



SQL

Занятие 1

Введение и операторы

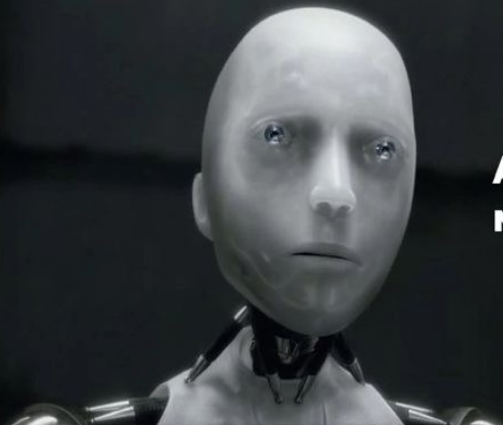
Бояр Владислав

Знакомство

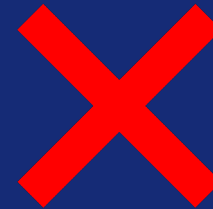
Может ли ИИ  
написать  
эффективный  
SQL-запрос?



А ты, типа,  
можешь?



# Результат



### О курсе:

- 10 занятий
- 9 домашних заданий
- Дедлайн – 1 неделя после выдачи дз
- 2 дз можно сдать в течение недели после дедлайна без штрафа (необходимо написать об этом в комментарии при сдаче)
- На последнем занятии итоговый тест
- Если у группы будут общие вопросы по дз или материалу курса, возможно проведение онлайн-консультаций

# Результат



### Формула оценки:

- дз и итоговый тест оцениваются по 10-балльной шкале
- дз зачтено, если набрано 4+ баллов
- Чтобы получить зачёт по модулю, нужно успешно выполнить 6 дз или 5 дз и итоговый тест

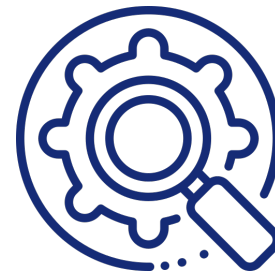
# Результат



# Занятие состоит из:



Теория



Практика

# Системы управления базами данных (СУБД)

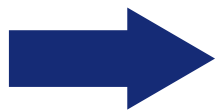


Почему СУБД и хранилища  
данных лучше Excel и Google  
таблиц?





