Базы данных

Модуль 3. Перспективные направления развития систем обработки данных. Обзор современных СУБД

Лекция 14. Системы поддержки принятия решений. Хранилища данных и системы анализа данных.

Преподаватель:
Семенов Геннадий Николаевич, к.т.н., доцент

2023

Два класса оперативных ИС

Среди фактографических ИС важное место занимают два класса: системы ***операционной обработки данных*** и системы, ориентированные на ***анализ данных и поддержку принятия решений***

В современной литературе эти классы систем обозначаются:

OLTP - On-Line Transaction Processing - оперативная обработка транзакций;

OLAP - On-Line Analysis Processing - оперативная аналитическая обработка

Характеристики OLTP-системы:

- рассчитаны на быстрое обслуживание относительно простых запросов большого числа пользователей;
- требуют защиты от несанкционированного доступа, нарушений целостности, аппаратных и программных сбоев;
- время ожидания выполнения типичных запросов не должно превышать нескольких секунд;
- логическая единица функционирования - транзакция;
- сфера применения - банковские и биржевые системы, резервирование мест.

Аналитическая информационная система

OLAP-ИС предназначены для выполнения запросов, требующих статистической обработки накопленных в течение времени данных, моделирования процессов предметной области, прогнозирования развития явлений.

Эти системы оперируют большими объемами данных и позволяют выделить из них содержательную информацию, т.е. получить из данных ***знания***.

Пример запроса в OLTP-ИС: «Есть ли свободные места в купе поезда Москва-Сочи, отправляющегося 6 июля в 20.35?»

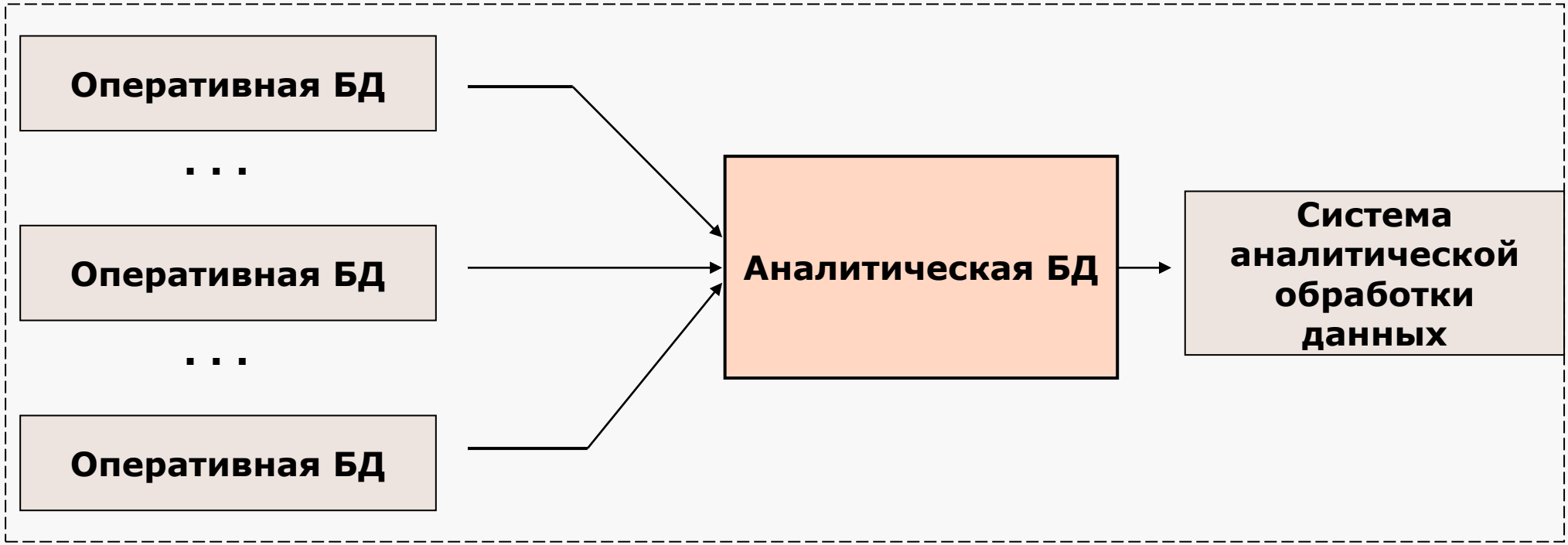
Пример запроса в OLAP-ИС: «Каким будет объем продажи железнодорожных билетов в следующем квартале с учетом сезонных колебаний?»

Пример запроса в OLTP-ИС: «Каков размер счета клиента Иванова И.И.?»

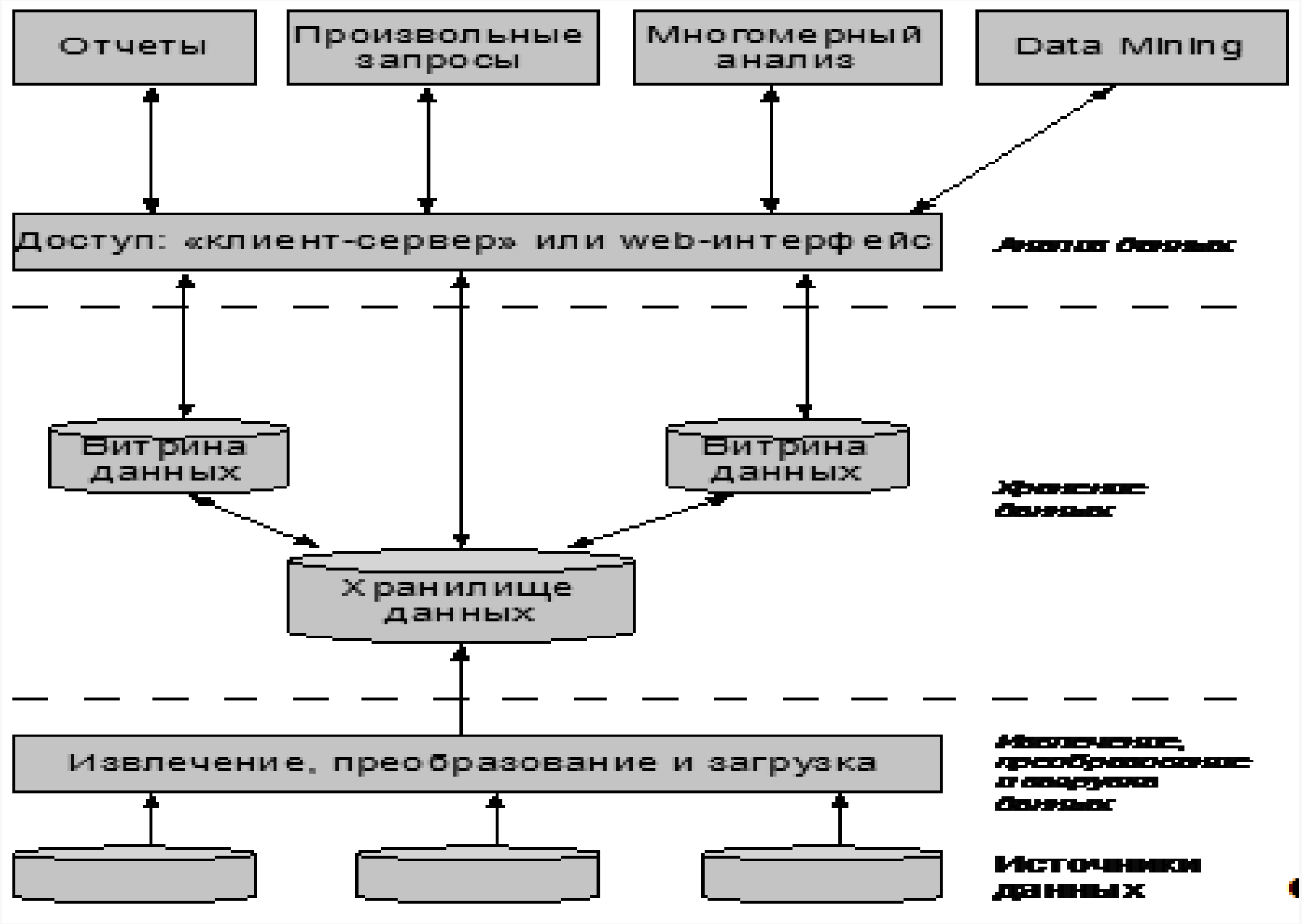
Пример запроса в OLAP-ИС: «Найти среднее значение промежутка времени между выставлением счета и оплатой его клиентом в текущем и прошедшем году отдельно для разных групп клиентов.»

