

Занятие 1

Основные принципы и понятия Data Warehouse (DWH)

Бояр Владислав

План курса

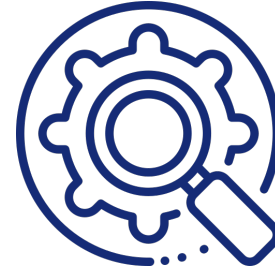
- ① Основные принципы и понятия Data Warehouse (DWH)
- ② Распределенные файловые системы. Hadoop, Spark
- ③ Massive parallel processing (MPP) - системы. Greenplum
- ④ ETL и оркестрирование. Cron, Airflow
- ⑤ Обеспечение качества данных

Занятие состоит из:



Теория:

- Виды СУБД;
- Что такое DWH и DataLake;
- Слои DWH.



Практика:


- Проектирование и создание многослойной DWH.

Системы управления базами данных (СУБД)

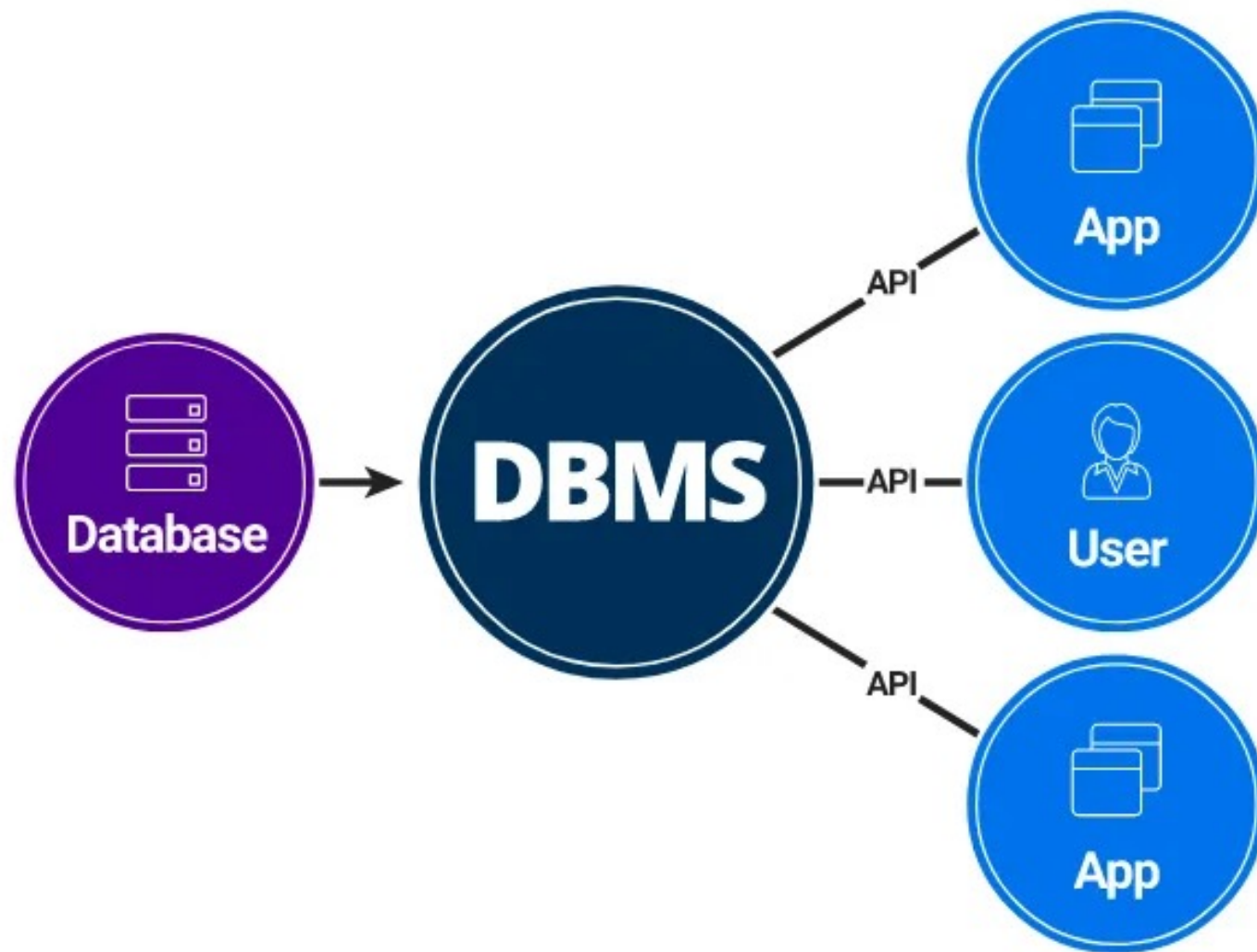
Что такое СУБД?

База данных (БД) – структурированный набор данных (файл с данными на компьютере сервере);

Система управления базами данных (СУБД) – программа, позволяющая манипулировать данными в БД (проводить выборку/вставку/удаление элементов и т.д.)



Взаимосвязь БД и СУБД

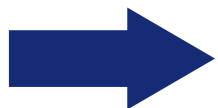




Почему СУБД и хранилища
данных лучше Excel и Google
таблиц?



Почему СУБД и хранилища данных лучше Excel и Google таблиц?

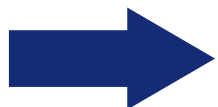


Позволяют хранить больше информации, чем электронные таблицы:

- xls ~ 65 тыс. строк;
- xlsx ~ 1 млн. строк.



Работают быстрее на больших данных;



Процессы доступа быстрее и безопаснее;



Проще интегрировать с другими источниками данных.

Классификации СУБД

СУБД классифицируются в зависимости от того, как структурирована информация и как с ней взаимодействовать.



