

IBM DB2 Cube Views



Сценарии моделирования

Версия 8

Примечание

Перед тем как использовать эту информацию и продукт, описанный в ней, прочтите общие сведения под заголовком “Замечания” на стр. 31.

Этот документ содержит информацию, которая является собственностью IBM. Она предоставляется в соответствии с лицензионным соглашением и защищена законами об авторском праве. Информация в данной публикации не включает никаких гарантий на продукт и никакое из утверждений в данном руководстве не следует понимать подобным образом.

Заказать публикации IBM можно через Интернет или у местного представителя IBM.

- Чтобы заказать публикации через Интернет, перейдите на Web-страницу Центра публикаций IBM (IBM Publications Center): www.ibm.com/shop/publications/order
- Чтобы найти местное представительство IBM, перейдите на страницу IBM Directory of Worldwide Contacts по адресу www.ibm.com/planetwide

Чтобы заказать публикации DB2 через отдел DB2 Marketing and Sales в Соединенных Штатах или Канаде, позвоните по телефону 1800-IBM-4YOU (425-4968).

Отсылая информацию IBM, вы тем самым даете IBM неисключительное право использовать или распространять эту информацию любым способом, как фирма сочтет нужным, без каких-либо обязательств перед вами.

Содержание

Об этой книге	v	Подробности сценария	19
Для кого предназначена эта книга	v	Шаги создания показателя	20
Глава 1. Вычисление потоков и стоимости товаров на складе в течение времени	1	Глава 5. Ранжирование данных о продажах	23
Подробности сценария	1	Подробности сценария	23
Шаги создания показателей	4	Шаги создания показателя	25
Глава 2. Вычисление корреляции между затратами на рекламу и объемом продаж	7	Глава 6. Использование данных из таблицы фактов для создания измерения времени	27
Подробности сценария	7	Подробности сценария	27
Шаги для создания показателя	7	Шаги для создания атрибутов и измерения	29
Глава 3. Вычисление прибыли и уровня прибыли для магазина	13	Замечания	31
Подробности сценария	13	Товарные знаки	33
Шаги создания показателей	14		
Глава 4. Определение числа заказов через Интернет	19		

Об этой книге

Эта книга содержит дополнительную информацию к книге *Руководство по установке и использованию DB2 Cube Views*, и описывает моделирование реальных бизнес-сценариев с использованием метаданных DB2 Cube Views.

Для кого предназначена эта книга

Прочитайте эту книгу, если вы администратор базы данных, работающий с метаданными OLAP и DB2 Universal Database (DB2). Вы должны знать:

- Понятия OLAP, такие как схема типа "звезда"
- Объекты метаданных DB2 Cube Views, такие как модели кубов, фактографические объекты, изменения, объединения, показатели и атрибуты.

Глава 1. Вычисление потоков и стоимости товаров на складе в течение времени

Торговая компания XYZ Retail, прежде чем посылать товары в конкретный магазин для продажи, хранит их на складе. XYZ Retail сохраняет данные о нахождении товаров на складе в течение времени и теперь желает анализировать эти данные. В частности, в компании хотят исследовать две характеристики работы склада:

- Поток товаров, поступающих на склад и отправляемых с него
- Стоимость товаров на складе в данный момент времени

Для анализа первой характеристики (потока товаров) нужно использовать данные за определенный промежуток времени. Для получения второй характеристики (стоимости товаров) нужно получить информацию о состоянии склада в определенный момент времени.

Подробности сценария

В XYZ Retail есть таблица фактов со следующими столбцами, относящимися к складу: QUANTITY_IN (число поступивших), QUANTITY_OUT (число отправленных), CURRENT_QUANTITY (текущее число), PRODUCT_VALUE (стоимость продукта), PRODUCT_ID (ID продукта) и TIME_ID (ID времени). Эти данные вводятся в таблицу еженедельно. В базе данных есть также таблицы Product (продукт) и Time (время). Пример набора данных таблицы фактов показан в Табл. 1.

Таблица 1.

PRODUCT_ID	TIME_ID	QUANTITY_IN	QUANTITY_OUT	CURRENT_QUANTITY	PRODUCT_VALUE
1234	1	5	0	5	5
1234	2	20	10	15	5
1234	3	10	20	5	5

Во всех этих трех записях данных одно и то же значение PRODUCT_ID - продукты одного типа могут многократно поступать на склад и убывать со склада.

Администратор базы данных компании XYZ Retail должен создать три различных показателя:

Flow In (входящий поток)

Моделирует поток товаров, поступающих на склад.

Flow Out (выходящий поток)

Моделирует поток товаров, отправляемых со склада.

Current Value (текущая стоимость)

Моделирует стоимость товаров в данный момент времени.

Для создания первых двух показателей, Flow In и Flow Out, администратор базы данных создает показатели, которые отображаются соответственно в столбцы QUANTITY_IN и QUANTITY_OUT и суммируют данные по всем измерениям. Такие показатели называют полностью аддитивными, поскольку для агрегации данных используется только функция суммирования (SUM) по всем измерениям. В Табл. 2 показан пример набора данных для столбцов QUANTITY_IN и QUANTITY_OUT за три месяца для продукта с PRODUCT_ID 1234. Показатели Flow In и Flow Out суммируют эти ежемесячные значения и показывают общее число поступивших и отправленных товаров за квартал.

Таблица 2. Вычисление данных примера для полностью аддитивных показателей Flow In и Flow Out для PRODUCT_ID 1234

	Январь	Февраль	Март	1 квартал
QUANTITY_IN	5	20	10	35
QUANTITY_OUT	0	10	20	30

Полностью аддитивные показатели - это самый простой и общий тип показателей; они часто используются как элементы более сложных показателей. Для числовых данных OLAP Center по умолчанию создает полностью аддитивный показатель.

Для создания третьего показателя, Current Value, администратор базы данных создает показатель, вычисляющий произведение значений PRODUCT_VALUE и CURRENT_QUANTITY. Например, если стоимость продукта с PRODUCT_ID=1234 равна 5, значения показателя Current Value для данных примера показаны в Табл. 3.

Таблица 3. Вычисление данных примера для показателя Current Value для PRODUCT_ID 1234

	Январь	Февраль	Март
CURRENT_QUANTITY	5	10	20
Current Value	25	50	100

Затем эти данные должны суммироваться по измерениям. Но поскольку этот показатель отражает значение в определенный момент времени, для него не нужно выполнять суммирование по измерению Time (время). Вместо этого выполняется суммирование по измерению Product (продукт) и вычисляется

среднее значение данных по времени. Такой показатель называется полуаддитивным, так как только часть агрегации содержит функцию суммирования (SUM).

Показатели, в которых вычисляется состояние на определенный момент времени, например, данные месячного учета, часто являются полуаддитивными, так как в них не производится суммирование по месяцам квартала. Если продукт оставался на складе целый квартал, он будет включен в данные CURRENT_QUANTITY складского учета для каждого из трех месяцев квартала. Если просуммировать данные CURRENT_QUANTITY по времени, продукт, оставшийся на складе в течение трех месяцев, будет посчитан трижды. Как показано в Табл. 4, значение 25 для 1 квартала неверно представляет работу склада. В этой таблице видно, что на складе никогда не было 25 продуктов, поэтому число 25 продуктов не имеет смысла.

Таблица 4. Вычисление данных примера для столбца CURRENT_QUANTITY - использование функции SUM по измерению времени для PRODUCT_ID 1234

	Январь	Февраль	Март	1 квартал
SUM(CURRENT_QUANTITY)	5	15	5	25

Вместо того, чтобы применять функцию SUM для суммирования по всем измерениям, для измерения времени лучше использовать другие функции агрегации, такие как AVG, MIN и MAX. Для тех же данных примера для трех месяцев (январь, февраль, март) можно применить другую функцию агрегации для измерения времени, как показано в Табл. 5, чтобы получить осмысленные значения для квартала. Показатель Current Value может представлять среднее число товаров, хранимых на складе в течение квартала, или максимальное или минимальное число товаров в какой-либо момент времени в этом квартале.

