МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ
МГПУ

Инструменты аналитика данных для решения прикладных задач
лекция 3

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ
МГПУ

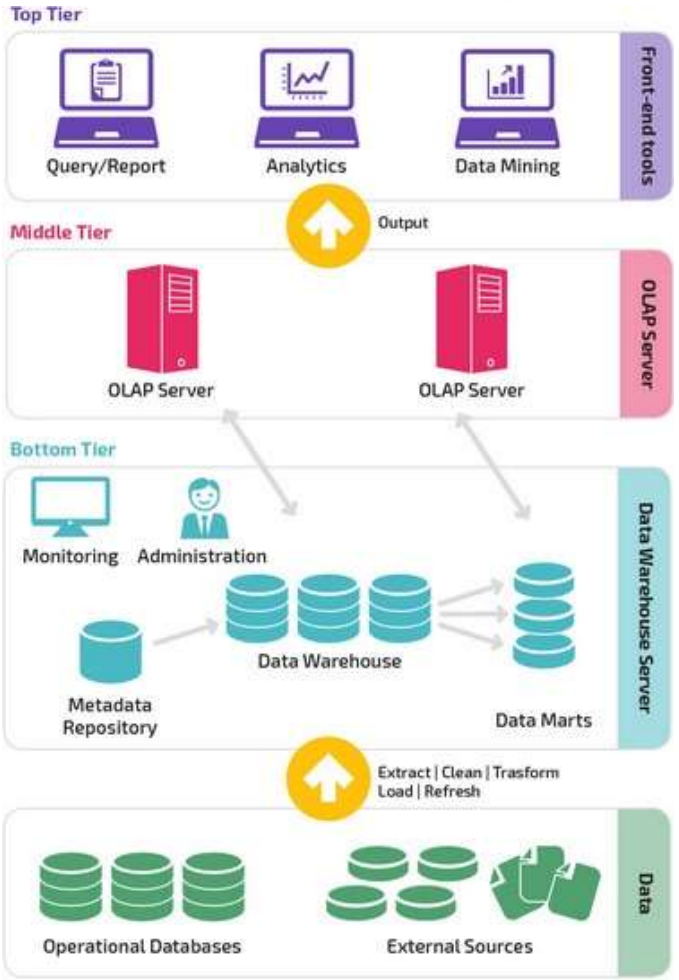
Архитектура хранилищ данных: традиционная и облачная

Традиционная архитектура хранилища данных

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

Трехуровневая архитектура

- **Нижний уровень:**
- этот уровень содержит сервер базы данных, используемый для извлечения данных из множества различных источников, например, из транзакционных баз данных, используемых для интерфейсных приложений.

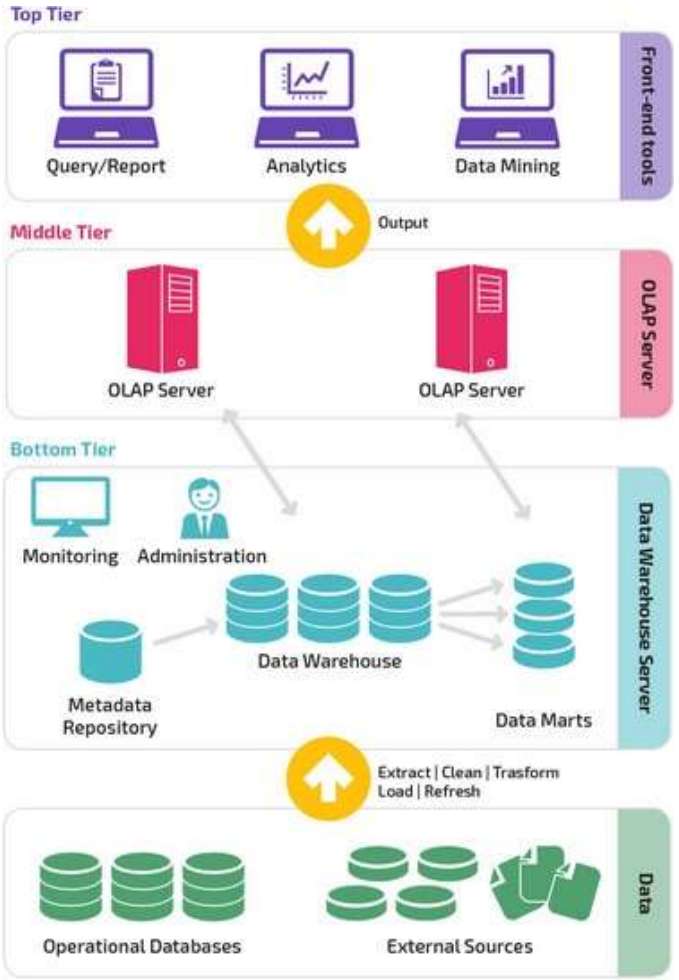


Традиционная архитектура хранилища данных

Трехуровневая архитектура

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

•**Средний уровень**:
средний уровень содержит сервер **OLAP**, который преобразует данные в структуру, лучше подходящую для анализа и сложных запросов. Сервер **OLAP** может работать двумя способами: либо в качестве расширенной системы управления реляционными базами данных, которая отображает операции над многомерными данными в стандартные реляционные операции (**Relational OLAP**), либо с использованием многомерной модели **OLAP**, которая непосредственно реализует многомерные данные и операции.



