



PUBLIC (公開)

SAP BusinessObjects Intelligence Suite

ドキュメントバージョン: 4.1 Support Package 7 – 2015/12/11

Web Intelligence における関数、式、計算の使用

目次

1	ドキュメント履歴:Web Intelligence の関数、式および計算.....	5
2	このガイドについて.....	7
3	標準計算とカスタム計算の使用.....	8
3.1	標準計算.....	8
3.2	式を使用したカスタム計算の作成.....	8
	変数を使った式の簡略化.....	9
3.3	関数の操作.....	9
	セルへの関数の挿入.....	9
	関数の構文.....	10
	関数の例.....	10
	関数と式の演算子.....	15
4	計算コンテキストの理解.....	17
4.1	入力コンテキスト.....	17
4.2	出力コンテキスト.....	18
4.3	デフォルトの計算コンテキスト.....	20
	垂直テーブルのデフォルトのコンテキスト.....	21
	水平テーブルのデフォルトのコンテキスト.....	21
	クロスタブテーブルのデフォルトのコンテキスト.....	22
	セクションのデフォルトのコンテキスト.....	22
	ブレイクのデフォルトのコンテキスト.....	23
4.4	拡張構文を使用したデフォルトの計算コンテキストの変更.....	24
	拡張構文の演算子.....	25
5	スマートメジャーを使用した値の計算.....	29
5.1	グループ化集合とスマートメジャー.....	29
	グループ化集合の管理.....	30
5.2	スマートメジャーと分析範囲.....	30
5.3	スマートメジャーと SQL.....	31
	グループ化集合と UNION 演算子.....	31
5.4	スマートメジャーと式.....	33
	スマートメジャーと式を含むディメンション.....	33
	式のスマートメジャー.....	33
5.5	スマートメジャーとフィルタ.....	34

	スマートメジャーとフィルタに関する制約	34
	スマートメジャーとディメンションのフィルタ	34
	スマートメジャーのフィルタ処理	35
	スマートメジャーとドリルフィルタ	36
	スマートメジャーおよびネストされた OR フィルタ	36
6	関数、演算子、およびキーワード	37
6.1	関数	37
	カスタム書式	37
	集計関数	40
	文字列関数	80
	日付と時間関数	102
	データプロバイダ関数	119
	ドキュメント関数	136
	論理関数	146
	数値関数	156
	セット関数	180
	その他の関数	201
6.2	関数と式の演算子	223
	数理的演算子	223
	条件演算子	224
	論理演算子	224
	関数固有の演算子	227
	拡張構文の演算子	236
	集合演算子	239
6.3	拡張構文キーワード	240
	Block キーワード	240
	Body キーワード	241
	Break キーワード	242
	Report キーワード	243
	Section キーワード	244
6.4	数値の四捨五入と切り捨て	245
6.5	階層内のメンバーおよびメンバーセットの参照	246
7	数式のトラブルシューティング	248
7.1	式自動書き換えメカニズム	248
7.2	式のエラーメッセージと情報メッセージ	249
	#COMPUTATION	249
	#CONTEXT	249
	#DATASYNC	250

#DIV/0.	250
#ERROR.	250
#EXTERNAL.	251
#INCOMPATIBLE.	251
#MIX.	251
#MULTIVALUE.	251
#N/A.	252
#OVERFLOW.	252
#PARTIALRESULT.	252
#RANK.	252
#RECURSIVE.	253
#REFRESH.	253
#SECURITY.	253
#SYNTAX.	253
#TOREFRESH.	254
#UNAVAILABLE.	254
8 関数を使用した値の比較.	255
8.1 Previous 関数を使用した値の比較.	255
8.2 RelativeValue 関数を使用した値の比較.	255
スライスディメンションと RelativeValue 関数.	256
スライスディメンションとセクション.	258
スライスディメンションの順序.	259
スライスディメンションと並べ替え.	261
クロスタブでの RelativeValue の使用.	262

1 ドキュメント履歴:Web Intelligence の関数、式および計算

以下の表は、最も重要なドキュメント変更の概要です。

バージョン	日付	説明
SAP BusinessObjects Web Intelligence 4.1 SP07	2015 年 11 月	<p>このガイドに、次のトピックが追加されました。</p> <ul style="list-style-type: none">• 週数のカスタム書式を登録するための文字列 カスタム書式 [37 ページ]• タイムゾーンは、日付/時刻の値で表示されます。 カスタム書式 [37 ページ]• Concatenation() 関数の動作が更新されました。 Concatenation [81 ページ]• RelativeDate() 関数の動作が更新されました。 RelativeDate [113 ページ]• 4.1 SP3 以降に利用可能な式自動書き換えメカニズムがドキュメント化されました。 式自動書き換えメカニズム [248 ページ]
SAP BusinessObjects Web Intelligence 4.1 SP05	2014 年 11 月	<p>format_string 構文の例がガイドに追加されました。これらの例を確認する場合は、FormatDate 関数の Format_string の例 [85 ページ] を参照してください。</p>
SAP BusinessObjects Web Intelligence 4.1	2013 年 5 月	<p>アラビア語ロケール向け RTL (右から左) のサポートを追加。RTL ロケールの以下の文字機能の意味が特に影響を受けます。</p> <ul style="list-style-type: none">• Left• LeftPad• LeftTrim• Right• RightPad• RightTrim
SAP BusinessObjects Web Intelligence 4.0 Feature Pack 3	2012 年 2 月	Feature Pack 3
SAP BusinessObjects Web Intelligence 4.0 SP02	2011 年 6 月	サポートパッケージ 2。Web Intelligence にリブランドされました。
SAP BusinessObjects Interactive Analysis 4.0 SP01	2011 年 3 月	サポートパッケージ 1

バージョン	日付	説明
SAP BusinessObjects Interactive Analysis 4.0	2010 年 11 月	商標変更された最初の Interactive Analysis。このドキュメントの初版です。

2 このガイドについて

Web Intelligence における関数、式、計算の使い方ガイドでは、データ分析を実行したときに使用できる高度な計算機能について詳しく説明します。また、使用できる関数と演算子に関する構文も示します。

3 標準計算とカスタム計算の使用

標準計算関数を使用して、データをすばやく計算できます。

標準計算でニーズが満たされない場合は、式言語を使用してカスタム計算を作成できます。

3.1 標準計算

標準計算関数を使用して、データをすばやく計算できます。

次の標準計算を使用できます。

計算	説明
合計	選択したデータの合計を計算します。
件数	メジャーオブジェクトのすべての行数を計算します。ディメンションオブジェクト、または詳細オブジェクトの固有行数を計算します。
平均	データの平均値を計算します。
最小	選択したデータの最大値を計算します。
最大	選択したデータの最大値を計算します。
パーセンテージ	選択したデータを全体に対するパーセンテージで表示します。パーセンテージの結果はテーブルに追加される列または行に表示されます。
<div><div>i 注記</div><p>パーセンテージは、選択したメジャーをテーブルまたはブロック全体のそのメジャー値の合計と比較したものです。選択したメジャーを他のメジャーと比較するパーセンテージ計算には、カスタムの式を作成する必要があります。</p></div>	
デフォルト	デフォルトの集計関数を標準のメジャーに適用するか、データベース集計関数をスマートメジャーに適用します。

テーブルの列に標準計算を適用すると、計算結果がフッタに表示されます。計算ごとに1つのフッタが追加されます。

3.2 式を使用したカスタム計算の作成

カスタム計算を作成し、基本オブジェクトや標準計算以外の計算をレポートに追加できます。

カスタム計算を追加するには、式を作成します。式は、レポート変数、関数、演算子、計算コンテキストで構成されます。

カスタム計算式は、レポートオブジェクト、関数、演算子で構成されます。式には、コンテキストを明示的に指定できる計算コンテキストを設定できます。

❖ 例

1 個あたりの平均売上げの表示

[売上げ]オブジェクトと[販売数量]オブジェクトのあるレポートで、1 個あたりの売上げを追加する場合を考えます。計算 $[売上げ] / [販売数]$ を使用すると、売上げを販売個数で割ることで 1 個あたりの売上げを計算できます。

3.2.1 変数を使った式の簡略化

式が複雑な場合、変数を使って簡略化できます。

変数を使って複雑な式を管理しやすいサイズに分割し、理解しやすくすることで、式の作成時のエラーが減少します。

式には、あらかじめ作成しておいた変数も他のレポートオブジェクトと同じように使用できます。変数は式エディタの[変数]フォルダに表示されます。

この変数名を式に直接入力したり、他のオブジェクトのように数式ツールバーに変数をドラッグすることができます。

3.3 関数の操作

[売上げ] / [販売数] などのように、カスタム計算にレポートオブジェクトだけが含まれる場合があります。計算には、レポートオブジェクトに加えて関数を含めることもできます。

関数は、入力としてゼロ個以上の値を受け取り、その値に基づいて出力を返します。たとえば Sum 関数は、メジャーのすべての値を合計し、その結果を出力します。式 $\text{Sum}([売上げ])$ は、売上げの合計を出力します。この例では、関数の入力は[売上げ]メジャー、出力は[売上げ]メジャーの合計です。

関連情報

[関数と式の演算子 \[15 ページ\]](#)

[関数 \[37 ページ\]](#)

3.3.1 セルへの関数の挿入

レポートセル内のテキストは、必ず "=" で始まります。

そのまま表示される文字列は引用符で囲んで、式は引用符無しで表示されます。たとえば式 $\text{Average}([売上げ])$ は、セルでは $=\text{Average}([売上げ])$ と表示され、テキスト「平均売上げ」は $=$ 「平均売上げ」と表示されます。

セルには、テキストだけを使うこともできますし、“+”演算子を使って式とテキストを組み合わせることもできます。セルで、テキスト「平均売上げ」の後ろに平均売上げの数値を表示する場合は、セルに「="平均売上げ：" + Average([売上げ])」と入力します。

テキストの最後に区切り記号やスペースを入れておくと、セル内のテキストと数値を離して表示できます。

3.3.2 関数の構文

関数を選択すると、**式エディタ**に関数の構文が表示されます。

関数を使用するには、その名前と必要な入力値の数、データ型を知っている必要があります。関数が出力するデータ型も知っておく必要があります。

たとえば Sum 関数は、入力値として数値型オブジェクト (売上げを示す計数など) を 1 つ取り、数値型データ (計数オブジェクトのすべての値の合計) を出力します。

次に、Abs 関数の構文を示します。

```
num Abs (number)
```

この構文を見れば、Abs 関数が入力に単一の数値を取り、出力として数値を返すことがわかります。

3.3.3 関数の例

このトピックでは、式で使用される関数の例を示します。

❖ 例

UserResponse 関数によるプロンプト入力の表示

年、四半期、売上げを表示したレポートがあります。表示されていませんが、[都道府県]オブジェクトもレポートデータに含まれています。ユーザがレポートを実行すると、都道府県の選択を要求するプロンプトが表示されます。選択した都道府県をレポートのタイトルに表示することになります。データプロバイダが "eFashion"、プロンプトのテキストが "都道府県の選択" の場合、タイトルに入力する式は次のようになります。

```
"Quarterly Revenues for " + UserResponse([Query 1];"Enter values for State:")
```

ユーザがデータプロバイダの最新表示時に州としてイリノイを選択すると、レポートは次のようになります。

京都の四半期別売上げ

年	四半期	売上げ
2001	第1	¥256,454
	第2	¥241,149
	第3	¥107,006
	第4	¥133,306
2001	合計:	¥737,914

年	四半期	売上げ
2002	第1	¥334,297
	第2	¥254,722
	第3	¥230,573
	第4	¥331,067
2002	合計:	¥1,150,659

年	四半期	売上げ
2003	第1	¥255,658
	第2	¥354,724
	第3	¥273,186
	第4	¥250,517
2003	合計:	¥1,134,085

❖ 例

Percentage 関数を使用したパーセンテージの計算

Percentage 関数はパーセンテージを計算します。この関数は、そのコンテキストにおけるパーセンテージを計算します。たとえば、売上げを年別、四半期別にまとめた以下のテーブルのパーセンテージ列には、式 Percentage ([Sales revenue]) が指定されています。

年	四半期	売上げ	パーセンテージ
2001	第1	¥ 2,660,700	0.073
2001	第2	¥ 2,278,693	0.063
2001	第3	¥ 1,367,841	0.038
2001	第4	¥ 1,788,580	0.049
2002	第1	¥ 3,326,172	0.091
2002	第2	¥ 2,840,651	0.078
2002	第3	¥ 2,879,303	0.079
2002	第4	¥ 4,186,120	0.115
2003	第1	¥ 3,742,989	0.103
2003	第2	¥ 4,006,717	0.11
2003	第3	¥ 3,953,395	0.109
2003	第4	¥ 3,356,041	0.092
合計:			1

この式では、総売上げに対するそれぞれの売上げのパーセンテージが計算されます。この計算でのコンテキストは売上げ合計です。これは、テーブルの年別、四半期別の内訳にはない、唯一の意味のある売上げの数値です。

レポートを[年]別のセクションに分割すると、テーブル外のコンテキストはセクションの売上げ合計になります。

2001			
年	四半期	売上げ	パーセンテージ
2001	第1	¥ 2,660,700	0.329
2001	第2	¥ 2,278,693	0.281
2001	第3	¥ 1,367,841	0.169
2001	第4	¥ 1,788,580	0.221
合計:			1

パーセンテージを表示するセルがテーブルの外、セクションの中にある場合は、コンテキストは売上げ合計になります。この場合、Percentage 関数はセクションの売上げ合計を売上げ総合計に対するパーセンテージで計算します。

2001		0.222	
年	四半期	売上げ	パーセンテージ
2001	第1	¥ 2,660,700	0.329
2001	第2	¥ 2,278,693	0.281
2001	第3	¥ 1,367,841	0.169
2001	第4	¥ 1,788,580	0.221
2002		0.364	
年	四半期	売上げ	パーセンテージ
2002	第1	¥ 3,326,172	0.251
2002	第2	¥ 2,840,651	0.215
2002	第3	¥ 2,879,303	0.218
2002	第4	¥ 4,186,120	0.316

❖ 例

Sum 関数を使用したパーセンテージの計算

Percentage 関数ではなく Sum 関数を使用すると、パーセンテージを計算するコンテキストをより制御することができます。一連の数字の 1 つをそれらの数字の合計で割ると、合計に対するパーセンテージが得られます。たとえば、式 [Sales revenue]/Sum([Sales revenue]) は、売上げの総合計に対するパーセンテージとして売上げを計算します。

次のテーブルの[合計に対するパーセンテージ]列には、以下の式が入力されています。

[売上げ] / (Sum([売上げ] In Report))

[年合計に対するパーセンテージ]の式は、次のようになります。

[売上げ] / (Sum([売上げ] In Section))

2001				
年	四半期	売上げ	総合計に対する割合	年合計に対する割合
2001	第1	¥ 2,660,700	0.073	0.329
2001	第2	¥ 2,278,693	0.063	0.281
2001	第3	¥ 1,367,841	0.038	0.169
2001	第4	¥ 1,788,580	0.049	0.221

これらの式では、拡張構文キーワードの Report と Section を使って売上げ総合計と、それに対応する年ごとの売上げを計算するよう Sum 関数に指示しています。

関連情報

[拡張構文を使用したデフォルトの計算コンテキストの変更 \[24 ページ\]](#)

3.3.3.1 変数を使った分散式の簡略化

分散とは統計用語です。一連の値の分散は、平均値付近の値の広がりを計測するものです。

Var 関数を使用すると 1 回の手順で分散を計算できますが、手動による分散の計算は、変数を使用して複雑な式を簡単にする方法を示す良い例となります。手動で分散を計算するには、次の操作を実行する必要があります。

- 販売数量の平均を計算します。
- 商品ごとの販売数量と平均の差を計算し、その値を 2 乗します。
- 2 乗した値をすべて合計します。
- この合計を、商品の数から 1 を引いた数で割ります。

四半期ごとの販売数量を示すレポートに分散を含めることにします。変数を使用せずに式を作成すると、式は次のように複雑になります。

```
Sum(((販売数量] - Average([販売数量] ForEach [四半期]) In Report) * ([販売数量] -  
Average([販売数量] ForEach [四半期]) In Report)) In [四半期]) / (Count ([販売数量]  
ForEach [四半期]) - 1)
```

分散式の作成

分散式を作成するには、いくつかのステップが必要です。それぞれのステップを変数に置き換えます。次の変数を作成します。

- 販売数量の平均
- 商品数(販売数量の値の数)
- 各販売数量と平均の差の 2 乗
- 上記 2 乗値の合計を商品数から 1 を引いた数で割った結果

変数式は次のとおりです。

変数	式
販売数量の平均	Average([販売数量] In ([四半期])) In Report
商品数	Count([販売数量] In ([四半期])) In Report
2 乗した値	Power((販売数量] - 販売数量の平均]);2)
分散	Sum([2 乗した値] In ([四半期])) / ([商品数] - 1)

最終的な式は次のようになります。

```
Sum ([2 乗した値]) / [商品数] - 1)
```

この式の方が理解しやすい式です。簡略化された式では内容が上位レベルで表示されるため、細かい部分で何を計算しているのかで頭を悩ますことがありません。上位レベルの式で参照されている変数の式を確認することで、式の構成要素を理解できます。

たとえば、この式で参照している変数[2 乗した値]は、さらに変数[販売数量の平均]を参照しています。[2 乗した値]と[販売数量の平均]の内容を確認することで、式をドリルダウンしてその詳細を把握することができます。

3.3.4 関数と式の演算子

演算子は、式のコンポーネントを結合するものです。

式には、数理的演算子、条件演算子、論理演算子、関数固有の演算子、拡張構文演算子を含めることができます。

3.3.4.1 数理的演算子

数理的演算子は、一般的な算数の計算です。

加算 (+)、減算 (-)、乗算 (*)、除算 (/) 演算子があり、数値計算を実行します。式 [売上げ] - [販売コスト] には、数理的演算子の減算が含まれています。

i 注記

文字列と使用する場合は、“+”演算子は文字列を連結する演算子になります。この場合は、2 つの文字列が結合されます。たとえば “John” + “ Smith” という式は、“John Smith” を返します。

3.3.4.2 条件演算子

条件演算子は、値の比較方法を決定する演算子です。

演算子	説明
=	等しい
>	より大きい
<	より小さい
>=	以上
<=	以下
<>	等しくない

条件演算子は、次のように If 関数と共に使用します。

```
If [売上げ]>10000 Then "High" Else "Low"
```

この式は、売上げが 10000 以上のすべての行について“高い”を、その他の行については“低い”を返します。

3.3.4.3 論理演算子

論理演算子には、And、Or、Not、Between、Inlist の 5 種類があります。

論理演算子は、True または False の値を返す論理式で使います。

3.3.4.4 コンテキスト演算子

コンテキスト演算子は、拡張計算構文の一部です。

拡張構文を使うと、メジャーまたは式の計算対象となるディメンションを定義できます。

3.3.4.5 関数固有の演算子

一部の関数には、特定の演算子を引数として指定できます。

たとえば、Previous 関数には Sel 演算子を指定できます。

すべての関数は) と (を使って引数を囲みます。複数のパラメータを指定できる関数では、; を使ってパラメータを区切ります。

4 計算コンテキストの理解

計算コンテキストは、計算を行って結果を算出するときに考慮するデータです。

つまり、メジャーから与えられる値は、メジャーの計算に使用されるディメンションによって決まります。

レポートには、次の 2 種類のオブジェクトが含まれています。

- ディメンションは、数値を生成するビジネスデータを表します。店舗、年、地域などがディメンションデータになります。たとえば、1 つの店舗、1 年、または 1 地域で売上げを生成できます。店舗別の売上げ、年間の売上げ、または地域別の売上げについて話し合うことができます。
- メジャーは、分析データによって生成される数値データです。売上げや販売数などがメジャーデータになります。たとえば、特定の店舗の販売数について話し合うことができます。

メジャーは、分析データの組み合わせによって生成することもできます。たとえば、2005 年度の特定の店舗の売上げについて話し合うことができます。

1 つのメジャーの計算コンテキストには、次の 2 つの構成要素があります。

- メジャー値を判断するディメンションまたはディメンションの一覧
- メジャー値を判断するディメンションデータの一部

計算コンテキストには、次の 2 つの構成要素があります。

- 入力コンテキスト
- 出力コンテキスト

関連情報

[入力コンテキスト \[17 ページ\]](#)

[出力コンテキスト \[18 ページ\]](#)

4.1 入力コンテキスト

メジャーまたは式の入力コンテキストは、計算に挿入されるディメンションの一覧です。

入力コンテキストのディメンションの一覧は、値を出力する関数のかっこ内に表示されます。また、ディメンションの一覧は、ディメンションが 1 つしか含まれていない場合であってもかっこで囲み、ディメンションはセミicolonで区切る必要があります。

❖ 例

入力コンテキストの指定

“年”セクションがあり、その各セクションのブロックに“顧客”および“売上げ”列が含まれるレポートがある場合、入力コンテキストは次のようになります。

レポートパーツ	入力コンテキスト
セクションヘッダとブロックフッタ	年
ブロック内の行	年、顧客

セクションヘッダとブロックヘッダには、年別に集計された売上げが表示され、ブロック内の各行には、年別および顧客別に集計された売上げ（該当の年におけるその顧客の売上げ）が表示されます。

式を明示的に指定すると、これらの入力コンテキストは次のようになります。

```
Sum ([売上げ] In ([年]))
```

```
Sum ([売上げ] In ([年]; [顧客]))
```

入力コンテキストのディメンションは、指定する入力コンテキストの関数（この場合は Sum）の括弧内に表示されません。

4.2 出力コンテキスト

ブレークのあるブロックのフッタに表示される値は、出力コンテキストのある式で出力されます。

❖ 例

出力コンテキストの指定

年別、四半期別の売上げを表示した次のレポートでは、年にブレークが作成されています。また、年別の最低売上げが計算されています。

年	四半期	売上げ
2001	第1	¥ 2,660,700
	第2	¥ 2,278,693
	第3	¥ 1,367,841
	第4	¥ 1,788,580
2001	最小:	¥ 1,367,841

年	四半期	売上げ
2002	第1	¥ 3,326,172
	第2	¥ 2,840,651
	第3	¥ 2,879,303
	第4	¥ 4,186,120
2002	最小:	¥ 2,840,651

年	四半期	売上げ
2003	第1	¥ 3,742,989
	第2	¥ 4,006,717
	第3	¥ 3,953,395
	第4	¥ 3,356,041
2003	最小:	¥ 3,356,041

ブレークのないブロックに年別の最低売上げを表示する場合は、式に出力コンテキストを指定します。この式は、次のようになります。

```
Min ([売上げ]) In ([年])
```

出力コンテキストは、指定する出力コンテキストの関数の括弧の後に表示されます。この場合、年別の最低売上げが計算されます。

この式を含む列をブロックに追加すると、結果は次のようになります。

年	四半期	売上げ	年度最小値
2001	第1	¥ 2,660,700	¥ 1,367,841
2001	第2	¥ 2,278,693	¥ 1,367,841
2001	第3	¥ 1,367,841	¥ 1,367,841
2001	第4	¥ 1,788,580	¥ 1,367,841
2002	第1	¥ 3,326,172	¥ 2,840,651
2002	第2	¥ 2,840,651	¥ 2,840,651
2002	第3	¥ 2,879,303	¥ 2,840,651
2002	第4	¥ 4,186,120	¥ 2,840,651
2003	第1	¥ 3,742,989	¥ 3,356,041
2003	第2	¥ 4,006,717	¥ 3,356,041
2003	第3	¥ 3,953,395	¥ 3,356,041
2003	第4	¥ 3,356,041	¥ 3,356,041

[年度最小値]列には、前のレポートでブレークのフッタに表示されていた最低売上げが表示されているのがわかります。

この例では、入力コンテキストはブロックのデフォルトコンテキストである([年]、[四半期])であるため、明示的には指定されていません。つまり、年別、四半期別のどちらの売上げが出力されるかは、出力コンテキストによって決まります。入力および出力コンテキストをすべて明示的に指定した式は、次のようになります。

```
Min ([売上げ] In ([年];[四半期])) In ([年])
```

この式は、年別、四半期別の売上げを計算してから、各年度の最低売上げを出力します。

[年別最低売上]列で出力コンテキストを指定しなかった場合は、この列の数字は[売上げ列]の値と同じになります。その理由は、ブロックのデフォルトコンテキストに、そのブロックのディメンションが含まれているためです。年/四半期の組み合わせには1つの売上げしかいないため、年別、四半期別の最低売上げは、年別、四半期別の売上げと同じになります。

4.3 デフォルトの計算コンテキスト

メジャーには、配置されているレポート上の位置に応じたデフォルトの計算コンテキストがあります。

メジャーから返される数値は、メジャーが関連付けられているディメンションによって異なります。このディメンションの組み合わせは、計算コンテキストを表します。

デフォルトのコンテキストは、拡張構文を使用して変更できます。つまり、メジャーの生成に使用されるディメンションのセットを決定できます。これが計算コンテキストを定義することになります。

❖ 例

レポートのデフォルトコンテキスト

この例では、単純なレポートでのメジャーのデフォルト計算コンテキストについて説明します。このレポートには、顧客別の売上げが示され、年別のセクションに分割されています。

2005	合計: 8000
顧客	売上げ
ハリス	1000
ジョーンズ	3000
ウォルシュ	4000
合計:	8000
レポート合計: 8000	

次の表に、このレポートのメジャーの計算コンテキストを示します。

メジャー	値	コンテキスト
レポート合計	20000	レポートの総売上げ
セクションヘッダの合計	8000	年
顧客合計	1000、3000、4000	年;顧客
ブロックフッタの合計	8000	年

関連情報

[計算コンテキストの理解 \[17 ページ\]](#)

[拡張構文を使用したデフォルトの計算コンテキストの変更 \[24 ページ\]](#)

4.3.1 垂直テーブルのデフォルトのコンテキスト

垂直テーブルは、上部にヘッダ、下部にフッタを持ち、データを上から下に表示する標準のレポートテーブルです。

次の表は、デフォルトのコンテキストを示したものです。

計算の実行場所	入力コンテキスト	出力コンテキスト
ヘッダ	ブロックのボディの生成に使用するディメンションとメジャー	すべてのデータを集計してから、計算関数で1つの値を返す。
ブロックのボディ	その行の生成に使用するディメンションとメジャー	入力コンテキストと同じ
フッタ	ブロックのボディの生成に使用するディメンションとメジャー	すべてのデータを集計してから、計算関数で1つの値を返す。

4.3.2 水平テーブルのデフォルトのコンテキスト

水平テーブルのデフォルトのコンテキストは、垂直テーブルと同じです。

水平テーブルは、垂直テーブルを横向きにしたものです。

水平テーブルの表示は、BI ラウンチパッドの基本設定で選択した [優先表示ロケール] に応じて異なります。ロケールには、英語ロケールのように左から右 (LTR) のインタフェース配置が使用されるものと、アラビア語ロケールのように右から左 (RTL) のインタフェース配置が使用されるものがあります。

LTR ロケールでは、ヘッダは左に、フッタは右に、データは左から右に表示されます。RTL ロケールでは、ヘッダは右に、フッタは左に、データは右から左に表示されます。

4.3.3 クロスタブテーブルのデフォルトのコンテキスト

クロスタブは、ディメンションの交差部分にメジャーを表示するマトリックスとしてデータを表示します。

クロスタブのデフォルトのコンテキストは次のとおりです。

計算の実行場所	入力コンテキスト	出力コンテキスト
ヘッダ	ブロックのボディの生成に使用するディメンションとメジャー	すべてのデータを集計してから、計算関数で1つの値を返す。
ブロックのボディ	ブロックのボディの生成に使用するディメンションとメジャー	入力コンテキストと同じ
フッタ	ブロックのボディの生成に使用するディメンションとメジャー	すべてのデータを集計してから、計算関数で1つの値を返す。
VBody フッタ	その列の生成に使用するディメンションとメジャー	すべてのデータを集計してから、計算関数で1つの値を返す。
Hbody フッタ	その行の生成に使用するディメンションとメジャー	すべてのデータを集計してから、計算関数で1つの値を返す。
VFooter	フッタと同じ	すべてのデータを集計してから、計算関数で1つの値を返す。
HFooter	フッタと同じ	すべてのデータを集計してから、計算関数で1つの値を返す。

❖ 例

クロスタブのデフォルトのコンテキスト

以下のレポートは、クロスタブのデフォルトのコンテキストを示しています。

		2003	2003	2003	2003	¥3,926,240
		第1	第2	第3	第4	¥3,926,240
京都	¥1,134,085	¥255,658	¥354,724	¥273,186	¥250,517	¥1,134,085
兵庫	¥1,135,479	¥314,430	¥273,608	¥294,798	¥252,644	¥1,135,479
大阪	¥1,656,676	¥421,390	¥423,867	¥432,215	¥379,205	¥1,656,676
合計:	¥3,926,240	¥991,477	¥1,052,199	¥1,000,199	¥882,365	¥3,926,240

4.3.4 セクションのデフォルトのコンテキスト

セクションは、ヘッダ、ボディ、フッタで構成されます。

セクションのデフォルトのコンテキストは次のとおりです。

計算の実行場所	入力コンテキスト	出力コンテキスト
本文	セクションに表示される制限されたセクションデータを提供するレポートのディメンションとメジャー	すべてのデータを集計してから、計算関数で1つの値を返す。

❖ 例

セクションのデフォルトのコンテキスト

以下のレポートは、セクションのデフォルトのコンテキストを示しています。

2001		¥8,095,814
四半期	売上げ	セクション
第1	¥2,660,700	¥2,660,700
第2	¥2,278,693	¥2,278,693
第3	¥1,367,841	¥1,367,841
第4	¥1,788,580	¥1,788,580
合計:	¥8,095,814	

2002		¥13,232,246
四半期	売上げ	セクション
第1	¥3,326,172	¥3,326,172
第2	¥2,840,651	¥2,840,651
第3	¥2,879,303	¥2,879,303
第4	¥4,186,120	¥4,186,120
合計:	¥13,232,246	

2003		¥15,059,143
四半期	売上げ	セクション
第1	¥3,742,989	¥3,742,989
第2	¥4,006,717	¥4,006,717
第3	¥3,953,395	¥3,953,395
第4	¥3,356,041	¥3,356,041
合計:	¥15,059,143	

4.3.5 ブレークのデフォルトのコンテキスト

ブレークは、ヘッダ、ボディ、フッタで構成されます。

ブレークのデフォルトのコンテキストは次のとおりです。

計算の実行場所	入力コンテキスト	出力コンテキスト
ヘッダ	ブレークの現在のインスタンス	すべてのデータを集計してから、計算関数で1つの値を返す。
フッタ	ブレークの現在のインスタンス	すべてのデータを集計してから、計算関数で1つの値を返す。

❖ 例

ブレークのデフォルトのコンテキスト

以下のレポートは、ブレークのデフォルトのコンテキストを示しています。

年	四半期	売上げ
2001	第1	¥2,660,700
	第2	¥2,278,693
	第3	¥1,367,841
	第4	¥1,788,580
2001	合計:	¥8,095,814

年	四半期	売上げ
2002	第1	¥3,326,172
	第2	¥2,840,651
	第3	¥2,879,303
	第4	¥4,186,120
2002	合計:	¥13,232,246

年	四半期	売上げ
2003	第1	¥3,742,989
	第2	¥4,006,717
	第3	¥3,953,395
	第4	¥3,356,041
2003	合計:	¥15,059,143

4.4 拡張構文を使用したデフォルトの計算コンテキストの変更

拡張構文では、式やメジャーに追加して計算コンテキストを指定できるコンテキスト演算子を使用します。

メジャーや式のコンテキストは、入力コンテキストと出力コンテキストで構成されます。

関連情報

[拡張構文キーワード \[240 ページ\]](#)

[拡張構文の演算子 \[25 ページ\]](#)

