



PUBLIC (ОБЩЕДОСТУПНО)

SAP BusinessObjects Business Intelligence Suite

Версия документа: 4.3 Support Package 4 – 2023-12-07

Использование функций, формул и вычислений в Web Intelligence

Содержание

1	Использование функций, формул и вычислений для анализа данных.	3
1.1	История документа: функции, формулы и вычисления Web Intelligence.	3
1.2	О данном руководстве.	4
1.3	Использование стандартных и пользовательских вычислений.	4
	Знакомство с редактором формул.	4
	Использование стандартных и пользовательских вычислений.	5
1.4	Сведения о контекстах вычисления.	13
	Сведения о контекстах вычисления.	13
1.5	Расчет значений с помощью интеллектуальных показателей.	26
	Расчет значений с помощью интеллектуальных мер.	26
1.6	Функции, операторы и ключевые слова.	35
	Функции.	35
	Операторы функций и формул.	272
	Ключевые слова расширенного синтаксиса.	291
	Округление и усечение чисел.	296
	Ссылки на элементы и наборы элементов в иерархиях.	297
1.7	Создание пользовательских функций.	298
	Обзор внешних функций.	298
	Определение пользовательского вычисления.	301
	Примеры.	308
	сообщение об ошибке #EXTERNAL.	309
	Сообщения об ошибках журнала трассировки.	309
1.8	Устранение связанных с формулами неполадок.	310
	Механизм автоматической перезаписи формулы.	310
	Сообщения об ошибках формул и информационные сообщения.	311
1.9	Сравнение значений с помощью функций.	318
	Сравнение значений с помощью функции Previous.	318
	Сравнение значений с помощью функции RelativeValue.	318

1 Использование функций, формул и вычислений для анализа данных

1.1 История документа: функции, формулы и вычисления Web Intelligence

В следующей таблице описываются наиболее важные изменения документа.

Версия	Дата	Описание
SAP BusinessObjects Web Intelligence 4.3 SP3	Декабрь 2022 г.	<p>Следующие разделы были обновлены или добавлены в настоящее руководство:</p> <ul style="list-style-type: none">Добавлены новые функции Reverse [страница 104], ElementLinkingFilters [страница 167], InputControlFilter [страница 168], DescriptionOf [страница 241], FormulaOf [страница 243] и Next [страница 252].Обновлены Функция ReportFilter [страница 172] и Функция ToDate [страница 131].
SAP BusinessObjects Web Intelligence 4.3 SP1	Декабрь 2020 г.	<p>Следующие разделы были обновлены или добавлены в настоящее руководство:</p> <ul style="list-style-type: none">Добавлена новая функция RPos [страница 108].Обновлено Функция Pos [страница 102].
SAP BusinessObjects Web Intelligence 4.3	Июнь 2020 г.	<p>Следующие разделы были обновлены или добавлены в настоящее руководство:</p> <ul style="list-style-type: none">Добавлены новые функции DocumentDescription [страница 161], DocumentParentFolder [страница 162], DocumentPath [страница 164] и NumberOfColumns [страница 149].Теперь можно добавлять комментарии в коде формулы.Обновлено:

Версия	Дата	Описание
		<ul style="list-style-type: none"> • Функция Trim [страница 110], Функция LeftTrim [страница 99] и Функция RightTrim [страница 107]. Теперь можно указывать символы, которые требуется удалить. • Функция QuerySummary [страница 171] и Функция DataProviderType [страница 139]. Обе функции возвращают новые типы поставщиков данных.

1.2 О данном руководстве

В руководстве "Использование функций, формул и вычислений в Web Intelligence" представлены подробные сведения о расширенных возможностях вычислений, доступных при анализе данных.

В нем также содержится справочник по синтаксису доступных функций и операторов.

1.3 Использование стандартных и пользовательских вычислений

1.3.1 Знакомство с редактором формул

Редактор формул – это центральное место для создания расширенных вычислений и переменных.



Он был разработан для упрощения манипулирования объектами и быстрого создания формул или переменных с помощью встроенного редактора кода.

Редактор кода



Редактор кода предлагает множество функций для написания формул:

- Сопоставление скобок
- Анализ синтаксиса
- Цветовая кодировка
- Автоматическое заполнение




- Комбинации клавиш (**Ctrl** + **C**, **Ctrl** + **Z** и т. д.)
- Нумерация строк

Используйте соответствующий переключатель, чтобы включить перенос строк () , а также анализ синтаксиса и цветовую кодировку ().

Упрощенная версия редактора кода доступна на панели формул и боковых панелях Web Intelligence. Например, упрощенную версию редактора кода можно найти в текстовых полях формул для ячеек,

блоков и отчетов при переходе к  >  > *Настройки просмотра* > *Скрыть* > *Скрывать, когда формула верна*.

Панели "Объекты", "Функции" и "Операторы"

Три панели рядом с текстовым редактором позволяют быстро перетащить объекты в пределах редактора кода: *Объекты* () , *Функции* () и *Операторы* (). Каждую из этих панелей можно скрыть или изменить ее размер при необходимости.

Можно использовать кнопку *Развернуть все* (или *Свернуть все*) в верхней части каждой панели, чтобы развернуть (или свернуть) содержимое боковой панели.

Для получения справки по любому доступному объекту наведите на него указатель мыши, чтобы получить доступ к всплывающей подсказке с подробной информацией. Если щелкнуть одну из доступных функций или операторов, то в правом нижнем углу диалогового окна будет ссылка, ведущая на подробную документацию на справочном портале. При наведении указателя мыши на любой объект отображается всплывающая подсказка, которая содержит информацию из словаря объектов.

1.3.2 Использование стандартных и пользовательских вычислений

Стандартные функции вычислений можно использовать для быстрого выполнения вычислений с данными.

Если возможностей стандартных вычислений недостаточно, можно использовать язык формул для создания пользовательских вычислений.

1.3.2.1 Стандартные вычисления

Стандартные функции вычислений можно использовать для быстрого выполнения вычислений с данными.

Существуют следующие стандартные вычисления:

Вычисление	Описание
Sum	Вычисляет сумму выбранных данных.
Count	Подсчитывает все строки для объекта меры или подсчитывает отдельные строки для объекта измерения или сведений.
Average	Вычисляет среднее значение для данных.
Мин.	Отображает минимальное значение среди выбранных данных.
Макс.	Отображает максимальное значение среди выбранных данных.
Percentage	Отображает выбранные данные в виде процента от суммарного значения. Результаты вычисления процентов отображаются в таблице в дополнительном столбце или дополнительной строке.

❗ Примечание

Значения в процентах вычисляются для выбранной меры по отношению к итоговому результату для этой меры в таблице или в разрыве. Для вычисления процентного значения одной меры по отношению к другой мере необходимо создать формулу.

При применении стандартных вычислений к столбцам таблицы результаты вычислений отображаются в нижних колонтитулах. Для каждого вычисления добавляется по одному нижнему колонтитулу.

1.3.2.2 Использование формул для создания пользовательских вычислений

Пользовательские вычисления позволяют добавлять дополнительные вычисления к отчету поверх основных объектов и стандартных вычислений.

Пользовательские вычисления добавляются путем создания формулы. Формула может состоять из основных переменных отчета, функций, операторов и контекстов вычислений.

Пользовательское вычисление – это формула, которая состоит из объектов отчета, функций и операторов. В формулах содержится контекст вычисления, который может быть отображен при необходимости.

Пример: Отображение среднего дохода от каждой продажи

Если в отчете есть объекты «Доход от продаж» и «Проданное количество» и необходимо добавить в отчет доход от каждой продажи, вычисление `[Sales Revenue] / [Number Sold]` позволяет получить это значение путем деления дохода на количество проданных позиций, в результате чего получается доход по каждой позиции.

1.3.2.2.1 Использование переменных для упрощения формул

Переменные удобно использовать для разбиения формул на управляемые части, упрощающие чтение. Они также позволяют сделать процесс создания формул менее подверженным ошибкам.

Переменные можно найти на панели *Объекты* в разделе *Переменные*, наряду с другими объектами в запросе.

Используйте поле *Описание* для предоставления контекста и подробной информации о конкретной переменной. Введенное описание затем отображается при наведении указателя на переменную на *панели запросов*. Это описание можно редактировать при создании, редактировании или переименовании переменной.

1.3.2.3 Работа с функциями

В пользовательском вычислении содержатся только объекты отчетов, например [Доход от продаж] / [Количество продаж]. В вычислениях, кроме объектов отчета, также могут содержаться функции.

Функция получает ноль или несколько значений в качестве входных данных и возвращает выходные данные на основе этих значений. Например, функция Sum суммирует все значения в мере и выводит результат. Формула Sum([Доход от продаж]) выводит итог доходов от продаж. В данном случае входные данные функции – это мера "Доход от продаж", а выходные данные функций – сумма всех мер "Доход от продаж".

Связанные сведения

[Операторы функций и формул \[страница 272\]](#)

[Функции \[страница 35\]](#)

1.3.2.3.1 Включение функций в ячейки

Текст в ячейках отчета всегда начинается со знака "=".

Буквенный текст отображается в кавычках, а формулы – без кавычек. Например, формула Average([Прибыль]) отображается в ячейке как =Average ([Прибыль]). Текст "Средняя прибыль?" отображается как = "Средняя прибыль?"

В ячейке можно использовать только текст или смесь формул и текста, с помощью оператора "+". Если необходимо, чтобы перед средней прибылью в ячейке отображался текст "Средняя прибыль:", текст ячейки должен выглядеть следующим образом: = "Средняя прибыль: " + Average ([Прибыль])

Обратите внимание на пробел в конце текстовой строки, который позволяет отделить в ячейке текст от значения.

1.3.2.3.2 Синтаксис функций

При выборе функции в *редакторе формул* отображается ее синтаксис.

Чтобы использовать функцию, необходимо знать ее имя, количество необходимых для нее входных переменных и типы данных этих переменных. Также необходимо знать тип данных, который выводится функцией.

Например, функция `Sum` принимает числовой объект на входе (например, меру, отображающую доход от продаж) и возвращает числовые данные (сумму всех значений объекта меры).

Ниже приведен синтаксис функции `Abs`:

```
num Abs ( number )
```

Этот синтаксис показывает, что функция `Abs` принимает одно число на входе и возвращает число на выходе.

1.3.2.3.3 Примеры функций

В этом разделе приводятся примеры использования функций в формулах.

Пример: Отображение входа подсказок с помощью функции `UserResponse`

В отчете содержатся сведения о доходах от продаж за год и квартал. Объект состояния также отображается в данных отчета, хотя в самом отчете он не отображается. Когда пользователь выполняет отчет, ему выводится подсказка и он должен выбрать состояние. Необходимо отобразить состояние, выбранное в заголовке отчета. Если поставщик данных называется "eFashion", а текст подсказки – "Выберите штат", то формула для заголовка выглядит следующим образом:

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

Отчет выглядит следующим образом, если при обновлении поставщика данных для штата указано значение "Иллинойс":

Quarterly Revenues for Illinois

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

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

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

Пример: Вычисление процентного соотношения с помощью функции Percentage

Функция Percentage рассчитывает процентное соотношение. С помощью данной функции вычисляется процентное соотношение числа по отношению к окружающему контексту. Например, в следующей таблице отображаются доходы по годам и по кварталам. В столбце процентного соотношения содержится формула `Percentage ([Sales revenue])`.

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

В данном случае формула вычисляет каждый доход как процентное соотношение общего дохода. Окружающий контекст – это итоговый доход; это единственное необходимое число дохода, кроме его разбиения по годам и кварталам в таблице.

Если отчет разделен на разделы по годам, окружающий контекст вне таблицы становится итоговым доходом в разделе.

2004

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

Если ячейка "Процентное соотношение" помещена вне таблицы, но внутри раздела, окружающий контекст становится итоговым доходом. В данном случае функция Percentage вычисляет итоговый доход для раздела в качестве процентного соотношения.

2004

0.22

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

2005

0.38

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

Пример: Вычисление процентного соотношения с помощью функции Sum

Можно усилить контроль над контекстом, в котором процентное соотношение вычисляется с помощью функции Sum, вместо функции Percentage. Если разделить одно число во множестве чисел на сумму этих чисел, то можно получить процентное соотношение итога; например формула $[\text{Sales revenue}] / \text{Sum}([\text{Sales revenue}])$ представляет доход от продаж в качестве процентного соотношения от общего дохода.

В следующей таблице в столбце "Процентное соотношение итога" есть формула:

```
[Sales revenue] / (Sum([Sales revenue] In Report))
```

в столбце "Процентное соотношение года" также есть формула:

```
[Sales revenue] / (Sum([Sales revenue] In Section))
```

2004

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

В данных формулах используются ключевые слова Report и Section расширенного синтаксиса, чтобы с помощью функции Sum можно было вычислять итоговый доход и годовой доход соответственно.

Связанные сведения

[Изменение контекста вычислений по умолчанию с расширенным синтаксисом \[страница 22\]](#)

1.3.2.3.1 Упрощение формулы для дисперсии с помощью переменных

Дисперсия – это статистический термин. Дисперсия множества значений служит мерой рассеяния этих значений относительно их среднего.

Функция Var вычисляет дисперсию за один шаг, но ручное вычисление дисперсии служит хорошим примером того, как можно упростить сложную формулу, используя переменные. Чтобы вычислить дисперсию вручную, необходимо выполнить следующие действия:

- вычислить среднее количество проданных товаров;
- вычислить разность между каждым количеством проданных товаров и средним значением, а затем возвести это значение в квадрат;
- сложить все полученные квадраты разностей;
- разделить итог на количество значений минус единица.

Предположим, есть отчет по количеству товаров, проданных за квартал, и в него требуется включить дисперсию. Если не использовать переменные для упрощения формулы, то формула будет выглядеть следующим образом:

```
Sum(((Quantity sold] - Average([Quantity sold] ForEach [Quarter]) In Report)*([Quantity sold] - Average([Quantity sold] ForEach [Quarter]) In Report)) In [Quarter])/(Count ([Quantity sold] ForEach [Quarter]) - 1)
```

Создание формулы для дисперсии

Создание формулы для дисперсии выполняется за несколько шагов. Эти несколько шагов можно свести к одному с помощью одной переменной. Необходимо создать следующие переменные:

- среднее количество проданных товаров;
- количество наблюдений (то есть количество отдельных значений количества проданных товаров);
- разность между наблюдением и средним значением, возведенная в квадрат;
- сумма этих квадратов разностей, разделенная на количество наблюдений минус единица

Формулы с переменными имеют следующий вид.

Переменная	Формула
Average Sold	Average([Quantity sold] In ([Quarter])) In Report
Number of Observations	Count([Quantity sold] In ([Quarter])) In Report
Difference Squared	Power(([Quantity sold] - [Average sold]);2)
Variance	Sum([Difference squared] In ([Quarter]))/([Number of Observations] - 1)

Окончательная формула имеет вид:

```
Sum ([Difference Squared])/([Number of Observations] - 1)
```

В таком виде формула более понятна. Такой упрощенный вариант формулы дает представление более высокого уровня о том, что вычисляется в формуле, вместо того чтобы глубоко вникать в сбивающие с толку детали. Затем можно проверить формулы для переменных, на которые есть ссылки в формулах более высокого уровня, что даст более полное представление об их структуре.

Например формула ссылается на переменную Difference Squared, которая, в свою очередь, ссылается на переменную Average Sold. Проанализировав формулы для переменных Difference Squared и Average sold, можно выполнить развертку формулы и тем самым лучше понять механизм ее работы.

1.4 Сведения о контекстах вычисления

1.4.1 Сведения о контекстах вычисления

Контекст вычисления – это данные, которые учитывает вычисление, чтобы сгенерировать результат.

Это значит, что значение, предоставленное мерой, определяется с помощью измерений, которые используются для вычисления меры.

В отчете содержится два вида объектов.

- Измерения представляют бизнес-данные, которые генерируют цифры. Филиалы магазина, года или регионы – это примеры данных измерений. Например, филиал магазина, год или регион могут генерировать доход. Можно говорить о доходе по магазину, году или региону.

- Меры – это числовые данные, сгенерированные с помощью данных измерений. Примеры мер: доход и количество продаж. Например, можно говорить о количестве продаж в определенном магазине.

Меры также можно генерировать с помощью комбинаций данных измерений. Например, можно говорить о доходе, сгенерированном в определенном магазине в 2005 году.

Контекст вычисления меры состоит из двух компонентов:

- измерение или список измерений, которые определяют значение меры
- часть данных измерения, которая определяет значение меры

Контекст вычисления состоит из двух компонентов:

- Контекст ввода
- Контекст вывода

Связанные сведения

[Контекст ввода \[страница 14\]](#)

[Контекст вывода \[страница 15\]](#)

1.4.1.1 Контекст ввода

Контекст ввода меры или формулы – это список измерений, которые подаются на вычисление.

Список измерений во контексте ввода помещается внутри скобок функции, которая выводит значение. Список измерений также должен заключаться в скобки (даже если он содержит только одно измерение), а измерения должны разделяться точками с запятой.

Пример: Определение контекста ввода

В отчете с разделами "Год" и блоком в каждом разделе со столбцами "Заказчик" и "Доход" контексты ввода будут следующими.

Часть отчета	Контекст ввода
Ячейки раздела и нижние колонтитулы блоков	Год
Строки блока	"Год", "Заказчик"

Другими словами, в ячейках разделов и нижних колонтитулах блоков выводится прибыль по годам, а в каждой строке блока – прибыль по годам и заказчику (прибыль, полученная от данного заказчика за рассматриваемый год).

При явном указании в формуле контексты ввода будут следующими:

```
Sum ([Доход] In ([Год]))
```

```
Sum ([Доход] In ([Год];[Заказчик]))
```

Таким образом, измерения в контексте ввода оказываются внутри круглых скобок той функции (в данном случае – Sum), для которой указан контекст ввода.

1.4.1.2 Контекст вывода

С помощью контекста вывода формула выводит значение, если она находится в нижнем колонтитуле блока с разбиением.

Пример: Определение контекста вывода

В следующем отчете отображена прибыль годам и по кварталам, с разбиением по году и минимальной прибылью, вычисленной по годам.

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

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

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

Что необходимо сделать, чтобы отобразить минимальный доход по годам в блоке с разбиением? Это можно осуществить, определив контекст вывода в формуле. В этом случае данная формула выглядит следующим образом.

```
min ([Доход от продаж]) In ([Год])
```

То есть контекст вывода отображается после круглых скобок функции, чей контекст вывода указывает пользователь. В этом случае контекст вывода рассчитывает минимальный доход за год.

Если добавить дополнительный столбец с данной формулой в блок, результат будет выглядеть следующим образом.

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

Можно увидеть, что в столбце "Минимум за каждый год" содержатся данные о минимальных доходах, которые отображаются в нижних колонтитулах разбиения предыдущего отчета.

Обратите внимание, что в данном примере контекст ввода не указан, так как это контекст по умолчанию (год, квартал) для блока. Другими словами, контекст вывода определяет, какой доход по годам и кварталам следует выводить. Если явно привести все входные и выходные формулы, то данная формула будет выглядеть следующим образом.

```
min ([Доход от продаж] In([Год];[Квартал])) In ([Год])
```

Эта формула вычисляет доходы за годы поквартально, а затем выводит наименьший из этих доходов за каждый год.

Что произойдет, если не определить контекст вывода в столбце "Минимум за год"? В этом случае данные цифры будут совпадать с цифрами в столбце "Доход от продаж". Причина: Следует помнить, что контекст по умолчанию в блоке содержит измерения в данном блоке. Минимальный доход за каждый год/квартал совпадает с доходом за каждый год/квартал, так как имеется лишь одно значение дохода для каждого сочетания "год/квартал".

1.4.1.3 Контексты вычислений по умолчанию

У меры есть контекст вычисления по умолчанию, зависящий от ее положения в отчете.

Значения, возвращаемые мерой, зависят от измерений, с которыми она связана. Это сочетание измерений представляет контекст вычисления.

Контекст по умолчанию можно изменить с помощью расширенного синтаксиса. Другими словами, можно определить набор измерений, который используется для создания меры. Это и есть определение контекста вычисления.

Пример: Контексты по умолчанию в отчете

В данном примере описывается контекст вычисления мер по умолчанию в простом отчете. В отчете отображается доход, созданный покупателями. Отчет разбит на разделы по годам.

2005	Всего: 8000
Покупатель	Доход
Харрис	1000
Джонс	3000
Уолш	4000
Всего:	8000
Итого в отчете: 8000	

В таблице, указанной выше, содержится контекст вычисления мер в следующем отчете.

Показатель	Значение	Контекст
Итого в отчете	20000	Итоговое значение общего дохода в отчете
Сумма ячейки раздела	8000	Год
Итого по всем заказчикам	1000, 3000, 4000	Год; Заказчик
Итоговое значение в нижнем колонтитуле	8000	Год

Связанные сведения

[Сведения о контекстах вычисления \[страница 13\]](#)

[Изменение контекста вычислений по умолчанию с расширенным синтаксисом \[страница 22\]](#)

1.4.1.3.1 Контексты по умолчанию в вертикальной таблице

Вертикальная таблица – это стандартная таблица отчета с верхними колонтитулами сверху, данными, которые идут сверху вниз, и нижними колонтитулами внизу.

Контексты по умолчанию в нижней таблице.

Положение вычисления	Контекст ввода	Контекст вывода
Верхний колонтитул	Измерения и меры используются для генерирования содержимого блока.	Все данные агрегированы, функция вычисления возвращает единственное значение.
Содержимое блока	Измерения и меры используются для генерирования текущей строки.	Идентично контексту ввода.
Нижний колонтитул	Измерения и меры используются для генерирования содержимого блока.	Все данные агрегированы, функция вычисления возвращает единственное значение.

1.4.1.3.2 Контексты по умолчанию в горизонтальной таблице

Контексты по умолчанию для горизонтальной таблицы выглядят так же, как и для вертикальной.

Горизонтальная таблица выглядит как вертикальная, повернутая на 90 градусов.

Внешний вид горизонтальной таблицы зависит от предпочтительного языкового стандарта для просмотра, выбранного в настройках стартовой панели BI. Для некоторых языковых стандартов (например, для английского) используется интерфейс с написанием слева направо (LTR), а для других (например, для арабского) – интерфейс с написанием справа налево (RTL).

В региональном стандарте с написанием слева направо верхние колонтитулы будут отображаться слева, данные будут выводиться слева направо, а нижние колонтитулы будут отображаться справа. В региональном стандарте с написанием справа налево верхние колонтитулы будут отображаться справа, данные будут выводиться справа налево, а нижние колонтитулы будут отображаться слева.

1.4.1.3.3 Контексты по умолчанию в кросс-таблице

В кросс-таблице отображаются данные в виде матрицы с мерами, которые отображаются на пересечении измерений.

Контексты по умолчанию в кросс-таблице.

Положение вычисления	Контекст ввода	Контекст вывода
Верхний колонтитул	Измерения и меры используются для генерирования содержимого блока.	Все данные агрегируются, затем функция вычисления возвращает единственное значение.
Содержимое блока	Измерения и меры используются для генерирования содержимого блока.	Идентично контексту ввода.

Положение вычисления	Контекст ввода	Контекст вывода
Нижний колонтитул	Измерения и меры используются для генерирования содержимого блока.	Все данные агрегируются, затем функция вычисления возвращает единственное значение.
Нижний колонтитул VBody	Измерения и меры используются для генерирования текущего столбца.	Все данные агрегируются, затем функция вычисления возвращает единственное значение.
Нижний колонтитул HBody	Измерения и меры используются для генерирования текущей строки.	Все данные агрегируются, затем функция вычисления возвращает единственное значение.
VFooter	Идентично нижнему колонтитулу.	Все данные агрегируются, затем функция вычисления возвращает единственное значение.
HFooter	Идентично нижнему колонтитулу.	Все данные агрегируются, затем функция вычисления возвращает единственное значение.

Пример: Контексты по умолчанию в кросс-таблице

В следующем отчете отображены контексты по умолчанию в кросс-таблице.

		FY2000	FY2000	FY2000	FY2000	1 115 730
		Q1	Q2	Q3	Q4	1 115 730
France	259 170	61 895	76 555	70 080	50 640	259 170
US	856 560	196 831	189 886	234 574	235 269	856 560
Сумма:	1 115 730	258 726	266 441	304 654	285 909	1 115 730

1.4.1.3.4 Контексты по умолчанию в разделе

Раздел состоит из верхнего колонтитула, текста сообщения и нижнего колонтитула.

Контексты по умолчанию в разделе:

Положение вычисления	Контекст ввода	Контекст вывода
Body	Измерения и меры в отчете отфильтрованы, чтобы ограничить данные до данных раздела.	Все данные агрегируются, затем функция вычисления возвращает единственное значение.

Пример: Контексты по умолчанию в разделе

В следующем отчете отображены контексты по умолчанию в разделе.

2001		8 096 123,60
Quarter	Sales revenue	
Q1	\$2 660 700	8 096 123,60
Q2	\$2 279 003	8 096 123,60
Q3	\$1 367 841	8 096 123,60
Q4	\$1 788 580	8 096 123,60
Сумма:	\$8 096 123,60	
2002		13 232 246
Quarter	Sales revenue	
Q1	\$3 326 172	13 232 246,00
Q2	\$2 840 651	13 232 246,00
Q3	\$2 879 303	13 232 246,00
Q4	\$4 186 120	13 232 246,00
Сумма:	\$13 232 246,00	
2003		15 059 142,80
Quarter	Sales revenue	
Q1	\$3 742 989	15 059 142,80
Q2	\$4 006 718	15 059 142,80
Q3	\$3 953 395	15 059 142,80
Q4	\$3 356 041	15 059 142,80
Сумма:	\$15 059 142,80	

1.4.1.3.5 Контексты по умолчанию в разбиении

Разбиение состоит из верхнего колонтитула, текста сообщения и нижнего колонтитула.

Контексты по умолчанию в разбиении:

Положение вычисления	Контекст ввода	Контекст вывода
Верхний колонтитул	Текущий экземпляр разбиения.	Все данные агрегируются, затем функция вычисления возвращает единственное значение.
Нижний колонтитул	Текущий экземпляр разбиения.	Все данные агрегируются, затем функция вычисления возвращает единственное значение.

Пример: Контексты по умолчанию в разбиении

В следующем отчете отображены контексты по умолчанию в разбиении:

Year	Quarter	\$8096123
	Q1	\$2 660 700
	Q2	\$2 279 003
	Q3	\$1 367 841
	Q4	\$1 788 580
2001		
	Сумма:	\$8 096 124

Year	Quarter	\$13232246
	Q1	\$3 326 172
	Q2	\$2 840 651
	Q3	\$2 879 303
	Q4	\$4 186 120
2002		
	Сумма:	\$13 232 246

1.4.1.4 Изменение контекста вычислений по умолчанию с расширенным синтаксисом

В расширенном синтаксисе используются операторы контекста, которые добавляются к формуле или мере для указания контекста вычислений.

Контекст меры или формулы состоит из контекста ввода и вывода.

Связанные сведения

[Ключевые слова расширенного синтаксиса \[страница 291\]](#)

[Операторы расширенного синтаксиса \[страница 23\]](#)

1.4.1.4.1 Операторы расширенного синтаксиса

Контексты ввода и вывода задаются явным образом при помощи операторов контекста.

В следующей таблице перечислены операторы контекста:

Оператор	Описание
In	Определяет подробный список измерений для использования в контексте.
ForEach	Добавляет измерения в контекст по умолчанию
ForAll	Удаляет измерения из контекста по умолчанию

Операторы ForAll и ForEach могут оказаться полезными, когда в контексте по умолчанию содержится множество измерений. Часто может быть проще добавлять или удалять измерения из контекста с помощью операторов ForAll и ForEach, чем создавать список с использованием In.

1.4.1.4.1.1 Оператор In Context

Оператор In Context определяет измерения в контексте в явной форме.

Пример: Использование оператора In Context для определения измерений в контексте

В этом примере у нас есть отчет, в котором отображается "Год" и "Доход от продаж". В поставщике данных также содержится объект "Квартал", однако это измерение в блок не включено. Вместо этого, требуется включить дополнительный столбец для отображения максимального значения дохода за каждый квартал каждого года. Отчет выглядит следующим образом:

Год	Доход от продаж	Максимальный квартальный доход
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

Чтобы узнать, откуда берутся значения в столбце "Максимальный доход за квартал", просмотрите этот блок совместно с блоком, содержащим измерение "Квартал":

Год	Квартал	Доход от продаж
2001	K1	\$2,660,699.50
2001	K2	\$2,279,003.00
2001	K3	\$1,367,841.00
2001	K4	\$1,788,580.00
	Максимум:	\$2,660,699.50

Год	Квартал	Доход от продаж
	K1	\$3,326,172.00
	K2	\$2,840,651.00
	K3	\$2,879,303.00
	K4	\$4,186,120.00
	Максимум:	\$4,186,120.00

Год	Квартал	Доход от продаж
	K1	\$3,742,989.00
	K2	\$4,006,717.50
	K3	\$3,953,395.00
	K4	\$3,356,041.00
	Максимум:	\$4,006,717.50

Столбец "Максимальный доход за квартал" содержит максимальные показатели дохода за каждый год. Например, K4 содержит значение максимального дохода в 2002 году, таким образом, "Максимальный доход за квартал" в строке 2002 года отобразит K4.

Использование оператора In, формула для "Максимального дохода за квартал"

```
Max ([Доход от продаж] In ([Год];[Квартал])) In ([Год])
```

Эта формула рассчитывает максимальный доход от продаж для каждого из сочетаний (Год, Квартал), а затем выводит полученное значение по годам.

📌 Примечание

Так как по умолчанию контекстом для вывода блока служит "Год", особым образом указывать контекст вывода в этой формуле не требуется.

1.4.1.4.1.2 Контекстный оператор ForEach

Оператор ForEach добавляет в контекст измерения.

Пример: Использование оператора ForEach для добавления измерений в контекст

В следующей таблице показан максимальный доход по каждому кварталу в отчете, который содержит измерение "Квартал", не включенное в блок:

Год	Доход от продаж	Максимальный квартальный доход
2001	8096123.60	2660699.50
2002	13232246.00	4186120.00
2003	15059142.80	4006717.50

Для столбца "Максимальный доход за квартал" можно создать формулу, которая не содержит оператор ForEach:

```
Max ([Доход от продаж] In ([Год];[Квартал])) In ([Год])
```

Контекстный оператор "ForEach" позволяет добиться того же результата с помощью следующей формулы:

```
Max ([Доход от продаж] ForEach ([Квартал])) In ([Год])
```

Причина: Измерение "Год" является в блоке контекстом ввода по умолчанию. При использовании оператора "ForEach" измерение "Квартал" добавляется в контекст, в результате чего получается контекст ввода ([Год];[Квартал]).

1.4.1.4.1.3 Контекстный оператор ForAll

Контекстный оператор ForAll удаляет измерения из контекста.

Пример: Использование оператора ForAll для удаления измерений из контекста

Есть отчет, в котором отображается год, квартал и доход от продаж, и необходимо добавить столбец, в котором отображается суммарный доход по каждому году, как показано в следующем блоке:

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

Для суммирования доходов по годам контекстом ввода должен быть (Year); по умолчанию это (Year; Quarter). Поэтому необходимо удалить из контекста ввода квартал, указав в формуле ForAll ([Quarter]), что будет выглядеть следующим образом:

```
Sum([Sales revenue] ForAll ([Quarter]))
```

Следует отметить, что для получения такого же результата можно использовать оператор In. В этом случае формула будет выглядеть следующим образом:

```
Sum([Sales revenue] In ([Year]))
```

В данной версии формулы в качестве контекста явно указывается год, а не удаляется квартал, чтобы оставить год.

1.5 Расчет значений с помощью интеллектуальных показателей

1.5.1 Расчет значений с помощью интеллектуальных мер

Интеллектуальные меры – это меры, чьи значения рассчитываются базой данных (реляционной или OLAP), которую использует юниверс.

Этим они отличаются от классических мер, которые вычисляются из детализированных значений, возвращаемых базой данных. Способы агрегирования данных, возвращаемых интеллектуальными мерами, не поддерживаются на внутреннем уровне компонентом Web Intelligence платформы SAP BusinessObjects Business Intelligence.

Запросы, содержащие интеллектуальные меры, вычисляют их во всех контекстах, необходимых отчету. Эти контексты могут меняться по мере изменения отчета. В результате запрос изменяется при каждом обновлении данных после изменения нужных контекстов.

При редактировании такого отчета в него автоматически вставляется сообщение #TOREFRESH с напоминанием о необходимости обновления отчета для отражения изменений. Отчет можно обновлять автоматически, выбрав параметр "Автоматически обновить документ" в диалоговом окне "Свойства документа".

📘 Примечание

Делегирование меры осуществляется статически и определяется на базе определения отчета на этапе разработки. В некоторых случаях (формула на основе `if [choice]= 1 then [dimension 1] else [dimension 2]`) контекст измерения во время выполнения имеет переменные характеристики. В таких случаях система не может делегировать вычисление меры и возвращает пустое значение.

Интеллектуальные меры ведут себя не так, как классические, поддерживающие основной набор функций агрегирования (Max, Min, Count, Sum, Average), которые можно вычислить во всех контекстах без участия базы данных. Например, при создании запроса, содержащего измерения [Страна] и [Регион] и мера [Доход] (которая вычисляет сумму дохода), исходно в блоке отображается страна, регион и доход. Если из блока удалить "Регион", итоговый доход для каждой страны все еще можно вычислить без обновления данных суммированием доходов для всех регионов страны. В этой ситуации интеллектуальная мера потребует обновления данных.

Контексты вычисления в созданном запросе представляют наборы группировок.

1.5.1.1 Наборы группировок и интеллектуальные меры

Набор группировок — это набор измерений, создающий результаты для меры.

Созданный SQL, возвращающий данные в интеллектуальную меру, содержит наборы группировок для всех уровней агрегирования данной меры, включенных в отчет.

Пример: Наборы группировок в запросе

Запрос содержит измерения [Страна], [Регион], [Город] и интеллектуальная мера [Доход]. Эти объекты предполагают, что для вычисления дохода во всех возможных контекстах будут использованы следующие наборы группировок:

- Итоговое значение интеллектуальной меры
- значение интеллектуальной меры по (Страна, Регион, Город)
- значение интеллектуальной меры по (Страна, Город)
- значение интеллектуальной меры по (Город)
- значение интеллектуальной меры по (Регион, Город)
- значение интеллектуальной меры по (Регион)
- значение интеллектуальной меры по (Страна, Регион)

- значение интеллектуальной меры по (Страна)

Если база данных поддерживает оператор UNION, в созданном SQL каждый набор группировок будет представлен в предложении UNION.

Наборы группировок обновляются с учетом необходимых отчету контекстов вычисления, которые могут измениться в ответ на изменения в структуре отчета.

1.5.1.1.1 Управление наборами группировок

При первом создании и выполнении запроса, включая интеллектуальные меры, созданный SQL содержит набор группировок, необходимый для вычисления интеллектуальных мер на самом детализированном уровне, подразумеваемом объектами запроса.

Например, при создании запроса, содержащего измерения [Страна], [Регион] и [Город] и интеллектуальную меру [Доход], в созданный SQL включается набор группировок (Страна, Регион, Город). Самый детализированный набор группировок включается в SQL всегда. Добавление и удаление других наборов группировок происходит в соответствии с изменениями в отчете.

При удалении измерения [Город] из блока потребуется набор группировок (Страна, Регион), чтобы возвращать значения дохода. Данный набор группировок еще не доступен в SQL-запросе, поэтому в ячейках [Доход] отображается #TOREFRESH. После обновления данных #TOREFRESH заменяется значениями дохода.

Затем при замене измерения [Город] в блоке набор группировок (Страна, Регион) больше не потребуется. Он удаляется из SQL-запроса и при следующем обновлении данных его значения отбрасываются.

Каждый раз при обновлении данных отчета наборы группировок включаются или отбрасываются в соответствии с контекстами вычисления, необходимыми отчету.

В некоторых ситуациях отобразить значение интеллектуальной меры невозможно. В этом случае в ячейках меры отображается значение #UNAVAILABLE.

1.5.1.2 Интеллектуальные меры и область анализа

При создании запроса с областью анализа исходный набор группировок содержит объекты результатов, но не объекты области.

Запрос не создает все возможные наборы группировок из комбинации объектов результатов и объектов области.

Пример: Запрос с областью анализа и интеллектуальной мерой

Запрос содержит объекты результата [Страна] и [Доход]. В область анализа включены измерения [Регион] и [Город]. При выполнении запроса его SQL-сценарий содержит набор группировок (Страна) и отображает в блоке объекты [Страна] и [Доход].

1.5.1.3 Интеллектуальные меры и язык SQL

1.5.1.3.1 Наборы группировок и оператор UNION

Некоторые базы данных поддерживают наборы группировок явно, используя оператор `GROUPING SETS`.

При формировании запроса, содержащего интеллектуальные меры, созданный SQL-сценарий использует множественные результирующие наборы и оператор `UNION` для моделирования эффекта `GROUPING SETS`.

Пример: Наборы группировок, извлеченные с помощью оператора UNION

В данном примере описывается запрос, содержащий измерения [Страна], [Регион], [Город] и интеллектуальную меру [Доход].

❗ Примечание

Для простоты интеллектуальная мера вычисляет сумму. На практике для такого агрегирования интеллектуальная мера не требуется, поскольку функция суммирования встроена в Web Intelligence.

При первом выполнении запроса набор группировок – это (Страна, Регион, Город). SQL-запрос в целом возвращает этот набор группировок, и необходимости в операторе `UNION` в SQL нет.

Если удалить из таблицы измерение [Город], для представления дохода (отображается как #TOREFRESH) потребуется набор группировок (Страна, Регион). После обновления данных SQL выглядит следующим образом:

```
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

```

Каждый набор группировок представлен оператором `SELECT` и имеет свой собственный идентификатор (столбец `GID`). В наборы группировок, которые не содержат полного набора измерений, включаются пустые столбцы (`SELECT ' '`), поскольку у каждого оператора `SELECT` в запросе, в том числе `UNION`, должно быть одинаковое число столбцов.

Если в отчет добавляется новый блок, содержащий [Страна] и [Доход], потребуется набор группировок (Страна). Теперь созданный SQL содержит три следующих набора группировок:

```

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

```

```
country.country_name,  
region.region_name,  
city.city_name
```

1.5.1.4 Интеллектуальные меры и формулы

1.5.1.4.1 Интеллектуальные меры и измерения, содержащие формулы

Если формула или переменная отображаются в контексте вычисления интеллектуальной меры как измерение и формула определяет необходимый мере набор группировок, значения для интеллектуальной меры не могут быть отображены.

Например, интеллектуальные меры и измерения теперь возвращают значения для следующих элементов:

- URL, созданный при помощи мастера гиперссылок.
- Простое соединение измерения (удаление пустых символов).
- Если используется FormatDate для [date]

📌 Примечание

Сообщение #UNAVAILABLE по-прежнему возвращается для следующих функций: ForEach, ForAll, In, Where, Rank, Previous, RelativeValue, RelativeDate, TimeDim, а также функций агрегирования, если в формулах используются функции Min, Max, Last или First: if ([selection] =1) then [dim1] else [dim2]

1.5.1.4.2 Интеллектуальные меры в формулах

Интеллектуальные меры, включенные в формулы, могут возвращать значения, даже если формуле нужен контекст вычисления, отличный контекста, подразумеваемого положением формулы.

Пусть, например, отчет содержит следующий блок:

Страна	Регион	Доход
США	Север	10000
США	Юг	15000
США	Восток	14000
США	Запад	12000

Если в эту таблицу добавить дополнительный столбец с формулой

```
[Доход] ForAll ([Регион])
```

исходным значением столбца будет #TOREFRESH, поскольку формуле, исключающей регионы из вычисления, нужен набор группировок (Страна). Обновление данных добавляет в запрос набор группировок (Страна) и отображает значения меры.

Связанные сведения

[Контекстный оператор ForAll \[страница 25\]](#)

1.5.1.5 Интеллектуальные меры и фильтры

1.5.1.5.1 Ограничение связано с интеллектуальными мерами и фильтрами

Интеллектуальная мера может оцениваться в теле таблицы, когда в таблице нет фильтра, или в родительском контексте (фильтр отчета).

В следующей таблице описано, как оцениваются интеллектуальные меры при наличии фильтров.

Принцип оценки интеллектуальных мер при наличии фильтра в отчете

Если фильтр включен ...	Интеллектуальная мера оценивается следующим образом
Мера	Интеллектуальная мера будет оценена правильно, но некоторые строки будут удалены из таблицы.
Измерение, которое уже является частью оси таблицы	Интеллектуальная мера будет оценена правильно, но некоторые строки будут удалены из таблицы. Интеллектуальная мера может быть оценена, поскольку после фильтрации не было агрегирования.
Измерение, которое не является частью оси таблицы, и когда операнд фильтра имеет одно значение (фильтр будет возвращать одно значение/строку).	Интеллектуальная мера будет оценена правильно. Интеллектуальная мера может быть оценена, поскольку после фильтрации не было агрегирования.
Измерение, которое не является частью оси таблицы, и если операнд фильтра имеет несколько значений (фильтр может возвращать несколько значений/строк).	Интеллектуальная мера не может быть оценена (отображается #UNAVAILABLE), так как в этом случае перед агрегированием выполняется фильтрация и для одной строки таблицы требуется агрегирование.

1.5.1.5.2 Интеллектуальные меры и фильтры для измерений

Если многозначный фильтр применяется к измерению, от которого зависит интеллектуальное значение, но измерение не отображается явно в контексте вычисления меры, интеллектуальная мера не может вернуть значение, и в ячейке отображается #UNAVAILABLE.

Такое поведение также наблюдается в тех случаях, когда фильтр отчета определяется элементом управления вводом.

#UNAVAILABLE отображается, поскольку мера должна быть отфильтрована в запросе, а затем агрегирована, однако после применения фильтра уровня отчета интеллектуальная мера не может быть агрегирована. Расчет меры возможен при добавлении фильтра запроса в созданный SQL, но при таком решении есть риск повлиять на другие отчеты, созданные на основе того же запроса.