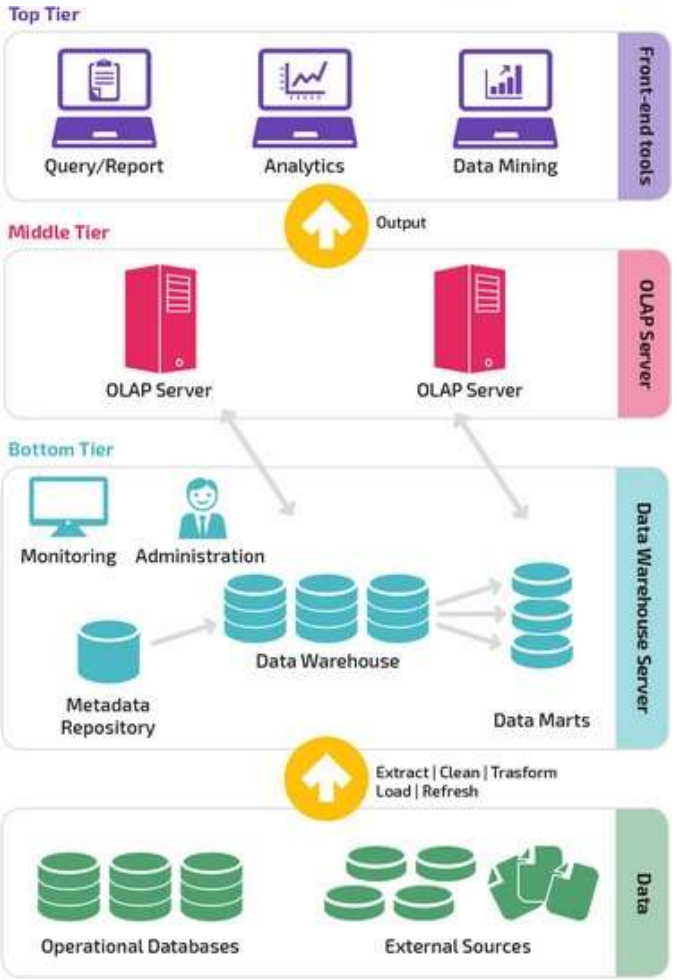
Традиционная архитектура хранилища данных

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

Трехуровневая архитектура

- **Верхний уровень:**

верхний уровень — это уровень клиента. Этот уровень содержит инструменты, используемые для высокоуровневого анализа данных, создания отчетов и анализа данных.



Kimball vs. Inmon

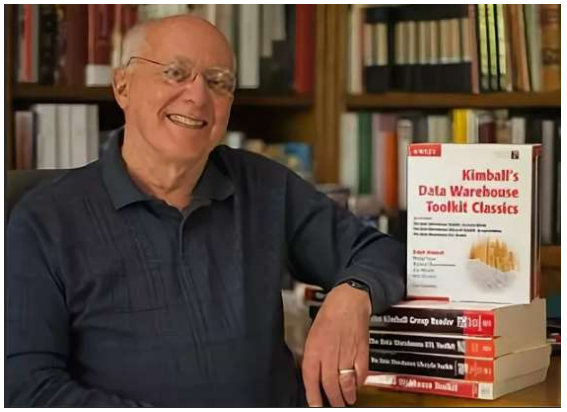
МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ
МГПУ

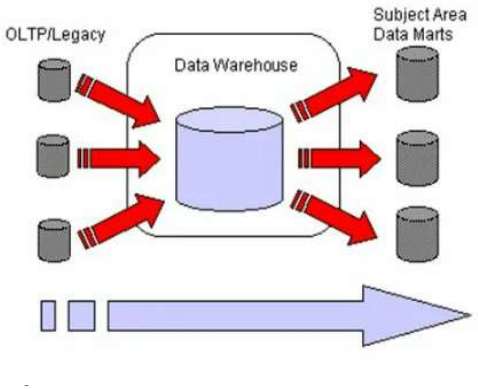
Два пионера хранилищ данных: **Билл Инмон** и **Ральф Кимбалл** предлагают разные подходы к проектированию.

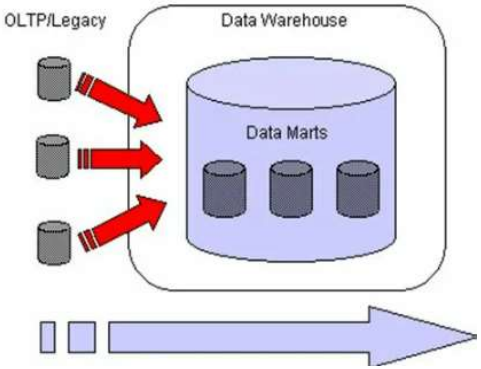
Kimball vs. Inmon

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

- Подход **Ральфа Кимбалла** основывается на важности витрин данных, которые являются хранилищами данных, принадлежащих конкретным направлениям бизнеса.
- Хранилище данных — это просто **сочетание различных витрин данных**, которые облегчают отчетность и анализ. Проект хранилища данных по принципу Кимбалла использует подход «снизу вверх».



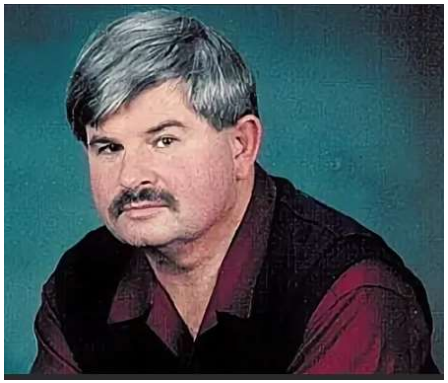


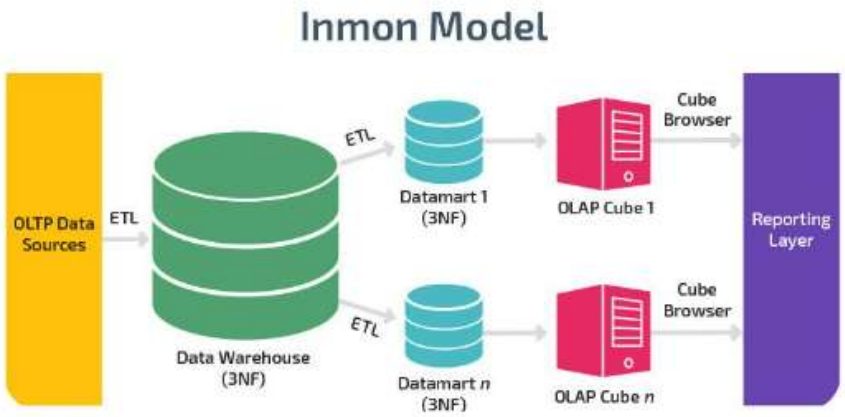


Kimball vs. Inmon

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

- Подход **Билла Инмона** основывается на том, что хранилище данных является централизованным хранилищем всех корпоративных данных. При таком подходе организация сначала создает **нормализованную модель** хранилища данных. Затем создаются витрины размерных данных на основе модели хранилища. Это известно как **нисходящий подход к хранилищу данных**.