Реляционные




Нереляционные
(NoSQL)

Реляционные СУБД

Реляционные СУБД

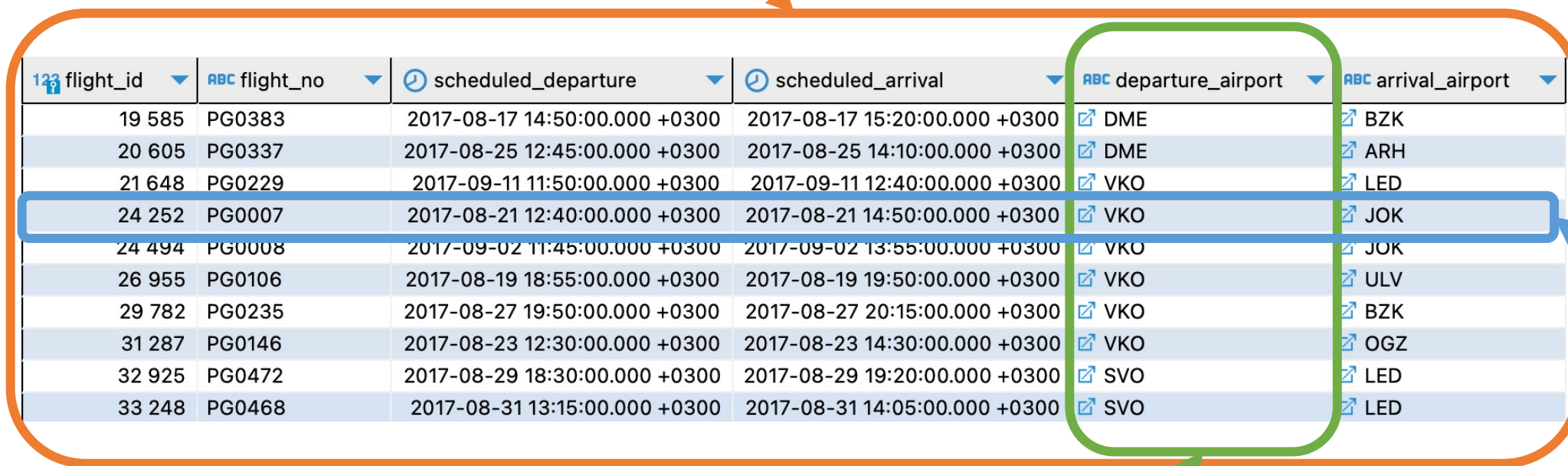
Реляционные СУБД - представляют собой множество сущностей (таблиц) и связей между ними

Основные сущности:

- Основной способ доступа к данным – SQL запросы;
 - Таблицы и их составляющие (атрибуты, кортежи);
 - Связи между таблицами (FK);
 - Ограничения (constraints) – PK, FK, Unique, Not Null, Default, Check.
- 

Реляционные СУБД. Термины

отношение, таблица



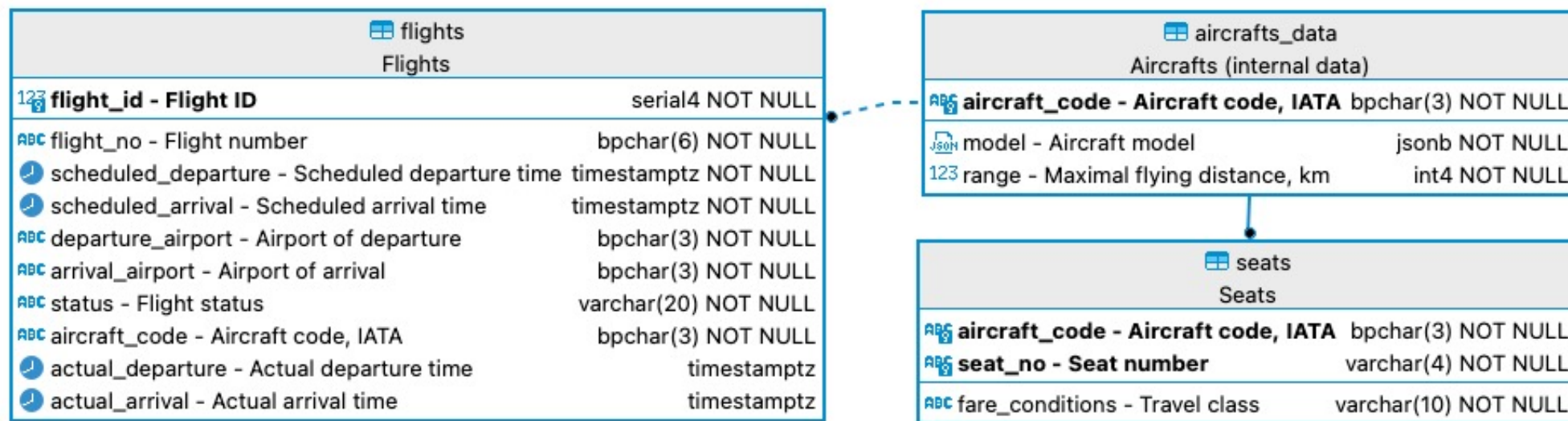
123 flight_id	ABC flight_no	🕒 scheduled_departure	🕒 scheduled_arrival	ABC departure_airport	ABC arrival_airport
19 585	PG0383	2017-08-17 14:50:00.000 +0300	2017-08-17 15:20:00.000 +0300	✈ DME	✈ BZK
20 605	PG0337	2017-08-25 12:45:00.000 +0300	2017-08-25 14:10:00.000 +0300	✈ DME	✈ ARH
21 648	PG0229	2017-09-11 11:50:00.000 +0300	2017-09-11 12:40:00.000 +0300	✈ VKO	✈ LED
24 252	PG0007	2017-08-21 12:40:00.000 +0300	2017-08-21 14:50:00.000 +0300	✈ VKO	✈ JOK
24 494	PG0008	2017-09-02 11:45:00.000 +0300	2017-09-02 13:55:00.000 +0300	✈ VKO	✈ JOK
26 955	PG0106	2017-08-19 18:55:00.000 +0300	2017-08-19 19:50:00.000 +0300	✈ VKO	✈ ULV
29 782	PG0235	2017-08-27 19:50:00.000 +0300	2017-08-27 20:15:00.000 +0300	✈ VKO	✈ BZK
31 287	PG0146	2017-08-23 12:30:00.000 +0300	2017-08-23 14:30:00.000 +0300	✈ VKO	✈ OGZ
32 925	PG0472	2017-08-29 18:30:00.000 +0300	2017-08-29 19:20:00.000 +0300	✈ SVO	✈ LED
33 248	PG0468	2017-08-31 13:15:00.000 +0300	2017-08-31 14:05:00.000 +0300	✈ SVO	✈ LED