📘 Примечание

Многозначный фильтр фильтрует множественные значения, используя такие операторы, как "Больше", "В списке" или "Меньше". Можно применять однозначные фильтры, например "Равно", не создавая при этом ошибку #UNAVAILABLE.

📘 Примечание

Если агрегирование не требуется, то возможен обходной путь. Определите формулу в качестве переменной, квалификацией которой является мера, убедитесь, что используемое измерение добавлено в блок переменной (для оптимизации отображения этот столбец можно скрыть).

Пример: Интеллектуальная мера и фильтр для измерения

Запрос содержит измерения "Страна" и "Продукт", а также интеллектуальная мера "Доход". "Страна" и "Доход" отображаются в блоке. Если применяется фильтр отчета, ограничивающий значения атрибута "Продукт" строками "Платья" или "Куртки", в ячейках "Доход" отображается #UNAVAILABLE.

Страна	Доход
Франция	#UNAVAILABLE
США	#UNAVAILABLE
Сумма:	#UNAVAILABLE

Значения отображаются, если ограничить значения атрибута "Продукт" только строкой "Куртки".

Страна	Доход
США	971 444
Сумма:	971 444

📘 Примечание

Многозначный фильтр по измерению "Страна" определен в таблице и возвращает значения.

1.5.1.5.3 Фильтрация интеллектуальных мер

В нижнем колонтитуле таблицы должно отображаться агрегированное значение данных таблицы.

Если к данным таблицы применяется локальный фильтр, система не может вернуть делегированное агрегированное значение отфильтрованных таким образом данных.

Пример: Фильтрация интеллектуальной меры

Страна	OrderAmountDel
Бразилия	28,833.36
Китай	51,384.33
Франция	68,630.22
США	3,529,511.14
Итого:	3,678,359.05
Сумма:	3,678,359.05

При фильтрации данных следующей таблицы по условию OrderAmountDel > 60,000

В таблице отображаются строки, для которых значение OrderAmountDel в соответствующем контексте (по стране) превышает 60,000.

В нижнем колонтитуле отображается сумма видимых строк.

В строке итогов возвращается сообщение #UNAVAILABLE, поскольку при вычислении агрегирование делегируется на серверную часть, что невозможно из-за локальной фильтрации в Web Intelligence.

Страна	OrderAmountDel
Франция	68,630.22
США	3,529,511.14
Итого:	#UNAVAILABLE
Сумма:	3,598,141.36

1.5.1.5.4 Интеллектуальные меры и фильтры детализации

Фильтр детализации является однозначным.

Детализацию можно выполнить непосредственно из панели детализации.

1.5.1.5.5 Интеллектуальные меры и вложенные фильтры OR

Вложенные фильтры OR, в которых хотя бы одно из отфильтрованных измерений не представлено в блоке, порождают ошибку #UNAVAILABLE для интеллектуальной меры в блоке.

Это связано с необходимостью выполнить локальное агрегирование интеллектуальной меры после локальной обработки (например, фильтрации или применения некоторых формул Web Intelligence).

1.6 Функции, операторы и ключевые слова

1.6.1 Функции

Функции формул подразделяются на несколько категорий.

📌 Примечание

На следующие языки функции не переводятся: китайский, японский, болгарский, польский, турецкий, тайский и русский. Они выводятся в интерфейсе по-английски.

Категория	Описание
Агрегирование	Объединяют данные (например, суммируя или усредняя набор значений)
Символьные	Оперируют со строками символов
Дата и время	Возвращают данные даты или времени
Документ	Возвращают данные о документе
Поставщик данных	Возвращает данные о поставщике данных документа
Логические	Возвращают значения True или False
Числовые	Возвращают числовые данные
Разные	Функции, которые не попадают ни в одну из указанных выше категорий
Набор	Возвращает наборы элементов из иерархий

1.6.1.1 Пользовательские форматы

Как отображаются числовые значения или значения даты/времени, можно определить с помощью пользовательских форматов.

В следующей таблице представлены маркеры, которые можно использовать для создания таких пользовательских форматов:

Маркер формата	Описание	Пример
#	Соответствующая цифра. Если количество символов в числе меньше, чем число символов формата #, дополнительные нули не подставляются.	"12345" в формате #, ##0 отображается как "12,345" (если в вашем языковом стандарте в качестве разделителя разрядов используется запятая) или "12 345" (если в вашем языковом стандарте в качестве разделителя разрядов используется пробел)
0	Соответствующая цифра. Если количество символов в числе меньше, чем число символов формата 0, перед числом ставятся дополнительные нули.	"123" в формате #0, 000 отображается как "0,123"
,	Разделитель разрядов определяется параметрами вашего языкового стандарта.	"1234567" в формате #, ##0 отображается как "1,234,567" (если в вашем языковом стандарте в качестве разделителя разрядов используется запятая) или "1 234 567" (если в вашем языковом стандарте в качестве разделителя разрядов используется неразрывный пробел)
.	Разделитель десятичных знаков определяется параметрами вашего языкового стандарта.	"12.34" в формате #.#0 отображается как "12.34" (если в вашем языковом стандарте части целого отделяются точкой) или "12,34" (если в вашем языковом стандарте части целого отделяются запятой)
[%] %	Отображает знак процента (%) после результата (результат при этом умножается на 100).	0,50 превращается в 50%.
%	Символ (%) после значения результата, не умножать результат на 100.	0,50 превращается в 0,50%
	Неразрывный пробел ()	"1234567" в формате # ##0 отображается как "1234 567"
1, 2, 3, а, Ъ, с, \$, £, € (и т. д.)	Буквенно-числовой символ.	"705.15" в формате \$#. #0 отображается как "\$705.15", а в формате #.#0 € – как "705,15 €"

ⓘ Примечание

Буквенно-цифровые символы следует заключать в одинарные кавычки, иначе они будут приняты за символы форматирования. Например, ## будет преобразовано в "123 4", а '# #' — в '# 1234'.

Маркер формата	Описание	Пример
COMPACT	Округление числового значения и его отображение с сокращенным суффиксом. Суффикс зависит от языкового стандарта.	-1234 в формате COMPACT отображается как -1K согласно языковому стандарту "Английский (США)".
LONG_COMPACT	Округление числового значения и его отображение с суффиксом. Суффикс отображается в полном имени и зависит от языкового стандарта.	-1234 в формате COMPACT отображается как -1 thousand согласно языковому стандарту "Английский (США)".
CURRENCY	Отображение значения в виде валюты и применение правил предпочтительного языкового стандарта для просмотра для денежных значений.	-1234 в формате CURRENCY отображается как -₽1,234.00 согласно языковому стандарту "Английский (США)".
ACCOUNTING_CURRENCY	Отображение значения в виде валюты и применение правил предпочтительного языкового стандарта для просмотра для значений бухгалтерского учета.	-1234 в формате ACCOUNTING_CURRENCY отображается как (₽1,234.00) согласно языковому стандарту "Английский (США)".
CURRENCY_COMPACT	Отображение значения в виде валюты с сокращенным суффиксом.	-1234 в формате CURRENCY_COMPACT отображается как -₽1K согласно языковому стандарту "Английский (США)".
[MIN_DEC:n]	Используется с форматами COMPACT, LONG_COMPACT и CURRENCY_COMPACT для определения минимального числа отображаемых десятичных разрядов. По умолчанию используется значение 0.	-1234 в формате COMPACT[MIN_DEC:5] отображается как -1.23400K согласно языковому стандарту "Английский (США)".
[MAX_DEC:n]	Используется с форматами COMPACT, LONG_COMPACT и CURRENCY_COMPACT для определения максимального числа отображаемых десятичных разрядов. По умолчанию значение равно MIN_DEC.	-1234 в формате COMPACT[MAX_DEC:2] отображается как -1.23K согласно языковому стандарту "Английский (США)".
[CURRENCY:c]	Используется с форматами CURRENCY, ACCOUNTING_CURRENCY и CURRENCY_COMPACT для определения символа валюты. По умолчанию используется значение ₽.	-1234 в формате CURRENCY отображается как -\$1,234.00 согласно языковому стандарту "Английский (США)".
[Red], [Blue], [Green], [Yellow], [Gray], [White], [Dark Red], [Dark Blue], [Dark Green]	Значение в указанном цвете.	"150" в формате #,##0[Red] отображается как "150" красным шрифтом, а в формате #,##0[Blue] – как "150" синим шрифтом.
Маркеры дня/даты	(day, date)	

Маркер формата	Описание	Пример
d	Количество дней месяца без дополнительных нулей. Если день даты состоит из одного символа, в начало добавляется один ноль.	Первый день месяца в формате d отображается как "1"
dd	Номер дня с начальными нулями. Если день даты состоит из одного символа, то ноль в начало не добавляется.	Первый день месяца в формате dd отображается как "01"
ddd	Сокращенное название дня недели. Если в выбранном языковом стандарте названия дней недели должны выделяться заглавной буквой, первая буква будет заглавной.	"Понедельник" в формате ddd отображается как "Пон". Во французском языке "lundi" отображается как "lun".
Dddd	Первая буква дня недели принудительно делается заглавной во всех языковых стандартах.	"Понедельник" в формате Dddd отображается как "Пон". Во французском языке "lundi" отображается как "Lun".
dddd	Название дня недели в полном формате. Если в выбранном языковом стандарте названия дней недели должны выделяться заглавной буквой, первая буква будет заглавной.	"Понедельник" в формате dddd отображается как "Понедельник". Во французском языке этот день отображается как "lundi".
DDDD	Полное название дня недели в верхнем регистре.	"Понедельник" в формате DDDD отображается как "ПОНЕДЕЛЬНИК". Во французском языке этот день отображается как "LUNDI".
dddd dd	После названия дня недели стоит пробел и номер дня.	"Понедельник" в формате dddd dd отображается как "Понедельник 01"
Маркеры календаря (week, month, year)		
M	Число месяцев без начальных нулей. Если номер месяца состоит из одного символа, то ноль в начало не добавляется.	"Январь" в формате M отображается как "1"
MM	Номер месяца с начальными нулями. Если номер месяца состоит из одного символа, то в начало добавляется ноль.	"Январь" в формате MM отображается как "01"
mmm	Сокращенное название месяца. Если в выбранном языковом стандарте названия должны выделяться заглавной буквой, первая буква будет заглавной.	"Январь" в формате mmm отображается как "Янв". Во французском языке этот месяц отображается как "jan".
Mmmm	Сокращенное название месяца. Первая буква делается заглавной во всех языковых стандартах.	"Январь" в формате mmm отображается как "Янв". Во французском языке этот месяц отображается как "Jan".

Маркер формата	Описание	Пример
mmmm	Название месяца в полном формате. Если в выбранном языковом стандарте названия должны выделяться заглавной буквой, первая буква будет заглавной.	"Январь" в формате mmmm отображается как "Январь". Во французском языке этот месяц отображается как "janvier".
MMMM	Название месяца в верхнем регистре.	"Январь" в формате MMMM отображается как "ЯНВАРЬ". Во французском языке этот месяц отображается как "JANVIER".
ww	Номер недели года.	9-е января 2015 г. в формате ww отображается как "02", поскольку приходится на седьмую неделю 2015 года.
w	Номер недели года без начального нуля.	9-е января 2015 г. в формате w отображается как "2", поскольку приходится на седьмую неделю 2015 года.
W	Номер недели месяца.	9-е января 2015 г. в формате W отображается как "2", поскольку приходится на вторую неделю января.
yy	Две последние цифры года.	"2003" в формате yy отображается как "03"
yyyy	Все четыре цифры года.	"2003" в формате yyyy отображается как "2003"
Маркеры времени		
	(hours, minutes, seconds, am/pm)	
hh:mm:ss a	Часы без дополнительных нулей, минуты и секунды с дополнительными нулями. Символ "a" отображает AM или PM после значения времени (по возможности).	"21:05:03" в формате hh:mm:ss a отображается как "9:05:03 PM" (английский языковой стандарт).
h	Час в 24-часовом формате времени, начиная с 0. Значения меньше 10 состоят из одной цифры.	"21:00" в формате h отображается как "21". Допустимые значения: от 0 до 23.
HH	Час в 24-часовом формате с отсчетом от 0.	"21:00" в формате HH отображается как "21". Допустимые значения: от 00 до 23.
k	Час в 24-часовом формате времени, начиная с 1. Значения меньше 10 состоят из одной цифры.	"21:00" в формате k отображается как "21". Допустимые значения: от 1 до 24.
kk	Час в 24-часовом формате с отсчетом от 01.	"21:00" в формате kk отображается как "21". Допустимые значения: от 01 до 24.
hh	Указание времени в 12-часовом формате.	"21:00" в формате hh отображается как "09".
HH:mm	Час и минуты, значения часа до 10 отображаются с нулем спереди.	"7:15" в формате HH:mm отображается как "07:15".

Маркер формата	Описание	Пример
HH:mm:ss	Час, минуты и секунды, значения часа до 10 отображаются с нулем спереди.	"7:15" в формате HH:mm:ss отображается как "7:15:00".
mm:ss	Минуты и секунды, значения часа до 10 отображаются с нулем спереди.	"7:15:03" в формате mm:ss отображается как "15:03".
x	Часовой пояс в формате "часы".	-08, +0530, +00
xx	Часовой пояс в формате "часы минуты".	-0800, +0530, +0000
xxx	Часовой пояс в формате "часы:минуты".	-08:00, +05:30, +00:00
xxxx	Часовой пояс в формате "часы минуты секунды".	-0800, +075228, +0000
xxxxxx	Часовой пояс в формате "часы:минуты:секунды".	-08:00, +07:52:28, +00:00
X	Аналогично x, за исключением того, что для часового пояса UTC отображается "Z".	-08, +0530, Z
XX	Аналогично xx, за исключением того, что для часового пояса UTC отображается "Z".	-0800, +0530, Z
XXX	Аналогично xxx, за исключением того, что для часового пояса UTC отображается "Z".	-08:00, +05:30, Z
XXXX	Аналогично xxxx, за исключением того, что для часового пояса UTC отображается "Z".	-0800, +075228, Z
XXXXXX	Аналогично xxxxxx, за исключением того, что для часового пояса UTC отображается "Z".	-08:00, +07:52:28, Z
VV	Ид. часового пояса.	America/Los_Angeles
O	Часовой пояс в формате "часы" относительно GMT.	GMT-8
OOOO	Часовой пояс в формате "часы и минуты" относительно GMT (заменяет устаревший формат "z").	GMT-08:00

Маркер формата	Описание	Пример
z	Название часового пояса. Если у часового пояса нет названия, в z отображается временная разница.	CEST или PST. Если у часового пояса есть название. Если нет названия, в z отображается временная разница следующим образом: +02, +530,...

ⓘ Примечание

Документы, которые созданы в версиях, предшествующих 4.3 и использующих формат z, преобразуются автоматически для отображения надлежащего результата в версии 4.3. Устаревший формат z интерпретируется как формат OOOO, приведенный в таблице.

[TIMEZONE:t]	Используется для указания часового пояса значения даты/времени (по умолчанию в Web Intelligence дата и время указаны в часовом поясе UTC). Поддерживаемые часовые пояса перечислены ниже.	Для 1 января 2015 г. 12:00:00 AM, HH' : 'mm' : 'ss[TIMEZONE:US / Eastern] z отображается как 19:00:00 EST
--------------	---	--

Список часовых поясов с маркером [TIMEZONE:t]:

Африка/Абиджан	Америка/Гранд-Терк	Азия/Багдад	Австралия/Перт	Европа/Ульяновск
Африка/Аккра	Америка/Гренада	Азия/Бахрейн	Австралия/Квинсленд	Европа/Ужгород
Африка/Аддис-Абеба	Америка/Гваделупа	Азия/Баку	Австралия/Юг	Европа/Вадуц
Африка/Алжир	Америка/Гватемала	Азия/Бангкок	Австралия/Сидней	Европа/Ватикан
Африка/Асмара	Америка/Гуаякиль	Азия/Барнаул	Австралия/Тасмания	Европа/Вена
Африка/Асмэра	Америка/Гайана	Азия/Бейрут	Австралия/Виктория	Европа/Вильнюс
Африка/Бамако	Америка/Галифакс	Азия/Бишкек	Австралия/Запад	Европа/Волгоград
Африка/Банги	Америка/Гавана	Азия/Бруней	Австралия/Янковина	Европа/Варшава
Африка/Банжул	Америка/Эрмосильо	Азия/Калькутта	Бразилия/Акри	Европа/Загреб
Африка/Бисау	Америка/Индиана/Индианаполис	Азия/Чита	Бразилия/ДеНоронья	Европа/Запорожье
Африка/Блантайр	Америка/Индиана/Нокс	Азия/Чойбалсан	Бразилия/Восток	Европа/Цюрих
Африка/Браззавиль	Америка/Индиана/Маренго	Азия/Чунцин	Бразилия/Запад	Великобритания
Африка/Бужумбура	Америка/Индиана/Петербург	Азия/Чунцин	СЕТ	Великобритания: Ирландия

Африка/Каир	Америка/Индиана/ Телл-Сити	Азия/Коломбо	CST6CDT	GMT
Африка/Касабланка	Америка/Индиана/ Вевей	Азия/Дакка	Канада/Атлантиче- ский регион	GMT+0
Африка/Сеута	Америка/Индиана/ Винсеннес	Азия/Дамаск	Канада/Центральная Канада	GMT-0
Африка/Конакри	Америка/Индиана/ Уинамак	Азия/Дакка	Канада/Восточная Канада	GMT0
Африка/Дакар	Америка/Индианапо- лис	Азия/Дили	Канада/Горный ре- гион	Гринвич
Африка/Дар-эс-Са- лам	Америка/Инувик	Азия/Дубай	Канада/Ньюфаунд- ленд	HST
Африка/Джибути	Америка/Икалуит	Азия/Душанбе	Канада/Тихоокеан- ский регион	Гонконг
Африка/Дуала	Америка/Ямайка	Азия/Фамагуста	Канада/Саскачеван	Исландия
Африка/Эль-Аюн	Америка/Жужуй	Азия/Газа	Канада/Юкон	Индийский океан/ Антананариву
Африка/Фритаун	Америка/Джуно	Азия/Харбин	Чили/Континенталь- ная часть	Индийский океан/Ча- гос
Африка/Габороне	Америка/Кентукки/ Луисвилл	Азия/Хеврон	Чили/Остров Пасхи	Регион Индийского океана/Остров Ро- ждества
Африка/Хараре	Америка/Кентукки/ Монтичелло	Азия/Хошимин	Куба	Индийский океан/Ко- косовые острова
Африка/Йоханнес- бург	Америка/Нокс, Ин- диана	Азия/Гонконг	EET	Регион Индийского океана/Коморские острова
Африка/Джуба	Америка/Кралендейк	Азия/Ховд	EST	Индийский океан/Маэ
Африка/Кампала	Америка/Ла-Пас	Азия/Иркутск	EST5EDT	Индийский океан/ Мальдивы
Африка/Хартум	Америка/Лима	Азия/Стамбул	Египет	Индийский океан/ Маврикий
Африка/Кигали	Америка/Пос-Андже- лес	Азия/Джакарта	Ирландия	Индийский океан/ Майотта
Африка/Киншаса	Америка/Луисвилл	Азия/Джаяпура	Etc/GMT	Индийский океан/ Реюньон
Африка/Лагос	Америка/Лоуэр- Принс	Азия/Иерусалим	Etc/GMT+0	Иран
Африка/Либревиль	Америка/Масейо	Азия/Кабул	Etc/GMT+1	Израиль
Африка/Поме	Америка/Манагуа	Азия/Камчатка	Etc/GMT+10	Ямайка
Африка/Луанда	Америка/Манаус	Азия/Карачи	Etc/GMT+11	Япония
Африка/Лубумбаши	Америка/Мариго	Азия/Кашгар	Etc/GMT+12	Кваджалейн

Африка/Лусака	Америка/Мартиника	Азия/Катманду	Etc/GMT+2	Ливия
Африка/Малабо	Америка/Матаморос	Азия/Катманду	Etc/GMT+3	MET
Африка/Мапуту	Америка/Масатлан	Азия/Хандыга	Etc/GMT+4	MST
Африка/Масеру	Америка/Мендоса	Азия/Калькутта	Etc/GMT+5	MST7MDT
Африка/Мбабане	Америка/Меномини	Азия/Красноярск	Etc/GMT+6	Мексика/Северная Нижняя Калифорния
Африка/Мобадишо	Америка/Мерида	Азия/Куала-Лумпур	Etc/GMT+7	Мексика/Южная Нижняя Калифорния
Африка/Монровия	Америка/Метлакатла	Азия/Кучинг	Etc/GMT+8	Мексика/общее
Африка/Найроби	Америка/Мехико	Азия/Кувейт	Etc/GMT+9	Новая Зеландия
Африка/Нджамена	Америка/Микелон	Азия/Макао	Etc/GMT-0	NZ-CHAT
Африка/Ниамей	Америка/Монктон	Азия/Макао	Etc/GMT-1	Навахо
Африка/Нуакшот	Америка/Монтеррей	Азия/Магадан	Etc/GMT-10	КНР
Африка/Уагадугу	Америка/Монтеви- део	Азия/Макассар	Etc/GMT-11	PST8PDT
Африка/Порто-Ново	Америка/Монреаль	Азия/Манила	Etc/GMT-12	Тихий океан/Апиа
Африка/Сан-Томе	Америка/Монтсеррат	Азия/Маскат	Etc/GMT-13	Тихий океан/Окленд
Африка/Тимбукту	Америка/Нассау	Азия/Никосия	Etc/GMT-14	Тихий океан/Буген- виль
Африка/Триполи	Америка/Нью-Йорк	Азия/Новокузнецк	Etc/GMT-2	Тихий океан/Чатем
Африка/Тунис	Америка/Нипигон	Азия/Новосибирск	Etc/GMT-3	Тихий океан/Чуук
Африка/Виндхук	Америка/Ном	Азия/Омск	Etc/GMT-4	Тихий океан/Пасха
Америка/Адак	Америка/Норонья	Азия/Уральск	Etc/GMT-5	Тихий океан/Эфате
Америка/Анкоридж	Америка/Северная Дакота/Бойла	Азия/Пномпень	Etc/GMT-6	Тихий океан/Эндер- бери
Америка/Ангилья	Америка/Северная Дакота/Сентер	Азия/Понтианак	Etc/GMT-7	Тихий океан/Фа- каофо
Америка/Антигуа	Америка/Северная Дакота/Нью-Салем	Азия/Пхеньян	Etc/GMT-8	Тихий океан/Фиджи
Америка/Арагуаина	Америка/Охинага	Азия/Катар	Etc/GMT-9	Тихий океан/Фуна- фути
Америка/Аргентина/ Буэнос-Айрес	Америка/Панама	Азия/Костанай	Etc/GMT0	Тихий океан/Галапа- гос
Америка/Аргентина/ Катамарка	Америка/Пангниртун	Азия/Кызылорда	Etc/Гринвич	Тихий океан/Гамбье
Америка/Аргентина/ Комодоро-Ривадавия	Америка/Парама- рибо	Азия/Рангун 30 м	Etc/UCT	Тихий океан/Гуадал- канал
Америка/Аргентина/ Кордова	Америка/Феникс	Азия/Эр-Рияд	Etc/UTC	Тихий океан/Гуам

Америка/Аргентина/ Жужуй	Америка/Порт-о- Пренс	Азия/Сайгон	Еtc/универсальное время	Тихий океан/Гоно- лулу
Америка/Аргентина/ Ла-Риоха	Америка/Порт-оф- Спейн	Азия/Сахалин	Еtc/зулу	Тихий океан/Джон- стон
Америка/Аргентина/ Мендоса	Америка/Порту-Акри	Азия/Самарканд	Европа/Амстердам	Тихий океан/Кирити- мати
Америка/Аргентина/ Рио-Гальегос	Америка/Порту-Ве- лью	Азия/Сеул	Европа/Андорра	Тихий океан/Косрае
Америка/Аргентина/ Сальта	Америка/Пуэрто- Рико	Азия/Шанхай	Европа/Астрахань	Тихий океан/Квад- жалейн
Америка/Аргентина/ Сан-Хуан	Америка/Пунта-Аре- нас	Азия/Сингапур	Европа/Афины	Тихий океан/Мад- журо
Америка/Аргентина/ Сан-Луис	Америка/Рейни-Ри- вер	Азия/Среднеколымск	Европа/Белфаст	Тихий океан/Маркиз- ские острова
Америка/Аргентина/ Тукуман	Америка/Ранкин-Ин- лет	Азия/Тайбэй	Европа/Белград	Тихоокеанский ре- гион/Мидуэй
Америка/Аргентина/ Ушуая	Америка/Ресифи	Азия/Ташкент	Европа/Берлин	Тихий океан/Науру
Америка/Аруба	Америка/Реджайна	Азия/Тбилиси	Европа/Братислава	Тихий океан/Ниуэ
Америка/Асунсьон	Америка/Ресольют	Азия/Тегеран	Европа/Брюссель	Тихоокеанский ре- гион/Норфолк
Америка/Атикокан	Америка/Рио-Бранку	Азия/Тель-Авив	Европа/Бухарест	Тихий океан/Нумеа
Америка/Атка	Америка/Росарио	Азия/Тхимбу	Европа/Будапешт	Тихоокеанский ре- гион/Паго-Паго
Америка/Баия	Америка/Санта-Ис- абель	Азия/Тхимпху	Европа/Бюзинген	Тихий океан/Палау
Америка/Баия-Бан- дерас	Америка/Сантарен	Азия/Токио	Европа/Кишинев	Тихий океан/Питкэрн
Америка/Барбадос	Америка/Сантьяго	Азия/Томск	Европа/Копенгаген	Тихий океан/Понпеи
Америка/Белен	Америка/Санто-До- минго	Азия/Уджунг-Пан- данг	Европа/Дублин	Тихий океан/Понпеи
Америка/Белиз	Америка/Сан-Паулу	Азия/Улан-Батор	Европа/Гибралтар	Тихоокеанский ре- гион/Порт-Морсби
Америка/Блан-Са- блон	Америка/Скорсби- сунн	Азия/Улан-Батор	Европа/Гернси	Тихий океан/Раро- тонга
Америка/Боа-Виста	Америка/Шипрок	Азия/Урумчи	Европа/Хельсинки	Тихий океан/Сайпан
Америка/Богота	Америка/Ситка	Азия/Усть-Нера	Европа/Остров Мэн	Тихий океан/Самоа
Америка/Бойсе	Америка/Сен-Барте- леми	Азия/Вьентьян	Европа/Стамбул	Тихий океан/Таити
Америка/Буэнос-Ай- рес	Америка/Сент- Джонс	Азия/Владивосток	Европа/Джерси	Тихий океан/Тарава

Америка/Кембридж-Бей	Америка/Сент-Китс	Азия/Якутск	Европа/Калининград	Тихий океан/Тонгатапу
Америка/Кампу-Гранде	Америка/Сент-Люсия	Азия/Янгон	Европа/Киев	Тихий океан/Трук
Америка/Канкун	Америка/Сент-Томас	Азия/Екатеринбург	Европа/Киров	Тихий океан/Уэйк
Америка/Каракас	Америка/Сент-Винсент	Азия/Ереван	Европа/Лиссабон	Тихий океан/Уоллис
Америка/Катамарка	Америка/Свифт-Керрент	Атлантический регион/Азорские острова	Европа/Любляна	Тихий океан/Яп
Америка/Кайенна	Америка/Тегусигальпа	Атлантический океан/Бермудские острова	Европа/Лондон	Польша
Америка/Кайман	Америка/Туле	Атлантический океан/Канарские острова	Европа/Люксембург	Португалия
Америка/Чикаго	Америка/Тандер-Бей	Атлантический регион/Кабо-Верде	Европа/Мадрид	ROC
Америка/Чиуауа	Америка/Тихуана	Атлантический регион/Фарерские острова	Европа/Мальта	ROK
Америка/Корал-Харбор	Америка/Торонто	Атлантический океан/Фарерские острова	Европа/Мариехамн	Сингапур
Америка/Кордова	Америка/Тортола	Атлантический регион/Ян-Майен	Европа/Минск	Турция
Америка/Коста-Рика	Америка/Ванкувер	Атлантический регион/Мадейра	Европа/Монако	УСТ
Америка/Крестон	Америка/Вирджин	Атлантический регион/Рейкьявик	Европа/Москва	США/Аляска
Америка/Куяба	Америка/Уайтхорс	Атлантический океан/Южная Георгия	Европа/Никосия	США/Алеутские острова
Америка/Кюрасао	Америка/Виннипег	Атлантический регион/Остров Святой Елены	Европа/Осло	США/Аризона
Америка/Данмарк-схавн	Америка/Якутат	Атлантический океан/Стэнли	Европа/Париж	США/Центральная часть
Америка/Доусон	Америка/Йеллоунайф	Австралия/АСТ	Европа/Подгорица	США/Восточная Индия
Америка/Досон-Крик	Антарктика/Маккуори	Австралия/Аделаида	Европа/Прага	США/Восточная часть
Америка/Денвер	Антарктика/МакМердо	Австралия/Брисбен	Европа/Рига	США/Гавайи

Америка/Детройт	Антарктида/Палмер	Австралия/Брокен-Хилл	Европа/Рим	США/Индиана и Старк
Америка/Доминика	Антарктика/Южный полюс	Австралия/Канберра	Европа/Самара	США/Мичиган
Америка/Эдмонтон	Арктика/Лонгйир	Австралия/Карри	Европа/Сан-Марино	США/Горный регион
Америка/Эйрунепе	Азия/Аден	Австралия/Дарвин	Европа/Сараево	США/Тихоокеанский регион
Америка/Сальвадор	Азия/Алматы	Австралия/Евкла	Европа/Саратов	США/Тихоокеанский регион – новый
Америка/Энсенада	Азия/Амман	Австралия/Хобарт	Европа/Симферополь	США/Самоа
Америка/Форт-Нельсон	Азия/Анадырь	Австралия/ЛМС	Европа/Скопье	UTC
Америка/Форт-Уэйн	Азия/Актау	Австралия/Линдеман	Европа/София	Универсальный
Америка/Форталеза	Азия/Актобе	Австралия/Лорд-Хау	Европа/Стокгольм	W-SU
Америка/Глас Бэй	Азия/Ашхабад	Австралия/Мельбурн	Европа/Таллин	WET
Америка/Готхоб	Азия/Ашхабад	Австралия/Новый Южный Уэльс	Европа/Тирана	Зулу
Америка/Гуз-Бей	Азия/Атырау	Австралия/Север	Европа/Тирасполь	

1.6.1.2 Функции агрегирования

1.6.1.2.1 Агрегирование

Описание

Возвращает агрегирование меры по умолчанию для данного набора элементов

Группа функций

Агрегирование

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязателен
measure	Любая мера	Мера	Да
member_set	Набор элементов, используемый для вычисления агрегирования	Набор элементов	Нет

Примечания

- С функцией `Aggregate` можно использовать контекстные операторы расширенного синтаксиса.
- При включении `member_set` `Aggregate` возвращает агрегированное значение меры для всех элементов в наборе элементов.
- В `member_set` могут входить различные наборы, разделяемые точкой с запятой (;).
- Список наборов элементов должен быть заключен в {}.
- Если выражением набора элементов не указывается точный элемент или узел, в таблице должна присутствовать используемая в ссылках иерархия, то выражение набора элементов ссылается на текущий элемент в иерархии в таблице. Если этой иерархии нет в таблице, функция возвращает сообщение `#MULTIVALUE`.
- Агрегирование делегированных мер возвращает значение `#TOREFRESH`, если необходимое агрегирование недоступно в запросе. Чтобы получить доступ к новому уровню агрегирования, обновите документ. Например, это может понадобиться при использовании панели фильтров, если пользователь выбрал значение перед выбором фильтра "Все значения" или наоборот выбрал "Все значения" перед выбором значения.

Примеры

Если агрегирование по умолчанию меры [Доходы от продаж] представляет собой Sum, а [Калифорния] является элементом иерархии [География] (Страна > Штат > Город): `Aggregate([Доходы от продаж]; {Descendants([География]&[США].[Калифорния]; 1)})` возвращает итоговую выручку от продаж во всех городах Калифорнии.

Связанные сведения

[Ссылки на элементы и наборы элементов в иерархиях \[страница 297\]](#)

1.6.1.2.2 Выбор элементов в функциях агрегирования

Описание

Для некоторых функций агрегирования можно задать выбор элемента, чтобы определить контекст агрегирования, когда блок содержит иерархию.

Группа функций

Агрегирование

Синтаксис

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

Ввод

Параметр	Описание	Тип	Обязательный
AggregationFunction	Должна быть одной из нижеперечисленных: <ul style="list-style-type: none">AggregateAverageCountMaxMinSum	Функция Aggregate	Да
мой объект	Измерение или показатель	Измерение или показатель	Да
memberselection	Определенный элемент или набор элементов, вычисленный с помощью функции множества. Элемент memberselection должен быть заключен в фигурные скобки. Каждый элемент в наборе элементов отделяется точкой с запятой <pre>{[member one];[member two];CalculatedMemberSet() }</pre>	Элемент или набор элементов, вычисленный с помощью функции множества.	Да

Параметр	Описание	Тип	Обязательный
	<p>Где CalculatedMemberSet использует одну из функций множества:</p> <ul style="list-style-type: none"> • Ancestor • Descendant • Lag • Children • Parent • Siblings 		

Описание

Функции множества использует в качестве параметров объект, уровень или элемент. Если указан только объект, и это иерархический объект, присутствующий в блоке, то он будет использовать текущий элемент. Также можно указать определенный элемент с использованием следующего синтаксиса:

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

Для источников Microsoft и Essbase .UNIX можно выбрать уровень:

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

Примеры

Все следующие примеры взяты из источника данных на английском языке.

❖ Пример

В следующем примере нужно определить разницу в объеме интернет-продаж за 2002 и 2001 годы.

```
=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]})
```

Либо выберите два элемента при выборе элемента:

```
=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	

❖ Пример

В следующем примере имеется иерархия товаров, и нужно узнать объем интернет-продаж всех товаров, связанных с велосипедами. Но два из них находятся в различных ветвях:

```
=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

❖ Пример

В следующем примере нужно сравить объем интернет-продаж в странах Североамериканского континента, сначала сравнивая с Канадой и США, а затем с другими странами мира.

Для начала получим итог для стран Североамериканского континента; в данном случае нас интересуют Канада и США:

```
=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

Затем будем сравнивать все страны с Северной Америкой:

```
=([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

Мы видим, что общее число клиентов во всем мире в два с половиной раза превышает этот показатель для Северной Америки, и что общее число клиентов в Австралии составляет 80% от этого же показателя для Северной Америки.

Связанные сведения

[Агрегирование \[страница 46\]](#)

1.6.1.2.3 Average

Описание

Возвращает среднее значение меры

Группа функций

Агрегирование

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязателен
measure	Любая мера	Мера	Да
member_set	Набор элементов	Набор элементов	Нет
IncludeEmpty	Включать в вычисление пустые строки	Ключевое слово	Нет (Пустые строки по умолчанию исключаются)

Примечания

- С функцией `Average` можно использовать операторы контекста расширенного синтаксиса.
- Если указывается `набор_элементов`, функция `Average` возвращает среднее значение меры для всех элементов в наборе элементов.
- В `member_set` могут входить различные наборы, разделяемые точкой с запятой (;).
- Список наборов элементов должен быть заключен в {}.
- Если выражением набора элементов не указывается точный элемент или узел, в таблице должна присутствовать используемая в ссылках иерархия, то выражение набора элементов ссылается на текущий элемент в иерархии в таблице. Если этой иерархии нет в таблице, функция возвращает сообщение `#MULTIVALUE`.

- Агрегирование делегированных мер возвращает значение #TOREFRESH, если необходимое агрегирование недоступно в запросе. Чтобы получить доступ к новому уровню агрегирования, обновите документ. Например, это может понадобиться при использовании панели фильтров, если пользователь выбрал значение перед выбором фильтра "Все значения" или наоборот выбрал "Все значения" перед выбором значения.
- Делегированная мера, определенная относительно группы, использует локальное агрегирование (агрегирование значения меры по сгруппированным значениям) и в связи с этим возвращает значение #UNAVAILABLE.
Даже если выполнить принудительное локальное агрегирование такой меры с использованием формулы "if then else" или значения группы, по-прежнему будет возвращаться сообщение #MULTIVALUE.

Примеры

Если для меры [Доходы от продаж] заданы значения 41569, 30500, 40000 и 50138, функция `Average([Доходы от продаж])` возвращает 405552.

Если [Калифорния] является элементом иерархии [География] (Страна > Штат > Город), то `Average([Доходы от продаж]; {[География]&[США].[Калифорния].children})` возвращает средний доход от продаж для всех городов в Калифорнии.

Связанные сведения

[Ссылки на элементы и наборы элементов в иерархиях \[страница 297\]](#)

[Оператор IncludeEmpty \[страница 281\]](#)

1.6.1.2.4 Количество

Описание

Возвращает число значений в наборе значений

Группа функций

Агрегирование

Синтаксис

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

Входные данные

Параметр	Описание	Тип	Обязательный
aggregated_data	Любое измерение, мера, иерархия, уровень или набор элементов	Измерение, мера, иерархия, набор элементов	Да
member_set	Набор элементов, используемый для расчета числа	Набор элементов	Нет
IncludeEmpty	Включает пустые значения в вычисление	Ключевое слово	Нет
Distinct All	Включает только точные значения (по умолчанию для измерений) или все значения (по умолчанию для мер) в вычисление	Ключевое слово	Нет

Примечания

- Можно использовать операторы контекста расширенного синтаксиса с функцией Count.
- Если в качестве второго аргумента определено значение IncludeEmpty функция будет учитывать пустые (null) значения в вычислении.
- Если параметр Distinct|All не задан, то по умолчанию используются следующие значения: Distinct для измерений и All для мер.
- Если включен member_set, Count ограничивает число количеством значений в member_set.
- В member_set могут входить различные наборы, разделяемые точкой с запятой (;).
- Список наборов элементов должен быть заключен в {}.
- Если выражением набора элементов не указывается точный элемент или узел, в таблице должна присутствовать используемая в ссылках иерархия, то выражение набора элементов ссылается на текущий элемент в иерархии в таблице. Если этой иерархии нет в таблице, функция возвращает сообщение #MULTIVALUE.
- Агрегирование делегированных мер возвращает значение #TOREFRESH, если необходимое агрегирование недоступно в запросе. Чтобы получить доступ к новому уровню агрегирования, обновите документ. Например, это может понадобиться при использовании панели фильтров, если пользователь выбрал значение перед выбором фильтра *Все значения* или, наоборот, выбрал *Все значения* перед выбором значения.
- Делегированная мера, указанная для группы, возвращает значение #UNAVAILABLE, поскольку требует локального агрегирования (агрегирования значения меры сгруппированных значений).

Даже если выполнить принудительное локальное агрегирование такой меры с использованием формулы "if then else" или значения группы, по-прежнему будет возвращаться сообщение #MULTIVALUE.

📌 Примечание

В очень специфичных рабочих процессах объединенные значения объектов не агрегируются одинаково в версии XI 3.x и 4.x.

В версии XI 3.x агрегация значений объединенных элементов объектов в контексте этого объединенного объекта (с использованием формулы ForEach()) приводит к отфильтрованному списку агрегированных значений (таких, которые соответствуют объединенным значениям).

В версии 4.x тот же рабочий процесс приводит к полному списку агрегированных значений: фильтр не применяется.

Примеры

`Count ("Тест")` возвращает 1

`Count ([City];Distinct)` возвращает 5, если в списке городов есть 5 различных городов, даже если в списке более 5 строк из-за повторов.

`Count ([City];All)` возвращает 10, если в списке городов есть 10 городов, даже если некоторые из них повторяются.

`Count ([City];IncludeEmpty)` возвращает 6, если существует 5 городов и одна пустая строка в списке городов.

`Count ([Продукт];{[География]&[Штат]})` возвращает общее число продуктов на уровне [Штат] в иерархии [География].

Связанные сведения

[Оператор IncludeEmpty \[страница 281\]](#)

[Операторы Distinct/All \[страница 280\]](#)

1.6.1.2.5 First

Описание

Возвращает первое значение из множества данных

Группа функций

Агрегирование

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
dimension measure	Любое измерение или мера	Измерение или мера	Да

Примечания

- При нахождении в нижнем колонтитуле разрыва функция `First` возвращает первое значение в разрыве.
- При нахождении в нижнем колонтитуле таблицы функция `First` возвращает первое значение в таблице.
- При нахождении в теле таблицы функция `First` возвращает непредсказуемый результат, который зависит от порядка набора данных в источнике.

Примеры

При размещении в нижнем колонтитуле таблицы функция `First ([Доход])` возвращает первое значение `[Доход]` в этой таблице.

1.6.1.2.6 Функция Interpolation

Описание

Вычисляет пустые значения мер путем интерполяции

Группа функций

Агрегирование

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
measure	Любая мера	Показатель	Да
PointToPoint Linear	Метод интерполяции <ul style="list-style-type: none">PointToPoint – двухточечная интерполяцияLinear – линейная регрессия с интерполяцией методом наименьших квадратов	Ключевое слово	Нет (PointToPoint – параметр по умолчанию)
NotOnBreak reset_dims	<ul style="list-style-type: none">NotOnBreak предотвращает сброс вычисления функции при нахождении разбиений блоков или разделов.reset_dims – список измерений, используемых для сброса интерполяции.	Ключевое слово список измерений	Нет
Row Col	Устанавливает направление вычисления	Ключевое слово	(Row параметр по умолчанию)

Примечания

- Функция `Interpolation` особенно необходима при создании линейного графика с использованием меры, в которой содержатся отсутствующие значения. При использовании этой

функции обеспечивается непрерывное построение графика без получения разорванных линий и точек.