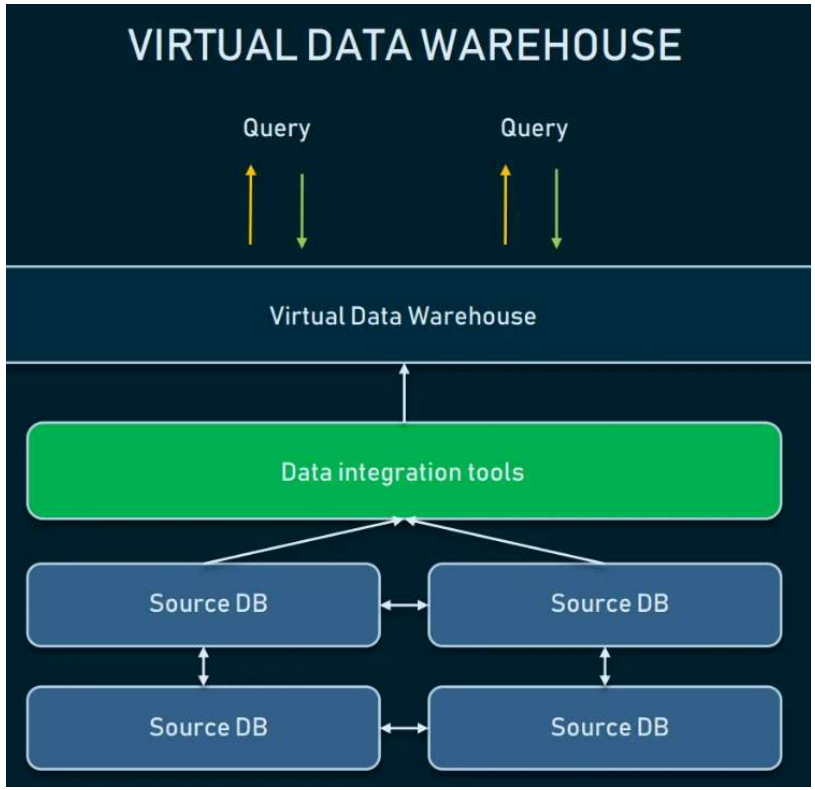
МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ
МГПУ

Модели хранилищ данных

Модели хранилищ данных

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

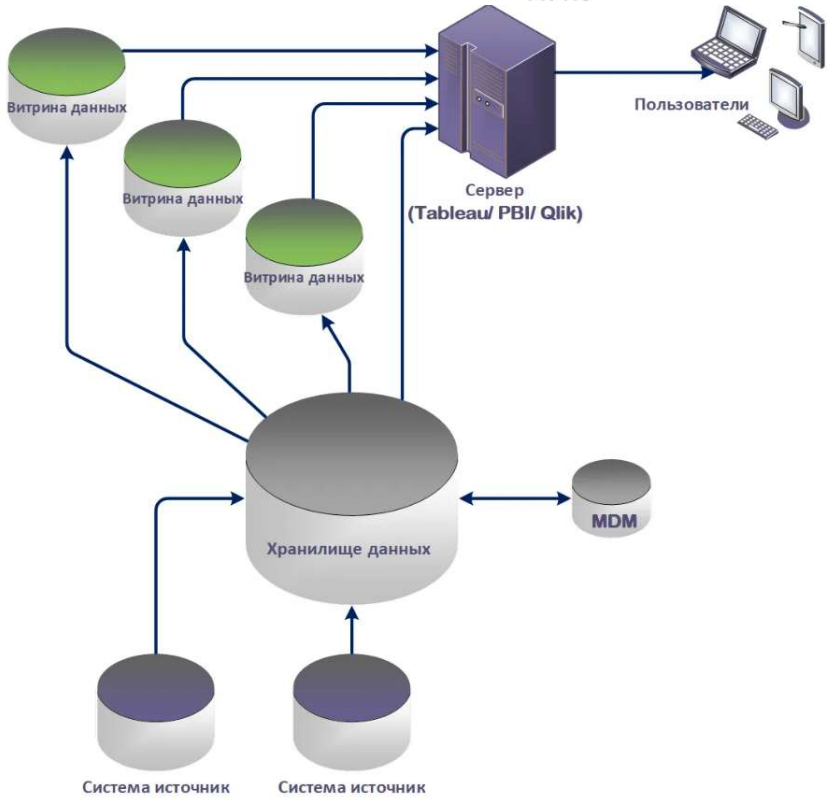
- **Виртуальное хранилище данных** — это набор отдельных баз данных, которые можно использовать совместно, чтобы пользователь мог эффективно получать доступ ко всем данным, как если бы они хранились в одном хранилище данных;