Таблица 5. Вычисление данных примера для столбца CURRENT_QUANTITY - использование функций AVG, MAX и MIN для измерения времени для PRODUCT_ID 1234

	Январь	Февраль	Март	1 квартал
AVG(CURRENT_QUANTITY)	5	15	5	8.3
MAX(CURRENT_QUANTITY)	5	15	5	15
MIN(CURRENT_QUANTITY)	5	15	5	5

Шаги создания показателей

Следующие шаги объясняют, как можно использовать окно Свойства фактографического объекта OLAP Center для создания показателей Flow In, Flow Out и Current Value в существующем фактографическом объекте:

1. Чтобы открыть окно Свойства фактографического объекта, щелкните правой кнопкой мыши по фактографическому объекту в дереве объектов OLAP Center и выберите **Изменить показатели**. Откроется окно Свойства фактографического объекта.
2. Создайте показатель Flow In:
 - a. На странице Показатели нажмите кнопку **Создать вычисленный показатель**, чтобы создать показатель Flow In. Откроется окно Построитель выражений SQL.
 - b. В окне Построитель выражений SQL введите FLOW IN в поле **Имя**.
 - c. Чтобы создать выражение для этого показателя:
 - Раскройте папку **Столбцы** и таблицу фактов в списке **Данные**.
 - Дважды щелкните по столбцу **QUANTITY_IN**, чтобы добавить его в выражение.
 - Нажмите кнопку **ОК**, чтобы закрыть окно Построитель выражений SQL. Не изменяйте функцию агрегации по умолчанию, SUM, заданную на странице Агрегации. Функция SUM по умолчанию используется для показателя Flow In, так как у него числовой источник данных и показатель ссылается на столбец, а не только на другие показатели.
3. Создайте показатель Flow Out:
 - a. На странице Показатели нажмите кнопку **Создать вычисленный показатель**, чтобы создать показатель Flow Out. Откроется окно Построитель выражений SQL.
 - b. В окне Построитель выражений SQL введите FLOW OUT в поле **Имя**.
 - c. Чтобы создать выражение для этого показателя:
 - Раскройте папку **Столбцы** и таблицу фактов в списке **Данные**.
 - Дважды щелкните по столбцу **QUANTITY_OUT**.
 - d. Нажмите кнопку **ОК**, чтобы закрыть окно Построитель выражений SQL. Не изменяйте функцию агрегации по умолчанию, SUM, заданную на странице Агрегации. Функция SUM по умолчанию используется для показателя Flow Out, так как у него числовой источник данных и показатель ссылается на столбец, а не только на другие показатели.
4. Создайте показатель Current Value:
 - a. На странице Показатели нажмите кнопку **Создать вычисленный показатель**, чтобы создать показатель Current Value. Откроется окно Построитель выражений SQL.
 - b. В окне Построитель выражений SQL введите CURRENT VALUE в поле **Имя**.
 - c. Чтобы создать выражение для этого показателя:

- Раскройте папку **Столбцы** и таблицу фактов в списке **Данные**.
- Дважды щелкните по столбцу **PRODUCT_VALUE** в списке **Данные**.
- Дважды щелкните по операции ***** в списке **Операторы**.
- Дважды щелкните по столбцу **CURRENT_QUANTITY** в списке **Данные**.

На рис. 1 показано выражение показателя Current Value, которое можно создать в окне Построитель выражений SQL.

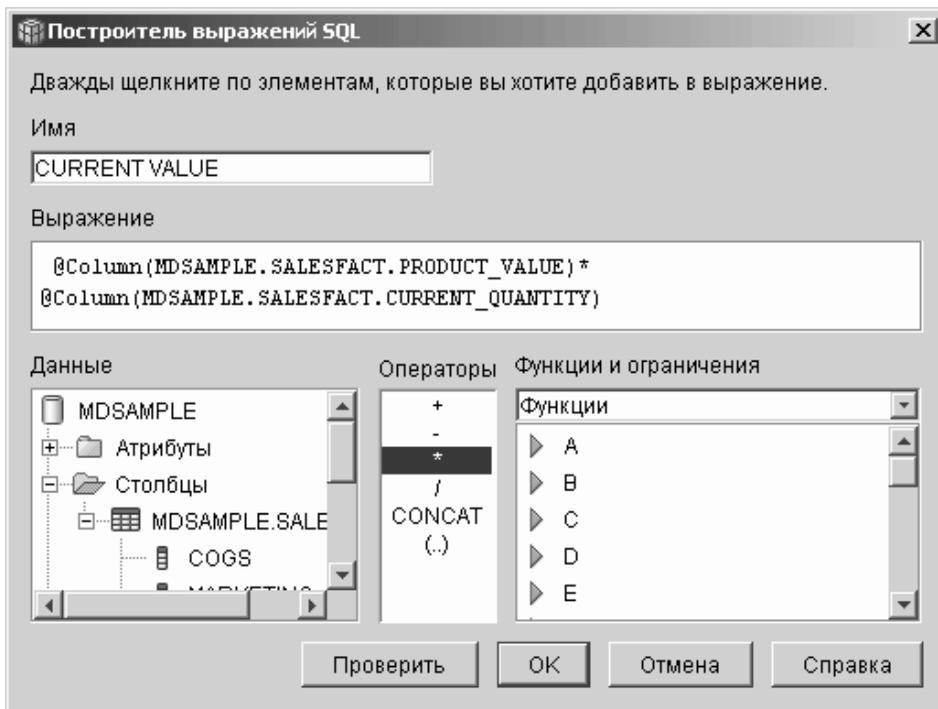


Рисунок 1. Полное выражение показателя Current Value в окне Построитель выражений SQL

- Нажмите кнопку **ОК**, чтобы закрыть окно Построитель выражений SQL.
- На странице Агрегации выберите агрегацию для показателя Current Value и выберите **Сценарий агрегации** из списка. Откроется окно Построитель сценариев агрегации. Сценарий агрегации по умолчанию содержит функцию SUM для всех измерений.
- При необходимости переместите измерение Time вниз, выбрав **Time** и нажав кнопку , чтобы это измерение стало последним в списке для этого сценария. Выбрав измерение Time, дважды щелкните по функции **AVG** в списке **Функции столбцов**. Сценарий агрегации, показанный на

рис. 2, будет суммировать данные по всем измерениям, кроме измерения Time; данные по этому измерению будут усредняться.

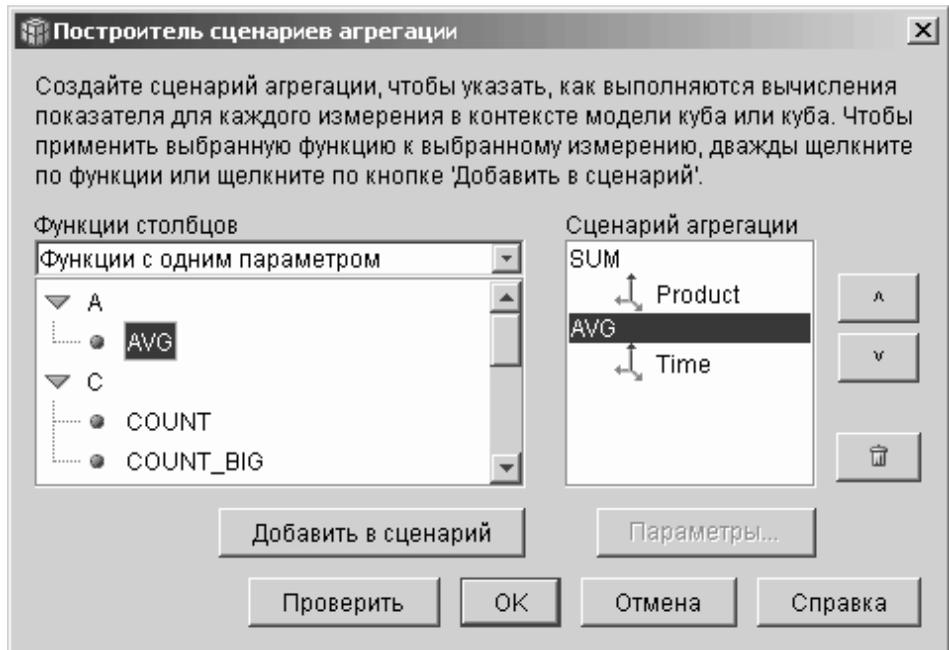


Рисунок 2. Сценарий агрегации для показателя Current Value

- g. В окне Построитель сценариев агрегации нажмите кнопку **Проверить**, чтобы проверить правильность сценария агрегации. Нажмите кнопку **ОК**, чтобы сохранить сценарий агрегации и закрыть это окно.
5. Нажмите кнопку **ОК**, чтобы сохранить изменения в фактографическом объекте и закрыть окно Свойства фактографического объекта.

Теперь у вас есть три вычисляемых показателя для товаров на складе. Эти показатели можно использовать для анализа поведения входящих и исходящих потоков на складе.

Глава 2. Вычисление корреляции между затратами на рекламу и объемом продаж

В агентстве по продаже автомобилей обсуждается повышение расходов на рекламу. Чтобы принять обоснованное решение, агентству нужно сначала проанализировать взаимосвязь между затратами на рекламу и объемом продаж. Нужно выяснить, как влияют на продажи различные типы рекламы, и в особенности, насколько повышение расходов на рекламу приводит к повышению объема продаж.

Подробности сценария

В базе данных этого агентства есть таблица фактов со столбцами Sales (объем продаж) и Ad Costs (расходы на рекламу). В базе данных есть также некоторые таблицы измерений. Администратор базы данных может создать показатели, использующие функцию DB2 CORRELATION, чтобы определить корреляцию между расходами и объемом продаж. Функция CORRELATION - это функция с несколькими параметрами, для нее требуются два входных параметра. В данном случае администратор базы данных использует в качестве этих двух входных параметров столбцы Sales и Ad Costs.