кортеж,
запись,
строка

атрибут, поле, колонка

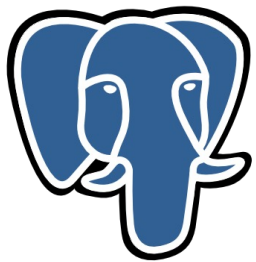
Связи между таблицами

ER (Entity–relationship) - диаграмма



Где на диаграмме **первичные** ключи, а где **вторичные** (внешние)?

Примеры реляционных СУБД



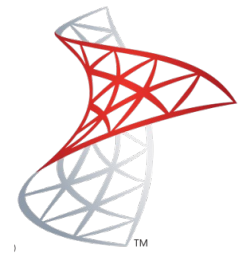
PostgreSQL



MySQL



Oracle




Microsoft SQL Server

Нереляционные СУБД (NoSQL)

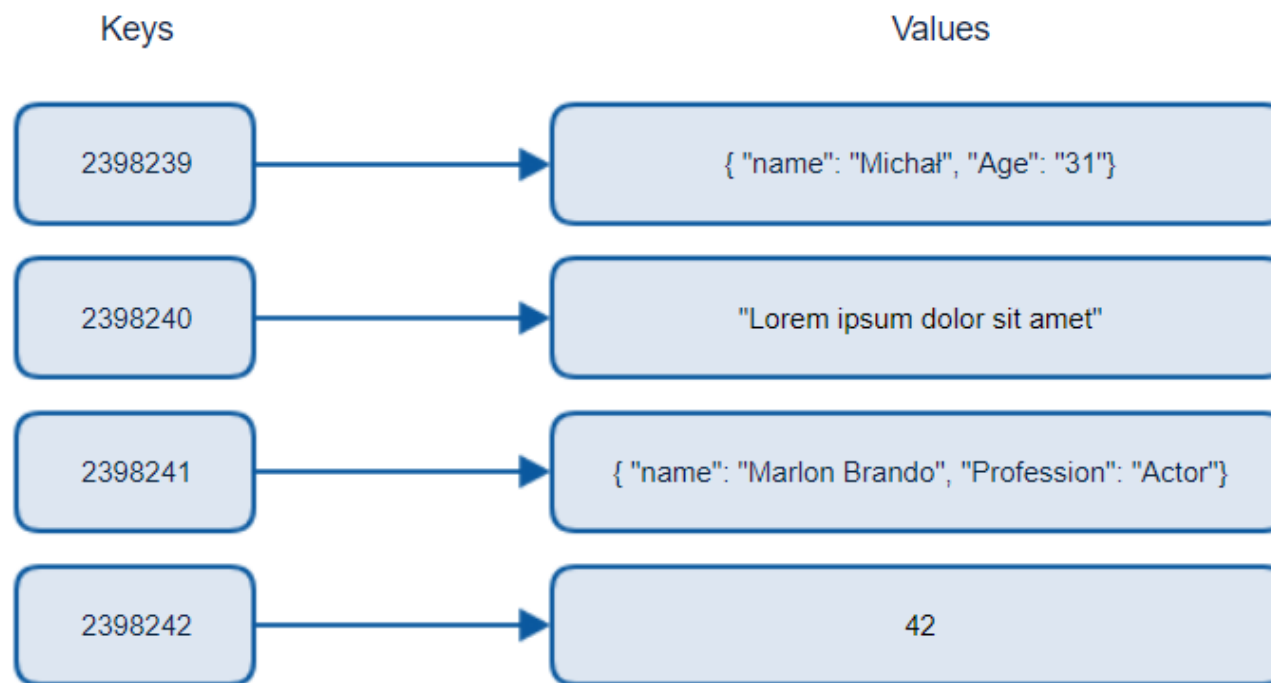
Нереляционные СУБД (NoSQL)

Нереляционные СУБД для доступа к данным не предполагают SQL запросы:

- Ключ-значение (Key Value)
 - Документоориентированные (Document Oriented)
 - Колоночные (Column Oriented)
 - Графовые (Graph)
- 

Key-value DB

В Key-value данные хранятся в ассоциативных массивах (словарях, хэш-таблицах). Часто используется как прослойка между пользователями/сервисом и реляционной БД.



Примеры: Хранилище сессий подключений, корзина интернет-магазина.

Key-value DB

Плюсы:

- + Простота реализации;
- + Быстрый доступ к данным;
- + Возможность хранить неструктурированные данные;
- + Легко масштабируемые.

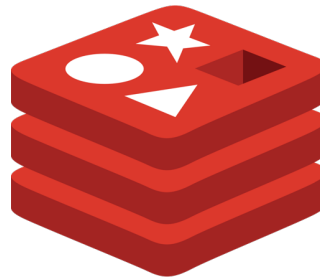
Минусы:

- Структура данных практически отсутствует;
- Обновление данных происходит только целиком;
- Нельзя проводить фильтрацию по значению.

