

SAP BusinessObjects Intelligence Suite
Версия документа: 4.1 Support Package 7 – 11.12.2015

Использование функций, формул и вычислений в Web Intelligence

Содержание

1	История документа: функции, формулы и вычисления Web Intelligence.	5
2	О данном руководстве.	7
3	Использование стандартных и пользовательских вычислений.	8
3.1	Стандартные вычисления.	8
3.2	Использование формул для создания пользовательских вычислений.	9
	Использование переменных для упрощения формул.	9
3.3	Работа с функциями.	9
	Включение функций в ячейки.	10
	Синтаксис функций.	10
	Примеры функций.	10
	Операторы функций и формул.	15
4	Сведения о контекстах вычисления.	18
4.1	Контекст ввода.	18
4.2	Контекст вывода.	19
4.3	Контексты вычислений по умолчанию.	21
	Контексты по умолчанию в вертикальной таблице.	22
	Контексты по умолчанию в горизонтальной таблице.	23
	Контексты по умолчанию в кросс-таблице.	23
	Контексты по умолчанию в разделе.	24
	Контексты по умолчанию в разбиении.	25
4.4	Изменение контекста вычислений по умолчанию с расширенным синтаксисом.	26
	Операторы расширенного синтаксиса.	27
5	Расчет значений с помощью интеллектуальных мер.	31
5.1	Наборы группировок и интеллектуальные меры.	31
	Управление наборами группировок.	32
5.2	Интеллектуальные меры и область анализа.	33
5.3	Интеллектуальные меры и язык SQL.	33
	Наборы группировок и оператор UNION.	33
5.4	Интеллектуальные меры и формулы.	35
	Интеллектуальные меры и измерения, содержащие формулы.	35
	Интеллектуальные меры в формулах.	35
5.5	Интеллектуальные меры и фильтры.	36
	Ограничение связано с интеллектуальными мерами и фильтрами.	36
	Интеллектуальные меры и фильтры для измерений.	37

	Фильтрация интеллектуальных мер.	38
	Интеллектуальные меры и фильтры детализации.	39
	Интеллектуальные меры и вложенные фильтры OR.	39
6	Функции, операторы и ключевые слова.	40
6.1	Функции.	40
	Пользовательские форматы.	40
	Функции агрегирования.	44
	Символьные функции.	85
	Функции даты и времени.	108
	Функции провайдера данных.	126
	Функции документа.	142
	Логические функции.	152
	Числовые функции.	163
	Функции множества.	186
	Прочие функции.	207
6.2	Операторы функций и формул.	230
	Математические операторы.	230
	Условные операторы.	230
	Логические операторы.	231
	Специальные операторы функций.	234
	Операторы расширенного синтаксиса.	243
	Операторы наборов.	246
6.3	Ключевые слова расширенного синтаксиса.	247
	Ключевое слово Block.	247
	Ключевое слово Body.	248
	Ключевое слово Break.	249
	Ключевое слово Report.	250
	Ключевое слово Section.	251
6.4	Округление и усечение чисел.	252
6.5	Ссылки на элементы и наборы элементов в иерархиях.	253
7	Устранение связанных с формулами неполадок.	255
7.1	Механизм автоматической перезаписи формулы.	255
7.2	Сообщения об ошибках формул и информационные сообщения.	256
	#COMPUTATION.	256
	#CONTEXT.	257
	#DATASYNC.	257
	#DIV/0.	257
	#ERROR.	258
	#EXTERNAL.	258
	#INCOMPATIBLE.	258

#MIX.	258
#MULTIVALUE.	259
#N/A.	259
#OVERFLOW.	259
#PARTIALRESULT.	259
#RANK.	260
#RECURSIVE.	260
#REFRESH.	260
#SECURITY.	261
#SYNTAX.	261
#TOREFRESH.	261
#UNAVAILABLE.	262
8 Сравнение значений с помощью функций.	263
8.1 Сравнение значений с помощью функции Previous.	263
8.2 Сравнение значений с помощью функции RelativeValue.	263
Измерения срезов и функция RelativeValue.	264
Измерения срезов и разделы.	266
Порядок измерений срезов.	267
Сортировка измерений срезов.	269
Использование функции RelativeValue в кросс-таблицах.	270

1 История документа: функции, формулы и вычисления Web Intelligence

В следующей таблице описываются наиболее важные изменения документа.

Версия	Дата	Описание
SAP BusinessObjects Web Intelligence 4.1 SP07	Ноябрь 2015 г.	<p>В документ добавлены следующие разделы:</p> <ul style="list-style-type: none">• Строки для создания пользовательских форматов недель Пользовательские форматы [страница 40]• В значении даты и времени может отображаться часовой пояс Пользовательские форматы [страница 40]• Обновлено поведение функции Concatenation() Функция Concatenation [страница 87]• Обновлено поведение функции RelativeDate() Функция RelativeDate [страница 119]• Механизм автоматической перезаписи формулы, доступный с 4.1 SP3, теперь задокументирован. Механизм автоматической перезаписи формулы [страница 255]
SAP BusinessObjects Web Intelligence 4.1 SP05	Ноябрь 2014 г.	<p>В руководство были добавлены примеры синтаксиса format_string. Для просмотра примеров см. Примеры Format_string для функции FormatDate [страница 90].</p>
SAP BusinessObjects Web Intelligence 4.1	Май 2013 г.	<p>Включает поддержку написания справа налево для арабского языкового стандарта; она особенно влияет на работу следующих символьных функций для языковых стандартов с написанием справа налево:</p> <ul style="list-style-type: none">• Left• LeftPad• LeftTrim• Right• RightPad• RightTrim

Версия	Дата	Описание
SAP BusinessObjects Web Intelligence 4.0 с пакетом компонентов 3	Февраль 2012 г.	Пакет компонентов 3
SAP BusinessObjects Web Intelligence 4.0 SP02	Июнь 2011 г.	Пакет поддержки 2. Возвращено название Web Intelligence
SAP BusinessObjects Interactive Analysis 4.0 SP01	Март 2011 г.	Пакет поддержки 1
SAP BusinessObjects Interactive Analysis 4.0	Ноябрь 2010 г.	Изначально обновленный Interactive Analysis. Первый выпуск документа

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.

