



Министерство образования Республики Беларусь

Учреждение образования
«Гомельский государственный технический
университет имени П. О. Сухого»

Кафедра «Информационные технологии»

МНОГОМЕРНЫЙ АНАЛИЗ ДАННЫХ

ПОСОБИЕ

**по выполнению контрольных
и лабораторных работ по дисциплине
«Автоматизация принятия
управленческих решений»**

**для студентов экономических специальностей
дневной и заочной форм обучения**

Электронный аналог печатного издания

Гомель 2007

УДК 004.43(075.8)
ББК 32.973-018.1я73
М73

*Рекомендовано к изданию научно-методическим советом
факультета автоматизированных и информационных систем
ГГТУ им. П. О. Сухого
(протокол № 10 от 26.06.2006 г.)*

Авторы-составители: *Н. В. Водополова, Г. П. Косинов*

Рецензент: начальник сектора автоматизированных систем управления
вычислительного центра ГГТУ им. П. О. Сухого
Н. С. Шестакова

Многомерный анализ данных : пособие по выполнению контрол. и лаб. работ по
М73 дисциплине «Автоматизация принятия управленческих решений» для студентов экон.
специальностей днев. и заоч. форм обучения / авт.-сост.: Н. В. Водополова, Г. П. Коси-
нов. – Гомель : ГГТУ им. П. О. Сухого, 2007. – 20 с.– Систем. требования: PC не ниже
Intel Celeron 300 МГц ; 32 Mb RAM ; свободное место на HDD 16 Mb ; Windows 98 и вы-
ше ; Adobe Acrobat Reader. – Режим доступа: <http://gstu.local/lib>. – Загл. с титул. экрана.

ISBN 978-985-420-636-3.

Рассмотрены вопросы технологии многомерной обработки данных OLAP. Изложены ос-
новные положения концепции многомерности представления, хранения и обработки данных:
OLAP-кубы, хранилища данных, источники данных, Сервер, Клиент, а также архитектура
OLAP-систем. Подробно рассмотрены отличительные особенности организации новых техно-
логий обработки данных. Весь теоретический материал проиллюстрирован примерами, табли-
цами и рисунками.

Для студентов экономических специальностей дневной и заочной форм
обучения.

УДК 004.43(075.8)
ББК 32.973-018.1я73

ISBN 978-985-420-636-3

© Водополова Н. В., Косинов Г. П.,
составление, 2007

© Учреждение образования

«Гомельский государственный технический
университет имени П. О. Сухого», 2007

ВВЕДЕНИЕ В OLAP-ТЕХНОЛОГИЮ

Успешность любого объекта хозяйствования напрямую связана с эффективностью принимаемых решений. Уровень обоснованности и качество любого управленческого решения зависят от количества обрабатываемой информации. Очевидно, что чем больше объем данных, на основании анализа которых принято то или иное решение, тем оно успешнее. Однако увеличение объема данных и количества показателей, используемых для принятия управленческого решения, не только усложняет анализ, но и увеличивает время их обработки. Широкое внедрение информационных технологий позволило автоматизировать многие процессы, в том числе и принятие решений. Информационные системы, призванные содействовать принятию решений нередко так и называются – системы поддержки принятия решений (СППР).

Как правило, данные, используемые в СППР, хранятся в базах данных и файлах оперативных систем. Обработка таких данных при больших объемах и разнообразии хранимой информации на определенном этапе становится весьма затруднительной, а иногда и невозможной. Это объясняется различными причинами: разрозненностью данных, например, что не редкость, наличие в каждом отделе предприятия своей базы данных; хранением их в форматах различных СУБД и в разных «уголках» корпоративной сети. Даже при хранении всех данных на центральном сервере баз данных структура данных бывает настолько разнообразна и сложна, что невозможно сделать выборку данных, необходимую для принятия того или иного решения.

Данные информационных систем организованы в виде реляционных таблиц. Реляционные базы данных (БД) хранят данные в отдельных таблицах, которые обычно хорошо нормализованы. Эта структура удобна для операционных БД, но сложные многотабличные запросы в ней выполняются относительно медленно. Поэтому сложные аналитические запросы к оперативной информации тормозят текущую работу компании, надолго блокируя таблицы и захватывая ресурсы сервера. Кроме того, существует и верхний предел объема данных, который может содержаться в реляционной БД и позволяющий достаточно эффективно осуществлять анализ.

Вышеуказанные причины послужили толчком к разработке и переходу к новым технологиям организации, хранения и обработки данных. Технология комплексного многомерного анализа данных получила название OLAP (*On-Line Analytical Processing*) – аналитическая обработка в реальном времени. Следует особо подчеркнуть, что

OLAP – это не отдельно взятый программный продукт, не язык программирования и даже не конкретная технология. OLAP – совокупность концепций, принципов и требований, лежащих в основе программных продуктов, облегчающих доступ к данным и их анализ.

Впервые концепция OLAP была описана в 1993 г. известным исследователем БД и автором реляционной модели данных Эдгаром Коддом. В 1995 г. на основе требований, изложенных Коддом, был сформирован так называемый тест FASMI (*Fast Analysis of Shared Multidimensional Information* – быстрый анализ разделяемой многомерной информации), который включает следующие требования к приложениям для многомерного анализа:

- **Fast** (*Быстрый*) – предоставление пользователю результатов анализа за приемлемое время (обычно не более 5 с), пусть даже ценой менее детального анализа;

- **Analysis** (*Анализ*) – возможность осуществления любого логического и статистического анализа, характерного для данного приложения, и его сохранения в доступном для конечного пользователя виде;

- **Shared** (*Разделяемый*) – многопользовательский доступ к данным с поддержкой соответствующих механизмов блокировок и средств авторизованного доступа;

- **Multidimensional** (*Многомерный*) – многомерное концептуальное представление данных, включая полную поддержку для иерархий и множественных иерархий (это – ключевое требование OLAP);

- **Information** (*Информация*) – возможность обращаться к любой нужной информации независимо от ее объема и места хранения.

