

Яндекс

Яндекс

Базы данных

Александр Ткаченко, разработчик интерфейсов

Немного теории



БД

База данных (БД) — это набор структурированной хранимой информации, предназначенный для использования программными средствами

БД != СУБД

Для работы с БД существуют СУБД

Система управления базами данных (СУБД) — комплекс программ, позволяющих создавать БД и манипулировать данными

Т.о. БД - это сами данные, СУБД - средство доступа к ним

БД != СУБД



Задачи СУБД

- › Хранение данных на дисках
- › Хранение данных в оперативной памяти (кэш)
- › Управление доступом (пользователи и их права)
- › API доступа
- › Служебные функции (логирование, резервное копирование...)

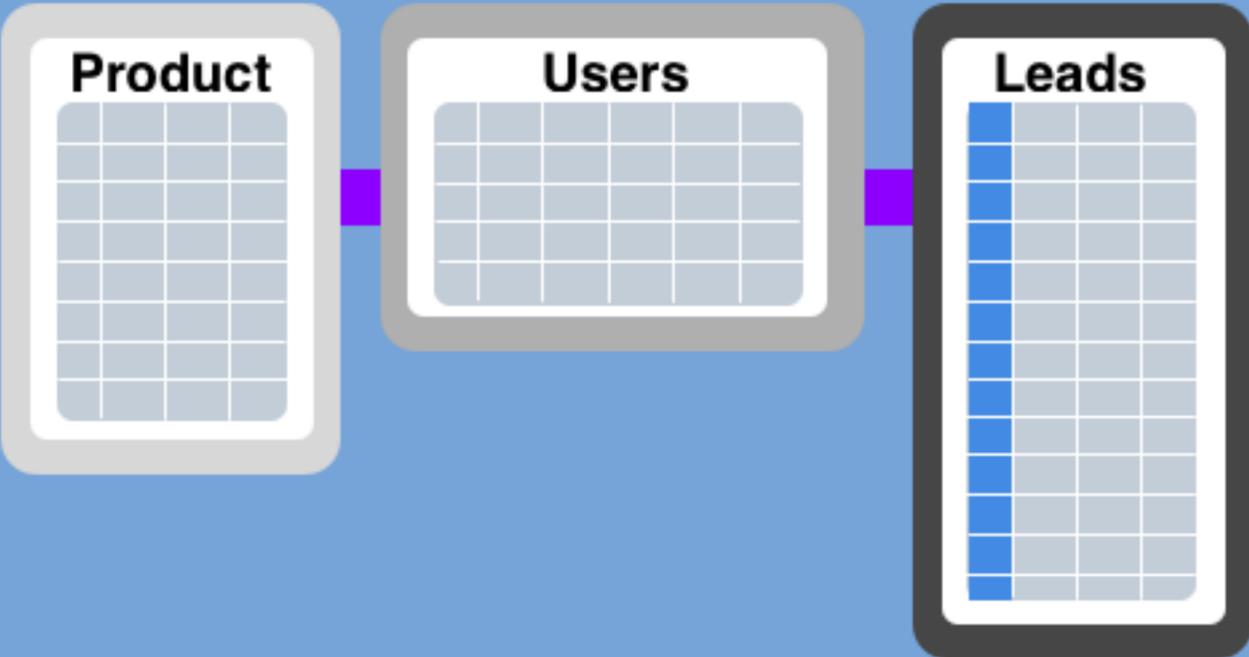
Типы СУБД

Реляционные (SQL) — традиционные базы на основе строго структурированных типизированных схем данных с заранее определенными отношениями (relations).
Для доступа к данным используется язык запросов SQL.

Нереляционные (NoSQL) — все остальные.
Отличаются повышенной скоростью и гибкостью в ущерб согласованности данных.
Для доступа к данным используются собственные API.

