Министерство образования и науки России

ФГБОУ ВО "Севастопольский государственный университет"

Кафедра ИС

Реферат на тему:

Введение в OLAP. Основы OLAP

Выполнил:

ст. гр. ИTб-33д

Лисянский А. И.

Севастополь

2015

## Что такое хранилище данных

Информационные системы масштаба предприятия, как правило, содержат приложения, предназначенные для комплексного многомерного анализа данных, их динамики, тенденций и т.п. Такой анализ в конечном итоге призван содействовать принятию решений. Нередко эти системы так и называются — системы поддержки принятия решений.

Принять любое управленческое решение невозможно не обладая необходимой для этого информацией, обычно количественной. Для этого необходимо создание хранилищ данных (Data warehouses), то есть процесс сбора, отсеивания и предварительной обработки данных с целью предоставления результирующей информации пользователям для статистического анализа (а нередко и создания аналитических отчетов).

Ральф Кимбалл (Ralph Kimball), один из авторов концепции хранилищ данных, описывал хранилище данных как "место, где люди могут получить доступ к своим данным" (см., например, Ralph Kimball, "The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses", John Wiley & Sons, 1996 и "The Data Webhouse Toolkit: Building the Web-Enabled Data Warehouse", John Wiley & Sons, 2000). Он же сформулировал и основные требования к хранилищам данных:

* поддержка высокой скорости получения данных из хранилища;
* поддержка внутренней непротиворечивости данных;
* возможность получения и сравнения так называемых срезов данных (slice and dice);
* наличие удобных утилит просмотра данных в хранилище;
* полнота и достоверность хранимых данных;
* поддержка качественного процесса пополнения данных.

Удовлетворять всем перечисленным требованиям в рамках одного и того же продукта зачастую не удается. Поэтому для реализации хранилищ данных обычно используется несколько продуктов, одни их которых представляют собой собственно средства хранения данных, другие — средства их извлечения и просмотра, третьи — средства их пополнения и т.д.

Типичное хранилище данных, как правило, отличается от обычной реляционной базы данных. Во-первых, обычные базы данных предназначены для того, чтобы помочь пользователям выполнять повседневную работу, тогда как хранилища данных предназначены для принятия решений. Например, продажа товара и выписка счета производятся с использованием базы данных, предназначенной для обработки транзакций, а анализ динамики продаж за несколько лет, позволяющий спланировать работу с поставщиками, — с помощью хранилища данных.

Во-вторых, обычные базы данных подвержены постоянным изменениям в процессе работы пользователей, а хранилище данных относительно стабильно: данные в нем обычно обновляются согласно расписанию (например, еженедельно, ежедневно или ежечасно — в зависимости от потребностей). В идеале процесс пополнения представляет собой просто добавление новых данных за определенный период времени без изменения прежней информации, уже находящейся в хранилище.

И в-третьих, обычные базы данных чаще всего являются источником данных, попадающих в хранилище. Кроме того, хранилище может пополняться за счет внешних источников, например статистических отчетов.

## Что такое OLAP

Системы поддержки принятия решений обычно обладают средствами предоставления пользователю агрегатных данных для различных выборок из исходного набора в удобном для восприятия и анализа виде. Как правило, такие агрегатные функции образуют многомерный (и, следовательно, нереляционный) набор данных (нередко называемый гиперкубом или метакубом), оси которого содержат параметры, а ячейки — зависящие от них агрегатные данные - причем храниться такие данные могут и в реляционных таблицах, но в данном случае мы говорим о логической организации данных, а не о физической реализации их хранения). Вдоль каждой оси данные могут быть организованы в виде иерархии, представляющей различные уровни их детализации. Благодаря такой модели данных пользователи могут формулировать сложные запросы, генерировать отчеты, получать подмножества данных.

Технология комплексного многомерного анализа данных получила название OLAP (On-Line Analytical Processing). OLAP — это ключевой компонент организации хранилищ данных. Концепция OLAP была описана в 1993 году Эдгаром Коддом, известным исследователем баз данных и автором реляционной модели данных (см. E.F. Codd, S.B. Codd, and C.T.Salley, Providing OLAP (on-line analytical processing) to user-analysts: An IT mandate. Technical report, 1993). В 1995 году на основе требований, изложенных Коддом, был сформулирован так называемый тест FASMI (Fast Analysis of Shared Multidimensional Information — быстрый анализ разделяемой многомерной информации), включающий следующие требования к приложениям для многомерного анализа:

* предоставление пользователю результатов анализа за приемлемое время (обычно не более 5 с), пусть даже ценой менее детального анализа;
* возможность осуществления любого логического и статистического анализа, характерного для данного приложения, и его сохранения в доступном для конечного пользователя виде;
* многопользовательский доступ к данным с поддержкой соответствующих механизмов блокировок и средств авторизованного доступа;
* многомерное концептуальное представление данных, включая полную поддержку для иерархий и множественных иерархий (это — ключевое требование OLAP);
* возможность обращаться к любой нужной информации независимо от ее объема и места хранения.