Модели хранилищ данных

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

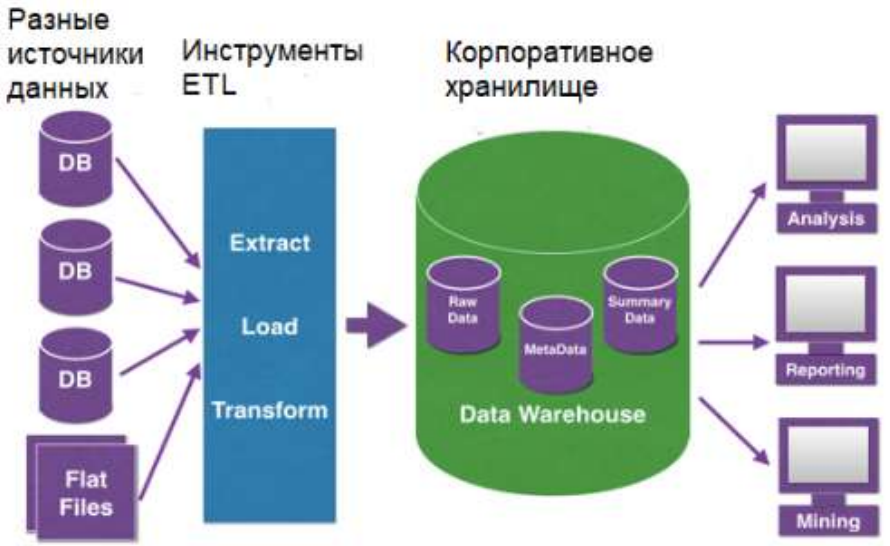
- **Модель витрины данных** используется для отчетности и анализа конкретных бизнес-линий. В этой модели хранилища – агрегированные данные из ряда исходных систем, относящихся к конкретной бизнес-сфере, такой как продажи или финансы;



Модели хранилищ данных

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

- **Модель корпоративного хранилища** данных предполагает хранение агрегированных данных, охватывающих всю организацию. Эта модель рассматривает хранилище данных как сердце информационной системы предприятия с интегрированными данными всех бизнес-единиц



МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ
МГПУ

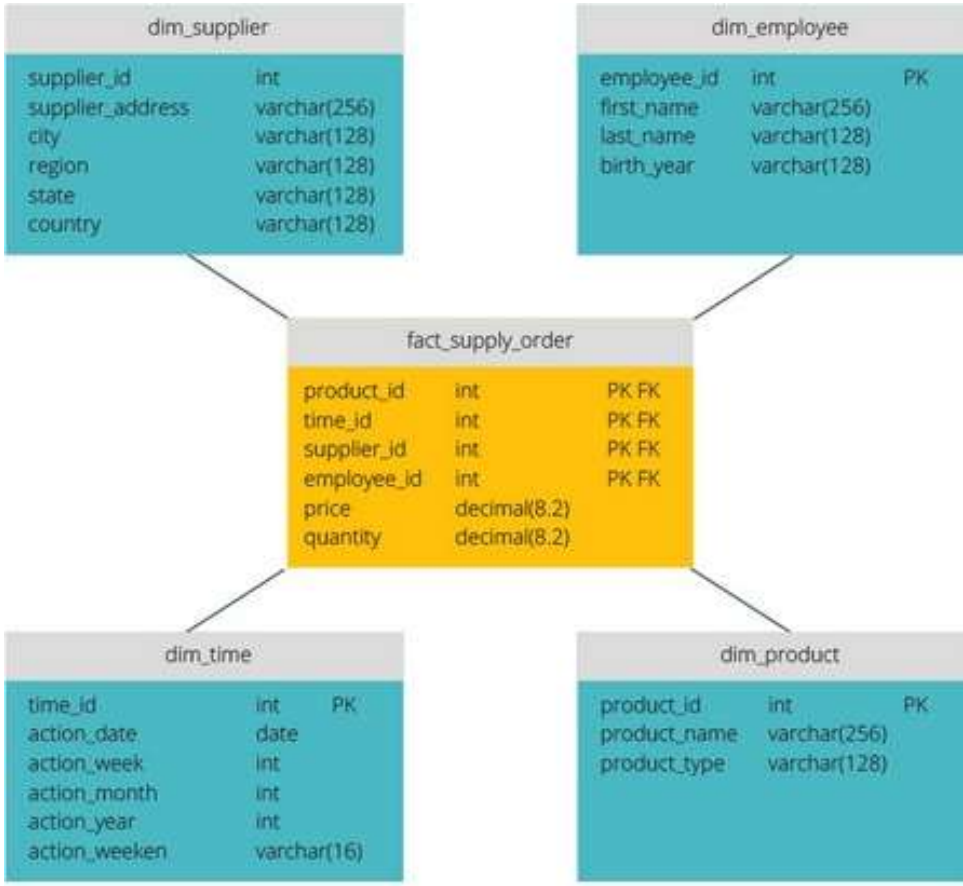
Способы структурирования хранилищ данных

Звезда vs. Снежинка

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ
МГПУ

Схема типа **«звезда»** имеет централизованное хранилище данных, которое хранится в таблице фактов. Схема разбивает таблицу фактов на ряд денормализованных таблиц измерений. Таблица фактов содержит агрегированные данные, которые будут использоваться для составления отчетов, а таблица измерений описывает хранимые данные.