Аналитическая информационная система

Компоненты:
- **Оперативные БД** как источники информации
- **Аналитическая БД** для сбора, хранения и предоставления необходимой информации
- **Система аналитической обработки** данных для их анализа(приложения)



Архитектура корпоративной OLAP-системы



Системы поддержки принятия решений

С помощью OLAP-ИС можно получать сведения как об обслуживаемой организации, так и о сфере ее деятельности.

Данные, накапливаемые в массивах данных, содержат скрытые закономерности, из которых можно вывести правила функционирования предметной области, моделируемой информационной системой.

Эти правила могут быть использованы для стратегического планирования, принятия решений и прогнозирования их последствий.

Человеко-машинный вычислительный комплекс, ориентированный на анализ данных для получения информации, необходимой при разработке решений в сфере управления, называют ***системой поддержки принятия решений*** (***СППР***).

Задачи, традиционно решаемые с помощью СППР:

- оценка альтернатив решений;
- прогнозирование;
- классификация;
- кластеризация;
- выявление ассоциаций;
- и т.д.

Аналитический запрос невозможно сформулировать в терминах языка SQL. Разработаны специализированные языки, например Express 4GL фирмы Oracle.

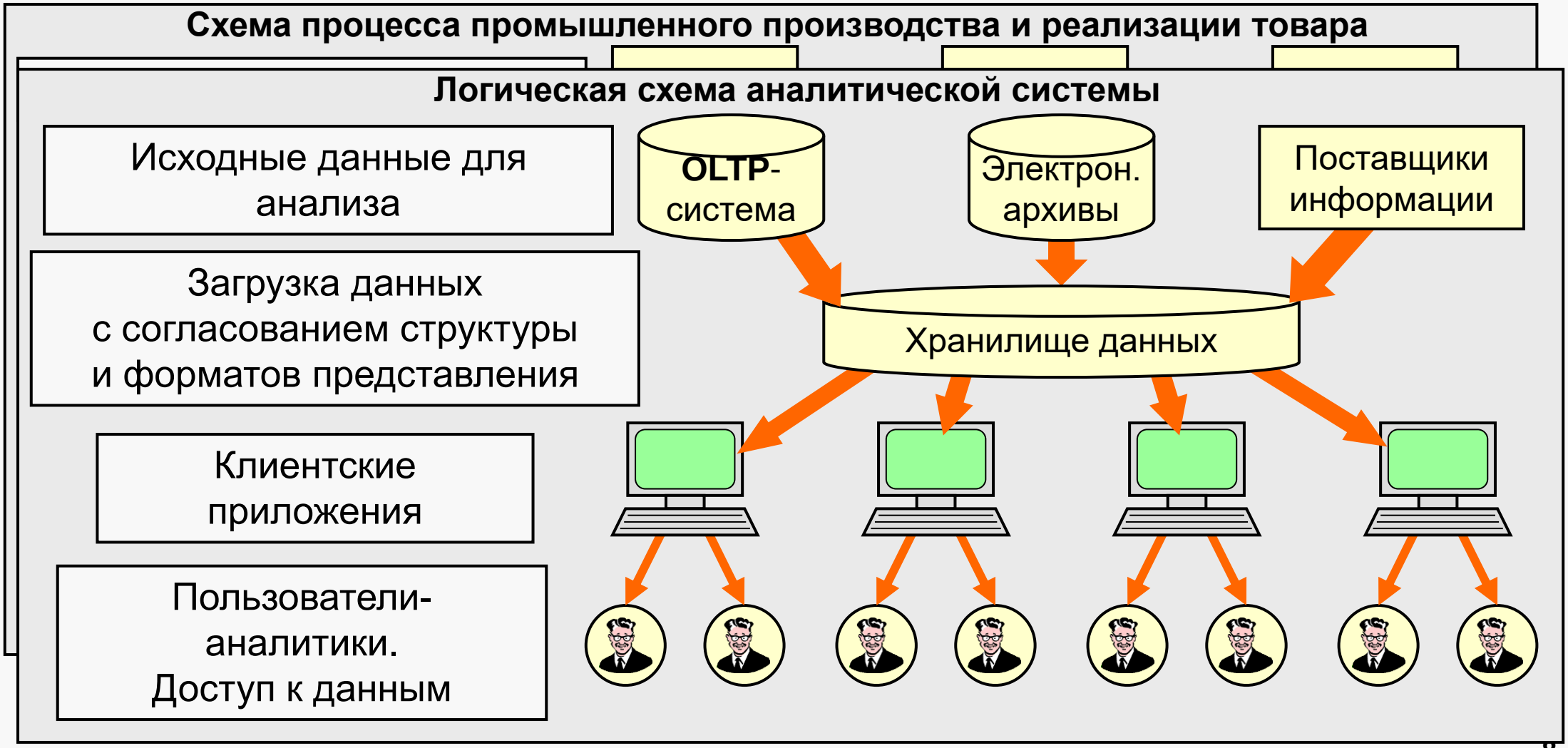
Свойства данных в OLTP и СППР

Данные, их свойства и принципы хранения
в системах операционной обработки и СППР различны.

Свойство	OLTP	СППР
Назначение данных	Оперативный поиск, несложные виды обработки	Аналитическая обработка, прогнозирование, моделирование
Уровень агрегации	Детализированные данные	Агрегированные данные
Период хранения	От нескольких месяцев до одного года	От нескольких лет до десятков лет
Частота обновления	Высокая, малыми порциями	Малая, большими порциями

Функционирование СППР

Процедура анализа данных имеет много общего с процессом изготовления промышленной продукции.



Концепция хранилища данных (ХД)

Оперативность обработки больших объемов данных достигается за счет применения мощных многопроцессорных компьютеров и специальных ***хранилищ данных - Data Warehouse.***

Билл Инмон: «Хранилище - предметно-ориентированный, интегрированный, неизменяемый и поддерживающий хронологию набор данных, предназначенный для обеспечения принятия управленческих решений.

Функциональные особенности хранилища данных (ХД):

- накопление данных за большой период времени;
- накопление данных из различных источников информации;
- обеспечение быстрого доступа к данным.

- при выполнении аналитических запросов обрабатываются большие инфор-мационные массивы. Реляционная БД обрабатывается относительно медленно;
- не происходит обновление данных - производится лишь накопление;
- требование хронологической упорядоченности данных;
- чаще используются не детальные, а обобщенные и агрегированные данные.

Цели концепции ХД в СППР

Использование концепции хранилища данных
в системе поддержки принятия решений
преследует определенные цели.

Цели:

- обеспечение аналитиков информацией для выработки решений;
- создание единой модели данных организации;
- создание интегрированного источника данных, предоставляющего:
  - доступ к разнородной информации;
  - гарантию получения одинаковых ответов на одинаковые вопросы.

В СППР критерии поиска и состав выдаваемой в виде отчета информации не фиксируются при разработке концепции ХД, в отличие от OLTP.

Пользователи оперируют заранее не регламентированными запросами.

Свойства хранилища данных

Свойства, присущие хранилищам данных,
следуют из определения Билла Инмона.

1. Ориентация на предметную область.

- ХД разрабатывается с учетом специфики предметной области, а не приложений, оперирующих данными.
- Структура ХД должна отражать представление аналитика об информации, с которой ему приходится работать.

2. Интегрированность.

- Единый синтаксический и семантический вид данных разных приложений.
- Проверка поступающих данных на целостность и непротиворечивость.
- Данные агрегируются при загрузке, а не группируются при запросе.

3. Неизменяемость данных.

- Единственное изменение - добавление записей.
- Нет проблем с откатом транзакций, снятием взаимных блокировок процессов
- Главная решаемая задача - высокая скорость доступа к данным.
- Использование надежного оборудования для обеспечения защиты от сбоев.

4. Поддержка хронологии.

- Аналитический запрос - анализ тенденции развития явлений во времени.

Задачи создания ХД

Загрузка данных в СППР на основе ХД выполняется сравнительно редко, но большими порциями -
до нескольких миллионов записей за один раз.

Основные задачи, решаемые при создании ХД:

- выбор структуры хранения данных, оптимальной по времени отклика на аналитические запросы и требуемого объема памяти;
- начальное заполнение и последующее пополнение хранилища данными;
- обеспечение удобства доступа пользователей к данным.