Примеры Key-Value BD



Redis



Apache Ignite



Memcached

Document Oriented

Документоориентированные БД позволяют хранить данные в виде документов в полуструктурированных форматах (JSON, XML). Являются более сложной версией хранилищ “ключ-значение”

```
{
  "_id": 2,
  "first_name": "Donna",
  "email": "donna@example.com",
  "spouse": "Joe",
  "likes": [
    "spas",
    "shopping",
    "live tweeting"
  ],
  "businesses": [
    {
      "name": "Castle Realty",
      "status": "Thriving",
      "date_founded": {
        "$date": "2013-11-21T04:00:00Z"
      }
    }
  ]
}
```

Примеры:

- Каталоги;
- Пользовательские данные;
- Логи;
- Ответы внешних источников (API).

Document Oriented

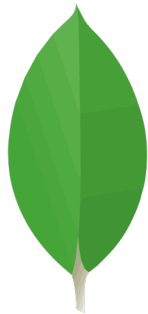
Плюсы:

- + Свободно изменяемое количество атрибутов у объектов(документов);
- + Изменение атрибутов одного документа не влияет на другие;
- + Большая глубина вложенности атрибутов.

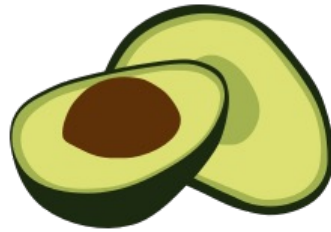
Минусы:

- Плохо работает с системами, где присутствует множество связей между объектами.

Примеры Document-oriented



MongoDB



ArangoDB



CouchDB

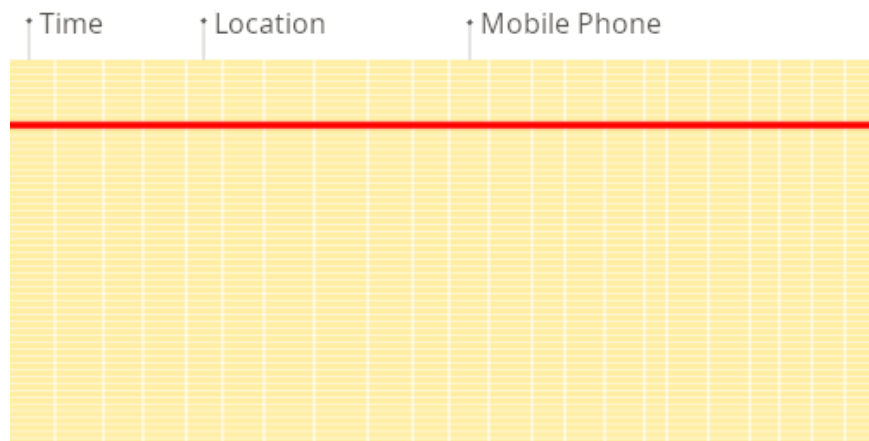
Column Oriented

В колоночных БД данные каждого столбца хранятся отдельно (независимо) от других столбцов;

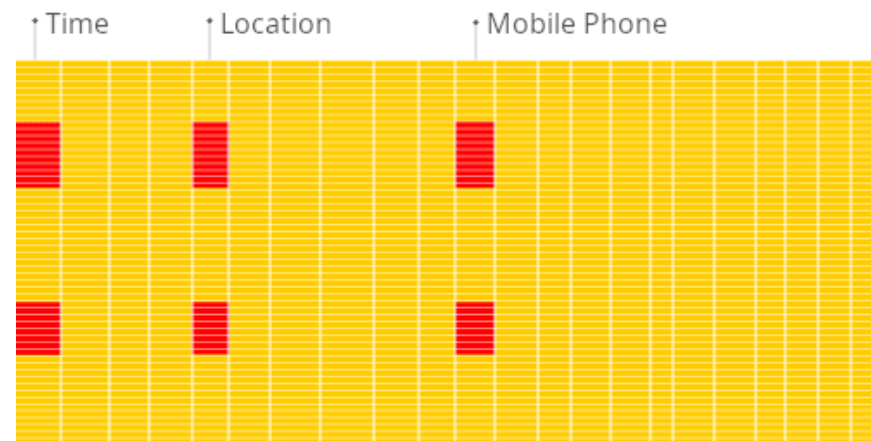
Нужны для обработки и хранения больших данных;

Используются в аналитических приложениях, BI

Реляционные БД



Колоночные БД



Column-Oriented

Плюсы:

- + Быстрые операции над колонками
- + Удобные для работы «широкие» таблицы
- + Нет необходимости делать много джоинов
- + Агрегация запросов на больших объемах данных

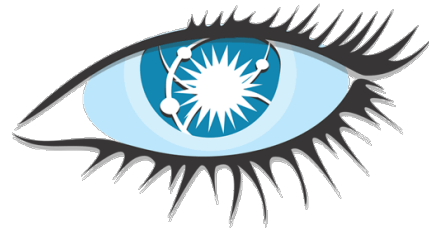
Минусы:

- «Дорогие» операции над строками
- Тяжелые операции объединения (join)

Примеры Column-Oriented



ClickHouse



Apache Cassandra

VERTICA

Vertica



Apache HBase

Graph

В графовых БД связи обозначены узлами, рёбрами и свойствами.

Записи в этих БД могут иметь любое количество связанных с ними свойств.

Структура похожа на связанные списки.

Используются для анализа соцсетей, рекомендательных сервисов, антифрода.



Примеры Graph



Data WareHouse (DWH)

Data WareHouse (DWH)

DWH – это:

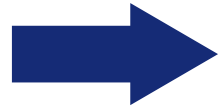
- Единое хранилище данных;
- Является получателем данных из различных источников;
- Является источником данных для внутренних и внешних потребителей.



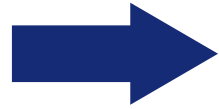
Для чего компании создают
хранилища данных?



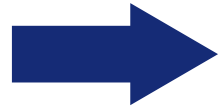
Для чего компания создают хранилища данных?



