe

Aggregation

Syntax

```
Num LaufendeAnzahl (Dimension | Kennzahl [ ; Zeile | Spalte ] [ ; EinschließlichLeer ] [ ;  
(Dimensionen_zurücksetzen) ] )
```

Um die Funktion LaufendeAnzahl bei jeder Sektion zurückzusetzen, wird die folgende Syntax empfohlen:

```
num RunningCount (dimension | measure ; section)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Dimension Kennzahl	Beliebige Dimension oder Kennzahl	Dimension oder Kennzahl	Ja
Zeile Spalte	Legt die Berechnungsrichtung fest	Schlüsselwort	Nein
EinschließlichLeer	Fügt leere Werte in die Berechnung ein	Schlüsselwort	Nein
Dimensionen_zurücksetzen	Setzt die Berechnung für die angegebenen Dimensionen zurück	Dimensionsliste	Nein

Parameter	Beschreibung	Typ	Erforderlich
Sektion	Dimension, für die die Sektion festgelegt ist	Schlüsselwort	Ja, wenn eine Sektion zurückgesetzt wird

Hinweise

- Sie können mit der Funktion `LaufendeAnzahl` Kontextoperatoren der erweiterten Syntax verwenden.
- Mithilfe der Operatoren `Zeile` und `Spalte` können Sie die Berechnungsrichtung festlegen.
- Wenn Sie eine Sortierung auf die Kennzahl anwenden, auf die `LaufendeAnzahl` verweist, wird die laufende Anzahl berechnet, nachdem die Kennzahl sortiert wurde.
- Dimensionen müssen selbst dann immer in Klammern eingeschlossen werden, wenn die Liste der zurückgesetzten Dimensionen nur eine Dimension enthält.
- Beim Festlegen einer Reihe zurückgesetzter Dimensionen müssen Sie diese durch Semikolons trennen.
- Durch `LaufendeAnzahl` wird die Anzahl nach einem Gruppenwechsel für einen Block oder nach einer neuen Sektion nicht automatisch zurückgesetzt.

Beispiele

`LaufendeAnzahl ([Umsatz])` gibt diese Ergebnisse in der folgenden Tabelle zurück:

Land	Erholungsort	Umsatz	Laufende Anzahl
USA	Hawaiian Club	1,479,660	1
USA	Bahamas Beach	971,444	2
Frankreich	Französische Riviera	835,420	3

`LaufendeAnzahl ([Umsatz] ; ([Land]))` gibt diese Ergebnisse in der folgenden Tabelle zurück:

Land	Erholungsort	Umsatz	Laufende Anzahl
USA	Hawaiian Club	1,479,660	1
USA	Bahamas Beach	971,444	2
Frankreich	Französische Riviera	835,420	1

In einem Beispiel, in dem Sie die Funktion `LaufendeAnzahl` in einer Sektion für [Woche] mit der Formel `LaufendeAnzahl ([Linien] ; ([Woche]))` und einem Eingabesteuerelement für [Umsatz] verwenden, womit die Liste auf Umsätze von über 30.000 \$ eingeschränkt wird, werden die folgenden Ergebnisse zurückgegeben:

Woche 1		
Linien	Umsatz	Laufende Anzahl
Sweat-T-Shirts	\$186,191	1

Hemdbluse	\$139,082	2
Kleider	\$70,931	3
<i>Woche 2</i>		
<i>Linien</i>	<i>Umsatz</i>	<i>Laufende Anzahl</i>
Accessoires	\$344,617	1
Sweat-T-Shirts	\$196,976	2
Hemdbluse	\$105,597	3
Kleider	\$76,290	4
Pullover	\$68,364	5

Beachten Sie, dass es in Woche 1 drei Linien mit einem Umsatz gibt, die den Betrag von 30.000 \$ überschreiten und in Woche 2 fünf Produktlinien, die den Betrag von 30.000 \$ überschreiten.

Weitere Informationen

[EinschließlichLeer \(Operator\) \[Seite 280\]](#)

[Zeile/Spalte \(Operatoren\) \[Seite 283\]](#)

[EinschließlichLeer \(Operator\) \[Seite 280\]](#)

[EinschließlichLeer \(Operator\) \[Seite 280\]](#)

1.6.1.2.17 LaufendesMaximum

Beschreibung

Gibt den laufenden Höchstwert einer Dimension oder Kennzahl zurück.

Funktionsgruppe

Aggregation

Syntax

```
Eingabetyp LaufendesMaximum(Dimension|Kennzahl[;Zeile|Spalte][;
(Dimensionen_zurücksetzen)])
```


Um die Funktion `LaufendesMaximum` bei jeder Sektion zurückzusetzen, wird die folgende Syntax empfohlen:

```
num RunningMax(measure;section)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Dimension Kennzahl	Beliebige Dimension oder Kennzahl	Dimension oder Kennzahl	Ja
Zeile Spalte	Legt die Berechnungsrichtung fest	Schlüsselwort	Nein
Dimensionen_zurücksetzen	Setzt die Berechnung für die angegebenen Dimensionen zurück	Dimensionsliste	Nein
Sektion	Dimension, für die die Sektion festgelegt ist	Schlüsselwort	Ja, wenn eine Sektion zurückgesetzt wird

Hinweise

- Sie können mit der Funktion `LaufendesMaximum` Kontextoperatoren der erweiterten Syntax verwenden.
- Mithilfe der Operatoren `Zeile` und `Spalte` können Sie die Berechnungsrichtung festlegen.
- Wenn Sie eine Sortierung auf die Kennzahl anwenden, auf die `LaufendesMaximum` verweist, wird das laufende Maximum berechnet, nachdem die Kennzahl sortiert wurde.
- Dimensionen müssen selbst dann immer in Klammern eingeschlossen werden, wenn die Liste der zurückgesetzten Dimensionen nur eine Dimension enthält.
- Beim Festlegen einer Reihe zurückgesetzter Dimensionen müssen Sie diese durch Semikolons trennen.
- Durch `LaufendesMaximum` wird der Höchstwert nach einem Gruppenwechsel für einen Block oder nach einer neuen Sektion nicht automatisch zurückgesetzt.

Beispiele

`LaufendesMaximum([Umsatz])` gibt diese Ergebnisse in der folgenden Tabelle zurück:

<i>Land</i>	<i>Erholungsort</i>	<i>Umsatz</i>	<i>Laufendes Maximum</i>
Frankreich	Französische Riviera	835,420	835,420
USA	Bahamas Beach	971,444	971,444
USA	Hawaiian Club	1,479,660	1,479,660

In einem Beispiel, in dem Sie die Funktion `LaufendesMaximum` in einer Sektion für [Stadt] mit der Formel `LaufendesMaximum([Umsatz];([Stadt]))` verwenden, erhalten Sie die folgenden Ergebnisse:

Austin

Quartal	Umsatz	Laufendes Maximum
Q1	\$775,482.70	\$775,482.70
Q2	\$667,850.30	\$775,482.70
Q3	\$581,470.40	\$775,482.70
Q4	\$674,869.80	\$775,482.70

Boston

Quartal	Umsatz	Laufendes Maximum
Q1	\$312,896.40	\$312,896.40
Q2	\$291,431.00	\$312,896.40
Q3	\$249,529.00	\$312,896.40
Q4	\$429,850.20	\$429,850.20

Weitere Informationen

[EinschließlichLeer \(Operator\) \[Seite 280\]](#)

[Zeile/Spalte \(Operatoren\) \[Seite 283\]](#)

1.6.1.2.18 LaufendesMinimum

Beschreibung

Gibt den laufenden Mindestwert einer Dimension oder Kennzahl zurück.

Funktionsgruppe

Aggregation

Syntax

```
Eingabe_Typ LaufendesMinimum(Dimension|Kennzahl;[Zeile|Spalte];  
[(Dimensionen_zurücksetzen)])
```

Um die Funktion `LaufendesMinimum` bei jeder Sektion zurückzusetzen, wird die folgende Syntax empfohlen:

```
num RunningMin(measure;section)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Dimension Detail Kennzahl	Beliebige Dimension oder Kennzahl	Dimension oder Kennzahl	Ja
Zeile Spalte	Legt die Berechnungsrichtung fest	Schlüsselwort	Nein
Dimensionen_zurücksetzen	Setzt die Berechnung für die angegebenen Dimensionen zurück	Dimensionsliste	Nein
Sektion	Dimension, für die die Sektion festgelegt ist	Schlüsselwort	Ja, wenn eine Sektion zurückgesetzt wird

Hinweise

- Sie können mit der Funktion `LaufendesMinimum` Kontextoperatoren der erweiterten Syntax verwenden.
- Mithilfe der Operatoren `Zeile` und `Spalte` können Sie die Berechnungsrichtung festlegen.
- Wenn Sie eine Sortierung auf die Kennzahl anwenden, auf die `LaufendesMinimum` verweist, wird das laufende Minimum berechnet, nachdem die Kennzahl sortiert wurde.
- Dimensionen müssen selbst dann immer in Klammern eingeschlossen werden, wenn die Liste der zurückgesetzten Dimensionen nur eine Dimension enthält.
- Beim Festlegen einer Reihe zurückgesetzter Dimensionen müssen Sie diese durch Semikolons trennen.
- Durch `LaufendesMinimum` wird das Minimum nach einem Gruppenwechsel für einen Block oder nach einer neuen Sektion nicht automatisch zurückgesetzt.

Beispiele

`LaufendesMinimum([Umsatz])` gibt diese Ergebnisse in der folgenden Tabelle zurück:

<i>Land</i>	<i>Erholungsort</i>	<i>Umsatz</i>	<i>Laufendes Minimum</i>
Frankreich	Französische Riviera	835,420	835,420
USA	Bahamas Beach	971,444	835,420
USA	Hawaiian Club	1,479,660	835,420

In einem Beispiel, in dem Sie die Funktion `LaufendesMinimum` in einer Sektion für [Stadt] mit der Formel `LaufendesMinimum([Umsatz];([Stadt]))` verwenden, erhalten Sie die folgenden Ergebnisse:

Austin

Quartal	Umsatz	Laufendes Minimum
Q1	\$775,482.70	\$775,482.70
Q2	\$667,850.30	\$667,850.30
Q3	\$581,470.40	\$581,470.40
Q4	\$674,869.80	\$581,470.40

Boston

Quartal	Umsatz	Laufendes Minimum
Q1	\$312,896.40	\$312,896.40
Q2	\$291,431.00	\$291,431.00
Q3	\$249,529.00	\$249,529.00
Q4	\$429,850.20	\$249,529.00

Weitere Informationen

[EinschließlichLeer \(Operator\) \[Seite 280\]](#)

[Zeile/Spalte \(Operatoren\) \[Seite 283\]](#)

1.6.1.2.19 LaufendesProdukt

Beschreibung

Gibt das laufende Produkt einer Kennzahl zurück.

Funktionsgruppe

Aggregation

Syntax

```
Num LaufendesProdukt(Kennzahl[;Zeile|Spalte][;(Dimensionen_zurücksetzen)])
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Kennzahl	Beliebige Kennzahl	Kennzahl	Ja
Zeile Spalte	Legt die Berechnungsrichtung fest	Schlüsselwort	Nein
Dimensionen_zurücksetzen	Setzt die Berechnung für die angegebenen Dimensionen zurück	Dimensionsliste	Nein

Hinweise

- Sie können mit der Funktion `LaufendesProdukt` Kontextoperatoren der erweiterten Syntax verwenden.
- Mithilfe der Operatoren `Zeile` und `Spalte` können Sie die Berechnungsrichtung festlegen.
- Wenn Sie eine Sortierung auf die Kennzahl anwenden, auf die `LaufendesProdukt` verweist, wird das laufende Produkt berechnet, nachdem die Kennzahl sortiert wurde.
- Dimensionen müssen selbst dann immer in Klammern eingeschlossen werden, wenn die Liste der zurückgesetzten Dimensionen nur eine Dimension enthält.
- Beim Festlegen einer Reihe zurückgesetzter Dimensionen müssen Sie diese durch Semikolons trennen.
- Durch `LaufendesProdukt` wird das Produkt nach einem Gruppenwechsel für einen Block oder nach einer neuen Sektion nicht automatisch zurückgesetzt.

Beispiele

`LaufendesProdukt ([Kundenanzahl])` gibt die Ergebnisse in der folgenden Tabelle zurück:

<i>Herkunftsland</i>	<i>Stadt</i>	Kundenanzahl	Laufendes Produkt
Japan	Kobe	6	6
Japan	Osaka	4	24
USA	Chicago	241	5.784

`LaufendesProdukt ([Kundenanzahl] ; ([Herkunftsland]))` gibt die Ergebnisse in der folgenden Tabelle zurück:

<i>Herkunftsland</i>	<i>Stadt</i>	Kundenanzahl	Laufendes Produkt
Japan	Kobe	6	6
Japan	Osaka	4	24
USA	Chicago	241	5784

Weitere Informationen

[EinschließlichLeer \(Operator\) \[Seite 280\]](#)

[Zeile/Spalte \(Operatoren\) \[Seite 283\]](#)

1.6.1.2.20 LaufendeSumme

Beschreibung

Gibt die laufende Summe einer Kennzahl zurück.

Funktionsgruppe

Aggregation

Syntax

```
Num LaufendeSumme(Kennzahl[;Zeile|Spalte][;(Dimensionen_zurücksetzen)])
```

Um die Funktion LaufendeSumme bei jeder Sektion zurückzusetzen, wird die folgende Syntax empfohlen:

```
num RunningSum(measure;section)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Kennzahl	Beliebige Kennzahl	Kennzahl	Ja
Zeile Spalte	Legt die Berechnungsrichtung fest	Schlüsselwort	Nein
Dimensionen_zurücksetzen	Setzt die Berechnung für die angegebenen Dimensionen zurück	Dimensionsliste	Nein
Sektion	Dimension, für die die Sektion festgelegt ist	Schlüsselwort	Ja, wenn eine Sektion zurückgesetzt wird

Hinweise

- Mit der Funktion `LaufendeSumme` können Sie Kontextoperatoren der erweiterten Syntax verwenden.
- Mithilfe der Operatoren `Zeile` und `Spalte` können Sie die Berechnungsrichtung festlegen.
- Wenn Sie eine Sortierung auf die Kennzahl anwenden, auf die die Funktion `LaufendeSumme` verweist, wird die laufende Summe berechnet, nachdem die Kennzahl sortiert wurde.
- Dimensionen müssen selbst dann immer in Klammern eingeschlossen werden, wenn die Liste der zurückgesetzten Dimensionen nur eine Dimension enthält.
- Beim Festlegen einer Reihe zurückgesetzter Dimensionen müssen Sie diese durch Semikolons trennen.
- Durch `LaufendeSumme` wird die Summe nach einem Gruppenwechsel für einen Block oder nach einer neuen Sektion nicht automatisch zurückgesetzt.

Beispiel

`LaufendeSumme([Umsatz])` gibt diese Ergebnisse in der folgenden Tabelle zurück:

<i>Land</i>	<i>Erholungsort</i>	<i>Umsatz</i>	<i>Laufende Summe</i>
Frankreich	Französische Riviera	835,420	835,420
USA	Bahamas Beach	971,444	1,806,864
USA	Hawaiian Club	1,479,660	3,286,524

`LaufendeSumme([Umsatz];([Land]))` gibt diese Ergebnisse in der folgenden Tabelle zurück:

<i>Land</i>	<i>Erholungsort</i>	<i>Umsatz</i>	<i>Laufende Summe</i>
Frankreich	Französische Riviera	835,420	835,420
USA	Bahamas Beach	971,444	971,444
USA	Hawaiian Club	1,479,660	2,451,104

In einem Beispiel, in dem Sie die Funktion `LaufendeSumme` in einer Sektion für [Quartal] mit der Formel `LaufendeSumme([Umsatz];([Quartal]))` verwenden, erhalten Sie die folgenden Ergebnisse:

Q1

<i>Stadt</i>	<i>Umsatz</i>	<i>Laufende Summe</i>
New York	\$1,987,114.70	\$1,987,114.70
Houston	\$1,544,627.80	\$3,531,742.50
Los Angeles	\$1,129,177.60	\$4,660,920.10

Q2

<i>Stadt</i>	<i>Umsatz</i>	<i>Laufende Summe</i>
New York	\$2,028,090.70	\$2,028,090.70

Houston	\$1,380,838.20	\$3,408,928.90
Los Angeles	\$980,405.30	\$4,389,334.20

Weitere Informationen

[EinschließlichLeer \(Operator\) \[Seite 280\]](#)

[Zeile/Spalte \(Operatoren\) \[Seite 283\]](#)

1.6.1.2.21 StdAbw

Beschreibung

Gibt die Standardabweichung einer Kennzahl zurück.

Funktionsgruppe

Aggregation

Syntax

```
Num StdAbw(Kennzahl)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Kennzahl	Beliebige Kennzahl	Kennzahl	Ja

Hinweise

Die Standardabweichung ist eine Kennzahl für die statistische Streuung in einer Menge von Zahlen. Sie wird wie folgt berechnet:

- Ermitteln des Durchschnitts der Zahlenmenge;
- Subtrahieren des Durchschnitts von jeder Zahl in der Menge und Quadrieren der Differenz;
- Addieren aller quadrierten Differenzen;
- Dividieren dieser Summe durch (`<Anzahl der Zahlen in der Menge> - 1`);
- Berechnen der Quadratwurzel des Ergebnisses.

Beispiele

Wenn `Kennzahl` die Wertemenge (2, 4, 6, 8) aufweist, gibt `StdAbw([Kennzahl])` 2,58 zurück.

Weitere Informationen

[Var \[Seite 84\]](#)

1.6.1.2.22 StAbwP

Beschreibung

Gibt die Populationsstandardabweichung einer Kennzahl zurück.

Funktionsgruppe

Aggregation

Syntax

```
Num StAbwP( Kennzahl )
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Kennzahl	Beliebige Kennzahl	Kennzahl	Ja

Hinweise

Die Populationsstandardabweichung ist eine Kennzahl für die statistische Streuung in einer Menge von Zahlen. Sie wird wie folgt berechnet:

- Ermitteln des Durchschnitts der Zahlenmenge;
- Subtrahieren des Durchschnitts von jeder Zahl in der Menge und Quadrieren der Differenz;
- Addieren aller quadrierten Differenzen;
- Dividieren dieser Summe durch (`<Anzahl der Zahlen in der Menge>`);
- Berechnen der Quadratwurzel des Ergebnisses.

Sie können erweiterte Syntax-Kontextoperatoren mit der Funktion `StAbwP` verwenden.

Beispiele

Wenn `Kennzahl` die Wertemenge (2, 4, 6, 8) aufweist, gibt `StAbwP ([Kennzahl])` 2,24 zurück.

1.6.1.2.23 Summe

Beschreibung

Gibt die Summe einer Kennzahl zurück.

Funktionsgruppe

Aggregation

Syntax

```
Num Summe ( Kennzahl [ ; Elementmenge ] )
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Kennzahl	Beliebige Kennzahl	Kennzahl	Ja
Elementmenge	Eine Menge von Elementen	Elementmenge	Nein

Hinweise

- Sie können erweiterte Syntax-Kontextoperatoren mit der Funktion `Summe` verwenden.
- Wenn Sie `Elementmenge` einschließen, gibt `Summe` die Summe der Kennzahl für alle Elemente der Elementmenge zurück.
- `Elementmenge` kann mehrere, durch Semikolon (;) getrennte Mengen enthalten.
- Die Liste der Elementmengen muss in {} eingeschlossen werden.
- Wenn im Ausdruck "Elementmenge" kein bestimmtes Element oder kein bestimmter Knoten angegeben ist, muss die referenzierte Hierarchie in der Tabelle vorhanden sein. Dann referenziert der Ausdruck "Elementmenge" das aktuelle Element in der Hierarchie in der Tabelle. Befindet sich die Hierarchie nicht in der Tabelle, gibt die Funktion die Meldung `#MEHRFACHWERT` zurück.
- Bei der delegierten Kennzahlaggregation wird `#ZUREGENERIEREN` zurückgegeben, wenn die erforderliche Aggregation in der Abfrage nicht verfügbar ist. Der Benutzer muss das Dokument regenerieren, um die neue Aggregationsebene abzurufen. Dies ist beispielsweise bei Verwendung der Filterleiste der Fall, wenn der Benutzer einen Wert auswählt, bevor er "Alle Werte" auswählt, und umgekehrt bei Auswahl von "Alle Werte" vor Auswahl eines Werts.
- Bei der Migration von XIR2 nach XIR3 müssen Aggregationsfunktionen, die IN- und WHERE-Klauseln in XI2-Abfragen enthalten, durch Verwendung von Klammern wie folgt definitiv in die Summenfunktion eingeschlossen werden:
Die Formel in XIR2: `=Summe([Kennzahl] In ([Dim 1];[Dim 2])) In ([Dim 1]) Wo ([Dim 3]="Konstante")`
Geänderte Deklaration ab XI3: `=Summe(([Kennzahl] In ([Dim 1];[Dim 2])) In ([Dim 1]) Wo ([Dim 3]="Konstante"))`
- Eine für eine Gruppe bereitgestellte delegierte Kennzahl gibt `#NICHT VERFÜGBAR` zurück, da eine lokale Aggregation erforderlich ist (Aggregation des Kennzahlwerts der gruppierten Werte). Selbst wenn Sie die lokale Aggregation für eine delegierte Kennzahl für eine If-then-Else-Formel oder einen Gruppenwert erzwingen, wird die Meldung `#MEHRFACHWERT` zurückgegeben.

Beispiele

Wenn die Kennzahl Umsatz die Werte 2000, 3000, 4000 und 1000 umfasst, dann gibt `Summe([Umsatz])` 10000 zurück.

Wenn [Kalifornien] ein Element der Hierarchie [Geografie] (Land > Bundesstaat > Stadt) ist, gibt `Summe([Umsatz]; {Nachfolger([Geografie]&[USA].[Kalifornien];1)})` den Gesamtumsatz aller Städte in Kalifornien zurück.

1.6.1.2.24 Var

Beschreibung

Gibt die Varianz einer Kennzahl zurück.

Funktionsgruppe

Aggregation

Syntax

```
Num Var ( Kennzahl )
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Kennzahl	Beliebige Kennzahl	Kennzahl	Ja

Hinweise

Die Varianz ist eine Kennzahl für die statistische Streuung in einer Zahlenmenge. Sie wird wie folgt berechnet:

- Ermitteln des Durchschnitts der Zahlenmenge;
- Subtrahieren des Durchschnitts von jeder Zahl in der Menge und Quadrieren der Differenz;
- Addieren aller quadrierten Differenzen;
- Dividieren dieser Summe durch (**<Anzahl der Zahlen in der Menge> - 1**);

Die Varianz ist das Quadrat der Standardabweichung.

Sie können erweiterte Syntax-Kontextoperatoren mit der Funktion `var` verwenden.

Beispiele

Wenn `Kennzahl` die Wertemenge (2, 4, 6, 8) aufweist, gibt `Var ([Kennzahl])` 6,67 zurück.

Weitere Informationen

[StdAbw \[Seite 80\]](#)

1.6.1.2.25 VarianzP (Funktion)

Beschreibung

Gibt die Populationsvarianz einer Kennzahl zurück.

Funktionsgruppe

Aggregation

Syntax

```
Num VarianzP(Kennzahl)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Kennzahl	Beliebige Kennzahl	Kennzahl	Ja

Hinweise

Die Populationsvarianz ist eine Kennzahl für die statistische Streuung in einer Zahlenmenge. Sie wird wie folgt berechnet:

- Ermitteln des Durchschnitts der Zahlenmenge;
- Subtrahieren des Durchschnitts von jeder Zahl in der Menge und Quadrieren der Differenz;
- Addieren aller quadrierten Differenzen;
- Dividieren dieser Summe durch ([<Anzahl der Zahlen in der Menge>](#)).

Die Populationsvarianz entspricht dem Quadrat der Populationsstandardabweichung.

Sie können erweiterte Syntax-Kontextoperatoren mit der Funktion `VarianzP` verwenden.

Beispiele

Wenn Kennzahl die Wertemenge (2, 4, 6, 8) aufweist, gibt `VarianzP([Kennzahl])` 5 zurück.

Weitere Informationen

[StAbwP \[Seite 81\]](#)

1.6.1.3 Zeichenfunktionen

1.6.1.3.1 Asc

Beschreibung

Gibt den ASCII-Wert eines Zeichens zurück.

Funktionsgruppe

Zeichen

Syntax

```
Ganzz Asc(Zeichenfolge)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
	Beliebige Zeichenfolge	Zeichenfolge	Ja

Hinweise

Wenn `zeichenfolge` mehr als ein Zeichen enthält, gibt die Funktion den ASCII-Wert des ersten Zeichens in der Zeichenfolge zurück.

Beispiele

`Asc("A")` gibt 65 zurück.

`Asc("ab")` gibt 97 zurück.

`Asc([Land])` gibt 85 zurück, wenn der Wert von `[Land]` "USA" ist.

1.6.1.3.2 Zeich

Beschreibung

Gibt das einem ASCII-Code entsprechende Zeichen zurück.

Funktionsgruppe

Zeichen

Syntax

```
Zeichenfolge Zeich(ASCII_Code)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
ASCII_Code	Ein ASCII-Code	Numerisch	Ja

Hinweise

Wenn `zahl` eine Dezimalzahl ist, werden die Dezimalstellen von der Funktion ignoriert.

Beispiel

S

`Zeichen(123)` gibt "{" zurück.

1.6.1.3.3 Verkettung

Beschreibung

Verkettet (verknüpft) zwei Zeichenfolgen. Bei Zahlen summiert die Funktion die Werte, statt diese zu verketteten.

ⓘ Hinweis

Wenn mindestens ein Eingabeparameter eine Zeichenfolge ist, werden alle anderen Eingabeparameter in Zeichenfolgen umgewandelt.

Funktionsgruppe

Zeichen

Syntax

```
Zeichenfolge Verkettung(erste_Zeichenfolge; zweite_Zeichenfolge)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
<code>erste_Zeichenfolge</code>	Erste Zeichenfolge	Zeichenfolge oder Zahl	Ja

Parameter	Beschreibung	Typ	Erforderlich
zweite_Zeichenfolge	Die Zeichenfolge wurde der ersten Zeichenfolge hinzugefügt.	Zeichenfolge oder Zahl	Ja

Hinweise

Sie können auch den Operator '+' verwenden, um Zeichenfolgen zu verketteten.

"First " + "Second" gibt "First Second" zurück.

"First " + "Second" + " Third" gibt "First Second Third" zurück.

Sie können mithilfe der Verkettung mehrere Dimensionen in eine Aggregationsfunktion einschließen. Beispiel: `Count([Sales Person]+[Quarter]+[Resort])` entspricht der Syntax `Count(<Sales Person>,<Quarter>,<Resort>)`, die bei Desktop Intelligence zulässig ist.

Beispiele

`Verkettung("Erste";"Zweite")` gibt "Erste Zweite" zurück.

`Verkettung("Erste";Verkettung("Zweite";"Dritte"))` gibt "Erste Zweite Dritte" zurück.

Wenn [A] eine Zahl und [A] = 1 ist, gibt `Concatenation([A];[A])` den Wert "2" zurück.

Wenn [A] eine Zeichenfolge und [A] = 1 ist, gibt `Concatenation([A];[A])` den Wert "11" zurück.

Wenn [A] eine Zeichenfolge, [B] eine Zahl, [A] = 1 und [B] = 2 ist, gibt `Concatenation([A];[B])` den Wert "12" zurück.

1.6.1.3.4 Füllen

Beschreibung

Erstellt eine Zeichenfolge durch n-fache Wiederholung einer Zeichenfolge.

Funktionsgruppe

Zeichen

Syntax

```
Zeichenfolge Füllen(wiederholte_Zeichenfolge; Anz_Wiederholungen)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
wiederholte_Zeichenfolge	Zu wiederholende Zeichenfolge	Zeichenfolge	Ja
Anz_Wiederholungen	Anzahl der Wiederholungen	Numerisch	Ja

Beispiele

Füllen("New York ";2) gibt "New York New York" zurück.

1.6.1.3.5 FormatDatum

Beschreibung

Weist einem Datum das angegebene Format zu.

Funktionsgruppe

Zeichen

Syntax

```
Zeichenfolge FormatDatum(Datum;Format_Zeichenfolge)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Datum	Zu formatierendes Datum	Datum	Ja
Format_Zeichenfolge	Anzuwendendes Format	Zeichenfolge	Ja

Hinweise

- Das Ausgabeformat hängt vom Datumsformat ab, das auf die Zelle angewendet wurde.
- Die Farbformatierungszeichenfolgen (z.B. [Rot], [Blau] usw.) können nicht auf die Funktion `FormatDatum` angewendet werden.

Beispiele

`FormatDatum(AktuellesDatum() ; "dd/MM/yyyy")` gibt "15/12/2005" zurück, wenn das aktuelle Datum der 15. Dezember 2005 ist.

Weitere Informationen

[Benutzerdefinierte Formate \[Seite 36\]](#)

1.6.1.3.5.1 Beispiele zu "format_string" für die Funktion "FormatDate"

Für die Syntax `FormatDate` für `format_string` können Sie die Beispiele aus der nachfolgenden Tabelle verwenden.

Hinweis

Sie finden diese Beispiele im Dialogfeld [Zahl formatieren](#) im Rich-Client. Welche Beispiele angezeigt werden, hängt jedoch vom ausgewählten Produktgebietsschema in Ihren BI-Launchpad-Einstellungen ab. Wenn Sie beispielsweise [Englisch](#) auswählen, ist "September 21, 2004" ein verfügbares Beispiel.

Beispiel	Syntax
Dienstag, September 21, 2004	dddd', 'mmmm d', 'yyyy
September 21, 2004	mmmm d', 'yyyy

Beispiel	Syntax
Sep 21, 2004	mmm d', 'yyyy
9/21/04	M'/'d'/'yy
Sep 21, 2004 8:45:30 PM	mmm d', 'yyyy h': 'mm': 'ss a
9/21/04 8:45 PM	M'/'d'/'yy h': 'mm a
9/21/2004	M'/'d'/'yyyy
09/21/2004	MM'/'d'/'yyyy
9/21/04 8:45:30 PM	M'/'d'/'yy h': 'mm a
8:45:30 PM	h': 'mm': 'ss a
8:45 PM	h': 'mm a
20:45:30	HH': 'mm': 'ss
20h45	HH'h'mm

→ Tipp

Es wird empfohlen, tatsächlichen Text in der Syntax in Apostrophe zu setzen, damit der Text nicht fälschlicherweise als Mustersymbol interpretiert werden kann, wie z. B. im letzten Beispiel in der obigen Tabelle (das 'h' in "HH'h'mm").

Weitere Informationen

[FormatDatum \[Seite 90\]](#)

[Benutzerdefinierte Formate \[Seite 36\]](#)

1.6.1.3.6 FormatZahl

Beschreibung

Weist einer Zahl das angegebene Format zu.

Funktionsgruppe

Zeichen

Syntax

```
Zeichenfolge FormatZahl (Zahl;Format_Zeichenfolge)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Zahl	Zu formatierende Zahl	Zahl	Ja
Format_Zeichenfolge	Anzuwendendes Format	Zeichenfolge	Ja

Hinweise

- Das Ausgabeformat hängt vom Zahlenformat ab, das auf die Zelle angewendet wurde.
- Die Farbformatierungszeichenfolgen (z. B. [Rot], [Blau] usw.) können nicht auf die Funktion `FormatZahl` angewendet werden.

Beispiele

`FormatZahl ([Umsatz]; "#,##.00")` gibt 835.420,00 zurück, wenn der für [Umsatz] angegebene Betrag 835.420 ist.

Weitere Informationen

[Benutzerdefinierte Formate \[Seite 36\]](#)

1.6.1.3.7 HTMLEncode

Beschreibung

Wendet HTML-Codierungsregeln auf eine Zeichenfolge an.

Funktionsgruppe

Zeichen

Syntax

```
Zeichenfolge HTMLEncode(html)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
html	Eine HTML-Zeichenfolge	Zeichenfolge	Ja

Beispiele

`HTMLEncode("<Hello World! >")` gibt "<Hello World! >" zurück, da der Browser das Zeichen interpretiert. Intern wird "<Hello World!>" zurückgegeben.

1.6.1.3.8 GroßAnfang

Beschreibung

Formatiert den ersten Buchstaben einer Zeichenfolge als Großbuchstaben.

Funktionsgruppe

Zeichen

Syntax

```
Zeichenfolge GroßAnfang(Zeichenfolge)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Zeichenfolge	Zeichenfolge, die in Großbuchstaben geschrieben werden soll	Zeichenfolge	Ja

Beispiele

GroßAnfang("was lange währt, wird endlich gut") gibt "Was lange währt, wird endlich gut" zurück.

1.6.1.3.9 Links

Beschreibung

Gibt die ersten Zeichen von links in einer Zeichenfolge zurück.

Hinweis

Diese Funktion gibt die ersten Zeichen vom logischen Anfang der Zeichenfolge zurück. Sprachen im Anzeige-/Lesemodus von rechts nach links, wie z. B. Arabisch oder Hebräisch, werden unterstützt.

Funktionsgruppe

Zeichen

Syntax

```
Zeichenfolge Links(Zeichenfolge;num_Zeichen)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Zeichenfolge	Eingabezeichenfolge	Zeichenfolge	Ja
num_Zeichen	Anzahl der Zeichen, die vom Anfang der Zeichenfolge zurückgegeben werden sollen	Zahl	Ja

Beispiel

`Links([Land];2)` gibt "Fr" zurück, wenn das angegebene [Land] "Frankreich" ist.

1.6.1.3.10 AuffüllenLinks

Beschreibung

Ergänzt eine Zeichenfolge insoweit, dass die angegebene Mindestlänge erreicht wird. Dies erfolgt durch Hinzufügen einer bestimmten Zeichenfolge auf der linken Seite der Zeichenfolge.

Hinweis

Diese Funktion ergänzt die Zeichenfolgen vom logischen Anfang. Sprachen im Anzeige-/Lesemodus von rechts nach links, wie z. B. Arabisch oder Hebräisch, werden unterstützt.

Funktionsgruppe

Zeichen

Syntax

```
Zeichenfolge AuffüllenLinks (aufgefüllte_Zeichenfolge;Länge;linke_Zeichenfolge)
```


Eingabe

Parameter	Beschreibung	Typ	Erforderlich
aufgefüllte_Zeichenfolge	Die ursprüngliche Zeichenfolge.	Zeichenfolge	Ja
Länge	Die Länge der Ausgabezeichenfolge.	Zahl	Ja
linke_Zeichenfolge	Zeichenfolge, die am Anfang von <code>padded_string</code> eingefügt werden soll	Zeichenfolge	Ja

Hinweise

- Wenn `length` kleiner als die Längen von `left_string` und `padded_string` zusammen ist, wird `left_string` abgeschnitten.
- Wenn `length` kleiner als die Länge oder gleich der Länge von `padded_string` ist, gibt die Funktion `padded_string` zurück.
- Wenn `length` größer als die Längen von `padded_string` und `left_string` zusammen ist, wird `left_string` wiederholt oder teilweise wiederholt, bis die Länge erreicht ist.

Beispiele

`AuffüllenLinks("York";8,"New")` gibt "New York" zurück.

`AuffüllenLinks("York",6,"New")` gibt "NeYork" zurück.

`AuffüllenLinks("York";11,"New")` gibt "New NewYork" zurück.

`AuffüllenLinks("New";2;"York")` gibt "New" zurück.

1.6.1.3.11 LöschLeerLinks

Beschreibung

Entfernt führende Leerzeichen und Sonderzeichen aus einer Zeichenfolge.

ⓘ Hinweis

Diese Funktion entfernt die ersten Zeichen aus dem logischen Anfang der Zeichenfolge. Sprachen im Anzeige-/Lesemodus von rechts nach links, wie z. B. Arabisch oder Hebräisch, werden unterstützt.

Funktionsgruppe

Zeichen

Syntax

```
Zeichenfolge LöschLeerLinks(gekürzte_Zeichenfolge[ ;char])
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
gekürzte_Zeichenfolge	Die Zeichenfolge, in der Stellen am Anfang bzw. Ende gelöscht werden sollen	Zeichenfolge	Ja
Zeichenfolge	zu entfernendes Zeichen	Zeichenfolge	Nein

Beispiele

- `LeftTrim([Produkt])` gibt "Laptop" zurück, wenn [Produkt] "Laptop" ist.
- `LeftTrim([Produkt]; "=")` gibt "Laptop" zurück, wenn [Produkt] "==Laptop" ist.

1.6.1.3.12 Länge

Beschreibung

Gibt die Anzahl der in einer Zeichenfolge enthaltenen Zeichen zurück.

Funktionsgruppe

Zeichen

Syntax

```
Ganzz Länge( Zeichenfolge )
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Zeichenfolge	Einzugebende Gesamtzeichenfolge	Zeichenfolge	Ja

Beispiele

Länge([Familienname]) gibt 5 zurück, wenn als [Familienname] "Meier" angegeben wurde.

1.6.1.3.13 KleinAlleBuchst

Beschreibung

Konvertiert eine Zeichenfolge in Kleinbuchstaben.

Funktionsgruppe

Zeichen

Syntax

```
Zeichenfolge KleinAlleBuchst( Zeichenfolge )
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Zeichenfolge	Zeichenfolge, die in Kleinbuchstaben konvertiert werden soll	Zeichenfolge	Ja

Beispiele

`KleinAlleBuchst("New York")` gibt "new york" zurück.

1.6.1.3.14 Vergleich

Beschreibung

Ermittelt, ob eine Zeichenfolge mit einem Muster übereinstimmt.

Funktionsgruppe

Zeichen

Syntax

```
Bool Vergleich(Test_Zeichenfolge;Struktur)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Test_Zeichenfolge	Zeichenfolge, die mit dem Textmuster verglichen werden soll	Zeichenfolge	Ja
Struktur	Textmuster	Zeichenfolge	Ja

Hinweise

- Das Muster kann die Platzhalterzeichen "*" (für jede beliebige Zeichenfolge) oder "?" (für ein beliebiges einzelnes Zeichen) enthalten.

Beispiele

Vergleich([Land]; "F*") gibt "Wahr" zurück, wenn als [Land] "Frankreich" angegeben wurde.

Vergleich([Land]; "?S?") gibt "Wahr" zurück, wenn als [Land] "USA" angegeben wurde.

Vergleich("New York"; "P*") gibt "Falsch" zurück.

1.6.1.3.15 Pos

Beschreibung

Gibt die Anfangsposition eines Textmusters in einer Zeichenfolge zurück.

Funktionsgruppe

Zeichen

Syntax

```
Ganzz Pos(Testzeichenfolge;Struktur[;Anfang][;Ende])
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Testzeichenfolge	Zeichenfolge, die mit dem Textmuster verglichen werden soll	Zeichenfolge	Ja
Struktur	Textmuster	Zeichenfolge	Ja

Parameter	Beschreibung	Typ	Erforderlich
Anfang	Startposition der Suche nach der Zeichenfolge	Ganzzahl	Nein
Ende	Endposition der Suche nach der Zeichenfolge	Ganzzahl	Nein

Hinweise

- Wenn das Muster mehr als einmal vorkommt, gibt `Pos` die Position des ersten Vorkommens zurück.
- Die Suche erfolgt zwischen Start- und Endposition (eingeschlossen).

Beispiele

`Pos("New York"; "Ne")` gibt 1 zurück.

`Pos("New York, New York"; "Ne")` gibt 1 zurück.

`Pos("New York"; "York")` gibt 5 zurück.

`Pos("Hello World World"; "World"; 7)` gibt 7 zurück.

`Pos("Hello World World"; "World"; 8)` gibt 13 zurück.

`Pos("Hello World World"; "World"; 8; 13)` gibt 13 zurück.

`Pos("Hello World World"; "World"; 8; 10)` gibt 0 zurück.

1.6.1.3.16 Ersetzen

Beschreibung

Ersetzt einen Teil einer Zeichenfolge durch eine andere Zeichenfolge.

Funktionsgruppe

Zeichen

Syntax

```
Zeichenfolge Ersetzen(ersetzen_in;ersetzte_Zeichenfolge;ersetzen_durch)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
ersetzen_in	Zeichenfolge, die den zu ersetzenden Text enthält	Zeichenfolge	Ja
ersetzte_Zeichenfolge	Zu ersetzender Text	Zeichenfolge	Ja
ersetzen_durch	Text, der ersetzte_Zeichenfolge ersetzt	Zeichenfolge	Ja

Beispiele

Ersetzen("New YORK" ; "ORK" ; "ork") gibt "New York" zurück.

1.6.1.3.17 Umkehren

Beschreibung

Gibt die umgekehrte Zeichenfolge zurück.

Funktionsgruppe

Zeichen

Syntax

```
Zeichenfolge Umkehren(Zeichenfolge)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Zeichenfolge	Die umzukehrende Zeichenfolge	Zeichenfolge	Ja

Beispiele

`Reverse("abc123")` gibt "321cba" zurück.

1.6.1.3.18 Rechts

Beschreibung

Gibt die ersten Zeichen von rechts in einer Zeichenfolge zurück (die Zeichen am Ende der Zeichenfolge).

Hinweis

Diese Funktion gibt die ersten Zeichen vom logischen Ende der Zeichenfolge zurück. Sprachen im Anzeige-/Lesemodus von rechts nach links, wie z. B. Arabisch oder Hebräisch, werden unterstützt.