Администратор базы данных должен использовать эту функцию агрегации с несколькими параметрами в качестве первой функции в сценарии агрегации. Эту функцию с несколькими параметрами можно применить для всех измерений; другой вариант - применить ее сначала для всех измерений, кроме измерения времени, а затем применить к измерению времени другую функцию, например, MAX. Администратор базы данных определяет для показателя выражение SQL, напрямую отображаемое в столбец Ad Costs. Это выражение SQL будет первым из двух параметров, используемых в этой функции с несколькими параметрами. Администратор базы данных определяет второй параметр как выражение SQL, напрямую отображаемое в столбец Sales. Функция CORRELATION определена как единственная функция агрегации, поэтому этот показатель будет вычислять статистическую корреляцию между затратами на рекламу и объемом продаж по всем измерениям.

Шаги для создания показателя

Следующие шаги объясняют, как можно использовать окно Свойства фактографического объекта OLAP Center для создания показателя Advertising-Sales Correlation (корреляция между затратами на рекламу и объемом продаж) в существующем фактографическом объекте:

1. Откройте окно Свойства фактографического объекта, для чего щелкните правой кнопкой мыши по фактографическому объекту в дереве объектов OLAP Center и выберите **Изменить показатели**.
2. Нажмите кнопку **Создать вычисленный показатель**. Откроется окно Построитель выражений SQL.
3. В окне Построитель выражений SQL введите ADVERTISING-SALES CORRELATION в поле **Имя**.
4. Определите для показателя выражение, которое будет использоваться в качестве первого параметра функции CORRELATION в сценарии агрегации. Чтобы определить это выражение, раскройте папку **Показатели** в списке **Данные** и дважды щелкните по показателю **AD COSTS**, чтобы добавить его в список **Выражение**. На рис. 3 показано выражение, созданное в окне Построитель выражений SQL.

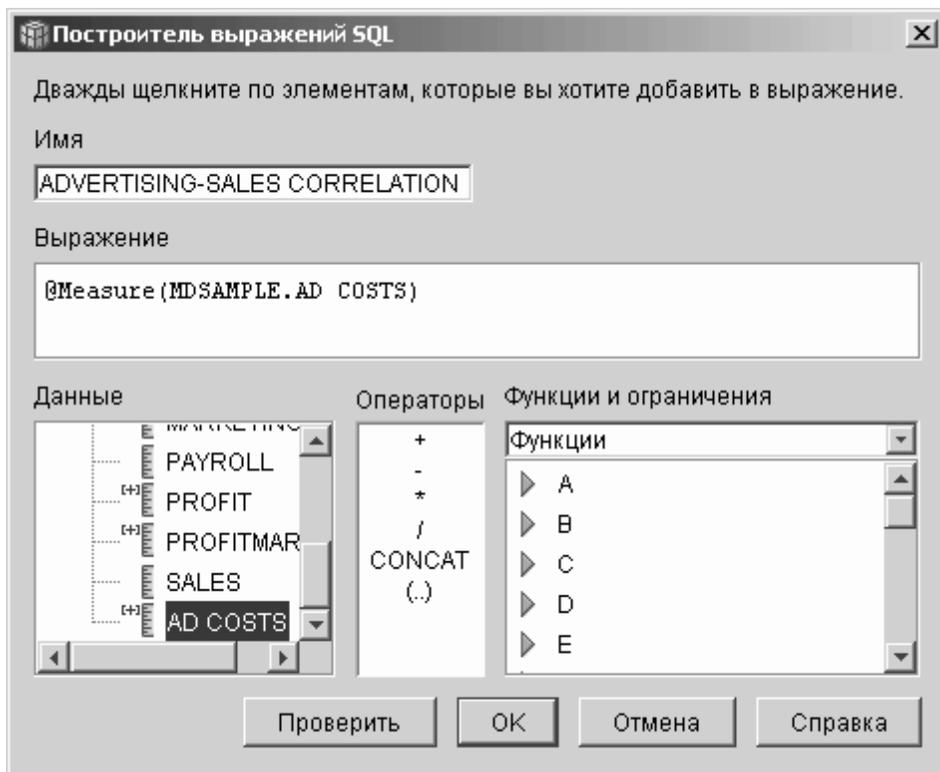


Рисунок 3. Полное выражение показателя Advertising-Sales Correlation в окне Построитель выражений SQL

5. На странице Агрегации выберите функцию агрегации для показателя **ADVERTISING-SALES CORRELATION** и выберите **Сценарий агрегации**, как

показано на рис. 4. Откроется окно Построитель сценариев агрегации.

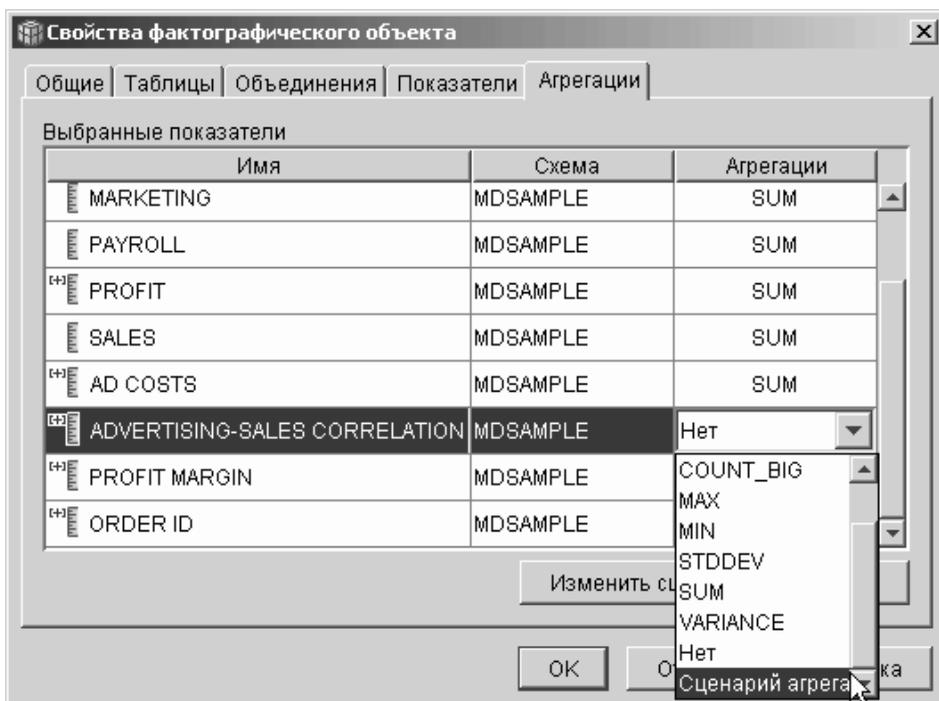


Рисунок 4. Страница Агрегации окна Свойства фактографического объекта

- В поле **Функции столбцов** выберите **Функции с несколькими параметрами**. В списке функций с несколькими параметрами выберите функцию **CORRELATION** и нажмите кнопку **Добавить в сценарий**. На рис. 5 на стр. 10 показано, что функция CORRELATION находится в начале списка измерений в сценарии.

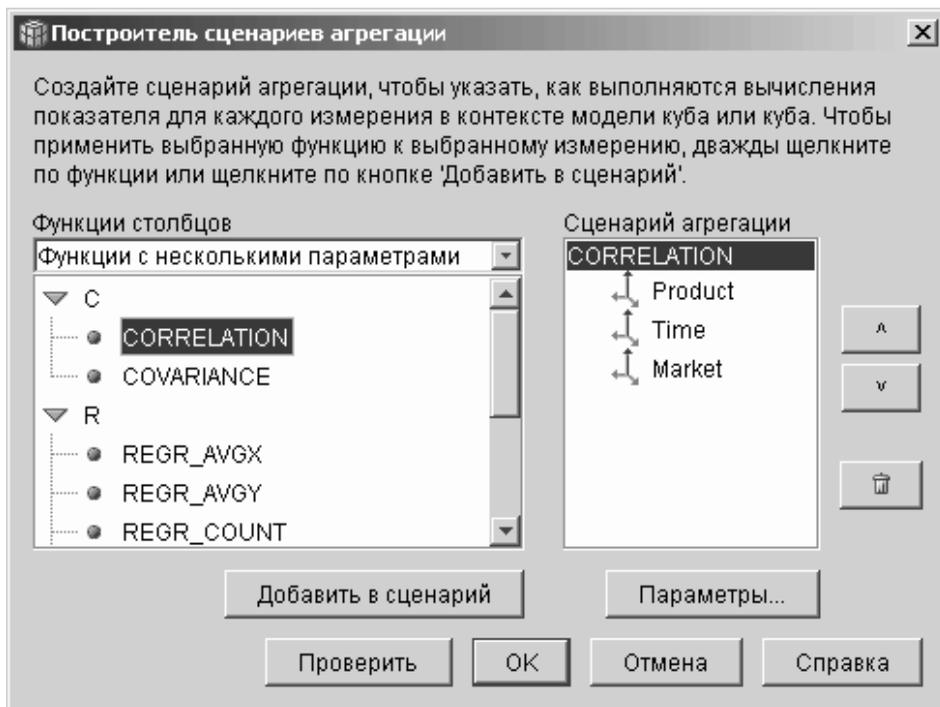


Рисунок 5. Сценарий агрегации для показателя Advertising-Sales Correlation

7. Нажмите кнопку **Параметры**, чтобы задать второй параметр для функции CORRELATION. Откроется окно Параметры функции. Выберите **Использовать существующий показатель** и выберите показатель **SALES**. На рис. 6 на стр. 11 показано окно Параметры функции с показателем Sales, заданным в качестве второго параметра. Нажмите кнопку **ОК**, чтобы сохранить выбранные параметры и закрыть окно Параметры функции.