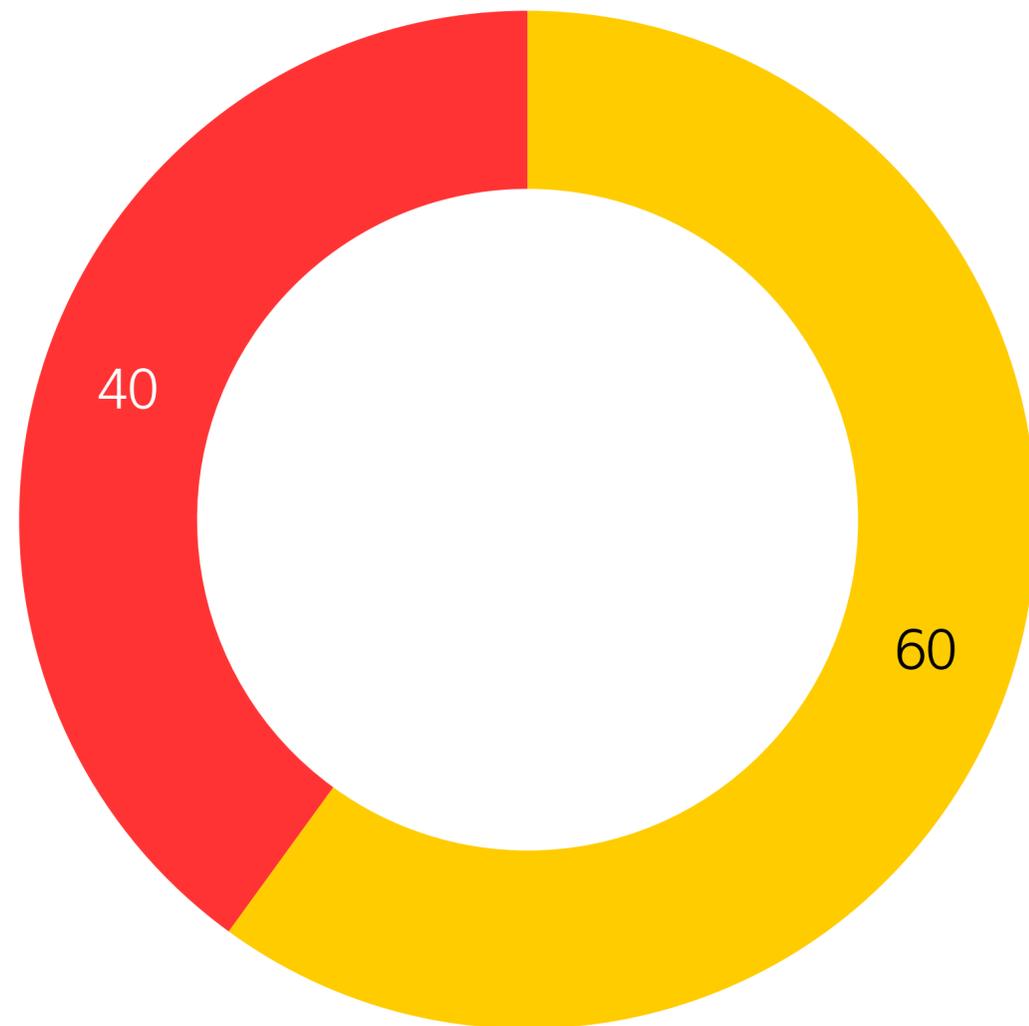
Типы СУБД

SQL vs. NoSQL



Типы СУБД

Доля рынка, 2019г.



Актуальность



Где используют БД в web-е

Везде *

* ну почти

Зачем это нам (фронтендерам)?

