|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ  Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА − Российский технологический университет»**  **РТУ МИРЭА** |
|  |
| Институт Информационных технологий  *наименование института(полностью)* |
| Кафедра Инструментального и прикладного программного обеспечения  *наименование кафедры (полностью)* |

|  |
| --- |
| **УТВЕРЖДАЮ** |
| зав. кафедрой ИППО |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_/В.А. Мордвинов/ |
| «\_\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_ |

**КОНСПЕКТ ЛЕКЦИЙ ПО ДИСЦИПЛИНЕ**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Хранилища данных и OLAP-технологии** | | | | | | |
| *(наименование дисциплины)* | | | | | | |
| Направление подготовки | | | | | **09.03.04 «Программная инженерия»** | |
|  | | | | | *(код и наименование)* | |
|  |  | | | | | |
| Профиль | **«Интеллектуальные программные системы и комплексы»** | | | | | |
|  | *(код и наименование)* | | | | | |
|  | |  | | | | |
| Форма обучения | | | **очная** | | | |
|  | | | *(очная, очно-заочная, заочная)* | | | |
|  | | | |  | | |
| Программа подготовки | | | | **академический** | | |
|  | | | | *(академический, прикладной бакалавриат)* | | |
|  | | | | | |  |
| Квалификация выпускника | | | | | | **Бакалавр** |
|  | | | | | |  |

Содержание

[Содержание 2](#_Toc528230356)

[Тема 1.Введение. Системы поддержки принятия решений 3](#_Toc528230357)

[1.1 Введение 3](#_Toc528230358)

[1.2 Системы поддержки принятия решений и основные решаемые с их помощью задачи 5](#_Toc528230359)

[1.3 OLTP-системы (Системы оперативной обработки транзакций) 8](#_Toc528230360)

[Тема 2. Хранилища данных 11](#_Toc528230361)

[2.1 Общая информация о хранилищах данных 11](#_Toc528230362)

[2.2 Сравнение OLTP-систем и храниилищ данных 13](#_Toc528230363)

[Тема 3. Архитектура хранилищ данных 15](#_Toc528230364)

[3.1 Архитектура и основные компоненты ХД 15](#_Toc528230365)

[3.2 Информационные потоки в ХД 20](#_Toc528230366)

[3.3 Магазины данных 24](#_Toc528230367)

[3.4 Предпосылки для создания магазинов данных 27](#_Toc528230368)

[3.5 Характеристики магазинов данных 27](#_Toc528230369)

[3.6 Витрины данных 30](#_Toc528230370)

[Тема 4. OLAP-системы 34](#_Toc528230371)

[4.1. Правила Кодда для OLAP-систем 34](#_Toc528230372)

[4.2. Многомерная OLAP-технология 36](#_Toc528230373)

[Многомерное пространство данных (МПД) 40](#_Toc528230374)

[Атрибуты измерений 40](#_Toc528230375)

[4.3. Категориии нструментов OLAP –технологий 41](#_Toc528230376)

[4.4. Схемы для реализации многомерного представления данных 46](#_Toc528230377)

[4.5. Управляемая среда запросов МОЕ 50](#_Toc528230378)

[Список литературы 51](#_Toc528230379)

# Тема 1.Введение. Системы поддержки принятия решений

1.1 Введение

В информационной области наступил этап, когда начали использовать Хранилища Данных и OLAP-технологии. OLAP (On-Line Analytical Processing) — технологии комплексного многомерного анализа данных. Концепция OLAP была описана в 1993 году Эдгаром Коддом, известным исследователем баз данных и автором реляционной модели данных. В OLAP-технологии используются агрегатные данные, которые образуют многомерный набор данных, называемый OLAP-куб. Благодаря такой модели данных пользователи могут формулировать сложные запросы, генерировать отчеты, получать подмножества данных.

В настоящее время современные вычислительные системы и компьютерные сети позволяют накапливать большие массивы данных для решения задач обработки и анализа.

Еще лет пять назад мало, кто слышал об этих технологиях. Сегодня хранилища данных и OLAP становятся неотъемлемой частью современных корпоративных систем поддержки принятия решений. Это одно из наиболее динамично развивающихся направлений индустрии создания программного обеспечения.

Концепция информационных хранилищ, зародилась в 80-х годах в недрах IBM. Идея хранилищ данных обязана своим развитием многим людям. Хотя эту идею предвосхищали в своих работах многие исследователи, можно смело утверждать, что **первой публикацией**, посвященной именно хранилищам данных, была статья Девлина (Devlin) и Мэрфи (Murphy) , вышедшая в 1988 году. В 1992 году Уильям Г.Инмон (William H. Inmon), который был техническим директором компании Prism и написал монументальную монографию «Building the Data Warehouse» («Построение хранилищ данных»), в которой дал **определение** хранилища данных:

Эффективное управление крупным и средним бизнесом сегодня немыслимо без применения передовых информационных технологий — систем поддержки принятия решений (СППР).

Процесс управления сводится к решению 3 задач:

* Где мы находимся?
* Куда мы хотим прийти?
* Как мы туда попадем?

Процесс управления — итерационный характер (принятие решения — применение управляющего воздействия — оценка состояния системы — оценка правильности выбранного решения — при наличии отклонений снова принятие решения).

Современные информационные технологии позволяют аналитику формулировать и решать следующие классы:

* Аналитические (вычисление заданных показателей и статистических характеристик).
* Визуализация данных
* Добыча знаний (datamining —проверка статистических гипотез, кластеризация, нахождение ассоциаций и временных шаблонов и т.п.)
* Имитационные (проведение на ЭВМ экспериментов на моделях, описывающих поведение сложных систем, например, в интервалы времени для анализа возможных последствий принятия того или иного решения)
* Синтез управления (для определения допустимых управляющих воздействий, обеспечивающих достижение заданной цели, оценка достижимости цели, определение множества возможных управляющих воздействий)
* Оптимизационные (интеграция имитационных, управленческих, оптимизационных и статистических методов моделирования и прогнозирования, выбор наиболее эффективного решения).

Однако в настоящее время нет информационных средств для решения всех задач в комплексе.

Бизнес — это сложный объект, который состоит из множества различных по свойствам подсистем, между которыми действует большое число разнородных связей. В кибернетике такие объекты получили название **сложных систем**, а методы их изучения — **системным анализом** (эта наука развивается с начала 40-х гг. в период 2-й мировой войны).

Общая с точки зрения теории познания триада имеет вид:

***Гипотеза — модель — решение.***

**Гипотеза** — это открытие, которое является новым положением, осуществляется на основе интуиции (из глубин человеческого подсознания, сформированного на основе личного опыта).

По гипотезе строится **модель** — формальное математическое описание — и находится решение. Полученное решение проверяется в эксперименте (отвергается или принимается). В результате получается знание, которым можно руководствоваться в практике.

**Проблемы** (в бизнесе):

1. динамичное изменение экономической ситуации, что мешает применять накопленный опыт, не успевает вырабатываться интуиция.
2. в условиях свободного рынка нет возможности проводить целенаправленные эксперименты.

В настоящее время актуальна разработка и использование комплексного ПО, реализующего задачи 1, 2 и 3-го классов. Сейчас стремительно развиваются OLAP- технологии.

Сейчас более 100 крупных производителей программ включились в конкуренцию.

**OLAP** — это инструменты оперативного анализа данных, содержащихся в хранилище, которые предназначены для общения аналитика с проблемой, а не с компьютером.

Целью OLAP-анализа является проверка возникающих гипотез. Аналитик выдвигает гипотезы, основываясь на своих знаниях и опыте. Технологии OLAP тесно связаны с технологиями построения Хранилищ данных (Data Warehouse) и методами их интеллектуальной обработки - Data Mining.

Термин Data Mining – добыча данных появился в 1996 году. Одним из основателей метода является Пятецкий-Шапиро. Идея метода Data Mining заключается в обнаружении знаний, содержащихся в **терабайтах** накопленной информации. Эти знания человек-аналитик не в состоянии исследовать самостоятельно. Data Mining относится к методам интеллектуального анализа данных с использованием хранилищ данных.

В 2008 г. был введен термин "Большие данные (Big Data)"Автором этого термина является Клиффорд Линчем, редактором журнала «Nature».С « big data» связывают задачи:

- создания хранилищ с использованием новых технологий для хранения

больших объёмов данных;

- управления;

- работы с неструктурированной информацией;

- разработки методов анализа big data.

1.2 Системы поддержки принятия решений и основные решаемые с их помощью задачи

В настоящее время информационные технологии находятся на следующем этапе:

1. Созданы для многих предприятий, технологических процессов информационные системы с использованием СУБД, локальных сетей, распределенных СУБД. Идет создание информационных систем для распределенных структур, фирм, располагающихся в различных городах или даже странах с помощью облачных технологий.

2. В созданных информационных системах идет непрерывный рост объема накопленных данных. в области информационных технологий в настоящее время акцент смещается в область аналитической обработки этих данных. В целом, человечество вошло в область создания информационного общества. С появлением первых ЭВМ наступил этап информатизации разных сторон человеческой деятельности. Если раньше человек основное внимание уделял веществу, затем энергии (рис. 1.1), то сегодня можно без преувеличения сказать, что наступил этап осознания процессов, связанных с информацией. Вычислительная техника создавалась, прежде всего, для обработки данных.



Рисунок 1.1 – Уровень использования человеком различных объектов материального мира.

СППР — это системы, обладающие средствами ввода, хранения и анализа данных, относящихся к определенной предметной области, с целью поиска решений. Ввод данных в СППР осуществляется либо автоматически от датчиков, характеризующих состояние среды или процесса, либо человеком-оператором. В первом случае данные накапливаются путем циклического опроса или по сигналу готовности, возникающему при появлении информации. Во втором случае СППР должны предоставлять пользователям удобные средства ввода данных, контролирующие корректность вводимых данных и выполняющие сопутствующие вычисления. Если ввод осуществляется одновременно несколькими операторами, то система должна решать проблемы параллельного доступа и модификации одних и тех же данных. Постоянное накопление данных приводит к непрерывному росту их объема. В связи с этим на СППР ложится задача обеспечить надежное хранение больших объемов данных. На СППР также могут быть возложены задачи предотвращения несанкционированного доступа, резервного хранения данных, архивирования и т. п.