Успешное решение этих задач зависит от:

- выбора модели данных;
- архитектуры хранилища;
- регламента поступления данных в хранилище;
- методов аналитической обработки данных в хранилище.

Модели данных для построения ХД

Особенность накапливаемых в ХД данных определяет способ их представления и организации.

Данные в ХД чаще всего содержат сведения о значениях некоторых параметров, характеризующих предметную область:

- в определенные ***моменты времени;***
- за определенные ***промежутки времени.***

***Пример*:** информация об изменении социально-экономической обстановки в России собирается Госкомстатом из субъектов РФ:

- помесячно, поквартально, за год;
- объем производства, индекс потребительских цен и т.д.

Для организации таких БД используются модели:

- многомерная модель БД - ***MOLAP*** - Multidimentional OLAP;
- реляционная модель БД - ***ROLAP*** - Relational OLAP.
- гибридная модель (***HOLAP***) Hybrid OLAP,

Конкурирующие и взаимодополняющие модели данных.

Хранение активных данных в многомерной БД

- В этом случае данные **OLAP** хранятся в **многомерных СУБД**, использующих оптимизированные для такого типа данных конструкции. Обычно многомерные СУБД поддерживают и все типовые для **OLAP** операции, включая **агрегацию** по требуемым уровням иерархии и т.д.
- Этот тип хранения данных в каком-то смысле можно назвать классическим для **OLAP**. Для него необходимы все шаги по **предварительной подготовке данных**.
- Обычно данные многомерной **СУБД** хранятся на диске, однако, в некоторых случаях, для ускорения обработки данных такие системы позволяют хранить данные в оперативной памяти. Для тех же целей иногда применяется и хранение в БД заранее рассчитанных **агрегатных значений** и прочих расчётных величин.
- Среди условных недостатков, характерных для некоторых реализаций многомерных **СУБД** и базирующихся на них **OLAP-систем** можно отметить их подверженность непредсказуемому с пользовательской точки зрения росту объёмов занимаемого БД места.
- Этот эффект вызван желанием максимально уменьшить время реакции системы, диктующим хранить заранее рассчитанные значения агрегатных показателей и иных величин в БД, что вызывает **нелинейный рост объёма** хранящейся в БД информации с добавлением в неё новых значений данных или измерений.

Хранение активных данных в реляционной БД

- Могут храниться данные **OLAP** и в традиционной РСУБД. В большинстве случаев этот подход используется при попытке «безболезненной» интеграции OLAP с существующими учётными системами, либо базирующимися на РСУБД хранилищами данных. Вместе с тем, этот подход требует от **РСУБД** для обеспечения эффективного выполнения требований **FASMI-теста** (в частности, обеспечения минимального времени реакции системы) некоторых дополнительных возможностей. Обычно данные **OLAP** хранятся в денормализованном виде, а часть заранее рассчитанных агрегатов и значений хранится в специальных таблицах. При хранении же в нормализованном виде эффективность **РСУБД** в качестве метода хранения активных данных снижается.
- Проблема выбора эффективных подходов и алгоритмов хранения предрассчитанных данных также актуальна для **OLAP-систем**, базирующихся на РСУБД, поэтому производители таких систем обычно акцентируют внимание на достоинствах применяемых подходов.
- В целом считается, что базирующиеся на **РСУБД** OLAP-системы медленнее систем, базирующихся на многомерных СУБД, в том числе за счет менее эффективных для задач **OLAP** структур хранения данных, однако на практике это зависит от особенностей конкретной системы.
- Среди достоинств хранения данных в **РСУБД** обычно называют большую масштабируемость таких систем.

Гибридный подход к хранению данных

- Большинство производителей **OLAP-систем**, продвигающих свои комплексные решения, часто включающие помимо собственно **OLAP-системы** СУБД, инструменты **ETL (Extract Transform Load)** и отчетности, в настоящее время используют гибридный подход к организации хранения активных данных системы, распределяя их тем или иным образом между **РСУБД** и специализированным хранилищем, а также между дисковыми структурами и кэшированием в оперативной памяти.
- Так как эффективность такого решения зависит от конкретных подходов и алгоритмов, применяемых производителем для определения того, *какие данные и где хранить*, то поспешно делать выводы о изначально большей эффективности таких решений как класса без оценки конкретных особенностей рассматриваемой системы.

Хранение активных данных в «плоских» файлах

- Этот подход предполагает хранение порций данных в обычных файлах.
- Обычно он используется как дополнение к одному из двух основных подходов с целью ускорения работы за счет кэширования актуальных данных на диске или в оперативной памяти клиентского ПК.

OLAP (англ. on-line analytical processing)

- **OLAP** (англ. on-line analytical processing) – совокупность методов динамической обработки многомерных запросов в аналитических базах данных.
- Такие источники данных обычно имеют довольно большой объем, и в применяемых для их обработки средствах одним из наиболее важных требований является высокая скорость.
- В реляционных БД информация хранится в отдельных таблицах, которые хорошо нормализованы. Но сложные многотабличные запросы в них выполняются довольно медленно.
- Значительно лучшие показатели по скорости обработки в **MOLAP** - системах достигаются за счет особенности структуры хранения данных.
- Вся информация здесь четко организована, и применяются два типа хранилищ данных: ***измерения*** (содержат справочники, разделенные по категориям, например, точки продаж, клиенты, сотрудники, услуги и т.д.) и ***факты*** (характеризуют взаимодействие элементов различных измерений,

Основные понятия многомерной модели данных

- **Показатель** - это величина (обычно числового типа), которая собственно и является предметом анализа. Один OLAP-куб может обладать одним или несколькими показателями.
- **Измерение** (dimension) - это множество объектов одного или нескольких типов, организованных в виде иерархической структуры и обеспечивающих информационный контекст числового показателя. Измерение принято визуализировать в виде ребра многомерного куба.
- Объекты, совокупность которых и образует измерение, называются **членами измерений** (members). Члены измерений визуализируют как точки или участи, откладываемые на осях гиперкуба. Например, временное измерение: Дни, Месяцы, Кварталы, Годы - наиболее часто используемые в анализе, могут содержать следующие члены: 8 мая 2002 года, май 2002 года, 2-ой квартал 2002 года и 2002 год. Объекты в измерениях могут быть различного типа, например "производители" - "марки автомобиля" или "годы" - "кварталы". Эти объекты должны быть организованы в иерархическую структуру так, чтобы объекты одного типа принадлежали только одному уровню иерархии.
- **Ячейка** (cell) - атомарная структура куба, соответствующая конкретному значению некоторого показателя. Ячейки при визуализации располагаются внутри куба и здесь же принято отображать соответствующее значение показателя.

Роль измерений в кубе

- **Измерения** играют роль **инёдексов**, используемых для идентификации значений показателей, находящихся в ячейках гиперкуба.
- **Комбинация** членов различных измерений играют роль координат, которые определяют значение определенного показателя.
- Поскольку для куба может быть определено несколько показателей, то комбинация членов всех измерения будет определять несколько ячеек со значениями каждого из показателей.
- Поэтому для однозначной идентификации ячейки необходимо указать комбинацию членов всех измерений и показатель.

- **Агрегатами** называют агрегированные по определенным условиям исходные значения показателей. Обычно под агрегацией понимается любая процедура формирования меньшего количества значений (агрегатов) на основании большего количества исходных значений (чаще это сумма).
- Заблаговременное формирование и сохранение агрегатов с целью уменьшения времени отклика на пользовательский запрос является основным свойством систем поддержки оперативного анализа.