3 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.

3.1 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.
Minimum	Отображает минимальное значение среди выбранных данных.
Maximum	Отображает максимальное значение среди выбранных данных.
Percentage	<p>Отображает выбранные данные в виде процента от суммарного значения. Результаты вычисления процентов отображаются в таблице в дополнительном столбце или дополнительной строке.</p> <div>i Примечание Значения в процентах вычисляются для выбранной меры по отношению к итоговым результатам для этой меры в таблице или в разрыве. Для вычисления процентного значения одной меры по отношению к другой мере необходимо создать формулу.</div>
Default	Применяет функцию агрегирования по умолчанию к стандартной мере или функцию агрегирования по базе данных для интеллектуальной меры.

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

3.2 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", необходимо добавить в отчет доход от каждой продажи. Для получения данного значения при вычислении $\text{[Доход от продаж]} / \text{[Проданное количество]}$ выполняется деление дохода на количество проданных товаров, что дает величину дохода с каждого товара.

3.2.1 Использование переменных для упрощения формул

Если формула сложная, можно использовать переменные, чтобы упростить ее.

С помощью переменных можно разбить сложную формулу на управляемые части, чтобы облегчить ее чтение, а также уменьшить число ошибок в процессе создания формул.

В формуле можно использовать ранее созданные переменные точно так же, как и остальные объекты отчета. Переменные отображаются в редакторе формул в папке "Переменные".

В формуле можно вводить имя переменной или перетаскивать переменную на панель инструментов "Формула", как и в случае с другими объектами отчета.

3.3 Работа с функциями

В пользовательском вычислении содержатся только объекты отчетов, например $\text{[Доход от продаж]} / \text{[Количество продаж]}$. 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 $\text{Sum}(\text{[Доход от продаж]})$ 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 \[страница 15\]](#)

[Функции \[страница 40\]](#)

3.3.1 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.

3.3.2 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.

3.3.3 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":

Квартальные доходы для Иллинойса

Year	Quarter	Sales revenue
2001	Q1	\$256 454
2001	Q2	\$241 458
2001	Q3	\$107 006
2001	Q4	\$133 306
2001	Total	\$738 223,80

Year	Quarter	Sales revenue
2002	Q1	\$334 297
2002	Q2	\$254 722
2002	Q3	\$230 573
2002	Q4	\$331 067
2002	Total	\$1 150 658,80

Year	Quarter	Sales revenue
2003	Q1	\$255 658
2003	Q2	\$354 724
2003	Q3	\$273 186
2003	Q4	\$250 517
2003	Total	\$1 134 085,40

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
2001	Q1	\$2 660 700	0,07
2001	Q2	\$2 279 003	0,06
2001	Q3	\$1 367 841	0,04
2001	Q4	\$1 788 580	0,05
2002	Q1	\$3 326 172	0,09
2002	Q2	\$2 840 651	0,08
2002	Q3	\$2 879 303	0,08
2002	Q4	\$4 186 120	0,12
2003	Q1	\$3 742 989	0,10
2003	Q2	\$4 006 718	0,11
2003	Q3	\$3 953 395	0,11
2003	Q4	\$3 356 041	0,09
		Сумма:	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.

2001			
Year	Quarter	Sales revenue	Percentage
2001	Q1	\$2 660 700	0,33
2001	Q2	\$2 279 003	0,28
2001	Q3	\$1 367 841	0,17
2001	Q4	\$1 788 580	0,22
		Сумма:	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.

2001	0,22
------	------

Year	Quarter	Sales revenue
2001	Q1	\$2 660 700
2001	Q2	\$2 279 003
2001	Q3	\$1 367 841
2001	Q4	\$1 788 580

2002	0,36
------	------

Year	Quarter	Sales revenue
2002	Q1	\$3 326 172
2002	Q2	\$2 840 651
2002	Q3	\$2 879 303
2002	Q4	\$4 186 120