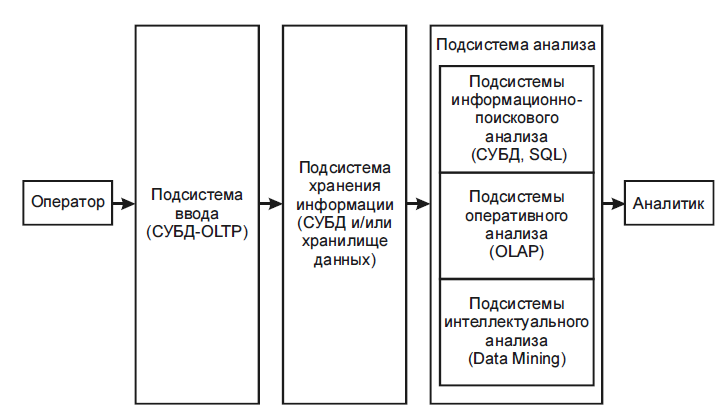
Основная задача СППР — предоставить аналитикам инструмент для выполнения анализа данных. Необходимо отметить, что для эффективного использования СППР ее пользователь-аналитик должен обладать соответствующей квалификацией. Система не генерирует правильные решения, а только предоставляет аналитику данные в соответствующем виде (отчеты, таблицы, графики и т. п.) для изучения и анализа, именно поэтому такие системы обеспечивают выполнение функции поддержки принятия решений. Очевидно, что, с одной стороны, качество принятых решений зависит от квалификации аналитика. С другой стороны, рост объемов, анализируемых данных, высокая скорость обработки и анализа, а также сложность использования машинной формы представления данных стимулируют исследования и разработку интеллектуальных СППР. Для таких СППР характерно наличие функций, реализующих отдельные умственные возможности человека. По степени "интеллектуальности" обработки данных при анализе выделяют три класса задач анализа:

- информационно-поисковый — СППР осуществляет поиск необходимых данных. Характерной чертой такого анализа является выполнение заранее определенных запросов;

- оперативно-аналитический — СППР производит группирование и обобщение данных в любом виде, необходимом аналитику. В отличие от информационно-поискового анализа в данном случае невозможно заранее предсказать необходимые аналитику запросы;

- интеллектуальный — СППР осуществляет поиск функциональных и логических закономерностей в накопленных данных, построение моделей и правил, которые объясняют найденные закономерности и/или прогнозируют развитие некоторых процессов (с определенной вероятностью).

Таким образом, обобщенная архитектура СППР может быть представлена следующим образом (рис. 1.2).

 Рисунок 1.2. Обобщенная архитектура системы поддержки принятия решений.

Рассмотрим отдельные подсистемы более подробно.

- Подсистема ввода данных. В таких подсистемах, называемых OLTP (Onlinetransaction processing), выполняется операционная (транзакционная) обработка данных. Для реализации этих подсистем используют обычные системы управления базами данных (СУБД).

- Подсистема хранения. Для реализации данной подсистемы используют

современные СУБД и концепцию хранилищ данных.

- Подсистема анализа. Данная подсистема может быть построена на основе:

• подсистемы информационно-поискового анализа на базе реляционных СУБД и статических запросов с использованием языка структурных запросов SQL (Structured Query Language);

• подсистемы оперативного анализа. Для реализации таких подсистем применяется технология оперативной аналитической обработки данных OLAP (On-line analytical processing), использующая концепцию многомерного представления данных;

• подсистемы интеллектуального анализа. Данная подсистема реализует методы и алгоритмы Data Mining ("добыча данных").

1.3 OLTP-системы (Системы оперативной обработки транзакций)

OLTP (Online Transaction Processing), транзакционная система — обработка транзакций в реальном времени. Способ организации БД, при котором система работает с небольшими по размерам транзакциями, но идущими большим потоком, и при этом клиенту требуется от системы минимальное время отклика.

Термин OLTP применяют также к системам (приложениям). OLTP-системы предназначены для ввода, структурированного хранения и обработки информации (операций, документов) в режиме реального времени.

Должна обладать 4 свойствами: 1. Атомарность (неделимость): выполняется как одинарная операция доступа к БД, должна выполняться полностью или не выполняться совсем. 2. Согласованность – гарантирует взаимную целостность данных после окончания обработки транзакций. 3. Изолированность (каждая транзакция может изменять данное, которое временно находится в несогласованном состоянии). При этом доступ других транзакций к этим данным запрещен, пока транзакция не завершится. 4. Долговечности – если транзакция выполнена успешно, то изменения не будут потеряны. Результатом выполнения транзакции может быть её фиксация (действие по фиксации изменений в БД) или откат (отмена транзакции и возврат БД в состояние до начала её). Механизм фиксации и откат основан на использовании журнала транзакций, где сохраняется состояние ДО (в нескольких итерациях) и ПОСЛЕ. Некоторые диалекты SQL включают операторы промежуточной фиксации (откат от точки к точке).

Мониторы обработки транзакций (Transaction Processing Monitor — TPM)— это программные системы (относят к посредническому или промежуточному программному обеспечению), решающие задачу эффективного управления информационно-вычислительными ресурсами в распределенной системе. Они представляют собой гибкую, открытую среду для разработки и управления мобильными приложениями, ориентированными на оперативную обработку распределенных транзакций. В числе важнейших характеристик TPM — масштабируемость, поддержка функциональной полноты и целостности приложений, достижение максимальной производительности при обработке данных при невысоких стоимостных показателях, поддержка целостности данных в гетерогенной среде. TPM опираются на трехзвенную модель "клиент-сервер"

На современном рынке мониторов транзакций основными "действующими лицами" являются такие системы, как ACMS (DEC), CICS (IBM), TOP END (NCR), TUXEDO Sytem (Novell).

**Совместное использование данных**

OLTP - системы оперативной обработки транзакций, характеризуются большим количеством изменений, одновременным обращением множества пользователей к одним и тем же данным для выполнения разнообразных операций - чтения, записи, удаления или модификации данных. Для нормальной работы множества пользователей применяются блокировки и транзакции. Эффективная обработка транзакций и поддержка блокировок входят в число важнейших требований к системам оперативной обработки транзакций.

Современные технологии БД предъявляют определенные требования в области архитектуры. До недавнего времени выделялось три класса задач:

* задачи оперативной обработки транзакций;
* задачи пакетной обработки;
* задачи принятия решений.

OLTP-системы - системы оперативной обработки транзакций. Основная функция подобных систем заключается в одновременном выполнении большого количества коротких транзакций от большого числа пользователей. Сами транзакции выглядят относительно просто, например, "снять сумму денег со счета А, добавить эту сумму на счет В" (рис.1.3). Исторически такие системы возникли в первую очередь, поскольку реализовывали потребности в учете, скорости обслуживания, сборе данных и пр.

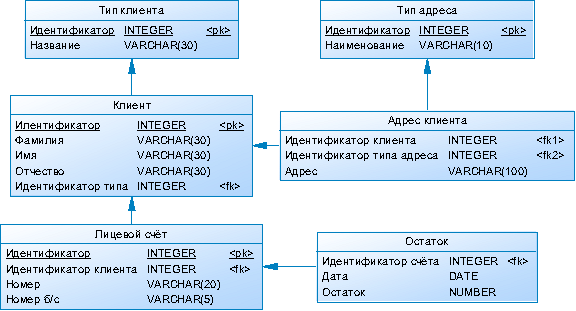
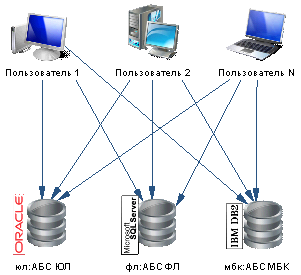
**

Рисунок 1.3. Обработка информации (транзакций) в реальном времени.

.

Системы OLTP характеризуются:

* поддержкой большого числа пользователей;
* малым временем отклика на запрос;
* относительно короткими запросами;
* короткими транзакциями;
* участие в запросах небольшого числа таблиц.

Практически все запросы к базе данных в OLTP-системах состоят из команд вставки, обновления, удаления. Запросы на выборку в основном предназначены для предоставления пользователям возможности выбора из различных справочников. Большая часть запросов, таким образом, известна заранее еще на этапе проектирования системы. Таким образом, критическим для OLTP-приложений является скорость и надежность выполнения коротких операций обновления данных.

Сервер оперативной обработки транзакций строится в предположении:

* OLTP- операции поддерживают большое число пользователей;
* наиболее часто используются короткие простые транзакции;
* обычно транзакции не использую одинаковые данные;
* операторы обычно затрагивают небольшое число строк;
* время отклика - доли секунды;
* только несколько таблиц имеют большие размеры или могут быть изменены.

Реализация такого сервера опирается на:

* физические методики сокращений операций с дисками;
* обработку небольших объемов данных в памяти;
* примитивный оптимизатор запросов;

требование к приложениям - исключить конкуренцию запросов в использовании ресурсов и данных.

# Тема 2. Хранилища данных

* 1. Общая информация о хранилищах данных

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: Buildingthe 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.

Концепция ХД так или иначе обсуждалась специалистами в области информационных систем достаточно давно. Первые статьи, посвященные именно ХД, появились в 1988 г., их авторами были Б. Девлини П. Мэрфи. В 1992 г. У. Инмон подробно описал данную концепцию в своей монографии "Построение хранилищ данных" ("Building the Data Warehouse", second edition —QED Publishing Group, 1996). В основе концепции ХД лежит идея разделения данных, используемых для оперативной обработки и для решения задач анализа. Это позволяет применять структуры данных, которые удовлетворяют требованиям их хранения с учетом использования в OLTP-системах и системах анализа. Такое разделение позволяет оптимизировать как структуры данных оперативного хранения (оперативные БД, файлы, электронные таблицы и т. п.) для выполнения операций ввода, модификации, удаления и поиска, так и структуры данных, используемые для анализа (для выполнения аналитических запросов). В СППР эти два типа данных называются соответственно оперативными источниками данных (ОИД) и хранилищем данных.

**Хранилище данных** – предметно-ориентированный, интегрированный, привязанный ко времени и неизменяемый набор данных, предназначенный для поддержки принятия решений.

В приведенном выше определении указанные характеристики данных рассматриваются следующим образом.

•Предметная ориентированность. Хранилище данных организовано вокруг основных предметов (или субъектов) организации (например, клиенты, товары и сбыт), а не вокруг прикладных областей деятельности (выставление счета клиенту, контроль запасов и продажа товаров). Это свойство отражает необходимость хранения данных, предназначенных для поддержки принятия решений, а необычных оперативно-прикладных данных.

•Интегрированность. Смысл этой характеристики состоит в том, что оперативно-прикладные данные обычно поступают из разных источников, которые часто имеют несогласованное представление одних и тех же данных, например, используют разный формат. Для предоставления пользователю единого обобщенного представления данных необходимо создать интегрированный источник, обеспечивающий согласованность хранимой информации.

•Привязка ко времени. Данные в хранилище точны и действительны только в том случае, если они привязаны к некоторому моменту или промежутку времени. Необходимость привязки хранилища данных ко времени следует из большой длительности того периода, за который была накоплена сохраняемая в нем информация, из явной или неявной связи временных отметок со всеми с данными, а также из того факта, что хранимая информация фактически представляет собой набор снимков состояния данных.

•Неизменяемость. Это означает, что данные не обновляются в оперативном режиме, а лишь регулярно пополняются за счет информации з оперативных систем обработки. При этом новые данные никогда не заменяют, а лишь дополняют прежние. Таким образом, база данных хранилища постоянно пополняется новыми данными, последовательно интегрируемыми с уже накопленной информацией.

Особенности хранилищ данных:

• Хранилища данных содержат информацию, собранную из нескольких оперативных баз данных. Хранилища, как правило, на порядок больше оперативных баз, зачастую имея объем от сотен гигабайт до нескольких террабайт.