Основным в OLAP-технологии является концепция многомерности. Многомерность в OLAP-приложениях может быть трех уровней:

1. Многомерное представление данных – средства конечного пользователя, обеспечивающие многомерную визуализацию и манипулирование данными. Слой многомерного представления абстрагирован от физической структуры данных и воспринимает данные как многомерные.

2. Многомерное хранение – средства физической организации данных, обеспечивающие эффективное выполнение многомерных запросов.

3. Многомерная обработка – средство (язык) формулирования многомерных запросов (традиционный реляционный язык запросов

SQL в этом случае непригоден) и процессор, обеспечивающий обработку и выполнения такого запроса.

Первый и третий уровни в обязательном порядке присутствуют во всех OLAP-средствах. Второй уровень широко распространен, но не обязателен, т. к. данные для многомерного представления могут извлекаться и из обычных реляционных структур. Процессор многомерных запросов в этом случае транслирует их в SQL-запросы, которые выполняются реляционной СУБД.

1. КОНЦЕПЦИЯ МНОГОМЕРНОСТИ

1.1. МНОГОМЕРНОЕ ПРЕДСТАВЛЕНИЕ ДАННЫХ

OLAP предоставляет удобные быстродействующие средства доступа, просмотра и анализа деловой информации. В основе OLAP-технологии лежит возможность получения мгновенного снимка реляционной БД и структуризация ее в пространственную модель для выполнения многомерных запросов. При этом пользователь получает естественную, интуитивно понятную модель данных, которая организована в виде многомерных наборов данных.

Основные термины и понятия:

OLAP-куб (*Cub*) – многомерный набор данных конкретной предметной области.

Измерения куба (*Dimensions*) – оси многомерной системы координат. Измерениями являются основные атрибуты анализируемого бизнес-процесса. Например, для продаж это могут быть *Регион*, *Товар*, *Время*. Пример такого трехмерного куба изображен на рис. 1. Добавив измерение Тип покупателя, получим четырехмерный куб. Количество измерений куба зависит от решаемой задачи. Мощные OLAP-продукты рассчитаны на обработку кубов порядка 20, а простые приложения, как правило, поддерживают 6 измерений.

Члены измерения, или метки (*Members*) – значения, откладываемые на осях. Например, членами измерения «Регион» могут быть «Брест», «Гомель» и т. д.

Меры (*Measures*) – данные, количественно характеризующие процесс и располагающиеся в ячейках куба, т. е. на пересечении измерений или осей куба.

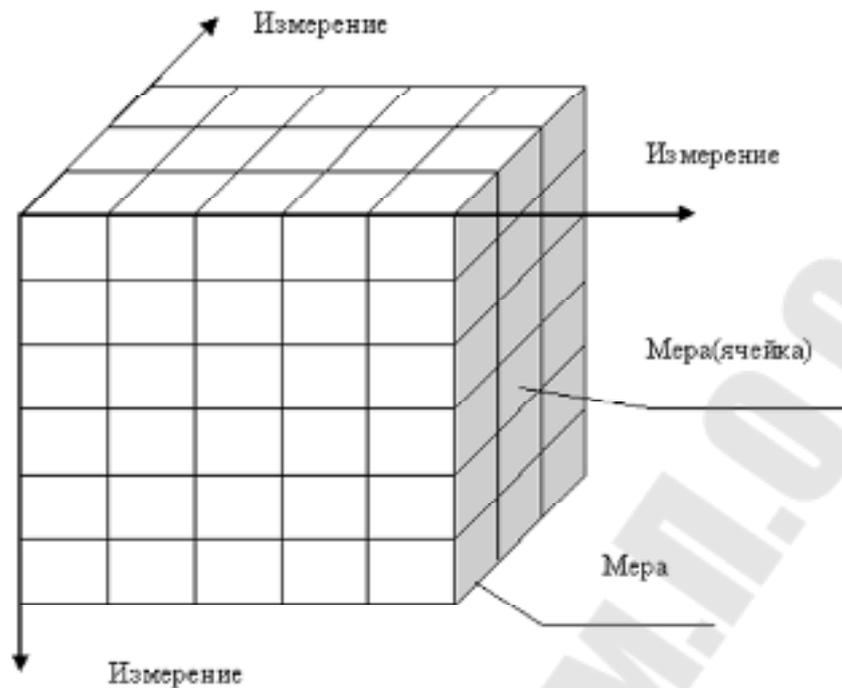


Рис. 1. Представление данных в виде гиперкуба

Над гиперкубом могут выполняться следующие операции:

1. Срез (Slice) (рис. 2) – формируется подмножество многомерного массива данных, соответствующее единственному значению одного или нескольких элементов измерений, не входящих в это подмножество. Если рассматривать термин «срез» с точки зрения конечного пользователя, то наиболее часто его роль играет двумерная проекция куба.