4.4.1 拡張構文の演算子

コンテキスト演算子では、入力コンテキストと出力コンテキストを明示的に指定します。

次の表に、コンテキスト演算子を示します。

演算子	説明
In	コンテキストで使用するディメンションの明示的な一覧を指定します。
ForEach	デフォルトのコンテキストにディメンションを追加します。
ForAll	デフォルトのコンテキストからディメンションを除外します。

ForAll と ForEach 演算子は、デフォルトのコンテキストに多数のディメンションがある場合に便利です。コンテキストの追加と除外は多くの場合、In 演算子で明示的に指定するよりも、ForAll と ForEach を使用の方が簡単です。

4.4.1.1 In コンテキスト演算子

In コンテキスト演算子は、コンテキストのディメンションを明示的に指定します。

❖ 例

In によるコンテキストのディメンションの指定

次のレポートは、年と売上げを表示したレポートです。データプロバイダには[四半期]オブジェクトも含まれていますが、ブロックにこのディメンションは含まれていません。このレポートに、各年の四半期別の最高売上げを示す列を追加することになります。レポートは次のようになります。

年	売上げ	四半期別最高売上げ
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

このブロックと"四半期"ディメンションを含むブロックを比べると、[四半期別最高売上げ]列の値がどのように導き出されたかわかります。

年	四半期	売上げ
2001	Q1	\$2,660,699.50

年	四半期	売上げ
2001	Q2	\$2,279,003.00
2001	Q3	\$1,367,841.00
2001	Q4	\$1,788,580.00
	最大:	\$2,660,699.50
年	四半期	売上げ
	Q1	\$3,326,172.00
	Q2	\$2,840,651.00
	Q3	\$2,879,303.00
	Q4	\$4,186,120.00
	最大:	\$4,186,120.00
年	四半期	売上げ
	Q1	\$3,742,989.00
	Q2	\$4,006,717.50
	Q3	\$3,953,395.00
	Q4	\$3,356,041.00
	最大:	\$4,006,717.50

[四半期別最高売上げ]列は、各年の四半期別の最高売上げを示しています。たとえば、2002 年は第 4 四半期の売上げが最大であるため、2002 年の[四半期別最高売上げ]列には第 4 四半期の売上げが表示されています。

In 演算子を使用すると、四半期別最高売上げの式は次のようになります。

```
Max ([売上げ] In ([年];[四半期])) In ([年])
```

この式は、各 (年、四半期) の組み合わせについて最高売上げを計算し、年別にこの数字を出力します。

i 注記

ブロックのデフォルトの出力コンテキストは年なので、出力コンテキストを明示的に指定する必要はありません。

4.4.1.2 ForEach コンテキスト演算子

ForEach 演算子は、コンテキストにディメンションを追加します。

❖ 例

ForEach によるコンテキストへのディメンションの追加

次の表は、“四半期”ディメンションを含むが、ブロックにはそのディメンションが含まれていないレポートの各四半期に対する最大売上げを示しています。

年	売上げ	四半期別最高売上げ
2001	8096123.60	2660699.50
2002	13232246.00	4186120.00
2003	15059142.80	4006717.50

[四半期別最高売上げ]列で ForEach 演算子を含まない次の式を作成することができます。

```
Max ([売上げ] In ([年];[四半期])) In ([年])
```

ForEach コンテキスト演算子を使って、次の式を使った場合と同じ結果を導き出すことができます。

```
Max ([売上げ] ForEach ([四半期])) In ([年])
```

その理由は、“年”ディメンションは、ブロックのデフォルトの入力コンテキストだからです。ForEach 演算子を使用してコンテキストに[四半期]ディメンションを追加すると、入力コンテキストは([年];[四半期])になります。

4.4.1.3 ForAll コンテキスト演算子

ForAll コンテキスト演算子は、コンテキストからディメンションを除外します。

❖ 例

ForAll によるコンテキストからディメンションの除外

年、四半期、売上げを示すレポートで、次のブロックのように年別の売上げ合計を示す列を追加する場合を考えます。

年	四半期	売上げ	年度合計
2001	第1	¥ 2,660,700	¥ 8,095,814
2001	第2	¥ 2,278,693	¥ 8,095,814
2001	第3	¥ 1,367,841	¥ 8,095,814
2001	第4	¥ 1,788,580	¥ 8,095,814
2002	第1	¥ 3,326,172	¥ 13,232,246
2002	第2	¥ 2,840,651	¥ 13,232,246
2002	第3	¥ 2,879,303	¥ 13,232,246
2002	第4	¥ 4,186,120	¥ 13,232,246
2003	第1	¥ 3,742,989	¥ 15,059,143
2003	第2	¥ 4,006,717	¥ 15,059,143
2003	第3	¥ 3,953,395	¥ 15,059,143
2003	第4	¥ 3,356,041	¥ 15,059,143

年別に売上げを合計するには、入力コンテキストが(年)でなければなりません。しかし、デフォルトコンテキストは(年; 四半期)です。このため、式に `ForAll ([四半期])` と指定することで入力コンテキストから四半期を除外できます。結果は、次のような式になります。

```
Sum([売上げ] ForAll ([四半期]))
```

In 演算子を使用しても同じことができます。この場合は次の式になります。

```
Sum([売上げ] In ([年]))
```

この式では、四半期を除外して年を残すのではなく、コンテキストとして年を明白に指定しています。

5 スマートメジャーを使用した値の計算

スマートメジャーは、ユニバースに基づくデータベース (リレーショナルまたは OLAP) によって値が計算されるメジャーです。

このようなメジャーは、データベースによって返される詳細値から計算されるクラシックメジャーとは異なります。スマートメジャーから返されるデータは、SAP BusinessObjects Business Intelligence プラットフォームの Web Intelligence コンポーネントでネイティブではサポートされない方法で集計されます。

スマートメジャーを含むクエリでは、レポートに必要なすべての計算コンテキストでメジャーが計算されます。これらのコンテキストは、レポートの変更に伴って変更される場合があります。そのため、必要なコンテキストが変更された後、データが最新表示されるたびにクエリが変更されます。

このようなレポートを編集する場合、#TOREFRESH メッセージがレポートに自動的に挿入され、変更を反映させるにはレポートを最新表示する必要があることが通知されます。レポートを自動的に更新するには、[ドキュメントのプロパティ] ダイアログで [自動最新表示を有効にする] オプションを選択します。

i 注記

メジャー依頼は静的で、デザイン時のレポート定義に基づいて定義されます。場合によっては ("if [choice]= 1 then [dimension 1] else [dimension 2]" に基づく式など)、実行時にディメンションコンテキストが変更されることがあります。この場合、システムはメジャー計算を依頼することができず、空の値を返します。

スマートメジャーは、データベースの支援なしにすべてのコンテキストで計算を実行できる集計関数 (Max、Min、Count、Sum、Averag) の基本セットをサポートするクラシックメジャーとは動作が異なります。たとえば、"国" と "地域" ディメンションおよび "売上げ" メジャー (売上げの合計を計算する) を含むクエリを作成すると、最初にブロック内に、"国"、"地域" および "売上げ" が表示されます。ブロックから "地域" を削除しても、データを最新表示することなく、国のすべての地域の売上げを集計して各国の合計売上げを計算できます。スマートメジャーでは、次のような場合にデータの最新表示が必要になります。

生成されるクエリで、計算コンテキストがグループ化集合によって表される場合。

5.1 グループ化集合とスマートメジャー

グループ化集合は、メジャーの結果を生成する一連のディメンションです。

スマートメジャーにデータを返す生成された SQL には、レポートに含まれるそのメジャーのすべての集計のグループ化集合が含まれます。

❖ 例

クエリ内のグループ化集合

あるクエリに、"国"、"地域"、"都市"ディメンションおよび"売上げ"スマートメジャーが含まれているとします。これらのオブジェクトは、可能なすべてのコンテキストで売上げを計算する次のグループ化集合を示します。

- スマートメジャーの合計値
- (国, 地域, 都市)によるスマートメジャー値
- (国, 都市)によるスマートメジャー値
- (都市)によるスマートメジャー値
- (地域, 都市)によるスマートメジャー値
- (地域)によるスマートメジャー値
- (国, 地域)によるスマートメジャー値
- (国)によるスマートメジャー値

データベースで UNION がサポートされる場合は、生成される SQL の UNION 句に各グループ化集合が含まれます。

グループ化集合は、レポートに必要な計算コンテキストに従って更新されます。この計算コンテキストは、レポート構造の変更に応じて変更される場合があります。

5.1.1 グループ化集合の管理

スマートメジャーを含むクエリを最初に作成して実行すると、生成される SQL には、クエリオブジェクトで示される最も詳細なレベルでスマートメジャーを計算するために必要なグループ化集合が含まれます。

たとえば、"国"、"地域"、"都市" デイメンションと "売上げ" スマートメジャーを含むクエリを作成すると、生成された SQL には (国, 地域, 都市) グループ化要素が含まれます。詳細レベルが最も高いグループ化集合は常に SQL に表示されます。他のグループ化集合は、レポートの変更に応じて追加および削除されます。

"都市" デイメンションをブロックから削除した場合は、売上げを返すために (国, 地域) グループ化集合が必要になります。このグループ化集合はクエリ SQL では使用できないため、"売上げ" セルには #TOREFRESH と表示されます。データを最新表示すると、#TOREFRESH は売上げ値で置き換えられます。

ブロック内に"都市"デイメンションを挿入すると、(国, 地域) グループ化集合は必要なくなります。このグループ化集合はクエリ SQL から削除され、その値は次にデータを最新表示するときに破棄されます。

グループ化集合は、レポートデータを最新表示するたびに、レポートに必要な計算コンテキストに従って含められるか破棄されます。

場合によっては、スマートメジャーの値を表示できない場合があります。この場合、メジャーセルに #UNAVAILABLE が表示されます。

5.2 スマートメジャーと分析範囲

分析範囲を含むクエリを作成すると、初期のグループ化集合には結果オブジェクトが含まれますが、範囲オブジェクトは含まれません。

クエリでは、結果オブジェクトと範囲オブジェクトの組み合わせから可能なすべてのグループ化集合が生成されるわけではありません。

❖ 例

分析範囲とスマートメジャーを含むクエリ

クエリに結果オブジェクト"国"および"売上げ"が含まれています。分析範囲には、"国"および"都市"ディメンションが含まれています。クエリを実行すると、SQL に (国) グループ化集合が含まれ、ブロックに "国" および "地域" が表示されます。

5.3 スマートメジャーと SQL

5.3.1 グループ化集合と UNION 演算子

一部のデータベースは、GROUPING SETS 演算子を使用して、グループ化集合を明示的にサポートします。

スマートメジャーを含むクエリを作成すると、生成される SQL は複数の結果セットと UNION 演算子を使用して、GROUPING SETS の結果をシミュレートします。

❖ 例

UNION 演算子で取得されるグループ化集合

この例では、"国"、"地域"、"都市"ディメンションおよび"売上げ"スマートメジャーを含むクエリについて説明します。

i 注記

簡略化のため、スマートメジャーで合計値を計算します。実際には、Web Intelligence が合計をネイティブでサポートするため、スマートメジャーはこの集計に必要ありません。

クエリが最初に行われると、グループ化集合は(国, 地域, 都市)になります。SQL クエリ全体でこのグループ化集合が返され、SQL で UNION 演算子を使用する必要はありません。

"都市" ディメンションをテーブルから削除する場合、売上げ (最初に #TOREFRESH として表示される) を表示するために (国, 地域) グループ化集合が必要になります。データが最新表示されると、SQL は次のようになります。

```
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

```

各グループ化集合は、SELECT 文で表され、それぞれ独自の ID(GID 列)を持ちます。ディメンションの完全なセットが含まれていないグループ化集合には、空の列(SELECT ' ')が含まれます。これは、UNION を含むクエリ内の各 SELECT 文に同じ数の行が必要だからです。

"国" および "売上げ" を含む新しいブロックをレポートに追加する場合は、(国) グループ化集合が必要です。生成された SQL には、次のように 3 つのグループ化集合が含まれます。

```

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

```

5.4 スマートメジャーと式

5.4.1 スマートメジャーと式を含むディメンション

式または変数がスマートメジャーの計算コンテキストのディメンションとして表示され、式によってメジャーに必要なグループ化集合が決定される場合、スマートメジャーの値が表示されます。

たとえば、スマートメジャーとディメンションは以下の値を返します。

- ハイパーリンクウィザードで作成された URL。
- ディメンションの単純連結 (または空白除去)。
- 日付に対して FormatDate を使用する場合

i 注記

以下の関数の場合は、現在も #UNAVAILABLE メッセージが返されます: ForEach、ForAll、In、Where、Rank、Previous、RelativeValue、RelativeDate、TimeDim、および式 if ([selection] =1) then [dim1] else [dim2] で Min、Max、Last、または First が使用される場合の集計関数。

5.4.2 式のスマートメジャー

スマートメジャーが式に含まれている場合は、式がその位置で示されるコンテキストとは異なる計算コンテキストを必要としている場合でも、スマートメジャーの値を返すことができます。

たとえば、あるレポートに次のようなブロックが含まれているとします。

国	地域	売上げ
アメリカ	北部	10000
アメリカ	南部	15000
アメリカ	東部	14000
アメリカ	西部	12000

次の式を含む追加の列をテーブルに含めると、

```
[売上げ] ForAll ([地域])
```

列の最初の値は、#TOREFRESH になります。これは、計算から地域を除外するこの式にはグループ化集合 (国) が必要であるためです。データを最新表示すると、(国) グループ化集合がクエリに追加され、メジャー値が表示されます。

関連情報

[ForAll コンテキスト演算子 \[27 ページ\]](#)

5.5 スマートメジャーとフィルタ

5.5.1 スマートメジャーとフィルタに関する制約

スマートメジャーは、テーブル内または親コンテキスト (レポートフィルタ) 内にフィルタがない場合に、テーブルのボディで評価されます。

フィルタが存在する場合のスマートメジャーの評価方法は以下の表のとおりです。

レポート内にフィルタが存在する場合のスマートメジャーの評価方法

フィルタの適用対象	スマートメジャーの評価方法
メジャー	スマートメジャーは正しく評価されますが、一部の行がテーブルから削除されます。
すでにテーブルの軸の一部となっているディメンション	スマートメジャーは正しく評価されますが、一部の行がテーブルから削除されます。フィルタの適用後に集計されないため、スマートメジャーは評価可能です。
テーブルの軸の一部となっていないディメンションで、フィルタのオペランドが単一値 (フィルタにより 1 つの値または行が返される)	スマートメジャーは正しく評価されます。フィルタの適用後に集計されないため、スマートメジャーは評価可能です。
テーブルの軸の一部となっていないディメンションで、フィルタのオペランドが複数値 (フィルタにより複数の値または行が返される)	この場合、フィルタが集計の前に適用され、テーブルの 1 行で集計が必要になるため、スマートメジャーは評価不可能です (#UNAVAILABLE と表示されます)。

5.5.2 スマートメジャーとディメンションのフィルタ

スマートメジャーの値が依存するディメンションに複数値のフィルタが適用されても、ディメンションがメジャーの計算コンテキストに明示的に表示されない場合、スマートメジャーは値を返すことができず、セルには #UNAVAILABLE が表示されます。

これは、入力コントロールによるレポートフィルタにも適用されます。

#UNAVAILABLE が表示されるのは、メジャーはレポートでフィルタ処理されてから集計される必要があるのに対して、スマートメジャーはレポートレベルのフィルタの適用後には集計することができないためです。生成される SQL にクエリフィルタを追加することでメジャーを計算することはできますが、この解決策には、同じクエリに基づく他のレポートに影響を与えるというリスクがあります。

i 注記

複数値のフィルタでは、より大きい、リスト内、より小さいなどの演算子を使用して複数値をフィルタ処理します。「等しい」などの単一値フィルタを適用する場合は、#UNAVAILABLE エラーは生成されません。

i 注記

集計が不要な場合には回避策があります。資格がメジャーである変数として式を定義し、使用されるディメンションがその変数を含むブロックに含まれるようにします (この列は非表示にすると見やすくなります)。

❖ 例

スマートメジャーとディメンションのフィルタ

あるクエリに、"国" と "製品" ディメンション、および "売上げ" スマートメジャーが含まれています。"国" および "売上げ" がブロック内に表示されています。"製品" の値を "ドレス" または "ジャケット" に制限するレポートフィルタを適用すると、"売上げ" セルには #UNAVAILABLE が表示されます。

国	売上げ
フランス	#UNAVAILABLE
アメリカ	#UNAVAILABLE
合計:	#UNAVAILABLE

"製品" を "ジャケット" のみに制限すると、値が表示されます。

国	売上げ
アメリカ	971,444
合計:	971,444

i 注記

フィルタがテーブル内にあるため、国に対する複数値フィルタは結果を返します。

5.5.3 スマートメジャーのフィルタ処理

テーブルフッタの値は、テーブルに表示されるデータの集計である必要があります。

テーブルに表示されるデータがローカルでフィルタリングされる場合、システムはローカルでフィルタリングされたデータの依頼集計を返すことはできません。



例

スマートメジャーのフィルタ

国	OrderAmountDel
ブラジル	28,833.36
中国	51,384.33
フランス	68,630.22
米国	3,529,511.14
合計:	3,678,359.05
合計:	3,678,359.05

以下のテーブルでデータを OrderAmountDel > 60,000 でフィルタリングする場合

テーブルには、国ごとのテーブルのコンテキストの OrderAmountDel が 60,000 を超える行が表示されます。

フッタの合計は、表示される行の合計を計算します。

合計は、#UNAVAILABLE を返します。これは、計算では集計をバックエンドにプッシュする一方、Web Intelligence はローカルのフィルタリングを実行するために集計を依頼することができないためです。

国	OrderAmountDel
フランス	68,630.22
米国	3,529,511.14
合計:	#UNAVAILABLE
合計:	3,598,141.36

5.5.4 スマートメジャーとドリルフィルタ

ドリルフィルタは単位値フィルタです。

ドリルバーを使用して直接ドリルすることができます。

5.5.5 スマートメジャーおよびネストされた OR フィルタ

少なくとも1つのフィルタ対象ディメンションがブロックに表示されていないネストされた OR フィルタにより、ブロック内のスマートメジャーの #UNAVAILABLE エラーが生成されます。

これは、スマートメジャーが何らかのローカル処理 (フィルタリングや特定の Web Intelligence 式など) の後にローカルで集計される必要がある一方、この集計は依頼されないためです。

6 関数、演算子、およびキーワード

6.1 関数

式関数は、複数のカテゴリに分類されます。

i 注記

以下の言語では、関数、キーワード、および演算子は翻訳されません。中国語、日本語、ハンガリー語、韓国語、ポーランド語、トルコ語、タイ語、ロシア語。関数は、英語でインタフェースに表示されます。

カテゴリ	説明
集計	データを集計(一連の数値の合計や平均など)します。
文字列	文字列を操作します。
日付と時刻	日付または時間のデータを返します。
ドキュメント	ドキュメントに関するデータを返します。
データプロバイダ	ドキュメントのデータプロバイダに関するデータを返します。
論理	TRUE または FALSE を返します。
数値	数値データを返します。
その他	上記のカテゴリに該当しない関数です。
集合	階層からメンバーの集合を返します。

6.1.1 カスタム書式

テーブルで、カスタム書式の種類を使用して、セルにカスタムの書式を定義できます。

Web Intelligence 関数では、以下の日/日付定義、カレンダー定義、および時刻の文字定義が適用されます。

i 注記

カスタム書式は、Web Intelligence HTML インタフェースのテーブル内の数値および日付には適用できません。

次の表は、カスタム書式の作成に使用できる文字列の一覧です。

文字	表示	例
#	対応する数字。数値の桁数が書式を指定するのに使用されている # 文字の数よりも少ない場合、先頭にゼロは挿入しません。	'12345' を #, ##0 で書式設定する場合: '12,345' (ロケールでカンマが桁区切り記号として定義されている場合) または '12 345' (ロケールでスペースが桁区切り記号として定義されている場合) を返します。
0	対応する桁数。数値の桁数が書式を指定するのに使用されている 0 文字の数よりも少ない場合、数字の前にゼロを挿入します。	'123' を #0,000 で書式設定する場合: '0,123'
,	ロケールで定義されている桁区切り記号。	'1234567' を #, ##0 で書式設定する場合: '1,234,567' (ロケールでカンマが桁区切り記号として定義されている場合) または '1 234 567' (ロケールでスペースが桁区切り記号として定義されている場合) を返します。
.	ロケールで定義されている小数点区切り記号。	'12.34' を #.##0 で書式設定する場合: '12.34' (ロケールでピリオドが小数点区切り記号として定義されている場合) または '12,34' (ロケールでカンマが小数点区切り記号として定義されている場合) を返します。
[%]%	結果の後にパーセント記号 (%) を表示し、結果を 100 倍します。	0.50 は 50% になります。
%	結果の後に % 記号を表示するが、結果を 100 倍しません。	0.50 は 0.50% になります。
	スペース ()	'1234567' を # ##0 で書式設定する場合: '1234 567' を返します。
1, 2, 3, a, b, c, \$, £, 英数字。 € (など)		'705.15' を \$#.##0 で書式設定する場合: '\$705.15' を返します。#,##0 € で書式設定する場合: '705,15 €' を返します。
<div>i 注記</div> <p>英数字は、一重引用符で囲む必要があります。一重引用符で囲まないと、書式設定文字と解釈される可能性があります。たとえば、## は '123 4' になり、'# #' は '# 1234' になります。</p>		
[Red], [Blue], [Green], [Yellow], [Gray], [White], [Dark Red], [Dark Blue], [Dark Green]	指定された色の値。	'150' を #,##0 [Red] で書式設定する場合: '150' を赤色で返します。#,##0 [Blue] で書式設定する場合: '150' を青色で返します。
Day/date characters	(day, date)	
d	月を基準にした日にち番号 (前にゼロを付けない)。日にちが 1 桁の場合、前にゼロを付けずに表示されます。	月の最初の日を d で書式設定する場合: '1' を返します。

文字	表示	例
dd	日にち番号 (前にゼロを付ける)。日にちが1桁の場合、前にゼロを付けて表示されます。	月の最初の日を dd で書式設定する場合: '01' を返します。
ddd	曜日の短縮名。選択しているロケールで先頭文字が大文字化された曜日名が使用されている場合は、最初の文字は大文字になります。	英語の 'Monday' を ddd で書式設定する場合: 'Mon' を返します。フランス語の lundi の場合: lun を返します。
Dddd	すべてのロケールで曜日名の先頭文字を強制的に大文字化します。	英語の 'Monday' を Dddd で書式設定する場合: 'Mon' を返します。フランス語の lundi の場合: Lun を返します。
dddd	曜日の完全名。選択しているロケールで先頭文字の大文字化された曜日名が使用されている場合は、最初の文字は大文字になります。	英語の 'Monday' を dddd で書式設定する場合: 'Monday' を返します。フランス語の場合: lundi。
DDDD	曜日の完全名 (大文字)。	英語の 'Monday' を DDDD で書式設定する場合: 'MONDAY' を返します。フランス語の場合: LUNDI。
dddd dd	曜日と、これに続くスペースと週の何日目かを表す数字。	'Monday' を dddd dd で書式設定する場合: 'Monday 01' を返します。
Calendar characters	(week, month, year)	
M	月番号 (前にゼロは付けない)。月番号が1桁の場合、前にゼロを付けずに表示されます。	'January' を M で書式設定する場合: '1' を返します。
MM	月番号 (前にゼロを付ける)。月番号が1桁の場合、前にゼロを付けて表示されます。	'January' を MM で書式設定する場合: '01' を返します。
mmm	月の短縮名。選択しているロケールで先頭文字の大文字化が使用されている場合は、最初の文字が大文字になります。	英語の 'January' を mmm で書式設定する場合: Jan を返します。フランス語の場合: 'jan'。
Mmmm	月の短縮名。すべてのロケールで最初の文字が大文字化されます。	英語の 'January' を mmm で書式設定する場合: Jan を返します。フランス語の場合: 'Jan'。
mmmm	月の完全名。選択しているロケールで先頭文字の大文字化が使用されている場合は、最初の文字が大文字になります。	英語の 'January' を mmmm で書式設定する場合: January を返します。フランス語の場合: janvier を返します。
MMMM	月の完全名 (すべて大文字)。	英語の 'January' を MMMM で書式設定する場合: JANUARY を返します。フランス語の場合: JANVIER を返します。
ww	その年の何週目にあたるかを示す数値。	2015 年 1 月 9 日は、ww で書式設定する場合: 02'。2015 年の 7 週目にあたるからです。
w	暦週は先頭のゼロを付けずに出力されます。	2015 年 1 月 9 日は、w で書式設定する場合: 2'。2015 年の 7 週目にあたるからです。
W	その月の何週目にあたるかを示す数値	2015 年 1 月 9 日は、W で書式設定する場合: '2'。1 月の第 2 週目にあたるからです。

文字	表示	例
yy	西暦の下 2 桁。	'2003' を yy で書式設定する場合: '03'
yyyy	西暦の 4 桁すべて。	'2003' を yyyy で書式設定する場合: '2003'
Time of day characters	(hours, minutes, seconds, am/pm)	
hh:mm:ss a	時 (前にゼロを付けない) と分秒 (前にゼロを付ける)。 "a" 文字を指定すると、可能な場合、時間の後に AM または PM が表示されます。	'21:05:03' を hh:mm:ss a で書式設定する場合: 英語ロケールでは '9:05:03 PM' を返します。
H	0 から始まる 24 時間表記の時間 (1 桁の時間の場合、前にゼロを付けない)。	'21:00' を H で書式設定する場合: '21' を返します。表示可能な値は 0 ~ 23。
HH	0 から始まる 24 時間表記の時間。	'21:00' を HH で書式設定する場合: '21' を返します。表示可能な値は 00 ~ 23。
k	1 から始まる 24 時間表記の時間 (1 桁の時間の場合、前にゼロを付けない)。	'21:00' を k で書式設定する場合: '21' を返します。表示可能な値は 1 ~ 24。
kk	01 から始まる 24 時間表記の時間。	'21:00' を kk で書式設定する場合: '21' を返します。表示可能な値は 01 ~ 24。
hh	12 時間表記の時間。	'21:00' を hh で書式設定する場合: '09' を返します。
HH:mm	時と分 (1 桁の時間の前にゼロを付ける)。	'7:15' を HH:mm で書式設定する場合: '7:15' を返します。
HH:mm:ss	時分秒 (1 桁の時間の前にゼロを付ける)。	'7:15' を HH:mm:ss で書式設定する場合: '7:15:00' を返します。
mm:ss	分と秒 (1 桁の時間の前にゼロを付ける)。	'7:15:03' を mm:ss で書式設定する場合: '15:03' を返します。
z	日付/時刻の値についてのタイムゾーン情報は以下のとおりです。GMT+/-HH:mm	

6.1.2 集計関数

6.1.2.1 Aggregate

説明

指定されたメンバーセットのメジャーのデフォルトの集計を返します。

関数グループ

集計

構文

```
num Aggregate(measure[;member_set])
```

入力

パラメータ	説明	種類	必須
measure	任意のメジャー	メジャー	はい
member_set	集計の計算に使用されるメンバーセット	メンバーセット	いいえ

注

- Aggregate では、拡張構文のコンテキスト演算子を使用できます。
- member_set を含む場合、Aggregate はメンバーセット内すべてのメンバーのメジャーの集計値を返します。
- member_set には、複数のセットをセミコロン (;) で区切って入力することができます。
- メンバーセットのリストは {} で囲む必要があります。
- メンバーセット式では正確なメンバーまたはノードを指定しない場合、参照される階層がテーブルに存在している必要があります。これにより、メンバーセット式は、テーブル内の階層に存在する現在のメンバーを参照します。階層がテーブルにない場合、関数は #MULTIVALUE というメッセージを返します。
- 必要な集計がクエリで使用できない場合、依頼メジャー集計は #TOREFRESH を返します。ユーザが新しいレベルの集計を取得するには、ドキュメントを最新表示する必要があります。この状況は、たとえばユーザがフィルタバーを使用し、"すべての値" の前に値を選択した場合、およびその逆で、選択された値の前に "すべての値" を選択した場合に発生します。

例

"売上げ" メジャーのデフォルト集計が Sum で、"カリフォルニア州" が "地理" 階層 (国 > 州 > 市町村) のメンバーである場合、Aggregate([売上げ]; {Descendants([地理] & [米国].[カリフォルニア州]; 1)}) は、カリフォルニア州の全市町村の売上げの合計を返します。

関連情報

[階層内のメンバーおよびメンバーセットの参照 \[246 ページ\]](#)

6.1.2.2 集計関数におけるメンバー選択

説明

特定の集計関数に対し、ブロックに階層が含まれている場合に集計コンテキストを定義するためのメンバー選択を定義することができます。

関数グループ

集計

構文

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

入力

パラメータ	説明	型	必須
AggregationFunction	次のいずれかである必要があります。 <ul style="list-style-type: none">AggregateAverageCountMaxMinSum	集計関数	○
マイオブジェクト	ディメンションまたはメジャー	ディメンションまたはメジャー	○

パラメータ	説明	型	必須
memberselection	<p>定義されたメンバー、またはセット関数を使用するメンバーの計算されたセット。memberselection は中括弧で囲む必要があります。メンバーセットの各部分はセミコロンで区切ります:</p> <pre>{ [member one]; [member two]; CalculatedMemberSet () }</pre> <p>ここで、CalculatedMemberSet は次のセット関数のいずれかを使用します。</p> <ul style="list-style-type: none"> • Ancestor • Descendant • Lag • Children • Parent • Siblings 	メンバーまたはセット関数を使用するメンバーの計算されたセット	

説明

セット関数は、パラメータとして Object、Level、または Member を使用します。Object のみを指定し、オブジェクトがブロックにある階層オブジェクトである場合には、現在のメンバーが使用されます。次の構文を使用して特定のメンバーを定義することもできます:

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

Microsoft および Essbase .UNIX ソースの場合、Level を選択できます:

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

例

次の例は、すべて英語のデータソースからのものです。

❖ 例

次のサンプルでは、2002 年と 2001 年のインターネットセールスの差異を得ようとしています:

```
=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]})
```

あるいは、メンバー選択で 2 つのメンバーを選択するとします:

```
=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	

❖ 例

次のサンプルでは、製品階層があり、自転車に関連するすべての製品のインターネットセールスを把握しようとしています。しかしながら、製品のうち2つは異なる支社に存在しています：

```
=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

❖ 例

次のサンプルでは、まず最初にカナダと米国を、次に世界中のその他の国を比較し、北米地域の国の間でインターネットセールス額を比較しようとしています。

最初に、北米の国の合計を取得します。このサンプルでは、カナダと米国にのみ関心があります：

```
=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

さらに、北米のすべての国を比較します：

```
=([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

世界の顧客の合計は北米の 2.5 倍であり、オーストラリアは北米と比較してその 80% であることがわかります。

関連情報

[Aggregate \[40 ページ\]](#)

6.1.2.3 Average

説明

メジャーの平均値を返します。

関数グループ

集計

構文

```
num Average (measure [ ; member_set ] [ ; IncludeEmpty ])
```

入力

パラメータ	説明	種類	必須
measure	任意のメジャー	メジャー	はい
member_set	一連のセット	メンバーセット	いいえ
IncludeEmpty	計算に空の行を含めます	キーワード	いいえ (デフォルトでは、空の行は除外されます)

注

- Average 関数では、拡張構文のコンテキスト演算子を使用できます。
- member_set を含む場合、Average はメンバーセット内すべてのメンバーのメジャーの平均値を返します。
- member_set には、複数のセットをセミコロン (;) で区切って入力することができます。
- メンバーセットのリストは {} で囲む必要があります。
- メンバーセット式では正確なメンバーまたはノードを指定しない場合、参照される階層がテーブルに存在している必要があります。これにより、メンバーセット式は、テーブル内の階層に存在する現在のメンバーを参照します。階層がテーブルにない場合、関数は #MULTIVALUE というメッセージを返します。
- 必要な集計がクエリで使用できない場合、依頼メジャー集計は #TOREFRESH を返します。ユーザが新しいレベルの集計を取得するには、ドキュメントを最新表示する必要があります。この状況は、たとえばユーザがフィルタバーを使