* Как правило, хранилище данных поддерживается независимо от оперативных баз данных организации, поскольку требования к функциональности и производительности аналитических приложений отличаются от требований к транзакционным системам.
* Хранилища данных создаются специально для приложений поддержки принятия решений и предоставляют накопленные за определенное время, сводные и консолидированные данные, которые более приемлемы для анализа, чем детальные индивидуальные записи. Рабочая нагрузка состоит из нестандартных, сложных запросов, которые обращаются к миллионам записей и выполняют огромное количество операций сканирования, соединения и агрегирования. Время ответа на запрос в данном случае важнее, чем пропускная способность.
  1. Сравнение OLTP-систем и храниилищ данных

СУБД, созданная для поддержки оперативной обработки транзакций (OLTP),обычно рассматривается как непригодная для организации хранилищ данных, поскольку к этим двум типам систем предъявляются совершенно разные требования. Например, системы OLTP проектируются с целью обеспечения максимально интенсивной обработки фиксированных транзакций, тогда как хранилища данных—прежде всего для обработки единичных произвольных запросов (ad hoc query). В таблице 2.1 для сравнения приведены основные характеристики типичных систем OLTP и хранилищ данных.

Таблица 2.1.

Сравнение основных характеристик типичных систем OLTP и Хранилищ данных

| Система OLTP | Хранилище данных |
| --- | --- |
| Содержит текущие данные | Содержит исторические данные |
| Хранит подробные сведения | Хранит подробные сведения, а также  частично и полностью обобщенные данные |
| Данные являются динамическими | Данные в основном являются статическими |
| Повторяющийся способ обработки данных | Нерегламентированный,  неструктурированный и эвристический  способ обработки данных |
| Высокая интенсивность обработки транзакций | Средняя и низкая интенсивность обработки транзакций |
| Предсказуемый способ использования данных | Непредсказуемый способ использования  данных |
| Предназначена для обработки транзакций | Предназначено для проведения анализа |
| Ориентирована на прикладные области | Ориентировано на предметные области |
| Поддержка принятия повседневных решений | Поддержка принятия стратегических решений |
| Обслуживает большое количество работников исполнительного звена | Обслуживает относительно малое количество работников руководящего звена |

Организация обычно имеет несколько различных систем OLTP, предназначенных для поддержки таких деловых процессов, как управление запасами, выставление счетов клиентам и продажа товаров. Эти системы вырабатывают оперативные данные, которые являются очень подробными, текущими и подверженными изменениям. Системы OLTP оптимально подходят для интенсивной обработки транзакций, которые проектируются заранее, многократно повторяются и связаны преимущественно с обновлением данных. В соответствии с этим и особенностями, данные в системах OLTP организованысогласнотребованиямконкретныхделовыхприложенийипозволяютприниматьповседневныерешениябольшомуколичествупараллельноработающихпользователей-исполнителей.

Хранилища данных предназначены для обработки относительно небольшого количества транзакций, которые имеют непредсказуемый характер и требуют ответа на произвольные, неструктурированные и эвристические запросы. Информация в хранилище данных организована в соответствии с требованиями возможных запросов и предназначена для поддержки принятия долговременных стратегических решений относительно небольшим количеством руководящих работников.

*Характерный запрос для OLTP-систем:*

Какова средняя цена объектов недвижимости в крупнейших городах страны?

*Характерные запросы для ХД:*

- Какие типы объектов недвижимости продаются по ценам выше средней цены объектов недвижимости в крупнейших городах страны, и как эти объекты коррелируются с демографическими данными?

- Какие три района в обслуживаемых городах были более популярны с точки зрения аренды объектов недвижимости в 2017 г., и как эти данные связаны с данными за предыдущие два года?

# Тема 3. Архитектура хранилищ данных

* 1. Архитектура и основные компоненты ХД

В общем случае модель данных современных Систем Поддержки Принятия Решений (СППР) строится на основе **пяти классов данных**:

* источники данных,
* хранилища данных (в узком смысле),
* оперативный склад данных,
* витрины данных,
* метаданные.

Рисунок 3.1 Архитектура хранилища данных.

На рисунке 3.1. цифрами отмечено:

1. Оперативные данные

2. Менеджер загрузки

3. Менеджер хранилища

4. Менеджер запросов

5. Детальные данные

6.1. Частично-обобщённые данные

6.2. Глубоко-обобщённые данные

7. Архивная и резервные копии

8. Метаданные

9. Средства доступа к данным конечного пользователя

**1. Оперативные данные**

Исходящие данные, помещённые в хранилище поступают из следующих источников:

1. Оперативные данные, содержащиеся в БД первого поколения (иерархические и сетевые).

2. Данные различных подразделений, сохраняемые в БД и реляционных СУБД.

3. БД внешних систем, принадлежащих поставщикам или клиентам организаций.

**2. Менеджер загрузки**

Менеджер загрузки, который часто называют внешним (frontend) компонентом, выполняет все операции, связанные с извлечением и загрузкой данных в хранилище. Данные могут извлекаться непосредственно из источников данных, а в последнее время для этого чаще всего применяются хранилища оперативных данных. Операции, выполняемые менеджером загрузки, включают простые преобразования данных, необходимые для их подготовки к вводу в хранилище. Размеры и сложность данного компонента зависят от типа хранилища данных, а в его состав обычно входят не только программы собственной разработки, но и инструменты загрузки, созданные независимыми поставщиками.

**3. Менеджер хранилища**

Менеджер хранилища выполняет все операции, связанные с управлением информацией, помещенной в хранилище данных. Этот компонент может также включать программы собственной разработки и инструменты, предоставленные независимыми компаниями. Менеджер хранилища выполняет следующие операции:

•анализ непротиворечивости данных;

•преобразование и перемещение исходных данных из временной области

Хранения в основные таблицы хранилища данных;

•создание индексов и представлений для базовых таблиц;

•денормализация данных (в случае необходимости);

•агрегирование данных (в случае необходимости);

•резервное копирование и архивирование данных.

Внекоторыхслучаяхменеджерхранилищатакжеанализируетпрофилизапросовдляопределениянеобходимыхиндексовитребованийкагрегированиюданных. Профиль запроса может создаваться для отдельного пользователя, группы пользователей или хранилища данных в целом. Он формируется на основе информации, описывающей такие характеристики запросов, как частота выполнения, набор используемых таблиц и размер результирующего набора данных.

**4. Менеджер запросов**

Менеджер запросов, который часто называют внутренним (backend) компонентом, выполняет все операции, связанные с управлением пользовательскими запросами. Этот компонент обычно создается на основе предоставляемых разработчиком СУБД инструментов доступа к данным, инструментов мониторинга хранилища и программ собственной разработки, использующих весь набор функциональных возможностей СУБД. Сложность менеджера запросов определяется функциональными средствами, которые предоставляются инструментами доступа к данным и самой СУБД. К числувыполняемыхэтимкомпонентомоперацийотносятсяуправлениезапросамиксоответствующимтаблицамисоставлениеграфиковвыполненияэтихзапросов. В некоторых случаях менеджер запросов формирует также профили запросов, позволяющие менеджеру хранилища определить набор необходимых индексов и агрегированных данных.

**5. Детальные данные**

В этот блок загружаются все данные, описанные в концептуальной модели данных хранилища.

**6. Частично обобщённые данные и глубоко-обобщённые данные**

Здесь размещаются все данные, предварительно обработанные менеджером хранилища с целью частичного или глубоко обобщения. Эта часть хранилища является временной, так как постоянно подвергается изменениям в соответствии с вопросами, возникающими у пользователей.

**7. Архивная и резервная копия**

Этот компонент хранилища отвечает за подготовку детальной и обобщённой информации к помещению в резервные и архивные копии. В архивные данные могут быть записаны заранее обобщённые данные, которые требуются для часто повторяемых запросов.

**8. Метаданные (данные о данных)**

В этой области хранилища данных хранятся все те метаданные, которые используются любыми процессами хранилища. Метаданные могут применяться для разных целей, включая перечисленные ниже.

•Извлечение и загрузка данных. Метаданные используются для отображения источников данных на общее представление информации внутри хранилища.

•Обслуживание хранилища. Метаданные применяются для автоматизации подготовки таблиц с итоговой информацией.

•Часть процесса обслуживания запросов. Метаданные используются для направления запроса к наиболее подходящему источнику данных.

Структураметаданныхдляразныхпроцессовможетразличатьсявзависимостиотихназначения. Это означает, что для одного и того же элемента данных в хранилище может храниться сразу несколько вариантов метаданных. Кроме того, большинство компаний-разработчиков применяет собственные версии структуры метаданных в своих инструментах управления копированием данных и средствах доступа, предназначенных для конечных пользователей. В частности, инструменты управления копированием данных используют метаданные для определения правил отображения, которые необходимо применить для преобразования исходных данных в общепринятую форму. Средства доступа конечных пользователей к данным используют метаданные для выбора способа построения запроса. Управлениеметаданнымивнутрихранилищаданныхявляетсяоченьсложнойзадачей, которую не следует недооценивать.

**Метаданные** — это любые данные о данных. Метаданные играют важную роль в построении Систем Поддержки Принятия Решений (СППР). Одновременно это один из наиболее сложных и недостаточно практически проработанных объектов. В общем случае можно выделить по крайней мере три аспекта метаданных, которые должны присутствовать в системе.

* 1. **С точки зрения пользователей:**
     + метаданные для бизнес-аналитиков,
     + метаданные для администраторов,
     + метаданные для разработчиков.
  2. **С точки зрения предметных областей:**
     + структуры данных хранилища,
     + модели бизнес-процессов,
     + описания пользователей,
     + технологические и пр.
  3. **С точки зрения функциональности системы:** 
     + метаданные о процессах трансформации,
     + метаданные по администрированию системы,
     + метаданные о приложениях, метаданные о представлении данных
     + пользователям.

Присутствие трех перечисленных аспектов метаданных подразумевает, что, например, прикладные пользователи и разработчики системы будут иметь различное видение технологических аспектов трансформации данных из источников: прикладные пользователи - семантику, состав и периодичность пополнения хранилища данными из источника, разработчики - ER-диаграммы, правила трансформации и интерфейс доступа к данным источника.

В настоящее время отсутствует единая промышленная технология проектирования, создания и сопровождения метаданных. Поэтому вопросы, связанные с управлением метаданными, рассматриваются отдельно, применительно к каждому конкретному проекту построения СППР.

**9. Средства доступа к данным конечного пользователя**

Основным назначением ХД является представление конечным пользователям информации, необходимой им для принятия стратегических решений. Пользователи взаимодействуют с хранилищем с помощью специальных инструментов доступа к данным. ХД должно обеспечивать эффективное выполнение произвольных запросов и представлять средства для проведения анализов. Высокая производительность хранилища достигается за счёт тщательного предварительного планирования и составления периодических таблиц, которые могут потребоваться конечным пользователям.

Выделяют 5 основных групп в этих средствах доступа к данным:

- Инструмент создания отчётов и запросов;

- Инструменты разработки приложений;

- Инструменты информационной системы руководителя;

- Инструменты оперативно-аналитической разработки (OLAP-

инструменты);

- Инструменты разработки данных (data mining и big data).

Инструменты создания отчётов подразделяются на:

- Итоговые отчёты

- Редакторы отчётов

Информационная система руководителя разработана для поддержкипринятия высокоуровневых стратегических решений.