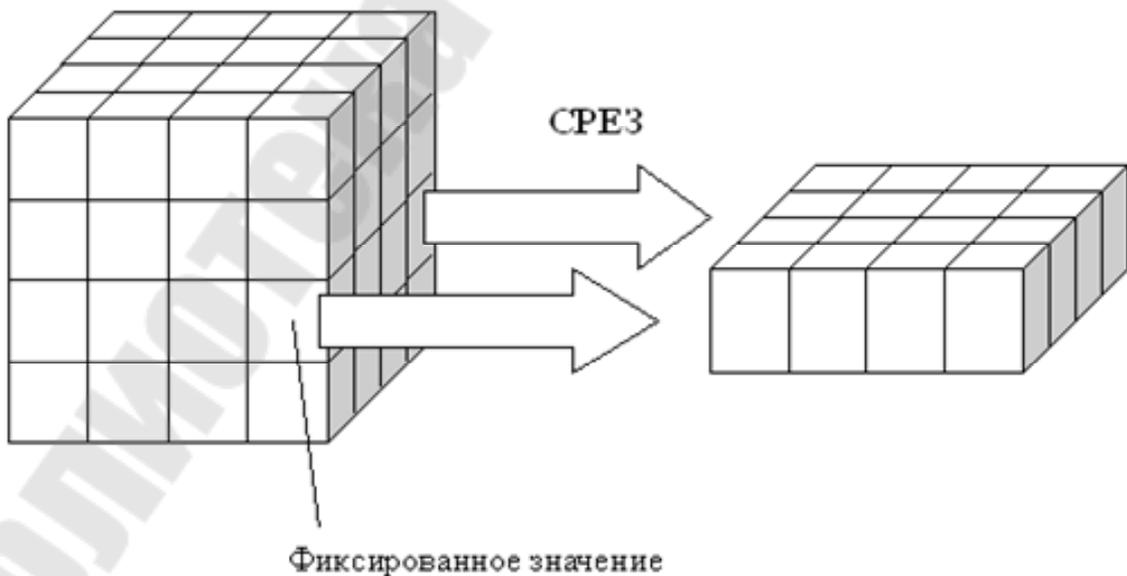


Рис. 2. Операция среза

2. Вращение (Rotate) (рис. 3) – изменение расположения измерений, представленных в отчете или на отображаемой странице. Например, операция вращения может заключаться в перестановке местами строк и столбцов таблицы или перемещения интересующих измерений в столбцы или строки создаваемого отчёта, что позволяет придать ему желаемый вид. Кроме того, вращением куба данных является перемещение внетабличных измерений на место измерений, представленных на отображаемой странице, и наоборот (при этом внетабличное измерение становится новым измерением строки или измерением столбца).

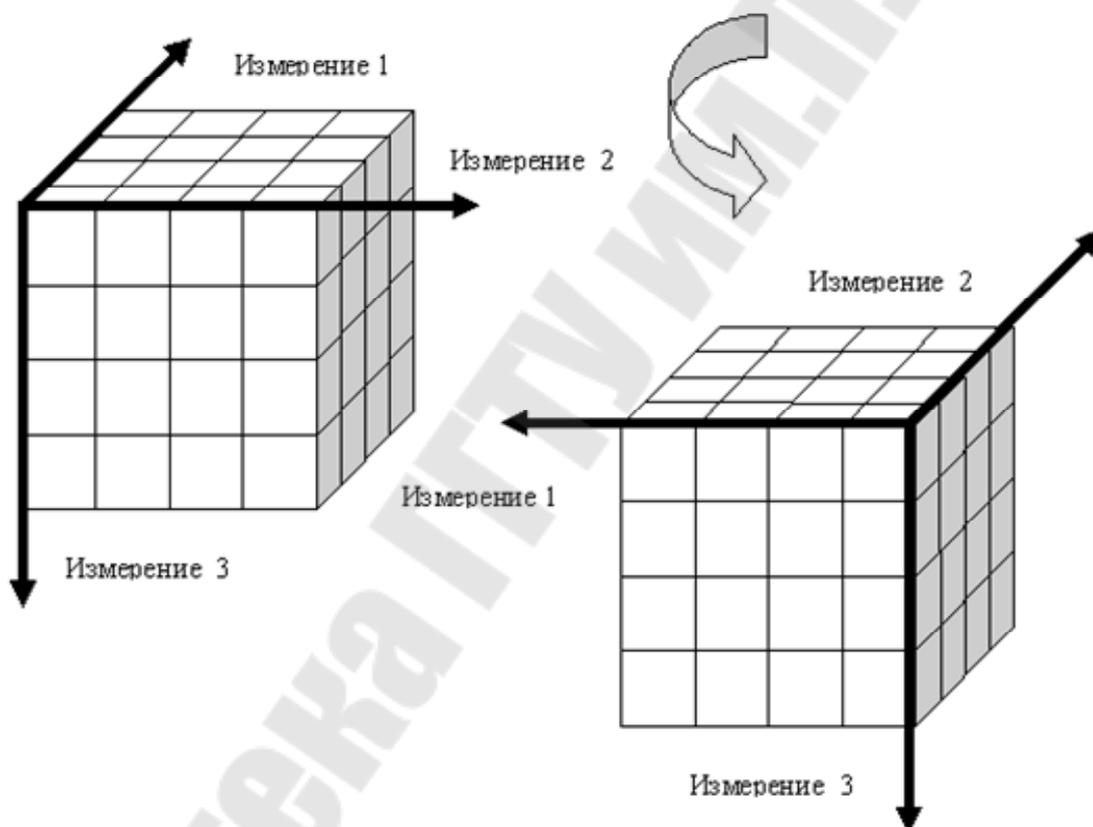


Рис. 3. Операция вращения

3. Консолидация (Drill Up) и детализация (Drill Down) (рис. 4.) – операции, которые определяют переход вверх по направлению от детального (down) представления данных к агрегированному (up) и наоборот, соответственно. Направление детализации (обобщения) может быть задано как по иерархии отдельных измерений, так и согласно прочим отношениям, установленным в рамках измерений или между измерениями.