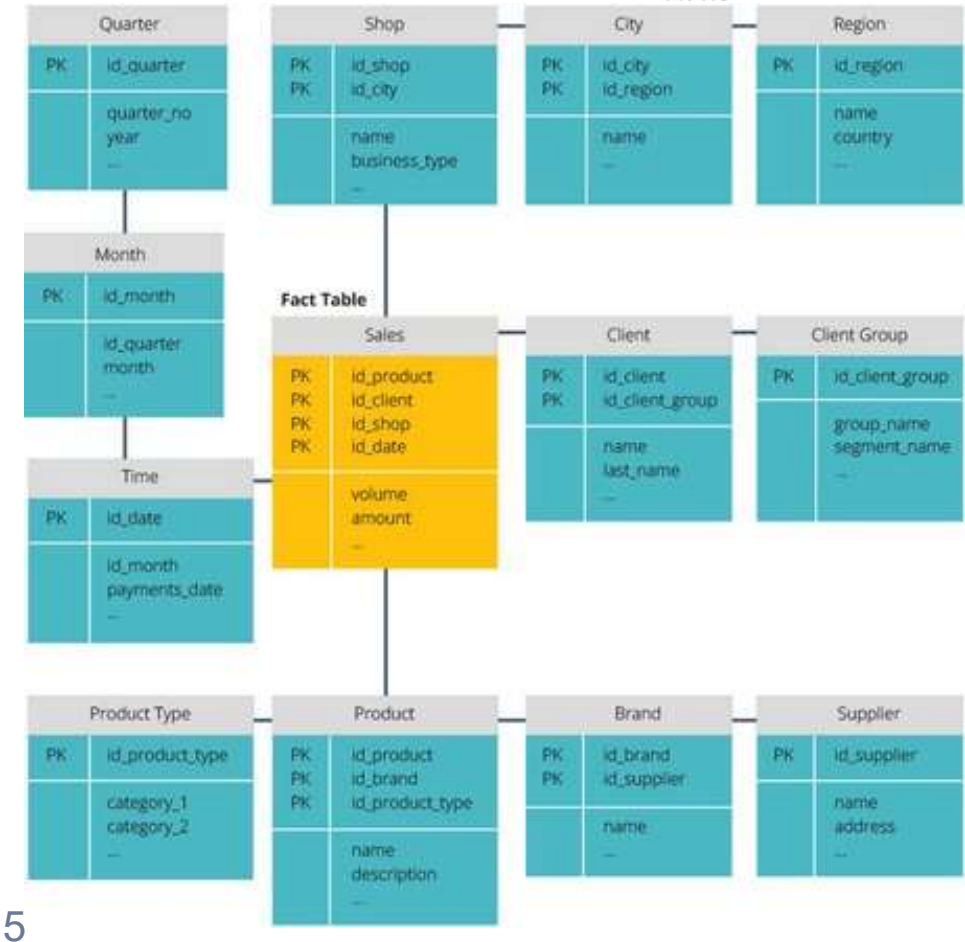
Денормализованные проекты менее сложны, потому что данные сгруппированы. Таблица фактов использует только одну ссылку для присоединения к каждой таблице измерений. Более простая конструкция звездообразной схемы значительно упрощает написание сложных запросов.



Звезда vs. Снежинка

Схема типа **«снежинка»** отличается тем, что использует нормализованные данные. Нормализация означает эффективную организацию данных так, чтобы все зависимости данных были определены, и каждая таблица содержала минимум избыточности. Таким образом, отдельные таблицы измерений разветвляются на отдельные таблицы измерений.

Схема **«снежинки»** использует меньше дискового пространства и лучше сохраняет целостность данных. Основным недостатком является сложность запросов, необходимых для доступа к данным — каждый запрос должен пройти несколько соединений таблиц, чтобы получить соответствующие данные.



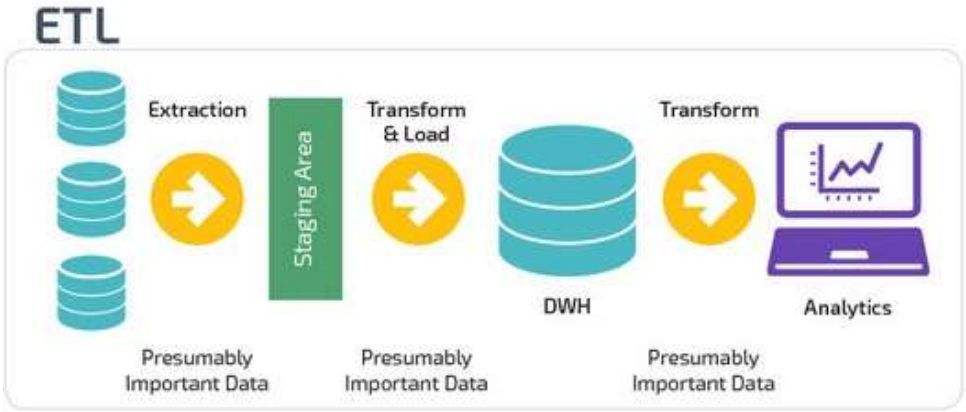
МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ
МГПУ

Способы загрузки данных в хранилище

ETL vs. ELT

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

ETL (Extract, Transform, Load) сначала извлекают данные из пула источников данных. Данные хранятся во временной промежуточной базе данных. Затем выполняются операции преобразования, чтобы структурировать и преобразовать данные в подходящую форму для целевой системы хранилища данных. Затем структурированные данные загружаются в хранилище и готовы к анализу.



ETL vs. ELT

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

В случае **ELT (Extract, Load, Transform)** данные сразу же загружаются после извлечения из исходных пулов данных. Промежуточная база данных отсутствует, что означает, что данные немедленно загружаются в единый централизованный репозиторий. Данные преобразуются в системе хранилища данных для использования с инструментами бизнес-аналитики и аналитики.