Многомерная модель ХД (MOLAP)

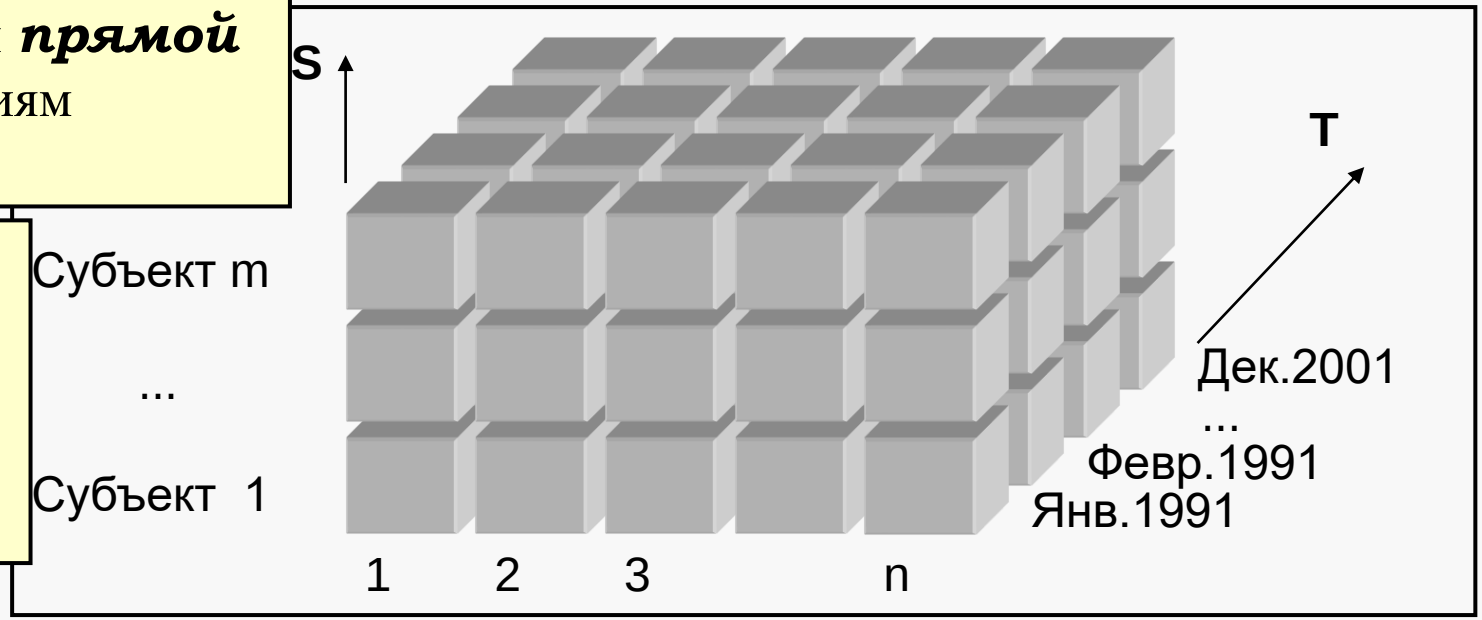
В многомерной модели данные представляются в виде упорядоченного многомерного массива – ***гиперкуба (OLAP-куба)***.

Пример: индекс потребительских цен в Москве в декабре 1996 года был равен 101% - задается точкой в трехмерном пространстве (N,S,T), где:

- N - наименование параметра - *индекс потребительских цен*;
- S - субъект федерации - *Москва*;
- T - момент времени - *декабрь 1996 года*.

В этой модели обеспечивается ***прямой доступ*** к данным по значениям координат пространства.

Среднее время ответа на сложный аналитический запрос в 10-100 раз мень-ше, чем в реляционной СУБД с нормализованной структурой.



Число измерений теоретически не ограничено.

Характеристики многомерной модели ХД

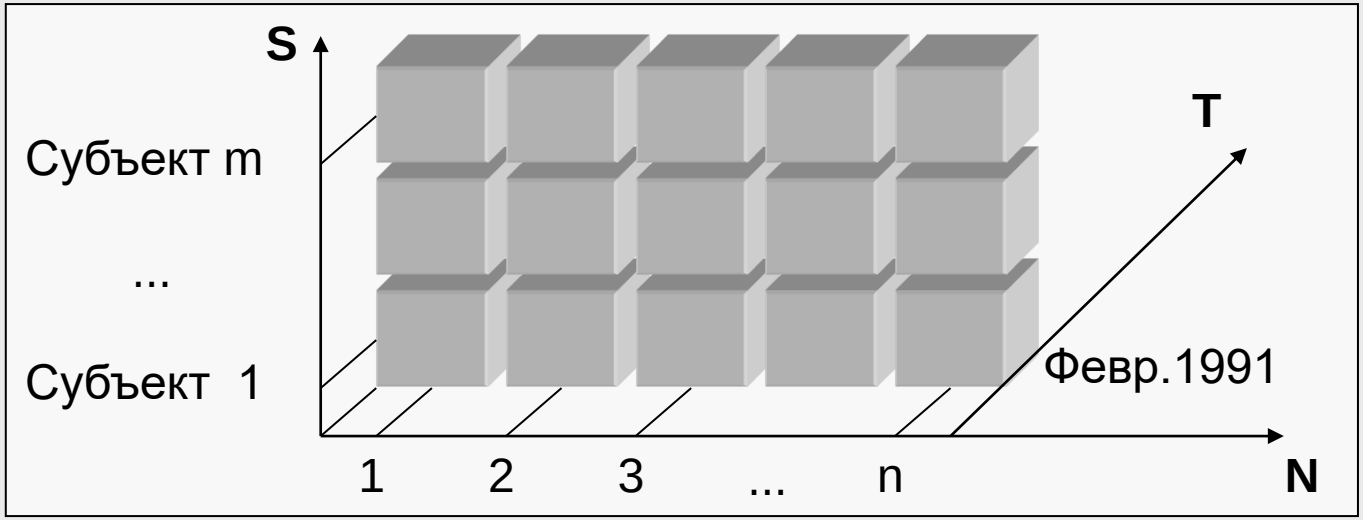
Основные понятия многомерной модели - ***измерение*** и ***значение***.

Измерение - это множество, образующее грань гиперкуба. Оно является индексом для идентификации значений в ячейках гиперкуба.

Значение - количественное или качественное данное в ячейке гиперкуба.

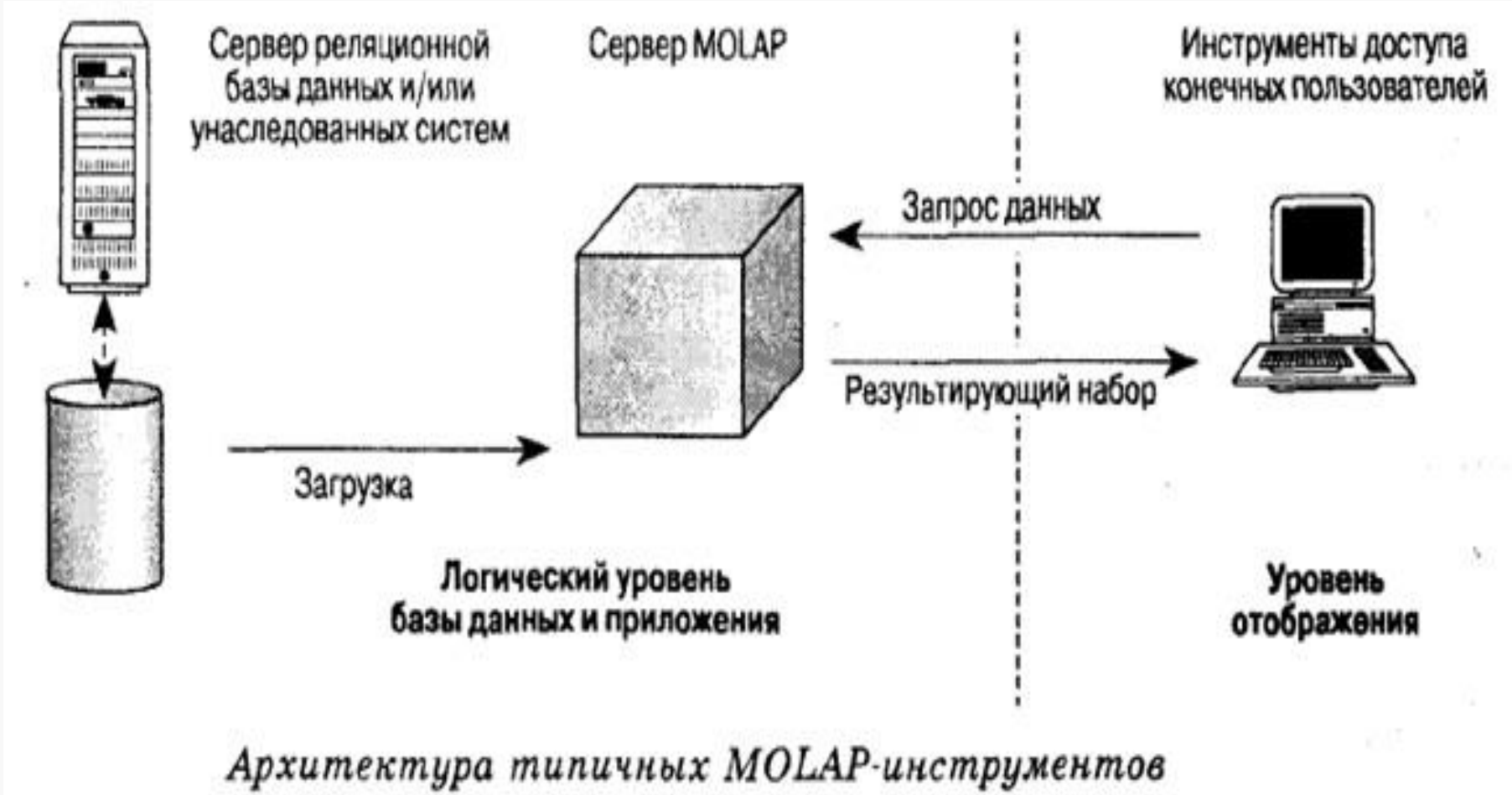
Операции манипулирования измерениями:

- *сечение* - формирование подмножества с фиксированным измерением;
- *вращение* - изменение порядка представления измерений;
- *свертка* - замена значения значением более высокого уровня иерархии;
- *детализация* - переход от обобщенных данных к детализированным.