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))
```

2001

Year	Quarter	Sales revenue	Percentage of Total	Percentage of Year
2001	Q1	\$2 660 700	0,07	0,33
2001	Q2	\$2 279 003	0,06	0,28
2001	Q3	\$1 367 841	0,04	0,17
2001	Q4	\$1 788 580	0,05	0,22

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 \[страница 26\]](#)

3.3.3.1 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.

3.3.4 Операторы функций и формул

Операторы связывают различные элементы формул.

Формулы могут содержать математические, условные, логические, специальные функциональные операторы или операторы расширенного синтаксиса.

3.3.4.1 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.

i 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".

3.3.4.2 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.

3.3.4.3 Логические операторы

К логическим операторам относятся следующие: And, Or, Not, Between и Inlist.

Логические операторы используются в булевых выражениях, возвращающих значения True или False.

3.3.4.4 Контекстные операторы

Контекстные операторы образуют часть расширенного синтаксиса вычислений.

Расширенный синтаксис позволяет определять, какие измерения будут учитываться формулой или мерой при вычислении.

3.3.4.5 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 `;`.

4 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 \[страница 18\]](#)

[Контекст вывода \[страница 19\]](#)

4.1 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
Верхние колонтитулы разделов и нижние колонтитулы блоков	Год
Строки блока	"Год", "Заказчик"

Другими словами, в верхних колонтитулах 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.

4.2 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
2001	Q1	\$2 660 699,50
2001	Q2	\$2 279 003,00
2001	Q3	\$1 367 840,70
2001	Q4	\$1 788 580,40
2001		
	Минимум:	\$1 367 840,70

Year	Quarter	Sales revenue
2002	Q1	\$3 326 172,20
2002	Q2	\$2 840 650,80
2002	Q3	\$2 879 303,00
2002	Q4	\$4 186 120,00
2002		
	Минимум:	\$2 840 650,80

Year	Quarter	Sales revenue
2003	Q1	\$3 742 988,90
2003	Q2	\$4 006 717,50
2003	Q3	\$3 953 395,30
2003	Q4	\$3 356 041,10
2003		
	Минимум:	\$3 356 041,10

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
2001	Q1	\$2 660 699,50	\$1 367 840,70
2001	Q2	\$2 279 003,00	\$1 367 840,70
2001	Q3	\$1 367 840,70	\$1 367 840,70
2001	Q4	\$1 788 580,40	\$1 367 840,70
2002	Q1	\$3 326 172,20	\$2 840 650,80
2002	Q2	\$2 840 650,80	\$2 840 650,80
2002	Q3	\$2 879 303,00	\$2 840 650,80
2002	Q4	\$4 186 120,00	\$2 840 650,80
2003	Q1	\$3 742 988,90	\$3 356 041,10
2003	Q2	\$4 006 717,50	\$3 356 041,10
2003	Q3	\$3 953 395,30	\$3 356 041,10
2003	Q4	\$3 356 041,10	\$3 356 041,10

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".

4.3 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	Итого: 8000
------	-------------

Заказчик	Доход
Харрис	1000
Джонс	3000
Уолш	4000
Итого:	8000

Итого в отчете: 8000

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

Мера	Значение	Контекст
Итого в отчете	20000	Итоговое значение общего дохода в отчете
Итоговое значение в верхнем колонтитуле	8000	Год
Итого по всем заказчикам	1000, 3000, 4000	Год; Заказчик
Итоговое значение в нижнем колонтитуле	8000	Год

Связанные сведения

[Сведения о контекстах вычисления \[страница 18\]](#)

[Изменение контекста вычислений по умолчанию с расширенным синтаксисом \[страница 26\]](#)

4.3.1 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.

4.3.2 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.

4.3.3 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

4.3.4 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	

4.3.5 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

4.4 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 \[страница 247\]](#)

[Операторы расширенного синтаксиса \[страница 27\]](#)

4.4.1 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.

4.4.1.1 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.

i Примечание

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

4.4.1.2 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]).

4.4.1.3 Контекстный оператор `ForAll`

Контекстный оператор `ForAll` удаляет из контекста измерения.

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 Total
2001	Q1	\$2 660 700	\$8 096 124
2001	Q2	\$2 279 003	\$8 096 124
2001	Q3	\$1 367 841	\$8 096 124
2001	Q4	\$1 788 580	\$8 096 124
2002	Q1	\$3 326 172	\$13 232 246
2002	Q2	\$2 840 651	\$13 232 246
2002	Q3	\$2 879 303	\$13 232 246
2002	Q4	\$4 186 120	\$13 232 246
2003	Q1	\$3 742 989	\$15 059 143
2003	Q2	\$4 006 718	\$15 059 143
2003	Q3	\$3 953 395	\$15 059 143
2003	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 ([Квартал]), что будет выглядеть следующим образом:

```
Sum([Доход от продаж] ForAll ([Квартал]))
```

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

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

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

5 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".

i 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.

5.1 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.

5.1.1 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`.

5.2 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].

5.3 Интеллектуальные меры и язык SQL

5.3.1 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

```

5.4 Интеллектуальные меры и формулы

5.4.1 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]

i 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]

5.4.2 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 \[страница 29\]](#)

5.5 Интеллектуальные меры и фильтры

5.5.1 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.

Если фильтр включен ...	Интеллектуальная мера оценивается следующим образом
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.

5.5.2 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

Страна	Доход
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.

5.5.3 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
Итог:	3,678,359.05
Сумма:	3,678,359.05

При фильтрации данных следующей таблицы по условию $\text{OrderAmountDel} > 60,000$

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

Страна	OrderAmountDel
Итог:	#UNAVAILABLE
Сумма:	3,598,141.36

5.5.4 Интеллектуальные меры и фильтры детализации

Фильтр детализации является однозначным.

Детализацию можно выполнить непосредственно из панели детализации.

5.5.5 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).

6 Функции, операторы и ключевые слова

6.1 Функции

Функции формул подразделяются на несколько категорий.

Примечание

Функции, ключевые слова и операторы не переводятся на следующие языки: китайский, японский, венгерский, корейский, 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

6.1.1 Пользовательские форматы

В таблицах можно использовать тип пользовательского формата, чтобы определить специальный формат для любой ячейки.

В функциях Web Intelligence действуют следующие определения символов дня, даты, календаря и времени.

Примечание

Пользовательские форматы нельзя применять к числам и датам в таблицах из интерфейса Web Intelligence HTML.

В следующей таблице представлены строки, используемые для создания пользовательских форматов:

Символ(ы)	Отображение(я)	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, a, b, c, \$, £, € (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'.

Символ(ы)	Отображение(я)	Пример
[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.
Day/date characters	(day, date)	
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"
Calendar characters	(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"

Символ(ы)	Отображение(я)	Пример
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".
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"
Time of day characters	(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.

Символ(ы)	Отображение(я)	Пример
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".
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".
z	Информация о часовом поясе в значении даты и времени выглядит следующим образом: GMT+/-HH:mm	

6.1.2 Функции агрегирования

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 \[страница 253\]](#)

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">AggregateAverageCounMaMiSum	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 \[страница 44\]](#)

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 \[страница 253\]](#)

[Оператор IncludeEmpty \[страница 237\]](#)

6.1.2.4 Count

Описание

Возвращает число значений в наборе значений

Группа функций

Агрегирование

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязателен
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, 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

`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 \[страница 237\]](#)

[Операторы Distinct/All \[страница 237\]](#)

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	Обязательное
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 раздела функция `First` возвращает первое значение в разделе.

Примеры

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

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 получают 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 \[страница 238\]](#)

[Оператор PointToPoint \[страница 239\]](#)

6.1.2.7 Функция Last

Описание

Возвращает последнее значение в измерении или мере

Группа функций

Агрегирование

Синтаксис

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

Ввод

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

Примечания

- При нахождении в нижнем колонтитуле разрыва функция Last возвращает последнее значение в разрыве.
- При размещении в нижнем колонтитуле раздела функция Last возвращает последнее значение в разделе.

Примеры

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

6.1.2.8 Функция Max

Описание

Возвращает наибольшее значение из набора значений

Группа функций

Агрегирование

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязателен
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

- 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.

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.

6.1.2.10 Функция Min

Описание

Возвращает наименьшее значение из набора значений

Группа функций

Агрегирование

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
agggregated_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].

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].