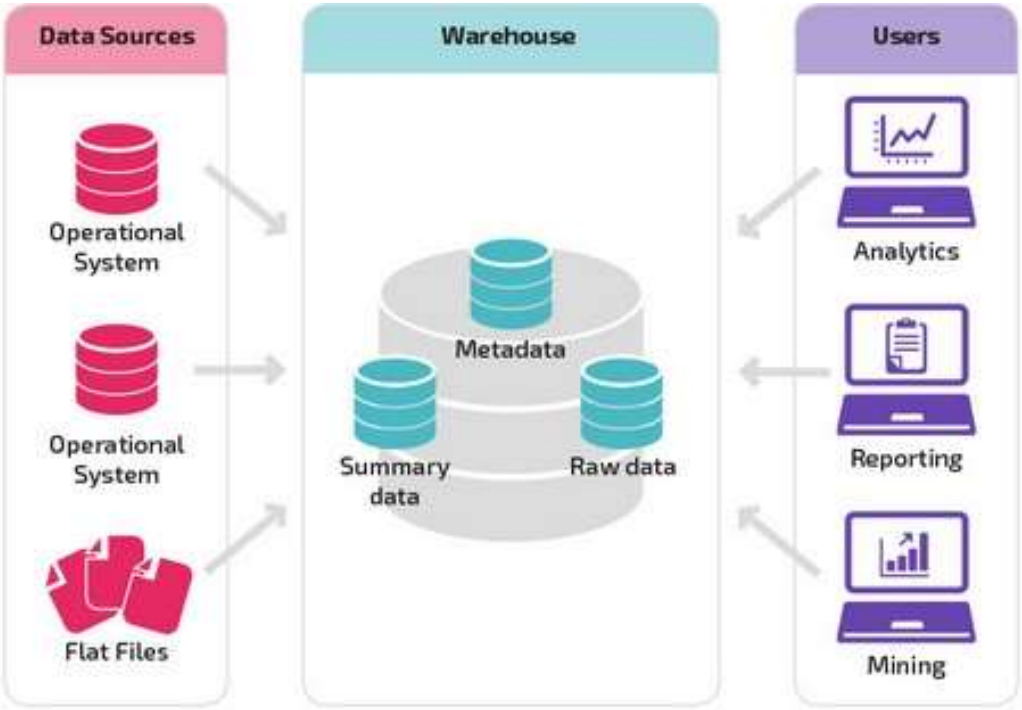
МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ
МГПУ

Структура хранилища данных

Базовая структура

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

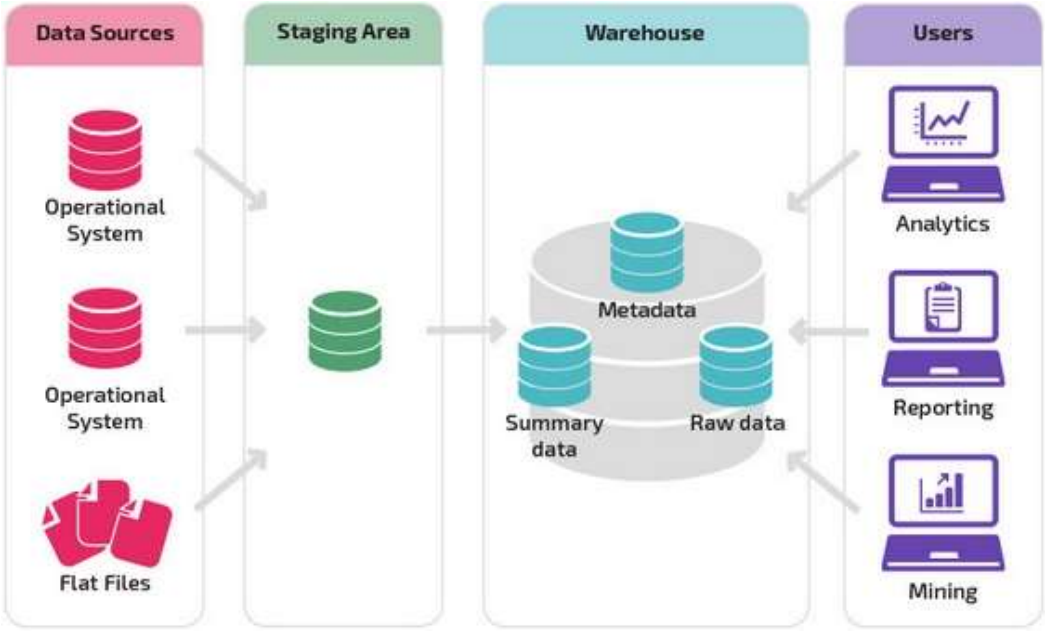
Базовая структура позволяет конечным пользователям хранилища напрямую получать доступ к сводным данным, полученным из исходных систем, создавать отчеты и анализировать эти данные. Эта структура полезна для случаев, когда источники данных происходят из одних и тех же типов систем баз данных.