用し、"すべての値" の前に値を選択した場合、およびその逆で、選択された値の前に "すべての値" を選択した場合に発生します。

- グループに指定されている依頼メジャーは、ローカル集計 (グループ化された値のメジャー値の集計) が必要なため、#UNAVAILABLE を返します。
"if then else" 式またはグループ値に対する依頼メジャーでローカル集計を強制しても、#MULTIVALUE というメッセージを返します。

例

メジャー [売上げ] の値が 41569、30500、40000、50138 の場合、`Average([売上げ])` は 40552 を返します。

"カリフォルニア州" が "地理" 階層 (国 > 州 > 市町村) のメンバーの場合、`Average([売上げ]; {[地理]&[米国]}.[カリフォルニア州].children)` は、カリフォルニア州の全市町村の売上げの平均を返します。

関連情報

[階層内のメンバーおよびメンバーセットの参照 \[246 ページ\]](#)

[IncludeEmpty 演算子 \[230 ページ\]](#)

6.1.2.4 Count

説明

値のセット内の値の数を返します。

関数グループ

集計

構文

```
integer Count(aggregated_data[;member_set][;IncludeEmpty][;Distinct|All])
```


入力

パラメータ	説明	種類	必須
aggregated_data	任意のディメンション、メジャー、階層、レベル、またはメンバーセット	ディメンション、メジャー、階層、メンバーセット	はい
member_set	カウントの計算に使用されるメンバーセット	メンバーセット	いいえ
IncludeEmpty	計算に空の値を含めます	キーワード	いいえ
Distinct All	計算に特定の値のみ(ディメンションの場合のデフォルト)またはすべての値(メジャーの場合のデフォルト)を含めます。	キーワード	いいえ

注

- Count では、拡張構文のコンテキスト演算子を使用できます。
- 2 番目の引数として IncludeEmpty を指定すると、関数は空の値 (null) を含めて計算します。
- Distinct|All パラメータを指定しないと、デフォルト値は、ディメンションに対しては Distinct、メジャーに対しては All となります。
- member_set を含めると、Count はカウントを member_set 内の値の数に限定します。
- member_set には、複数のセットをセミコロン (;) で区切って入力することができます。
- メンバーセットのリストは {} で囲む必要があります。
- メンバーセット式では正確なメンバーまたはノードを指定しない場合、参照される階層がテーブルに存在している必要があります。これにより、メンバーセット式は、テーブル内の階層に存在する現在のメンバーを参照します。階層がテーブルにない場合、関数は #MULTIVALUE というメッセージを返します。
- 必要な集計がクエリで使用できない場合、依頼メジャー集計は #TOREFRESH を返します。ユーザが新しいレベルの集計を取得するには、ドキュメントを最新表示する必要があります。この状況は、たとえばユーザがフィルタバーを使用し、[すべての値] の前に値を選択した場合、およびその逆で、選択された値の前に [すべての値] を選択した場合に発生します。
- グループに指定されている依頼メジャーは、ローカル集計 (グループ化された値のメジャー値の集計) が必要なため、#UNAVAILABLE を返します。
"if then else" 式またはグループ値に対する依頼メジャーでローカル集計を強制しても、#MULTIVALUE というメッセージを返します。

例

Count ("Test") は、1 を返します。

都市を表示する列に、重複しない 5 つの都市がある場合、1 つの都市が繰り返し何回も表示されていても Count ([都市]; Distinct) は 5 を返します。

都市を表示する列に、重複を含む 10 の都市がある場合、`Count([都市];All)` は 10 を返します。

都市を表示する列に 5 つの都市と 1 行の空白行がある場合、`Count([都市];IncludeEmpty)` は 6 を返します。

`Count([製品];{[地理]&[州]})` は、"地理" 階層の "州" レベルの製品の合計数を返します。

関連情報

[IncludeEmpty 演算子 \[230 ページ\]](#)

[Distinct/All 演算子 \[230 ページ\]](#)

6.1.2.5 First

説明

データセットの最初の値を返します。

関数グループ

集計

構文

```
input_type First(dimension|measure)
```

入力

パラメータ	説明	種類	必須
dimension measure	任意のディメンションまたはメジャー	ディメンションまたはメジャー	はい

注

- ブレークフッタで使用了した場合、`First` はブレーク内の最初の値を返します。
- セクションフッタで使用了した場合、`First` はセクション内の最初の値を返します。

例

テーブルフッタで使用了した場合、`First([売上げ])` はテーブル内の [売上げ] の最初の値を返します。

6.1.2.6 Interpolation

説明

空のメジャー値を補間で計算します。

関数グループ

集計

構文

```
num Interpolation(measure[;PointToPoint|Linear] [;NotOnBreak|(reset_dims)] [;Row|Col])
```

入力

パラメータ	説明	型	必須
measure	任意のメジャー	メジャー	○
PointToPoint Linear	補間方法 <ul style="list-style-type: none">• <code>PointToPoint</code>: ポイントツーポイント補間	キーワード	×

パラメータ	説明	型	必須
	<ul style="list-style-type: none"> Linear: 最小二乗補間による線形回帰 		
NotOnBreak reset_dims	<ul style="list-style-type: none"> NotOnBreak: ブロックまたはセクションブレイク上で、関数によって計算がリセットされるのを防止します。 reset_dims: 補間をリセットするために使用されるディメンションの一覧 	キーワード ディメンションの一覧	×
Row Col	計算の方向を設定します。	キーワード	(Row はデフォルトです)

注

- Interpolation は、値が欠落しているメジャーで折れ線チャートを作成する場合に特に便利です。関数を使用して、途切れた線と点ではなく、つながった折れ線のチャートを作成します。
- 最小二乗補間による線形回帰は、使用可能なすべてのメジャー値をできる限り厳密に渡す、 $f(x) = ax + b$ という形の一次方程式を計算することによって、欠落した値を計算します。
- ポイントツーポイント補間は、欠落した隣接する 2 つの値を渡す、 $f(x) = ax + b$ という形式の一次方程式を計算することにより、欠落した値を計算します。
- メジャーの並べ替え順序は、Interpolation で返される値に影響します。
- Interpolation が含まれる式には、並べ替えまたは順位を適用できません。
- 値の一覧に値が 1 つしかない場合、Interpolation はこの値を使用して、欠落しているすべての値を表示します。
- 補間メジャーに適用されたフィルタは、フィルタが影響を与える値に応じて、Interpolation で返される値を変更できます。

例

Interpolation([値]) は、デフォルトのポイントツーポイント補間メソッドを使用して、欠落している以下の値を表示します。

日	値	Interpolation([値])
月曜	12	12
火曜	14	14
水曜		15
木曜	16	16

日	値	Interpolation([値])
金曜		17
土曜		18
日曜	19	19

関連情報

[Linear 演算子 \[231 ページ\]](#)

[PointToPoint 演算子 \[233 ページ\]](#)

6.1.2.7 Last

説明

ディメンションまたはメジャー内の最後の値を返します。

関数グループ

集計

構文

```
input_type Last (dimension|measure)
```

入力

パラメータ	説明	種類	必須
dimension measure	任意のディメンションまたはメジャー	ディメンションまたはメジャー	はい

注

- ブレークフッタで使用了した場合、Last はブレーク内の最後の値を返します。
- セクションフッタで使用了した場合、Last はセクション内の最後の値を返します。

例

テーブルフッタで使用了した場合、First([売上げ]) はテーブル内の [売上げ] の最初の値を返します。

6.1.2.8 Max

説明

一連の値の最大値を返します。

関数グループ

集計

構文

```
input_type Max (aggregated_data [,member_set])
```

入力

パラメータ	説明	種類	必須
aggregated_data	任意のディメンション、メジャー、階層、レベル、またはメンバーセット	ディメンション、メジャー、階層、レベル、またはメンバーセット	はい
member_set	一連のセット	メンバーセット	いいえ

注

- `Max` では、拡張構文のコンテキスト演算子を使用できます。
- `member_set` を含めると、`Max` はメンバーセット内のすべてのメンバーの集計データの最大値を返します。
- `member_set` には、複数のセットをセミコロン (;) で区切って入力することができます。
- メンバーセットのリストは {} で囲む必要があります。
- メンバーセット式では正確なメンバーまたはノードを指定しない場合、参照される階層がテーブルに存在している必要があります。これにより、メンバーセット式は、テーブル内の階層に存在する現在のメンバーを参照します。階層がテーブルにない場合、関数は `#MULTIVALUE` というメッセージを返します。
- 必要な集計がクエリで使用できない場合、依頼メジャー集計は `#TOREFRESH` を返します。ユーザが新しいレベルの集計を取得するには、ドキュメントを最新表示する必要があります。この状況は、たとえばユーザがフィルタバーを使用し、"すべての値" の前に値を選択した場合、およびその逆で、選択された値の前に "すべての値" を選択した場合に発生します。
- グループに指定されている依頼メジャーは、ローカル集計 (グループ化された値のメジャー値の集計) が必要なため、`#UNAVAILABLE` を返します。
"if then else" 式またはグループ値に対する依頼メジャーでローカル集計を強制しても、`#MULTIVALUE` というメッセージを返します。

例

"売上げ" メジャーの値が 3000、60034、および 901234 の場合、`Max([Sales Revenue])` は 901234 を返します。

"都市" ディメンションの値が "Aberdeen" と "London" の場合、`Max([City])` は "London" を返します。

"米国" が "地理" 階層 (国 > 州 > 市町村) のメンバーの場合、`Max([売上げ]; {[地理].[米国].Children})` は米国の州の最大の売上げを返します。

6.1.2.9 Median

説明

メジャーのメジアン (中央値) を返します。

関数グループ

集計

構文

```
num Median(measure)
```

入力

パラメータ	説明	種類	必須
measure	任意のメジャー	メジャー	はい

注

一連の数値の個数が偶数の場合、Median は中央の 2 つの数値の平均値を返します。

例

[売上げ] の値が 835420、971444、1479660 の場合、Median([売上げ]) は 971,444 を返します。

6.1.2.10 Min

説明

一連の値の最小値を返します。

関数グループ

集計

構文

```
input_type Min(aggregated_data[;member_set])
```


入力

パラメータ	説明	型	必須
aggregated_data	任意のディメンション、メジャー、階層、レベル、またはメンバーセット	ディメンション、メジャー、階層、レベル、またはメンバーセット	○
member_set	一連のセット	メンバーセット	×

注

- Min では、拡張構文のコンテキスト演算子を使用できます。
- member_set を含めると、Min はメンバーセット内のすべてのメンバーの集計データの最小値を返します。
- member_set には、複数のセットをセミコロン (;) で区切って入力することができます。
- メンバーセットのリストは {} で囲む必要があります。
- メンバーセット式では正確なメンバーまたはノードを指定しない場合、参照される階層がテーブルに存在している必要があります。これにより、メンバーセット式は、テーブル内の階層に存在する現在のメンバーを参照します。階層がテーブルにない場合、関数は #MULTIVALUE というメッセージを返します。
- 必要な集計がクエリで使用できない場合、依頼メジャー集計は #TOREFRESH を返します。ユーザが新しいレベルの集計を取得するには、ドキュメントを最新表示する必要があります。この状況は、たとえばユーザがフィルタバーを使用し、"すべての値" の前に値を選択した場合、およびその逆で、選択された値の前に "すべての値" を選択した場合に発生します。
- グループに指定されている依頼メジャーは、ローカル集計 (グループ化された値のメジャー値の集計) が必要なため、#UNAVAILABLE を返します。
"if then else" 式またはグループ値に対する依頼メジャーでローカル集計を強制しても、#MULTIVALUE というメッセージを返します。

例

計数[売上げ]の値が 3000、60034、901234 の場合、Min([売上げ]) は 3000 を返します。

"都市" ディメンションに値 "Aberdeen" および "London" が含まれる場合、Min([City]) は "Aberdeen" を返します。

Min([売上げ]; {[地理]&[米国].children}) は、"米国" が "地理" 階層内のレベル "国" > "州" > "市町村" のメンバーの場合は、米国の州の最低売上げを返します。

6.1.2.11 Mode

説明

データ セットに最も頻繁に発生する値を返します。

関数グループ

集計

構文

```
input_type Mode (dimension|measure)
```

入力

パラメータ	説明	型	必須
dimension measure	任意のディメンションまたはメジャー	メジャー	○

注

- データセットに、その他すべての値よりも頻繁に発生する1つの値が含まれない場合、Mode は NULL を返します。

例

[売上げ] の値が 100、200、300、200 の場合、Mode ([売上げ]) は 200 を返します。

Mode ([国]) は、最も頻繁に発生する [国] の値を返します。

6.1.2.12 Percentage

説明

メジャー値を埋め込みコンテキストのパーセンテージとして表します。

関数グループ

集計

構文

```
num Percentage (measure [;Break] [;Row|Col])
```

入力

パラメータ	説明	種類	必須
measure	任意のメジャー	メジャー	はい
Break	テーブルのブレイクを考慮します。	キーワード	いいえ
Row Col	計算の方向を設定します。	キーワード	いいえ

例

次のテーブルの「パーセンテージ」列には、式 `Percentage ([売上げ])` が入力されています。

年	売上げ	パーセンテージ
2001	1000	10
2002	5000	50
2003	4000	40
合計:	10000	100

デフォルトで埋め込まれるコンテキストは、テーブル内のメジャーの合計値です。オプションの `Break` 引数を使って、テーブルに適用されたブレイクを考慮させることができます。この場合はデフォルトの埋め込みコンテキストは、テーブルセクションとなります。

次のテーブルの「パーセンテージ」列には、式 `Percentage ([売上げ];Break)` が入力されています。

年	四半期	売上げ	パーセンテージ
2001	第 1	1000	10
	第 2	2000	20
	第 3	5000	50
	第 4	2000	20
2001	合計:	10000	100
年	四半期	売上げ	パーセンテージ
2002	第 1	2000	20

	第 2	2000	20
	第 3	5000	50
	第 4	1000	10
2002	合計:	10000	100

Percentage 関数は、行方向にも列方向にも使用できます。方向は、Row|Col 引数を使って明示的に指定できます。たとえば、次のクロスタブの [%] 列には、Percentage ([売上];Row) という式が入力されています。

	第 1	%	第 2	%	第 3	%	第 4	%
2001	1000	10	2000	20	5000	50	2000	20
2002	2000	20	2000	20	5000	50	1000	10

6.1.2.13 Percentile

説明

メジャーの第 n 分位数を返します。

関数グループ

数値

構文

```
num Percentile (measure;percentile)
```

入力

パラメータ	説明	種類	必須
measure	任意のメジャー	メジャー	はい
percentile	小数で示された百分率	数値	はい

注

n 百分位数は、一連の数値の n% 以上の数値です。n% は 0.n のフォームで表します。

例

[メジャー]が、一連の number (10;20;30;40;50) の場合、Percentile ([measure];0.3) は、22 を返します。これは、一連の数値の 30% 以上の値を示します。

6.1.2.14 Product

説明

メジャーの値を乗算します。

関数グループ

集計

構文

```
num Product (measure)
```

入力

パラメータ	説明	種類	必須
measure	任意のメジャー	メジャー	はい

例

[メジャー] の値が 2、3、5 の場合、Product ([メジャー]) は 30 を返します。

6.1.2.15 RunningAverage

説明

メジャーの積算平均値を返します。

関数グループ

集計

構文

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

各セクションで RunningAverage をリセットするには、次の構文を使用することをお勧めします。

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

入力

パラメータ	説明	型	必須
measure	任意のメジャー	メジャー	○
Row Col	計算の方向を設定します。	キーワード	×
IncludeEmpty	計算に空の値を含めます	キーワード	×
reset_dims	指定したディメンションに対する計算をリセットします。	ディメンションの一覧	×
section	セクションを設定するディメンション	キーワード	○ (セクションリセットの場合)

注

- RunningAverage では、拡張構文のコンテキスト演算子を使用できます。
- Row および Col 演算子を使用して、計算の方向を設定できます。
- RunningAverage によって参照されるメジャーに並べ替えを適用すると、積算平均値はメジャーを並べ替えた後に計算されます。

- 指定するリセットディメンションの数が1つだけでも、ディメンションは常にかっこの中に配置する必要があります。
- 複数のリセットディメンションを指定する場合、それらをセミコロンで区切る必要があります。
- RunningAverage は、ブロックブレイクまたは新しいセクションの後に平均を自動的にリセットすることはありません。

例

RunningAverage ([売上げ]) は、次の結果を返します。

国	リゾート	売上げ	積算平均値
アメリカ	ハワイアンクラブ	1,479,660	835,420
アメリカ	パハマビーチ	971,444	1,225,552
フランス	フランスリビエラ	835,420	1,095,508

RunningAverage ([売上げ]; ([国])) は、次の結果を返します。

国	リゾート	売上げ	積算平均値
アメリカ	ハワイアンクラブ	1,479,660	835,420
アメリカ	パハマビーチ	971,444	1,225,552
フランス	フレンチリヴィエラ	835,420	835,420

[四半期] セクションで RunningAverage を使用する例で、式 RunningAverage ([売上げ]; ([四半期])) を使用すると、次の結果が返されます。

Q1

都市	売上げ	積算平均値
ニューヨーク	\$1,987,114.70	\$1,987,114.70
ヒューストン	\$1,544,627.80	\$1,765,871.25
ロサンゼルス	\$1,129,177.60	\$1,553,640.03

Q2

都市	売上げ	積算平均値
ニューヨーク	\$2,028,090.70	\$2,028,090.70
ヒューストン	\$1,380,838.20	\$1,704,464.45
ロサンゼルス	\$980,405.30	\$1,463,111.40

関連情報

[IncludeEmpty 演算子 \[230 ページ\]](#)

6.1.2.16 RunningCount

説明

数値一式の積算件数を返します。

関数グループ

集計

構文

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

各セクションで RunningCount をリセットするには、次の構文を使用してください。

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

入力

パラメータ	説明	型	必須
dimension measure	任意のディメンションまたはメジャー	ディメンションまたはメジャー	○
Row Col	計算の方向を設定します。	キーワード	×
IncludeEmpty	計算に空の値を含めます	キーワード	×
reset_dims	指定したディメンションに対する計算をリセットします。	ディメンションの一覧	×
section	セクションを設定するディメンション	キーワード	○ (セクションリセットの場合)

注

- RunningCount では、拡張構文のコンテキスト演算子を使用できます。
- Row および Col 演算子を使用して、計算の方向を設定できます。
- RunningCount によって参照されるメジャーに並べ替えを適用すると、積算カウント値はメジャーを並べ替えた後に計算されます。
- 指定するリセットディメンションの数が1つだけでも、ディメンションは常にかっこの中に配置する必要があります。
- 複数のリセットディメンションを指定する場合、それらをセミコロンで区切る必要があります。
- RunningCount は、ブロックブレイクまたは新しいセクションの後に件数を自動的にリセットすることはありません。

例

次のテーブル内で使用した場合、RunningCount ([売上げ]) は、次の結果を返します。

国	リゾート	売上げ	積算件数
アメリカ	ハワイアンクラブ	1,479,660	1
アメリカ	パハマビーチ	971,444	2
フランス	フランスリビエラ	835,420	3

次のテーブル内で使用した場合、RunningCount ([売上]; ([国])) は、次の結果を返します。

国	リゾート	売上げ	積算件数
アメリカ	ハワイアンクラブ	1,479,660	1
アメリカ	パハマビーチ	971,444	2
フランス	フレンチリヴィエラ	835,420	1

[週] セクションで RunningCount を使用する例で、入力制限で [売上げ] を 30,000 ドルを超える売上げのリストに制限して、式 RunningCount ([品目]; ([週])) を使用すると、次の結果が返されます。

週 1

品目	売上げ	積算件数
スエット T シャツ	\$186,191	1
シャツウェスト	\$139,082	2
ドレス	\$70,931	3

週 2

品目	売上げ	積算件数
アクセサリー	\$344,617	1
スエット T シャツ	\$196,976	2
シャツウェスト	\$105,597	3

ドレス	\$76,290	4
セーター	\$68,364	5

週 1 は売上げが 30,000 ドルを超える品目は 3 つであるのに対して、週 2 は 30,000 ドルを超える品目が 5 つであることに注目してください。

関連情報

[IncludeEmpty 演算子 \[230 ページ\]](#)

[Row/Col 演算子 \[233 ページ\]](#)

[IncludeEmpty 演算子 \[230 ページ\]](#)

[IncludeEmpty 演算子 \[230 ページ\]](#)

6.1.2.17 RunningMax

説明

ディメンションまたはメジャーの積算最大値を返します。

関数グループ

集計

構文

```
input_type RunningMax (dimension | measure [ ; Row | Col ] [ ; (reset_dims) ] )
```

各セクションで RunningMax をリセットするには、次の構文を使用することをお勧めします。

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

入力

パラメータ	説明	型	必須
dimension measure	任意のディメンションまたはメジャー	ディメンションまたはメジャー	○
Row Col	計算の方向を設定します。	キーワード	×
reset_dims	指定したディメンションに対する計算をリセットします。	ディメンションの一覧	×
section	セクションを設定するディメンション	キーワード	○ (セクションリセットの場合)

注

- RunningMax では、拡張構文のコンテキスト演算子を使用できます。
- Row および Col 演算子を使用して、計算の方向を設定できます。
- RunningMax によって参照されるメジャーに並べ替えを適用すると、積算最大値はメジャーを並べ替えた後に計算されます。
- 指定するリセットディメンションの数が1つだけでも、ディメンションは常にかっこの中に配置する必要があります。
- 複数のリセットディメンションを指定する場合、それらをセミコロンで区切る必要があります。
- RunningMax は、ブロックブレイクまたは新しいセクションの後に最大値を自動的にリセットすることはありません。

例

次のテーブル内で使用した場合、RunningMax ([売上]) は、次の結果を返します。

国	リゾート	売上げ	積算最大値
フランス	フランスリビエラ	835,420	835,420
アメリカ	パハマビーチ	971,444	971,444
アメリカ	ハワイアンクラブ	1,479,660	1,479,660

[都市] セクションで RunningMax を使用する例で、式 RunningMax ([売上げ]; ([都市])) を使用すると、次の結果が返されます。

オースチン		
四半期	売上げ	積算最大値
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
ポストン		
四半期	売上げ	積算最大値
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

関連情報

[IncludeEmpty 演算子 \[230 ページ\]](#)

[Row/Col 演算子 \[233 ページ\]](#)

6.1.2.18 RunningMin

説明

ディメンションまたはメジャーの積算最小値を返します。

関数グループ

集計

構文

```
input_type RunningMin (dimension|measure; [Row|Col]; [(reset_dims)])
```

各セクションで RunningMin をリセットするには、次の構文を使用することをお勧めします。

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

入力

パラメータ	説明	型	必須
dimension detail measure	任意のディメンションまたはメジャー	ディメンションまたはメジャー	○
Row Col	計算の方向を設定します。	キーワード	×
reset_dims	指定したディメンションに対する計算をリセットします。	ディメンションの一覧	×
section	セクションを設定するディメンション	キーワード	○ (セクションリセットの場合)

注

- RunningMin では、拡張構文のコンテキスト演算子を使用できます。
- Row および Col 演算子を使用して、計算の方向を設定できます。
- RunningMin によって参照されるメジャーに並べ替えを適用すると、積算最小値はメジャーを並べ替えた後に計算されます。
- 指定するリセットディメンションの数が1つだけでも、ディメンションは常にかっこの中に配置する必要があります。
- 複数のリセットディメンションを指定する場合、それらをセミコロンで区切る必要があります。
- RunningMin は、ブロックブレイクまたは新しいセクションの後に最小値を自動的にリセットすることはありません。

例

次のテーブル内で使用した場合、RunningMin ([売上げ]) は、次の結果を返します。

国	リゾート	売上げ	積算最小値
フランス	フランスリビエラ	835,420	835,420
アメリカ	パハマビーチ	971,444	835,420
アメリカ	ハワイアンクラブ	1,479,660	835,420

[都市] セクションで RunningMin を使用する例で、式 RunningMin ([売上げ]; ([都市])) を使用すると、次の結果が返されます。

オースチン		
四半期	売上げ	積算最小値
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
ポストン		
四半期	売上げ	積算最小値
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

関連情報

[IncludeEmpty 演算子 \[230 ページ\]](#)

[Row/Col 演算子 \[233 ページ\]](#)

6.1.2.19 RunningProduct

説明

メジャーの積算積を返します。

関数グループ

集計

構文

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

入力

パラメータ	説明	種類	必須
measure	任意のメジャー	メジャー	はい

パラメータ	説明	種類	必須
Row Col	計算の方向を設定します。	キーワード	いいえ
reset_dims	指定したディメンションに対する計算をリセットします。	ディメンションの一覧	いいえ

注

- RunningProduct 関数では、拡張構文のコンテキスト演算子を使用できます。
- Row および Col 演算子を使用して、計算の方向を設定できます。
- RunningProduct によって参照されるメジャーに並べ替えを適用すると、積算積値はメジャーをソート後に計算されます。
- 指定するリセットディメンションの数が1つだけでも、ディメンションは常にかっこの中に配置する必要があります。
- 複数のリセットディメンションを指定する場合、それらをセミコロンで区切る必要があります。
- RunningProduct は、ブロックブレイクまたは新しいセクションの後に積を自動的にリセットすることはありません。

例

次のテーブル内で使用した場合、RunningProduct ([ゲストの数]) は、次の結果を返します。

国籍	都市	ゲストの数	累積積
日本	神戸	6	6
日本	大阪	4	24
アメリカ	シカゴ	241	5,784

次のテーブル内で使用した場合、RunningProduct ([ゲストの数]; ([国籍])) は、次の結果を返します。

国籍	都市	ゲストの数	累積積
日本	神戸	6	6
日本	大阪	4	24
アメリカ	シカゴ	241	5784

関連情報

[IncludeEmpty 演算子 \[230 ページ\]](#)

[Row/Col 演算子 \[233 ページ\]](#)

6.1.2.20 RunningSum

説明

メジャーの積算合計値を返します。

関数グループ

集計

構文

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

各セクションで RunningSum をリセットするには、次の構文を使用することをお勧めします。

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

入力

パラメータ	説明	型	必須
measure	任意のメジャー	メジャー	○
Row Col	計算の方向を設定します。	キーワード	×
reset_dims	指定したディメンションに対する計算をリセットします。	ディメンションの一覧	×
section	セクションを設定するディメンション	キーワード	○ (セクションリセットの場合)

注

- RunningSum では、拡張構文のコンテキスト演算子を使用できます。
- Row および Col 演算子を使用して、計算の方向を設定できます。
- RunningSum 関数によって参照されるメジャーに並べ替えを適用すると、積算合計値はメジャーをソート後に計算されます。
- 指定するリセットディメンションの数が1つだけでも、ディメンションは常にかっこの中に配置する必要があります。

- 複数のリセットディメンションを指定する場合、それらをセミコロンで区切る必要があります。
- RunningSum は、ブロックブレイクまたは新しいセクションの後に合計を自動的にリセットすることはありません。

例

次のテーブル内で使用した場合、RunningSum([売上]) は、次の結果を返します。

国	リゾート	売上げ	積算合計値
フランス	フランスリビエラ	835,420	835,420
アメリカ	パハマビーチ	971,444	1,806,864
アメリカ	ハワイアンクラブ	1,479,660	3,286,524

次のテーブルで使用した場合、RunningSum([売上];([国])) は、次の結果を返します。

国	リゾート	売上げ	積算合計値
フランス	フレンチリヴィエラ	835,420	835,420
アメリカ	パハマビーチ	971,444	971,444
アメリカ	ハワイアンクラブ	1,479,660	2,451,104

[四半期] セクションで RunningSum を使用する例で、式 RunningSum([売上げ];([四半期])) を使用すると、次の結果が返されます。

Q1		
都市	売上げ	積算合計値
ニューヨーク	\$1,987,114.70	\$1,987,114.70
ヒューストン	\$1,544,627.80	\$3,531,742.50
ロサンゼルス	\$1,129,177.60	\$4,660,920.10
Q2		
都市	売上げ	積算合計値
ニューヨーク	\$2,028,090.70	\$2,028,090.70
ヒューストン	\$1,380,838.20	\$3,408,928.90
ロサンゼルス	\$980,405.30	\$4,389,334.20

関連情報

[IncludeEmpty 演算子 \[230 ページ\]](#)

[Row/Col 演算子 \[233 ページ\]](#)

6.1.2.21 StdDev

説明

メジャーの標準偏差を返します。

関数グループ

集計

構文

```
num StdDev (measure)
```

入力

パラメータ	説明	種類	必須
measure	任意のメジャー	メジャー	はい

注

標準偏差とは、統計的な対象となる分散の度合いを表す数値です。母集団の分散は、次のように計算されます。

- 集団の平均値を計算します。
- それぞれの値から平均値を引き、その差を 2 乗します。
- 2 乗した値をすべて合計します。
- 合計値を (<集団の値の個数> -1) で割ります。
- 計算結果の平方根を求めます。

例

measure の値が (2, 4, 6, 8) の場合、StdDev ([メジャー]) は 2.58 を返します。

関連情報

[Var \[77 ページ\]](#)

6.1.2.22 StdDevP

説明

メジャーの母標準偏差を返します。

関数グループ

集計

構文

```
num StdDevP(measure)
```

入力

パラメータ	説明	種類	必須
measure	任意のメジャー	メジャー	はい

注

母集団の標準偏差とは、統計的な対象となる分散の度合いを表す数値です。母集団の分散は、次のように計算されます。

- 集団の平均値を計算します。
- それぞれの値から平均値を引き、その差を 2 乗します。
- 2 乗した値をすべて合計します。
- 合計値を (<集団の値の個数>) で割ります。
- 計算結果の平方根を求めます。

StdDevP では、拡張構文のコンテキスト演算子を使用できます。

例

`measure` の値が (2, 4, 6, 8) の場合、`StdDev([メジャー])` は 2.24 を返します。

6.1.2.23 Sum

説明

メジャーの合計値を返します。

関数グループ

集計

構文

```
num Sum(measure[;member_set])
```

入力

パラメータ	説明	種類	必須
measure	任意のメジャー	メジャー	はい
member_set	一連のセット	メンバーセット	いいえ

注

- `Sum` では、拡張構文のコンテキスト演算子を使用できます。
- `member_set` を含めると、`Sum` はメンバーセット内のすべてのメンバーのメジャーの合計を返します。
- `member_set` には、複数のセットをセミコロン (;) で区切って入力することができます。
- メンバーセットのリストは {} で囲む必要があります。
- メンバーセット式では正確なメンバーまたはノードを指定しない場合、参照される階層がテーブルに存在している必要があります。これにより、メンバーセット式は、テーブル内の階層に存在する現在のメンバーを参照します。階層がテーブルにない場合、関数は `#MULTIVALUE` というメッセージを返します。

- 必要な集計がクエリで使用できない場合、依頼メジャー集計は #TOREFRESH を返します。ユーザが新しいレベルの集計を取得するには、ドキュメントを最新表示する必要があります。この状況は、たとえばユーザがフィルタバーを使用し、"すべての値" の前に値を選択した場合、およびその逆で、選択された値の前に "すべての値" を選択した場合に発生します。
- XIR2 から XIR3 に移行する際、XIR2 クエリに IN 句および WHERE 句が含まれる集計関数は、次のような括弧を使用することで、絶対に Sum 関数に含める必要があります。
XIR2 の式は、`=Sum([Measure] In ([Dim 1];[Dim 2])) In ([Dim 1]) Where ([Dim 3]="Constant")` です。
XIR3 からは、宣言を `=Sum((([Measure] In ([Dim 1];[Dim 2])) In ([Dim 1]) Where ([Dim 3]="Constant"))` のように変更します。
- グループに指定されている依頼メジャーは、ローカル集計 (グループ化された値のメジャー値の集計) が必要なため、#UNAVAILABLE を返します。
"if then else" 式またはグループ値に対する依頼メジャーでローカル集計を強制しても、#MULTIVALUE というメッセージを返します。