Следует отметить, что OLAP-функциональность может быть реализована различными способами, начиная с простейших средств анализа данных в офисных приложениях и заканчивая распределенными аналитическими системами, основанными на серверных продуктах. Но прежде чем говорить о различных реализациях этой функциональности, давайте рассмотрим, что же представляют собой кубы OLAP с логической точки зрения.

## Многомерные кубы

В данном разделе мы более подробно рассмотрим концепцию OLAP и многомерных кубов. В качестве примера реляционной базы данных, который мы будем использовать для иллюстрации принципов OLAP, воспользуемся базой данных Northwind, входящей в комплекты поставки Microsoft SQL Server или Microsoft Access и представляющей собой типичную базу данных, хранящую сведения о торговых операциях компании, занимающейся оптовыми поставками продовольствия. К таким данным относятся сведения о поставщиках, клиентах, компаниях, осуществляющих доставку, список поставляемых товаров и их категорий, данные о заказах и заказанных товарах, список сотрудников компании. Подробное описание базы данных Northwind можно найти в справочных системах Microsoft SQL Server или Microsoft Access — здесь за недостатком места мы его не приводим.

Для рассмотрения концепции OLAP воспользуемся представлением Invoices и таблицами Products и Categories из базы данных Northwind, создав запрос, в результате которого получим подробные сведения о всех заказанных товарах и выписанных сетах.

Этот запрос обращается к представлению Invoices, содержащему сведения обо всех выписанных счетах, а также к таблицам Categories и Products, содержащим сведения о категориях продуктов, которые заказывались, и о самих продуктах соответственно. В результате этого запроса мы получим набор данных о заказах, включающий категорию и наименование заказанного товара, дату размещения заказа, имя сотрудника, выписавшего счет, город, страну и название компании-заказчика, а также наименование компании, отвечающей за доставку.

Для удобства сохраним этот запрос в виде представления, назвав его Invoices1. Результат обращения к этому представлению приведен на рис. 1.

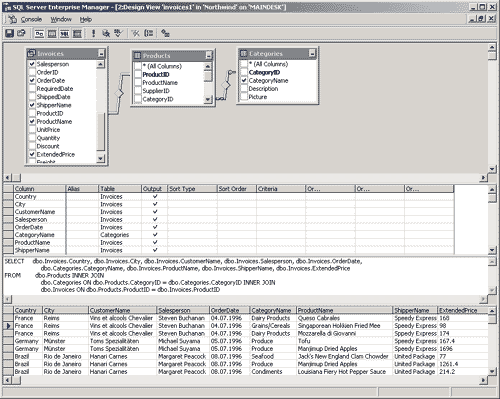


Рис. 1. Результат обращения к представлению Invoices1

Какие агрегатные данные мы можем получить на основе этого представления? Обычно это ответы на вопросы типа:

* Какова суммарная стоимость заказов, сделанных клиентами из Франции?
* Какова суммарная стоимость заказов, сделанных клиентами из Франции и доставленных компанией Speedy Express?
* Какова суммарная стоимость заказов, сделанных клиентами из Франции в 1997 году и доставленных компанией Speedy Express?

Переведем эти вопросы в запросы на языке SQL (аналогичные вопросы, сформулированные на английском языке, можно превратить в SQL-запросы с помощью Microsoft English Query, однако рассмотрение подобных средств выходит за рамки данной статьи).

|  |  |
| --- | --- |
| Вопрос | SQL-запрос |
| Какова суммарная стоимость заказов, сделанных клиентами из Франции? | SELECT SUM (ExtendedPrice) FROM invoices1 WHERE Country=’France’ |
| Какова суммарная стоимость заказов, сделанных клиентами из Франции и доставленных компанией Speedy Express? | SELECT SUM (ExtendedPrice) FROM invoices1 WHERE Country=’France’ AND ShipperName=’Speedy Express’ |
| Какова суммарная стоимость заказов, сделанных клиентами из Франции в 1996 году и доставленных компанией Speedy Express? | SELECT SUM (ExtendedPrice) FROM Ord\_pmt WHERE CompanyName=’Speedy Express’ AND OrderDate BETWEEN ‘December 31, 1995’ AND ‘April 1, 1996’ AND ShipperName=’Speedy Express’ |

Результатом любого из перечисленных выше запросов является число. Если в первом из запросов заменить параметр ‘France’ на ‘Austria’ или на название иной страны, можно снова выполнить этот запрос и получить другое число. Выполнив эту процедуру со всеми странами, мы получим следующий набор данных (ниже показан фрагмент):

|  |  |
| --- | --- |
| Country | SUM (ExtendedPrice) |
| Argentina | 7327.3 |
| Austria | 110788.4 |
| Belgium | 28491.65 |
| Brazil | 97407.74 |
| Canada | 46190.1 |
| Denmark | 28392.32 |
| Finland | 15296.35 |
| France | 69185.48 |
| Germany | 209373.6 |
| … | … |