Многомерные СУБД неэффективно используют память.

Типичная архитектура MOLAP



Архитектура типичных MOLAP-инструментов

Преимущества MOLAP

- Высокая производительность запросов благодаря оптимизированному хранилищу, многомерному индексированию и кэшированию.
- Меньший размер данных на диске по сравнению с данными, хранящимися в реляционной базе данных, благодаря методам сжатия.
- Автоматизированное вычисление агрегированных данных более высокого уровня.
- Он очень компактен для наборов данных малой размерности.
- Массивные модели обеспечивают естественную индексацию.
- Эффективное извлечение данных достигается за счет предварительного структурирования агрегированных данных.

Недостатки MOLAP

- В некоторых системах MOLAP этап обработки (загрузка данных) может быть довольно длительным, особенно при больших объемах данных.
- Обычно это исправляется путем выполнения только инкрементной обработки, то есть обработки только тех данных, которые изменились (обычно новых данных), вместо повторной обработки всего набора данных.
- Некоторые методологии MOLAP вводят избыточность данных.

Продукты:

Примерами коммерческих продуктов, использующих **MOLAP**, являются Cognos Powerplay, Oracle Database OLAP Option, MicroStrategy, Microsoft Analysis Services, Essbase, TM1, Jedox и icCube.

Реляционная модель ХД (ROLAP)

- **ROLAP** работает непосредственно с реляционными базами данных и не требует предварительных вычислений. Базовые данные и таблицы измерений хранятся в виде реляционных таблиц, а для хранения агрегированной информации создаются новые таблицы. Это зависит от специализированной схемы проектирования.
- Эта методология основана на манипулировании данными, хранящимися в реляционной базе данных, чтобы создать видимость традиционной функциональности OLAP для нарезки и нарезки кубиками. По сути, каждое действие нарезки и нарезки кубиками эквивалентно добавлению предложения "WHERE" в инструкцию SQL.
- Инструменты ROLAP не используют предварительно рассчитанные кубы данных, а вместо этого представляют запрос к стандартной реляционной базе данных и ее таблицам, чтобы получить данные, необходимые для ответа на вопрос. Инструменты ROLAP позволяют задавать любые вопросы, поскольку методология не ограничивается содержимым куба. ROLAP также имеет возможность детализации до самого низкого уровня детализации в базе данных.
- База данных, которая была разработана для OLTP, не будет хорошо функционировать как база данных ROLAP. Таким образом, ROLAP по-прежнему предполагает создание дополнительной копии данных. Однако, поскольку это база данных, для заполнения базы данных могут использоваться различные технологии.

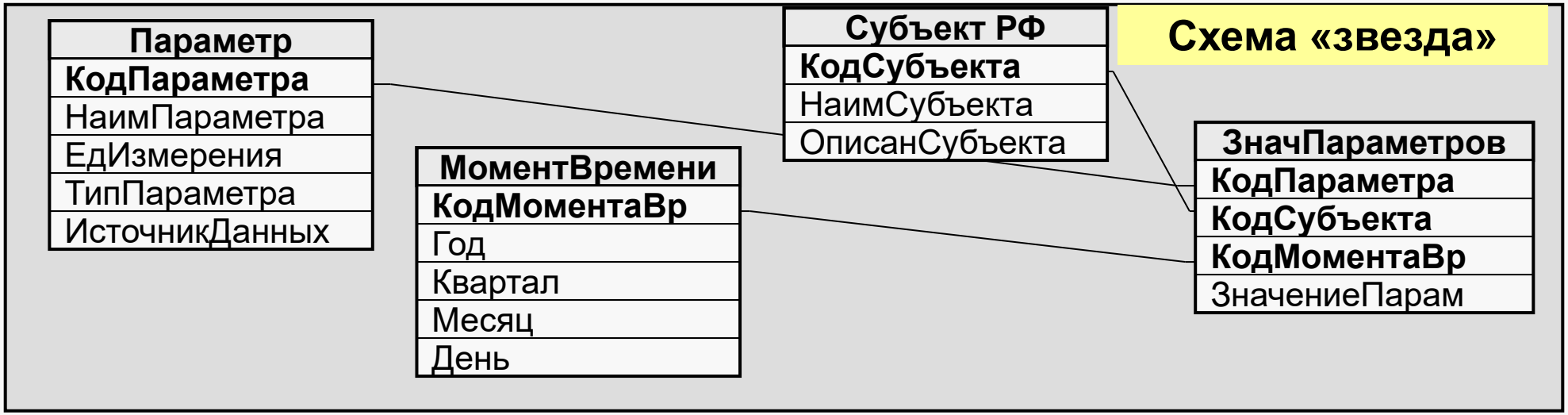
Реляционная модель ХД (ROLAP)

В реляционной модели для организации данных используется два типа таблиц:
таблицы измерений и ***фактологическая таблица***.

Запись фактологической таблицы соответствует ячейке куба (до 1 млрд. зап.).

Таблица измерений содержит все значения соответствующего измерения куба.

Фактологическая таблица индексируется по составному ключу, скомпонован-ному из первичных ключей таблиц измерений.



Объем данных в реляционной СУБД не ограничен.

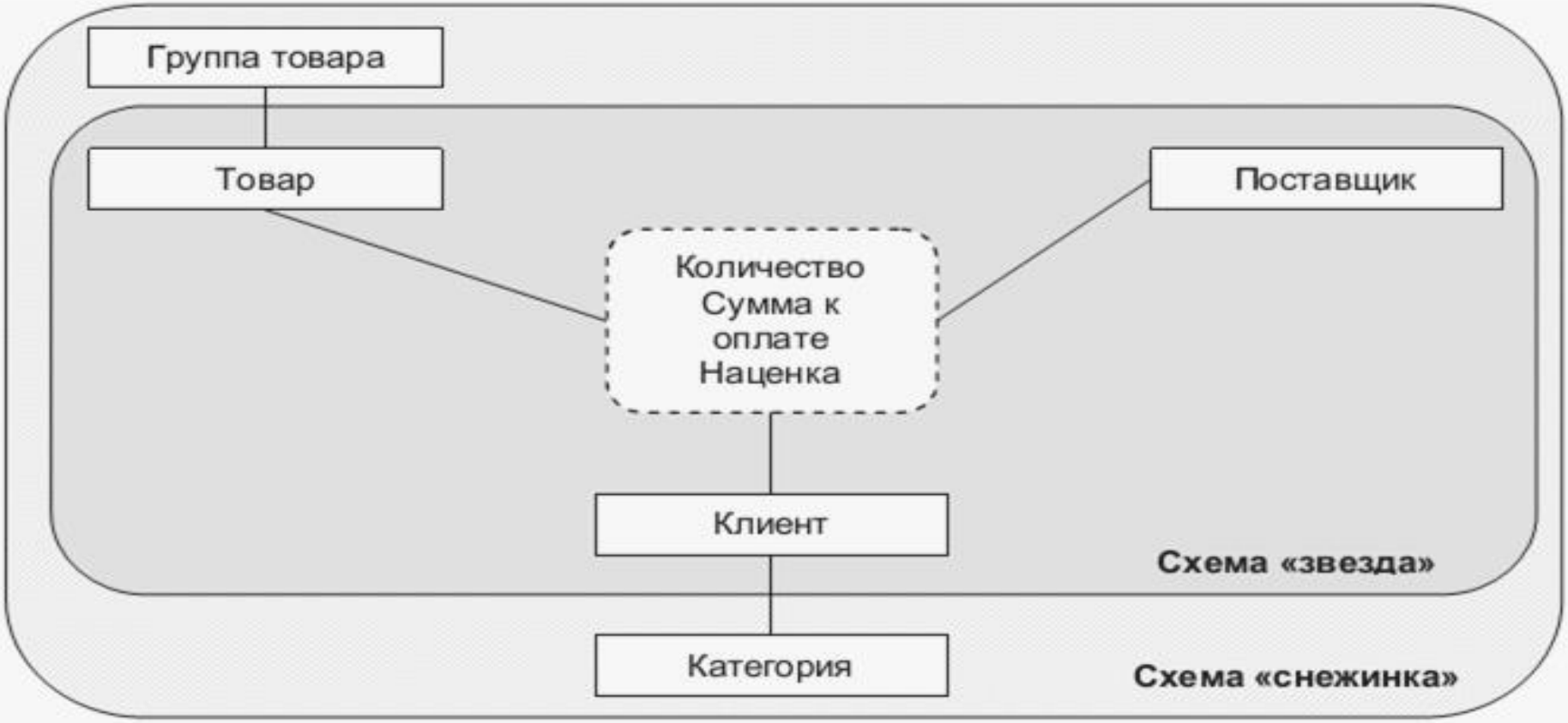
Основные составляющие структуры хранилищ данных