例

計数[売上げ]の値が 2000、3000、4000、1000 の場合、`Sum([売上げ])` は 10000 を返します。

"カリフォルニア州" が "地理" 階層 (国 > 州 > 市町村) のメンバーの場合、`Sum([売上げ]; {Descendants([地理] & [アメリカ], [カリフォルニア州]; 1)})` は、カリフォルニア州の全市町村の売上げ合計を返します。

6.1.2.24 Var

説明

メジャーの分散を返します。

関数グループ

集計

構文

```
num Var (measure)
```

入力

パラメータ	説明	種類	必須
measure	任意のメジャー	メジャー	はい

注

分散とは、統計的な対象となる分散の度合いを表す数値です。母集団の分散は、次のように計算されます。

- 集団の平均値を計算します。
- それぞれの値から平均値を引き、その差を 2 乗します。
- 2 乗した値をすべて合計します。
- 合計値を (**<集団の値の個数>** -1) で割ります。

分散は、標準偏差の 2 乗です。

Var 関数では、拡張構文のコンテキスト演算子を使用できます。

例

メジャーの値が (2, 4, 6, 8) の場合、Var([メジャー]) は 6.67 を返します。

関連情報

[StdDev \[74 ページ\]](#)

6.1.2.25 VarP

説明

メジャーの母集団の分散を返します。

関数グループ

集計

構文

```
num VarP (measure)
```

入力

パラメータ	説明	種類	必須
measure	任意のメジャー	メジャー	はい

注

母集団の分散とは、統計的な対象となる分散の度合いを表す数値です。母集団の分散は、次のように計算されます。

- 集団の平均値を計算します。
- それぞれの値から平均値を引き、その差を 2 乗します。
- 2 乗した値をすべて合計します。
- 合計値を (<集団の値の個数>) で割ります。

母集団の分散は、母集団の標準偏差の 2 乗です。

VarP 関数では、拡張構文のコンテキスト演算子を使用できます。

例

メジャーの値が (2, 4, 6, 8) の場合、VarP ([メジャー]) は 5 を返します。

関連情報

[StdDevP \[75 ページ\]](#)

6.1.3 文字列関数

6.1.3.1 Asc

説明

文字の ASCII 値を返します。

関数グループ

文字

構文

```
int Asc(string)
```

入力

パラメータ	説明	種類	必須
string	任意の文字列	文字列	はい

注

string に複数の文字が含まれる場合、文字列内の最初の文字の ASCII 値を返します。

例

Asc("A") は、65 を返します。

Asc("ab") は、97 を返します。

Asc([国]) は、[国] が“US”の場合、85 を返します。

6.1.3.2 Char

説明

指定された ASCII コードに関連する文字を返します。

関数グループ

文字

構文

```
string Char(ascii_code)
```

入力

パラメータ	説明	種類	必須
ascii_code	ASCII コード	数値	はい

注

`number` が小数の場合、小数部分は無視されます。

例

`Char(123)` は "{" を返します。

6.1.3.3 Concatenation

説明

2 つの文字列を連結します。数値の場合、連結するのではなく値を合計します。

i 注記

少なくとも1つの入力パラメータが文字列の場合、その他の入力パラメータはすべて文字列に変換されます。

関数グループ

文字列

構文

```
string Concatenation (first_string;second_string)
```

入力

パラメータ	説明	型	必須
first_string	1 番目の文字列	文字列または数値	○
second_string	1 番目の文字列に追加された文字列	文字列または数値	○

注

"+" 演算子を使っても、同じ結果を得ることができます。

"First " + "Second" は "First Second" を返します。

"First " + "Second" + " Third" は、"First Second Third" を返します。

連結を使用して、集計関数に複数のディメンションを含めることができます。たとえば、Count ([Sales Person]+[Quarter]+[Resort]) は、Desktop Intelligence で使用可能な構文 Count (<Sales Person>,<Quarter>,<Resort>) と同等です。

例

Concatenation ("First "; "Second") は、"First Second" を返します。

Concatenation ("First ";Concatenation ("Second "; "Third")) は、"First Second Third" を返します。

[A] が数値で [A] = 1 の場合、Concatenation([A]; [A]) は "2" を返します。

[A] が文字列で [A] = 1 の場合、Concatenation([A]; [A]) は "11" を返します。

[A] が文字列、[B] が数値、[A] = 1、[B] = 2 の場合、Concatenation([A]; [B]) は "12" を返します。

6.1.3.4 Fill

説明

文字列を n 回繰り返すことによって、文字列を作成します。

関数グループ

文字

構文

```
string Fill(repeating_string; num_repeats)
```

入力

パラメータ	説明	種類	必須
repeating_string	繰り返す文字列	文字列	はい
num_repeats	繰り返す回数	数値	はい

例

Fill ("New York"; 2) は、"New York New York"を返します。

6.1.3.5 FormatDate

説明

指定した書式に従って日付を表示します。

関数グループ

文字

構文

```
string FormatDate(date;format_string)
```

入力

パラメータ	説明	型	必須
date	書式設定する日付	日付	○
format_string	適用する書式	文字列	○

注

- 出力の形式は、セルに適用した日付の形式によって異なります。
- 色の書式設定文字列 ([Red]、[Blue] など) は、FormatDate に適用できません。

例

今日の日付が 2005 年 12 月 15 日の場合、FormatDate(CurrentDate();"dd/MM/yyyy") は"15/12/2005"を返します。

関連情報

[カスタム書式 \[37 ページ\]](#)

6.1.3.5.1 FormatDate 関数の Format_string の例

FormatDate の format_string 構文では、次の表に示す例を使用できます。

i 注記

これらのサンプルは、Web Intelligence リッチクライアントまたはアプレットインタフェースの**数値の書式設定**ダイアログボックスに表示されますが、表示されるサンプルは BI ラウンチパッド基本設定で選択している製品ロケールによって異なります。たとえば、**英語**を選択している場合は、“September 21, 2004” が使用できるサンプルになります。

サンプル	構文
Tuesday, September 21, 2004	dddd, 'mmmm d', 'yyyy
September 21, 2004	mmmm d', 'yyyy
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

→ ヒント

テキストがパターン記号と混同されないように、実際のテキストはアポストロフィで囲む構文で表すことをお勧めします。たとえば、上の表の最後のサンプル、“HH'h'mm” の ‘h’ のようにします。

関連情報

[FormatDate \[84 ページ\]](#)

[カスタム書式 \[37 ページ\]](#)

6.1.3.6 FormatNumber

説明

指定した書式に従って数値を表示します。

関数グループ

文字

構文

```
string FormatNumber(number;format_string)
```

入力

パラメータ	説明	型	必須
number	書式設定する数値	数値	○
format_string	適用する書式	文字列	○

注

- 出力の形式は、セルに適用した数値の形式によって異なります。
- 色の書式設定文字列 ([Red]、[Blue] など) は、FormatNumber に適用できません。

例

[売上げ] が 835,420 の場合、FormatNumber([売上げ];"#,##.00") は 835,420.00 を返します。

関連情報

[カスタム書式 \[37 ページ\]](#)

6.1.3.7 HTMLEncode

説明

文字列に HTML エンコード規則を適用します。

関数グループ

文字

構文

```
string HTMLEncode(html)
```

入力

パラメータ	説明	種類	必須
html	HTML 文字列	文字列	はい

例

`HTMLEncode("http://www.sap.com")` は、`"http%3A%2F%2Fwww%2Esap%2Ecom"` を返します。

6.1.3.8 InitCap

説明

最初の文字列を大文字にします。

関数グループ

文字

構文

```
string InitCap(string)
```

入力

パラメータ	説明	種類	必須
string	大文字にする文字列	文字列	はい

例

`InitCap("we hold these truths to be self-evident")` は、`"We hold these truths to be self-evident"` を返します。

6.1.3.9 Left

説明

文字列の左端から指定された文字数分の文字列を返します。

i 注記

選択したインタフェースロケールがアラビア語 (右から左へ表示/読む) の場合、この関数は、文字列の論理的開始位置から先頭の文字を返します。

関数グループ

文字

構文

```
string Left(string;num_chars)
```


入力

パラメータ	説明	種類	必須
string	指定する文字列	文字列	はい
num_chars	文字列の最初から取り出す文字数	数値	はい

例

[国] の値が“France”の場合、`Left([国];2)` は“Fr”を返します。

6.1.3.10 LeftPad

説明

文字列の左に文字をバイト数単位で追加します。文字列の長さをバイト数で指定します。

i 注記

選択したインタフェースロケールがアラビア語 (右から左へ表示/読む) の場合、この関数は、文字列の論理的開始位置の先頭に別の文字列の文字を追加します。

関数グループ

文字

構文

```
string LeftPad(padded_string;length;left_string)
```

入力

パラメータ	説明	種類	必須
padded_string	元の文字列	文字列	はい
length	出力文字列の長さ	数値	はい
left_string	padded_string の最初に追加する文字列	文字列	はい

注

- length が left_string と padded_string を合わせた長さよりも短い場合、left_string は切り捨てられます。
- length が padded_string の長さより短い、または等しい長さである場合、関数は padded_string を返します。
- length が padded_string と left_string を合わせた長さよりも長い場合、長さを埋めるのに十分な数だけ left_string が繰り返されるか、または部分的に繰り返されます。

例

LeftPad("York";8;"New ") は、“New York”を返します。

LeftPad("York";6;"New ") は、“NeYork”を返します。

LeftPad("York";11;"New ") は、“New NewYork”を返します。

LeftPad("New ";2;"York") は“New”を返します。

6.1.3.11 LeftTrim

説明

文字列から先行のスペースを切り取ります。

i 注記

選択したインタフェースロケールがアラビア語 (右から左へ表示/読む) の場合、この関数は、文字列の論理的開始位置から先頭のスペースを削除します。

関数グループ

文字

構文

```
string LeftTrim(trimmed_string)
```

入力

パラメータ	説明	種類	必須
trimmed_string	指定する文字列	文字列	はい

例

[国] の値が“France”の場合、`LeftTrim([国])` は“France”を返します。

6.1.3.12 Length

説明

文字列のバイト数を返します。

関数グループ

文字

構文

```
int Length(string)
```

入力

パラメータ	説明	種類	必須
string	指定する文字列	文字列	はい

例

[名前] が "Smith" の場合、`Length([名前])` は 5 を返します。

6.1.3.13 Lower

説明

文字列を小文字に変換します。

関数グループ

文字

構文

```
string Lower(string)
```

Input

パラメータ	説明	種類	必須
string	小文字に変換する文字列	文字列	はい

例

`Lower("New York")` は "new york" を返します。

6.1.3.14 Match

説明

文字列がパターンに一致するかどうかを調べます。

関数グループ

文字

構文

```
bool Match(test_string;pattern)
```

入力

パラメータ	説明	種類	必須
test_string	テキストパターンと照らし合わせる文字列	文字列	はい
pattern	テキストパターン	文字列	はい

注

- パターンには、ワイルドカード“*” (任意の複数の文字)と“?”(任意の1文字)を使用できます。

例

[国] の値が“France”の場合、Match([国], "F*") は TRUE を返します。

[国]の値が“USA”の場合、Match([国]; "?S?") は、TRUE を返します。

Match("New York";"P*") は、FALSE を返します。

6.1.3.15 Pos

説明

文字列内のテキストパターンの開始位置を返します。

関数グループ

文字

構文

```
int Pos(test_string;pattern)
```

入力

パラメータ	説明	種類	必須
test_string	テキストパターンと照らし合わせる文字列	文字列	はい
pattern	テキストパターン	文字列	はい

注

- 複数のパターンが出現する場合、Pos は最初に出現するパターンの位置を返します。

例

Pos("New York";"Ne") は、1を返します。

Pos("New York, New York";"Ne") は、1を返します。

Pos("New York"; "York") は、5を返します。

6.1.3.16 Replace

説明

文字列の一部を指定した文字列で置き換えます。

関数グループ

文字

構文

```
string Replace(replace_in;replaced_string;replace_with)
```

入力

パラメータ	説明	種類	必須
replace_in	テキストが置換される文字列	文字列	はい
replaced_string	置換されるテキスト	文字列	はい
replace_with	replaced_string に置換するテキスト	文字列	はい

例

`Replace("New YORK";"ORK";"ork")` は、“New York”を返します。

6.1.3.17 Right

説明

文字列の右端の文字列 (文字列の終わりにある文字列) を返します。

i 注記

選択したインタフェースロケールがアラビア語 (右から左へ表示/読む) の場合、この関数は、文字列の論理的開始位置から先頭の文字を返します。

関数グループ

文字

構文

```
string Right(string;num_chars)
```

入力

パラメータ	説明	種類	必須
string	任意の文字列	文字列	はい
num_chars	右端から戻す文字数	数値	はい

例

[国] の値が“France”の場合、`Right([国];2)` は“ce”を返します。

6.1.3.18 RightPad

説明

文字列の右に別の文字列を追加します (元の文字列の最初に文字列を追加します)。

i 注記

選択したインタフェースロケールがアラビア語 (右から左へ表示/読む) の場合、この関数は、文字列の論理的開始位置の先頭の文字に文字列を追加します。

関数グループ

文字

構文

```
string RightPad(padded_string;length;right_string)
```

入力

パラメータ	説明	種類	必須
padded_string	元の文字列	文字列	はい
length	出力文字列の長さ	数値	はい
right_string	padded_string の最後に追加する文字列	文字列	はい

注

- length が right_string と padded_string を合わせた長さよりも短い場合、right_string は切り捨てられます。
- length が padded_string の長さより短い、または等しい長さである場合、関数は padded_string を返します。
- length が padded_string と right_string を合わせた長さよりも長い場合、長さを埋めるのに十分な数だけ right_string が繰り返されるか、または部分的に繰り返されます。

例

RightPad("New ";8;"York") は、“New York”を返します。

RightPad("New "; 6;"York") は、“New Yo”を返します。

RightPad("New "; 11;"York") は、“New YorkYor”を返します。

RightPad("New ";2;"York") は“New”を返します。

6.1.3.19 RightTrim

説明

文字列から末尾のスペースを切り取ります。

i 注記

選択したインターフェースロケールがアラビア語 (右から左へ表示/読む) の場合、この関数は、文字列の論理的終端で末尾のスペースを削除します。

関数グループ

文字

構文

```
string RightTrim(trimmed_string)
```

入力

パラメータ	説明	種類	必須
trimmed_string	指定する文字列	文字列	はい

例

[国] の値が“France”の場合、RightTrim([国]) は“France”を返します。

6.1.3.20 Substr

説明

文字列の一部を返します。

関数グループ

文字

構文

```
string SubStr(string;start;length)
```

入力

パラメータ	説明	種類	必須
string	任意の文字列	文字列	はい
start	抽出された文字列の開始位置	数値	はい
length	取り出す文字数	数値	はい

例

SubStr ("Great Britain";1;5) は Great を返します。

SubStr("Great Britain";7;7) は Britain を返します。

6.1.3.21 Trim

説明

文字列から先頭と末尾のスペースを切り取ります。

関数グループ

文字

構文

```
string Trim(trimmed_string)
```

入力

パラメータ	説明	種類	必須
string	指定する文字列	文字列	はい

例

Trim(" Great Britain ") は“Great Britain”を返します。

6.1.3.22 Upper

説明

文字列を大文字に変換します。

関数グループ

文字

構文

```
string Upper(string)
```

入力

パラメータ	説明	種類	必須
string	変換される文字列	文字列	はい

例

`Upper("New York")` は、“NEW YORK”を返します。

6.1.3.23 UriEncode

説明

文字列に URL エンコード規則を適用します。

関数グループ

文字

構文

```
string UriEncode(html)
```

入力

パラメータ	説明	種類	必須
html	エンコードされる URL	文字列	はい

例

`UriEncode("http://www.sap.com")` は、“http%3A%2F%2Fwww%2Esap%2Ecom”を返します。

6.1.3.24 WordCap

説明

文字列の各単語の最初の文字を大文字にします。

関数グループ

文字

構文

```
string WordCap(string)
```

入力

パラメータ	説明	種類	必須
string	大文字にする文字列	文字列	はい

例

`WordCap("Sales revenue for March")` は、“Sales Revenue For March”を返します。

6.1.4 日付と時間関数

6.1.4.1 CurrentDate

説明

地域設定に従って書式設定された現在の日付を返します。

関数グループ

日付と時刻

構文

```
date CurrentDate()
```

例

今日が 2002 年 9 月 10 日だとすると、CurrentDate() は“2002 年 9 月 10 日”を返します。

6.1.4.2 CurrentTime

説明

地域設定に従って書式設定された現在の時刻を返します。

関数グループ

日付と時刻

構文

```
time CurrentTime()
```

例

現在の時刻が午後 11:15 の場合は、CurrentTime は 11:15 を返します。

6.1.4.3 DayName

説明

日付の曜日名を返します。

関数グループ

日付と時刻

構文

```
string DayName (date)
```

入力

パラメータ	説明	種類	必須
date	指定する日付	日付	はい

例

[予約日]が 2001 年 12 月 15 日(土)の場合、DayName ([予約日]) は Saturday を返します。

注

入力データは変数にする必要があります。DayName ("07/15/2001") のように、日付を直接指定することはできません。

6.1.4.4 DayNumberOfMonth

説明

月の日付の数値を返します。

関数グループ

日付と時刻

構文

```
int DayNumberOfMonth (date)
```

入力

パラメータ	説明	種類	必須
date	指定する日付	日付	はい

例

[予約日]が 2001 年 12 月 15 日の場合、DayNumberOfMonth ([予約日]) は 15 を返します。

6.1.4.5 DayNumberOfWeek

説明

曜日の番号を返します。

関数グループ

日付と時刻

構文

```
int DayNumberOfWeek (date)
```

入力

パラメータ	説明	種類	必須
date	指定する日付	日付	はい

注

この関数は、月曜日を週の最初の日として処理します。

例

[予約日] が 2005 年 5 月 2 日 (月) の場合、DayNumberOfWeek ([予約日]) は 1 を返します。

6.1.4.6 DayNumberOfYear

説明

年における日付の番号を返します。

関数グループ

日付と時刻

構文

```
int DayNumberOfYear (date)
```

入力

パラメータ	説明	種類	必須
date	指定する日付	日付	はい

例

[予約日]が2003年12月15日の場合、`DayNumberOfYear([予約日])`は349を返します。

6.1.4.7 DaysBetween

説明

2つの日付の間の日数を返します。

関数グループ

日付と時刻

構文

```
int DaysBetween(first_date;last_date)
```

i 注記

引数に使用する日付は、同じタイムゾーンである必要があります。これは、比較や計算など、すべての日付操作に適用されます。

入力

パラメータ	説明	種類	必須
first_date	最初の日	日付	はい

パラメータ	説明	種類	必須
last_date	最後の日	日付	はい

例

[販売日] が 2001 年 12 月 15 日、[請求日] が 2001 年 12 月 17 日の場合、DaysBetween([Sale Date]; [Invoice Date]) は 2 を返します。

6.1.4.8 LastDayOfMonth

説明

月内の最後の日付を返します。

関数グループ

日付と時刻

構文

```
date LastDayOfMonth(date)
```

Input

パラメータ	説明	種類	必須
date	月内の任意の日付	日付	はい

例

[販売日] が 2005 年 12 月 11 日の場合、LastDayOfMonth([販売日]) は、“2005 年 12 月 11 日”を返します。

6.1.4.9 LastDayOfWeek

説明

週内の最後の日付を返します。

関数グループ

日付と時刻

構文

```
date LastDayOfWeek (date)
```

Input

パラメータ	説明	種類	必須
date	週内の任意の日付	日付	はい

注

この関数は、月曜日を週の最初の日として処理します。

例

[販売日] が 2005 年 5 月 11 日の場合、LastDayOfWeek ([販売日]) は、“15 May 2005” (日曜日) を返します。

6.1.4.10 Month

説明

日付の月名を返します。

関数グループ

日付と時刻

構文

```
string Month(date)
```

入力

パラメータ	説明	種類	必須
date	指定する日付	日付	はい

例

[予約日] が 2005 年 12 月 15 日の場合、Month([予約日]) は“12 月”を返します。

6.1.4.11 MonthNumberOfYear

説明

日付の月を返します。

関数グループ

日付と時刻

構文

```
int MonthNumberOfYear(date)
```

入力

パラメータ	説明	種類	必須
date	年内の任意の日付	日付	はい

例

[予約日] が 2005 年 12 月 15 日の場合、MonthNumberOfYear([予約日]) は 12 を返します。

6.1.4.12 MonthsBetween

説明

2つの日付の間の月数を返します。

関数グループ

日付と時刻

構文

```
int MonthsBetween(first_date;last_date)
```

入力

パラメータ	説明	種類	必須
first_date	最初の日	日付	はい
last_date	最後の日	日付	はい

例

[販売日] が 2005 年 12 月 2 日、[請求日] が 2006 年 1 月 2 日の場合、MonthsBetween([販売日]; [請求日]) は 1 を返します。

[販売日] が 2008 年 3 月 31 日、[請求日] が 2008 年 4 月 30 日の場合、MonthsBetween([販売日]; [請求日]) は 1 を返します。

[販売日] が 1993 年 1 月 7 日、[請求日] が 2002 年 11 月 6 日の場合、MonthsBetween([販売日]; [請求日]) は 1 を返します。

6.1.4.13 Quarter

説明

日付の四半期を返します。

関数グループ

日付と時刻

構文

```
int Quarter(date)
```

入力

パラメータ	説明	種類	必須
date	四半期内の任意の日付	日付	はい

例

[予約日] が 2005 年 12 月 15 日の場合、Quarter([予約日]) は 4 を返します。

6.1.4.14 RelativeDate

説明

別の日付に関する日付を返します。

関数グループ

日付と時刻

構文

```
date RelativeDate(start_date;num;period)
```

入力

パラメータ	説明	型	必須
start_date	開始日	日付	○
num	開始日に追加された期間の単位数	数値	○
期間	開始日に追加された期間のタイプ	定義済み	オプション

注

- num パラメータは、定数、関数の数値結果、メジャー値または数値のディメンション値、および整数にすることができます。
- num パラメータは、start_date よりも前の日付を返すように負の値にすることができます。
- 省略する場合、period パラメータは、日数 (DayPeriod) と共に使用します。
- 月数 (SemesterPeriod、QuarterPeriod および MonthPeriod) を追加または削除したときに、指定された日が返された月に存在しない場合は、返された月の最終日が使用されます。
- 期間パラメータに指定可能な値は、以下のとおりです。MillisecondPeriod、SecondPeriod、MinutePeriod、HourPeriod、DayPeriod、WeekPeriod、MonthPeriod、QuarterPeriod、SemesterPeriod、YearPeriod

例

[予約日] が 2005 年 12 月 15 日の場合、`RelativeDate([Reservation Date];2)` は 2005 年 12 月 17 日を返します。

[予約日] が 2007 年 1 月 12 日の場合、`RelativeDate([Reservation Date];-3)` は 2007 年 1 月 9 日を返します。

[予約日] が 2007 年 1 月 12 日の場合、`RelativeDate([Reservation Date];1;MonthPeriod)` は 2007 年 2 月 12 日を返します。

6.1.4.15 TimeDim

説明

`TimeDim` 時系列ディメンションを使用すると、日付型ユニバースオブジェクトから時間軸を作成できます。`Timedim` は、1 つめのパラメータで指定した日付の、2 つめのパラメータで指定した期間にわたるデータを返します。データがない期間がある場合は、空白の期間それぞれの最初の日が返されます。これにより、指定した期間の完全な軸を取得できます。次のことが保証されます。

- 軸は、自然な時間順 (最も古いオブジェクトが最初、最新のオブジェクトが最後) を保つ。
- 軸は、現在のコンテキストの最小日付から最大日付のすべての期間を含む。

i 注記

`TimeDim` 関数を使用してフィルタ、入力コントロール、要素リンク、フィルタ/ドリルバーなどの式をフィルタリングすることはできません。基になる日付ディメンションを直接フィルタリングする必要があります。

関数グループ

日付と時刻

構文

```
TimeDim([Date Type]; Period Type)
```

入力

パラメータ	説明	型	必須
Date Type	InvoiceDate など、レポートの日付オブジェクト。	日付	○
Period Type	次のいずれかの値の、結果の期間。 <ul style="list-style-type: none">DayPeriodMonthPeriodQuarterPeriodYearPeriod 値を選択しないと、デフォルトで DayPeriod が使用されます。このオブジェクトは、データプロバイダオブジェクトとし、レポートオブジェクトから使用できる必要があります。変数にすることはできません。	定義済み	オプション

次の関数と組み合わせて、上の関数を使用します。

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

例

次の 1 つめの表に、特定の日付のみに関係するデータを示します。次のクエリ例で、結果を解釈する方法を示します。

請求日	売上げ
2000/01/03	31,607
2000/01/08	31,244
2000/07/03	38,154

次の式 `DayName (TimeDim ([請求日] ; QuarterPeriod)` は、上の表から日ごとの値を返します。

請求日	売上げ
2000/01/03	31,607
2000/01/08	31,244
2000/04/01	
2000/07/03	38,154

次の表の結果を得るには、Quarter 関数で Timedim 関数の結果の書式を整えて、Quarter (Q1、Q2 など) の結果を返す必要があります。

請求日	売上げ
Q1	62,851
Q2	
Q3	38,154

6.1.4.16 ToDate

説明

文字列を日付に変換します。日付書式をパラメータとして指定し、文字列を日付に変換する方法を Web Intelligence に対して指示します。指定する日付書式は、元の文字列の日付書式と一致している必要があります。使用できる日付書式については、下のリンクを参照してください。

関数グループ

日付と時刻

構文

```
date ToDate(date_string;format)
```

または

```
date ToDate(date_string;"INPUT_DATE_TIME")
```

i 注記

ユーザによって優先表示ロケールが異なる可能性があるシナリオでは、(特定のロケールで) 書式を固定化することは適切ではありません。この場合、上記 2 番目の例のように INPUT_DATE_TIME を使用します。

入力

パラメータ	説明	型	必須
date_string	日付として解釈される文字列。	文字列	○
format	文字列ごとに使用される日付形式。 “INPUT_DATE_TIME” を使用して、優先表示ロケールの書式を使用します。	文字列	○*

* 上述の注を参照してください。ニーズに応じ、この書式または INPUT_DATE_TIME を使用します。

例

ToDate("12/15/2002";"MM/dd/yyyy") は“12”を月、“15”を日、および“2002”を年として解釈します。

ToDate("Dec/02";"Mmm/yy") は“Dec”を月の短縮名、“02”を年の下二桁として解釈します。

ToDate("15-December-02";"dd-Mmmm-yy") は“15”を日、“December”を月、および“02”を年の下二桁として解釈します。

ToDate("12/15/02 11:00:00";"INPUT_DATE_TIME") は、“12/15/02 11:00:00”をユーザのマシンの優先表示ロケールで使用されている書式で解釈します。

i 注記

- INPUT_DATE_TIME では、date_string 入力文字列で日付と時刻の両方を指定する必要があります。
- date_string を指定された書式の有効な日付として解釈できない場合、ToDate 式は #ERROR を返します。
- セルにおける日付の表示方法は、そのセルで選択された日付書式に依存します。たとえば、選択された日付書式が "MM/dd/yyyy" である場合、ToDate("Dec/15/02";"MMM/dd/yy") は 12/15/2002 と表示されます。

関連情報

[カスタム書式 \[37 ページ\]](#)

6.1.4.17 Week

説明

何週目にあたるかを数値 (1 ~ 53) で返します。

関数グループ

日付と時刻

構文

```
int Week(date)
```

入力

パラメータ	説明	種類	必須
date	指定する日付	日付	はい

例

[予約日]が2004年1月4日(2004年の最初の週)の場合、Week([予約日])は1を返します。

6.1.4.18 Year

説明

日付の年を返します。

関数グループ

日付と時刻

構文

```
int Year(date)
```

入力

パラメータ	説明	種類	必須
date	指定する日付	日付	はい

例

[予約日]が2005年12月15日の場合、`Year([予約日])` は2005を返します。

6.1.5 データプロバイダ関数

6.1.5.1 Connection

説明

データプロバイダが使用したデータベース接続のパラメータを返します。

関数グループ

データプロバイダ

構文

```
string Connection(dp)
```

入力

パラメータ	説明	種類	必須
dp	データプロバイダ	データプロバイダ	はい

注

- データプロバイダの名前は角かっこで囲む必要があります。
- セキュリティ上の理由から、関数の出力にはデータベースホスト名、ユーザ名およびユーザパスワードは含まれません。

6.1.5.2 DataProvider

説明

レポートオブジェクトを含むデータプロバイダの名前を返します。

関数グループ

データプロバイダ

構文

```
string DataProvider(obj)
```

入力

パラメータ	説明	型	必須
obj	レポートオブジェクト	レポートオブジェクト	○

例

`DataProvider([総売上げ])` では、“売上げ”という名前のデータプロバイダに[総売上げ]というメジャーが含まれている場合、“売上げ”を返します。

i 注記

`DataProvider` は、データプロバイダ名を返すためにオブジェクト名を必要とします。`DataProvider` のパラメータとして、ディメンション変数など、オブジェクト名を提供しない他の関数を使用する場合、`DataProvider` 関数ではエラーが返されます。

6.1.5.3 DataProviderKeyDate

説明

データプロバイダのキー日付を返します。

関数グループ

データプロバイダ

構文

```
date DataProviderKeyDate(dp)
```

入力

パラメータ	説明	種類	必須
dp	データプロバイダ	データプロバイダ	はい

注

- データプロバイダの名前は角かっこで囲む必要があります。
- 返されたキー日付はドキュメントロケールに従って書式設定されます。

例

売上げデータプロバイダのキー日付が 2007 年 8 月 3 日の場合、`DataProviderKeyDate([売上げ])` は 2007 年 8 月 3 日を返します。

6.1.5.4 DataProviderKeyDateCaption

説明

データプロバイダのキー日付のキャプションを返します。

関数グループ

データプロバイダ

構文

```
string DataProviderKeyDateCaption(dp)
```

入力

パラメータ	説明	種類	必須
dp	データプロバイダ	データプロバイダ	はい

注

データプロバイダの名前は角かっこで囲む必要があります。

例

売上げデータプロバイダのキー日付のキャプションが“現在のカレンダー日付”の場合、`DataProviderKeyDateCaption([売上げ])` は“現在のカレンダー日付”を返します。

6.1.5.5 DataProviderSQL

説明

データプロバイダによって生成された SQL を返します。

関数グループ

データプロバイダ

構文

```
string DataProviderSQL(dp)
```

入力

パラメータ	説明	種類	必須
dp	データプロバイダ	データプロバイダ	はい

注

データプロバイダの名前は角かっこで囲む必要があります。

例

`DataProviderSQL([Query 1])` は、データプロバイダの SQL が `SELECT country.country_name FROM country` の場合、`SELECT country.country_name FROM country` を返します。

6.1.5.6 DataProviderType

説明

データプロバイダの種類を返します。

関数グループ

データプロバイダ

構文

```
string DataProviderType (dp)
```

入力

パラメータ	説明	種類	必須
dp	データプロバイダ	データプロバイダ	はい

注

- DataProviderType は、ユニバースデータプロバイダには“ユニバース”を返し、個人用データプロバイダには“個人用データ”を返します。
- データプロバイダの名前は角かっこで囲む必要があります。