- При использовании линейной регрессии с интерполяцией методом наименьших квадратов отсутствующие значения вычисляются с помощью линейного выражения $f(x) = ax + b$, которое позволяет получить линию, наиболее приближенную ко всем доступным значениям меры.
- При двухточечной интерполяции отсутствующие значения получаются путем вычисления значений линейной функции $f(x) = ax + b$, проходящей через две соседние с отсутствующими значениями точки.
- От порядка сортировки меры зависят значения, полученные с помощью функции `Interpolation`.
- Применение сортировки или ранжирования в формуле с функцией `Interpolation` недопустимо.
- В списке значений содержится только одно значение, и это значение используется в функции `Interpolation` для получения всех отсутствующих значений.
- Фильтры, примененные к интерполированной мере, могут повлиять на значения, выведенные функцией `Interpolation`, в зависимости от значений, к которым применяются фильтры.

Примеры

`Interpolation([Значение])` возвращает следующие отсутствующие значения при использовании метода интерполяции "точка-точка" по умолчанию:

День	Значение	Interpolation([Значение])
Понедельник	12	12
Вторник	14	14
Среда		15
Четверг	16	16
Пятница		17
Суббота		18
Воскресенье	19	19

Связанные сведения

[Оператор Linear \[страница 282\]](#)

[Оператор PointToPoint \[страница 283\]](#)

1.6.1.2.7 Last

Описание

Возвращает последнее значение из набора данных

Группа функций

Агрегирование

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
dimension measure	Любое измерение или мера	Измерение или мера	Да

Примечания

- При нахождении в нижнем колонтитуле таблицы функция Last возвращает последнее значение в разрыве.
- При нахождении в нижнем колонтитуле таблицы функция Last возвращает последнее значение в таблице.
- При нахождении в теле таблицы функция Last возвращает непредсказуемый результат, который зависит от порядка набора данных в источнике.
- По техническим причинам функция Last может возвращать значение Null, если в качестве входного параметра передается объединенный объект.

Примеры

При размещении в нижнем колонтитуле таблицы функция Last ([Revenue]) возвращает последнее значение [Revenue] в этой таблице.

1.6.1.2.8 Функция Max

Описание

Возвращает наибольшее значение из набора значений

Группа функций

Агрегирование

Синтаксис

```
input_type Max(aggreated_data[;member_set])
```

Данные ввода

Параметр	Описание	Тип	Обязателен
aggreated_data	Любое измерение, мера, иерархия, уровень или набор элементов	Измерение, мера, иерархия, уровень или набор элементов	Да
member_set	Набор элементов	Набор элементов	Нет

Примечания

- Можно использовать контекстные операторы расширенного синтаксиса с функцией max.
- При включении member_set функция max возвращает максимальное значение агрегированных данных для всех элементов в наборе элементов.
- В member_set могут входить различные наборы, разделяемые точкой с запятой (;).
- Список наборов элементов должен быть заключен в {}.
- Если выражением набора элементов не указывается точный элемент или узел, в таблице должна присутствовать используемая в ссылках иерархия, то выражение набора элементов ссылается на текущий элемент в иерархии в таблице. Если этой иерархии нет в таблице, функция возвращает сообщение #MULTIVALUE.
- Агрегирование делегированных мер возвращает значение #TOREFRESH, если необходимое агрегирование недоступно в запросе. Чтобы получить доступ к новому уровню агрегирования, обновите документ. Например, это может понадобиться при использовании панели фильтров, если

пользователь выбрал значение перед выбором фильтра "Все значения" или наоборот выбрал "Все значения" перед выбором значения.

- Делегированная мера, определенная относительно группы, использует локальное агрегирование (агрегирование значения меры по сгруппированным значениям) и, в связи с этим, возвращает значение #UNAVAILABLE.
Даже если выполнить принудительное локальное агрегирование такой меры с использованием формулы "if then else" или значения группы, по-прежнему будет возвращаться сообщение #MULTIVALUE.

Примеры

Если мера [Доход с продаж] содержит значения 3000, 60 034 и 901 234, функция `Max ([Доход от продаж])` возвращает 901 234.

Если измерение [Город] имеет значения "Абердин" и "Лондон", `Max ([Город])` возвращает "Лондон".

Если [США] является элементом иерархии [География] (Страна > Штат > Город), то функция `Max ([Доходы от продаж] ; { [География] . [США] . Children })` возвращает наибольший доход от продаж для штата в США.

1.6.1.2.9 Функция Median

Описание

Возвращает медиану (среднее число) меры

Группа функций

Агрегирование

Синтаксис

```
num Median(measure)
```

Данные ввода

Параметр	Описание	Тип	Обязателен
measure	Любая мера	Мера	Да

Примечания

Если в наборе чисел четное количество значений, `Median` берет среднее от двух значений из середины.

Примеры

`Median([Доход])` возвращает 971 444, если [Доход] имеет значения 835 420, 971 444 и 147 966.

1.6.1.2.10 Функция Min

Описание

Возвращает наименьшее значение из набора значений

Группа функций

Агрегирование

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
aggregated_data	Любое измерение, мера, иерархия, уровень или набор элементов	Измерение, мера, иерархия, уровень или набор элементов	Да
member_set	Набор элементов	Набор элементов	Нет

Примечания

- С функцией `min` можно использовать контекстные операторы расширенного синтаксиса.
- При включении `member_set` `min` возвращает минимальное значение агрегированных данных из всех элементов в наборе элементов.
- В `member_set` могут входить различные наборы, разделяемые точкой с запятой (;).
- Список наборов элементов должен быть заключен в {}.
- Если выражением набора элементов не указывается точный элемент или узел, в таблице должна присутствовать используемая в ссылках иерархия, то выражение набора элементов ссылается на текущий элемент в иерархии в таблице. Если этой иерархии нет в таблице, функция возвращает сообщение `#MULTIVALUE`.
- Агрегирование делегированных мер возвращает значение `#TOREFRESH`, если необходимое агрегирование недоступно в запросе. Чтобы получить доступ к новому уровню агрегирования, обновите документ. Например, это может понадобиться при использовании панели фильтров, если пользователь выбрал значение перед выбором фильтра "Все значения" или наоборот выбрал "Все значения" перед выбором значения.
- Делегированная мера, указанная для группы, возвращает сообщение `#UNAVAILABLE`, поскольку требует локального агрегирования (агрегирования значения меры для сгруппированных значений). Даже если выполнить принудительное локальное агрегирование такой меры с использованием формулы "if then else" или значения группы, по-прежнему будет возвращаться сообщение `#MULTIVALUE`.

Примеры

Если мера [Доход от продаж] содержит значения 3000, 60034 и 901234, функция `min([Доход от продаж])` возвращает 3000.

Если измерение [Город] содержит значения "Абердин" и "Лондон", функция `min([Город])` возвращает "Абердин".

`min([Доход от продаж]; {[География]&[США].children})` возвращает наименьший доход от продаж в штате США, если [США] является элементом иерархии [География] с уровнями [Страна] > [Штат] > [Город].

1.6.1.2.11 Функция Mode

Описание

Возвращает наиболее часто встречающееся значение в наборе данных

Группа функций

Агрегирование

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
dimension measure	Любое измерение или показатель	Показатель	Да

Примечания

- Функция mode возвращает пустое значение, если во множестве данных не содержится значения, встречающегося чаще других значений.

Примеры

Функция mode ([Доход]) возвращает значение 200, если объект [Доход] имеет значения 100, 200, 300, 200.

Функция mode ([Страна]) возвращает наиболее часто встречающееся значение в объекте [Страна].

1.6.1.2.12 Функция Percentage

Описание

Выражает значение меры в процентном соотношении его внедренного контекста

Группа функций

Агрегирование

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязателен
measure	Любая мера	Мера	Да
Break	Счета для разбиений таблиц	Ключевое слово	Нет
Row Col	Устанавливает направление вычисления	Ключевое слово	Нет

Примеры

В следующей таблице для столбца "Процентное соотношение" задана формула Percentage ([Доход от продаж])

Год	Доход с продаж	Процентное соотношение
2001	1000	10
2002	5000	50
2003	4000	40
Сумма:	10000	100

По умолчанию присвоенный контекст – это мера итогового значения в таблице. Можно задать в функции учет разбиения таблицы с использованием дополнительного аргумента Break. В этом случае внедренный контекст по умолчанию становится разделом таблицы.

В следующей таблице для столбца "Процентное соотношение" задана формула Percentage ([Доход от продаж] ; Break)

Год	Квартал	Доход с продаж	Процентное соотношение
2001	K1	1000	10
	K2	2000	20
	K3	5000	50
	K4	2000	20
2001	Сумма:	10000	100

Год	Квартал	Доход с продаж	Процентное соотношение
2002	K1	2000	20
	K2	2000	20
	K3	5000	50
	K4	1000	10
2002	Сумма:	10000	100

Функцию Percentage можно применять к строкам или столбцам; это можно сделать явно с помощью дополнительного аргумента Row|Col. Например, в следующей кросс-таблице столбец процентного соотношения описан формулой Percentage ([Доход с продаж] ; Row)

	K1	%	K2	%	K3	%	K4	%
2001	1000	10	2000	20	5000	50	2000	20
2002	2000	20	2000	20	5000	50	1000	10

1.6.1.2.13 Функция Percentile

Описание

Возвращает процентиль nth меры

Группа функций

Числовые

Синтаксис

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

Ввод

Параметр	Описание	Тип	Обязателен
measure	Любая мера	Мера	Да
percentile	Процентное значение, выраженное десятичной дробью	Число	Да

Примечания

N-й процентиль – это число, которое больше или равно n% чисел из набора. N% выражается в форме 0,n.

Примеры

Если [мера] содержит множество значений (10;20;30;40;50), то `Percentile([мера];0,3)` возвращает 22, что больше или равно 30% чисел из множества.

1.6.1.2.14 Функция Product

Описание

Умножает значения меры

Группа функций

Агрегирование

Синтаксис

```
num Product(measure)
```

Ввод

Параметр	Описание	Тип	Обязателен
measure	Любая мера	Мера	Да

Примеры

`Product ([Мера])` возвращает 30, если [Мера] имеет значения 2, 3, 5.

1.6.1.2.15 RunningAverage

Описание

Возвращает скользящее среднее значение меры

Группа функций

Агрегирование

Синтаксис

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

Чтобы сбросить `RunningAverage` для каждого раздела, рекомендуется использовать следующий синтаксис:

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

Данные ввода

Параметр	Описание	Тип	Обязательный
measure	Любая мера	Мера	Да
Row Col	Устанавливает направление вычисления	Ключевое слово	Нет
IncludeEmpty	Включает пустые значения в вычисление	Ключевое слово	Нет
reset_dims	Сбрасывает вычисление на указанные измерения	Список измерений	Нет
раздел	Измерение, в котором задан раздел	Ключевое слово	"Да" в случае сброса раздела

Примечания

- С функцией `RunningAverage` можно использовать операторы контекста расширенного синтаксиса.
- Направление вычисления можно задать с помощью операторов `Row` и `Col`.
- В случае применения сортировки к мере, на которую ссылается `RunningAverage`, скользящее среднее рассчитывается после сортировки меры.
- Измерения нужно всегда помещать в круглые скобки, даже если в списке измерений сброса присутствует только одно измерение.
- Измерения сброса в наборе должны разделяться двоеточиями.
- Функция `RunningAverage` не производит автоматический сброс среднего значения после разбиения блока или нового раздела.

Примеры

`RunningAverage([Доход])` возвращает следующие результаты:

<i>Страна</i>	<i>Курорт</i>	Выручка	Промежуточное среднее
США	Hawaiian Club	1 479 660	1 479 660
США	Bahamas Beach	971 444	1 225 552
Франция	French Riviera	835 420	1 095 508

`RunningAverage([Доход]; ([Страна]))` возвращает следующие результаты:

<i>Страна</i>	<i>Курорт</i>	Доход	Промежуточное среднее
США	Hawaiian Club	1 479 660	1 479 660

США	Bahamas Beach	971 444	1 225 552
Франция	French Riviera	835 420	835 420

В примере, где RunningAverage используется в разделе с [Кварталом], при использовании формулы RunningAverage([Доходы с продаж];([Квартал])) будут выведены следующие результаты:

K1

Город	Доходы с продаж	Промежуточное среднее
Нью-Йорк	1 987 114,70 \$	1 987 114,70 \$
Хьюстон	1 544 627,80\$	1 765 871,25\$
Лос-Анджелес	1 129 177,60\$	1 553 640,03\$

K2

Город	Доходы с продаж	Промежуточное среднее
Нью-Йорк	2 028 090,70\$	2 028 090,70\$
Хьюстон	1 380 838,20\$	1 704 464,45\$
Лос-Анджелес	980 405,30\$	1 463 111,40\$

Связанные сведения

[Оператор IncludeEmpty \[страница 281\]](#)

[Операторы Row/Col \[страница 283\]](#)

1.6.1.2.16 Функция RunningCount

Описание

Возвращает текущий счет набора чисел

Группа функций

Агрегирование

Синтаксис

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

Чтобы сбросить RunningCount для каждого раздела, рекомендуется использовать следующий синтаксис:

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

Данные ввода

Параметр	Описание	Тип	Обязательный
dimension measure	Любое измерение или мера	Измерение или мера	Да
Row Col	Устанавливает направление вычисления	Ключевое слово	Нет
IncludeEmpty	Включает пустые значения в вычисление	Ключевое слово	Нет
reset_dims	Сбрасывает вычисление на указанные измерения	Список измерений	Нет
раздел	Измерение, в котором задан раздел	Ключевое слово	"Да" в случае сброса раздела

Примечания

- Контекстные операторы расширенного синтаксиса можно использовать с функцией RunningCount.
- Направление вычисления можно задать с помощью операторов Row и Col.
- В случае применения сортировки к мере, на которую ссылается RunningCount, нарастающий счетчик вычисляется после сортировки меры.
- Измерения нужно всегда помещать в круглые скобки, даже если в списке измерений сброса присутствует только одно измерение.
- Измерения сброса в наборе должны разделяться двоеточиями.
- Функция RunningCount не производит автоматического сброса счетчика после разбиения по блокам или создания нового раздела.

Примеры

RunningCount ([Доход с продаж]) возвращает эти результаты в следующей таблице:

Страна	Курорт	Доход с продаж	Промежуточное количество
США	Hawaiian Club	1,479,660	1
США	Bahamas Beach	971,444	2
Франция	French Riviera	835,420	3

RunningCount ([Доход] ; ([Страна])) возвращает эти результаты в следующей таблице:

Страна	Курорт	Доход	Промежуточное количество
США	Hawaiian Club	1,479,660	1
США	Bahamas Beach	971,444	2
Франция	French Riviera	835,420	1

В примере, где RunningCount используется в разделе с [Неделя], при использовании формулы RunningCount ([Строки] ; ([Неделя])) и элемента управления вводом [Доход с продаж], ограничивающего список только доходами свыше 30 000 долл., будут выведены следующие результаты:

Неделя 1

Строки	Доход с продаж	Промежуточное количество
Футболки	\$186,191	1
Блузки	\$139,082	2
Платья	\$70,931	3

Неделя 2

Строки	Доход с продаж	Промежуточное количество
Аксессуары	\$344,617	1
Футболки	\$196,976	2
Блузки	\$105,597	3
Платья	\$76,290	4
Свитеры	\$68,364	5

Обратите внимание, что в неделе 1 есть три строки с доходом свыше 30 000 долл., тогда как в неделе 2 таких продуктов пять.

Связанные сведения

[Оператор IncludeEmpty \[страница 281\]](#)

[Операторы Row/Col \[страница 283\]](#)

[Оператор IncludeEmpty \[страница 281\]](#)

[Оператор IncludeEmpty \[страница 281\]](#)

1.6.1.2.17 Функция RunningMax

Описание

Возвращает промежуточный максимум измерения или меры

Группа функций

Агрегирование

Синтаксис

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

Чтобы сбросить RunningMax для каждого раздела, рекомендуется использовать следующий синтаксис:

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

Данные ввода

Параметр	Описание	Тип	Обязательный
dimension measure	Любое измерение или мера	Измерение или мера	Да
Row Col	Устанавливает направление вычисления	Ключевое слово	Нет
reset_dims	Сбрасывает вычисление на указанные измерения	Список измерений	Нет
раздел	Измерение, в котором задан раздел	Ключевое слово	"Да" в случае сброса раздела

Примечания

- С помощью функции RunningMax можно использовать операторы контекста расширенного синтаксиса.
- Направление вычисления можно задать с помощью операторов Row и Col.
- В случае применения сортировки к мере, на которую ссылается RunningMax, промежуточный максимум вычисляется после сортировки меры.

- Измерения нужно всегда помещать в круглые скобки, даже если в списке измерений сброса присутствует только одно измерение.
- Измерения сброса в наборе должны разделяться двоеточиями.
- Функция RunningMax не производит автоматического сброса максимума после разбиения блока или создания нового раздела.

Примеры

RunningMax ([Доход]) возвращает результаты, приведенные в следующей таблице:

Страна	Курорт	Доход	Текущий максимум
Франция	French Riviera	835,420	835,420
США	Bahamas Beach	971,444	971,444
США	Hawaiian Club	1,479,660	1,479,660

В примере, где RunningMax используется в разделе с [Городом], при использовании формулы RunningMax ([Доходы с продаж] ; ([Город])) будут выведены следующие результаты:

Остин

Квартал	Доход с продаж	Текущий максимум
K1	\$775,482.70	\$775,482.70
K2	\$667,850.30	\$775,482.70
K3	\$581,470.40	\$775,482.70
K4	\$674,869.80	\$775,482.70

Бостон

Квартал	Доход с продаж	Текущий максимум
K1	\$312,896.40	\$312,896.40
K2	\$291,431.00	\$312,896.40
K3	\$249,529.00	\$312,896.40
K4	\$429,850.20	\$429,850.20

Связанные сведения

[Оператор IncludeEmpty \[страница 281\]](#)

[Операторы Row/Col \[страница 283\]](#)

1.6.1.2.18 Функция RunningMin

Описание

Возвращает промежуточный минимум измерения или меры

Группа функций

Агрегирование

Синтаксис

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

Чтобы сбросить RunningMin для каждого раздела, рекомендуется использовать следующий синтаксис:

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

Данные ввода

Параметр	Описание	Тип	Обязательный
dimension detail measure	Любое измерение или мера	Измерение или мера	Да
Row Col	Задаёт направление вычисления	Ключевое слово	Нет
reset_dims	Сбрасывает вычисление на указанные измерения	Список измерений	Нет
раздел	Измерение, в котором задан раздел	Ключевое слово	"Да" в случае сброса раздела

Примечания

- С функцией RunningMin можно использовать контекстные операторы расширенного синтаксиса.
- Направление вычисления можно задать с помощью операторов Row и Col.
- В случае применения сортировки к мере, на которую ссылается RunningMin, промежуточный минимум вычисляется после сортировки меры.

- Измерения нужно всегда помещать в круглые скобки, даже если в списке измерений сброса присутствует только одно измерение.
- Измерения сброса в наборе должны разделяться двоеточиями.
- Функция `RunningMin` не производит автоматический сброс минимума после разбиения блока или нового раздела.

Примеры

`RunningMin([Доход с продаж])` возвращает эти результаты в следующей таблице:

<i>Страна</i>	<i>Курорт</i>	<i>Доход с продаж</i>	<i>Текущий минимум</i>
Франция	French Riviera	835,420	835,420
США	Bahamas Beach	971,444	835,420
США	Hawaiian Club	1,479,660	835,420

В примере, где `RunningMin` используется в разделе с [Городом], при использовании формулы `RunningMin([Доходы с продаж]; ([Город]))` будут выведены следующие результаты:

Остин

<i>Квартал</i>	<i>Доход с продаж</i>	<i>Текущий минимум</i>
K1	\$775,482.70	\$775,482.70
K2	\$667,850.30	\$667,850.30
K3	\$581,470.40	\$581,470.40
K4	\$674,869.80	\$581,470.40

Бостон

<i>Квартал</i>	<i>Доход с продаж</i>	<i>Текущий минимум</i>
K1	\$312,896.40	\$312,896.40
K2	\$291,431.00	\$291,431.00
K3	\$249,529.00	\$249,529.00
K4	\$429,850.20	\$249,529.00

Связанные сведения

[Оператор IncludeEmpty \[страница 281\]](#)

[Операторы Row/Col \[страница 283\]](#)

1.6.1.2.19 Функция RunningProduct

Описание

Возвращает промежуточное произведение меры

Группа функций

Агрегирование

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязателен
measure	Любая мера	Мера	Да
Row Col	Устанавливает направление вычисления	Ключевое слово	Нет
reset_dims	Сбрасывает вычисление на указанные измерения	Список измерений	Нет

Примечания

- С помощью функции `RunningProduct` можно использовать контекстные операторы расширенного синтаксиса.
- Направление вычисления можно задать операторами `Row` и `Col`.
- В случае применения сортировки к мере, на которую ссылается `RunningProduct`, промежуточное произведение вычисляется после сортировки меры.
- Измерения нужно всегда помещать в круглые скобки, даже если в списке измерений сброса присутствует только одно измерение.
- Измерения сброса в наборе должны разделяться двоеточиями.
- Функция `RunningProduct` не производит автоматического сброса продукта после разбиения блока или создания нового раздела.

Примеры

`RunningProduct ([Количество гостей])` возвращает результаты, приведенные в следующей таблице:

<i>Страна происхождения</i>	<i>Город</i>	Количество гостей	Промежуточное произведение
Япония	Кобе	6	6
Япония	Осака	4	24
США	Чикаго	241	5 784

`RunningProduct ([Количество гостей]; ([Страна происхождения]))` возвращает результаты, приведенные в следующей таблице:

<i>Страна происхождения</i>	<i>Город</i>	Количество гостей	Промежуточное произведение
Япония	Кобе	6	6
Япония	Осака	4	24
США	Чикаго	241	5 784

Связанные сведения

[Оператор IncludeEmpty \[страница 281\]](#)

[Операторы Row/Col \[страница 283\]](#)

1.6.1.2.20 RunningSum

Описание

Возвращает текущую сумму измерения

Группа функций

Агрегирование

Синтаксис

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

Чтобы сбросить RunningSum для каждого раздела, рекомендуется использовать следующий синтаксис:

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

Данные ввода

Параметр	Описание	Тип	Обязательный
measure	Любая мера	Мера	Да
Row Col	Устанавливает направление вычисления	Ключевое слово	Нет
reset_dims	Сбрасывает вычисление на указанные измерения	Список измерений	Нет
раздел	Измерение, в котором задан раздел	Ключевое слово	"Да" в случае сброса раздела

Примечания

- Функцию RunningSum можно использовать с контекстными операторами расширенного синтаксиса.
- Направление вычисления можно задать с помощью операторов Row и Col.
- В случае применения сортировки к мере, на которую ссылается функция RunningSum, промежуточная сумма вычисляется после сортировки меры.
- Измерения нужно всегда помещать в круглые скобки, даже если в списке измерений сброса присутствует только одно измерение.
- Измерения сброса в наборе должны разделяться двоеточиями.
- Функция RunningSum не производит автоматический сброс суммы после разбиения блока или создания нового раздела.

Пример

RunningSum([Доход]) возвращает результаты, приведенные в следующей таблице:

Страна	Курорт	Доход	Текущая сумма
Франция	French Riviera	835,420	835,420

США	Bahamas Beach	971,444	1,806,864
США	Hawaiian Club	1,479,660	3,286,524

RunningSum([Доход] ; ([Страна])) возвращает результаты, приведенные в следующей таблице:

Страна	Курорт	Доход	Текущая сумма
Франция	French Riviera	835,420	835,420
США	Bahamas Beach	971,444	971,444
США	Hawaiian Club	1,479,660	2,451,104

В примере, где RunningSum используется в разделе с [Кварталом], при использовании формулы RunningSum([Доходы с продаж] ; ([Квартал])) будут выведены следующие результаты:

K1

Город	Доход с продаж	Текущая сумма
Нью-Йорк	\$1,987,114.70	\$1,987,114.70
Хьюстон	\$1,544,627.80	\$3,531,742.50
Лос-Анджелес	\$1,129,177.60	\$4,660,920.10

K2

Город	Доход с продаж	Текущая сумма
Нью-Йорк	\$2,028,090.70	\$2,028,090.70
Хьюстон	\$1,380,838.20	\$3,408,928.90
Лос-Анджелес	\$980,405.30	\$4,389,334.20

Связанные сведения

[Оператор IncludeEmpty \[страница 281\]](#)

[Операторы Row/Col \[страница 283\]](#)

1.6.1.2.21 Функция StdDev

Описание

Возвращает стандартное отклонение меры

Группа функций

Агрегирование

Синтаксис

```
num StdDev(measure)
```

Данные ввода

Параметр	Описание	Тип	Обязателен
measure	Любая мера	Мера	Да

Примечания

Среднеквадратичное отклонение – это мера статистической дисперсии во множестве чисел. Она вычисляется следующим образом:

- поиск среднего значения во множестве чисел
- вычитание среднего значения из каждого числа в множестве и возведение разности в квадрат
- сложение всех возведенных в квадрат разностей
- деление полученной суммы на (**<количество чисел во множестве>** – 1).
- вычисление квадратного корня из результата.

Примеры

Если мера содержит набор значений (2, 4, 6, 8), StdDev([measure]) возвращает 2,58.

Связанные сведения

[Функция Var \[страница 85\]](#)

1.6.1.2.22 Функция StdDevP

Описание

Возвращает стандартное отклонение по совокупности для меры

Группа функций

Агрегирование

Синтаксис

```
num StdDevP(measure)
```

Данные ввода

Параметр	Описание	Тип	Обязателен
measure	Любая мера	Мера	Да

Примечания

Среднеквадратичное отклонение по совокупности – это мера статистического разброса множества значений. Она вычисляется следующим образом:

- поиск среднего значения в наборе чисел;
- вычитание среднего значения из каждого числа в наборе и возведение разности в квадрат;
- сложение всех возведенных в квадрат разностей;
- полученная сумма делится на (**<количество чисел во наборе>**).
- вычисление квадратного корня из результата.

Функцию StdDevP можно использовать с операторами контекста расширенного синтаксиса.

Примеры

Если мера содержит набор значений (2, 4, 6, 8) StdDevP([measure]) вернет 2,24.

1.6.1.2.23 Функция Sum

Описание

Возвращает сумму меры

Группа функций

Агрегирование

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязателен
measure	Любая мера	Мера	Да
member_set	Набор элементов	Набор элементов	Нет

Примечания

- Можно использовать операторы контекста расширенного синтаксиса с функцией Sum.
- При указании member_set, функция Sum возвращает сумму меры для всех элементов в наборе элементов.
- В member_set могут входить различные наборы, разделяемые точкой с запятой (;).
- Список наборов элементов должен быть заключен в {}.
- Если выражением набора элементов не указывается точный элемент или узел, в таблице должна присутствовать используемая в ссылках иерархия, то выражение набора элементов ссылается на текущий элемент в иерархии в таблице. Если этой иерархии нет в таблице, функция возвращает сообщение #MULTIVALUE.
- Агрегирование делегированных мер возвращает значение #TOREFRESH, если необходимое агрегирование недоступно в запросе. Чтобы получить доступ к новому уровню агрегирования, обновите документ. Например, это может понадобиться при использовании панели фильтров, если пользователь выбрал значение перед выбором фильтра "Все значения" или наоборот выбрал "Все значения" перед выбором значения.

- При миграции из XIR2 в XIR3 функции агрегирования, содержащие операторы IN и WHERE в запросах XIR2, следует явным образом включить в функцию Sum, используя круглые скобки:
В XIR2 формула имеет следующий вид: `=Sum([Measure] In ([Dim 1];[Dim 2])) In ([Dim 1]) Where ([Dim 3]="константа")`
Начиная с версии XIR3 необходимо изменить декларацию: `=Sum([Measure] In ([Dim 1];[Dim 2])) In ([Dim 1]) Where ([Dim 3]="константа")`
- Делегированная мера, определенная относительно группы, использует локальное агрегирование (агрегирование значения меры по сгруппированным значениям) и, в связи с этим, возвращает значение #UNAVAILABLE.
Даже если выполнить принудительное локальное агрегирование такой меры с использованием формулы "if then else" или значения группы, по-прежнему будет возвращаться сообщение #MULTIVALUE.

Примеры

Если для меры "Доходы от продаж" заданы значения 2000, 3000, 4000 и 1000, функция `Sum ([Доходы от продаж])` возвращает 10000.

Если [Калифорния] является элементом иерархии [География] (Страна > Штат > Город), то `Sum ([Доходы от продаж] ; {Descendants ([География]&[США].[Калифорния];1)})` возвращает итоговую выручку от продаж во всех городах Калифорнии.

1.6.1.2.24 Функция Var

Описание

Возвращает отклонение для меры

Группа функций

Агрегирование

Синтаксис

```
num Var(measure)
```

Данные ввода

Параметр	Описание	Тип	Обязателен
measure	Любая мера	Мера	Да

Примечания

Дисперсия – это мера статистического отклонения для множества чисел. Она вычисляется следующим образом:

- поиск среднего значения во множестве чисел
- вычитание среднего значения из каждого числа в множестве и возведение разности в квадрат
- сложение всех возведенных в квадрат разностей
- деление полученной суммы на (**<количество чисел во множестве>** – 1).

Дисперсия – это возведенное в квадрат значение среднеквадратичного отклонения.

Можно использовать операторов контекста расширенного синтаксиса с функцией `var`.

Примеры

Если мера содержит набор значений (2, 4, 6, 8), `var ([мера])` возвращает 6,67.

Связанные сведения

[Функция StdDev \[страница 81\]](#)

1.6.1.2.25 Функция VarP

Описание

Возвращает отклонение по совокупности для меры

Группа функций

Агрегирование

Синтаксис

```
num VarP(measure)
```

Данные ввода

Параметр	Описание	Тип	Обязателен
measure	Любая мера	Мера	Да

Примечания

Дисперсия генеральной совокупности – это мера статистического отклонения для множества чисел. Она вычисляется следующим образом:

- поиск среднего значения во множестве чисел
- вычитание среднего значения из каждого числа в множестве и возведение разности в квадрат
- сложение всех возведенных в квадрат разностей
- полученная сумма делится на (**<количество чисел во множестве>**)

Дисперсия генеральной совокупности – это возведенное в квадрат значение среднеквадратичного отклонения генеральной совокупности.

Можно использовать контекстные операторы расширенного синтаксиса с функцией `VarP`.

Примеры

Если мера содержит набор значений (2, 4, 6, 8), `VarP([measure])` возвращает 5.

Связанные сведения

[Функция StdDevP \[страница 83\]](#)

1.6.1.3 Символьные функции

1.6.1.3.1 Функция Asc

Описание

Возвращает код ASCII для символа

Группа функций

Символьные

Синтаксис

```
int Asc(string)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
string	Любая строка	Строка	Да

Примечания

Если строка содержит несколько символов, функция возвращает код ASCII для первого символа в строке.

Примеры

Asc ("A") возвращает 65.

Asc ("ab") возвращает 97.

Asc ([Страна]) возвращает 85, если значение измерения "Страна" равно "US".

1.6.1.3.2 Функция Char

Описание

Возвращает символ, связанный с кодом ASCII

Группа функций

Символьные

Синтаксис

```
string Char(ascii_code)
```

Данные ввода

Параметр	Описание	Тип	Требуется
ascii_code	Код ASCII	Число	Да

Примечания

Если число десятичное, данная функция игнорирует десятичную часть.

Например

s

Char(123) возвращает "{".

1.6.1.3.3 Функция Concatenation

Описание

Соединяет две символьных строки. Для чисел функция указывает их сумму вместо соединения.

📘 Примечание

Если хотя бы один из входных параметров является строкой, все остальные входные параметры преобразуются в строки.

Группа функций

Символьные

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
first_string	Первая строка	Строка или число	Да
second_string	Строка добавлена к первой строке	Строка или число	Да

Примечания

Для соединения строк также можно использовать оператор "+".

"First " + "Second" возвращает "First Second".

"First " + "Second" + " Third" возвращает "First Second Third".

Для включения нескольких измерений в функцию агрегирования можно использовать конкатенацию. Например, Count ([Sales Person]+[Quarter]+[Resort]) эквивалентно синтаксису Count (<Sales Person> , <Quarter> , <Resort>) , разрешенному Desktop Intelligence.

Примеры

`Concatenation("Первый" ; "Второй")` возвращает "Первый Второй".

`Concatenation("Первый" ; Concatenation("Второй" ; "Третий"))` возвращает "Первый Второй Третий".

Если `[A]` – число и `[A] = 1`, `Concatenation([A] ; [A])` возвращает "2".

Если `[A]` – строка и `[A] = 1`, `Concatenation([A] ; [A])` возвращает "11".

Если `[A]` – строка, `[B]` – число, `[A] = 1` и `[B] = 2`, `Concatenation([A] ; [B])` возвращает "12".

1.6.1.3.4 Функция Fill

Описание

Создает строку путем повторения строки `n` раз

Группа функций

Символьные

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
<code>repeating_string</code>	Повторяющаяся строка	Строка	Да
<code>num_repeats</code>	Количество повторений	Число	Да

Примеры

`Fill ("Нью-Йорк" ; 2)` возвращает "Нью-Йорк Нью-Йорк".

1.6.1.3.5 Функция FormatDate

Описание

Форматирует дату в соответствии с указанным форматом

Группа функций

Символьные

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
date	Дата в формате	Дата	Да
format_string	Формат, который будет применен к числу	Строка	Да

Примечания

- Формат вывода зависит от формата даты, примененного к ячейке.
- Форматирование цвета строк (например: [Красный], [Синий] и т. д.) нельзя применять к FormatDate.

Примеры

FormatDate(CurrentDate() ; "dd/MM/yyyy") возвращает "15/12/2005", если текущая дата – 15 декабря 2005 года.

Связанные сведения

[Пользовательские форматы \[страница 35\]](#)

1.6.1.3.5.1 Примеры Format_string для функции FormatDate

В синтаксисе FormatDate для format_string можно использовать примеры из следующей таблицы.

📘 Примечание

Эти примеры можно найти в диалоговом окне [Форматировать число](#) в Rich Client. Однако примеры отображаются в зависимости от языкового стандарта продукта, выбранного в предпочтениях стартовой панели BI. Например, если выбрать [Английский](#), то в качестве доступного образца будет представлено "Сентябрь 21, 2004".

Образец	Синтаксис
Вторник, сентябрь 21, 2004	dddd', 'mmmm d', 'yyyy
Сентябрь 21, 2004	mmmm d', 'yyyy
Сен 21, 2004	mmm d', 'yyyy
9/21/04	M'/'d'/'yy
Сен 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

→ Совет

Непосредственно текст рекомендуется заключать в одинарные кавычки, чтобы он не был принят за символы шаблона. Это продемонстрировано в последнем примере в таблице выше, где 'h' – это текст в шаблоне "HH'h'mm".

Связанные сведения

[Функция FormatDate \[страница 92\]](#)

[Пользовательские форматы \[страница 35\]](#)

1.6.1.3.6 Функция FormatNumber

Описание

Форматирует число согласно указанному формату

Группа функций

Символьные

Синтаксис

```
string FormatNumber(число;format_string)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
number	Число для форматирования	Число	Да
format_string	Формат, который будет применен к числу	Строка	Да

Примечания

- Формат вывода зависит от числового формата ячейки.
- Форматирование цвета строк (например: [Красный], [Синий] и т. д.) нельзя применять к FormatNumber.

Примеры

`FormatNumber([Доход]; "#,##.00")` возвращает 835 420,00, если [Доход] равен 835 420.

Связанные сведения

[Пользовательские форматы \[страница 35\]](#)

1.6.1.3.7 Функция HTMLEncode

Описание

Применяет к строке правила преобразования HTML

Группа функций

Символьные

Синтаксис

```
string HTMLEncode(html)
```

Ввод

Параметр	Описание	Тип	Обязательный
html	Строка HTML	Строка	Да

Примеры

`HTMLEncode("<Hello World!>")` возвращает "<Hello World!>", поскольку браузер интерпретирует символы. Внутренней системы она возвращает "<Hello World!>".

1.6.1.3.8 Функция InitCap

Описание

Делает заглавной первую букву строки

Группа функций

Символьные

Синтаксис

```
string InitCap(string)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
string	Строка для преобразования в заглавные буквы	Строка	Да

Примеры

InitCap("мы исходим из той очевидной истины") возвращает "Мы исходим из той очевидной истины".

1.6.1.3.9 По левому краю

Описание

Возвращает крайние левые символы строки.

📘 Примечание

Эта функция возвращает первые символы от логического начала строки. Поддерживаются языки с отображением/чтением справа налево, такие как арабский и иврит.

Группа функций

Символьные

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
string	Входная строка	string	Да
num_chars	Число символов, возвращаемых от начала строки.	number	Да

Пример

`Left([Страна];2)` возвращает "Фр", если [Страна] – "Франция"

1.6.1.3.10 Функция LeftPad

Описание

Заполняет строку до заданной минимальной длины, добавляя слева предварительно назначенную строку.

📘 Примечание

Эта функция заполняет строки от логического начала. Поддерживаются языки с отображением/чтением справа налево, такие как арабский и иврит.

Группа функций

Символьные

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
padded_string	Исходная строка	Строка	Да
length	Длина строки вывода	Число	Да
left_string	Строка, которую необходимо добавить в начало строки padded_string	Строка	Да

Примечания

- Если длина меньше, чем общая длина left_string и padded_string, left_string сокращается.
- Если длина меньше или равна длине padded_string, функция возвращает padded_string.
- Если длина больше, чем общая длина padded_string и left_string, left_string повторяется или частично повторяется достаточное количество раз, чтобы заполнить длину.

Примеры

LeftPad("Йорк" ; 8 ; "Нью-") возвращает "Нью-Йорк"

LeftPad("Йорк" ; 6 ; "Нью") возвращает "Нью-Йорк"

LeftPad("Йорк" ; 11 ; "Нью") возвращает "Нью Нью-Йорк"

LeftPad("Нью" ; 2 ; "Йорк") возвращает "Нью".

1.6.1.3.11 Функция LeftTrim

Описание

Удаляет начальные пробелы и специальные символы в строке.

📘 Примечание

Эта функция удаляет первые символы от логического начала строки. Поддерживаются языки с отображением/чтением справа налево, такие как арабский и иврит.

Группа функций

Символьные

Синтаксис

```
string LeftTrim(trimmed_string[;char])
```

Входные данные

Параметр	Описание	Тип	Обязательный
trimmed_string	Строка для усечения	String	Да
string	Символ для удаления	String	Нет

Примеры

- LeftTrim([Продукт]) возвращает "Ноутбук", если в поле [Продукт] введено " Ноутбук".
- LeftTrim([Продукт]; "=") возвращает "Ноутбук", если в поле [Продукт] введено "==Ноутбук".

1.6.1.3.12 Функция Length

Описание

Возвращает число символов в строке

Группа функций

Символьные

Синтаксис

```
int Length(строка)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
string	Входная строка	Строка	Да

Примеры

Функция `Length([Фамилия])` возвращает 6, если для объекта [Фамилия] задано значение "Иванов".

1.6.1.3.13 Функция Lower

Описание

Преобразует строку в нижний регистр

Группа функций

Символьные

Синтаксис

```
string Lower(string)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
string	Строка, преобразуемая в нижний регистр	Строка	Да

Примеры

Функция `Lower` ("Нью-Йорк") возвращает "нью-йорк".

1.6.1.3.14 Функция Match

Описание

Определяет соответствие строки шаблону

Группа функций

Символьные

Синтаксис

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

Входные данные

Параметр	Описание	Тип	Обязательный
test_string	Строка для проверки соответствия образцу текста	строка	Да
pattern	Шаблон текста	строка	Да

Примечания

- Образец содержать символы подстановки "*" (заменяет набор любых символов) или "?" (заменяет один символ).

Примеры

`Match([Страна]; "Ф*")` возвращает значение True, если [Страна] = "Франция".

`Match([Страна]; "?Ш?")` возвращает значение True, если [Страна] = "США".

`Match("Нью-Йорк"; "П*")` возвращает значение False

1.6.1.3.15 Функция Pos

Описание

Возвращает начальную позицию образца текста в строке.

Группа функций

Символьные

Синтаксис

```
int Pos(test_string;pattern[;start][;end])
```

Данные ввода

Параметр	Описание	Тип	Обязательный
test_string	Строка для проверки вхождения в нее образца текста	string	Да
pattern	Шаблон текста	string	Да
start	Начальная позиция поиска строки	integer	Нет
end	Конечная позиция поиска строки	integer	Нет

Примечания

- Если образец встречается больше одного раза, Pos возвращает позицию первого экземпляра.
- Поиск выполняется между начальной и конечной позициями (включительно).

Примеры

Pos("Нью-Йорк" ; "Нью") возвращает 1

Pos("Нью-Йорк, Нью-Йорк" ; "Нью") возвращает 1.

Pos("Нью-Йорк" ; "Йорк") возвращает 5.

Pos ("Hello World World" ; "World" ; 7) возвращает 7.

Pos ("Hello World World" ; "World" ; 8) возвращает 13.

Pos ("Hello World World" ; "World" ; 8 ; 13) возвращает 13.

Pos ("Hello World World" ; "World" ; 8 ; 10) возвращает 0.

1.6.1.3.16 Функция Replace

Описание

Заменяет часть строки другой строкой

Группа функций

Символьные

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Требуемый
replace_in	Строка, в которой заменяется текст	строка	Да
replaced_string	Заменяемый текст	строка	Да
replace_with	Текст, который заменяет replaced_string	строка	Да

Примеры

`Replace("Нью-Йорк" ; "ОРК" ; "орк")` возвращает "Нью-Йорк".

1.6.1.3.17 Reverse

Описание

Возвращает обратный порядок строки.

Группа функций

Символьные

Синтаксис

```
string Reverse(string)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
string	Строка для изменения порядка символов на обратный	String	Да

Примеры

`Reverse("abc123")` возвращает "321cba".

1.6.1.3.18 Право

Описание

Возвращает крайние правые символы строки (символы в конце строки).

📘 Примечание

Эта функция возвращает первые символы от логического конца строки. Поддерживаются языки с отображением/чтением справа налево, такие как арабский и иврит.