Funktionsgruppe

Zeichen

Syntax

```
Zeichenfolge Rechts(Zeichenfolge;num_Zeichen)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Zeichenfolge	Beliebige Zeichenfolge	Zeichenfolge	Ja

Parameter	Beschreibung	Typ	Erforderlich
num_Zeichen	Anzahl der von rechts zurückzugebenden Zeichen	Zahl	Ja

Beispiele

`Rechts ([Land] ; 2)` gibt "ch" zurück, wenn das angegebene [Land] "Frankreich" ist.

1.6.1.3.19 AuffüllenRechts

Beschreibung

Ergänzt eine Zeichenfolge insoweit, dass die angegebene Mindestlänge erreicht wird. Dies erfolgt durch Hinzufügen einer bestimmten Zeichenfolge zum rechten Ende der Zeichenfolge.

Hinweis

Diese Funktion ergänzt die Zeichenfolgen vom logischen Ende. Sprachen im Anzeige-/Lesemodus von rechts nach links, wie z. B. Arabisch oder Hebräisch, werden unterstützt.

Funktionsgruppe

Zeichen

Syntax

```
Zeichenfolge AuffüllenRechts (aufgefüllte_Zeichenfolge;Länge;rechte_Zeichenfolge)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
aufgefüllte_Zeichenfolge	Die ursprüngliche Zeichenfolge.	Zeichenfolge	Ja

Parameter	Beschreibung	Typ	Erforderlich
Länge	Die Länge der Ausgabezeichenfolge.	Zahl	Ja
rechte_Zeichenfolge	Zeichenfolge, die am Ende von <code>padded_string</code> eingefügt werden soll	Zeichenfolge	Ja

Hinweise

- Wenn `length` kleiner als die Längen von `right_string` und `padded_string` zusammen ist, wird `right_string` abgeschnitten.
- Wenn `length` kleiner als die Länge oder gleich der Länge von `padded_string` ist, gibt die Funktion `padded_string` zurück.
- Wenn `length` größer als die Längen von `padded_string` und `right_string` zusammen ist, wird `right_string` wiederholt oder teilweise wiederholt, bis die Länge erreicht ist.

Beispiele

- `AuffüllenRechts ("New " ; 8 ; "York")` gibt "New York" zurück.
- `AuffüllenRechts ("New " ; 6 ; "York")` gibt "New Yo" zurück.
- `AuffüllenRechts ("New" ; 11 ; "York")` gibt "New YorkYor" zurück.
- `AuffüllenRechts ("New" ; 2 ; "York")` gibt "New" zurück.

1.6.1.3.20 LösSchLeerRechts

Beschreibung

Entfernt nachstehende Leerzeichen und Sonderzeichen aus einer Zeichenfolge.

Hinweis

Diese Funktion entfernt die letzten Zeichen aus dem logischen Ende der Zeichenfolge. Sprachen im Anzeige-/Lesemodus von rechts nach links, wie z. B. Arabisch oder Hebräisch, werden unterstützt.

Funktionsgruppe

Zeichen

Syntax

```
Zeichenfolge LöschesRechts(gekürzte_Zeichenfolge[;char])
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
gekürzte_Zeichenfolge	Die Zeichenfolge, in der Stellen am Anfang bzw. Ende gelöscht werden sollen	Zeichenfolge	Ja
Zeichenfolge	zu entfernendes Zeichen	Zeichenfolge	Nein

Beispiele

- `RightTrim([Produkt])` gibt "Laptop" zurück, wenn [Produkt] "Laptop" ist.
- `RightTrim([Produkt]; "=")` gibt "Laptop" zurück, wenn [Produkt] "Laptop==" ist.

1.6.1.3.21 RPos

Beschreibung

Gibt die Anfangsposition des letzten Vorkommens eines Textmusters in einer Zeichenfolge zurück

Funktionsgruppe

Zeichen

Syntax

```
Ganzz RPos(Testzeichenfolge;Struktur[;Anfang][;Ende])
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Testzeichenfolge	Zeichenfolge, die mit dem Textmuster verglichen werden soll	Zeichenfolge	Ja
Struktur	Textmuster	Zeichenfolge	Ja
Anfang	Startposition der Suche nach der Zeichenfolge	Ganzzahl	Nein
Ende	Endposition der Suche nach der Zeichenfolge	Ganzzahl	Nein

Hinweise

Die Suche erfolgt zwischen Start- und Endposition (eingeschlossen). Die Suche erfolgt vom Ende der Zeichenfolge aus rückwärts.

Beispiele

`RPos ("Hello World World"; "World")` gibt 13 zurück.

`RPos ("Hello World World"; "World"; 7)` gibt 13 zurück.

`RPos ("Hello World World"; "World"; 8)` gibt 13 zurück.

`RPos ("Hello World World"; "World"; 8; 13)` gibt 13 zurück.

`RPos ("Hello World World"; "World"; 1; 10)` gibt 7 zurück.

`RPos ("Hello World World"; "World"; 1; 6)` gibt 0 zurück.

1.6.1.3.22 Teilzeichenfolge

Beschreibung

Gibt einen Teil einer Zeichenfolge zurück.

Funktionsgruppe

Zeichen

Syntax

```
Zeichenfolge Teilzeichenfolge ( Zeichenfolge ; Anfang ; Länge )
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Zeichenfolge	Beliebige Zeichenfolge	Zeichenfolge	Ja
Anfang	Die Startposition der extrahierten Zeichenfolge	Numerisch	Ja
Länge	Die Länge der extrahierten Zeichenfolge	Numerisch	Ja

Beispiele

Teilzeichenfolge ("Großbritannien" ; 1 ; 4) gibt "Groß" zurück.

Teilzeichenfolge ("Großbritannien" ; 5 ; 7) gibt "britann" zurück.

1.6.1.3.23 Löschen

Beschreibung

Entfernt führende und nachstehende Leerzeichen aus der zu durchsuchenden Zeichenfolge. Wenn ein anderes Zeichen als Parameter übergeben wird, wird dieses Zeichen entfernt.

Funktionsgruppe

Zeichen

Syntax

```
Zeichenfolge Löschen ( Zeichenfolge [ ; char ] )
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Zeichenfolge	Die Zeichenfolge, in der Stellen am Anfang bzw. Ende gelöscht werden sollen	Zeichenfolge	Ja
Zeichenfolge	zu entfernendes Zeichen	Zeichenfolge	Nein

Beispiele

Löschen(" Großbritannien ") gibt "Großbritannien" zurück.

Trim (Trim ("---=Hello= ---" ; "-") ; "=") gibt "-Hello=" zurück.

1.6.1.3.24 GroßAlleBuchst

Beschreibung

Konvertiert eine Zeichenfolge in Großbuchstaben.

Funktionsgruppe

Zeichen

Syntax

```
Zeichenfolge GroßAlleBuchst( Zeichenfolge )
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Zeichenfolge	Zu konvertierende Zeichenfolge	Zeichenfolge	Ja

Beispiele

`GroßAlleBuchst("New York")` gibt "NEW YORK" zurück.

1.6.1.3.25 URLKodierung

Beschreibung

Wendet URL-Kodierungsregeln auf eine Zeichenfolge an.

Funktionsgruppe

Zeichen

Syntax

```
Zeichenfolge URLKodierung(html)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
html	URL, die kodiert werden soll	Zeichenfolge	Ja

Beispiele

`URLKodierung("http://www.sap.com")` gibt "http%3A%2F%2Fwww%2Esap%2Ecom" zurück.

1.6.1.3.26 GroßAlleWörter

Beschreibung

Formatiert den ersten Buchstaben aller Wörter in einer Zeichenfolge als Großbuchstaben.

Funktionsgruppe

Zeichen

Syntax

```
Zeichenfolge GroßAlleWörter(Zeichenfolge)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
string	Zeichenfolge, die in Großbuchstaben geschrieben werden soll	Zeichenfolge	Ja

Beispiele

`GroßAlleWörter("Umsatz für den Monat März")` gibt "Umsatz Für Den Monat März" zurück.

1.6.1.4 Datums- und Uhrzeitfunktionen

1.6.1.4.1 AktuellesDatum

Beschreibung

Gibt das aktuelle Datum mit dem in den Ländereinstellungen festgelegten Format zurück.

Funktionsgruppe

Datum und Uhrzeit

Syntax

```
Datum AktuellesDatum()
```

Beispiele

`AktuellesDatum()` gibt den 10. September 2002 zurück, wenn das aktuelle Datum dem 10. September 2002 entspricht.

1.6.1.4.2 AktuelleUhrzeit

Beschreibung

Gibt die aktuelle Zeit mit dem in den Ländereinstellungen festgelegten Format zurück.

Funktionsgruppe

Datum und Uhrzeit

Syntax

```
Zeit AktuelleUhrzeit()
```

Beispiele

`AktuelleUhrzeit` gibt 11:15 zurück, wenn die aktuelle Uhrzeit 11:15 ist.

1.6.1.4.3 Funktion "DatesBetween"

Beschreibung

Gibt die Anzahl der Zeiträume zwischen zwei Datumsangaben unabhängig der Uhrzeit zurück.

Funktionsgruppe

Datum und Uhrzeit

Syntax

```
int DatesBetween(first_date;last_date;period)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
first_date	Das erste Datum des Zeitraums	Datum	Ja
last_date	Das letzte Datum des Zeitraums	Datum	Ja
period	Der Typ der Periode, der im Zeitraum gezählt wird.	Vordefiniert	Ja

Hinweis

- Folgende Werte sind für den period-Parameter möglich: DayPeriod, WeekPeriod, MonthPeriod, QuarterPeriod, SemesterPeriod, YearPeriod.

⚠ Achtung

Wenn Sie den weekPeriod-Parameter verwenden, geht die Anwendung nicht davon aus, dass eine Woche aus sieben Tagen besteht. Eine Wochendifferenz kann einen bis sieben Tage umfassen, je nachdem, welcher Tag für die Berechnung verwendet wird. Außerdem ist Montag, entsprechend dem Standard ISO 8601, als erster Tag der Woche definiert, d.h., dass zwischen einem Montag und dem vorausgegangenen Sonntag immer eine Differenz von einer Woche liegt.

- Wenn der Rückgabewert für "int" außerhalb des gültigen Bereichs liegt, wird ein #OVERFLOW-Fehler zurückgegeben.

Beispiele

- DatesBetween([Begin Date];[End Date];MonthPeriod) gibt 2 zurück, wenn [Begin Date] der 30. Juni 2016 und [End Date] der 3. August 2016 ist.

- `DatesBetween([Begin Date];[End Date];DayPeriod)` gibt -10 zurück, wenn [Begin Date] der 30. Juni 2016 und [End Date] der 20. Juni 2016 ist.
- `DatesBetween([Begin Date];[End Date];QuarterPeriod)` gibt 6 zurück, wenn [Begin Date] der 30. Juni 2016 und [End Date] der 17. November 2017 ist.
- `DatesBetween([Begin Date];[End Date];MonthPeriod)` gibt 1 zurück, wenn [Begin Date] der 31. Dezember 2015 und [End Date] der 1. Januar 2016 ist.
- `DatesBetween([Begin Date];[End Date];DayPeriod)` gibt 1 zurück, wenn [Begin Date] der 31. Dezember 2015 und [End Date] der 1. Januar 2016 ist.
- `DatesBetween([Begin Date];[End Date];WeekPeriod)` gibt 0 zurück, wenn [Begin Date] der 31. Dezember 2015 und [End Date] der 1. Januar 2016 ist, weil beide Tage in der gleichen Woche liegen.

1.6.1.4.4 NameDesTages

Beschreibung

Gibt den Namen des Tages in einer Datumsangabe zurück.

Funktionsgruppe

Datum und Uhrzeit

Syntax

Zeichenfolge `NameDesTages(Datum)`

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Datum	Einzugebendes Datum	Datum	Ja

Beispiele

`NameDesTages([Reservierungsdatum])` gibt "Samstag" zurück, wenn das als [Reservierungsdatum] angegebene Datum der 15. Dezember 2001 ist (ein Samstag).

Hinweis

Das Eingabedatum muss eine Variable sein. Sie können das Datum nicht direkt angeben, wie in `NameDesTages ("15/07/2001")`

1.6.1.4.5 NummerTagDesMonats

Beschreibung

Gibt den Zahlenwert des Tages in einer Monatsangabe zurück.

Funktionsgruppe

Datum und Uhrzeit

Syntax

```
Ganzz NummerTagDesMonats (Datum)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Datum	Einzugebendes Datum	Datum	Ja

Beispiele

`NummerTagDesMonats ([Reservierungsdatum])` gibt 15 zurück, wenn das als `[Reservierungsdatum]` angegebene Datum der 15. Dezember 2001 ist.

1.6.1.4.6 NummerTagDerWoche

Beschreibung

Gibt den Zahlenwert des Tages in einer Wochenangabe zurück.

Funktionsgruppe

Datum und Uhrzeit

Syntax

```
Ganzz NummerTagDerWoche (Datum)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Datum	Einzugebendes Datum	Datum	Ja

Hinweise

Die Funktion sieht Montag als ersten Tag der Woche an.

Beispiele

NummerTagDerWoche ([Reservierungsdatum]) gibt 1 zurück, wenn das Datum in [Reservierungsdatum] der 2. Mai 2005 (ein Montag) ist.

1.6.1.4.7 NummerTagDesJahres

Beschreibung

Gibt den Zahlenwert des Tages in einer Jahresangabe zurück.

Funktionsgruppe

Datum und Uhrzeit

Syntax

```
Ganzz NummerTagDesJahres (Datum)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Datum	Einzugebendes Datum	Datum	Ja

Beispiele

NummerTagDesJahres ([Reservierungsdatum]) gibt 349 zurück, wenn das als [Reservierungsdatum] angegebene Datum der 15. Dezember 2001 ist.

1.6.1.4.8 TageZwischen

Beschreibung

Gibt die Anzahl der Tage zwischen zwei Datumsangaben zurück.

Funktionsgruppe

Datum und Uhrzeit

Syntax

```
Ganzz TageZwischen(erstes_Datum;eeztes_Datum)
```

Hinweis

Sie müssen sicherstellen, dass die in den Argumenten bereitgestellten Datumsangaben aus derselben Zeitzone stammen. Dies gilt für alle Datumsvorgänge: Vergleich und Berechnung.

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
erstes_Datum	Erstes	Datum	Ja
letztes_Datum	Letztes Datum	Datum	Ja

Beispiele

`TageZwischen([Verkaufsdatum]; [Rechnungsdatum])` gibt 2 zurück, wenn das als [Verkaufsdatum] angegebene Datum der 15. Dezember 2001 und das als [Rechnungsdatum] angegebene Datum der 17. Dezember 2001 ist.

1.6.1.4.9 LetzterTagDesMonats

Beschreibung

Gibt das Datum des letzten Tages eines Monats zurück.

Funktionsgruppe

Datum und Uhrzeit

Syntax

```
Datum LetzterTagDesMonats (Datum)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Datum	Beliebiges Datum im Monat	Datum	Ja

Beispiele

`LetzterTagDesMonats ([Verkaufsdatum])` gibt den 31. Dezember 2005 zurück, wenn [Verkaufsdatum] dem 11. Dezember 2005 entspricht.

1.6.1.4.10 LetzterTagDerWoche

Beschreibung

Gibt das Datum des letzten Tages einer Woche zurück.

Funktionsgruppe

Datum und Uhrzeit

Syntax

```
Datum LetzterTagDerWoche (Datum)
```


Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Datum	Beliebiges Datum in der Woche	Datum	Ja

Hinweise

Die Funktion sieht Montag als ersten Tag der Woche an.

Beispiele

`LetzterTagDerWoche ([Verkaufsdatum])` gibt den 15. Mai 2005 (einen Sonntag) zurück, wenn [Verkaufsdatum] dem 11. Mai 2005 entspricht.

1.6.1.4.11 Monat

Beschreibung

Gibt den Monatsnamen in einer Datumsangabe zurück.

Funktionsgruppe

Datum und Uhrzeit

Syntax

```
Zeichenfolge Monat (Datum)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Datum	Einzugebendes Datum	Datum	Ja

Beispiele

`Monat([Reservierungsdatum])` gibt "Dezember" zurück, wenn das als [Reservierungsdatum] angegebene Datum der 15. Dezember 2005 ist.

1.6.1.4.12 NummerMonatDesJahres

Beschreibung

Gibt den Zahlenwert des Monats in einer Datumsangabe zurück.

Funktionsgruppe

Datum und Uhrzeit

Syntax

```
Ganzz NummerMonatDesJahres(Datum)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Datum	Beliebiges Datum im Jahr	Datum	Ja

Beispiel

`NummerMonatDesJahres([Reservierungsdatum])` gibt 12 zurück, wenn das als [Reservierungsdatum] angegebene Datum der 15. Dezember 2005 ist.

1.6.1.4.13 MonateZwischen

Beschreibung

Gibt die Anzahl der Monate zwischen zwei Datumsangaben zurück.

Funktionsgruppe

Datum und Uhrzeit

Syntax

```
Ganzzahl MonateZwischen(erstes_Datum;letztes_Datum)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
erstes_Datum	Erstes	Datum	Ja
letztes_Datum	Letztes Datum	Datum	Ja

Beispiele

`MonateZwischen([Verkaufsdatum];[Rechnungsdatum])` gibt 1 zurück, wenn das als [Verkaufsdatum] angegebene Datum der 2. Dezember 2005 und das als [Rechnungsdatum] angegebene Datum der 2. Januar 2006 ist.

`MonateZwischen([Verkaufsdatum];[Rechnungsdatum])` gibt 1 zurück, wenn das als [Verkaufsdatum] angegebene Datum der 31.03.2008 und das als [Rechnungsdatum] angegebene Datum der 30.04.2008 ist.

MonateZwischen([Verkaufsdatum]; [Rechnungsdatum]) gibt 118 zurück, wenn das als [Verkaufsdatum] angegebene Datum der 07.01.1993 und das als [Rechnungsdatum] angegebene Datum der 06.11.2002 ist.

1.6.1.4.14 Quartal

Beschreibung

Gibt den Zahlenwert des Quartals in einer Datumsangabe zurück.

Funktionsgruppe

Datum und Uhrzeit

Syntax

```
Ganzz Quartal(Datum)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Datum	Beliebiges Datum im Quartal	Datum	Ja

Beispiele

Quartal([Reservierungsdatum]) gibt 4 zurück, wenn das als [Reservierungsdatum] angegebene Datum der 15. Dezember 2005 ist.

1.6.1.4.15 DatumRelativ

Beschreibung

Gibt ein Datum relativ zu einem anderen Datum zurück.

Funktionsgruppe

Datum und Uhrzeit

Syntax

```
Datum DatumRelativ(start_date;num;period)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Startdatum	Startdatum	Datum	Ja
num	Anzahl der dem Startdatum hinzugefügten Zeitraum-Einheiten	Nummer	Ja
period	Dem Startdatum hinzugefügter Zeitraumtyp	Vordefiniert	Optional

Hinweise

- Bei dem Parameter `num` kann es sich um eine Konstante, das numerische Ergebnis einer Funktion, einen Kennzahlenwert oder einen numerischen Dimensionswert handeln; allerdings muss der Parameter eine Ganzzahl sein.
- Der Parameter `num` kann einen negativen Wert aufweisen, um ein Datum zurückzugeben, das vor dem `start_date` (Startdatum) liegt.
- Wird kein Wert angegeben, verwendet der Parameter `period` (Zeitraum) Tage (`DayPeriod`).
- Beim Addieren oder Subtrahieren von Monaten (für `SemesterPeriod` (Semesterzeitraum), `QuarterPeriod` (QuartalZeitraum) und `MonthPeriod` (MonatZeitraum)) muss der letzte Tag des zurückgegebenen Monats verwendet werden, wenn der betreffende Tag im zurückgegebenen Monat nicht vorhanden ist.
- Folgende Werte sind für den Zeitraum-Parameter ("period") möglich: `MillisecondPeriod` (Millisekundenzeitraum), `SecondPeriod` (Sekundenzeitraum), `MinutePeriod` (MinuteZeitraum), `HourPeriod` (StundeZeitraum), `DayPeriod` (TagZeitraum), `WeekPeriod` (WocheZeitraum), `MonthPeriod` (MonatZeitraum), `QuarterPeriod` (QuartalZeitraum), `SemesterPeriod` (Semesterzeitraum), `YearPeriod` (JahrZeitraum).

Beispiele

`DatumRelativ([Reservierungsdatum];2)` gibt den 17. Dezember 2005 zurück, wenn das als `[Reservierungsdatum]` angegebene Datum dem 15. Dezember 2005 entspricht.

`DatumRelativ([Reservierungsdatum];-3)` gibt den 9. Januar 2007 zurück, wenn das als `[Reservierungsdatum]` angegebene Datum dem 12. Januar 2007 entspricht.

`DatumRelativ([Reservierungsdatum];1;MonatZeitraum)` gibt den 12. Februar 2007 zurück, wenn das als `[Reservierungsdatum]` angegebene Datum dem 12. Januar 2007 entspricht.

1.6.1.4.16 Funktion "TimeBetween"

Beschreibung

Gibt die Anzahl der Perioden zwischen zwei Datumsangaben zurück, wobei die Zeit berücksichtigt wird.

Funktionsgruppe

Datum und Uhrzeit

Syntax

```
int TimeBetween(first_date;last_date;period)
```

Hinweis

Achten Sie darauf, dass die in den Argumenten angegebenen Datumsangaben für die gleiche Zeitzone gelten, da bei der Berechnung des Rückgabewerts kein Zeitonenversatz verwendet wird.

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
<code>first_date</code>	Das erste Datum des Zeitraums	Zeit	Ja
<code>last_date</code>	Das letzte Datum des Zeitraums	Zeit	Ja

Parameter	Beschreibung	Typ	Erforderlich
period	Der Typ der Periode, der im Zeitraum gezählt wird.	Vordefiniert	Ja

Hinweise

- Folgende Werte sind für den period-Parameter möglich: DayPeriod (Millisekundenzeitraum), WeekPeriod (Sekundenzeitraum), MonthPeriod (MinuteZeitraum), QuarterPeriod (StundeZeitraum), SemesterPeriod (TagZeitraum), YearPeriod (WocheZeitraum), HourPeriod (MonatZeitraum), MinutePeriod (QuartalZeitraum), SecondPeriod (Semesterzeitraum), MillisecondPeriod (JahrZeitraum).
- Wenn der Rückgabewert für "int" außerhalb des gültigen Bereichs liegt, wird ein #OVERFLOW-Fehler zurückgegeben.

Beispiele

- TimeBetween([Begin Date];[End Date];HourPeriod) gibt 2 zurück, wenn [Begin Date] der 30. Juni 2016, 8:45 Uhr, und [End Date] der 30. Juni 2016, 10:05 Uhr, ist.
- TimeBetween([Begin Date];[End Date];MinutePeriod) gibt -10 zurück, wenn [Begin Date] der 30. Juni 2016, 8:45 Uhr, und [End Date] der 30. Juni 2016, 08:35 Uhr, ist.
- TimeBetween([Begin Date];[End Date];SecondPeriod) gibt 120 zurück, wenn [Begin Date] der 30. Juni 2016, 8:45 Uhr, und [End Date] der 30. Juni 2016, 08:47 Uhr, ist.
- TimeBetween([Begin Date];[End Date];MonthPeriod) gibt 1 zurück, wenn [Begin Date] der 31. Dezember 2015, 11:45 Uhr, und [End Date] der 1. Januar 2016, 8:47 Uhr, ist.
- TimeBetween([Begin Date];[End Date];DayPeriod) gibt 1 zurück, wenn [Begin Date] der 31. Dezember 2015, 11:45 Uhr, und [End Date] der 1. Januar 2016, 8:47 Uhr, ist.
- TimeBetween([Begin Date];[End Date];WeekPeriod) gibt 0 zurück, wenn [Begin Date] der 31. Dezember 2015, 11:45 Uhr, und [End Date] der 1. Januar 2016, 8:47 Uhr, ist, weil beide Tage in der gleichen Woche liegen.

1.6.1.4.17 ZeitDim

Beschreibung

Die Zeitdimension `ZeitDim` ermöglicht es Ihnen, eine Zeitachse aus einem Datumstyp-Universumsobjekt zu erstellen. `ZeitDim` gibt die Daten für die als ersten Parameter angegebenen Datumsangaben im Laufe der als zweiten Parameter angegebenen Zeiträume zurück. Wenn Zeiträume ohne Daten vorhanden sind, wird der erste Tag jedes leeren Zeitraums zurückgegeben. So wird eine vollständige Achse für den angegebenen Zeitraum gewährleistet. Dadurch wird Folgendes sichergestellt:

- Die Achse behält die natürliche zeitliche Reihenfolge bei (älteste Objekte an erster Stelle, aktuelle Objekte an letzter Stelle).
- Die Achse enthält alle Zeiträume zwischen den Mindest- und Höchstdatumsangaben im aktuellen Kontext.

Hinweis

Sie können mit der ZeitDim-Funktion keine Filtervorgänge für Formeln ausführen (zum Beispiel in einem Filter, einem Eingabesteuerelement, einer Elementverknüpfung, einer Filter-/Drill-Leiste). Stattdessen sollten Sie den Filter direkt für die zugrunde liegende Datumsdimension ausführen.

Funktionsgruppe

Datum und Uhrzeit

Syntax

```
ZeitDim([Datumstyp]; Zeitraumtyp)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Datumstyp	Das Datumsobjekt für den Bericht, beispielsweise Rechnungsdatum.	Datum	Ja
Zeitraumtyp	<p>Der Zeitraum für die Ergebnisse aus folgenden Werten:</p> <ul style="list-style-type: none"> • DayPeriod • MonthPeriod • QuarterPeriod • YearPeriod <p>Wenn kein Wert ausgewählt ist, wird standardmäßig TagZeitraum verwendet. Dieses Objekt sollte ein Datenprovider-Objekt sein. Es muss von Berichtsobjekten verfügbar und darf keine Variable sein.</p>	Vordefiniert	Optional

Verwenden Sie obige Funktion zusammen mit folgenden Funktionen:

- DayName
- DayNumberOfMonth
- DayNumberOfWeek
- DayNumberOfYear
- Month
- MonthNumberOfYear
- Quarter
- Year
- FormatDate

Beispiel

Die erste der folgenden Tabellen enthält Daten, die nur bestimmte Datumsangaben betreffen. Die anschließenden Abfrage-Beispiele zeigen, wie die Ergebnisse interpretiert werden.

Rechnungsdatum	Umsatz
03.01.2000	31.607
08.01.2000	31.244
03.07.2000	38.154

Die folgende Formel `NameDesTages(ZeitDim([Rechnungsdatum] ; ZeitraumQuartal)` gibt Tageswerte aus der obigen Tabelle zurück.

Rechnungsdatum	Umsatz
03.01.2000	31.607
08.01.2000	31.244
01.04.2000	
03.07.2000	38.154

Die Ergebnisse der `zeitDim`-Funktion sollten mit der `Quartal`-Funktion formatiert werden, um die Ergebnisse nach Quartal (Q1, Q2...) zurückzugeben, damit Sie die folgende Ergebnistabelle erhalten:

Rechnungsdatum	Umsatz
Q1	62.851
Q2	
Q3	38.154

1.6.1.4.18 ToDate

Beschreibung

Wandelt eine Zeichenfolge in ein Datum um. Geben Sie das Datumsformat als Parameter an, um Web Intelligence anzuweisen, wie die Zeichenfolge in ein Datum zu konvertieren ist. Das bereitgestellte Datumsformat muss mit dem Format des Datums in der ursprünglichen Zeichenfolge übereinstimmen. Die möglichen Datumsformate erfahren Sie über den Hyperlink unten.

Funktionsgruppe

Datum und Uhrzeit

Syntax

```
date ToDate(date_string;format[;cutoff_year])
```

oder

```
date ToDate(date_string;"INPUT_DATE_TIME"[;cutoff_year])
```

ⓘ Hinweis

In Szenarios, in denen das *bevorzugte Anzeigebereichsschema* je nach Benutzer unterschiedlich sein kann, ist ein festes Format (für ein bestimmtes Gebietsschema) nicht geeignet. Verwenden Sie in diesem Fall INPUT_DATE_TIME, wie oben im zweiten Beispiel gezeigt.

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
date_string	Die als Datum zu interpretierende Zeichenfolge	Zeichenfolge	Ja
format	Das von der Zeichenfolge verwendete Datumsformat. Verwenden Sie "INPUT_DATE_TIME", um das Format des bevorzugten	Zeichenfolge	Ja*

Parameter	Beschreibung	Typ	Erforderlich
	Anzeigegebietsschemas zu verwenden.		
cutoff_year	Optionaler Parameter zur Angabe des Jahres, das für den Abschluss verwendet wird. Standardmäßig ist das das Jahr 2029.	Ganzzahl	Nein

* Siehe Hinweis oben. Verwenden Sie je nach Bedarf das Format oder `INPUT_DATE_TIME`.

Beispiele

Bei `ToDate("12/15/2002" ; "MM/dd/yyyy")` wird "12" als Monatszahl, "15" als Zahlenwert des Tages und "2002" als Jahr interpretiert.

Bei `ToDate("Dec/02" ; "Mmm/yy")` wird "Dec" als Abkürzung des Monatsnamens und "02" als die letzten beiden Ziffern einer Jahreszahl interpretiert.

Bei `ToDate("15-December-02" ; "dd-Mmmm-yy")` wird "15" als Nummer eines Tages, "December" als Monat und "02" als die letzten beiden Ziffern einer Jahreszahl interpretiert.

`ToDate("12/15/02 11:00:00" ; "INPUT_DATE_TIME")` wird "12/15/02 11:00:00" in dem Format, das in dem *bevorzugten Anzeigegebietsschema* auf dem Rechner des Benutzers verwendet wird interpretiert.

→ Tipp

Um Verwirrung oder unerwünschte Ergebnisse zu verhindern, verwenden Sie für das Jahresargument vier Ziffern. "07" könnte z. B. "1907" oder "2007" bedeuten.

ⓘ Hinweis

- Bei `INPUT_DATE_TIME` müssen in der Eingabezeichenfolge `date_string` sowohl das Datum als auch die Uhrzeit angegeben werden.
- Wenn `date_string` nicht als gültiges Datum mit dem angegebenen Format interpretiert werden kann, gibt die Formel `ToDate()` die Fehlermeldung `#ERROR` zurück.
- Wie ein Datum in einer Zelle angezeigt wird, hängt von dem in der jeweiligen Zelle ausgewählten Datumsformat ab. Wenn das ausgewählte Datumsformat beispielsweise `"MM/dd/yyyy"` lautet, wird `ToDate("Dec/15/02" ; "Mmm/dd/yy")` im Format 12/15/2002 angezeigt.
- Wenn das Jahr in `date_string` aus zwei Ziffern besteht und ein Sperrjahr (`cutoff_year`) angegeben wird, gilt Folgendes:
 - `cutoff_year` liegt im aktuellen Jahrhundert.
 - Die letzten beiden Ziffern von `cutoff_year` definieren den Schwellenwert zur Verwendung des aktuellen Jahrhunderts von diesem `cutoff_year` oder dem vorherigen.
- Wenn `cutoff_year` unter 100 liegt, wird eine Fehlermeldung ausgegeben.

Weitere Informationen

[Benutzerdefinierte Formate \[Seite 36\]](#)

1.6.1.4.19 Woche

Beschreibung

Gibt den Zahlenwert der Woche in einer Jahresangabe zurück.

Funktionsgruppe

Datum und Uhrzeit

Syntax

```
Ganzz Woche(Datum)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Datum	Einzugebendes Datum	Datum	Ja

Beispiele

Woche([Reservierungsdatum]) gibt 1 zurück, wenn das als [Reservierungsdatum] angegebene Datum der 4. Januar 2004 ist (Tag der ersten Woche des Jahres 2004).

1.6.1.4.20 Jahr

Beschreibung

Gibt das Jahr in einer Datumsangabe zurück.

Funktionsgruppe

Datum und Uhrzeit

Syntax

```
Ganzz Jahr(Datum)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Datum	Einzugebendes Datum	Datum	Ja

Beispiele

Jahr([Reservierungsdatum]) gibt 2005 zurück, wenn das als [Reservierungsdatum] angegebene Datum der 15. Dezember 2005 ist.

1.6.1.5 Datenprovider-Funktionen

1.6.1.5.1 Verbindung

Beschreibung

Gibt die Parameter der von einem Datenprovider verwendeten Datenbankverbindung zurück.

Funktionsgruppe

Datenprovider

Syntax

```
Zeichenfolge Verbindung(dp)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
dp	Der Datenprovider	Datenprovider	Ja

Hinweise

- Der Name des Datenproviders muss in eckige Klammern eingeschlossen werden.
- Aus Sicherheitsgründen enthält die Ausgabe der Funktion weder den Datenbank-Hostnamen noch den Benutzernamen bzw. das Benutzerkennwort.

1.6.1.5.2 Datenprovider

Beschreibung

Gibt den Namen der Abfrage zurück, wie er vom Benutzer im Dialogfeld [Abfrageeigenschaften](#) definiert wurde.

Funktionsgruppe

Datenprovider

Syntax

```
Zeichenfolge Datenprovider(Obj)
```

Zeichenfolge Datenprovider(dp)

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Obj	Ein Berichtsobjekt	Berichtsobjekt	Ja
dp	Ein Datenprovider.	Datenprovider	Ja

Beispiele

Datenprovider([Gesamtumsatz]) gibt "Verkauf" zurück, wenn sich die Variable [Gesamtumsatz] in einem Datenprovider mit dem Namen "Verkauf" befindet.

Hinweis

Datenprovider erfordert, dass ein Objektname dessen Datenprovidernamen zurückgibt. Wenn Sie eine andere Funktion als Parameter von Datenprovider verwenden, beispielsweise eine Dimensionsvariable, die keinen Objektnamen zurückgibt, gibt die Datenprovider-Funktion einen Fehler zurück.

1.6.1.5.3 DatenproviderSchlüsseldatum

Beschreibung

Gibt das Schlüsseldatum eines Datenproviders zurück.

Funktionsgruppe

Datenprovider

Syntax

Datum DatenproviderSchlüsseldatum(dp)

Eingabe

Parameter	Beschreibung	Art	Erforderlich
dp	Der Datenprovider	Datenprovider	Ja

Hinweise

- Der Name des Datenproviders muss in eckige Klammern eingeschlossen werden.
- Das zurückgegebene Schlüsseldatum wird entsprechend dem Gebietsschema des Dokuments formatiert.
- Diese Funktion wird nur für Datenprovider unterstützt, die auf BEx Querys mit SAP-Variablen vom Typ KEYDATE basieren.
- Diese Funktion wird nur für ältere OLAP-UNV-Universen in BW unterstützt. Sie wird nicht für BEx Direct Access in Web Intelligence oder für BEx-UNX-Universen unterstützt. Diese Funktion ist veraltet, da UNV-Universen ab 4.3 nicht mehr unterstützt werden.

Beispiele

`DataProviderKeyDate([Sales])` gibt 3. August 2007 zurück, wenn das Schlüsseldatum für den Datenprovider [Umsatz] der 3. August 2007 ist.

1.6.1.5.4 DataProviderKeyDateCaption

Beschreibung

Gibt die Schlüsseldatumbeschriftung eines Datenproviders zurück.

Funktionsgruppe

Datenprovider

Syntax

```
Zeichenfolge DatenproviderSchlüsseldatumBeschriftung(dp)
```


Eingabe

Parameter	Beschreibung	Typ	Erforderlich
dp	Der Datenprovider	Datenprovider	Ja

Hinweise

- Der Name des Datenproviders muss in eckige Klammern eingeschlossen werden.
- Diese Funktion wird nur für Datenprovider unterstützt, die auf BEx Querys mit SAP-Variablen vom Typ KEYDATE basieren.
- Diese Funktion wird nur für ältere OLAP-UNV-Universen in BW unterstützt. Sie wird nicht für BEx Direct Access in Web Intelligence oder für BEx-UNX-Universen unterstützt. Diese Funktion ist veraltet, da UNV-Universen ab 4.3 nicht mehr unterstützt werden.

Beispiele

`DataProviderKeyDateCaption([Sales])` gibt "Aktuelles Kalenderdatum" zurück, wenn im Datenprovider [Umsatz] die Beschriftung für das Schlüsseldatum "Aktuelles Kalenderdatum" ist.

1.6.1.5.5 DatenproviderSQL

Beschreibung

Gibt die von einem Daten Provider generierte SQL zurück.

Funktionsgruppe

Datenprovider

Syntax

```
Zeichenfolge DatenproviderSQL(dp)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
dp	Der Datenprovider	Datenprovider	Ja

Hinweise

Der Name des Datenproviders muss in eckige Klammern eingeschlossen werden.

Beispiele

`DatenproviderSQL([Abfrage 1])` gibt `SELECT country.country_name FROM country` zurück, wenn die SQL des Datenproviders `SELECT country.country_name FROM country` lautet.

1.6.1.5.6 DatenproviderTyp

Beschreibung

Gibt den Typ eines Datenproviders zurück.

Funktionsgruppe

Datenprovider

Syntax

```
Zeichenfolge DatenproviderTyp(dp)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
dp	Der Datenprovider	Datenprovider	Ja

Hinweise

- `DataProviderType` gibt den Typ eines Datenproviders zurück. Mögliche Rückgabewerte: `Universe`, `Web Intelligence`, `Excel`, `Text`, `Free-hand SQL`, `SAP HANA`, `SAP BW` oder `Web Service`.
- Der Name des Datenproviders muss in eckige Klammern eingeschlossen werden.

Beispiele

`DatenproviderTyp([Umsatz])` gibt "Universum" zurück, wenn der Datenprovider [Umsatz] auf einem Universum basiert.

1.6.1.5.7 Datenquellenbeschreibung

Beschreibung

Gibt die Beschreibung der Datenquelle eines Datenproviders zurück.

Funktionsgruppe

Datenprovider

Syntax

```
Zeichenfolge Datenquellenbeschreibung ( dp )
```

```
string DataSourceDescription ( obj )
```

Eingabe

Parameter	Beschreibung	Art	Erforderlich
Obj	Ein Berichtsbjekt	Berichtsobjekt	Ja
dp	Ein Datenprovider.	Datenprovider	Ja

Hinweise

Die zurückgegebene Zeichenfolge wird abgerufen und im Dokument gespeichert, wenn diese Datenquelle regeneriert wird. Wenn diese Datenquelle nie regeneriert wurde, gibt diese Funktion eine leere Zeichenfolge zurück.

1.6.1.5.8 Datenquellenspeicherortart

Beschreibung

Gibt die Speicherortart der Datenquelle zurück.

Funktionsgruppe

Datenprovider

Syntax

```
Zeichenfolge Datenquellenspeicherortart ( dp )
```

```
string DataSourceLocationType ( obj )
```

Eingabe

Parameter	Beschreibung	Art	Erforderlich
Obj	Ein Berichtsbjekt	Berichtsobjekt	Ja

Parameter	Beschreibung	Art	Erforderlich
dp	Ein Datenprovider.	Datenprovider	Ja

Hinweise

Die zurückgegebene Zeichenfolge wird abgerufen und im Dokument gespeichert, wenn diese Datenquelle regeneriert wird. Wenn diese Datenquelle nie regeneriert wurde, gibt diese Funktion eine leere Zeichenfolge zurück.

1.6.1.5.9 Datenquellenpfad

Beschreibung

Gibt den vollständigen Verzeichnisordner einer Datenquelle zurück.

Funktionsgruppe

Datenprovider

Syntax

```
Zeichenfolge Datenquellenpfad ( dp )
```

```
string DataSourcePath ( obj )
```

Eingabe

Parameter	Beschreibung	Art	Erforderlich
Obj	Ein Berichtsobjekt	Berichtsobjekt	Ja
dp	Ein Datenprovider.	Datenprovider	Ja

Hinweise

Die zurückgegebene Zeichenfolge wird abgerufen und im Dokument gespeichert, wenn diese Datenquelle regeneriert wird. Wenn diese Datenquelle nie regeneriert wurde, gibt diese Funktion eine leere Zeichenfolge zurück.

1.6.1.5.10 DatenquelleÜbergeordneterOrdner

Beschreibung

Gibt den Namen des übergeordneten Ordners zurück, der die Datenquelle des Datenproviders enthält.

Funktionsgruppe

Datenprovider

Syntax

```
Zeichenfolge DatenquelleÜbergeordneterOrdner ( dp )
```

```
string DataSourceParentFolder ( obj )
```

Eingabe

Parameter	Beschreibung	Art	Erforderlich
Obj	Ein Berichtsobjekt	Berichtsobjekt	Ja
dp	Ein Datenprovider.	Datenprovider	Ja

Hinweise

Die zurückgegebene Zeichenfolge wird abgerufen und im Dokument gespeichert, wenn diese Datenquelle regeneriert wird. Wenn diese Datenquelle nie regeneriert wurde, gibt diese Funktion eine leere Zeichenfolge zurück.

1.6.1.5.11 Datenquellenname

Beschreibung

Gibt den Namen der Datenquelle zurück.

Funktionsgruppe

Datenprovider

Syntax

```
Zeichenfolge Datenquellenname ( dp )
```

```
string DataSourceName ( obj )
```

Eingabe

Parameter	Beschreibung	Art	Erforderlich
Obj	Ein Berichtsobjekt	Berichtsobjekt	Ja
dp	Ein Datenprovider.	Datenprovider	Ja

Hinweise

Die zurückgegebene Zeichenfolge wird abgerufen und im Dokument gespeichert, wenn diese Datenquelle regeneriert wird. Wenn diese Datenquelle nie regeneriert wurde, gibt diese Funktion eine leere Zeichenfolge zurück.

1.6.1.5.12 WurdeEingabeaufforderungBeantwortet

Beschreibung

Ermittelt, ob eine Eingabeaufforderung beantwortet wurde.

Funktionsgruppe

Datenprovider

Syntax

```
Bool WurdeEingabeaufforderungBeantwortet ([dp;]Eingabeaufforderungszeichenfolge)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
dp	Datenprovider, der die Eingabeaufforderung enthält	Datenprovider	Nein
Eingabeaufforderungszeichenfolge	Eingabeaufforderungstext	Zeichenfolge	Ja

Hinweise

- Der Name des Datenproviders muss in eckige Klammern eingeschlossen werden.
- WurdeEingabeaufforderungBeantwortet gibt einen booleschen Wert zurück, den Sie gemeinsam mit der Wenn-Funktion verwenden können.
- Wenn Sie WurdeEingabeaufforderungBeantwortet direkt in eine Spalte einfügen, wird eine Ganzzahl (1=wahr; 0=falsch) zurückgegeben. Sie können diese Ganzzahl mit einem booleschen Zahlenformat formatieren.

