



Использование функций, формул и вычислений в Web Intelligence

- SAP BusinessObjects Business Intelligence Suite 4.1 Support Package 1

2013-09-19

Авторские права

© SAP AG или аффилированная компания SAP, 2013. 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 AG запрещены. Информация, содержащаяся в настоящей публикации, может быть изменена без предварительного уведомления. Некоторые программные продукты, предлагаемые на рынке компанией SAP AG 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 AG и ее аффилированными компаниями ("SAP Group") исключительно в информационных целях, без предоставления каких-либо гарантий. SAP Group не несет ответственности за ошибки или пропуски в настоящих материалах. Гарантии, если таковые предоставляются, в отношении продуктов и услуг SAP Group 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 другие указанные продукты и услуги SAP, как и соответствующие логотипы являются товарными знаками или зарегистрированными товарными знаками SAP AG на территории Германии и других стран. Для получения дополнительных сведений и уведомлений о товарных знаках см. <http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark>.

2013-09-19

Содержание

Глава 1	История документа: функции, формулы и вычисления Web Intelligence.....	7
Глава 2	О данном руководстве.....	9
Глава 3	Использование стандартных и пользовательских вычислений.....	11
3.1	Стандартные вычисления.....	11
3.2	Использование формул для создания пользовательских вычислений.....	12
3.2.1	Использование переменных для упрощения формул.....	12
3.3	Работа с функциями.....	13
3.3.1	Включение функций в ячейки.....	13
3.3.2	Синтаксис функций.....	14
3.3.3	Примеры функций.....	14
3.3.4	Операторы функций и формул.....	19
Глава 4	Определение контекстов вычисления.....	23
4.1	Определенные контексты вычислений.....	23
4.1.1	Контекст ввода.....	24
4.1.2	Контекст вывода.....	24
4.2	Контексты вычислений по умолчанию.....	26
4.2.1	Контексты по умолчанию в вертикальной таблице.....	28
4.2.2	Контексты по умолчанию в горизонтальной таблице.....	28
4.2.3	Контексты по умолчанию в кросс-таблице.....	29
4.2.4	Контексты по умолчанию в разделе.....	30
4.2.5	Контексты по умолчанию в разбиении.....	31
4.3	Изменение контекста вычислений по умолчанию с расширенным синтаксисом.....	32
4.3.1	Операторы расширенного синтаксиса.....	32
4.3.2	Ключевые слова расширенного синтаксиса.....	36
Глава 5	Расчет значений с помощью интеллектуальных мер.....	45
5.1	Наборы группировок и интеллектуальные меры.....	45
5.1.1	Управление наборами группировок.....	46

5.2	Интеллектуальные меры и область анализа.....	47
5.3	Интеллектуальные меры и язык SQL.....	47
5.3.1	Наборы группировок и оператор UNION.....	47
5.4	Интеллектуальные меры и формулы.....	49
5.4.1	Интеллектуальные меры и измерения, содержащие формулы.....	49
5.4.2	Интеллектуальные меры в формулах.....	50
5.5	Интеллектуальные меры и фильтры.....	50
5.5.1	Ограничение связано с интеллектуальными мерами и фильтрами.....	50
5.5.2	Интеллектуальные меры и фильтры для измерений.....	51
5.5.3	Фильтрация интеллектуальных мер.....	52
5.5.4	Интеллектуальные меры и фильтры детализации.....	53
5.5.5	Интеллектуальные меры и вложенные фильтры OR.....	53
Глава 6	Функции, операторы и ключевые слова.....	55
6.1	Функции.....	55
6.1.1	Пользовательские форматы.....	55
6.1.2	Функции агрегирования.....	60
6.1.3	Символьные функции.....	86
6.1.4	Функции даты и времени.....	101
6.1.5	Функции поставщика данных.....	113
6.1.6	Функции документа.....	124
6.1.7	Логические функции.....	131
6.1.8	Числовые функции.....	139
6.1.9	Функции множества.....	159
6.1.10	Дополнительные функции.....	168
6.2	Операторы функций и формул.....	187
6.2.1	Математические операторы.....	188
6.2.2	Условные операторы.....	188
6.2.3	Логические операторы.....	188
6.2.4	Специальные операторы функций.....	191
6.2.5	Операторы расширенного синтаксиса.....	199
6.2.6	Операторы наборов.....	203
6.3	Ключевые слова расширенного синтаксиса.....	204
6.3.1	Ключевое слово Block.....	204
6.3.2	Ключевое слово Body.....	205
6.3.3	Ключевое слово Break.....	206
6.3.4	Ключевое слово Report.....	207
6.3.5	Ключевое слово Section.....	208
6.4	Округление и усечение чисел.....	209
6.5	Ссылки на элементы и наборы элементов в иерархиях.....	210

Глава 7	Устранение связанных с формулами неполадок.....	213
7.1	Сообщения об ошибках формул и информационные сообщения.....	213
7.1.1	#COMPUTATION.....	213
7.1.2	#CONTEXT.....	213
7.1.3	#DATASYNC.....	214
7.1.4	#DIV/0.....	214
7.1.5	#ERROR.....	215
7.1.6	#EXTERNAL.....	215
7.1.7	#INCOMPATIBLE.....	215
7.1.8	#MIX.....	215
7.1.9	#MULTIVALUE.....	216
7.1.10	#N/A.....	216
7.1.11	#OVERFLOW.....	216
7.1.12	#PARTIALRESULT.....	217
7.1.13	#RANK.....	217
7.1.14	#RECURSIVE.....	218
7.1.15	#REFRESH.....	218
7.1.16	#REPFORMLA.....	218
7.1.17	#SECURITY.....	218
7.1.18	#SYNTAX.....	219
7.1.19	#TOREFRESH.....	219
7.1.20	#UNAVAILABLE.....	220
Глава 8	Сравнение значений с помощью функций.....	221
8.1	Сравнение значений с помощью функции Previous.....	221
8.2	Сравнение значений с помощью функции RelativeValue.....	221
8.2.1	Измерения срезов и функция RelativeValue.....	222
8.2.2	Измерения срезов и разделы.....	224
8.2.3	Порядок измерений срезов.....	226
8.2.4	Сортировка измерений срезов.....	228
8.2.5	Использование функции RelativeValue в кросс-таблицах.....	229
Приложение А	Дополнительная информация.....	231
Указатель		235

История документа: функции, формулы и вычисления Web Intelligence

В следующей таблице описываются наиболее важные изменения документа.

Версия	Дата	Описание
SAP BusinessObjects Interactive Analysis 4.0	30 ноября 2011 г.	Изначально обновленный Interactive Analysis. Первый выпуск документа
SAP BusinessObjects Interactive Analysis 4.0, пакет обновления 1	25 февраля 2011 г.	Пакет поддержки 1
BusinessObjects Web Intelligence 4.0, пакет обновления 2	15 июня 2011 г.	Пакет поддержки 2. Выполнен возврат к Web Intelligence
BusinessObjects Web Intelligence 4.0, пакет компонентов 3	20 февраля 2012 г.	Пакет компонентов 3
BusinessObjects Web Intelligence 4.1	10 мая 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

О данном руководстве

В руководстве *Использование функций, формул и вычислений в Web Intelligence* представлены подробные сведения о расширенных возможностях вычислений, доступных во время выполнения анализа данных. В нем также содержится справочник по синтаксису доступных функций и операторов.

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> <p>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.</p>

Вычисление	Описание
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 $\frac{[\text{Доход от продаж}]}{[\text{Проданное количество}]}$ выполняется деление дохода на количество проданных товаров, что дает величину дохода с каждого товара.

См. также

- [Определенные контексты вычислений](#)

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.

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 сделать процесс создания формул менее подверженным ошибкам.

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.

См. также

- [Упрощение формулы для дисперсии с помощью переменных](#)

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

3.3 Работа с функциями

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

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

См. также

- [Операторы функций и формул](#)

Операторы связывают различные элементы формул.

- [Функции](#)

Функции формул подразделяются на несколько категорий.

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( "eFashion";"Choose a State")
```

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

Квартальные продажи для Иллинойс

Year	Quarter	Sales revenue
2001	Q1	\$251 454
2001	Q2	\$241 453
2001	Q3	\$107 005
2001	Q4	\$133 335
2001	Total	\$733 250

Year	Quarter	Sales revenue
2002	Q1	\$254 367
2002	Q2	\$254 712
2002	Q3	\$235 573
2002	Q4	\$231 367
2002	Total	\$976 020

Year	Quarter	Sales revenue
2003	Q1	\$255 660
2003	Q2	\$254 714
2003	Q3	\$231 065
2003	Q4	\$230 917
2003	Total	\$972 356

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 Процентное соотношение ([Доход от продаж]).

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 [Доход от продаж]/Sum([Доход от продаж]) 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.

См. также

- [Изменение контекста вычислений по умолчанию с расширенным синтаксисом](#)

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

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)

Переменная	Формула
Variance	Sum([Difference Squared] In ([Quarter]))/([Number of Observations] – 1)

Окончательная формула принимает вид

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

что гораздо легче воспринимать. 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.

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 [Revenue]>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 `;`.

Определение контекстов вычисления

4.1 Определенные контексты вычислений

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

См. также

- [Контекст ввода](#)

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

- [Контекст вывода](#)

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

4.1.1 Контекст ввода

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

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

Пример: Определение контекста ввода

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

Часть отчета	Контекст ввода
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 ([Revenue] In ([Year]))
```

```
Sum ([Revenue] In ([Year];[Customer]))
```

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

4.1.2 Контекст вывода

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

Пример: Определение контекста вывода

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

Year	Quarter	Sales revenue
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 ([Revenue]) In ([Year])
```

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 ([Sales Revenue] In([Year];[Quarter])) In ([Year])
```

Действие этой формулы можно описать следующим образом: она 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".

4.2 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

См. также

- [Определенные контексты вычислений](#)

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.

- [Изменение контекста вычислений по умолчанию с расширенным синтаксисом](#)

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

4.2.1 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.

4.2.2 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.

4.2.3 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.
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

4.2.4 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 кросс-таблице.

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	

4.2.5 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

4.3 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.

4.3.1 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.

4.3.1.1 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 с 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	Доходы с 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	Доходы с 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	Доходы с 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 ([Sales Revenue] In ([Year];[Quarter])) In ([Year])
```

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

Примечание:

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

4.3.1.2 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	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

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

```
Max ([Sales Revenue] In ([Year];[Quarter])) In ([Year])
```

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

```
Max ([Sales Revenue] ForEach ([Quarter])) In ([Year])
```

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]).

4.3.1.3 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 итоговый 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([Sales Revenue] ForAll ([Quarter]))
```

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

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

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

4.3.2 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.

4.3.2.1 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([Sales revenue]) 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]).

4.3.2.2 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 ([Sales Revenue]) 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]).

4.3.2.3 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	Не применимо

Пример: Ключевое слово Break

В отчете отображаются год, квартал и доход от продаж:

Год	Квартал	Доходы с 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 ([Sales Revenue]) 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]).

4.3.2.4 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)	Не применимо
За пределами всех блоков или разделов	Не применимо

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" присутствует формула.

```
Average([Sales revenue] In Section)
```

Формула также присутствует в столбце "В среднем за первое полугодие".

```
Average ([Sales revenue]) In Block
```

Можно проследить за тем, как ключевое слово Block учитывает фильтр блока.

4.3.2.5 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	Доходы с 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 ([Sales Revenue]) 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.

4.3.2.6 Использование ключевых слов для обобщения отчетов

Ключевые слова расширенного синтаксиса защищают отчет от изменений.