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])`

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

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)`

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

<i>Год</i>	<i>Квартал</i>	<i>Доход с продаж</i>	<i>Процентное соотношение</i>
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

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 перцентиль – 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.

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.

6.1.2.15 Функция 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:

Страна	Курорт	Доход	Промежуточное среднее
США	Hawaiian Club	1,479,660	835,420
США	Bahamas Beach	971,444	1,225,552
Франция	French Riviera	835,420	1,095,508

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

Страна	Курорт	Доход	Промежуточное среднее
США	Hawaiian Club	1,479,660	835,420
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 \[страница 237\]](#)

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 \[страница 237\]](#)

[Операторы Row/Col \[страница 240\]](#)

[Оператор IncludeEmpty \[страница 237\]](#)

[Оператор IncludeEmpty \[страница 237\]](#)

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
<i>Бостон</i>		
<i>Квартал</i>	<i>Доход с продаж</i>	<i>Текущий максимум</i>
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 \[страница 237\]](#)

[Операторы Row/Col \[страница 240\]](#)

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:

Страна	Курорт	Доход с продаж	Текущий минимум
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		
Квартал	Доход с продаж	Текущий минимум
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
<i>Бостон</i>		
<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 \[страница 237\]](#)

[Операторы Row/Col \[страница 240\]](#)

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 \[страница 237\]](#)

[Операторы Row/Col \[страница 240\]](#)

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

Параметр	Описание	Тип	Обязательный
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
<i>K2</i>		
<i>Город</i>	<i>Доход с продаж</i>	<i>Текущая сумма</i>
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 \[страница 237\]](#)

[Операторы Row/Col \[страница 240\]](#)

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 \[страница 83\]](#)

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.

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 потребоваться 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.

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 операторы 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 \[страница 79\]](#)

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 \[страница 80\]](#)

6.1.3 Символьные функции

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".

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 "{".

6.1.3.3 Функция Concatenation

Описание

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

i 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

Параметр	Описание	Тип	Обязательный
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".

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
repeating_string	Повторяющаяся строка	Строка	Да
num_repeats	Количество повторений	Число	Да

Примеры

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

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 \[страница 40\]](#)

6.1.3.5.1 Примеры `Format_string` для функции `FormatDate`

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

i Примечание

Эти образцы можно найти в диалоговом окне [Форматирование числа](#) в интерфейсах микроприложения или Web Intelligence Rich Client. Однако отображаемые образцы зависят от выбранного языка продукта 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

Образец	Синтаксис
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 \[страница 89\]](#)

[Пользовательские форматы \[страница 40\]](#)

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 \[страница 40\]](#)

6.1.3.7 Функция HTML Encode

Описание

Применяет к строке правила преобразования HTML

Группа функций

Символьные

Синтаксис

```
string HTMLEncode(html)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
html	Строка HTML	Строка	Да

Примеры

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

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".

6.1.3.9 Функция Left

Описание

Возвращает крайние левые символы строки.

i Примечание

Если выбран арабский языковой стандарт интерфейса (отображение и чтение справа налево), эта функция возвращает первые символы от логического начала строки.

Группа функций

Символьные

Синтаксис

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

Ввод

Параметр	Описание	Тип	Обязательный
string	Строка ввода	строка	Да
num_chars	Число символов, возвращаемых от начала строки.	число	Да

Пример

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

6.1.3.10 Функция LeftPad

Описание

Заполняет строку слева другой строкой.

Примечание

Если выбран арабский языковой стандарт интерфейса (отображение и чтение справа налево), эта функция заполняет строку другой строкой до ее логического начала.

Группа функций

Символьные

Синтаксис

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

Ввод

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".

6.1.3.11 Функция LeftTrim

Описание

Удаляет начальные пробелы из строки.

i Примечание

Если выбран арабский языковой стандарт интерфейса (отображение и чтение справа налево), эта функция удаляет первые символы пробела от логического начала строки.

Группа функций

Символьные

Синтаксис

```
string LeftTrim(trimmed_string)
```

Данные ввода

Параметр	Описание	Тип	Требуемая
trimmed_string	Строка для усечения	Строка	Да

Примеры

Функция `LeftTrim([Страна])` возвращает "Франция", если `[Страна]` имеет значение " Франция".

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".

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".

6.1.3.14 Функция Match

Описание

Определяет соответствие строки шаблону

Группа функций

Символьные

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Требуемая
test_string	Строка для проверки соответствия образцу текста	string	Да
pattern	Строка	Образец текста	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

6.1.3.15 Функция Pos

Описание

Возвращает начальную позицию образца текста в строке

Группа функций

Символьные

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Требуемая
test_string	Строка для проверки вхождения в нее образца текста	Строка	Да
pattern	Строка	Образец текста	Да

Примечания

- Если образец встречается больше одного раза, Pos возвращает позицию первого экземпляра.

Примеры

Pos ("Нью-Йорк"; "Нью") возвращает 1

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

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

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".

6.1.3.17 Функция Right

Описание

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

i Примечание

Если выбран арабский языковой стандарт интерфейса (отображение и чтение справа налево), эта функция возвращает первые символы от логического начала строки.

Группа функций

Символьные

Синтаксис

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

Данные ввода

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

Примеры

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

6.1.3.18 Функция RightPad

Описание

Заполняет строку справа другой строкой (добавляет строку к началу исходной строки).

i Примечание

Если выбран арабский языковой стандарт интерфейса (отображение и чтение справа налево), эта функция добавляет строку к первым символам от логического начала строки.

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".

6.1.3.19 Функция RightTrim

Описание

Удаляет пробелы из конца строки.

Примечание

Если выбран арабский языковой стандарт интерфейса (отображение и чтение справа налево), эта функция удаляет пробелы в логическом конце строки.

Группа функций

Символьные

Синтаксис

```
string RightTrim(trimmed_string)
```

Данные ввода

Параметр	Описание	Тип	Требуемая
trimmed_string	Строка для усечения	Строка	Да

Примеры

`RightTrim([Страна])` возвращает "Франция", если `[Страна]` – "Франция ".

6.1.3.20 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".

6.1.3.21 Функция Trim

Описание

Удаляет в начале и конце строки

Группа функций

Символьные

Синтаксис

```
string Trim(trimmed_string)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
string	Строка для удаления	Строка	Да

Примеры

Trim (" Великобритания ") возвращает "Великобритания".