Beispiele

`WurdeEingabeaufforderungBeantwortet ("Ort wählen")` gibt "Wahr" zurück, wenn die durch den Text "Ort wählen" identifizierte Eingabeaufforderung beantwortet wurde.

`WurdeEingabeaufforderungBeantwortet ([Umsatz]; "Ort wählen")` gibt "Wahr" zurück, wenn die durch den Text "Ort wählen" identifizierte Eingabeaufforderung im Datenprovider [Umsatz] beantwortet wurde.

1.6.1.5.13 DatumLetzteAusführung

Beschreibung

Gibt das Datum zurück, an dem ein Datenprovider zuletzt aktualisiert wurde.

Funktionsgruppe

Datenprovider

Syntax

```
Datum DatumLetzteAusführung ( dp )
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
dp	Der Datenprovider	Datenprovider	Ja

Hinweise

- Wenn der Bericht nur über einen Datenprovider verfügt, können Sie den Parameter `dp` auslassen.
- Der Name des Datenproviders muss in eckige Klammern eingeschlossen werden.
- Sie können die Funktion `Datenprovider` verwenden, um einen Verweis auf einen Datenprovider einzufügen.

Beispiele

`DatumLetzteAusführung ([Umsatz-Abfrage])` gibt "4.3.2002" zurück, wenn der Datenprovider "Umsatz-Abfrage" zuletzt am 4. März 2002 regeneriert wurde.

Weitere Informationen

[Datenprovider \[Seite 134\]](#)

1.6.1.5.14 DauerLetzteAusführung

Beschreibung

Gibt die Zeit (in Sekunden) zurück, die für die letzte Regenerierung eines Datenproviders benötigt wurde.

Funktionsgruppe

Datenprovider

Syntax

```
Num DauerLetzteAusführung(dp)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
dp	Der Datenprovider	Datenprovider	Ja

Hinweise

Der Name des Datenproviders muss in eckige Klammern eingeschlossen werden.

Beispiele

`DauerLetzteAusführung([Umsatz])` gibt 3 zurück, wenn Datenprovider [Umsatz] 3 Sekunden zur Rückgabe der Daten bei der letzten Ausführung benötigt hat.

1.6.1.5.15 UhrzeitLetzteAusführung

Beschreibung

Gibt die Uhrzeit zurück, zu der ein Datenprovider zuletzt aktualisiert wurde.

Funktionsgruppe

Datenprovider

Syntax

```
Uhrzeit UhrzeitLetzteAusführung(dp)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
dp	Der Datenprovider	Datenprovider	Ja

Hinweise

- Wenn der Bericht nur über einen Datenprovider verfügt, können Sie den Parameter `dp` auslassen.
- Sie können die Funktion `Datenprovider` verwenden, um einen Verweis auf einen Datenprovider einzufügen.
- Der Name des Datenproviders muss in eckige Klammern eingeschlossen werden.

Beispiele

`UhrzeitLetzteAusführung([Umsatz-Abfrage])` gibt "14:48:00" zurück, ob der Datenprovider "Umsatz-Abfrage" zuletzt um 14:48:00 regeneriert wurde.

Weitere Informationen

[Datenprovider \[Seite 134\]](#)

1.6.1.5.16 AnzahlSpalten

Beschreibung

Gibt die Anzahl der Spalten in einem Datenprovider zurück

Funktionsgruppe

Datenprovider

Syntax

```
Ganzz AnzahlSpalten([dp])
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
dp	Der Datenprovider	Datenprovider	Ja

Beispiel

AnzahlSpalten([Abfrage 1]) gibt 5 zurück, wenn der Datenprovider "Abfrage 1" über 5 Zeilen verfügt.

1.6.1.5.17 AnzahlDatenprovider

Beschreibung

Gibt die Anzahl der in einem Bericht enthaltenen Datenprovider zurück.

Funktionsgruppe

Datenprovider

Syntax

```
Ganzz AnzahlDatenprovider()
```

Beispiele

AnzahlDatenprovider() gibt 2 zurück, wenn der Bericht über zwei Datenprovider verfügt.

1.6.1.5.18 AnzahlZeilen

Beschreibung

Gibt die Anzahl der Zeilen in einem Datenprovider zurück.

Funktionsgruppe

Datenprovider

Syntax

```
Ganzz AnzahlZeilen(dp)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
dp	Der Datenprovider	Datenprovider	Ja

Hinweise

- Der Name des Datenproviders muss in eckige Klammern eingeschlossen werden.
- Sie können die Funktion `Datenprovider` verwenden, um einen Verweis auf einen Datenprovider einzufügen.

Beispiele

`AnzahlZeilen([Abfrage 1])` gibt 10 zurück, wenn der Datenprovider [Abfrage 1] über 10 Zeilen verfügt.

Weitere Informationen

[Datenprovider \[Seite 134\]](#)

1.6.1.5.19 Abfragenname

Beschreibung

Gibt den Namen der Abfrage des Datenproviders zurück.

Funktionsgruppe

Datenprovider

Syntax

```
Zeichenfolge Abfragenname ( dp )
```

```
string QueryName ( obj )
```

Eingabe

Parameter	Beschreibung	Art	Erforderlich
Obj	Ein Berichtsobjekt	Berichtsobjekt	Ja
dp	Ein Datenprovider.	Datenprovider	Ja

1.6.1.5.20 RefWertDatum

Beschreibung

Gibt das Datum der für das Datentracking verwendeten Referenzdaten zurück.

Funktionsgruppe

Datenprovider

Syntax

```
Datum RefWertDatum( )
```

Beispiele

RefWertDatum() gibt den 15. Dezember 2008 zurück, wenn das Referenzdatum der 15. Dezember 2008 ist.

1.6.1.5.21 ReferenzwertAntwortBenutzer

Beschreibung

Gibt die Antwort auf eine Eingabeaufforderung zu der Zeit, als die Referenzdaten den aktuellen Daten entsprachen, zurück.

Funktionsgruppe

Datenprovider

Syntax

```
Zeichenfolge  
ReferenzwertAntwortBenutzer([dp;]Eingabeaufforderungszeichenfolge[;Index])
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
dp	Der Datenprovider	Datenprovider	Nein
Eingabeaufforderungszeichenfolge	Eingabeaufforderungstext	Zeichenfolge	Ja
Index	Weist die Funktion an, die Datenbankprimärschlüssel der Eingabeaufforderungswerte zurückzugeben	Schlüsselwort	Nein

Hinweise

- Die Funktion gibt eine leere Zeichenfolge zurück, wenn das Nachverfolgen von Daten nicht aktiviert ist.
- Der Name des Datenproviders muss in eckige Klammern eingeschlossen werden.
- Sie können die Funktion `Datenprovider` verwenden, um einen Verweis auf einen Datenprovider einzufügen.
- Wenn Sie mehr als einen Wert als Antwort auf eine Eingabeaufforderung ausgewählt haben, gibt die Funktion eine Zeichenfolge zurück, die aus einer durch Semikolons getrennten Werteliste (oder bei Angabe des `Index`-Operators aus Primärschlüsseln) besteht.

Beispiele

`ReferenzwertAntwortBenutzer("Welche Stadt?")` gibt "Los Angeles" zurück, wenn Sie "Los Angeles" in der Eingabeaufforderung "Welche Stadt?" zu der Zeit eingegeben haben, als die Referenzdaten mit den aktuellen Daten übereinstimmen.

`ReferenzwertAntwortBenutzer([Umsatz-Abfrage]; "Welche Stadt?")` gibt "Los Angeles" zurück, wenn Sie "Los Angeles" in der Eingabeaufforderung "Welche Stadt?" im Datenprovider "Umsatz-Abfrage" zu der Zeit eingegeben haben, als die Referenzdaten mit den aktuellen Daten übereinstimmten.

1.6.1.5.22 Serverwert

Beschreibung

Gibt den Datenbankwert einer Kennzahl zurück.

Funktionsgruppe

Datenprovider

Syntax

```
Num Serverwert([Kennzahl])
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Kennzahl	Beliebige Kennzahl	Kennzahl	Ja

Hinweise

- `Serverwert` ignoriert alle auf Dimensionen oder Hierarchien angewendeten lokalen Filter zum Berechnen der Kennzahl.

Beispiel

`Serverwert([Internet-Umsatzbetrag])` gibt den Datenbankwert der Kennzahl `[Internet-Umsatzwert]` zurück.

1.6.1.5.23 Universumsname

Beschreibung

Gibt den Namen des Universums oder der Datei zurück, auf dem bzw. der ein Datenprovider für Universen, Text, Excel- und FHSQL-Datenquellen basiert.

Funktionsgruppe

Datenprovider

Syntax

```
Zeichenfolge Universumsname(dp[, Zeichenfolge])
```

```
Zeichenfolge Universumsname(Obj[, Zeichenfolge])
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
dp	Der Datenprovider	Datenprovider	Ja

Hinweise

- Für SAP-BW- und SAP-HANA-Datenquellen geben die Funktionen den View-Namen zurück. Folgende Optionen können verwendet werden:
 - `key`: Gibt den technischen Namen des View zurück.
 - `caption`: Gibt die Beschriftung des View zurück.
 - `full`: Gibt den vollständigen Pfad des View zurück (Paket und Schlüssel). Zum Beispiel "sales.sales::revenue" für SAP HANA bzw. "xxx" für SAP BW.
- Der `dp`-Wert in der Formel wird automatisch aktualisiert, wenn sich der Name des Datenproviders ändert. Wenn der Datenprovider in "Q1" umbenannt wird, wird die Formel in `UniverseName([Q1])` geändert.
- Der Name des Datenproviders muss in eckige Klammern eingeschlossen werden.
- Sie können die Funktion `DataProvider` verwenden, um einen Verweis auf einen Datenprovider einzufügen.

Beispiele

Universumsname([Abfrage 1]) gibt "eFashion" zurück, wenn der Datenprovider [Abfrage 1] auf dem Universum "eFashion" basiert.

Weitere Informationen

[Datenprovider \[Seite 134\]](#)

1.6.1.5.24 AntwortBenutzer

Beschreibung

Gibt die Antwort auf eine Eingabeaufforderung zurück.

Funktionsgruppe

Datenprovider

Syntax

```
Zeichenfolge AntwortBenutzer([dp;]Eingabeaufforderungszeichenfolge[;Index])
```

```
string UserResponse ([dp;] prompt_string [;Index] [;multi_separator])
```

```
string UserResponse ([dp;] prompt_string [;Index] [;multi_separator|  
DefaultSeparator; between_separator])
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
dp	Der Datenprovider	Datenprovider	Nein
Eingabeaufforderungszeichenfolge	Eingabeaufforderungstext	Zeichenfolge	Ja

Parameter	Beschreibung	Typ	Erforderlich
Index	Weist die Funktion an, die Datenbankprimärschlüssel der Eingabeaufforderungswerte zurückzugeben	Schlüsselwort	Nein
Multitrennzeichen	Die Zeichenfolge zum Trennen verschiedener beantworteter Werte. Standardmäßig ist dies ";".	Zeichenfolge	Nein
Standardtrennzeichen	Die Standardzeichenfolge (";") zum Trennen verschiedener beantworteter Werte.	Schlüsselwort	Nein
Zwischentrennzeichen	Die Zeichenfolge, die verwendet wird, um die beiden beantworteten Werte eines Bereichs oder „Zwischen“-Operators in einer Eingabeaufforderung zu trennen.	Zeichenfolge	Nein

Hinweise

- Der Name des Datenproviders muss in eckige Klammern eingeschlossen werden.
- Sie können die Funktion `DataProvider` verwenden, um einen Verweis auf einen Datenprovider einzufügen.
- Wenn Sie mehr als einen Wert als Antwort auf eine Eingabeaufforderung auswählen, gibt die Funktion eine Zeichenfolge zurück, die aus einer durch Semikolons getrennten Werteliste (oder bei Angabe des Index-Operators aus Primärschlüsseln) besteht.
- Die Ausgabe der Funktion hängt vom Aufforderungstyp ab.

Beispiele

`AntwortBenutzer ("Welche Stadt?")` gibt "Madrid" zurück, wenn Sie für die Eingabeaufforderung "Welche Stadt?" den Ort "Madrid" eingegeben haben.

`AntwortBenutzer ([Umsatz-Abfrage] ; "Welche Stadt?")` gibt "Madrid" zurück, wenn Sie für die Eingabeaufforderung "Welche Stadt?" im Datenprovider "Umsatz-Abfrage" den Ort "Madrid" eingegeben haben.

`AntwortBenutzer ([Umsatz-Abfrage] ; "Welche Stadt?" ; Index)` gibt 23 zurück, wenn Sie für die Eingabeaufforderung "Welche Stadt?" im Datenprovider "Umsatz-Abfrage" den Ort "Madrid" eingegeben haben und der Datenbankprimärschlüssel von Madrid 23 lautet.

UserResponse("Which country?") gibt "Frankreich" zurück, wenn es sich um eine einwertige Eingabeaufforderung handelt.

UserResponse("Which country?") gibt "Frankreich;Holland;USA" zurück, wenn es sich um eine Mehrfachwert-Eingabeaufforderung handelt.

UserResponse("Which country?") gibt "Frankreich - USA" zurück, wenn es sich um eine Eingabeaufforderung vom Typ "Intervall" mit einem einzigen Eintrag handelt.

UserResponse("Which country?") gibt "Frankreich - Holland;Japan - USA" zurück, wenn es sich um eine Eingabeaufforderung vom Typ "Intervall" mit mehreren Einträgen handelt.

UserResponse("Which country?") gibt "Frankreich" zurück, wenn es sich um eine komplexe Eingabeaufforderung handelt, die mit dem Operator "Gleich" verwendet wird.

UserResponse("Which country?") gibt ">Frankreich" zurück, wenn es sich um eine komplexe Eingabeaufforderung handelt, die mit dem Operator "Größer als" verwendet wird.

UserResponse("Which country?") gibt ">=Frankreich" zurück, wenn es sich um eine komplexe Eingabeaufforderung handelt, die mit dem Operator "Größer als" oder "Gleich" verwendet wird.

UserResponse("Which country?") gibt "<Frankreich" zurück, wenn es sich um eine komplexe Eingabeaufforderung handelt, die mit dem Operator "Kleiner als" verwendet wird.

UserResponse("Which country?") gibt "<=Frankreich" zurück, wenn es sich um eine komplexe Eingabeaufforderung handelt, die mit dem Operator "Kleiner als" oder "Gleich" verwendet wird.

UserResponse("Which country?") gibt "Frankreich - USA" zurück, wenn es sich um eine komplexe Eingabeaufforderung handelt, die mit dem Operator "Zwischen" verwendet wird.

UserResponse("Which country?") gibt "!Frankreich" zurück, wenn es sich um eine komplexe Eingabeaufforderung handelt, die mit dem Operator "Ungleich" verwendet wird.

UserResponse("Which country?") gibt "Österreich;Frankreich - Holland;>Japan" zurück, wenn es sich um eine komplexe Eingabeaufforderung mit mehreren Einträgen handelt.

1.6.1.6 Dokumentfunktionen

1.6.1.6.1 DokumentAutor

Beschreibung

Gibt den InfoView-Anmeldenamen des Dokumenterstellers zurück.

Funktionsgruppe

Dokument

Syntax

```
Zeichenfolge DokumentAutor()
```

Beispiele

DokumentAutor() gibt "gkn" zurück, wenn die Anmeldung des Dokumentautors "gkn" lautet.

1.6.1.6.2 DokumenterstellungDatum

Beschreibung

Gibt das Datum zurück, an dem ein Dokument erstellt wurde.

Funktionsgruppe

Dokument

Syntax

```
Datum DokumenterstellungDatum()
```

Beispiele

DokumenterstellungDatum() gibt den 15. Dezember 2008 zurück, wenn das Dokument am 15. Dezember 2008 erstellt wurde.

1.6.1.6.3 DokumenterstellungUhrzeit

Beschreibung

Gibt die Uhrzeit zurück, zu der ein Dokument erstellt wurde.

Funktionsgruppe

Dokument

Syntax

```
Uhrzeit DokumenterstellungUhrzeit()
```

Beispiele

DokumenterstellungUhrzeit() gibt 11:15 zurück, wenn das Dokument um 11:15 erstellt wurde.

1.6.1.6.4 DokumentDatum

Beschreibung

Gibt das Datum zurück, an dem ein Dokument zuletzt gespeichert wurde.

Funktionsgruppe

Dokument

Syntax

```
Datum DokumentDatum()
```

Beispiele

DokumentDatum() gibt den 8. August 2005 zurück, wenn das Dokument zuletzt am 8. August 2005 gespeichert wurde.

1.6.1.6.5 DocumentDescription

Beschreibung

Gibt die Dokumentbeschreibung im bevorzugten Anzeigegebietsschema des Benutzers zurück.

Funktionsgruppe

Dokument

Syntax

```
string DocumentDescription()
```

Beispiel

`DocumentDescription()` "Analyse der Umsatzzahlen der vier Quartale von 2019" wird zurückgegeben, wenn die Dokumentbeschreibung "Analyse der Umsatzzahlen der vier Quartale von 2019" ist.

1.6.1.6.6 DokumentName

Beschreibung

Gibt den Namen des Dokuments zurück.

Funktionsgruppe

Dokument

Syntax

```
Zeichenfolge DokumentName( )
```

Beispiele

DokumentName() gibt "Verkaufsbericht" zurück, wenn das Dokument den Namen "Verkaufsbericht" trägt.

1.6.1.6.7 Dokumenteigentümer

Beschreibung

Gibt den BI-Launchpad-Anmelde-/Benutzernamen des Eigentümers des Dokuments (der Person, die das Dokument zuletzt gespeichert hat) zurück. (Um den ursprünglichen Autor/Verfasser) des Dokuments zurückzugeben, verwenden Sie die Funktion "DokumentAutor".)

Funktionsgruppe

Dokument

Syntax

```
Zeichenfolge Dokumenteigentümer( )
```

Beispiele

Dokumenteigentümer() gibt "gkn" zurück, wenn die Person, die das Dokument zuletzt gespeichert hat, den Benutzer- oder Anmeldenamen "gkn" hat.

1.6.1.6.8 DokumentÜbergeordneterOrdner

Beschreibung

Gibt den Namen des Ordners zurück, der das Dokument enthält.

Funktionsgruppe

Dokument

Syntax

```
string DocumentParentFolder()
```

Hinweise

- Diese Funktion gibt den Ordner zurück, der das aktuelle Dokument enthält, in dem diese Funktion verwendet wird.
- Wenn sie in einer Instanz verwendet wird, gibt diese Funktion den Ordner zurück, der das zeitgesteuerte Dokument enthält.
- Wenn das Dokument im Posteingang oder in den Favoriten eines Benutzers gespeichert ist, wird der Name dieses Benutzers zurückgegeben.

Beispiele

- Die Funktion `DokumentÜbergeordneterOrdner()` gibt `Root Folder` zurück, wenn sie in einem Dokument verwendet wird, das in "Öffentliche Ordner" gespeichert ist.
- `DocumentParentFolder()` gibt `Web Intelligence Samples` für ein Dokument zurück, das in diesem Ordner gespeichert ist.
- `DokumentÜbergeordneterOrdner()` gibt `userA` zurück, wenn sich das Dokument im Eingangs- oder Favoritenordner des Benutzers befindet.
- `DokumentÜbergeordneterOrdner()` gibt `ZZ_Charting` zurück, wenn die Funktion für eine Instanz dieses Dokuments verwendet wird.

1.6.1.6.9 DokumentTeilRegeneriert

Beschreibung

Ermittelt, ob ein Dokument teilweise regeneriert ist.

Funktionsgruppe

Dokument

Syntax

```
Bool DokumentTeilRegeneriert()
```

Hinweise

`DokumentTeilRegeneriert()` gibt einen booleschen Wert zurück, den Sie in der `IF`-Funktion verwenden können.

Beispiele

`DokumentTeilRegeneriert()` gibt "Wahr" zurück, wenn das Dokument teilweise regeneriert wurde.

1.6.1.6.10 DokumentPfad

Beschreibung

Gibt den Dokumentpfad zurück. Gibt in einer Dokumentinstanz den Pfad des Dokuments und seinen Namen zurück.

Funktionsgruppe

Dokument

Syntax

```
string DocumentPath()
```

Hinweise

Bei einem Dokument erhält der Pfad das Suffix "/". Dies ist für die Instanz eines Dokuments nicht der Fall.

Beispiele

- `DocumentPath()` gibt "Öffentliche Ordner/Web-Intelligence-Beispieldaten/" für ein in diesem Ordner gespeichertes Dokument zurück.
- `DocumentPath()` gibt "Öffentliche Ordner/Web-Intelligence-Beispieldaten/ZZ_Diagramme" zurück, wenn der Pfad für eine Instanz dieses zeitgesteuerten Dokuments verwendet wird.

1.6.1.6.11 DokumentUhrzeit

Beschreibung

Gibt die Uhrzeit zurück, zu der ein Dokument zuletzt gespeichert wurde.

Funktionsgruppe

Dokument

Syntax

```
Uhrzeit DokumentUhrzeit()
```

Hinweise

Das Format der zurückgegebenen Uhrzeit variiert abhängig vom Zellformat.

Beispiel:

`DokumentUhrzeit()` gibt 15:45 zurück, wenn das Dokument zuletzt um 15:45 gespeichert wurde.

1.6.1.6.12 DrillFilter

Beschreibung

Gibt die Ergebnisse von Drill-Filtern zurück, die auf ein Dokument oder Objekt in einem angegebenen Bericht im Drill-Modus angewendet wurden. Sie können im Dokument einen anderen Bericht angeben. Wenn Sie keinen Bericht angeben, wird der gerade aktive Bericht verwendet.

Funktionsgruppe

Dokument

Syntax

```
Zeichenfolge DrillFilter([Obj|Trennzeichen[;Bericht]])
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Obj	Ein Berichtsobjekt	Berichtsobjekt	Entweder Obj oder Trennzeichen ist erforderlich
Trennzeichen	Trennzeichen zwischen Drillfilterwerten	Zeichenfolge	Entweder Obj oder Trennzeichen ist erforderlich

Parameter	Beschreibung	Typ	Erforderlich
Bericht	Optional. Der Name des zu verwendenden Berichts. Er muss sich im Dokument befinden. Wenn kein Bericht angegeben ist, wird der aktuelle Bericht verwendet.	Zeichenfolge	Entweder Obj oder Trennzeichen ist erforderlich

Hinweise

- Sie können die Funktion `DrillFilter` direkt einfügen, ohne die Formel manuell durch Einfügen einer `DrillFilter`-Zelle eingeben zu müssen.
- Wenn kein Objekt angegeben wird, gibt die Funktion alle auf das Dokument angewendeten Drillfilter zurück.

Beispiele

`DrillFilter()` gibt "USA" zurück, wenn das Dokument einen Drillfilter aufweist, der das Objekt [Land] auf USA einschränkt.

`DrillFilter()` gibt "USA – 1999" zurück, wenn das Dokument einen Filter aufweist, der das Objekt [Land] auf "USA" und das Objekt [Jahr] auf 1999 einschränkt.

`DrillFilter(" / ")` gibt "USA / 1999" zurück, wenn das Dokument Filter aufweist, die das Objekt [Land] auf "USA" und [Jahr] auf 1999 einschränken.

`DrillFilter([Quartal])` gibt "Q3" zurück, wenn das Dokument einen Filter aufweist, der das Objekt [Quartal] auf "Q3" einschränkt.

1.6.1.6.13 ElementLinkingFilters

Beschreibung

Gibt die für den Block "Blockname" ausgewählten Werte für Elementverknüpfungen zurück.

Funktionsgruppe

Dokument

Syntax

```
string ElementLinkingFilters(string Block Name[:string separator[:string report]])
```

Eingabe

Parameter	Beschreibung	Art	Erforderlich
Zeichenfolge Block Name	Das Argument <code>Block Name</code> legt den Block fest, für den die Elementverknüpfung festgelegt wurde.	Zeichenfolge	Ja
Zeichenfolge Trennzeichen	Dies ist die Zeichenfolge, die zum Trennen verschiedener ausgewählter Werte verwendet wird. Standardmäßig ist dies "-".	Zeichenfolge	Nein
Zeichenfolge Report	Wenn ein Argument <code>report</code> angegeben wird, werden die ausgewählten Werte der Elementverknüpfung im angegebenen Bericht zurückgegeben. Andernfalls werden die Werte in dem Bericht zurückgegeben, in dem die Formel verwendet wird.	Zeichenfolge	Nein

Hinweise

- In einem Dokument, das aus einer vorherigen Version migriert wurde, funktioniert `ElementLinkingFilters` nur, nachdem neue Werte für die Elementverknüpfung ausgewählt und das Dokument gespeichert wurde.

1.6.1.6.14 InputControlFilter

Beschreibung

Gibt die vom Benutzer ausgewählten Werte über die Eingabesteuerung zurück.

Funktionsgruppe

Dokument

Syntax

```
string InputControlFilter(string InputControlName [;string separator][;string report])
```

Eingabe

Parameter	Beschreibung	Art	Erforderlich
Zeichenfolge InputControlName	Dies ist der Name des Eingabesteuerelements. Nur die von diesem Eingabesteuerelement ausgewählten Werte werden von der Funktion zurückgegeben.	Zeichenfolge	Ja
Zeichenfolge Trennzeichen	Dies ist die Zeichenfolge, die verwendet wird, um verschiedene ausgewählte Werte oder die für verschiedene Eingabesteuerelemente ausgewählten Werte zu trennen. Standardmäßig ist dies ";".	Zeichenfolge	Nein
Zeichenfolge Report	Wenn ein Argument <code>report</code> angegeben wird, werden die ausgewählten Werte des Eingabesteuerelements im angegebenen Bericht zurückgegeben. Andernfalls werden die Werte in dem Bericht zurückgegeben, in dem die Formel verwendet wird.	Zeichenfolge	Nein

Hinweise

- In einem Dokument, das aus einer vorherigen Version migriert wurde, funktioniert `InputControlFilter` nur, nachdem neue Werte für das Eingabesteuerelement ausgewählt und das Dokument gespeichert wurde.

1.6.1.6.15 Funktion "PromptSummary"

Beschreibung

Gibt den Eingabeaufforderungstext und die Benutzerantwort für alle Eingabeaufforderungen in einem Dokument zurück.

Funktionsgruppe

Dokument

Syntax

```
string PromptSummary([dp];[sorting_order];[show_definitive_prompts])
```

Eingabe

Parameter	Typ	Beschreibung	Erforderlich
dp	Datenprovider	Ein Datenprovider.	Nein
sorting_order	Vordefiniert	Sortierreihenfolge der Funktionsausgabe	Nein (Standardwert = ascending)
show_definitive_prompts	Boolesch	Definitive BW-Eingabeaufforderungen in der Funktionsausgabe anzeigen	Nein (Standardwert = false)

Hinweise

- Der optionale Einsatz von Filtern `dp` ermöglicht Ihnen, die Ausgabe der Eingabeaufforderungszusammenfassung () für die jeweilige Abfrage zu filtern.

- Folgende Werte sind für den `sorting_order`-Parameter möglich: `default`, `ascending` oder `descending`
- Wenn `sorting_order` nicht in der Funktion verwendet wird, wird die Datenquellenreihenfolge verwendet.
- Wenn der Parameter `show_definitive_prompts` nicht verwendet wurde, sind definitive BW-Eingabeaufforderungen in der Funktionsausgabe verborgen.

Beispiel

Ein Beispiel für die Ausgabe der Funktion `PromptSummary` sieht folgendermaßen aus:

```
Enter Quantity Sold: 5000
Enter value(s) for State (optional): California, Texas, Utah
Enter Customer (optional):
```

1.6.1.6.16 Abfragezusammenfassung

Beschreibung

Gibt Informationen zu den Abfragen in einem Dokument zurück. Für jede Abfrage gibt die Methode den Datenprovidertyp, den Datenprovidernamen, Informationen zur Regenerierung, die Abfrageeigenschaften und die Abfragedefinition (Ergebnisobjekte und Filter) zurück.

Funktionsgruppe

Dokument

Syntax

```
Zeichenfolge Abfragezusammenfassung( )
```

```
Zeichenfolge Abfragezusammenfassung([dp])
```

```
Zeichenfolge Abfragezusammenfassung([dp];[Datenstatus])
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
dp	Ein Datenprovider.	Datenprovider	Nein
Datenstatus	BW-Status der Daten	Boolesch	Nein

Hinweise

- Wenn Sie den Namen eines Datenproviders als Parameter übergeben, gibt die Funktion ausschließlich Informationen zu diesem Datenprovider zurück. Der Name dieses Datenproviders muss in eckige Klammern eingeschlossen werden.
- Vor dem Namen des Datenproviders steht dessen Typ: Universe, Web Intelligence, Excel, Text, Free-hand SQL, SAP HANA, SAP BW oder Web Service.
- Der BW-Status der Daten gibt das Datum der letzten Regenerierung des BW-Info-Providers an, er steht in der letzten von der Funktion zurückgegebenen Zeile.

Beispiele

`QuerySummary()` gibt Informationen zu allen Abfragen in einem Dokument zurück.

`QuerySummary([Query 1])` gibt auf der Grundlage des Datenproviders [Abfrage 1] Informationen zu den Abfragen zurück.

Ausgabebeispiel:

```
*** Query Name:Query 1 ***
** Query Properties:
  Universe:eFashion
  Last Refresh Date:4/1/20 5:15 PM
  Last Execution Duration: 2
  Number of rows: 586
  Refreshable: ON
  Retrieve Duplicate Rows: ON
  Retrieve Empty Rows: OFF
  Max Retrieval Time (s): /
  Max Rows Retrieved: /
  Query Stripping: OFF
** Query Definition:
  Result Objects: State, Year, Sales revenue, City, Quarter, Month
```

1.6.1.6.17 ReportFilter

Beschreibung

Gibt die auf ein Objekt angewendeten Berichtsfiler zurück.

Funktionsgruppe

Dokument

Syntax

```
Zeichenfolge ReportFilter(Obj[;Trennzeichen])
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Obj	Ein Berichtsobjekt	Berichtsobjekt	Ja
Trennzeichen	Dieses Zeichen kann verwendet werden, um verschiedene Werte in der Liste zu trennen. Das Standardtrennzeichen ist ";".	Zeichenfolge	Nein

Beispiele

`ReportFilter([Land])` gibt "USA" zurück, wenn das Objekt "Land" über einen Berichtsfiler verfügt, der das Objekt auf "USA" beschränkt.

1.6.1.6.18 Berichtfilterzusammenfassung

Beschreibung

Gibt eine Zusammenfassung der Berichtsfiler in einem Dokument oder Bericht zurück.

Funktionsgruppe

Dokument

Syntax

```
Zeichenfolge Berichtfilterzusammenfassung(Berichtsname)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Berichtsname	Name des Berichts	Zeichenfolge	Nein

Hinweise

Wenn `Berichtsname` nicht angegeben wird, gibt `Berichtfilterzusammenfassung` eine Zusammenfassung aller Berichtsfilter in allen Berichten des Dokuments zurück.

Beispiele

`Berichtfilterzusammenfassung()` gibt die Informationen zu allen Berichtsfiltern in einem Dokument zurück.

`Berichtfilterzusammenfassung("Bericht1")` gibt die Informationen zu den Berichtsfiltern in "Bericht1" zurück.

Ein Beispiel für die Ausgabe der Funktion `Berichtfilterzusammenfassung` sieht folgendermaßen aus:

```
Filters on Report1:
    (Sales Revenue Greater Than 1000000
    Or (Sales Revenue Less Than 3000))
Filters on Section on City:
    (City InList{"Los Angeles";"San Diego";})
Ranking Filter:
    (Top 10 & Bottom 10 [Customer] Based on [Sales
Revenue] (Count))
```

1.6.1.7 Logische Funktionen

1.6.1.7.1 Gerade

Beschreibung

Bestimmt, ob eine Zahl gerade ist.

Funktionsgruppe

Logisch

Syntax

```
Bool Gerade(Zahl)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Zahl	Eine beliebige Zahl	Nummer	Ja

Hinweise

- `Gerade` gibt einen booleschen Wert zurück, den Sie in der `IF`-Funktion verwenden können.
- Wenn Sie `Gerade` direkt in eine Spalte einfügen, wird eine Ganzzahl (1=wahr; 0=falsch) angezeigt. Sie können diese Ganzzahl mit einem booleschen Zahlenformat formatieren.
- `Even` gibt für jede gerade Ganzzahl "Wahr" und für jede Dezimalzahl "Falsch" zurück.

Beispiele

`Gerade(4)` gibt "Wahr" zurück.

`Gerade(3)` gibt "Falsch" zurück.

`Gerade(23,2)` gibt "Falsch" zurück.

Gerade(-4) gibt "Wahr" zurück.

Gerade(-2,2) gibt "Falsch" zurück.

1.6.1.7.2 IstDatum

Beschreibung

Ermittelt, ob es sich bei einem Wert um ein Datum handelt.

Funktionsgruppe

Logisch

Syntax

```
Bool IstDatum(Obj)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Obj	Beliebiges Berichtsobjekt.	Berichtsobjekt	Ja

Hinweise

- IstDatum gibt einen booleschen Wert zurück, den Sie in der If-Funktion verwenden können.
- Wenn Sie IstDatum direkt in eine Spalte einfügen, wird eine Ganzzahl (1=wahr; 0=falsch) angezeigt. Sie können diese Ganzzahl mit einem booleschen Zahlenformat formatieren.

Beispiele

IstDatum([Reservierungsdatum]) gibt "Wahr" zurück, wenn die sich bei [Reservierungsdatum] um ein Datum handelt.

Oder eine der folgenden Funktionen gibt "Datum" zurück, wenn es sich bei [Reservierungsdatum] um ein Datum handelt:

- `If(IsDate([Reservation Date])) Then "Date" Else "Not a date"`
- `If IsDate([Reservation Date]) Then "Date" Else "Not a date"`

Weitere Informationen

[Wenn...Dann...Sonst \[Seite 249\]](#)

1.6.1.7.3 IstFehler

Beschreibung

Ermittelt, ob ein Objekt einen Fehler zurückgibt.

Funktionsgruppe

Logische Operatoren

Syntax

```
Bool IstFehler(Obj)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Obj	Beliebiges Berichtsobjekt.	Berichtsobjekt	Ja

Hinweise

- `IstFehler` gibt einen booleschen Wert zurück, den Sie in der `If`-Funktion verwenden können.

- Wenn Sie `IstFehler` direkt in eine Spalte einfügen, wird eine Ganzzahl (1=wahr; 0=falsch) zurückgegeben. Sie können diese Ganzzahl mit einem booleschen Zahlenformat formatieren.

Beispiele

`IstFehler([Umsatz])` gibt "Falsch" zurück, wenn die Variable [Umsatz] keinen Fehler zurückgibt.

`IstFehler([Durchschnittliche Gästezahl])` gibt "Wahr" zurück, wenn die Variable [Durchschnittliche Gästezahl] einen #DIV/0-Fehler (Division durch null) zurückgibt.

`If IstFehler([Durchschnittliche Gästezahl]) Then "Fehler" Else "Kein Fehler"` gibt "Fehler" zurück, wenn die Variable [Durchschnittliche Gästezahl] einen #DIV/0-Fehler (Division durch null) zurückgibt.

Weitere Informationen

[Wenn...Dann...Sonst \[Seite 249\]](#)

1.6.1.7.4 IstBoolesch

Beschreibung

Ermittelt, ob ein Wert boolesch ist.

Funktionsgruppe

Logische Operatoren

Syntax

```
Bool IstBoolesch(Obj)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Obj	Beliebiges Berichtsobjekt.	Berichtsobjekt	Ja

Hinweise

- `IstBoolesch` gibt einen booleschen Wert zurück, den Sie in der `IF`-Funktion verwenden können.
- Wenn Sie `IstBoolesch` direkt in eine Spalte einfügen, wird eine Ganzzahl (1=wahr; 0=falsch) angezeigt. Sie können diese Ganzzahl mit einem booleschen Zahlenformat formatieren.

Beispiele

`IstBoolesch(IstZeichenfolge([Land]))` gibt "Wahr" zurück.

`IstBoolesch([Land])` gibt "Falsch" zurück, wenn [Land] einen anderen als den booleschen Datentyp enthält.

`If IstBoolesch(IstDatum([Land])) Then "Boolesch" Else "Nicht boolesch"` gibt "Boolesch" zurück.

Weitere Informationen

[Wenn...Dann...Sonst \[Seite 249\]](#)

1.6.1.7.5 IstNull

Beschreibung

Ermittelt, ob ein Wert einem Nullwert entspricht.

Funktionsgruppe

Logische Operatoren

Syntax

```
Bool IstNull(Obj)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Obj	Beliebiges Berichtsobjekt.	Berichtsobjekt	Ja

Hinweise

- `IstNull` gibt einen booleschen Wert zurück, den Sie in der `If`-Funktion verwenden können.
- Wenn Sie `IstNull` direkt in eine Spalte einfügen, wird eine Ganzzahl (1=wahr; 0=falsch) angezeigt. Sie können diese Ganzzahl mit einem booleschen Zahlenformat formatieren.

Beispiele

`IstNull([Umsatz])` gibt "Falsch" zurück, wenn die Variable `[Umsatz]` ungleich null ist.

`IstNull([Durchschnittliche Gästezahl])` gibt "Wahr" zurück, wenn die Variable `[Durchschnittliche Gästezahl]` null ist.

Weitere Informationen

[Wenn...Dann...Sonst \[Seite 249\]](#)

1.6.1.7.6 IstZahl

Beschreibung

Ermittelt, ob es sich bei einem Wert um eine Zahl handelt.

Funktionsgruppe

Logische Operatoren

Syntax

```
Bool IstZahl(Obj)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Obj	Beliebiges Berichtsobjekt.	Berichtsobjekt	Ja

Hinweise

- `IstZahl` gibt einen booleschen Wert zurück, den Sie in der `IF`-Funktion verwenden können.
- Wenn Sie `IstZahl` direkt in eine Spalte einfügen, wird eine Ganzzahl (1=wahr; 0=falsch) angezeigt. Sie können diese Ganzzahl mit einem booleschen Zahlenformat formatieren.

Beispiele

`IstZahl([Umsatz])` gibt "Wahr" zurück, wenn die Variable [Umsatz] eine Zahl ist.

`IstZahl([Kundenname])` gibt "Falsch" zurück, wenn die Variable [Kundenname] keine Zahl ist.

`If IstZahl([Kundenname]) Then "Zahl" Else "Keine Zahl"` gibt "Keine Zahl" zurück, wenn die Variable [Kundenname] keine Zahl ist.

Weitere Informationen

[Wenn...Dann...Sonst \[Seite 249\]](#)

1.6.1.7.7 IstZeichenfolge

Beschreibung

Ermittelt, ob es sich bei einem Wert um eine Zeichenfolge handelt.

Funktionsgruppe

Logische Operatoren

Syntax

```
Bool IstZeichenfolge(Obj)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Obj	Beliebiges Berichtsobjekt.	Berichtsobjekt	Ja

Hinweise

- `IstZeichenfolge` gibt einen booleschen Wert zurück, den Sie in der `If`-Funktion verwenden können.
- Wenn Sie `IstZeichenfolge` direkt in eine Spalte einfügen, wird eine Ganzzahl (1=wahr; 0=falsch) angezeigt. Sie können diese Ganzzahl mit einem booleschen Zahlenformat formatieren.

Beispiele

`IstZeichenfolge([Umsatz])` gibt "Falsch" zurück, wenn die Variable `[Umsatz]` keine Zeichenfolge ist.

`IstZeichenfolge([Kundenname])` gibt "Wahr" zurück, wenn die Variable `[Kundenname]` eine Zeichenfolge ist.

`If IstZeichenfolge([Kundenname]) Then "Zeichenfolge" Else "Keine Zeichenfolge"` gibt "Zeichenfolge" zurück, wenn die Variable `[Kundenname]` eine Zeichenfolge ist.

Weitere Informationen

[Wenn...Dann...Sonst \[Seite 249\]](#)

1.6.1.7.8 IstUhrzeit

Beschreibung

Ermittelt, ob es sich bei einer Variablen um eine Zeitangabe handelt.