例

“売上げ”データプロバイダがユニバースに基づいている場合、DataProviderType([売上げ]) は“ユニバース”を返します。

6.1.5.7 IsPromptAnswered

説明

プロンプトが回答されたかどうかを判別します。

関数グループ

データプロバイダ

構文

```
bool IsPromptAnswered([dp;]prompt_string)
```

入力

パラメータ	説明	型	必須
dp	プロンプトが設定されているデータプロバイダ	データプロバイダ	×
prompt_string	プロンプトテキスト	文字列	○

注

- データプロバイダの名前は角かっこで囲む必要があります。
- IsPromptAnswered は、If 関数と共に使用できる論理値を返します。
- IsPromptAnswered は、列に直接配置された場合に、整数を返します (1=true、0=false)。この整数は、論理数値書式を使用して書式設定できます。

例

テキスト“都市の選択”によって識別されるプロンプトが回答されている場合、IsPromptAnswered ("都市の選択") は TRUE を返します。

[売上げ] データプロバイダのテキスト“都市の選択”によって識別されるプロンプトが回答されている場合、IsPromptAnswered ([売上げ]; "都市の選択") は TRUE を返します。

6.1.5.8 LastExecutionDate

説明

データプロバイダが最後に更新された日付を返します。

関数グループ

データプロバイダ

構文

```
date LastExecutionDate (dp)
```

入力

パラメータ	説明	種類	必須
dp	データプロバイダ	データプロバイダ	はい

注

- レポートにデータプロバイダが1つしかない場合、dp パラメータは省略できます。
- データプロバイダの名前は角かっこで囲む必要があります。
- DataProvider 関数を使用して、データプロバイダへの参照を提供できます。

例

LastExecutionDate([売上げクエリ]) は、売上げクエリのデータプロバイダが最後に更新された日付が 2002 年 3 月 4 日の場合、“3/4/2002”を返します。

関連情報

[DataProvider](#) [120 ページ]

6.1.5.9 LastExecutionDuration

説明

データプロバイダの最後の更新にかかった時間を秒単位で返します。

関数グループ

データプロバイダ

構文

```
num LastExecutionDuration (dp)
```

入力

パラメータ	説明	種類	必須
dp	データプロバイダ	データプロバイダ	はい

注

データプロバイダの名前は角かっこで囲む必要があります。

例

“売上げ”データプロバイダが最後に実行されたデータを返すのに 3 秒かかった場合、`LastExecutionDuration([売上げ])` は 3 を返します。

6.1.5.10 LastExecutionTime

説明

データプロバイダが最後に更新された時刻を返します。

関数グループ

データプロバイダ

構文

```
time LastExecutionTime(dp)
```

入力

パラメータ	説明	種類	必須
dp	データプロバイダ	データプロバイダ	はい

注

- レポートにデータプロバイダが1つしかない場合、dp パラメータは省略できます。
- DataProvider 関数を使用して、データプロバイダへの参照を提供できます。
- データプロバイダの名前は角カッコで囲む必要があります。

例

LastExecutionTime([売上げクエリ]) は、売上げクエリのデータプロバイダが最後に更新された時刻が 2:48:00PM の場合、“2:48:00 PM”を返します。

関連情報

[DataProvider](#) [120 ページ]

6.1.5.11 NumberOfDataProviders

説明

レポート内のデータプロバイダの数を返します。

関数グループ

データプロバイダ

構文

```
int NumberOfDataProviders()
```

例

`NumberOfDataProviders()` は、レポートに 2 つのデータプロバイダがある場合、2 を返します。

6.1.5.12 NumberOfRows

説明

指定したデータプロバイダの行数を返します。

関数グループ

データプロバイダ

構文

```
int NumberOfRows (dp)
```

入力

パラメータ	説明	種類	必須
dp	データプロバイダ	データプロバイダ	はい

注

- データプロバイダの名前は角かっこで囲む必要があります。
- `DataProvider` 関数を使用して、データプロバイダへの参照を提供できます。

例

`NumberOfRows ([クエリ 1])` は、`[クエリ1]` データプロバイダに 10 行ある場合、10 を返します。

関連情報

[DataProvider \[120 ページ\]](#)

6.1.5.13 RefValueDate

説明

データ追跡に使用した参照データの日付を返します。

関数グループ

データプロバイダ

構文

```
date RefValueDate()
```

例

RefValueDate() は、参照日付が 2008 年 12 月 15 日の場合に 2008 年 12 月 15 日を返します。

6.1.5.14 RefValueUserReponse

説明

参照データが現在のデータであった場合に、プロンプトへの応答を返します。

関数グループ

データプロバイダ

構文

```
string RefValueUserResponse([dp;]prompt_string[;Index])
```

入力

パラメータ	説明	種類	必須
dp	データプロバイダ	データプロバイダ	いいえ
prompt_string	プロンプトテキスト	文字列	はい
Index	プロンプト値のデータベースのプライマリキーを返すよう、関数に指示します。	キーワード	いいえ

注

- データ追跡がアクティブでない場合、この関数は空の文字列を返します。
- データプロバイダの名前は角かっこで囲む必要があります。
- `DataProvider` 関数を使用して、データプロバイダへの参照を提供できます。
- プロンプトへの回答で複数の値を選択すると、セミicolonで区切られた値の一覧で構成される文字列 (または `Index` 演算子が指定されている場合はプライマリー) が返されます。

例

`RefValueUserResponse ("表示する都市を指定")` は、参照データが現在のデータであるときに、ユーザが“表示する都市を指定”プロンプトに“Los Angeles”を入力していた場合、“Los Angeles”を返します。

`RefValueUserResponse ([販売クエリ]; "表示する都市を指定")` は、参照データが現在のデータであるときに、ユーザが“販売クエリ”データプロバイダの“表示する都市を指定”プロンプトに“Los Angeles”を入力していた場合、“Los Angeles”を返します。

6.1.5.15 ServerValue

説明

メジャーのデータベース値を返します。

関数グループ

データプロバイダ

構文

```
num ServerValue([measure])
```

入力

パラメータ	説明	型	必須
measure	任意のメジャー	メジャー	○

注

- `ServerValue` は、メジャーの計算に使用するディメンションまたは階層に適用されている、すべてのローカルフィルタを無視します。

例

`ServerValue([インターネット売上金額])` は、"インターネット売上金額" メジャー のデータベース値を返します。

6.1.5.16 UniverseName

説明

データプロバイダのベースとなっているユニバース名を返します。

関数グループ

データプロバイダ

構文

```
string UniverseName(dp)
```

入力

パラメータ	説明	種類	必須
dp	データプロバイダ	データプロバイダ	はい

注

- データプロバイダの名前が変更された場合、式内の `dp` の値は自動的に更新されます。たとえば、データプロバイダの名前が "Q1" に変更されると、式は `UniverseName([Q1])` になります。
- データプロバイダの名前は角かっこで囲む必要があります。
- `DataProvider` 関数を使用して、データプロバイダへの参照を提供できます。

例

[Query 1] データプロバイダが eFashion ユニバースをベースにしている場合、`UniverseName([Query 1])` は、“eFashion”を返します。

関連情報

[DataProvider \[120 ページ\]](#)

6.1.5.17 UserResponse

説明

プロンプトに対する回答を返します。

関数グループ

データプロバイダ

構文

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

入力

パラメータ	説明	種類	必須
dp	データプロバイダ	データプロバイダ	いいえ
prompt_string	プロンプトテキスト	文字列	はい
Index	プロンプト値のデータベースのプライマリーキーを返すよう、関数に指示します。	キーワード	いいえ

注

- データプロバイダの名前は角かっこで囲む必要があります。
- `DataProvider` 関数を使用して、データプロバイダへの参照を提供できます。
- プロンプトへの回答で複数の値を選択すると、セミコロンで区切られた値の一覧で構成される文字列 (または `Index` 演算子が指定されている場合はプライマリーキー) が返されます。

例

`UserResponse("表示する都市を指定")` は、ユーザが“表示する都市を指定”プロンプトに“Los Angeles”を入力していた場合、“Los Angeles”を返します。

`UserResponse([販売クエリ]; "表示する都市を指定")` は、“販売クエリ”データプロバイダの“表示する都市を指定”プロンプトに“Los Angeles”を入力していた場合、“Los Angeles”を返します。

`UserResponse([販売クエリ]; "表示する都市を指定"; Index)` は、“販売クエリ”データプロバイダの“表示する都市を指定”プロンプトに“Los Angeles”を入力し、Los Angeles のデータベースプライマリーキーが 23 の場合、23 を返します。

6.1.6 ドキュメント関数

6.1.6.1 DocumentAuthor

説明

ドキュメント作成者の InfoView ログイン名を返します。

関数グループ

ドキュメント

構文

```
string DocumentAuthor()
```

例

ドキュメント作成者のログイン名が“gkn”の場合、DocumentAuthor() は“gkn”を返します。

6.1.6.2 DocumentCreationDate

説明

ドキュメントの作成日を返します。

関数グループ

ドキュメント

構文

```
date DocumentCreationDate()
```

例

ドキュメントを作成したのが 2008 年 12 月 15 日の場合、`DocumentCreationDate()` は、“15 December 2008”を返します。

6.1.6.3 DocumentCreationTime

説明

ドキュメントの作成時刻を返します。

関数グループ

ドキュメント

構文

```
時刻 DocumentCreationTime()
```

例

ドキュメントの作成時刻が 11:15 の場合、`DocumentCreationTime()` は 11:15 を返します。

6.1.6.4 DocumentDate

説明

ドキュメントの最終保存日を返します。

関数グループ

ドキュメント

構文

```
date DocumentDate()
```

例

ドキュメントを最後に保存した日付が 2005 年 8 月 8 日の場合、`DocumentDate()` は、“2005 年 8 月 8 日”を返します。

6.1.6.5 DocumentName

説明

ドキュメントの名前を返します。

関数グループ

ドキュメント

構文

```
string DocumentName()
```

例

ドキュメント名が“Sales Report”の場合、`DocumentName()` は“Sales Report”を返します。

6.1.6.6 DocumentOwner

説明

ドキュメント所有者 (ドキュメントを最後に保存したユーザ) の BI ラウンチパッドログオン/ユーザ名を返します。(ドキュメントの元の作成者を返すには、DocumentAuthor 関数を使用します)。

関数グループ

ドキュメント

構文

```
string DocumentOwner()
```

例

ドキュメントを最後に保存したユーザのユーザ名またはログイン名が "gkn" である場合、DocumentOwner() は、"gkn" を返します。

6.1.6.7 DocumentPartiallyRefresed

説明

ドキュメントが部分的に最新表示されているかどうかを返します。

関数グループ

ドキュメント

構文

```
bool DocumentPartiallyRefreshed()
```

注

`DocumentPartiallyRefreshed` は、`If` 関数で利用できる論理値を返します。

例

ドキュメントが部分的に最新表示されている場合、`DocumentPartiallyRefreshed()` は `True` を返します。

6.1.6.8 DocumentTime

説明

ドキュメントが最後に保存された時刻を返します。

関数グループ

ドキュメント

構文

```
時刻 DocumentTime()
```

注

返される時刻の書式は、セルの書式によって異なります。

例

ドキュメントの最終保存時刻が 15:45 の場合、DocumentTime () は 15:45 を返します。

6.1.6.9 DrillFilters

説明

ドリルモードで宣言したレポートのドキュメントまたはオブジェクトに適用されたドリルフィルタの結果を返します。ドキュメント内で別のレポートを宣言することもできます。レポートを宣言しない場合、現在のアクティブなレポートが使用されます。

関数グループ

ドキュメント

構文

```
string DrillFilters([obj|separator[;report]])
```

入力

パラメータ	説明	種類	必須
obj	レポートオブジェクト	レポートオブジェクト	obj または separator が 必要
separator	ドリルフィルタの区切り記号	文字列	obj または separator が 必要
report	オプション。使用するレポートの名前。ドキュメント内にある必要があります。宣言されたレポートがない場合、現在のレポートが使用されます。	文字列	obj または separator が 必要

注

- `DrillFilters` は、`DrillFilters` セルを挿入することによって式全体を手動で入力することなく、直接挿入することができます。
- オブジェクトを指定しない場合は、ドキュメントに適用されているすべてのドリルフィルタを返します。

例

[都道府県] オブジェクトの値を大阪に限定するドリルフィルタがドキュメントに適用されている場合、`DrillFilters()` は大阪を返します。

[都道府県] オブジェクトの値を“大阪”に、[年] オブジェクトの値を“2003”に限定するドリルフィルタがドキュメントに適用されている場合、`DrillFilters()` は“大阪 - 2003”を返します。

[都道府県] オブジェクトの値を“大阪”に、[年] オブジェクトの値を“2003”に限定するドリルフィルタがドキュメントに適用されている場合、`DrillFilters("/")` は“大阪/2003”を返します。

[四半期] オブジェクトの値を“第 3”に限定するドリルフィルタがドキュメントに適用されている場合、`DrillFilters` ([四半期]) は“第 3”を返します。

6.1.6.10 PromptSummary

説明

ドキュメント内のすべてのプロンプトのプロンプトテキストとユーザの応答を返します。

関数グループ

ドキュメント

構文

```
string PromptSummary()
```

例

PromptSummary 関数の出力例を以下に示します。

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

6.1.6.11 QuerySummary

説明

ドキュメントのクエリに関する情報を返します。

関数グループ

ドキュメント

構文

```
string QuerySummary([dp])
```

入力

パラメータ	説明	種類	必須
dp	データプロバイダ	データプロバイダ	いいえ

注

- データプロバイダの名前は角かっこで囲む必要があります。

例

`QuerySummary()` は、ドキュメント内のすべてのクエリに関する情報を返します。

`QuerySummary([クエリ 1])` は、`[クエリ 1]` データプロバイダに基づくクエリの情報を返します。

出力例

```
Query 1:
Universe: eFashion
Last execution time: 1s
NB of rows: 34500
Result objects: State, Year, Sales Revenue
Scope of analysis: State, City, Year, Quarter,
Month
Filters:
(State inlist{"US";"France";}
And (Sales Revenue Greater Than 1000000
Or Sales Revenue Less Than 10000))
Query 2:
Source file: D:¥Data¥datacar.xls
Result objects: State, Year, Sales Revenue
```

6.1.6.12 ReportFilter

説明

オブジェクトまたはレポートに適用されるレポートフィルタを返します。

関数グループ

ドキュメント

構文

```
string ReportFilter(obj)
```


入力

パラメータ	説明	種類	必須
obj	レポートオブジェクト	レポートオブジェクト	はい

例

国オブジェクトを“US”に限定するレポートフィルタがある場合、`ReportFilter([国])` は“US”を返します。

6.1.6.13 ReportFilterSummary

説明

ドキュメントまたはレポート内のレポートフィルタの要約を返します。

関数グループ

ドキュメント

構文

```
string ReportFilterSummary(report_name)
```

入力

パラメータ	説明	型	必須
report_name	レポートの名前	文字列	×

注

`report_name` を省略すると、`ReportFilterSummary` はドキュメント内のすべてのレポートにおけるすべてのレポートフィルタの要約を返します。

例

`ReportFilterSummary()` は、ドキュメント内のすべてのレポートフィルタに関する情報を返します。

`ReportFilterSummary("レポート 1")` は、“レポート1”レポートのレポートフィルタに関する情報を返します。

`ReportFilterSummary` 関数の出力例を以下に示します。

```
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))
```

6.1.7 論理関数

6.1.7.1 Even

説明

数値が偶数かどうかを判別します。

関数グループ

論理

構文

```
bool Even(number)
```

入力

パラメータ	説明	種類	必須
number	任意の数値	数値	はい

注

- `Even` は、`If` 関数で利用できる論理値を返します。
- `Even` を列に直接配置する場合は、整数を返します (1=true、0=false)。この整数は、論理数値書式を使用して書式設定できます。

例

`Even(4)` は `True` を返します。

`Even(3)` は `False` を返します。

`Even(23.2)` は `False` を返します。

`Even(-4)` は `True` を返します。

`Even(-2.2)` は `False` を返します。

6.1.7.2 IsDate

説明

値が日付型かどうかを判別します。

関数グループ

論理

構文

```
bool IsDate(obj)
```

入力

パラメータ	説明	型	必須
obj	任意のレポートオブジェクト	レポートオブジェクト	○

注

- `IsDate` は、`If` 関数で利用できる論理値を返します。
- `IsDate` を列に直接配置する場合は、整数を返します (1=true、0=false)。この整数は、論理数値書式を使用して書式設定できます。

例

`IsDate([予約日])` は、`[予約日]` が日付である場合に `TRUE` を返します。

あるいは、以下のどちらかは、`[Reservation Date]` が日付である場合に `"Date"` を返します。

- `If(IsDate([Reservation Date])) Then "Date" Else "Not a date"`
- `If IsDate([Reservation Date]) Then "Date" Else "Not a date"`

関連情報

[If...Then...Else \[208 ページ\]](#)

6.1.7.3 IsError

説明

オブジェクトがエラーを返すかどうかを判別します。

関数グループ

論理

構文

```
bool IsError(obj)
```

入力

パラメータ	説明	種類	必須
obj	任意のレポートオブジェクト	レポートオブジェクト	はい

注

- IsError は、If 関数で利用できる論理値を返します。
- IsError を列に直接配置すると、整数を返します (1=true、0=false)。この整数は、論理数値書式を使用して書式設定できます。

例

変数 [売上げ] がエラーを返さない場合、IsError([売上げ]) は FALSE を返します。

変数 [平均顧客数] が #DIV/0 エラー(0 による除算)を返す場合、IsError([平均顧客数]) は TRUE を返します。

If IsError([平均顧客数]) Then "エラー" Else "エラーでない" は、変数 [平均顧客数] が #DIV/0 エラー(0 による除算)を返す場合、"エラー"を返します。

関連情報

[If...Then...Else \[208 ページ\]](#)

6.1.7.4 IsLogical

説明

値が論理型であるかどうかを判別します。

関数グループ

論理

構文

```
bool IsLogical(obj)
```

入力

パラメータ	説明	種類	必須
obj	任意のレポートオブジェクト	レポートオブジェクト	はい

注

- IsLogical は、If 関数で利用できる論理値を返します。
- IsLogical を列に直接配置する場合は、整数を返します (1=true、0=false)。この整数は、論理数値書式を使用して書式設定できます。

例

IsLogical(IsString([国])) は TRUE を返します。

IsLogical([国]) は、[国] が論理型以外のデータ型を返す場合、FALSE を返します。

If IsLogical(IsDate([国])) Then "論理型" Else "論理型でない" は、“論理型”を返します。

関連情報

[If...Then...Else \[208 ページ\]](#)

6.1.7.5 IsNull

説明

値が NULL であるかどうかを判別します。

関数グループ

論理

構文

```
bool IsNull(obj)
```

入力

パラメータ	説明	種類	必須
obj	任意のレポートオブジェクト	レポートオブジェクト	はい

注

- IsNull は、If 関数で使用できる論理値を返します。
- IsNull を列に直接配置する場合は、整数を返します (1=true、0=false)。この整数は、論理数値書式を使用して書式設定できます。

例

変数 [売上げ] が NULL でない場合、IsNull([売上げ]) は FALSE を返します。

変数 [平均顧客数] が NULL の場合、IsNull([平均顧客数]) は TRUE を返します。

関連情報

[If...Then...Else \[208 ページ\]](#)

6.1.7.6 IsNumber

説明

値が数値型かどうかを判別します。

関数グループ

論理

構文

```
bool IsNumber(obj)
```

入力

パラメータ	説明	種類	必須
obj	任意のレポートオブジェクト	レポートオブジェクト	はい

注

- IsNumber は、If 関数で使用できる論理値を返します。
- IsNumber を列に直接配置する場合は、整数を返します (1=true、0=false)。この整数は、論理数値書式を使用して書式設定できます。

例

変数 [売上げ] が数値の場合、IsNumber([売上げ]) は TRUE を返します。

変数 [顧客名] が数値ではない場合、IsNumber ([顧客名]) は FALSE を返します。

If IsNumber ([顧客名]) Then "数値" Else "数値でない" は、変数 [顧客名] が数値でない場合、“数値でない”を返します。

関連情報

[If...Then...Else \[208 ページ\]](#)

6.1.7.7 IsString

説明

値が文字列かどうかを判別します。

関数グループ

論理

構文

```
bool IsString(obj)
```

入力

パラメータ	説明	種類	必須
obj	任意のレポートオブジェクト	レポートオブジェクト	はい

注

- IsString は、If 関数で利用できる論理値を返します。
- IsString を列に直接配置する場合は、整数を返します (1=true、0=false)。この整数は、論理数値書式を使用して書式設定できます。

例

変数[売上げ]が文字列でない場合、IsString([売上げ])はFALSEを返します。

変数[顧客]が文字列の場合、IsString([顧客])はTRUEを返します。

変数[顧客名]が文字列の場合、If IsString([顧客名]) Then "文字列" Else "文字列でない" は“文字列”を返します。

関連情報

[If...Then...Else \[208 ページ\]](#)

6.1.7.8 IsTime

説明

変数が時間変数かどうかを判別します。

関数グループ

論理

構文

```
bool IsTime(obj)
```

入力

パラメータ	説明	種類	必須
obj	任意のレポートオブジェクト	レポートオブジェクト	はい

注

- `IsTime` は、`If` 関数で利用できる論理値を返します。
- `IsTime` を列に直接配置する場合は、整数を返します (1=true、0=false)。この整数は、論理数値書式を使用して書式設定できます。

例

変数[予約時間]が時間変数の場合、`IsTime([予約時間])` は TRUE を返します。

変数[平均顧客数]が時間変数ではない場合、`IsTime([平均顧客数])` は FALSE を返します。

`If IsTime([平均顧客数]) Then "時間" Else "時間でない"` は、変数 [平均顧客数] が時間変数でない場合に“時間でない”を返します。

関連情報

[If...Then...Else \[208 ページ\]](#)

6.1.7.9 Odd

説明

変数が奇数かどうかを判別します。

関数グループ

論理

構文

```
bool Odd(number)
```

入力

パラメータ	説明	種類	必須
number	任意の数値	数値	はい

注

- Odd は、If 関数で使用する論理値を返します。
- Odd を列に直接配置する場合は、整数を返します (1=true、0=false)。この整数は、論理数値書式を使用して書式設定できます。
- Odd は、数値の小数部分を無視します。

例

Odd (5) は TRUE を返します。

Odd (4) は FALSE を返します。

Odd (23.2) は TRUE を返します。

Odd (24.2) は TRUE を返します。

Odd (-23.2) は TRUE を返します。

Odd (-24.2) は TRUE を返します。

関連情報

[If...Then...Else \[208 ページ\]](#)

6.1.8 数値関数

6.1.8.1 Abs

説明

引数の絶対値を返します。

関数グループ

数値

構文

```
num Abs (number)
```

入力

パラメータ	説明	種類	必須
number	任意の数値	数値	はい

例

Abs (25) は、25 を返します。

Abs (-11) は、11 を返します。

6.1.8.2 Ceil

説明

数値を切り上げた整数を返します。

関数グループ

数値

構文

```
num Ceil (number)
```

入力

パラメータ	説明	種類	必須
number	任意の数値	数値	はい

例

`Ceil(2.4)` は、3 を返します。

`Ceil(3.1)` は、4 を返します。

`Ceil(-3.1)` は、-3 を返します。

6.1.8.3 Cos

説明

角度のコサインを返します。

関数グループ

数値

構文

```
num Cos(angle)
```

入力

パラメータ	説明	種類	必須
angle	ラジアンで指定した角度	数値	はい

例

Cos(180) は、-0.6 を返します。

6.1.8.4 EuroConvertFrom

説明

ユーロから他の通貨に変換します。

関数グループ

数値

構文

```
num EuroConvertFrom(euro_amount;curr_code;round_level)
```

入力

パラメータ	説明	種類	必須
euro_amount	ユーロでの金額	数値	はい
curr_code	変換する通貨の ISO コード	文字列	はい
round_level	四捨五入する小数点以下の桁数	数値	はい

注

指定できる通貨コードは、2002 年 1 月の通貨廃止に先立ち固定レートが適用された次の EU 加盟国 12 カ国のいずれかの通貨です。それ以外の通貨が指定されると、#ERROR を返します。通貨は次のとおりです。

BEF	ベルギーフラン
-----	---------

DEM	ドイツマルク
GRD	ギリシャドラクマ
ESP	スペインペセタ
FRF	フランスフラン
IEP	アイルランドポンド
ITL	イタリアリラ
LUF	ルクサンブルグフラン
NLG	オランダギルダ
ATS	オーストリアシリング
PTS	ポルトガルエスクード
FIM	フィンランドマルカ

例

`EuroConvertFrom(1000;"FRF";2)` は、6559.57 を返します。

`EuroConvertFrom(1000;"FRF";1)` は、6559.60 を返します。

`EuroConvertFrom(1000.04;"DEM";2)` は、1955.83 を返します。

`EuroConvertFrom(1000.04;"DEM";1)` は、1955.80 を返します。

関連情報

[数値の四捨五入と切り捨て \[245 ページ\]](#)

6.1.8.5 EuroConvertTo

説明

他の通貨からユーロに変換します。

関数グループ

数値

構文

```
num EuroConvertTo(noneuro_amount;curr_code;round_level)
```

入力

パラメータ	説明	種類	必須
noneuro_amount	他の通貨の金額	数値	はい
curr_code	他の通貨の ISO コード	文字列	はい
round_level	四捨五入する小数点以下の桁数	数値	はい

例

EuroConvertTo(6559;"FRF";2) は、999.91 を返します。

EuroConvertTo(6559;"FRF";1) は、999.90 を返します。

EuroConvertTo(1955;"DEM";2) は、999.58 を返します。

EuroConvertTo(1955;"DEM";1) は、999.60 を返します。

注

指定できる通貨コードは、2002 年 1 月の通貨廃止に先立ち固定レートが適用された次の EU 加盟国 12 カ国のいずれかの通貨です。それ以外の通貨が指定されると、#ERROR を返します。通貨は次のとおりです。

BEF	ベルギーフラン
DEM	ドイツマルク
GRD	ギリシャドラクマ
ESP	スペインペセタ
FRF	フランスフラン
IEP	アイルランドポンド
ITL	イタリアリラ
LUF	ルクサンブルグフラン
NLG	オランダギルダ

ATS	オーストリアシリング
PTS	ポルトガルエスクード
FIM	フィンランドマルカ

関連情報

[数値の四捨五入と切り捨て \[245 ページ\]](#)

6.1.8.6 EuroFromRoundError

説明

ユーロから他の通貨への換算時の誤差を返します。

関数グループ

数値

構文

```
num EuroFromRoundError(euro_amount;curr_code;round_level)
```

入力

パラメータ	説明	型	必須
euro_amount	ユーロでの金額	数値	○
curr_code	変換する通貨の ISO コード	文字列	○
round_level	四捨五入する小数点以下の桁数	数値	○

出力

換算時の四捨五入誤差

例

`EuroFromRoundError(1000;"FRF";2)` は 0 を返します (四捨五入しない換算と、小数点第 2 位で四捨五入する換算には違いはありません)。

`EuroFromRoundError(1000;"FRF";1)` は 0.03 を返します (四捨五入しない換算は 6559.57 です。小数点第 1 位で四捨五入する換算は 6559.60 です。四捨五入のエラーは 0.03 です)。

`EuroFromRoundError(1000;"DEM";2)` は 0 を返します (四捨五入しない換算と、小数点第 2 位で四捨五入する換算には違いはありません)。

`EuroFromRoundError(1000;"DEM";1)` は -0.01 を返します (四捨五入しない換算は 1955.83 です。小数点第 1 位で四捨五入する換算は 1995.80 です。四捨五入のエラーは -0.03 です)。

注

指定できる通貨コードは、2002 年 1 月の通貨廃止に先立ち固定レートが適用された次の EU 加盟国 12 カ国のいずれかの通貨です。それ以外の通貨が指定されると、#ERROR を返します。通貨は次のとおりです。

BEF	ベルギーフラン
DEM	ドイツマルク
GRD	ギリシャドラクマ
ESP	スペインペセタ
FRF	フランスフラン
IEP	アイルランドポンド
ITL	イタリアリラ
LUF	ルクセンブルグフラン
NLG	オランダギルダ
ATS	オーストリアシリング
PTS	ポルトガルエスクード
FIM	フィンランドマルカ

関連情報

[数値の四捨五入と切り捨て \[245 ページ\]](#)

6.1.8.7 EuroToRoundError

説明

ユーロへの換算時の誤差を返します。

関数グループ

数値

構文

```
num EuroToRoundError (noneuro_amount; curr_code; round_level)
```

入力

パラメータ	説明	型	必須
euro_amount	他の通貨の金額	数値	○
curr_code	他の通貨の ISO コード	文字列	○
round_level	四捨五入する小数点以下の桁数	数値	○

例

`EuroToRoundError(6559;"FRF";2)` は 0 を返します。(四捨五入しない換算と、小数点第 2 位で四捨五入する換算には違いはありません。)

`EuroToRoundError(6559;"FRF";1)` は -0.01 を返します。(四捨五入しない換算は 999.91 です。小数点第 1 位で四捨五入する換算は 999.90 です。四捨五入のエラーは -0.01 です。)

`EuroToRoundError(1955;"DEM";2)` は 0 を返します。(四捨五入しない換算と、小数点第 2 位で四捨五入する換算には違いはありません。)

`EuroToRoundError(1955;"DEM";1)` は 0.02 を返します。(四捨五入しない換算は 999.58 です。小数点第 1 位で四捨五入する換算は 999.60 です。四捨五入のエラーは 0.02 です。)

注

指定できる通貨コードは、2002 年 1 月の通貨廃止に先立ち固定レートを適用された次の EU 加盟国 12 カ国のいずれかの通貨です。それ以外の通貨が指定されると、#ERROR を返します。通貨は次のとおりです。

BEF	ベルギーフラン
DEM	ドイツマルク
GRD	ギリシャドラクマ
ESP	スペインペセタ
FRF	フランスフラン
IEP	アイルランドポンド
ITL	イタリアリラ
LUF	ルクサンブルグフラン
NLG	オランダギルダ
ATS	オーストリアシリング
PTS	ポルトガルエスクード
FIM	フィンランドマルカ

関連情報

[数値の四捨五入と切り捨て \[245 ページ\]](#)

6.1.8.8 Exp

説明

指数 (e を底とした数値の累乗) を返します。

関数グループ

数値

構文

```
num Exp(power)
```

入力

パラメータ	説明	種類	必須
power	指数	数値	はい

注

指数は、 e (2.718...) を底とした数値の累乗である定数です。

例

`Exp(2.2)` は 9.03 を返します。

6.1.8.9 Fact

説明

数値の階乗を返します。

関数グループ

数値

構文

```
int Fact(number)
```

入力

パラメータ	説明	種類	必須
number	任意の数値	数値	はい

注

number の階乗は、1 から number までのすべての整数の積です。

例

Fact (4) は 24 を返します。

Fact (5.9) は 120 を返します。

6.1.8.10 Floor

説明

数値を切り下げた整数を返します。

関数グループ

数値

構文

```
int Floor(number)
```

入力

パラメータ	説明	種類	必須
number	任意の数値	数値	はい

例

`Floor(24.4)` は 24 を返します。

6.1.8.11 Ln

説明

数値の自然対数を返します。

関数グループ

数値

構文

```
num Ln(number)
```

入力

パラメータ	説明	型	必須
number	任意の数値	数値	○

例

`Ln(10)` は 2.3 を返します。

6.1.8.12 Log

説明

指定した数値を底とする対数を返します。

関数グループ

数値

構文

```
num Log (number;base)
```

入力

パラメータ	説明	種類	必須
number	任意の数値	数値	はい
base	対数の底とする数値	数値	はい

例

Log (125;5) は 3 を返します。

6.1.8.13 Log10

説明

10 を底とした対数を返します。

関数グループ

数値

構文

```
num Log10 (number)
```

入力

パラメータ	説明	種類	必須
number	任意の数値	数値	はい

例

Log10 (100) は 2 を返します。

6.1.8.14 Mod

説明

2つの数値の除算から余りを返します。

関数グループ

数値

構文

```
num Mod (dividend;divisor)
```

入力

パラメータ	説明	種類	必須
dividend	被除数	数値	はい
divisor	除数	数値	はい

例

`Mod (10;4)` は 2 を返します。

`Mod (10.2;4.2)` は 1.8 を返します。

6.1.8.15 Power

説明

指定した数値の累乗を返します。

関数グループ

数値

構文

```
num Power (number;power)
```

入力

パラメータ	説明	種類	必須
number	累乗を求める数値	数値	はい
power	指数	数値	はい

例

`Power(10;2)` は 100 を返します。

6.1.8.16 Rank

説明

メジャーをディメンション別に順位付けします

関数グループ

数値

構文

```
int Rank(measure; [ranking_dims] [; Top|Bottom] [; (reset_dims)])
```

入力

パラメータ	説明	種類	必須
measure	順位付けするメジャー	メジャー	はい
ranking_dims	メジャーの順位付けに使用するディメンション	ディメンションの一覧	いいえ
Top Bottom	次のように順位を設定します。 <ul style="list-style-type: none">Top: 降順Bottom: 昇順	キーワード	いいえ (デフォルトは Top です)
reset_dims	順位をリセットするディメンション	ディメンションの一覧	いいえ