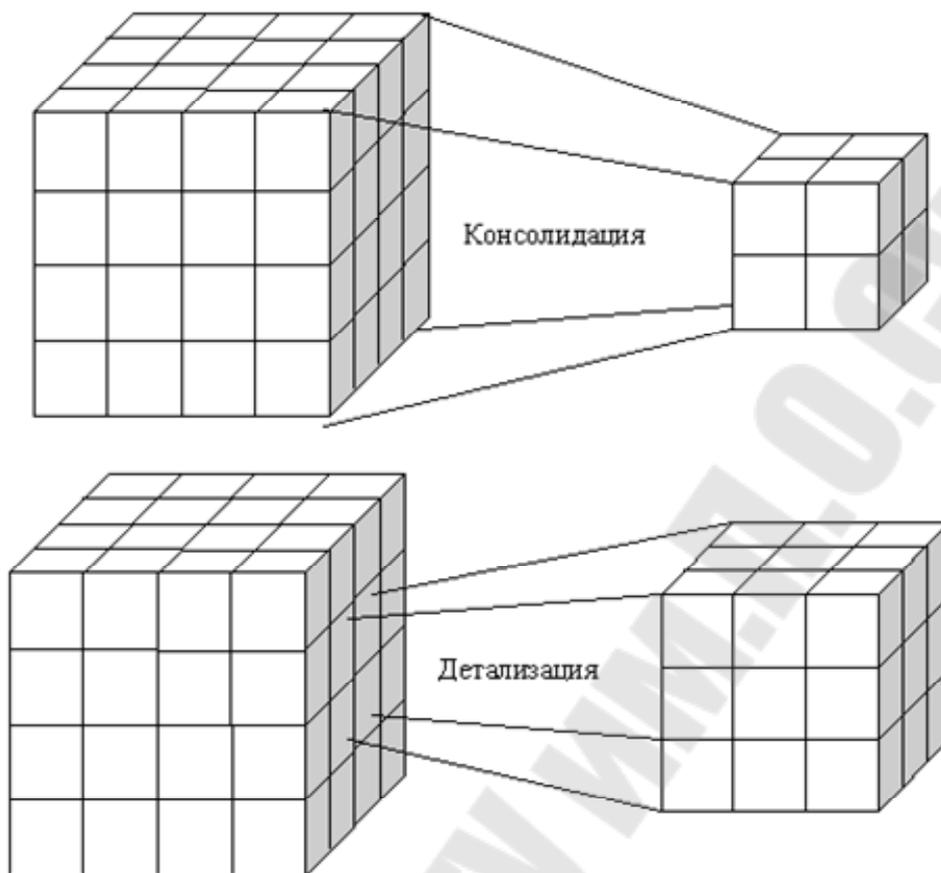


Рис. 4. Операции консолидации и детализации

Наряду с исходными данными в ячейках OLAP-куба могут находиться агрегатные данные, т. е. содержаться результаты выполнения агрегатных функций языка SQL (MIN, MAX, AVG, COUNT), а в некоторых случаях – и других (дисперсии, среднеквадратичного отклонения и т. д.). В кубе рис. 5 представлены суммарные данные.

	Март			
	Февраль			
	Январь			
	Брест	Гомель		
Напитки	25000	28000		
Колбасы	10000	13500		
Молочные	3500	5100		

Рис. 5. Пример OLAP-куба

Для визуализации данных, хранящихся в кубе, применяются привычные двухмерные таблицы, но имеющие сложные иерархические заголовки строк и столбцов. Данные куба рис. 5 могут быть представлены таблицей рис. 6.

	Январь		Февраль		Март	
	Брест	Гомель	Брест	Гомель	Брест	Гомель
Напитки						
Колбасы						
Молочные продукты						

Рис. 6. Пример визуализации данных трехмерного куба

Пользователь, анализирующий информацию, может «разрезать» куб по разным направлениям, получая сводные (например, по годам) или, наоборот, детальные (по декадам) сведения. Для «разрезания» куба используются метки. Например, «разрезав» куб рис. 1 по значению метки «Январь» измерения *Время*, получим обычную двухмерную таблицу рис. 3. «Разрезав» этот же куб по метке «Брест» измерения *Регион*, получим таблицу рис. 4.

Метки используются как для «разрезания» куба, так и для ограничения (фильтрации) выбираемых данных. В этом случае для анализа остается только необходимое подмножество значений, например, колбасы копченые, напитки безалкогольные и т. д.

	Брест	Гомель
Напитки		
Колбасы		
Молочные продукты		

Рис. 7

	Январь	Февраль	Март
Напитки			
Колбасы			
Молочные продукты			

Рис. 8

Метки могут объединяться в иерархии, состоящие из одного или нескольких уровней. Например, метки измерения *Регион*, естественно, объединяются в иерархию с такими уровнями:

Мир
 Страна
 Область
 Город

В соответствии с уровнями иерархии вычисляются агрегатные значения, например, объем продаж для Республики Беларусь (уровень **Страна**) или для Гомельской области (уровень **Область**). В одном измерении можно реализовать более одной иерархии. Например, для измерения *Регион* возможны {Мир, Страна, Область, Город} и {Мир, Континент, Страна} или {Область, Район, Город} и {Область, Город, Район города}.