- Схема звезда обычно содержит одну большую таблицу, называемую таблицей **факта (*fact table*)**, помещенную в центр, и окружающие ее меньшие таблицы,
- называемые **таблицами размерности (*dimensional table*)**, соединенные с таблицей факта в виде звезды радиальными связями. В этих связях таблицы размерности являются родительскими, таблица факта - дочерней.
- Схема звезда может иметь также **консольные таблицы *(outrigger table)***, присоединенные к таблице размерности. Консольные таблицы являются родительскими, таблицы размерности - дочерними.

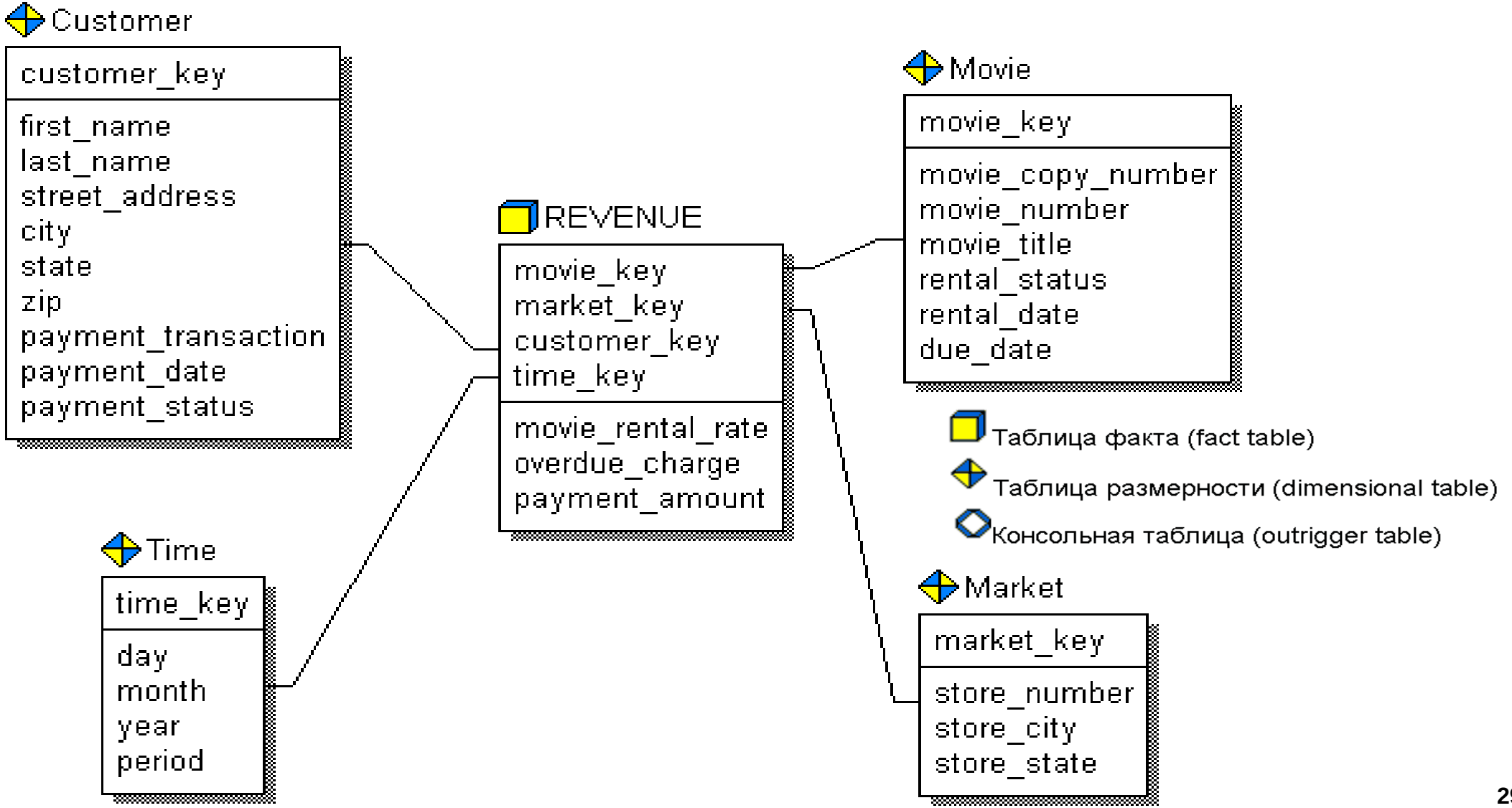
Схема «снежинка»

В схеме«звезда» измерение может ссылаться только на таблицу **фактов**, а в«снежинке» измерение может ссылаться на другие измерения, которые в свою очередь ссылаются на таблицу фактов. «звезда» – это частный случай схемы«снежинка».

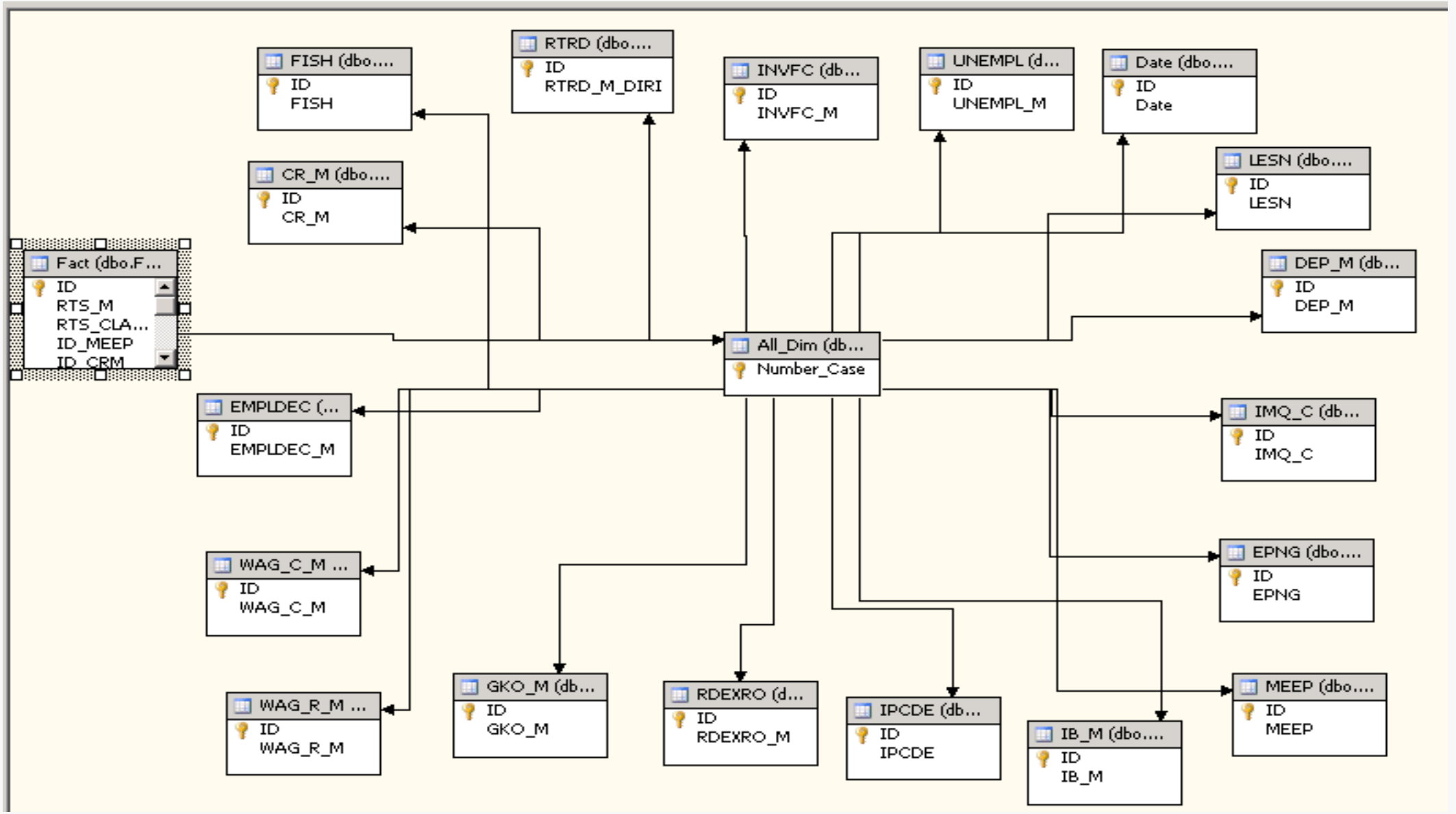
Таблицы **размерностей** нормализованы, что снижает избыточность информации и ускоряет выполнения некоторых запросов.