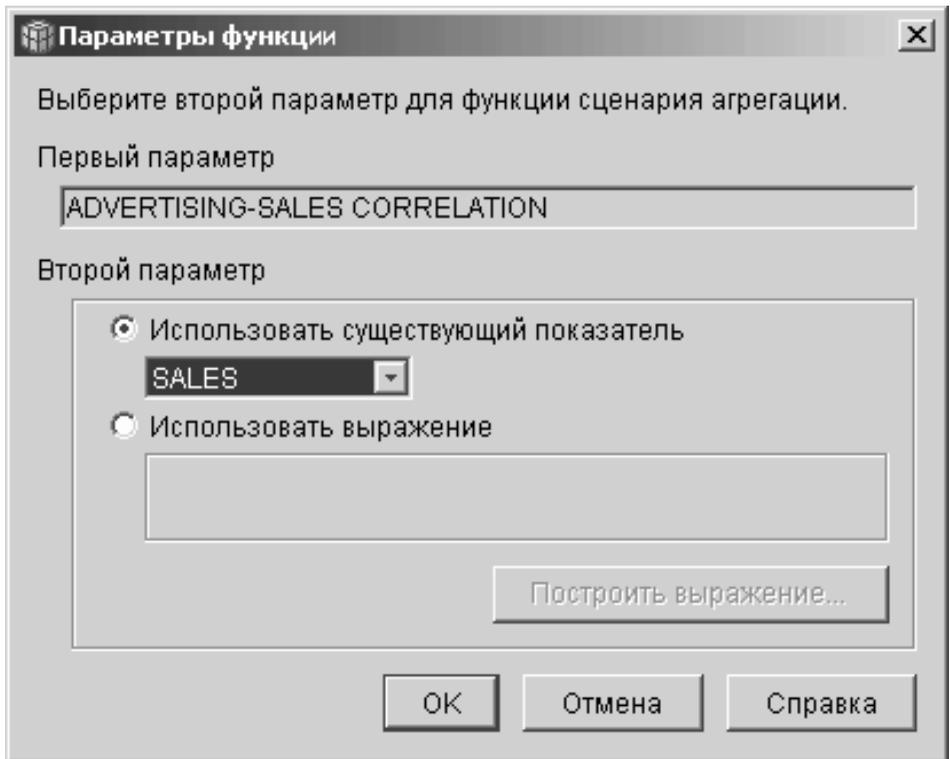


Рисунок 6. Показатель Sales задан в качестве второго параметра в окне Параметры функции specified as the second parameter in the Function Parameters window

8. В окне Построитель сценариев агрегации нажмите кнопку **Проверить**, чтобы проверить правильность сценария агрегации. Нажмите кнопку **ОК**, чтобы сохранить сценарий агрегации и закрыть это окно.
9. Нажмите кнопку **ОК**, чтобы сохранить изменения в фактографическом объекте и закрыть окно Свойства фактографического объекта.

Теперь у вас есть показатель, вычисляющий корреляцию между двумя типами данных в базе данных. Этот показатель можно использовать для принятия решений о будущих затратах на рекламу на основе обнаруженных закономерностей.

Глава 3. Вычисление прибыли и уровня прибыли для магазина

Генеральный менеджер магазина игрушек хочет иметь возможность анализировать то, как различные факторы (например, время года или тип продукта) влияют на прибыль и уровень прибыли. Прежде, чем можно будет проводить более сложный анализ, администратор базы данных магазина игрушек должен создать показатели Profit (прибыль) и Profit Margin (уровень прибыли). Затем администратор базы данных может создать дополнительные показатели для вычисления корреляции и сравнения различных факторов с этими показателями прибыли и уровня прибыли.

Подробности сценария

В базе данных магазина игрушек есть таблица фактов со столбцами Sales (объем продаж), COGS (Costs of Goods Sold - стоимость проданных товаров) и Expense (затраты), а также соответствующие столбцы внешних ключей для каждой из отдельных таблиц измерений. Администратор базы данных уже создал показатели Sales, COGS и Expense, отображаемые на столбцы Sales, COGS и Expense, соответственно. На основе этих показателей можно создать показатели Profit и Profit Margin.

Чтобы создать показатель Profit, администратор базы данных создает показатель, значение которого вычисляется как выражение SQL `SALES-(COGS+EXPENSE)`, и суммирует вычисленные данные по всем измерениям. При создании показателя Profit можно использовать существующие показатели, столбцы или и то, и другое.

Создав показатель Profit, администратор базы данных может создать показатель Profit Margin. Показатель Profit Margin - это отношение двух существующих показателей, выраженное в процентах - $(\text{Profit} / \text{Sales}) * 100$, поэтому для него не требуется функция агрегации. Функция агрегации не требуется, так как показатель ссылается только на другие показатели, данные которых уже являются результатами функций агрегации. Если используется составной показатель (то есть показатель, ссылающийся только на другие показатели) для вычисления относительных значений, не нужно определять дополнительные агрегации. Обычно функции агрегации (например, SUM) нет смысла использовать для относительных значений. Например, если у магазина игрушек в четырех последовательных кварталах уровень прибыли был равен 40%, 32%, 28% и 37%, суммирование этих значений по времени даст уровень прибыли 137% за год, что не имеет смысла.

Шаги создания показателей

Следующие шаги объясняют, как можно использовать окно Свойства фактографического объекта OLAP Center для создания показателей Profit и Profit Margin в существующем фактографическом объекте:

1. Чтобы открыть окно Свойства фактографического объекта, щелкните правой кнопкой мыши по фактографическому объекту в дереве объектов OLAP Center и выберите **Изменить показатели**. Откроется окно Свойства фактографического объекта.
2. Создайте показатель Profit:
 - a. На странице Показатели нажмите кнопку **Создать вычисленный показатель**. Откроется окно Построитель выражений SQL.
 - b. В окне Построитель выражений SQL введите PROFIT в поле **Имя**.
 - c. Чтобы создать выражение для показателя Profit, раскройте папку **Показатели** в списке **Данные** и выполните следующие действия:
 - Дважды щелкните по показателю **SALES** в списке **Данные**, чтобы добавить его в выражение.
 - Дважды щелкните по операции – в списке **Операторы**.
 - Дважды щелкните по показателю **COGS** в списке **Данные**.
 - Дважды щелкните по операции **+** в списке **Операторы**.
 - Дважды щелкните по показателю **EXPENSE** в списке **Данные**.
 - В поле **Выражение** выделите следующий фрагмент выражения: @Measure(MDSAMPLE.COGS)+@Measure(MDSAMPLE.EXPENSE) и дважды щелкните по операции **(. .)** в списке **Операторы**, чтобы заключить выделенный фрагмент выражения в скобки.

На рис. 7 на стр. 15 показано выражение показателя Profit, которое можно создать в окне Построитель выражений SQL.