Формирование единого источника данных

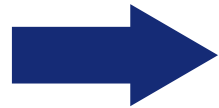


Нормализация данных с целью уменьшения занимаемого ими дискового пространства



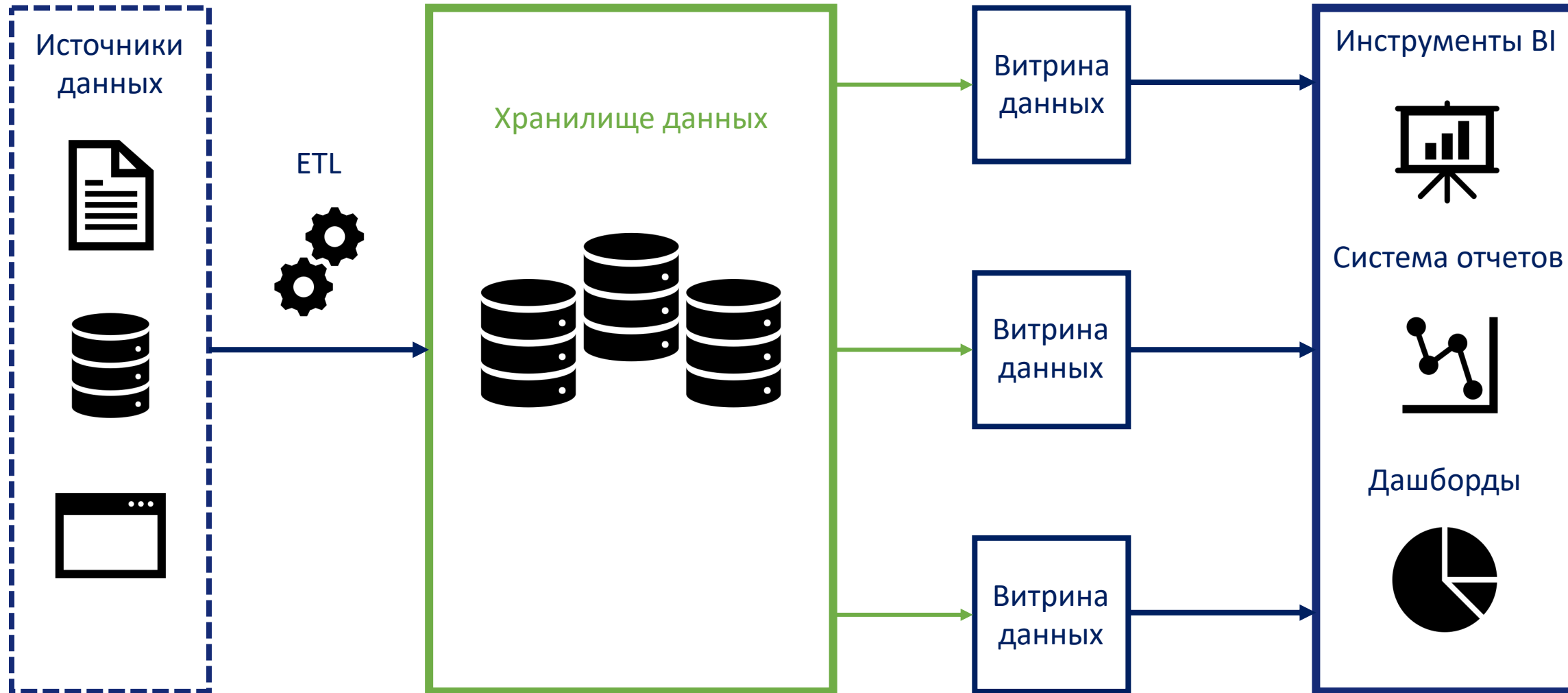
Предоставление данных для внутренних и внешних потребителей:

- DA;
- BI;
- DS (ML).