- › Общее представление о том, что происходит на бэкенде
- › Возможность без привлечения бэкенда запилить несложный сервер
- › Понадобится на хакатонах

MongoDB



MongoDB

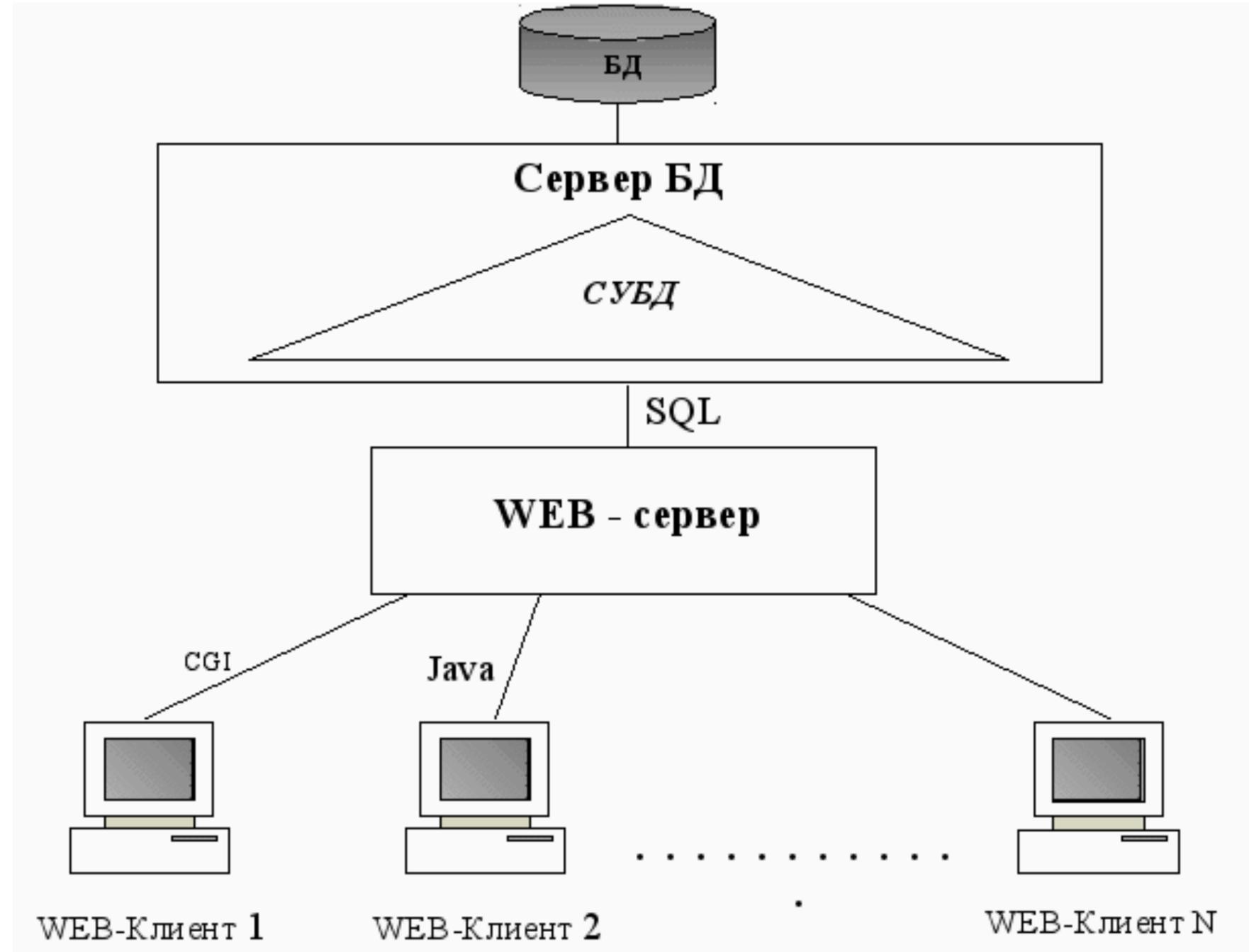
MongoDB - документоориентированная СУБД типа NoSQL

Почему именно она:

- › Низкий порог входа
- › Открытый исходный код
- › Кроссплатформенная
- › Популярность (~25% рынка*)