При явном определении данных (измерение определяется с помощью операторов In, ForEach или ForAll) отчеты могут возвращать непредвиденные данные, если добавлять или удалять измерения. Это показано в следующем примере.

Пример: Использование ключевого слова Report для отображения процентного соотношения

В данном примере в блоке содержатся объекты "Год", "Квартал" и "Доход от продаж". Необходимо отобразить доходы за каждый год и квартал, а также процентное соотношение итогового дохода в отчете, который представляет каждый индивидуальный доход, как показано ниже.

Year	Quarter	Sales revenue	Percentage of Total
2001	Q1	\$2 660 700	7,31
2001	Q2	\$2 279 003	6,26
2001	Q3	\$1 367 841	3,76
2001	Q4	\$1 788 580	4,92
2002	Q1	\$3 326 172	9,14
2002	Q2	\$2 840 651	7,81
2002	Q3	\$2 879 303	7,91
2002	Q4	\$4 186 120	11,50
2003	Q1	\$3 742 989	10,29
2003	Q2	\$4 006 718	11,01
2003	Q3	\$3 953 395	10,86
2003	Q4	\$3 356 041	9,22
		Сумма:	100

Формула для столбца "Процентное соотношение итога" выглядит следующим образом.

```
(([Sales revenue]/(Sum([Sales revenue]) In Report)) * 100
```

В блоке ключевое слово Report включает в себя все данные в отчете, поэтому данная формула выглядит следующим образом.

```
(([Sales revenue]/Sum([Sales revenue] ForAll ([Year];[Quarter]))) * 100
```

Эта формула удаляет "Год" и "Квартал" из контекста вывода (то есть вычисляет итоговую сумму, так как в отчете нет других измерений). Затем в формуле каждый доход делится на итоговую сумму, чтобы найти его процентное соотношение от общего дохода.

Хотя в данной ситуации можно использовать оператор ForAll, предпочтительнее использовать ключевое слово Report. Причина Что произойдет, если добавить измерение "Месяц" в отчет позже? С помощью версии формулы, которая используется в ключевом слове Report, каждое

процентное соотношение вычисляется правильно, но версия, которая определяет измерения "Год" и "Квартал", сейчас не является верной.

Year	Quarter	Month	Sales revenue	Percentage of Total
2001	Q1	1	\$1 003 541,20	26,13
2001	Q1	2	\$630 073,20	29,97
2001	Q1	3	\$1 027 085,10	27,12
2001	Q2	4	\$895 259,80	28,10
2001	Q2	5	\$865 615,10	24,30
2001	Q2	6	\$517 818,50	21,77
2001	Q3	7	\$525 903,50	20,42
2001	Q3	8	\$173 756,40	11,11
2001	Q3	9	\$668 180,80	16,45
2001	Q4	10	\$655 206,40	18,04
2001	Q4	11	\$484 024,20	18,55
2001	Q4	12	\$649 349,80	21,01
2002	Q1	1	\$1 335 401,90	34,77
2002	Q1	2	\$609 012,80	28,97
2002	Q1	3	\$1 381 757,50	36,49
2002	Q2	4	\$1 068 308,90	33,53
2002	Q2	5	\$1 081 884,80	30,38
2002	Q2	6	\$690 457,10	29,03
2002	Q3	7	\$801 954,70	31,14
2002	Q3	8	\$581 093,50	37,15
2002	Q3	9	\$1 496 254,80	36,84
2002	Q4	10	\$1 545 871,80	42,57
2002	Q4	11	\$1 081 915,30	41,47
2002	Q4	12	\$1 558 332,90	50,43
2003	Q1	1	\$1 501 366,70	39,09
2003	Q1	2	\$863 451,90	41,07
2003	Q1	3	\$1 378 170,30	36,39
2003	Q2	4	\$1 222 329,40	38,37
2003	Q2	5	\$1 614 147,30	45,32
2003	Q2	6	\$1 170 240,80	49,20
2003	Q3	7	\$1 247 313,50	48,44
2003	Q3	8	\$809 365,40	51,74
2003	Q3	9	\$1 896 716,40	46,70
2003	Q4	10	\$1 430 300,10	39,39
2003	Q4	11	\$1 043 098,80	39,98
2003	Q4	12	\$882 642,20	28,56
			Сумма:	1 200

В чем причина? Проблема заключается в следующем.

```
Sum ([Sales Revenue] ForAll ([Year];[Quarter]))
```

Когда измерения "Год" и "Квартал" были единственными измерениями в отчете, это было эквивалентно итоговой сумме всех доходов. Как только было добавлено измерение "Месяц", данное выражение удалило объекты "Год" и "Квартал" из контекста вывода, но объект "Месяц" остался.

Теперь формула включает "разбиение" по месяцу. Другими словами, в каждой строке, где у объекта "Месяц" значение 1, данное выражение означает "итоговый доход всех месяцев со значением 1s". В каждой строке, где у объекта "Месяц" значение 2, данное выражение означает "итоговый доход всех месяцев со значением 2s". В результате процентное соотношение не соответствует ожидаемому.

Расчет значений с помощью интеллектуальных мер

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

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

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

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

Примечание:

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

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

Контексты вычисления в созданном запросе представляют «наборы группировок».

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,
  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]

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	10 000
США	Юг	15 000
США	Восток	14 000
США	Запад	12 000

Если в эту таблицу добавить дополнительный столбец с формулой

```
[Revenue] ForAll ([Region])
```

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.

См. также

- [Контекстный оператор ForAll](#)

Контекстный оператор ForAll удаляет из контекста измерения.

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.

Таблица 5-2: Принцип оценки интеллектуальных мер при наличии фильтра в отчете

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

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

Пример: Интеллектуальная мера и фильтр для измерения

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

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

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

Страна	Доход
США	971 444
Сумма:	971 444

Примечание:

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

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 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
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).

Функции, операторы и ключевые слова

6.1 Функции

Функции формул подразделяются на несколько категорий.

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
Разное	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 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 12,345 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 языковом стандарте части целого отделяются точкой) или "12,34" (если в 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%

Символ(ы)	Отображение(я)	Пример
	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 €"
[Red], [Blue], [Green], [Yellow], [Gray], [White], [Dark Red], [Dark Blue], [Dark Green]	Значение в указанном цвете.	"150" в формате #, # # 0 [Red] отображается как "150" #, # # 0 [Blue] дает "150"
Символы даты и времени	(день, дата)	
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".

Символ(ы)	Отображение(я)	Пример
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 отображается как "Понедельник 01"
Символы календаря	(месяц, год)	
M	Число месяцев без начальных нулей. Если номер месяца состоит из одного символа, то ноль в начало не добавляется.	"Январь" в формате M отображается как "1"
MM	Номер месяца с начальными нулями. Если номер месяца состоит из одного символа, то в начало добавляется ноль.	"Январь" в формате MM отображается как "01"
mmm	Сокращенное название месяца. Если в выбранном языковом стандарте названия должны выделяться заглавной буквой, первая буква будет заглавной.	"Январь" в формате mmm отображается как "Янв". Во французском языке этот месяц отображается как "jan".
Mmmm	Сокращенное название месяца. Первая буква делается заглавной во всех языковых стандартах.	"Январь" в формате mmmm 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".
yy	Две последние цифры года.	"2003" в формате yy отображается как "03"

Символ(ы)	Отображение(я)	Пример
yyyy	Все четыре цифры года.	"2003" в формате yyyy отображается как "2003"
Символы времени су ток	(часы, минуты, секунды, до/после полудня)	
hh:mm:ss a	Часы без дополнительных нулей, минуты и секунды с дополнительными нулями. Символ a обозначает АМ (до полудня) или РМ (после полудня) после значения времени, если такое обозначение доступно.	"21:05:03" в формате hh:mm:ss a отображается как "9:05:03 PM" (английский языковой стандарт)
H	Час в 24-часовом формате с отсчетом от 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 с отсчетом от 1. Значения меньше 10 состоят из одной цифры.	"21:00" в формате k отображается как "21". Допустимые значения: от 1 до 24.
kk	Час в 24-часовом формате с отсчетом от 01.	"21:00" в формате kk отображается как "21". Допустимые значения: от 01 до 24.
hh	Указание времени в 12-часовом формате.	"21:00" в формате hh отображается как "09"
HH:mm	Часы и минуты с дополнительными нулями.	"7.15 am" в формате HH:mm отображается как "07:15"
HH:mm:ss	Часы, минуты и секунды с дополнительными нулями.	"7.15 am" в формате HH:mm:ss отображается как "07:15:00"
mm:ss	Минуты и секунды с дополнительными нулями.	"07:15:03" в формате mm:ss отображается как "15:03"

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_setAggregate` 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.

См. также

- [Ссылки на элементы и наборы элементов в иерархиях](#)

Ссылки на элементы и наборы элементов в функциях оформляются с помощью синтаксиса `[иерархия]&path.function`.

6.1.2.2 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.

См. также

- [Ссылки на элементы и наборы элементов в иерархиях](#)

Ссылки на элементы и наборы элементов в функциях оформляются с помощью синтаксиса `[иерархия]&path.function`.

- [Оператор IncludeEmpty](#)

Оператор `IncludeEmpty` используется с функциями агрегирования.

6.1.2.3 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
<code>aggregated_data</code>	Любое измерение, мера, иерархия, уровень или набор элементов	Измерение, мера, иерархия, набор элементов	Да
<code>member_set</code>	Набор элементов, используемый для расчета числа	Набор элементов	Нет
<code>IncludeEmpty</code>	Включает пустые значения в вычисление	Ключевое слово	Нет
<code>Distinct All</code>	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].

См. также

- [Оператор IncludeEmpty](#)

Оператор `IncludeEmpty` используется с функциями агрегирования.

- [Операторы Distinct/All](#)

Операторы `Distinct/All` используются с функцией `Count`.

6.1.2.4 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.

6.1.2.5 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.

6.1.2.6 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.

6.1.2.7 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.

6.1.2.8 Функция Min

Описание

Возвращает наименьшее значение из набора значений

Группа функций

Агрегирование

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязателен
<code>aggregated_data</code>	Любое измерение, мера, иерархия, уровень или набор элементов	Измерение, мера, иерархия, уровень или набор элементов	Да
<code>member_set</code>	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, 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 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].

6.1.2.9 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	Обязательное
dimension measure	Любое измерение или мера	Мера	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].

6.1.2.10 Функция Percentage

Описание

Выражает значение меры в процентном соотношении его внедренного контекста

Группа функций

Агрегирование

Синтаксис

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

Данные ввода

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

Примеры

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

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

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

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

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

Год	Квартал	Доход с продаж	Процентное соотношение
2002	K1	2000	20

	K2	2000	20
	K3	5000	50
	K4	1000	10
2002	Сумма:	10000	100

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

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

6.1.2.11 Функция `Percentile`

Описание

Возвращает процентиль `nth` меры

Группа функций

Числовые

Синтаксис

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

Ввод

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

Примечания

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

Примеры

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

6.1.2.12 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.

6.1.2.13 Функция RunningAverage**Описание**

Возвращает скользящее среднее значение меры

Группа функций

Агрегирование

Синтаксис

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


Данные ввода

Параметр	Описание	Тип	Обязателен
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

Примечания

- С функцией `RunningAverage` можно использовать операторы контекста расширенного синтаксиса.
- Направление вычисления можно задать операторами `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 результаты, приведенные в следующей таблице:

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 результаты, приведенные в следующей таблице:

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

См. также

- [Оператор IncludeEmpty](#)

Оператор `IncludeEmpty` используется с функциями агрегирования.