Funktionsgruppe

Logische Operatoren

Syntax

```
Bool IstUhrzeit(Obj)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Obj	Beliebiges Berichtsobjekt.	Berichtsobjekt	Ja

Hinweise

- `IstUhrzeit` gibt einen booleschen Wert zurück, den Sie in der `If`-Funktion verwenden können.
- Wenn Sie `IstUhrzeit` direkt in eine Spalte einfügen, wird eine Ganzzahl (1=wahr; 0=falsch) angezeigt. Sie können diese Ganzzahl mit einem booleschen Zahlenformat formatieren.

Beispiele

`IstUhrzeit([Uhrzeit der Reservierung])` gibt "Wahr" zurück, wenn die Variable `[Uhrzeit der Reservierung]` eine Zeitangabe ist.

`IstUhrzeit([Durchschnittliche Gästezahl])` gibt "Falsch" zurück, wenn die Variable `[Durchschnittliche Gästezahl]` keine Zeitangabe ist.

`If IstUhrzeit([Durchschnittliche Gästezahl]) Then "Uhrzeit" Else "Keine Uhrzeit"` gibt "Keine Uhrzeit" zurück, wenn die Variable `[Durchschnittliche Gästezahl]` keine Zeitangabe ist.

Weitere Informationen

[Wenn...Dann...Sonst \[Seite 249\]](#)

1.6.1.7.9 Ungerade

Beschreibung

Bestimmt, ob eine Zahl ungerade ist.

Funktionsgruppe

Logisch

Syntax

```
Bool Ungerade(Zahl)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Zahl	Eine beliebige Zahl	Nummer	Ja

Hinweise

- `Ungerade` gibt einen booleschen Wert zurück, den Sie in der `IF`-Funktion verwenden können.
- Wenn Sie `Ungerade` direkt in eine Spalte einfügen, wird eine Ganzzahl (1=wahr; 0=falsch) angezeigt. Sie können diese Ganzzahl mit einem booleschen Zahlenformat formatieren.
- `odd` gibt für jede Dezimalzahl "Wahr" und für jede gerade Ganzzahl "Falsch" zurück.

Beispiele

`Ungerade (5)` gibt "Wahr" zurück.

`Ungerade (4)` gibt "Falsch" zurück.

`Ungerade (23 , 2)` gibt "Wahr" zurück.

`Ungerade (24 , 2)` gibt "Wahr" zurück.

`Ungerade (-23 , 2)` gibt "Wahr" zurück.

`Ungerade (-24 , 2)` gibt "Wahr" zurück.

Weitere Informationen

[Wenn...Dann...Sonst \[Seite 249\]](#)

1.6.1.8 Numerische Funktionen

1.6.1.8.1 Abs

Beschreibung

Gibt den absoluten Wert einer Zahl zurück.

Funktionsgruppe

Numerisch

Syntax

```
Num Abs (Zahl)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Zahl	Eine beliebige Zahl	Numerisch	Ja

Beispiele

Abs (25) gibt 25 zurück.

Abs (-11) gibt 11 zurück.

1.6.1.8.2 Obergrenze

Beschreibung

Gibt eine auf die nächste Ganzzahl aufgerundete Zahl zurück.

Funktionsgruppe

Numerisch

Syntax

```
Num Obergrenze (Zahl)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Zahl	Eine beliebige Zahl	Numerisch	Ja

Beispiele

`Obergrenze(2,4)` gibt 3 zurück.

`Obergrenze(3,1)` gibt 4 zurück.

`Obergrenze(-3,1)` gibt -3 zurück.

1.6.1.8.3 Cos

Beschreibung

Gibt den Kosinus eines Winkels zurück.

Funktionsgruppe

Numerisch

Syntax

```
Num Cos(Winkel)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Winkel	Winkel im Bogenmaß	Numerisch	Ja

Beispiel

`Cos(180)` gibt -0,6 zurück.

1.6.1.8.4 AusEuroKonvertieren

Beschreibung

Rechnet einen Euro-Betrag in eine andere Währung um.

Funktionsgruppe

Numerisch

Syntax

```
Num AusEuroKonvertieren(BetragEuro;Währungscode;Rundungsebene)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
BetragEuro	Betrag in Euro	Numerisch	Ja
Währungscode	ISO-Code der Zielwährung	Zeichenfolge	Ja
Rundungsebene	Anzahl der Dezimalstellen, auf die das Ergebnis auf- bzw. abgerundet werden soll	Numerisch	Ja

Hinweise

Der Währungscode muss der Code einer der 12 EU-Währungen sein, deren Werte im Verhältnis zum Euro vor der Abschaffung dieser Währungen im Januar 2002 festgelegt wurden. Andernfalls gibt die Funktion die Meldung #FEHLER zurück. Folgende Währungen sind verfügbar:

BEF	Belgische Franc
DEM	Deutsche Mark
GRD	Griechische Drachmen
ESP	Spanische Peseten
FRF	Französische Franc
IEP	Irische Pfund
ITL	Italienische Lire
LUF	Luxemburgische Franc
NLG	Holländische Gulden
ATS	Österreichische Schilling
PTS	Portugiesischer Escudo
FIM	Finnmark

Beispiele

`AusEuroKonvertieren(1000;"FRF";2)` gibt 6559,57 zurück.

`AusEuroKonvertieren(1000;"FRF";1)` gibt 6559,60 zurück.

`AusEuroKonvertieren(1000,04;"DEM";2)` gibt 1955,83 zurück.

`AusEuroKonvertieren(1000,04;"DEM";1)` gibt 1955,80 zurück.

Weitere Informationen

[Runden und Abschneiden von Zahlen \[Seite 295\]](#)

1.6.1.8.5 InEuroKonvertieren

Beschreibung

Rechnet einen Betrag in Euro um.

Funktionsgruppe

Numerisch

Syntax

```
Num InEuroKonvertieren(BetragNichtEuro;Währungscode;Rundungsebene)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
BetragNichtEuro	Betrag in der Ausgangswährung	Numerisch	Ja
Währungscode	ISO-Code der Ausgangswährung	Zeichenfolge	Ja
Rundungsebene	Anzahl der Dezimalstellen, auf die das Ergebnis auf- bzw. abgerundet werden soll	Numerisch	Ja

Beispiel

`InEuroKonvertieren(6559;"FRF";2)` gibt 999,91 zurück.

`InEuroKonvertieren(6559;"FRF";1)` gibt 999,90 zurück.

`InEuroKonvertieren(1955;"DEM";2)` gibt 999,58 zurück.

`InEuroKonvertieren(1955;"DEM";1)` gibt 999,60 zurück.

Hinweis

Der Währungscode muss der Code einer der 12 EU-Währungen sein, deren Werte im Verhältnis zum Euro vor der Abschaffung dieser Währungen im Januar 2002 festgelegt wurden. Andernfalls gibt die Funktion die Meldung #FEHLER zurück. Folgende Währungen sind verfügbar:

BEF	Belgische Franc
DEM	Deutsche Mark
GRD	Griechische Drachmen
ESP	Spanische Peseten
FRF	Französische Franc
IEP	Irische Pfund
ITL	Italienische Lire

LUF	Luxemburgische Franc
NLG	Holländische Gulden
ATS	Österreichische Schilling
PTS	Portugiesischer Escudo
FIM	Finnmark

Weitere Informationen

[Runden und Abschneiden von Zahlen \[Seite 295\]](#)

1.6.1.8.6 AusEuroRundungsfehler

Beschreibung

Gibt den Rundungsfehler bei der Umwandlung von Euro-Beträgen zurück.

Funktionsgruppe

Numerisch

Syntax

```
Num AusEuroRundungsfehler (BetragEuro; Währungscode; Rundungsebene)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
BetragEuro	Betrag in Euro	Zahl	Ja
Währungscode	ISO-Code der Zielwährung	Zeichenfolge	Ja
Rund_ebene	Anzahl der Dezimalstellen, auf die das Ergebnis auf- bzw. abgerundet werden soll	Zahl	Ja

Ausgabe

Rundungsfehler in der Berechnung

Beispiele

`AusEuroRundungsfehler(1000;"FRF";2)` gibt 0 zurück. (Der nicht gerundete Umrechnungsbetrag und der auf zwei Dezimalstellen auf- bzw. abgerundete Umrechnungsbetrag sind gleich.)

`AusEuroRundungsfehler(1000;"FRF";1)` gibt 0,03 zurück. (Der nicht gerundete Umrechnungsbetrag ist 6559,57. Der auf eine Dezimalstelle gerundete Umrechnungsbetrag ist 6559,60. Der Rundungsfehler beträgt 0,03.)

`AusEuroRundungsfehler(1000;"DEM";2)` gibt 0 zurück. (Der nicht gerundete Umrechnungsbetrag und der auf zwei Dezimalstellen auf- bzw. abgerundete Umrechnungsbetrag sind gleich.)

`AusEuroRundungsfehler(1000;"DEM";1)` gibt -0,01 zurück. (Der nicht gerundete Umrechnungsbetrag ist 1955,83. Der auf eine Dezimalstelle gerundete Umrechnungsbetrag ist 1995,80. Der Rundungsfehler beträgt -0,03.)

Hinweis

Der Währungscode muss der Code einer der 12 EU-Währungen sein, deren Werte im Verhältnis zum Euro vor der Abschaffung dieser Währungen im Januar 2002 festgelegt wurden. Wenn dies nicht der Fall ist, gibt die Funktion `#FEHLER` zurück. Folgende Währungen sind verfügbar:

BEF	Belgische Franc
DEM	Deutsche Mark
GRD	Griechische Drachmen
ESP	Spanische Peseten
FRF	Französische Franc
IEP	Irische Pfund
ITL	Italienische Lire
LUF	Luxemburgische Franc
NLG	Holländische Gulden
ATS	Österreichische Schilling
PTS	Portugiesischer Escudo
FIM	Finnmark

Weitere Informationen

[Runden und Abschneiden von Zahlen \[Seite 295\]](#)

1.6.1.8.7 InEuroRundungsfehler

Beschreibung

Gibt den Rundungsfehler bei der Umwandlung in Euro-Beträge zurück.

Funktionsgruppe

Numerisch

Syntax

```
Num InEuroRundungsfehler(BetragNichtEuro;Währungscode;Rundungsebene)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
BetragEuro	Betrag in der Ausgangswährung	Zahl	Ja
Währungscode	ISO-Code der Ausgangswährung	Zeichenfolge	Ja
Rund_ebene	Anzahl der Dezimalstellen, auf die das Ergebnis auf- bzw. abgerundet werden soll	Zahl	Ja

Beispiele

`InEuroRundungsfehler(6559;"FRF";2)` gibt 0 zurück. (Der nicht gerundete Umrechnungsbetrag und der auf zwei Dezimalstellen auf- bzw. abgerundete Umrechnungsbetrag sind gleich.)

`InEuroRundungsfehler(6559;"FRF";1)` gibt -0,01 zurück. (Der nicht gerundete Umrechnungsbetrag ist 999,91. Der auf eine Dezimalstelle gerundete Umrechnungsbetrag ist 999,90. Der Rundungsfehler beträgt -0,01.)

`InEuroRundungsfehler(1955;"DEM";2)` gibt 0 zurück. (Der nicht gerundete Umrechnungsbetrag und der auf zwei Dezimalstellen auf- bzw. abgerundete Umrechnungsbetrag sind gleich.)

`InEuroRundungsfehler(1955;"DEM";1)` gibt 0,02 zurück. (Der nicht gerundete Umrechnungsbetrag ist 999,58. Der auf eine Dezimalstelle gerundete Umrechnungsbetrag ist 999,60. Der Rundungsfehler beträgt 0,02.)

Hinweis

Der Währungscode muss der Code einer der 12 EU-Währungen sein, deren Werte im Verhältnis zum Euro vor der Abschaffung dieser Währungen im Januar 2002 festgelegt wurden. Wenn dies nicht der Fall ist, gibt die Funktion `#FEHLER` zurück. Folgende Währungen sind verfügbar:

BEF	Belgische Franc
DEM	Deutsche Mark
GRD	Griechische Drachmen
ESP	Spanische Peseten
FRF	Französische Franc
IEP	Irische Pfund
ITL	Italienische Lire
LUF	Luxemburgische Franc
NLG	Holländische Gulden
ATS	Österreichische Schilling
PTS	Portugiesischer Escudo
FIM	Finnmark

Weitere Informationen

[Runden und Abschneiden von Zahlen \[Seite 295\]](#)

1.6.1.8.8 Exp

Beschreibung

Gibt eine Exponentialzahl (in eine Potenz erhobene Zahl) zurück.

Funktionsgruppe

Numerisch

Syntax

```
Num Exp( Zahl )
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Zahl	Zu verwendende Potenz	Numerisch	Ja

Hinweise

Eine Exponentialzahl ist eine Konstante e (2,718...), die in eine Potenz erhoben ist.

Beispiele

`Exp(2 , 2)` gibt 9,03 zurück.

1.6.1.8.9 Fakultät

Beschreibung

Gibt die Fakultät einer Zahl zurück.

Funktionsgruppe

Numerisch

Syntax

```
Ganzz Fakultät(Zahl)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Zahl	Eine beliebige Zahl	Numerisch	Ja

Hinweise

Die Fakultät von `zahl` ist das Produkt aller Ganzzahlen von 1 bis `zahl`.

Beispiele

`Fakultät(4)` gibt 24 zurück.

`Fakultät(5,9)` gibt 120 zurück.

1.6.1.8.10 Untergrenze

Beschreibung

Gibt eine auf die nächste Ganzzahl abgerundete Zahl zurück.

Funktionsgruppe

Numerisch

Syntax

```
Ganzz Untergrenze(Zahl)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Zahl	Eine beliebige Zahl	Nummer	Ja

Beispiel

`Untergrenze(24,4)` gibt 24 zurück.

1.6.1.8.11 Ln

Beschreibung

Gibt den natürlichen Logarithmus einer Zahl zurück.

Funktionsgruppe

Numerisch

Syntax

```
Num Ln(Zahl)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Zahl	Eine beliebige Zahl	Zahl	Ja

Beispiele

`Ln(10)` gibt 2,3 zurück.

1.6.1.8.12 Log

Beschreibung

Gibt den Logarithmus einer Zahl für die angegebene Basis zurück.

Funktionsgruppe

Numerisch

Syntax

```
Num Log(Zahl;Basis)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Zahl	Eine beliebige Zahl	Numerisch	Ja
Basisrechte	Basis des Logarithmus	Numerisch	Ja

Beispiele

`Log(125;5)` gibt 3 zurück.

1.6.1.8.13 Log10

Beschreibung

Gibt den Logarithmus einer Zahl für die Basis 10 zurück.

Funktionsgruppe

Numerisch

Syntax

```
Num Log10(Zahl)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Zahl	Eine beliebige Zahl	Nummer	Ja

Beispiele

Log10(100) gibt 2 zurück.

1.6.1.8.14 Rest

Beschreibung

Gibt den Rest der Division zweier Zahlen zurück.

Funktionsgruppe

Numerisch

Syntax

```
Num Rest(Dividend;Divisor)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Dividend	Der Dividend (Zahl)	Numerisch	Ja
Divisor	Der Divisor (Zahl)	Numerisch	Ja

Beispiele

`Rest(10;4)` gibt 2 zurück.

`Rest(10,2;4,2)` gibt 1,8 zurück.

1.6.1.8.15 Potenz

Beschreibung

Gibt eine in eine Potenz erhobene Zahl zurück.

Funktionsgruppe

Numerisch

Syntax

```
Num Potenz(Zahl;Potenz)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Zahl	Zahl, die in eine Potenz erhoben werden soll	Numerisch	Ja
Zahl	Zu verwendende Potenz	Numerisch	Ja

Beispiel

Potenz(10;2) gibt 100 zurück.

1.6.1.8.16 Rang

Beschreibung

Ordnet eine Kennzahl auf der Grundlage von Dimensionen in einer Rangfolge ein.

Funktionsgruppe

Numerisch

Syntax

```
Ganzz Rang(Kennzahl;[Rangfolge_Dimensionen][;Erster|Letzter];  
(Dimensionen_zurücksetzen))
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Kennzahl	Kennzahl, für die die Rangfolge erstellt werden soll	Kennzahl	Ja
Rangfolge_Dimensionen	Dimensionen, auf deren Grundlage die Rangfolge für die Kennzahl erstellt werden soll	Dimensionsliste	Nein
Erster Letzter	Legt die Richtung der Rangfolge fest: <ul style="list-style-type: none">Erster – absteigendLetzter – aufsteigend	Schlüsselwort	Nein (Erster ist Standard)
Dimensionen_zurücksetzen	Zum Zurücksetzen der Rangfolge verwendete Dimensionen	Dimensionsliste	Nein

Hinweise

- Wenn Sie keine Rangfolgen-Dimensionen angeben, verwendet die Funktion den Standardberechnungskontext zum Berechnen der Rangfolge.
- Dimensionen müssen selbst dann immer in Klammern eingeschlossen werden, wenn die Liste der Rangfolge- oder Rücksetzungsdimensionen nur eine Dimension enthält.
- Beim Festlegen einer Reihe von Rangfolge- oder Rücksetzungsdimensionen müssen Sie diese durch Semikolons trennen.
- Bei einem Gruppenwechsel für eine Sektion oder einen Block wird die Rangfolge standardmäßig zurückgesetzt.

Beispiele

In der folgenden Tabelle wird der Rang anhand von `Rang([Umsatz];([Land]))` ermittelt:

Land	Umsatz	Rang
Frankreich	835.420	2
USA	2.451.104	1

In der folgenden Tabelle wird der Rang anhand von `Rang([Umsatz];([Land]);Letzter)` ermittelt. Durch das Argument `Letzter` wird festgelegt, dass die Kennzahlen in aufsteigender Rangfolge angeordnet werden.

Land	Umsatz	Rang
Frankreich	835.420	1
USA	2.451.104	2

In der folgenden Tabelle wird der Rang anhand von `Rang([Umsatz];([Land];[Erholungsort]))` ermittelt:

Land	Erholungsort	Umsatz	Rang
Frankreich	French Riviera	835.420	3
USA	Bahamas Beach	971.444	2
USA	Hawaiian Club	1.479.660	1

In der folgenden Tabelle wird der Rang anhand von `Rang([Umsatz];([Land];[Jahr]);([Land]))` ermittelt. Der Rang wird für die Dimension Land zurückgesetzt.

Land	Jahr	Umsatz	Rang
Frankreich	FJ1998	295.940	1
Frankreich	FJ1999	280.310	2
Frankreich	FJ2000	259.170	3
USA	FJ1998	767.614	3

USA	FJ1999	826.930	2
USA	FJ2000	856.560	1

Weitere Informationen

[Letzter/Erster \(Operatoren\) \[Seite 277\]](#)

1.6.1.8.17 Runden

Beschreibung

Rundet eine Zahl.

Funktionsgruppe

Numerisch

Syntax

```
Num Runden(Zahl;Rund_ebene)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Zahl	Die zu rundende Zahl	Numerisch	Ja
Rund_ebene	Anzahl der Dezimalstellen, auf die die Zahl auf- bzw. abgerundet werden soll	Numerisch	Ja

Beispiele

`Runden(9,44;1)` gibt 9,4 zurück.

Runden(9,45;1) gibt 9,5 zurück.

Runden(9,45;0) gibt 9 zurück.

Runden(9,45;-1) gibt 10 zurück.

Runden(4,45;-1) gibt 0 zurück.

Weitere Informationen

[Runden und Abschneiden von Zahlen \[Seite 295\]](#)

1.6.1.8.18 Zeichen

Beschreibung

Gibt das Vorzeichen einer Zahl zurück.

Funktionsgruppe

Numerisch

Syntax

```
Ganzz Zeichen(Zahl)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Zahl	Eine beliebige Zahl	Numerisch	Ja

Hinweise

Zeichen gibt -1 zurück, wenn Zahl negativ ist, 0, wenn Zahl Null ist und 1, wenn Zahl positiv ist.

Beispiele

`Zeichen(3)` gibt 1 zurück.

`Zeichen(-27,5)` gibt -1 zurück.

1.6.1.8.19 Sin

Beschreibung

Gibt den Sinus eines Winkels zurück.

Funktionsgruppe

Numerisch

Syntax

```
Num Sin(Winkel)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Winkel	Winkel im Bogenmaß	Zahl	Ja

Beispiel

`sin(234542)` kann, je nach Einstellung der Dezimalstellen, -0,116992 oder -0,12 zurückgeben.

1.6.1.8.20 Quadratwurzel

Beschreibung

Gibt die Quadratwurzel einer Zahl zurück.

Funktionsgruppe

Numerisch

Syntax

```
Num Quadratwurzel (Zahl)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Zahl	Eine beliebige Zahl	Nummer	Ja

Beispiel

`Quadratwurzel(25)` gibt 5 zurück.

1.6.1.8.21 Tan

Beschreibung

Gibt den Tangens eines Winkels zurück.

Funktionsgruppe

Numerisch

Syntax

```
Num Tan(Winkel)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Winkel	Winkel im Bogenmaß	Numerisch	Ja

Beispiele

Tan(90) gibt -2 zurück.

1.6.1.8.22 ToDecimal

Beschreibung

Gibt eine Dezimalzahl zurück.

Funktionsgruppe

Numerisch

Syntax

```
Num ToDecimal(Zahl|Zeichenfolge)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Zahl Zeichenfolge	Eine Zahl oder Zeichenfolge, die als Zahl interpretiert werden kann.	Zahl oder Zeichenfolge	Ja

Hinweise

Wenn `string` keine Zahl ist, gibt `ToDecimal` den Wert `#ERROR` zurück.

Beispiele

`ToDecimal("1234567890.1234567890")` gibt `1234567890.1234567890` zurück.

`ToDecimal("1234567890.12345")` gibt `1234567890.12345` zurück.

`ToDecimal("abcdefghijkl")` gibt `#ERROR` zurück.

1.6.1.8.23 InZahl

Beschreibung

Gibt eine Zeichenfolge als Zahl zurück.

Funktionsgruppe

Numerisch

Syntax

```
Num InZahl(Zeichenfolge)
```

oder

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Zeichenfolge	Zahl als Zeichenfolge	Zeichenfolge	Ja

Hinweise

Wenn es sich bei `zeichenfolge` nicht um eine Zahl oder einen DatumUhrzeit-Wert handelt, gibt die Funktion `InZahl` einen `#ERROR` zurück.

Beispiele

`InZahl ("45")` gibt 45 zurück.

1.6.1.8.24 Abschneiden

Beschreibung

Schneidet eine Zahl ab.

Funktionsgruppe

Numerisch

Syntax

```
Num Abschneiden(Zahl;Abschneid_Ebene)
```


Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Zahl	Die zu rundende Zahl	Numerisch	Ja
Abschneid_Ebene	Anzahl der Dezimalstellen, bis auf die die Zahl abgeschnitten werden soll	Numerisch	Ja

Hinweise

Beispiel

`Abschneiden(3,423;2)` gibt 3,42 zurück.

Weitere Informationen

[Runden und Abschneiden von Zahlen \[Seite 295\]](#)

1.6.1.9 Set-Funktionen

1.6.1.9.1 Vorgänger

Beschreibung

Gibt ein Vorgänger-Element eines Elements zurück

Funktionsgruppe

Menge

Syntax

```
Element Vorgänger(Element;Ebene|Entfernung)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Element	Ein beliebiges Element	Element	Ja
Ebene	Die Ebene des Vorgängers	Ebene	Es ist entweder die Ebene oder die Entfernung erforderlich
Entfernung	Der Abstand der Vorgängerebene von der aktuellen Ebene	Ganzzahl	Es ist entweder die Ebene oder die Entfernung erforderlich

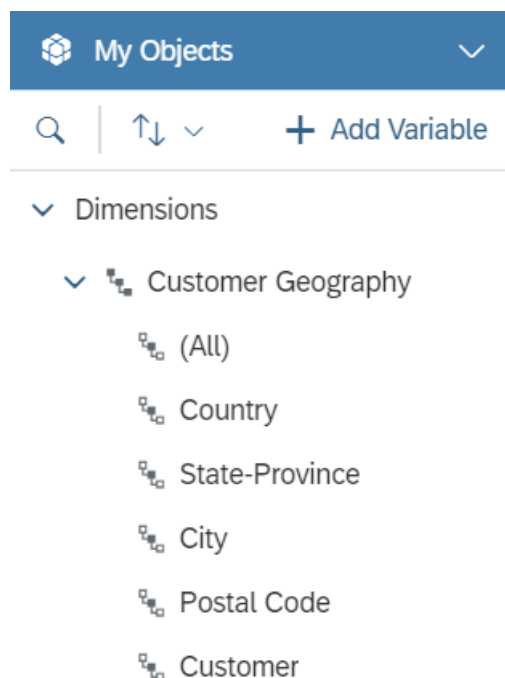
Hinweise

- `Vorgänger` wird nicht als eigenständige Funktion verwendet. Sie wird im Eingabeparameter von Aggregatfunktionen verwendet, der die Elementmenge für die Aggregation angibt.
- `Element` ist das aktuelle Element einer Hierarchie. Wenn die Hierarchie nicht im Kontext des Blocks enthalten ist, gibt die Formel einen leeren Wert zurück.
- Die `Entfernung` muss einen positiven Wert haben.

Beispiele

Die folgenden Beispiele stammen alle aus einer englischsprachigen Datenquelle.

Ausgehend von der folgenden geografischen Hierarchie möchten Sie die Auswirkungen des Internet-Umsatzbetrags jedes einzelnen Kunden, unabhängig von dessen Stadt, ermitteln.



Zuerst möchten Sie für jede Stadt den Internet-Umsatzbetrag für das zugehörige Land ermitteln:

```
=Sum([Query 2].[Internet Sales].[Internet Sales Amount];{Ancestor([Customer Geography];[Customer Geography].[City]))
```

Customer Geography	Sales Amount per State/Province	Internet Sales Amount
▼ All Customers		29,358,677.22
▼ Australia		9,061,000.58
▼ New South Wales	3,934,485.73	3,934,485.73
▼ Coffs Harbour	3,934,485.73	235,454.97
▼ 2450	3,934,485.73	235,454.97
Adriana Smith	3,934,485.73	5,333.25
Aimee Guo	3,934,485.73	77.27
Allison R. Young	3,934,485.73	39.98
Ann A. Sara	3,934,485.73	39.98
Antonio G. Patterson	3,934,485.73	8,068.03
Ariana Stewart	3,934,485.73	6,070.59
Arthur Kapoor	3,934,485.73	23.97
Barbara W. Lal	3,934,485.73	2,795.01
Bobby D. Saunders	3,934,485.73	120.48
Brianna J. Johnson	3,934,485.73	38.98

Dann berechnen Sie den Beitrag jeder Stadt zum gesamten Internet-Umsatzbetrag des Landes:

```
=([Query 2].[Internet Sales].[Internet Sales Amount] / Sum([Query 2].[Internet Sales].[Internet Sales Amount];{Ancestor([Customer Geography];[Customer Geography].[City]))
```

Customer Geography	City Contribution	Internet Sales Amount
▼ All Customers		29,358,677.22
▼ Australia		9,061,000.58
▼ New South Wales	100.00%	3,934,485.73
▼ Coffs Harbour	5.98%	235,454.97
▼ 2450	5.98%	235,454.97
Adriana Smith	0.14%	5,333.25
Aimee Guo	0.00%	77.27
Allison R. Young	0.00%	39.98
Ann A. Sara	0.00%	39.98
Antonio G. Patterson	0.21%	8,068.03
Ariana Stewart	0.15%	6,070.59
Arthur Kapoor	0.00%	23.97
Barbara W. Lal	0.07%	2,795.01
Bobby D. Saunders	0.00%	120.48
Brianna J. Johnson	0.00%	38.98

ⓘ Hinweis

Wenn Sie BICS-Verbindungen zu SAPBW-Providern verwenden, müssen Sie eine Offset-Ebene angeben anstatt die Ebene zu benennen:

```
=[Query 2].[Internet Sales].[Internet Sales Amount] / Sum([Query 2].[Internet Sales].[Internet Sales Amount];{Ancestor([Customer Geography];2)})
```

In diesem Fall erhalten Sie auch Ergebnisse für Bundesland und Land.

Weitere Informationen

[Aggregation \[Seite 45\]](#)

[Durchschnitt \[Seite 52\]](#)

[Anzahl \[Seite 53\]](#)

[Max \[Seite 59\]](#)

[Min \[Seite 62\]](#)

[Summe \[Seite 82\]](#)

1.6.1.9.2 Untergeordnete Elemente

Beschreibung

Gibt untergeordnete Elemente eines Hierarchieelements in einer Aggregationsfunktion zurück.

Funktionsgruppe

Menge

Syntax

```
Elementmenge Element.Untergeordnete Elemente
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Element	Ein beliebiges Element	Element	Ja

Hinweise

- `UntergeordneteElemente` wird nicht als eigenständige Funktion verwendet. Sie wird im Eingabeparameter von Aggregatfunktionen verwendet, der die Elementmenge für die Aggregation angibt.
- `Element` ist das aktuelle Element einer Hierarchie. Wenn die Hierarchie nicht im Kontext des Blocks enthalten ist, gibt die Formel einen leeren Wert zurück.

Beispiele

`[Geographie].[USA].[Kalifornien].UntergeordneteElemente` gibt `[Los Angeles]`, `[San Francisco]`, `[San Diego]` zurück.

`[Geographie].UntergeordneteElemente` gibt `[Los Angeles]`, `[San Francisco]`, `[San Diego]` zurück, wenn `[Kalifornien]` das aktuelle Element in der Hierarchie `[Geographie]` ist.

Weitere Informationen

[Aggregation \[Seite 45\]](#)

[Durchschnitt \[Seite 52\]](#)

[Anzahl \[Seite 53\]](#)

[Max \[Seite 59\]](#)

[Min \[Seite 62\]](#)

[Summe \[Seite 82\]](#)

1.6.1.9.3 Tiefe

Beschreibung

Gibt die Tiefe eines Elements in einer Hierarchie zurück

Funktionsgruppe

Menge

Syntax

```
int member.Depth
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Element	Ein beliebiges Element	Element	Ja

Hinweise

- Die Tiefe ist die Entfernung des Elements von der obersten Ebene der Hierarchie.
- Die oberste Hierarchieebene lautet 0.
- `Element` ist das aktuelle Element einer Hierarchie. Wenn die Hierarchie nicht im Kontext des Blocks enthalten ist, gibt die Formel einen leeren Wert zurück.

Beispiele

Sie möchten die Tiefe eines Hierarchieelements erfahren:

```
=[Calendar].[Date.Calendar].Depth
```

Date.Calendar	'=[Calendar].[Date.Calendar].Depth
[-] All Periods	0
[-] CY 2001	1
[-] H2 CY 2001	2
[-] Q3 CY 2001	3
[-] July 2001	4
July 1, 2001	5
July 2, 2001	5
July 3, 2001	5
July 4, 2001	5
July 5, 2001	5
July 6, 2001	5
July 7, 2001	5

Um diese Funktion mit den Funktionen von untergeordneten Elemente zu kombinieren, z. B. um zu überprüfen, ob alle Tage des Monats aufgeführt sind, tun Sie dies wie folgt:

```
=If [Calendar].[Date.Calendar].Depth = 4 Then Count([Internet Sales].[Internet Sales Amount];{[Calendar].[Date.Calendar].Children()})
```

Date.Calendar	Internet Sales Amount	[Date.Calendar].Depth	[Date.Calendar].Children()
[-] All Periods	29,358,677.22	0	
[-] CY 2001	3,266,373.66	1	
[-] H2 CY 2001	3,266,373.66	2	
[-] Q3 CY 2001	1,453,522.89	3	
[+] July 2001	473,388.16	4	31
[+] August 2001	506,191.69	4	30
[+] September 2001	473,943.03	4	29
[-] Q4 CY 2001	1,812,850.77	3	
[+] October 2001	513,329.47	4	30
[+] November 2001	543,993.41	4	30
[+] December 2001	755,527.89	4	31

1.6.1.9.4 Nachfolger

Beschreibung

Gibt Nachfolger eines Hierarchieelements in einer Aggregationsfunktion zurück.

Funktionsgruppe

Menge

Syntax

```
Elementmenge Nachfolger(Element[;Ebene|Entfernung][;NachfolgerKennz])
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Element	Ein beliebiges Element	Element	Ja

Parameter	Beschreibung	Typ	Erforderlich
Ebene	Die Ebene des Nachfolgers	Ebene	Nein (die Ebene des Elements ist der Standard)
Entfernung	Der Entfernung der Nachfolgerebene von der aktuellen Ebene	Ganzzahl	Nein (die Ebene des Elements ist der Standard)
NachfolgerKennz	Legt fest, welche Nachfolgerelemente zurückgegeben werden	Schlüsselwort	Nein (Standard ist Selbst)

Hinweise

- `Nachfolger` wird nicht als eigenständige Funktion verwendet. Sie wird im Eingabeparameter von Aggregatfunktionen verwendet, der die Elementmenge für die Aggregation angibt.
- `Element` ist das aktuelle Element einer Hierarchie. Wenn die Hierarchie nicht im Kontext des Blocks enthalten ist, gibt die Formel einen leeren Wert zurück.
- `Selbst` in `NachfolgerKennz` bezieht sich auf die von dem Parameter `Ebene|Entfernung` angegebene Ebene.
- `Vor` in `NachfolgerKennz` bezieht sich auf alle Ebenen oberhalb der von dem Parameter `Ebene|Entfernung` angegebenen Ebene.
- `Nach` in `NachfolgerKennz` bezieht sich auf alle Ebenen unterhalb der von dem Parameter `Ebene|Entfernung` angegebenen Ebene.
- `NachfolgerKennz` hat die folgenden Werte:

Selbst	Die Nachfolger auf der vom Parameter <code>Ebene Entfernung</code> angegebenen Ebene, auch das aktuelle Element, sofern es sich auf dieser Ebene befindet, werden zurückgegeben.
Vor	Das aktuelle Element und alle Nachfolger oberhalb der vom Parameter <code>Ebene Entfernung</code> angegebenen Ebene werden zurückgegeben.
Nach	Die Nachfolger unterhalb der vom Parameter <code>Ebene Entfernung</code> angegebenen Ebene werden zurückgegeben.
Self_Before	Das aktuelle Element und alle Nachfolger oberhalb und einschließlich der vom Parameter <code>Ebene Entfernung</code> angegebenen Ebene werden zurückgegeben.
Self_After	Das aktuelle Element und alle Nachfolger auf und unterhalb der vom Parameter <code>Ebene Entfernung</code> angegebenen Ebene werden zurückgegeben.
Before_After	Das aktuelle Element und alle Nachfolger mit Ausnahme der Nachfolger auf der vom Parameter

	Ebene Entfernung angegebenen Ebene werden zurückgegeben.
Self_Before_After	Das aktuelle Element und alle Nachfolger werden zurückgegeben.
Blätter	Alle Elemente zwischen dem aktuellen Element und der vom Parameter Ebene Entfernung angegebenen Ebene, die keine untergeordneten Elemente haben.

- Die Entfernung muss einen positiven Wert haben.

Beispiel

Sie gehen von einer Finanzhierarchie aus, bei der einige Knoten nicht unbedingt kumulativ sind, Sie aber deren Nachfolger ermitteln möchten. In diesem Beispiel erhalten Sie die Summe der Nachfolger jedes Bilanzelements, nur 1 Ebene darunter:

```
=Sum([Query 3 (1)].[Financial Reporting].[Amount];
{Descendants([Accounts]&[Balance Sheet];1)})
```

Accounts			
<input checked="" type="checkbox"/> Balance Sheet	0		27,481,462
<input checked="" type="checkbox"/> Assets	13,740,731		
<input checked="" type="checkbox"/> Liabilities and Owners Equity	13,740,731		
<input checked="" type="checkbox"/> Net Income	12,609,503		

```
=Sum([Query 3 (1)].[Financial Reporting].[Amount];
{Descendants([Accounts]&[Balance Sheet].[Assets].[Current Assets];1;Leaves)})
```

[-] Balance Sheet	0	12,445,628
[-] Assets	13,740,731	
[-] Current Assets	12,445,628	
Cash	3,236,799	
[-] Receivables	3,475,923	
Trade Receivables	3,371,580	
Other Receivables	104,343	
Allowance for Bad Debt	67,429	
[-] Inventory	4,143,398	
Raw Materials	2,007,586	
Work in Process	1,393,582	
Finished Goods	742,230	
Deferred Taxes	505,424	
Prepaid Expenses	341,992	
Intercompany Receivable	674,663	

Jetzt möchten Sie die Summe aller Elemente unter „Gesamtumlaufvermögen“ berechnen:

```
=Sum([Query 3 (1)].[Financial Reporting].[Amount];
{Descendants([Accounts]&[Balance Sheet].[Assets].[Current Assets];0;After)})
```

[-] Balance Sheet	0	20,064,949
[-] Assets	13,740,731	
[-] Current Assets	12,445,628	
Cash	3,236,799	
[-] Receivables	3,475,923	
Trade Receivables	3,371,580	
Other Receivables	104,343	
Allowance for Bad Debt	67,429	
[-] Inventory	4,143,398	
Raw Materials	2,007,586	
Work in Process	1,393,582	
Finished Goods	742,230	
Deferred Taxes	505,424	
Prepaid Expenses	341,992	
Intercompany Receivable	674,663	

Jetzt addieren Sie das „Gesamtumlaufvermögen“ selbst:

```
=Sum([Query 3 (1)].[Financial Reporting].[Amount];
{Descendants([Accounts]&[Balance Sheet].[Assets].[Current Assets];0;Self_After)})
```

[-] Balance Sheet	0		32,510,577
[-] Assets	13,740,731		
[-] Current Assets	12,445,628		
Cash	3,236,799		
[-] Receivables	3,475,923		
Trade Receivables	3,371,580		
Other Receivables	104,343		
Allowance for Bad Debt	67,429		
[-] Inventory	4,143,398		
Raw Materials	2,007,586		
Work in Process	1,393,582		
Finished Goods	742,230		
Deferred Taxes	505,424		
Prepaid Expenses	341,992		
Intercompany Receivable	674,663		

Weitere Informationen

[Aggregation \[Seite 45\]](#)

[Durchschnitt \[Seite 52\]](#)

[Anzahl \[Seite 53\]](#)

[Max \[Seite 59\]](#)

[Min \[Seite 62\]](#)

[Summe \[Seite 82\]](#)

1.6.1.9.5 IstBlattelement

Beschreibung

Festlegen, ob ein Element ein Blattelement ist.

Funktionsgruppe

Sonstige

Syntax

```
bool member.IsLeaf
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Element	Ein beliebiges Element	Element	Ja

Hinweise

- Ein Blattelement ist ein Element, das keine untergeordneten Elemente hat.
- `Element` ist das aktuelle Element einer Hierarchie. Wenn die Hierarchie nicht im Kontext des Blocks enthalten ist, gibt die Formel einen leeren Wert zurück.

Beispiele

Sie möchten wissen, ob es sich bei der Zeile um einen Tag handelt:

```
=[Calendar].[Date.Calendar].IsLeaf()
```

Date.Calendar	'=[Query 1].[Calendar].[Date.Calendar].IsLeaf
[-] All Periods	0
[-] CY 2001	0
[-] H2 CY 2001	0
[-] Q3 CY 2001	0
[-] July 2001	0
July 1, 2001	1
July 2, 2001	1
July 3, 2001	1
July 4, 2001	1
July 5, 2001	1
July 6, 2001	1
July 7, 2001	1
July 8, 2001	1

1.6.1.9.6 Schlüssel

Beschreibung

Zurückgeben des Schlüssels eines Elements

Syntax

`string member.Key`

Funktionsgruppe

Menge

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Element	Ein beliebiges Element	Element	Ja

Hinweise

- Der Schlüssel ist die interne Kennung eines Elements.
- `Element` ist das aktuelle Element einer Hierarchie. Wenn die Hierarchie nicht im Kontext des Blocks enthalten ist, gibt die Formel einen leeren Wert zurück.

Beispiel

`[Geografie].[USA].Schlüssel` gibt "XYZ" zurück, wenn der Schlüssel des Elements [USA] "XYZ" ist.

1.6.1.9.7 Abstand

Beschreibung

Gibt ein Element auf derselben Ebene wie das aktuelle Element und in einer bestimmten Entfernung danach innerhalb einer aggregierten Funktion zurück

Syntax

```
member member.Lag(distance)
```

Funktionsgruppe

Menge

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Element	Ein beliebiges Element	Element	Ja
Entfernung	Der Entfernung des Elements vom aktuellen Element	int	Ja

Hinweise

- Abstand wird nicht als eigenständige Funktion verwendet. Sie wird im Eingabeparameter von Aggregatfunktionen verwendet, der die Elementmenge für die Aggregation angibt.
- Wenn distance positiv ist, gibt Lag das Element distance Stellen nach member zurück. Wenn distance negativ ist, gibt Lag das Element distance Stellen vor member zurück.
- Element ist das aktuelle Element einer Hierarchie. Wenn die Hierarchie nicht im Kontext des Blocks enthalten ist, gibt die Formel einen leeren Wert zurück.
- Abstand verwendet die Elementreihenfolge in der Hierarchie und der Abfrage, um das zugehörige Element zurückzugeben.

Beispiele

Sie möchten die Änderungen im Verkauf über das Internet auf wöchentlicher Basis erhalten.