Группа функций

Символьные

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
string	Любая строка	string	Да
num_chars	Количество символов справа, которые следует вернуть	number	Да

Примеры

Функция `Right ([Страна] ; 2)` возвращает "ия", если объект [Страна] имеет значение "Франция".

1.6.1.3.19 Функция RightPad

Описание

Заполняет строку до заданной минимальной длины, добавляя справа назначенную строку.

📘 Примечание

Эта функция заполняет строки от логического конца. Поддерживаются языки с отображением/чтением справа налево, такие как арабский и иврит.

Группа функций

Символьные

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
padded_string	Исходная строка	Строка	Да
length	Длина строки вывода	Число	Да
right_string	Строка, которую необходимо добавить в конец строки padded_string	Строка	Да

Примечания

- Если длина меньше, чем общая длина right_string и padded_string, right_string сокращается.
- Если длина меньше или равна длине padded_string, функция возвращает padded_string.
- Если длина больше, чем общая длина padded_string и right_string, right_string повторяется или частично повторяется достаточное количество раз, чтобы заполнить всю длину строки.

Примеры

- RightPad("Нью-" ; 8 ; "Йорк") возвращает "Нью-Йорк"
- RightPad("Нью-" ; 6 ; "Йорк") возвращает "Нью-Йо"
- RightPad("Нью-" ; 11 ; "Йорк") возвращает "Нью-ЙоркЙор"
- RightPad("Нью-" ; 2 ; "Йорк") возвращает "Нью".

1.6.1.3.20 Функция RightTrim

Описание

Удаляет конечные пробелы и специальные символы в строке.

📌 Примечание

Эта функция удаляет последние символы от логического конца строки. Поддерживаются языки с отображением/чтением справа налево, такие как арабский и иврит.

Группа функций

Символьные

Синтаксис

```
string RightTrim(trimmed_string[;char])
```

Входные данные

Параметр	Описание	Тип	Обязательный
trimmed_string	Строка для усечения	String	Да
string	Символ для удаления	String	Нет

Примеры

- `RightTrim([Продукт])` возвращает "Ноутбук", если в поле [Продукт] введено "Ноутбук".
- `RightTrim([Продукт]; "=")` возвращает "Ноутбук", если в поле [Продукт] введено "Ноутбук==".

1.6.1.3.21 RPos

Описание

Возвращает начальную позицию последнего вхождения шаблона текста в строке.

Группа функций

Символьные

Синтаксис

```
int RPos(test_string;pattern[;start][;end])
```

Данные ввода

Параметр	Описание	Тип	Обязательный
test_string	Строка для проверки вхождения в нее шаблона текста	string	Да
pattern	Шаблон текста	string	Да
start	Начальная позиция поиска строки	integer	Нет
end	Конечная позиция поиска строки	integer	Нет

Примечания

Поиск выполняется между начальной и конечной позициями (включительно). Поиск выполняется от конца строки в обратном направлении.

Примеры

RPos ("Hello World World"; "World") возвращает 13.

RPos ("Hello World World"; "World"; 7) возвращает 13.

RPos ("Hello World World"; "World"; 8) возвращает 13.

RPos ("Hello World World"; "World"; 8; 13) возвращает 13.

RPos ("Hello World World"; "World"; 1; 10) возвращает 7.

RPos ("Hello World World"; "World"; 1; 6) возвращает 0.

1.6.1.3.22 Функция Substr

Описание

Возвращает часть строки

Группа функций

Символьные

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
string	Любая строка	Строка	Да
start	Начальная позиция извлеченной строки	Число	Да
length	Длина извлеченной строки	Число	Да

Примеры

SubStr ("Великобритания" ; 1 ; 5) возвращает "Велик".

SubStr ("Великобритания" ; 7 ; 7) возвращает "Британия".

1.6.1.3.23 Функция Trim

Описание

Удаляет начальные и конечные пробелы из строки для сканирования. Если в качестве параметра указан другой символ, удаляется указанный символ.

Группа функций

Символьные

Синтаксис

```
string Trim(trimmed_string[;char])
```

Входные данные

Параметр	Описание	Тип	Обязательный
string	Строка для усечения	Строка	Да
string	Символ для удаления	Строка	Нет

Примеры

Trim (" Великобритания ") возвращает "Великобритания".

Trim (Trim ("---Hello= ---" ; "-") ; "=") возвращает "-Hello= ".

1.6.1.3.24 Функция Upper

Описание

Преобразует строку символов в верхний регистр

Группа функций

Символьные

Синтаксис

```
string Upper(string)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
string	Строка для преобразования	Строка	Да

Примеры

`Upper ("Нью-Йорк")` возвращает "НЬЮ-ЙОРК".

1.6.1.3.25 Функция UriEncode

Описание

Применяет к строке правила кодировки URL

Группа функций

Символьные

Синтаксис

```
string UriEncode(html)
```


Данные ввода

Параметр	Описание	Тип	Обязательный
html	URL-адрес, который необходимо закодировать	Строка	Да

Примеры

`UrlEncode("http://www.sap.com")` возвращает `"http%3A%2F%2Fwww%2Esap%2Ecom"`.

1.6.1.3.26 Функция WordCap

Описание

Перевод первых букв всех слов в строке в верхний регистр

Группа функций

Символьные

Синтаксис

```
string WordCap(string)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
string	Строка, которую необходимо перевести в верхний регистр	Строка	Да

Примеры

`WordCap("Доход с продаж за март")` возвращает "Доход С Продаж За Март".

1.6.1.4 Функции даты и времени

1.6.1.4.1 Функция `CurrentDate`

Описание

Возвращает текущую дату, отформатированную в соответствии с национальными установками

Группа функций

Дата и время

Синтаксис

```
date CurrentDate()
```

Примеры

`CurrentDate()` возвращает 10 сентября 2002 года, если дата – 10 сентября 2002 года.

1.6.1.4.2 Функция `CurrentTime`

Описание

Возвращает текущее время, отформатированное в соответствии с национальными установками

Группа функций

Дата и время

Синтаксис

```
time CurrentTime()
```

Примеры

CurrentTime возвращает 11:15, если текущее время 11:15.

1.6.1.4.3 Функция DatesBetween

Описание

Возвращает число периодов между двумя датами, без учета времени.

Группа функций

Дата и время

Синтаксис

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

Входной

Параметр	Описание	Тип	Обязательный
first_date	Первая дата в диапазоне времени	Дата	Да
last_date	Последняя дата в диапазоне времени	Дата	Да
period	Тип периодов, подсчитываемых в заданном диапазоне	Предварительно определенный	Да

Примечание

- Возможные значения параметра `period`: `DayPeriod`, `WeekPeriod`, `MonthPeriod`, `QuarterPeriod`, `SemesterPeriod`, `YearPeriod`.

⚠ Предупреждение

При использовании параметра `weekPeriod` приложение не учитывает семидневную неделю. Недельная разница может быть любой, от одного до семи дней в зависимости от того, какой день используется для расчета. Также, понедельник задан в качестве первого дня недели, согласно стандарту ISO 8601, что означает, что между понедельником и воскресеньем перед ним всегда существует недельная разница.

- Если возвращаемое значение вне допустимого диапазона для типа `int`, возвращается ошибка `#OVERFLOW`.

Примеры

- `DatesBetween([Begin Date];[End Date];MonthPeriod)` возвращает 2, если `[Begin Date]` = 30 июня 2016 г., а `[End Date]` = 3 августа 2016 г.
- `DatesBetween([Begin Date];[End Date];DayPeriod)` возвращает -10, если `[Begin Date]` = 30 июня 2016 г., а `[End Date]` = 20 июня 2016 г.
- `DatesBetween([Begin Date];[End Date];QuarterPeriod)` возвращает 6, если `[Begin Date]` = 30 июня 2016 г., а `[End Date]` = 17 ноября 2017 г.
- `DatesBetween([Begin Date];[End Date];MonthPeriod)` возвращает 1, если `[Begin Date]` = 31 декабря 2015 г., а `[End Date]` = 1 января 2016 г.
- `DatesBetween([Begin Date];[End Date];DayPeriod)` возвращает 1, если `[Begin Date]` = 31 декабря 2015 г., а `[End Date]` = 1 января 2016 г.
- `DatesBetween([Begin Date];[End Date];WeekPeriod)` возвращает 0, если `[Begin Date]` = 31 декабря 2015 г., а `[End Date]` = 1 января 2016 г., так как оба дня относятся к одной неделе.

1.6.1.4.4 Функция DayName

Описание

Возвращает название дня в дате

Группа функций

Дата и время

Синтаксис

```
string DayName ( date )
```

Данные ввода

Параметр	Описание	Тип	Обязательный
date	Дата ввода	Дата	Да

Примеры

DayName ([Дата резервирования]) возвращает "Суббота", если дата в [Дата резервирования] – 15 декабря 2001 года (которая приходится на субботу).

Примечание

Дата на входе должна быть представлена переменной. Прямое указание даты, например DayName (" 07 / 15 / 2001 "), не допускается.

1.6.1.4.5 Функция DayNumberOfMonth

Описание

Возвращает номер дня в месяце

Группа функций

Дата и время

Синтаксис

```
int DayNumberOfMonth(дата)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
date	Дата ввода	Дата	Да

Примеры

DayNumberOfMonth([Дата резервирования]) возвращает 15, если в [Дата сохранения] стоит 15 декабря 2001.

1.6.1.4.6 Функция DayNumberOfWeek

Описание

Возвращает номер дня в неделе

Группа функций

Дата и время

Синтаксис

```
int DayNumberOfWeek(date)
```

Ввод

Параметр	Описание	Тип	Обязательный
date	Дата ввода	Дата	Да

Примечания

Первым днем недели в этой функции считается понедельник.

Примеры

DayNumberOfWeek([Дата резервирования]) возвращает 1, если дата в [Дата резервирования] – 2 мая 2005 года (понедельник).

1.6.1.4.7 Функция DayNumberOfYear

Описание

Возвращает номер дня в году

Группа функций

Дата и время

Синтаксис

```
int DayNumberOfYear(date)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
date	Дата ввода	Дата	Да

Примеры

`DayNumberOfYear([Дата резервирования])` возвращает 349, если в качестве значения параметра [Дата резервирования] установлено 15 декабря 2001.

1.6.1.4.8 Функция DaysBetween

Описание

Возвращает количество дней между двумя датами

Группа функций

Дата и время

Синтаксис

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

📘 Примечание

Необходимо убедиться, что заданные в аргументах даты относятся к одному и тому же часовому поясу. Это относится ко всем операциям на датами: сравнению и вычислениям.

Данные ввода

Параметр	Описание	Тип	Обязательный
first_date	Первая дата	Дата	Да
last_date	Последняя дата	Дата	Да

Примеры

DaysBetween([Дата продажи]; [Дата выставления счета]) возвращает 2, если [Дата продажи] – 15 декабря 2001 г., а [Дата выставления счета] – 17 декабря 2001 г.

1.6.1.4.9 Функция LastDayOfMonth

Описание

Возвращает дату последнего дня в месяце

Группа функций

Дата и время

Синтаксис

```
date LastDayOfMonth(date)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
date	Любая дата в месяце	Дата	Да

Примеры

`LastDayOfMonth([Дата продажи])` возвращает 31 декабря 2005 года, если [Дата продажи] равна 11 декабря 2005 года.

1.6.1.4.10 Функция LastDayOfWeek

Описание

Возвращает дату последнего дня недели

Группа функций

Дата и время

Синтаксис

```
date LastDayOfWeek(date)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
date	Любая дата недели	Дата	Да

Примечания

Первым днем недели в этой функции считается понедельник.

Примеры

Функция `LastDayOfWeek([Дата продажи])` возвращает 15 мая 2005 года (воскресенье), если [Дата продажи] имеет значение 11 мая 2005 года.

1.6.1.4.11 Функция Month

Описание

Возвращает название месяца в дате

Группа функций

Дата и время

Синтаксис

```
string Month(date)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
date	Дата ввода	Дата	Да

Примеры

Функция Month([Дата резервирования]) возвращает "Декабрь", когда [Дата резервирования] – 15 декабря 2005 г.

1.6.1.4.12 Функция MonthNumberOfYear

Описание

Возвращает номер месяца в дате

Группа функций

Дата и время

Синтаксис

```
int MonthNumberOfYear(дата)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
date	Любая дата в году	Дата	Да

Пример

Функция `MonthNumberOfYear([Дата бронирования])` возвращает 12, если [Дата бронирования] – 15 декабря 2005 г.

1.6.1.4.13 Функция MonthsBetween

Описание

Возвращает количество месяцев между двумя датами

Группа функций

Дата и время

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
first_date	Первая дата	Дата	Да
last_date	Последняя дата	Дата	Да

Примеры

Функция `MonthsBetween([Дата продажи]; [Дата выставления счета])` возвращает 1, если [Дата продажи] имеет значение 2 декабря 2005 г., а [Дата выставления счета] имеет значение 2 января 2006 г.

Функция `MonthsBetween([Дата продажи]; [Дата выставления счета])` возвращает 1, если [Дата продажи] имеет значение 31/03/2008 и [Дата выставления счета] имеет значение 30/04/2008.

Функция `MonthsBetween([Дата продажи]; [Дата выставления счета])` возвращает 118, если [Дата продажи] имеет значение 07/01/1993 и [Дата выставления счета] имеет значение 06/11/2002.

1.6.1.4.14 Функция Quarter

Описание

Возвращает номер квартала в дате

Группа функций

Дата и время

Синтаксис

```
int Quarter(date)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
date	Любая дата в квартале	Дата	Да

Примеры

Формула `Quarter([Дата резервирования])` возвращает значение 4, если дата в переменной [Дата резервирования] равна 15 декабря 2005 года.

1.6.1.4.15 Функция RelativeDate

Описание

Возвращает дату, отстоящую от другой даты.

Группа функций

Дата и время

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
start_date	Дата начала	Дата	Да
num	Количество единиц периода, добавленных к дате начала	Число	Да

Параметр	Описание	Тип	Обязательный
period	Тип периода, добавленного к дате начала	Предварительно определенные	Дополнительно

Примечания

- Параметр num может быть константой, числовым результатом функции, значением показателя или значением числового измерения и должен быть целым.
- Параметр num может быть отрицательным, чтобы возвращать более раннюю дату, чем start_date.
- В случае пропуска параметр period работает с днями (DayPeriod).
- Если при добавлении или вычитании месяцев (для SemesterPeriod, QuarterPeriod и MonthPeriod) день в возвращенном месяце не существует, нужно использовать последний день возвращенного месяца.
- Возможные значения для параметра периода: MillisecondPeriod, SecondPeriod, MinutePeriod, HourPeriod, DayPeriod, WeekPeriod, MonthPeriod, QuarterPeriod, SemesterPeriod, YearPeriod.

Примеры

RelativeDate([Reservation Date];2) возвращает 17 декабря 2005 года, если для объекта [Reservation Date] установлено значение 15 декабря 2005 года.

RelativeDate([Reservation Date];-3) возвращает 9 января 2007 года, если [Reservation Date] – 12 января 2007 года.

RelativeDate([Reservation Date];1;MonthPeriod) возвращает 12 февраля 2007 года, если [Reservation Date] – 12 января 2007 года.

1.6.1.4.16 Функция TimeBetween

Описание

Возвращает число периодов между двумя датами, учитывая время.

Группа функций

Дата и время

Синтаксис

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

📘 Примечание

Убедитесь, что даты, указанные в качестве аргументов, относятся к одному часовому поясу, так как при расчете возвращаемого значения смещение часового пояса не используется.

Данные ввода

Параметр	Описание	Тип	Обязательный
first_date	Первая дата в диапазоне времени	Время	Да
last_date	Последняя дата в диапазоне времени	Время	Да
period	Тип периодов, подсчитываемых в заданном диапазоне	Предварительно определенный	Да

Примечания

- Возможные значения параметра period: DayPeriod, WeekPeriod, MonthPeriod, QuarterPeriod, SemesterPeriod, YearPeriod, HourPeriod, MinutePeriod, SecondPeriod, MillisecondPeriod.
- Если возвращаемое значение вне допустимого диапазона для типа int, возвращается ошибка #OVERFLOW.

Примеры

- TimeBetween([Begin Date];[End Date];HourPeriod) возвращает 2, если [Begin Date] = 30 июня 2016 г. 8:45, а [End Date] = 30 июня 2016 г. 10:05.
- TimeBetween([Begin Date];[End Date];MinutePeriod) возвращает -10, если [Begin Date] = 30 июня 2016 г. 8:45, а [End Date] = 30 июня 2016 г. 8:35.
- TimeBetween([Begin Date];[End Date];SecondPeriod) возвращает 120, если [Begin Date] = 30 июня 2016 г. 8:45, а [End Date] = 30 июня 2016 г. 8:47.
- TimeBetween([Begin Date];[End Date];MonthPeriod) возвращает 1, если [Begin Date] = 31 декабря 2015 г. 11:45, а [End Date] = 1 января 2016 г. 8:47.
- TimeBetween([Begin Date];[End Date];DayPeriod) возвращает 1, если [Begin Date] = 31 декабря 2015 г. 11:45, а [End Date] = 1 января 2016 г. 8:47.

- `TimeBetween([Begin Date];[End Date];WeekPeriod)` возвращает 0, если `[Begin Date]` = 31 декабря 2015 г. 11:45, а `[End Date]` = 1 января 2016 г. 8:47, так как оба дня относятся к одной неделе.

1.6.1.4.17 Функция TimeDim

Описание

Измерение времени `TimeDim` позволяет строить оси времени на основе объекта юниверса типа данных. Измерение `TimeDim` возвращает данные для дат, задаваемых первым параметром, за периоды времени, задаваемые вторым параметром. Для периодов, не содержащих данных, возвращается первый день периода. Это позволяет построить полную ось для любого заданного периода. Такой подход гарантирует следующее:

- На оси сохраняется естественный порядок дат и времени (сначала идут самые старые объекты, за ними более новые).
- На оси будут представлены все периоды, заключенные между минимальной и максимальной датой текущего контекста.

ⓘ Примечание

Функцию `TimeDim` нельзя использовать для фильтрации формул (например, в фильтре, элементе управления вводом, ссылке на элемент, панели фильтрации или детализации). Вместо этого следует применять фильтр непосредственно к базовому измерению даты.

Группа функций

Дата и время

Синтаксис

```
TimeDim([Тип даты]; Тип периода)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
Тип даты	Объект даты для отчета, например, <code>InvoiceDate</code> .	Дата	Да

Параметр	Описание	Тип	Обязательный
Тип периода	<p>Период для результатов. Может иметь следующие значения:</p> <ul style="list-style-type: none"> DayPeriod MonthPeriod QuarterPeriod YearPeriod <p>Если значение не выбрано, по умолчанию используется значение DayPeriod. Данный объект должен быть объектом поставщика данных, доступным из объектов отчета, и не может быть переменной.</p>	Предварительно определенные	Дополнительно

Приведенные выше функции необходимо использовать совместно со следующими функциями:

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

Пример

В первой таблице приведены данные, относящиеся к датам. В приведенных ниже примерах запросов показан порядок интерпретации результатов.

Дата счета-фактуры	Доход
03.01.2000	31 607
08.01.2000	31 244
03.07.2000	38 154

Следующая формула `DayName(TimeDim([Invoice Date] ; QuarterPeriod)` возвращает значения за каждый день из приведенной выше таблицы.

Дата счета-фактуры	Доход
03.01.2000	31 607
08.01.2000	31 244
01.04.2000	
03.07.2000	38 154

Результаты функции TimeDim следует отформатировать с помощью функции Quarter. Результаты, возвращаемые функцией Quarter (Q1, Q2...), формируют следующую таблицу результатов:

Дата счета-фактуры	Доход
K1	62 851
K2	
K3	38 154

1.6.1.4.18 Функция ToDate

Описание

Преобразует строку символов в дату. Чтобы определить способ преобразования строки в дату в Web Intelligence, укажите формат даты в качестве параметра. Этот формат должен соответствовать формату даты исходной строки. Допустимые форматы даты приведены по следующей ссылке.

Группа функций

Дата и время

Синтаксис

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

или

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

📘 Примечание

В сценариях, когда для разных пользователей могут быть определены разные *Предпочтительный языковой стандарт* для просмотра, использование фиксированного формата для конкретного языкового стандарта не допускается. В подобных ситуациях следует использовать параметр INPUT_DATE_TIME, как показано в примере выше.

Данные ввода

Параметр	Описание	Тип	Обязательный
date_string	Строка будет учтена как дата.	String	Да
format	Формат даты, который используется в строке. Используйте "INPUT_DATE_TIME" для применения формата предпочтительного языкового стандарта просмотра.	String	Да*
cutoff_year	Необязательный параметр, указывающий год, используемый для отсечения. По умолчанию для этого года используется значение 2029.	Integer	Нет

*См. примечание выше. Используйте формат или INPUT_DATE_TIME в зависимости от ситуации.

Примеры

В `ToDate("12/15/2002" ; "мм/дд/yyyy")` "12" интерпретируется как номер месяца, "15" как номер дня и "2002" как год.

В `ToDate("Dec/02" ; "Ммм/yy")` "Dec" интерпретируется как сокращенное название месяца и "02" как две последние цифры значения года.

В `ToDate("15-December-02" ; "dd-Mmmm-yy")` "15" интерпретируется как номер дня, "December" как месяц, и "02" как две последние цифры года.

`ToDate("12/15/02 11:00:00" ; "INPUT_DATE_TIME")` интерпретируется как "12/15/02 11:00:00" в формате, который определен в региональных настройках *Предпочтительный языковой стандарт* на компьютере пользователя.

→ Совет

Используйте четыре цифры для аргумента года, чтобы избежать путаницы или нежелательных результатов. Например, "07" может означать "1907" или "2007".

ⓘ Примечание

- В случае INPUT_DATE_TIME и дата, и время должны быть указаны в строке ввода date_string.
- Если date_string невозможно интерпретировать как допустимую дату с указанным форматом, формула `ToDate()` возвращает #ERROR.

- Способ отображения даты в ячейке также зависит от выбранного формата даты в ячейке. Например, если выбран формат даты "мм/дд/yyyy", тогда ToDate("Dec/15/02" ; "MMM/dd/yy") будет отображено как 12/15/2002.
- Если в date_string год состоит из двух цифр и указан cutoff_year, то:
 - Текущий век является одним из значений этого cutoff_year.
 - Последние две цифры этого cutoff_year определяют пороговое значение для использования текущего века этого cutoff_year или предыдущего.
- Если cutoff_year строго ниже 100, возвращается сообщение об ошибке.

Связанные сведения

[Пользовательские форматы \[страница 35\]](#)

1.6.1.4.19 Функция Week

Описание

Возвращает номер недели в году

Группа функций

Дата и время

Синтаксис

```
int Week(date)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
date	Дата ввода	Дата	Да

Примеры

`week` ([Дата бронирования]) возвращает 1, когда [Дата бронирования] — 4 января 2004 г. (первая неделя 2004 года).

1.6.1.4.20 Функция Year

Описание

Возвращает год в дате

Группа функций

Дата и время

Синтаксис

```
int Year(date)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
date	Дата ввода	Дата	Да

Примеры

`Year` ([Дата записи]) возвращает 2005, если дата [Дата записи] — 15 декабря 2005.

1.6.1.5 Функции поставщика данных

1.6.1.5.1 Функция Connection

Описание

Возвращает параметры соединения базы данных, используемого поставщиком данных

Группа функций

Источник данных

Синтаксис

```
string Connection(dp)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
dp	Поставщик данных	Поставщик данных	Да

Примечания

- Имя поставщика данных необходимо указывать в квадратных скобках.
- По соображениям безопасности вывод этой функции не содержит имя хоста базы данных, а также имя и пароль пользователя.

1.6.1.5.2 Функция DataProvider

Описание

Возвращает имя запроса, определенное пользователем в диалоговом окне [Свойства запроса](#).

Группа функций

Поставщик данных

Синтаксис

```
string DataProvider(obj)
```

```
string DataProvider(dp)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
obj	Объект отчета	Объект отчета	Да
dp	Поставщик данных	Поставщик данных	Да

Примеры

`DataProvider([Общий доход])` возвращает "Продажи", если мера [Общий доход] содержится в поставщике данных "Продажи".

📘 Примечание

Функция `DataProvider` требует имя объекта и возвращает имя поставщика данных. При использовании в качестве параметра `DataProvider` другой функции (например, переменной измерения), не возвращающей имя объекта, функция `DataProvider` выдаст ошибку.

1.6.1.5.3 Функция `DataProviderKeyDate`

Описание

Возвращает ключевую дату поставщика данных

Группа функций

Поставщик данных

Синтаксис

```
date DataProviderKeyDate(dp)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
dp	Поставщик данных	Поставщик данных	Да

Примечания

- Имя поставщика данных необходимо указывать в квадратных скобках.
- Возвращенная ключевая дата форматируется в соответствии с языковым стандартом документа.
- Эта функция поддерживается только для поставщиков данных, основанных на запросах BEx с переменными SAP типа KEYDATE.
- Эта функция поддерживается только для устаревших UNV-юниверсов OLAP в BW. Она не поддерживается для прямого доступа к BEx в Web Intelligence и UNX-юниверсов BEx. Эта функция больше не используется, так как UNV-юниверсы исключены из версии 4.3.

Примеры

`DataProviderKeyDate([Продажи])` возвращает 3 августа 2007 года, если ключевая дата для поставщика данных продаж – 3 августа 2007 года.

1.6.1.5.4 Функция DataProviderKeyDateCaption

Описание

Возвращает заголовок ключевой даты поставщика данных

Группа функций

Поставщик данных

Синтаксис

```
string DataProviderKeyDateCaption(dp)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
dp	Поставщик данных	Поставщик данных	Да

Примечания

- Имя поставщика данных необходимо указывать в квадратных скобках.
- Эта функция поддерживается только для поставщиков данных, основанных на запросах BEx с переменными SAP типа KEYDATE.
- Эта функция поддерживается только для устаревших UNV-юниверсов OLAP в BW. Она не поддерживается для прямого доступа к BEx в Web Intelligence и UNX-юниверсов BEx. Эта функция больше не используется, так как UNV-юниверсы исключены из версии 4.3.

Примеры

`DataProviderKeyDateCaption([Продажи])` возвращает "Текущая календарная дата", если в заголовке ключевой даты поставщика данных "Продажи" стоит "Текущая календарная дата".

1.6.1.5.5 Функция DataProviderSQL

Описание

Возвращает SQL, созданный поставщиком данных

Группа функций

Источник данных

Синтаксис

```
string DataProviderSQL(dp)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
dp	Поставщик данных	Поставщик данных	Да

Примечания

Имя поставщика данных необходимо указывать в квадратных скобках.

Примеры

`DataProviderSQL([Запрос 1])` возвращает `SELECT country.country_name FROM country`, если код SQL для поставщика данных указан как `SELECT country.country_name FROM country`.

1.6.1.5.6 Функция DataProviderType

Описание

Возвращает тип поставщика данных

Группа функций

Поставщик данных

Синтаксис

```
string DataProviderType(dp)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
dp	Поставщик данных	Поставщик данных	Да

Примечания

- DataProviderType возвращает тип поставщика данных. Возможные возвращаемые значения: Universe, Web Intelligence, Excel, Text, Free-hand SQL, SAP HANA, SAP BW или Web Service.
- Имя поставщика данных необходимо указывать в квадратных скобках.

Примеры

DataProviderType([Продажи]) возвращает "Юниверс", если поставщик данных "Продажи" основан на юниверсе.

1.6.1.5.7 DataSourceDescription

Описание

Возвращает описание источника данных поставщика данных.

Группа функций

Поставщик данных

Синтаксис

```
string DataSourceDescription ( dp )
```

```
string DataSourceDescription ( obj )
```

Данные ввода

Параметр	Описание	Тип	Обязательный
obj	Объект отчета	Объект отчета	Да
dp	Поставщик данных	Поставщик данных	Да

Примечания

Возвращенная строка извлекается и сохраняется в документе при обновлении этого источника данных. Если этот источник данных никогда не обновлялся, то эта функция возвращает пустую строку.

1.6.1.5.8 DataSourceLocationType

Описание

Возвращает тип местоположения источника данных.

Группа функций

Поставщик данных

Синтаксис

```
string DataSourceLocationType ( dp )
```

```
string DataSourceLocationType ( obj )
```

Данные ввода

Параметр	Описание	Тип	Обязательный
obj	Объект отчета	Объект отчета	Да
dp	Поставщик данных	Поставщик данных	Да

Примечания

Возвращенная строка извлекается и сохраняется в документе при обновлении этого источника данных. Если этот источник данных никогда не обновлялся, то эта функция возвращает пустую строку.

1.6.1.5.9 DataSourcePath

Описание

Возвращает папку полного пути к источнику данных.

Группа функций

Поставщик данных

Синтаксис

```
string DataSourcePath ( dp )
```

```
string DataSourcePath ( obj )
```

Данные ввода

Параметр	Описание	Тип	Обязательный
obj	Объект отчета	Объект отчета	Да
dp	Поставщик данных	Поставщик данных	Да

Примечания

Возвращенная строка извлекается и сохраняется в документе при обновлении этого источника данных. Если этот источник данных никогда не обновлялся, то эта функция возвращает пустую строку.

1.6.1.5.10 DataSourceParentFolder

Описание

Возвращает имя родительской папки, содержащей источник данных поставщика данных.

Группа функций

Поставщик данных

Синтаксис

```
string DataSourceParentFolder ( dp )
```

```
string DataSourceParentFolder ( obj )
```

Данные ввода

Параметр	Описание	Тип	Обязательный
obj	Объект отчета	Объект отчета	Да
dp	Поставщик данных	Поставщик данных	Да

Примечания

Возвращенная строка извлекается и сохраняется в документе при обновлении этого источника данных. Если этот источник данных никогда не обновлялся, то эта функция возвращает пустую строку.

1.6.1.5.11 DataSourceName

Описание

Возвращает имя источника данных.

Группа функций

Поставщик данных

Синтаксис

```
string DataSourceName ( dp )
```

```
string DataSourceName ( obj )
```

Данные ввода

Параметр	Описание	Тип	Обязательный
obj	Объект отчета	Объект отчета	Да
dp	Поставщик данных	Поставщик данных	Да

Примечания

Возвращенная строка извлекается и сохраняется в документе при обновлении этого источника данных. Если этот источник данных никогда не обновлялся, то эта функция возвращает пустую строку.

1.6.1.5.12 Функция IsPromptAnswered

Описание

Определяет, был ли получен ответ на подсказку

Группа функций

Поставщик данных

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
dp	Поставщик данных, содержащий подсказку	Поставщик данных	Нет
prompt_string	Текст подсказки	Строка	Да

Примечания

- Имя поставщика данных необходимо указывать в квадратных скобках.
- IsPromptAnswered возвращает булево значение, которое можно использовать в функции.
- Если поместить функцию IsPromptAnswered непосредственно в столбец, она возвращает целочисленное значение (1=true; 0=false). Потом это значение можно форматировать с использованием формата булевых чисел.

Примеры

Функция IsPromptAnswered("Выберите город") возвращает значение True, если на подсказку с текстом "Выберите город", был получен ответ.

Функция `IsPromptAnswered ([Продажи]; "Выберите город")` возвращает значение `True`, если на подсказку с текстом "Выберите город" в поставщике данных `[Продажи]` был получен ответ.

1.6.1.5.13 Функция `LastExecutionDate`

Описание

Возвращает дату последнего обновления поставщика данных

Группа функций

Источник данных

Синтаксис

```
date LastExecutionDate(dp)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
dp	Поставщик данных	Поставщик данных	Да

Примечания

- Если в отчете содержится только один поставщик данных, параметр `dp` можно пропустить
- Имя поставщика данных необходимо указывать в квадратных скобках.
- Можно использовать функцию `DataProvider` для предоставления ссылки на поставщика данных.

Примеры

`LastExecutionDate ([Запрос продаж])` возвращает "4/3/2002", если поставщик данных "Запрос продаж" последний раз обновлялся 4 марта 2002 года.

Связанные сведения

[Функция DataProvider \[страница 135\]](#)

1.6.1.5.14 Функция LastExecutionDuration

Описание

Возвращает время последнего обновления поставщика данных в секундах

Группа функций

Источник данных

Синтаксис

```
num LastExecutionDuration(dp)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
dp	Поставщик данных	Поставщик данных	Да

Примечания

Имя поставщика данных необходимо указывать в квадратных скобках.

Примеры

Функция `LastExecutionDuration([Продажи])` возвращает 3, если поставщик данных "Продажи" затратил 3 секунды на возврат данных во время его последнего запуска.

1.6.1.5.15 Функция LastExecutionTime

Описание

Возвращает время последнего обновления поставщика данных

Группа функций

Источник данных

Синтаксис

```
time LastExecutionTime(dp)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
dp	Поставщик данных	Поставщик данных	Да

Примечания

- Если в отчете содержится только один поставщик данных, параметр dp можно опустить.
- Функцию DataProvider можно использовать для указания ссылки на поставщик данных.
- Имя поставщика данных необходимо указывать в квадратных скобках.

Примеры

Функция LastExecutionTime([Запрос продаж]) возвращает "14:48:00", если последнее обновление поставщика данных "Запрос продаж" производилось в 14:48:00.

Связанные сведения

[Функция DataProvider \[страница 135\]](#)

1.6.1.5.16 NumberOfColumns

Описание

Возвращает количество столбцов поставщика данных

Группа функций

Поставщик данных

Синтаксис

```
int NumberOfColumns (dp)
```

Входные данные

Параметр	Описание	Тип	Обязательный
dp	Поставщик данных	Поставщик данных	Да

Пример

`NumberOfRows ([Запрос 1])` возвращает 5, если в поставщике данных "Запрос 1" содержится 5 строк.

1.6.1.5.17 NumberOfDataProvider

Описание

Возвращает количество поставщиков данных в отчете

Группа функций

Источник данных

Синтаксис

```
int NumberOfDataProviders()
```

Примеры

Функция `NumberOfDataProviders()` возвращает 2, если в отчете указаны два поставщика данных.

1.6.1.5.18 Функция NumberOfRows

Описание

Возвращает количество строк в поставщике данных

Группа функций

Источник данных

Синтаксис

```
int NumberOfRows(dp)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
dp	Поставщик данных	Поставщик данных	Да

Примечания

- Имя поставщика данных необходимо указывать в квадратных скобках.
- Можно использовать функцию `DataProvider` для предоставления ссылки на поставщика данных.

Примеры

`NumberOfRows ([Запрос 1])` возвращает 10, если в поставщике данных "Запрос 1" содержится 10 строк.

Связанные сведения

[Функция DataProvider \[страница 135\]](#)

1.6.1.5.19 QueryName

Описание

Возвращает имя запроса поставщика данных.

Группа функций

Поставщик данных

Синтаксис

```
string QueryName ( dp )
```

```
string QueryName ( obj )
```

Данные ввода

Параметр	Описание	Тип	Обязательный
obj	Объект отчета	Объект отчета	Да
dp	Поставщик данных	Поставщик данных	Да

1.6.1.5.20 Функция RefValueDate

Описание

Возвращает справочные данные, используемые для отслеживания данных

Группа функций

Источник данных

Синтаксис

```
date RefValueDate()
```

Примеры

Функция RefValueDate() возвращает значение 15 декабря 2008 г., если опорная дата – 15 декабря 2008 г.

1.6.1.5.21 Функция RefValueUserReponse

Описание

Возвращает ответ на подсказку, когда опорные данные были текущими данными

Группа функций

Источник данных

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
dp	Поставщик данных	Поставщик данных	Нет
prompt_string	Текст подсказки	Строка	Да
Index	Указывает функции на то, что необходимо возвращать основные ключи базы данных для значений подсказки	Ключевое слово	Нет

Примечания

- Эта функция возвращает пустую строку, если отслеживание данных не включено.
- Имя поставщика данных необходимо указывать в квадратных скобках.
- Для указания ссылки на поставщик данных можно использовать функцию `DataProvider`.
- При выборе более одного значения для ответа на подсказку функция возвращает строку, состоящую из списка значений (или основных ключей, если указан оператор `Index`), разделенных между собой точками с запятой.

Примеры

`RefValueUserResponse("Какой город?")` возвращает "Лос-Анджелес", если в поле "Какой город?" было указано "Лос Анджелес" подсказка в момент времени, когда опорные данные были текущими.

`RefValueUserResponse([Запрос продаж]; "Какой город?")` возвращает "Лос-Анджелес", если в поле "Какой город?" было указано "Лос Анджелес" подсказка в поставщике данных "Запрос продаж" в момент времени, когда опорные данные были текущими.

1.6.1.5.22 Функция ServerValue

Описание

Возвращает значение базы данных для меры

Группа функций

Поставщик данных

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
measure	Любая мера	Показатель	Да

Примечания

- ServerValue игнорирует все локальные фильтры, применяемые к измерениям или иерархиям, используемым в расчете меры

Пример

ServerValue([Сумма Интернет-продаж]) возвращает значение базы данных для меры [Сумма Интернет-продаж]

1.6.1.5.23 Функция UniverseName

Описание

Возвращает имя юниверса или файла, на котором основан поставщик данных для юниверсов, текстов, источников данных Excel и FHSQL.

Группа функций

Поставщик данных

Синтаксис

```
string UniverseName(dp[,string])
```

```
string UniverseName(obj[,string])
```

Данные ввода

Параметр	Описание	Тип	Обязательный
dp	Поставщик данных	Поставщик данных	Да

Примечания

- Для источников данных SAP BW и SAP HANA функции возвращают имя представления. Возможные параметры:
 - `key`: возвращает техническое имя представления;
 - `caption`: возвращает заголовок представления;
 - `full`: возвращает полный путь представления (пакет + ключ). Например, "sales.sales::revenue" для SAP HANA или "xxx" для SAP BW
- Значение параметра `dp` в формуле автоматически обновляется при изменении имени поставщика данных. Если поставщик данных переименован в "Q1", то формула изменяется на `UniverseName([Q1])`.
- Имя поставщика данных необходимо указывать в квадратных скобках.
- Можно использовать функцию `DataProvider` для предоставления ссылки на поставщика данных.

Примеры

`UniverseName([Запрос 1])` возвращает "eFashion", если поставщик данных основан на юниверсе eFashion.

Связанные сведения

[Функция DataProvider \[страница 135\]](#)

1.6.1.5.24 Функция UserResponse

Описание

Возвращает ответ на подсказку

Группа функций

Поставщик данных

Синтаксис

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

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

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

Ввод

Параметр	Описание	Тип	Обязательный
dp	Поставщик данных	Поставщик данных	Нет
prompt_string	Текст подсказки	Строка	Да

Параметр	Описание	Тип	Обязательный
Index	Указывает функции на то, что необходимо возвращать первичные ключи базы данных для значений подсказки	Ключевое слово	Нет
multi_separator	Строка, используемая для разделения различных значений с ответами. По умолчанию используется ";".	Строка	Нет
DefaultSeparator	Строка по умолчанию (";"), используемая для разделения различных значений с ответами.	Ключевое слово	Нет
between_separator	Строка, используемая для разделения двух значений с ответами оператора диапазона или between в подсказке.	Строка	Нет

Примечания

- Имя поставщика данных необходимо указывать в квадратных скобках.
- Можно использовать функцию `DataProvider` для предоставления ссылки на поставщика данных.
- При выборе нескольких значений для ответа на подсказку функция возвращает строку, состоящую из списка значений (или первичных ключей, если указан оператор `Index`), разделенных между собой точкой с запятой.
- Вывод функции зависит от типа подсказки.

Примеры

`UserResponse("Which city?")` возвращает "Los Angeles", если в подсказке "Which city?" было указано "Los Angeles".

`UserResponse([Sales Query];"Which city?")` возвращает "Los Angeles", если в подсказке "Which city?" поставщика данных "Sales Query" было указано "Los Angeles".

`UserResponse([Sales Query];"Which city?";Index)` возвращает 23, если в подсказке "Which city?" поставщика данных "Sales Query" было указано "Los Angeles" и первичный ключ базы данных для Los Angeles равен 23.

`UserResponse("Which country?")` возвращает "France", если это подсказка с одним значением.

`UserResponse("Which country?")` возвращает "France;Holland;USA", если это подсказка с несколькими значениями.

UserResponse("Which country?") возвращает "France - USA", если это – подсказка типа "интервал" с одной записью.

UserResponse("Which country?") возвращает "France - Holland;Japan - USA", если это – подсказка типа "интервал" с несколькими записями.

UserResponse("Which country?") возвращает "France", если это – сложная подсказка, используемая с оператором "Equal To".

UserResponse("Which country?") возвращает ">France", если это – сложная подсказка, используемая с оператором "Greater Than".

UserResponse("Which country?") возвращает ">=France", если это – сложная подсказка, используемая с оператором "Greater Than" или "Equal To".

UserResponse("Which country?") возвращает "<France", если это – сложная подсказка, используемая с оператором "Less Than".

UserResponse("Which country?") возвращает "<=France", если это – сложная подсказка, используемая с оператором "Less Than" или "Equal To".

UserResponse("Which country?") возвращает "France - USA", если это – сложная подсказка, используемая с оператором "Between".

UserResponse("Which country?") возвращает "!France", если это – сложная подсказка, используемая с оператором "Not Equal To".

UserResponse("Which country?") возвращает "Austria;France - Holland;>Japan", если это – сложная подсказка с несколькими записями.

1.6.1.6 Функции документа

1.6.1.6.1 DocumentAuthor

Описание

Возвращает регистрационное имя создателя документа в приложении InfoView

Группа функций

Документ

Синтаксис

```
string DocumentAuthor()
```

Примеры

`DocumentAuthor()` возвращает "gkn", если регистрационное имя автора документа – "gkn".

1.6.1.6.2 Функция `DocumentCreationDate`

Описание

Возвращает дату создания документа

Группа функций

Документ

Синтаксис

```
date DocumentCreationDate()
```

Примеры

`DocumentCreationDate()` возвращает 15 декабря 2008 года, если документ был создан 15 декабря 2008 года.

1.6.1.6.3 Функция `DocumentCreationTime`

Описание

Возвращает время создания документа.