1.2. МНОГОМЕРНОЕ ХРАНЕНИЕ ДАННЫХ

Как уже отмечалось, конечной целью использования OLAP является анализ данных и представление результатов этого анализа в виде, удобном для восприятия и принятия решений. Основная идея OLAP заключается в построении многомерных кубов, которые будут доступны для пользовательских запросов. Однако исходные данные для построения OLAP-кубов обычно хранятся в реляционных базах данных. Нередко это специализированные реляционные базы данных, называемые также хранилищами данных.

Хранилище данных (Data Warehouse) – это предметно-ориентированное, привязанное ко времени и неизменяемое собрание данных для поддержки процесса принятия управляющих решений.

Следует понимать, что хранилища данных – это не гигантское скопление данных, а организация данных, позволяющая производить быструю обработку сложных запросов, и в силу этого должна отвечать следующим требованиям:

- 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 (процедур) просмотра данных в хранилище;
- полнота и достоверность хранимых данных;
- поддержка качественного процесса пополнения данных.

Типичное хранилище данных, как правило, отличается от обычной реляционной базы данных:

1. 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.

2. В обычных базах данных хранятся исходные данные. В хранилищах данных наряду с детальными данными, извлекаемыми из оперативных систем, хранятся и суммарные показатели (агрегированные данные, агрегаты), такие, как суммы объемов продаж по месяцам, по категориям товаров и т. п. Агрегаты хранятся в явном виде с единственной целью – ускорить выполнение запросов. Это связано с тем, что в хранилище накапливается, как правило, очень большой объем данных, но аналитиков в большинстве случаев интересуют не детальные данные, а обобщенные показатели. И если каждый раз для вычисления суммы продаж за год пришлось бы суммировать миллионы индивидуальных продаж, скорость обработки, скорее всего, была бы неприемлемой. Поэтому при загрузке данных в хранилище вычисляются и сохраняются все суммарные показатели или их часть.

3. 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.

4. Обычные базы данных чаще всего являются источником данных, попадающих в хранилище. Кроме того, хранилище может пополняться за счет внешних источников, например статистических отчетов.

5. При организации работы с обычной базой все операции по созданию и поддержке данных из БД, а также по извлечению информации осуществляются одним и тем же программным продуктом, например, СУБД Access, табличным процессором Excel и т. д. Для реализации хранилищ данных обычно используется несколько продуктов, одни из которых представляют собой собственно средства хранения данных, другие – средства их извлечения и просмотра, третьи – средства их пополнения и т. д.

Основными составляющими структуры хранилищ данных являются таблица фактов и таблицы измерений.

Таблица фактов (*fact table*) – это основная таблица хранилища данных, содержит сведения об объектах или событиях, совокупность которых будет в дальнейшем анализироваться. Различают четыре типа фактов:

1) факты, связанные с транзакциями (*transaction facts*), т. е. с отдельными событиями (снятие денег со счета с помощью банкомата, факт покупки конкретного товара);

2) факты, связанные с «моментальными снимками» (*snapshot facts*) состояния объекта в определенные моменты времени (наличие денег банковского счета на конец дня или месяца, объем продаж за день);

3) факты, связанные с элементами документа (*line-item facts*) и содержат подробную информацию об элементах этого документа (элементами счета за товар могут быть наименование товара, количество, цена, процент скидки);

4) факты, связанные с возникновением события или состояния объекта (*event or state facts*) без подробностей о нем (факт продажи конкретного товара или отсутствие такового без каких-либо подробностей).

Очевидно, что для многомерного анализа пригодны только такие таблицы фактов, которые содержат как можно более подробные данные. Таковыми являются данные, соответствующие членам самых нижних уровней иерархии измерений. То есть предпочтительнее взять за основу факты продажи товаров отдельным заказчикам, а не суммы продаж для разных стран – последние все равно будут вычислены OLAP-средством.

Следует особо подчеркнуть, что в таблице фактов не хранятся сведения о том, как группировать записи при вычислении агрегатных данных. Например, если таблица фактов содержит идентификаторы продуктов или клиентов, то информация о том, к какой категории относится данный продукт или в каком городе находится данный клиент, отсутствует. Эти сведения, в дальнейшем используемые для построения иерархий в измерениях куба, содержатся в таблицах измерений.

Таблицы измерений (*dimension tables*) служат для хранения редко изменяемых данных. Как правило, одна запись таблицы измерений соответствует каждому члену нижнего уровня иерархии в измерении. То есть каждая таблица измерений должна находиться с таблицей фактов в отношении «один ко многим».

Отдельная запись таблицы измерений, как минимум, может содержать одно описательное поле (обычно с именем члена измерения) и, как правило, целочисленное ключевое поле для однозначной идентификации члена измерения.

Если измерение, описанное в таблице измерений, является элементом иерархии, то таблица измерений обязательно должна содержать поля, указывающие на «родителя» данного члена иерархии. Иногда таблица измерений может содержать и поля, указывающие на «прародителей», и иных «предков» в данной иерархии.