# 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;



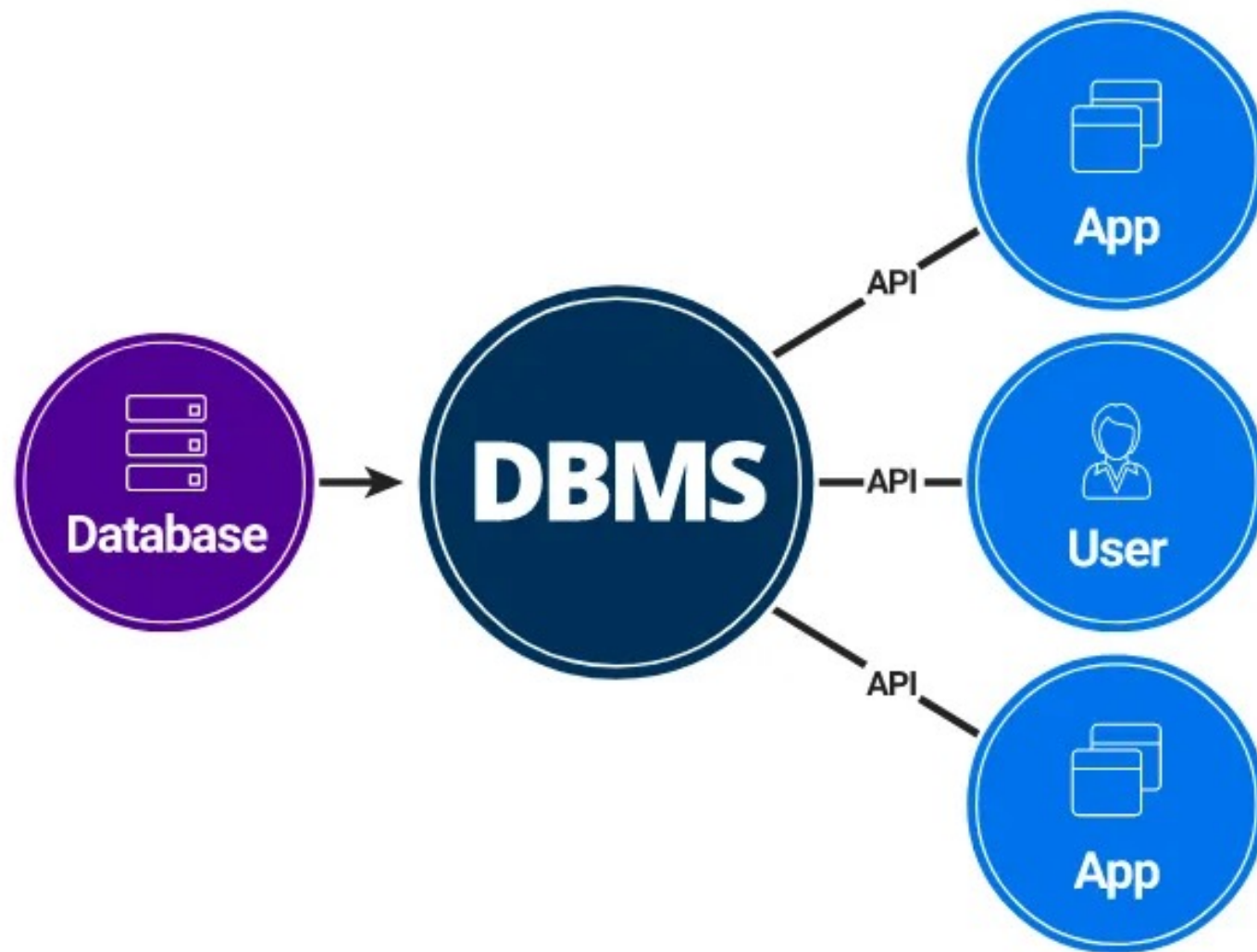
Проще интегрировать с другими источниками данных.

# 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



# Классификации СУБД

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



Реляционные



Нереляционные

# Реляционные СУБД

# Реляционные СУБД

**Реляционные СУБД** - представляют собой множество сущностей (таблиц) и связей между ними. Самый распространённый тип БД.

## Основные сущности:

- SQL;
- Таблицы и их составляющие (атрибуты, кортежи);
- Связи между таблицами (FK);
- Ограничения (constraints) – PK, Unique, NotNull, Default, Check;