Группа функций

Документ

Синтаксис

```
time DocumentCreationTime()
```

Примеры

`DocumentCreationTime()` возвращает 11:15, если документ создан в 11:15.

1.6.1.6.4 Функция `DocumentDate`

Описание

Возвращает дату последнего сохранения документа

Группа функций

Документ

Синтаксис

```
date DocumentDate()
```

Примеры

`DocumentDate()` возвращает 8 августа 2005 года, если документ был в последний раз сохранен 8 августа 2005 года.

1.6.1.6.5 DocumentDescription

Описание

Возвращает описание документа в предпочитаемом пользователем языковом стандарте просмотра.

Группа функций

Документ

Синтаксис

```
string DocumentDescription()
```

Пример

`DocumentDescription()` возвращает "Анализ показателей продаж за четыре квартала 2019 г.", если документ имеет описание "Анализ показателей продаж за четыре квартала 2019 г.".

1.6.1.6.6 Функция DocumentName

Описание

Возвращает имя документа

Группа функций

Документ

Синтаксис

```
string DocumentName()
```

Примеры

`DocumentName()` возвращает "Отчет о продажах", если документ назван "Отчет о продажах".

1.6.1.6.7 Функция `DocumentOwner`

Описание

Возвращает имя для входа в систему/имя пользователя в стартовой панели BI владельца документа (последнего лица, сохранившего документ). (Для получения сведений об авторе (пользователе, создавшем документ) используется функция `DocumentAuthor`.)

Группа функций

Документ

Синтаксис

```
string DocumentOwner()
```

Примеры

`DocumentOwner()` возвращает "gkn", если "gkn" является именем пользователя или именем для входа в систему лица, сохранившего документ.

1.6.1.6.8 `DocumentParentFolder`

Описание

Возвращает имя папки, содержащей документ.

Группа функций

Документ

Синтаксис

```
string DocumentParentFolder()
```

Примечания

- Эта функция возвращает папку, содержащую текущий документ, в котором используется эта функция.
- При использовании в экземпляре эта функция возвращает папку, содержащую запланированный документ.
- Если документ хранится в папке "Входящие" или "Избранное" пользователя, возвращается имя этого пользователя.

Примеры

- `DocumentParentFolder()` возвращает "Root Folder", если используется в документе, размещенном в разделе "Общие папки".
- `DocumentParentFolder()` возвращает "Web Intelligence Samples" для документа из этой папки.
- `DocumentParentFolder()` возвращает "userA", если документ расположен в папке "Входящие" или "Избранное" пользователя.
- `DocumentParentFolder()` возвращает "ZZ_Charting", если это экземпляр документа.

1.6.1.6.9 Функция DocumentPartiallyRefreshed

Описание

Определяет, обновлен ли документ частично.

Группа функций

Документ

Синтаксис

```
логическое значение DocumentPartiallyRefreshed()
```

Примечания

`DocumentPartiallyRefreshed()` возвращает логическое значение, которое можно использовать в функции `If`.

Примеры

`DocumentPartiallyRefreshed()` возвращает значение `True`, если документ обновлен частично.

1.6.1.6.10 DocumentPath

Описание

Возвращает путь к документу. В экземпляре документа возвращает путь к документу и его имя.

Группа функций

Документ

Синтаксис

```
string DocumentPath()
```

Примечания

Для документа к пути добавляется суффикс "/". Это правило не распространяется на экземпляр документа.

Примеры

- `DocumentPath()` возвращает "Public Folders/Web Intelligence Samples/" для документа, хранящегося в этой папке.
- `DocumentPath()` возвращает "Public Folders/Web Intelligence Samples/ZZ_Charting", если используется в экземпляре этого запланированного документа.

1.6.1.6.11 Функция DocumentTime

Описание

Возвращает время последнего сохранения документа

Группа функций

Документ

Синтаксис

```
time DocumentTime()
```

Примечания

Формат возвращаемого времени зависит от формата ячейки.

Например

`DocumentTime()` возвращает 15:45, если документ был последний раз сохранен в 15:45.

1.6.1.6.12 Функция DrillFilters

Описание

Возвращает результат применения к документу или к объекту фильтра детализации в объявленном отчете в режиме детализации. В документе можно объявить другой отчет. Если не объявлять отчет, будет использоваться текущий активный отчет.

Группа функций

Документ

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
obj	Объект отчета	Объект отчета	Необходим obj или separator
separator	Разделитель фильтров детализации	Строка	Необходим obj или separator
report	Необязательный параметр. Имя отчета, который	Строка	Необходим obj или separator

Параметр	Описание	Тип	Обязательный
	требуется использовать. Он должен входить в документ. Если отчет не объявлен, используется текущий отчет.		

Примечания

- Можно вставить `DrillFilters` напрямую, вставив ячейку `DrillFilters`, без необходимости ввода формулы вручную.
- Если объект не указан, функция возвращает значения всех фильтров детализации, примененных к документу.

Примеры

`DrillFilters()` возвращает "США", если в документе применен фильтр детализации, допускающий в объекте [Страна] только значение "США".

`DrillFilters()` возвращает "США – 1999", если в документе применен фильтр, допускающий в объекте [Страна] только значение "США" и в объекте [Год] только значение "1999"

`DrillFilters(" / ")` возвращает "США / 1999", если в документе применены фильтры, допускающие в объекте [Страна] только значение "США" и в объекте [Год] только значение "1999"

`DrillFilters([Квартал])` возвращает "К3", если в документе применен фильтр детализации, допускающий в объекте [Квартал] только значение "К3"

1.6.1.6.13 ElementLinkingFilters

Описание

Возвращает значения, выбранные в имени блока для связи элементов.

Группа функций

Документ

Синтаксис

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

Входной

Параметр	Описание	Тип	Обязательный
string Block Name	Аргумент Block Name определяет блок, в котором была определена связь элементов.	String	Да
string separator	Это строка, используемая для разделения различных выбранных значений. По умолчанию используется "_".	String	Нет
string report	Если указан аргумент report, выбранные значения связи элементов возвращаются в указанный отчет. В противном случае значения возвращаются в отчет, в котором используется формула.	String	Нет

Примечания

- В документе, перенесенном из предыдущей версии, ElementLinkingFilters работает должным образом только после выбора новых значений для связи элементов и сохранения документа.

1.6.1.6.14 InputControlFilter

Описание

Возвращает значения, выбранные пользователем с помощью элемента управления вводом.

Группа функций

Документ

Синтаксис

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

Входной

Параметр	Описание	Тип	Обязательный
string InputControlName	Это имя элемента управления вводом. Функция возвращает значения, выбранные только с помощью этого элемента управления вводом.	String	Да
string separator	Это строка, используемая для разделения различных выбранных значений или значений, выбранных для разных элементов управления вводом. По умолчанию используется ";".	String	Нет
string report	Если указан аргумент report, выбранные значения элемента управления вводом возвращаются в указанный отчет. В противном случае значения возвращаются в отчет, в котором используется формула.	String	Нет

Примечания

- В документе, перенесенном из предыдущей версии, InputControlFilter работает должным образом только после выбора новых значений для элемента управления вводом и сохранения документа.

1.6.1.6.15 Функция PromptSummary

Описание

Возвращает текст подсказки и ответ пользователя на все подсказки в документе

Группа функций

Документ

Синтаксис

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

Данные ввода

Параметр	Тип	Описание	Обязательный
dp	Поставщик данных	Поставщик данных	Нет
sorting_order	Предварительно определенный	Порядок сортировки выходных данных функции	Нет (по умолчанию ascending)
show_definitive_prompts	Логический	Вывод развернутых подсказок BW в выходных данных функции	Нет (по умолчанию false)

Примечания

- Необязательное использование `dp` позволяет фильтровать вывод `PromptSummary()` для указанного запроса.
- Возможные значения параметра `sorting_order`: `default`, `ascending`, `descending`.
- Если параметр `sorting_order` в функции не указан, используется порядок, установленный в источнике данных.
- Если параметр `show_definitive_prompts` не используется, развернутые подсказки BW будут скрыты в выходных данных функции.

Пример

Ниже приведен пример вывода функции PromptSummary:

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

1.6.1.6.16 Функция QuerySummary

Описание

Возвращает информацию о запросах в документе. Для каждого запроса метод возвращает тип поставщика данных, имя поставщика данных, сведения об обновлении, свойства запроса и определение запроса (объекты результата и фильтры).

Группа функций

Документ

Синтаксис

```
string QuerySummary()
```

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

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

Входной

Параметр	Описание	Тип	Обязательный
dp	Поставщик данных	Поставщик данных	Нет
StatusOfData	Статус BW, присвоенный данным	Логический	Нет

Примечания

- Если вы передаете имя поставщика данных в качестве параметра, функция возвращает сведения только для этого поставщика данных. Имя этого поставщика данных необходимо заключить в квадратные скобки.
- Префиксом имени поставщика данных является его тип, который может быть следующим: Universe, Web Intelligence, Excel, Text, Free-hand SQL, SAP HANA, SAP BW или Web Service.
- Статус BW, присвоенный данным, указывает на дату последнего обновления инфо-провайдера BW и отображается в последней строке, возвращаемой этой функцией.

Примеры

QuerySummary() возвращает сведения о всех запросах в документе.

QuerySummary([Запрос 1]) возвращает сведения о запросах, основанных на поставщике данных [Запрос 1].

Пример вывода:

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

1.6.1.6.17 Функция ReportFilter

Описание

Возвращает фильтры отчета, примененные к объекту.

Группа функций

Документ

Синтаксис

```
string ReportFilter(obj[:separator])
```

Данные ввода

Параметр	Описание	Тип	Обязательный
obj	Объект отчета	Объект отчета	Да
separator	Эта строка может использоваться для разделения различных значений в списке. По умолчанию используется разделитель ";".	String	Нет

Примеры

Функция `ReportFilter([Country])` возвращает значение "US", если к объекту `Country` применен фильтр отчета, допускающий только значения "US".

1.6.1.6.18 Функция ReportFilterSummary

Описание

Возвращает общие сведения о фильтрах отчета в документе или отчете

Группа функций

Документ

Синтаксис

```
string ReportFilterSummary(report_name)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
report_name	Имя отчета	Строка	Нет

Примечания

Если параметр `report_name` опущен, функция `ReportFilterSummary` возвращает общие сведения обо всех фильтрах отчетов в документе.

Примеры

Функция `ReportFilterSummary()` возвращает сведения обо всех фильтрах отчетов в документе.

Функция `ReportFilterSummary("Report1")` возвращает сведения обо всех фильтрах отчетов в отчете "Отчет1".

Ниже приведен пример вывода функции `ReportFilterSummary`:

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

1.6.1.7 Логические функции

1.6.1.7.1 Функция Even

Описание

Определяет, является ли число четным

Группа функций

Логические

Синтаксис

```
bool Even(number)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
число	Любое число	Число	Да

Примечания

- Функция `Even` возвращает булево значение, которое можно использовать в функции `If`.
- Если поместить функцию `Even` непосредственно в столбец, она возвращает целочисленное значение (1=true; 0=false). Потом это значение можно форматировать с использованием формата булевых чисел.
- `Even` возвращает значение `True` для каждого четного целого числа и `False` для каждого десятичного числа.

Примеры

`Even(4)` возвращает значение `True`.

`Even(3)` возвращает значение `False`.

`Even(23.2)` возвращает значение `False`.

`Even(-4)` возвращает значение `True`.

`Even(-2.2)` возвращает значение `False`.

1.6.1.7.2 Функция IsDate

Описание

Определяет, является ли значение датой

Группа функций

Логические

Синтаксис

```
bool IsDate(obj)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
obj	Любой объект отчета	Объект отчета	Да

Примечания

- IsDate возвращает булево значение, которое можно использовать в функции If.
- Если поместить функцию IsDate непосредственно в столбец, она возвращает целочисленное значение (1=true; 0=false). Потом это значение можно форматировать с использованием формата булевых чисел.

Примеры

IsDate([Reservation Date]) возвращает значение True, если [Дата бронирования] является датой.

Или одно из следующего для возврата "Дата", если [Дата бронирования] является датой:

- If(IsDate([Дата бронирования])) Then "Date" Else "Not a date"
- If IsDate([Дата бронирования]) Then "Date" Else "Not a date"

Связанные сведения

[Функция If...Then...Else \[страница 249\]](#)

1.6.1.7.3 Функция IsError

Описание

Определяет, возвращает ли объект ошибку

Группа функций

Логические

Синтаксис

```
bool IsError(объект)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
obj	Любой объект отчета	Объект отчета	Да

Примечания

- Функция IsError возвращает булево значение, которое можно использовать в функции If.
- Если поместить функцию IsError непосредственно в столбец, она возвращает целочисленное значение (1=true; 0=false). Потом это значение можно форматировать с использованием формата булевых чисел.

Примеры

Функция IsError([Доход]) возвращает значение False, если переменная [Доход] не возвращает ошибку.

Функция IsError([Среднее число гостей]) возвращает значение True, если переменная [Среднее число гостей] возвращает ошибку деления на ноль (#DIV/0).

Функция `If IsError([Среднее число гостей]) Then "Ошибка" Else "Нет ошибок"` возвращает значение "Ошибка", если переменная `[Среднее число гостей]` возвращает ошибку деления на ноль (`#DIV/0!`).

Связанные сведения

[Функция If...Then...Else \[страница 249\]](#)

1.6.1.7.4 Функция IsLogical

Описание

Определяет, является ли значение булевым

Группа функций

Логические

Синтаксис

```
bool IsLogical(объект)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
obj	Любой объект отчета	Объект отчета	Да

Примечания

- Функция `IsLogical` возвращает булево значение, которое можно использовать в функции `If`.
- Если поместить функцию `IsLogical` непосредственно в столбец, она возвращает целочисленное значение (1=true; 0=false). Потом это значение можно форматировать с использованием формата булевых чисел.

Примеры

Функция `IsLogical(IsString([Страна]))` возвращает значение `True`.

Функция `IsLogical([Страна])` возвращает значение `False`, если страна возвращает какой-либо тип данных, отличный от булева.

Функция `If IsLogical(IsDate([Страна])) Then "Булев" Else "Не булев"` возвращает "Булев".

Связанные сведения

[Функция If...Then...Else \[страница 249\]](#)

1.6.1.7.5 Функция IsNull

Описание

Определяет, является ли значение неопределенным

Группа функций

Логические

Синтаксис

```
bool IsNull(obj)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
obj	Любой объект отчета	Объект отчета	Да

Примечания

- `IsNull` возвращает логическое значение, которое можно использовать в функции `If`.
- Если поместить функцию `IsNull` непосредственно в столбец, она возвращает целочисленное значение (1=true; 0=false). Потом это значение можно форматировать с использованием формата булевых чисел.

Примеры

`IsNull ([Доход])` возвращает значение `False`, если переменная `[Доход]` не является неопределенным значением.

`IsNull ([Гостей в среднем])` возвращает значение `True`, если переменная `[Гостей в среднем]` не является неопределенным значением.

Связанные сведения

[Функция If...Then...Else \[страница 249\]](#)

1.6.1.7.6 IsNumber

Описание

Определяет, является ли значение числом

Группа функций

Логические

Синтаксис

```
bool IsNumber (obj)
```

Ввод

Параметр	Описание	Тип	Обязательный
obj	Любой объект отчета	Объект отчета	Да

Примечания

- `IsNumber` возвращает булево значение, которое можно использовать в функции `If`.
- Если поместить функцию `IsNumber` непосредственно в столбец, она возвращает целочисленное значение (1=true; 0=false). Потом это значение можно форматировать с использованием формата булевых чисел.

Примеры

`IsNumber([Доход])` возвращает значение `True`, если переменная `[Доход]` является числом.

`IsNumber([Имя клиента])` возвращает значение `False`, если переменная `[Имя клиента]` не является числом.

`If IsNumber([Имя клиента]) Then "Число" Else "Не число"` возвращает "Не число", если переменная `[Имя клиента]` не является числом.

Связанные сведения

[Функция If...Then...Else \[страница 249\]](#)

1.6.1.7.7 Функция IsString

Описание

Определяет, является ли значение строкой

Группа функций

Логические

Синтаксис

```
bool IsString(obj)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
obj	Любой объект отчета	Объект отчета	Да

Примечания

- Функция `IsString` возвращает булево значение, которое можно использовать в функции `If`.
- Если поместить функцию `IsString` непосредственно в столбец, она возвращает целочисленное значение (1=true; 0=false). Потом это значение можно форматировать с использованием формата булевых чисел.

Примеры

Функция `IsString([Доход])` возвращает значение `False`, если переменная `[Доход]` не является строкой.

Функция `IsString([Имя покупателя])` возвращает значение `True`, если переменная `[Имя покупателя]` является строкой.

Функция `If IsString([Имя покупателя]) Then "Строка" Else "Не строка"` возвращает значение "Строка", если переменная `[Имя покупателя]` является строкой.

Связанные сведения

[Функция If...Then...Else \[страница 249\]](#)

1.6.1.7.8 Функция IsTime

Описание

Определяет, является ли переменная переменной времени

Группа функций

Логические

Синтаксис

```
bool IsTime(obj)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
obj	Любой объект отчета	Объект отчета	Да

Примечания

- IsTime возвращает булево значение, которое можно использовать в функции If.
- Если поместить функцию IsTime непосредственно в столбец, она возвращает целочисленное значение (1=true; 0=false). Потом это значение можно форматировать с использованием формата булевых чисел.

Примеры

IsTime([Срок резервации]) возвращает значение True, если переменная [Срок резервации] является переменной времени. .

IsTime([Гостей в среднем]) возвращает значение False, если переменная [Гостей в среднем] не является переменной времени.

If IsTime([Гостей в среднем]) Then "Время" Else "Не время" возвращает "Не время", если переменная [Гостей в среднем] не является переменной времени.

Связанные сведения

[Функция If...Then...Else \[страница 249\]](#)

1.6.1.7.9 Функции Odd

Описание

Определяет, является ли число нечетным

Группа функций

Логические

Синтаксис

```
bool Odd ( number )
```

Данные ввода

Параметр	Описание	Тип	Обязательный
число	Любое число	Число	Да

Примечания

- odd возвращает булево значение, которое можно использовать в функции if.
- Если поместить функцию odd непосредственно в столбец, она возвращает целочисленное значение (1=true; 0=false). Потом это значение можно форматировать с использованием формата булевых чисел.
- odd возвращает значение True для каждого десятичного числа и False для каждого четного целого числа.

Примеры

odd (5) возвращает значение True.

odd (4) возвращает значение False.

odd (23 . 2) возвращает значение True.

`odd (24 . 2)` возвращает значение `True`.

`odd (-23 . 2)` возвращает значение `True`.

`odd (-24 . 2)` возвращает значение `True`.

Связанные сведения

[Функция If...Then...Else \[страница 249\]](#)

1.6.1.8 Числовые функции

1.6.1.8.1 Функция Abs

Описание

Возвращает абсолютное значение числа

Группа функций

Числовые

Синтаксис

```
num Abs ( number )
```

Данные ввода

Параметр	Описание	Тип	Обязательный
number	Любое число	Число	Да

Примеры

`Abs (25)` возвращает 25.

`Abs (-11)` возвращает 11.

1.6.1.8.2 Функция Ceil

Описание

Возвращает число, округленное до ближайшего целого числа

Группа функций

Числовые

Синтаксис

```
num Ceil(number)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
number	Любое число	Число	Да

Примеры

`Ceil(2.4)` возвращает 3.

`Ceil(3.1)` возвращает 4.

`Ceil(-3.1)` возвращает -3.

1.6.1.8.3 Функция Cos

Описание

Возвращает косинус угла

Группа функций

Числовые

Синтаксис

```
num Cos ( angle )
```

Данные ввода

Параметр	Описание	Тип	Обязательный
angle	Угол в радианах	Число	Да

Пример

`Cos (180)` возвращает -0.6.

1.6.1.8.4 Функция EuroConvertFrom

Описание

Преобразует сумму в евро в другую валюту

Группа функций

Числовые

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
noneuro_amount	Сумма в евро	Число	Да
curr_code	Код ISO для целевой валюты	Строка	Да
round_level	Количество десятичных разрядов, до которых округляется результат	Число	Да

Примечания

Код валюты должен соответствовать коду одной из 12 валют ЕС, курс которых был зафиксирован по отношению к евро перед их отменой в январе 2002 года. Если это не выполняется, функция возвращает ошибку: #ERROR. Валюты:

BEF	Бельгийский франк
DEM	Немецкая марка
GRD	Греческая драхма
ESP	Испанская песета
FRF	Французский франк
IEP	Ирландский фунт
ITL	Итальянская лира
LUF	Люксембургский франк
NLG	Голландский гульден
ATS	Австрийский шиллинг
PTS	Португальское эскудо
FIM	Финская марка

Примеры

EuroConvertFrom(1000;"FRF";2) возвращает 6559,57.

EuroConvertFrom(1000;"FRF";1) возвращает 6559,60.

EuroConvertFrom(1000.04;"DEM";2) возвращает 1955,83.

EuroConvertFrom(1000.04;"DEM";1) возвращает 1955,80.

Связанные сведения

[Округление и усечение чисел \[страница 296\]](#)

1.6.1.8.5 EuroConvertTo

Описание

Преобразует сумму в евро

Группа функций

Числовые

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
noneuro_amount	Сумма в валюте, отличной от евро	Число	Да
curr_code	Код ISO валюты, отличной от евро	Строка	Да
round_level	Количество десятичных разрядов, до которых округляется результат	Число	Да

Пример

`EuroConvertTo(6559;"FRF";2)` возвращает 999,91.

`EuroConvertTo(6559;"FRF";1)` возвращает 999,90.

`EuroConvertTo(1955;"DEM";2)` возвращает 999,58.

`EuroConvertTo(1955;"DEM";1)` возвращает 999,60.

Примечание

Код валюты должен соответствовать коду одной из 12 валют ЕС, курс которых был зафиксирован по отношению к евро перед их отменой в январе 2002 года. Если это не выполняется, функция возвращает ошибку: #ERROR. Валюты:

BEF	Бельгийский франк
DEM	Немецкая марка
GRD	Греческая драхма
ESP	Испанская песета
FRF	Французский франк
IEP	Ирландский фунт
ITL	Итальянская лира
LUF	Люксембургский франк
NLG	Голландский гульден
ATS	Австрийский шиллинг
PTS	Португальское эскудо
FIM	Финская марка

Связанные сведения

[Округление и усечение чисел \[страница 296\]](#)

1.6.1.8.6 Функция EuroFromRoundError

Описание

Возвращает ошибку округления при преобразовании из евро

Группа функций

Числовые

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
euro_amount	Сумма в евро	Число	Да
curr_code	Код ISO для целевой валюты	Строка	Да
round_level	Количество десятичных разрядов, до которых округляется результат	Число	Да

Данные вывода

Ошибка округления при вычислениях

Примеры

`EuroFromRoundError(1000;"FRF";2)` возвращает значение 0. (Нет разницы между неокругленной конвертацией и конвертацией, округленной до 2 десятичных знаков.)

`EuroFromRoundError(1000;"FRF";1)` возвращает 0,03. (Неокругленная конвертация – 6559,57. Конвертация, округленная до 1 десятичного знака – 6559,60. Ошибка округления – 0,03.)

`EuroFromRoundError(1000;"DEM";2)` возвращает значение 0. (Нет разницы между неокругленной конвертацией и конвертацией, округленной до 2 десятичных знаков.)

`EuroFromRoundError(1000;"DEM";1)` возвращает -0,01. (Неокругленная конвертация – 1955,83. Конвертация, округленная до 1 десятичного знака – 1995,80. Ошибка округления – -0,03.)

Примечание

Код валюты должен соответствовать коду одной из 12 валют ЕС, курс которых был зафиксирован по отношению к евро перед их отменой в январе 2002 года. Если это не выполняется, функция возвращает ошибку: #ERROR. Валюты:

BEF	Бельгийский франк
DEM	Немецкая марка
GRD	Греческая драхма
ESP	Испанская песета
FRF	Французский франк
IEP	Ирландский фунт
ITL	Итальянская лира
LUF	Люксембургский франк
NLG	Голландский гульден
ATS	Австрийский шиллинг
PTS	Португальское эскудо
FIM	Финская марка

Связанные сведения

[Округление и усечение чисел \[страница 296\]](#)

1.6.1.8.7 Функция EuroToRoundError

Описание

Возвращает ошибку округления при преобразовании в евро

Группа функций

Числовые

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
noneuro_amount	Сумма в валюте, отличной от евро	Число	Да
curr_code	Код ISO валюты, отличной от евро	Строка	Да
round_level	Количество десятичных разрядов, до которых округляется результат	Число	Да

Примеры

`EuroToRoundError (6559 ; "FRF" ; 2)` возвращает значение 0. (Нет разницы между неокругленной конвертацией и конвертацией, округленной до 2 десятичных знаков.)

`EuroToRoundError (6559 ; "FRF" ; 1)` возвращает -0,01. (Неокругленная конвертация – 999,91. Конвертация, округленная до 1 десятичного знака – 999,90. Ошибка округления – 0,01.)

`EuroToRoundError (1955 ; "FRF" ; 2)` возвращает значение 0. (Нет разницы между неокругленной конвертацией и конвертацией, округленной до 2 десятичных знаков.)

`EuroToRoundError (1955 ; "DEM" ; 1)` возвращает 0,02. (Неокругленная конвертация – 999,58. Конвертация, округленная до 1 десятичного знака – 999,60. Ошибка округления – 0,02.)

Примечание

Код валюты должен соответствовать коду одной из 12 валют ЕС, курс которых был зафиксирован по отношению к евро перед их отменой в январе 2002 года. Если это не выполняется, функция возвращает ошибку: #ERROR. Валюты:

BEF	Бельгийский франк
DEM	Немецкая марка
GRD	Греческая драхма
ESP	Испанская песета

FRF	Французский франк
IEP	Ирландский фунт
ITL	Итальянская лира
LUF	Люксембургский франк
NLG	Голландский гульден
ATS	Австрийский шиллинг
PTS	Португальское эскудо
FIM	Финская марка

Связанные сведения

[Округление и усечение чисел \[страница 296\]](#)

1.6.1.8.8 Функция Exp

Описание

Возвращает значение экспоненциальной функции (число e , возведенное в указанную степень)

Группа функций

Числовые

Синтаксис

```
num Exp(power)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
power	Степень	Число	Да

Примечания

Экспоненциальная функция вычисляется путем возведения константы e (2,718...) в степень.

Примеры

`Exp(2, 2)` возвращает 9,03.

1.6.1.8.9 Функция Fact

Описание

Возвращает факториал числа

Группа функций

Числовые

Синтаксис

```
int Fact(number)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
number	Любое число	Число	Да

Примечания

Факториал числа – это произведение всех целых чисел от 1 до этого числа.

Примеры

`Fact (4)` возвращает 24.

`Fact (5 , 9)` возвращает 120.

1.6.1.8.10 Функция Floor

Описание

Возвращает число, округленное до ближайшего целого числа

Группа функций

Числовые

Синтаксис

```
int Floor(number)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
number	Любое число	Число	Да

Пример

`Floor (24 . 4)` возвращает 24.

1.6.1.8.11 Ln

Описание

Возвращает натуральный логарифм числа

Группа функций

Числовые

Синтаксис

```
num Ln(число)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
число	Любое число	Число	Да

Примеры

$\text{Ln}(10)$ возвращает 2,3.

1.6.1.8.12 Функция Log

Описание

Возвращает логарифм числа по указанному основанию

Группа функций

Числовые

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
number	Любое число	Число	Да
основание	Основание логарифма	Число	Да

Примеры

Log(125;5) возвращает 3.

1.6.1.8.13 Функция Log10

Описание

Возвращает логарифм по основанию 10 для числа

Группа функций

Числовые

Синтаксис

```
num Log10(number)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
number	Любое число	Число	Да

Примеры

`Log10(100)` возвращает 2.

1.6.1.8.14 Функция Mod

Описание

Возвращает остаток от деления двух чисел

Группа функций

Числовые

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
dividend	Делимое	Число	Да
divisor	Делитель	Число	Да

Примеры

`Mod(10;4)` возвращает 2.

`Mod(10,2;4,2)` возвращает 1,8

1.6.1.8.15 Функция Power

Описание

Возвращает число, возведенное в степень

Группа функций

Числовые

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
number	Число, возводимое в степень	Число	Да
power	Степень	Число	Да

Пример

Формула `Power(10;2)` возвращает 100.

1.6.1.8.16 Функция Rank

Описание

Ранжирует меру по измерениям

Группа функций

Числовые

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязателен
measure	Мера для ранжирования	Мера	Да
ranking_dims	Измерения, используемые для ранжирования меры	Список измерений	Нет
Top Bottom	Устанавливает порядок ранжирования: <ul style="list-style-type: none">• Top – по убыванию• Bottom – по возрастанию	Ключевое слово	Нет (Top по умолчанию)
reset_dims	Измерения, которые сбрасывают ранжирование	Список измерений	Нет

Примечания

- Эта функция использует установленный по умолчанию контекст вычисления для ранжирования, если не указаны измерения ранжирования.
- Измерения необходимо всегда помещать в круглые скобки, даже если в списке ранжирования или измерений сброса присутствует только одно измерение.
- Указываемые наборы ранжирования или измерений сброса должны разделяться точкой с запятой.

- По умолчанию ранжирование сбрасывается при переходе к разделу или разрыву блока.

Примеры

В следующей таблице ранжирование задается функцией `Rank ([Доход] ; ([Страна]))`:

	Доход	Ранжирование
Франция	835 420	2
США	2 451 104	1

В следующей таблице ранжирование задается функцией `Rank ([Доход] ; ([Страна]) ; Bottom)`. Аргумент `Bottom` означает, что ранжирование мер происходит в порядке убывания.

Страна	Доход	Ранжирование
Франция	835 420	1
США	2 451 104	2

В следующей таблице ранжирование задается функцией `Rank ([Доход] ; ([Страна] ; [Курорт]))`:

Страна	Курорт	Доход	Ранжирование
Франция	French Riviera	835 420	3
США	Bahamas Beach	971 444	2
США	Hawaiian Club	1 479 660	1

В следующей таблице ранжирование задается функцией `Rank ([Доход] ; ([Страна] ; [Год]) ; ([Страна]))`. Ранжирование сбрасывается на измерении страны.

Страна	Год	Доход	Ранжирование
Франция	FY1998	295 940	1
Франция	FY1999	280 310	2
Франция	FY2000	259 170	3
США	FY1998	767 614	3
США	FY1999	826 930	2
США	FY2000	856 560	1

Связанные сведения

[Операторы Bottom/Top \[страница 278\]](#)

1.6.1.8.17 Функция Round

Описание

Округляет число

Группа функций

Числовые

Синтаксис

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

Ввод

Параметр	Описание	Тип	Обязательный
number	Округляемое число	Число	Да
round_level	Количество десятичных знаков, до которого округляется число	Число	Да

Примеры

Round (9 . 44 ; 1) возвращает 9.4.

Round (9 . 45 ; 1) возвращает 9.5.

Round (9 . 45 ; 0) возвращает 9.

Round (9 . 45 ; -1) возвращает 10.

Round (4 . 45 ; -1) возвращает 0.

Связанные сведения

[Округление и усечение чисел \[страница 296\]](#)

1.6.1.8.18 Функция Sign

Описание

Возвращает знак числа

Группа функций

Числовые

Синтаксис

```
int Sign(number)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
number	Любое число	Число	Да

Примечания

Знак возвращает 1, если число является отрицательным, 0 если число равно нулю, 1 если число является положительным.

Примеры

Знак (3) возвращает 1.

Знак (27 . 5) возвращает 1.

1.6.1.8.19 Функция Sin

Описание

Возвращает синус угла.

Группа функций

Числовые

Синтаксис

```
num Sin(angle)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
angle	Угол в радианах	Число	Да

Пример

`Sin(234542)` может вернуть -0,116992 или -0,12 (в зависимости от настроек десятичного разделителя).

1.6.1.8.20 Функция Sqrt

Описание

Возвращает квадратный корень из числа

Группа функций

Числовые

Синтаксис

```
num Sqrt (number)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
number	Любое число	Число	Да

Например

`Sqrt (25)` возвращает 5.

1.6.1.8.21 Функция Tan

Описание

Возвращает тангенс угла

Группа функций

Числовые

Синтаксис

```
num Tan (angle)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
angle	Угол в радианах	Число	Да

Примеры

`Tan (90)` возвращает -2.

1.6.1.8.22 ToDecimal

Описание

Возвращает десятичное.

Группа функций

Числовые

Синтаксис

```
num ToDecimal(number|string)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
number string	Число или строка, которая может быть воспринята как число	Число или строка	Да

Примечания

Если `string` не является числом, `ToDecimal` возвращает `#ERROR`.

Примеры

`ToDecimal("1234567890.1234567890")` возвращает `1234567890.1234567890`.

`ToDecimal("1234567890.12345")` возвращает `1234567890.12345`.

`ToDecimal("abcdefghijkl")` возвращает `#ERROR`.

1.6.1.8.23 Функция ToNumber

Описание

Преобразует строку в число

Группа функций

Числовые

Синтаксис

```
num ToNumber(string)
```

или

Данные ввода

Параметр	Описание	Тип	Обязательный
<code>string</code>	Число в качестве строки	Строка	Да

Примечания

Если строка не является числом, функция ToNumber возвращает #ERROR.

Примеры

ToNumber ("45") возвращает 45.

1.6.1.8.24 Функция Truncate

Описание

Обрезает число

Группа функций

Числовые

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
number	Округляемое число	Число	Да
truncate_level	Количество знаков после запятой, до которого обрезается число	Число	Да

Примечания

Например

`Truncate (3,423; 2)` возвращает 3,42.

Связанные сведения

[Округление и усечение чисел \[страница 296\]](#)

1.6.1.9 Функции множества

1.6.1.9.1 Функция Ancestor

Описание

Возвращает элемент-предок для элемента

Группа функций

Набор

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
member	Любой элемент	member	Да

Параметр	Описание	Тип	Обязательный
level	Уровень предка	level	Необходимо указать либо параметр level, либо параметр distance
distance	Расстояние уровня предка от текущего уровня	int	Необходимо указать либо параметр level, либо параметр distance

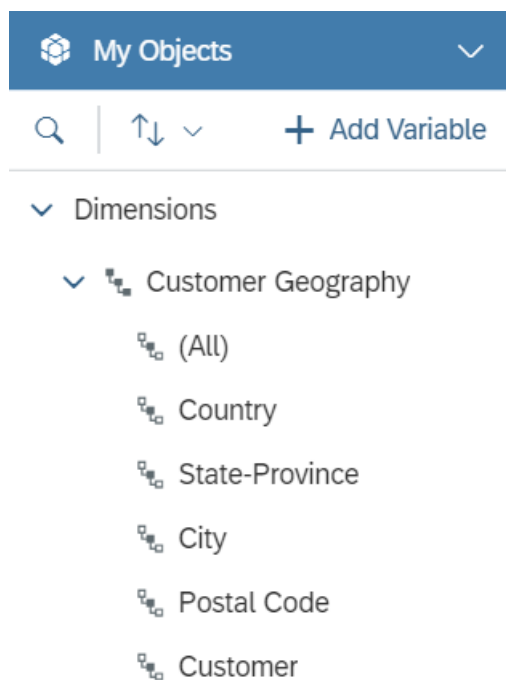
Примечания

- `ancestor` не применяется в качестве отдельной функции. Эта функция используется в качестве входного параметра для функций агрегирования, задающего набор элементов для агрегирования.
- `member`— текущий элемент иерархии. Если иерархия не является контекстом для блока, формула вернет пустое значение.
- `distance` (расстояние) должно быть положительным значением.

Примеры

Все следующие примеры взяты из источника данных на английском языке.

На основании следующей географической иерархии требуется узнать влияние объема интернет-продаж каждого клиента независимо от города клиента.



Сначала для каждого города требуется получить значение объема интернет-продаж для страны:

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

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

Затем рассчитать долю каждого города в объеме интернет-продаж для страны:

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

Customer Geography	City Contribution	Internet Sales Amount
▼ All Customers		29,358,677.22
▼ Australia		9,061,000.58
▼ New South Wales	100.00%	3,934,485.73
▼ Coffs Harbour	5.98%	235,454.97
▼ 2450	5.98%	235,454.97
Adriana Smith	0.14%	5,333.25
Aimee Guo	0.00%	77.27
Allison R. Young	0.00%	39.98
Ann A. Sara	0.00%	39.98
Antonio G. Patterson	0.21%	8,068.03
Ariana Stewart	0.15%	6,070.59
Arthur Kapoor	0.00%	23.97
Barbara W. Lal	0.07%	2,795.01
Bobby D. Saunders	0.00%	120.48
Brianna J. Johnson	0.00%	38.98

📌 Примечание

При использовании соединений BICS с поставщиками SAPBW требуется указать уровень смещения вместо наименования уровня:

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

В этом случае также будут получены результаты для штата-провинции и страны.

Связанные сведения

[Агрегирование \[страница 46\]](#)

[Average \[страница 53\]](#)

[Количество \[страница 54\]](#)

[Функция Max \[страница 61\]](#)

[Функция Min \[страница 63\]](#)

[Функция Sum \[страница 84\]](#)

1.6.1.9.2 Функция Children

Описание

Возвращает дочерние элементы для элемента иерархии в функции агрегирования.

Группа функций

Набор

Синтаксис

```
member_set member.Children
```

Данные ввода

Параметр	Описание	Тип	Обязательный
member	Любой элемент	member	Да

Примечания

- `Children` не используется в качестве отдельной функции. Эта функция используется в качестве входного параметра для функций агрегирования, задающего набор элементов для агрегирования.
- `member` — текущий элемент иерархии. Если иерархия не является контекстом для блока, формула вернет пустое значение.

Примеры

`[География].[США].[Калифорния].Children` вернет `[Лос-Анджелес]`, `[Сан-Франциско]`, `[Сан-Диего]`.

`[География].Children` вернет `[Лос-Анджелес]`, `[Сан-Франциско]`, `[Сан-Диего]`, если `[Калифорния]` является текущим элементом иерархии `[География]`.

Связанные сведения

[Агрегирование \[страница 46\]](#)

[Average \[страница 53\]](#)

[Количество \[страница 54\]](#)

[Функция Max \[страница 61\]](#)

[Функция Min \[страница 63\]](#)

[Функция Sum \[страница 84\]](#)

1.6.1.9.3 Функция Depth (Глубина)

Описание

Возвращает глубину элемента в иерархии

Группа функций

Набор

Синтаксис

```
int member.Depth
```

Данные ввода

Параметр	Описание	Тип	Обязательный
member	Любой элемент	member	Да

Примечания

- Глубина – это расстояние элемента от верхнего уровня иерархии.
- Самым верхним уровнем иерархии является уровень 0.
- member — текущий элемент иерархии. Если иерархия не является контекстом для блока, формула вернет пустое значение.

Примеры

Требуется узнать глубину элементов иерархии:

```
=[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

Затем скомбинировать с функцией Children, чтобы проверить наличие всех дней, перечисленных в каждом месяце:

```
=If [Calendar].[Date.Calendar].Depth = 4 Then Count([Internet Sales].[Internet Sales Amount];{[Calendar].[Date.Calendar].Children()})
```


Date.Calendar	Internet Sales Amount	[Date.Calendar].Depth	[Date.Calendar].Children()
[-] All Periods	29,358,677.22	0	
[-] CY 2001	3,266,373.66	1	
[-] H2 CY 2001	3,266,373.66	2	
[-] Q3 CY 2001	1,453,522.89	3	
[-] July 2001	473,388.16	4	31
[-] August 2001	506,191.69	4	30
[-] September 2001	473,943.03	4	29
[-] Q4 CY 2001	1,812,850.77	3	
[-] October 2001	513,329.47	4	30
[-] November 2001	543,993.41	4	30
[-] December 2001	755,527.89	4	31

1.6.1.9.4 Функция Descendants

Описание

Возвращает потомки элемента иерархии в функции агрегирования.

Группа функций

Набор

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
member	Любой элемент	member	Да

Параметр	Описание	Тип	Обязательный
level	Уровень потомка	level	Нет (по умолчанию используется уровень member)
distance	Расстояние уровня потомка от текущего уровня	int	Нет (по умолчанию используется уровень member)
desc_flag	Определяет возвращаемые элементы-потомки	ключевое слово	Нет (по умолчанию Self)

Примечания

- Descendants не применяется в качестве отдельной функции. Эта функция используется в качестве входного параметра для функций агрегирования, задающего набор элементов для агрегирования.
- member — текущий элемент иерархии. Если иерархия не является контекстом для блока, формула вернет пустое значение.
- Значение Self в desc_flag относится к уровню, определяемому параметром level|distance.
- Значение Before в desc_flag относится ко всем уровням выше уровня, определяемого параметром level|distance.
- Значение After в desc_flag относится ко всем уровням ниже уровня, определяемого параметром level|distance.
- Параметр desc_flag имеет следующие значения:

Self	Возвращает потомки на уровне, определяемом параметром level distance, включая текущий элемент, если он находится на этом уровне.
Before	Возвращает текущий элемент и все потомки выше уровня, определяемого параметром level distance.
After	Возвращает потомки ниже уровня, определяемого параметром level distance.
Self_Before	Возвращает текущий элемент и все потомки выше уровня, определяемого параметром level distance, включая этот уровень.
Self_After	Возвращает текущий элемент и все потомки ниже уровня, определяемого параметром level distance, включая этот уровень.
Before_After	Возвращает текущий элемент и все потомки за исключением тех, что находятся на уровне, определяемом параметром level distance.
Self_Before_After	Возвращает текущий элемент и все потомки.

Leaves

Возвращает все элементы между текущим элементом и уровнем, определяемым параметром level|distance, не имеющие дочерних элементов.

- distance (расстояние) должно быть положительным значением.