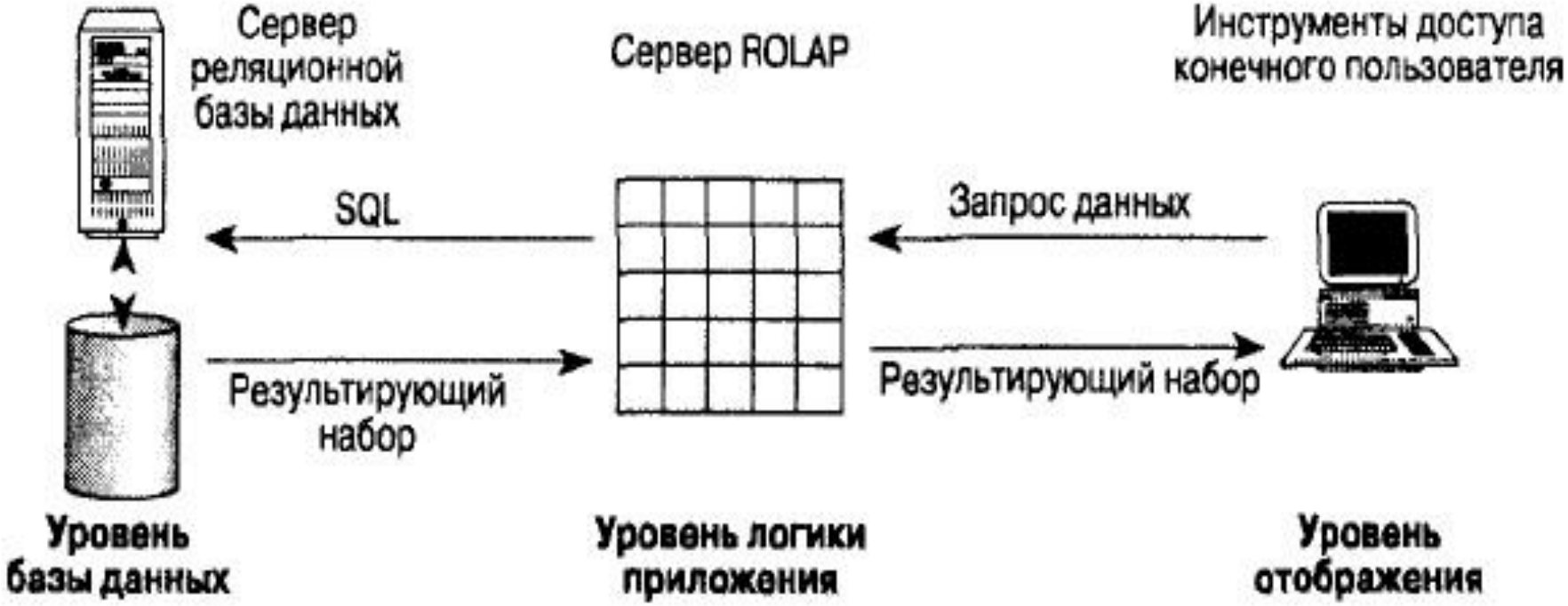
Первичный ключ (таблица факта “REVENUE”) составлен из четырех внешних ключей: movie_key, market_key, customer_key и time_key



Структура ХД - снежинка



Типичная архитектура реляционных OLAP



Архитектура реляционных OLAP-инструментов

Преимущества ROLAP

- **ROLAP** считается более масштабируемым при обработке больших объемов данных, особенно моделей с размерами с очень высокой мощностью
- Благодаря разнообразию доступных инструментов загрузки данных и возможности точной настройки кода извлечения, преобразования, загрузки (ETL) для конкретной модели данных, время загрузки обычно намного короче, чем при автоматической загрузке **MOLAP**.
- Данные хранятся в стандартной реляционной базе данных и могут быть доступны с помощью любого инструмента SQL (инструмент не обязательно должен быть инструментом **OLAP**).
- Инструменты **ROLAP** лучше справляются *с неагрегируемыми фактами* (например, текстовыми описаниями). Инструменты **MOLAP**, как правило, страдают от низкой производительности при запросе этих элементов.
- **ROLAP-хранилище** данных, может успешно моделировать данные, которые в противном случае не вписывались бы в строгую многомерную модель.
- Подход **ROLAP** может использовать средства управления авторизацией базы данных, такие как безопасность на уровне строк, посредством чего результаты запроса фильтруются в зависимости от заданных критериев, применяемых, например, к данному пользователю или группе пользователей (предложение SQL **WHERE**).

Недостатки ROLAP

- В **IT**-отрасли существует консенсус в отношении того, что инструменты ROLAP имеют более низкую производительность, чем инструменты MOLAP.
- Загрузка *сводных таблиц* должна управляться пользовательским кодом **ETL**. Инструменты **ROLAP** не помогают с этой задачей. Это означает дополнительное время разработки и больше кода для поддержки.
- Когда этап создания сводных таблиц пропускается, производительность запросов снижается, поскольку необходимо запрашивать более подробные таблицы большего размера. Это можно частично исправить, добавив дополнительные сводные таблицы, однако по-прежнему нецелесообразно создавать сводные таблицы для всех комбинаций измерений / атрибутов.
- **ROLAP** полагается на базу данных общего назначения для запросов и кэширования, и поэтому некоторые специальные методы, используемые инструментами **MOLAP**, недоступны. Однако современные инструменты **ROLAP** используют последние усовершенствования языка **SQL**, такие как операторы **КУБА** и свертки, а также другие расширения **SQL OLAP**. Эти улучшения **SQL** могут уменьшить преимущества инструментов **MOLAP**.
- Поскольку инструменты **ROLAP** полагаются на **SQL** для всех вычислений, они не подходят, когда модель перегружена вычислениями, которые плохо переводятся в **SQL**. Примерами таких моделей являются составление бюджета, распределение средств, финансовая отчетность и другие сценарии.