- [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.

6.1.2.14 Функция RunningCount

Описание

Возвращает текущий счет набора чисел

Группа функций

Агрегирование

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательное
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

Примечания

- Контекстные операторы расширенного синтаксиса можно использовать с функцией `RunningCount`.
- Направление вычисления можно задать операторами `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])` 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	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

См. также

- [Оператор IncludeEmpty](#)

Оператор `IncludeEmpty` используется с функциями агрегирования.

- [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.

- [Оператор IncludeEmpty](#)

Оператор `IncludeEmpty` используется с функциями агрегирования.

- [Оператор IncludeEmpty](#)

Оператор `IncludeEmpty` используется с функциями агрегирования.

6.1.2.15 Функция `RunningMax`

Описание

Возвращает промежуточный максимум измерения или меры

Группа функций

Агрегирование

Синтаксис

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

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

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	Промежуточный максимум
Франция	French Riviera	835 420	835 420
США	Bahamas Beach	971 444	971 444
США	Hawaiian Club	1 479 660	1 479 660

См. также

- [Оператор IncludeEmpty](#)

Оператор `IncludeEmpty` используется с функциями агрегирования.

- 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.

6.1.2.16 Функция `RunningMin`

Описание

Возвращает промежуточный минимум измерения или меры

Группа функций

Агрегирование

Синтаксис

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

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

Примечания

- С функцией `RunningMin` можно использовать контекстные операторы расширенного синтаксиса.
- Направление вычисления можно задать операторами `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])` возвращает результаты в виде таблицы, представленной ниже:

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

См. также• [Оператор IncludeEmpty](#)

Оператор `IncludeEmpty` используется с функциями агрегирования.

• [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.

6.1.2.17 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:

Страна происхождения	Город	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:

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

См. также

- [Оператор IncludeEmpty](#)

Оператор `IncludeEmpty` используется с функциями агрегирования.

- [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.

6.1.2.18 Функция RunningSum

Описание

Возвращает текущую сумму измерения

Группа функций

Агрегирование

Синтаксис

```
num RunningSum(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

- Функцию RunningSum можно использовать с контекстными операторами расширенного синтаксиса.
- Направление вычислений 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

См. также

- [Оператор IncludeEmpty](#)

Оператор `IncludeEmpty` используется с функциями агрегирования.

- [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.

6.1.2.19 Функция ServerValue

Описание

Возвращает значение базы данных для меры

Группа функций

Агрегирование

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязателен
measure	Любая мера	Мера	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]`

6.1.2.20 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.

См. также

- [Функция Var](#)

6.1.2.21 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.

6.1.2.22 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.

6.1.2.23 Функция Var

Описание

Возвращает отклонение для меры

Группа функций

Агрегирование

Синтаксис

```
num Var (measure)
```

Данные ввода

Параметр	Описание	Тип	Обязателен
measure	Любая мера	Мера	Да

Примечания

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

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

Дисперсия – это возведенное в квадрат значение среднеквадратичного отклонения.

Можно использовать операторов контекста расширенного синтаксиса с функцией Var.

Примеры

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

См. также

- [Функция StdDev](#)

6.1.2.24 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.

См. также

- [Функция StdDevP](#)

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

Группа функций

Символьные

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Требуется
first_string	Первая строка	Строка	Да
second_string	Вторая строка	Строка	Да

Примечания

Для соединения строк также можно использовать оператор "+".

"Первый" + "Второй" возвращает "Первый Второй".

"Первый" + "Второй" + "Третий" возвращает "Первый Второй Третий".

Для включения нескольких измерений в функцию агрегирования можно использовать конкатенацию. Например, выражение `Count ([Продавец] + [Квартал] + [Курорт])` равнозначно синтаксису `Count (<Продавец>, <Квартал>, <Курорт>)`, разрешенному в Desktop Intelligence.

Примеры

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

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

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/мм/yyyy") возвращает "15/12/2005", если текущая дата – 15 декабря 2005 года.

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.
- Строки цветового форматирования (например, [Красный], [Голубой] и так далее) применить к функции `FormatNumber` невозможно.

Примеры

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

6.1.3.7 Функция `HTMLEncode`

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.

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.

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
<code>trimmed_string</code>	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).

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).

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
<code>padded_string</code>	Исходная строка	Строка	Да
<code>length</code>	Длина строки вывода	Число	Да
<code>right_string</code>	Строка, которую необходимо добавить в конец строки <code>padded_string</code>	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 UriEncode**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)
```

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_days)
```

Ввод

Параметр	Описание	Тип	Обязательный
start_date	Начальная дата	Дата	Да
num_days	Количество дней от начальной даты	Число	Да

Примечания

Параметр `num_days` может быть отрицательным, чтобы возвращать дату, ранее чем `start_date`.

Примеры

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

`RelativeDate([Дата резервирования]; -3)` возвращает 9 января 2007 года, если [Дата резервирования] – 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.

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	Обязателен
Тип даты	Объект даты для отчета, например, <code>InvoiceDate</code> .	Дата	Да
Тип периода	<p>Период для результатов. Может иметь следующие значения:</p> <ul style="list-style-type: none"> • <code>DayPeriod</code> • <code>MonthPeriod</code> • <code>QuarterPeriod</code> • <code>YearPeriod</code> <p>Если значение не выбрано, по умолчанию используется значение <code>DayPeriod</code>. 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
1/3/00	31607
1/8/00	31244
7/3/00	38154

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

Дата счета-фактуры	Доход
1/3/00	31607
1/8/00	31244
4/1/00	
7/3/00	38154

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

Дата счета-фактуры	Доход
K1	62851
K2	
K3	38154

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)
```

Ввод

Параметр	Описание	Тип	Обязательный
date_string	Дата для форматирования	строка	Да
формат	Формат даты.	Строка	Да

Примеры

`ToDate ("15/12/2002"; "dd/мм/yyyy")` возвращает 15/12/2002.

`ToDate ("15/12/02"; "dd/мм/yy")` возвращает 15/12/02.

`ToDate ("15/12/02"; "dd/MMMM/yy")` возвращает 15/DECEMBER/02.

См. также

- [Пользовательские форматы](#)

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

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 поставщика 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

Источник 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".

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

Источник 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

Источник 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.

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.

См. также

- [Функция DataProvider](#)

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.

См. также

- [Функция DataProvider](#)

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.

См. также

- [Функция DataProvider](#)

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 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.

См. также

- [Функция DataProvider](#)

6.1.5.16 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 "КЗ", если в документе применен фильтр детализации, допускающий в объекте [Квартал] только значение "КЗ"

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	Обязательное
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)`

Ввод

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.

`If(IsDate([Дата бронирования]) Then "Дата" Else "Не дата"` возвращает "Дата", если [Дата бронирования] – это дата.

См. также

- [Функция If...Then...Else](#)

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`).

См. также

- [Функция If...Then...Else](#)

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".

См. также

- [Функция If...Then...Else](#)

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.

См. также

- [Функция If...Then...Else](#)

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.

См. также

- [Функция If...Then...Else](#)

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.

См. также

- [Функция If...Then...Else](#)

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.

См. также

- [Функция If...Then...Else](#)

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`.

См. также

- [Функция If...Then...Else](#)

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

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.

См. также

- [Округление и усечение чисел](#)

Некоторые функции содержат параметр, определяющий, до какого знака функция округляет или обрезает возвращаемое значение.

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

См. также

- [Округление и усечение чисел](#)

Некоторые функции содержат параметр, определяющий, до какого знака функция округляет или обрезает возвращаемое значение.

6.1.8.6 Функция EuroFromRoundError

Описание

Возвращает ошибку округления при преобразовании из евро

Группа функций

Числовые

Синтаксис

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

Данные ввода

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

Данные вывода

Ошибка округления при вычислениях

Примеры

`EuroFromRoundErr(1000;"FRF";2)` возвращает 0. (Нет разницы между неокругленным преобразованием и преобразованием, округленным до 2 десятичных разрядов.)

`EuroFromRoundErr(1000;"FRF";1)` возвращает 0,03. (Неокругленное преобразование – 6559,57. Преобразование, округленное до 1 десятичного разряда, равно 6559,60. Ошибка округления составляет 0,03.)

`EuroFromRoundErr(1000;"DEM";2)` возвращает 0. (Нет разницы между неокругленным преобразованием и преобразованием, округленным до 2 десятичных разрядов.)

`EuroFromRoundErr(1000;"DEM";1)` возвращает -0,01. (Неокругленное преобразование – 1955,83. Преобразование, округленное до 1 десятичного разряда, 1995,80. Ошибка округления составляет -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

См. также

- [Округление и усечение чисел](#)

Некоторые функции содержат параметр, определяющий, до какого знака функция округляет или обрезает возвращаемое значение.

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

`EuroToRoundErr(6559; "FRF"; 2)` возвращает 0. (Нет разницы между неокругленной конвертацией и конвертацией, округленной до 2 десятичных знаков.)

`EuroToRoundErr(6559; "FRF"; 1)` возвращает -0.01. (Неокругленная конвертация – 999,91. Конвертация, округленная до 1 десятичного знака – 999,90. Ошибка округления – -0,01).

`EuroToRoundErr(1955; "DEM"; 2)` возвращает 0. (Нет разницы между неокругленной конвертацией и конвертацией, округленной до 2 десятичных знаков.)

`EuroToRoundErr(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

См. также

- [Округление и усечение чисел](#)

Некоторые функции содержат параметр, определяющий, до какого знака функция округляет или обрезает возвращаемое значение.

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 Функция Interpolation**Описание**

Вычисляет пустые значения мер путем интерполяции

Группа функций

Числовые

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязателен
measure	Любая мера	Мера	Да
PointToPoint Linear	Метод интерполяции <ul style="list-style-type: none"> PointToPoint – двухточечная интерполяция Linear – 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 устанавливается по умолчанию)

Примечания

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

Примеры

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

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

См. также

- [Оператор Linear](#)

Оператор `Linear` используется с функцией `Interpolation`.

- [Оператор PointToPoint](#)

С помощью оператора `PointToPoint` функция `Interpolation` использует двухточечную интерполяцию для получения отсутствующих значений мер.

6.1.8.12 Ln**Описание**

Возвращает натуральный логарифм числа

Группа функций

Числовые

Синтаксис

```
num Ln(число)
```

Входные данные

Параметр	Описание	Тип	Обязательный
number	Любое число	Число	Да

Примеры

Функция $\text{Ln}(10)$ возвращает 2.

6.1.8.13 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.

6.1.8.14 Функция Log10**Описание**

Возвращает логарифм по основанию 10 для числа

Группа функций

Числовые

Синтаксис

```
num Log10 (number)
```

Данные ввода

Параметр	Описание	Тип	Обязательный
number	Любое число	Число	Да

Примеры

`Log10(100)` возвращает 2.

6.1.8.15 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

6.1.8.16 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.

6.1.8.17 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
-----	--------	---------	---

См. также

- [Операторы Bottom/Top](#)

Операторы Bottom/Top используются с функцией Rank.

6.1.8.18 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.

См. также

- [Округление и усечение чисел](#)

Некоторые функции содержат параметр, определяющий, до какого знака функция округляет или обрезает возвращаемое значение.

6.1.8.19 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.

6.1.8.20 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)` возвращает -0,116992.

6.1.8.21 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.

6.1.8.22 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.

6.1.8.23 Функция ToNumber**Описание**

Преобразует строку в число

Группа функций

Числовые

Синтаксис

```
num ToNumber(string)
```

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.