```
=Max([Internet Sales].[Internet Sales Amount];{[Calendar].[Date.Calendar].Lag(7)})
```

Date.Calendar	Internet Sales Amount	=Max([Query 1].[Internet Sales].[Internet Sales Amount];[Query 1].[Calendar].[Date.Calendar].Lag(7))
All Periods	29,358,677.22	
CY 2001	3,266,373.66	
H2 CY 2001	3,266,373.66	
Q3 CY 2001	1,453,522.89	1,623,971.06
July 2001	473,388.16	550,816.69
July 1, 2001	14,477.34	7,855.64
July 2, 2001	13,931.52	20,909.78
July 3, 2001	15,012.18	10,556.53
July 4, 2001	7,156.54	14,313.08
July 5, 2001	15,012.18	14,134.8
July 6, 2001	14,313.08	7,156.54
July 7, 2001	7,855.64	25,047.89
July 8, 2001	7,855.64	11,230.63
July 9, 2001	20,909.78	14,313.08
July 10, 2001	10,556.53	14,134.8

Oder Sie möchten ein bestimmtes Jahr mit dem vorletzten Jahr vergleichen:

Date.Calendar	Internet Sales Amount	
[-] All Periods	29,358,677.22	
[-] CY 2001	3,266,373.66	11.13%
[+] H2 CY 2001	3,266,373.66	100.00%
[-] CY 2002	6,530,343.53	22.24%
[+] H1 CY 2002	3,805,710.59	58.28%
[+] H2 CY 2002	2,724,632.94	41.72%
[-] CY 2003	9,791,060.3	33.35%
[+] H1 CY 2003	3,037,501.36	31.02%
[+] H2 CY 2003	6,753,558.94	68.98%
[+] CY 2004	9,770,899.74	33.28%

CY 2002	CY 2002.Lag(2)	CY 2002 - CY 2002.Lag(2)
6,530,343.53	9,770,899.74	-3,240,556.21

Nun möchten Sie "Abstand" mit "IstBlattelement" kombinieren, um die Unterschiede im Verkauf für einen Zeitraum von einer Woche zu erfahren. Hierzu muss die Formel in der letzten Spalte wie folgt aussehen:

```
=If [Calendar].[Date.Calendar].IsLeaf() Then [Internet Sales].[Internet Sales Amount] - Max([Internet Sales].[Internet Sales Amount];{[Calendar].[Date.Calendar].Lag(7)})
```

Date.Calendar	Internet Sales Amount	'[Calendar].[Date.Calendar].Lag(7))	Difference week to week
[-] All Periods	29,358,677.22		
[-] CY 2001	3,266,373.66		
[-] H2 CY 2001	3,266,373.66		
[-] Q3 CY 2001	1,453,522.89	1,623,971.06	
[-] July 2001	473,388.16	550,816.69	
July 1, 2001	14,477.34	7,855.64	6,621.7
July 2, 2001	13,931.52	20,909.78	-6,978.26
July 3, 2001	15,012.18	10,556.53	4,455.65
July 4, 2001	7,156.54	14,313.08	-7,156.54
July 5, 2001	15,012.18	14,134.8	877.38
July 6, 2001	14,313.08	7,156.54	7,156.54
July 7, 2001	7,855.64	25,047.89	-17,192.25
July 8, 2001	7,855.64	11,230.63	-3,374.99
July 9, 2001	20,909.78	14,313.08	6,596.7
July 10, 2001	10,556.53	14,134.8	-3,578.27
July 11, 2001	14,313.08	6,953.26	7,359.82

Weitere Informationen

[Aggregation \[Seite 45\]](#)

[Durchschnitt \[Seite 52\]](#)

[Anzahl \[Seite 53\]](#)

[Max \[Seite 59\]](#)

[Min \[Seite 62\]](#)

[Summe \[Seite 82\]](#)

1.6.1.9.8 MemberAtDepth

Beschreibung

Gibt die Elemente einer Hierarchie bis zu einer ausgewählten Tiefe zurück.

Funktionsgruppe

Menge

Syntax

```
string MemberAtDepth(hierarchy;depth)
```

Eingabe




Parameter	Beschreibung	Typ	Erforderlich
hierarchy	Hierarchisches Objekt	Dimension	Ja
depth	Die Ebene der Elementmenge in der ausgewählten Hierarchie	int	Ja

Hinweise

- `depth` muss ein positiver Wert oder Null sein, andernfalls gibt die Funktion einen Fehler zurück.
- Wenn bei der angegebenen Tiefe kein Element vorhanden ist, gibt `MemberAtDepth()` den Wert `Null` zurück.
- Wenn das Eingabeobjekt keine Hierarchie ist, gibt `MemberAtDepth()` für Tiefen über Null den Wert `Null` zurück.

Beispiele

Die folgende Hierarchie [Country] wurde dahingehend gefiltert, dass nur zwei untergeordnete Elemente des Knotens EUROPA übrig bleiben.

Country	Quantity
 WORLD	262,461
 EUROPE	31,009
France	6,965
Germany	6,331
 NORTH_AMERICA	219,944
Canada	17,754
USA	202,190
ASIA_PAC	9,065

MemberAtDepth([Country];0) gibt Folgendes zurück:

	Order Quantity
WORLD	755,719

"WELT" ist der Stamm und der einzige Wert auf der Ebene 0.

Wenn die Option *Duplikatgruppierung vermeiden* deaktiviert ist, aggregiert die Funktion die Werte aller Elemente, da sie alle vom Stamm "WELT" abstammen. Wenn Sie alle aggregierten Werte prüfen möchten, aktivieren Sie die Option *Duplikatgruppierung vermeiden*. Die Hierarchie würde dann folgendermaßen aussehen:

	Order Quantity
WORLD	262,461
WORLD	31,009
WORLD	6,965
WORLD	6,331
WORLD	219,944
WORLD	17,754
WORLD	202,190
WORLD	9,065
Sum:	755,719

MemberAtDepth([Country];1) gibt Folgendes zurück:

	Order Quantity
	262,461
ASIA_PAC	9,065
EUROPE	44,305
NORTH_AME	439,888

Die erste Zeile enthält einen NULL-Wert, da der Stamm keine Werte auf Ebene 1 enthält.

In anderen Zeilen aggregiert die Funktion die Werte aller Elemente der Hierarchie [Country] auf der Ebene 1: "ASIEN_PAZIFIK", "EUROPA" und "NORDAMERIKA". Wenn Sie alle aggregierten Werte prüfen

möchten, aktivieren Sie die Option *Duplikatgruppierung vermeiden*. Die Hierarchie würde dann folgendermaßen aussehen:

	Order Quantity
	262,461
ASIA_PAC	9,065
EUROPE	31,009
EUROPE	6,965
EUROPE	6,331
NORTH_AME	219,944
NORTH_AME	17,754
NORTH_AME	202,190
Sum:	755,719

MemberAtDepth([Country];2) gibt Folgendes zurück:

	Order Quantity
	522,479
Canada	17,754
France	6,965
Germany	6,331
USA	202,190

Die erste Zeile aggregiert wiederum all diejenigen Elemente, die keinen Wert auf der Ebene 2 aufweisen, d. h. den Stamm und jeden Knoten. In anderen Zeilen werden die Werte aller Elemente der Hierarchie [Country] auf der Ebene 2 aggregiert:

MemberAtDepth([Country];3) gibt Folgendes zurück:

	Order Quantity
	755,719

Es bleibt nur eine Zeile mit einem NULL-Wert übrig, da die Hierarchie keine dritte Ebene aufweist. Deshalb werden alle Knoten und die Werte aller Blätter der Hierarchie aggregiert.

Wenn Sie die gesamte Hierarchie visualisieren möchten, fügen Sie Spalten hinzu, die die Ebenen der Hierarchie der vorhandenen Tabelle enthalten, und verwenden Sie dann die Funktion, um den Ebenen der Hierarchie unterschiedliche Werte zu geben. Sie würde dann folgendermaßen aussehen:

Level 0	Level 1	Level 2	Order Quantity
WORLD			262,461
WORLD	ASIA_PAC		9,065
WORLD	EUROPE		31,009
WORLD	EUROPE	France	6,965
WORLD	EUROPE	Germany	6,331
WORLD	NORTH_AMERICA		219,944
WORLD	NORTH_AMERICA	Canada	17,754
WORLD	NORTH_AMERICA	USA	202,190

Sie können die Formel `IsLeaf` verwenden, um die Blätter der Hierarchie in der oben stehenden Tabelle zu filtern: Nähere Informationen zur Formel `IsLeaf` finden Sie unter [IstBlattelement \[Seite 220\]](#).

Level 0	Level 1	Level 2	Order Quantity	IsLeaf
WORLD			262,461	false
WORLD	ASIA_PAC		9,065	true
WORLD	EUROPE		31,009	false
WORLD	EUROPE	France	6,965	true
WORLD	EUROPE	Germany	6,331	true
WORLD	NORTH_AMERICA		219,944	false
WORLD	NORTH_AMERICA	Canada	17,754	true
WORLD	NORTH_AMERICA	USA	202,190	true

Anschließend können Sie die Spalte `IsLeaf` ausblenden, um das Äquivalent einer geglätteten Hierarchietabelle zu erhalten:

Level 0	Level 1	Level 2	Order Quantity
WORLD	ASIA_PAC		9,065
WORLD	EUROPE	France	6,965
WORLD	EUROPE	Germany	6,331
WORLD	NORTH_AMERICA	Canada	17,754
WORLD	NORTH_AMERICA	USA	202,190

1.6.1.9.9 Übergeordnetes Element

Beschreibung

Gibt das übergeordnete Element eines Hierarchieelements in einer Aggregationsfunktion zurück.

Funktionsgruppe

Menge

Syntax

```
Element Element.ÜbergeordnetesElement
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Element	Ein beliebiges Element	Element	Ja

Hinweise

- `Übergeordnetes Element` wird nicht als eigenständige Funktion verwendet. Sie wird im Eingabeparameter von Aggregatfunktionen verwendet, der die Elementmenge für die Aggregation angibt.
- `Element` ist das aktuelle Element einer Hierarchie. Wenn die Hierarchie nicht im Kontext des Blocks enthalten ist, gibt die Formel einen leeren Wert zurück.

Beispiele

Die zweite Spalte enthält die Formel, mit der Sie das übergeordnete Element für jedes Hierarchieelement ermitteln können:

```
=Max([Customer Geography];{[Customer Geography].Parent})
```

[-] All Customers	
[-] Australia	All Customer:
[-] New South Wales	Australia
[+] Alexandria	New South W
[-] Coffs Harbour	New South W
[-] 2450	Coffs Harbou
Adriana Smith	2450
Aimee Guo	2450
Allison R. Young	2450
Ann A. Sara	2450

Weitere Informationen

[Aggregation \[Seite 45\]](#)

[Durchschnitt \[Seite 52\]](#)

[Anzahl \[Seite 53\]](#)

[Max \[Seite 59\]](#)

[Min \[Seite 62\]](#)

[Summe \[Seite 82\]](#)

1.6.1.9.10 GleichgeordneteElemente

Beschreibung

Gibt das Element und die gleichgeordneten Elemente eines Hierarchieelements in einer Aggregationsfunktion zurück.

Funktionsgruppe

Menge

Syntax

```
Elementmenge Element.GleichgeordneteElemente
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Element	Ein beliebiges Element	Element	Ja

Hinweise

- `GleichgeordneteElemente` wird nicht als eigenständige Funktion verwendet. Sie wird im Eingabeparameter von Aggregatfunktionen verwendet, der die Elementmenge für die Aggregation angibt.
- `Element` ist das aktuelle Element einer Hierarchie. Wenn die Hierarchie nicht im Kontext des Blocks enthalten ist, gibt die Formel einen leeren Wert zurück.
- Gleichgeordnete Elemente sind Elemente, die sich auf der gleichen Ebene wie das `Element` befinden und das gleiche übergeordnete Element haben.

Beispiele

Sie haben eine Zeithierarchie und möchten den prozentualen Anteil für jedes Quartal eines Jahres oder den prozentualen Anteil für das Jahr innerhalb des Zeitraums ermitteln.

```
=[Query 1].[Internet Sales].[Internet Sales Amount] / Sum([Query 1].[Internet Sales].[Internet Sales Amount];{[Query 1].[Calendar].[Date.Calendar].Siblings()})
```

Date.Calendar	Internet Sales Amount	
[-] All Periods	29,358,677.22	
[-] CY 2001	3,266,373.66	11.13%
[+] H2 CY 2001	3,266,373.66	100.00%
[-] CY 2002	6,530,343.53	22.24%
[+] H1 CY 2002	3,805,710.59	58.28%
[+] H2 CY 2002	2,724,632.94	41.72%
[-] CY 2003	9,791,060.3	33.35%
[+] H1 CY 2003	3,037,501.36	31.02%
[+] H2 CY 2003	6,753,558.94	68.98%
[+] CY 2004	9,770,899.74	33.28%

In einer individuellen Zelle möchten Sie den Beitrag des Jahres 2004 in dem gesamten Zeitraum ermitteln:

```
=Sum([Query 1].[Internet Sales].[Internet Sales Amount];{[Query 1].[Calendar].[Date.Calendar]&[All Periods].[CY 2004]}) / Sum([Query 1].[Internet Sales].[Internet Sales Amount];{[Query 1].[Calendar].[Date.Calendar]&[All Periods].[CY 2004].Siblings()})
```

{CY 2001;CY 2002}	2004 percentage in 2001 to 2004 period
9,796,717.18	33.28%

Date.Calendar	Internet Sales Amount	
[-] All Periods	29,358,677.22	
[-] CY 2001	3,266,373.66	11.13%
[+] H2 CY 2001	3,266,373.66	100.00%
[-] CY 2002	6,530,343.53	22.24%
[+] H1 CY 2002	3,805,710.59	58.28%
[+] H2 CY 2002	2,724,632.94	41.72%
[-] CY 2003	9,791,060.3	33.35%
[+] H1 CY 2003	3,037,501.36	31.02%
[+] H2 CY 2003	6,753,558.94	68.98%
[+] CY 2004	9,770,899.74	33.28%

Weitere Informationen

[Aggregation \[Seite 45\]](#)

[Durchschnitt \[Seite 52\]](#)

[Anzahl \[Seite 53\]](#)

[Max \[Seite 59\]](#)

[Min \[Seite 62\]](#)

[Summe \[Seite 82\]](#)

1.6.1.10 Sonstige Funktionen

1.6.1.10.1 Blockname

Beschreibung

Gibt den Namen des Blocks zurück.

Funktionsgruppe

Sonstige

Syntax

```
Zeichenfolge Blockname( )
```

Beispiele

`Blockname()` gibt "Block1" zurück, wenn sich die Funktion in einem Block namens "Block1" befindet.

1.6.1.10.2 Abschlussperiode

Beschreibung

Gibt die Kennzahl am letzten Datum der Periode im aktuellen Kontext und in dem in der Zeitdimension definierten Zeitbereich zurück.

Funktionsgruppe

Sonstige

Syntax

```
Eingabetyp Abschlussperiode(Kennzahl;Zeitraum)
```

Eingabe

Parameter	Beschreibung	Art	Erforderlich
Kennzahl	Beliebige Kennzahl oder Variable.	Kennzahl	Ja
Zeitraum	Der Zeitraum, der den Berechnungskontext im Block bereitstellt.	Zeitraum	Ja

ⓘ Hinweis

- Das Zeitobjekt muss ein im Block verfügbarer Zeitraum sein. Wenn der Block den Zeitraum nicht enthält, gibt die Funktion den Fehler #BERECHNUNG zurück.
- Die Berichtsfiler für die Zeitdimension können sich auf die Funktionsergebnisse auswirken. Sie können die Funktion mit der Funktion `NoFilter` kombinieren, um Berichtsfiler in der Funktionsauswertung zu ignorieren.

Beispiele

Die Spalte „Abschlussperiode“ in der Tabelle unten enthält folgende Formel:

```
ClosingPeriod([Revenue]; [Time Dimension].[Year])
```

Jahr	Umsatz	Abschlussperiode
2016	1000	2000
2017	2000	2000

```
ClosingPeriod([Revenue]; [Time Dimension].[Semester])
```

Jahr	Semester	Umsatz	Abschlussperiode
2016	H1 2016	400	1500
2016	H2 2016	600	1500
2017	H1 2017	500	1500
2017	H2 2017	1500	1500

Weitere Informationen

[#BERECHNUNG \[Seite 311\]](#)

1.6.1.10.3 SpalteNummer

Beschreibung

Gibt die Nummer der Spalte zurück.

Funktionsgruppe

Sonstige

Syntax

```
Ganzz SpalteNummer( )
```

Beispiele

`SpalteNummer()` gibt 2 zurück, wenn sich die Formel in der zweiten Spalte einer Tabelle befindet.

1.6.1.10.4 Kommentar

Beschreibung

Gibt den Kommentar einer Zelle zurück

Funktionsgruppe

Sonstige

Syntax

```
string Comment ( )
```

Hinweis

Abhängig davon, wie Sie den Parameter in den *Dokumenteigenschaften* eingestellt haben, handelt es sich bei dem durch die Funktion zurückgegebenen Kommentar entweder um den ersten oder den letzten Kommentar in der Zelle.

Beispiel

Comment () gibt "Bruttomarge in Q3 steigern" zurück, wenn der Kommentar in der Zelle "Bruttomarge in Q3 steigern" lautet.

1.6.1.10.5 AktuellerBenutzer

Beschreibung

Zurückgeben des BI-Launchpad-Anmeldenamens des aktuellen Benutzers

Funktionsgruppe

Sonstige

Syntax

```
Zeichenfolge AktuellerBenutzer ( )
```

Beispiele

AktuellerBenutzer () gibt "gkn" zurück, wenn der Anmeldename des aktuellen Benutzers "gkn" ist.

1.6.1.10.6 BenutzerdefinierteEigenschaften

Beschreibung

Gibt die Liste der definierten benutzerdefinierten Eigenschaften zurück

Funktionsgruppe

Sonstige

Syntax

```
Zeichenfolge BenutzerdefinierteEigenschaften ( )
```

ⓘ Hinweis

- Wenn im Dokument keine benutzerdefinierte Eigenschaft vorhanden ist, gibt die Funktion die leere Zeichenfolge zurück.
- Wenn mehrere benutzerdefinierte Eigenschaften im Dokument vorhanden sind, werden sie in der von der Funktion zurückgegebenen Zeichenfolge durch Semikolons getrennt.

Beispiele

Wenn das Dokument zwei benutzerdefinierte Eigenschaften mit der Bezeichnung „Lines“ und „Category“ enthält,

gibt `CustomProperties()` den Wert `"Lines;Category"` zurück

1.6.1.10.7 BenutzerdefinierteEigenschaftWert

Beschreibung

Gibt den Wert der definierten benutzerdefinierten Eigenschaft zurück.

Funktionsgruppe

Sonstige

Syntax

```
Zeichenfolge BenutzerdefinierteEigenschaftWert (benutzerdefinierte_Eigenschaft)
```

Eingabe

Parameter	Beschreibung	Art	Erforderlich
custom_property	Name einer benutzerdefinieren Eigenschaft	Zeichenfolge	Ja

Hinweis

- Wenn die benutzerdefinierte Eigenschaft nicht vorhanden ist, gibt die Funktion die leere Zeichenfolge zurück.

Beispiele

Wenn das Dokument zwei benutzerdefinierte Eigenschaften enthält: „Lines“ mit dem Wert „Dresses“ und „Category“ mit dem Wert „Skirts“, dann gilt Folgendes:

- `CustomPropertyValue("Lines")` gibt "Dresses" zurück
- `CustomPropertyValue("Category")` gibt "Skirts" zurück
- `CustomPropertyValue("Color")` gibt " " zurück

1.6.1.10.8 DescriptionOf

Beschreibung

Gibt die Beschreibung eines Objekts zurück.

Funktionsgruppe

Sonstige

Syntax

```
string DescriptionOf(obj)
```

Eingabe

Parameter	Beschreibung	Art	Erforderlich
Obj	Ein Berichtsobjekt	Berichtsobjekt	Ja

Hinweise

- Wenn keine Beschreibung festgelegt wurde, wird eine leere Zeichenfolge zurückgegeben.
- Die Beschreibung wird in Ihrem bevorzugten Anzeigegebietsschema zurückgegeben. Wenn die Beschreibung nicht in Ihrem bevorzugten Anzeigegebietsschema definiert wurde, wird die Beschreibung ggf. im Fallback-Gebietsschema zurückgegeben.

1.6.1.10.9 ZusammenführungErzwingen

Beschreibung

Berücksichtigt synchronisierte Dimensionen bei Kennzahlberechnungen, wenn die Dimensionen nicht im Berechnungskontext der Kennzahl enthalten sind.

Funktionsgruppe

Sonstige

Syntax

```
num ZusammenführungErzwingen ( Kennzahl )
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Kennzahl	Beliebige Kennzahl	Kennzahl	Ja

Ausgabe

Das Ergebnis der Berechnung unter Berücksichtigung der synchronisierten Dimensionen

Hinweise

- `ZusammenführungErzwingen` gibt bei Anwendung auf eine intelligente Kennzahl #MEHRFACHWERT zurück, da die zum Berechnen der intelligenten Kennzahl erforderliche Gruppierung nicht vorhanden ist.
- `ZusammenführungErzwingen` ist die Entsprechung der BusinessObjects/Desktop Intelligence-Funktion `MultiCube`.

Beispiele

`ZusammenführungErzwingen ([Umsatz])` gibt den Wert von [Umsatz] unter Berücksichtigung aller synchronisierten Dimensionen zurück, die nicht im selben Block wie die Kennzahl [Umsatz] angezeigt werden.

1.6.1.10.10 FormulaOf

Beschreibung

Gibt die Formel zurück, die eine Variable definiert, wenn das Objekt eine Variable ist. Wenn das Objekt keine Variable ist, wird eine leere Zeichenfolge zurückgegeben.

Funktionsgruppe

Sonstige

Syntax

```
string FormulaOf(obj)
```

Eingabe

Parameter	Beschreibung	Art	Erforderlich
Obj	Ein Berichtsobjekt	Berichtsobjekt	Ja

Hinweise

- Die Formel wird in Ihrem bevorzugten Anzeigegebietsschema zurückgegeben.

1.6.1.10.11 InhaltsgebietsschemaAbrufen

Beschreibung

Gibt das Gebietsschema der im Dokument enthaltenen Daten zurück (das Dokumentgebietsschema)

Funktionsgruppe

Sonstige

Syntax

```
Zeichenfolge InhaltsgebietsschemaAbrufen()
```

Hinweise

Das Dokumentgebietsschema wird zum Formatieren von Daten in einem Dokument verwendet.

Beispiele

`InhaltsgebietsschemaAbrufen` gibt "fr_FR" zurück, wenn das Dokumentgebietsschema "Französisch (Frankreich)" ist.

1.6.1.10.12 DominantesBevorzugtesAnzeigegebietsschemaAbrufen

Beschreibung

Gibt das dominante Gebietsschema in der Gruppe "Bevorzugtes Anzeigegebietsschema" des Benutzers zurück

Funktionsgruppe

Sonstige

Syntax

```
Zeichenfolge DominantesBevorzugtesAnzeigegebietsschemaAbrufen( )
```

Hinweise

- Jede Gruppe zugehöriger Gebietsschemas besitzt ein dominantes Gebietsschema, das als Basis für alle anderen Gebietsschemas in der Gruppe verwendet wird. So ist beispielsweise US-Englisch ("en_US") das dominante Gebietsschema in der Gruppe der englischen Gebietsschemas. Neuseeländisches Englisch ("en_NZ") gehört ebenfalls zu dieser Gruppe.
- Im Benutzerhandbuch *Übersetzungsmanager* sind alle dominanten bevorzugten Anzeigegebietsschemas aufgeführt.

Beispiele

`DominantesBevorzugtesAnzeigegebietsschemaAbrufen` gibt "en_US" zurück, wenn das bevorzugte Anzeigegebietsschema "Englisch (Neuseeland)" ist.

Weitere Informationen

[BevorzugtesAnzeigegebietsschemaAbrufen \[Seite 248\]](#)

1.6.1.10.13 GebietsschemaAbrufen

Beschreibung

Gibt das Gebietsschema des Benutzers zum Formatieren der Oberfläche (das Produktgebietsschema) zurück

Funktionsgruppe

Sonstige

Syntax

```
Zeichenfolge GebietsschemaAbrufen()
```

Hinweise

Das Produktgebietsschema ist das Gebietsschema der Benutzeroberfläche (z.B. Menüeinträge und Schaltflächentext).

Beispiele

`GebietsschemaAbrufen` gibt "en_US" zurück, wenn das Produktgebietsschema des Benutzers "Englisch (US)" ist.

1.6.1.10.14 LokalisierteAbrufen

Beschreibung

Gibt eine Zeichenfolge zurück, die gemäß dem bevorzugten Anzeigegebietsschema des Benutzers lokalisiert wurde

Syntax

```
Zeichenfolge LokalisierteAbrufen(Zeichenfolge[;Kommentar])
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Zeichenfolge	Die zu übersetzende Zeichenfolge	Zeichenfolge	Ja
Kommentar	Ein Kommentar für Übersetzer	Zeichenfolge	Nein

Hinweise

- Der Parameter `zeichenfolge` kann eine Zeichenfolge in einer beliebigen Formel sein (z.B. in einer Zelle, einer Alerter-Meldung oder einer Variablendefinition).
- Beim Erstellen eines Berichts können Sie den Parameter `kommentar` verwenden, um Übersetzern zusätzliche Informationen zur Verfügung zu stellen. Der Kommentar wird zusammen mit der Zeichenfolge im Übersetzungsmanager angezeigt, den Übersetzer zum Übersetzen von Berichten verwenden.
- Jedes Paar aus `zeichenfolge` und `kommentar` erzeugt eine separate Zeichenfolge, die im Übersetzungsmanager übersetzt wird. Daher werden mit `LokalisierteAbrufen("Product Total";"Max. 20 characters")` und `LokalisierteAbrufen("Product Total";"Use no more than 20 characters")` möglicherweise unterschiedliche Übersetzungen zurückgegeben.

Beispiele

`LokalisierteAbrufen("Total for all products")` gibt die französische Übersetzung von "Total for all products" zurück, wenn das bevorzugte Anzeigegebietsschema "fr_FR" ist.

`LokalisierteAbrufen("Total for all products";"Try not to use more than 20 characters")` gibt die deutsche Übersetzung von "Total for all products" zurück, wenn das bevorzugte

Anzeigegebietsschema "de_DE" ist. Außerdem wird der Übersetzer mit der Funktion angewiesen, bei der Übersetzung der Zeichenfolge möglichst nicht mehr als 20 Zeichen zu verwenden.

Weitere Informationen

[BevorzugtesAnzeigegebietsschemaAbrufen \[Seite 248\]](#)

1.6.1.10.15 BevorzugtesAnzeigegebietsschemaAbrufen

Beschreibung

Gibt das vom Benutzer bevorzugte Gebietsschema zum Anzeigen der Dokumentdaten zurück (das bevorzugte Anzeigegebietsschema)

Funktionsgruppe

Sonstige

Syntax

```
Zeichenfolge BevorzugtesAnzeigegebietsschemaAbrufen( )
```

Beispiele

BevorzugtesAnzeigegebietsschemaAbrufen gibt "en_US" zurück, wenn das bevorzugte Anzeigegebietsschema "Englisch (US)" ist.

Weitere Informationen

[LokalisierteAbrufen \[Seite 247\]](#)

[DominantesBevorzugtesAnzeigegebietsschemaAbrufen \[Seite 245\]](#)

1.6.1.10.16 Wenn...Dann...Sonst

Beschreibung

Gibt einen Wert zurück, der auf dem Ergebnis eines Booleschen Ausdrucks (Wahr/Falsch) beruht.

Funktionsgruppe

Sonstige

Syntax

```
Wenn Bool_Wert Dann True_Wert [Sonst False_Wert]
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Bool_Wert	Ein boolescher Wert	Boolesch	Ja
True_Wert	Wert, der zurückgegeben wird, wenn <code>bool_value</code> wahr ist	Beliebig	Ja
False_Wert	Wert, der zurückgegeben wird, wenn <code>bool_value</code> falsch ist	Beliebig	Ja, wenn Else enthalten ist

Hinweise

- `true_value` und `false_value` können mit verschiedenen Datentypen verwendet werden.
- Sie können die booleschen Operatoren `And`, `Between`, `InList`, `Or` und `Not` mit der Funktion `If` verwenden.
- Sie können die `If`-Bedingungen verschachteln, indem Sie `Else`-Bedingungen durch `ElseIf`-Bedingungen ersetzen. Die folgende Syntax beschreibt eine Verschachtelungsebene:

```
Wenn Bool_Wert Dann True_Wert [SonstWenn Bool_Wert Dann True_Wert Sonst  
False_Wert...]
```

- Die Originalsyntax der `If`-Funktion, `If(bool_value;true_value;false_value)`, wird ebenfalls unterstützt.

Beispiele

If [Umsatz]>1000000 Then "Hoher Umsatz" gibt für alle Zeilen mit einem Umsatz über 1.000.000 "Hoher Umsatz" zurück und für alle anderen Zeilen nichts.

If [Umsatz]>1000000 Then "Hoher Umsatz" Else [Umsatz] gibt für alle Zeilen mit einem Umsatz über 1.000.000 "Hoher Umsatz" und für alle anderen Zeilen den Umsatzwert zurück.

If [Umsatz]>1000000 Then "Hoher Umsatz" Else "Niedriger Umsatz" gibt für alle Zeilen mit einem Umsatz über 1.000.000 "Hoher Umsatz" und für alle Zeilen mit einem Umsatz unter 1.000.000 "Niedriger Umsatz" zurück.

If [Umsatz] > 1000000 Then "Hoher Umsatz" ElseIf [Umsatz] > 800000 Then "Mittlerer Umsatz" Else "Niedriger Umsatz" gibt für alle Zeilen mit einem Umsatz über 1.000.000 "Hoher Umsatz", für alle Zeilen mit einem Umsatz zwischen 800.000 und 1.000.000 "Mittlerer Umsatz" und für alle Zeilen mit einem Umsatz unter 1.000.000 "Niedriger Umsatz" zurück.

Weitere Informationen

[Wenn \[Seite 250\]](#)

[Und \(Operator\) \[Seite 273\]](#)

[Zwischen \(Operator\) \[Seite 274\]](#)

[InListe \(Operator\) \[Seite 275\]](#)

[Oder \(Operator\) \[Seite 273\]](#)

[Nicht \(Ungleich-Operator\) \[Seite 274\]](#)

1.6.1.10.17 Wenn

Beschreibung

Gibt einen Wert zurück, der auf dem Ergebnis eines Booleschen Ausdrucks (Wahr/Falsch) beruht.

Funktionsgruppe

Sonstige

Syntax

```
Wenn(bool_Wert;true_Wert>false_Wert)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Bool_Wert	Ein boolescher Wert	Boolesch	Ja
True_Wert	Wert, der zurückgegeben wird, wenn Bool_Wert wahr ist	Beliebig	Ja
False_Wert	Wert, der zurückgegeben wird, wenn Bool_Wert falsch ist	Beliebig	Ja

Hinweise

- True_Wert und False_Wert können mit verschiedenen Datentypen verwendet werden.
- Sie können wenn-Bedingungen verschachteln, indem Sie False_Wert durch zusätzliche wenn-Bedingungen ersetzen: Diese Syntax zeigt eine Verschachtelungsebene:

```
Wenn(bool_Wert;true_Wert;Wenn(bool_Wert;true_Wert>false_Wert);false_Wert)
```

- Die Wenn...Dann...Sonst-Syntax wird ebenfalls unterstützt.

Beispiele

Wenn([Umsatz]>1000000;"Hoher Umsatz";"Niedriger Umsatz") gibt "Hoher Umsatz" für alle Zeilen zurück, deren Umsatz über 1.000.000 liegt, und "Niedriger Umsatz" für alle Zeilen, deren Umsatz unter 1.000.000 liegt.

Wenn([Umsatz]>1000000;"Hoher Umsatz";[Umsatz]) gibt für alle Zeilen mit einem Umsatz über 1.000.000 "Hoher Umsatz" und für alle anderen Zeilen den Umsatzwert zurück.

Weitere Informationen

[Wenn...Dann...Sonst \[Seite 249\]](#)

1.6.1.10.18 ZeileNummer

Beschreibung

Gibt die Zeilennummer in einer Tabelle zurück.

Funktionsgruppe

Sonstige

Syntax

```
Ganzz ZeileNummer()
```

Hinweise

Die Nummerierung der Zeilen in einer Tabelle beginnt mit dem Kopf, der Zeile 1 darstellt.

Beispiele

`ZeileNummer()` gibt 2 zurück, wenn sich die Funktion in der zweiten Zeile einer Tabelle befindet.

1.6.1.10.19 Next

Beschreibung

Gibt den nächsten Wert eines Objekts zurück.

Funktionsgruppe

Sonstige

Syntax

```
input_type Next(dimension|measure [;Row|Col][;reset_dims][;offset][;NotNull])
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Dimension/Kennzahl	Dies ist die Dimension oder Kennzahl, deren nächsten Wert die Funktion zurückgibt.	Dimension oder Kennzahl	Ja
Zeile/Spalte	Legt die Berechnungsrichtung fest. Sie wird in einer Kreuztabelle verwendet, um zu definieren, ob der nächste zurückgegebene Wert der Wert in der Zeile oder der Spalte ist.	Schlüsselwort	Nein
Dimensionen_zurücksetzen	Dies ist die Liste der zum Zurücksetzen der Berechnung verwendeten Dimensionen.	Dimensionsliste	Nein
Offset	Gibt den nächsten Wert zurück, der um so viele Zeilen, wie durch <code>offset</code> angegeben, nach der aktuellen Zeile liegt.	Ganzzahl	Nein (Standardwert ist "1")
NichtNull	Weist die Funktion an, den ersten Wert zurückzugeben, der nicht Null ist, angefangen mit dem Offset.	Schlüsselwort	Nein

Hinweise

- `Next` ist nicht kompatibel mit anzeigeabhängigen Funktionen wie `ColumnNumber`, `LineNumber`, `PageNumber`, `Page` oder `PageInSection`. Eine Kombination dieser Funktionen führt zu einem `#RECURSIVE`-Fehler. Als Behelfslösung wird empfohlen, die Funktionen `ColumnNumber` und `LineNumber` durch eine laufende Summe zu ersetzen. Für `PageNumber`, `Page` und `PageInSection` gibt es keine Behelfslösung.
- Der Standardwert für `offset` ist 1. `Next ([Revenue] ; 1)` und `Next ([Revenue])` sind funktional gleich.
- Wenn die Funktion mit dem Argument `NoNull` verwendet wird, gibt sie den ersten Wert zurück, der nicht Null ist, beginnend mit der Zelle, die sich `offset` Zeilen vor der aktuellen Zeile befindet, und rückwärts zählend.
- Sie können erweiterte Syntax-Kontextoperatoren mit der Funktion `Next` verwenden.
- Dimensionen müssen selbst dann immer in Klammern eingeschlossen werden, wenn die Liste der zurückgesetzten Dimensionen nur eine Dimension enthält.
- Beim Festlegen einer Reihe zurückgesetzter Dimensionen müssen Sie diese durch Semikolons trennen.
- `Next` wird angewendet, nachdem alle Berichts-, Sektions- und Blockfilter sowie alle Sortierungen angewendet wurden.

- Es können keine Sortierungen oder Filter auf Formeln angewendet werden, die `Next` verwenden.
- Wenn `Next` auf eine Kennzahl angewendet wird und die Kennzahl einen nicht definierten Wert zurückgibt, gibt "Next" selbst dann einen nicht definierten Wert zurück, wenn von der vorherigen Zeile ein Wert zurückgegeben wurde.
- Wenn `Next` außerhalb eines Gruppenwechselkopfs oder -fußes platziert wird, werden Gruppenwechsel ignoriert.
- Wenn `Next` innerhalb eines Gruppenwechselfußes platziert wird, wird der Wert in der vorherigen Instanz des Gruppenwechselfußes zurückgegeben.
- `Next` wird in jeder Berichtssektion zurückgesetzt.
- Bei Verwendung von `Next` in einer Kreuztabelle wird der letzte Wert einer Zeile nicht als vorheriger Wert des ersten Werts der nächsten Zeile behandelt.

1.6.1.10.20 KeinFilter

Beschreibung

Ignoriert Filter beim Berechnen eines Wertes. `KeinFilter` wird mit Kennzahlobjekten verwendet. Es gilt nicht für Dimensionen.

Funktionsgruppe

Sonstige

Syntax

```
Eingabetyp KeinFilter(Obj[;Alle|Drill])
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Obj	Beliebiges Berichtsobjekt.	Berichtsobjekt	Ja
Alle Drill	<ul style="list-style-type: none"> Kein Schlüsselwort angegeben – Bericht ignorieren und Filter blockieren 	Schlüsselwort	Nein

Parameter	Beschreibung	Typ	Erforderlich
	<ul style="list-style-type: none"> • <code>Alle</code> – alle Filter ignorieren • <code>Drill</code> – Berichts- und Drillfilter ignorieren 		

Hinweise

- `KeinFilter(Obj;Drill)` funktioniert im Abfrage-Drill-Modus nicht, da die Drillfilter nicht auf die Berichtsdaten angewendet, sondern zur Abfrage hinzugefügt werden.
- Wenn der Drill-Modus mit aktivierten Drillfiltern beendet wird, werden die Drillfilter zu Berichtsfiltern und können den Wert jedes Objekts ändern, auf das `KeinFilter(Obj;Drill)` angewendet wird.

Beispiele

Wenn `KeinFilter(Summe([Umsatz]))` in einer Blockfußzeile platziert ist, wird der Gesamtumsatz aller möglichen Zeilen im Block zurückgegeben, selbst wenn Zeilen aus dem Block herausgefiltert werden.

`KeinFilter(Summe([Umsatz]);Alle)` gibt die Umsatzsumme für alle Länder einschließlich Frankreich zurück, auch wenn ein Filter vorhanden ist, durch den Frankreich aus dem Bericht ausgeschlossen wird.

`KeinFilter(Summe([Umsatz]);Drill)` gibt die Umsatzsumme für alle Länder zurück, auch wenn ein Drill-Filter für die Dimension [Land] vorhanden ist.

1.6.1.10.21 AnzahlSeiten

Beschreibung

Gibt die Anzahl der in einem Bericht enthaltenen Seiten zurück

Funktionsgruppe

Sonstige

Syntax

```
Ganzz AnzahlSeiten()
```

Hinweise

Wenn Sie die Funktion `NumberOfPages` in eine Zelle einfügen, deren Eigenschaften für die automatische Anpassung der Höhe und Breite festgelegt sind, gibt die Zelle `#REKURSIV` zurück, da das Einfügen dieser Formel in eine Zelle mit automatischer Anpassung eine zirkuläre Abhängigkeit erzeugt. Diese Funktion benötigt die exakte Größe des Berichts, um einen Wert zurückzugeben. Allerdings wird die Größe der Zelle, die Auswirkungen auf die Größe des Berichts hat, vom Zelleninhalt bestimmt.

Beispiele

`AnzahlSeiten()` gibt 2 zurück, wenn der Bericht aus zwei Seiten besteht.

1.6.1.10.22 Öffnungsperiode

Beschreibung

Gibt die Kennzahl am ersten Datum des Zeitraums im aktuellen Kontext und in dem in der Zeitdimension definierten Zeitraum zurück.

Funktionsgruppe

Sonstige

Syntax

```
Eingabetyp Öffnungsperiode(Kennzahl;Zeitraum)
```


Eingabe

Parameter	Beschreibung	Art	Erforderlich
Kennzahl	Beliebige Kennzahl oder Variable.	Kennzahl	Ja
Zeitraum	Der Zeitraum, der den Berechnungskontext im Block bereitstellt.	Zeitraum	Ja

ⓘ Hinweis

- Das Zeitobjekt muss ein im Block verfügbarer Zeitraum sein. Wenn der Block den Zeitraum nicht enthält, gibt die Funktion den Fehler #BERECHNUNG zurück.
- Die Berichtsfiler für die Zeitdimension können sich auf die Funktionsergebnisse auswirken. Sie können die Funktion mit der Funktion `NoFilter` kombinieren, um Berichtsfiler in der Funktionsauswertung zu ignorieren.

Beispiele

Die Spalte "Öffnungsperiode" in der Tabelle unten enthält folgende Formel:

```
OpeningPeriod([Revenue]; [Time Dimension].[Year])
```

Jahr	Umsatz	Öffnungsperiode
2016	1000	1000
2017	2000	1000

```
OpeningPeriod([Revenue]; [Time Dimension].[Semester])
```

Jahr	Semester	Umsatz	Öffnungsperiode
2016	H1 2016	400	400
2016	H2 2016	600	400
2017	H1 2017	500	400
2017	H2 2017	1500	400

Weitere Informationen

[#BERECHNUNG \[Seite 311\]](#)

1.6.1.10.23 Seite

Beschreibung

Gibt die aktuelle Seitenzahl im Bericht zurück.

Funktionsgruppe

Sonstige

Syntax

```
Ganzz Seite()
```

Hinweise

Wenn Sie die Funktion `Page` in eine Zelle einfügen, deren Eigenschaften für die automatische Anpassung der Höhe und Breite festgelegt sind, gibt die Zelle `#REKURSIV` zurück, da das Einfügen dieser Formel in eine Zelle mit automatischer Anpassung eine zirkuläre Abhängigkeit erzeugt. Diese Funktion benötigt die exakte Größe des Berichts, um einen Wert zurückzugeben. Allerdings wird die Größe der Zelle, die Auswirkungen auf die Größe des Berichts hat, vom Zelleninhalt bestimmt.

Beispiel

`Page ()` gibt 2 zurück, wenn sich die Funktion auf der zweiten Seite des Berichts befindet.

1.6.1.10.24 PageInSection

Beschreibung

Gibt die Seitenzahl in der aktuellen Sektionsinstanz in einem bestimmten Abschnitt zurück.

Funktionsgruppe

Sonstige

Syntax

```
integer PageInSection([section_level])
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Sektionsebene	Hierarchie-Ebene der Sektion	integer	Nein

Hinweise

- Wenn Sie die Funktion `PageInSection` in eine Zelle einfügen, deren Eigenschaften für die automatische Anpassung der Höhe und Breite festgelegt sind, gibt die Zelle `#REKURSIV` zurück, da das Einfügen dieser Formel in eine Zelle mit automatischer Anpassung eine zirkuläre Abhängigkeit erzeugt. Diese Funktion benötigt die exakte Größe des Berichts, um einen Wert zurückzugeben. Allerdings wird die Größe der Zelle, die Auswirkungen auf die Größe des Berichts hat, vom Zelleninhalt bestimmt.
- `PageInSection()` muss innerhalb einer Sektionsinstanz liegen. Andernfalls wird 0 zurückgegeben.
- Die Hierarchie-Ebenen der Sektion fangen bei 1 an (oberste Ebene).
- Wenn die `section_level` (Sektionsebene) nicht angegeben ist, gibt die Funktion die Nummer in der aktuellen Sektionsebene zurück.
- Wenn Sie eine Sektionsebene angeben, die nicht vorhanden ist, gibt die Funktion 0 zurück.
- Variablen, die mit der Funktion erstellt werden, müssen Kennzahlen sein.