6.1.3.22 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".

6.1.3.23 Функция UrlEncode

Описание

Применяет к строке правила кодировки URL

Группа функций

Символьные

Синтаксис

```
string UrlEncode(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".

6.1.3.24 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".

6.1.4 Функции даты и времени

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.

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.

6.1.4.3 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.

6.1.4.4 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.

6.1.4.5 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).

6.1.4.6 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.

6.1.4.7 Функция DaysBetween

Описание

Возвращает количество дней между двумя датами

Группа функций

Дата и время

Синтаксис

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

i 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.

6.1.4.8 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.

6.1.4.9 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.

6.1.4.10 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.

6.1.4.11 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.

6.1.4.12 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.

6.1.4.13 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.

6.1.4.14 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
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.

6.1.4.15 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.

i 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, InvoiceDate.	Дата	Да
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

6.1.4.16 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)
```

или

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

i 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.	Строка	Да
формат	Формат даты, который используется в строке. Используйте "INPUT_DATE_TIME", чтобы использовать формат предпочтительного языкового стандарта просмотра.	Строка	Да*

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

Примеры

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

В `ToDate ("Dec/02"; "Mmm/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.

i Примечание

- В случае с функцией `INPUT_DATE_TIME` оба значения даты и времени должны быть указаны в строке ввода `date_string`.
- Если формат `date_string` не может быть учтен как допустимая дата с определенным форматом, функция `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.

Связанные сведения

[Пользовательские форматы \[страница 40\]](#)

6.1.4.17 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).

6.1.4.18 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.

6.1.5 Функции провайдера данных

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.

6.1.5.2 Функция DataProvider

Описание

Возвращает имя поставщика данных, в котором содержится объект отчета

Группа функций

Поставщик данных

Синтаксис

```
string DataProvider(obj)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
obj	Объект отчета	Объект отчета	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 другой функции (напр., переменной измерения), не возвращающей имя объекта, в качестве параметра для `DataProvider`, функция `DataProvider` выдаст ошибку.

6.1.5.3 Функция `DataProviderKeyDate`

Описание

Возвращает ключевую дату поставщика данных

Группа функций

Источник 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.

Примеры

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

6.1.5.4 Функция DataProviderKeyDateCaption

Описание

Возвращает заголовок ключевой даты поставщика данных

Группа функций

Источник 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.

Примеры

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

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`.

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 "Юниверс" для поставщиков данных юниверса или "Персональные данные" для поставщиков персональных данных.
- Имя поставщика данных необходимо указывать в квадратных скобках.

Примеры

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

6.1.5.7 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.

6.1.5.8 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 \[страница 126\]](#)

6.1.5.9 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.

6.1.5.10 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 \[страница 126\]](#)

6.1.5.11 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.

6.1.5.12 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` \[страница 126\]](#)

6.1.5.13 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.

6.1.5.14 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.

6.1.5.15 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]`

6.1.5.16 Функция `UniverseName`

Описание

Возвращает имя юниверса, на котором основан поставщик данных

Группа функций

Источник данных

Синтаксис

```
string UniverseName(dp)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
dp	Поставщик данных	Поставщик данных	Да

Примечания

- 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 \[страница 126\]](#)

6.1.5.17 Функция UserResponse

Описание

Возвращает ответ на подсказку

Группа функций

Источник данных

Синтаксис

```
string UserResponse([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

- Имя поставщика данных необходимо указывать в квадратных скобках.
- Функцию `DataProvider` можно использовать для указания ссылки на поставщика данных.
- При выборе нескольких значений для ответа на подсказку функция возвращает строку, состоящую из списка значений (или основных ключей, если указан оператор `Index`), разделенных между собой точкой с запятой.

Примеры

`UserResponse ("Какой город?")` возвращает "Лос-Анджелес", если в поле запросов "Какой город?" было указано "Лос-Анджелес" подсказка.

`UserResponse ([Запрос продаж]; "Какой город?")` возвращает "Лос-Анджелес", если в поле "Какой город?" было указано "Лос Анжелес" подсказка в поставщике данных "Запрос продаж".

`UserResponse ([Запрос продаж]; "Какой город?"; Индекс)` возвращает 23, если в поле запросов "Какой город" было указано "Лос-Анджелес" подсказка в поставщике данных "Запрос продаж", а первичный ключ значения "Лос-Анджелес" в базе данных – 23.

6.1.6 Функции документа

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".

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.

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.

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.

6.1.6.5 Функция DocumentName

Описание

Возвращает имя документа

Группа функций

Документ

Синтаксис

```
string DocumentName()
```

Примеры

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

6.1.6.6 Функция DocumentOwner

Описание

Возвращает имя для входа в систему/имя пользователя в стартовой панели ВІ 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.

6.1.6.7 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.

6.1.6.8 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.

6.1.6.9 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 требуется использовать. Он должен входить в документ. Если отчет не объявлен, используется текущий отчет.	Строка	Необходим obj или separator

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"

6.1.6.10 Функция PromptSummary

Описание

Возвращает текст подсказки и ответ пользователя на все подсказки в документе

Группа функций

Документ

Синтаксис

```
string PromptSummary()
```

Примеры

Ниже приведен пример вывода функции PromptSummary:

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

6.1.6.11 Функция QuerySummary

Описание

Возвращает информацию о запросах в документе

Группа функций

Документ

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
dp	Поставщик данных	Поставщик данных	Нет

Примечания

- Имя поставщика данных необходимо указывать в квадратных скобках.

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

6.1.6.12 Функция ReportFilter

Описание

Возвращает фильтры, примененные к объекту или отчету

Группа функций

Документ

Синтаксис

```
string ReportFilter(obj)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
obj	Объект отчета	Объект отчета	Да

Примеры

Функция `ReportFilter([Страна])` возвращает значение "США", если для объекта "Страна" задан фильтр отчета, которым допускаются только значения "США".