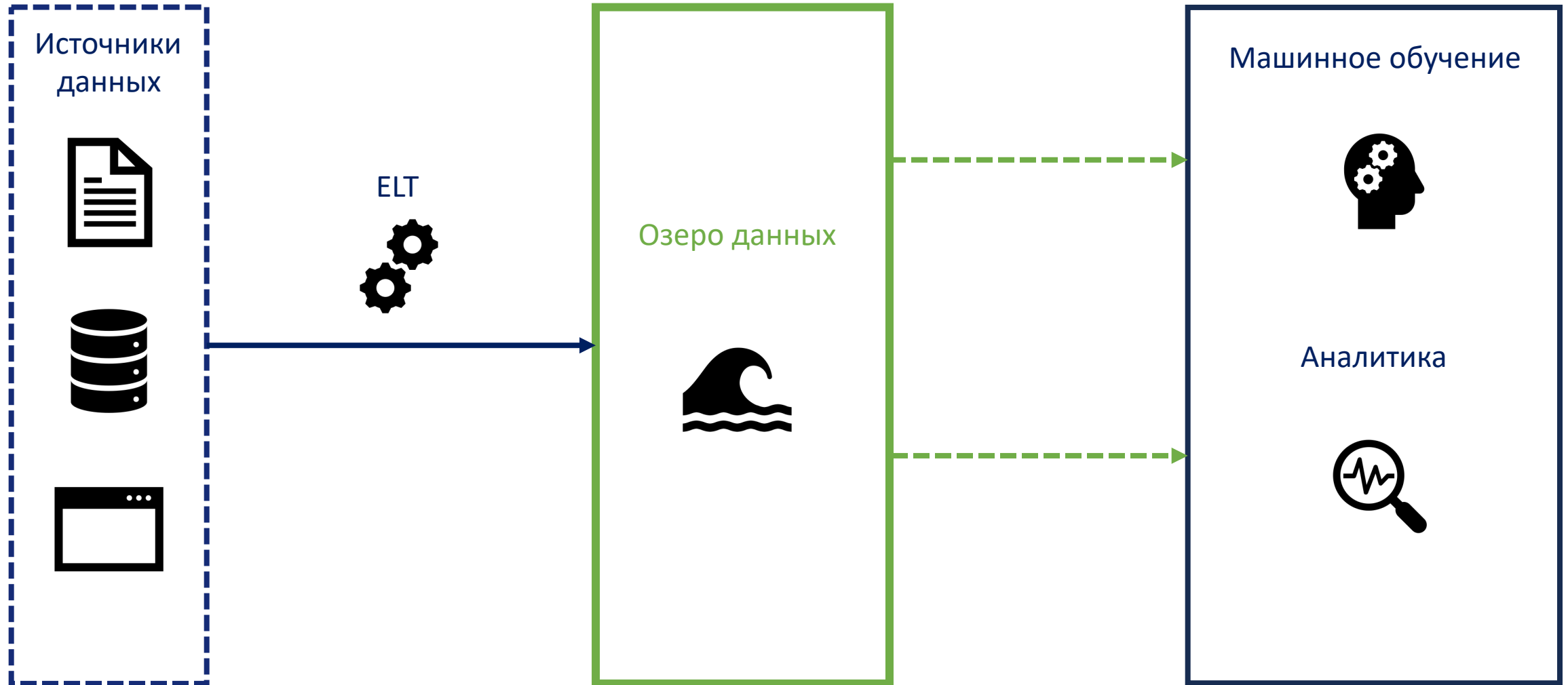


Разграничение доступа к данным;