Beispiele

In einem Dokument mit einer Sektion für Jahr (= 2010, 2011, 2012) und einer Untersektion für Bundesstaat (= Kalifornien, Florida, Texas):

- Wenn die Funktion `PageInSection(1)` (`SeitenInSection(1)`) in der Sektion Jahr wiederholt wird, wird "2" auf der zweiten Seite von 2010, 2011 und 2012 zurückgegeben.
- Wenn die Funktion `PageInSection(2)` (`SeitenInSection(2)`) in der Untersektion Bundesstaat wiederholt wird, wird "1" auf der ersten Seite von Kalifornien, Florida und Texas wiedergegeben.

1.6.1.10.25 ParallelePeriode

Beschreibung

Gibt die Kennzahl zum Datum einer Periode zurück, die parallel zu den Datumsangaben im Block im aktuellen Kontext ist. Ausgewählte Daten werden um eine Reihe von Intervallen zeitlich nach vorne oder nach hinten verschoben.

Funktionsgruppe

Sonstige

Syntax

```
Eingabetyp ParallelePeriode(Kennzahl;Zeitraum;Offset)
```

Eingabe

Parameter	Beschreibung	Art	Erforderlich
Kennzahl	Beliebige Kennzahl oder Variable.	Kennzahl	Ja
Zeitraum	Der Zeitraum, der den Berechnungskontext im Block bereitstellt.	Zeitraum	Ja
Offset	Gibt die Anzahl der Intervalle entweder vorwärts oder rückwärts in der Zeit an	Ganzzahl	Ja

ⓘ Hinweis

- Das Zeitobjekt muss ein im Block verfügbarer Zeitraum sein. Wenn der Block den Zeitraum nicht enthält, gibt die Funktion den Fehler #BERECHNUNG zurück.
- Die Funktion ist unabhängig von der Sortierreihenfolge der Zeitdimension im Block.
- Die Berichtsfilter für die Zeitdimension können sich auf die Funktionsergebnisse auswirken. Sie können die Funktion mit der Funktion `NOFILTER` kombinieren, um Berichtsfilter in der Funktionsauswertung zu ignorieren.

Beispiele

Die Spalte „ParallelePeriode“ in der Tabelle unten enthält folgende Formel:

```
ParallelPeriod([Revenue]; [Time Dimension].[Year];-1
```

Jahr	Umsatz	Abschlussperiode
2015	600	–
2016	1000	600
2017	2000	2000

```
ParallelPeriod ([Revenue]; [Time Dimension].[Semester];-1)
```

Jahr	Semester	Umsatz	ParallelePeriode
2015	H1 2015	200	
2015	H2 2015	400	200
2016	H1 2016	400	400
2016	H2 2016	600	400
2017	H1 2017	500	600
2017	H2 2017	1500	500

Weitere Informationen

[#BERECHNUNG \[Seite 311\]](#)

1.6.1.10.26 PeriodeKumuliert

Beschreibung

Gibt die laufende Summe der Werte der Kennzahl über den Zeitraum bis zum heutigen Datum im aktuellen Kontext zurück. Der Benutzer kann beispielsweise „Jahr“ auswählen, um die YTD-Werte (Jahr kumuliert) für jeden Monat neben den Monatswerten anzuzeigen.

Funktionsgruppe

Sonstige

Syntax

```
Eingabetyp PeriodeKumuliert (Kennzahl; Zeitraum; [Summe | Max | Min | Anzahl | Durchschnitt | Produkt])
```

Eingabe

Parameter	Beschreibung	Art	Erforderlich
Kennzahl	Beliebige Kennzahl oder Variable.	Kennzahl	Ja
Zeitraum	Der Zeitraum, der den Berechnungskontext im Block bereitstellt.	Zeitraum	Ja
LaufendeFunktion	Anzuwendende laufende Funktion. Mögliche Werte: <ul style="list-style-type: none">• Sum (Standard)• Max• Min• Count• Average• Product	Enumeration	Nein

ⓘ Hinweis

- Das Zeitobjekt muss ein im Block verfügbarer Zeitraum sein. Wenn der Block den Zeitraum nicht enthält, gibt die Funktion den Fehler #BERECHNUNG zurück.
- Wenn der Block andere Dimensionen enthält, werden diese letzten Dimensionen implizit als zurückgesetzte Dimensionen für die ausgeführte Funktion verwendet.
- Wenn der Block andere Zeiträume als die in der Funktion definierten enthält, wird die laufende Summe auf den Zeitraum mit der niedrigsten Zeitgranularität angewendet.
- Die Funktion ist unabhängig von der Sortierreihenfolge der Zeitdimension im Block.
- Die Berichtsfiler für die Zeitdimension können sich auf die Funktionsergebnisse auswirken. Sie können die Funktion mit der Funktion `NOFILTER` kombinieren, um Berichtsfiler in der Funktionsauswertung zu ignorieren.

Beispiele

Die Spalte "PeriodeKumuliert" in der folgenden Tabelle enthält folgende Formel:

```
PeriodToDate([Revenue]; [Time Dimension].[Year])
```

Jahr	Semester	Umsatz	Abschlussperiode
2015	H1 2015	200	200
2015	H2 2015	400	600
2016	H1 2016	400	400
2016	H2 2016	600	1000
2017	H1 2017	500	500
2017	H2 2017	1500	2000

Jahr	Semester	Produktfamilie	Umsatz	PeriodeKumuliert
2015	H1 2015	Lebensmittel	50	50
2015	H1 2015	Elektronik	150	150
2015	H2 2015	Lebensmittel	100	150
2015	H2 2015	Elektronik	300	450
2016	H1 2016	Lebensmittel	150	150
2016	H1 2016	Elektronik	250	250
2016	H2 2016	Lebensmittel	200	350
2016	H2 2016	Elektronik	400	650
2017	H1 2017	Lebensmittel	200	200
2017	H1 2017	Elektronik	300	300
2017	H2 2017	Lebensmittel	500	700
2017	H2 2017	Elektronik	1000	1300

Weitere Informationen

[#BERECHNUNG \[Seite 311\]](#)

1.6.1.10.27 Vorherige

Beschreibung

Gibt den vorherigen Wert eines Objekts zurück.

Funktionsgruppe

Sonstige

Syntax

```
Eingabetyp Vorherige(Dimension|Kennzahl|Selbst [;Zeile|Spalte]  
[(Dimensionen_zurücksetzen)[;Offset][;NotNull])
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Dimension Kennzahl Selbst	Dimension oder Kennzahl, deren vorherigen Wert die Funktion zurückgibt, oder das Selbst-Schlüsselwort	Dimension, Kennwort oder Schlüsselwort	Ja
Zeile/Spalte	Legt die Berechnungsrichtung fest	Schlüsselwort	Nein
Dimensionen_zurücksetzen	Die Liste der zum Zurücksetzen der Berechnung verwendeten Dimensionen	Dimensionsliste	Nein
Offset	Gibt den Wert für <code>dimension</code> oder <code>measure</code> an, die sich <code>offset</code> Zeilen vor der aktuellen Zeile befindet.	Ganzzahl	Nein (Standardwert ist "1")
NotNull	Weist die Funktion an, den ersten Wert zurückzugeben, der nicht Null ist, angefangen mit dem Offset	Schlüsselwort	Nein

Hinweise

- `Previous` ist nicht kompatibel mit anzeigeabhängigen Funktionen wie `ColumnNumber`, `LineNumber`, `PageNumber`, `Page` oder `PageInSection`. Eine Kombination dieser Funktionen führt zu einem `#RECURSIVE`-Fehler. Als Behelfslösung wird empfohlen, die Funktionen `ColumnNumber` und `LineNumber` durch eine laufende Summe zu ersetzen. Für `PageNumber`, `Page` und `PageInSection` gibt es keine Behelfslösung.
- Der Standardwert für `offset` ist 1. `Previous([Revenue];1)` und `Previous([Revenue])` sind funktional gleich.
- Wenn die Funktion mit dem Argument `NotNull` verwendet wird, gibt sie den ersten Wert zurück, der nicht Null ist, beginnend mit der Zelle, die sich `offset` Zeilen vor der aktuellen Zeile befindet, und rückwärts zählend.
- Sie können erweiterte Syntax-Kontextoperatoren mit der Funktion `Previous` verwenden.
- Mithilfe des Operators `self` können Sie auf den vorherigen Wert einer Zelle verweisen, wenn sie einen anderen Inhalt als ein Berichtsobjekt aufweist.

- Dimensionen müssen selbst dann immer in Klammern eingeschlossen werden, wenn die Liste der zurückgesetzten Dimensionen nur eine Dimension enthält.
- Beim Festlegen einer Reihe zurückgesetzter Dimensionen müssen Sie diese durch Semikolons trennen.
- `Previous` wird angewendet, nachdem alle Berichts-, Sektions- und Blockfilter sowie alle Sortierungen angewendet wurden.
- Es können keine Sortierungen oder Filter auf Formeln angewendet werden, die `Previous` verwenden.
- Wenn `Previous` auf eine Kennzahl angewendet wird und die Kennzahl einen nicht definierten Wert zurückgibt, gibt `Previous` selbst dann einen nicht definierten Wert zurück, wenn von der vorherigen Zeile ein Wert zurückgegeben wurde.
- Wenn `Previous` außerhalb eines Gruppenwechselkopfs oder -fußes platziert wird, werden Gruppenwechsel ignoriert.
- Wenn `Previous` innerhalb eines Gruppenwechselkopfs platziert wird, wird der Wert in der vorherigen Instanz des Gruppenwechselkopfs zurückgegeben.
- `Previous` wird in jeder Berichtssektion zurückgesetzt.
- Bei Verwendung von `Previous` in einer Kreuztabelle wird der letzte Wert einer Zeile nicht als vorheriger Wert des ersten Werts der nächsten Zeile behandelt.

Beispiele

`Previous ([Country] ; 1)` gibt die folgenden Werte in der folgenden Tabelle zurück:

Land	Umsatz	Vorherige
USA	5.000.000	
GB	2.000.000	USA
Frankreich	2.100.000	GB

`Previous ([Revenue])` gibt die Werte in der folgenden Tabelle zurück:

Land	Umsatz	Vorherige
USA	5.000.000	
GB	2.000.000	5.000.000
Frankreich	2.100.000	2.000.000

`Previous ([Revenue] ; ([Country]))` gibt die Werte in der folgenden Tabelle zurück:

Land	Region	Umsatz	Vorherige
USA	Norden	5.000.000	
	Süden	7.000.000	5.000.000
GB	Norden	3.000.000	
	Süden	4.000.000	3.000.000

Previous ([Revenue]) gibt die Werte in der folgenden Kreuztabelle zurück:

	2004	Vorherige	2005	Vorherige
USA	5.000.000		6.000.000	5.000.000
GB	2.000.000		2.500.000	2.000.000
Frankreich	3.000.000		2.000.000	3.000.000

Previous ([Revenue]) gibt die Werte in der folgenden Tabelle mit einem Gruppenwechsel für [Land] zurück:

Land	Region	Umsatz	Vorherige
USA	Norden	5.000.000	
	Süden	7.000.000	5.000.000
USA		12.000.000	
Land	Region	Umsatz	Vorherige
GB	Norden	3.000.000	7.000.000
	Süden	4.000.000	3.000.000
GB		7.000.000	12.000.000

Previous ([Revenue]) ; 2 ; NoNull) gibt die Werte in der folgenden Tabelle zurück:

Jahr	Quartal	Umsatz	Vorherige
2008	Q1	500	
2008	Q2		
2008	Q3	400	500
2008	Q4	700	500
2008	Q1	300	400
2008	Q2		700
2008	Q3		300
2008	Q4	200	300

2*Previous (Self) gibt die Sequenz 2, 4, 6, 8, 10... zurück.

Weitere Informationen

[Vergleichen von Werten mithilfe der Funktion "Vorherige" \[Seite 317\]](#)

[Selbst \(Operator\) \[Seite 284\]](#)

1.6.1.10.28 RefWert

Beschreibung

Gibt bei Aktiviertem Datentracking den Referenzwert eines Berichtsobjekts zurück.

Funktionsgruppe

Sonstige

Syntax

```
Eingabetyp RefWert (Obj)
```

Beispiele

`RefWert ([Umsatzstärkste Region])` gibt "Südwest" zurück, wenn der Wert der Variablen `[Umsatzstärkste Region]` in den Referenzdaten "Südwest" ist.

`RefWert ([Umsatz])` gibt 1000 zurück, wenn der Wert der Kennzahl `[Umsatz]` in den Referenzdaten 1000 ist.

Hinweise

- Die `RefWert()`-Funktion kann mit einem Kennzahl- oder einem Dimensionsobjekt verwendet werden. Wird sie jedoch in einer Variablen verwendet, die als Dimension oder Detail qualifiziert ist, gibt die `RefWert()`-Funktion anstelle der Referenzwerte die aktuellen Werte dieses Objekts zurück. Um die Referenzwerte zu erhalten, muss die Variable als Kennzahl qualifiziert sein.
- Wenn eine Formel direkt in einem Abschnitt, einer Tabelle, einem Formular oder Diagramm erstellt wurde, wird eine Formel immer als Kennzahl qualifiziert. Das bedeutet, wenn in der Formel die `RefWert()`-Funktion verwendet wurde, gibt diese die erwarteten Referenzwerte zurück.

Beispiel für die RefWert-Funktion mit einer Variablen

Es liegt eine Liste der Werte für die Dimension `[Bundesstaat]` vor: Kalifornien, Florida, Texas und New York. Nach der Datenregenerierung sieht die Liste wie folgt aus: Arizona, Kalifornien, Florida, Texas und New York. Eine Variable wie `Variable=RefWert([Bundesstaat])` gibt eine der folgenden Optionen zurück:

Variable ist qualifiziert als

Die Liste der zurückgegebenen Werte lautet

Dimension oder Detail	Arizona, Kalifornien, Florida, Texas und New York.
-----------------------	--

Kennzahl	(Nullwert), Kalifornien, Florida, Texas und New York.
----------	---

1.6.1.10.29 RelativerWert

Beschreibung

Gibt vorherige oder nachfolgende Werte eines Objekts zurück.

Funktionsgruppe

Sonstige

Syntax

```
Eingabetyp RelativerWert(Kennzahl|Detail;Ausschnittsdimensionen;Offset)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Kennzahl Detail	Beliebige Kennzahl oder Detail einer Dimension im Block	Kennzahl oder Detail	Ja
Ausschnittsdimensionen	Die Dimensionen, die den Berechnungskontext bieten	Dimensionsliste	Ja
Offset	Gibt den Wert für <code>measure</code> oder <code>detail</code> an, die bzw. das sich <code>offset</code> Zeilen von der aktuellen Zeile entfernt befindet.	Ganzzahl	Ja

Hinweise

- Das Objekt muss eine Kennzahl oder ein Detail einer Dimension sein, die im Block verfügbar ist.
- Die Sortierreihenfolge der Werteliste der Ausschnittsdimensionen wird verwendet, um die Ausgabe der Funktion zu bestimmen.
Die Sortierreihenfolge wird durch zwei Faktoren bestimmt: Sortierungen, die auf die Ausschnittsdimensionen angewendet werden, und die Reihenfolge, in der die Ausschnittsdimensionen in der Funktion aufgelistet werden.
- Eine Dimension, die als führende Sektion einer Sektion verwendet wird, kann als Ausschnittsdimension festgelegt werden.
- Alle Ausschnittsdimensionen müssen im Block oder der Sektionszelle des Blocks, in den die Funktion platziert wird, vorhanden sein. Wenn eine Ausschnittsdimension später aus dem Block entfernt wird, gibt die Funktion den Fehler #BERECHNUNG zurück.
- Wenn der Offset die Anzahl der Zeilen in der Werteliste der Ausschnittsdimension übersteigt, gibt die Funktion den Wert null zurück.
- `RelativeValue` kann nicht rekursiv verwendet werden.
- Dimensionen müssen selbst dann immer in Klammern eingeschlossen werden, wenn die Liste der Ausschnittsdimensionen nur eine Dimension enthält.

Beispiele

Die Spalte "RelativerWert" in der folgenden Tabelle enthält folgende Formel:

```
RelativerWert([Umsatz];([Jahr]);-1)
```

Jahr	Quartal	Vertriebsmitarbeiter	Umsatz	RelativerWert
2007	Q1	Schmidt	1000	
2007	Q2	Müller	2000	
2007	Q3	Krause	1500	
2007	Q4	Meier	3000	
2008	Q1	Schmidt	4000	1000
2008	Q2	Müller	3400	2000
2008	Q3	Krause	2000	1500
2008	Q4	Meier	1700	3000

Weitere Informationen

[#BERECHNUNG \[Seite 311\]](#)

[Vergleichen von Werten mithilfe der Funktion RelativerWert. \[Seite 317\]](#)

1.6.1.10.30 Berichtsname

Beschreibung

Gibt den Namen eines Berichts zurück

Funktionsgruppe

Sonstige

Syntax

```
Zeichenfolge Berichtsname()
```

Beispiele

`Berichtsname()` gibt "Verkaufsbericht" zurück, wenn das Dokument den Namen "Verkaufsbericht" trägt.

1.6.1.10.31 ZeilenIndex

Beschreibung

Gibt die Nummer einer Zeile zurück.

Funktionsgruppe

Sonstige

Syntax

```
Ganzz ZeilenIndex()
```

Hinweise

- Die Zeilennummerierung beginnt bei 0.
- `zeilenIndex` gibt #MEHRFACHWERT zurück, wenn die Funktion in einen Tabellenkopf oder -fuß eingefügt wird.

Beispiele

`zeilenIndex` gibt 0 zurück, wenn sich die Funktion in der ersten Zeile einer Tabelle befindet.

1.6.1.10.32 EindeutigerNameVon

Beschreibung

Gibt den eindeutigen Namen eines Objekts zurück.

Funktionsgruppe

Sonstige

Syntax

```
Zeichenfolge EindeutigerNameVon(Obj)
```

Eingabe

Parameter	Beschreibung	Typ	Erforderlich
Obj	Beliebiges Berichtsobjekt.	Berichtsobjekt	Ja

Beispiele

`EindeutigerNameVon([Reservierungsdatum])` gibt "Reservierungsdatum" zurück.

1.6.2 Funktions- und Formeloperatoren

Mithilfe von Operatoren werden die einzelnen Komponenten einer Formel miteinander verknüpft.

Formeln können mathematische, bedingte, logische und funktionspezifische Operatoren sowie Operatoren für die erweiterte Syntax enthalten.

1.6.2.1 Mathematische Operatoren

Bei den mathematischen Operatoren handelt es sich um jene, die im Rahmen gängiger Arithmetikoperationen verwendet werden.

Es gibt Operatoren für Addition (+), Subtraktion (-), Multiplikation (*) und Division (/), mit denen mathematische Operationen in einer Formel ausgeführt werden können. Die Formel `[Umsatz] - [Umsatzkosten]` enthält einen mathematischen Operator in Form einer Subtraktion.

📘 Hinweis

Wenn der Operator '+' in Zusammenhang mit einer Zeichenfolge verwendet wird, dient er als Verkettungsoperator. In diesem Fall verknüpft er zwei Zeichenfolgen. So gibt beispielsweise die Formel `"John" + "Schmidt"` das Ergebnis "John Schmidt" zurück.

1.6.2.2 Bedingungsoperatoren

Bedingungsoperatoren bestimmen, auf welche Art Werte miteinander verglichen werden.

Operator	Beschreibung
=	Gleich
>	Größer als
<	Kleiner als
>=	Größer als oder gleich
<=	Kleiner als oder gleich
<>	Ungleich

Bedingungsoperatoren werden mit der Wenn-Funktion verwendet. Beispiel:

```
If [Umsatz]>10000 Then "Hoch" Else "Niedrig"
```

In diesem Beispiel wird "Hoch" für sämtliche Zeilen zurückgegeben, in denen der Umsatz über oder gleich 10000 ist, und "Niedrig" für alle restlichen Zeilen.

1.6.2.3 Logische Operatoren

Es gibt fünf logische Operatoren: `Und`, `Oder`, `Nicht`, `Zwischen` und `InListe`.

Logische Operatoren werden in Booleschen Ausdrücken verwendet, die `Wahr` oder `Falsch` zurückgeben.

1.6.2.3.1 Und (Operator)

Der Operator `Und` verknüpft boolesche Werte.

Beschreibung

Wenn alle booleschen Werte, die durch `Und` verknüpft sind, "Wahr" zurückgeben, gibt auch die Kombination aller Werte "Wahr" zurück.

Syntax

```
Bool_Wert Und Bool_Wert [Und Bool_Wert...]
```

Beispiele

`If [Erholungsort] ="Bahamas Beach" Und [Umsatz]>100000 Then "Hoher Umsatz für Bahamas"` gibt "Hoher Umsatz für Bahamas" zurück, wenn der Erholungsort "Bahamas Beach" und der Umsatz höher als 100.000 ist.

1.6.2.3.2 Oder (Operator)

Der Operator `Oder` verknüpft boolesche Werte.

Beschreibung

Wenn einer der booleschen Werte, die durch `Oder` verknüpft sind, "Wahr" zurückgibt, gibt auch die Kombination aller Werte "Wahr" zurück.

Syntax

```
Bool_Wert Oder Bool_Wert [Oder Bool_Wert...]
```

Beispiele

If [Erholungsort] = "Bahamas Beach" Oder [Erholungsort] = "Hawaiian Club" Then "USA" Else "Frankreich" gibt "USA" zurück, wenn [Erholungsort] = "Bahamas Beach" oder "Hawaiian Club". Andernfalls wird "Frankreich" zurückgegeben.

1.6.2.3.3 Nicht (Ungleich-Operator)

Beschreibung

Der Operator `Nicht` gibt das Gegenteil eines booleschen Ausdrucks zurück.

Syntax

```
Bool Nicht (Bool_Wert)
```

Beispiele

Wenn `Nicht([Land] = "USA")` Dann "Nicht USA" gibt "Nicht USA" zurück, wenn [Land] einen anderen Wert als "USA" hat.

1.6.2.3.4 Zwischen (Operator)

Beschreibung

Der Operator `Zwischen` stellt fest, ob eine Variable zwischen zwei Werten liegt.

Syntax

```
Bool Zwischen(erster_Wert;zweiter_Wert)
```

Hinweise

- Der Operator `Zwischen` kann mit der Funktion `If` und dem Operator `wo` verwendet werden.
- Änderungen des Gebietsschemas des Dokuments können das von dem `Zwischen`-Operator zurückgegebene Ergebnis beeinflussen.

Beispiele

`If [Umsatz] Zwischen(800000;900000) Then "Mittlerer Umsatz" gibt "Mittlerer Umsatz" zurück, wenn [Umsatz] zwischen 800000 und 900000 liegt.`

`[Umsatz] Zwischen (10000;20000) gibt "Wahr" zurück, wenn der Umsatz zwischen 10000 und 20000 liegt.`

`If ([Umsatz] Zwischen (200000;500000); "Mittlerer Umsatz"; "Geringer/Hoher Umsatz") gibt "Mittlerer Umsatz" zurück, wenn die Variable [Umsatz] gleich 300000 ist.`

Weitere Informationen

[Wenn...Dann...Sonst \[Seite 249\]](#)

[Wo \(Operator\) \[Seite 285\]](#)

1.6.2.3.5 InListe (Operator)

Beschreibung

Der Operator `InListe` stellt fest, ob sich ein Wert in einer Werteliste befindet.

Syntax

```
Bool Testwert InListe(Werteliste)
```

Hinweise

Nur durch die Kombination von `Testwert + InListe` wird ein boolescher Wert zurückgegeben, nicht durch `InListe` allein.

Beispiele

```
If Nicht ([Land] InListe("England";"Schottland";"Wales")) Then "Nicht  
Großbritannien" Else "Großbritannien" gibt "Nicht Großbritannien" zurück, wenn [Land] nicht  
"England", "Schottland" oder "Wales" ist. Andernfalls wird "Großbritannien" zurückgegeben.
```

```
If [Erholungsort] InListe("Bahamas Beach";"Hawaiian Club") Then "US-Erholungsort"  
gibt "US-Erholungsort" zurück, wenn [Erholungsort] gleich "Bahamas Beach" oder "Hawaiian Club".
```

Weitere Informationen

[Wenn...Dann...Sonst \[Seite 249\]](#)

[Wo \(Operator\) \[Seite 285\]](#)

1.6.2.4 Funktionsspezifische Operatoren

Einige Funktionen akzeptieren spezifische Operatoren als Argumente.

Die `Vorherige`-Funktion akzeptiert beispielsweise den `Selbst`-Operator.

Alle Funktionen verwenden `)` und `(`, um die Funktionsargumente einzuschließen. Funktionen, die mehrere Parameter akzeptieren, verwenden `;`, um die einzelnen Parameter zu trennen.

1.6.2.4.1 Alle (Operator)

Der Operator `Alle` weist die Funktion `KeinFilter` an, alle Filter zu ignorieren.

Der Operator `Alle` kann auch die Funktion `Anzahl` anweisen, alle Werte einschließlich Duplikaten zu zählen.

Weitere Informationen

[Anzahl \[Seite 53\]](#)

[Einmalig/Alle \(Operatoren\) \[Seite 279\]](#)

[KeinFilter](#) [Seite 254]

[Alle/Drill \(Operatoren\)](#) [Seite 277]

1.6.2.4.2 Alle/Drill (Operatoren)

Die Operatoren `Alle/Drill` sind mit der Funktion `KeinFilter` kompatibel.

Beschreibung

Die Operatoren `Alle/Drill` bestimmen, welche Filter von der Funktion `KeinFilter` ignoriert werden.

- Nicht angegeben – `KeinFilter` ignoriert Berichts- und Blockfilter
- `Alle` – `KeinFilter` ignoriert alle Filter
- `Drill` – `KeinFilter` ignoriert Berichtsfilter und Drill-Filter

1.6.2.4.3 Aufsteigend

Der Operator `Ascending` (Aufsteigend) ist ein Argument der Funktion `PromptSummary`.

Beschreibung

Ist das Kennzeichen gesetzt, werden die Eingabeaufforderungen durch die Funktion `PromptSummary` in aufsteigender Reihenfolge sortiert.

Weitere Informationen

[Funktion "PromptSummary"](#) [Seite 169]

1.6.2.4.4 Letzter/Erster (Operatoren)

Die Operatoren `Letzter/Erster` sind mit der Funktion `Rang` kompatibel.

Beschreibung

Die Operatoren `Letzter`/`Erster` weisen die Funktion `Rang` an, eine Rangfolge in aufsteigender oder absteigender Reihenfolge zu erstellen.

- `Erster` – absteigende Rangfolge
- `Letzter` – aufsteigende Rangfolge

Beispiele

`Rang ([Umsatz] ; ([Land]) ; Erster` legt für Länder eine Rangfolge nach Umsatz, beginnend mit dem höchsten Umsatz, fest.

Weitere Informationen

[Rang \[Seite 200\]](#)

1.6.2.4.5 Gruppenwechsel (Operator)

Der Operator `Gruppenwechsel` ist mit der Funktion `Prozent` kompatibel.

Beschreibung

Der Operator `Gruppenwechsel` weist die Funktion `Prozent` an, Gruppenwechsel in Tabellen zu berücksichtigen.

Beispiele

Die Formel `Prozent ([Umsatz])` gibt die in der folgenden Tabelle aufgelisteten Ergebnisse zurück (Prozentwerte werden für den Gesamtumsatz im Block berechnet):

Jahr	Quartal	Umsatz	Prozent
2005	Q1	10000	10 %
2005	Q2	20000	20 %

2006	Q1	30000	30 %
2006	Q2	40000	40 %

Die Formel `Prozent ([Umsatz];Gruppenwechsel)` gibt die in der folgenden Tabelle aufgelisteten Ergebnisse zurück (Prozentwerte werden für den Gesamtumsatz in jedem Teil des Blocks berechnet):

Jahr	Quartal	Umsatz	Prozent
2005	Q1	10000	33,3 %
2005	Q2	20000	66,6 %
2006	Q1	30000	42,9 %
2006	Q2	40000	57,1 %

Weitere Informationen

[Prozent \[Seite 64\]](#)

1.6.2.4.6 Absteigend

Der Operator `Descending` (Absteigend) ist ein Argument der Funktion `PromptSummary`.

Beschreibung

Ist das Kennzeichen gesetzt, werden die Eingabeaufforderungen durch die Funktion `PromptSummary` in absteigender Reihenfolge sortiert.

Weitere Informationen

[Funktion "PromptSummary" \[Seite 169\]](#)

1.6.2.4.7 Einmalig/Alle (Operatoren)

Die Operatoren `Einmalig/Alle` sind mit der Funktion `Anzahl` kompatibel.

Die Operatoren `Einmalig`/`Alle` weisen die Funktion `Anzahl` an, nur einmalige Werte oder alle Werte zu zählen.

Beispiele

`Anzahl([Umsatz];Einmalig)` gibt 3 zurück, wenn `[Umsatz]` die Werte (5;5;6;4) enthält.

`Anzahl([Umsatz];Alle)` gibt 4 zurück, wenn `[Umsatz]` die Werte (5;5;6;4) enthält.

Weitere Informationen

[Anzahl \[Seite 53\]](#)

1.6.2.4.8 EinschließlichLeer (Operator)

Der Operator `EinschließlichLeer` ist mit Aggregationsfunktionen kompatibel.

Beschreibung

Der Operator `EinschließlichLeer` weist einige Aggregatfunktionen (`Durchschnitt`, `Anzahl`, `LaufenderDurchschnitt`, `LaufendeAnzahl`) an, leere Werte in Berechnungen einzuschließen.

Beispiele

`Durchschnitt([Umsatz]; EinschließlichLeer)` gibt 3 zurück, wenn `[Umsatz]` die Werte (5;3;<leer>;4) aufweist.

Weitere Informationen

[Durchschnitt \[Seite 52\]](#)

[Anzahl \[Seite 53\]](#)

[RunningAverage \[Seite 68\]](#)

[LaufendeAnzahl \[Seite 70\]](#)

1.6.2.4.9 Index (Operator)

Der Operator `Index` ist mit den Funktionen `AntwortBenutzer` und `ReferenzwertAntwortBenutzer` kompatibel.

Beschreibung

Der Operator `Index` weist die Funktionen `AntwortBenutzer` und `ReferenzwertAntwortBenutzer` an, den Datenbankprimärschlüssel der Eingabeaufforderungsantwort zurückzugeben.

Weitere Informationen

[AntwortBenutzer \[Seite 155\]](#)

[ReferenzwertAntwortBenutzer \[Seite 151\]](#)

1.6.2.4.10 Linear (Operator)

Der Operator `Linear` ist mit der Funktion `Interpolation` kompatibel.

Beschreibung

Der Operator `Linear` weist die Funktion `Interpolation` an, zur Angabe fehlender Kennzahlwerte eine lineare Regression mit der Interpolation nach kleinsten Quadraten zu verwenden.

Lineare Regression mit der Interpolation nach kleinsten Quadraten berechnet fehlende Werte mit einer linearen Gleichung in Form von " $f(x) = ax + b$ ", die alle verfügbaren Werte der Kennzahl so genau wie möglich weitergibt.

Weitere Informationen

[Interpolation \[Seite 56\]](#)

1.6.2.4.11 NichtNull (Operator)

Der Operator `NichtNull` ist mit der Funktion `Vorherige` kompatibel.

Beschreibung

Der Operator `NichtNull` weist die Funktion `Vorher` an, Nullwerte zu ignorieren.

Wenn die Funktion mit `NichtNull` verwendet wird, gibt `Vorher` den ersten Wert, der nicht Null ist, beginnend mit der Zelle `offset`-Zeilen vor der aktuellen Zeile und rückwärts zählend, zurück.

Weitere Informationen

[Vorherige \[Seite 263\]](#)

1.6.2.4.12 NichtInGruppenwechsel (Operator)

Der Operator `NichtInGruppenwechsel` ist mit der Funktion `Interpolation` kompatibel.

Beschreibung

Der Operator `NichtInGruppenwechsel` legt fest, dass die Funktion `Interpolation` Abschnitts- und Blockwechsel ignoriert.

Weitere Informationen

[Interpolation \[Seite 56\]](#)

1.6.2.4.13 PunktzuPunkt (Operator)

Der Operator `PunktzuPunkt` weist die Funktion `Interpolation` an, zur Angabe fehlender Kennzahlwerte eine Punkt-zu-Punkt-Interpolation zu verwenden.

Beschreibung

Mit der Punkt-zu-Punkt-Interpolation werden die Werte durch Berechnung einer linearen Gleichung in Form von " $f(x) = ax + b$ " berechnet, die durch die beiden neben dem fehlenden Wert liegenden Werte verläuft.

Weitere Informationen

[Interpolation \[Seite 56\]](#)

1.6.2.4.14 Zeile/Spalte (Operatoren)

Bei Verwendung des `zeile`-Operators wird jeder Wert in der Zeile als Prozentsatz des Gesamtwerts aller Zeilen im Berechnungskontext berechnet. Bei Verwendung des `spalte`-Operators wird jeder Wert in der Spalte als Prozentsatz des Gesamtwerts aller Spalten im Berechnungskontext berechnet.

Beschreibung

Die Operatoren `zeile`/`spalte` legen die Berechnungsrichtung der folgenden Funktionen fest: `Prozent`, `Vorherige`, `LaufenderDurchschnitt`, `LaufendeAnzahl`, `LaufendesMax`, `LaufendesMin`, `LaufendesProdukt`, `LaufendeSumme`.

Hinweise

In einer Kreuztabelle wird der Wert der einzelnen Zellen standardmäßig als Prozentsatz des Gesamtwerts in der Kreuztabelle berechnet. Bei Verwendung des `zeile`-Operators werden die Werte in den Zeilen als Prozentsatz des Gesamtwerts für die Zeile berechnet. Bei Verwendung des `spalte`-Operators werden die Werte in den Spalten als Prozentsatz des Gesamtwerts für die Spalte berechnet.

Beispiele

In einer Kreuztabelle liefert `Prozent ([Kennzahl])` die folgenden Ergebnisse:

Kennzahl	Prozent	Kennzahl	Prozent
100	10 %	500	50 %
200	20 %	200	20 %

`Prozent ([Kennzahl] ; Zeile)` liefert die folgenden Ergebnisse:

Kennzahl	Prozent	Kennzahl	Prozent
100	16,7 %	500	83,3 %

200	50 %	200	50 %
-----	------	-----	------

`Prozent([Kennzahl];Spalte)` liefert die folgenden Ergebnisse:

Kennzahl	Prozent	Kennzahl	Prozent
100	33,3 %	500	83,3 %
200	66,6 %	200	16,7 %

Der Zeile-Operator berechnet die laufende Aggregation nach Zeile. Der Spalte-Operator berechnet die laufende Aggregation nach Spalte.

In einer Kreuztabelle erzielt `LaufendeSumme([Kennzahl])` oder `LaufendeSumme([Kennzahl];Zeile)` folgendes Ergebnis:

Kennzahl	LaufendeSumme	Kennzahl	LaufendeSumme
100	100	200	300
400	700	250	950

In einer Kreuztabelle ergibt `LaufendeSumme([Kennzahl];Spalte)` folgendes Ergebnis:

Kennzahl	LaufendeSumme	Kennzahl	LaufendeSumme
100	100	200	700
400	500	250	950

Weitere Informationen

[Prozent \[Seite 64\]](#)

[RunningAverage \[Seite 68\]](#)

[LaufendeAnzahl \[Seite 70\]](#)

[LaufendesMaximum \[Seite 72\]](#)

[LaufendesMinimum \[Seite 74\]](#)

[LaufendesProdukt \[Seite 76\]](#)

[LaufendeSumme \[Seite 78\]](#)

1.6.2.4.15 Selbst (Operator)

Der Operator `Selbst` ist mit der Funktion `vorherige` kompatibel.

Beschreibung

Verweist die Funktion `Vorherige` auf die vorherige Zelle, wenn sie kein Berichtsobjekt enthält.

Beispiele

`5 + Vorher(Selbst)` gibt die Sequenz 5, 10, 15, 20, 25, 30... zurück.

`1 + 0,5 * Vorher(Selbst)` gibt die Sequenz 1, 1,5; 1,75; 1,88... zurück.

Weitere Informationen

[Vorherige \[Seite 263\]](#)

1.6.2.4.16 Wo (Operator)

Beschreibung

Der Operator `wo` beschränkt die für die Berechnung einer Kennzahl verwendeten Daten.

Beispiele

Die Formel `Durchschnitt ([Umsatz]) Wo ([Land] = "USA")` berechnet den Durchschnittsumsatz, wenn das Land "USA" ist.

Die Formel `Durchschnitt ([Umsatz]) Wo ([Land] = "USA" Oder [Land] = "Frankreich")` berechnet den Durchschnittsumsatz, wenn das Land "USA" oder "Frankreich" ist.

Die Formel `[Umsatz] Wo (Nicht ([Land] InListe("USA"; "Frankreich")))` berechnet den Umsatz für andere Länder als die USA und Frankreich.

Die Variable `[Hoher Umsatz]` hat die Formel `[Umsatz] Wo [Umsatz > 500000]`. Wenn `[Hoher Umsatz]` in einem Block platziert wird, zeigt die Variable den Umsatz an, wenn der Wert größer als 500000 oder leer ist. Wenn die Formel `Durchschnitt ([Hoher Umsatz])` in einer Fußzeile unten in der Spalte `[Hoher Umsatz]` platziert ist, gibt sie den Durchschnitt aller Umsätze größer als 500000 zurück.

Weitere Informationen

[Und \(Operator\) \[Seite 273\]](#)

[Zwischen \(Operator\) \[Seite 274\]](#)

[InListe \(Operator\) \[Seite 275\]](#)

[Oder \(Operator\) \[Seite 273\]](#)

[Nicht \(Ungleich-Operator\) \[Seite 274\]](#)

1.6.2.5 Operatoren der erweiterten Syntax

Sie geben Eingabe- und Ausgabekontexte explizit mit Kontextoperatoren an.

In der folgenden Tabelle sind die Kontextoperatoren aufgelistet:

Operator	Beschreibung
In	Gibt eine explizite Liste mit Dimensionen an, die im Kontext verwendet werden sollen.
FürJeden	Fügt dem Standardkontext Dimensionen hinzu.
FürAlle	Entfernt Dimensionen aus dem Standardkontext.

Die Operatoren "FürAlle" und "FürJeden" sind nützlich, wenn Sie über einen Standardkontext verfügen, in dem zahlreiche Dimensionen aufgeführt sind. Es ist meistens einfacher, den Kontext unter Verwendung von "FürAlle" und "FürJeden" zu ergänzen oder zu reduzieren, als die Liste mithilfe des Operators "In" ausdrücklich zu definieren.

1.6.2.5.1 Der Kontextoperator "In"

Der Kontextoperator `In` gibt Dimensionen ausdrücklich in einem Kontext an.