6.1.6.13 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))
```

6.1.7 Логические функции

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
number	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.

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`.

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 \[страница 214\]](#)

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 \[страница 214\]](#)

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 \[страница 214\]](#)

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 \[страница 214\]](#)

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 \[страница 214\]](#)

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` [\[страница 214\]](#)

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 \[страница 214\]](#)

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
number	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 опускает дробные части десятичных чисел.

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 \[страница 214\]](#)

6.1.8 Числовые функции

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.

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.

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.

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 \[страница 252\]](#)

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 \[страница 252\]](#)

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
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

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 \[страница 252\]](#)

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 \[страница 252\]](#)

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.

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.

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.

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

Ln (10) возвращает 2,3.

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

$\text{Log}(125; 5)$ возвращает 3.

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

$\text{Log10}(100)$ возвращает 2.

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

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.

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 \[страница 235\]](#)

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 \[страница 252\]](#)

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.

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).

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.

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.

6.1.8.22 Функция ToNumber

Описание

Преобразует строку в число

Группа функций

Числовые

Синтаксис

```
num ToNumber(string)
```

или

Данные ввода

Параметр	Описание	Тип	Обязательный
string	Число в качестве строки	Строка	Да

Примечания

Если строка не является числом, функция `ToNumber` возвращает `#ERROR`.

Примеры

`ToNumber ("45")` возвращает 45.

6.1.8.23 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 обрезаются число	Число	Да

Примечания

Например

`Truncate (3,423; 2)` возвращает 3,42.

Связанные сведения

[Округление и усечение чисел \[страница 252\]](#)

6.1.9 Функции множества

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

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

i 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 \[страница 44\]](#)

[Average \[страница 51\]](#)

[Count \[страница 52\]](#)

[Функция Max \[страница 58\]](#)

[Функция Min \[страница 61\]](#)

[Функция Sum \[страница 81\]](#)

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 \[страница 44\]](#)

[Average \[страница 51\]](#)

[Count \[страница 52\]](#)

[Функция Max \[страница 58\]](#)

[Функция Min \[страница 61\]](#)

[Функция Sum \[страница 81\]](#)

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

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 \[страница 44\]](#)

[Average \[страница 51\]](#)

[Count \[страница 52\]](#)

[Функция Max \[страница 58\]](#)

[Функция Min \[страница 61\]](#)

[Функция Sum \[страница 81\]](#)

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

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".

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, Lag возвращает distance позиций после member. Если distance отрицательное число, 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 \[страница 44\]](#)

[Average \[страница 51\]](#)

[Count \[страница 52\]](#)

[Функция Max \[страница 58\]](#)

[Функция Min \[страница 61\]](#)

[Функция Sum \[страница 81\]](#)

6.1.9.8 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 \[страница 44\]](#)

[Average \[страница 51\]](#)

[Count \[страница 52\]](#)

[Функция Max \[страница 58\]](#)

[Функция Min \[страница 61\]](#)

[Функция Sum \[страница 81\]](#)

6.1.9.9 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 perdioid
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 \[страница 44\]](#)

[Average \[страница 51\]](#)

[Count \[страница 52\]](#)

[Функция Max \[страница 58\]](#)

[Функция Min \[страница 61\]](#)

[Функция Sum \[страница 81\]](#)

6.1.10 Прочие функции

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".

6.1.10.2 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.

6.1.10.3 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".

6.1.10.4 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].

6.1.10.5 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)".

6.1.10.6 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` \[страница 213\]](#)

6.1.10.7 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)".

6.1.10.8 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` \[страница 213\]](#)

6.1.10.9 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` \[страница 212\]](#)

[Функция `GetDominantPreferredViewingLocale` \[страница 210\]](#)

6.1.10.10 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, если bool_value имеет значение 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 этом 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 [Доход с продаж]>1000000 Then "Высокий доход" возвращает "Высокий доход" только для тех строк, где значение дохода превышает 1 000 000.

If [Доход с продаж] >1000000 Then "Высокий доход" Else [Доход] возвращает "Высокий доход" для всех строк, где значение дохода превышает 1 000 000, для всех остальных строк просто выводится значение дохода.

If [Доход с продаж]>1000000 Then "Высокий доход" Else "Низкий доход" возвращает "Высокий доход" для всех строк, где значение дохода превышает 1 000 000 и "Низкий доход" для строк, где значение дохода не достигает 1 000 000.

If [Доход с продаж]>1000000 Then "Высокий доход" ElseIf [Доход с продаж] > 800000 Then "Средний доход" Else "Низкий доход" возвращает "Высокий доход" для строк, где значение дохода превышает 1 000 000, "Средний доход" для строк, где значение дохода находится в пределах 800 000 и 1 000 000 и "Низкий доход" во всех остальных случаях.

Связанные сведения

[Функция If \[страница 216\]](#)

[Оператор And \[страница 231\]](#)

[Оператор Between \[страница 232\]](#)

[Оператор InList \[страница 233\]](#)

[Оператор Or \[страница 231\]](#)

[Оператор Not \[страница 232\]](#)

6.1.10.11 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 \[страница 214\]](#)

6.1.10.12 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.

6.1.10.13 Функция NameOf

Описание

Возвращает имя объекта

Группа функций

Разные

Синтаксис

```
string NameOf(obj)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
obj	Любой объект отчета	Объект отчета	Да

Примечания

Функция `NameOf` отображается в заголовках столбцов и строк в отчете.

Примеры

Функция `NameOf([Дата бронирования])` возвращает "Дата бронирования".

6.1.10.14 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].

6.1.10.15 Функция NumberOfPages

Описание

Возвращает количество страниц в отчете

Группа функций

Разные

Синтаксис

```
integer NumberOfPages()
```

Примеры

Примеры `NumberOfDataPages()` возвращает 2, если в отчете содержатся две страницы.

6.1.10.16 Функция Page

Описание

Возвращает номер текущей страницы в отчете.

Группа функций

Разные

Синтаксис

```
integer Page()
```

Например

Page () возвращает 2, если отображается на второй странице отчета.