OLAP – инструменты – инструменты оперативной системы обработки данных. Создаются на основе концепции многомерной базы данных. Эти модели позволяют квалифицированным пользователям анализировать данные с помощью сложных многомерных представлений.

Методы разработки данных превосходят возможности OLAP-технологий, так как они создают предсказательные модели в отличие от ретроспективных моделей OLAP-технологий.

* 1. Информационные потоки в ХД

В технологии хранилищ данных главное внимание уделяется управлению пятью основными информационными потоками: входным, восходящим, нисходящим, выходными метапотоком. Место этих потоков в структуре хранилища данных схематически показано на рис. 3.2. С каждым из этих потоков связаны определенные процессы, которые представлены ниже.

Таблица 3.1.

Информационные потоки в хранилище данных и их процессы.

|  |  |
| --- | --- |
| Информационный поток | Процесс |
| Входной поток | Процессы/связанные с извлечением, очисткой и загрузкой информации из источников данных в хранилище данных |
| Восходящий поток | Процессы, связанные с повышением ценности представленных в хранилище данных посредством суммирования, документирования и распределения исходных данных. |
| Нисходящий поток | Процессы, связанные с архивированием и резервным кодированием информации в хранилище данных. |
| Выходной поток | Процессы, связанные с предоставлением данных пользователям. |
| Метапоток | Процессы, связанные с управлением метаданными |

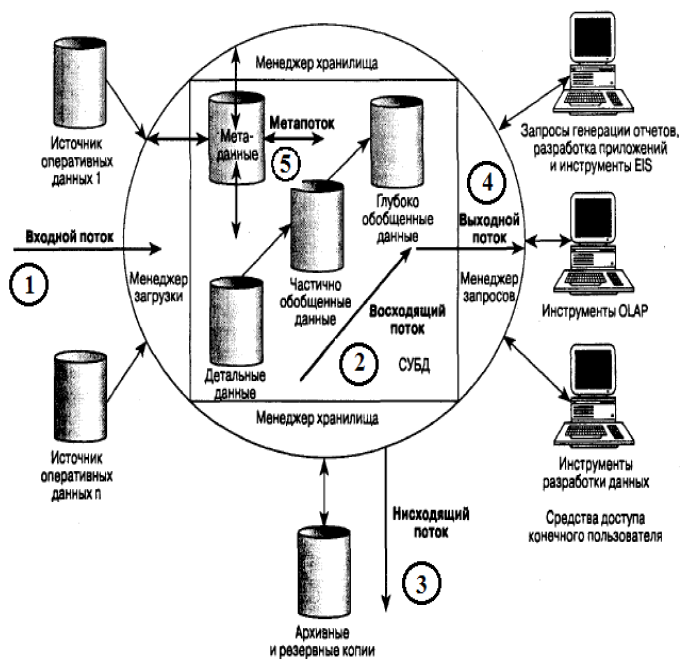


Рисунок 3.2.Информационныепотокивхранилищеданных.

**1. Входной поток**

Входной поток связан с выборкой информации из источников, данных с целью их последующей загрузки в хранилище данных. В качестве альтернативы, данные могут первоначально загружаться в хранилище оперативных данных (Operationl Data Store — ODS) перед передачей в хранилище данных. Поскольку исходные данные вырабатываются в основном системами OLTP, они должны быть перестроены в соответствии с требованиями хранилища данных. Перестройка данных включает следующие операции:

•очистка данных;

•преобразование данных в соответствии с требованиями хранилища данных, включая добавление и/или удаление полей и денормализацию данных;

•проверка внутренней не противоречивости данных и их непротиворечивости по отношению к данным, уже загруженным в хранилище.

Для эффективного управления входным потоком необходимо подобрать механизм определения момента начала извлечения данных с последующим выполнением требуемых преобразований и проверкой не противоречивости данных. Для создания единого не противоречивого представления корпоративных данных очень важно в процессе извлечения информации из источников убедиться в том, что она находится в согласованном состоянии. Сложность процесса извлечения информации зависит от степени взаимной согласованности между различными источниками данных.

После извлечения из источника данные обычно загружаются во временное хранилище с целью выполнения очистки и проверки их не противоречивости.

Поскольку этот процесс достаточно сложен, важно, чтобы он был полностью автоматизирован и сопровождался выдачей сообщений о любых возникающих проблемах или сбоях. Для обслуживания входного потока предусмотрены специальные коммерческие инструменты. Однако если необходимая процедура не является относительно простой, то в случае использования коммерческих инструментов потребуется выполнить их предварительную настройку.

**2. Восходящий поток**

Обслуживание восходящего потока включает выполнение приведенных ниже действий.

Суммирование данных выполняется с помощью операций выборки, проекции, соединения и группирования реляционных данных, для получения представлений, которые являются более удобными и полезными для конечных пользователей. Суммирование может включать выполнение не только простых реляционных операций, но и проведение сложного статистического анализа, включая определение тенденций, кластеризацию и получение выборочных данных.

•Документирование данных. Выполняется путем преобразования фактических или суммарных данных в более удобные форматы представления, такие как электронные таблицы, текстовые документы, диаграммы и другие графические презентации, закрытые базы данных и анимационные материалы.

•Распределение исходных данных. Предусматривает распределение данных на соответствующие группы для повышения их подготовленности к использованию и доступности.

При планировании процедур повышения ценности данных следует учитывать  
необходимость увеличения общей производительности хранилища, а также снижения текущих расходов на его сопровождение. Все эти требования противоречат друг другу, что вынуждает разработчиков либо повышать производительность обработки запросов, либо сокращать расходы на сопровождение. Иначе говоря, администратор хранилища данных должен разработать проект базы, наиболее подходящий для удовлетворения всех существующих требований, для чего часто приходится идти на компромисс.

**3. Нисходящий поток** – архивирование данных

Архивирование устаревших данных играет важную роль при обеспечении высокой эффективности и производительности хранилища данных за счет переноса устаревших данных с ограниченной ценностью на архивный носитель, например на магнитную ленту или оптические диски. Но если схема секционирования базы данных выполнена правильно, то общее количество оперативных данных не должно влиять на производительность хранилища данных. Секционирование — это один из способов проектирования логической организации очень крупных баз данных, который позволяет разбивать таблицы, содержащие слишком большое количество записей, на несколько меньших таблиц. Для секционирования каждой конкретной таблицы применяется правило, в котором учтены такие характеристики данных, как период, к которому они относятся, или географический регион.

**4. Выходной поток**

Именно благодаря выходному потоку информации у сотрудников организации создается представление об истинной ценности хранилища данных. Полученные данные могут потребовать перестройки всех деловых процессов организации с целью повышения ее конкурентоспособности.

В качестве основных действий, связанных с выходным потоком, следует упомянуть перечисленные ниже.

• Доступ к данным. Обеспечивает выполнение запросов конечных пользователей к нужным им данным. Главная цель заключается в создании такой  
среды, в которой пользователи смогли бы эффективно использовать инструменты создания запросов для получения доступа к наиболее подходящему источнику данных. Частота выполнения отдельных запросов пользователей может изменяться от однократно выполняемого произвольного запроса до регулярно выполняемых запросов или даже запросов, выполняемых в реальном времени. При подготовке графика выполнения запросов пользователей очень важно обеспечить использование системных ресурсов максимально эффективным образом.

• Доставка. Означает своевременную доставку информации на рабочие станции конечных пользователей. Это — относительно новая область обработки информации в хранилищах данных, связанная с процессами типа публикации/подписки. Хранилище данных публикует различные деловые объекты, которые периодически подвергаются пересмотру с учетом интенсивности их использования. Пользователи могут подписаться на такой набор деловых объектов, который в наибольшей степени соответствует их потребностям. Важной особенностью управления выходным потоком является активная популяризация хранилища данных среди пользователей, что может оказать существенное влияние на функционирование всей организации. В числе дополнительных действий по обслуживанию выходного потока следует также назвать направление запросов к соответствующим целевым таблица^ и обработку профилей запросов.

**5. Поток метаданных**

Предыдущие потоки характеризуют управление хранилищем данных в отношении перемещения данных в хранилище и из него. Метапоток — это процесс, связанный с перемещением метаданных, т.е. данных о других потоках. Метаданные — это описание информационного содержания хранилища данных: что в нем содержится, откуда что поступает, какие операции выполнялись во время очистки, как осуществлялись интеграция и обобщение данных. В ответ на изменение деловых потребностей постепенно менялись и продолжают изменяться любые традиционные унаследованные системы. Поэтому при управлении хранилищем данных необходимо учитывать эти продолжающиеся изменениями

* 1. Магазины данных

Хранилища данных в общем случае представляют собой единый источник информации для любой обработки, связанной с поддержкой принятия решений. Однако в начале 90-х годов, когда хранилища данных только приобретали популярность, было обнаружено, что чаще всего пользователи составляли пространные отчеты и выполняли различные операции анализа данных на относительно небольшом подмножестве полного объема информации в хранилище данных. И действительно, пользователи повторяли те же самые операции на том же самом подмножестве данных каждый раз после их обновления. Более того, некоторые из этих операций— например, предикативный анализ (прогноз), имитация, моделирование "что если" на основе деловых данных — включали создание новых схем и данных с последующим обновлением этих новых данных.

Неоднократное повторное выполнение таких операций на одном и том же подмножестве информации полного хранилища данных, безусловно, не очень эффективно. Поэтому возникла очевидная идея построения некоторого ограниченного "хранилища" специального назна­чения, которое подходило бы для достижения рассматриваемых целей. Кроме того, в некото­рых случаях можно извлекать и обрабатывать данные непосредственно из локальных источ­ников, предоставляя более быстрый доступ к данным по сравнению с тем, который мог быть предоставлен при синхронизации со всеми остальными данными, загруженными в полное хранилище. Подобные соображения привели к появлению концепции магазинов данных.

На самом деле вокруг точного определения термина *магазин данных* еще ведутся споры. Для наших целей магазин данных можно определить как "специализированное, предметно-ориентированное, интегрированное, непостоянное, изменяемое во времени хранилище данных для поддержки конкретного подмножества управленческих решений". Как видим, ключевое отличие между магазинами данных и хранилищами данных заключается в том, что магазины данных — *специализированные* и *непостоянные.* Под характеристикой *специализированные* подразумевается, что они содержат данные для поддержки лишь некоторой конкретной области делового анализа, а под характеристикой *непостоянные* подразумевается, что пользователи могут обновлять данные и, возможно, даже создавать в каких-то целях новые данные, например новые таблицы.

*Магазин данных* – подмножество хранилища данных, которое поддерживает требования отдельного подразделения или деловой сферы организации. Магазин данных содержит некоторое подмножество данных хранилища данных, которое обычно представлено в виде обобщенной информации, связанной с некоторым подразделением или деловой сферой предприятия. Магазин данных может быть независимым или определенным образом, связанным с централизованным хранилищем данных. Популярность магазинов данных основана на том очевидном факте, что корпоративные хранилища данных создавать и использовать сложнее.

На рис. 3.3 показана типичная архитектура хранилища данных и связанных с ним магазинов данных.

Ниже перечислены основные различия между магазином и хранилищем данных.

•Магазин данных отвечает требованиям пользователей только одного из подразделений организации или некоторой ее деловой сферы.