6.1.8.24 Функция Truncate**Описание**

Обрезает число

Группа функций

Числовые

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязательный
number	Округляемое число	Число	Да
truncate_level	Количество знаков после запятой, до которого обрезается число	Число	Да

Примечания**Например**

Truncate (3,423; 2) возвращает 3,42.

См. также

- [Округление и усечение чисел](#)

Некоторые функции содержат параметр, определяющий, до какого знака функция округляет или обрезает возвращаемое значение.

6.1.9 Функции множества

6.1.9.1 Функция Children

Описание

Возвращает дочерний элемент для элемента

Группа функций

Набор

Синтаксис

```
member_set member.Children
```

Данные ввода

Параметр	Описание	Тип	Обязателен
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` либо указывается явным образом, либо считается равным текущему элементу иерархии. Если элемент указывается явно, необходимо указать полный путь в иерархии. Если указывается иерархия, находящаяся вне контекста формулы, то формула возвращает 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]`.

См. также

- [Агрегирование](#)
- [Average](#)
- [Count](#)
- [Функция Max](#)
- [Функция Min](#)
- [Функция Sum](#)

6.1.9.2 Функция Depth (Глубина)**Описание**

Возвращает глубину элемента в иерархии

Группа функций

Набор

Синтаксис

```
int member.Depth
```


Данные ввода

Параметр	Описание	Тип	Обязателен
member	Любой элемент	member	Да

Примечания

- Глубина – это расстояние элемента от верхнего уровня иерархии.
- Самым верхним уровнем иерархии является уровень 0.
- `Member` либо указывается явным образом, либо считается равным текущему элементу иерархии. Если элемент указывается явно, необходимо указать полный путь в иерархии. Если указывается иерархия, находящаяся вне контекста формулы, то формула возвращает пустое значение.

Примеры

[География] . [США] . [Калифорния] . [Лос-Анджелес] .Depth возвращает 2.

6.1.9.3 Функция Descendants**Описание**

Возвращает элементы, являющиеся потомками элемента

Группа функций

Набор

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязателен
member	Любой элемент	member	Да
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 либо указывается явным образом, либо считается равным текущему элементу иерархии. Если элемент указывается явно, необходимо указать полный путь в иерархии. Если указывается иерархия, находящаяся вне контекста формулы, то формула возвращает 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.

Примеры

Все примеры основаны на следующих данных:

			США			
	Калифорния				Невада	
Лос-Анджелес	Сан-Диего	Сан-Франциско		Лас-Вегас	Рино	Карсон-Сити

Функция `Descendants([Географическое положение].[США].[Калифорния];[Географическое положение].[Город])` возвращает [Сан-Франциско], [Сан-Диего], [Лос-Анджелес].

Функция `Descendants([Географическое положение];1)` возвращает [Калифорния], [Невада], если текущий элемент – [США].

Функция `Descendants([Географическое положение].[США];2;Before)` возвращает [США], [Калифорния], [Невада].

Функция `Descendants([Географическое положение].[США];[Географическое положение].[Город],Self_Before)` возвращает [США], [Калифорния], [Невада], [Сан-Франциско], [Сан-Диего], [Лос-Анджелес], [Лас-Вегас], [Рено], [Карсон-Сити].

Функция `Descendants([Географическое положение];[Географическое положение].[Штат];After)` возвращает [Сан-Франциско], [Сан-Диего], [Лос-Анджелес], [Лас-Вегас], [Рено], [Карсон-Сити], если текущий элемент – [США].

`Descendants([Географическое положение];1;Self_After)` возвращает [США], [Калифорния], [Невада], [Сан-Франциско], [Сан-Диего], [Лос-Анджелес], [Лас-Вегас], [Рено], [Карсон-Сити], если текущий элемент – [США].

Функция `Descendants([Географическое положение].[США];[Географическое положение].[Штат];Before_After)` возвращает [США], [Сан-Франциско], [Сан-Диего], [Лос-Анджелес], [Лас-Вегас], [Рено], [Карсон-Сити].

Функция `Descendants([Географическое положение].[США];[Географическое положение].[Штат],Self_Before_After)` возвращает [США], [Калифорния], [Невада], [Сан-Франциско], [Сан-Диего], [Лос-Анджелес], [Лас-Вегас], [Рено], [Карсон-Сити].

Функция `Descendants ([Географическое положение] . [США] ; [Географическое положение] . [Штат] ; Leaves)` возвращает [Сан-Франциско], [Сан-Диего], [Лос-Анжелес], [Лас-Вегас], [Рено], [Карсон-Сити].

См. также

- [Агрегирование](#)
- [Average](#)
- [Count](#)
- [Функция Max](#)
- [Функция Min](#)
- [Функция Sum](#)

6.1.9.4 Функция IsLeaf

Описание

Определяет, является ли элемент элементом конечного узла

Группа функций

Разные

Синтаксис

`bool member.IsLeaf`

Данные ввода

Параметр	Описание	Тип	Обязателен
member	Любой элемент	member	Да

Примечания

- Элемент конечного узла – это элемент, не имеющий дочерних элементов.
- `Member` либо указывается явным образом, либо считается равным текущему элементу иерархии. Если элемент указывается явно, необходимо указать полный путь в иерархии. Если указывается иерархия, находящаяся вне контекста формулы, то формула возвращает пустое значение.

Примеры

`[География] . [США] . [Калифорния] . IsLeaf` возвращает значение `False`, если элемент [Калифорния] имеет дочерние элементы [Лос-Анджелес], [Сан-Диего], [Сан-Франциско].

[Географическое положение].IsLeaf возвращает значение True, если текущим элементом иерархии [Географическое положение] является [Невада] и [Невада] не имеет дочерних элементов.

6.1.9.5 Функция Key

Описание

Возвращает ключ элемента

Синтаксис

```
string member.Key
```

Группа функций

Набор

Данные ввода

Параметр	Описание	Тип	Обязателен
member	Любой элемент	member	Да

Примечания

- Ключ является внутренним идентификатором элемента.
- Member либо указывается явным образом, либо считается равным текущему элементу иерархии. Если элемент указывается явно, необходимо указать полный путь в иерархии. Если указывается иерархия, находящаяся вне контекста формулы, то формула возвращает пустое значение.

Пример

[География].[США].Key возвращает "XYZ", если элемент [США] имеет ключ "XYZ".

6.1.9.6 Функция Lag

Описание

Возвращает элемент на том же уровне, что и текущий элемент и заданное расстояние перед ним

Синтаксис

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

Группа функций

Набор

Данные ввода

Параметр	Описание	Тип	Обязателен
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 значение distance положительно, функция Lag возвращает элемент, находящийся на расстоянии distance перед member. Если значение distance отрицательно, функция Lag возвращает элемент, находящийся на расстоянии distance после member.
- Member либо указывается явным образом, либо считается равным текущему элементу иерархии. Если элемент указывается явно, необходимо указать полный путь в иерархии. Если указывается иерархия, находящаяся вне контекста формулы, то формула возвращает пустое значение.
- Функция Lag использует порядок элементов в иерархии и запросе для возвращения связанного элемента.

Примеры

[География] . [США] . [Калифорния] . [Сан-Франциско] . Lag (1) возвращает [Сан-Диего], если [Сан-Диего] является предыдущим элементом для [Сан-Франциско] на уровне "Город".

См. также

- [Агрегирование](#)
- [Average](#)
- [Count](#)
- [Функция Max](#)
- [Функция Min](#)
- [Функция Sum](#)

6.1.9.7 Функция Parent

Описание

Возвращает родительский элемент для данного элемента

Группа функций

Набор

Синтаксис

```
member member.Parent
```

Данные ввода

Параметр	Описание	Тип	Обязателен
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` либо указывается явным образом, либо считается равным текущему элементу иерархии. Если элемент указывается явно, необходимо указать полный путь в иерархии. Если указывается иерархия, находящаяся вне контекста формулы, то формула возвращает пустое значение.

Примеры

[География] . [США] . [Калифорния] . [Лос-Анджелес] .Parent возвращает [Калифорния].

См. также

- [Агрегирование](#)
- [Average](#)
- [Count](#)
- [Функция Max](#)
- [Функция Min](#)
- [Функция Sum](#)

6.1.9.8 Функции Siblings

Описание

Возвращает элемент и его элементы с общим родителем.

Группа функций

Набор

Синтаксис

```
member_set member.Siblings
```

Данные ввода

Параметр	Описание	Тип	Обязателен
member	Любой элемент	member	Да

Примечания

- 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` либо указывается явным образом, либо считается равным текущему элементу иерархии. Если элемент указывается явно, необходимо указать полный путь в иерархии. Если указывается иерархия, находящаяся вне контекста формулы, то формула возвращает пустое значение.
- Элементы с общим родителем – это элементы того же уровня и с тем же общим родителем, что и у элемента.

Примеры

`[География].[США].[Калифорния].Siblings` (Элементы с общим родителем) возвращает [Невада], [Аризона], если [Невада] и [Аризона] находятся на том же уровне, что и [Калифорния].

См. также

- [Агрегирование](#)
- [Average](#)
- [Count](#)
- [Функция Max](#)
- [Функция Min](#)
- [Функция Sum](#)

6.1.10 Дополнительные функции

6.1.10.1 Функция Ancestor

Описание

Возвращает элемент-предок для элемента

Группа функций

Разные

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязателен
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` либо указывается явным образом, либо считается равным текущему элементу иерархии. Если элемент указывается явно, необходимо указать полный путь в иерархии. Если указывается иерархия, находящаяся вне контекста формулы, то формула возвращает пустое значение.
- `distance` (расстояние) должно быть положительным значением.

Примеры

Все примеры основаны на иерархии [География] (Страна > Штат > Город). В следующей таблице показано подмножество данных иерархии.

			США			
	Калифорния				Невада	
Лос-Анджелес	Сан-Диего	Сан-Франциско		Лас-Вегас	Рино	Карсон-Сити

`Ancestor ([География] . [США] . [Калифорния] . [Лос-Анджелес] ; 1)` вернет [Калифорния].

`Ancestor ([География] ; [География] . [Штат])` вернет [Калифорния], если текущий элемент равен [Сан-Диего].

См. также

- [Агрегирование](#)
- [Average](#)
- [Count](#)
- [Функция Max](#)
- [Функция Min](#)
- [Функция Sum](#)

6.1.10.2 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".

6.1.10.3 Функция ColumnNumber

Описание

Возвращает порядковый номер столбца

Группа функций

Разные

Синтаксис

```
int ColumnNumber()
```

Примеры

ColumnNumber() возвращает 2, если данная формула помещена во второй столбец таблицы.

6.1.10.4 Функция CurrentUser

Описание

Возвращает параметры входа на стартовую панель ВІ для текущего пользователя

Группа функций

Разные

Синтаксис

```
строка CurrentUser()
```

Примеры

`CurrentUser()` возвращает "gkn", если текущее регистрационное имя пользователя – "gkn".

6.1.10.5 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	Обязателен
measure	Любая мера	Мера	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].

6.1.10.6 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)".

6.1.10.7 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)".

См. также

- [Функция `GetPreferredViewingLocale`](#)

6.1.10.8 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)".

6.1.10.9 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.

См. также

- [Функция `GetPreferredViewingLocale`](#)

6.1.10.10 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)".

См. также

- [Функция `GetLocalized`](#)
- [Функция `GetDominantPreferredViewingLocale`](#)

6.1.10.11 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	Требуемое
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.

См. также

- [Функция If](#)
- [Оператор And](#)

Оператор And связывает булевы значения.

- [Оператор Between](#)
- [Оператор InList](#)
- [Оператор Or](#)

Оператор Or связывает булевы значения.