Полученный набор агрегатных значений (в данном случае — сумм) может быть интерпретирован как одномерный набор данных. Этот же набор данных можно получить и в результате запроса с предложением GROUP BY следующего вида:

SELECT Country, SUM (ExtendedPrice) FROM invoices1

GROUP BY Country

Теперь обратимся ко второму из приведенных выше запросов, который содержит два условия в предложении WHERE. Если выполнять этот запрос, подставляя в него все возможные значения параметров Country и ShipperName, мы получим двухмерный набор данных следующего вида (ниже показан фрагмент):

|  |  |  |  |
| --- | --- | --- | --- |
|  | ShipperName | | |
| Country | Federal Shipping | Speedy Express | United Package |
| Argentina | 1 210.30 | 1 816.20 | 5 092.60 |
| Austria | 40 870.77 | 41 004.13 | 46 128.93 |
| Belgium | 11 393.30 | 4 717.56 | 17 713.99 |
| Brazil | 16 514.56 | 35 398.14 | 55 013.08 |
| Canada | 19 598.78 | 5 440.42 | 25 157.08 |
| Denmark | 18 295.30 | 6 573.97 | 7 791.74 |
| Finland | 4 889.84 | 5 966.21 | 7 954.00 |
| France | 28 737.23 | 21 140.18 | 31 480.90 |
| Germany | 53 474.88 | 94 847.12 | 81 962.58 |
| … | … | … | … |

Такой набор данных называется сводной таблицей (pivot table) или кросс-таблицей (cross table, crosstab). Создавать подобные таблицы позволяют многие электронные таблицы и настольные СУБД — от Paradox для DOS до Microsoft Excel 2000. Вот так, например, выглядит подобный запрос в Microsoft Access 2000:

TRANSFORM Sum(Invoices1.ExtendedPrice) AS SumOfExtendedPrice

SELECT Invoices1.Country

FROM Invoices1

GROUP BY Invoices1.Country

PIVOT Invoices1.ShipperName;

Агрегатные данные для подобной сводной таблицы можно получить и с помощью обычного запроса GROUP BY:

SELECT Country,ShipperName, SUM (ExtendedPrice) FROM invoices1

GROUP BY COUNTRY,ShipperName

Отметим, однако, что результатом этого запроса будет не сама сводная таблица, а лишь набор агрегатных данных для ее построения (ниже показан фрагмент):

|  |  |  |
| --- | --- | --- |
| Country | ShipperName | SUM (ExtendedPrice) |
| Argentina | Federal Shipping | 845.5 |
| Austria | Federal Shipping | 35696.78 |
| Belgium | Federal Shipping | 8747.3 |
| Brazil | Federal Shipping | 13998.26 |
| … | … | … |

Третий из рассмотренных выше запросов имеет уже три параметра в условии WHERE. Варьируя их, мы получим трехмерный набор данных (рис. 2).

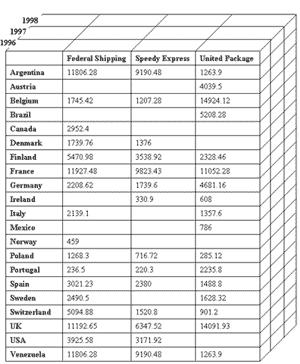


Рис. 2. Трехмерный набор агрегатных данных

Ячейки куба, показанного на рис. 2, содержат агрегатные данные, соответствующие находящимся на осях куба значениям параметров запроса в предложении WHERE.

Можно получить набор двухмерных таблиц с помощью сечения куба плоскостями, параллельными его граням (для их обозначения используют термины cross-sections и slices).

Очевидно, что данные, содержащиеся в ячейках куба, можно получить и с помощью соответствующего запроса с предложением GROUP BY. Кроме того, некоторые электронные таблицы (в частности, Microsoft Excel 2000) также позволяют построить трехмерный набор данных и просматривать различные сечения куба, параллельные его грани, изображенной на листе рабочей книги (workbook).

Если в предложении WHERE содержится четыре или более параметров, результирующий набор значений (также называемый OLAP-кубом) может быть 4-мерным, 5-мерным и т.д.

Рассмотрев, что представляют собой многомерные OLAP-кубы, перейдем к некоторым ключевым терминам и понятиям, используемым при многомерном анализе данных.