•Магазин данных обычно не содержит подробных оперативных сведений (в отличие от хранилища данных).

•Поскольку магазин данных содержит меньше информации, чем хранилище, структура информации магазина данных более понятна и проста в управлении.

Существует несколько подходов к созданию магазинов данных. Один из них заключается в создании нескольких магазинов данных, предназначенных для последующего объединения в хранилище данных, а другой подход предусматривает создание инфраструктуры для корпоративного хранилища данных и вместе с тем формирование одного или нескольких магазинов данных для удовлетворения неотложных деловых потребностей.

Для магазина данных можно выбрать двух- или трехуровневую архитектуру.

При этом хранилище данных (если оно поставляет данные для магазина данных) образует первый уровень (в случае необходимости), магазин данных —второй уровень, а рабочая станция пользователя—третий (см. рис. 3.3). Данные распределены между всеми этими тремя уровнями.

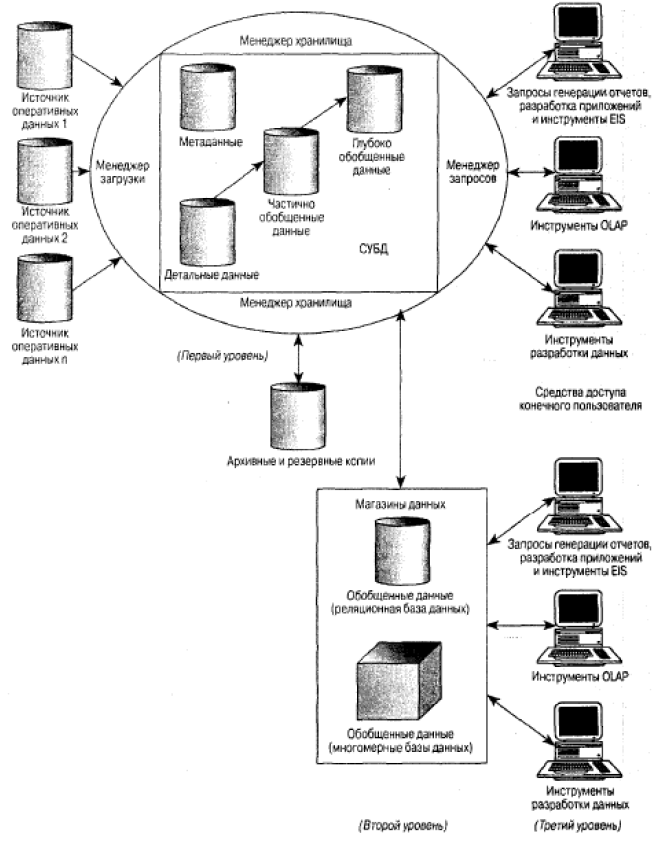


Рисунок 3.3. Магазины данных.

* 1. Предпосылки для создания магазинов данных

Существует несколько перечисленных ниже причин, по которым следует создавать магазины данных.

* Для предоставления пользователям доступа к данным, которые приходится анализировать чаще других.
* Для предоставления данных группе пользователей некоторого отдела или деловой сферы в форме, которая соответствует их коллективному представлению о данных.
* Для сокращения времени ответа на запрос (за счет сокращения объема обрабатываемых данных).
* Для предоставления данных, структурированных в соответствии с требованиями доступа к данным. Для многих инструментов доступа к данным, таких как инструменты разработки данных или оперативной аналитической обработки (OLAP), может потребоваться создать отдельную внутреннюю структуру базы данных. На практике подобные инструменты часто создают свои собственные магазины данных, предназначенные исключительно для поддержки их специфических функций.
* Магазины данных обычно содержат меньший объем данных, поэтому такие задачи, как очистка, загрузка, преобразование и интеграция данных, выполняются проще. Следовательно, реализация и настройка магазина данных требует меньше усилий, чем разработка и реализация корпоративного хранилища данных.
* Стоимость реализации магазина данных обычно существенно ниже, чем стоимость создания хранилища данных.
* Круг потенциальных пользователей магазина данных более четко определен, поэтому учесть их требования и организовать необходимую поддержку проще, чем в случае корпоративного хранилища данных.
  1. Характеристики магазинов данных

Проблемы, связанные с разработкой и управлением магазинами данных, перечислены ниже:

* Функциональность  
  Размер базы данных  
  Производительность загрузки данных  
  Доступ пользователей к данным в нескольких магазинах данных  
  Доступ к магазинам данных через Internet/внутреннюю сеть  
  Администрирование  
  Установка

**Функциональность**

Возможности магазина данных возрастают вместе с ростом их популярности. В отличие от небольших и легкодоступных баз данных, некоторые магазины данных должны масштабироваться вплоть до гигабайтовых размеров и предоставлять средства для выполнения сложного анализа с использованием инструментов оперативной аналитической обработки (OLAP) и/или инструментов разработки данных. Кроме того, должна быть предоставлена возможность дистанционного доступа к магазину данных для сотен пользователей. Сложность и размер некоторых магазинов данных порой соответствуют сложности и размерам

**Размер базы данных**

Пользователям магазина данных требуется обеспечить меньшее время реакции системы по сравнению со временем ответа при обработке запросов в хранилище данных. Однако производительность выполнения запросов падает по мере возрастания объема данных магазина. Некоторые компании-разработчики магазинов данных исследуют способы сокращения объемов данных магазинов с целью повышения их производительности. Например, возможности динамической

**Производительность загрузки данных**

В магазине данных должен поддерживаться компромисс между двумя важными параметрами: временем реакции системы и производительностью загрузки данных. Магазин данных, созданный для быстрого получения ответа на запросы пользователя, должен иметь большое количество сводных таблиц и агрегированных данных. К сожалению, создание таких таблиц и подготовка таких данных существенно увеличивают продолжительность загрузки данных. Компании разработчики внимательно исследуют этот вопрос и пытаются повысить автоматически и непрерывно адаптируются к обрабатываемым данным, или предоставляют возможность инкрементного обновления базы данных, что позволяет обновлять только данные, подвергшиеся изменениям, а не всю структуру ММБД.

**Доступ пользователей к данным в нескольких магазинах данных**

Одно из решений этой задачи заключается в репликации данных между различными магазинами данных, или, другими словами, в создании виртуальных

магазинов данных. Виртуальные магазины данных являются представлениями скольких физических магазинов данных или корпоративных хранилищ данных, настроенных согласно требованиям особых групп пользователей. В настоящее время уже ведутся коммерческие поставки программных продуктов, предназначенных для управления виртуальными магазинами данных.

**Доступ к магазинам данных через Internet/внутреннюю сеть**

Технологии Internet/внутренней сети предоставляют пользователям дешевые средства доступа к магазинам и хранилищам данных с помощью таких Web браузеров, как Netscape Navigator и Microsoft Internet Explorer. Инструменты доступа к магазинам данных через Internet/внутреннюю сеть обычно располагаются между Web-браузером и инструментами анализа данных. Компании разработчики продолжают создавать продукты, обладающие все более широкими возможностями работы в среде Web.

**Администрирование**

По мере увеличения количества магазинов данных в организации возрастает необходимость обеспечить централизованное управление и координацию деятельности всех магазинов данных. После копирования в магазины данные могут стать противоречивыми, поскольку каждый пользователь может изменять их в своем магазине данных с целью выполнения различных видов анализа. Администрирование нескольких магазинов данных в пределах всей организации является сложной задачей, поскольку для ее решения приходится выполнять поддержку версий, контролировать непротиворечивость и целостность данных и метаданных, обеспечивать защиту информации в пределах всего предприятия, а также выполнять настройку производительности. Уже ведутся коммерческие поставки инструментов администрирования магазинов данных.

**Установка**

Процесс создания магазина данных со временем все больше усложняется. Компании-разработчики предлагают программные продукты, называемые "полностью готовыми к установке магазинами данных", которые действительно представляют собой недорогой набор инструментов для работы с магазинами данных.

* 1. Витрины данных

Одним из вариантов реализации на практике хранилища данных является построение витрин данных (Data Marts). Иногда их называют также киосками данных.

Витрина данных — это предметно-ориентированная совокупность данных, имеющая специфическую организацию. Содержание витрин данных, как правило, предназначено для решения некоего круга однородных задач одной или нескольких смежных предметных областей, либо для выполнения конкретных бизнес-функций, либо для конкретных подразделений. Например, для решения задач, связанных с анализом кредитных услуг банка, используется одна витрина, а для работ по анализу деятельности банка на фондовом рынке — другая.

Витрины данных (data mart) – это упрощенный вариант хранилища данных, содержащий только тематически объединенные данные.

Витрины данных являются объектами хранения аналитической информации, нацеленными на поддержку конкретных бизнес-функций, конкретных подразделений компании. На уровне базы данных витрины обычно реализуются по схеме «звезда» или «снежинка» и содержат данные из области детальных данных (System of records). Также могут быть реализованы в виде многомерного OLAP-куба. Витрины данных являются основой, обеспечивающей возможность проведения многомерного анализа (OLAP) данных.

Ниже представлены основные принципы проектирования витрин данных.

1. Витрины данных ориентированы на бизнес и при их проектировании необходимо учесть все измерения, показатели и иерархии, необходимые пользователям.
2. При проектировании витрин данных необходимо учитывать особенности BI-приложения, используемого на проекте. Например, в Oracle Discoverer нет возможности создавать несбалансированные иерархии и это нужно учитывать.

Следовательно, витрина данных — это относительно небольшие и специализированные хранилища данных, содержащие только тематически ориентированные данные и предназначенные для использования конкретным функциональным подразделением. Итак, функционально-ориентированные витрины данных представляют собой структуры данных, обеспечивающие решение аналитических задач в конкретной функциональной области или подразделении компании, например управление прибыльностью, анализ рынков, анализ ресурсов, анализ денежных потоков, анализ клиентской базы, маркетинговое исследование, управление активами и пассивами и т.д. Таким образом, витрины данных можно рассматривать как маленькие тематические хранилища, которые создаются с целью информационного обеспечения аналитических задач конкретных управленческих подразделений компании.

Организация данных в витрину определяется необходимостью обеспечить возможности анализа данных той или иной предметной области наиболее оптимальными средствами.

Витрины данных и хранилище данных значительно отличаются друг от друга. Хранилище данных создается для решения корпоративных задач, присутствующих в корпоративной информационной системе. Обычно хранилища данных создаются и приобретаются организациями с центральным подчинением, такими как классические организации информационных технологий, например банк. Хранилище данных составляется усилиями всей корпорации.

Витрина данных разрабатывается для удовлетворения потребностей в решении конкретного однородного круга задач. Поэтому в одной компании может быть много различных витрин данных, каждая из которых имеет свой собственный внешний вид и содержание.

Следующее отличие состоит в степени детализации данных, поскольку витрина данных содержит уже агрегированные данные. В хранилище данных, наоборот, находятся максимально детализированные данные.

Поскольку уровень интеграции в витринах данных более высок, чем в хранилищах, нельзя легко разложить степень детализации витрины данных в степень детализации хранилища. Но всегда можно последовать в обратном направлении и агрегировать отдельные данные в обобщенные показатели.