- [Оператор Not](#)

6.1.10.12 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	Требуемое
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.

См. также

- [Функция If...Then...Else](#)

6.1.10.13 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.

6.1.10.14 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".

6.1.10.15 Функция NoFilter**Описание**

Игнорирует фильтры при вычислении значения

Группа функций

Разные

Синтаксис

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

Данные ввода

Параметр	Описание	Тип	Обязатель- ный
obj	Любой объект отчета	Объект отчета	Да
All Drill	<ul style="list-style-type: none"> Нет указанного ключевого слова – игнорировать отчет и блокировать фильтры All – игнорировать все фильтры Drill – 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].

6.1.10.16 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.

6.1.10.17 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.

6.1.10.18 Функция `Previous`

Описание

Возвращает предыдущее значение объекта

Группа функций

Разные

Синтаксис

```
input_type Previous(измерение|меря|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)
NotNull	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 `NotNull` функция возвращает первое ненулевое значение объекта, начиная с ячейки строк смещения, предшествующих текущей строке и отсчитанных в обратном направлении.
- С функцией `Previous` (Предыдущее) можно использовать контекстные операторы расширенного синтаксиса
- Оператор `SELF` позволяет обращаться к предыдущему значению ячейки, если ее содержимое не является объектом отчета.
- Измерения нужно всегда помещать в круглые скобки, даже если в списке измерений сброса присутствует только одно измерение.
- Измерения сброса в наборе должны разделяться двоеточиями.
- Функция `Previous` применяется после применения всех фильтров отчета, раздела и блокировки, а также после всех сортировок.
- Применение сортировок 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;NotNull)` 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...

См. также

- [Сравнение значений с помощью функции Previous](#)

Функция `Previous` возвращает предыдущее значение выражения для сравнения.

- [Оператор Self](#)

Оператор `Self` используется с функцией `Previous`.

6.1.10.19 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.

6.1.10.20 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([Revenue];([Year]);-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

Год	Квартал	Продавец	Доход	RelativeValue
2008	K2	Джонс	3400	2000
2008	K3	Уилсон	2000	1500
2008	K4	Харрис	1700	3000

См. также• [#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.

• [Сравнение значений с помощью функции `RelativeValue`](#)

Функция `RelativeValue` возвращает значения выражения для сравнения. Функция возвращает данные значения независимо от макета отчета.

6.1.10.21 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".

6.1.10.22 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.

6.1.10.23 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.

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 [Revenue]>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] Between (800000;900000) Then "Средний доход" возвращает значение "Средний доход", если [Доход] 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.

См. также

- [Функция If...Then...Else](#)
- [Оператор Where](#)

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".

См. также

- [Функция If...Then...Else](#)
- [Оператор Where](#)

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.

См. также

- [Count](#)
- [Операторы Distinct/All](#)

Операторы `Distinct/All` используются с функцией `Count`.

- [Функция NoFilter](#)
- [Операторы All/Drill](#)

Операторы `All/Drill` используются с функцией `NoFilter`.

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.

См. также

- [Функция Rank](#)

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%

См. также

- [Функция Percentage](#)

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)

См. также

- [Count](#)

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.

См. также

- [Average](#)
- [Count](#)
- [Функция RunningAverage](#)
- [Функция RunningCount](#)

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.

См. также

- [Функция `UserResponse`](#)
- [Функция `RefValueUserResponse`](#)

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.

См. также

- [Функция `Interpolation`](#)

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 `NotNull` функция `Previous` возвращает первое ненулевое значение объекта, обнаруженного при отсчете в обратном направлении, начиная со строк смещения ячеек перед текущей строкой.

См. также

- [Функция Previous](#)

6.2.4.10 Оператор `NotOnBreak`

Оператор `NotOnBreak` используется с функцией `Interpolation`.

Описание

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

См. также

- [Функция Interpolation](#)

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.

См. также

- [Функция Interpolation](#)

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

См. также

- [Функция Percentage](#)
- [Функция RunningAverage](#)
- [Функция RunningCount](#)
- [Функция RunningMax](#)
- [Функция RunningMin](#)
- [Функция RunningProduct](#)
- [Функция RunningSum](#)

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 + Previous(Self)` возвращает последовательность 5, 10, 15, 20, 25, 30...

`1 + 0.5 * Previous(Self)` возвращает последовательность 1, 1.5, 1.75, 1.88...

См. также

- [Функция Previous](#)

6.2.4.14 Оператор Where

Описание

Оператор `Where` ограничивает данные, используемые для вычисления меры.

Примеры

Формулой `Average ([Доход с продаж]) Where ([Страна] = "США")` вычисляется средний доход с продаж для страны "США".

Формулой `Average ([Доход с продаж]) Where ([Страна] = "США" Or [Страна] = "Франция")` вычисляется средний доход с продаж для страны "США" или "Франция".

Формулой `[Доход] Where (Not ([Страна] InList ("США"; "Франция")))` вычисляется доход для стран, отличных от "США" и "Франция".

Переменная [Высокий доход] описана формулой `[Доход] Where [Доход > 500000]`. При размещении в блок переменной [Высокий доход] отображаются только значения дохода, превышающие 500000. При помещении в нижний колонтитул в нижней части столбца [Высокий доход], формула `Average ([Высокий доход])` возвращает среднее значение всех доходов, превышающих 500000.

См. также

- [Оператор And](#)

Оператор `And` связывает булевы значения.

- [Оператор Between](#)

- [Оператор InList](#)

- [Оператор Or](#)

Оператор `Or` связывает булевы значения.

- [Оператор Not](#)

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 с 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	Доходы с 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	Доходы с 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	Доходы с 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 ([Sales Revenue] In ([Year];[Quarter])) In ([Year])
```

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

Примечание:

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

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	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

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

```
Max ([Sales Revenue] In ([Year];[Quarter])) In ([Year])
```

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

```
Max ([Sales Revenue] ForEach ([Quarter])) In ([Year])
```

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 итоговый 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([Sales Revenue] ForAll ([Quarter]))
```

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

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

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)	Не применимо

Если расположено в...	Ссылается на эти данные...
За пределами всех блоков или разделов	Не применимо

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" присутствует формула.

```
Average([Sales revenue] In Section)
```

Формула также присутствует в столбце "В среднем за первое полугодие".

```
Average ([Sales revenue]) 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	Доходы с 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 ([Sales Revenue]) 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	Не применимо

Пример: Ключевое слово Break

В отчете отображаются год, квартал и доход от продаж:

Год	Квартал	Доходы с 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 ([Sales Revenue]) 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

Если расположено в...	Ссылается на эти данные...
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([Sales revenue]) 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

Если расположено в...	Ссылается на эти данные...
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 ([Sales Revenue]) 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 целое число, которое должно быть больше нуля, меньше нуля или равно нулю. В следующей таблице описываются способы округления и усечения чисел в каждом из этих случаев:

Параметр	Описание
> 0	<p>Эта функция округляет или обрезает до десятичных знаков <параметра>.</p> <p>Примеры.</p> <p>Round(3,13; 1) возвращает 3,1</p> <p>Round(3,157; 2) возвращает 3,16</p>
0	<p>Эта функция округляет или обрезает до ближайшего целого числа</p> <p>Примеры.</p> <p>Truncate(3,7; 0) возвращает 3</p> <p>Truncate(4,164; 0) возвращает 4</p>
< 0	<p>Эта функция округляет или обрезает до ближайшего числа 10 (параметр = -1), 100 (параметр = -2), 1000 (параметр = -3) и так далее.</p> <p>Примеры.</p> <p>Round(123,76; -1) возвращает 120</p> <p>Round(459,9; -2) возвращает 500</p> <p>Truncate(1600; -3) возвращает 1000</p>

Примечание:

Числа представляются в виде двойных значений и имеют точность до шестнадцатой цифры.

См. также

- [Функция Round](#)
- [Функция Truncate](#)
- [EuroConvertTo](#)
- [Функция EuroConvertFrom](#)
- [Функция EuroFromRoundError](#)
- [Функция EuroToRoundError](#)

6.5 Ссылки на элементы и наборы элементов в иерархиях

Ссылки на элементы и наборы элементов в функциях оформляются с помощью синтаксиса [иерархия] &path.function.

Элементы `path` и `function` не являются обязательными. `Path` содержит ссылки на элементы в квадратных скобках, при этом они разделяются точкой. Имена элементов и уровней указываются с учетом регистра.

Примечание:

Наборы элементов используются для переопределения контекста вычисления по умолчанию для иерархии. В функциях с наборами элементов последние заключаются в `{}`.

Для обозначения диапазонов элементов между начальным и конечным элементами указывается двоеточие (`:`) и для каждого элемента указывается полный путь. Диапазон содержит все элементы того же уровня, что и указанные элементы.

Пример синтаксиса для диапазона: `[Иерархия продаж]&[Customer_Type].[ENTERPRISE].[Крупный].[Нэнси Дэволио]: [Иерархия продаж]&[Customer_Type].[ENTERPRISE].[Крупный].[Эндрю Смит]`.

Пример: Ссылки на элементы и наборы элементов

Имеется следующая иерархия:

Иерархия продаж	Сумма заказа
Customer_Type	277 290 434
ENTERPRISE	180 063 361
Крупный	113 905 997
Нэнси Дэволио	44 855 689
Джэнет Леверлинг	44 050 308
Эндрю Смит	30 000 000
GLOBAL	91 157 363

- `[Иерархия продаж]&[Customer_Type].[ENTERPRISE].[Крупный].Children` ссылается на элементы `[Нэнси Дэволио]`, `[Джэнет Леверлинг]` и `[Эндрю Смит]`.
- `Sum([Сумма заказа];{[Иерархия продаж]&[Customer_Type].[ENTERPRISE].[Крупный].children})` возвращает значение **113 905 997** (сумма мер трех дочерних элементов).
- `[Иерархия продаж]&[Customer_Type].[ENTERPRISE].[Крупный].[Джэнет Леверлинг]` ссылается на элемент `[Джэнет Леверлинг]`.
- `Sum([Сумма заказа];{[Иерархия продаж]&[Customer_Type].[ENTERPRISE].[Крупный].[Джэнет Леверлинг];[Иерархия продаж]&[Customer_Type].[ENTERPRISE].[Крупный].[Нэнси Дэволио]})` возвращает результат **88 905 997** (сумма мер двух элементов).
- `[Иерархия продаж]&[Customer_Type].[ENTERPRISE].[Крупный].[Нэнси Дэволио]:[Иерархия продаж]&[Customer_Type].[ENTERPRISE].[Крупный].[Эндрю Смит]` ссылается на элементы `[Нэнси Дэволио]`, `[Джэнет 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 мер для трех элементов в диапазоне).**
 - `[Иерархия продаж].children` **ссылается на все элементы в иерархии [Иерархия продаж].**
 - `Sum([Сумма заказа];{[Иерархия продаж].children})` **возвращает значение 277 290 434.**
-

Устранение связанных с формулами неполадок

7.1 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.

7.1.1 #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.

См. также

- [Функция `RelativeValue`](#)

7.1.2 #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).

7.1.3 #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.

7.1.4 #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.

7.1.5 #ERROR

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

7.1.6 #EXTERNAL

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

7.1.7 #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.

7.1.8 #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.

7.1.9 #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

7.1.10 #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.

7.1.11 #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).

7.1.12 #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.

7.1.13 #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.

7.1.14 #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.

7.1.15 #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.

7.1.16 #REPFORMULA

Не удалось найти эквивалент Web Intelligence.

7.1.17 #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.

7.1.18 #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.