Пример

Существует финансовая иерархия, некоторые узлы не всегда являются кумулятивными, но требуется получить сумму их потомков. В этом примере будет получена сумма потомков для каждого элемента баланса только на один уровень ниже:

```
=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		

Затем можно получить сумму всех элементов под текущими активами:

```
=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	

Потом добавить текущие активы:

```
=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	

Связанные сведения

[Агрегирование \[страница 46\]](#)

[Average \[страница 53\]](#)

[Количество \[страница 54\]](#)

[Функция Max \[страница 61\]](#)

[Функция Min \[страница 63\]](#)

[Функция Sum \[страница 84\]](#)

1.6.1.9.5 Функция IsLeaf

Описание

Определяет, является ли элемент элементом конечного узла

Группа функций

Разные

Синтаксис

```
bool member.IsLeaf
```

Данные ввода

Параметр	Описание	Тип	Обязательный
member	Любой элемент	member	Да

Примечания

- Элемент конечного узла – это элемент, не имеющий дочерних элементов.
- `member` — текущий элемент иерархии. Если иерархия не является контекстом для блока, формула вернет пустое значение.

Примеры

Требуется узнать, является ли строка значением дня:

```
=[Calendar].[Date.Calendar].IsLeaf()
```

Date.Calendar	'=[Query 1].[Calendar].[Date.Calendar].IsLeaf
[-] All Periods	0
[-] CY 2001	0
[-] H2 CY 2001	0
[-] Q3 CY 2001	0
[-] July 2001	0
July 1, 2001	1
July 2, 2001	1
July 3, 2001	1
July 4, 2001	1
July 5, 2001	1
July 6, 2001	1
July 7, 2001	1
July 8, 2001	1

1.6.1.9.6 Функция Key

Описание

Возвращает ключ элемента

Синтаксис

```
string member.Key
```

Группа функций

Набор

Данные ввода

Параметр	Описание	Тип	Обязательный
member	Любой элемент	member	Да

Примечания

- Ключ является внутренним идентификатором элемента.
- `member` — текущий элемент иерархии. Если иерархия не является контекстом для блока, формула вернет пустое значение.

Пример

`[Geography].[US].Key` возвращает "XYZ", если элемент `[US]` имеет ключ "XYZ".

1.6.1.9.7 Функция Lag

Описание

Возвращает элемент на том же уровне, что и текущий элемент, и заданное расстояние после него в функции агрегирования.

Синтаксис

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

Группа функций

Набор

Данные ввода

Параметр	Описание	Тип	Обязательный
member	Любой элемент	member	Да
distance	Расстояние элемента от текущего элемента	int	Да

Примечания

- Lag не применяется в качестве отдельной функции. Эта функция используется в качестве входного параметра для функций агрегирования, задающего набор элементов для агрегирования.
- Если distance положительное число, функция Lag возвращает distance позиций после member. Если distance отрицательное число, функция Lag возвращает distance позиций перед member.
- member— текущий элемент иерархии. Если иерархия не является контекстом для блока, формула вернет пустое значение.
- Функция Lag использует порядок элементов в иерархии и запросе для возвращения связанного элемента.

Примеры

Требуется получить значения разниц в объемах интернет-продаж от недели к неделе.

```
=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

Или требуется сравнить определенный год с другим годом за два года до него:

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

Затем следует сочетать функции Lag и IsLeaf, чтобы определить разницу проданного объема за одну неделю. Формула, указанная в последнем столбце:

```
=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

Связанные сведения

[Агрегирование \[страница 46\]](#)

[Average \[страница 53\]](#)

[Количество \[страница 54\]](#)

[Функция Max \[страница 61\]](#)

[Функция Min \[страница 63\]](#)

[Функция Sum \[страница 84\]](#)

1.6.1.9.8 MemberAtDepth

Описание

Возвращает элемент иерархии с заданной глубиной.

Группа функций

Набор

Синтаксис

```
string MemberAtDepth(hierarchy;depth)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
hierarchy	Иерархический объект	измерение	Да
depth	Уровень набора элементов в выбранной иерархии.	int	Да

Примечания

- Параметр `depth` должен иметь положительное или нулевое значение, иначе функция возвращает ошибку.
- Если элементы с заданной глубиной не обнаружены, функция `MemberAtDepth()` возвращает значение `Null`.
- Если объект ввода не является иерархией, функция `MemberAtDepth()` возвращает значение `Null` для любого значения глубины больше нуля.

Примеры

Следующая иерархия [Country] отфильтрована так, чтобы она содержала только два дочерних элемента узла EUROPE.

Country	Quantity
WORLD	262,461
EUROPE	31,009
France	6,965
Germany	6,331
NORTH_AMERICA	219,944
Canada	17,754
USA	202,190
ASIA_PAC	9,065

MemberAtDepth([Country];0) возвращает:

	Order Quantity
WORLD	755,719

"WORLD" – это корень и единственное значение на уровне 0.

Если параметр *Не допускать агрегирования повторяющихся строк* отключен, функция агрегирует значения всех элементов, так как все они исходят из корня "WORLD". Если нужно проверить все агрегируемые значения, включите параметр *Не допускать агрегирования повторяющихся строк*. Тогда иерархия будет иметь следующий вид:

	Order Quantity
WORLD	262,461
WORLD	31,009
WORLD	6,965
WORLD	6,331
WORLD	219,944
WORLD	17,754
WORLD	202,190
WORLD	9,065
Sum:	755,719

MemberAtDepth([Country];1) возвращает:

	Order Quantity
	262,461
ASIA_PAC	9,065
EUROPE	44,305
NORTH_AME	439,888

Первая строка содержит значение NULL, так как корень не содержит значений на уровне 1.

В других строках функция агрегирует значения всех элементов иерархии [Country] на уровне 1: "ASIA_PAC", "EUROPE" и "NORTH_AMERICA". Если нужно проверить все агрегируемые значения,

включите параметр *Не допускать агрегирования повторяющихся строк*. Тогда иерархия будет иметь следующий вид:

	Order Quantity
	262,461
ASIA_PAC	9,065
EUROPE	31,009
EUROPE	6,965
EUROPE	6,331
NORTH_AME	219,944
NORTH_AME	17,754
NORTH_AME	202,190
Sum:	755,719

MemberAtDepth([Country]; 2) возвращает:

	Order Quantity
	522,479
Canada	17,754
France	6,965
Germany	6,331
USA	202,190

Обратите внимание, что первая строка агрегирует все элементы, не имеющие значений на уровне 2, то есть корень и каждый узел. В других строках агрегируются значения всех элементов иерархии [Country] на уровне 2.

MemberAtDepth([Country]; 3) возвращает:

	Order Quantity
	755,719

Остается только одна строка, со значением NULL, поскольку в иерархии нет третьего уровня. Следовательно, агрегируются значения всех узлов и конечных узлов иерархии.

Если нужно просмотреть всю иерархию, добавьте столбцы, содержащие уровни иерархии, в существующую таблицу, а затем воспользуйтесь функцией и передайте ей значения различных уровней. Тогда она будет иметь следующий вид:

Level 0	Level 1	Level 2	Order Quantity
WORLD			262,461
WORLD	ASIA_PAC		9,065
WORLD	EUROPE		31,009
WORLD	EUROPE	France	6,965
WORLD	EUROPE	Germany	6,331
WORLD	NORTH_AMERICA		219,944
WORLD	NORTH_AMERICA	Canada	17,754
WORLD	NORTH_AMERICA	USA	202,190

Можно воспользоваться формулой `IsLeaf` для фильтрации конечных узлов иерархии в вышеуказанной таблице: Дополнительные сведения о формуле `IsLeaf` см. в разделе [Функция IsLeaf](#) [страница 221].

Level 0	Level 1	Level 2	Order Quantity	IsLeaf
WORLD			262,461	false
WORLD	ASIA_PAC		9,065	true
WORLD	EUROPE		31,009	false
WORLD	EUROPE	France	6,965	true
WORLD	EUROPE	Germany	6,331	true
WORLD	NORTH_AMERICA		219,944	false
WORLD	NORTH_AMERICA	Canada	17,754	true
WORLD	NORTH_AMERICA	USA	202,190	true

Когда это сделано, можно скрыть столбец `IsLeaf`, чтобы получить эквивалент таблицы плоской иерархии:

Level 0	Level 1	Level 2	Order Quantity
WORLD	ASIA_PAC		9,065
WORLD	EUROPE	France	6,965
WORLD	EUROPE	Germany	6,331
WORLD	NORTH_AMERICA	Canada	17,754
WORLD	NORTH_AMERICA	USA	202,190

1.6.1.9.9 Функция Parent

Описание

Возвращает дочерний элемент для элемента иерархии в функции агрегирования.

Группа функций

Набор

Синтаксис

```
member member.Parent
```

Данные ввода

Параметр	Описание	Тип	Обязательный
member	Любой элемент	member	Да

Примечания

- `Parent` не используется в качестве отдельной функции. Эта функция используется в качестве входного параметра для функций агрегирования, задающего набор элементов для агрегирования.
- `member` — текущий элемент иерархии. Если иерархия не является контекстом для блока, формула вернет пустое значение.

Примеры

Второй столбец содержит формулу, которая позволяет получить родительский элемент каждого элемента иерархии:

```
=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

Связанные сведения

[Агрегирование \[страница 46\]](#)

[Average \[страница 53\]](#)

[Количество \[страница 54\]](#)

[Функция Max \[страница 61\]](#)

[Функция Min \[страница 63\]](#)

[Функция Sum \[страница 84\]](#)

1.6.1.9.10 Функция Siblings

Описание

Возвращает элемент и элементы с общим родителем для элемента иерархии в функции агрегирования.

Группа функций

Набор

Синтаксис

```
member_set member.Siblings
```

Данные ввода

Параметр	Описание	Тип	Обязательный
member	Любой элемент	member	Да

Примечания

- `Siblings` — элементы с общим родителем; не используется в качестве отдельной функции. Эта функция используется в качестве входного параметра для функций агрегирования, задающего набор элементов для агрегирования.
- `member` — текущий элемент иерархии. Если иерархия не является контекстом для блока, формула вернет пустое значение.
- Элементы с общим родителем – это элементы того же уровня и с тем же общим родителем, что и у элемента.

Примеры

Существует иерархия времени, требуется получить процентное соотношение каждого квартала в году или процентное соотношение каждого года в периоде.

```
=[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%

В ячейке произвольного формата требуется получить значение доли 2004 года за весь период:

```
=Sum([Query 1].[Internet Sales].[Internet Sales Amount];{[Query 1].[Calendar].[Date.Calendar]&[All Periods].[CY 2004]}) / Sum([Query 1].[Internet Sales].[Internet Sales Amount];{[Query 1].[Calendar].[Date.Calendar]&[All Periods].[CY 2004].Siblings()})
```

{CY 2001;CY 2002}	2004 percentage in 2001 to 2004 period
9,796,717.18	33.28%

Date.Calendar	Internet Sales Amount	
[-] All Periods	29,358,677.22	
[-] CY 2001	3,266,373.66	11.13%
[+] H2 CY 2001	3,266,373.66	100.00%
[-] CY 2002	6,530,343.53	22.24%
[+] H1 CY 2002	3,805,710.59	58.28%
[+] H2 CY 2002	2,724,632.94	41.72%
[-] CY 2003	9,791,060.3	33.35%
[+] H1 CY 2003	3,037,501.36	31.02%
[+] H2 CY 2003	6,753,558.94	68.98%
[+] CY 2004	9,770,899.74	33.28%

Связанные сведения

[Агрегирование \[страница 46\]](#)

[Average \[страница 53\]](#)

[Количество \[страница 54\]](#)

[Функция Max \[страница 61\]](#)

[Функция Min \[страница 63\]](#)

[Функция Sum \[страница 84\]](#)

1.6.1.10 Дополнительные функции

1.6.1.10.1 Функция BlockName

Описание

Возвращает имя блока

Группа функций

Разные

Синтаксис

```
string BlockName()
```

Примеры

BlockName () возвращает "Блок 1", если он помещен в блок под названием "Блок 1".

1.6.1.10.2 ClosingPeriod

Описание

Возвращает показатель на последнюю дату периода в текущем контексте и в диапазоне времени, определенном в измерении времени.

Группа функций

Разные

Синтаксис

```
input_type ClosingPeriod(measure;timeperiod)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
measure	Любой показатель или переменная.	Показатель	Да
timeperiod	Период времени, предоставляющий контекст расчета в блоке.	Период времени	Да

📘 Примечание

- Объект времени должен быть периодом времени, доступным в блоке. Если в блоке нет периода времени, функция возвращает ошибку #COMPUTATION.
- Фильтры отчета по измерению времени могут влиять на результаты функции. Функцию можно комбинировать с функцией NoFilter, чтобы игнорировать фильтры отчета в вычислении функции.

Примеры

Столбец ClosingPeriod в таблице ниже содержит следующую формулу:

```
ClosingPeriod([Revenue]; [Time Dimension].[Year])
```

Год	Доход	ClosingPeriod
2016	1000	2000
2017	2000	2000

```
ClosingPeriod([Revenue]; [Time Dimension].[Semester])
```

Год	Полугодие	Доход	ClosingPeriod
2016	H1 2016	400	1500
2016	H2 2016	600	1500
2017	H1 2017	500	1500
2017	H2 2017	1500	1500

Связанные сведения

[#COMPUTATION \[страница 312\]](#)

1.6.1.10.3 Функция ColumnNumber

Описание

Возвращает порядковый номер столбца

Группа функций

Разные

Синтаксис

```
int ColumnNumber()
```

Примеры

ColumnNumber () возвращает 2, если данная формула помещена во второй столбец таблицы.

1.6.1.10.4 Комментарий

Описание

Возвращает комментарий к ячейке

Группа функций

Разное

Синтаксис

```
string Comment()
```

Примечание

Функция возвращает первый или последний комментарий в ячейке, в зависимости от параметров, указанных в [свойствах документа](#).

Пример

`Comment()` возвращает "Увеличение маржи брутто в третьем квартале", если ячейка содержит комментарий "Увеличение маржи брутто в третьем квартале".

1.6.1.10.5 Функция CurrentUser

Описание

Возвращает параметры входа на стартовую панель BI для текущего пользователя

Группа функций

Разные

Синтаксис

```
строка CurrentUser()
```

Примеры

`CurrentUser()` возвращает "gkn", если текущее регистрационное имя пользователя – "gkn".

1.6.1.10.6 CustomProperties

Описание

Возвращает список определенных пользовательских свойств.

Группа функций

Разные

Синтаксис

```
string CustomProperties ()
```

ⓘ Примечание

- Если в документе нет пользовательского свойства, функция возвращает пустую строку.
- Если в документе существует несколько пользовательских свойств, они разделяются точкой с запятой в строке, возвращаемой функцией.

Примеры

Если документ содержит два пользовательских свойства с именами Lines и Category, то:

`CustomProperties()` возвращает "Lines;Category"

1.6.1.10.7 CustomPropertyValue

Описание

Возвращает значение определенного пользовательского свойства

Группа функций

Разные

Синтаксис

```
string CustomPropertyValue ( custom_property )
```

Данные ввода

Параметр	Описание	Тип	Обязательный
custom_property	Имя пользовательского свойства	Строка	Да

Примечание

- Если нет пользовательского свойства, функция возвращает пустую строку.

Примеры

Если документ содержит два пользовательских свойства Lines со значением Dresses и Category со значением Skirts, то:

- CustomPropertyValue("Lines") возвращает "Dresses"
- CustomPropertyValue("Category") возвращает "Skirts"
- CustomPropertyValue("Color") возвращает ""

1.6.1.10.8 DescriptionOf

Описание

Возвращает описание объекта.

Группа функций

Разные

Синтаксис

```
string DescriptionOf(obj)
```

Входной

Параметр	Описание	Тип	Обязательный
obj	Объект отчета	Report object	Да

Примечания

- Если описание не задано, возвращается пустая строка.
- Описание возвращается с использованием предпочтительного языкового стандарта для просмотра. Если описание не определено в предпочтительном языковом стандарте для просмотра, оно может быть возвращено в резервном языковом стандарте.

1.6.1.10.9 Функция ForceMerge

Описание

Включает синхронизированные измерения в вычисления мер, если эти измерения не содержатся в контексте вычисления меры

Группа функций

Разные

Синтаксис

```
num ForceMerge(measure)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
measure	Любая мера	Показатель	Да

Данные вывода

Результат вычисления с учетом синхронизированных измерений

Примечания

- `ForceMerge` возвращает значение #НЕОДНОЗНАЧНО, если функция применяется к интеллектуальной мере, поскольку набора для группировки, необходимого для вычисления интеллектуальной меры, не существует.
- Функция `ForceMerge` эквивалентна функции `Multicube` приложения `BusinessObjects/Desktop Intelligence`.

Примеры

`ForceMerge([Доход])` возвращает значение объекта `[Доход]` с учетом всех синхронизированных измерений, которые не отображаются в одном блоке с мерой `[Доход]`.

1.6.1.10.10 FormulaOf

Описание

Возвращает формулу, определяющую переменную, если объект является переменной. Если объект не является переменной, возвращает пустую строку.

Группа функций

Разные

Синтаксис

```
string FormulaOf(obj)
```

Входной

Параметр	Описание	Тип	Обязательный
obj	Объект отчета	Report object	Да

Примечания

- Формула возвращается с использованием предпочтительного языкового стандарта для просмотра.

1.6.1.10.11 Функция GetContentLocale

Описание

Возвращает языковой стандарт данных, содержащихся в документе (языковой стандарт документа)

Группа функций

Разные

Синтаксис

```
string GetContentLocale( )
```

Примечания

Языковой стандарт используется для форматирования данных в документе.

Примеры

Функция `GetContentLocale()` возвращает "fr_FR", если в качестве языкового стандарта документа установлен "Французский (Франция)".

1.6.1.10.12 Функция `GetDominantPreferredViewingLocale`

Описание

Возвращает основной языковой стандарт в группе предпочтительного языкового стандарта для просмотра, выбранного пользователем

Группа функций

Разные

Синтаксис

```
string GetDominantPreferredViewingLocale()
```

Примечания

- Каждая группа связанных языковых стандартов имеет основной языковой стандарт, используемый в качестве базового для всех остальных языковых стандартов в данной группе. Например, английский (США) ("en_US") является основным языковым стандартом в группе стандартов английского языка. Английский (Новая Зеландия) также входит в эту группу.
- В *Руководстве к диспетчеру переводов* перечислены все основные предпочтительные языковые стандарты для просмотра.

Примеры

Функция `GetDominantPreferredViewingLocale` возвращает "en_US", если в качестве предпочтительного языкового стандарта для просмотра установлен "Английский (Новая Зеландия)".

Связанные сведения

[Функция GetPreferredViewingLocale \[страница 248\]](#)

1.6.1.10.13 Функция GetLocale

Описание

Возвращает пользовательский языковой стандарт, применяемый для форматирования пользовательского интерфейса (языковой стандарт продукта)

Группа функций

Разные

Синтаксис

```
string GetLocale()
```

Примечания

Языковой стандарт продукта – это языковой стандарт, используемый для отображения пользовательского интерфейса (например, элементов меню и текста кнопок).

Примеры

Функция `GetLocale()` возвращает "en_US", если в качестве пользовательского языкового стандарта продукта установлен "Английский (США)".

1.6.1.10.14 Функция GetLocalized

Описание

Возвращает локализованную строку в соответствии с предпочтительным языковым стандартом для просмотра, выбранным пользователем.

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
string	Строка для перевода	Строка	Да
comment	Комментарий, добавляемый переводчиками	Строка	Нет

Примечания

- Параметр `string` может быть строкой в любой формуле (например, в ячейке, сообщении предупреждения или определении переменной).
- При создании отчета можно использовать параметр `comment` для ввода дополнительной информации, помогающей переводчикам перевести строку. Этот комментарий появляется вместе со строкой в инструменте "Диспетчер переводов", используемом переводчиками для перевода отчетов.
- Каждая пара строка + комментарий создает отдельную строку для перевода в инструменте "Диспетчер переводов". В результате `GetLocalized("Итого по продукту"; "Максимум 20 символов")` и `GetLocalized("Итого по продукту"; "Использовать не более 20 символов")` могут возвращать различные переводы.

Примеры

Функция `GetLocalized("Итого для всех продуктов")` возвращает перевод на французский язык строки "Итого для всех продуктов", если в качестве предпочтительного языкового стандарта для просмотра установлен "fr_FR".

Функция `GetLocalized("Итого для всех продуктов", "Старайтесь не использовать больше 20 символов")` возвращает перевод на немецкий язык строки "Итого для всех продуктов", если в качестве предпочтительного языкового стандарта для просмотра установлен "de_DE". Эта функция также сообщает переводчику отчета, что не рекомендуется использовать больше 20 символов при переводе этой строки.

Связанные сведения

[Функция `GetPreferredViewingLocale` \[страница 248\]](#)

1.6.1.10.15 Функция `GetPreferredViewingLocale`

Описание

Возвращает пользовательский предпочтительный языковой стандарт для просмотра данных в документе (предпочтительный языковой стандарт для просмотра)

Группа функций

Разные

Синтаксис

```
string GetPreferredViewingLocale()
```

Примеры

Функция `GetPreferredViewingLocale` возвращает "en_US", если в качестве предпочтительного языкового стандарта для просмотра установлен "Английский (США)".

Связанные сведения

[Функция `GetLocalized` \[страница 247\]](#)

1.6.1.10.16 Функция If...Then...Else

Описание

Возвращает значение в зависимости от того, является ли значение выражения TRUE или FALSE

Группа функций

Разные

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
bool_value	Логическое значение	Логическое значение	Да
true_value	Возвращаемое значение в случае, если bool_value = true	Любой	Да
false_value	Возвращаемое значение, если bool_value имеет значение False	Любой	Да, если Else включено

Примечания

- true_value и false_value могут смешивать типы данных.
- Можно использовать операторы логических выражений And, Between, InList, Or и Not с функцией If.

- Условия If можно вкладывать в другие условия путем замены оператора Else оператором ElseIf. В следующем синтаксисе приведен пример одноуровневого вложения:

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

- Также поддерживается исходный синтаксис функции If:
If (bool_value; true_value; false_value).

Примеры

If [Sales Revenue] > 1000000 Then "High Revenue" возвращает "High Revenue" для всех строк, где значение дохода превышает 1 000 000; для остальных строк ничего не возвращается.

If [Sales Revenue] > 1000000 Then "High Revenue" Else [Revenue] возвращает "High Revenue" для всех строк, где значение дохода превышает 1 000 000, для всех остальных строк просто выводится значение дохода.

If [Sales Revenue] > 1000000 Then "High Revenue" Else "Low Revenue" возвращает "High Revenue" для всех строк, где значение дохода превышает 1 000 000, и "Low Revenue" для строк, где значение дохода не достигает 1 000 000.

If [Sales Revenue] > 1000000 Then "Высокий доход" ElseIf [Sales Revenue] > 800000 Then "Средний доход" Else "Низкий доход" возвращает "Высокий доход" для строк, где значение дохода превышает 1000000, "Средний доход" для строк, где значение дохода находится в пределах 800000 и 1000000 и "Низкий доход" во всех остальных случаях.

Связанные сведения

[Функция If \[страница 250\]](#)

[Оператор And \[страница 273\]](#)

[Оператор Between \[страница 275\]](#)

[Оператор InList \[страница 276\]](#)

[Оператор Or \[страница 274\]](#)

[Оператор Not \[страница 275\]](#)

1.6.1.10.17 Функция If

Описание

Возвращает значение в зависимости от того, является ли значение выражения TRUE или FALSE

Группа функций

Разные

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
bool_value	Логическое значение	Логическое значение	Да
true_value	Возвращаемое значение, если bool_value имеет значение True	Любой	Да
false_value	Возвращаемое значение, если bool_value имеет значение False	Любой	Да

Примечания

- true_value и false_value могут смешивать типы данных.
- Условия If можно вкладывать в другие условия путем замены параметра false_value дополнительными условиями If. В этом синтаксисе приведен пример одноуровневого вложения:

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

- Также поддерживается синтаксис If...Then...Else.

Примеры

Функция `If ([Доход с продаж]>1000000;"Высокий доход";"Низкий доход")` возвращает "Высокий доход" для всех строк, где значение дохода превышает 1 000 000, и "Низкий доход" для тех строк, где значение дохода не достигает 1 000 000.

Функция `If ([Доход с продаж]>1000000;"Высокий доход";[Доход])` возвращает "Высокий доход" для всех строк, где значение дохода превышает 1 000 000, для всех остальных строк просто выводится значение дохода.

Связанные сведения

[Функция If...Then...Else \[страница 249\]](#)

1.6.1.10.18 Функция LineNumber

Описание

Возвращает номер строки в таблице

Группа функций

Разные

Синтаксис

```
int LineNumber( )
```

Примечания

Нумерация строк таблицы начинается с заголовка, который является строкой за номером 1.

Примеры

`LineNumber()` возвращает 2, если функция отображается на второй строке в таблице.

1.6.1.10.19 Next

Описание

Возвращает следующее значение объекта.

Группа функций

Разные

Синтаксис

```
input_type Next(dimension|measure [;Row|Col][;reset_dims][;offset][;NotNull])
```

Входной

Параметр	Описание	Тип	Обязательный
dimension/measure	Это измерение или мера, следующее значение которой возвращает функция.	Dimension или measure	Да
Row/col	Задаёт направление вычисления. Используется в кросс-таблице, чтобы определить, является ли следующее возвращаемое значение значением в строке или столбце.	Keyword	Нет
reset_dims	Список измерений, используемых для сброса вычисления.	Dimension list	Нет
offset	Возвращает следующее значение, которое является строкой <code>offset</code> после текущей строки.	Integer	Нет (по умолчанию 1)
NotNull	Указывает функции на необходимость возвращения первого значения, не равного NULL, начиная со смещения.	Keyword	Нет

Примечания

- Функция `Next` несовместима с зависящими от отображения функциями, такими как `ColumnNumber`, `LineNumber`, `PageNumber`, `Page` и `PageInSection`. При использовании комбинации этих функций генерируется ошибка `#RECURSIVE`. В качестве обходного решения замените функции

ColumnNumber и LineNumber текущей суммой. Для PageNumber, Page и PageInSection обходного решения не существует.

- По умолчанию для offset используется значение 1. Next ([Revenue] ; 1) и Next ([Revenue]) функционально одинаковы.
- В случае включения аргумента nonNull функция возвращает первое значение, не являющееся NULL, объекта, начиная с ячейки строк offset, предшествующих текущей строке и отсчитанных в обратном направлении.
- С функцией Next можно использовать контекстные операторы расширенного синтаксиса.
- Измерения нужно всегда помещать в круглые скобки, даже если в списке измерений сброса присутствует только одно измерение.
- Измерения сброса в наборе должны разделяться точками с запятой.
- Функция Next применяется после применения всех фильтров отчета, раздела и блокировки, а также после всех сортировок.
- Применение сортировок и фильтров в формулах, использующих функцию Next, невозможно.
- Если функция Next применяется к мере, а мера возвращает неопределенное значение, функция Next возвращает неопределенное значение, даже если в предыдущей строке возвращается значение.
- При помещении за пределами верхнего или нижнего колонтитула разрыва функция Next игнорирует разрывы.
- Если функция Next помещена в нижний колонтитул разрыва, то она возвращает значение из предыдущего экземпляра нижнего колонтитула.
- Функция Next сбрасывается в каждом разделе отчета.
- При использовании в кросс-таблице функция Next не обрабатывает последнее значение в строке как предыдущее значение для первого значения в следующей строке.

1.6.1.10.20 Функция NoFilter

Описание

Игнорирует фильтры при вычислении значения. Функция "NoFilter" используется с объектами показателей. Она не подходит для работы с измерениями.

Группа функций

Разные

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
obj	Любой объект отчета	Объект отчета	Да
All Drill	<ul style="list-style-type: none">Нет указанного ключевого слова – игнорировать отчет и блокировать фильтрыAll – игнорировать все фильтрыDrill – игнорировать фильтры отчета и фильтры детализации	Ключевое слово	Нет

Примечания

- NoFilter(obj;Drill) не работает в режиме детализации запроса, т. к. фильтры детализации добавляются в запрос, но не применяются для фильтра данных отчета.
- Если работа в режиме детализации завершена, а фильтры детализации остались активны, они становятся фильтрами отчета и могут изменять значения любых выражений, на которые распространяется действие функции NoFilter(obj;Drill).

Примеры

Если поместить функцию NoFilter(Sum([Доходы с продаж])) в нижний колонтитул блока, она возвращает итоговый объем доходов с продаж по всем строкам блока, даже если применяемый фильтр исключает некоторые строки из блока.

Функция NoFilter(Sum([Доходы с продаж]);All) возвращает общий объем доходов с продаж для всех стран, включая Францию, несмотря на то, что применяемый фильтр исключает Францию из отчета.

Функция NoFilter(Sum([Доходы с продаж]);Drill) возвращает общий объем доходов с продаж для всех стран, игнорируя при этом фильтр детализации по измерению [Страна].

1.6.1.10.21 Функция NumberOfPages

Описание

Возвращает количество страниц в отчете

Группа функций

Разные

Синтаксис

```
integer NumberOfPages()
```

Примечания

Если функция `NumberOfPages` помещается в ячейку с установленными свойствами "Автоматический подбор высоты" или "Автоматический подбор ширины", ячейка возвращает значение `#RECURSIVE`, так как в этом случае возникает циклическая зависимость. Чтобы вернуть значение, этой функции нужен точный размер отчета, но на размер отчета влияет размер ячейки, который определяется содержимым ячейки.

Примеры

Примеры `NumberOfDataPages()` возвращает 2, если в отчете содержатся две страницы.

1.6.1.10.22 OpeningPeriod

Описание

Возвращает показатель на первую дату периода в текущем контексте и в диапазоне времени, определенном в измерении времени.

Группа функций

Разные

Синтаксис

```
input_type OpeningPeriod(measure;timeperiod)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
measure	Любой показатель или переменная.	Показатель	Да
timeperiod	Период времени, предоставляющий контекст расчета в блоке.	Период времени	Да

📘 Примечание

- Объект времени должен быть периодом времени, доступным в блоке. Если в блоке нет периода времени, функция возвращает ошибку #COMPUTATION.
- Фильтры отчета по измерению времени могут влиять на результаты функции. Функцию можно комбинировать с функцией NoFilter, чтобы игнорировать фильтры отчета в вычислении функции.

Примеры

Столбец OpeningPeriod в таблице ниже содержит следующую формулу:

```
OpeningPeriod([Revenue]; [Time Dimension].[Year])
```

Год	Доход	OpeningPeriod
2016	1000	1000
2017	2000	1000

```
OpeningPeriod([Revenue]; [Time Dimension].[Semester])
```

Год	Полугодие	Доход	OpeningPeriod
2016	H1 2016	400	400
2016	H2 2016	600	400
2017	H1 2017	500	400
2017	H2 2017	1500	400

Связанные сведения

[#COMPUTATION](#) [страница 312]

1.6.1.10.23 Функция Page

Описание

Возвращает номер текущей страницы в отчете.

Группа функций

Разные

Синтаксис

```
integer Page()
```

Примечания

Если функция Page помещается в ячейку с установленными свойствами "Автоматический подбор высоты" или "Автоматический подбор ширины", ячейка возвращает значение #RECURSIVE, так как в этом случае возникает циклическая зависимость. Чтобы вернуть значение, этой функции нужен точный размер отчета, но на размер отчета влияет размер ячейки, который определяется содержимым ячейки.

Пример

Page () возвращает 2, если отображается на второй странице отчета.

1.6.1.10.24 PageInSection

Описание

Возвращает количество страниц в текущей инстанции раздела в указанном разделе

Группа функций

Разное

Синтаксис

```
integer PageInSection([section_level])
```

Входной

параметр	Описание	Тип	Обязательный
section_level	Иерархический уровень раздела	Целое число	Нет

Примечания

- Если функция `PageInSection` помещается в ячейку с установленными свойствами "Автоматический подбор высоты" или "Автоматический подбор ширины", ячейка возвращает значение `#RECURSIVE`, так как в этом случае возникает циклическая зависимость. Чтобы вернуть значение, этой функции нужен точный размер отчета, но на размер отчета влияет размер ячейки, который определяется содержимым ячейки.
- `PageInSection()` должно быть в пределах инстанции раздела. В противном случае возвращается значение 0.
- Иерархические уровни разделов начинаются с 1 (верхний уровень).
- Если `section_level` не указан, функция возвращает количество в текущем уровне раздела.
- Если указан несуществующий уровень раздела, то функция возвращает 0.
- Переменные, созданные с помощью этой функции, должны быть показателями.

Примеры

В документе с разделом Year (= 2010, 2011, 2012) и подразделом State (= California, Florida, Texas):

- повторение `PageInSection(1)` в разделе Year возвращает 2 на второй странице для 2010, 2011 и 2012.
- повторение `PageInSection(2)` в подразделе State возвращает 1 на первой странице для California, Florida и Texas.

1.6.1.10.25 ParallelPeriod

Описание

Возвращает показатель на дату периода параллельно с датами в блоке в текущем контексте. Выбранные даты смещаются на ряд интервалов вперед или назад во времени.

Группа функций

Разные

Синтаксис

```
input_type ParallelPeriod(measure;timeperiod;offset)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
measure	Любой показатель или переменная.	Показатель	Да
timeperiod	Период времени, предоставляющий контекст расчета в блоке.	Период времени	Да
offset	Указывает число интервалов вперед или назад во времени.	Целое число	Да

❗ Примечание

- Объект времени должен быть периодом времени, доступным в блоке. Если в блоке нет периода времени, функция возвращает ошибку #COMPUTATION.
- Функция не зависит от порядка сортировки измерения времени в блоке.
- Фильтры отчета по измерению времени могут влиять на результаты функции. Функцию можно комбинировать с функцией NoFilter, чтобы игнорировать фильтры отчета в вычислении функции.

Примеры

Столбец ParallelPeriod в таблице ниже содержит следующую формулу:

```
ParallelPeriod([Revenue]; [Time Dimension].[Year];-1)
```

Год	Доход	ClosingPeriod
2015	600	-
2016	1000	600
2017	2000	2000

```
ParallelPeriod ([Revenue]; [Time Dimension].[Semester];-1)
```

Год	Полугодие	Доход	ParallelPeriod
2015	H1 2015	200	
2015	H2 2015	400	200
2016	H1 2016	400	400
2016	H2 2016	600	400
2017	H1 2017	500	600
2017	H2 2017	1500	500

Связанные сведения

[#COMPUTATION \[страница 312\]](#)

1.6.1.10.26 PeriodToDate

Описание

Возвращает текущую сумму значений показателя за временной период по дате в текущем контексте. Например, пользователь может выбрать "Год", чтобы просмотреть значения года нарастающим итогом (YTD) для каждого месяца рядом с месячными значениями.

Группа функций

Разные

Синтаксис

```
input_type PeriodToDate(measure;timeperiod;[Sum|Max|Min|Count|Average|Product])
```

Данные ввода

Параметр	Описание	Тип	Обязательный
measure	Любой показатель или переменная.	Показатель	Да
timeperiod	Период времени, предоставляющий контекст расчета в блоке.	Период времени	Да
runningfunct	Выполняющаяся функция для применения. Возможные значения: <ul style="list-style-type: none">Сумма (по умолчанию)Макс.Мин.КоличествоСреднееПродукт	Перечисление	Нет

📘 Примечание

- Объект времени должен быть периодом времени, доступным в блоке. Если в блоке нет периода времени, функция возвращает ошибку #COMPUTATION.
- Если блок содержит другие измерения, последние из них будут неявно использоваться в качестве измерений сброса для выполняющейся функции.

- Если блок содержит другие периоды времени, которые отличаются от определенных в функции, текущая сумма будет применена к периоду времени с наименьшей гранулярностью времени.
- Функция не зависит от порядка сортировки измерения времени в блоке.
- Фильтры отчета по измерению времени могут влиять на результаты функции. Функцию можно комбинировать с функцией `NoFilter`, чтобы игнорировать фильтры отчета в вычислении функции.

Примеры

Столбец `PeriodToDate` в таблице ниже содержит следующую формулу:

```
PeriodToDate([Revenue]; [Time Dimension].[Year])
```

Год	Полугодие	Доход	ClosingPeriod
2015	H1 2015	200	200
2015	H2 2015	400	600
2016	H1 2016	400	400
2016	H2 2016	600	1000
2017	H1 2017	500	500
2017	H2 2017	1500	2000

Год	Полугодие	Семейство продук- тов	Доход	PeriodToDate
2015	H1 2015	Продукты питания	50	50
2015	H1 2015	Электроника	150	150
2015	H2 2015	Продукты питания	100	150
2015	H2 2015	Электроника	300	450
2016	H1 2016	Продукты питания	150	150
2016	H1 2016	Электроника	250	250
2016	H2 2016	Продукты питания	200	350
2016	H2 2016	Электроника	400	650
2017	H1 2017	Продукты питания	200	200
2017	H1 2017	Электроника	300	300
2017	H2 2017	Продукты питания	500	700
2017	H2 2017	Электроника	1000	1300

Связанные сведения

[#COMPUTATION \[страница 312\]](#)

1.6.1.10.27 Функция Previous

Описание

Возвращает предыдущее значение объекта

Группа функций

Разные

Синтаксис

```
input_type Previous(измерение|мера|Self [;Row|col][;(reset_dims)][;offset]
[;NotNull])
```

Входные данные

Параметр	Описание	Тип	Обязательный
измерение мера Self	Измерение или мера, чье предыдущее значение возвращает функция, или ключевое слово Self	Измерение, мера или ключевое слово	Да
Row/Col	Устанавливает направление вычисления	Ключевое слово	Нет
reset_dims	Список измерений, используемых для сброса вычислений	Список измерений	Нет
offset	Указывает значение измерения или меры, то есть строки со смещением,	Целое	Нет (по умолчанию 1)

Параметр	Описание	Тип	Обязательный
	предшествующие текущей строке		
NotNull	Указывает функции на то, что необходимо возвращать первое ненулевое значение, начиная со смещения	Ключевое слово	Нет

Примечания

- Функция `Previous` несовместима с зависящими от отображения функциями, такими как `ColumnNumber`, `LineNumber`, `PageNumber`, `Page` и `PageInSection`. При использовании комбинации этих функций генерируется ошибка `#RECURSIVE`. В качестве обходного решения замените функции `ColumnNumber` и `LineNumber` текущей суммой. Для `PageNumber`, `Page` и `PageInSection` обходного решения не существует.
- По умолчанию для `offset` используется значение 1. `Previous([Revenue];1)` и `Previous([Revenue])` функционально одинаковы.
- В случае включения аргумента `NotNull` функция возвращает первое ненулевое значение объекта, начиная с ячейки строк смещения, предшествующих текущей строке и отсчитанных в обратном направлении.
- С функцией `Previous` (Предыдущее) можно использовать контекстные операторы расширенного синтаксиса
- Оператор `SELF` позволяет обращаться к предыдущему значению ячейки, если ее содержимое не является объектом отчета.
- Измерения нужно всегда помещать в круглые скобки, даже если в списке измерений сброса присутствует только одно измерение.
- Измерения сброса в наборе должны разделяться двоеточиями.
- Функция `Previous` применяется после применения всех фильтров отчета, раздела и блокировки, а также после всех сортировок.
- Применение сортировок и фильтров в формулах, использующих функцию `Previous`, невозможно.
- Если функция `Previous` применяется к мере, а мера возвращает неопределенное значение, функция `Previous` возвращает неопределенное значение, даже если в предыдущей строке возвращается значение.
- При помещении за пределами верхнего или нижнего колонтитула разрыва функция `Previous` игнорирует разрывы.
- Если функция `Previous` помещена в нижний колонтитул разрыва, то она возвращает значение из предыдущего экземпляра нижнего колонтитула.
- Функция `Previous` сбрасывается в каждом разделе отчета.
- При использовании в кросс-таблице, функция `Previous` не обрабатывает последнее значение в строке как предыдущее значение для первого значения в следующей строке.

Примеры

`Previous ([Country] ; 1)` возвращает следующие значения в следующей таблице:

Страна	Доход	Предыдущее
США	5 000 000	
Великобритания	2 000 000	США
Франция	2 100 000	Великобритания

`Previous ([Revenue])` возвращает следующие значения в следующей таблице:

Страна	Доход	Предыдущее
США	5 000 000	
Великобритания	2 000 000	5 000 000
Франция	2 100 000	2 000 000

`Previous ([Revenue] ; ([Country]))` возвращает следующие значения в следующей таблице:

Страна	Регион	Доход	Предыдущее
США	Север	5 000 000	
	Юг	7 000 000	5 000 000
Великобритания	Север	3 000 000	
	Юг	4 000 000	3 000 000

`Previous ([Revenue])` возвращает следующие значения в следующей кросс-таблице:

	2004	Предыдущее	2005	Предыдущее
США	5 000 000		6 000 000	5 000 000
Великобритания	2 000 000		2 500 000	2 000 000
Франция	3 000 000		2 000 000	3 000 000

`Previous ([Revenue])` возвращает следующие значения в следующей таблице с разделением по [Странам]:

Страна	Регион	Доход	Предыдущее
США	Север	5 000 000	
	Юг	7 000 000	5 000 000
США		12 000 000	
Великобритания	Север	3 000 000	7 000 000

	Юг	4 000 000	3 000 000
Великобритания		7 000 000	12 000 000

`Previous([Revenue]); 2;NotNull)` возвращает следующие значения в следующей таблице:

Год	Квартал	Доход	Предыдущее
2008	K1	500	
2008	K2		
2008	K3	400	500
2008	K4	700	500
2008	K1	300	400
2008	K2		700
2008	K3		300
2008	K4	200	300

`2*Previous(Self)` возвращает последовательность 2, 4, 6, 8, 10...

Связанные сведения

[Сравнение значений с помощью функции Previous \[страница 318\]](#)

[Оператор Self \[страница 285\]](#)

1.6.1.10.28 Функция RefValue

Описание

Возвращает опорное значение для объекта отчета, когда включено отслеживание данных

Группа функций

Разные

Синтаксис

```
input_type RefValue(obj)
```

Примеры

`RefValue([Самый эффективный регион])` возвращает "Юго-запад", если значение переменной [Самый эффективный регион] в опорных данных равно "Юго-запад".

`RefValue([Доход])` возвращает 1000, если значение меры [Доход] в опорных данных равно 1000.