注

- "順位" ディメンションを指定しない場合、この関数はデフォルトの計算コンテキストを使用して順位を計算します。

- 順位またはリセットディメンションの一覧にあるディメンションの数が1つだけでも、ディメンションは常にかっこの中に配置する必要があります。
- 複数の順位またはリセットディメンションを指定する場合、それらをセミコロンで区切る必要があります。
- デフォルトでは、セクションまたはブロックブレイクを越えると順位はリセットされます。

例

次のテーブルでは、順位は `Rank([売上];([国]))` で指定されます。

国	売上げ	順位
フランス	835,420	2
アメリカ	2,451,104	1

次のテーブルでは、順位は `Rank([売上げ];([国];Bottom))` で指定されます。Bottom 引数は、メジャーが降順で並べられることを意味します。

国	売上げ	順位
フランス	835,420	1
アメリカ	2,451,104	2

次のテーブルでは、順位は `Rank([売上];([国];[リゾート]))` で指定されます。

国	リゾート	売上げ	順位
フランス	フランスリビエラ	835,420	3
アメリカ	バハマビーチ	971,444	2
アメリカ	ハワイアンクラブ	1,479,660	1

次のテーブルでは、順位は `Rank([売上];([国];[年度]);([国]))` で指定されます。順位は、“国”ディメンションではリセットされます。

国	年	売上げ	順位
フランス	FY1998	295,940	1
フランス	FY1999	280,310	2
フランス	FY2000	259,170	3
アメリカ	FY1998	767,614	3
アメリカ	FY1999	826,930	2
アメリカ	FY2000	856,560	1

関連情報

[Bottom/Top 演算子 \[228 ページ\]](#)

6.1.8.17 Round

説明

数値を四捨五入します。

関数グループ

数値

構文

```
num Round (number;round_level)
```

入力

パラメータ	説明	種類	必須
number	切り捨てる数値	数値	はい
round_level	四捨五入する小数点以下の桁数	数値	はい

例

Round(9.44;1) は 9.4 を返します。

Round(9.45;1) は 9.5 を返します。

Round(9.45;0) は 9 を返します。

Round(9.45;-1) は 10 を返します。

Round(4.45;-1) は 0 を返します。

関連情報

[数値の四捨五入と切り捨て \[245 ページ\]](#)

6.1.8.18 Sign

説明

数値の符号を返します。

関数グループ

数値

構文

```
int Sign(number)
```

入力

パラメータ	説明	種類	必須
number	任意の数値	数値	はい

注

Sign は、number が負数の場合に -1 を返し、number がゼロの場合に 0 を返し、number が正数の場合に 1 を返します。

例

Sign(3) は 1 を返します。

Sign(-27.5) は -1 を返します。

6.1.8.19 Sin

説明

角度のサインを返します。

関数グループ

数値

構文

```
num Sin (angle)
```

入力

パラメータ	説明	型	必須
angle	ラジアンで指定した角度	数値	○

例

`Sin (234542)` は、小数点の設定に応じて、-0.116992 または -0.12 を返します。

6.1.8.20 Sqrt

説明

数値の平方根を返します。

関数グループ

数値

構文

```
num Sqrt (number)
```

入力

パラメータ	説明	種類	必須
number	任意の数値	数値	はい

例

Sqrt (25) は 5 を返します。

6.1.8.21 Tan

説明

指定した角度のタンジェントを返します。

関数グループ

数値

構文

```
num Tan (angle)
```

入力

パラメータ	説明	種類	必須
angle	ラジアンで指定した角度	数値	はい

例

`Tan(90)` は -2 を返します。

6.1.8.22 ToNumber

説明

文字列を数値として返します。

関数グループ

数値

構文

```
num ToNumber(string)
```

または

入力

パラメータ	説明	型	必須
文字列	文字列としての数値	文字列	○

注

`string` が数値または日時ではない場合、`ToNumber` は `#ERROR` を返します。

例

`ToNumber("45")` は 45 を返します。

6.1.8.23 Truncate

説明

数値を切り捨てます。

関数グループ

数値

構文

```
num Truncate(number;truncate_level)
```

入力

パラメータ	説明	種類	必須
number	切り捨てる数値	数値	はい
truncate_level	切り捨てる小数点以下の桁数	数値	はい

注

例

`Truncate(3.423;2)` は 3.42 を返します。

関連情報

[数値の四捨五入と切り捨て \[245 ページ\]](#)

6.1.9 セット関数

6.1.9.1 Ancestor

説明

メンバーの祖先メンバーを返します。

関数グループ

集合

構文

```
member Ancestor(member;level|distance)
```

入力

パラメータ	説明	型	必須
member	任意のメンバー	メンバー	○
level	祖先のレベル	レベル	level または distance のいずれかが必要

パラメータ	説明	型	必須
distance	現在のレベルから祖先レベルまでの距離	整数	level または distance のいずれかが必要

注

- `Ancestor` は、スタンドアロン関数としては使用されません。集計のメンバーセットを指定する集計関数の入力パラメータで使用されます。
- `member` は階層の現在のメンバーです。その階層がブロックのコンテキストにない場合は、式は空の値を返します。
- `distance` は正である必要があります。

例

次の例は、すべて英語のデータソースからのものです。

以下の地理階層に基づいて、顧客の都市と無関係な各顧客のインターネット売上金額の影響を調べます。



最初に、都市ごとに、その国のインターネット売上金額を計算します。

```
=Sum([Query 2].[Internet Sales].[Internet Sales Amount];{Ancestor([Customer Geography];[Customer Geography].[City])})
```

Customer Geography	'=Sum([Query 2].[Internet Sales].[Internet Sales Amount
[-] All Customers		29,358,677.22
[-] Australia		9,061,000.58
[-] New South Wales		3,934,485.73
[-] Coffs Harbour	235,454.97	235,454.97
[-] 2450	235,454.97	235,454.97
Adriana Smith	235,454.97	5,333.25
Aimee Guo	235,454.97	77.27
Allison R. Young	235,454.97	39.98
Ann A. Sara	235,454.97	39.98
Antonio G. Pattersor	235,454.97	8,068.03
Ariana Stewart	235,454.97	6,070.59
Arthur Kapoor	235,454.97	23.97
Barbara W. Lal	235,454.97	2,795.01
Bobby D. Saunders	235,454.97	120.48
Brianna J. Johnson	235,454.97	38.98

次に、その国のグローバルインターネット売上金額に占める各都市の割合を計算します。

```
=([Query 2].[Internet Sales].[Internet Sales Amount] / Sum([Query 2].[Internet Sales].[Internet Sales Amount];{Ancestor([Customer Geography];[Customer Geography].[City])))
```

Customer Geography	'=Sum([Query 2].[Internet Sales].[Internet Sales Amount
[-] All Customers		29,358,677.22
[-] Australia		9,061,000.58
[-] New South Wales		3,934,485.73
[-] Coffs Harbour	100.00%	235,454.97
[-] 2450	100.00%	235,454.97
Adriana Smith	2.27%	5,333.25
Aimee Guo	0.03%	77.27
Allison R. Young	0.02%	39.98
Ann A. Sara	0.02%	39.98
Antonio G. Pattersor	3.43%	8,068.03
Ariana Stewart	2.58%	6,070.59
Arthur Kapoor	0.01%	23.97
Barbara W. Lal	1.19%	2,795.01
Bobby D. Saunders	0.05%	120.48
Brianna J. Johnson	0.02%	38.98
Bruce G. Madan	0.03%	65.96

i 注記

SAPBW プロバイダへの BICS 接続を使用している場合には、レベルを指定する代わりに、オフセットレベルを指定する必要があります。

```
=[Query 2].[Internet Sales].[Internet Sales Amount] / Sum([Query 2].[Internet Sales].[Internet Sales Amount];{Ancestor([Customer Geography];2)})
```

この場合には、州/都道府県と国の結果も得られます。

関連情報

[Aggregate \[40 ページ\]](#)

[Average \[47 ページ\]](#)

[Count \[48 ページ\]](#)

[Max \[54 ページ\]](#)

[Min \[56 ページ\]](#)

[Sum \[76 ページ\]](#)

6.1.9.2 Children

説明

集計関数内の階層メンバーの子メンバーを返します。

関数グループ

集合

構文

```
member_set member.Children
```

入力

パラメータ	説明	型	必須
member	任意のメンバー	member	○

注

- `Children` は、スタンドアロン関数としては使用されません。集計のメンバーセットを指定する集計関数の入力パラメータで使用されます。
- `member` は階層の現在のメンバーです。その階層がブロックのコンテキストにない場合は、式は空の値を返します。

例

`[地理].[米国].[カリフォルニア州].Children` は `[ロサンゼルス]`、`[サンフランシスコ]`、`[サンディエゴ]` を返します。

`"地理".Children` は、`"カリフォルニア州"` が `"地理"` 階層の現在のメンバーの場合、`"ロサンゼルス"`、`"サンフランシスコ"`、`"サンディエゴ"` を返します。

関連情報

[Aggregate \[40 ページ\]](#)

[Average \[47 ページ\]](#)

[Count \[48 ページ\]](#)

[Max \[54 ページ\]](#)

[Min \[56 ページ\]](#)

[Sum \[76 ページ\]](#)

6.1.9.3 Depth

説明

階層内のメンバーの奥行きを返します。

関数グループ

集合

構文

```
int member.Depth
```

入力

パラメータ	説明	型	必須
member	任意のメンバー	メンバー	○

注

- 奥行きとは階層の最上位レベルからのメンバーの距離です。
- 階層の最上位レベルはレベル 0 です。
- `member` は階層の現在のメンバーです。その階層がブロックのコンテキストにない場合は、式は空の値を返します。

例

階層メンバーの奥行きを調べます。

```
=[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

各月のすべての日がリストされているかどうかをチェックするために、子関数と組み合わせます。

```
=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

6.1.9.4 Descendants

説明

集計関数内の階層メンバーの子孫を返します。

関数グループ

集合

構文

```
member_set Descendants (member[;level|distance][;desc_flag])
```

入力

パラメータ	説明	型	必須
member	任意のメンバー	メンバー	○

パラメータ	説明	型	必須
level	子孫のレベル	レベル	いいえ (member のレベルはデフォルト)
distance	現在のレベルからの子孫レベルまでの距離	整数	いいえ (member のレベルはデフォルト)
desc_flag	どの子孫レベルメンバーを返すかを決定します。	キーワード	いいえ (デフォルトは Self)

注

- Descendants は、スタンドアロン関数としては使用されません。集計のメンバーセットを指定する集計関数の入力パラメータで使用されます。
- member は階層の現在のメンバーです。その階層がブロックのコンテキストにない場合は、式は空の値を返します。
- desc_flag の Self は、level|distance パラメータによって指定されたレベルを参照します。
- desc_flag の Before は、level|distance パラメータによって指定されたレベルの上のすべてのレベルを参照します。
- desc_flag の After は、level|distance パラメータによって指定されたレベルの下すべてのレベルを参照します。
- desc_flag の値は、以下のとおりです。

Self	level distance パラメータによって指定されたレベルでの子孫を返します。現在のメンバーがこのレベルにある場合は、現在のメンバーも含めて返します。
Before	現在のメンバーと、level distance パラメータによって指定されたレベルの上のすべての子孫を返します。
After	level distance パラメータによって指定されたレベルの下の子孫を返します。
Self_Before	現在のメンバーとその上のすべての子孫と、level distance パラメータによって指定されたレベルを含めて返します。
Self_After	現在のメンバーと、level distance パラメータによって指定されたレベル以下のすべての子孫を返します。
Before_After	現在のメンバーと、level distance パラメータによって指定されたレベルを除くすべての子孫を返します。
Self_Before_After	現在のメンバーとすべての子孫レベルを返します。
LEAVES	現在のメンバーと、子メンバーを含まない level distance パラメータによって指定されたレベル間のすべてのメンバーを返します。

- distance は正である必要があります。

例

財務階層があります。一部のノードは必ずしも累積ノードではありませんが、それらのノードの子孫を合計します。この例では、各貸借対照表メンバーの1つだけ下のレベルの子孫を以下のように合計します。

```
=Sum([Query 3 (1)].[Financial Reporting].[Amount];  
{Descendants([Accounts]&[Balance Sheet];1)})
```

Accounts			
[-] Balance Sheet	0		27,481,462
[-] Assets	13,740,731		
[-] Liabilities and Owners Equity	13,740,731		
[-] 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		

流動資産のすべてのメンバーを合計します。

```
=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	

流動資産自体を追加します。

```
=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	

関連情報

[Aggregate \[40 ページ\]](#)

[Average \[47 ページ\]](#)

[Count \[48 ページ\]](#)

[Max \[54 ページ\]](#)

[Min \[56 ページ\]](#)

[Sum \[76 ページ\]](#)

6.1.9.5 IsLeaf

説明

メンバーがリーフメンバーかどうかを決定します。

関数グループ

その他

構文

```
bool member.IsLeaf
```

入力

パラメータ	説明	型	必須
member	任意のメンバー	メンバー	○

注

- リーフメンバーとは子メンバーを持たないメンバーです。
- `member` は階層の現在のメンバーです。その階層がブロックのコンテキストにない場合は、式は空の値を返します。

例

ラインが日かどうかを調べます。

```
=[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

6.1.9.6 Key

説明

メンバーのキーを返します。

構文

```
string member.Key
```


関数グループ

集合

入力

パラメータ	説明	型	必須
member	任意のメンバー	メンバー	○

注

- キーとはメンバーの内部識別子です。
- `member` は階層の現在のメンバーです。その階層がブロックのコンテキストにない場合は、式は空の値を返します。

例

"米国" メンバーのキーが "XYZ" の場合、`[地理].[米国].Key` は "XYZ" を返します。

6.1.9.7 Lag

説明

集計関数内で、現在のメンバーから後ろに指定した距離だけ離れた同じレベルのメンバーを返します。

構文

```
member member.Lag(distance)
```

関数グループ

集合

入力

パラメータ	説明	型	必須
member	任意のメンバー	メンバー	○
distance	現在のメンバーからのメンバーの距離	整数	○

注

- Lag は、スタンドアロン関数としては使用されません。集計のメンバーセットを指定する集計関数の入力パラメータで使用されます。
- distance が正の場合、Lag は member の distance 個後ろのメンバーを返します。distance が負の場合、Lag は member の distance 個前のメンバーを返します。
- member は階層の現在のメンバーです。その階層がブロックのコンテキストにない場合は、式は空の値を返します。
- Lag は階層およびクエリ内のメンバー順序を使用して関連メンバーを返します。

例

毎週のインターネットセールスの差異を取得します。

```
=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

あるいは、特定の年とその 2 年前の年を比較します。

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

LagとIsLeafを組み合わせて、1週間の売り上げ金額の差異を調べます。最後の列には、以下のとおりに式が設定されます。

```
=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

関連情報

[Aggregate \[40 ページ\]](#)

[Average \[47 ページ\]](#)

[Count \[48 ページ\]](#)

[Max \[54 ページ\]](#)

[Min \[56 ページ\]](#)

[Sum \[76 ページ\]](#)

6.1.9.8 Parent

説明

集計関数内の階層メンバーの親メンバーを返します。

関数グループ

集合

構文

```
member member.Parent
```

入力

パラメータ	説明	型	必須
member	任意のメンバー	メンバー	○

注

- `Parent` は、スタンドアロン関数としては使用されません。集計のメンバーセットを指定する集計関数の入力パラメータで使用されます。
- `member` は階層の現在のメンバーです。その階層がブロックのコンテキストにない場合は、式は空の値を返します。

例

第 2 列には、各階層メンバーの親の取得を可能にする式が含まれます。

```
=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

関連情報

[Aggregate \[40 ページ\]](#)

[Average \[47 ページ\]](#)

[Count \[48 ページ\]](#)

[Max \[54 ページ\]](#)

[Min \[56 ページ\]](#)

[Sum \[76 ページ\]](#)

6.1.9.9 Siblings

説明

集計関数内の階層メンバーのメンバーおよび兄弟メンバーを返します。

関数グループ

集合

構文

```
member_set member.Siblings
```

入力

パラメータ	説明	型	必須
member	任意のメンバー	メンバー	○

注

- `Siblings` は、スタンドアロン関数としては使用されません。集計のメンバーセットを指定する集計関数の入力パラメータで使用されます。
- `member` は階層の現在のメンバーです。その階層がブロックのコンテキストにない場合は、式は空の値を返します。

- 兄弟メンバーとは、同じレベルで member と親が同じであるメンバーです。

例

時間階層で、1 年間の各四半期のパーセント、または期間内の各年のパーセントを調べます。

```
=[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%

自由形式セルで、期間全体に占める 2004 年の割合を調べます。

```
=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 perdiol
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%

関連情報

[Aggregate \[40 ページ\]](#)

[Average \[47 ページ\]](#)

[Count \[48 ページ\]](#)

[Max \[54 ページ\]](#)

[Min \[56 ページ\]](#)

[Sum \[76 ページ\]](#)

6.1.10 その他の関数

6.1.10.1 BlockName

説明

ブロック名を返します。

関数グループ

その他

構文

```
string BlockName()
```

例

“Block1”という名前のブロックにある場合、BlockName() は“Block1”を返します。

6.1.10.2 ColumnNumber

説明

列番号を返します。

関数グループ

その他

構文

```
int ColumnNumber()
```

例

式がテーブルの 2 番目の列にある場合、ColumnNumber() は 2 を返します。

6.1.10.3 CurrentUser

説明

現在のユーザの BI ラウンチパッドログインを返します。

関数グループ

その他

構文

```
string CurrentUser()
```

例

現在のユーザのログインが "gkn" の場合、CurrentUser() は "gkn" を返します。

6.1.10.4 ForceMerge

説明

同期されたディメンションがメジャーの計算コンテキストにない場合に、メジャーの計算にそれらのディメンションを組み込みます。

関数グループ

その他

構文

```
num ForceMerge (measure)
```

入力

パラメータ	説明	型	必須
measure	任意のメジャー	メジャー	○

出力

同期化されたディメンションを考慮して計算された式の結果

注

- スマートメジャーに適用された場合、ForceMerge は #MULTIVALUE を返します。これは、スマートメジャーの計算に必要なグループ化集合が存在しないからです。
- ForceMerge は、BusinessObjects/Desktop Intelligence の Multicube 関数に相当するものです。

例

ForceMerge ([売上げ]) は、[売上げ] メジャーと同じブロックに表示されない任意の同期されたディメンションが考慮された [売上げ] の値を返します。

6.1.10.5 GetContentLocale

説明

ドキュメントに含まれるデータのロケール (ドキュメントのロケール) を返します。

関数グループ

その他

構文

```
string GetContentLocale()
```

注

ドキュメントのロケールは、ドキュメント内のデータの書式設定に使用されます。

例

ドキュメントのロケールが“フランス語 (フランス)”の場合、`GetContentLocale()` は“fr_FR”を返します。

6.1.10.6 GetDominantPreferredViewingLocale

説明

ユーザの優先表示ロケールグループにおける最優先表示ロケールを返します。

関数グループ

その他

構文

```
string GetDominantPreferredViewingLocale()
```

注

- 関連ロケールの各グループには、グループのその他すべてのロケールの基準として使用される最優先ロケールがあります。たとえば、英語ロケールグループでは、アメリカ英語 ("en_US") が最優先ロケールとなります。ニュージーランド英語 ("en_NZ") も、このグループのメンバーです。
- トランスレーションマネージャガイドでは、すべての最優先表示ロケールが一覧にされています。

例

優先表示ロケールが“英語 (ニュージーランド)”である場合、`GetDominantPreferredViewingLocale` は“en_US”を返します。

関連情報

[GetPreferredViewingLocale \[207 ページ\]](#)

6.1.10.7 GetLocale

説明

ユーザインタフェースの書式設定に使用されるユーザのロケール (製品ロケール) を返します。

関数グループ

その他

構文

```
string GetLocale()
```

注

製品ロケールは、ユーザインタフェース（メニュー項目やボタンテキストなど）のロケールです。

例

ユーザの製品ロケールが“英語（アメリカ）”である場合、`GetLocale()` は“en_US”を返します。

6.1.10.8 GetLocalized

説明

ユーザの優先表示ロケールに従ってローカライズされた文字列を返します。

構文

```
string GetLocalized(string[;comment])
```

入力

パラメータ	説明	型	必須
文字列	翻訳する文字列	文字列	○
comment	翻訳者をサポートするコメント	string	×

注

- `string` パラメータは、任意の式の文字列にすることができます (たとえばセルではアラートメッセージや変数定義など)。
- レポートの設計時には、`comment` パラメータを使用して、翻訳者が文字列を翻訳するのをサポートする追加情報を提供することができます。コメントは、翻訳者がレポートの翻訳に使用するトランスレーションマネージャツールに、文字列とともに表示されます。
- `string` と `comment` の各ペアにより、トランスレーションマネージャツールで翻訳する個々の文字列が生成されます。その結果、`GetLocalized("Product Total"; "Max 20 characters")` と `GetLocalized("Product Total"; "Use no more than 20 characters")` が異なる翻訳を返す可能性があります。

例

優先表示ロケールが“fr_FR”である場合、`GetLocalized("Total for all products")` は“Total for all products”のフランス語訳を返します。

優先表示ロケールが“de_DE”である場合、`GetLocalized("Total for all products"; "Try not to use more than 20 characters")` は“Total for all products”のドイツ語訳を返します。またこの関数は、レポートの翻訳者に対し、文字列の翻訳時に、可能であれば 20 を超える文字数を使用しないよう指示します。

関連情報

[GetPreferredViewingLocale \[207 ページ\]](#)

6.1.10.9 GetPreferredViewingLocale

説明

ドキュメントデータを表示する際に使用される、ユーザの優先ロケール (優先表示ロケール) を返します。

関数グループ

その他

構文

```
string GetPreferredViewingLocale()
```

例

優先表示ロケールが“英語 (アメリカ)”である場合、GetPreferredViewingLocale は“en_US”を返します。

関連情報

[GetLocalized \[206 ページ\]](#)

[GetDominantPreferredViewingLocale \[204 ページ\]](#)

6.1.10.10 If...Then...Else

説明

表現式が TRUE か FALSE かに応じて指定された値を返します。

関数グループ

その他

構文

```
If bool_value Then true_value [Else false_value]
```


入力

パラメータ	説明	型	必須
bool_value	論理値	ブール型	はい
true_value	bool_value が true である場合に返される値	任意	はい
false_value	bool_value が false である場合に返される値	任意	Else が含まれる場合は、はい

注

- true_value と false_value ではデータ型を混合できます。
- If とともに論理演算子の And、Between、InList、Or、および Not を使用できます。
- Else 句を ElseIf 句と置き換えることによって If 条件をネストできます。次に、ネストレベルが1の構文を示します。

```
If bool_value Then true_value [ElseIf bool_value Then true_value Else false_value...]
```

- If 関数の元の構文、If (bool_value; true_value; false_value) もサポートされます。

例

If [売上げ] > 1000000 Then "高売上げ" は、売上げが1,000,000を超えるすべての行で "高売上げ" を返し、その他すべての行では何も返しません。

If [売上げ] > 1000000 Then "高売上げ" Else [売上げ] は、売上げが1,000,000を超えるすべての行で "高売上げ" を返し、その他すべての行ではその売上げを返します。

If [売上げ] > 1000000 Then "高売上げ" Else "低売上げ" は、売上げが1,000,000を超えるすべての行で "高売上げ" を返し、売上げが1,000,000未満のすべての行で "低売上げ" を返します。

If [売上げ] > 1000000 Then "高売上げ" ElseIf [売上げ] > 800000 Then "中売上げ" Else "低売上げ" は、売上げが1,000,000を超えるすべての行で "高売上げ" を返し、売上げが800,000と1,000,000との間のすべての行で "中売上げ" を返し、その他すべての行で "低売上げ" を返します。

関連情報

[If \[210 ページ\]](#)

[And 演算子 \[224 ページ\]](#)

[Between 演算子 \[226 ページ\]](#)

[InList 演算子 \[227 ページ\]](#)

[Or 演算子 \[225 ページ\]](#)

[Not 演算子 \[225 ページ\]](#)

6.1.10.11 If

説明

表現式が TRUE か FALSE かに応じて指定された値を返します。

関数グループ

その他

構文

```
If(bool_value;true_value>false_value)
```

入力

パラメータ	説明	型	必須
bool_value	論理値	論理値	○
true_value	bool_value が true である場合に返される値	任意	○
false_value	bool_value が false である場合に返される値	任意	○

注

- true_value と false_value ではデータ型を混合できます。
- false_value を追加の If 条件と置き換えることによって、If 条件をネストできます。次に、ネストレベルが 1 の構文を示します。

```
If(bool_value;true_value;If(bool_value;true_value>false_value);false_value)
```

- If...Then...Else 構文もサポートされます。

例

If([売上げ]>1000000;"高レベル";"低レベル") は、売上げが 1,000,000 を超えるすべての行で“高レベル”を返し、売上げが 1,000,000 以下のすべての行で“低レベル”を返します。

If([売上げ] >1000000;"高売上げ";[売上げ]) は、売上げが 1,000,000 を超えるすべての行で“高売上げ”を返し、その他すべての行ではその売上げ値を返します。

関連情報

[If...Then...Else \[208 ページ\]](#)

6.1.10.12 LineNumber

説明

テーブルの行番号を返します。

関数グループ

その他

構文

```
int LineNumber()
```

注

テーブル内の行番号はヘッダを 1 行目として開始します。

例

`LineNumber()` がテーブルの 2 行目にある場合、関数は 2 を返します。

6.1.10.13 NameOf

説明

オブジェクトの名前を返します。

関数グループ

その他

構文

```
string NameOf(obj)
```

入力

パラメータ	説明	種類	必須
obj	任意のレポートオブジェクト	レポートオブジェクト	はい

注

`NameOf` 関数は、レポートの列ヘッダと行ヘッダに表示されます。

例

`NameOf([予約日])` は、“予約日”を返します。

6.1.10.14 NoFilter

説明

値の計算時にフィルタを無視します。NoFilter はメジャーオブジェクトとともに使用されます。ディメンションには適用されません。

関数グループ

その他

構文

```
input_type NoFilter(obj; [All|Drill])
```

入力

パラメータ	説明	型	必須
obj	任意のレポートオブジェクト	レポートオブジェクト	○
All Drill	<ul style="list-style-type: none">キーワードの指定なし: レポートおよびブロックフィルタを無視All: すべてのフィルタを無視Drill: レポートフィルタおよびドリルフィルタを無視	キーワード	いいえ

注

- NoFilter (obj;Drill) は、クエリドリルモードで正しく機能しません。これは、ドリルフィルタがレポートデータに適用されるのではなく、クエリに追加されるからです。
- ドリルフィルタを適用した状態でドリルモードを終了した場合、ドリルフィルタがレポートフィルタとなり、NoFilter (obj;Drill) が適用されるすべてのオブジェクトの値を変更する場合があります。

例

ブロックフッタで使用された場合、`NoFilter(Sum([売上げ]))` は、行がフィルタ処理して除外されている場合でも、ブロック内のすべての行の総売り上げを返します。

`NoFilter(Sum([売上げ]);All)` は、レポートからフランスを排除するフィルタが存在しても、フランスを含むすべての国の売上げの合計を返します。

`NoFilter(Sum([売上げ]);Drill)` は、“国”ディメンションに対するドリルフィルタが存在しても、すべての国の売上げの合計を返します。

6.1.10.15 NumberOfPages

説明

レポートのページ数を返します。

関数グループ

その他

構文

```
integer NumberOfPages()
```

例

2 ページのレポートの場合、`NumberOfDataPages()` は 2 を返します。

6.1.10.16 Page

説明

レポートの現在のページ番号を返します。

関数グループ

その他

構文

```
integer Page()
```

例

レポートの 2 ページ目にある場合、Page () は 2 を返します。

6.1.10.17 Previous

説明

オブジェクトの直前の値を返します。

関数グループ

その他

構文

```
input_type Previous (dimension|measure|Self [;Row|col] [; (reset_dims)] [;offset]  
[;NotNull])
```

入力

パラメータ	説明	種類	必須
dimension measure Self	関数が返す前の値をもつディメンションまたはメジャー、あるいは Self キーワード	ディメンション、メジャー、またはキーワード	はい
Row/Col	計算の方向を設定します。	キーワード	いいえ
reset_dims	計算をリセットするために使用されるディメンションの一覧	ディメンションの一覧	いいえ
offset	現在行の前の offset 行で ある dimension または measure の値を指定します。	整数	いいえ (デフォルトは 1)
NotNull	オフセットから開始して最初の NULL でない値を返すように関数に指示します。	キーワード	いいえ

注

- offset のデフォルト値は 1 です。Previous([売上げ];1) と Previous([売上げ]) は機能的には同じです。
- NotNull 引数を使用すると、この関数は、現在の行から offset 行分だけ前にあるセルから開始して後方へカウントされ、最初の Null でない値を返します。
- Previous 関数では、拡張構文のコンテキスト演算子を使用できます。
- Self 演算子は、セルが 1 つのレポートオブジェクト以外のコンテンツを含む場合に前のセルの値を参照できます。
- 指定するリセットディメンションの数が 1 つだけでも、ディメンションは常にかつこの中に配置する必要があります。
- 複数のリセットディメンションを指定する場合、それらをセミコロンで区切る必要があります。
- Previous は、レポート、セクション、ブロックのすべてのフィルタ、およびすべての並べ替えが適用された後に適用されます。
- Previous を使用する式に、並べ替えまたはフィルタを適用することはできません。
- Previous がメジャーに適用され、そのメジャーが未定義の値を返す場合、Previous は、前の行が値を返した場合でも、未定義の値を返します。
- Previous は、ブレイクヘッダまたはフッタ以外に配置されたブレイクは無視します。
- Previous は、ブレイクフッタに配置されると、フッタの前のインスタンスの値を返します。
- Previous は、各レポートセクションでリセットされます。
- クロスタブで使用される場合、Previous は行の最後の値を次の行の最初の値の前の値として処理しません。

例

次のテーブル内で使用した場合、Previous([国];1) は、次の値を返します。

国	売上げ	直前の値
アメリカ	5,000,000	
イギリス	2,000,000	アメリカ
フランス	2,100,000	イギリス

次のテーブル内で使用した場合、Previous ([売上]) は、次の値を返します。

国	売上げ	直前の値
アメリカ	5,000,000	
イギリス	2,000,000	5,000,000
フランス	2,100,000	2,000,000

次のテーブル内で使用した場合、Previous ([売上]; ([国])) は、次の値を返します。

国	地域	売上げ	直前の値
アメリカ	北部	5,000,000	
	南部	7,000,000	5,000,000
イギリス	北部	3,000,000	
	南部	4,000,000	3,000,000

次のクロスタブで使用した場合、Previous ([売上]) は、次の値を返します。

	2004	直前の値	2005	直前の値
アメリカ	5,000,000		6,000,000	5,000,000
イギリス	2,000,000		2,500,000	2,000,000
フランス	3,000,000		2,000,000	3,000,000