7.1.19 #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.

Для некоторых мер выполняется "делегирование" (для BW это относится, главным образом, к мере, не агрегированной с SUM): при определении таблицы или вычисления на основе меры, выполняется запрос этой меры в конкретном контексте агрегирования (мера дается 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.

7.1.20 #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.

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.

См. также

- [Функция Previous](#)
- [Функция RelativeValue](#)
- [Сравнение значений с помощью функции RelativeValue](#)

Функция `RelativeValue` возвращает значения выражения для сравнения. Функция возвращает данные значения независимо от макета отчета.

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([Revenue];([Year]);-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.

См. также

- [Функция RelativeValue](#)

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([Revenue];([Year];[Quarter]);-1)
```

- Выражение – [Доход];
- Измерения срезов – ([Год];[Квартал]);
- Смещение равно минус единице (функция сразу возвращает предыдущее значение в списке).

Год	Квартал	Продавец	Доход	RelativeValue
2007	K1	Смит	1000	
2007	K2	Смит	2000	
2007	K3	Смит	1500	
2007	K4	Смит	3000*	
2007	K1	Джонс	4000	
2007	K2	Джонс	3400	
2007	K3	Джонс	2000	
2007	K4	Джонс	1700	
2008	K1	Смит	5000**	3000*
2008	K2	Смит	3000***	5000**
2008	K3	Смит	2700****	3000***
2008	K4	Смит	6800	2700****

Формула, выраженная в деловом вопросе, возвращает доход, обеспеченный тем же продавцом в предыдущем квартале.

Вычисление, выполняемое в формуле, возвращает значение [Дохода] из строки, в которой значения [Год] и [Квартал] представляют предыдущее значение из списка значений ([Год];[Квартал]), а значение [Продавец] совпадает со значением в текущей строке.

Функция использует для поиска сравниваемого дохода список значений измерений срезов.

Год	Квартал	
2007	K1	
2007	K2	
2007	K3	
2007	K4	*
2008	K1	**
2008	K2	***
2008	K3	****
2008	K4	

Порядок сортировки в измерениях срезов определяет выходные данные функции. Знак "*" в таблицах отображает порядок сортировки.

См. также

- [Функция RelativeValue](#)

8.2.2 Измерения срезов и разделы

Измерение среза может находиться в основной ячейке раздела отчета.

Пример:

В указанной ниже таблице столбец RelativeValue содержит следующую формулу:

```
RelativeValue([Revenue];([Year];[Quarter]);-1)
```

2007			
Квартал	Продавец	Доход	RelativeValue
K1	Смит	1000	
K2	Смит	2000	
K3	Смит	1500	
K4	Смит	3000*	

Квартал	Продавец	Доход	RelativeValue
K1	Джонс	4000	
K2	Джонс	3400	
K3	Джонс	2000	
K4	Джонс	1700	

2008

Квартал	Продавец	Доход	RelativeValue
K1	Смит	5000**	3000*
K2	Смит	3000***	5000**
K3	Смит	2700****	3000***
K4	Смит	6800	2700****

Функция использует для поиска сравниваемого дохода список значений измерений срезов.

Год	Квартал	
2007	K1	
2007	K2	
2007	K3	
2007	K4	*
2008	K1	**
2008	K2	***
2008	K3	****
2008	K4	

Порядок сортировки в измерениях срезов определяет выходные данные функции. Знак "*" в таблицах отображает порядок сортировки.

См. также

- [Функция RelativeValue](#)

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([Revenue];([Year];[Quarter]);-1)
```

Год	Квартал	Продавец	Доход	RelativeValue
2007	K1	Смит	1000	
2007	K2	Смит	2000	
2007	K3	Смит	1500	
2007	K4	Смит	3000*	
2007	K1	Джонс	4000	
2007	K2	Джонс	3400	
2007	K3	Джонс	2000	
2007	K4	Джонс	1700	
2008	K1	Смит	5000**	3000*
2008	K2	Смит	3000***	5000**
2008	K3	Смит	2700****	3000***
2008	K4	Смит	6800	2700****

Формула, выраженная в деловом вопросе, возвращает доход, обеспеченный тем же продавцом в предыдущем квартале.

Порядок сортировки измерений срезов выглядит следующим образом.

Год	Квартал	
2007	K1	
2007	K2	
2007	K3	
2007	K4	*

Год	Квартал	
2008	K1	**
2008	K2	***
2008	K3	****
2008	K4	

Название функции изменяется на:

```
RelativeValue([Revenue];([Quarter];[Year]);-1)
```

Порядок сортировки измерений срезов становится следующим.

Квартал	Год	
K1	2007	*
K1	2008	**
K2	2007	***
K2	2008	****
K3	2007	*****
K3	2008	*****
K4	2007	*****
K4	2008	*****

Порядок сортировки оказывает следующее влияние на результат функции.

Год	Квартал	Продавец	Доход	RelativeValue
2007	K1	Смит	1000*	
2007	K2	Смит	2000***	
2007	K3	Смит	1500*****	
2007	K4	Смит	3000*****	
2007	K1	Джонс	4000	
2007	K2	Джонс	3400	
2007	K3	Джонс	2000	
2007	K4	Джонс	1700	
2008	K1	Смит	5000**	1000*

Год	Квартал	Продавец	Доход	RelativeValue
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

См. также

- [Функция RelativeValue](#)

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([Revenue];([Year];[Quarter]);-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	

Год	Квартал	Продавец	Доход	RelativeValue
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.

См. также

- [Функция RelativeValue](#)

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.

См. также

- [Функция RelativeValue](#)

Дополнительная информация

Источник информации	Местоположение
Информация о продуктах SAP BusinessObjects	http://www.sap.com
Справочный портал SAP	<p>Перейдите к http://help.sap.com/businessobjects/ и на боковой панели «BusinessObjects Overview» выберите All Products.</p> <p>Справочный портал SAP предоставляет доступ к актуальной документации по всем продуктам SAP BusinessObjects и их развертыванию. Можно загрузить документы в формате PDF или устанавливаемые HTML-библиотеки.</p> <p>Некоторые руководства находятся на веб-сайте SAP Service Marketplace и недоступны на справочном портале SAP. На данном портале перечислены эти руководства и даны соответствующие ссылки на SAP Service Marketplace. Клиенты, заключившие соглашение о техническом обслуживании, получают идентификатор авторизованного пользователя для доступа к этому веб-сайту. Для получения идентификатора обратитесь к представителю службы поддержки пользователей.</p>
SAP Service Marketplace	<p>http://service.sap.com/bosap-support > Документация</p> <ul style="list-style-type: none"> • Руководства по установке: https://service.sap.com/bosap-instguides • Примечания к выпуску: http://service.sap.com/releasenotes <p>На веб-сайте SAP Service Marketplace содержатся некоторые руководства по установке, модернизации, миграции и развертыванию, а также примечания к версиям и документация по поддерживаемым платформам. Клиенты, заключившие соглашение о техническом обслуживании, получают идентификатор авторизованного пользователя для доступа к этому веб-сайту. Для получения идентификатора обратитесь к представителю службы поддержки пользователей. Если вас перенаправили на веб-сайт SAP Service Marketplace со справочного портала SAP, с помощью меню в навигационной панели слева перейдите в категорию, содержащую нужные вам документы.</p>

Источник информации	Местоположение
Docupedia	https://cw.sdn.sap.com/cw/community/docupedia Docupedia предоставляет дополнительные ресурсы документации, объединенную среду для создания контента и интерактивный канал обратной связи.
Ресурсы разработчика	https://bos.sdn.sap.com/ https://www.sdn.sap.com/irj/sdn/businessobjects-sdklibrary
Статьи SAP BusinessObjects в сети сообщества SAP	https://www.sdn.sap.com/irj/boc/businessobjects-articles Подобные статьи ранее назывались технической документацией.
Примечания	https://service.sap.com/notes Эти примечания ранее назывались статьями базы знаний.
Форумы в сети сообщества SAP	https://www.sdn.sap.com/irj/scn/forums
Обучение	http://www.sap.com/services/education Мы можем предложить обучающий пакет, соответствующий вашим потребностям и предпочтительным формам обучения – от классических занятий в классах до специализированных курсов eLearning.
Интерактивная служба поддержки пользователей	http://service.sap.com/bosap-support На портале службы поддержки SAP представлены сведения о программах и услугах поддержки. Здесь также содержатся ссылки на самую разнообразную техническую информацию и множество файлов для загрузки. Клиенты, заключившие соглашение о техническом обслуживании, получают идентификатор авторизованного пользователя для доступа к этому веб-сайту. Для получения идентификатора обратитесь к представителю службы поддержки пользователей.

Источник информации	Местоположение
Консалтинговые услуги	http://www.sap.com/services/bysubject/businessobjectsconsulting Наши консультанты готовы оказывать вам поддержку на всех этапах – от начального анализа до развертывания системы. Советы экспертов можно найти, например, в темах, посвященных относительным и многомерным базам данных, возможностям подключения, средствам разработки баз данных и технологии встраивания с индивидуальными настройками.

Указатель

Символы

#ERROR, сообщение об ошибке 144, 145, 158, 215
#REPERFORMULA, сообщение об ошибке 218
#UNAVAILABLE
и интеллектуальные меры 52

A

All, оператор 178
Asc, функция 86
Average, функция 61

B

Between, оператор 190
BlockName, функция 170
Bottom, оператор 153

C

Ceil, функция 140
Char, функция 87
ColumnNumber, функция 170
Concatenation, функция 88
Connection, функция 113
Cos, функция 140
Count, функция 62
CurrentDate, функция 102
CurrentTime, функция 102
CurrentUser, функция 170

D

DataProvider, функция 114
DataProviderKeyDate, функция 114
DataProviderKeyDateCaption, функция 115
DataProviderSQL, функция 116
DataProviderType, функция 116
DayName, функция 102
DayNumberOfMonth, функция 103
DayNumberOfWeek, функция 103
DayNumberOfYear, функция 104
DaysBetween, функция 105
DocumentDate, функция 126
DocumentName, функция 126

DocumentPartiallyRefreshed, функция 127
DocumentTime, функция 127

E

EuroConvertFrom, функция 141
EuroConvertTo, функция 142
EuroFromRoundError, функция 144
EuroToRoundError, функция 145
Exp, функция 147

F

Fact, функция 147
Fill, функция 88
First, функция 64
Floor, функция 148
ForceMerge, функция 171
FormatDate, функция 89
FormatNumber, функция 90

G

GetContentLocale, функция 171
GetLocale, функция 173

H

HTMLEncode, функция 90

I

If, функция 176
If...Then...Else, условие 175
InitCap, функция 91
InList, оператор 191
Interpolation, функция 148, 195
IsDate, функция 132
IsError, функция 133
IsLogical, функция 134
IsNull, функция 135
IsNumber, функция 136
IsPromptAnswered, функция 117
IsString, функция 136
IsTime, функция 137

L

Last, функция 65
LastDayOfMonth, функция 105
LastDayOfWeek, функция 106
LastExecutionDate, функция 118
LastExecutionDuration, функция 118
LastExecutionTime, функция 119
Left, функция 91
LeftPad, функция 92
LeftTrim, функция 93
Length, функция 93
LineNumber, функция 177
Ln, функция 150
Log, функция 151
Log10, функция 151
Lower, функция 94

M

Match, функция 94
Max, функция 65
measures
возвращение последующих значений 184
возвращение предыдущих значений 184
Median, функция 67
Min, функция 67
Mod, функция 152
Month 106
MonthNumberOfYear, функция 107
MonthsBetween, функция 107