В отличие от хранилища витрина данных содержит лишь незначительный объем исторической информации, привязанной только к небольшому отрезку времени и существенной только в момент, когда она отвечает требованиям решения задачи. Витрины данных можно представить в виде логически или физически разделенных подмножеств хранилища данных. На рис. 3.4 представлена взаимосвязь витрин данных и хранилища данных на примере банковской сферы.

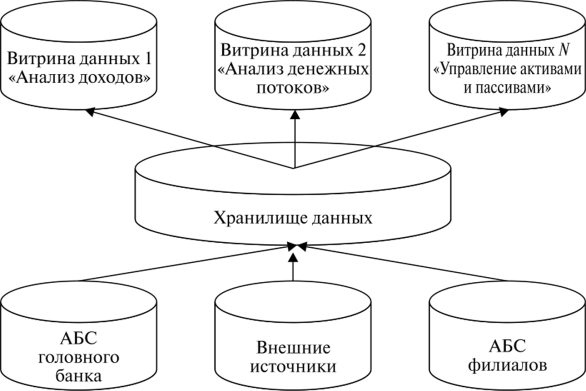


Рисунок 3.4. Взаимосвязь витрин данных и хранилища данных.

Витрины данных, как правило, размещаются в многоуровневой технологии, которая оптимальна для гибкости анализа, но не оптимальна для больших объемов данных.

Структура витрин данных также ориентирована на многомерную организацию данных в виде куба. Однако их построение в силу ограниченности информационного диапазона, обеспечивающего потребности одной функциональной области, значительно проще и выгоднее.

Существуют два типа витрин данных — зависимые и независимые. Зависимая витрина данных — это та, источником которой является хранилище данных. Источником независимой витрины данных является среда первичных программных приложений. Зависимые витрины данных стабильны и имеют прочную архитектуру. Независимые витрины данных нестабильны и имеют неустойчивую архитектуру, по крайней мере при пересылке данных.

Надо отметить, что витрины данных являются идеальным решением наиболее существенного конфликта при проектировании хранилища данных — производительность или гибкость. В общем, чем более стандартизирована и гибка модель хранилища данных, тем менее продуктивно она отвечает на запросы. Это связано с тем, что запросы, поступающие в стандартно спроектированную систему, требуют значительно больше предварительных операций, чем в оптимально спроектированной системе. Направляя все запросы пользователя в витрины данных, поддерживая гибкую модель для хранилища данных, разработчики могут достичь гибкости и продолжительной стабильности структуры хранилища, а также оптимальной производительности для запросов пользователей.

Данные, попав в хранилище, могут быть распространены среди многих витрин данных для доступа пользовательских запросов. Эти витрины данных могут принимать различные формы — от баз данных «клиент—сервер» до баз данных на рабочем столе, ОЬАР- кубов или даже динамических электронных таблиц. Выбор инструментов для пользовательских запросов может быть широким и отображать предпочтения и опыт конкретных пользователей. Широкий выбор таких инструментов и простота их применения сделают их внедрение наиболее дешевой частью реализации проекта хранилища данных. Если данные в хранилище имеют хорошую структуру и проверенное качество, то их передача в другие витрины данных станет рутинной и дешевой операцией.

Использование технологий витрин данных, как зависимых, так и независимых, позволяет решать задачу консолидации данных из различных источников в целях наиболее эффективного решения задач анализа данных. При этом источниками могут быть различные учетные и справочные системы, различающиеся по архитектуре и функциональности, в частности территориально разрозненные.

# Тема 4. OLAP-системы

* 1. Правила Кодда для OLAP-систем

**OLAP** — это инструменты оперативного анализа данных, содержащихся в хранилище, которые предназначены для общения аналитика с проблемой, а не с компьютером.

Основной вопрос при обработке информации заключается в том, как обрабатывать всё более и более крупные БД, которые содержат информацию с постоянной усложняющейся структурой, и как создать систему, отвечающую на запросы за приемлемое время.

OLAP занимается динамическим синтезом, анализом и обобщением больших объемов многомерных данных. Эта технология должна поддерживать сложные аналитические системы.

12 правил Б.Кодда:

1. Многомерное концептуальное представление данных

2. Прозрачность

3. Доступность

4. Неизменно высокая производительность подготовки отчетов

5. Архитектура клиент-сервер

6. Универсальность измерений

7. Динамическое управление разреженных матриц

8. Многопользовательская поддержка

9. Неограниченные перекрестные операции между размерностями

10. Поддержка удобных средств манипулирования данными

11. Гибкость средств формирования отчетов

12. Неограниченное число измерений и уровней агрегирования.

Рассмотрим более детально каждое из правил.

1. **Многомерность.**

OLAP-система на концептуальном уровне должна представлять данные в виде многомерной модели, что упрощает процессы анализа и восприятия информации.

1. **Прозрачность.**

OLAP-система должна скрывать от пользователя реальную реализацию многомерной модели, способ организации, источники, средства обработки и хранения.

1. **Доступность.**

OLAP-система должна предоставлять пользователю единую, согласованную и целостную модель данных, обеспечивая доступ к данным независимо от того, как и где они хранятся.

1. **Постоянная производительность при разработке отчетов.**

Производительность OLAP-систем не должна значительно уменьшаться при увеличении количества измерений, по которым выполняется анализ.

1. **Клиент-серверная архитектура.**

OLAP-система должна быть способна работать в среде "клиент-сервер", т. к. большинство данных, которые сегодня требуется подвергать оперативной аналитической обработке, хранятся распределено. Главной идеей здесь является то, что серверный компонент инструмента OLAP должен быть достаточно интеллектуальным и позволять строить общую концептуальную схему на основе обобщения и консолидации различных логических и физических схем корпоративных БД для обеспечения эффекта прозрачности.

1. **Равноправие измерений.**

OLAP-система должна поддерживать многомерную модель, в которой все измерения равноправны. При необходимости дополнительные характеристики могут быть предоставлены отдельным измерениям, но такая возможность должна быть у любого измерения.

1. **Динамическое управление разреженными матрицами.**

OLAP-система должна обеспечивать оптимальную обработку разреженных матриц. Скорость доступа должна сохраняться вне зависимости от расположения ячеек данных и быть постоянной величиной для моделей, имеющих разное число измерений и различную степень разреженности данных.

1. **Поддержка многопользовательского режима.**

OLAP-система должна предоставлять возможность нескольким пользователям работать совместно с одной аналитической моделью или должна создавать для них различные модели из единых данных. При этом возможны как чтение, так и запись данных, поэтому система должна обеспечивать их целостность и безопасность.

1. **Неограниченные перекрестные операции.**

OLAP-система должна обеспечивать сохранение функциональных отношений, описанных с помощью определенного формального языка между ячейками гиперкуба при выполнении любых операций среза, вращения, консолидации или детализации. Система должна самостоятельно (автоматически) выполнять преобразование установленных отношений, не требуя от пользователя их переопределения.

10.**Интуитивная манипуляция данными.**

OLAP-система должна предоставлять способ выполнения операций среза, вращения, консолидации и детализации над гиперкубом без необходимости пользователю совершать множество действий с интерфейсом. Измерения, определенные в аналитической модели, должны содержать всю необходимую информацию для выполнения вышеуказанных операций.

11.**Гибкие возможности получения отчетов.**

OLAP-система должна поддерживать различные способы визуализации данных, т. е. средства формирования отчетов должны представлять синтезируемые данные или информацию, следующую из модели данных, в ее любой возможной ориентации. Это означает, что строки, столбцы или страницы должны показывать одновременно от 0 *ro N  измерений*, где *N*– число измерений всей аналитической модели. Кроме того, каждое измерение содержимого, показанное в одной записи, колонке или странице, должно позволять показывать любое подмножество элементов (значений), содержащихся в измерении, в любом порядке.

12.**Неограниченная размерность и число уровней агрегации.**

Исследование о возможном числе необходимых измерений, требующихся в аналитической модели, показало, что одновременно могут использоваться до 19 измерений. Отсюда вытекает настоятельная рекомендация, чтобы аналитический инструмент мог одновременно предоставить хотя бы 15, а предпочтительнее – и 20 измерений. Более того, каждое из общих измерений не должно быть ограничено по числу определяемых пользователем-аналитиком уровней агрегации и путей консолидации.

* 1. Многомерная OLAP-технология

В процессе принятия решений пользователь генерирует некоторые гипотезы. Для превращения их в законченные решения эти гипотезы должны быть проверены. Проверка гипотез осуществляется на основании информации об анализируемой предметной области. Как правило, наиболее удобным способом представления такой информации для человека является зависимость между некоторыми параметрами. Например, зависимость объемов продаж от региона, времени, категории товара и т. п. Другим примером может служить зависимость количества выздоравливающих пациентов от применяемых средств лечения, возраста и т. п.

В процессе анализа данных, поиска решений часто возникает необходимость в построении зависимостей между различными параметрами. Кроме того, число таких параметров может варьироваться в широких пределах. Как уже отмечалось ранее, традиционные средства анализа, оперирующие данными, которые представлены в виде таблиц реляционной БД, не могут в полной мере удовлетворять таким требованиям. В 1993 г. Е. Кодд — основоположник реляционной модели БД — рассмотрел ее недостатки, указав в первую очередь на невозможность "объединять, просматривать и анализировать данные с точки зрения множественности измерений, т. е. самым понятным для аналитиков способом".

Измерение — это последовательность значений одного из анализируемых параметров. Например, для параметра "время" это последовательность календарных дней, для параметра "регион" это может быть список городов.

Множественность измерений предполагает представление данных в виде многомерной модели. По измерениям в многомерной модели откладывают  
параметры, относящиеся к анализируемой предметной области.

По Кодду, многомерное концептуальное представление (multi-dimensional  
conceptual view) — это множественная перспектива, состоящая из нескольких независимых измерений, вдоль которых могут быть проанализированы определенные совокупности данных. Одновременный анализ по нескольким измерениям определяется как многомерный анализ.

Каждое измерение может быть представлено в виде иерархической структуры. Например, измерение "Исполнитель" может иметь следующие иерархические уровни: "предприятие — подразделение — отдел — служащий". Более того, некоторые измерения могут иметь несколько видов иерархического представления. Например, измерение "Время" может включать две иерархии со следующими уровнями: "год — квартал — месяц — день" и "неделя — день".

На пересечениях осей измерений (Dimensions) располагаются данные, количественно характеризующие анализируемые факты, — меры (Measures). Это могут быть объемы продаж, выраженные в единицах продукции или в денежном выражении, остатки на складе, издержки и т. п.

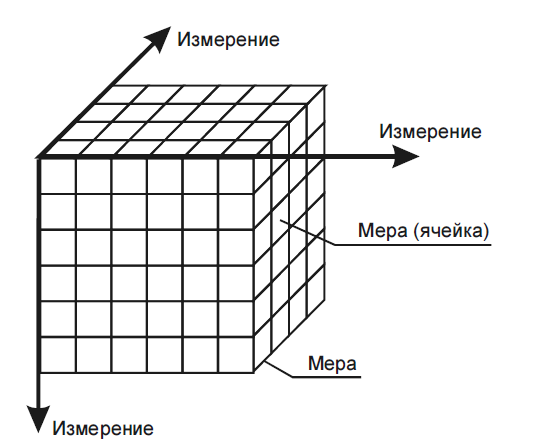


Рисунок 4.1. Представление данных в виде гиперкуба.