* <https://postgrespro.ru/list/thread-id/2429985>

Клиент-серверная архитектура СУБД



Установка и запуск сервера mongo

1. Устанавливаем и запускаем сервер **MongoDB Community Edition** по инструкции:

<https://docs.mongodb.com/v4.0/installation/>

2. Устанавливаем GUI клиент **Compass** по инструкции:

<https://docs.mongodb.com/compass/current/install/>

3. Проверяем работу сервера через Compass

MongoDB

- › Данные в mongo хранятся в виде JSON-подобных “документов”.
- › Документы объединены в коллекции.
- › У каждого документа есть уникальный идентификатор - поле `_id`

NodeJS + MongoDB

1. Устанавливаем драйвер **mongodb** в проект nodejs

```
npm install mongodb
```

2. Подключаемся к базе данных

```
const MongoClient = require("mongodb").MongoClient;
MongoClient.connect(url).then(function (client) {
    console.log('Соединение установлено');
});
```

Код на белом фоне

```
const MongoClient = require("mongodb").MongoClient;

// Адрес сервера
const url = 'mongodb://localhost:27017';

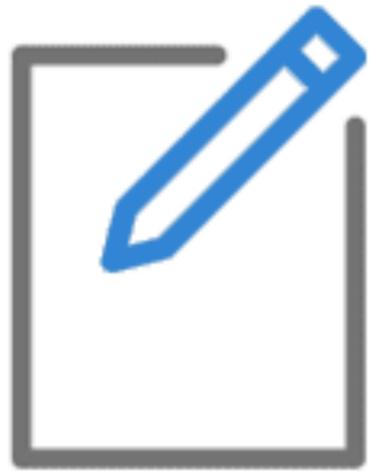
// Открываем соединение с сервером БД
MongoClient.connect(url)
  .then(function (client) {
    console.info('Соединение с сервером БД установлено');

    // Закрываем соединение
    client.close();
  })
  .catch(err => console.error('Не удалось подключиться к серверу БД', err));
```

CRUD



CRUD



CREATE



READ



UPDATE



DELETE

C

R

U

D

Create

```
function create(db) {  
  const doc = { name: 'Мой тестовый документ' };  
  
  // Получаем доступ к коллекции  
  const collection = db.collection('Temp');  
  // Сохраняем документ  
  return collection.insertOne(doc)  
    .then(() => console.log('Заметка сохранена'))  
    .catch(err => console.error('Не удалось сохранить заметку', err));  
}
```

Read

```
function read(db) {  
  // Получаем доступ к коллекции  
  const collection = db.collection('Temp');  
  
  // Читаем все документы  
  return collection.find({}).toArray()  
    .then(docs => console.log('Все документы', docs))  
    .catch(err => console.error('Не удалось прочитать документы', err));  
}
```

Update

```
function update(db) {  
  // Получаем доступ к коллекции  
  const collection = db.collection('Temp');  
  // Обновляем документ  
  return collection  
    .updateOne(  
      { _id: ObjectId('5cd175ade90d85871d46e00f') },  
      { $set: { name: 'бла-бла' } }  
    )  
    .then(result => console.log('Результат обновления', result.result))  
    .catch(err => console.error('Не удалось обновить документы', err));  
}
```

Delete

```
function remove(db) {  
  // Получаем доступ к коллекции  
  const collection = db.collection('Temp');  
  // Обновляем документ  
  return collection  
    .deleteOne({ _id: ObjectId('5cd175ade90d85871d46efff') })  
    .then(result => console.log('Результат удаления', result))  
    .catch(err => console.error('Не удалось удалить документ', err));  
}
```



Приложение с заметками

**Хотите больше?
SQL-СУБД: SQLite,
MySQL, PostgreSQL**



Яндекс

Спасибо

Ткаченко Александр

Разработчик интерфейсов



asany4@yandex-team.ru



[@asanych](https://www.instagram.com/asanych)