## Некоторые термины и понятия

Наряду с суммами в ячейках OLAP-куба могут содержаться результаты выполнения иных агрегатных функций языка SQL, таких как MIN, MAX, AVG, COUNT, а в некоторых случаях — и других (дисперсии, среднеквадратичного отклонения и т.д.). Для описания значений данных в ячейках используется термин summary (в общем случае в одном кубе их может быть несколько), для обозначения исходных данных, на основе которых они вычисляются, — термин measure, а для обозначения параметров запросов — термин dimension (переводимый на русский язык обычно как "измерение", когда речь идет об OLAP-кубах, и как "размерность", когда речь идет о хранилищах данных). Значения, откладываемые на осях, называются членами измерений (members).

Говоря об измерениях, следует упомянуть о том, что значения, наносимые на оси, могут иметь различные уровни детализации. Например, нас может интересовать суммарная стоимость заказов, сделанных клиентами в разных странах, либо суммарная стоимость заказов, сделанных иногородними клиентами или даже отдельными клиентами. Естественно, результирующий набор агрегатных данных во втором и третьем случаях будет более детальным, чем в первом. Заметим, что возможность получения агрегатных данных с различной степенью детализации соответствует одному из требований, предъявляемых к хранилищам данных, — требованию доступности различных срезов данных для сравнения и анализа.

Поскольку в рассмотренном примере в общем случае в каждой стране может быть несколько городов, а в городе — несколько клиентов, можно говорить об иерархиях значений в измерениях. В этом случае на первом уровне иерархии располагаются страны, на втором — города, а на третьем — клиенты рис. 3.

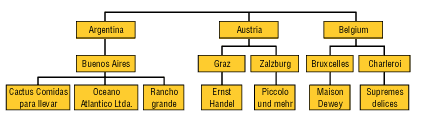


Рис. 3. Иерархия в измерении, связанном с географическим положением клиентов

Отметим, что иерархии могут быть сбалансированными (balanced), как, например, иерархия, представленная на рис. 3, а также иерархии, основанные на данных типа "дата—время", и несбалансированными (unbalanced). Типичный пример несбалансированной иерархии — иерархия типа "начальник—подчиненный" (ее можно построить, например, используя значения поля Salesperson исходного набора данных из рассмотренного выше примера), представлен на рис. 4.

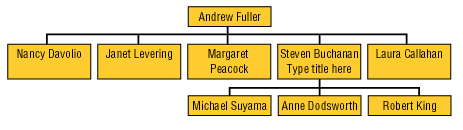


Рис. 4. Несбалансированная иерархия

Иногда для таких иерархий используется термин Parent-child hierarchy.

Существуют также иерархии, занимающие промежуточное положение между сбалансированными и несбалансированными (они обозначаются термином ragged — "неровный"). Обычно они содержат такие члены, логические "родители" которых находятся не на непосредственно вышестоящем уровне (например, в географической иерархии есть уровни Country, City и State, но при этом в наборе данных имеются страны, не имеющие штатов или регионов между уровнями Country и City; (рис. 5).

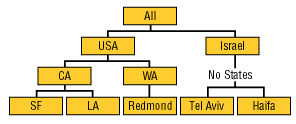


Рис. 5. "Неровная" иерархия

Отметим, что несбалансированные и "неровные" иерархии поддерживаются далеко не всеми OLAP-средствами. Например, в Microsoft Analysis Services 2000 поддерживаются оба типа иерархии, а в Microsoft OLAP Services 7.0 — только сбалансированные. Различным в разных OLAP-средствах может быть и число уровней иерархии, и максимально допустимое число членов одного уровня, и максимально возможное число самих измерений.

## Заключение

В данной статье мы ознакомились с основами OLAP. Мы узнали следующее:

* Назначение хранилищ данных — предоставление пользователям информации для статистического анализа и принятия управленческих решений.
* Хранилища данных должны обеспечивать высокую скорость получения данных, возможность получения и сравнения так называемых срезов данных, а также непротиворечивость, полноту и достоверность данных.
* OLAP (On-Line Analytical Processing) является ключевым компонентом построения и применения хранилищ данных. Эта технология основана на построении многомерных наборов данных — OLAP-кубов, оси которого содержат параметры, а ячейки — зависящие от них агрегатные данные.
* Приложения с OLAP-функциональностью должны предоставлять пользователю результаты анализа за приемлемое время, осуществлять логический и статистический анализ, поддерживать многопользовательский доступ к данным, осуществлять многомерное концептуальное представление данных и иметь возможность обращаться к любой нужной информации.

Кроме того, мы рассмотрели основные принципы логической организации OLAP-кубов, а также узнали основные термины и понятия, применяемые при многомерном анализе. И наконец, мы выяснили, что представляют собой различные типы иерархий в измерениях OLAP-кубов.

В следующей статье данного цикла мы рассмотрим типичную структуру хранилищ данных, поговорим о том, что представляет собой клиентский и серверный OLAP, а также остановимся на некоторых технических аспектах многомерного хранения данных.