# Реляционные СУБД. Пример

отношение, таблица

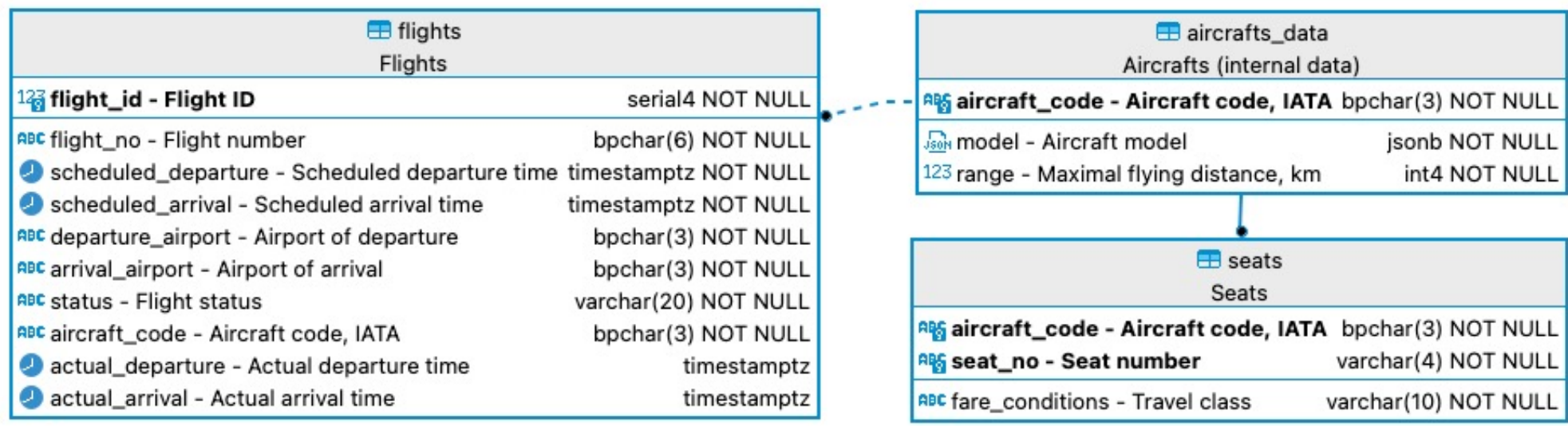
123 flight_id	ABC flight_no	🕒 scheduled_departure	🕒 scheduled_arrival	ABC departure_airport	ABC arrival_airport
19 585	PG0383	2017-08-17 14:50:00.000 +0300	2017-08-17 15:20:00.000 +0300	✈ DME	✈ BZK
20 605	PG0337	2017-08-25 12:45:00.000 +0300	2017-08-25 14:10:00.000 +0300	✈ DME	✈ ARH
21 648	PG0229	2017-09-11 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

атрибут, поле, колонка

# Связи между таблицами

## ER-диаграмма



Где на диаграмме **первичные** ключи, а где **вторичные**?