N

NameOf, функция 178
NoFilter, функция 178
Not, оператор 190
NOT, оператор 20, 189
NotOnBreak, оператор 148, 196
NumberOfDataProviders, функция 120
NumberOfPages, функция 179
NumberOfRows, функция 120

O

OR, оператор 20, 189

P

Page, функция 180
 Percentage, функция 69
 Percentile, функция 71
 Pos, функция 95
 Power, функция 152
 Previous, функция 180, 195, 198
 PromptSummary, функция 129

Q

Quarter, функция 108
 QuerySummary, функция 129

R

Rank, функция 153
 RefValue, функция 184
 RefValueDate, функция 121
 RefValueUserResponse, функция 121
 RelativeDate, функция 109
 Replace, функция 96
 ReportFilter, функция 130
 ReportFilterSummary, функция 130
 ReportName, функция 186
 Right, функция 97
 RightPad, функция 97
 RightTrim, функция 98
 Round, функция 155
 RowIndex, функция 186
 RunningAverage, функция 72
 RunningCount, функция 74
 RunningMax, функция 76
 RunningProduct, функция 78
 RunningSum, функция 80

S

Sign, функция 156
 Sin, функция 156
 SQL 116
 и наборы группировок 45, 46
 и обновление данных 46
 оператор GROUPING SETS 47
 оператор UNION 45, 47
 Sqrt, функция 157
 StdDev, функция 82
 StdDevP, функция 82
 Substr, функция 99

T

ToDate, функция 111
 ToNumber, функция 158

Trim, функция 99
 Truncate, функция 158

U

UniqueNameOf, функция 187
 UniverseName, функция 122
 Upper, функция 100
 URL-адреса
 применение правил кодировки к 100
 URLEncode, функция 100
 UserResponse, функция 123, 195

V

Var, функция 85
 VarP, функция 85

W

Week, функция 112
 Where, оператор 199
 WordCap, функция 101

Y

Year, функция 113

Б

блоки
 отображение имен 170
 булевы выражения
 возвращение противоположного 190
 связывание с помощью оператора And 189
 булевы значения
 определение 134
 проверка 175, 176
 булевы форматы
 пользовательские 55

В

валюты
 преобразование между европейскими валютами 141, 142
 вертикальные таблицы
 контексты вычислений по умолчанию в 28

вложенные фильтры OR
 и интеллектуальные меры 54
 входной контекст
 определенный 24
 выходной контекст
 определенные 25
 вычисления
 пользовательские 11
 стандартные 11

Г

горизонтальные таблицы
 контексты вычислений по умолчанию в 28

Д

данные
 обновление 46
 даты
 вычисление относительных дат 109
 идентификация 132
 форматирование 89, 111
 даты и время
 форматирование 55
 денежные значения
 форматирование 55
 диаграммы
 отображение имен 170
 дисперсия 85
 дисперсия генеральной совокупности 85
 документы
 отображение автора 124
 отображение владельца 126
 отображение времени последнего сохранения 127
 отображение даты последнего сохранения 126
 отображение даты создания 125
 отображение имени 126
 просмотр запросов в 129

Е

евро
 преобразование в 142
 преобразование из 141
 европейские валюты
 преобразование между 141, 142

3

завершающие пробелы
удаление из строк символов 98
зависящие от функций операторы
21, 191
запросы
просмотр сводки 129
значения
округление 209
подсчет 62, 74
сравнение с помощью
RelativeValue 221, 223, 224,
226, 228, 229
сравнение с помощью функции
Previous 221
точность 209
усечение 209
форматирование булева
значения 55
форматирование валюты 55
форматирование даты и времени
55
форматирование числа 55
значения времени
определение 137
значения меры
интерполяция 148, 196

И

измерения
включение нескольких
измерений в функции
агрегирования 88
добавление к контексту
вычисления 35, 202
и наборы группировок 49
и сообщение об ошибке
#DATASYNC 214
и сообщение об ошибке
#INCOMPATIBLE 215
измерения среза 221, 223
срезы 224
удаление из контекста
вычисления 35, 202
указание в контексте вычислений
33, 200
измерения среза
влияние порядка сортировки 228
интеллектуальные меры
влияние фильтров на 51
и вложенные фильтры OR 54
и наборы группировок 45
и область анализа 47
и операторы контекста 50

интеллектуальные меры
(*(продолжение)*)
и ошибка #UNAVAILABLE 52
и переменные 49
и сообщение об ошибке
#PARTIALRESULT 217
и сообщение об ошибке
#TOREFRESH 219
и сообщение об ошибке
#UNAVAILABLE 220
и стандартные вычисления 11
и фильтры детализации 53
и формулы 49, 50
невозможность вычисления 220
определенные 45
фильтрация 52
интерполяция значений меры 196

К

квадратный корень 157
ключевое слово "Разбиение" 39, 207
ключевое слово Body 41, 206
ключевое слово Report 42
ключевое слово блока 39, 204
ключевое слово отчета 37, 207
ключевое слово раздела 38, 208
ключевые слова
Block 39, 204
Body 41, 206
Break 39, 207
Report 37, 42, 207
Section 38, 208
ключевые слова расширенного
синтаксиса 36, 204
Отчет 42
создание отчетов, общих с 42
ключевые слова расширенного
синтаксиса 36, 204
использование вместе с
функцией Average 39, 204
использование вместе с
функцией Sum 37, 38, 39, 41,
206, 207, 208
создание отчетов, общих с 42
коды ASCII 86
возвращение символов,
связанных с 87
конечные пробелы
удаление из строк символов 99
контекстные операторы 20
контексты вычислений
входной контекст 23, 24
выходной контекст 23, 25
изменение с расширенным
синтаксисом 26

контексты вычислений
(*(продолжение)*)
определенные 23
по умолчанию 26
контексты вычислений по
умолчанию
в вертикальных таблицах 28
в горизонтальных таблицах 28
в кросс-таблицах 29
в разбиениях 31
в разделах 30
изменение с расширенным
синтаксисом 32
контексты вычисления
и интеллектуальные меры 50
косинус 140
кросс-таблицы
и функция RelativeValue 229
контексты вычислений по
умолчанию в 29

Л

логарифмы 150, 151
логарифмы по основанию 10 151
логарифмы по основанию n 151
логические операторы 20, 189
логические функции
описание 55

М

максимальные значения
вычисление промежуточного
максимума 76
математические операторы 19, 188
медиана 67
меры
и контексты вычислений по
умолчанию 26
определение интеллектуальных
мер 45
минимальные значения
вычисление промежуточного
минимума 77
мода 69

Н

наборы группировок 45
и область анализа 47
и оператор UNION 47
определенные 45
управление для 46

наборы элементов
 ссылки в функциях 211
 натуральные логарифмы 150
 начальные пробелы
 удаление из строк символов 93, 99
 нечетные числа
 определение 131, 138
 номера страниц
 отображение в отчетах 180

О

область анализа
 и интеллектуальные меры 47
 и наборы группировок 47
 обновление данных 46
 объекты
 отображение имени 187
 просмотр фильтров по 130
 оператор All 192, 194
 оператор And 20, 189
 оператор Between 20, 189
 оператор Bottom 192
 оператор Break 193
 оператор Col 72, 76, 77, 78, 80, 197
 оператор Distinct 194
 оператор Drill 178, 192
 оператор ForAll 35, 42, 202
 оператор ForEach 35, 202
 оператор In 33, 200
 оператор IncludeEmpty 194
 оператор Index 123
 оператор InList 20, 189
 оператор Linear 148, 195
 оператор NoNull 180, 195
 оператор OR 189
 оператор PointToPoint 148, 196
 оператор Row 72, 76, 77, 78, 80, 197
 оператор Self 180, 198
 оператор Top 153, 192
 оператор UNION 47
 оператор диапазона 203
 операторы
 All 178, 192, 194
 And 20, 189
 Between 20, 189, 190
 Bottom 153, 192
 Break 193
 Col 72, 74, 76, 77, 78, 80, 180, 197
 Distinct 194
 Drill 178, 192
 ForAll 35, 42, 202
 ForEach 35, 202
 In 33, 200
 IncludeEmpty 74, 194

операторы ((продолжение))
 Index 123, 195
 InList 20, 189, 191
 Linear 148, 195
 NoNull 180, 195
 Not 20, 189, 190
 NotOnBreak 148, 196
 Or 20, 189
 PointToPoint 148, 196
 Row 72, 74, 76, 77, 78, 80, 180, 197
 Self 180, 198
 Where 199
 диапазон 203
 зависящие от функций 21, 191
 контекст 20
 логический 20, 189
 математические 19, 188
 набор 203
 оператор Index 195
 определенные 19, 187
 Сверху 153, 192
 условные 20, 188
 операторы наборов 203
 опорные данные
 возвращение даты 121
 основной предпочтительный
 языковой стандарт для
 просмотра
 отображение 172
 ось времени
 построение 110
 отдельные ячейки
 и сообщение об ошибке
 #MULTIVALUE 216
 отчеты
 отображение имен объектов в
 178
 отображение количества страниц
 в 179
 отображение номеров страниц в
 180
 просмотр имен 186
 ошибки
 определение 133
 ошибки BW
 и сообщение об ошибке #N/A 216
 ошибки округления 144, 145

П

переменные
 и интеллектуальные меры 49
 упрощение формул с 13, 18

подсказки
 отображение ответов на 14, 121, 123
 подсчет значений 74
 пользовательские вычисления 11
 использование формул для
 построения 12
 пользовательские форматы даты и
 времени 55
 пользовательские числовые
 форматы 55
 порядок сортировки
 и измерения среза 223, 226, 228
 поставщики данных 120
 отображение имени юниверса
 122
 отображение количества строк в
 120
 просмотр кода SQL,
 сгенерированного 116
 Предпочтительный языковой
 стандарт для просмотра
 отображение 174
 пример
 наборы группировок 47
 произведения
 вычисление промежуточного
 произведения 78
 промежуточное количество 74
 процентиль 71
 пустые значения
 определение 135

Р

разбиения
 контексты вычислений по
 умолчанию в 31
 разделы
 и измерения среза 224
 и функция RelativeValue 224
 контексты вычислений по
 умолчанию в 30
 разные функции
 описание 55
 ранжирование
 и функции промежуточного
 агрегирования 217
 расширенный синтаксис 20
 изменение контекста вычислений
 по умолчанию с 32
 ключевое слово Block 39, 204
 ключевое слово Body 41, 206
 ключевое слово Break 39, 207
 ключевое слово Report 37, 42, 207

расширенный синтаксис
 ((продолжение))
 ключевое слово Section 38, 208
 оператор ForAll 32, 35, 199, 202
 оператор ForEach 32, 35, 199, 202
 оператор In 32, 33, 199, 200
 Редактор формул
 отображение синтаксиса
 функций в 14
 Руководство "Использование
 функций, формул и вычислений"
 сведения 9

C

свойства запроса
 MaxRowsRetrieved 217
 символы
 возвращение кодов ASCII 86
 отображение из кодов ASCII 87
 символьные функции
 описание 55
 синтаксис функций
 пример 14
 синус 156
 сообщение об ошибке
 #COMPUTATION 184, 213
 сообщение об ошибке #CONTEXT
 214
 сообщение об ошибке #DATASYNC
 214
 сообщение об ошибке #ERROR 141,
 142
 сообщение об ошибке #EXTERNAL
 215
 сообщение об ошибке
 #INCOMPATIBLE 214, 215
 Сообщение об ошибке #MIX 216
 сообщение об ошибке
 #MULTIVALUE 186, 216
 сообщение об ошибке #N/A 216
 сообщение об ошибке #OVERFLOW
 217
 сообщение об ошибке
 #PARTIALRESULT 217
 и интеллектуальные меры 217
 сообщение об ошибке #RANK 217
 сообщение об ошибке #RECURSIVE
 218
 сообщение об ошибке #REFRESH
 218
 сообщение об ошибке #SECURITY
 219
 сообщение об ошибке #SYNTAX 219