次のテーブルで使用した場合、Previous ([売上]) は、[国] でブレイクして次の値を返します。

国	地域	売上げ	直前の値
アメリカ	北部	5,000,000	
	南部	7,000,000	5,000,000
アメリカ		12,000,000	

国	地域	売上げ	直前の値
イギリス	北部	3,000,000	7,000,000
	南部	4,000,000	3,000,000
イギリス		7,000,000	12,000,000

次のテーブル内で使用した場合、Previous ([売上げ]; 2; NoNull) は、次の値を返します。

年	四半期	売上げ	直前の値
2008	第 1	500	
2008	第 2		
2008	第 3	400	500
2008	第 4	700	500
2008	第 1	300	400
2008	第 2		700
2008	第 3		300
2008	第 4	200	300

2*Previous (Self) は、2、4、6、8、10.. を返します。

関連情報

[Previous 関数を使用した値の比較 \[255 ページ\]](#)

[Self 演算子 \[235 ページ\]](#)

6.1.10.18 RefValue

説明

データ追跡が有効な場合、レポートオブジェクトの参照値を返します。

関数グループ

その他

構文

```
input_type RefValue(obj)
```

例

参照データで“上位の地域”変数の値が“南西部”の場合、`RefValue([上位の地域])` は“南西部”を返します。

参照データで“売上げ”メジャーの値が1000の場合、`RefValue([売上げ])` は1000を返します。

注

- `RefValue()` 関数は、メジャーオブジェクトまたはディメンションオブジェクトのいずれかとともに使用することができます。ただし、ディメンションまたは詳細としての資格が設定された変数で使用されると、`RefValue()` 関数はその参照値ではなくそのオブジェクトの現在の値を返します。参照値を取得するには、変数にメジャーとしての資格が設定されている必要があります。
- セクション、テーブル、フォーム、またはチャート内で直接作成された式には、常にメジャーとしての資格が設定されます。このため、式で `RefValue()` 関数が使用される場合、想定された参照値が返されます。

変数を使用した **RefValue** 関数の例

“カリフォルニア”、“フロリダ”、“テキサス”、“ニューヨーク”という“都道府県”ディメンションの値の一覧があるとします。データの最新表示後、この一覧は、“アリゾナ”、“カリフォルニア”、“フロリダ”、“テキサス”、および“ニューヨーク”になります。`Variable=RefValue([都道府県])` のような変数により、以下が返されます。

変数の資格	返される値の一覧
ディメンションまたは詳細	"アリゾナ"、"カリフォルニア"、"フロリダ"、"テキサス"、"ニューヨーク"
メジャー	(NULL 値)、"カリフォルニア"、"フロリダ"、"テキサス"、"ニューヨーク"

6.1.10.19 RelativeValue

説明

オブジェクトの前と後の値を返します。

関数グループ

その他

構文

```
input_type RelativeValue (measure|detail;slicing_dims;offset)
```

入力

パラメータ	説明	型	必須
measure detail	ブロック内のメジャーまたはディメンションの詳細	メジャーまたは詳細	○
slicing_dims	計算コンテキストを提供するディメンション	ディメンションの一覧	○
offset	現在行から削除された offset 行である measure または detail の値を指定します。	整数	○

注

- オブジェクトはブロック内で使用可能なメジャーまたはディメンションの詳細であることが必要です。
- スライスディメンションの値のリストの並べ替え順を使用して、関数の出力が決定されます。並べ替え順は 2 つの要因によって決定されます。並べ替えはスライスディメンションに適用され、スライスディメンションにある並べ替えは関数の中に表示されます。
- セクションマスタとして使用されるディメンションは、スライスディメンションとして指定できます。
- すべてのスライスディメンションはブロック内または関数が配置されているブロックのセクションヘッダ内に存在することが必要です。スライスディメンションが後でブロックから削除された場合、関数は #COMPUTATION エラーを返します。
- オフセットがスライスディメンション値のリストの行数を超過した場合、関数は NULL を返します。
- RelativeValue は再帰的には使用できません。
- 指定するスライスディメンションの数が 1 つだけでも、ディメンションは常にかっこの中に配置する必要があります。

例

次の表の RelativeValue 列には下記の式が含まれます。

```
RelativeValue ([売上げ]; ([年]); -1)
```

年	四半期	販売担当者	売上げ	RelativeValue
2007	Q1	スミス	1000	

年	四半期	販売担当者	売上げ	RelativeValue
2007	Q2	ジョーンズ	2000	
2007	Q3	ウィルソン	1500	
2007	Q4	ハリス	3000	
2008	Q1	スミス	4000	1000
2008	Q2	ジョーンズ	3400	2000
2008	Q3	ウィルソン	2000	1500
2008	Q4	ハリス	1700	3000

関連情報

[#COMPUTATION \[249 ページ\]](#)

[RelativeValue 関数を使用した値の比較 \[255 ページ\]](#)

6.1.10.20 ReportName

説明

レポートの名前を返します。

関数グループ

その他

構文

```
string ReportName()
```

例

使用されているレポートの名前が“売上げレポート”の場合、ReportName() は“売上げレポート”を返します。

6.1.10.21 RowIndex

説明

行番号を返します。

関数グループ

その他

構文

```
integer RowIndex()
```

注

- 行番号は 0 から開始します。
- RowIndex は、テーブルヘッダまたはフッタに配置されると #MULTIVALUE を返します。

例

RowIndex は、テーブルの 1 行目にある場合は 0 を返します。

6.1.10.22 UniqueNameOf

説明

オブジェクトの一意の名前を返します。

関数グループ

その他

構文

```
string UniqueNameOf(obj)
```

入力

パラメータ	説明	種類	必須
obj	任意のレポートオブジェクト	レポートオブジェクト	はい

例

`UniqueNameOf([予約日])` は“予約日”を返します。

6.2 関数と式の演算子

演算子は、式のコンポーネントを結合するものです。

式には、数理的演算子、条件演算子、論理演算子、関数固有の演算子、拡張構文演算子を含めることができます。

6.2.1 数理的演算子

数理的演算子は、一般的な算数の計算です。

加算 (+)、減算 (-)、乗算 (*)、除算 (/) 演算子があり、数値計算を実行します。式 `[売上げ] - [販売コスト]` には、数理的演算子の減算が含まれています。

i 注記

文字列と使用する場合は、“+”演算子は文字列を連結する演算子になります。この場合は、2つの文字列が結合されます。たとえば `“John” + “Smith”` という式は、“John Smith”を返します。

6.2.2 条件演算子

条件演算子は、値の比較方法を決定する演算子です。

演算子	説明
=	等しい
>	より大きい
<	より小さい
>=	以上
<=	以下
<>	等しくない

条件演算子は、次のように If 関数と共に使用します。

```
If [売上げ]>10000 Then "High" Else "Low"
```

この式は、売上げが 10000 以上のすべての行について“高い”を、その他の行については“低い”を返します。

6.2.3 論理演算子

論理演算子には、And、Or、Not、Between、Inlist の 5 種類があります。

論理演算子は、True または False の値を返す論理式で使用します。

6.2.3.1 And 演算子

And 演算子は、論理値をリンクします。

説明

And によってリンクされたすべての論理型が TRUE を返す場合、すべての値の組み合わせも TRUE を返します。

構文

```
bool_value And bool_value [And bool_value...]
```


例

If [リゾート] = "バハマビーチ" And [売上げ]>100000 Then "バハマの高売上げ" は、[リゾート] が“バハマビーチ”で [売上げ] が 100000 を超える場合に“バハマの高売上げ”を返します。

6.2.3.2 Or 演算子

Or 演算子は、論理値をリンクします。

説明

Or によってリンクされた 1 つの論理値が TRUE を返すと、すべての値の組み合わせも TRUE を返します。

構文

```
bool_value Or bool_value [Or bool_value...]
```

例

If [リゾート] = "バハマビーチ" Or [リゾート]="ハワイアンクラブ" Then "US" Else "France" は、[リゾート] が“バハマビーチ”または“ハワイアンクラブ”の場合に“US”、それ以外の場合に“France”を返します。

6.2.3.3 Not 演算子

説明

Not 演算子は、論理値の逆を返します。

構文

```
bool Not (bool_value)
```

例

If Not ([国] = "US") Then "US でない" は、[国] が“US”以外の値である場合に“US でない”を返します。

6.2.3.4 Between 演算子

説明

Between 演算子は、変数が2つの値の間にあるかどうかを判別します。

構文

```
bool Between(first_value;second_value)
```

注

- Between は、If 関数および Where 演算子と共に使用します。
- ドキュメントロケールを変更すると、Between 演算子によって返される結果に影響を与えることができます。

例

If [売上げ] Between (800000;900000) Then "平均的売上げ" は、[売上げ] が 800000 と 900000 の間である場合に"平均的売上げ"を返します。

[売上げ] Between (10000;20000) は、売上げが 10000 と 20000 の間である場合に TRUE を返します。

If ([売上げ] Between (200000;500000); "平均的売上げ"; "高売上げ/低売上げ") は、[売上げ] の値が 300000 の場合に"平均的売上げ"を返します。

関連情報

[If...Then...Else \[208 ページ\]](#)

[Where 演算子 \[235 ページ\]](#)

6.2.3.5 InList 演算子

説明

InList 演算子は、値が値の一覧にあるかどうかを判別します。

構文

```
bool test_value InList(value_list)
```

注

これは、InList 単独ではなく、test_value と InList の組み合わせで論理値を返します。

例

If Not ([国] InList("England";"Scotland";"Wales")) Then "Britain でない" Else
"Britain" は、[国] が"England"、"Scotland"、"Wales"に等しくない場合に"Not Britain"を返し、等しい場合に
"Britain"を返します。

If [リゾート] InList("バハマビーチ";"ハワイアンクラブ") Then "US リゾート" は、[リゾート] が"バハ
マビーチ"または"ハワイアンクラブ"に等しい場合に"US リゾート"を返します。

関連情報

[If...Then...Else \[208 ページ\]](#)

[Where 演算子 \[235 ページ\]](#)

6.2.4 関数固有の演算子

一部の関数には、特定の演算子を引数として指定できます。

たとえば、Previous 関数には Sel 演算子を指定できます。

すべての関数は) と (を使って引数を囲みます。複数のパラメータを指定できる関数では、; を使ってパラメータを区切ります。

6.2.4.1 All 演算子

All 演算子は、すべてのフィルタを無視するよう NoFilter 関数に指示します。

All 演算子は、Count 関数に重複を含むすべての値をカウントするように指示することもできます。

関連情報

[Count \[48 ページ\]](#)

[Distinct/All 演算子 \[230 ページ\]](#)

[NoFilter \[213 ページ\]](#)

[All/Drill 演算子 \[228 ページ\]](#)

6.2.4.2 All/Drill 演算子

All/Drill 演算子は、NoFilter 関数と共に使用します。

説明

All/Drill 演算子は、NoFilter 関数が無視するフィルタを判別します。

- 指定なし - NoFilter はレポートおよびブロックフィルタを無視
- All - NoFilter はすべてのフィルタを無視
- Drill - NoFilter はレポートフィルタおよびドリルフィルタを無視

6.2.4.3 Bottom/Top 演算子

Bottom/Top 演算子は、Rank 関数と共に使用します。

説明

Bottom/Top 演算子は、Rank 関数に降順または昇順に順位付けするように指示します。

- Top: 降順に順位付けします。
- Bottom: 昇順に順位付けします。

例

`Rank([売上げ];([国]);Top)` は、売上げの多い国から順に順位を付けます。

関連情報

[Rank \[172 ページ\]](#)

6.2.4.4 Break 演算子

`Break` 演算子は、`Percentage` 関数と共に使用します。

説明

`Break` 演算子は、`Percentage` 関数にテーブルのブレイクを考慮するように指示します。

例

式 `Percentage([売上げ])` は、下の表のような結果になります。パーセンテージはブロック内の総売上げに対して計算されています。

年	四半期	売上げ	パーセンテージ
2005	第 1	10000	10%
2005	第 2	20000	20%
2006	第 1	30000	30%
2006	第 2	40000	40%

式 `Percentage([売上げ];Break)` は、下の表のような結果になります。パーセンテージはブロック内の各部分の総売上げに対して計算されています。

年	四半期	売上げ	パーセンテージ
2005	第 1	10000	33.3%

2005	第 2	20000	66.6%
2006	第 1	30000	42.9%
2006	第 2	40000	57.1%

関連情報

[Percentage \[58 ページ\]](#)

6.2.4.5 Distinct/All 演算子

Distinct/All 演算子は、Count 関数と共に使用します。

Distinct/All 演算子は、Count 関数に個々の値のみ、またはすべての値をカウントするように指示します。

例

Count ([売上げ]; Distinct) は、[売上げ] の値が (5;5;6;4) の場合に 3 を返します。

Count ([売上げ]; All) は、[売上げ] の値が (5;5;6;4) の場合に 4 を返します。

関連情報

[Count \[48 ページ\]](#)

6.2.4.6 IncludeEmpty 演算子

IncludeEmpty 演算子は、集計関数と共に使用します。

説明

IncludeEmpty 演算子は、空の値を計算に組み込むように、一部の集計関数 (Average、Count、RunningAverage、RunningCount) に指示します。

例

`Average([売上げ]; IncludeEmpty)` は、`[売上げ]` の値が (5;3;<empty>;4) の場合に 3 を返します。

関連情報

[Average \[47 ページ\]](#)

[Count \[48 ページ\]](#)

[RunningAverage \[62 ページ\]](#)

[RunningCount \[64 ページ\]](#)

6.2.4.7 Index 演算子

`Index` 演算子は、`UserResponse` 関数および `RefValueUserResponse` 関数と共に使用します。

説明

`Index` 演算子は、`UserResponse` 関数および `RefValueUserResponse` 関数にプロンプト応答のデータベースのプライマリキーを返すように指示します。

関連情報

[UserResponse \[134 ページ\]](#)

[RefValueUserReponse \[131 ページ\]](#)

6.2.4.8 Linear 演算子

`Linear` 演算子は、`Interpolation` 関数と共に使用します。

説明

`Linear` 演算子は、最小二乗補間による線形回帰を使用して欠落したメジャー値を設定するよう `Interpolation` 関数に指示します。

最小二乗補間による線形回帰は、使用可能なすべてのメジャー値をできる限り厳密に渡す、 $f(x) = ax + b$ という形の一次方程式を計算することによって、欠落した値を計算します。

関連情報

[Interpolation \[51 ページ\]](#)

6.2.4.9 NoNull 演算子

NoNull 演算子は、Previous 関数と共に使用します。

説明

NoNull 演算子は、NULL 値を無視するよう Previous 関数に指示します。

Previous を NoNull と併用した場合、現在の行よりオフセット行分だけ前にあるセルから開始して、さかのぼってカウントし、最初の NULL でないオブジェクトの値を返します。

関連情報

[Previous \[215 ページ\]](#)

6.2.4.10 NotOnBreak 演算子

NotOnBreak 演算子は、Interpolation 関数と共に使用します。

説明

NotOnBreak 演算子は、セクションおよびブロックブレイクを無視するよう Interpolation 関数に指示します。

関連情報

[Interpolation \[51 ページ\]](#)

6.2.4.11 PointToPoint 演算子

PointToPoint 演算子は、Interpolation 関数にポイントツーポイント補間を使用して欠落したメジャー値を表示するよう指示します。

説明

ポイントツーポイント補間は、欠落した隣接する 2 つの値を渡す、 $f(x) = ax + b$ という形式の一次方程式を計算することにより、欠落した値を計算します。

関連情報

[Interpolation \[51 ページ\]](#)

6.2.4.12 Row/Col 演算子

Row 演算子は、埋め込みコンテキストのすべての行の合計値のパーセンテージとして行の各値を計算します。Col 演算子は、埋め込みコンテキストのすべての列の合計値のパーセンテージとして列の各値を計算します。

説明

Row/Col 演算子は Percentage、Previous、RunningAverage、RunningCount、RunningMax、RunningMin、RunningProduct、RunningSum の各関数の計算方向を設定します。

注

クロスタブでは、各セルの値はデフォルトでクロスタブ内の合計値のパーセンテージとして計算されます。Row 演算子は、行の合計値のパーセンテージとして行の値を計算します。Col 演算子は、列の合計値パーセンテージとして列の値を計算します。

例

クロスタブでは、Percentage ([メジャー]) は次のような結果になります。

メジャー	パーセンテージ	メジャー	パーセンテージ
100	10%	500	50%
200	20%	200	20%

Percentage ([メジャー]; Row) は次のような結果になります。

メジャー	パーセンテージ	メジャー	パーセンテージ
100	16.7%	500	83.3%
200	50%	200	50%

Percentage ([メジャー]; Col) は次のような結果になります。

メジャー	パーセンテージ	メジャー	パーセンテージ
100	33.3%	500	83.3%
200	66.6%	200	16.7%

Row 演算子は、行ごとに積算集計を計算します。Col 演算子は、列ごとに積算集計を計算します。

クロスタブでは、RunningSum ([メジャー]) または RunningSum ([メジャー]; Row) は次のような結果になります。

メジャー	RunningSum	メジャー	RunningSum
100	100	200	300
400	700	250	950

クロスタブでは、RunningSum ([メジャー]; Col) は次のようになります。

メジャー	RunningSum	メジャー	RunningSum
100	100	200	700
400	500	250	950

関連情報

[Percentage \[58 ページ\]](#)

[RunningAverage \[62 ページ\]](#)

[RunningCount \[64 ページ\]](#)

[RunningMax \[66 ページ\]](#)

[RunningMin \[68 ページ\]](#)

[RunningProduct \[70 ページ\]](#)

[RunningSum \[72 ページ\]](#)

6.2.4.13 Self 演算子

Self 演算子は、Previous 関数と共に使用します。

説明

レポートオブジェクトが含まれていない場合に、Previous 関数が前のセルを参照するようにします。

例

`5 + Previous(Self)` は、5、10、15、20、25、30...を返します。

`1 + 0.5 * Previous(Self)` は、1、1.5、1.75、1.88...を返します。

関連情報

[Previous \[215 ページ\]](#)

6.2.4.14 Where 演算子

説明

Where 演算子は、メジャーの計算に使用されるデータを制限します。

例

`Average ([売上げ]) Where ([国] = "US")` という式は、国が“US”である売上げの平均を計算します。

`Average ([売上げ]) Where ([国] = "US" Or [国] = "フランス")` という式は、国が“US”または“フランス”である売上げの平均を計算します。

`[売上げ] Where (Not ([国] Inlist ("US"; "フランス")))` という式は、US およびフランス以外の国の売上げを計算します。

変数 [高売上げ] には、`[売上げ] Where [売上げ] > 500000` という式が含まれています。ブロックで使用された場合、[高売上げ] は 500000 よりも大きい値の売上げを示すか、または何も表示しないかのいずれかです。[高売上げ] 列

の下の方で使われた場合、Average（「高売上げ」）という式は、500000 よりも大きいすべての売上げの平均を返します。

関連情報

[And 演算子 \[224 ページ\]](#)

[Between 演算子 \[226 ページ\]](#)

[InList 演算子 \[227 ページ\]](#)

[Or 演算子 \[225 ページ\]](#)

[Not 演算子 \[225 ページ\]](#)

6.2.5 拡張構文の演算子

コンテキスト演算子では、入力コンテキストと出力コンテキストを明示的に指定します。

次の表に、コンテキスト演算子を示します。

演算子	説明
In	コンテキストで使用するディメンションの明示的な一覧を指定します。
ForEach	デフォルトのコンテキストにディメンションを追加します。
ForAll	デフォルトのコンテキストからディメンションを除外します。

ForAll と ForEach 演算子は、デフォルトのコンテキストに多数のディメンションがある場合に便利です。コンテキストの追加と除外は多くの場合、In 演算子で明示的に指定するよりも、ForAll と ForEach を使用の方が簡単です。

6.2.5.1 In コンテキスト演算子

In コンテキスト演算子は、コンテキストのディメンションを明示的に指定します。

❖ 例

In によるコンテキストのディメンションの指定

次のレポートは、年と売上げを表示したレポートです。データプロバイダには[四半期]オブジェクトも含まれていますが、ブロックにこのディメンションは含まれていません。このレポートに、各年の四半期別の最高売上げを示す列を追加することになります。レポートは次のようになります。

年	売上げ	四半期別最高売上げ
2001	\$8,096,123.60	\$2,660,699.50

年	売上げ	四半期別最高売上げ
2002	\$13,232,246.00	\$4,186,120.00
2003	\$15,059,142.80	\$4,006,717.50

このブロックと"四半期"ディメンションを含むブロックを比べると、[四半期別最高売上げ]列の値がどのように導き出されたかわかります。

年	四半期	売上げ
2001	Q1	\$2,660,699.50
2001	Q2	\$2,279,003.00
2001	Q3	\$1,367,841.00
2001	Q4	\$1,788,580.00
	最大:	\$2,660,699.50

年	四半期	売上げ
	Q1	\$3,326,172.00
	Q2	\$2,840,651.00
	Q3	\$2,879,303.00
	Q4	\$4,186,120.00
	最大:	\$4,186,120.00

年	四半期	売上げ
	Q1	\$3,742,989.00
	Q2	\$4,006,717.50
	Q3	\$3,953,395.00
	Q4	\$3,356,041.00
	最大:	\$4,006,717.50

[四半期別最高売上げ]列は、各年の四半期別の最高売上げを示しています。たとえば、2002 年は第 4 四半期の売上げが最大であるため、2002 年の[四半期別最高売上げ]列には第 4 四半期の売上げが表示されています。

In 演算子を使用すると、四半期別最高売上げの式は次のようになります。

```
Max ([売上げ] In ([年];[四半期])) In ([年])
```

この式は、各 (年、四半期) の組み合わせについて最高売上げを計算し、年別にこの数字を出力します。

i 注記

ブロックのデフォルトの出力コンテキストは年なので、出力コンテキストを明示的に指定する必要はありません。

6.2.5.2 ForEach コンテキスト演算子

ForEach 演算子は、コンテキストにディメンションを追加します。

❖ 例

ForEach によるコンテキストへのディメンションの追加

次の表は、“四半期”ディメンションを含むが、ブロックにはそのディメンションが含まれていないレポートの各四半期に対する最大売上げを示しています。

年	売上げ	四半期別最高売上げ
2001	8096123.60	2660699.50
2002	13232246.00	4186120.00
2003	15059142.80	4006717.50

[四半期別最高売上げ]列で ForEach 演算子を含まない次の式を作成することができます。

```
Max ([売上げ] In ([年];[四半期])) In ([年])
```

ForEach コンテキスト演算子を使って、次の式を使った場合と同じ結果を導き出すことができます。

```
Max ([売上げ] ForEach ([四半期])) In ([年])
```

その理由は、“年”ディメンションは、ブロックのデフォルトの入力コンテキストだからです。ForEach 演算子を使用してコンテキストに[四半期]ディメンションを追加すると、入力コンテキストは([年];[四半期])になります。

6.2.5.3 ForAll コンテキスト演算子

ForAll コンテキスト演算子は、コンテキストからディメンションを除外します。

❖ 例

ForAll によるコンテキストからディメンションの除外

年、四半期、売上げを示すレポートで、次のブロックのように年別の売上げ合計を示す列を追加する場合を考えます。

年	四半期	売上げ	年度合計
2001	第1	¥ 2,660,700	¥ 8,095,814
2001	第2	¥ 2,278,693	¥ 8,095,814
2001	第3	¥ 1,367,841	¥ 8,095,814
2001	第4	¥ 1,788,580	¥ 8,095,814
2002	第1	¥ 3,326,172	¥ 13,232,246
2002	第2	¥ 2,840,651	¥ 13,232,246
2002	第3	¥ 2,879,303	¥ 13,232,246
2002	第4	¥ 4,186,120	¥ 13,232,246
2003	第1	¥ 3,742,989	¥ 15,059,143
2003	第2	¥ 4,006,717	¥ 15,059,143
2003	第3	¥ 3,953,395	¥ 15,059,143
2003	第4	¥ 3,356,041	¥ 15,059,143

年別に売上げを合計するには、入力コンテキストが(年) でなければなりません。しかし、デフォルトコンテキストは(年; 四半期) です。このため、式に ForAll ([四半期]) と指定することで入力コンテキストから四半期を除外できます。結果は、次のような式になります。

```
Sum([売上げ] ForAll ([四半期]))
```

In 演算子を使用しても同じことができます。この場合は次の式になります。

```
Sum([売上げ] In ([年]))
```

この式では、四半期を除外して年を残すのではなく、コンテキストとして年を明白に指定しています。

6.2.6 集合演算子

集合演算子は階層型データのメンバーに対して使用できます。

6.2.6.1 範囲演算子

説明

範囲演算子(:) は、同じレベルにある 2 つのメンバーとその間にあるメンバーの集合を返します。

構文

```
first_member:last_member
```

例

[地理]&[米国].[カリフォルニア].[ロサンゼルス]:[地理]&[米国].[カリフォルニア].[サンフランシスコ] は、このレベルのメンバーが ...[ロサンゼルス],[サンディエゴ],[サンフランシスコ]... の順になっている場合、[ロサンゼルス],[サンディエゴ],[サンフランシスコ] を返します。

Sum([売上げ];{[地理]&[米国].[カリフォルニア].[ロサンゼルス]:[地理]&[米国].[カリフォルニア].[サンフランシスコ]}) は、ロサンゼルス、サンディエゴ、サンフランシスコの売上げ合計を返します。

6.3 拡張構文キーワード

拡張構文キーワードは、拡張構文のディメンションを明示的に指定しなくても参照できる、"速記"のようなものです。

これらのキーワードは、レポートを将来にわたって使用する場合に便利です。式にディメンションをハードコード化して参照しないことで、レポートにディメンションが追加されたり、削除されても、式はそのまま有効に機能します。

拡張構文のキーワードには、Report、Section、Break、Block および Body の 5 つがあります。

6.3.1 Block キーワード

このトピックでは、Block キーワードで参照されるディメンションを、レポート内のキーワードが配置される場所に基づいて示します。Block キーワードには、多くの場合、Section キーワードと同じデータが含まれます。

ただし、Block では、Section キーワードが無視するブロックのフィルタも考慮されます。

場所	参照されるデータ
ブロック	ブロック全体のデータで、ブレイクは無視するが、フィルタは考慮する
ブロックのブレイク(ヘッダまたはフッタ)	ブロック全体のデータで、ブレイクは無視するが、フィルタは考慮する
セクション(ヘッダ、フッタ、またはブロックの外)	使用できません。
ブロックまたはセクションの外	適用外

❖ 例

Block キーワード

年、四半期、売上げを表示したレポートがあります。このレポートには、[年]セクションがあります。ブロックには第 3 と第 4 四半期を除外するフィルタが適用されています。

2001

四半期	売上げ	前期平均	年平均
第1	¥2,660,700	¥2,469,696	¥8,095,814
第2	¥2,278,693	¥2,469,696	¥8,095,814
合計:	¥4,939,393		

2002

四半期	売上げ	前期平均	年平均
第1	¥3,326,172	¥3,083,412	¥13,232,246
第2	¥2,840,651	¥3,083,412	¥13,232,246
合計:	¥6,166,823		

2003

四半期	売上げ	前期平均	年平均
第1	¥3,742,989	¥3,874,853	¥15,059,143
第2	¥4,006,717	¥3,874,853	¥15,059,143
合計:	¥7,749,706		

年平均列の式は、次のようになります。

```
Average([Sales revenue] In Section)
```

前期平均列の式は、次のようになります。

```
Average ([Sales revenue]) In Block
```

Block キーワードがブロックへのフィルタを考慮していることがわかります。

6.3.2 Body キーワード

このトピックでは、Body ブロック内のキーワードで参照されるディメンションを、レポート内のキーワードが配置される場所に基づいて示します。

場所	参照されるデータ
ブロック	ブロックのデータ
ブロックのブレーク(ヘッダまたはフッタ)	ブロックのデータ
セクション(ヘッダ、フッタ、またはブロックの外)	セクションのデータ

場所	参照されるデータ
ブロックまたはセクションの外	レポートのデータ

❖ 例

Body キーワード

年、四半期、売上げを表示し、[年]にブレイクが適用されているレポートがあります。レポートには、[年]セクションがあり、四半期にブレイクが適用されています。

年	四半期	売上げ	本文
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	

[Body]列の式は、次のようになります。

```
Sum ([Sales Revenue]) In Body
```

[Body]列の合計は[売上げ]列と同じですが、これは Body キーワードがブロックのデータを参照しているためです。[月]オブジェクトを除外すると、[Body]列の値は[売上げ]列の値の変更に合わせて変わります。レポートのフッタにこの式を入力すると、ボディの売上げ合計が返されます。

6.3.3 Break キーワード

次の表は、Break キーワードで参照されるディメンションを、レポート内のキーワードが配置される場所に基づいて示したものです。

場所	参照されるデータ
ブロック	ブレイクで区切られたブロック部分のデータ
ブロックのブレイク(ヘッダまたはフッタ)	ブレイクで区切られたブロック部分のデータ
セクション(ヘッダ、フッタ、またはブロックの外)	使用できません。
ブロックまたはセクションの外	使用できません。

❖ 例

Break キーワード

年、四半期および売上げをあらわすレポートがあります。

年	四半期	売上げ	ブレイク合計
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

このレポートには、[年] にブレイクが適用されています。[ブレイクの合計] 列には、次のような式があります。

```
Sum ([Sales Revenue]) In Break
```

Break キーワードを指定しない場合は、この列にはデフォルトの出力コンテキスト ([年];[四半期]) に従って [売上げ] 列と同じ値が表示されます。

6.3.4 Report キーワード

このトピックでは、Report キーワードで参照されるデータを、レポート内のキーワードが配置される場所に基づいて示します。

場所	参照されるデータ
ブロック	レポートのすべてのデータ
ブロックのブレイク(ヘッダまたはフッタ)	レポート内の全データ
セクション(ヘッダ、フッタ、またはブロックの外)	レポート内の全データ
ブロックまたはセクションの外	レポート内の全データ

❖ 例

Report キーワード

年、四半期、売上げを表示したレポートがあります。このレポートには、レポートのすべての売上げ合計を示す、[このレポートの総合計]列があります。

年	四半期	売上げ	レポート総合計
2001	第1	¥ 2,660,700	¥ 36,387,203
2001	第2	¥ 2,278,693	¥ 36,387,203
2001	第3	¥ 1,367,841	¥ 36,387,203
2001	第4	¥ 1,788,580	¥ 36,387,203
2002	第1	¥ 3,326,172	¥ 36,387,203
2002	第2	¥ 2,840,651	¥ 36,387,203
2002	第3	¥ 2,879,303	¥ 36,387,203
2002	第4	¥ 4,186,120	¥ 36,387,203
2003	第1	¥ 3,742,989	¥ 36,387,203
2003	第2	¥ 4,006,717	¥ 36,387,203
2003	第3	¥ 3,953,395	¥ 36,387,203
2003	第4	¥ 3,356,041	¥ 36,387,203

[このレポートの総合計] 列の式は、以下のとおりです。

```
Sum([Sales revenue]) In Report
```

Report キーワードを指定しない場合、この列にはデフォルトの出力コンテキスト ([年];[四半期]) に従って[売上げ] 列と同じ値が表示されます。

6.3.5 Section キーワード

このトピックでは、Section のキーワードで参照されるデータを、レポート内のキーワードが配置される場所に基づいて示します。

場所	参照されるデータ
ブロック	セクションのすべてのデータ
ブロックのブレイク(ヘッダまたはフッタ)	セクション内の全データ
セクション(ヘッダ、フッタ、またはブロックの外)	セクション内の全データ
ブロックまたはセクションの外	使用できません。

❖ 例

Section キーワード

年、四半期、売上げを表示したレポートがあります。

2001		
四半期	売上げ	セクション合計
第1	¥2,660,700	¥8,095,814
第2	¥2,278,693	¥8,095,814
第3	¥1,367,841	¥8,095,814
第4	¥1,788,580	¥8,095,814