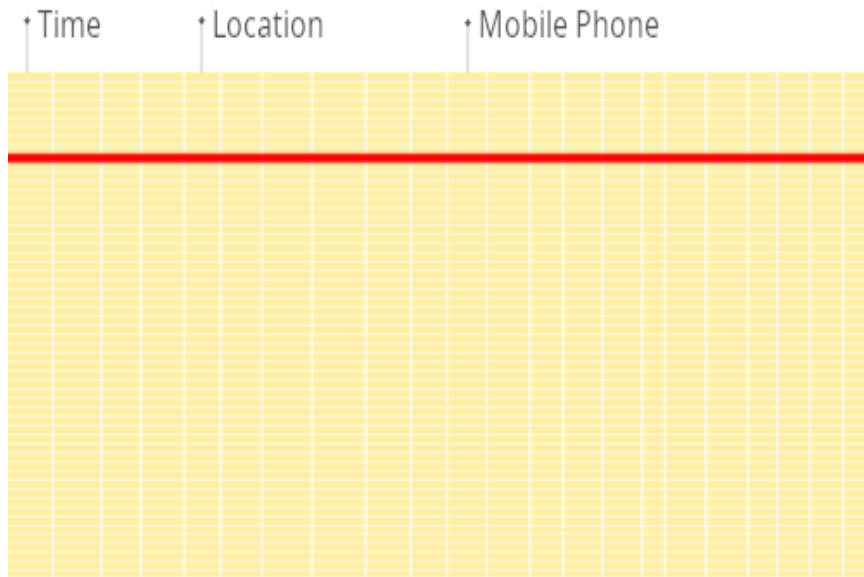
# Классификации СУБД

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

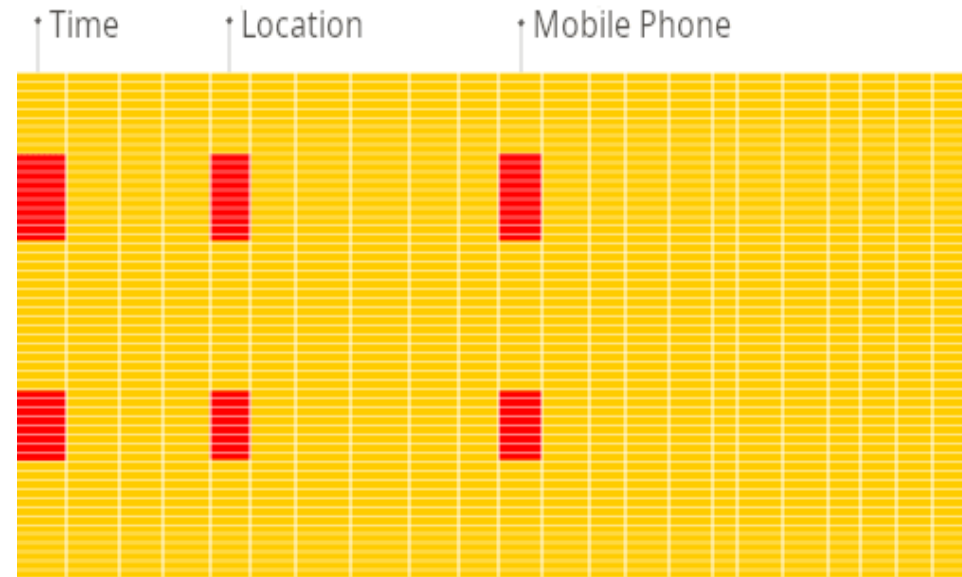


# Строковое и колоночное хранение

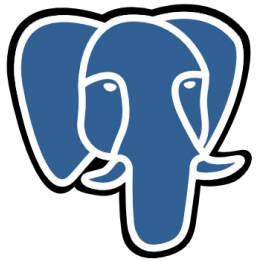
## Строковые БД



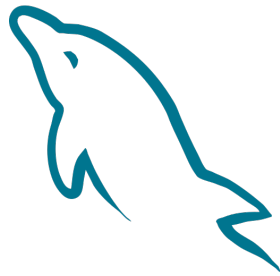
## Колоночные БД



# Примеры строковых СУБД



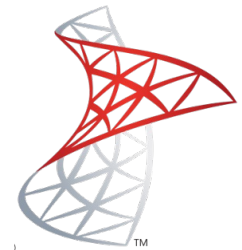
PostgreSQL



MySQL



Oracle

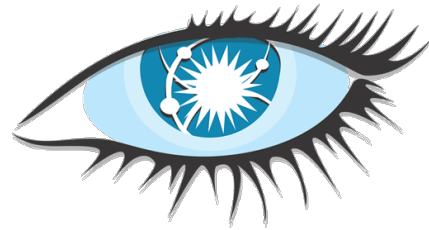


Microsoft SQL Server

# Примеры колоночных СУБД



ClickHouse



Apache Cassandra



Vertica



Apache HBase

# Примеры MPP СУБД



GreenPlum

VERTICA

Vertica

teradata.