сообщение об ошибке #TOREFRESH
 46, 50, 219
 и интеллектуальные меры 219
 сообщение об ошибке
 #UNAVAILABLE 51, 220
 и интеллектуальные меры 220
 сообщение об ошибке #ДЕЛЕНИЕ/0
 214
 сообщения об ошибках
 #COMPUTATION 184, 213
 #CONTEXT 214
 #DATASYNC 214
 #DIV/0 214
 #ERROR 215
 #ERRPR 141, 142, 144, 145, 158
 #EXTERNAL 215
 #INCOMPATIBLE 214, 215
 #MIX 216
 #MULTIVALUE 186, 216
 #N/A 216
 #OVERFLOW 217
 #PARTIALRESULT 217
 #RANK 217
 #RECURSIVE 218
 #REFRESH 218
 #REPFORMLA 218
 #SECURITY 219
 #SYNTAX 219
 #TOREFRESH 46, 50, 219
 #UNAVAILABLE 51, 220
 сообщения об ошибках в формулах
 213
 сортировки
 и измерения среза 223, 226, 228
 среднее
 вычисление скользящего
 среднего 72
 среднеквадратическое отклонение
 82
 среднеквадратическое отклонение
 генеральной совокупности 82
 стандартное вычисление функции
 Average 11
 стандартное вычисление функции
 Count 11
 стандартное вычисление функции
 Default 11
 стандартное вычисление функции
 Maximum 11
 стандартное вычисление функции
 Minimum 11
 стандартное вычисление функции
 Percentage 11
 стандартное вычисление функции
 Sum 11
 стандартные вычисления 11

столбцы
 отображение номеров 170
 страницы
 отображение количества в
 отчетах 179
 строки
 определение 136
 отображение количества в
 поставщике данных 120
 отображение номера строки 186
 подсчет в таблицах 177
 подсчет всех значений 192
 подсчет различных значений 192
 строки символов
 возврат (первых) крайних левых
 символов 91
 возвращение самых правых
 символов 97
 вычисление длины 93
 замена частей 96
 заполнение другими строками
 92, 97
 извлечение разделов из 99
 объединение/конкатенация 19,
 88, 188
 перевод в верхний регистр 100
 перевод в верхний регистр всех
 первых букв в 101
 перевод в верхний регистр
 первых букв 101
 перевод в числа 158
 перевод первой буквы в верхний
 регистр 91
 повторение 88
 преобразование в нижний
 регистр 94
 применение правил кодировки
 URL к 100
 соответствие шаблону 94
 удаление конечных пробелов из
 98, 99
 удаление начальных пробелов
 из 93, 99

T

таблицы
 вертикальная 28
 горизонтальная 28
 отображение имен 170
 подсчет строк в 177
 тангенсы 157

У

условные операторы 20, 188

Ф

факториал 147

фильтры

блок 130

детализация 192

и интеллектуальные меры 53

на измерения 51

отображение всех значений 192

отображение фильтров

детализации 192

отчет 130, 192

раздел 130

фильтрация интеллектуальных мер 52

фильтры детализации 53

фильтры детализации 192

и интеллектуальные меры 53

игнорирование 192

отображение 192

фильтры для измерений

влияние на интеллектуальные меры 51

влияние на интеллектуальные меры 51

фильтры отчета 130, 192

игнорирование 192

просмотр сводки 130

форматы валют

пользовательские 55

форматы даты и времени

пользовательские 55

форматы чисел

пользовательские 55

формулы

и интеллектуальные меры 49

интеллектуальные меры в 50

использование операторов в 19, 187

создание пользовательских

вычислений с помощью 12

сообщения об ошибках,

сгенерированные 213

упрощение с переменными 13, 18

функции

Abs 139

Aggregate 60

Ancestor 168

Asc 86

Average 39, 61, 194, 204

BlockName 170

Ceil 140

Char 87

Children 159

ColumnNumber 170

функции ((продолжение))

Concatenation 88

Connection 113

Cos 140

Count 62, 192, 194

CurrentDate 102

CurrentTime 102

CurrentUser 170

DataProvider 114

DataProviderKeyDate 114

DataProviderKeyDateCaption 115

DataProviderSQL 116

DataProviderType 116

DayName 102

DayNumberOfMonth 103

DayNumberOfWeek 103

DayNumberOfYear 104

DaysBetween 105

Descendants 161

DocumentAuthor 124

DocumentCreationDate 125

DocumentCreationTime 125

DocumentCreationTime, функция 125

DocumentDate 126

DocumentName 126

DocumentOwner 126

DocumentPartiallyRefreshed 127

DocumentTime 127

DrillFilters 128

EuroConvertFrom 141, 209

EuroConvertTo 142, 209

EuroFromRoundError 144, 209

EuroToRoundError 145, 209

Even 131

Exp 147

Fact 147

Fill 88

First 64

Floor 148

ForceMerge 171

FormatDate 89

FormatNumber 90

GetContentLocale 171

GetDominantPreferredViewingLocale 172

GetLocale 173

GetLocalized 173

GetPreferredViewingLocale 174

HTMLEncode 90

If 20, 176, 188

InitCap 91

Interpolation 148, 195, 196

IsDate 132

IsError 133

IsLeaf 164

функции ((продолжение))

IsLogical 134

IsNull 135

IsNumber 136

IsPromptAnswered 117

IsString 136

IsTime 137

Lag 165

Last 65

LastDayOfMonth 105

LastDayOfWeek 106

LastExecutionDate 118

LastExecutionDuration 118

LastExecutionTime, функция 119

Left 91

LeftPad 92

LeftTrim 93

Length 93

LineNumber 177

Ln 150

Log 151

Log10 151

Lower 94

Match 94

Max 35, 65, 202

Median 67

Min 67

Mod 152

Mode 69

Month 106

MonthNumberOfYear 107

MonthsBetween 107

NameOf 178

NoFilter 178, 192

NumberOfDataProviders 120

NumberOfPages 179, 218

NumberOfRows 120

Odd 138

Page 180

Parent 166

Percentage 14, 69, 193, 197

Percentile 71

Pos 95

Power 152

Previous 180, 195, 197, 198, 217, 221

Product 72

PromptSummary 129

Quarter 108

QuerySummary 129

Rank 192

RefValue 184

RefValueDate 121

RefValueUserResponse 121, 195

RelativeDate 109

функции ((продолжение))

RelativeValue 184, 213, 221, 224, 226, 228, 229
 Replace 96
 ReportFilter 130
 ReportFilterSummary 130
 ReportName 186
 Right 97
 RightPad 97
 RightTrim 98
 Round 155, 209
 RowIndex 186
 RunningAverage 72, 194, 197
 RunningCount 74, 194, 197
 RunningMax 76, 197
 RunningMin 77, 197
 RunningProduct 78, 197
 RunningSum 80, 197
 RunningSum, функция 197
 ServerValue 81
 Siblings 167
 Sign 156
 Sin 156
 Sqrt 157
 StdDev 82
 StdDevP 82
 Substr 99
 Sum 14, 35, 37, 38, 39, 41, 42, 83, 202, 206, 207, 208
 Tan 157
 TimeDim 110
 ToDate 111
 ToNumber 158
 Trim 99
 Truncate 158, 209
 UniqueNameOf 187
 UniverseName 122
 Upper 100
 URLEncode 100
 UserResponse 14, 195
 Var 85
 VarP 85
 Week 112
 WordCap 101
 Year 113
 включение в ячейки 14
 глубина 160
 категории 55
 Ключ 165
 определенные 13
 ОтветПользователя 123
 примеры 14
 Ранжирование 153
 синтаксис функций 14
 смесь с текстом в ячейках 14

функции ((продолжение))

ссылки на наборы элементов в 211
 ссылки на элементы в 211
 функция DocumentCreationDate 125
 функция Interpolation 196
 функция Percentage 197
 функция RunningAverage 197
 функция RunningCount 197
 функция RunningMax 197
 функция RunningMin 77, 197
 функция RunningProduct 197
 функция Sum 83
 функции агрегирования 60
 включение нескольких измерений в 88
 описание 55
 функции даты и времени
 описание 55
 функции для поставщиков данных
 описание 55
 функции документов
 описание 55
 функции промежуточного агрегирования
 и ранжирование 217
 и сообщение об ошибке #RANK 217
 функция Abs 139
 функция Ancestor 168
 функция Average 194
 использование вместе с ключевыми словами расширенного синтаксиса 39, 204
 функция Children 159
 функция Count 192, 194
 Функция Depth 160
 Функция Descendants 161
 Функция DocumentAuthor 124
 Функция DocumentOwner 126
 функция DrillFilters 128
 функция EuroConvertFrom 209
 функция EuroConvertTo 209
 функция EuroFromRoundError 209
 функция EuroToRoundError 209
 функция Even 131
 функция
 GetDominantPreferredViewingLocale 172
 функция GetLocalized 173
 функция GetPreferredViewingLocale 174
 функция If 20, 188
 функция Interpolation 196

функция IsLeaf 164

Функция Key 165
 функция Lag 165
 функция Max
 использование с операторами контекста 35, 202
 функция Mode 69
 функция NoFilter 192
 функция NumberOfPages
 и сообщение об ошибке #RECURSIVE 218
 функция Odd 138
 функция Parent 166
 функция Percentage 14
 функция Previous
 и сообщение об ошибке #RANK 217
 сравнение значений с помощью 221
 функция Product 72
 функция Rank 192
 функция RefValueUserResponse 195
 функция RelativeValue 184
 и измерения среза 223, 224, 228
 и кросс-таблицы 229
 и разделы 224
 и сообщение об ошибке #COMPUTATION 213
 сравнение значений с помощью 221, 223, 224, 226, 228, 229
 функция Round 209
 функция RunningAverage 194
 функция RunningCount 194
 функция ServerValue 81
 функция Siblings 167
 функция Sum 14
 использование вместе с ключевыми словами расширенного синтаксиса 37, 38, 39, 41, 42, 206, 207, 208
 использование с операторами контекста 35, 202
 функция Tan 157
 функция TimeDim 110
 функция Truncate 209
 функция UserResponse 14

Ч

четные числа
 идентификация 131, 138
 числа
 возведение в степень 152
 вычисление модуля 152
 деление 152
 идентификация 136

числа ((продолжение))

- модуль 152
- округление 140, 155
- округление с недостатком 148
- округление чисел 155
- округление чисел с недостатком 148
- определение
 - четности/нечетности 131, 138
- перевод строк символов в 158
- умножение 72, 78
- усечение 158
- усечение чисел 158
- форматирование 55, 90
- числовые функции
 - описание 55

Э

- экспоненциальная функция 147
- элементы
 - ссылки в функциях 211

Ю

- юниверсы
 - отображение имен 122

Я

- Языковой стандарт документа
 - отображение 171

- Языковой стандарт продукта
 - отображение 173

- языковые стандарты
 - отображение основного
 - предпочтительного языкового
 - стандарта для просмотра 172
 - отображение языкового
 - стандарта документа 171
 - отображение языкового
 - стандарта продукта 173

ячейки

- включая текст в 14
- включение функций в ячейки 14
- форматирование дат в 55
- форматирование чисел в 55