Примечания

- Функция `RefValue()` используется как с объектами показателей, так и с объектами измерения. Однако при использовании с переменной, квалифицированной как измерение или как объект-описание, функция `RefValue()` возвращает текущие, а не ссылочные значения объекта. Чтобы получить ссылочные значения, необходимо квалифицировать переменную как показатель.
- При создании непосредственно в разделе, таблице, форме или диаграмме формула всегда будет квалифицирована как показатель, поэтому в случае, когда формула использует функцию `RefValue()`, будут возвращены ожидаемые ссылочные значения.

Пример функции RefValue с переменной

Допустим, есть список значений измерения [State] (штат): California, Florida, Texas, New York. После обновления список имеет следующий вид: Arizona, California, Florida, Texas, New York. Переменная вида `Variable=RefValue([State])` вернет следующее значение:

Переменная квалифициро- вана как	Список возвращенных значений
Измерение или объект-описание	Arizona, California, Florida, Texas, New York
Показатель	(нулевое значение), California, Florida, Texas, New York

1.6.1.10.29 Функция RelativeValue

Описание

Возвращает предыдущие или последующие значения объекта

Группа функций

Разные

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
measure detail	Любая мера или сведения измерения в блоке	Мера или сведения	Да
slicing_dims	Измерения, которые служат контекстом вычислений	Список измерений	Да
offset	Указывает значение меры или сведений , которая является строкой смещения , удаленной из текущей строки	Целое	Да

Примечания

- Объект должен быть мерой или сведениями измерения, доступной в блоке.
- Порядок сортировки списка значений измерений срезов используется для определения выходных данных функции.
Порядок сортировки определяется двумя факторами: сортировкой, применяемой к измерениям срезов, и порядком, в котором измерения срезов перечислены в функции.
- Измерение, которое используется как начальный элемент раздела, может быть указано как измерение среза.
- Все измерения срезов должны быть представлены в блоке или ячейке раздела блока, где находится функция. Если измерение срезов затем удаляется из блока, функция возвращает ошибку #COMPUTATION.
- Если смещение превышает количество строк в списке значений измерения срезов, функция возвращает нулевое значение.
- `RelativeValue` невозможно использовать рекурсивно.
- Необходимо всегда заключать измерения в круглые скобки, даже если в списке измерений срезов есть только одно измерение.

Примеры

Столбец RelativeValue в таблице ниже содержит следующую формулу.

```
RelativeValue([Доход];([Год]);-1)
```

Год	Квартал	Продавец	Доход	RelativeValue
2007	K1	Смит	1000	
2007	K2	Джонс	2000	
2007	K3	Уилсон	1500	
2007	K4	Харрис	3000	
2008	K1	Смит	4000	1000
2008	K2	Джонс	3400	2000
2008	K3	Уилсон	2000	1500
2008	K4	Харрис	1700	3000

Связанные сведения

[#COMPUTATION \[страница 312\]](#)

[Сравнение значений с помощью функции RelativeValue \[страница 318\]](#)

1.6.1.10.30 Функция ReportName

Описание

Возвращает имя отчета

Группа функций

Разные

Синтаксис

```
string ReportName()
```

Примеры

Функция `ReportName()` возвращает значение "Отчет о продажах", если находится в отчете под названием "Отчет о продажах".

1.6.1.10.31 Индекс `RowIndex`

Описание

Возвращает номер строки

Группа функций

Разные

Синтаксис

```
integer RowIndex()
```

Примечания

- Нумерация строк начинается с 0.
- Функция `RowIndex` возвращает значение `#MULTIVALUE` при ее размещении в верхнем или нижнем колонтитуле таблицы.

Примеры

Функция `RowIndex` возвращает 0, если расположена в первой строке таблицы.

1.6.1.10.32 Функция UniqueNameOf

Описание

Возвращает уникальное имя объекта

Группа функций

Разные

Синтаксис

```
string UniqueNameOf(obj)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
obj	Любой объект отчета	Объект отчета	Да

Примеры

UniqueNameOf ([Дата сохранения]) возвращает "Дата сохранения".

1.6.2 Операторы функций и формул

Операторы связывают различные элементы формул.

Формулы могут содержать математические, условные, логические, специальные функциональные операторы или операторы расширенного синтаксиса.

1.6.2.1 Математические операторы

Математические операторы знакомы по арифметическим вычислениям, выполняемым ежедневно.

Это операторы сложения (+), вычитания (-), умножения (*), деления (/), которые позволяют выполнять математические операции в формуле. В формуле [Доход продаж] – [Издержки при продажах] содержится математический оператор вычитания.

📘 Примечание

При использовании с символьными строками оператор "+" ставится оператором конкатенации строк. А это значит, что он объединяет символьные строки. Например, формула "Иван" + "Иванов" возвращает значение "Иван Иванов".

1.6.2.2 Условные операторы

Условные операторы определяют тип сравнения двух значений.

Оператор	Описание
=	Равно
>	Больше
<	Меньше
>=	Больше или равно
<=	Меньше или равно
<>	Не равно

Условные операторы используются с функцией If, например:

```
If [Доход]>10000 Then "High" Else "Low"
```

которая возвращает значение "Высокая" для всех строк, в которых указанная прибыль больше или равна 10000, и значение "Низкая" для всех остальных строк.

1.6.2.3 Логические операторы

К логическим операторам относятся следующие: And, Or, Not, Between и Inlist.

Логические операторы используются в булевых выражениях, возвращающих значения True или False.

1.6.2.3.1 Оператор And

Оператор And связывает булевы значения.

Описание

Если все булевы значения, связанные оператором `And`, возвращают значение `True`, то комбинация всех значений также будет возвращать значение `True`.

Синтаксис

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

Примеры

Функция `If [Курорт] = "Багамский пляж" And [Доход]>100000 Then "Высокий доход с Багамских островов"` возвращает значение "Высокий доход с Багамских островов", если выполнено условие `[Курорт] = "Багамский пляж" And [Доход]>100000`.

1.6.2.3.2 Оператор Or

Оператор `Or` связывает булевы значения.

Описание

Если одно любое булево значение, связанное оператором `Or`, возвращает значение `True`, совокупность всех значений также возвращает значение `True`.

Синтаксис

```
bool_value Or bool_value [Or bool_value...]
```

Примеры

`If [Курорт] = "Багамский пляж" Or [Курорт]="Гавайский клуб" Then "США" Else "Франция"` возвращает "США", если `[Курорт]="Багамский пляж"` или `"Гавайский клуб"` или, в противном случае, возвращает "Франция".

1.6.2.3.3 Оператор Not

Описание

Оператор `Not` возвращает значение, противоположное логическому.

Синтаксис

```
bool Not(bool_value)
```

Примеры

`If Not([Страна] = "США") Then "Не США"` возвращает "Не США", если [Страна] имеет значение, отличное от "США".

1.6.2.3.4 Оператор Between

Описание

Оператор `Between` определяет, заключено ли значение переменной между двумя значениями.

Синтаксис

```
bool Between(first_value;second_value)
```

Примечания

- Функция `Between` используется вместе с функцией `If` и оператором `where`.
- При изменении региональных параметров документа значение, возвращаемое оператором `Between`, может измениться.

Примеры

Функция `If [Доход от продаж] Between(800000;900000) Then "Средний доход"` возвращает значение "Средний доход", если [Доход от продаж] заключен между 800000 и 900000.

Функция `[Доход от продаж] Between (10000;20000)` возвращает значение `True`, если доход от продаж находится между 10000 и 20000.

Функция `If ([Доход от продаж] Between (200000;500000); "Средний доход"; "Низкий/Высокий доход")` возвращает значение "Средний доход", если [Доход от продаж] составляет 300000.

Связанные сведения

[Функция If...Then...Else \[страница 249\]](#)

[Оператор Where \[страница 286\]](#)

1.6.2.3.5 Оператор InList

Описание

Оператор `InList` определяет, принадлежит ли значение списку значений.

Синтаксис

```
bool test_value InList(value_list)
```

Примечания

Именно комбинация `test_value + InList` возвращает булево значение, а не только один оператор `InList`.

Примеры

`If Not ([Страна] InList("Англия"; "Шотландия"; "Уэльс")) Then "Не Британия" Else "Британия"` возвращает "Не Британия", если [Страна] не совпадает с "Англия", "Шотландия" или "Уэльс" или, в противном случае, возвращает "Британия".

If [Курорт] InList("Багамский пляж" ; "Гавайский клуб") Then "Курорт США" возвращает "Курорт США", если [Курорт] совпадает с "Багамский пляж" или "Гавайский клуб".

Связанные сведения

[Функция If...Then...Else \[страница 249\]](#)

[Оператор Where \[страница 286\]](#)

1.6.2.4 Специальные операторы функций

Некоторые функции могут принимать в качестве аргументов определенные операторы.

Например, функция Previous может использовать оператор Self.

Аргументы всех функций заключаются в скобки:) и (. В функциях, работающих с несколькими параметрами, для их разделения используется знак ;.

1.6.2.4.1 Оператор All

При выборе оператора All функция NoFilter будет игнорировать все фильтры.

Кроме того, при выборе оператора All функция Count будет учитывать все значения, включая повторяющиеся.

Связанные сведения

[Количество \[страница 54\]](#)

[Операторы Distinct/All \[страница 280\]](#)

[Функция NoFilter \[страница 254\]](#)

[Операторы All/Drill \[страница 277\]](#)

1.6.2.4.2 Операторы All/Drill

Операторы All/Drill используются с функцией NoFilter.

Описание

С помощью операторов All/Drill можно определить, какие фильтры игнорирует функция NoFilter.

- Не задано – NoFilter игнорирует фильтры отчета и блока
- All – NoFilter игнорирует все фильтры
- Drill – NoFilter игнорирует фильтры отчета и фильтры детализации

1.6.2.4.3 По возрастанию

Оператор Ascending является аргументом функции PromptSummary.

Описание

Когда установлена функция PromptSummary, запросы на ввод сортируются по возрастанию.

Связанные сведения

[Функция PromptSummary \[страница 170\]](#)

1.6.2.4.4 Операторы Bottom/Top

Операторы Bottom/Top используются с функцией Rank.

Описание

Операторы Bottom/Top указывают функции Rank, ранжировать в порядке убывания или возрастания.

- Top – ранжирует в порядке убывания
- Bottom – ранжирует в порядке возрастания

Примеры

Rank([Доход];([Страна]);Top) ранжирует страны по доходу от высокого до низкого.

Связанные сведения

[Функция Rank \[страница 201\]](#)

1.6.2.4.5 Оператор Break

Оператор Break используется с функцией Percentage.

Описание

Оператор Break указывает функции Percentage, что необходимо учитывать разбиения таблиц.

Примеры

Формула Percentage ([Доход]) возвращает следующий результат в данной таблице (процентные соотношения вычисляются на основе общего значения дохода в блоке):

Год	Квартал	Доход	Процентное соотношение
2005	K1	10000	10%
2005	K2	20000	20%
2006	K1	30000	30%
2006	K2	40000	40%

Формула Percentage ([Доход] ; Break) возвращает следующий результат в данной таблице (процентные соотношения вычисляются на основе итогового значения дохода в каждой части блока):

Год	Квартал	Доход	Процентное соотношение
2005	K1	10000	33,3%
2005	K2	20000	66,6%
2006	K1	30000	42,9%
2006	K2	40000	57,1%

Связанные сведения

[Функция Percentage \[страница 66\]](#)

1.6.2.4.6 По убыванию

Оператор `Descending` является аргументом функции `PromptSummary`.

Описание

Когда установлена функция `PromptSummary`, запросы на ввод сортируются по убыванию.

Связанные сведения

[Функция PromptSummary \[страница 170\]](#)

1.6.2.4.7 Операторы Distinct/All

Операторы `Distinct/All` используются с функцией `Count`.

Операторы `Distinct/All` сообщают функции `Count`, каким образом следует подсчитывать число значений: учитывать только различные или учитывать все

Примеры

`Count ([Доход] ; Distinct)` возвращает 3, если `[Доход]` имеет значения (5;5;6;4)

`Count ([Доход] ; All)` возвращает 4, если `[Доход]` имеет значения (5;5;6;4)

Связанные сведения

[Количество \[страница 54\]](#)

1.6.2.4.8 Оператор IncludeEmpty

Оператор IncludeEmpty используется с функциями агрегирования.

Описание

Оператор IncludeEmpty указывает некоторым функциям агрегирования (Average, Count, RunningAverage, RunningCount) на необходимость включать пустые значения в вычисления.

Примеры

Формула `Average([Доход]; IncludeEmpty)` возвращает 3 при следующих значениях переменной [Доход]: 5; 3; <пусто>; 4.

Связанные сведения

[Average \[страница 53\]](#)

[Количество \[страница 54\]](#)

[RunningAverage \[страница 69\]](#)

[Функция RunningCount \[страница 71\]](#)

1.6.2.4.9 Оператор Index

Оператор Index используется с функциями UserResponse и RefValueUserResponse.

Описание

Оператор Index указывает функциям UserResponse и RefValueUserResponse возвращать основной ключ базы данных ответа на подсказку.

Связанные сведения

[Функция UserResponse \[страница 156\]](#)

[Функция RefValueUserReponse \[страница 152\]](#)

1.6.2.4.10 Оператор Linear

Оператор `Linear` используется с функцией `Interpolation`.

Описание

С помощью оператора `Линейный` можно использовать в функции `Интерполяция` линейную регрессию с интерполяцией методом наименьших квадратов для получения отсутствующих значений мер.

При использовании линейной регрессии с интерполяцией методом наименьших квадратов отсутствующие значения вычисляются с помощью линейного выражения $f(x) = ax + b$, которое позволяет получить линию, наиболее приближенную ко всем доступным значениям меры.

Связанные сведения

[Функция `Interpolation` \[страница 57\]](#)

1.6.2.4.11 Оператор NoNull

Оператор `NoNull` используется с функцией `Previous`.

Описание

С помощью оператора `NoNull` функция `Previous` игнорирует нулевые значения.

С помощью оператора `NoNull` функция `Previous` возвращает первое ненулевое значение объекта, обнаруженного при отсчете в обратном направлении, начиная со строк **смещения** ячеек перед текущей строкой.

Связанные сведения

[Функция `Previous` \[страница 264\]](#)

1.6.2.4.12 Оператор NotOnBreak

Оператор `NotOnBreak` используется с функцией `Interpolation`.

Описание

С помощью оператора `NotOnBreak` функция `Interpolation` игнорирует разбиения разделов и блоков.

Связанные сведения

[Функция `Interpolation` \[страница 57\]](#)

1.6.2.4.13 Оператор PointToPoint

С помощью оператора `PointToPoint` функция `Interpolation` использует двухточечную интерполяцию для получения отсутствующих значений мер.

Описание

При двухточечной интерполяции отсутствующие значения получаются путем вычисления значений линейной функции $f(x) = ax + b$, проходящей через две соседние с отсутствующими значениями точки.

Связанные сведения

[Функция `Interpolation` \[страница 57\]](#)

1.6.2.4.14 Операторы Row/Col

Оператор `Row` вычисляет каждое значение в строке как процент от итогового значения всех строк во внедренном контексте. Оператор `Col` вычисляет каждое значение в столбце как процент от итогового значения всех столбцов во внедренном контексте.

Описание

Операторы Row/Col задают направление вычисления следующих функций: Percentage, Previous, RunningAverage, RunningCount, RunningMax, RunningMin, RunningProduct, RunningSum.

Примечания

В кросс-таблице значение в каждой из ячеек по умолчанию рассчитывается как процент от общего значения кросс-таблицы. Оператор Row вычисляет значения в строках как процент от общего значения для строки. Оператор Col вычисляет значения в столбцах как процент от общего значения для столбца.

Примеры

В кросс-таблице Percentage ([Мера]) дает следующий результат:

Мера	Процентное соотношение	Мера	Процентное соотношение
100	10%	500	50%
200	20%	200	20%

Percentage ([Мера] ; ROW) дает следующий результат:

Мера	Процентное соотношение	Мера	Процентное соотношение
100	16,7%	500	83,3%
200	50%	200	50%

Percentage ([Мера] ; COL) дает следующий результат:

Мера	Процентное соотношение	Мера	Процентное соотношение
100	33,3%	500	83,3%
200	66,6%	200	16,7%

Оператор Row вычисляет промежуточное агрегатное значение по строкам. Оператор Col вычисляет промежуточное агрегатное значение по столбцам.

В кросс-таблице функции RunningSum ([Measure]) и RunningSum ([Measure] ; Row) дают следующие результаты:

Мера	RunningSum	Мера	RunningSum
100	100	200	300

400	700	250	950
-----	-----	-----	-----

В кросс-таблице `Percentage ([Мера])` дает следующий результат:

Мера	RunningSum	Мера	RunningSum
100	100	200	700
400	500	250	950

Связанные сведения

[Функция Percentage \[страница 66\]](#)

[RunningAverage \[страница 69\]](#)

[Функция RunningCount \[страница 71\]](#)

[Функция RunningMax \[страница 74\]](#)

[Функция RunningMin \[страница 76\]](#)

[Функция RunningProduct \[страница 78\]](#)

[RunningSum \[страница 79\]](#)

1.6.2.4.15 Оператор Self

Оператор `Self` используется с функцией `Previous`.

Описание

Указывает функции `Previous` на предыдущую ячейку, если она не содержит объект отчета.

Примеры

`5 + Previous(Self)` возвращает последовательность 5, 10, 15, 20, 25, 30...

`1 + 0.5 * Previous(Self)` возвращает последовательность 1, 1.5, 1.75, 1.88...

Связанные сведения

[Функция Previous \[страница 264\]](#)

1.6.2.4.16 Оператор Where

Описание

Оператор where ограничивает данные, используемые для вычисления меры.

Примеры

Формулой `Average ([Доход с продаж] where ([Страна] = "США"))` вычисляется средний доход с продаж для страны "США".

Формулой `Average ([Доход с продаж] where ([Страна] = "США" or [Страна] = "Франция"))` вычисляется средний доход с продаж для страны "США" или "Франция".

Формулой `[Доход] where (Not ([Страна] Inlist ("США"; "Франция")))` вычисляется доход для стран, отличных от "США" и "Франция".

Переменная [Высокий доход] описана формулой `[Доход] where [Доход] > 500000`. При размещении в блок переменной [Высокий доход] отображаются только значения дохода, превышающие 500000. При помещении в нижний колонтитул в нижней части столбца [Высокий доход], формула `Average ([Высокий доход])` возвращает среднее значение всех доходов, превышающих 500000.

Связанные сведения

[Оператор And \[страница 273\]](#)

[Оператор Between \[страница 275\]](#)

[Оператор InList \[страница 276\]](#)

[Оператор Or \[страница 274\]](#)

[Оператор Not \[страница 275\]](#)

1.6.2.5 Операторы расширенного синтаксиса

Контексты ввода и вывода задаются явным образом при помощи операторов контекста.

В следующей таблице перечислены операторы контекста:

Оператор	Описание
In	Определяет подробный список измерений для использования в контексте.

Оператор	Описание
ForEach	Добавляет измерения в контекст по умолчанию
ForAll	Удаляет измерения из контекста по умолчанию

Операторы ForAll и ForEach могут оказаться полезными, когда в контексте по умолчанию содержится множество измерений. Часто может быть проще добавлять или удалять измерения из контекста с помощью операторов ForAll и ForEach, чем создавать список с использованием In.

1.6.2.5.1 Оператор In Context

Оператор In Context определяет измерения в контексте в явной форме.

Пример: Использование оператора In Context для определения измерений в контексте

В этом примере у нас есть отчет, в котором отображается "Год" и "Доход от продаж". В поставщике данных также содержится объект "Квартал", однако это измерение в блок не включено. Вместо этого, требуется включить дополнительный столбец для отображения максимального значения дохода за каждый квартал каждого года. Отчет выглядит следующим образом:

Год	Доход от продаж	Максимальный квартальный доход
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

Чтобы узнать, откуда берутся значения в столбце "Максимальный доход за квартал", просмотрите этот блок совместно с блоком, содержащим измерение "Квартал":

Год	Квартал	Доход от продаж
2001	K1	\$2,660,699.50
2001	K2	\$2,279,003.00
2001	K3	\$1,367,841.00
2001	K4	\$1,788,580.00
	Максимум:	\$2,660,699.50

Год	Квартал	Доход от продаж
	K1	\$3,326,172.00
	K2	\$2,840,651.00
	K3	\$2,879,303.00

Год	Квартал	Доход от продаж
	K4	\$4,186,120.00
	Максимум:	\$4,186,120.00
Год	Квартал	Доход от продаж
	K1	\$3,742,989.00
	K2	\$4,006,717.50
	K3	\$3,953,395.00
	K4	\$3,356,041.00
	Максимум:	\$4,006,717.50

Столбец "Максимальный доход за квартал" содержит максимальные показатели дохода за каждый год. Например, K4 содержит значение максимального дохода в 2002 году, таким образом, "Максимальный доход за квартал" в строке 2002 года отобразит K4.

Использование оператора In, формула для "Максимального дохода за квартал"

```
Max ([Доход от продаж] In ([Год];[Квартал])) In ([Год])
```

Эта формула рассчитывает максимальный доход от продаж для каждого из сочетаний (Год, Квартал), а затем выводит полученное значение по годам.

📘 Примечание

Так как по умолчанию контекстом для вывода блока служит "Год", особым образом указывать контекст вывода в этой формуле не требуется.

1.6.2.5.2 Контекстный оператор ForEach

Оператор ForEach добавляет в контекст измерения.

Пример: Использование оператора ForEach для добавления измерений в контекст

В следующей таблице показан максимальный доход по каждому кварталу в отчете, который содержит измерение "Квартал", не включенное в блок:

Год	Доход от продаж	Максимальный квартальный доход
2001	8096123.60	2660699.50
2002	13232246.00	4186120.00
2003	15059142.80	4006717.50

Для столбца "Максимальный доход за квартал" можно создать формулу, которая не содержит оператор ForEach:

```
Max ([Доход от продаж] In ([Год];[Квартал])) In ([Год])
```

Контекстный оператор "ForEach" позволяет добиться того же результата с помощью следующей формулы:

```
Max ([Доход от продаж] ForEach ([Квартал])) In ([Год])
```

Причина: Измерение "Год" является в блоке контекстом ввода по умолчанию. При использовании оператора "ForEach" измерение "Квартал" добавляется в контекст, в результате чего получается контекст ввода ([Год];[Квартал]).

1.6.2.5.3 Контекстный оператор ForAll

Контекстный оператор ForAll удаляет измерения из контекста.

Пример: Использование оператора ForAll для удаления измерений из контекста

Есть отчет, в котором отображается год, квартал и доход от продаж, и необходимо добавить столбец, в котором отображается суммарный доход по каждому году, как показано в следующем блоке:

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

Для суммирования доходов по годам контекстом ввода должен быть (Year); по умолчанию это (Year; Quarter). Поэтому необходимо удалить из контекста ввода квартал, указав в формуле ForAll ([Quarter]), что будет выглядеть следующим образом:

```
Sum([Sales revenue] ForAll ([Quarter]))
```

Следует отметить, что для получения такого же результата можно использовать оператор In. В этом случае формула будет выглядеть следующим образом:

```
Sum([Sales revenue] In ([Year]))
```

В данной версии формулы в качестве контекста явно указывается год, а не удаляется квартал, чтобы оставить год.

1.6.2.6 Операторы наборов

Операторы наборов работают с элементами в иерархических данных.

1.6.2.6.1 Оператор диапазона

Описание

Оператор диапазона (:) возвращает набор элементов, лежащих между двумя элементами на одном и том же уровне, включая и сами эти два элемента

Синтаксис

```
first_member:last_member
```

Примеры

[География]&[США].[Калифорния].[Лос-Анджелес]:[География]&[США].[Калифорния].[Сан-Франциско] возвратит [Лос-Анджелес], [Сан-Диего], [Сан-Франциско], если элементы на этом уровне идут в порядке ...[Лос-Анджелес], [Сан-Диего], [Сан-Франциско]...

Sum([Доход];{[География]&[США].[Калифорния].[Лос-Анджелес]:[География]&[США].[Калифорния].[Сан-Франциско]}) возвращает итоговый доход для Лос-Анджелеса, Сан-Диего и Сан-Франциско.

1.6.3 Ключевые слова расширенного синтаксиса

Ключевые слова с расширенным синтаксисом – это форма условного обозначения, которая позволяет обращаться к измерениям с расширенным синтаксисом без явного указания этих измерений.

Эти ключевые слова позволяют обеспечить создание неустаревающих отчетов. Если в формулах не содержатся жестко запрограммированные ссылки на измерения, то они будут оставаться действительными даже при добавлении измерений в отчет или удалении измерений из отчета.

Предусмотрено пять ключевых слов расширенного синтаксиса: Report, Section, Break, Block и Body.

1.6.3.1 Ключевое слово Block

В этом разделе описываются измерения, на которые ссылается ключевое слово Block в зависимости от его расположения в отчете. Ключевые слова Block и Section зачастую охватывают одни и те же данные.

Вся разница заключается в том, что Block принимает во внимание фильтры блока, а Section – нет.

Если расположено в...	Ссылается на эти данные...
Блок	Данные во всем блоке, без учета разрывов, но с учетом фильтров
Разбиение по блокам (верхний или нижний колонтитул)	Данные во всем блоке без учета разрывов, но с учетом фильтров
Раздел (верхний колонтитул, нижний колонтитул или данные за пределами блока)	Неприменимо
За пределами всех блоков или разделов	Неприменимо

Пример: Ключевое слово Block

В отчете содержатся сведения о доходах от продаж за год и квартал. В этом отчете создан раздел на основе объекта "Год". Фильтры блока опускают показатели за третий и четвертый кварталы.

2001

Quarter	Sales revenue	First Half Average	Yearly Average
Q1	\$2 660 700	\$2 469 851.25	\$8 096 123.60
Q2	\$2 279 003	\$2 469 851.25	\$8 096 123.60
Сумма:	4 939 702.5		

2002

Quarter	Sales revenue	First Half Average	Yearly Average
Q1	\$3 326 172	\$3 083 411.50	\$13 232 246.00
Q2	\$2 840 651	\$3 083 411.50	\$13 232 246.00
Сумма:	6 166 823		

2003

Quarter	Sales revenue	First Half Average	Yearly Average
Q1	\$3 742 989	\$3 874 853.20	\$15 059 142.80
Q2	\$4 006 718	\$3 874 853.20	\$15 059 142.80
Сумма:	7 749 706.4		

В столбце "В среднем за год" используется следующая формула:

```
Average([Доход от продаж] In Section)
```

В столбце "В среднем за первое полугодие" используется следующая формула:

```
Average ([Доход от продаж]) In Block
```

Можно проследить за тем, как ключевое слово Block учитывает фильтр блока.

1.6.3.2 Ключевое слово Body

В этом разделе описываются измерения, на которые ссылается ключевое слово Body в зависимости от его расположения в отчете.

Если расположено в...	Ссылается на эти данные...
Блок	Данные в блоке
Разбиение по блокам (верхний или нижний колонтитул)	Данные в блоке
Раздел (верхний колонтитул, нижний колонтитул или данные за пределами блока)	Данные в разделе
За пределами всех блоков или разделов	Данные в отчете

Пример: Ключевое слово Body

В отчете отображаются объекты "Год", "Квартал" и "Доход с продаж" с разбиением по объекту "Год". В отчете содержится раздел, основанный на объекте "Год", и разбиение по объекту "Квартал".

Год	Квартал	Доход от продаж	Body
2001	K1	2,660,700	2,660,699.5
	K2	2,279,003	2,279,003
	K3	1,367,841	1,367,840.7
	K4	1,788,580	1,788,580.4
2001		8,096,123.6	

В столбце Body записана формула

```
Sum ([Доход от продаж]) In Body
```

Итоги в столбце Body и "Доход с продаж" одинаковы, поскольку ключевое слово Body относится к данным в блоке. Если удалить объект "Месяц", цифры в столбце "Блок" изменятся для обеспечения соответствия цифрам в столбце "Доход с продаж". Если формулу разместить в нижнем колонтитуле отчета, то будет вычисляться совокупный доход для Body.

1.6.3.3 Ключевое слово Break

В следующей таблице описываются измерения, на которые ссылается ключевое слово Break в зависимости от его расположения в отчете:

Если расположено в...	Ссылается на эти данные...
Блок	Данные в части блока, разделенные разбиением
Разбиение по блокам (верхний или нижний колонтитул)	Данные в части блока, разделенные разбиением
Раздел (верхний колонтитул, нижний колонтитул или данные за пределами блока)	Не применимо
За пределами всех блоков или разделов	Неприменимо

Пример: Ключевое слово Break

В отчете содержатся сведения о доходах от продаж за год и квартал:

Год	Квартал	Доход от продаж	Итог разбиения
2001	K1	\$2,660,700	\$8,096,124
	K2	\$2,279,003	\$8,096,124

Год	Квартал	Доход от продаж	Итог разбиения
	К3	\$1,367,841	\$8,096,124
	К4	\$1,788,580	\$8,096,124

В отчете содержится разбиение по объекту "Год". В столбце "Итог разбиения" записана формула:

```
Sum ([Доход от продаж]) In Break
```

При отсутствии ключевого слова Break в этом столбце будут дублироваться цифры из столбца "Доход с продаж", поскольку будет использоваться контекст вывода по умолчанию ([Год];[Квартал]).

1.6.3.4 Ключевое слово Report

В этом разделе описываются данные, на которые ссылается ключевое слово Report в зависимости от его расположения в отчете.

Если расположено в...	Ссылается на эти данные...
Блок	Все данные в отчете
Разбиение по блокам (верхний или нижний колонтитул)	Все данные в отчете
Раздел (верхний колонтитул, нижний колонтитул или данные за пределами блока)	Все данные в отчете
За пределами всех блоков или разделов	Все данные в отчете

Пример: Ключевое слово Report

В отчете содержатся сведения о доходах от продаж за год и квартал. В отчете есть столбец "Итог по отчету", в котором отображается сумма всех доходов из отчета.

Year	Quarter	Sales revenue	Report Total
2001	Q1	\$2 660 700	36 387 512.4
2001	Q2	\$2 279 003	36 387 512.4
2001	Q3	\$1 367 841	36 387 512.4
2001	Q4	\$1 788 580	36 387 512.4
2002	Q1	\$3 326 172	36 387 512.4
2002	Q2	\$2 840 651	36 387 512.4
2002	Q3	\$2 879 303	36 387 512.4
2002	Q4	\$4 186 120	36 387 512.4
2003	Q1	\$3 742 989	36 387 512.4
2003	Q2	\$4 006 718	36 387 512.4
2003	Q3	\$3 953 395	36 387 512.4
2003	Q4	\$3 356 041	36 387 512.4

Формула столбца "Итог по отчету" выглядит следующим образом:

```
Sum([Доход от продаж]) In Report
```

Без ключевого слова Report в данном столбце воспроизводились бы цифры из столбца "Доходы от продаж", поскольку в нем использовался бы контекст вывода по умолчанию ([Год];[Квартал]).

1.6.3.5 Ключевое слово Section

В этом разделе описываются данные, на которые ссылается ключевое слово Section в зависимости от его расположения в отчете.

Если расположено в...	Ссылается на эти данные...
Блок	Все данные в разделе
Разбиение по блокам (верхний или нижний колонтитул)	Все данные в разделе
Раздел (верхний колонтитул, нижний колонтитул или данные за пределами блока)	Все данные в разделе
За пределами всех блоков или разделов	Не применимо

Пример: Ключевое слово Section

В отчете отображаются объекты "Год", "Квартал" и "Доход с продаж".

2001		
Quarter	Sales revenue	Section Total
Q1	\$2 660 700	8 095 814
Q2	\$2 278 693	8 095 814
Q3	\$1 367 841	8 095 814
Q4	\$1 788 580	8 095 814

В этом отчете создан раздел на основе объекта "Год". В столбце "Итог раздела" записана формула:

```
Sum([Доход от продаж]) In Section
```

Цифра в столбце "Итог раздела" обозначает совокупный доход за 2001 год, поскольку разбиение раздела выполнено в объекте "Год". При отсутствии ключевого слова Section в этом столбце будут дублироваться цифры из столбца "Доход с продаж", поскольку будет использоваться контекст вывода по умолчанию ([Год];[Квартал]).

1.6.4 Округление и усечение чисел

Некоторые функции содержат параметр, определяющий, до какого знака функция округляет или обрезает возвращаемое значение.

Этот параметр принимает целочисленное значение, которое должно быть больше нуля, меньше нуля или равно нулю. В следующей таблице описываются способы округления и усечения чисел в каждом из этих случаев:

Параметр	Описание
> 0	<p>Эта функция округляет или обрезает до десятичных знаков <параметра>.</p> <p>Примеры.</p> <p>Round (3 , 13 ; 1) возвращает 3,1</p> <p>Round (3 , 157 ; 2) возвращает 3,16</p>
0	<p>Эта функция округляет или обрезает до ближайшего целого числа</p> <p>Примеры:</p> <p>Truncate (3 , 7 ; 0) возвращает 3</p> <p>Truncate (4 , 164 ; 0) возвращает 4</p>
< 0	<p>Эта функция округляет или обрезает до ближайшего числа 10 (параметр = -1), 100 (параметр = -2), 1000 (параметр = -3) и так далее.</p> <p>Примеры:</p> <p>Round(123,76; -1) возвращает 120</p> <p>Round(459,9; -2) возвращает 500</p> <p>Truncate(1600; -3) возвращает 1000</p>

📌 Примечание

Числа представляются в формате двойной точности с плавающей запятой, определены точно от 15 до 17 цифр.

Связанные сведения

[Функция Round \[страница 203\]](#)

[Функция Truncate \[страница 209\]](#)

[EuroConvertTo \[страница 189\]](#)

[Функция EuroConvertFrom \[страница 187\]](#)

[Функция EuroFromRoundError \[страница 190\]](#)

1.6.5 Ссылки на элементы и наборы элементов в иерархиях

Ссылки на элементы и наборы элементов в функциях оформляются с помощью синтаксиса `[hierarchy]&path.function`.

Элементы `path` и `function` не являются обязательными. `Path` содержит ссылки на элементы в квадратных скобках, при этом они разделяются точкой. Имена элементов и уровней указываются с учетом регистра.

ⓘ Примечание

Наборы элементов используются для переопределения контекста вычисления по умолчанию для иерархии. В функциях с наборами элементов последние заключаются в {}.

Для обозначения диапазонов элементов между начальным и конечным элементами указывается двоеточие (:) и для каждого элемента указывается полный путь. Диапазон содержит все элементы того же уровня, что и указанные элементы.

Пример синтаксиса для диапазона: `[Иерархия продаж]&[Customer_Type].[ENTERPRISE]; [Крупный].[Нэнси Дэволио]: [Иерархия продаж]&[Customer_Type].[ENTERPRISE]; [Крупный].[Эндрю Смит]`.

Пример: Ссылки на элементы и наборы элементов

Имеется следующая иерархия:

Иерархия продаж	Сумма заказа
Customer_Type	277,290,434
ENTERPRISE	180,063,361
Крупный	113,905,997
Нэнси Дэволио	44,855,689
Джэнет Леверлинг	44,050,308
Эндрю Смит	30,000,000
GLOBAL	91,157,363

- `[Иерархия продаж]&[Customer_Type].[ENTERPRISE].[Крупный].Children` ссылается на элементы [Нэнси Дэволио], [Джэнет Леверлинг] и [Эндрю Смит].
- `Sum([Сумма заказа]; {[Иерархия продаж]&[Customer_Type].[ENTERPRISE].[Крупный].children})` возвращает результат 113 905 997 (сумма значений трех дочерних элементов).

- `[Иерархия продаж]&[Customer_Type].[ENTERPRISE].[Крупный].[Дженет Леверлинг]` ссылается на элемент `[Дженет Леверлинг]`.
- `Sum([Сумма заказа];{[Иерархия продаж]&[Customer_Type].[ENTERPRISE].[Крупный].[Дженет Леверлинг];[Иерархия продаж]&[Customer_Type].[ENTERPRISE].[Крупный].[Нэнси Дэволио]})` возвращает результат 88 905 997 (сумма значений двух элементов).
- `[Иерархия продаж]&[Customer_Type].[ENTERPRISE].[Крупный].[Нэнси Дэволио]:[Иерархия продаж]&[Customer_Type].[ENTERPRISE].[Крупный].[Эндрю Смит]` ссылается на элементы `[Нэнси Дэволио]`, `[Дженет Леверлинг]` и `[Эндрю Смит]`.
- `Sum([Сумма заказа];{[Иерархия продаж]&[Customer_Type].[ENTERPRISE].[Крупный].[Нэнси Дэволио]:[Иерархия продаж]&[Customer_Type].[ENTERPRISE].[Крупный].[Эндрю Смит]})` возвращает результат 113 905 997 (сумма значений трех элементов диапазона).
- `[Иерархия продаж].children` ссылается на все элементы в иерархии `[Иерархия продаж]`.
- `Sum([Сумма заказа];{Иерархия продаж}.children)` возвращает результат 277 290,434.

1.7 Создание пользовательских функций

1.7.1 Обзор внешних функций

Расширения для вычислений — это пользовательские вычисления для отчетов Web Intelligence, расширяющие список существующих функций Web Intelligence.

Чтобы использовать библиотеку расширений вычислений, создайте внешнюю библиотеку C++, используя определенный API.

1.7.1.1 Внешние функции

Внешние функции отображаются и могут использоваться подобно другим стандартным функциям Web Intelligence. Можно создать формулу с помощью функций, реализующих собственную логику пользователя.

📌 Примечание

Можно определить любое необходимое количество функций. Поддерживаются только функции, использующие параметры с единственным значением. Можно использовать не более пяти параметров с одним значением.

Чтобы определить функцию, выполните следующие действия.

1. Объявите в файле XML описание внешних функций, используя заданную структуру XML.
2. Внедрите функцию в библиотеку C++, используя заданный API.
3. Скопируйте файл XML и библиотеку в соответствующую папку в каталоге установки Business Objects Enterprise для сервера и клиента на настольном ПК).
4. Перезапустите систему, чтобы автоматически добавить внешнюю функцию в список функций, доступных для создания формул.

Внешняя функция основана на уникальном идентификаторе, чтобы при применении в отчете ее нельзя было ошибочно перепутать с другой используемой внешней библиотекой.

Если система не может загрузить библиотеку или отсутствует информация для внешней функции, есть несогласованное объявление XML, отсутствует библиотека или присутствует дубликат функции, отображается сообщение об ошибке. Кроме того, система записывает ошибки в журнал трассировки.

Связанные сведения

[сообщение об ошибке #EXTERNAL \[страница 309\]](#)

1.7.1.2 Развертывание пользовательских функций

Для развертывания пользовательских функций необходимо выполнить несколько действий вручную. Администратор BusinessObjects должен поместить файл XML и файл связанной библиотеки DLL в папку библиотек на сервере и на каждом компьютере с установленным полным настольным клиентом.

Предупреждение

Замена или добавление библиотеки в папку пользовательских библиотек может представлять угрозу для системы. Поскольку эта библиотека загружается автоматически, внешняя библиотека может получить доступ к внутренним критически важным данным или процессам, создавая опасность для системы.

Убедитесь, что администратор установил надлежащие права доступа к соответствующей папке, чтобы с ней могли работать только авторизованные пользователи.


1.7.1.3 Объявление библиотеки

В различных операционных системах для файлов библиотек используются разные расширения:

- DLL для Windows
- SO для Linux или UNIX

Используются следующие типы файлов:

Тип	Описание
Объявление каталогов XML	Существует только один файл этого типа, у которого должно быть имя <code>externalcatalogs.xml</code> . Этот файл содержит список всех файлов определений функций XML.

Тип	Описание
Объявление функций XML	<p>Этот файл, определяющий список функций и связанных библиотек, приведен в файле объявления каталогов XML.</p> <div>  Примечание Файл каталогов может содержать или указывать библиотеки объявления функций. </div>
файл библиотеки	<p>Этот файл содержит код в C++ для функций пользователя.</p> <p>Файл библиотеки содержит реализацию функции пользователя, определенную в объявлении функции XML.</p>

1.7.1.4 Использование образцов файлов Web Intelligence

Проверьте, что установлены следующие приложения:

- Visual Studio C++ VS2015 или выше
- Web Intelligence 4.1 или выше

В примерах данного документа используются образцы файлов из файла `Samples.zip`, который находится в каталоге `[каталог установки]\userlibs\WebI\Samples\`.

1. Распакуйте файл `Samples.zip`.
2. Чтобы открыть примеры, запустите файл `OpenSolution.bat`.

Файл `OpenSolution.bat` задает временную переменную среду `<WEBICALCPPLUGINAPI>`, используемую этим решением для поиска определенных заголовочных файлов Web Intelligence.

⚠ Предупреждение

Если нужная версия Web Intelligence не установлена, файл `OpenSolution.bat` использовать нельзя. В этом случае задайте вручную для переменной среды `<WEBICALCPPLUGINAPI>` путь к папке, которая содержит файлы заголовка Web Intelligence, и откройте `Samples\WebICalcPlugin`.

Связанные сведения

[Примеры \[страница 308\]](#)

1.7.2 Определение пользовательского вычисления

Настройка функции в Web Intelligence:

1. Определите объявление функции XML.
2. Определите объявление каталога XML.
3. Реализуйте библиотеку в C++, используя конкретный API для внешней функции.
4. Скомпилируйте исходный файл.
5. Скопируйте определение XML и библиотеку в соответствующую папку `WebiCalcPlugin` (на стороне сервера и каждого "толстого" клиента).
6. Перезапустите сервер Web Intelligence.

❗ Примечание

В примерах данной главы используются образцы файлов, включенные в программное обеспечение Web Intelligence.

Система автоматически добавляет функцию в список функций в редакторе формул и контекстную справку панели формул.

Если формула использует функцию, для которой недоступна внешняя библиотека, появляется сообщение об ошибке `#EXTERNAL`.

❗ Примечание

Поддерживаются только функции, использующие параметры с единственным значением. Параметры таблицы для экземпляра не поддерживаются.

1.7.2.1 Объекты функции XML

Определение XML содержит объекты, определяющие пользовательскую функцию. Пользовательские функции XML расширяют список функций для языка формул, чтобы для формулы, использующей эту функцию, можно было выполнить синтаксический разбор в соответствии с подписью XML и преобразование в форму с маркерами. Внешней функции следует назначить глобальный уникальный идентификатор (GUID), чтобы ее нельзя было повторно использовать или спутать с другими пользовательскими библиотеками.

Определение XML включает следующие объекты:

Тег	Атрибут XML	Объект определения XML
<code><CATALOG></code>		Корневой XML
<code><LIBRARY></code>	<code>file</code>	Имя файла библиотеки, содержащего код реализации C++ Файл библиотеки может включать несколько функций. Не следует задавать расширение библиотеки.

Тег	Атрибут XML	Объект определения XML
<FUNCTION>	guid	<p>Уникальный GUID функции</p> <div> <p>→ Совет</p> <p>Заблаговременно определите все GUID и убедитесь, что все они являются уникальными с глобальной точки зрения.</p> </div> <p>Для Windows можно использовать инструмент для генерации GUID, входящий в ПО Visual Studio, или загрузить его с веб-сайта Microsoft. Для Linux инструмент <code>usr/bin/uuidgen</code> можно найти в пакете <code>libuuid1</code> (Debian).</p>
	name	<p>Имя функции, отображаемое в редакторе формул</p> <p>Имя функции должно удовлетворять следующим условиям:</p> <ul style="list-style-type: none"> • быть простым и уникальным именем для функции, • начинаться с буквы, • содержать строчные и прописные буквы, числа или символ <code>—</code>, • отсутствовать в библиотеке Web Intelligence. <div> <p>ⓘ Примечание</p> <p>Имя не будет переводиться на другие языки.</p> </div>
<ARGLIST>		<p>Список параметров</p> <p>Число параметров должно быть меньше или равно 5.</p>
<ARG>	type	<p>Типы параметров</p> <p>Можно использовать следующие типы параметров:</p> <ul style="list-style-type: none"> • Числовой • Логический • Дата • Строка
	name	<p>Имя каждого параметра в том виде, в каком оно будет отображаться в редакторе формул</p> <p>Это имя показывает прототип метода для пользователя. Используйте только буквы и цифры.</p>

Тег	Атрибут XML	Объект определения XML
<RETURN>	type	<p>Тип возвращаемых значений</p> <p>Могут возвращаться следующие значения:</p> <ul style="list-style-type: none"> • Числовое • Логическое • Дата • Строка
<CATEGORY>	type	<p>Категория, в которой функция будет отображаться в редакторе формул</p> <p>Действуйте последовательно, поместите строки в категорию символов и числа в числовую категорию. Доступные категории:</p> <ul style="list-style-type: none"> • Символьные • Дата • Документ • Поставщик данных • Разное • Логические • Num
<HINT>	value	<p>Подсказка, отображаемая в редакторе формул</p> <p>Эта подсказка поясняет использование функции.</p>

1.7.2.2 Определение объявления функции XML

XML для подписи использует следующую структуру:

```
Function_list
```

XML для подписи использует следующую структуру:

```
Function_list := [Function*]
Function := [name, GUID, data_type = Numeric|Boolean|Date|
String, category = character|Date|Document|DP|Misc|Logical|Num,
parameter_list, (online_help_signature?),
(online_help_description?),library_name)]
parameter_list := [parameter*]
parameter := [name, data_type =Numeric|Boolean|Date|String]
```

1. Задайте корневой тег XML как CATALOG
2. В CATALOG добавьте теги LIBRARY.
3. В LIBRARY добавьте имя файла библиотеки без расширения DLL или SO. Это атрибут файла.
4. В LIBRARY добавьте теги FUNCTION.

У тега `FUNCTION` должен быть уникальный GUID и дополнительное уникальное имя атрибута, определяющее имя функции.

Тег `FUNCTION` должен содержать:

- тег `ARGLIST` с тегами `ARG`. Теги `ARG` должны включать первый атрибут, определяющий тип этого параметра, и второй атрибут, определяющий имя этого параметра.
Тип `ARG` может быть булевым, числовым, датой или строкой. Имя `ARG` содержит только буквы и цифры.

📌 Примечание

Можно использовать максимум пять параметров.

- тег `RETURN`, определяющий тип атрибута.
Тип `RETURN` может быть булевым, числовым, датой или строкой.
 - тег `CATEGORY`, определяющий атрибут типа.
Тип `CATEGORY` может быть символом, датой, документом, DP, смешанным, логическим или числовым.
 - тег `HINT`, определяющий атрибут значения.
5. Поместите определение XML в соответствующую папку (на стороне сервера и каждого клиента Rich client).

Пример: SampleMath.xml

```
<CATALOG>
  <LIBRARY file="SampleMath">
    <FUNCTION guid="CC3E9742-67A7-4844-9DBF-2CCD4F6ECABE" name="MySquareFct">
      <ARGLIST>
        <ARG type="Numeric" name="input_number"/>
      </ARGLIST>
      <RETURN type="Numeric"/>
      <CATEGORY type="Num"/>
      <HINT value="My square function."/>
    </FUNCTION>
  </LIBRARY>
</CATALOG>
```

Связанные сведения

[Использование образцов файлов Web Intelligence \[страница 300\]](#)

1.7.2.3 Определение объявления каталога XML

Можно создать объявление каталога XML или добавить его в существующее объявление каталогов.

<CATALOG> указывает файл объявления функции XML или непосредственно определяет <CATALOG>, как показано в разделе, в котором определен формат объявления функций XML.

Создание объявления каталога

1. Присвойте этому объявлению имя `externalcatalogs.xml`.
2. Задайте корневой тег XML как `CATALOGS`.
3. В `CATALOGS` добавьте теги `CATALOG`.

Это действие определяет значение имени файла для объявлений функций XML.

4. Поместите библиотеку XML в соответствующую папку (на стороне сервера и каждого клиента Rich client).

Пример: externalcatalogs.xml

```
<CATALOGS>
  <CATALOG file="SampleMath.xml" />
</CATALOGS>
```

Связанные сведения

[Использование образцов файлов Web Intelligence \[страница 300\]](#)

1.7.2.4 Реализация файла C++

1. В это файл добавьте заголовок `ibovariant.h`.
2. Для каждого метода объявление должно начинаться с макрокоманды `BO_DECLARE_USER_FCT`.

Эта макрокоманда включает:

- имя функции, отображаемое в файле объявления функций XML,
- имя объекта возвращаемого значения,
- имя объекта параметра

📌 Примечание

Функция возвращает `BONORError`, если все в порядке, в противном случае в отчете появляется сообщение об ошибке `#EXTERNAL`.

Пример: Square.cpp

```
// Headers file include of the WebI headers
#include <ibovariant.h>
// To not repeat BOExtFunct::
using namespace BOExtFunct;
BO_DECLARE_USER_FCT (// Name of function as it was defined in the XML.
    MySquareFct,
    // Name of the return value object.
    retVal,
    // Name of the parameters object.
    parameters
)
{
    try // Always used a try{}catch(...) to be sure no
        // exception was thrown outside this Web
        // Intelligence user function.
    {
        // Get the first parameter.
        const iBOValue&param0 = parameters[0];
        // Transform the parameter to the correct type.
        double valPar0(param0);
        // Assign value to the return value.
        retVal = valPar0 * valPar0;
    }
    catch(...)
    {
        return BOERROR; // Unkonwn exception so notify WebI
    }
    return BONOERROR; // It's OK
}
```

Связанные сведения

[Использование образцов файлов Web Intelligence \[страница 300\]](#)

1.7.2.5 Компиляция исходного файла в Microsoft Visual Studio 2015

1. Чтобы создать проект, выберите ► *Файл* ► *Создать* ► *Проект* ►.
2. В списке *Типы проектов* выберите ► *Visual C++* ► *Общие* ►.
3. В списке *Шаблоны* выберите *Пустой проект*.
4. Укажите имя проекта.
5. Укажите папку для проекта.
6. Нажмите кнопку *ОК*.
7. Щелкните проект правой кнопкой мыши и выберите *Свойства*.
8. В разделе *Конфигурация* выберите *Все конфигурации*.
9. В разделе ► *Свойства конфигурации* ► *Общие* ► задайте для *типа конфигурации* значение *Динамическая библиотека (.dll)*.

10. Нажмите кнопку *ОК*.
11. Щелкните проект правой кнопкой мыши и выберите ► *Добавить* ► *Новый элемент* ►.
12. В разделе *Категория* выберите *Код*.
13. В разделе *Шаблон* выберите *Файл C++ (.CPP)*.
14. Укажите имя файла CPP.
15. Нажмите *Добавить*.
16. Щелкните проект правой кнопкой мыши и выберите *Свойства*.
17. В разделе *Конфигурация* выберите *Все конфигурации*.
18. В разделе ► *Свойства конфигурации* ► *C/C++* ► *Дополнительные каталоги включаемых файлов* ► добавьте папку, содержащую заголовки файла Business Objects.
19. Нажмите кнопку *Применить*.
20. В разделе *Конфигурация* выберите *Отладка*.
21. В разделе ► *Свойства конфигурации* ► *C/C++* ► *Создание кода* ► задайте для параметра *Библиотека времени выполнения* значение *Многопоточная DLL (/MD)*.

ⓘ Примечание

Если используется компьютер, на котором установлена среда Microsoft Visual Studio, можно использовать значение *Многопоточная DLL с возможностью отладки (/MDd)* вместо *Многопоточная DLL (/MD)*, чтобы использовать среду отладки.

22. Нажмите кнопку *Применить*.
23. В разделе *Конфигурация* выберите *Выпуск*.
24. В разделе ► *Свойства конфигурации* ► *C/C++* ► *Создание кода* ► задайте для параметра *Библиотека времени выполнения* значение *Многопоточная DLL (/MD)*.

ⓘ Примечание

Если используется компьютер, на котором установлена среда Microsoft Visual Studio, можно использовать значение *Многопоточная DLL с возможностью отладки (/MDd)* вместо *Многопоточная DLL (/MD)*, чтобы использовать среду отладки.

25. Нажмите кнопку *ОК*.
26. Добавьте код в файл CPP.
27. Выполните компиляцию.

1.7.2.6 Копирование файла в WebiCalcPlugin

Скопируйте объявление функций XML, объявление каталогов XML и файл DLL/SO в папку WebiCalcPlugin.

Эта папка доступна в каталоге:

[каталог установки]\[версия BusinessObjects]\[OS]_[PLATFORM]\WebiCalcPlugin

Где: [версия BusinessObjects] - версия продукта, например BusinessObjects Enterprise XI 4.0, а [OS] - операционная система, например win32 для ОС Windows или linux для ОС Linux, а [PLATFORM] - платформа, например x86 в 32-разрядном ЦП Intel.

1.7.3 Примеры

В этих примерах используются образцы файлов из архива `Samples.zip`, который находится в каталоге `[каталог установки]\userlibs\WebI\Samples\`.

Пример: Объявление каталога XML для `externalcatalogs.xml`

```
<CATALOGS>
  <CATALOG file="SampleString.xml"/>
</CATALOGS>
```

Пример: Объявление функции XML в `SampleString.xml`

```
<CATALOG>
  <LIBRARY file="SampleString">
    <FUNCTION guid="A91BD526-B8EB-4b09-90F2-FFCD350776A8" name="MyHelloWorld">
      <RETURN type="String"/>
      <CATEGORY type="Num"/>
      <HINT value="My simple hello world function."/>
    </FUNCTION>
  </LIBRARY>
</CATALOG>
```

Пример: Объявление файла C++ в `HelloWorld.cpp`

```
// Headers file include of the Web Intelligence hearders
#include <ibovariant.h>
// To not repeat BOExtFunc::
using namespace BOExtFunc;
BO_DECLARE_USER_FCT(
    // Name of function as it was defined in the XML.
    MyHelloWorld,
    // Name of the return value object.
    retVal
    // Don't use parameter.
    /*parameters*/
)
{
    try // Always used a try{}catch(...) to be sure no
        // exception was thrown outside this
        // Web Intelligence user function.
```

```

{
    // Create an std::wstring with wide char Hello world.
    std::wstring helloWorldStr = L"Hello world!!!";
    // Initialise the return value.
    retVal = helloWorldStr;
}
catch(...)
{
    // Unkonwn exception so notify Web Intelligence
    return BOERROR;
}
return BONOERROR;    // It's OK
}

```

Связанные сведения

[Использование образцов файлов Web Intelligence \[страница 300\]](#)

1.7.4 сообщение об ошибке #EXTERNAL

Сообщение об ошибке #EXTERNAL появляется при наличии следующих проблем:

- Формула ссылается на внешнюю функцию, которой нет в папке внешней библиотеки.
- Документ содержит внешний метод, который не может загрузить система. Несогласованное объявление или файл библиотеки не найден.
- Внешний метод не инициализировал возвращаемое значение.
- Внешний метод инициализировал возвращаемое значение с неправильным типом. Например, для строки задан тип "double".
- Внешний метод возвратил код ошибки.

Попросите администратора BusinessObjects развернуть нужную библиотеку, в которой реализована эта функция.

1.7.5 Сообщения об ошибках журнала трассировки

Если во время синтаксического разбора или проверки XML возникает ошибка, пользователю выдается сообщения, а ошибки записываются в журналы трассировки.

Тип журнала	Сообщения об ошибках
Журналы XML	File cannot be read or is missing. Bad XML structure due to: <ul style="list-style-type: none"> • Parent/Children relation invalid. • Missing field (ID function, name function). • Invalid field value.
Журналы DLL	File is missing. DLL cannot be loaded. Function is not found in the DLL.
Журналы функций	Function name is already in use. Function ID is already used. Function name is missing. Return type is invalid. ID is invalid. Number of parameters is invalid.
Журналы параметров	Parameter name is missing. Parameter type is invalid.
Журналы времени выполнения	The user function does not initialize the return value. The user function initializes the return value with a bad type. The user function returns the BOERROR error code.

1.8 Устранение связанных с формулами неполадок

1.8.1 Механизм автоматической перезаписи формулы

Пакеты исправлений Web Intelligence иногда могут приводить к разным результатам расчетов в разных версиях.

Начиная с версии 4.1 SP3, Web Intelligence предлагает механизм автоматической перезаписи формулы, который позволяет автоматически изменить набор формул (см. список ниже) в документе, перенесенном из предыдущей версии. Эти формулы строятся по определенному шаблону. После изменения формулы возвращают тот же результат, что и до изменения расчета. Поэтому рекомендуется сохранять документ, чтобы изменения сохранились в документе, завершая тем самым работу механизма перезаписи формулы.

Механизм автоматической перезаписи формулы по умолчанию доступен для документов, перенесенных в BI 4.1 SP3 и выше, для следующих шаблонов формул:

1. оператор Where() с измерением в качестве параметра в условии;
2. текущие расчеты со сбросом в разделах;
3. текущие расчеты со сбросом в кросс-таблицах.

Список правил может быть расширен в будущих пакетах дополнительными шаблонами формул.

Правило(1)

В предыдущих версиях расчет данных выполнялся определенным образом, если оператор Where() с измерением использовался в качестве параметра в условии. Измерение в этом случае добавлялось в контекст показателя. Правило(1) воспроизводит это прежнее поведение.

Это правило применяется к каждому документу, перенесенному из XI 3.1 FP3.6, XI 3.1 FP4.1, XI 3.1 FP5.1 и 4.0 SP5.

Правило(2)

В предыдущих версиях текущие расчеты в разделах выполнялись неправильно, так как расчеты сбрасывались для каждой инстанции раздела. Правило(2) воспроизводит это прежнее поведение.

Это правило применяется к каждому документу, перенесенному из XI R2 SP4.

Правило(3)

В предыдущих версиях текущие расчеты со сбросом в кросс-таблицах подразумевали выполнение расчетов по шаблону "N" (столбец за столбцом) вместо шаблона "Z" (строка за строкой).

Правило(3) вводит ключевое слово FORCE_COL для принудительного выполнения расчетов Web Intelligence по шаблону "N".

Например, для формулы RunningSum([Sales revenue];([State])) Правило(3) инициирует выполнение столбец за столбцом, если внести изменение RunningSum([Sales revenue];FORCE_COL;([State])).

Это правило применяется к каждому документу, перенесенному из каждой версии XI 3.x, 4.0 с программной вставкой 2.20, 4.0 SP5, 4.0 SP6, 4.0 SP7, 4.1 и 4.1 SP1.

1.8.2 Сообщения об ошибках формул и информационные сообщения

Данные отчета, которые возвращают сообщения об ошибках, можно форматировать с помощью условного форматирования.

В некоторых случаях формула не может вернуть значение и поэтому возвращает сообщение об ошибке или информационное сообщение, которое начинается с символа "#". Сообщение появляется в ячейке, в которую помещена формула.

1.8.2.1 #COMPUTATION

Сообщение об ошибке #COMPUTATION выводится в том случае, когда измерение со срезами, указанное в качестве параметра функции `RelativeValue`, более не доступно в контексте вычислений того блока, в котором эта функция расположена.

Ошибка #COMPUTATION также возникает при включении в отчет объединенного объекта, содержащего иерархию.

Сообщение об ошибке #COMPUTATION также может быть связано с неправильным использованием контекстных операторов в формуле.

Связанные сведения

[Функция `RelativeValue` \[страница 268\]](#)

1.8.2.2 #CONTEXT

#CONTEXT отображается в мере, если мера имеет несуществующий контекст вычисления.

Сообщение об ошибке #CONTEXT относится к сообщениям об ошибках #INCOMPATIBLE и #DATASYNC, которые отображаются в измерениях в случае, когда блок содержит несуществующий контекст вычисления.

В случае #INCOMPATIBLE контекст не существует по причине несовместимости измерений; в случае #DATASYNC контекст не существует потому, что измерения получены из нескольких несинхронизированных поставщиков данных.

Пример: Несуществующий контекст вычисления в запросе

Если блок, в основе которого лежит юниверс "Island Resorts Marketing", содержит объекты "Год резервирования" и "Доход", сообщение об ошибке #CONTEXT отображается потому, что невозможно агрегировать доход по году резервирования. (Резервирования еще не сгенерировали ни один доход).

1.8.2.3 #DATASYNC

Сообщение #DATASYNC появляется при размещении измерения от другого поставщика данных в блок, в котором содержатся измерения от отличных поставщиков данных, и два поставщика данных не синхронизированы в объединенном измерении.

Сообщение #DATASYNC появляется во всех измерениях в блоке, а сообщение #CONTEXT – в мерах.

Пример: Измерения от разных поставщиков данных в блоке

Если в отчете, основанном на Universe Island Resorts Marketing, содержатся поставщики данных с объектами (Год, Доход) и (Квартал), в блоке с объектами Год, Квартал и Доход отображается сообщение #DATASYNC в столбцах Год и Квартал, поскольку два поставщика данных не синхронизированы в объединенном измерении.

1.8.2.4 #DIV/O

Сообщение #ДЕЛЕНИЕ/О появляется, если в формуле выполняется попытка деления числа на ноль, что математически не представляется возможным.

Ноль никогда не может являться делителем.

Пример: Определение дохода для одного объекта

В отчете отображаются доходы с продаж, число проданных объектов и доход, полученный с продажи одного объекта (который вычисляется путем деления дохода с продаж на число проданных объектов).

Для квартала отсутствует доход с продаж; в столбце дохода с продажи одного элемента появляется сообщение #ДЕЛЕНИЕ/О для этого квартала, поскольку в формуле была предпринята попытка деления на ноль; т.е. деление дохода на нулевое число проданных объектов.

1.8.2.5 #ERROR

#ERROR – сообщение об ошибке по умолчанию, которое отображается при возникновении ситуации, которой не присвоено никакое другое сообщение об ошибке.

1.8.2.6 #EXTERNAL

Ошибка #EXTERNAL возникает в случае, когда формула ссылается на внешнюю функцию, которую нельзя использовать в Web Intelligence.

1.8.2.7 #INCOMPATIBLE

#INCOMPATIBLE – сообщение о наличии несовместимых объектов в блоке.

Пример: Несовместимые объекты в запросе

Если блок, в основе которого лежит юниверс Island Resorts Marketing (Маркетинг островных курортов), содержит измерения "Год" и "Год бронирования", в столбцах, в которых содержатся эти измерения, отображается сообщение #INCOMPATIBLE, поскольку эти объекты несовместимы.

1.8.2.8 #MIX

Ошибка #MIX возникает, когда в агрегированной мере обнаруживаются различные единицы измерения.

Например, в ячейке отображается сообщение об ошибке #MIX, если в ней агрегируются денежные значения, деноминированные в различных валютах.

1.8.2.9 #MULTIVALUE

Сообщение #НЕОДНОЗНАЧНО появляется при размещении формулы, которая возвращает несколько значений в ячейку, для которой предусмотрен вывод только одного значения.

Пример: Многозначность в ячейке

В отчет с объектами "Страна", "Курорт" и "Доход" добавляется ячейка с формулой [Доход] ForEach ([Страна]). В этой ячейке отображается сообщение #НЕОДНОЗНАЧНО, поскольку для объекта "Страна" в отчете заданы два значения: "США" и "Франция".

В одной ячейке невозможно одновременно отобразить доходы для США и Франции. В ячейке с доходом, расположенной за пределами таблицы, можно только определенным образом агрегировать доходы, указанные в таблице (например путем сложения или нахождения среднего значения).

Если в отчете выполнено разбиение на разделы по объекту "Страна", формула является правильной, если помещена в раздел, поскольку в одном разделе задано только одно значение для объекта "Страна". Однако за пределами раздела формула будет возвращать #НЕОДНОЗНАЧНО

1.8.2.10 #N/A

Если значение ячейки отчета базируется на значении отчета, недоступного в основании данных (например, ошибка BW в ячейке BEx), в ней отображается сообщение #N/A (недоступно). Оно свидетельствует о том, что ячейка пуста из-за ошибки при извлечении данных.

1.8.2.11 #OVERFLOW

Ошибка #OVERFLOW происходит, если при вычислении возвращается значение, слишком большое для обработки в программе.

Такое значение (в экспоненциальном представлении) должно превышать 1,7 E 308 (1,7 с 307 нолями).

1.8.2.12 #PARTIALRESULT

Ошибка #PARTIALRESULT возникает, если были извлечены не все связанные с объектом отчета строки.

Если в ваших отчетах часто возникает ошибка #PARTIALRESULT, и при наличии требуемых прав безопасности, измените свойство запроса **Максимальное количество извлекаемых строк**, чтобы разрешить извлечение большего объема данных. Если отсутствуют права на изменение запроса, обратитесь к администратору BI.

Если в отчете содержатся интеллектуальные меры, отображение ошибки #PARTIALRESULT более вероятно, так как для работы интеллектуальных мер необходимо извлекать большие объемы данных, чем для работы классических мер.

1.8.2.13 #RANK

Сообщение об ошибке #RANK возникает при попытке ранжирования данных, в основе которых лежит объект, зависящий от порядка значений.

От порядка значений зависят объекты, использующие функцию Previous или любую функцию промежуточного агрегирования.

Из-за ранжирования эти объекты пересчитывают свои значения, в результате чего изменяется ранжирование и возникает циклическая зависимость. Такая зависимость также может возникать при использовании диалогового окна "Ранг" для создания ранжирования или при использовании функции Rank.

Пример: Ранжирование по промежуточному среднему предыдущих значений

Если выполняется попытка ранжирования блока по столбцу, который содержит функцию `Previous` или любую функцию промежуточного агрегирования, ошибку `#RANK` возвращает весь блок.

1.8.2.14 #RECURSIVE

Ошибка `#RECURSIVE` возникает при невозможности выполнить вычисление из-за циклической зависимости.

Пример: Использование функций `NumberOfPages()`, `Page()` и `PageInSection()`

Если функции `NumberOfPages`, `Page` и `PageInSection` помещаются в ячейку с установленными свойствами "Автоматический подбор высоты" или "Автоматический подбор ширины", ячейка возвращает значение `#RECURSIVE`, так как в этом случае возникает циклическая зависимость. Чтобы вернуть значение, функциям нужен точный размер отчета, но на размер отчета влияет размер ячейки, который определяется содержимым ячейки.

1.8.2.15 #REFRESH

Ошибка `#REFRESH` возникает в отчетах с ячейками, производными от объектов, которые были удалены из запроса и повторно добавлены в запрос.

Ячейки удаляются из запроса, если установлено свойство запроса [Задействовать разбор запросов](#) и объект не используется ни в одном из основанных на запросе отчетов.

Ячейки повторно заполняются значениями из объектов при обновлении запроса.

1.8.2.16 #SECURITY

Ошибка `#SECURITY` возникает при попытке использования функции, для которой у пользователя нет прав защиты.

Пример: Использование функции DataProviderSQL()

Если пользователь, у которого нет прав просмотра поставщика данных SQL, размещает в ячейке функцию DataProviderSQL(), в ячейке отображается сообщение #SECURITY.

1.8.2.17 #SYNTAX

Ошибка #SYNTAX возникает в случае, когда формула ссылается на объект, который больше не существует в отчете.

Пример: Ссылка на несуществующий объект

Есть отчет, в котором исходно отображался доход от продаж по годам и кварталам, с дополнительным столбцом, в котором отображалась разница между доходом и средним доходом за год. Это значение представлено переменной "Разница со средним за год".

Если из отчета удалить переменную "Разница со средним за год", в столбце, в котором она находилась, возвращается ошибка #SYNTAX.

1.8.2.18 #TOREFRESH

#TOREFRESH – сообщение о том, что в ячейках, использующих интеллектуальные меры, возвращаемое ими значение не поддерживается.

Такая ситуация имеет место, когда в поставщике данных недоступен набор для группировки, содержащий значение.

Ошибка #TOREFRESH устраняется посредством обновления данных.

Некоторые меры являются "делегированными" (в BW это означает, что мера не агрегируется с помощью функции SUM). При определении для меры таблицы или вычисления эта мера запрашивается в определенном контексте агрегирования (мера указывается для набора измерений). Если этот набор измерений является подмножеством набора измерений запроса, меру следует агрегировать для указанного набора измерений (или набора группировок, ссылающегося на группу с помощью оператора SQL).

Для обычных мер система выполняет агрегирование, для делегированных мер это агрегирование делегируется в исходную базу данных. Для этого системе вновь требуется послать запрос этой базе данных. Поскольку данное действие не является автоматическим, выводится сообщение об ошибке #TOREFRESH и пользователь должен сам выполнить обновление. После обновления система отправляет дополнительный запрос для получения требуемого агрегирования, а затем заменяет #TOREFRESH подходящим значением.

1.8.2.19 #UNAVAILABLE

Ошибка #UNAVAILABLE возникает при невозможности расчета значения интеллектуальной меры.

Это происходит в случае, когда отобразить значения в отфильтрованной интеллектуальной мере невозможно без применения фильтра к запросу. Так как применение фильтра может затронуть другие отчеты, основанные на том же запросе, фильтр не применяется.

1.9 Сравнение значений с помощью функций

1.9.1 Сравнение значений с помощью функции Previous

Функция `Previous` возвращает предыдущее значение выражения для сравнения.

Возвращенное значение зависит от макета отчета.

Для использования расширенных возможностей сравнения применяйте функцию `RelativeValue`.

`RelativeValue` возвращает предыдущее или последующее значение выражения для сравнения.

Возвращенное значение не зависит от макета отчета.

Связанные сведения

[Функция Previous \[страница 264\]](#)

[Функция RelativeValue \[страница 268\]](#)

[Сравнение значений с помощью функции RelativeValue \[страница 318\]](#)

1.9.2 Сравнение значений с помощью функции RelativeValue

Функция `RelativeValue` возвращает значения выражения для сравнения. Функция возвращает данные значения независимо от макета отчета.

Во время использования функции `RelativeValue` необходимо указать следующие данные.

- Выражение, чье значение для сравнения необходимо найти (должно быть мерой или объектом-описанием измерения, доступными в блоке)
- Список измерений срезов
- Смещение

В данной функции используются измерения срезов, смещение и измерения вложенных осей (которые применяются измерениями срезов) для возврата значения для сравнения. Измерения вложенных осей – это все остальные измерения в контексте вычислений, не считая измерений срезов.

Выражаясь в общих чертах, функция `RelativeValue` возвращает значение выражения в строку, которая в списке значений измерений срезов является строкой **смещения**, удаленной из текущей строки, и в которой измерения вложенных осей такие же, как и в текущей строке.

❗ Примечание

Все измерения срезов должны находиться в контексте вычислений блока, куда помещена функция. Если измерение среза впоследствии удалить, функция возвращает `#COMPUTATION`.

Пример

В следующем примере в столбце `RelativeValue` содержится следующая формула.

```
RelativeValue([Доход];([Год]);-1)
```

- Выражение – `[Доход]`;
- Измерение среза – `[Год]`;
- Смещение равно минус единице (функция сразу возвращает предыдущее значение в списке).

Год	Квартал	Продавец	Доход	RelativeValue
2007	K1	Смит	1000	
2007	K2	Джонс	2000	
2007	K3	Уилсон	1500	
2007	K4	Харрис	3000	
2008	K1	Смит	4000	1000
2008	K2	Джонс	3400	2000
2008	K3	Уилсон	2000	1500
2008	K4	Харрис	1700	3000

Формула, выраженная в деловом вопросе, возвращает прибыль, полученную тем же продавцом за тот же квартал предыдущего года.

Вычисление, выполняемое в формуле, можно описать так: формула возвращает значение `[Доход]` (выражение) из строки, в которой значение `[Год]` (измерение среза) равно предыдущему значению из списка значений объекта `[Год]`, а значения `[Квартал]` и `[Продавец]` (измерения вложенных осей) совпадают со значениями в текущей строке.

Связанные сведения

[Функция `RelativeValue` \[страница 268\]](#)

1.9.2.1 Измерения срезов и функция RelativeValue

В функции `RelativeValue` список значений измерений срезов используется для поиска сравнимой строки.

Данная функция возвращает значение для сравнения с выражением, указанным в функции, находящейся за смещение строк от списка измерений срезов.

В результате порядок сортировки измерений срезов является решающим в определении выходных данных функции.

Пример: Несколько измерений срезов

В указанной ниже таблице столбец `RelativeValue` содержит следующую формулу:

```
RelativeValue([Доход];([Год];[Квартал]);-1)
```

- Выражение – `[Доход]`;
- Измерения срезов – `([Год];[Квартал])`;
- Смещение равно минус единице (функция сразу возвращает предыдущее значение в списке).

Год	Квартал	Продавец	Доход	RelativeValue
2007	K1	Смит	1000	
2007	K2	Смит	2000	
2007	K3	Смит	1500	
2007	K4	Смит	3000*	
2007	K1	Джонс	4000	
2007	K2	Джонс	3400	
2007	K3	Джонс	2000	
2007	K4	Джонс	1700	
2008	K1	Смит	5000**	3000*
2008	K2	Смит	3000***	5000**
2008	K3	Смит	2700****	3000***
2008	K4	Смит	6800	2700****

Формула, выраженная в деловом вопросе, возвращает доход, обеспеченный тем же продавцом в предыдущем квартале.

Вычисление, выполняемое в формуле, возвращает значение `[Дохода]` из строки, в которой значения `[Год]` и `[Квартал]` представляют предыдущее значение из списка значений `([Год];[Квартал])`, а значение `[Продавец]` совпадает со значением в текущей строке.

Функция использует для поиска сравниваемого дохода список значений измерений срезов.

Год	Квартал	
2007	K1	
2007	K2	
2007	K3	
2007	K4	*
2008	K1	**
2008	K2	***
2008	K3	****
2008	K4	

Порядок сортировки в измерениях срезов определяет выходные данные функции. Знак "*" в таблицах отображает порядок сортировки.

Связанные сведения

[Функция RelativeValue \[страница 268\]](#)

1.9.2.2 Измерения срезов и разделы

Измерение среза может находиться в начальной ячейке раздела отчета.

Пример: Измерение среза в ячейке раздела

В указанной ниже таблице столбец RelativeValue содержит следующую формулу:

```
RelativeValue([Доход];([Год];[Квартал]);-1)
```

2007

Квартал	Продавец	Доход	RelativeValue
K1	Смит	1000	
K2	Смит	2000	
K3	Смит	1500	
K4	Смит	3000*	
K1	Джонс	4000	
K2	Джонс	3400	

Квартал	Продавец	Доход	RelativeValue
K3	Джонс	2000	
K4	Джонс	1700	

2008

Квартал	Продавец	Доход	RelativeValue
K1	Смит	5000**	3000*
K2	Смит	3000***	5000**
K3	Смит	2700 ****	3000***
K4	Смит	6800	2700****

Функция использует для поиска сравниваемого дохода список значений измерений срезов.

Год	Квартал	
2007	K1	
2007	K2	
2007	K3	
2007	K4	*
2008	K1	**
2008	K2	***
2008	K3	****
2008	K4	

Порядок сортировки в измерениях срезов определяет выходные данные функции. Знак "*" в таблицах отображает порядок сортировки.

Связанные сведения

[Функция RelativeValue \[страница 268\]](#)

1.9.2.3 Порядок измерений срезов

Так как порядок сортировки списка значений измерений срезов определяет выходные данные функции RelativeValue, порядок, в котором указываются измерения срезов, влияет на выходные данные функции.

Пример: Порядок измерений срезов

В указанной ниже таблице столбец RelativeValue содержит следующую формулу:

```
RelativeValue([Доход];([Год];[Квартал]);-1)
```

Год	Квартал	Продавец	Доход	RelativeValue
2007	K1	Смит	1000	
2007	K2	Смит	2000	
2007	K3	Смит	1500	
2007	K4	Смит	3000*	
2007	K1	Джонс	4000	
2007	K2	Джонс	3400	
2007	K3	Джонс	2000	
2007	K4	Джонс	1700	
2008	K1	Смит	5000**	3000*
2008	K2	Смит	3000***	5000**
2008	K3	Смит	2700****	3000***
2008	K4	Смит	6800	2700****

Формула, выраженная в деловом вопросе, возвращает доход, обеспеченный тем же продавцом в предыдущем квартале.

Порядок сортировки измерений срезов выглядит следующим образом.

Год	Квартал	
2007	K1	
2007	K2	
2007	K3	
2007	K4	*
2008	K1	**
2008	K2	***
2008	K3	****
2008	K4	

Название функции изменяется на:

```
RelativeValue([Доход];([Квартал];[Год]);-1)
```

Порядок сортировки измерений срезов становится следующим.

Квартал	Год	
K1	2007	*

Квартал	Год	
K1	2008	**
K2	2007	***
K2	2008	****
K3	2007	*****
K3	2008	*****
K4	2007	*****
K4	2008	*****

Порядок сортировки оказывает следующее влияние на результат функции.

Год	Квартал	Продавец	Доход	RelativeValue
2007	K1	Смит	1000*	
2007	K2	Смит	2000***	
2007	K3	Смит	1500*****	
2007	K4	Смит	3000*****	
2007	K1	Джонс	4000	
2007	K2	Джонс	3400	
2007	K3	Джонс	2000	
2007	K4	Джонс	1700	
2008	K1	Смит	5000**	1000*
2008	K2	Смит	3000****	2000***
2008	K3	Смит	2700*****	1500*****
2008	K4	Смит	6800*****	3000*****

Выраженная в деловом вопросе формула теперь возвращает доход, получаемый тем же продавцом за тот же квартал предыдущего года.

Изменение в порядке сортировки измерения среза изменяет смысл формулы. Знак "*" в таблицах означает порядок сортировки

Связанные сведения

[Функция RelativeValue \[страница 268\]](#)

1.9.2.4 Сортировка измерений срезов

Так как порядок сортировки списка значений измерений срезов определяет выходные данные функции, сортировка, применяемая к измерению в измерениях срезов, влияет на выходные данные функции.

Пример: Пользовательская сортировка, применяемая к измерению среза

В указанной ниже таблице столбец RelativeValue содержит следующую формулу:

```
RelativeValue([Доход];([Год];[Квартал]);-1)
```

Пользовательская сортировка (K1, K2, K4, K3) применяется к [Квартал], предоставляя следующие результаты для функции.

Год	Квартал	Продавец	Доход	RelativeValue
2007	K1	Смит	1000	
2007	K2	Смит	2000	
2007	K4	Смит	3000	
2007	K3	Смит	1500*	
2007	K1	Джонс	4000	
2007	K2	Джонс	3400	
2007	K4	Джонс	1700	
2007	K3	Джонс	2000	
2008	K1	Смит	5000**	1500*
2008	K2	Смит	3000***	5000**
2008	K4	Смит	6800****	3000***
2008	K3	Смит	2700	6800****

Отсортированный список измерений срезов выглядит следующим образом.

Год	Квартал	
2007	K1	
2007	K2	
2007	K4	
2007	K3	*
2008	K1	**
2008	K2	***
2008	K4	****

Год	Квартал
2008	К3

Знак "*" в таблицах отображает порядок сортировки.

Связанные сведения

[Функция RelativeValue \[страница 268\]](#)

1.9.2.5 Использование функции RelativeValue в кросс-таблицах

Функция RelativeValue работает в кросс-таблицах точно так же, как и в вертикальных таблицах.

Размещение данных в кросс-таблицах не влияет на выходные данные функции.

Связанные сведения



[Функция RelativeValue \[страница 268\]](#)

Важные положения об отказе от ответственности в отношении правовых вопросов

Гиперссылки

Некоторые ссылки обозначаются значком и/или текстом, отображаемым при наведении мыши. Эти ссылки обеспечивают доступ к дополнительной информации.

Подробнее о значках:

- Ссылки со значком  Вы собираетесь перейти на сайт, размещенный не на сервере SAP. Используя такие ссылки, вы соглашаетесь (если иное не оговорено особо в соглашениях с SAP) со следующим:
 - Сайт по ссылке не содержит документацию SAP. Не разрешается подавать рекламации в отношении любых продуктов SAP на основе содержащейся на таком сайте информации.
 - SAP не выражает ни согласия, ни несогласия с информацией, содержащейся на сайте по ссылке, а также не гарантирует ее доступность и правильность. SAP не несет ответственности за любой ущерб, вызванный использованием такой информации, за исключением тех случаев, когда такой ущерб вызван намеренными нарушениями или халатностью со стороны компании SAP.
- Ссылки со значком  Вы закрываете документацию по определенному продукту или сервису SAP и переходите на веб-сайт, расположенный на сервере SAP. Используя такие ссылки, вы соглашаетесь (если иное не оговорено особо в соглашениях с SAP) с тем, что не разрешается подавать рекламации в отношении любых продуктов SAP на основе содержащейся на таком сайте информации.

Видео-ролики, размещенные на внешних платформах

Некоторые видео-ролики могут указывать на сторонние платформы размещения видео-роликов. SAP не может гарантировать в будущем доступность видео-роликов, сохраненных на этих платформах. Кроме того, любые рекламные объявления или другой контент, размещенные на этих платформах (например, предлагаемые видео-ролики или ссылки на другие видео-ролики, размещенные на одном сайте), не входят в сферу управления или ответственности SAP.

Бета-версии и другие экспериментальные функции

Экспериментальные функции не являются частью официально поставляемого SAP объема, гарантируемого для будущих версий. Это означает, что экспериментальные функции могут быть изменены компанией SAP в любое время и по любой причине без предварительного уведомления. Экспериментальные функции не предназначены для продуктивного использования. Не разрешается демонстрировать, тестировать, проверять, анализировать или иначе использовать экспериментальные функции в фактической операционной среде либо с использованием данных, для которых не выполнено резервное копирование. Экспериментальные функции предназначены для получения обратной связи, которая позволяет нашим клиентам и партнерам влиять на разработку будущих продуктов. Предоставляя обратную связь (например, в SAP Community), вы соглашаетесь с тем, что права на интеллектуальную собственность относительно ваших отзывов и производных работ останутся в исключительной собственности SAP.

Пример кода

Примером кода является любой код и/или фрагменты кода программного обеспечения. Они не предназначены для продуктивного использования. Этот код предназначен только для пояснения и иллюстрирования синтаксиса и правил составления текста программ. SAP не гарантирует правильность и полноту примеров кода. SAP не несет ответственности за любые ошибки и ущерб, вызванные использованием примеров кода, за исключением тех случаев, когда такой ущерб вызван намеренными нарушениями или халатностью со стороны компании SAP.

Язык, свободный от предрассудков

SAP поддерживает культуру многообразия и инклюзивности. Когда это возможно, в нашей документации мы используем безоценочный язык для обозначения людей из любой культуры или этнической группы, любого пола и уровня способностей.

© SAP SE или аффилированная компания SAP, 2024. Все права защищены.

Полное или частичное воспроизведение или передача в какой-либо форме и в каких-либо целях настоящей публикации без явного образом выраженного разрешения SAP SE или аффилированной компании SAP запрещены. Информация, содержащаяся в настоящей публикации, может быть изменена без предварительного уведомления.

Некоторые программные продукты, предлагаемые на рынке компанией SAP SE и ее дистрибьюторами, содержат компоненты программного обеспечения, исключительными правами в отношении которых обладают иные поставщики программного обеспечения. Возможны различные варианты спецификаций продуктов для разных стран.

Материалы предоставлены компанией SAP SE и ее аффилированной компанией исключительно в информационных целях, без предоставления каких-либо гарантий. Компания SAP или ее аффилированные компании не несут ответственности за ошибки или пропуски в настоящих материалах. Гарантии, если таковые предоставляются, в отношении продуктов и услуг компании SAP или ее аффилированной компании содержатся исключительно в документах, которые прилагаются к соответствующим продуктам и услугам. Ничто, изложенное в настоящем документе, не должно трактоваться как предоставление дополнительных гарантий.

SAP, а также упомянутые здесь продукты и услуги SAP, как и соответствующие логотипы, являются товарными знаками или зарегистрированными товарными знаками SAP SE (или аффилированной компании SAP) на территории Германии и других стран. Все иные названия продуктов и услуг являются товарными знаками соответствующих компаний.

Для получения дополнительной информации и уведомлений о товарных знаках см. <https://www.sap.com/cis/about/legal/trademark.html>.