Teradata




Arenadata

# Нереляционные СУБД

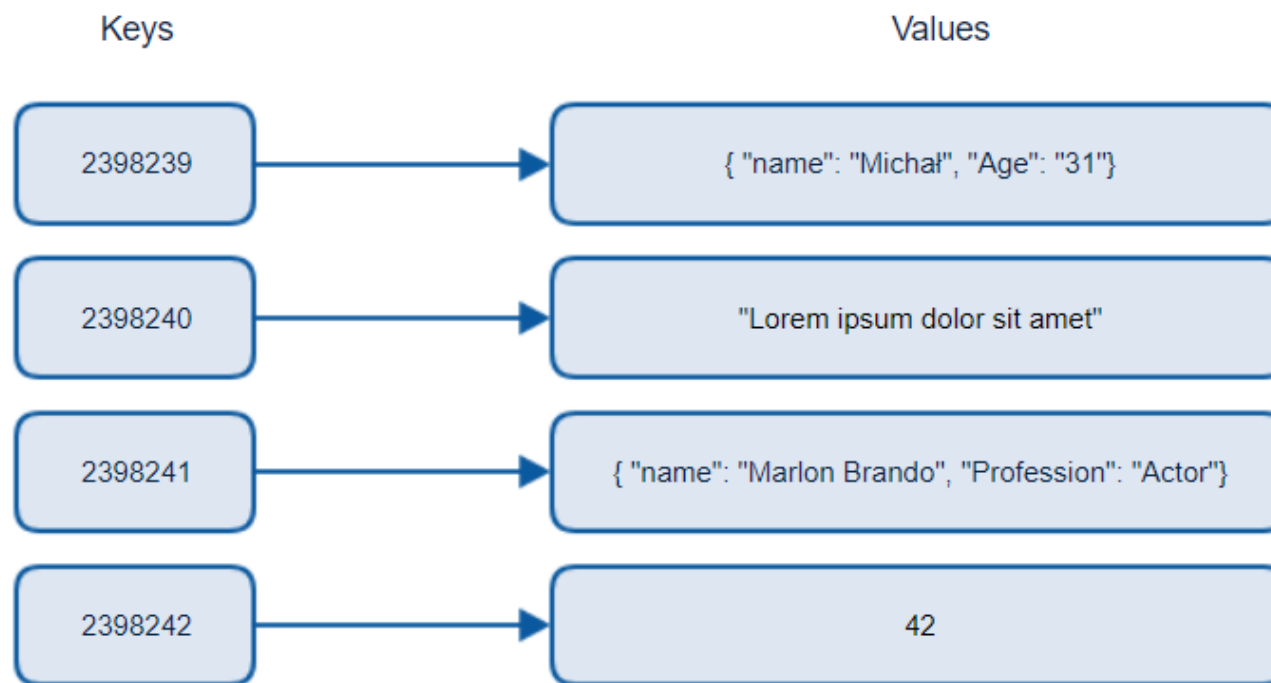
# Нереляционные СУБД (NoSQL)

**Нереляционные СУБД** для доступа не требуют SQL запросы:

- «Ключ-значение» (Key Value)
  - Документориентированные (Document 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.

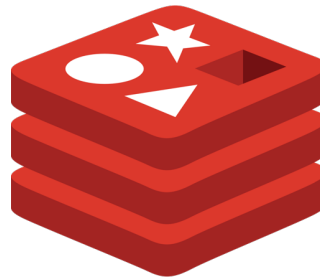
## Минусы:

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

# Примеры Key-Value DB



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

# Document Oriented

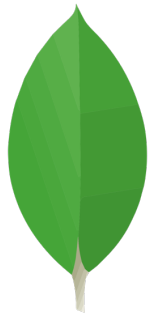
## Плюсы:

- + 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

# 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



SQL



# SQL

**SQL** (**S**tructured **Q**uery **L**anguage) – язык структурированных запросов.  
Универсальный способ доступа к данным реляционных СУБД.



# Группы операторов SQL

- **DML (Data Manipulation Language)** – манипуляции с данными
  - SELECT – выборка;
  - INSERT – вставка;
  - UPDATE – обновление;
  - DELETE – удаление.
- **DDL (Data Definition Language)** – работа с объектами БД
  - CREATE – создание;
  - ALTER – изменение;
  - DROP – удаление.
- **DCL (Data Control Language)** – определение доступа к данным
  - GRANT – предоставление прав;
  - REVOKE – отзыв прав;
  - DENY – запрет действий.
- **TCL (Transaction Control Language)** – управление транзакциями

# Основные операторы


# Основные операторы

```
SELECT [column_names]  
FROM schema_name.table_name
```

```
-- выбор данных из источника  
-- источник данных
```

# Основные операторы

```
SELECT [column_names]           -- выбор данных из источника
FROM schema_name.table_name    -- источник данных
WHERE column_name_1 > column_name_2 [<, =, !=] -- условие фильтрации
```