Сравнительные характеристики

Характеристика	OLTP	ROLAP	MOLAP
1	2	3	4
Типовая операция	Обновление	Отчет	Анализ
Уровень аналитических требований	Низкий	Средний	Высокий
Экраны	Неизменяемые	Определяемые пользователем	Определяемые пользователем
Объем данных на транзакцию	Небольшой	От малого до большого	Большой
Уровень данных	Детальные	Детальные и суммарные	Суммарные
Сроки хранения данных	Текущие	Исторические и текущие	Исторические, текущие и прогнозируемые
Структурные элементы	Записи	Записи	Массивы

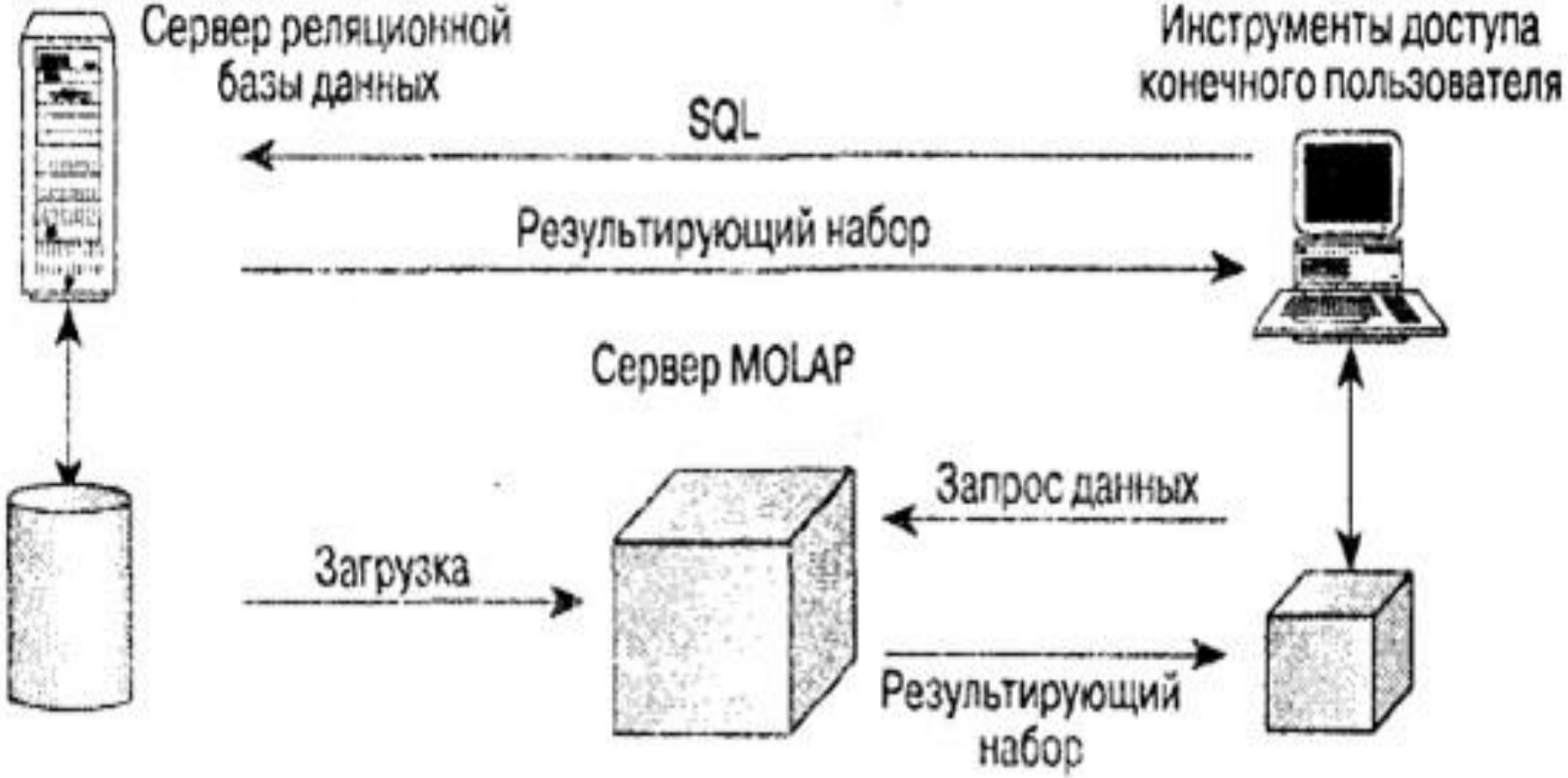
Управляемая среда запросов MQE

Концепция инструментов управляемой среды запросов (**MQE**) является относительно новой разработкой. Эти инструменты предоставляют ограниченные функции анализа либо непосредственно реляционным СУБД, либо с помощью промежуточного **MOLAP**-сервера

MQE-инструменты передают данные из СУБД (непосредственно или с помощью **MOLAP**-сервера) на настольный компьютер или локальный сервер в виде куба данных, который затем сохраняется, анализируется и сопровождается локально.

Эта технология характеризуется относительной простотой инсталляции и администрирования, а также меньшими затратами и усилиями на сопровождение.

Типичная архитектура MQE



Архитектура реляционных MQE-инструментов

Витрины данных

Витрина данных - комбинация многомерного и реляционного подходов к построению хранилищ данных.

Многомерная модель:

- позволяет производить быстрый анализ данных;
- не позволяет хранить большие объемы информации.

Реляционная модель:

- не обеспечивает должной скорости выполнения аналитических запросов.
- не имеет ограничений по объему накапливаемых данных;

Данные распределяют по тематическим хранилищам небольших объемов - ***витринами данных.***

Витрины данных реализуются в виде многомерных СУБД и выполняют роль мелких складов - ***Data Marts***.

Источником данных для витрин данных является централизованное **ХД** большого объема, представляющего собой реляционную СУБД.

Методы аналитич.обработки данных в ХД

Важная составная часть аналитических информационных систем - средства интеллектуального анализа данных (**Data Mining**).

Для обработки данных используется широкий спектр методов.

Традиционные статистические методы:

Регрессионный анализ;
Факторный анализ;
Дисперсионный анализ;
Анализ временных рядов.

Методы интеллектуального анализа данных (***data mining*** - добыча знаний):

Нейронные сети;
Нечеткая логика;
Генетические алгоритмы;
Методы извлечения знаний.

Методы интеллектуального анализа данных применяют, если невозможно использовать традиционные методы:

- при отсутствии точных зависимостей, описывающих анализируемые явления;
- для нахождения взаимосвязей в данных;
- для определения нелинейных зависимостей в данных;
- для выявления аномалий в данных - отклонений от общей закономерности.

Информационные системы на основе ХД

Информационные системы, использующие ХД, строятся на основе архитектуры клиент-сервер.

ХД размещается на специальном сервере - ***сервере ХД*** - мощной многопроцессорной вычислительной системе от фирм IBM, Hewlett-Packard, DEC, NRC и др.

Для них применяются СУБД, поддерживающие параллельную обработку запросов: Teradata (NCR), DB/2 (IBM), Oracle, Informix и др.

Витрины данных реализуются с использованием серверов многомерных БД: Essbase (Arbor Software), Oracle Express (Oracle), Gentium (Planning Sciences) и др.

В зависимости от объема используемых данных ХД подразделяются на:

Тип	Объем данных	Число строк в фактологической таблице
Маленькое	до 3 Гб	до нескольких миллионов
Среднее	до 25 Гб	до ста миллионов
Большое	до 200 Гб	несколько сотен миллионов
Сверхбольшое	свыше 200 Гб	миллиард и более

Приведен полезный объем. Общий объем ХД в 5-10 раз больше (индексы).

Другие типы OLAP

- **WOLAP** – веб-OLAP
- **DOLAP** – настольный OLAP
- **RTOLAP** – OLAP в реальном времени
- **GOLAP** – график OLAP
- **CaseOLAP** – контекстно-зависимый семантический OLAP, разработанный для биомедицинских приложений.
- Платформа **CaseOLAP** включает предварительную обработку данных (например, загрузку, извлечение и анализ текстовых документов), индексирование и поиск с помощью **Elasticsearch**, создавая функциональную структуру документа, называемую текстовым кубом, и количественная оценка определяемых пользователем отношений фраза-категория с использованием основной алгоритм **CaseOLAP**.

Продукты с открытым исходным кодом

Apache Pinot используется в LinkedIn, Cisco, Uber, Slack, Stripe, DoorDash, Target, Walmart, Amazon и Microsoft для предоставления масштабируемой аналитики в реальном времени с низкой задержкой.

Mondrian OLAP server - это OLAP-сервер с открытым исходным кодом, написанный на Java. Он поддерживает язык запросов MDX, XML для анализа и спецификации интерфейса olap4j.

Apache Druid - популярное распределенное хранилище данных с открытым исходным кодом для запросов OLAP, которое масштабно используется в производстве различными организациями.

Apache Kylin - это распределенное хранилище данных для запросов OLAP, первоначально разработанное eBay.

Cubes (OLAP server) - еще один облегченный инструментарий с открытым исходным кодом, реализующий функциональность OLAP на языке программирования Python со встроенным ROLAP.

ClickHouse - это довольно новая ориентированная на столбцы СУБД, ориентированная на быструю обработку и время отклика.

В числе наиболее популярных сегодня на российском рынке аналитических СУБД особенно интересны два проекта с открытым исходным кодом: **Greenplum и ClickHouse.**