Таким образом, можно сказать, что таблицы измерений содержат справочную информацию для данных, хранящихся в таблицах фактов. Такие данные являются неизменяемыми либо редко изменяемыми. Поэтому скорость роста таблиц измерений по сравнению со скоростью роста таблиц фактов является незначительной. Например, добавление новой записи в таблицу измерений, характеризующую товары, производится только при появлении нового товара, не продававшегося ранее.

Для связи данных таблицы фактов и таблиц измерений предназначены ключевые поля. Если ключевые поля таблиц измерений являются простыми (первичными) и используются только для идентификации конкретного члена измерения, то таблица фактов, как правило, содержит составной ключ, объединяющий первичные ключи таблиц измерений. Чаще всего это целочисленные значения или значения типа «дата/время».

OLAP-куб создается из соединения таблиц измерений и таблицы фактов в соответствии с определенной схемой. В центре любой схемы находится таблица фактов, содержащая факты, по которым делаются запросы. К таблице фактов присоединяются таблицы измерений. Если каждое измерение содержится в одной таблице, такая схема хранилища данных носит название «Звезда». На рис. 9 изображен пример такой схемы для базы данных Nortwind, входящая в комплект поставки Microsoft Server и Microsoft Access.

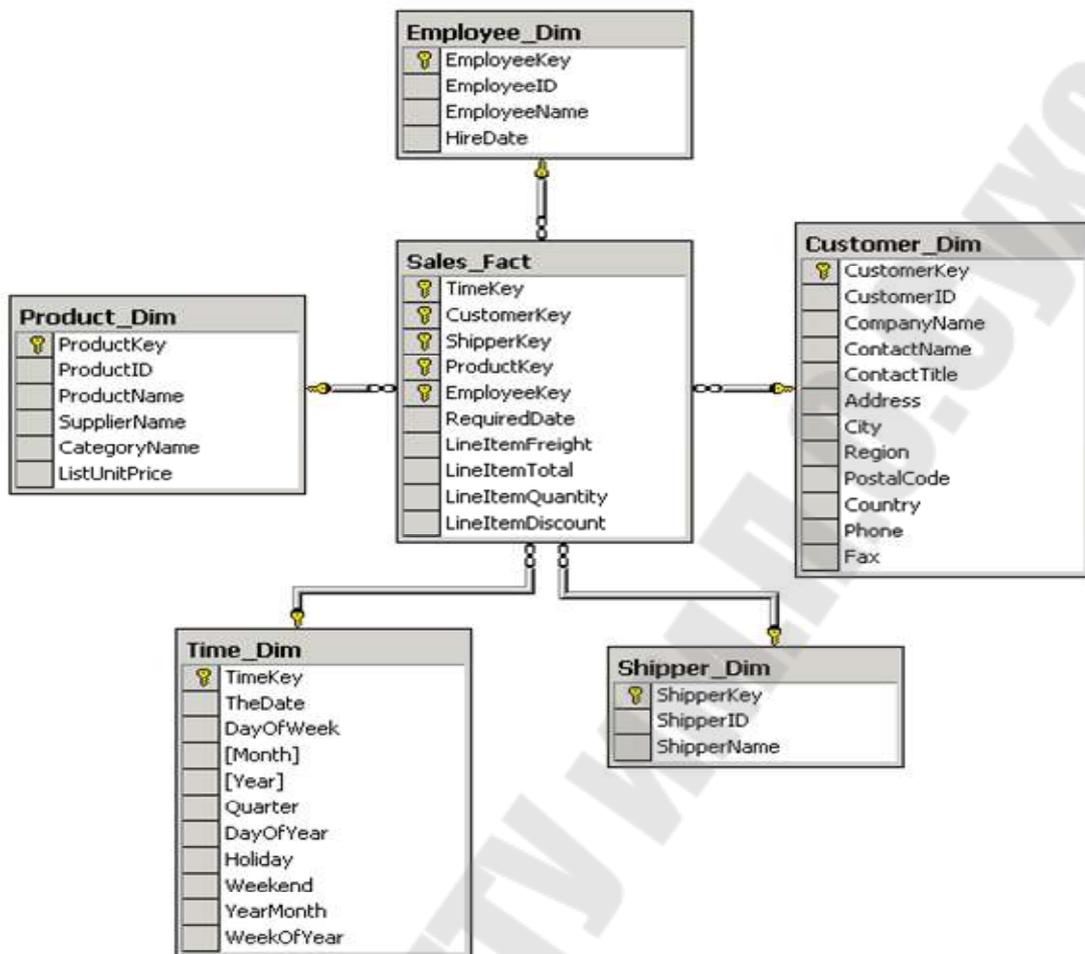


Рис. 9. Пример схемы «Звезда» для БД Nortwind

Таблица фактов **Sales_Fact** (сведения о продажах товаров) с помощью пятиэлементного составного ключа связана с пятью таблицами измерений:

- 1) **Product_Dim** – сведения о товаре;
- 2) **Employee_Dim** – сведения о служащих;
- 3) **Customer_Dim** – сведения о покупателях;
- 4) **Time_Dim** – даты продаж;
- 5) **Shipper_Dim** – сведения о грузчиках.

Если же хотя бы одно измерение содержится в нескольких таблицах, такая схема хранилища данных носит название «Снежинка» (рис. 10). В данном случае сведения о товарах хранятся в двух таблицах **Product2_Dim** и **ProductCategory_Dim**, т. е. для измерения **Product** добавляется еще один уровень иерархии: Категория товара.