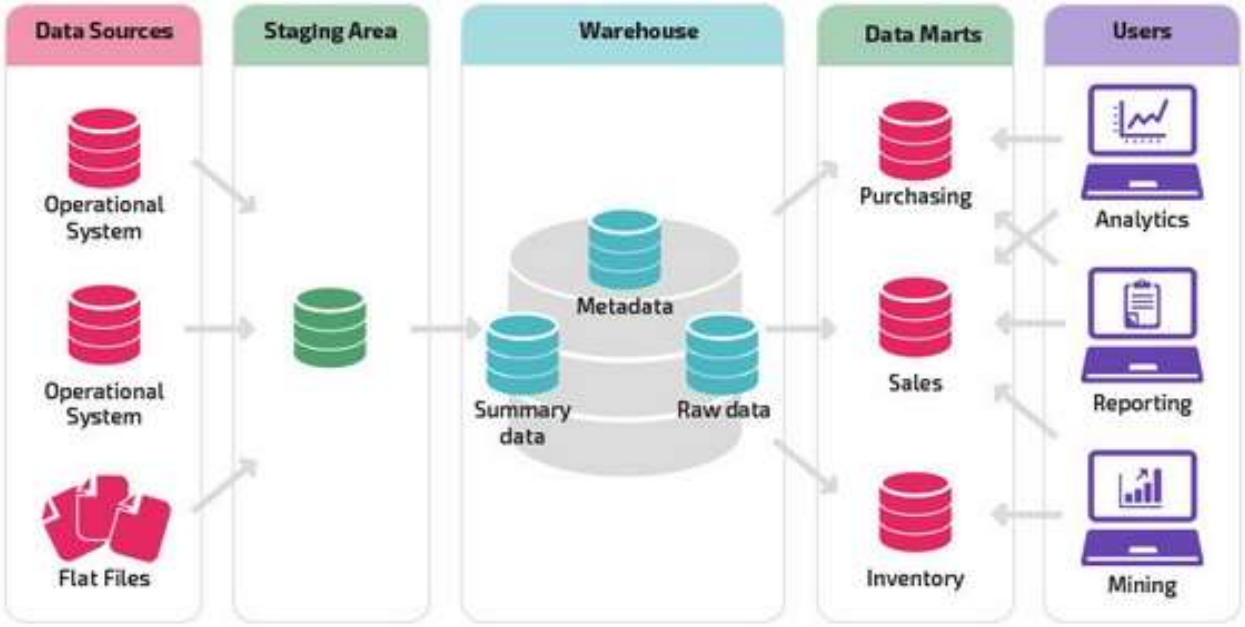
Хранилище с промежуточной областью

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

Хранилище с промежуточной областью является следующим логическим шагом в организации с разнородными источниками данных с множеством различных типов и форматов данных. Промежуточная область преобразует данные в обобщенный структурированный формат, который проще запрашивать с помощью инструментов анализа и отчетности.



Хранилище с промежуточной областью



Одной из разновидностей промежуточной структуры является **добавление витрин данных в хранилище данных**. В витринах данных хранятся сводные данные по конкретной сфере деятельности, что делает эти данные легко доступными для конкретных форм анализа.

Например, добавление витрин данных может позволить финансовому аналитику легче выполнять подробные запросы к данным о продажах, прогнозировать поведение клиентов. Витрины данных облегчают анализ, адаптируя данные специально для удовлетворения потребностей конечного пользователя.

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ
МГПУ

Новые архитектуры хранилищ данных

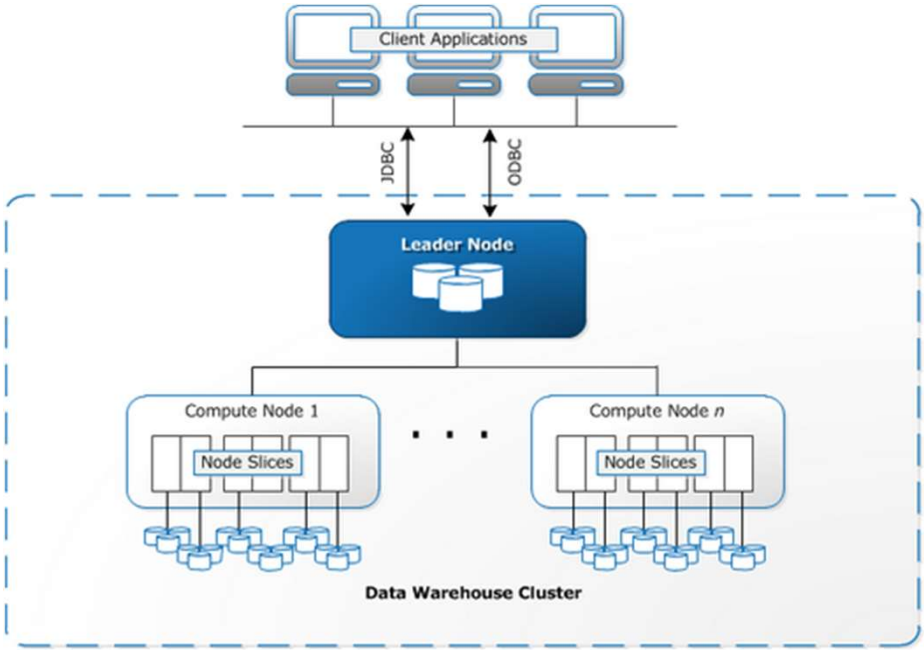
Amazon Redshift

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

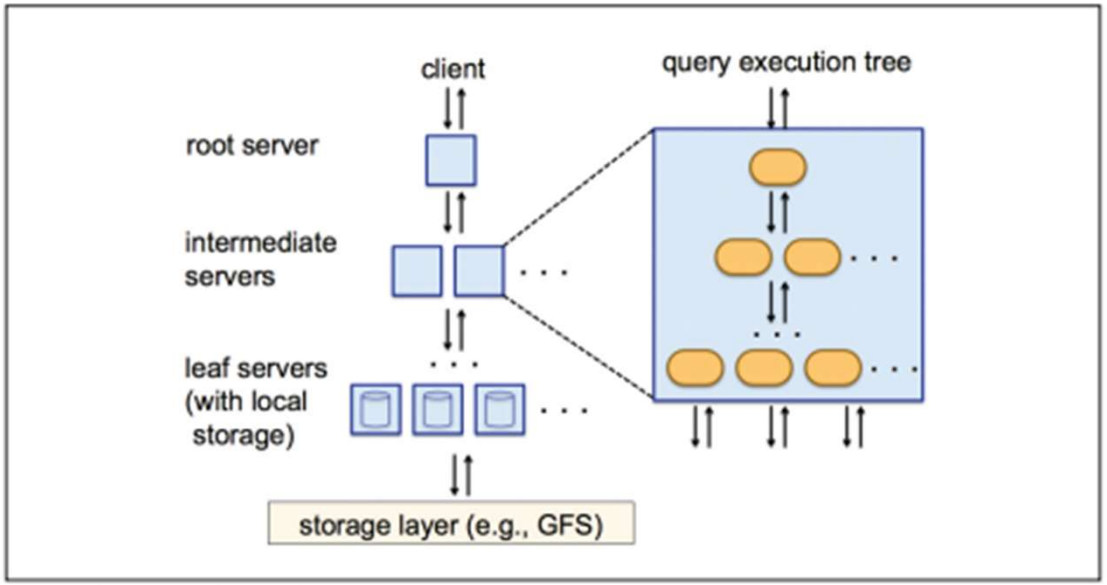
Amazon Redshift — это облачное представление традиционного хранилища данных.