6.1.10.17 Функция Previous

Описание

Возвращает предыдущее значение объекта

Группа функций

Разные

Синтаксис

```
input_type Previous(измерение|мера|Self [;Row|col] [;(reset_dims)] [;offset] [;NotNull])
```

Данные ввода

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
смещение	Указывает значение измерения или меры, то есть строки со смещением, предшествующие текущей строке	Целое	Нет (по умолчанию 1)
NoNull	Указывает функции на то, что необходимо возвращать первое ненулевое значение, начиная со смещения	Ключевое слово	Нет

Примечания

- По умолчанию смещение равно 1. `Previous ([Revenue] ; 1)` и `Previous ([Revenue])` функционально одно и то же.
- В случае включения аргумента `NoNull` 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 или фильтров к формулам, использующим функцию `Previous`, не поддерживается.
- 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	Предыдущий
США	5 000 000	
Великобритания	2 000 000	США
Франция	2 100 000	Великобритания

Previous ([Revenue]) возвращает следующие значения в следующей таблице:

Страна	Доход	Предыдущий
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	Предыдущий
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	Предыдущий	2005	Предыдущий
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	Предыдущий
США	Север	5 000 000	
	Юг	7 000 000	5 000 000
США		12 000 000	

Страна	Регион	Доход	Предыдущий
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 ; NoNull` возвращает следующие значения в следующей таблице:

Год	Квартал	Доход	Предыдущий
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 \[страница 263\]](#)

[Оператор Self \[страница 241\]](#)

6.1.10.18 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 квалифицирована как	Список возвращенных значений
Измерение или объект-описание	Arizona, California, Florida, Texas, New York

Переменная квалифициро- вана как	Список возвращенных значений
Показатель	(нулевое значение), California, Florida, Texas, New York

6.1.10.19 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: сортировки, которые применяются для измерений срезов, и порядок, в котором измерения срезов перечислены в функции.
- Измерение, которое используется как основной элемент раздела может быть указано как измерение срезов.
- Все измерения срезов должны быть представлены в блоке или в заголовке 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 \[страница 256\]](#)

[Сравнение значений с помощью функции `RelativeValue` \[страница 263\]](#)

6.1.10.20 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".

6.1.10.21 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.

6.1.10.22 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".

6.2 Операторы функций и формул

Операторы связывают различные элементы формул.

Формулы могут содержать математические, условные, логические, специальные функциональные операторы или операторы расширенного синтаксиса.

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.

i 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".

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.

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.

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.

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".

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".

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 \[страница 214\]](#)

[Оператор Where \[страница 242\]](#)

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 \[страница 214\]](#)

[Оператор Where \[страница 242\]](#)

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 `;`.

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

[Count \[страница 52\]](#)

[Операторы Distinct/All \[страница 237\]](#)

[Функция NoFilter \[страница 219\]](#)

[Операторы All/Drill \[страница 235\]](#)

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

6.2.4.3 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 \[страница 178\]](#)

6.2.4.4 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 \[страница 63\]](#)

6.2.4.5 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

[Count \[страница 52\]](#)

6.2.4.6 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 \[страница 51\]](#)

[Count \[страница 52\]](#)

[Функция RunningAverage \[страница 67\]](#)

[Функция RunningCount \[страница 69\]](#)

6.2.4.7 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 \[страница 141\]](#)

[Функция RefValueUserResponse \[страница 138\]](#)

6.2.4.8 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 \[страница 55\]](#)

6.2.4.9 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 \[страница 221\]](#)

6.2.4.10 Оператор NotOnBreak

Оператор `NotOnBreak` используется с функцией `Interpolation`.

Описание

С помощью оператора `NotOnBreak` функция `Interpolation` игнорирует разбиения разделов и блоков.

Связанные сведения

[Функция Interpolation \[страница 55\]](#)

6.2.4.11 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 \[страница 55\]](#)

6.2.4.12 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 \[страница 63\]](#)
[Функция RunningAverage \[страница 67\]](#)
[Функция RunningCount \[страница 69\]](#)
[Функция RunningMax \[страница 71\]](#)
[Функция RunningMin \[страница 73\]](#)
[Функция RunningProduct \[страница 75\]](#)
[RunningSum \[страница 77\]](#)

6.2.4.13 Оператор Self

Оператор Self используется с функцией Previous.

Описание

Указывает функции Previous на предыдущую ячейку, если она не содержит объект отчета.

Примеры

$5 + \text{Previous}(\text{Self})$ возвращает последовательность 5, 10, 15, 20, 25, 30...

$1 + 0.5 * \text{Previous}(\text{Self})$ возвращает последовательность 1, 1.5, 1.75, 1.88...

Связанные сведения

[Функция Previous \[страница 221\]](#)

6.2.4.14 Оператор Where

Описание

Оператор where ограничивает данные, используемые для вычисления меры.

Примеры

Формулой $\text{Average}([\text{Доход с продаж}]) \text{ Where } ([\text{Страна}] = \text{"США"})$ вычисляется средний доход с продаж для страны "США".

Формулой $\text{Average}([\text{Доход с продаж}]) \text{ Where } ([\text{Страна}] = \text{"США"} \text{ Or } [\text{Страна}] = \text{"Франция"})$ вычисляется средний доход с продаж для страны "США" или "Франция".

Формулой $[\text{Доход}] \text{ Where } (\text{Not } ([\text{Страна}] \text{ Inlist } (\text{"США"; "Франция"})))$ вычисляется доход для стран, отличных от "США" и "Франция".

Переменная [Высокий доход] описана формулой $[\text{Доход}] \text{ Where } [\text{Доход} > 500000]$. 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 $\text{Average}([\text{Высокий доход}])$ возвращает среднее значение всех доходов, превышающих 500000.

Связанные сведения

[Оператор And \[страница 231\]](#)

[Оператор Between \[страница 232\]](#)

[Оператор InList \[страница 233\]](#)

[Оператор Or \[страница 231\]](#)

[Оператор Not \[страница 232\]](#)

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.
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.

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
	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.

i Примечание

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

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]).

6.2.5.3 Контекстный оператор ForAll

Контекстный оператор ForAll удаляет из контекста измерения.