Таким образом, многомерную модель данных можно представить, как гиперкуб (рис. 4.1) (конечно, название не очень удачное, поскольку под кубом обычно понимают фигуру с равными ребрами, что в данном случае далеко не так). Ребрами такого гиперкуба являются измерения, а ячейками—меры.

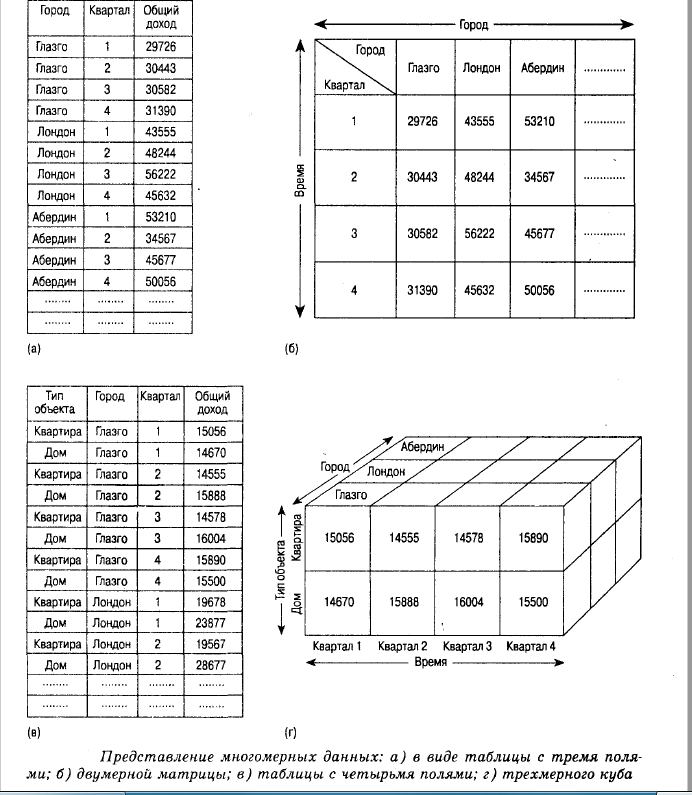


Рисунок 4.2. Представление многомерных данных:

а) в виде таблицы с тремя полями;

б) двумерной матрицы;

в) таблицы с четырьмя полями;

г) трехмерного куба.

За последние несколько лет анализ большого объёма данных для принятия стратегических решений и совершенствования процессов повседневного принятия решений вышел на первый план. Возникла потребность в системах, призванных обеспечить надёжный и быстрый доступ к большому объёму накопленной информации для её анализа.

OLAP системы имеют высокую скорость поиска информации в отличие от реляционных СУБД. Строится многомерная база данных на основе многомерной модели данных. Многомерная модель данных определяет представление данных на трёх уровнях:

* Концептуальной модели;
* Физической модели;
* Прикладной модели данных.

Концептуальная модель данных описывает представление данных в системе и методы их описания. В терминах этой модели описываются данные предприятия, их структура, методы расчёта.

Физическая модель данных определяет, как характеризуются данные на физических носителях, в каких типах файла, как хранятся в сети, как осуществляется доступ к ним, как данные кэшируются.

Прикладная модель данных определяет формат данных, в котором они передаются аналитическому приложению.

Многомерное пространство данных (МПД)

В реляционных базах данных имеем двухмерное пространство, то есть плоская таблица с координатами и атрибутами.

Многомерные данные – куб.

МПД может иметь любое количество измерений, которое не обязаны быть одинакового размера.

Если геометрические многомерные пространства непрерывны по осям, то МПД – дискретно.

Описание МПД

1. Измерение (dimension) – описывает элемент данных, по которому производится анализ.
2. Элемент – одна точка на измерении.
3. Значение элемента – уникальная характеристика элемента.
4. Атрибут – полная коллекция элементов одного типа.
5. Размер (кардинальное число) – кардинальность измерения – количество элементов измерения.

Кортеж – координата в многомерном пространстве.

Срез – секция или сечение многомерного пространства, которая может быть определена кортежем.

Многомерное пространство определяется набором координат (кортежем).

Атрибуты измерений

Если в запросе проходит время по месяцам, то по координатам могут быть выделены наиболее крупные элементы, например, квартал.

Данные на оси могут иметь разную степень детализации => иерархия измерений.

Три типа иерархии измерений:

Сбалансированная

Несбалансированная

Неровная

Ячейки. Если меняем шкалу измерения, то появляются новые позиции, соответствующие новым элементам атрибутов. Эти элементы атрибутов содержат множество новых точек в пространстве, но для них нет данных в исходных таблицах. Значения их могут быть вычислены из значений, заданных фактическими данными -> появляется новое пространство – логическое пространство данных.

Полный набор точек пространства, объединяющий фактические и логические пространства, называется многомерным кубом или моделью.

Мера – значение данных в ячейке.

Функция агрегирования данных – функция, позволяющая вычислять значения ячеек в логическом пространстве из значения ячеек фактического пространства.

Подкуб – часть полного пространства куба в виде некоторой многомерной фигуры в виде куба; срезы – это подкубы. Подкубы могут иметь нормальную и произвольную форму.

* 1. Категориии нструментов OLAP –технологий

OLAP-система включает в себя два основных компонента:

OLAP-сервер—обеспечивает хранение данных, выполнение над ними необходимых операций и формирование многомерной модели на концептуальном уровне. В настоящее время OLAP-серверы объединяют с ХД или ВД;

OLAP-клиент—представляет пользователю интерфейс к многомерной модели данных, обеспечивая его возможностью удобно манипулировать данными для выполнения задач анализа.

OLAP-серверы скрывают от конечного пользователя способ реализации многомерной модели. Они формируют гиперкуб, с которым пользователи посредством OLAP-клиента выполняют все необходимые манипуляции, анализируя данные. Между тем способ реализации очень важен, т. к. от него зависят такие характеристики, как производительность и занимаемые ресурсы. Основные способы реализации п 1-3 представлены в таблице 4.1, но также часто в литературе по OLAP-системам можно встретить аббревиатуры DOLAP и JOLA (п. 4-5).

Таблица 4.1. Способы реализации OLAP-систем

|  |  |  |
| --- | --- | --- |
| 1 | MOLAP | многомерный (multivariate) OLAP. Для реализации многомерной модели используют многомерные БД |
| 2 | ROLAP | реляционный (relational) OLAP. Для реализации многомерной модели используют реляционные БД |
| 3 | HOLAP | гибридный (hybrid) OLAP. Для реализации многомерной модели используют и многомерные, и реляционные БД |
| 4 | DOLAP | Настольный (desktop) OLAP. Является недорогой и простой в использовании OLAP-системой, предназначенной для локального анализа и представления данных, которые загружаются из реляционной или многомерной БД на машину клиента |
| 5 | JOLAP | новая, основанная на Java коллективная OLAP-API-инициатива,предназначеннаядлясозданияиуправленияданнымииметаданныминасерверах OLAP. Основной разработчик— Hyperion Solutions. Другими членами группы, определяющей предложенный API, являются компании IBM, Oracle идр. |

1. MOLAP – классическая форма OLAP, использующая многомерную БД. Данные представляют собой упорядоченные многомерные массивы, разделенные на гиперкубы, атрибуты которых имеют одинаковую размерность, и поликубы, в которых каждый атрибут хранится с собственным набором измерений.

Преимущества MOLAP:

легко встраиваются различные функции;

поиск и выборка выполняются быстрее, чем в реляционных БД.

Недостатки:

в многомерной БД резко уменьшается объем детализированных данных, что сокращает диапазон анализа;

увеличивается объем информации;

многомерная БД чувствительна к изменениям в многомерной модели.

Условия эффективности использования многомерных БД:

- объем исходных данных составляет порядка нескольких гигабайт;

- набор измерений стабилен;

- требуется широкое использование встроенных функций для выполнения вычислений над ячейками гиперкуба, в том числе написание пользовательских функций.

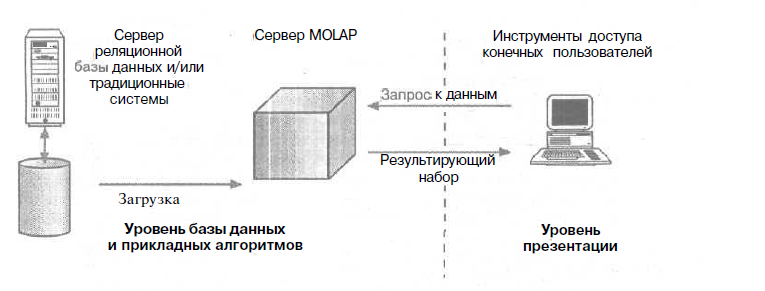


Рисунок 4.3.Архитектуратипичныхинструментов MOLAP.

1. HOLAP (hybrid OLAP) использует реляционные таблицы для хранения базовых данных и многомерные таблицы для хранения агрегированных данных.

В HOLAP-системах структура куба и предварительно обработанные агрегаты хранятся в многомерной базе данных. Это позволяет обеспечить быстрое извлечение агрегатов из структур MOLAP. Значения нижнего уровня иерархии в HOLAP остаются в реляционной витрине данных, которая служит источником данных для куба.

HOLAP не требует копирования листовых данных из витрины, хотя это и ведет к увеличению времени доступа при обращении к листовым данным. Данные в витрине доступны аналитику сразу после обновления. Таким образом, HOLAP-системы не вносят запаздывания в работу с данными нижнего уровня иерархии. По сути, HOLAP жертвует скоростью доступа к листовым данным ради устранения запаздывания при работе с ними и ускорения загрузки данных. В связи с этим HOLAP проигрывает по скорости MOLAP.

Преимущества и недостатки HOLAP • Преимуществом данной системы является обеспечение возможности связи с огромными наборами данных в реляционных таблицах и прирост производительности за счет использования многомерных хранилищ. • Недостаток состоит том, что количество проводимых преобразований между ROLAP и MOLAP системами может существенно влиять на общую эффективность.

2. ROLAP работает напрямую с реляционным хранилищем, факты и таблицы с измерениями хранятся в реляционных таблицах, и для хранения агрегатов создаются дополнительные реляционные таблицы. При этом размер хранилища не критичен, реляционная БД обеспечивает значительно больший уровень защиты данных и дает хорошие возможности разграничения прав доступа.

Основной недостаток ROLAP – невысокая производительность, однако при тщательной настройке ROLAP может быть приближена по скорости работы к MOLAP.

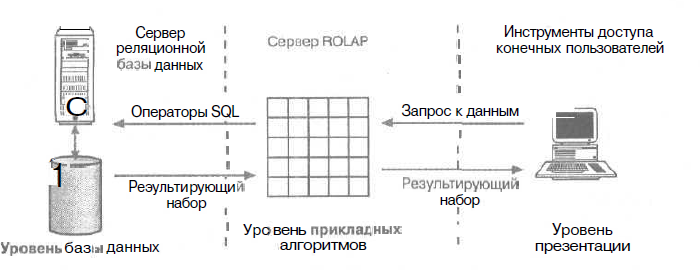


Рисунок 4.4. Архитектура реляционных инструментов OLAP.