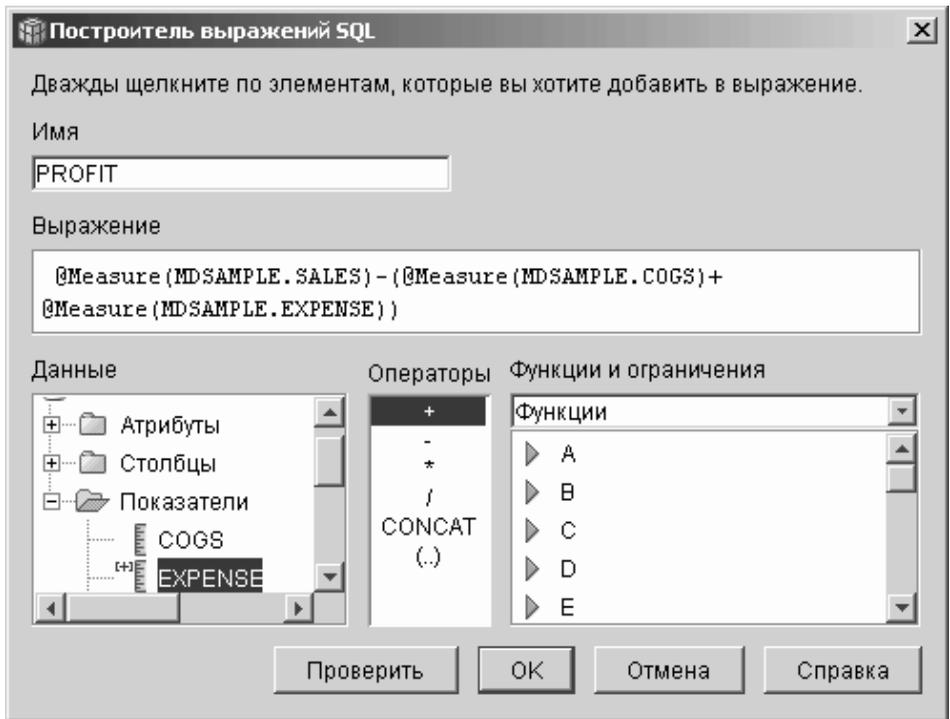


Рисунок 7. Полное выражение показателя Profit в окне Построитель выражений SQL

- d. Нажмите кнопку **ОК**, чтобы создать показатель Profit и закрыть окно Построитель выражений SQL.
 - e. На странице Агрегации выберите агрегацию для показателя Profit и выберите функцию SUM. Показатель Profit создан.
3. Создайте показатель Profit Margin:
 - a. На странице Показатели нажмите кнопку **Создать вычисленный показатель**. Откроется окно Построитель выражений SQL.
 - b. В окне Построитель выражений SQL введите PROFIT MARGIN в поле **Имя**.
 - c. Чтобы создать выражение для показателя Profit Margin, раскройте папку **Показатели** в списке **Данные** и выполните следующие действия:
 - Дважды щелкните по показателю **PROFIT** в списке **Данные**, чтобы добавить его в выражение.
 - Дважды щелкните по операции / в списке **Операторы**.
 - Дважды щелкните по показателю **SALES** в списке **Данные**.
 - Заключите все выражение в скобки, для чего введите их в поле **Выражение**.
 - Поместите курсор в конец выражения и дважды щелкните по операции * в списке **Операторы**.

- Введите число 100 в конце выражения в поле **Выражение**.

На рис. 8 показано выражение показателя Profit Margin, которое можно создать в окне Построитель выражений SQL.

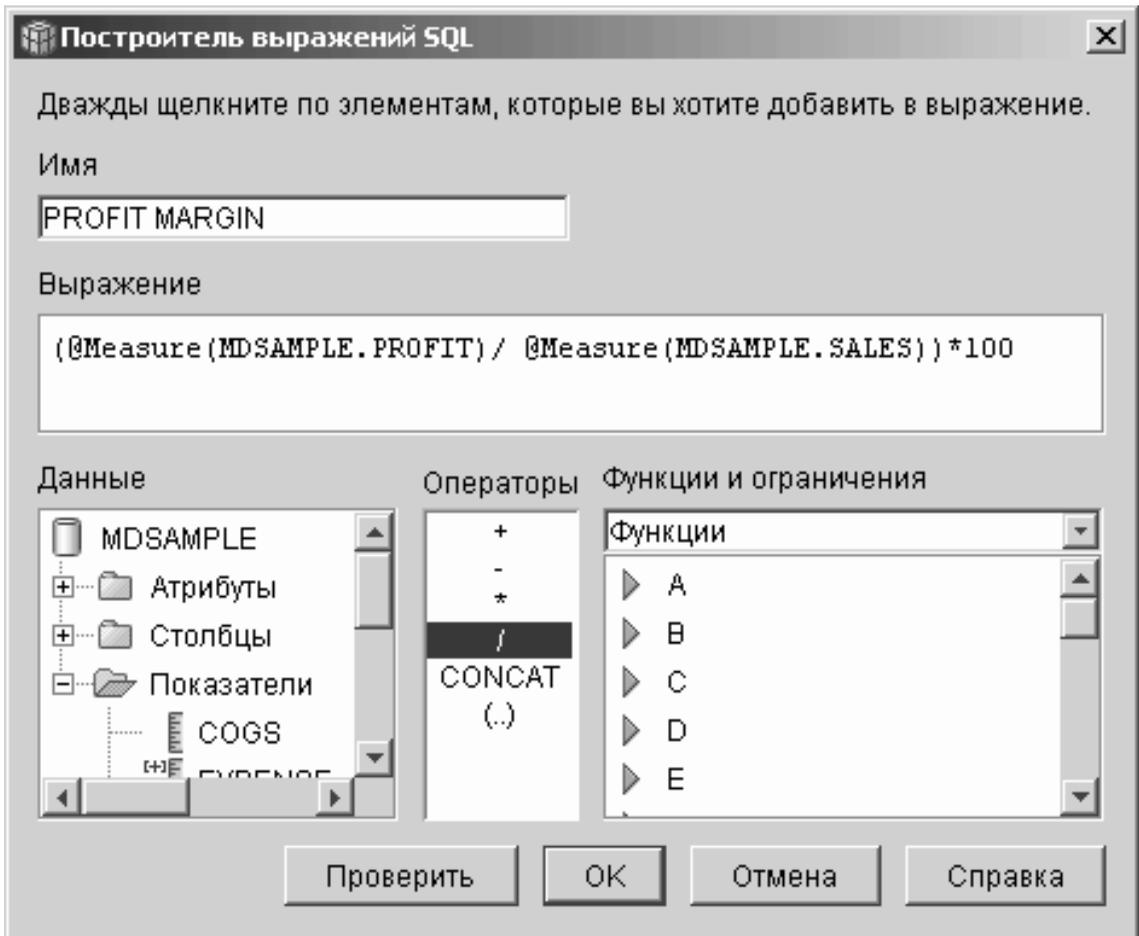


Рисунок 8. Полное выражение показателя Profit Margin в окне Построитель выражений SQL

- d. Нажмите кнопку **ОК**, чтобы создать показатель Profit Margin и закрыть окно Построитель выражений SQL.

OLAP Center задает на странице Агрегации функцию агрегации NONE - функцию по умолчанию для составных показателей, поэтому функцию агрегации изменять не нужно.

4. Нажмите кнопку **ОК**, чтобы закрыть окно Свойства фактографического объекта и сохранить эти два новые показателя в фактографическом объекте.

После того, как администратор базы данных создал эти два показателя, для них можно выполнять дополнительный анализ.

Глава 4. Определение числа заказов через Интернет

Торговая компания расширила свою деятельность, добавив в нее несколько лет тому назад торговлю через Интернет. Теперь в компании хотят проанализировать влияние продаж через Интернет. Одна из первых характеристик, которые нужны компании - это число заказов, сделанных через Интернет.

Подробности сценария

В базе данных компании есть таблица фактов для заказов через Интернет, содержащая столбцы ORDER_ID (ID заказа), PRODUCT_ID (ID продукта), QUANTITY (количество) и TIME_ID (ID времени). В столбце PRODUCT_ID записывается каждый проданный продукт для соответствующего заказа, а в столбце QUANTITY - количество единиц продукта, приобретенных в это заказе. Для заказов с несколькими продуктами в таблице будет по одной строке для каждого отдельного продукта. В Табл. 6 показан пример, где заказ 1 включает три продукта А, один продукт О и один продукт G.

Таблица 6. Часть содержимого таблицы фактов

ORDER_ID	PRODUCT_ID	QUANTITY
1	A	3
1	O	1
1	G	1
2	L	1
2	Q	2
3	P	5

Администратор базы данных может создать показатель Order Count (число заказов), подсчитывающий число различных элементов в столбце ORDER_ID. Для определения показателя Order Count используются ключевое слово DISTINCT в выражении SQL и функция COUNT для агрегации по всем измерениям. Выражение SQL этого показателя создаст список отдельных заказов, а функция агрегации подсчитает их число. Поскольку для этого показателя не используется суммирование, он называется неаддитивным показателем.

Неаддитивные показатели удобны также, когда нужно подсчитать число элементов символьных данных или данных других типов. Например, неаддитивные показатели можно использовать для подсчета числа почтовых индексов, на которые посланы продукты.

В этом примере администратор базы данных решил определить показатель Order ID (ID заказа), непосредственно отображаемый в столбец ORDER_ID. Однако вместо этого показателя можно тем же способом использовать сам столбец ORDER_ID. В зависимости от того, столбец или показатель используется в выражении SQL, агрегация по умолчанию будет различной, но в любом случае нужно изменить агрегацию по умолчанию на функцию COUNT, как описано в разделе “Шаги создания показателя”.

Шаги создания показателя

Следующие шаги объясняют, как можно использовать окно Свойства фактографического объекта OLAP Center для создания показателя Order Count в существующем фактографическом объекте:

1. Чтобы открыть окно Свойства фактографического объекта, щелкните правой кнопкой мыши по фактографическому объекту в дереве объектов OLAP Center и выберите **Изменить показатели**. Откроется окно Свойства фактографического объекта.
2. На странице Показатели нажмите кнопку **Создать вычисленный показатель**. Откроется окно Построитель выражений SQL.
3. В окне Построитель выражений SQL введите ORDER COUNT в поле **Имя**.
4. Чтобы создать выражение для показателя Order Count, раскройте папку **Показатели** в списке **Данные** и выполните следующие действия:
 - В поле **Функции и константы** выберите **Разное**. В списке разных функций и констант дважды щелкните по ключевому слову **DISTINCT**.
 - Дважды щелкните по показателю **ORDER ID** в списке **Данные**.