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 Total
2001	Q1	\$2 660 700	\$8 096 124
2001	Q2	\$2 279 003	\$8 096 124
2001	Q3	\$1 367 841	\$8 096 124
2001	Q4	\$1 788 580	\$8 096 124
2002	Q1	\$3 326 172	\$13 232 246
2002	Q2	\$2 840 651	\$13 232 246
2002	Q3	\$2 879 303	\$13 232 246
2002	Q4	\$4 186 120	\$13 232 246
2003	Q1	\$3 742 989	\$15 059 143
2003	Q2	\$4 006 718	\$15 059 143
2003	Q3	\$3 953 395	\$15 059 143
2003	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 ([Квартал]), что будет выглядеть следующим образом:

```
Sum([Доход от продаж] ForAll ([Квартал]))
```

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

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

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

6.2.6 Операторы наборов

Операторы наборов работают с элементами в иерархических данных.

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.

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.

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.

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.

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
	K3	\$1,367,841	\$8,096,124
	K4	\$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]).

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]).

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]).

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	Эта функция округляет или обрезает до десятичных знаков <параметра>. Примеры. Round (3,13; 1) возвращает 3,1 Round (3,157; 2) возвращает 3,16
0	Эта функция округляет или обрезает до ближайшего целого числа Примеры: Truncate (3,7; 0) возвращает 3 Truncate (4,164; 0) возвращает 4

Параметр	Описание
< 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>

i Примечание

Числа представляются в формате двойной точности с плавающей запятой, определены точно от 15 до 17 цифр.

Связанные сведения

[Функция Round \[страница 180\]](#)

[Функция Truncate \[страница 185\]](#)

[EuroConvertTo \[страница 167\]](#)

[Функция EuroConvertFrom \[страница 165\]](#)

[Функция EuroFromRoundError \[страница 168\]](#)

[Функция EuroToRoundError \[страница 170\]](#)

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.

i 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
Дженет 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], [Дженет 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.

7 Устранение связанных с формулами неполадок

7.1 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.

7.2 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.

7.2.1 #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` \[страница 226\]](#)

7.2.2 #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).

7.2.3 #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 Юниверсе 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.

7.2.4 #DIV/0

Сообщение #ДЕЛЕНИЕ/0 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/0 для этого квартала, поскольку в формуле была предпринята попытка деления на ноль; т.е. деление дохода на нулевое число проданных объектов.

7.2.5 #ERROR

#ERROR – сообщение об ошибке по умолчанию, которое отображается при возникновении ситуации, которой не присвоено никакое другое сообщение об ошибке.

7.2.6 #EXTERNAL

Ошибка #EXTERNAL возникает в случае, когда формула ссылается на внешнюю функцию, которую нельзя использовать в Web Intelligence.

7.2.7 #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.

7.2.8 #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.

7.2.9 #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

7.2.10 #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.

7.2.11 #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).

7.2.12 #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.

7.2.13 #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.

7.2.14 #RECURSIVE

Ошибка #RECURSIVE возникает при невозможности выполнить вычисление из-за циклической зависимости.

Пример

Использование функции NumberOfPages()

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, функции нужен точный размер отчета, но на размер отчета влияет размер ячейки, который определяется содержимым ячейки.

7.2.15 #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.

7.2.16 #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.

7.2.17 #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.

7.2.18 #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.

7.2.19 #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.

8 Сравнение значений с помощью функций

8.1 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 \[страница 221\]](#)

[Функция RelativeValue \[страница 226\]](#)

[Сравнение значений с помощью функции RelativeValue \[страница 263\]](#)

8.2 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 \[страница 226\]](#)

8.2.1 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	**

Год	Квартал	
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 \[страница 226\]](#)

8.2.2 Измерения срезов и разделы

Измерение среза может находиться в основной ячейке раздела отчета.

Пример

Измерения среза в заголовке 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	
K3	Джонс	2000	
K4	Джонс	1700	

2008

Квартал	Продавец	Доход	RelativeValue
K1	Смит	5000**	3000*

Квартал	Продавец	Доход	RelativeValue
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 \[страница 226\]](#)

8.2.3 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	

Год	Квартал	Продавец	Доход	RelativeValue
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	*
K1	2008	**
K2	2007	***
K2	2008	****
K3	2007	*****
K3	2008	*****
K4	2007	*****

Квартал	Год	
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 \[страница 226\]](#)

8.2.4 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	****
2008	K3	

Знак "*" в таблицах отображает порядок сортировки.

Связанные сведения

[Функция RelativeValue \[страница 226\]](#)

8.2.5 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 \[страница 226\]](#)

Важные положения об отказе от ответственности в отношении правовых вопросов

Образцы исходного кода

Любые фрагменты программ и/или строки кода («Код»), содержащиеся в настоящей документации, являются лишь примерами и не предназначены для использования в среде продуктивной системы. Этот Код предназначен только для объяснения и иллюстрации синтаксиса и правил составления текста программ. Компания SAP не гарантирует правильность и полноту приведенного Кода и не несет ответственности за ошибки или ущерб, вызванные использованием Кода, за исключением тех случаев, когда этот ущерб является результатом намеренных действий или халатности со стороны компании SAP.

Гендерно-нейтральный язык

В документации SAP везде, где это возможно, употребляются гендерно-нейтральные выражения. В зависимости от контекста используются либо безличные конструкции, касающиеся действий адресата (в английском варианте — личные конструкции с «you»), либо семантически нейтральные существительные (такие как «лицо» или «рабочие дни»). Однако в таких случаях, когда по отношению к представителям обоих полов нужно употребить местоимение третьего лица единственного числа или употребление нейтрального существительного невозможно, SAP сохраняет право использования формы мужского рода. Это обеспечивает удобство чтения документации.

Гиперссылки на ресурсы в Интернете

Документация SAP может содержать гиперссылки на ресурсы в Интернете. Эти гиперссылки указывают, где можно найти дополнительную информацию. SAP не гарантирует доступность и правильность такой дополнительной информации или ее пригодность для каких-либо целей. 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. Все ссылки для ясности разделены по категориям (см. <https://help.sap.com/viewer/disclaimer>).



**go.sap.com/registration/
contact.html**

© SAP SE или аффилированная компания SAP, 2018. Все права защищены.

Полное или частичное воспроизведение или передача в какой-либо форме и в каких-либо целях настоящей публикации без явным 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/corporate/en/legal/copyright.html>.