Разработан и реализован вариант системы с использованием ROLAP хранилища данных, позволяющего производить оперативно-аналитический анализ данных, а также с использованием программного обеспечения для интеллектуального анализа данных.

Поскольку большинство корпоративных систем используют реляционную модель хранения данных, то для хранения данных в данной конфигурации выбрана система управления реляционными базами данных MySQL (версия 5.5). Данная система выбрана с учетом простоты ее настройки и поддержки. Тем не менее, она может быть легко заменена на более новые ее аналоги, например, Maria DB или Postgre SQL.

Рассмотрим структуру хранилища данных.

В первую очередь, необходимо отметить, что для OLAP-хранилища данных необходим специальный OLAP-сервер. Есть несколько вариантов свободно распространяемый OLAP-серверов, но в данном конкретном случае используется Mondrian OLAP-Server, от компании Pentaho. Выбор именно этого сервера обусловлен моделью хранения данных: поскольку используется реляционная база данных, необходим сервер, поддерживающий ROLAP модель хранения данных. Mondrian данную модель поддерживает.

Для запуска Mondrian, необходим веб-сервер java-приложений. В качестве такого сервера выбран свободно распространяемый проект Apache Tomcat (ранее Catalina).

Для удобства построения запросов в хранилище данных необходим OLAP-клиент. В данной конфигурации используется плагин Saiku (Community Edition). Он имеет удобный графический интерфейс пользователя с возможностью визуального формирования запроса. Это позволяет любому желающему с легкостью составить любой запрос к хранилищу данных.

В дополнение к оперативно-аналитическому анализу данных, планируется интеллектуальный (статистический) анализ данных. Для этого в данной конфигурации решено использовать язык R.

Для удобства работы с этим языком используется среда разработки R-Studio. Для быстрого доступа к интерфейсу среды через веб-браузер используется R-Studio Server. Он устанавливается на основной сервер системы анализа данных.

Но также возможно использование клиентского приложения RStudio Desktop. Оно устанавливается на клиентскую машину и предоставляет свой интерфейс для работы. Общая структура рассмотренной системы анализа данных представлена на рисунке 4.5.

/Users/jlewka/Documents/MIREA/Master/MasterWork/Структура хранилища данных.png

Рисунок 4.5. ROLAP хранилище данных на базе MySQL

В настоящее время распространены две основные схемы реализации многомерного представления данных с помощью реляционных таблиц: схема «звезда» (рис.4.4) и схема «снежинка» (рис. 4.5).

* 1. Схемы для реализации многомерного представления данных

Существуют 2 схемы для реализации многомерного представления данных с помощью таблиц:

Звезда – каждое измерение хранится в одной таблице (рис. 4.6).

Снежинка – одно измерение может храниться в нескольких таблицах (рис. 4.7).



Рисунок 4.6. Схема «звезда».

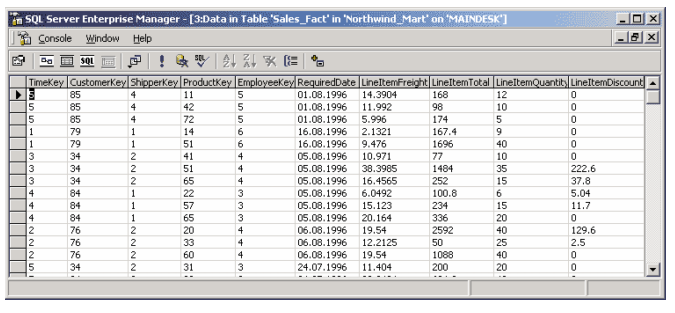


Рисунок 4.7. Схема «снежинка».

Каждая схема в центре имеет таблицу фактов, которая имеет связи с таблицами измерений типа «один» - «ко»-«многим». Основой для хранилища данных является денормализованная модель, составляющими которой являются:

- таблица фактов (fact table) – рисунок 4.8;

- таблицы измерений (dimension tables) – рисунок 4.9.



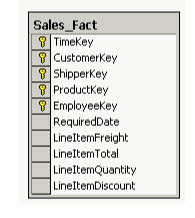


Рисунок 4.8. Пример таблицы фактов.

Таблица фактов является основной таблицей хранилища данных и как правило, содержит уникальный составной ключ, объединяющий первичные ключи таблиц измерений. Как правило, она содержит сведения об объектах или событиях, совокупность которых будет в дальнейшем анализироваться.

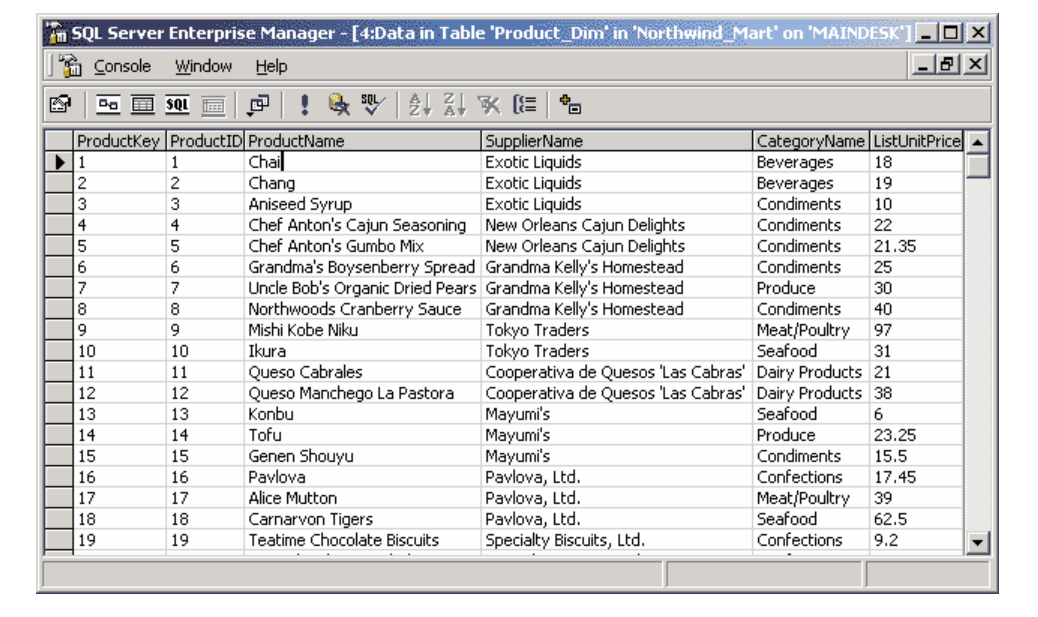
Обычно говорят о четырех наиболее часто встречающихся типах фактов. К ним относятся:

- факты, связанные с транзакциями (Transaction facts). Они основаны на отдельных событиях, типичными примерами которых являются телефонный звонок или снятие денег со счета с помощью банкомата;

- факты, связанные с «моментальными снимками» (Snapshot facts). Основаны на состоянии объекта (например, банковского счета) в определенные моменты времени, например, на конец дня или месяца. Типичными примерами таких фактов являются объем продаж за день или дневная выручка;

- факты, связанные с элементами документа (Line-item facts). Основаны на том или ином документе (например, счете за товар или услуги) и содержат подробную информацию об элементах этого документа (например, количестве, цене, проценте скидки);

- факты, связанные с событиями или состоянием объекта (Event or state facts). Представляют возникновение события без подробностей о нем (например, просто факт продажи или факт отсутствия таковой без иных подробностей).



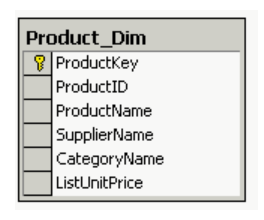


Рисунок 4.9. Пример таблицы измерений.

Таблицы измерений содержат неизменяемые либо редко изменяемые данные. В подавляющем большинстве случаев эти данные представляют собой по одной записи для каждого члена нижнего уровня иерархии в измерении.

* 1. Управляемая среда запросов МОЕ

Концепция инструментов управляемой среды запросов (MQE) является относительно новой. Эти инструменты предоставляют ограниченные функции анализа, применяемые либо непосредственно в реляционных СУБД, либо с помощью промежуточного сервера MOLAP. Инструменты MQE передают данные из СУБД (непосредственно или с помощью сервера MOLAP) на настольный компьютер или локальный сервер в виде куба данных, который затем хранится, анализируется и сопровождается локально. Компании-разработчики считают, чтоэтатехнологияпозволяетупроститьинсталляциюиадминистрирование, а также уменьшить затраты на сопровождение. Типичная архитектура инструментов MQE показана на рис. 4.10.

При разработке инструментов MQE необходимо учитывать следующие особенности.

•архитектураприводиткзначительномуувеличениюизбыточностиданныхиможетвызватьпроблемыприработевсетяхсомногимипользователями;

•способностькаждогопользователясоздаватьсвоисобственныекубыданныхможетпривестикутратенепротиворечивостиданных;

•всхемеможетэффективносопровождатьсятолькоограниченноеколичестводанных.

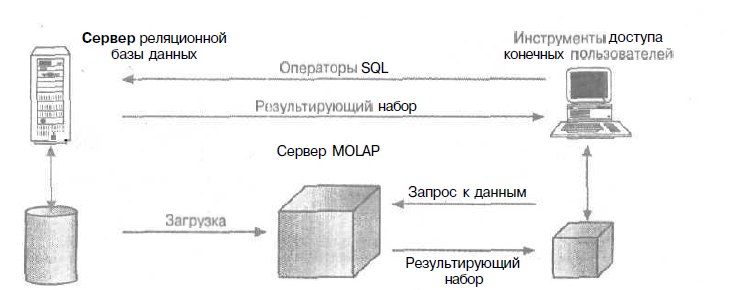


Рисунок 4.10. Архитектура реляционных инструментов MQE

# Список литературы

а) основная литература:

1. Макшанов, А.В. Технологии интеллектуального анализа данных [Электронный ресурс] : учебное пособие / А.В. Макшанов, А.Е. Журавлев. — Электрон. дан. — Санкт-Петербург : Лань, 2018. — 212 с. — Режим доступа: https://e.lanbook.com/book/109617. — Загл. с экрана.
2. Чубукова, И.А. Data Mining [Электронный ресурс]: учеб. пособие — Электрон. дан. — Москва: 2016. — 470 с. — Режим доступа: https://e.lanbook.com/book/100582. — Загл. с экрана.

б) дополнительная литература:

1. Введение в анализ данных: учебник / Б. Г. Миркин. — М.: Юрайт, 2015. — 174 с. — (Авторский учебник). — Библиогр.: с. 173-174 (27 назв.) (шифр в библиотеке МИРЭА: 004 М63).

2. Сидняев Н.И. Теория планирования эксперимента и анализ статистических данных : Учеб. пособие для вузов / Н. И. Сидняев. — М.: Юрайт, 2011. — 399 с.: ил. — (Магистр). — Библиогр.: с. 396-399 (50 назв.) (51 С34).

|  |  |  |
| --- | --- | --- |
| Конспект лекций по дисциплине разработан |  | Старший преподаватель кафедры ИППО  Богорадникова А.В. |
|  | *(подпись)* | *(степень, звание, должность, Фамилия И.О. разработчиков)* |