# Основные операторы

<b>SELECT</b> [column_names]	-- выбор данных из источника
<b>FROM</b> schema_name.table_name	-- источник данных
<b>WHERE</b> column_name_1 > column_name_2 [<, =, !=]	-- условие фильтрации
<b>ORDER BY</b> column_name_1 [ <b>ASC</b> , <b>DESC</b> ]	-- порядок сортировки

# Основные операторы

<b>SELECT</b> [column_names]	-- выбор данных из источника
<b>FROM</b> schema_name.table_name	-- источник данных
<b>WHERE</b> column_name_1 > column_name_2 [<, =, !=]	-- условие фильтрации
<b>ORDER BY</b> column_name_1 [ <b>ASC</b> , <b>DESC</b> ]	-- порядок сортировки
<b>LIMIT</b> 10	-- ограничение размера выборки

# Основные операторы

<b>SELECT</b> [column_names]	-- выбор данных из источника
<b>FROM</b> schema_name.table_name	-- источник данных
<b>WHERE</b> column_name_1 > column_name_2 [<, =, !=]	-- условие фильтрации
<b>ORDER BY</b> column_name_1 [ <b>ASC</b> , <b>DESC</b> ]	-- порядок сортировки
<b>LIMIT</b> 10	-- ограничение размера выборки
<b>OFFSET</b> 5	-- сдвиг выборки



# Основные операторы

<b>SELECT</b> [column_names]	-- выбор данных из источника
<b>FROM</b> schema_name_1.table_name_1 <b>AS</b> tn1	-- источник данных
<b>WHERE</b> column_name_1 > column_name_2 [<, =, !=]	-- условие фильтрации
<b>JOIN</b> schema_name_2.table_name_2 <b>AS</b> tn2	-- источник для объединения
<b>ON</b> tn1.id = tn2.id	-- условие объединения
<b>ORDER BY</b> column_name_1 [ <b>ASC</b> , <b>DESC</b> ]	-- порядок сортировки
<b>LIMIT</b> 10	-- ограничение размера выборки
<b>OFFSET</b> 5	-- сдвиг выборки

# Основные операторы

<b>SELECT</b> [column_names]	-- выбор данных из источника
<b>FROM</b> schema_name_1.table_name_1 <b>AS</b> tn1	-- источник данных
<b>WHERE</b> column_name_1 > column_name_2 [<, =, !=]	-- условие фильтрации
<b>JOIN</b> schema_name_2.table_name_2 <b>AS</b> tn2	-- источник для объединения
<b>ON</b> tn1.id = tn2.id	-- условие объединения
<b>GROUP BY</b> tn1.column_name	-- группировка данных
<b>ORDER BY</b> column_name_1 [ <b>ASC</b> , <b>DESC</b> ]	-- порядок сортировки
<b>LIMIT</b> 10	-- ограничение размера выборки
<b>OFFSET</b> 5	-- сдвиг выборки

# Основные операторы

<b>SELECT</b> [column_names]	-- выбор данных из источника
<b>FROM</b> schema_name_1.table_name_1 <b>AS</b> tn1	-- источник данных
<b>WHERE</b> column_name_1 > column_name_2 [<, =, !=]	-- условие фильтрации
<b>JOIN</b> schema_name_2.table_name_2 <b>AS</b> tn2	-- источник для объединения
<b>ON</b> tn1.id = tn2.id	-- условие объединения
<b>GROUP BY</b> tn1.column_name	-- группировка данных
<b>HAVING</b> sum(column_name) > 1	-- фильтрация после группировки
<b>ORDER BY</b> column_name_1 [ <b>ASC</b> , <b>DESC</b> ]	-- порядок сортировки
<b>LIMIT</b> 10	-- ограничение размера выборки
<b>OFFSET</b> 5	-- сдвиг выборки

Практика