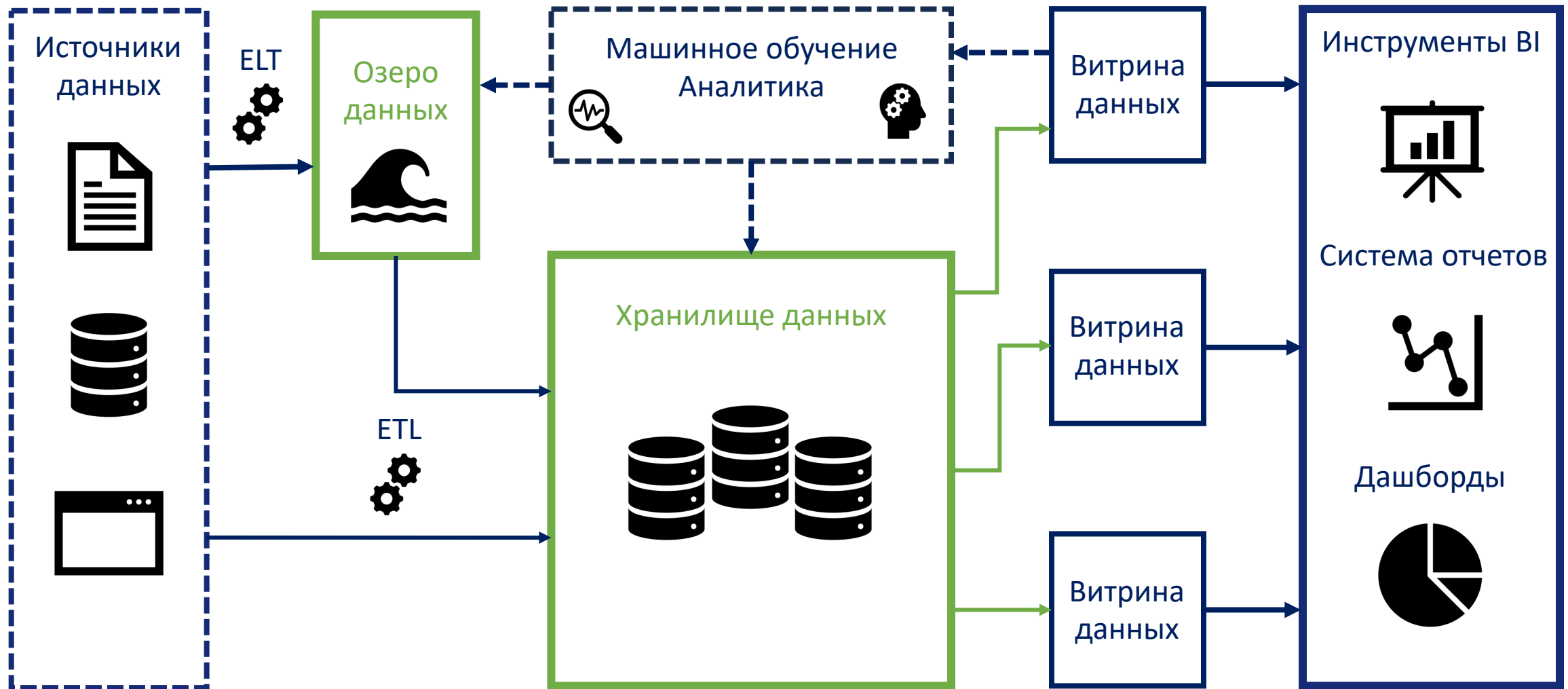
Структура DWH



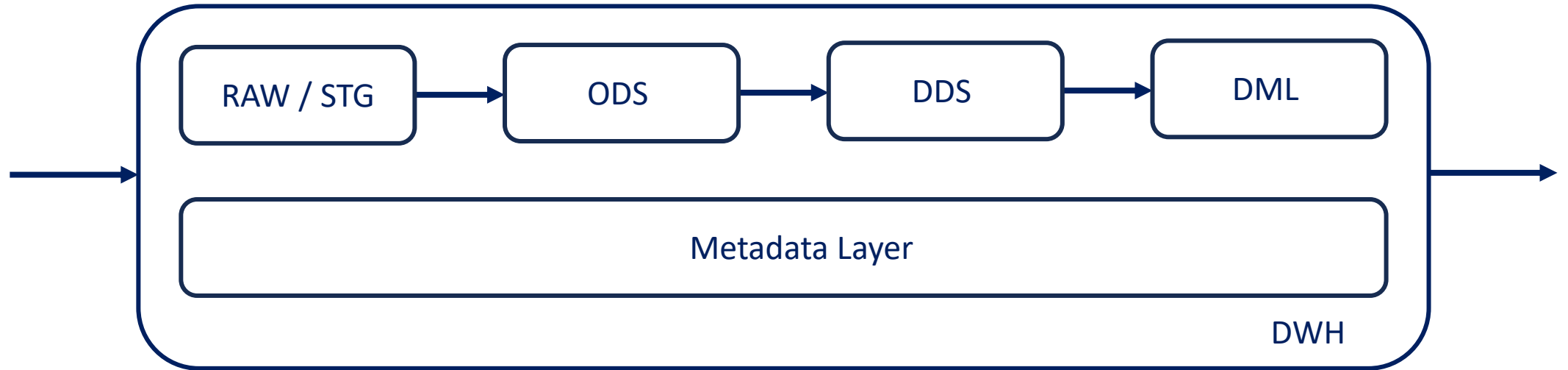
Структура Data Lake



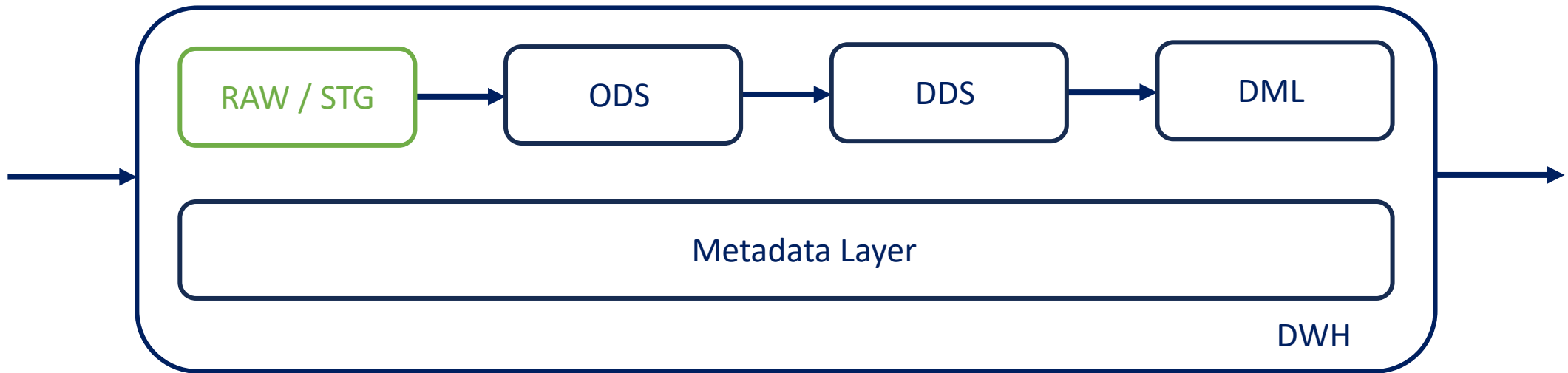
Структура LakeHouse



Слой DWH

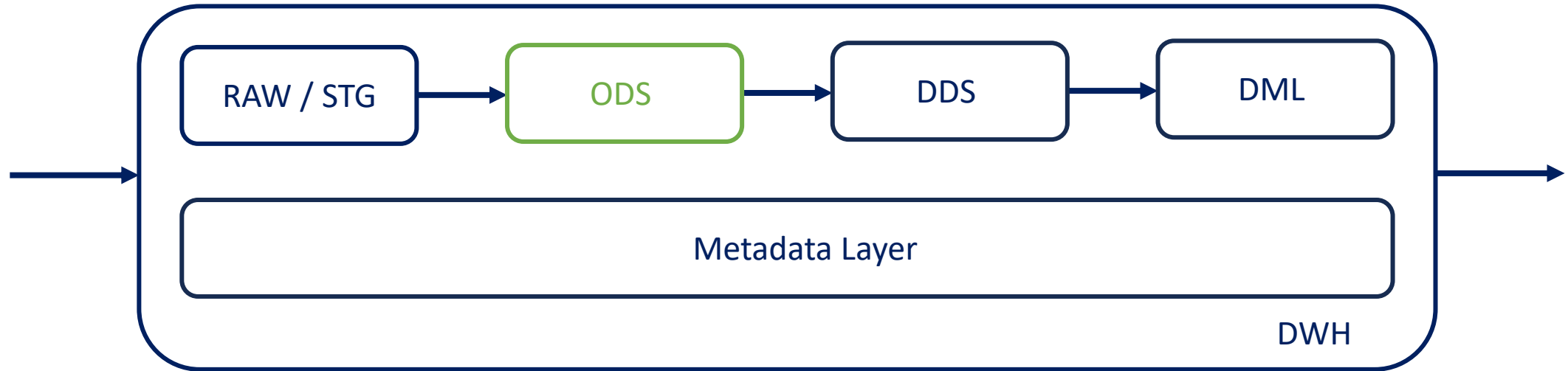


RAW / STAGING



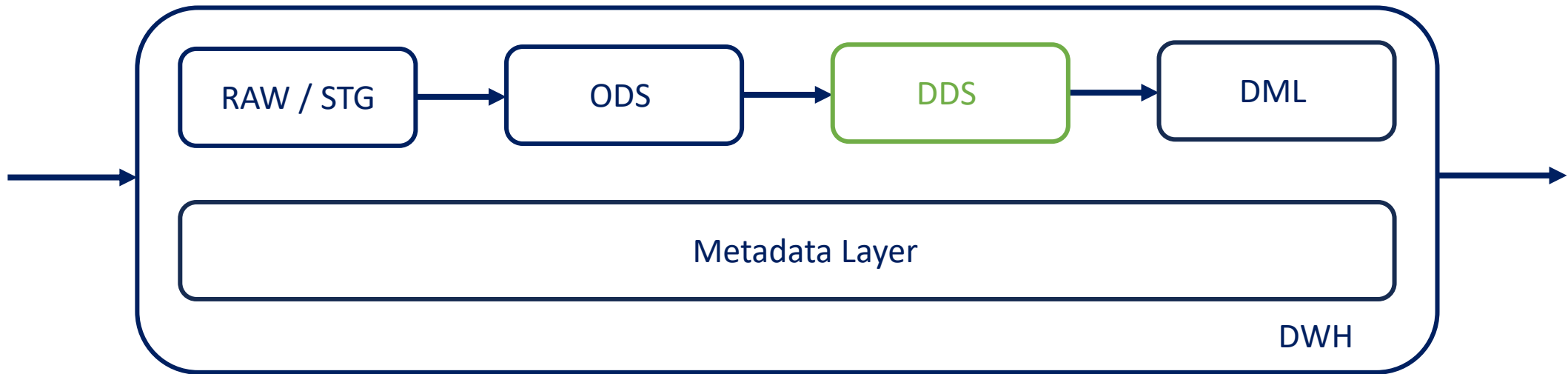
- Слой «сырых» данных;
- Используется для хранения данных из систем-источников;
- Форматы могут быть абсолютно различные: таблицы, csv, xml, json и т.д.
- Таблица, как правило, имеют префикс по названию источника.