このレポートには、[年]セクションがあります。[このセクションの合計]列には、次のような式があります。

```
Sum ([Sales Revenue]) In Section
```

セクションのブレイクが[年]オブジェクトに適用されているため、[このセクションの合計]列の値は 2001 年の売上げ合計になります。Section キーワードを指定しない場合は、この列にはデフォルトの出力コンテキスト ([年]:[四半期]) に従って[売上げ]列と同じ値が表示されます。

6.4 数値の四捨五入と切り捨て

一部の関数には、関数が返す値に対して四捨五入または切り捨てを行うレベルを決定するパラメータが含まれています。このパラメータには、0 以上、0、または 0 未満のいずれかの整数を指定できます。次の表は、数値を四捨五入したり切り捨てる方法を、それぞれの場合について説明しています。

パラメータ	説明
> 0	<p>関数は、<パラメータ> で指定した小数点以下の桁数へ四捨五入または切り捨てを行います。</p> <p>例:</p> <p>Round (3.13; 1) は 3.1 を返します。</p> <p>Round (3.157; 2) は 3.16 を返します。</p>
0	<p>関数は、数値を四捨五入した整数または数値を切り下げた整数を返します。</p> <p>例:</p> <p>Truncate (3.7; 0) は 3 を返します。</p> <p>Truncate (4.164; 0) は 4 を返します。</p>
< 0	<p>関数は、最も近い 10 単位の数値(パラメータが -1 の場合)、最も近い 100 単位の数値(パラメータが -2 の場合)、最も近い 1000 単位の数値(パラメータが -3 の場合)などに四捨五入または切り捨てます。</p>

パラメータ	説明
	例:
	Round(123.76;-1) は 120 を返します。
	Round(459.9;-2) は 500 を返します。
	Truncate(1600;-3) は 1000 を返します。

i 注記

数値は内部で倍精度浮動小数点形式で表示され、15 から 17 桁までの精度です。

関連情報

[Round \[174 ページ\]](#)

[Truncate \[179 ページ\]](#)

[EuroConvertTo \[160 ページ\]](#)

[EuroConvertFrom \[159 ページ\]](#)

[EuroFromRoundError \[162 ページ\]](#)

[EuroToRoundError \[164 ページ\]](#)

6.5 階層内のメンバーおよびメンバーセットの参照

関数のメンバーおよびメンバーセットを構文 `[hierarchy]&path.function` を使用して参照します。

`path` および `function` 部分はオプションです。`path` では、各メンバーを角かっこで囲んだうえ、ピリオドで区切って参照します。メンバーおよびレベルの名前では大文字と小文字が区別されます。

i 注記

階層のデフォルトの計算コンテキストを上書きするには、メンバーセットを使用します。メンバーセットを受け入れる関数の場合は、メンバーセットを `{}` で囲みます。

メンバーの範囲を参照するには、開始メンバーと終了メンバーの間にコロンの `(:)` を使用し、各メンバーの完全パスを指定します。範囲には、指定したメンバーと同じレベルのすべてのメンバーが含まれます。

範囲の構文の例は、`[Sales Hierarchy]&[Customer_Type].[ENTERPRISE];[Large].[Nancy Davolio]: [Sales Hierarchy]&[Customer_Type].[ENTERPRISE];[Large].[Andrew Smith]` です。

❖ 例

メンバーおよびメンバーセットの参照

最初に次のようなツリーを持っているとします。

営業階層	受注額
顧客タイプ	277,290,434
ENTERPRISE	180,063,361
大規模	113,905,997
Nancy Davolio	44,855,689
Janet Leverling	44,050,308
Andrew Smith	30,000,000
グローバル	91,157,363

- `[Sales Hierarchy]&[Customer_Type].[ENTERPRISE].[Large].Children` は、[Nancy Davolio]、[Janet Leverling] および [Andrew Smith] メンバーを参照します。
- `Sum([Order Amount];{[Sales Hierarchy]&[Customer_Type].[ENTERPRISE].[Large].children})` は 113,905,997 (3 つの子メンバーのメジャーの合計) を返します。
- `[Sales Hierarchy]&[Customer_Type].[ENTERPRISE].[Large].[Janet Leverling]` は [Janet Leverling] メンバーを参照します。
- `Sum([Order Amount];{[Sales Hierarchy]&[Customer_Type].[ENTERPRISE].[Large].[Janet Leverling];[Sales Hierarchy]&[Customer_Type].[ENTERPRISE].[Large].[Nancy Davolio]})` は 88,905,997 (2 つのメンバーのメジャーの合計) を返します。
- `[Sales Hierarchy]&[Customer_Type].[ENTERPRISE].[Large].[Nancy Davolio]:[Sales Hierarchy]&[Customer_Type].[ENTERPRISE].[Large].[Andrew Smith]` は、[Nancy Davolio]、[Janet Leverling] および [Andrew Smith] メンバーを参照します。
- `Sum([Order Amount];{[Sales Hierarchy]&[Customer_Type].[ENTERPRISE].[Large].[Nancy Davolio]:[Sales Hierarchy]&[Customer_Type].[ENTERPRISE].[Large].[Andrew Smith]})` は 113,905,997 (範囲の 3 つのメンバーのメジャーの合計) を返します。
- `[営業階層].children` は、[営業階層] 階層内のすべてのメンバーを参照します。
- `Sum([Order Amount];{Sales Hierarchy}.children)` は 277,290.434 を返します。

7 数式のトラブルシューティング

7.1 式自動書き換えメカニズム

Web Intelligence の一連の修正保守リリースでは、バージョン間で計算結果が異なることがあります。

バージョン 4.1 SP3 以降の Web Intelligence には、前のバージョンから移行されたドキュメントで式（以下の一覧を参照）の選択を自動的に変更する式自動書き換えメカニズムが用意されています。これらの式は特定のパターンに従っています。変更後、式は計算の変更前と同じ結果を返します。したがって、変更がドキュメントに保存され、式書き換えメカニズムが完了するように、ドキュメントを保存することをお奨めします。

式自動書き換えメカニズムは、デフォルトで、BI 4.1 SP3 以上に移行されたドキュメントに、以下の式パターンで 사용할 ことができます。

1. 条件にパラメータとしてディメンションを含む Where() 演算子
2. セクションでのリセットによる累積計算
3. クロステーブルでのリセットによる累積計算

このルールの一覧は、将来のリリースで、より多くの式パターンによって拡張される可能性があります。

ルール (1)

以前のバージョンでは、条件にパラメータとしてディメンションを含む Where() 演算子を使用すると、データが特定の方法で計算されていました。実際には、ディメンションがメジャーコンテキストに追加されました。ルール (1) ではこの以前の動作を再現しています。

このルールは、XI 3.1 FP3.6、XI 3.1 FP4.1、XI 3.1 FP5.1、および 4.0 SP5 に適用されます。

ルール (2)

以前のバージョンでは、各セクションインスタンスで計算がリセットされたため、セクションでの累計計算が適切に実行されませんでした。ルール (2) ではこの以前の動作を再現しています。

このルールは、XI R2 SP4 から移行されたすべてのドキュメントに適用されます。

ルール (3)

以前のバージョンでは、クロステーブルでのリセットによる累積計算は、計算が "Z" パターン（行から行）ではなく "N" パターン（列から列）で実行されることを意味していました。

ルール (3) では、Web Intelligence に "N" パターンによる累積計算を強制する FORCE_COL キーワードを導入しています。

たとえば、ルール (3) では、式 `RunningSum([Sales revenue];([State]))` を `RunningSum([Sales revenue];FORCE_COL;([State]))` に変更することにより、列から列への実行を強制します。

このルールは、XI 3.x、4.0 パッチ 2.20、4.0 SP5、4.0 SP6、4.0 SP7、4.1 および 4.1 SP1 のすべてのバージョンから移行されたすべてのドキュメントに適用されます。

7.2 式のエラーメッセージと情報メッセージ

エラーメッセージを返すレポートデータの書式を、条件付き書式設定を使用して設定することができます。

式から値を返すことができず、先頭に # が付いたエラーメッセージまたは情報メッセージが返される場合があります。メッセージは、式を入力したセル内に表示されます。

7.2.1 #COMPUTATION

#COMPUTATION は、`RelativeValue` 関数で指定されたスライスディメンションが、関数が配置されているブロックの計算コンテキストで使用できなくなった場合に発生します。

#COMPUTATION は、階層を含む結合オブジェクトがレポートに含まれる場合にも発生します。

また #COMPUTATION は、式でのコンテキスト演算子の使い方が間違っている場合に発生します。

関連情報

[RelativeValue \[219 ページ\]](#)

7.2.2 #CONTEXT

存在しない計算コンテキストがメジャーに含まれる場合、メジャーに #CONTEXT と表示されます。

#CONTEXT は、#INCOMPATIBLE および #DATASYNC エラーメッセージに関連します。これらのエラーメッセージは、存在しない計算コンテキストがブロックに含まれている場合にディメンションに表示されます。

#INCOMPATIBLE の場合は、ディメンションに互換性がないため、コンテキストは存在しません。#DATASYNC の場合は、ディメンションが複数の非同期のデータプロバイダから取得されるため、計算コンテキストは存在しません。

❖ 例

クエリ内の存在しない計算コンテキスト

[アイランドリゾートマーケティング]ユニバースに基づくブロックに、[予約年]および[売上げ]オブジェクトが含まれている場合は、予約年によって売上げを集計できないため、#CONTEXT エラーメッセージが表示されます。つまり、予約ではまだ売上げが発生していません。

7.2.3 #DATASYNC

#DATASYNC は、あるデータプロバイダのディメンションを含むブロックに異なるデータプロバイダのディメンションを配置し、2 つのデータプロバイダが結合ディメンションを通じて同期されていないと発生します。

#DATASYNC は、ブロック内のすべてのディメンションに表示され、#CONTEXT はメジャーに表示されます。

❖ 例

ブロック内の別のデータプロバイダからのディメンション

“アイランドリゾートマーケティング”ユニバースに基づくレポートに、オブジェクト(年, 売上げ) および(四半期)を含むデータプロバイダが含まれている場合は、2 つのデータプロバイダが結合ディメンションで同期されないため、年、四半期、売上げを含むブロックの“年”および“四半期”列には #DATASYNC が表示されます。

7.2.4 #DIV/0

#DIV/0 エラーは、数学的に不可能な 0 での除算を行おうとすると発生します。

0 は除数としては使用できません。

❖ 例

商品別の売上げ

売上げ、商品別販売個数、商品別売上げ(売上げを販売個数で割ったもの)を表示したレポートがあるとします。

特定の四半期の実績が非常に悪く、まったく売れなかった商品があった場合、この四半期の“商品別売上げ”列には #DIV/0 が表示されます。これは 0 による除算、つまり販売個数ゼロで売上げを割ろうとしたためです。

7.2.5 #ERROR

#ERROR は、他のエラーメッセージでは扱われないすべてのエラーを示すためのデフォルトのエラーメッセージです。

7.2.6 #EXTERNAL

#EXTERNAL は、式が Web Intelligence で使用できない外部関数を参照する場合に発生します。

7.2.7 #INCOMPATIBLE

#INCOMPATIBLE は、ブロックに互換性のないオブジェクトが含まれている場合に発生します。

❖ 例

クエリ内で互換性のないオブジェクト

[アイランドリゾートマーケティング]ユニバースに基づくブロックに[年]および[予約年]ディメンションが含まれている場合、これらのオブジェクトは互換性がないため、これらのディメンションを含む列には #INCOMPATIBLE が示されます。

7.2.8 #MIX

#MIX は、集計メジャーに異なる単位が含まれる場合に発生します。

たとえば、異なる通貨建ての通貨値が集計されるセルに #MIX が表示されます。

7.2.9 #MULTIVALUE

#MULTIVALUE は、1 つの値だけを出力するセルに複数の値を返す数式を入力すると表示されます。

❖ 例

セル内の複数の値

国、リゾート、売上げを示すレポートに、数式[売上げ]ForEach([国])を入力したセルを追加しますレポート内の国の値が“US”と“France”の 2 つだとすると、このセルは #MULTIVALUE を返します。

1 つのセルに、US と France の両方の売上げを表示することはできません。テーブルの外部に売上げを表示するセルがある場合は、そのセルではテーブル内の売上げをいくつかの方法 (合計、平均など) で集計することができます。

レポートが国別のセクションに分かれていれば、各セクションには 1 つの国の値しかないため、セクション内の数式は正しくなります。しかしセクション外では、この数式は #MULTIVALUE を返します。

7.2.10 #N/A

レポートのセルの値が、基盤となるデータベースで使用できないレポートの値に基づいており (たとえば BEx Cell の BW エラー)、セルに #N/A (使用不可) と表示されている場合、それは、データが取得できないためセルが空であるという意味です。

7.2.11 #OVERFLOW

#OVERFLOW は、計算がソフトウェアでは処理しきれないほど大きい値を返したときに発生します。

指数形式でのこの値は、1.7E308 (1.7 の後に 0 が 307 個) です。

7.2.12 #PARTIALRESULT

#PARTIALRESULT は、レポートオブジェクトに関連するすべての行を取得できなかった場合に表示されます。

#PARTIALRESULT がレポートで頻繁に発生し、相応するセキュリティ権限を持っている場合は、MaxRowsRetrieved クエリプロパティを変更してより多くのデータを取得できるようにします。クエリプロパティを変更する権限がない場合は、BI 管理者に連絡してください。

レポートにスマートメジャーが含まれている場合、スマートメジャーはクラシックメジャーよりも大量のデータを取得する必要があるため、#PARTIALRESULT が表示されやすくなります。

7.2.13 #RANK

#RANK は、値の順序に依存しているオブジェクトに基づいてデータに順位を付けようとすると発生します。

Previous 関数または実行集計関数を使用するオブジェクトは、値の順序に依存します。

順位付けによってこれらのオブジェクトの値が再計算されるので、順位が変更され、その結果、循環依存になります。このような循環依存は、[順位] ダイアログボックスを使用して順位を作成している場合、または Rank 関数を使用している場合に発生する可能性があります。

❖ 例

実行平均値または前の値に対する順位

Previous 関数または実行集計関数を含む列でブロックに順位を付けようとすると、ブロック全体が #RANK を返します。

7.2.14 #RECURSIVE

#RECURSIVE は、循環依存が原因で計算ができない場合に発生します。

❖ 例

NumberOfPages()関数の使用

高さの自動調整プロパティまたは幅の自動調整プロパティが設定されているセルに `NumberOfPages` 関数を指定すると、セルから #RECURSIVE が返されます。自動調整セルにこの式を指定すると循環依存が作成されるためです。この関数から値を返すためには、レポートの正確なサイズが必要です。ただし、レポートのサイズに影響を与えるセルのサイズは、セルコンテンツによって決定されます。

7.2.15 #REFRESH

#REFRESH が表示されるのは、クエリから除外された後でクエリに追加し直されたオブジェクトから派生する値を保持するレポートセルです。

[[クエリストリッピングを有効にする](#)] クエリプロパティが選択されていて、なおかつそのオブジェクトがクエリを使用するどのレポートにも寄与していない場合は、クエリからそのオブジェクトが除外されます。

クエリが最新表示されると、オブジェクトからセルに値が再入力されます。

7.2.16 #SECURITY

#SECURITY は、セキュリティ権限を持っていない関数を使用しようとすると発生します。

❖ 例

DataProviderSQL() 関数の使用

データプロバイダ SQL を表示する権限を持っていないユーザがセル内に `DataProviderSQL()` 関数を配置すると、#SECURITY メッセージが表示されます。

7.2.17 #SYNTAX

#SYNTAX は、式がレポートに存在していないオブジェクトを参照していると発生します。

❖ 例

削除されたオブジェクトの参照

年、四半期、売上げを示すレポートに、売上げと年平均売上げの差を示す列を追加したレポートがあります。この数値は[年平均との差]変数を使って表示しています。

[年平均との差]変数がレポートから削除されると、その変数を含む列は、#SYNTAX を返します。

7.2.18 #TOREFRESH

#TOREFRESH は、スマートメジャーから返された値が使用できない場合に、スマートメジャーに基づくセルに表示されません。

このエラーが発生するのは、値を含むグループ化集合がデータプロバイダで使用できない場合に発生します。

データを最新表示して、#TOREFRESH エラーを削除します。

一部のメジャーは "依頼" されます (BW では、SUM を使用して集計されないメジャーをこう呼びます)。メジャーでテーブルや計算を定義すると、集計における特定のコンテキストでこのメジャーがクエリされます (メジャーはディメンションのセットとして指定されます)。このディメンションのセットは、クエリディメンションセットのサブセットです。メジャーは任意のディメンションセット (または、SQL 内の GROUP BY 句を参照するグループセット) に従って集計される必要があります。

通常のメジャーの場合はシステムが集計を実行しますが、依頼されたメジャーの場合はこの集計が基盤となっているデータベースに依頼されます。そのため、システムでこのデータベースに対して再度クエリをする必要があります。これは自動的に行われないため、システムは #TOREFRESH を表示し、ユーザが最新表示して先に進めるのを待ちます。ユーザが最新表示すると、システムは追加のクエリを実行して要求された集計を取得してから、#TOREFRESH を適切な値に置換します。

7.2.19 #UNAVAILABLE

#UNAVAILABLE は、スマートメジャーの値を計算できない場合に表示されます。

このエラーは、クエリにフィルタを適用しないとフィルタ処理されたスマートメジャーで値を表示できない場合に発生します。これによって、同じクエリに基づく他のレポートに影響するリスクが発生するため、フィルタは適用されません。

8 関数を使用した値の比較

8.1 Previous 関数を使用した値の比較

Previous 関数は式の直前の比較値を返します。

返される値はレポートのレイアウトにより異なります。

より強力な比較機能が必要な場合は、RelativeValue 関数を使用します。RelativeValue は、式の直前または後続の比較値を返します。返される値はレポートのレイアウトに依存しません。

関連情報

[Previous \[215 ページ\]](#)

[RelativeValue \[219 ページ\]](#)

[RelativeValue 関数を使用した値の比較 \[255 ページ\]](#)

8.2 RelativeValue 関数を使用した値の比較

RelativeValue 関数は式の比較値を返します。返される値はレポートのレイアウトに依存しません。

RelativeValue を使用する場合は、以下を指定します。

- 検索する比較値が含まれる式。式はブロック内で使用可能なメジャーまたはディメンションの詳細であることが必要です。
- スライスディメンションの一覧
- オフセット

関数は、スライスディメンション、オフセット、サブ軸ディメンション(スライスディメンションによって示される)を使用して、比較値を返します。サブ軸ディメンションは、スライスディメンションを除く計算コンテキスト内の他のすべてのディメンションです。

一般的に記述される RelativeValue 関数は、スライスディメンションの値一覧の中にあり、現在の行から削除されたオフセット行にある式の値を返します。ここでは、サブ軸ディメンションの値は現在の行の値と同じです。

i 注記

すべてのスライスディメンションは、関数が配置されるブロックの計算コンテキスト内に常に存在する必要があります。スライスディメンションが後で削除された場合、関数は #COMPUTATION を返します。

❖ 例

以下に示す例では、RelativeValue 列には次の式が含まれます。

```
RelativeValue([売上げ];([年]);-1)
```

- 式は [売上げ]; です。
- スライスディメンションは [年]; です。
- オフセットは -1 です。関数はリスト内の直前の値を返します。

年	四半期	販売担当者	売上げ	RelativeValue
2007	Q1	スミス	1000	
2007	Q2	ジョーンズ	2000	
2007	Q3	ウィルソン	1500	
2007	Q4	ハリス	3000	
2008	Q1	スミス	4000	1000
2008	Q2	ジョーンズ	3400	2000
2008	Q3	ウィルソン	2000	1500
2008	Q4	ハリス	1700	3000

ビジネスクエスションとして記述された式は、前年の同じ四半期に同じ販売担当者が達成した売上げを返します。

単語を使用した計算として記述された式は、"年"(スライスディメンション)の値が"年"オブジェクトの値一覧で直前にある値になっていて、"四半期"および"販売担当者"(サブ軸ディメンション)の値が現在の行と同じ値になっている行の"売上げ"の値(式)を返します。

関連情報

[RelativeValue \[219 ページ\]](#)

8.2.1 スライスディメンションと RelativeValue 関数

RelativeValue 関数は、スライスディメンションの値の一覧を使用して比較行を検索します。

この関数は、スライスディメンションの一覧内でオフセット行数分だけ離れた関数で指定された式の比較値を返します。

その結果、スライスディメンションの並べ替え順は、関数の出力の決定に非常に重要なものとなります。

❖ 例

複数のスライスディメンション

次のテーブルで、RelativeValue 列には次の式が含まれます。

```
RelativeValue([売上げ];([年];[四半期]);-1)
```


- 式は [売上げ]; です。
- スライスディメンションは ([年]:[四半期]); です。
- オフセットは -1 です。関数はリスト内の直前の値を返します。

年	四半期	販売担当者	売上げ	RelativeValue
2007	Q1	スミス	1000	
2007	Q2	スミス	2000	
2007	Q3	スミス	1500	
2007	Q4	スミス	3000*	
2007	Q1	ジョーンズ	4000	
2007	Q2	ジョーンズ	3400	
2007	Q3	ジョーンズ	2000	
2007	Q4	ジョーンズ	1700	
2008	Q1	スミス	5000**	3000*
2008	Q2	スミス	3000***	5000**
2008	Q3	スミス	2700****	3000***
2008	Q4	スミス	6800	2700****

ビジネスクエスチョンとして記述された式は、前の四半期に同じ販売担当者が達成した売上げを返します。

単語を使用した計算として記述された式は、"年" と "四半期" の値が ("年","四半期") の値の一覧で直前にある値になっていて、"販売担当者" の値が現在の行と同じ値になっている行の "売上げ" の値を返します。

この関数は、スライスディメンションの値の一覧を使用して、比較対象となる売上げを検索します。

年	四半期	
2007	Q1	
2007	Q2	
2007	Q3	
2007	Q4	*
2008	Q1	**
2008	Q2	***
2008	Q3	****
2008	Q4	

スライスディメンションの並べ替え順により、関数の出力が決定されます。テーブル内の * は並べ替え順を示します。

関連情報

[RelativeValue \[219 ページ\]](#)

8.2.2 スライスディメンションとセクション

スライスディメンションは、レポートのセクションのマスタセル内に置くことができます。

❖ 例

スライスディメンションはセクションヘッダです。

次のテーブルで、RelativeValue 列には次の式が含まれます。

```
RelativeValue([売上げ];([年];[四半期]));-1)
```

2007

四半期	販売担当者	売上げ	RelativeValue
Q1	スミス	1000	
Q2	スミス	2000	
Q3	スミス	1500	
Q4	スミス	3000*	
Q1	ジョーンズ	4000	
Q2	ジョーンズ	3400	
Q3	ジョーンズ	2000	
Q4	ジョーンズ	1700	

2008

四半期	販売担当者	売上げ	RelativeValue
Q1	スミス	5000**	3000*
Q2	スミス	3000***	5000**
Q3	スミス	2700 ****	3000***
Q4	スミス	6800	2700****

この関数は、スライスディメンションの値の一覧を使用して、比較対象となる売上げを検索します。

年	四半期	
2007	Q1	
2007	Q2	
2007	Q3	
2007	Q4	*
2008	Q1	**
2008	Q2	***
2008	Q3	****
2008	Q4	

スライスディメンションの並べ替え順により、関数の出力が決定されます。テーブル内の * は並べ替え順を示します。

関連情報

[RelativeValue \[219 ページ\]](#)

8.2.3 スライスディメンションの順序

スライスディメンションの値のリストの並べ替え順により `RelativeValue` の出力が決定されるため、スライスディメンションを指定する順序は関数の出力に影響を与えます。

❖ 例

スライスディメンションの順序

次のテーブルで、`RelativeValue` 列には次の式が含まれます。

```
RelativeValue([売上げ];([年];[四半期]));-1)
```

年	四半期	販売担当者	売上げ	RelativeValue
2007	Q1	スミス	1000	
2007	Q2	スミス	2000	
2007	Q3	スミス	1500	
2007	Q4	スミス	3000*	
2007	Q1	ジョーンズ	4000	
2007	Q2	ジョーンズ	3400	
2007	Q3	ジョーンズ	2000	
2007	Q4	ジョーンズ	1700	
2008	Q1	スミス	5000**	3000*
2008	Q2	スミス	3000***	5000**
2008	Q3	スミス	2700****	3000***
2008	Q4	スミス	6800	2700****

ビジネスクエションとして記述された式は、前の四半期に同じ販売担当者が達成した売上げを返します。

スライスディメンションの並べ替え順は以下のとおりです。

年	四半期
2007	Q1
2007	Q2

年	四半期	
2007	Q3	
2007	Q4	*
2008	Q1	**
2008	Q2	***
2008	Q3	****
2008	Q4	

関数は次のように変更されます。

```
RelativeValue([四半期];([四半期];[年]);-1)
```

スライスディメンションの並べ替え順は以下のようになります。

四半期	年	
Q1	2007	*
Q1	2008	**
Q2	2007	***
Q2	2008	****
Q3	2007	*****
Q3	2008	*****
Q4	2007	*****
Q4	2008	*****

並べ替え順は、関数の結果に以下の影響を与えます。

年	四半期	販売担当者	売上げ	RelativeValue
2007	Q1	スミス	1000*	
2007	Q2	スミス	2000***	
2007	Q3	スミス	1500*****	
2007	Q4	スミス	3000*****	
2007	Q1	ジョーンズ	4000	
2007	Q2	ジョーンズ	3400	
2007	Q3	ジョーンズ	2000	
2007	Q4	ジョーンズ	1700	
2008	Q1	スミス	5000**	1000*
2008	Q2	スミス	3000****	2000***
2008	Q3	スミス	2700*****	1500*****
2008	Q4	スミス	6800*****	3000*****

ビジネスクエスチョンとして記述された式は、今度は前年の同じ四半期に同じ販売担当者が達成した売上げを返します。

スライスディメンションの並べ替え順を変更すると、式の意味が変わります。テーブル内の * は並べ替え順を示します。

関連情報

[RelativeValue \[219 ページ\]](#)

8.2.4 スライスディメンションと並べ替え

スライスディメンションの値のリストの並べ替え順により関数の出力が決定されるため、スライスディメンションのどのディメンションに適用される並べ替えも関数の出力に影響を与えます。

❖ 例

スライスディメンションに適用されるカスタムの並べ替え

次のテーブルで、RelativeValue 列には次の式が含まれます。

```
RelativeValue([売上げ];([年];[四半期]));-1)
```

カスタムの並べ替え (第 1、第 2、第 4、第 3) が [四半期] に適用され、関数の結果は以下のようになります。

年	四半期	販売担当者	売上げ	RelativeValue
2007	Q1	スミス	1000	
2007	Q2	スミス	2000	
2007	Q4	スミス	3000	
2007	Q3	スミス	1500*	
2007	Q1	ジョーンズ	4000	
2007	Q2	ジョーンズ	3400	
2007	Q4	ジョーンズ	1700	
2007	Q3	ジョーンズ	2000	
2008	Q1	スミス	5000**	1500*
2008	Q2	スミス	3000***	5000**
2008	Q4	スミス	6800****	3000***
2008	Q3	スミス	2700	6800****

スライスディメンションの並べ替えリストは以下のとおりです。

年	四半期	
2007	Q1	
2007	Q2	
2007	Q4	
2007	Q3	*
2008	Q1	**
2008	Q2	***
2008	Q4	****
2008	Q3	

テーブル内の * は並べ替え順を示します。

関連情報

[RelativeValue \[219 ページ\]](#)

8.2.5 クロスタブでの RelativeValue の使用

RelativeValue 関数はクロスタブ内でも、垂直のテーブル内と全く同じように機能します。

クロスタブ内のデータのレイアウトは、関数の出力に影響を与えません。



関連情報

[RelativeValue \[219 ページ\]](#)

重要免責事項および法的情報

ハイパーリンク

リンクの一部は、アイコンやマウスオーバーテキストで分類されています。これらのリンクから、追加の情報を得ることができます。アイコンについて。

-  このアイコンが付いたリンク: SAP がホストしているものではない Web サイトに移動します。これらのリンクを使用することで、お客様は (お客様と SAP との契約書に別段の明示的な記載がない限り) 以下のことに同意することになります。
 - リンク先のサイトのコンテンツが SAP のドキュメンテーションではないこと。お客様は、この情報に基づいて SAP に対する製品クレームを推断することはできません。
 - SAP が、リンク先のサイトのコンテンツについて同意することも反対することもなく、また SAP がその利用可能性や正確性について保証しないこと。SAP は、かかるコンテンツの使用により発生した損害が、SAP の重大な過失又は意図的な違法行為が原因で発生したものでない限り、その損害に対して一切責任を負いません。
-  このアイコンが付いたリンク: 当該の特定の SAP 製品又はサービスのドキュメンテーションから離れ、SAP がホストしている Web サイトに移動します。これらのリンクを使用することで、お客様は (お客様と SAP との契約書に別段の明示的な記載がない限り)、この情報に基づいて SAP に対する製品クレームを推断することはできないことに同意します。

ベータおよびその他の試験的機能

試験的機能は、SAP が将来のリリースを保証する正式に提供される機能の範囲外です。これは、試験的機能は、SAP により通知なく理由の如何を問わず随時変更される場合があることを意味します。試験的機能は、本稼働使用のためのものではありません。お客様は、試験的機能を実際の運用環境で、又は十分なバックアップがとられていないデータとともに、デモンストレーション、テスト、試験、評価その他の方法で使用してはなりません。

試験的機能の目的は、早期にフィードバックを得ることで、それに応じて顧客の皆様やパートナーが将来の製品に影響を与えることを可能にすることです。SAP コミュニティなどにおいてフィードバックを提供することで、お客様は、投稿物や二次的著作物の知的財産権が SAP の独占的所有物であり続けることを承認することになります。

コード例

ソフトウェアのコーディングやコードスニペットはすべて、例です。それらは、本稼働使用のためのものではありません。コード例は、構文や表現規則を分かりやすく説明し視覚化することのみを目的としています。SAP は、コード例の正確性や完全性について保証しません。SAP は、コード例の使用により発生した過誤や損害が、SAP の重大な過失又は意図的な違法行為が原因で発生したものでない限り、損害に対して一切責任を負いません。

性別関連の文言

SAP は、一方の性に特化した語形や記述を用いないようにしています。文脈や読みやすさのために適切な場合、SAP ではすべての性別を指すために男性形の語句を使用する場合があります。

© 2018 SAP SE or an SAP affiliate company. All rights reserved.

本書のいかなる部分も、SAP SE 又は SAP の関連会社の明示的な許可なくして、いかなる形式でも、いかなる目的にも複製又は伝送することはできません。本書に記載された情報は、予告なしに変更されることがあります。

SAP SE 及びその頒布業者によって販売される一部のソフトウェア製品には、他のソフトウェアベンダーの専有ソフトウェアコンポーネントが含まれています。製品仕様は、国ごとに変わる場合があります。

これらの文書は、いかなる種類の表明又は保証もなして、情報提供のみを目的として、SAP SE 又はその関連会社によって提供され、SAP 又はその関連会社は、これら文書に関する誤記脱落等の過失に対する責任を負うものではありません。SAP 又はその関連会社の製品及びサービスに対する唯一の保証は、当該製品及びサービスに伴う明示的保証がある場合に、これに規定されたものに限られます。本書のいかなる記述も、追加の保証となるものではありません。

本書に記載される SAP 及びその他の SAP の製品やサービス、並びにそれらの個々のロゴは、ドイツ及びその他の国における SAP SE (又は SAP の関連会社) の商標若しくは登録商標です。本書に記載されたその他すべての製品およびサービス名は、それぞれの企業の商標です。

商標に関する詳細の情報や通知については、<https://www.sap.com/japan/about/legal/trademark.html> をご覧ください。