На рис. 9 на стр. 21 показано выражение показателя Order Count, которое можно создать в окне Построитель выражений SQL.

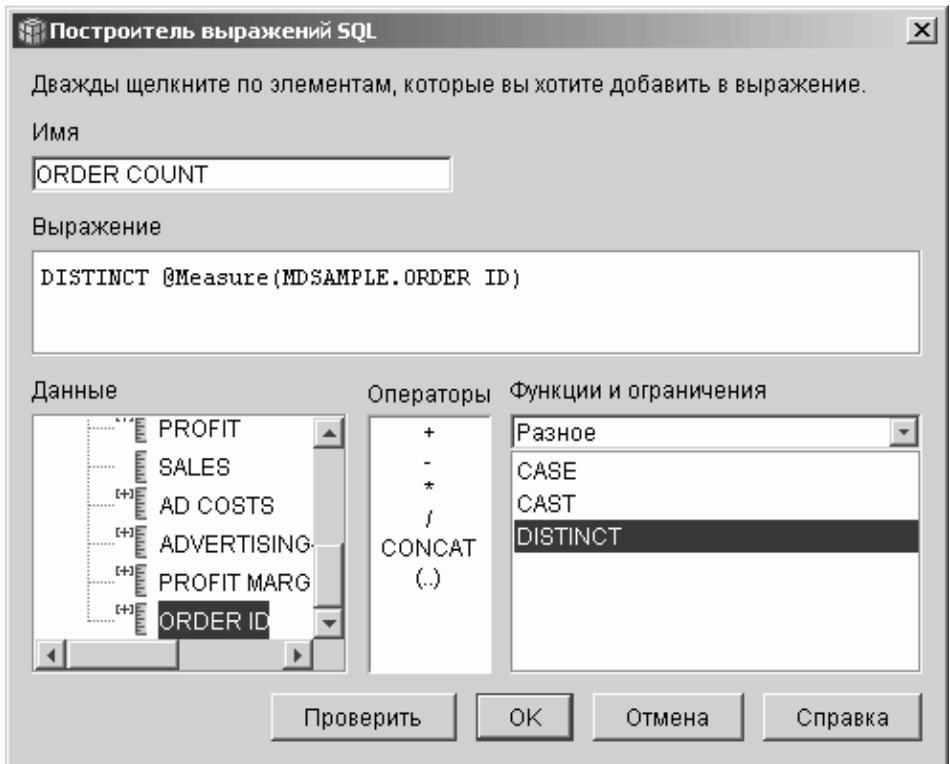


Рисунок 9. Полное выражение показателя Order Count в окне Построитель выражений SQL

5. Нажмите кнопку **ОК**, чтобы закрыть окно Построитель выражений SQL.
6. На странице Агрегации выберите агрегацию для показателя Order Count и выберите функцию **COUNT**.
7. Нажмите кнопку **ОК**, чтобы сохранить изменения в фактографическом объекте и закрыть окно Свойства фактографического объекта.

Теперь у вас есть показатель, подсчитывающий число строк с различными значениями ID заказа. Это показатель можно использовать вместе с другими показателями для дальнейшего анализа данных.

Глава 5. Ранжирование данных о продажах

Сеть магазинов канцелярских товаров быстро расширялась в течение нескольких последних лет. Ее руководители рассматривают вопрос о закрытии наименее прибыльных магазинов, чтобы сократить затраты и повысить прибыль. При принятии решения о закрытии магазина важным фактором являются данные об объеме продаж во времени. Аналитикам нужно иметь возможность ранжировать данные о продажах и затем выполнить углубление по изменениям для завершения анализа.

Подробности сценария

В базе данных этих магазинов канцелярских товаров есть таблица фактов, содержащая столбец Sales (объем продаж), а также другие столбцы. В базе данных есть также некоторые таблицы измерений. Администратор базы данных может создать показатель Sales Rank (ранг продаж), использующий функцию RANK - функцию OLAP, реализованную в DB2 Universal Database (DB2 UDB).

DB2 Cube Views поддерживает следующие функции OLAP, реализованные в DB2 UDB:

RANK

Упорядочивает строки и присваивает каждой из них ранг. Ранг определяется как число строго предшествующих ей с точки зрения критерия упорядочения строк плюс 1. Если относительный порядок двух или нескольких строк невозможно определить, поскольку они содержат одинаковые значения, им присваивается один и тот же номер ранга. Если есть повторяющиеся строки, в значениях результатов ранжирования могут быть пропуски. В Табл. 7 на стр. 25 показан пример результатов ранжирования после применения функции RANK для примера набора значений строк.

Обычный синтаксис функции RANK:

```
RANK ( ) OVER (ORDER BY выражение-ключа-сортировки порядок-сортировки)
```

где *выражение-ключа-сортировки* - набор данных для ранжирования, а *порядок-сортировки* - ключевое слово **ASC** или **DESC**, задающее восходящий или нисходящий порядок сортировки для значений выражения-ключа-сортировки. DB2 Cube Views требует, чтобы *выражение-ключа-сортировки* было существующим показателем, а не столбцом или атрибутом. Кроме того, DB2 Cube Views не поддерживает условие PARTITION BY, которое поддерживается DB2 UDB для этой

функции. Дополнительную информацию о функции RANK можно найти в Информационном центре DB2 UDB.

DENSERANK

Упорядочивает строки и присваивает ранг каждой из них. Ранг строки определяется как число всех строк, предшествующих этой строке, плюс 1. Следовательно, результаты ранжирования будут последовательными и не будут содержать пропусков в номерах рангов. В Табл. 7 на стр. 25 показан пример результатов ранжирования после применения функции DENSERANK для примера набора значений строк.

Обычный синтаксис функции DENSERANK:

DENSERANK () OVER (ORDER BY *выраж-ключа-сорт порядок-сортировки*)

где *выраж-ключа-сорт* - набор данных для ранжирования, а *порядок-сортировки* - ключевое слово **ASC** или **DESC**, задающее восходящий или нисходящий порядок сортировки для значений выражения-ключа-сортировки. DB2 Cube Views требует, чтобы *выражение-ключа-сортировки* было существующим показателем, а не столбцом или атрибутом. Кроме того, DB2 Cube Views не поддерживает условие PARTITION BY, которое поддерживается DB2 UDB для этой функции. Дополнительную информацию о функции DENSERANK можно найти в Информационном центре DB2 UDB.

ROWNUMBER

Вычисляет последовательный номер строки при упорядочении, начиная с 1 для первой строки. Если условие ORDER BY не задано, номера строк присваиваются в произвольном порядке.

Обычный синтаксис функции ROWNUMBER:

ROWNUMBER () OVER ([ORDER BY *выраж-ключа-сорт порядок-сортировки*])

где *выраж-ключа-сорт* - набор данных для ранжирования, а *порядок-сортировки* - ключевое слово **ASC** или **DESC**, задающее восходящий или нисходящий порядок сортировки для значений выражения-ключа-сортировки. DB2 Cube Views требует, чтобы в качестве источника данных для этой функции использовался существующий показатель, а не столбец или атрибут. Кроме того, DB2 Cube Views не поддерживает условие PARTITION BY, которое поддерживается DB2 UDB для этой функции. Дополнительную информацию о функции ROWNUMBER можно найти в Информационном центре DB2 UDB.

Эти функции OLAP не выводятся в списке функций и констант построителя выражений SQL.

Таблица 7. Результаты ранжирования для примера набора значений строк после применения функций RANK и DENSERANK

Значения строки	Упорядочение	Результаты ранжирования при помощи функции RANK	Результаты ранжирования при помощи функции DENSERANK
100	1	1	1
35	2	2	2
23	3	3	3
8	4	4	4
8	4	4	5
6	5	6	6

Шаги создания показателя

Следующие шаги объясняют, как можно использовать окно Свойства фактографического объекта OLAP Center для создания показателя Sales Rank в существующем фактографическом объекте:

1. Чтобы открыть окно Свойства фактографического объекта, щелкните правой кнопкой мыши по фактографическому объекту в дереве объектов OLAP Center и выберите **Изменить показатели**. Откроется окно Свойства фактографического объекта.
2. На странице Показатели нажмите кнопку **Создать вычисленный показатель**, чтобы создать показатель Sales Rank. Откроется окно Построитель выражений SQL.
3. В окне Построитель выражений SQL введите SALES RANK в поле **Имя**.
4. Чтобы создать выражение для этого показателя:
 - В поле Выражение введите следующее выражение: RANK () OVER (ORDER BY measure DESC).
 - Раскройте папку **Показатели** в списке **Данные**.
 - В выражении в поле **Выражение** выделите слово measure и дважды щелкните по показателю **SALES**, чтобы добавить его в выражение.

Готовое выражение показано на рис. 10 на стр. 26.