ODS (Operational Data Definition)



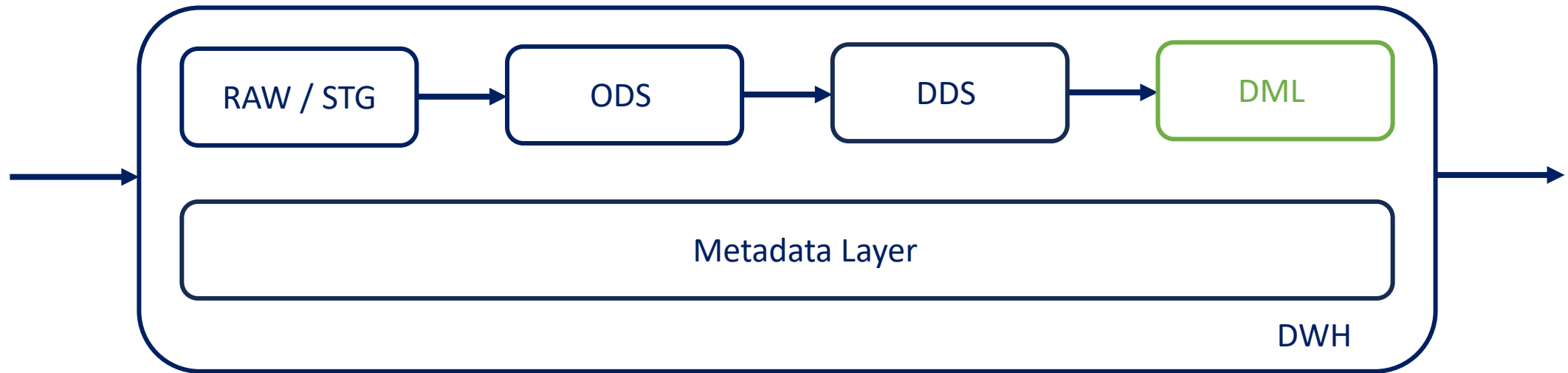
- Операционный слой;
- Загружаем данные в формате приближенном к реляционному;
- Чаще всего здесь происходит минимальная предобработка, генерация первичных ключей, формирование технических полей, преобразование типов данных.

DDS (Detail Data Store)



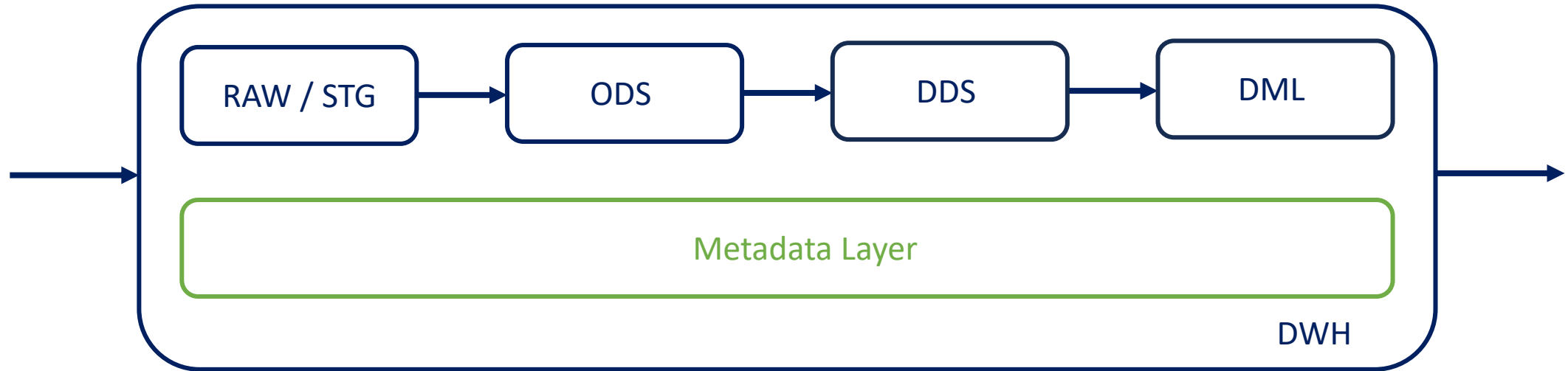
- Слой детальных данных (уровень детализации);
- Данные хранятся в нормализованном формате (3NF, Data Vault, Anchor Model);
- **Ключевой слой DWH;**
- Хранит историю изменения сущностей и связей между ними;

DML (Data Marts Layer)



- Слой витрин данных;
- Здесь формируются витрины данных и представления;
- Данные преобразуются в удобный для построения дашбордов вид;
- Данные этого слоя используются напрямую пользователями и BI-системами.

Metadata Layer



- Слой метаданных;
- Используется для осуществления загрузки данных и мониторинга загрузки;
- Позволяет анализировать метаданные и обеспечивать качество и целостность загружаемой информации.

Практика