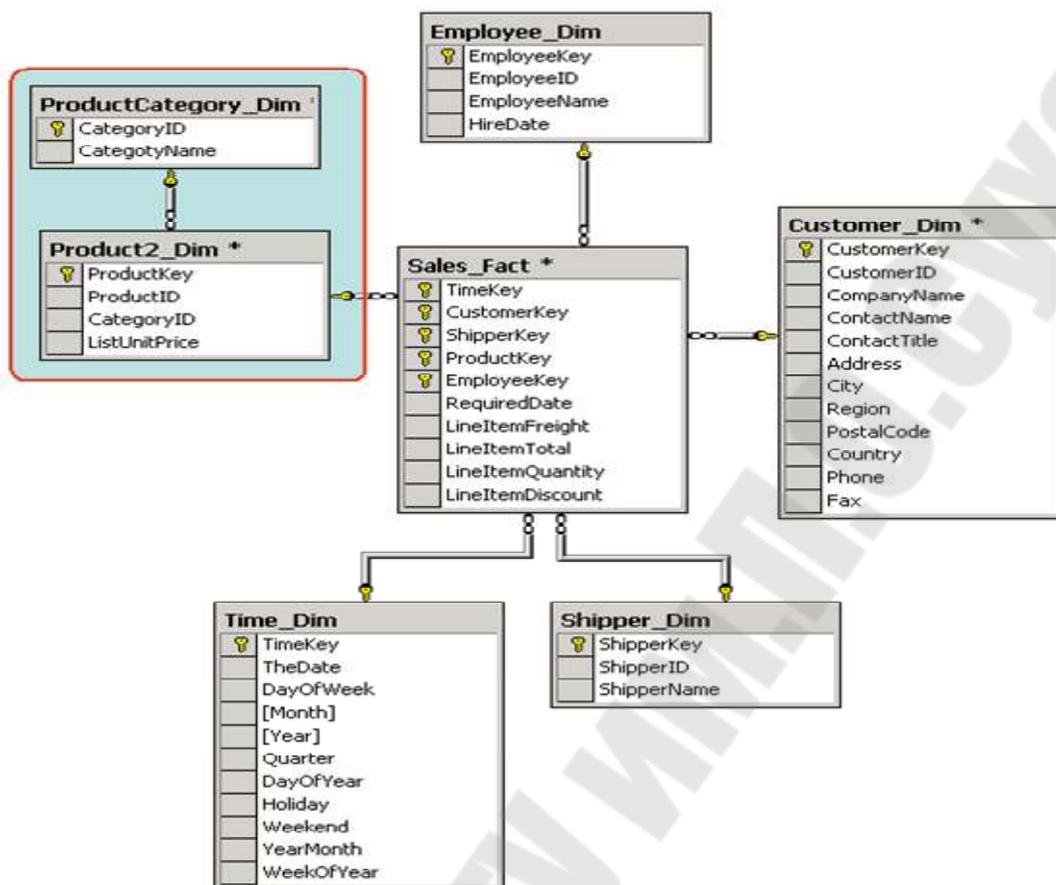


Рис. 10. Пример схемы «Снежинка» БД Nortwind

1.3. МНОГОМЕРНАЯ ОБРАБОТКА ДАННЫХ

Стратегия многомерного анализа данных определяется структурой OLAP-системы.

OLAP-система состоит из множества компонент. Основными же являются источники данных, OLAP-сервер и клиент.

Источник данных – это поставщик данных для анализа. В зависимости от области использования OLAP-продукта источником может служить:

- хранилище данных;
- наследуемая БД, содержащая общие данные;
- набор таблиц, объединяющих данные;
- любая комбинация перечисленных источников данных.

Данные из источника переносятся или копируются на OLAP-сервер, где они систематизируются и подготавливаются для более быстрого формирования ответов на запросы.

Сервер – это ядро, прикладная часть OLAP-системы, которая выполняет всю работу (в зависимости от модели системы), и хранит в себе всю информацию, к которой обеспечивается активный доступ. Поскольку как исходные, так и агрегатные данные могут храниться либо в реляционной БД, либо в многомерных структурах, в зависимости от выбранной концепции хранения данных существует несколько вариантов архитектуры OLAP-сервера:

- **MOLAP** (Multidimensional OLAP) – и детальные, и агрегатные данные хранятся в многомерной БД;

- **ROLAP** (Relation OLAP) – исходные данные хранятся в той же реляционной БД, где они изначально находились, а агрегатные данные помещаются в специально созданные для их хранения служебные таблицы, размещаемые в той же БД;

- **HOLAP** (Hybrid OLAP) – детальные данные хранятся в реляционной БД, а агрегатные данные хранятся в многомерной БД.

Большинство современных серверных OLAP-средств поддерживают все три способа хранения данных. Выбор способа хранения зависит от объема и структуры исходных данных, требований к скорости выполнения запросов и частоты обновления OLAP-кубов.

Клиент – это пользовательский интерфейс к OLAP-серверу, т. е. реализованная в виде специального приложения или функции программного продукта, возможность представления и манипуляции данными в БД.

Несмотря на то, что *Сервер* – это основа OLAP-решения, *Клиент* не менее важен. Если *Клиент* сложен и малофункционален, пользователь не сможет воспользоваться всеми преимуществами мощного *Сервера*. Чаще всего, при разработке *Клиентов* используется стандартный, привычный взгляд на интерфейс с заранее определенными функциями и структурой. Кроме того, *Клиент* позволяет быстро решать более или менее стандартные задачи. Например, популярны финансовые пакеты, которые позволяют специалистам использовать привычные финансовые инструменты без необходимости проектировать структуру БД или общепринятые формы и отчеты.