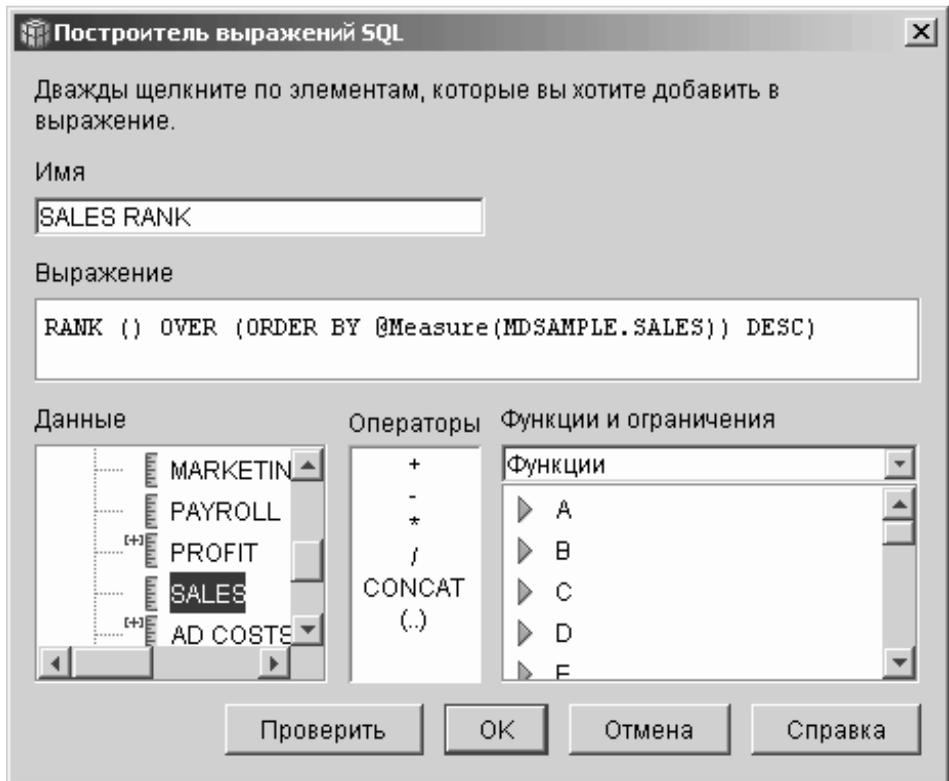


Рисунок 10. Полное выражение показателя Sales Rank в окне Построитель выражений SQL

5. Нажмите кнопку **Проверить**, чтобы проверить правильность этого выражения. Нажмите кнопку **ОК**, чтобы закрыть окно Построитель выражений SQL.

Не изменяйте агрегацию по умолчанию (Нет), заданную на странице Агрегации. Опция Нет используется по умолчанию для показателя Sales Rank, так как у него числовой источник данных и этот показатель ссылается только на другие показатели.

Используя в выражении показателя функцию RANK для упорядочения столбца Sales (объем продаж) в нисходящем порядке, аналитики могут углубиться по другим измерениям, чтобы определить магазин с худшими значениями продаж за последний год для определенной линии продуктов или в отношении других данных измерений, хранящихся в базе данных.

Глава 6. Использование данных из таблицы фактов для создания измерения времени

Торговая компания XYZ Retail использует DB2 Cube Views для моделирования данных продаж, что позволяет эффективнее анализировать эти данные. Однако для данных о продажах доступна только такая информация о времени, как дата продажи. Информация времени, моделируемая в измерении времени, необходима для многих распространенных типов вычислений, например, для анализа тенденций продаж за квартал или вычисления среднего числа товаров за неделю.

Многие администраторы баз данных избегают сохранения данных времени в виде даты или отметки времени, так как если в некоторые дни нет продаж, в данных будут пропуски и это может помешать точному вычислению агрегации данных и вывода их на экран. Обычно лучшим решением для моделирования данных времени оказывается использование таблицы времени. Однако администратор базы данных для компании XYZ Retail уверен, что каждый день будет хотя бы одна продажа, и поэтому он решил не менять существующую структуру данных.

Подробности сценария

В компании XYZ Retail есть таблица фактов с данными о каждой продаже, включающая данные столбцов Sales (объем продаж), Costs (стоимость), Quantity Sold (число проданных) и Date (дата). Кроме того, в базе данных есть таблицы измерений Region (регион) и Product (продукт). Проблема в том, что данные времени включены в таблицу фактов, а не хранятся в отдельной таблице измерения. Администратор базы данных должен создать объект измерения, основанный на данных столбца Date в таблице фактов.

Для создания измерения времени на основе одного столбца данных даты в таблице фактов есть два особых требования:

- Поскольку для всех объектов измерений в правильной модели куба должны быть заданы объединения с фактографическим объектом, а объект измерения времени и фактографический объект основаны на одной и той же таблице фактов, для объединения объекта измерения времени с фактографическим объектом нужно использовать автообъединение таблицы фактов с ней самой.
- Администратор базы данных должен построить вычисленные атрибуты для агрегации данных даты на таких уровнях, как неделя, месяц, квартал и год.

Автообъединение - это тип объединения, объединяющий таблицу с ней самой; в нашем случае это таблица фактов. В автообъединении должны задаваться один

или несколько столбцов, совокупность которых позволяет однозначно определить любую строку в таблице фактов. Лучше всего использовать первичный ключ. Но если первичный ключ еще не определен, можно использовать для его определения набор столбцов, используемых для объединения таблицы фактов с таблицами измерений. Для оптимизации модели куба первичный ключ должен быть определен. Объединение должно иметь мощность 1:1 и тип "внутреннее".

На рис. 11 на стр. 29 показано, как фактографический объект, измерение, основанное на таблице фактов, и объединение фактографического объекта с измерением могут отображаться на одну и ту же таблицу фактов.

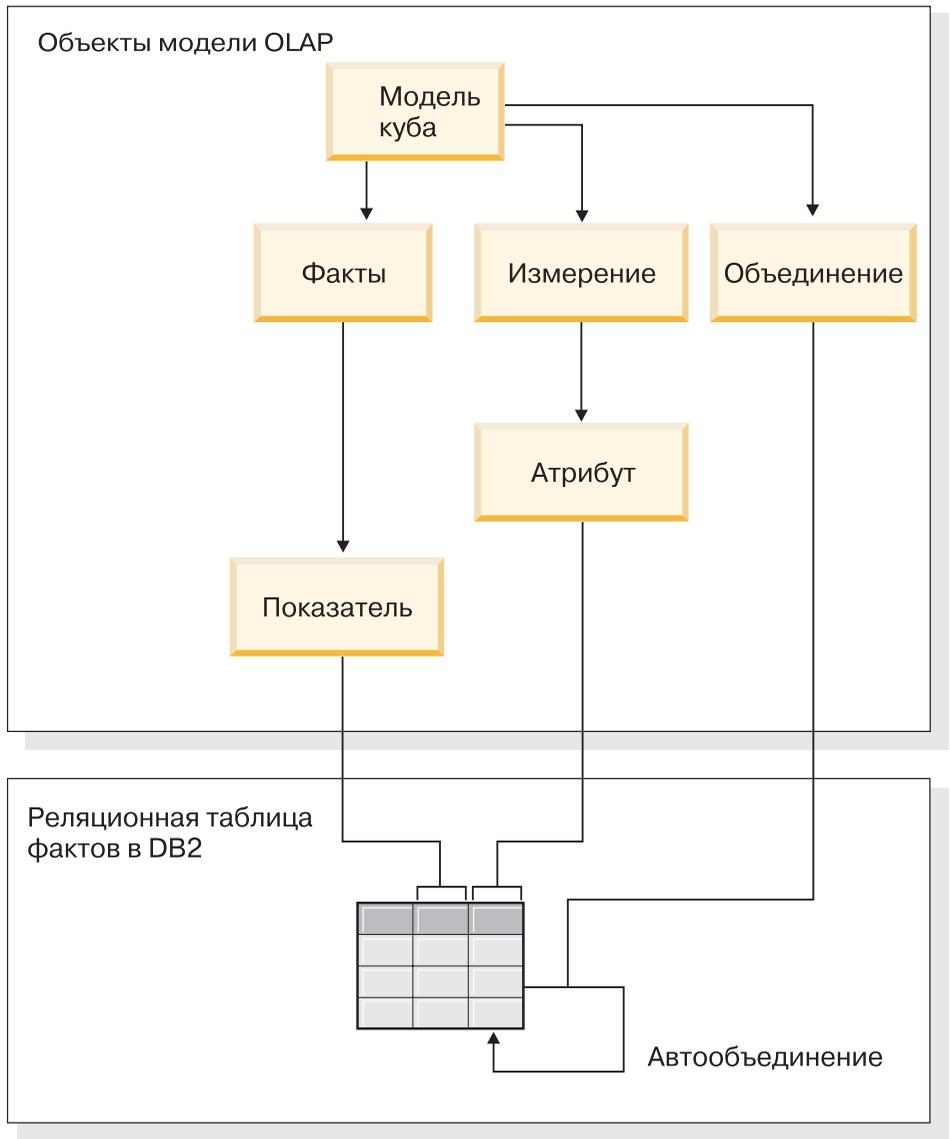


Рисунок 11. Автообъединение таблицы с ней самой

Шаги для создания атрибутов и измерения

Следующие шаги объясняют, как можно использовать окно Свойства фактографического объекта OLAP Center для создания измерения Time (время) и вычисленных атрибутов на основе таблицы фактов:

1. Чтобы открыть мастер измерений, щелкните правой кнопкой мыши по модели куба в дереве объектов OLAP Center и выберите **Создать измерение**. Откроется мастер измерений.
2. На странице Имя введите Time в поле **Имя**. Можно также изменить рабочее имя и ввести комментарий. Нажмите кнопку **Далее**.
3. Выберите таблицу фактов этой модели куба. Нажмите кнопку **Далее**. Объединения измерения задавать не нужно, так как в этом измерении только одна таблица. На странице Объединения измерения нажмите кнопку **Далее**.
4. На странице Атрибуты измерения выберите столбец **Timestamp** (Отметка времени).
5. Необязательно: Создайте дополнительные вычисленные атрибуты для агрегации данных этого столбца по интервалам времени, таким как месяц, квартал и год. Чтобы создать вычисленные атрибуты, нажмите кнопку **Создать вычисленный атрибут**, чтобы открыть построитель выражений SQL, и определите выражение для каждого атрибута, вычисляющего по данным отметки времени из исходного столбца значения для месяцев, кварталов и годов. Определив каждый вычисленный атрибут, нажмите кнопку **Проверить**, чтобы проверить правильность выражения, затем нажмите кнопку **ОК**, чтобы закрыть построитель выражений SQL и вернуться в мастер измерений. Создав все нужные атрибуты, нажмите кнопку **Далее**.
6. На странице Тип измерения выберите **Время**. Нажмите кнопку **Далее**.
7. На странице Объединение фактографического объекта с измерением нажмите кнопку **Создать объединение**. В открывшемся мастере объединений создайте автообъединение. Введите имя и нажмите кнопку **Далее**. В качестве как левого, так и правого атрибутов объединения выберите столбец или набор столбцов, которые однозначно определяют любую строку в таблице фактов, например, первичный ключ. Выбирайте за раз одну пару столбцов и нажимайте кнопку **Добавить**, чтобы добавить эту пару атрибутов в объединение. Выберите тип объединения "внутреннее" и мощность 1:1. Добавив необходимые пары атрибутов, нажмите кнопку **Готово**. Мастер объединений закроется.
8. На странице Объединение фактографического объекта с измерением нажмите кнопку **Готово**.

Когда в модели куба определено измерение времени, компания XYZ Retail расширит возможности анализа данных. Теперь можно выполнять анализ во временной области, включая в него данные о товарах.

Замечания

Эта информация разрабатывалась для продуктов и услуг, предлагаемых в США.

IBM может не предоставлять в других странах продукты, услуги и аппаратные средства, описанные в данном документе. За информацией о продуктах и услугах, предоставляемых в вашей стране, обращайтесь к местному представителю IBM. Ссылки на продукты, программы или услуги IBM не означают и не предполагают, что можно использовать только указанные продукты, программы или услуги. Вместо них разрешается использовать любые функционально эквивалентные продукты, программы или услуги, если при этом не нарушаются права IBM на интеллектуальную собственность. Однако ответственность за оценку и проверку их работы в сочетании с продуктами, программами или услугами других фирм (не IBM) несет пользователь.

IBM может располагать патентами или рассматриваемыми заявками на патенты, относящимися к предмету данной публикации. Получение данного документа не означает предоставления каких-либо лицензий на эти патенты. Запросы по поводу лицензий можно направлять в письменной форме по адресу:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

С запросами относительно информации о лицензиях для стран с двухбайтным набором символов (DBCS) обращайтесь в отдел интеллектуальной собственности IBM (IBM Intellectual Property Department) в вашей стране или направляйте письменные запросы по адресу:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

Приведенные в следующем абзаце положения не касаются Великобритании и прочих стран, в которых они входят в противоречие с местными законами:
КОРПОРАЦИЯ INTERNATIONAL BUSINESS MACHINES ПРЕДОСТАВЛЯЕТ ДАННУЮ ПУБЛИКАЦИЮ “КАК ЕСТЬ” БЕЗ КАКИХ-ЛИБО ЯВНЫХ ИЛИ ПРЕДПОЛАГАЕМЫХ ГАРАНТИЙ, ВКЛЮЧАЯ, НО НЕ ОГРАНИЧИВАЯСЬ ТАКОВЫМИ, ПРЕДПОЛАГАЕМЫЕ ГАРАНТИИ ОТНОСИТЕЛЬНО СОБЛЮДЕНИЯ ЧЬИХ-ЛИБО АВТОРСКИХ ПРАВ, КОММЕРЧЕСКОГО ИСПОЛЬЗОВАНИЯ ИЛИ ПРИГОДНОСТИ ДЛЯ ТЕХ

ИЛИ ИНЫХ ЦЕЛЕЙ. В некоторых странах для ряда сделок не допускается отказ от явных или предполагаемых гарантий; в таком случае данное положение к вам не относится.

В данном материале могут встретиться технические неточности или типографские опечатки. В публикацию время от времени вносятся изменения, которые будут отражены в следующих изданиях. IBM оставляет за собой право в любое время вносить усовершенствования и/или изменения в описанные в этом замечании продукты и/или программы.

IBM оставляет за собой право распространять любую предоставленную вами информацию любыми способами по своему усмотрению, не принимая на себя никаких обязательств по отношению к вам.

Если лицам, получившим лицензию на эту программу, нужна более подробная информация о ней для: (i) обмена данными между независимо созданными программами и прочими программами (включая и эту) и (ii) совместного использования этих данных, то они должны обратиться по адресу:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

Такая информация может быть предоставлена при соблюдении определенных положений и условий и, возможно, за определенную плату.

Описываемую в этой публикации лицензионную программу и все прилагаемые к ней лицензионные материалы IBM предоставляет на условиях IBM Customer Agreement (Соглашения между IBM и Заказчиком), Международного Соглашения о Лицензиях на Программы или любого эквивалентного соглашения.

Информация, касающаяся продуктов других компаний (не IBM) была получена от поставщиков этих продуктов, из опубликованных ими заявлений или из прочих общедоступных источников. IBM не производила тестирование этих продуктов и никак не может подтвердить информацию о точности их работы и совместимости, а также прочие заявления относительно продуктов других компаний (не IBM). Вопросы относительно возможностей продуктов других компаний (не IBM) следует адресовать поставщикам этих продуктов.

Приведенная здесь информация содержит примеры данных и отчетов, используемых в ежедневных бизнес-операциях. Чтобы примеры были как можно нагляднее, в них содержатся имена, названия компаний, фирменных товарных

знаков и продуктов. Все эти названия являются вымышленными и любое сходство с именами и адресами реальных предприятий является чисто случайным.

Все приводимые здесь данные о производительность получены в контролируемой среде. Результаты, полученные в других операционных средах, могут значительно от них отличаться. Некоторые измерения производились на компьютерах разработчиков, и нет никаких гарантий, что результаты будут такими же на обычно используемых компьютерах. Более того, некоторые результаты могли быть получены путем экстраполяции. Реальные результаты могут быть другими. Пользователи должны проверить данные в своей собственной среде.

Информация, касающаяся продуктов других компаний (не IBM) была получена от поставщиков этих продуктов, из опубликованных ими заявлений или из прочих общедоступных источников. IBM не производила тестирование этих продуктов и никак не может подтвердить информацию о точности их работы и совместимости, а также прочие заявления относительно продуктов других компаний (не IBM). Вопросы относительно возможностей продуктов других компаний (не IBM) следует адресовать поставщикам этих продуктов.

При просмотре этого материала на компьютере, фотографии и цветные иллюстрации могут не появиться на экране.

Товарные знаки

Перечисленные ниже термины являются товарными знаками International Business Machines Corporation в Соединенных Штатах и/или других странах:

DB2
DB2 Universal Database
IBM
Office Connect
Redbooks

Перечисленные ниже термины являются товарными знаками или зарегистрированными товарными знаками других компаний:

Microsoft, Windows, Windows NT, Windows 2000, Windows XP и Microsoft Excel - товарные знаки или зарегистрированные товарные знаки Корпорации Microsoft.

Java, все основанные на Java товарные знаки и логотипы и Solaris - товарные знаки Sun Microsystems, Inc. в США и/или других странах.

UNIX - зарегистрированный товарный знак в США и/или других странах, который лицензируется исключительно через X/Open Company Limited.

Linux - зарегистрированный товарный знак Линуса Торвальдса. Red Hat и все товарные знаки и логотипы на основе Hat - товарные знаки или зарегистрированные товарные знаки Red Hat, Inc. в Соединенных Штатах и других странах.

Прочие названия фирм, продуктов или услуг могут являться товарными знаками или сервисными марками других фирм.