Beispiel: Verwenden des Kontextoperators "In" zur ausdrücklichen Angabe der Dimensionen eines Kontextes

In dem folgenden Beispiel wird ein Bericht gezeigt, der die Objekte "Jahr" und "Umsatz" enthält. Ihr Datenprovider enthält darüber hinaus das Objekt "Quartal", allerdings fügen Sie diese Dimension nicht in den Block ein. Stattdessen möchten Sie eine zusätzliche Spalte einfügen, in der der Maximalumsatz pro Quartal für jedes Jahr gezeigt wird. Der Bericht sieht folgendermaßen aus:

Jahr	Umsatz	Maximaler vierteljährlicher Umsatz
2001	\$8,096,123.60	\$2,660,699.50
2002	\$13,232,246.00	\$4,186,120.00
2003	\$15,059,142.80	\$4,006,717.50

Woher die Werte der Spalte "Max. pro Quartal" stammen, können Sie nachvollziehen, indem Sie diesen Block mit dem Block vergleichen, der die Dimension "Quartal" enthält:

Jahr	Quartal	Umsatz
2001	Q1	\$2,660,699.50
2001	Q2	\$2,279,003.00
2001	Q3	\$1,367,841.00
2001	Q4	\$1,788,580.00
	Max:	\$2,660,699.50

Jahr	Quartal	Umsatz
	Q1	\$3,326,172.00
	Q2	\$2,840,651.00
	Q3	\$2,879,303.00
	Q4	\$4,186,120.00
	Max:	\$4,186,120.00

Jahr	Quartal	Umsatz
	Q1	\$3,742,989.00
	Q2	\$4,006,717.50
	Q3	\$3,953,395.00
	Q4	\$3,356,041.00
	Max:	\$4,006,717.50

Die Spalte Max. pro Quartal zeigt den höchsten vierteljährlich in jedem Jahr erwirtschafteten Umsatz. Q4 weist beispielsweise den höchsten Umsatz im Jahr 2002 auf. Die Spalte "Max. pro Quartal" zeigt den Umsatz von Q4 in der Zeile für 2002.

Unter Verwendung des Operators In lautet die Formel für Max. pro Quartal:

```
Max ([Umsatz] In ([Jahr];[Quartal])) In ([Jahr])
```

Mit dieser Formel wird der maximale Umsatz für jede Kombination (Jahr,Quartal) berechnet, und danach wird die Zahl nach Jahr ausgegeben.

ⓘ Hinweis

Da der Standardausgabekontext des Blocks "Jahr" ist, brauchen Sie den Ausgabekontext in dieser Formel nicht ausdrücklich zu definieren.

1.6.2.5.2 Der Kontextoperator "Für Jeden"

Der Operator FürJeden fügt Dimensionen zu einem Kontext hinzu.

Beispiel: Verwenden des Operators "FürJeden" zum Hinzufügen von Dimensionen zu einem Kontext

Die folgende Tabelle zeigt den maximalen Umsatz für jedes Quartal in einem Bericht, in dem die Dimension "Quartal" zwar enthalten, aber nicht in den Block eingeschlossen ist:

Jahr	Umsatz	Maximaler vierteljährlicher Umsatz
2001	8096123.60	2660699.50
2002	13232246.00	4186120.00
2003	15059142.80	4006717.50

Es ist möglich, eine Formel für die Spalte "Max. pro Quartal" zu erstellen, in der der Operator "FürJeden" nicht enthalten ist:

```
Max ([Umsatz] In ([Jahr];[Quartal])) In ([Jahr])
```

Mithilfe des Kontextoperators "FürJeden" können Sie über die folgende Formel dasselbe Ergebnis erzielen:

```
Max ([Umsatz] FürJeden ([Quartal])) In ([Jahr])
```

Warum? Da die Dimension "Jahr" der Standardeingabekontext im Block ist. Indem Sie den Operator "FürJeden" verwenden, fügen Sie die Dimension "Quartal" zum Kontext hinzu und definieren dadurch den Eingabekontext von ([Jahr];[Quartal]).

1.6.2.5.3 Der Kontextoperator "FürAlle"

Mit dem Kontextoperator `ForAll` werden Dimensionen aus einem Kontext entfernt.

Beispiel: Verwenden des Operators "FürAlle" zum Entfernen von Dimensionen aus einem Kontext

Sie verfügen über einen Bericht, der die Objekte "Jahr", "Quartal" und "Umsatz" enthält. Sie möchten eine Spalte hinzufügen, in der der Gesamtumsatz pro Jahr angezeigt wird, wie im folgenden Block dargestellt:

Year	Quarter	Sales revenue	Yearly Revenue
2004	Q1	\$2,660,700	\$8,096,124
2004	Q2	\$2,279,003	\$8,096,124
2004	Q3	\$1,367,841	\$8,096,124
2004	Q4	\$1,788,580	\$8,096,124
2005	Q1	\$3,326,172	\$13,232,246
2005	Q2	\$2,840,651	\$13,232,246
2005	Q3	\$2,879,303	\$13,232,246
2005	Q4	\$4,186,120	\$13,232,246
2006	Q1	\$3,742,989	\$15,059,143
2006	Q2	\$4,006,718	\$15,059,143
2006	Q3	\$3,953,395	\$15,059,143
2006	Q4	\$3,356,041	\$15,059,143

Um den Umsatz mit Bezug auf das Jahr zu berechnen, muss der Eingabekontext gleich (Jahr) sein. Die Standardeinstellung ist (Jahr; Quartal). Sie können "Quartal" aus dem Eingabekontext entfernen, indem Sie in der Formel `FürAlle ([Quartal])` angeben, was folgende Formel ergibt:

```
Summe([Umsatz] FürAlle ([Quartal]))
```

Sie können auch den Operator "In" verwenden, um dasselbe Ergebnis zu erzielen. In diesem Fall würde die Formel folgendermaßen aussehen:

```
Summe([Umsatz] In ([Jahr]))
```

Anstatt "Quartal" zu entfernen und "Jahr" zu belassen, wird in dieser Version der Formel "Jahr" ausdrücklich als Kontext definiert.

1.6.2.6 Mengenoperatoren

Mengenoperatoren werden für Elemente in hierarchischen Daten verwendet.

1.6.2.6.1 Bereichsoperator

Beschreibung

Der Bereichsoperator (:) gibt eine Menge von Elementen zwischen und einschließlich der beiden Elemente auf der gleichen Ebene zurück.

Syntax

`erstes_Element:letztes_Element`

Beispiele

`[Geografie]&[USA].[Kalifornien].[Los Angeles]:[Geografie]&[USA].[Kalifornien].[San Francisco]` gibt `[Los Angeles]`, `[San Diego]`, `[San Francisco]` zurück, wenn die Elemente auf der Ebene die folgende Reihenfolge haben ...`[Los Angeles]`, `[San Diego]`, `San Francisco`...

`Summe([Umsatz];{[Geografie]&[USA].[Kalifornien].[Los Angeles]:[Geografie]&[USA].[Kalifornien].[San Francisco]})` gibt den Gesamtumsatz für Los Angeles, San Diego und San Francisco zurück.

1.6.3 Erweiterte Syntax und Schlüsselwörter

Schlüsselwörter sind eine Art "Abkürzung", mit der Sie sich anhand der erweiterten Syntax auf Dimensionen beziehen können, die nicht ausdrücklich angegeben werden.

Mit diesen Schlüsselwörtern werden Berichte zukunftssicher. Wenn die Formeln keine hart codierten Verweise auf Dimensionen enthalten, bleiben sie auch dann funktionstüchtig, wenn Dimensionen zu dem Bericht hinzugefügt oder daraus entfernt werden.

Es gibt fünf Schlüsselwörter für die erweiterte Syntax: Bericht, Sektion, Gruppenwechsel, Block und Körper.

1.6.3.1 Das Schlüsselwort Block

In diesem Thema werden die Dimensionen beschrieben, auf die mit dem Schlüsselwort "Block" verwiesen wird. Dabei ist die Position relevant, an der das Schlüsselwort im Bericht verwendet wird. Das Schlüsselwort "Block" umfasst häufig dieselben Daten wie das Schlüsselwort "Sektion".

Der Unterschied besteht jedoch darin, dass das Schlüsselwort "Block" die auf einen Block angewendeten Filter im Gegensatz zu "Sektion" berücksichtigt.

Position	Verweis auf folgende Daten
In einem Block	Daten des ganzen Blocks, unter Nichtberücksichtigung von Gruppenwechseln und Berücksichtigung von Filtern
In einem Gruppenwechsel des Blocks (Kopf- oder Fußzeile)	Daten des ganzen Blocks, unter Nichtberücksichtigung von Gruppenwechseln und Berücksichtigung von Filtern
In einer Sektion (Kopfzeile, Fußzeile oder außerhalb eines Blocks)	Nicht anwendbar

Position	Verweis auf folgende Daten
Außerhalb der Blöcke oder Sektionen	Nicht anwendbar

Beispiel: Das Schlüsselwort Block

Sie verfügen über einen Bericht, der die Objekte Jahr, Quartal und Umsatz enthält. Der Bericht verfügt über eine Sektion, die auf "Jahr" basiert. Der Block wurde gefiltert, um das 3. und 4. Quartal auszuschließen.

2001			
Quartal	Umsatz	Mittelwert erstes Halbjahr	Jährlicher Mittelwert
Q1	€2.660.700	4.939.702,5	8.096.123,6
Q2	€2.279.003	4.939.702,5	8.096.123,6
Summe:	€4.939.703		

2002			
Quartal	Umsatz	Mittelwert erstes Halbjahr	Jährlicher Mittelwert
Q1	€3.326.172	6.166.823	13.232.246
Q2	€2.840.651	6.166.823	13.232.246
Summe:	€6.166.823		

2003			
Quartal	Umsatz	Mittelwert erstes Halbjahr	Jährlicher Mittelwert
Q1	€3.742.989	7.749.706,4	15.059.142,8
Q2	€4.006.718	7.749.706,4	15.059.142,8
Summe:	€7.749.706		

Die Spalte "Jährlicher Durchschnitt" verwendet folgende Formel:

```
Durchschnitt([Umsatz] In Sektion)
```

Die Spalte "Durchschnitt erstes Halbjahr" verwendet folgende Formel:

```
Durchschnitt ([Umsatz]) In Block
```

Das Schlüsselwort "Block" berücksichtigt den auf den Block angewendeten Filter.

1.6.3.2 Das Schlüsselwort Körper

In diesem Thema werden die Dimensionen beschrieben, auf die mit dem Schlüsselwort in einem Blockkörper verwiesen wird. Dabei ist die Position relevant, an der das Schlüsselwort im Bericht verwendet wird.

Position	Verweis auf folgende Daten
In einem Block	Daten im Block
In einem Gruppenwechsel des Blocks (Kopf- oder Fußzeile)	Daten im Block
In einer Sektion (Kopfzeile, Fußzeile oder außerhalb eines Blocks)	Daten im Block
Außerhalb der Blöcke oder Sektionen	Daten im Bericht

Beispiel: Das Schlüsselwort Körper

Sie verfügen über einen Bericht, der die Objekte Jahr, Quartal und Umsatz enthält, mit einem Gruppenwechsel bei Jahr. Der Bericht verfügt über eine Sektion, die auf "Jahr" basiert, und einen Gruppenwechsel für "Quartal".

Jahr	Quartal	Umsatz	Körper
2001	Q1	2,660,700	2,660,699.5
	Q2	2,279,003	2,279,003
	Q3	1,367,841	1,367,840.7
	Q4	1,788,580	1,788,580.4
2001		8,096,123.6	

Die Spalte "Körper" enthält die Formel

```
Summe ([Umsatz]) In Körper
```

Die Gesamtbeträge der Spalte Körper sind identisch mit jenen der Spalte Umsatz, da das Schlüsselwort Körper sich auf die im Block enthaltenen Daten bezieht. Durch Entfernen des Objekts Monat würden sich die Zahlen der Spalte Block ändern, um mit den geänderten Zahlen der Spalte Umsatz übereinstimmen zu können. Wenn Sie die Formel in den Berichtsfuß einfügen, würde dort der Gesamtumsatz dieses Körpers zurückgegeben werden.

1.6.3.3 Das Schlüsselwort Gruppenwechsel

In der folgenden Tabelle werden die Dimensionen beschrieben, auf die mit dem Schlüsselwort "Gruppenwechsel" verwiesen wird. Dabei ist die Position relevant, an der das Schlüsselwort im Bericht verwendet wird.

Position	Verweis auf folgende Daten
In einem Block	Daten in dem durch einen Gruppenwechsel eingeschränkten Teil des Blocks.
In einem Gruppenwechsel des Blocks (Kopf- oder Fußzeile)	Daten in dem durch einen Gruppenwechsel eingeschränkten Teil des Blocks
In einer Sektion (Kopfzeile, Fußzeile oder außerhalb eines Blocks)	Nicht anwendbar
Außerhalb der Blöcke oder Sektionen	Nicht zutreffend

Beispiel: Das Schlüsselwort Gruppenwechsel

Sie verfügen über einen Bericht, der Jahr, Quartal und Umsatz anzeigt:

Jahr	Quartal	Umsatz	Gesamt Gruppenwechsel
2001	Q1	\$2,660,700	\$8,096,124
	Q2	\$2,279,003	\$8,096,124
	Q3	\$1,367,841	\$8,096,124
	Q4	\$1,788,580	\$8,096,124

In dem Bericht wurde ein Gruppenwechsel auf Jahr angewendet. Die Spalte "Gesamtumsatz Gruppenwechsel" enthält folgende Formel:

```
Summe ([Umsatz]) In Gruppenwechsel
```

Ohne das Schlüsselwort Gruppenwechsel würde diese Spalte die Zahlen der Spalte Umsatz duplizieren, da sie den Standardausgabekontext ([Jahr]:[Quartal]) verwenden würde.

1.6.3.4 Das Schlüsselwort Bericht

In diesem Thema werden die Daten beschrieben, auf die mit dem Schlüsselwort "Bericht" verwiesen wird. Dabei ist die Position relevant, an der das Schlüsselwort im Bericht verwendet wird:

Position	Verweis auf folgende Daten
In einem Block	Alle Daten des Berichts
In einem Gruppenwechsel des Blocks (Kopf- oder Fußzeile)	Alle Daten des Berichts
In einer Sektion (Kopfzeile, Fußzeile oder außerhalb eines Blocks)	Alle Daten des Berichts
Außerhalb der Blöcke oder Sektionen	Alle Daten des Berichts

Beispiel: Das Schlüsselwort Bericht

Sie verfügen über einen Bericht, der die Objekte Jahr, Quartal und Umsatz enthält. Der Bericht enthält die Spalte "Gesamtumsatz Bericht", in der die Gesamtsumme aller Umsatzwerte des Berichts angezeigt wird.

Jahr	Quartal	Umsatz	Gesamtumsatz Bericht
2001	Q1	€2.660.700	36.387.512,4
2001	Q2	€2.279.003	36.387.512,4
2001	Q3	€1.367.841	36.387.512,4
2001	Q4	€1.788.580	36.387.512,4
2002	Q1	€3.326.172	36.387.512,4
2002	Q2	€2.840.651	36.387.512,4
2002	Q3	€2.879.303	36.387.512,4
2002	Q4	€4.186.120	36.387.512,4
2003	Q1	€3.742.989	36.387.512,4
2003	Q2	€4.006.718	36.387.512,4
2003	Q3	€3.953.395	36.387.512,4
2003	Q4	€3.356.041	36.387.512,4

Die Formel für die Spalte "Gesamtumsatz Bericht" sieht wie folgt aus:

```
Summe([Umsatz]) In Bericht
```

Ohne das Schlüsselwort Bericht würde diese Spalte die Zahlen der Spalte Umsatz duplizieren, da sie den Standardausgabekontext ([Jahr];[Quartal]) verwenden würde.

1.6.3.5 Das Schlüsselwort Sektion

In diesem Thema werden die Daten beschrieben, auf die mit dem Schlüsselwort für eine Sektion verwiesen wird. Dabei ist die Position relevant, an der das Schlüsselwort im Bericht verwendet wird.

Position	Verweis auf folgende Daten
In einem Block	Alle Daten der Sektion
In einem Gruppenwechsel des Blocks (Kopf- oder Fußzeile)	Alle Daten der Sektion
In einer Sektion (Kopfzeile, Fußzeile oder außerhalb eines Blocks)	Alle Daten der Sektion
Außerhalb der Blöcke oder Sektionen	Nicht anwendbar

Beispiel: Das Schlüsselwort Sektion

Sie verfügen über einen Bericht, der die Objekte Jahr, Quartal und Umsatz enthält.

2001

Quartal	Umsatz	Gesamtumsatz Sektion
Q1	€2.660.700	8.096.123,6
Q2	€2.279.003	8.096.123,6
Q3	€1.367.841	8.096.123,6
Q4	€1.788.580	8.096.123,6

Der Bericht verfügt über eine Sektion, die auf "Jahr" basiert. Die Spalte "Gesamtumsatz Sektion" enthält folgende Formel:

```
Summe ([Umsatz]) In Sektion
```

Die in der Spalte "Gesamtumsatz Sektion" angezeigte Zahl entspricht dem Gesamtumsatz von 2001, da der Sektionsgruppenwechsel auf das Objekt "Jahr" angewendet wurde. Ohne das Schlüsselwort Sektion würde diese Spalte die Zahlen der Spalte Umsatz duplizieren, da sie den Standardausgabekontext ([Jahr];[Quartal]) verwenden würde.

1.6.4 Runden und Abschneiden von Zahlen

Verschiedene Funktionen enthalten einen Parameter, der festlegt, ob der zurückgegebene Wert gerundet oder abgeschnitten werden soll.

Für den Parameter können Ganzzahlen eingegeben werden, die größer als, gleich oder kleiner als 0 sind. In der folgenden Tabelle wird veranschaulicht, wie Zahlen in diesen Fällen gerundet und abgeschnitten werden:

Parameter	Beschreibung
> 0	<p>Die Funktion rundet/schneidet bis auf <Parameter> Dezimalstellen auf/ab.</p> <p>Beispiele:</p> <p>Runden (3 , 13 ; 1) gibt 3,1 zurück.</p> <p>Runden (3 , 157 ; 2) gibt 3,16 zurück.</p>
0	<p>Die Funktion rundet/schneidet bis zur nächsten Ganzzahl auf/ab.</p> <p>Beispiele:</p>

Parameter	Beschreibung
	Abschneiden(3,7;0) gibt 3 zurück.
	Abschneiden(4,164;0) gibt 4 zurück.
< 0	<p>Die Funktion rundet/schneidet bis zur nächsten 10 (Parameter = -1), 100 (Parameter = -2), 1000 (Parameter = -3) usw. auf/ab.</p> <p>Beispiele:</p> <p>Runden(123,76;-1) gibt 120 zurück.</p> <p>Runden(459,9;-2) gibt 500 zurück.</p> <p>Abschneiden(1600;3) gibt 1000 zurück.</p>

ⓘ Hinweis

Zahlen werden intern als doppelt genaue Gleitkommazahlen dargestellt und weisen 15 bis 17 Dezimalstellen auf.

Weitere Informationen

[Runden \[Seite 202\]](#)

[Abschneiden \[Seite 208\]](#)

[InEuroKonvertieren \[Seite 188\]](#)

[AusEuroKonvertieren \[Seite 187\]](#)

[AusEuroRundungsfehler \[Seite 190\]](#)

[InEuroRundungsfehler \[Seite 192\]](#)

1.6.5 Verweisen auf Elemente und Elementmengen in Hierarchien

Sie verweisen auf Elemente und Elementmengen in Funktionen mit der Syntax:
[hierarchy]&path.function.

Pfad und Funktion sind optional. In Pfad verweisen Sie auf die einzelnen Elemente in eckigen Klammern, wobei die Elemente durch Punkte getrennt sind. Bei den Namen von Elementen und Ebenen muss die Groß-/Kleinschreibung beachtet werden.

ⓘ Hinweis

Sie verwenden Elementmengen zum Überschreiben des Standardberechnungskontexts für eine Hierarchie. In Funktionen, die Elementmengen akzeptieren, schließen Sie die Elementmenge zwischen {} ein.

Sie verweisen auf Elementbereiche, indem Sie einen Doppelpunkt (:) zwischen dem ersten und letzten Element verwenden und den vollständigen Pfad für jedes Element angeben. Ein Bereich umfasst alle Elemente, die sich auf der gleichen Ebene wie die angegebenen befinden.

Ein Beispiel für die Bereichssyntax ist: `[Vertriebshierarchie]&[Kundentyp].[UNTERNEHMEN];[Groß].[Nancy Davolio]:[Vertriebshierarchie]&[Kundentyp].[UNTERNEHMEN];[Groß].[Andrew Smith]`.

Beispiel: Verweisen auf Elemente und Elementmengen

Sie haben folgende Ausgangshierarchie:

Vertriebshierarchie	Auftragssumme
Kundentyp	277,290,434
UNTERNEHMEN	180,063,361
Groß	113.905,997
Nancy Davolio	44,855,689
Janet Leverling	44,050,308
Andrew Smith	30,000,000
WELTWEIT	91,157,363

- `[Vertriebshierarchie]&[Kundentyp].[UNTERNEHMEN].[Groß].Untergeordnete Elemente` verweist auf die Elemente `[Nancy Davolio]`, `[Janet Leverling]` und `[Andrew Smith]`.
- `Summe([Auftragssumme];{[Vertriebshierarchie]&[Kundentyp].[UNTERNEHMEN].[Groß].Untergeordnete Elemente})` gibt 113.905.997 zurück (die Summe der Kennzahl für die drei untergeordneten Elemente).
- `[Vertriebshierarchie]&[Kundentyp].[UNTERNEHMEN].[Groß].[Janet Leverling]` verweist auf das Element `[Janet Leverling]`.
- `Summe([Auftragssumme];{[Vertriebshierarchie]&[Kundentyp].[UNTERNEHMEN].[Groß].[Janet Leverling];[Vertriebshierarchie]&[Kundentyp].[UNTERNEHMEN].[Groß].[Nancy Davolio]})` gibt 88.905.997 zurück (die Summe der Kennzahl für die drei untergeordneten Elemente).
- `[Vertriebshierarchie]&[Kundentyp].[UNTERNEHMEN].[Groß].[Nancy Davolio]:[Vertriebshierarchie]&[Kundentyp].[UNTERNEHMEN].[Groß].[Andrew Smith]` verweist auf die Elemente `[Nancy Davolio]`, `[Janet Leverling]` und `[Andrew Smith]`.
- `Summe([Auftragssumme];{[Vertriebshierarchie]&[Kundentyp].[UNTERNEHMEN].[Groß].[Nancy Davolio]:[Vertriebshierarchie]&[Kundentyp].[UNTERNEHMEN].[Groß].[Andrew Smith]})` gibt 113.905.997 zurück (die Summe der Kennzahl für die drei Elemente im Bereich).
- `[Vertriebshierarchie].Untergeordnete Elemente` verweist auf alle Elemente in der Hierarchie `[Vertriebshierarchie]`.
- `Summe([Auftragssumme];{Vertriebshierarchie].Untergeordnete Elemente})` gibt 277.290.434 zurück.

1.7 Definieren benutzerdefinierter Funktionen

1.7.1 Übersicht über externe Funktionen

Berechnungserweiterungen sind benutzerdefinierte Web Intelligence-Reporting-Berechnungen, die die Liste der vorhandenen Web Intelligence-Funktionen erweitern.

Um die Bibliothek der erweiterten Berechnungsfunktionen zu verwenden, erstellen Sie anhand eines spezifischen API eine externe C++-Bibliothek.

1.7.1.1 Externe Funktionen

Externe Funktionen sind wie andere Web Intelligence-Standardfunktionen sichtbar und verwendbar. Sie können eine Formel mit Funktionen definieren, die Ihre eigene Logik implementieren.

ⓘ Hinweis

Sie können beliebig viele Funktionen definieren. Es werden nur Funktionen, die einzelne Werte für Parameter verwenden, unterstützt. Sie können maximal fünf einzelne Werte für Parameter verwenden.

So definieren Sie eine Funktion:

1. Deklarieren Sie in einer XML-Datei die Beschreibung der externen Funktion anhand einer bestimmten XML-Struktur.
2. Implementieren Sie die Funktion in einer C++-Bibliothek unter Verwendung einer bestimmten API.
3. Kopieren Sie die XML-Datei und die XML-Bibliothek in den entsprechenden Ordner im Business Objects Enterprise-Installationsverzeichnis für den Server und den Desktop-Client.
4. Führen Sie einen Neustart des Systems durch, um die externe Funktion automatisch zu der für die Erstellung von Formeln verfügbaren Funktionsliste hinzuzufügen.

Die externe Funktion besitzt eine eindeutige Kennung, um zu verhindern, dass sie bei Verwendung in einem Bericht falsch interpretiert werden kann, falls eine andere externe Bibliothek verwendet wird.

Wenn das System keine Bibliothek laden kann oder Informationen für eine externe Funktion fehlen, oder im Falle einer inkonsistenten XML-Deklaration, einer fehlenden Bibliothek oder einer doppelten Funktion, wird eine Fehlermeldung ausgegeben. Außerdem schreibt das System Fehler in das Ereignisprotokoll.

Weitere Informationen

[#EXTERNAL-Fehlermeldung \[Seite 308\]](#)

1.7.1.2 Implementieren von benutzerdefinierten Funktionen

Zur Implementierung von benutzerdefinierten Funktionen müssen einige manuelle Schritte vorgenommen werden. Der BusinessObjects-Administrator muss die XML-Datei und die zugehörige DLL-Bibliotheksdatei im Bibliotheksordner des Servers sowie auf jedem Rechner, auf dem der Rich Client (Desktop) installiert ist, ablegen.

⚠ Achtung

Das Ersetzen oder Hinzufügen einer Bibliothek in einem Ordner mit benutzerdefinierten Bibliotheken kann das System gefährden. Da die Bibliothek automatisch geladen wird, kann eine externe Bibliothek auf kritische interne Daten oder Prozesse zugreifen und dadurch das System in Gefahr bringen.

Stellen Sie sicher, dass der Standortadministrator den Zugriff auf den betreffenden Ordner ausreichend sichert, damit nur berechtigte Nutzer auf den benutzerdefinierten Ordner zugreifen können.

1.7.1.3 Bibliotheksdeklaration

Die Bibliotheksdateierweiterungen hängen vom Betriebssystem ab:

- DLL für Windows
- SO für Linux oder UNIX

Die Dateitypen lauten:

Typ	Beschreibung
XML-Katalogdeklaration	Es gibt nur eine Datei dieses Typs. Sie sollte <code>externalcatalogs.xml</code> genannt werden. Die Datei enthält die Liste aller XML-Funktionsdefinitionsdateien.
XML-Funktionsdeklaration	In der Datei ist eine Liste von Funktionen und ihre zugehörige Bibliothek definiert; sie ist in der XML-Katalogdeklarationsdatei aufgeführt.
<div><h3>📘 Hinweis</h3><p>Die Katalogdatei kann die Funktionsdeklarationsbibliotheken entweder enthalten oder referenzieren.</p></div>	
Bibliotheksdatei	Diese Datei enthält den Code C++ für die Anwenderfunktionen. Die Bibliotheksdatei enthält die Anwenderfunktionsimplementierung gemäß Definition in der XML-Funktionsdeklaration.

1.7.1.4 Web-Intelligence-Beispieldateien verwenden

Stellen Sie sicher, dass die folgenden Anwendungen installiert sind:

- Visual Studio C++ VS2015 oder höher
- Web Intelligence 4.1 oder höher

In den Beispielen in diesem Dokument werden die Beispieldateien aus der Datei `Samples.zip` verwendet, die sich im Verzeichnis `[Installationsverzeichnis]\userlibs\WebI\Samples\` befindet.

1. Entpacken Sie die Datei `Samples.zip`.
2. Starten Sie `OpenSolution.bat`, um die Beispiele zu öffnen.

Die Datei `OpenSolution.bat` legt die temporäre `<WEBICALCPLUGINAPI>`-Umgebungsvariable fest, die die Lösung zum Auffinden der spezifischen Web Intelligence-Header-Dateien benötigt.

⚠ Achtung

Wenn die erforderliche Web Intelligence-Version nicht installiert ist, kann die Datei `OpenSolution.bat` nicht verwendet werden. Ist dies der Fall, legen Sie die `<WEBICALCPLUGINAPI>`-Umgebungsvariable manuell auf den Pfad des Ordners mit den Web Intelligence-Header-Dateien fest, und öffnen Sie `Samples\WebICalcPlugin`.

Weitere Informationen

[Beispiele \[Seite 307\]](#)

1.7.2 Definieren von benutzerdefinierten Berechnungen

So erstellen Sie eine benutzerdefinierte Funktion in Web Intelligence:

1. Definieren Sie die XML-Funktionsdeklaration.
2. Definieren Sie die XML-Katalogdeklaration.
3. Implementieren Sie die Bibliothek in C++ anhand des spezifischen API für die externe Funktion.
4. Kompilieren Sie die Quelldatei.
5. Kopieren Sie die XML-Definition und die Bibliothek in den entsprechenden `webIcalcPlugin`-Ordner (auf dem Server und auf dem Rich Client).
6. Starten Sie den Web Intelligence-Server neu.

ℹ Hinweis

In den Beispielen im Kapitel werden die mit Web Intelligence gelieferten Beispieldateien verwendet.

Das System fügt die Funktion automatisch zu der Funktionsliste im Formel-Editor sowie zur Kontexthilfe der Formelleiste hinzu.

Wenn eine Formel eine Funktion verwendet, für die keine externe Bibliothek zur Verfügung steht, wird die `#EXTERNAL`-Fehlermeldung angezeigt.

ℹ Hinweis

Es werden nur Funktionen, die einzelne Werte für Parameter verwenden, unterstützt. Tabellenparameter werden beispielsweise nicht unterstützt.

1.7.2.1 XML-Funktionsobjekte

Die XML-Definition enthält Objekte zur Definition der benutzerdefinierten Funktion. Benutzerdefinierte XML-Funktionen erweitern die Funktionsliste der Formelsprache so, dass Formel, die diese Funktion verwenden, über ihre XML-Signatur analysiert und in Form von Tokens verwendet werden können. Sie ordnen der externen Funktion eine globale eindeutige ID (GUID) zu, damit sie nicht wiederverwendet oder mit anderen benutzerdefinierten Bibliotheken verwechselt werden kann.

Die XML-Definition enthält folgende Objekte:

Tag	XML-Attribut	XML-Definitionsobjekt
<CATALOG>		Der XML-Stamm
<LIBRARY>	Datei	<p>Der Name der Bibliotheksdatei, die den C++-Implementierungscode enthält</p> <p>Die Bibliotheksdatei kann mehrere Funktionen enthalten. Die Dateierweiterung der Bibliothek sollte nicht angegeben werden.</p>
<FUNCTION>	GUID	<p>Die eindeutige Funktions-GUID</p> <div><p>→ Tipp</p><p>Definieren Sie alle GUIDs im Voraus, und stellen Sie sicher, dass alle GUIDs global eindeutig sind.</p></div> <p>Für Windows können Sie das GUID-Tool verwenden, das im Lieferumfang von Visual Studio enthalten ist, oder es von der Microsoft-Website herunterladen. Für Linux befindet sich das Tool <code>usr/bin/uuidgen</code> im <code>libuuid1</code> (Debian)-Paket.</p>
	Name	<p>Der Funktionsname, der im Formel-Editor angezeigt wird</p> <p>Der Funktionsname</p> <ul style="list-style-type: none">• muss ein einfacher, eindeutiger Name für die Funktion sein• muss mit einem Buchstaben beginnen• muss Klein- und Großbuchstaben, Ziffern oder das Zeichen <code>_</code> enthalten• darf nicht bereits in einer Web Intelligence-Bibliothek enthalten sein <div><p>ⓘ Hinweis</p><p>Der Name wird nicht in eine andere Sprache übersetzt.</p></div>
<ARGLIST>		<p>Liste der Parameter</p> <p>Die Anzahl der Parameter sollte fünf nicht überschreiten.</p>

Tag	XML-Attribut	XML-Definitionsobjekt
<ARG>	Typ	<p>Die Parametertypen</p> <p>Es gibt folgende mögliche Parametertypen:</p> <ul style="list-style-type: none"> • Numerisch • Boolesch • Datum • Zeichenfolge
	Name	<p>Der Name der einzelnen Parameter, wie er im Formel-Editor angezeigt werden sollte</p> <p>Der Name zeigt dem Benutzer den Prototyp der Methode. Es dürfen nur alphanumerische Zeichen verwendet werden.</p>
<RETURN>	Typ	<p>Der Rückgabewert-Typ</p> <p>Rückgabewerte können sein:</p> <ul style="list-style-type: none"> • Numerisch • Boolesch • Datum • Zeichenfolge
	Typ	<p>Die Kategorie, in der die Funktion im Formel-Editor erscheint</p> <p>Achten Sie auf Konsistenz; setzen Sie Zeichenfolgen in die Zeichenkategorie und Zahlen in die numerische Kategorie. Folgende Kategorien sind verfügbar:</p> <ul style="list-style-type: none"> • Zeichen • Datum • Dokument • DP • Sonstige • Logisch • Num
<HINT>	Wert	<p>Ein im Formel-Editor angezeigter Hinweis</p> <p>Der Hinweis erklärt die Verwendung der Funktion.</p>

1.7.2.2 Definieren der XML-Funktionsdeklaration

Die XML für die Signatur verwendet folgende Struktur:

```
Function_list
```

Die XML für die Signatur verwendet folgende Struktur:

```
Function_list := [Function*]  
Function := [name, GUID, data_type = Numeric|Boolean|Date|  
String, category = character|Date|Document|DP|Misc|Logical|Num,  
parameter_list, (online_help_signature?),  
(online_help_description?),library_name)]  
parameter_list := [parameter*]  
parameter := [name, data_type =Numeric|Boolean|Date|String]
```

1. Setzen Sie das XML-Stammtag auf CATALOG.
2. Fügen Sie LIBRARY-Tags zu CATALOG hinzu.
3. Fügen Sie den Namen der Bibliotheksdatei ohne die Dateierweiterung DLL oder SO zu LIBRARY hinzu. Dies ist das Dateiattribut.
4. Fügen Sie FUNCTION-Tags zu LIBRARY hinzu.

Ein FUNCTION-Tag sollte eine eindeutige GUID und einen zusätzlichen, eindeutigen Attributnamen besitzen, der den Namen der Funktion definiert.

Das FUNCTION-Tag sollte Folgendes beinhalten:

- Ein ARGLIST-Tag mit ARG-Tags. Die ARG-Tags sollten einen ersten Attributtyp aufweisen, der den Typ dieses Parameters bestimmt sowie ein zweites Attribut, das den Namen dieses Parameters definiert. Der ARG-Typ kann "Boolean", "Numeric", "Date" oder "String" sein. Der ARG-Name enthält nur alphanumerische Zeichen.

ⓘ Hinweis

Sie dürfen maximal fünf Parameter verwenden.

- Ein RETURN-Tag, das ein Typattribut definiert. Der RETURN-Typ kann "Boolean", "Numeric", "Date" oder "String" sein.
 - Ein CATEGORY-Tag, das ein Typattribut definiert. Der CATEGORY-Typ kann "Character", "Date", "Document", "DP", "Misc", "Logical" oder "Num" sein.
 - Ein HINT-Tag, das ein Wertattribut definiert.
5. Fügen Sie die XML-Definition in den entsprechenden Ordner (auf dem Server und Rich Client) ein.

Beispiel: SampleMath.xml

```
<CATALOG>  
  <LIBRARY file="SampleMath">  
    <FUNCTION guid="CC3E9742-67A7-4844-9DBF-2CCD4F6ECABE" name="MySquareFct">  
      <ARGLIST>  
        <ARG type="Numeric" name="input_number"/>  
      </ARGLIST>  
      <RETURN type="Numeric"/>  
      <CATEGORY type="Num"/>  
      <HINT value="My square function."/>  
    </FUNCTION>  
  </LIBRARY>  
</CATALOG>
```

Weitere Informationen

[Web-Intelligence-Beispieldateien verwenden \[Seite 299\]](#)

1.7.2.3 Definieren der XML-Katalogdeklaration

Sie können eine XML-Katalogdeklaration erstellen oder sie zu einer vorhandenen Katalogdeklaration hinzufügen.

<CATALOG> referenziert eine XML-Funktionsdeklarationsdatei; der <CATALOG> kann jedoch auch direkt definiert werden, wie aus dem Abschnitt zur Definition eines XML-Funktionsformats ersichtlich.

Erstellen einer Katalogdeklaration

1. Geben Sie der Deklaration den Namen `externalcatalogs.xml`.
2. Setzen Sie das XML-Stammtag auf `CATALOGS`.
3. Fügen Sie zu den `CATALOGS` `CATALOG`-Tags hinzu.

Hiermit wird der Wert für den Dateinamen der XML-Funktionsdeklarationen definiert.

4. Fügen Sie die XML-Bibliothek in den entsprechenden Ordner (auf dem Server und Rich Client) ein.

Beispiel: `externalcatalogs.xml`

```
<CATALOGS>
  <CATALOG file="SampleMath.xml" />
</CATALOGS>
```

Weitere Informationen

[Web-Intelligence-Beispieldateien verwenden \[Seite 299\]](#)

1.7.2.4 Implementieren der C++-Datei

1. Fügen Sie den Header `ibovariant.h` in die Datei ein.
2. Beginnen Sie die Deklaration für jede Methode mit dem Makro `BO_DECLARE_USER_FCT`.

Das Makro enthält:

- den Funktionsnamen, wie er in der XML-Funktionsdeklarationsdatei angezeigt wird.
- den Rückgabewert-Objektnamen
- den Parameter-Objektnamen

Hinweis

Die Funktion gibt BONOERROR zurück, wenn keine Fehler auftreten, andernfalls wird die #EXTERNAL-Fehlermeldung im Bericht angezeigt.

Beispiel: Square.cpp

```
// Headers file include of the WebI headers
#include <ibovariant.h>
// To not repeat BOExtFunc::
using namespace BOExtFunc;
BO_DECLARE_USER_FCT (// Name of function as it was defined in the XML.
                    MySquareFct,
                    // Name of the return value object.
                    retVal,
                    // Name of the parameters object.
                    parameters
)
{
    try // Always used a try{}catch(...) to be sure no
        // exception was thrown outside this Web
        // Intelligence user function.
    {
        // Get the first parameter.
        const iBOValue&param0 = parameters[0];
        // Transform the parameter to the correct type.
        double valPar0(param0);
        // Assign value to the return value.
        retVal = valPar0 * valPar0;
    }
    catch(...)
    {
        return BOERROR; // Unknown exception so notify WebI
    }
    return BONOERROR; // It's OK
}
```

Weitere Informationen

[Web-Intelligence-Beispieldateien verwenden \[Seite 299\]](#)

1.7.2.5 Kompilieren der Quelldatei in Microsoft Visual Studio 2015

1. Um ein Projekt zu erstellen, wählen Sie **Datei > Neu > Projekt**.
2. Unter **Projekttypen** wählen Sie **Visual C++ > Allgemein** aus.
3. Wählen Sie unter **Vorlagen Leeres Projekt**.

4. Geben Sie den Namen des Projekts ein.
5. Geben Sie den Zielordner des Projekts ein.
6. Klicken Sie auf [OK](#).
7. Klicken Sie mit der rechten Maustaste auf das Projekt, und wählen Sie [Eigenschaften](#).
8. Wählen Sie unter [Konfiguration Alle Konfigurationen](#) aus.
9. Wählen Sie unter [Konfigurationseigenschaften](#) [Allgemein](#) den [Konfigurationstyp Dynamische Bibliothek \(.dll\)](#).
10. Klicken Sie auf [OK](#).
11. Klicken Sie mit der rechten Maustaste auf das Projekt, und wählen Sie [Hinzufügen](#) [Neues Element](#).
12. Wählen Sie unter [Kategorie Code](#).
13. Wählen Sie unter [Vorlage](#) die Option [C++-Datei \(.cpp\)](#) aus.
14. Geben Sie den Namen der CPP-Datei ein.
15. Klicken Sie auf [Hinzufügen](#).
16. Klicken Sie mit der rechten Maustaste auf das Projekt, und wählen Sie [Eigenschaften](#).
17. Wählen Sie unter [Konfiguration Alle Konfigurationen](#) aus.
18. Fügen Sie unter [Konfigurationseigenschaften](#) [C/C++](#) [Zusätzliche Includeverzeichnisse](#) den Ordner hinzu, der die BusinessObjects-Dateiheader enthält.
19. Klicken Sie auf [Übernehmen](#).
20. Wählen Sie unter [Konfiguration Debuggen](#).
21. Legen Sie unter [Konfigurationseigenschaften](#) [C/C++](#) [Codegenerierung](#) die [Laufzeitbibliothek](#) auf [Multithreaded-DLL \(/MD\)](#) fest.

Hinweis

Wenn auf Ihrem Computer Microsoft Visual Studio installiert ist, können Sie [Multithreaded-Debug-DLL \(/MDd\)](#) anstelle von [Multithreaded-DLL \(/MD\)](#) verwenden, um von der Debugging-Umgebung zu profitieren.

22. Klicken Sie auf [Übernehmen](#).
23. Wählen Sie unter [Konfiguration Release](#).
24. Legen Sie unter [Konfigurationseigenschaften](#) [C/C++](#) [Codegenerierung](#) die [Laufzeitbibliothek](#) auf [Multithreaded-DLL \(/MD\)](#) fest.

Hinweis

Wenn auf Ihrem Computer Microsoft Visual Studio installiert ist, können Sie [Multithreaded-Debug-DLL \(/MDd\)](#) anstelle von [Multithreaded-DLL \(/MD\)](#) verwenden, um von der Debugging-Umgebung zu profitieren.

25. Klicken Sie auf [OK](#).
26. Fügen Sie den Code zu der CPP-Datei hinzu.
27. Führen Sie die Kompilierung durch.

1.7.2.6 Dateien in das WebiCalcPlugin kopieren

Kopieren Sie die XML-Funktionsdeklaration, die XML-Katalogdeklaration und die DLL/SO-Datei in den Ordner WebiCalcPlugin.