Основу *Клиента* составляет инструмент запросов или генератор отчетов. Они имеют простой в использовании графический интерфейс и позволяют пользователям создавать отчеты перемещением объектов в отчет методом «*drag and drop*». Генераторы отчетов, поддерживающие OLAP, имеют функции не только формирования и форматирования отчетов (такие услуги предоставляют традиционные генераторы

отчетов), но возможности углубления в данные до уровня подробностей, вращения (*пивотинг*) отчетов, поддержки иерархий, *Add-Ins* (дополнения) электронных таблиц. Благодаря этим функциям у пользователя появляется возможность создать единственный отчет или диаграмму, а затем подвергать их различным манипуляциям с целью более детального анализа, не обременяя себя созданием всех возможных вариантов отчета или диаграммы.

2. АРХИТЕКТУРА OLAP-СИСТЕМ

Все компоненты OLAP-систем поддерживаются и объединяются в единую систему различными программными продуктами: OLAP-средствами, которые условно делятся на клиентские и серверные.

В *клиентских OLAP-средствах* все элементы OLAP-системы реализованы в одном приложении, т. е.:

– они самостоятельно выполняют все функции многомерного анализа: агрегирование данных (суммирование, вычисление средних величин, нахождение максимальных и минимальных значений), а также их отображение;

– агрегированные данные содержатся внутри адресного пространства такого OLAP-средства, что накладывает ограничение на количество числа измерений (обычно не более шести) и разнообразие значений этих параметров.

Некоторые клиентские OLAP-средства поддерживают возможность сохранения агрегатных данных в виде файла, что позволяет не производить их повторное вычисление и сэкономить память для других расчетов. Развитием этой идеи стала разработка серверных OLAP-средств.

Серверные OLAP-средства осуществляют поддержку хранилища данных, хранение и изменение агрегатных данных с помощью отдельного приложения или процесса, называемого *OLAP-сервером*. Благодаря этому серверные OLAP-средства позволяют обрабатывать большие объемы информации за незначительное время.

В последнее время все чаще используется идея применения серверных OLAP-средств в качестве промежуточного звена между хранилищем данных в виде реляционной БД и клиентским приложением. В этом случае основная задача OLAP-сервера – превращение данных из реляционного хранилища в форму, более удобную для соз-

дания аналитических отчетов, – в OLAP-кубы. Это позволяет значительно снизить время выполнения запросов и требования к ресурсам, потребляемым клиентским приложением. Примером такого клиентского средства является Microsoft Excel 2000.

Microsoft Excel 2000 предоставляет возможность создания и хранения в виде файла небольшого локального многомерного OLAP-куба, а также отображения его двух- или трехмерных сечений. Кроме того, он позволяет обращаться к серверным OLAP-хранилищам. OLAP-сервер фирмы Microsoft входит в комплект поставки Microsoft SQL Server 2000 Enterprise Edition и на сегодняшний день признан аналитиками Gartner Group одним из наиболее популярных продуктов этого класса.

ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ

1. Основные положения технологии комплексного многомерного анализа данных.
2. Концепция многомерности обработки данных.
3. Многомерное представление данных. Основные понятия.
4. Многомерное хранение данных. Хранилища данных.
5. Отличия организации хранилищ данных от реляционных баз данных.
6. Структурные компоненты и организация хранилищ данных.
7. Многомерная обработка данных.
8. Архитектура OLAP-систем.

ЛИТЕРАТУРА

1. Барсегян, А. Методы и модели анализа данных: OLAP и Data Mining / А. Барсегян [и др.]. – Санкт-Петербург : BHV, 2004.
2. Карабутов, Н. Создание интегрированных документов в Microsoft Office. Введение в анализ данных и подготовку документов / Н. Карабутов. – Москва : Солон, 2005.
3. Федоров, А. Введение в OLAP-технологии Microsoft / А. Федоров. – Москва : Диалог-МИФИ, 2002.
4. Федоров, А. Введение в OLAP / А. Федоров, Н. Елманова. – Москва : КомпьютерПресс 4, 2001.

СОДЕРЖАНИЕ

Введение в OLAP-технологию	3
1. Концепция многомерности	5
1.1. Многомерное представление данных	5
1.2. Многомерное хранение данных	10
1.3. Многомерная обработка данных	15
2. Архитектура OLAP-систем	17
Вопросы для самоконтроля	18
Литература	19

Учебное электронное издание комбинированного распространения

Учебное издание

МНОГОМЕРНЫЙ АНАЛИЗ ДАННЫХ
Пособие
по выполнению контрольных
и лабораторных работ по дисциплине
«Автоматизация принятия
управленческих решений»
для студентов экономических специальностей
дневной и заочной форм обучения

Авторы-составители: **Водополова** Наталия Виталиевна
Косинов Геннадий Петрович

Редактор
Компьютерная верстка

Н. В. Гладкова
Н. В. Широглазова

Подписано в печать 22.10.07.
Формат 60x84/16. Бумага офсетная. Гарнитура Таймс.
Цифровая печать. Усл. печ. л. 1,39. Уч.-изд. л. 1,1.
Изд. № 84.

E-mail: ic@gstu.gomel.by
<http://www.gstu.gomel.by>

Издатель и полиграфическое исполнение:
Издательский центр учреждения образования
«Гомельский государственный технический
университет имени П. О. Сухого».
ЛИ № 02330/0131916 от 30.04.2004 г.
246746, г. Гомель, пр. Октября, 48.