Redshift требует, чтобы вычислительные ресурсы были подготовлены и настроены в виде кластеров, которые содержат набор из одного или нескольких узлов. Каждый узел имеет свой собственный процессор, память и оперативную память. **Leader Node** компилирует запросы и передает их вычислительным узлам, которые выполняют запросы.

На каждом узле данные хранятся в блоках, называемых **срезами**. **Redshift** использует колоночное хранение, то есть каждый блок данных содержит значения из одного столбца в нескольких строках, а не из одной строки со значениями из нескольких столбцов.



Google BigQuery



Tree architecture of Dremel

Архитектура **BigQuery** не требует сервера, а это означает, что Google динамически управляет распределением ресурсов компьютера. Поэтому все решения по управлению ресурсами скрыты от пользователя.

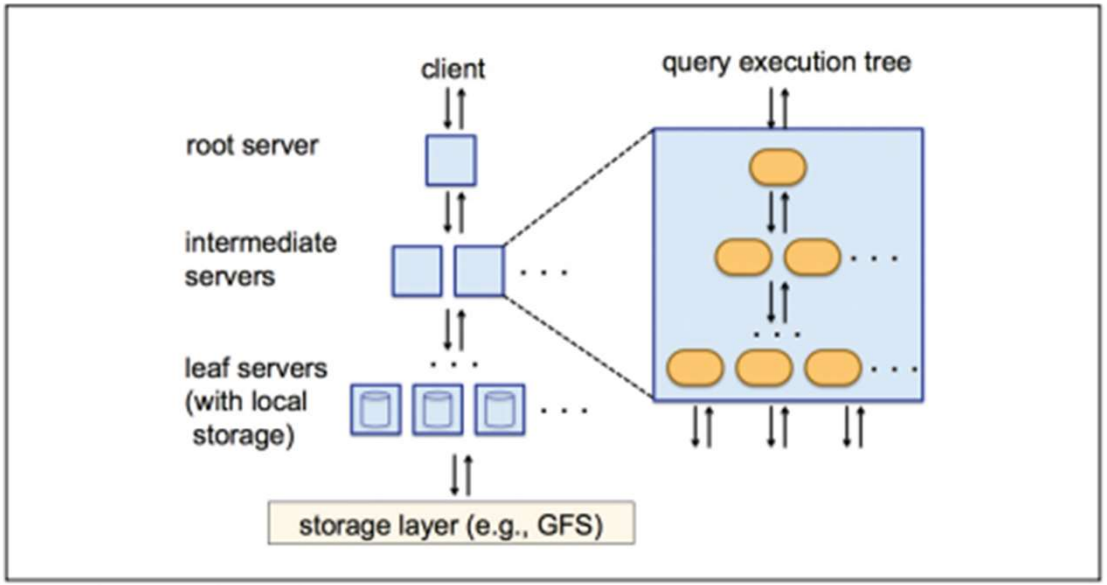
BigQuery позволяет клиентам загружать данные из Google Cloud Storage и других читаемых источников данных. Альтернативным вариантом является потоковая передача данных, что позволяет разработчикам добавлять данные в хранилище данных в режиме реального времени, строка за строкой, когда они становятся доступными.

BigQuery использует механизм выполнения запросов под названием Dremel, который может сканировать миллиарды строк данных всего за несколько секунд. Dremel использует массивно параллельные запросы для сканирования данных в базовой системе управления файлами Colossus. Colossus распределяет файлы на куски по 64 мегабайта среди множества вычислительных ресурсов, называемых узлами, которые сгруппированы в кластеры.

Dremel использует колоночную структуру данных, аналогичную Redshift. Древовидная архитектура отправляет запросы тысячам машин за считанные секунды.

Для выполнения запросов к данным используются простые команды SQL.

Google BigQuery



Tree architecture of Dremel

Архитектура **BigQuery** не требует сервера, а это означает, что Google динамически управляет распределением ресурсов компьютера. Поэтому все решения по управлению ресурсами скрыты от пользователя.

BigQuery позволяет клиентам загружать данные из Google Cloud Storage и других читаемых источников данных. Альтернативным вариантом является потоковая передача данных, что позволяет разработчикам добавлять данные в хранилище данных в режиме реального времени, строка за строкой, когда они становятся доступными.

BigQuery использует механизм выполнения запросов под названием Dremel, который может сканировать миллиарды строк данных всего за несколько секунд. Dremel использует массивно параллельные запросы для сканирования данных в базовой системе управления файлами Colossus. Colossus распределяет файлы на куски по 64 мегабайта среди множества вычислительных ресурсов, называемых узлами, которые сгруппированы в кластеры.

Dremel использует колоночную структуру данных, аналогичную Redshift. Древовидная архитектура отправляет запросы тысячам машин за считанные секунды.

Для выполнения запросов к данным используются простые команды SQL.

Источники

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

1. **Wiley Pentaho Kettle Solutions, Building Open Source ETL Solutions with Pentaho Data Integration** https://github.com/kyosuke1018/tips/blob/master/Wiley%20Pentaho%20Kettle%20Solutions%2C%20Building%20Open%20Source%20ETL%20Solutions%20with%20Pentaho%20Data%20Integration%20(2010).pdf

2. **Subsystems of ETL Revisited** https://www.kimballgroup.com/2007/10/subsystems-of-etl-revisited/

3. **Data Warehousing - 34 Kimball Subsytems** https://datacadamia.com/data/warehouse/subsystem

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ
МГПУ

Практика 3

В качестве практики вам необходимо выявить 8-10 подсистем в **ETL Pentaho DI и Talend,**
написать отчет, в котором Вы приложите **print screen** компонента (ETL подсистемы) и напишите про его свойства. Результат сохраните в вашем **Git**.

МОСКОВСКИЙ
ГОРОДСКОЙ
УНИВЕРСИТЕТ
МГПУ

СПАСИБО ЗА ВНИМАНИЕ