Der Ordner ist unter folgendem Pfad verfügbar:

```
[Installationsverzeichnis]\[BusinessObjects-Version]\[BS]_[PLATTFORM]
\WebiCalcPlugin
```

Dabei gilt Folgendes: [BusinessObjects-Version] ist die Version des Produkts, z. B. BusinessObjects Enterprise XI 4.0, [BS] ist das Betriebssystem, z. B. win32 (Windows-Betriebssystem) oder linux (Linux-Betriebssystem), und [PLATTFORM] ist die Plattform, z. B. x86 auf einer Intel-32-Bit-CPU.

1.7.3 Beispiele

In den Beispielen werden die Beispieldateien aus der Datei `Samples.zip` verwendet, die sich im Verzeichnis `[Installationsverzeichnis]\userlibs\WebI\Samples\` befindet.

Beispiel: XML-Katalogdeklaration für die `externalcatalogs.xml`

```
<CATALOGS>
  <CATALOG file="SampleString.xml"/>
</CATALOGS>
```

Beispiel: XML-Funktionsdeklaration in `SampleString.xml`

```
<CATALOG>
  <LIBRARY file="SampleString">
    <FUNCTION guid="A91BD526-B8EB-4b09-90F2-FFCD350776A8" name="MyHelloWorld">
      <RETURN type="String"/>
      <CATEGORY type="Num"/>
      <HINT value="My simple hello world function."/>
    </FUNCTION>
  </LIBRARY>
</CATALOG>
```

Beispiel: C++-Dateideklaration in `HelloWorld.cpp`

```
// Headers file include of the Web Intelligence headers
#include <ibovariant.h>
// To not repeat BOExtFunc::
using namespace BOExtFunc;
```

```

BO_DECLARE_USER_FCT(
    // Name of function as it was defined in the XML.
    MyHelloWorld,
    // Name of the return value object.
    retVal
    // Don't use parameter.
    /*parameters*/
)
{
    try // Always used a try{}catch(...) to be sure no
        // exception was thrown outside this
        // Web Intelligence user function.
    {
        // Create an std::wstring with wide char Hello world.
        std::wstring helloWorldStr = L"Hello world!!!";
        // Initialyse the return value.
        retVal = helloWorldStr;
    }
    catch(...)
    {
        // Unkonwn exception so notify Web Intelligence
        return BOERROR;
    }
    return BONOERROR; // It's OK
}

```

Weitere Informationen

[Web-Intelligence-Beispieldateien verwenden \[Seite 299\]](#)

1.7.4 #EXTERNAL-Fehlermeldung

Die #EXTERNAL-Fehlermeldung wird durch folgende Probleme ausgelöst:

- Eine Formel verweist auf eine externe Funktion, die sich nicht im externen Bibliotheksordner befindet.
- Ein Dokument enthält eine externe Methode, und das System kann diese nicht laden. Die Bibliotheksdatei wurde nicht gefunden, oder es ist eine inkonsistente Deklaration vorhanden.
- Der Rückgabewert wird von einer externen Methode nicht initialisiert.
- Der Rückgabewert wurde von einer externen Methode mit einem fehlerhaften Typ initialisiert. Beispielsweise wurde ein Wert irrtümlicherweise vom Typ "Doppelt" auf den Typ "Zeichenfolge" gesetzt.
- Eine externe Methode gibt einen Fehlercode zurück.

Bitten Sie den BusinessObjects-Administrator, die richtige Bibliothek zum Implementieren dieser Funktion bereitzustellen.

1.7.5 Ereignisprotokoll-Fehlermeldungen

Wenn bei einer XML-Analyse/Validierung ein Fehler auftritt, wird dem Benutzer eine Meldung angezeigt, und in den Ereignisprotokollen werden entsprechende Fehlermeldungen erstellt.

Protokolltyp	Fehlermeldungen
XML-Protokolle	<p>File cannot be read or is missing.</p> <p>Bad XML structure due to:</p> <ul style="list-style-type: none"> • Parent/Children relation invalid. • Missing field (ID function, name function). • Invalid field value.
DLL-Protokolle	<p>File is missing.</p> <p>DLL cannot be loaded.</p> <p>Function is not found in the DLL.</p>
Funktionsprotokolle	<p>Function name is already in use.</p> <p>Function ID is already used.</p> <p>Function name is missing.</p> <p>Return type is invalid.</p> <p>ID is invalid.</p> <p>Number of parameters is invalid.</p>
Parameterprotokolle	<p>Parameter name is missing.</p> <p>Parameter type is invalid.</p>
Laufzeitprotokolle	<p>The user function does not initialize the return value.</p> <p>The user function initializes the return value with a bad type.</p> <p>The user function returns the BOERROR error code.</p>

1.8 Fehlerbehebung bei Formeln

1.8.1 Mechanismus zum automatischen Umschreiben von Formeln

Bedingt durch die Abfolge der Releases zur korrektiven Wartung für Web Intelligence, kann es bei den Berechnungsergebnissen zu Unterschieden zwischen den Versionen kommen.

Ab Version 4.1 SP3 bietet Web Intelligence einen Mechanismus zum automatischen Umschreiben von Formeln, der ausgewählte Formeln (siehe untenstehende Liste) in einem Dokument automatisch ändert, das aus einer Vorgängerversion migriert wurde. Diese Formeln folgen einem bestimmten Muster. Nach der Änderung geben die Formeln dasselbe Ergebnis zurück wie vor der Berechnungsänderung. Daher ist es empfehlenswert, das Dokument mit den Änderungen zu speichern und somit den Vorgang der Formelumschreibung abzuschließen.

Der Mechanismus zum automatischen Umschreiben von Formeln ist standardmäßig für Dokumente verfügbar, die nach BI 4.1 SP3 und höher migriert wurden, und zwar für folgende Formelmuster:

1. Operator vom Typ Where() mit einer Dimension als Parameter in einer Bedingung,
2. laufende Berechnungen mit Zurücksetzen in Abschnitten,
3. laufende Berechnungen mit Zurücksetzen in Kreuztabellen.

Diese Regelliste wird möglicherweise für zukünftige Releases um weitere Formelmuster erweitert.

Regel(1)

In früheren Versionen wurden die Daten auf eine bestimmte Weise berechnet, wenn Sie einen Operator vom Typ Where() mit einer Dimension als Parameter in einer Bedingung verwendeten. Die Dimension wurde dem Kennzahlkontext hinzugefügt. Regel(1) reproduziert das frühere Verhalten.

Diese Regel gilt für jedes Dokument, das von den Versionen XI 3.1 FP3.6, XI 3.1 FP4.1, XI 3.1 FP5.1 und 4.0 SP5 migriert wurde.

Regel(2)

In früheren Versionen wurden laufende Berechnungen in Abschnitten nicht korrekt ausgeführt, da die Berechnungen bei jeder Sektionsinstanz zurückgesetzt wurden. Regel(2) reproduziert das frühere Verhalten.

Diese Regel gilt für jedes Dokument, das von Version XI R2 SP4 migriert wurde.

Regel(3)

In früheren Versionen waren laufende Berechnungen mit Zurücksetzen in Kreuztabellen Berechnungen, die nach einem "N"-Muster (Spalte für Spalte) statt nach einem "Z"-Muster (Zeile für Zeile) ausgeführt wurden.

Regel(3) führte das Schlüsselwort FORCE_COL ein, mit dem Web Intelligence gezwungen wird, Berechnungen nach einem "N"-Muster auszuführen.

Durch Regel(3) wird die Formel RunningSum([Sales revenue];([State])) z. B. gezwungen, eine Berechnung Spalte für Spalte auszuführen, wenn sie in RunningSum([Sales revenue];FORCE_COL;([State])) geändert wird.

Diese Regel gilt für jedes Dokument, das von einer der folgenden Versionen migriert wurde: XI 3.x, 4.0 Patch 2.20, 4.0 SP5, 4.0 SP6, 4.0 SP7, 4.1 und 4.1 SP1.

1.8.2 Formelfehler- und Informationsmeldungen

Mit der bedingten Formatierung können Sie Berichtsdaten formatieren, die Fehlermeldungen zurückgeben.

In bestimmten Fällen kann eine Formel keinen Wert zurückgeben und gibt stattdessen eine Fehler- oder Informationsmeldung zurück, die mit "#" beginnt. Die Meldung wird in der Zelle angezeigt, in der sich die Formel befindet.

1.8.2.1 #BERECHNUNG

Der Fehler #BERECHNUNG tritt auf, wenn eine Ausschnittsdimension, die in der Funktion `RelativerWert` angegeben ist, im Berechnungskontext des Blocks, in dem die Funktion platziert ist, nicht mehr zur Verfügung steht.

Der Fehler #BERECHNUNG tritt auch dann auf, wenn ein zusammengeführtes Objekt, das eine Hierarchie enthält, in einem Bericht eingeschlossen ist.

Der Fehler #BERECHNUNG bezieht sich also auf fehlerhafte Verwendung von Kontextoperatoren in einer Formel.

Weitere Informationen

[RelativerWert \[Seite 268\]](#)

1.8.2.2 #KONTEXT

#KONTEXT wird in einer Kennzahl angezeigt, wenn die Kennzahl einen nicht vorhandenen Berechnungskontext aufweist.

#KONTEXT steht in Bezug zu den Fehlermeldungen #INKOMPATIBEL und #DATENSYNCH, die in Dimensionen auftreten, wenn ein Block einen nicht vorhandenen Berechnungskontext aufweist.

Im Fall von #INKOMPATIBEL ist der Kontext nicht vorhanden, da die Dimensionen inkompatibel sind. Im Fall von #DATENSYNCH ist der Kontext nicht vorhanden, da die Dimensionen aus mehreren nicht synchronisierten Datenprovidern stammen.

Beispiel: Nicht-vorhandener Berechnungskontext in einer Abfrage

Wenn ein auf dem Universum "Insel-Reisen" basierender Block die Objekte Reservierungsjahr und Umsatz enthält, wird die Fehlermeldung #KONTEXT ausgegeben, da es nicht möglich ist, den Umsatz nach dem Reservierungsjahr zu verdichten. (Die Reservierungen haben noch keinen Umsatz erbracht.)

1.8.2.3 #DATENSYNCH

#DATENSYNCH wird angezeigt, wenn Sie eine Dimension aus einem unterschiedlichen Datenprovider in einem Block platzieren, der Dimensionen aus einem anderen Datenprovider enthält, und die beiden Datenprovider nicht durch eine zusammengeführte Dimension synchronisiert wurden.

#DATENSYNCH wird in allen Dimensionen des Blocks und #KONTEXT in den Kennzahlen angezeigt.

Beispiel: Dimensionen aus unterschiedlichen Daten Providern in einem Block

Wenn ein auf dem Universum "Insel-Reisen" basierender Bericht Datenprovider mit den Objekten (Jahr, Umsatz) und (Quartal) enthält, wird in den Spalten "Jahr" und "Quartal" eines Blocks, in dem "Jahr", "Quartal" und "Umsatz" enthalten sind, der Fehler #DATENSYNCH angezeigt, da die beiden Datenprovider nicht über eine zusammengeführte Dimension synchronisiert wurden.

1.8.2.4 #DIV/O

Der Fehler #DIV/O wird angezeigt, wenn eine Formel versucht, eine Zahl durch Null zu teilen. Vom mathematischen Standpunkt aus gesehen ist dies unmöglich.

Null kann niemals als Divisor verwendet werden.

Beispiel: Bestimmen des Umsatzes pro Artikel

Sie verfügen über einen Bericht, in dem der Umsatz, die Anzahl verkaufter Artikel und der Umsatz pro Artikel angezeigt werden. (Dieser kann berechnet werden, indem Sie den Umsatz durch die Anzahl verkaufter Artikel dividieren.)

Das letzte Quartal war für Sie ein regelrechter Misserfolg, da Sie keinerlei Umsatz verzeichnen konnten. Die Spalte "Umsatz pro Artikel" gibt für dieses Quartal den Wert #DIV/O zurück, da die Formel versucht, den Umsatzwert durch die Artikelanzahl Null zu teilen.

1.8.2.5 #FEHLER

#FEHLER ist die Standardfehlermeldung, die alle Fehler abdeckt, die nicht in den vorher aufgeführten Fehlermeldungskategorien enthalten sind.

1.8.2.6 #EXTERN

#EXTERN wird angezeigt, wenn eine Formel auf eine externe Funktion verweist, die in Web Intelligence nicht zur Verfügung steht.

1.8.2.7 #INKOMPATIBEL

#INKOMPATIBEL wird angezeigt, wenn ein Block inkompatible Objekte enthält.

Beispiel: Inkompatible Objekte in einer Abfrage

Wenn ein auf dem Universum "Insel-Reisen" basierender Block die Dimensionen Jahr und Reservierungsjahr enthält, wird in den Spalten, die diese Dimensionen enthalten, #INKOMPATIBEL angezeigt, da diese beiden Objekte nicht kompatibel sind.

1.8.2.8 #MIX

#MIX tritt auf, wenn eine aggregierte Kennzahl mehrere Einheiten hat.

Beispiel: In einer Zelle wird #MIX angezeigt, wenn aggregierte Währungswerte in mehreren Währungen angegeben sind.

1.8.2.9 #MEHRFACHWERT

#MEHRFACHWERT wird angezeigt, wenn Sie eine Formel, die mehrere Werte zurückgibt, in eine Zelle einfügen, die zur Ausgabe eines einzigen Werts vorgesehen ist.

Beispiel: Mehrfachwert in einer Zelle

Sie verfügen über einen Bericht, der die Objekte Land, Erholungsort und Umsatz anzeigt. Sie fügen eine Zelle in den Bericht ein, die die Formel [Umsatz] Für Jeden ([Land]) enthält. Diese Zelle gibt "#MEHRFACHWERT" zurück, da "Land" im Bericht über zwei Werte verfügt: "USA" und "Frankreich".

Der Umsatz der beiden Länder (USA und Frankreich) kann nicht in einer einzigen Zelle ausgegeben werden. Außerhalb einer Tabelle kann eine Zelle mit Umsatzangaben die einzelnen Werte nur auf eine ganz bestimmte Art verdichten (zum Beispiel durch Summieren oder Mitteln der Werte).

Wenn der Bericht in Sektionen pro "Land" unterteilt wird, ist die Formel innerhalb einer Sektion korrekt, da es in einer Sektion nur einen einzigen Wert pro "Land" gibt. Außerhalb einer Sektion hingegen gibt diese Formel weiterhin die Fehlermeldung "#MEHRFACHWERT" aus.

1.8.2.10 #NV

Wenn ein Bericht einen Wert für eine Zelle enthält, der auf einem Wert eines Berichts basiert, der in der zugrunde liegenden Datenbank nicht verfügbar ist (beispielsweise ein BW-Fehler in einer BEx-Zelle), zeigt die Zelle #NV (nicht verfügbar) an. Dies bedeutet, dass die Zelle leer ist, da die Daten nicht abgerufen werden können.

1.8.2.11 #ÜBERLAUF

#ÜBERLAUF wird angezeigt, wenn eine Berechnung einen Wert zurückgibt, der aufgrund seiner Größe nicht in der Software verarbeitet werden kann.

Ein solcher Wert kann in potenziert Form wie 1.7E308 vorliegen (1,7 gefolgt von 307 Nullen).

1.8.2.12 #TEILERGEBNIS

#TEILERGEBNIS tritt auf, wenn nicht alle mit einem Berichtsobjekt verknüpften Zeilen abgerufen wurden.

Wenn #TEILERGEBNIS häufig in Berichten angezeigt wird und Sie über die erforderlichen Sicherheitsrechte verfügen, ändern Sie die Abfrageeigenschaft `Max. abzurufende Zeilen`, um das Abrufen weiterer Daten zu ermöglichen. Falls Sie nicht zum Ändern der Abfrage berechtigt sind, wenden Sie sich an den BI-Administrator.

Wenn der Bericht intelligente Kennzahlen enthält, wird #TEILERGEBNIS wahrscheinlich häufiger angezeigt, da bei intelligenten Kennzahlen größere Datenmengen abgerufen werden müssen als bei klassischen Kennzahlen.

1.8.2.13 #RANG

#RANG wird beim Versuch angezeigt, Daten auf der Grundlage eines Objekts, das von der Reihenfolge der Werte abhängig ist, in eine Rangfolge zu stellen.

Objekte, die die Funktion `vorherige` bzw. laufende Aggregatfunktionen verwenden, sind von der Reihenfolge der Werte abhängig.

Durch das Erstellen einer Rangfolge werden die Werte dieser Objekte neu berechnet, wodurch die Rangfolge geändert wird. Dies führt zu einer zirkulären Abhängigkeit. Eine solche Abhängigkeit kann beispielsweise entstehen, wenn Sie eine Rangfolge im Dialogfeld "Rang" erstellen oder die `Rang`-Funktion verwenden.

Beispiel: Erstellen von Rangfolgen für Werte des laufenden Durchschnitts oder vorherige Werte

Wenn Sie versuchen, eine Rangfolge für einen Block auf der Grundlage einer Spalte zu erstellen, die die `vorher`-Funktion oder eine laufende Aggregatfunktion enthält, wird vom gesamten Block `#RANG` zurückgegeben.

1.8.2.14 #REKURSIV

`#REKURSIV` tritt auf, wenn aufgrund einer zirkulären Abhängigkeit keine Berechnung durchgeführt werden kann.

Beispiel: Nutzung der Funktionen `NumberOfPages()`, `Page()` und `PageInSection()`

Wenn Sie die Funktionen `NumberOfPages`, `Page` und `PageInSection` in eine Zelle einfügen, deren Eigenschaften für die automatische Anpassung der Höhe und Breite festgelegt sind, gibt die Zelle `#REKURSIV` zurück, da das Einfügen dieser Formeln in eine Zelle mit automatischer Anpassung eine zirkuläre Abhängigkeit erzeugt. Diese Funktionen benötigen die exakte Größe des Berichts, um einen Wert zurückzugeben. Allerdings wird die Größe der Zelle, die Auswirkungen auf die Größe des Berichts hat, vom Zelleninhalt bestimmt.

1.8.2.15 #REGENERIEREN

`#REGENERIEREN` wird in Berichtszellen angezeigt, deren Werte von Objekten abgeleitet wurden, die von einer Abfrage entfernt und dann erneut zu ihr hinzugefügt wurden.

Objekte werden von einer Abfrage entfernt, wenn die Abfrageeigenschaft [Abfragenentfernung aktivieren](#) ausgewählt wird und die Objekte nichts zu auf der Abfrage basierenden Berichten beitragen.

Wenn die Abfrage regeneriert wird, werden die Zellen erneut mit Werten von den Objekten aufgefüllt.

1.8.2.16 #SICHERHEIT

`#SICHERHEIT` wird angezeigt, wenn Sie versuchen, eine Funktion zu verwenden, für die Sie keine Sicherheitsrechte besitzen.

Beispiel: Verwenden der Funktion DatenproviderSQL()

Wenn ein Benutzer, der nicht berechtigt ist, Datenprovider-SQL anzeigen zu lassen, die Funktion DatenproviderSQL() in eine Zelle einfügt, wird die Meldung #SICHERHEIT in der Zelle ausgegeben.

1.8.2.17 #SYNTAX

#SYNTAX wird angezeigt, wenn eine Formel auf ein Objekt verweist, das nicht mehr im Bericht vorhanden ist.

Beispiel: Verweis auf ein nicht-vorhandenes Objekt

Sie verfügen über einen Bericht, der ursprünglich die Objekte Jahr, Quartal und Umsatz anzeigte. Sie haben eine Spalte hinzugefügt, in der die Differenz zwischen dem realen Umsatz und dem jährlichen Durchschnittsumsatz angezeigt wird. Diese Zahl wird anhand der Variablen "Differenz Jahresdurchschnitt" berechnet.

Wenn die Variable Differenz Jahresdurchschnitt aus dem Bericht gelöscht wird, wird von der Spalte, die die Variable enthält, #SYNTAX zurückgegeben.

1.8.2.18 #ZUREGENERIEREN

#ZUREGENERIEREN wird in Zellen angezeigt, die auf intelligenten Kennzahlen basieren, wenn der von der intelligenten Kennzahl zurückgegebene Wert nicht verfügbar ist.

Diese Situation tritt auf, wenn die Gruppierungsmenge, die den Wert enthält, im Datenprovider nicht verfügbar ist.

Um den Fehler #ZUREGENERIEREN zu entfernen, regenerieren Sie die Daten.

Einige der Kennzahlen werden "delegiert" (bei BW gilt dies für eine Kennzahl, die nicht mit SUMME aggregiert). Wenn Sie eine Tabelle oder Berechnung für eine Kennzahl definieren, wird diese Kennzahl im spezifischen Kontext der Aggregation abgefragt (die Kennzahl wird für eine Menge von Dimensionen angegeben). Wenn diese Dimensionsmenge eine Untermenge der Abfragedimensionsmenge ist, muss die Kennzahl entlang der vorgegebenen Dimensionsmenge (oder Gruppierungsmenge, die sich auf eine Gruppieren-nach-Klausel in SQL bezieht) aggregiert werden.

Bei regulären Kennzahlen führt das System die Aggregation durch, bei delegierten Kennzahlen wird diese Aggregation an die zugrunde liegende Datenbank delegiert. Das System muss diese Datenbank dazu erneut abfragen. Da dieser Vorgang nicht automatisch erfolgt, zeigt das System #ZUREGENERIEREN an und wartet auf die Regenerierung durch den Benutzer. Bei der Regenerierung führt das System die zusätzliche Abfrage durch, um die angeforderte Aggregation abzurufen, und ersetzt #ZUREGENERIEREN dann durch den entsprechenden Wert.

1.8.2.19 #NICHTVERFÜGBAR

#NICHTVERFÜGBAR wird angezeigt, wenn der Wert einer intelligenten Kennzahl nicht berechnet werden kann.

Dieser ist der Fall, wenn die Werte in einer gefilterten intelligenten Kennzahl nicht angezeigt werden können, ohne einen Filter auf die Abfrage anzuwenden. Da dies Auswirkungen auf andere, auf der gleichen Abfrage basierende Berichte haben kann, wird kein Filter angewendet.

1.9 Vergleichen von Werten mithilfe von Funktionen

1.9.1 Vergleichen von Werten mithilfe der Funktion "Vorherige"

Die Funktion `Vorherige` gibt zu Vergleichszwecken einen vorherigen Wert eines Ausdrucks zurück.

Der zurückgegebene Wert hängt vom Layout des Berichts ab.

Weitere leistungsstarke Vergleichsfunktionen stehen über die Funktion `RelativerWert` zur Verfügung. `RelativerWert` gibt zu Vergleichszwecken einen vorhergehenden oder nachfolgenden Wert eines Ausdrucks zurück. Der zurückgegebene Wert hängt nicht vom Layout des Berichts ab.

Weitere Informationen

[Vorherige \[Seite 263\]](#)

[RelativerWert \[Seite 268\]](#)

[Vergleichen von Werten mithilfe der Funktion RelativerWert. \[Seite 317\]](#)

1.9.2 Vergleichen von Werten mithilfe der Funktion RelativerWert.

Die Funktion `RelativerWert` gibt Vergleichswerte eines Ausdrucks zurück. Die Funktion gibt diese Werte unabhängig vom Layout des Berichts zurück.

Wenn Sie `RelativerWert` verwenden, geben Sie Folgendes an:

- Der Ausdruck, dessen Vergleichswert Sie suchen (der Ausdruck muss eine Kennzahl oder ein Detail einer im Block verfügbaren Dimension sein)
- Die Liste der Ausschnittsdimensionen
- Der Offset

Die Funktion verwendet die Ausschnittsdimensionen, den Offset und die Unterachsendimensionen (die von den Ausschnittsdimensionen impliziert werden), um einen Vergleichswert zurückzugeben. Die

Unterachsendimensionen sind alle anderen Dimensionen im Berechnungskontext mit Ausnahme der Ausschnittsdimensionen.

Allgemein ausgedrückt gibt `RelativerWert` den Wert des Ausdrucks in der Zeile zurück, die in der Werteliste der Ausschnittsdimensionen um die `offset`-Zeilen von der aktuellen Zeile entfernt ist, und wobei die Werte der Unterachsendimensionen denen der aktuellen Zeile entsprechen.

Hinweis

Alle Ausschnittsdimensionen müssen stets im Berechnungskontext des Blocks stehen, in dem die Funktion platziert ist. Wenn in der Folge eine Ausschnittsdimension entfernt wird, gibt die Funktion den Fehler `#BERECHNUNG` zurück.

Beispiel

In diesem Beispiel enthält die Zeile `RelativerWert` folgende Formel:

```
RelativerWert([Umsatz];([Jahr]);-1)
```

- Der Ausdruck ist `[Umsatz]`;
- Die Ausschnittsdimension ist `[Jahr]`;
- Der Offset beträgt -1 (die Funktion gibt den direkten vorherigen Wert in der Liste zurück).

Jahr	Quartal	Vertriebsmitarbeiter	Umsatz	RelativerWert
2007	Q1	Schmidt	1000	
2007	Q2	Müller	2000	
2007	Q3	Wilson	1500	
2007	Q4	Meier	3000	
2008	Q1	Schmidt	4000	1000
2008	Q2	Müller	3400	2000
2008	Q3	Krause	2000	1500
2008	Q4	Meier	1700	3000

Als Geschäftsfrage ausgedrückt gibt die Formel den vom selben Vertriebsmitarbeiter generierten Umsatz im selben Quartal des vorherigen Jahrs zurück.

Als Berechnung in Worten ausgedrückt gibt die Formel die Werte von `[Umsatz]` (der Ausdruck) in der Zeile zurück, in der der Wert von `[Jahr]` (die Ausschnittsdimension) der vorherige Wert aus der Werteliste des Objekts `[Jahr]` ist, und wobei Werte von `[Quartal]` und `[Vertriebsmitarbeiter]` (die Unterachsendimensionen) denen in der aktuellen Zeile entsprechen.

Weitere Informationen

[RelativerWert \[Seite 268\]](#)

1.9.2.1 Ausschnittsdimensionen und die Funktion "RelativerWert"

Die Funktion `RelativerWert` verwendet die Werteliste der Ausschnittsdimensionen, um die Zeile mit den Vergleichswerten zu suchen.

Die Funktion gibt den Vergleichswert des Ausdrucks zurück, der in der Funktion, die sich um die `offset`-Anzahl der Zeilen von der Liste der Ausschnittsdimensionen entfernt befindet, angegeben wird.

Daher ist die Sortierreihenfolge der Ausschnittsdimensionen wichtig für die Bestimmung der Funktionsausgabe.

Beispiel: Mehrere Ausschnittsdimensionen

In der folgenden Tabelle enthält die Spalte "RelativerWert" die folgende Formel:

```
RelativerWert([Umsatz];([Jahr];[Quartal]);-1)
```

- Der Ausdruck ist `[Umsatz]`;
- Die Ausschnittsdimensionen sind `([Jahr];[Quartal])`;
- Der Offset beträgt -1 (die Funktion gibt den direkten vorherigen Wert in der Liste zurück).

Jahr	Quartal	Vertriebsmitarbeiter	Umsatz	RelativerWert
2007	Q1	Schmidt	1000	
2007	Q2	Schmidt	2000	
2007	Q3	Schmidt	1500	
2007	Q4	Schmidt	3000*	
2007	Q1	Müller	4000	
2007	Q2	Müller	3400	
2007	Q3	Müller	2000	
2007	Q4	Müller	1700	
2008	Q1	Schmidt	5000**	3000*
2008	Q2	Schmidt	3000***	5000**
2008	Q3	Schmidt	2700****	3000***
2008	Q4	Schmidt	6800	2700****

Als Geschäftsfrage ausgedrückt gibt die Formel den vom selben Vertriebsmitarbeiter generierten Umsatz des vorherigen Quartals zurück.

Als Berechnung in Worten ausgedrückt gibt die Formel den Wert von `[Umsatz]` in der Zeile zurück, in der die Werte von `[Jahr]` und `[Quartal]` den vorherigen Wert in der `([Jahr];[Quartal])`-Werteliste, und in der der Wert von `[Vertriebsmitarbeiter]` dem Wert in der aktuellen Zeile entspricht.

Die Funktion verwendet die Werteliste der Ausschnittsdimensionen, um die Zeile mit dem Vergleichsumsatz zu suchen:

Jahr	Quartal	
2007	Q1	
2007	Q2	
2007	Q3	
2007	Q4	*
2008	Q1	**
2008	Q2	***
2008	Q3	****
2008	Q4	

Die Sortierreihenfolge der Ausschnittsdimensionen bestimmt die Ausgabe der Funktion. Das * in den Tabellen weist auf die Sortierreihenfolge hin.

Weitere Informationen

[RelativerWert \[Seite 268\]](#)

1.9.2.2 Ausschnittsdimensionen und Sektionen

Eine Ausschnittsdimension kann sich in der führenden Sektionszelle eines Berichts befinden.

Beispiel: Ausschnittsdimension in einer Sektionszelle

In der folgenden Tabelle enthält die Spalte "RelativerWert" die folgende Formel:

```
RelativerWert([Umsatz];([Jahr];[Quartal]);-1)
```

2007

Quartal	Vertriebsmitarbeiter	Umsatz	RelativerWert
Q1	Schmidt	1000	
Q2	Schmidt	2000	
Q3	Schmidt	1500	
Q4	Schmidt	3000*	
Q1	Müller	4000	
Q2	Müller	3400	

Quartal	Vertriebsmitarbeiter	Umsatz	RelativerWert
Q3	Müller	2000	
Q4	Müller	1700	

2008

Quartal	Vertriebsmitarbeiter	Umsatz	RelativerWert
Q1	Schmidt	5000**	3000*
Q2	Schmidt	3000***	5000**
Q3	Schmidt	2700****	3000***
Q4	Schmidt	6800	2700****

Die Funktion verwendet die Werteliste der Ausschnittsdimensionen, um die Zeile mit dem Vergleichsumsatz zu suchen:

Jahr	Quartal	
2007	Q1	
2007	Q2	
2007	Q3	
2007	Q4	*
2008	Q1	**
2008	Q2	***
2008	Q3	****
2008	Q4	

Die Sortierreihenfolge der Ausschnittsdimensionen bestimmt die Ausgabe der Funktion. Das * in den Tabellen weist auf die Sortierreihenfolge hin.

Weitere Informationen

[RelativerWert \[Seite 268\]](#)

1.9.2.3 Reihenfolge der Ausschnittsdimensionen

Da die Sortierreihenfolge der Werteliste der Ausschnittsdimensionen die Ausgabe von `RelativerWert` bestimmt, wirkt sich die Reihenfolge, in der die Ausschnittsdimensionen angegeben sind, auf die Ausgabe der Funktion aus.

Beispiel: Reihenfolge der Ausschnittsdimensionen

In der folgenden Tabelle enthält die Spalte "RelativerWert" die folgende Formel:

```
RelativerWert([Umsatz];([Jahr];[Quartal]);-1)
```

Jahr	Quartal	Vertriebsmitarbeiter	Umsatz	RelativerWert
2007	Q1	Schmidt	1000	
2007	Q2	Schmidt	2000	
2007	Q3	Schmidt	1500	
2007	Q4	Schmidt	3000*	
2007	Q1	Müller	4000	
2007	Q2	Müller	3400	
2007	Q3	Müller	2000	
2007	Q4	Müller	1700	
2008	Q1	Schmidt	5000**	3000*
2008	Q2	Schmidt	3000***	5000**
2008	Q3	Schmidt	2700****	3000***
2008	Q4	Schmidt	6800	2700****

Als Geschäftsfrage ausgedrückt gibt die Formel den vom selben Vertriebsmitarbeiter generierten Umsatz des vorherigen Quartals zurück.

Die Sortierreihenfolge der Ausschnittsdimensionen lautet wie folgt:

Jahr	Quartal
2007	Q1
2007	Q2
2007	Q3
2007	Q4
2008	Q1
2008	Q2
2008	Q3
2008	Q4

Die Funktion wird geändert in:

```
RelativerWert([Umsatz];([Quartal];[Jahr]);-1)
```

Die Sortierreihenfolge der Ausschnittsdimensionen wird zu:

Quartal	Jahr
Q1	2007

Quartal	Jahr	
Q1	2008	**
Q2	2007	***
Q2	2008	****
Q3	2007	*****
Q3	2008	*****
Q4	2007	*****
Q4	2008	*****

Die Sortierreihenfolge hat folgende Auswirkungen auf das Funktionsergebnis:

Jahr	Quartal	Vertriebsmitarbeiter	Umsatz	RelativerWert
2007	Q1	Schmidt	1000*	
2007	Q2	Schmidt	2000***	
2007	Q3	Schmidt	1500*****	
2007	Q4	Schmidt	3000*****	
2007	Q1	Müller	4000	
2007	Q2	Müller	3400	
2007	Q3	Müller	2000	
2007	Q4	Müller	1700	
2008	Q1	Schmidt	5000**	1000*
2008	Q2	Schmidt	3000****	2000***
2008	Q3	Schmidt	2700*****	1500*****
2008	Q4	Schmidt	6800*****	3000*****

Als Geschäftsfrage ausgedrückt gibt die Formel nun den vom selben Vertriebsmitarbeiter generierten Umsatz im selben Quartal des vorherigen Jahrs zurück.

Die Änderung der Sortierreihenfolge der Abschnittdimension ändert die Bedeutung der Formel. Das * in den Tabellen weist auf die Sortierreihenfolge hin.

Weitere Informationen

[RelativerWert \[Seite 268\]](#)

1.9.2.4 Ausschnittsdimensionen und Sortierungen

Da die Sortierreihenfolge der Werteliste der Ausschnittsdimensionen die Funktionsausgabe bestimmt, wirkt sich eine auf eine Dimension in den Ausschnittsdimensionen angewendete Sortierung auf die Funktionsausgabe aus.

Beispiel: Auf eine Ausschnittsdimension angewendete benutzerdefinierte Sortierung

In der folgenden Tabelle enthält die Spalte "RelativerWert" die folgende Formel:

```
RelativerWert([Umsatz];([Jahr];[Quartal]);-1)
```

Eine benutzerdefinierte Sortierung (Q1, Q2, Q4, Q3) wird auf [Quartal] angewendet und gibt folgendes Ergebnis für die Funktion zurück:

Jahr	Quartal	Vertriebsmitarbeiter	Umsatz	RelativerWert
2007	Q1	Schmidt	1000	
2007	Q2	Schmidt	2000	
2007	Q4	Schmidt	3000	
2007	Q3	Schmidt	1500*	
2007	Q1	Müller	4000	
2007	Q2	Müller	3400	
2007	Q4	Müller	1700	
2007	Q3	Müller	2000	
2008	Q1	Schmidt	5000**	1500*
2008	Q2	Schmidt	3000***	5000**
2008	Q4	Schmidt	6800****	3000***
2008	Q3	Schmidt	2700	6800****

Die sortierte Liste der Ausschnittsdimension lautet wie folgt:

Jahr	Quartal
2007	Q1
2007	Q2
2007	Q4
2007	Q3
2008	Q1
2008	Q2
2008	Q4

Jahr	Quartal
2008	Q3

Das * in den Tabellen weist auf die Sortierreihenfolge hin.

Weitere Informationen

[RelativerWert \[Seite 268\]](#)

1.9.2.5 Verwenden von "RelativerWert" in Kreuztabellen

Die Funktion `RelativerWert` funktioniert in Kreuztabellen auf dieselbe Weise wie in vertikalen Tabellen.

Das Layout der Daten in einer Kreuztabelle hat keine Auswirkungen auf die Funktionsausgabe.



Weitere Informationen

[RelativerWert \[Seite 268\]](#)

Ausschlussklauseln und rechtliche Aspekte

Hyperlinks

Einige Links werden durch ein Symbol und/oder einen Quick-Info-Text klassifiziert. Über diese Links erhalten Sie weitere Informationen. Informationen zu den Symbolen:

- Links zum Symbol : Sie rufen eine Website auf, die nicht von SAP gehostet wird. Durch die Nutzung solcher Links stimmen Sie Folgendem zu (sofern sich nicht aus Ihren Vereinbarungen mit SAP etwas anderes ergibt):
 - Der Inhalt der verlinkten Site ist keine SAP-Dokumentation. Basierend auf diesen Informationen ergibt sich für Sie keinerlei Produkthaftungsanspruch gegen SAP.
 - Weder widerspricht SAP dem Inhalt auf der verlinkten Site noch stimmt SAP ihm zu. Außerdem übernimmt SAP keine Gewährleistung für dessen Verfügbarkeit und Richtigkeit. SAP übernimmt keine Haftung für Schäden, die durch die Nutzung solchen Inhalts verursacht wurden, es sei denn, dass diese Schäden von SAP grob fahrlässig oder vorsätzlich verursacht wurden.
- Links zum Symbol : Sie verlassen die Dokumentation für das jeweilige SAP-Produkt oder den jeweiligen SAP-Service und rufen eine von SAP gehostete Website auf. Durch die Nutzung solcher Links stimmen Sie zu (sofern sich nicht aus Ihren Vereinbarungen mit SAP etwas anderes ergibt), dass sich basierend auf diesen Informationen für Sie keinerlei Produkthaftungsanspruch gegen SAP ergibt.

Videos, die auf externen Plattformen gehostet werden

Einige Videos verweisen möglicherweise auf Video-Hosting-Plattformen von Drittanbietern. SAP kann die zukünftige Verfügbarkeit von Videos, die auf diesen Plattformen gespeichert sind, nicht garantieren. Außerdem unterliegen alle Werbungen und anderen Inhalte, die auf diesen Plattformen gehostet werden (z.B. empfohlene Videos oder Navigation zu anderen gehosteten Videos auf derselben Site), nicht der Kontrolle oder Verantwortlichkeit von SAP.

Beta und andere experimentelle Funktionen

Experimentelle Funktionen sind nicht Teil des offiziellen Lieferumfangs, den SAP für künftige Releases garantiert. Dies bedeutet, dass experimentelle Funktionen von SAP jederzeit, aus beliebigen Gründen und ohne vorherige Ankündigung geändert werden können. Experimentelle Funktionen sind nicht zur Nutzung in einem Produktivsystem vorgesehen. Die experimentellen Funktionen dürfen nicht für Demonstrationen, Tests, Untersuchungen, Bewertungen oder anderweitige Zwecke in einer Produktivumgebung oder in Verbindung mit Daten, die nicht ausreichend gesichert wurden, verwendet werden. Der Zweck der experimentellen Funktionen besteht darin, frühzeitig Feedback zu erhalten und so Kunden und Partnern die Möglichkeit zu geben, das zukünftige Produkt entsprechend zu beeinflussen. Durch die Abgabe von Feedback (z.B. über SAP Community) stimmen Sie zu, dass die geistigen Eigentumsrechte der Beiträge oder daraus abgeleiteten Werke im ausschließlichen Besitz von SAP verbleiben.

Beispielcode

Bei dem Quelltext und/oder den Code-Snippets handelt es sich ausschließlich um beispielhafte Darstellungen. Sie sind nicht zur Nutzung in einem Produktivsystem vorgesehen. Der Beispielcode dient ausschließlich dem Zweck, Syntax- und Verphrasierungsregeln besser zu erläutern und zu visualisieren. SAP übernimmt keine Gewährleistung für die Richtigkeit und Vollständigkeit des Beispielcodes. SAP übernimmt keine Haftung für Fehler oder Schäden, die durch die Nutzung des Beispielcodes verursacht wurden, es sei denn, dass diese Fehler oder Schäden von SAP grob fahrlässig oder vorsätzlich verursacht wurden.

Vorurteilsfreie Sprache

SAP unterstützt eine Kultur der Vielfalt und Integration. Wann immer möglich, verwenden wir in unserer Dokumentation unvoreingenommene Sprache, um auf Menschen aller Kulturen, ethnischen Zugehörigkeiten, Geschlechter und Fähigkeiten zu verweisen.

© 2024 SAP SE oder ein SAP-Konzernunternehmen Alle Rechte vorbehalten.

Weitergabe und Vervielfältigung dieser Publikation oder von Teilen daraus sind, zu welchem Zweck und in welcher Form auch immer, ohne die ausdrückliche schriftliche Genehmigung durch SAP SE oder ein SAP-Konzernunternehmen nicht gestattet. In dieser Publikation enthaltene Informationen können ohne vorherige Ankündigung geändert werden.

Die von SAP SE oder deren Vertriebsfirmen angebotenen Softwareprodukte können Softwarekomponenten auch anderer Softwarehersteller enthalten. Produkte können länderspezifische Unterschiede aufweisen.

Die vorliegenden Unterlagen werden von der SAP SE oder einem SAP-Konzernunternehmen bereitgestellt und dienen ausschließlich zu Informationszwecken. Die SAP SE oder ihre Konzernunternehmen übernehmen keinerlei Haftung oder Gewährleistung für Fehler oder Unvollständigkeiten in dieser Publikation. Die SAP SE oder ein SAP-Konzernunternehmen steht lediglich für Produkte und Dienstleistungen nach der Maßgabe ein, die in der Vereinbarung über die jeweiligen Produkte und Dienstleistungen ausdrücklich geregelt ist. Keine der hierin enthaltenen Informationen ist als zusätzliche Garantie zu interpretieren.

SAP und andere in diesem Dokument erwähnte Produkte und Dienstleistungen von SAP sowie die dazugehörigen Logos sind Marken oder eingetragene Marken der SAP SE (oder von einem SAP-Konzernunternehmen) in Deutschland und verschiedenen anderen Ländern weltweit. Alle anderen Namen von Produkten und Dienstleistungen sind Marken der jeweiligen Firmen.

Zusätzliche Informationen zur Marke und Vermerke finden Sie auf der Seite <https://www.sap.com/germany/about/legal/trademark.html>.