

Введение в базы данных

Владислав Хорев
Ведущий программист в компании Andersen



Проверка связи



Если у вас нет звука:

- убедитесь, что на вашем устройстве и на колонках включён звук
- обновите страницу вебинара (или закройте страницу и заново присоединитесь к вебинару)
- откройте вебинар в другом браузере
- перезагрузите компьютер (ноутбук) и заново попытайтесь зайти



Поставьте в чат:

-  если меня видно и слышно
-  если нет

Владислав Хорев

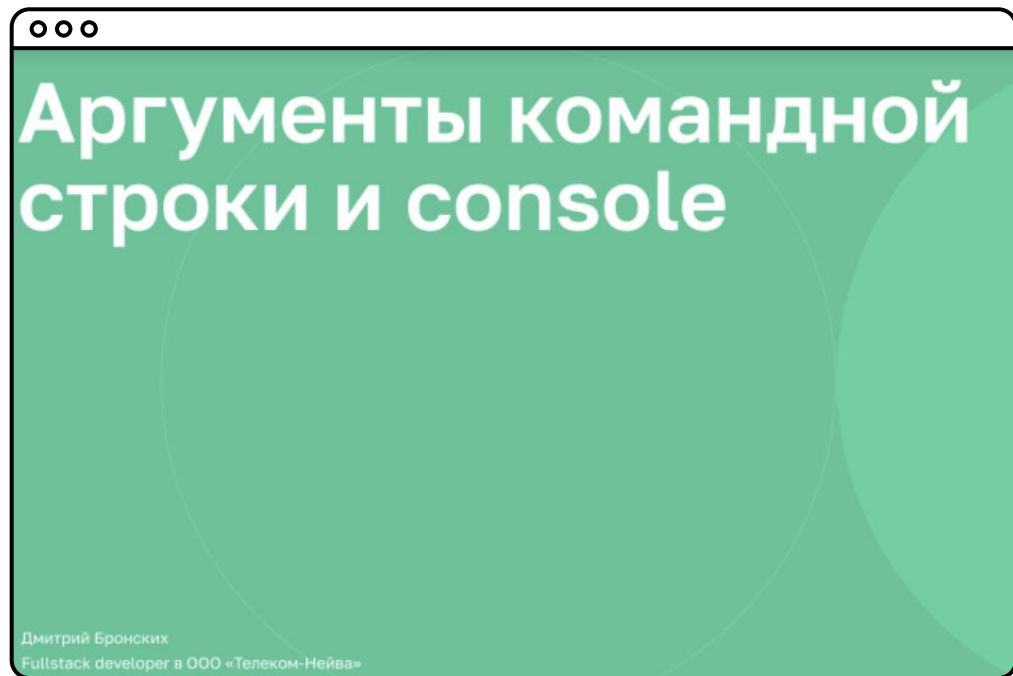
О спикере:

- Ведущий программист в компании Andersen
- Работает в IT с 2011 года
- Опыт разработки на C++ более 11 лет



Вебинары и видео

Записи, статьи и презентации экспертов — вся необходимая информация, чтобы ваше обучение было **максимально эффективным**



ooo

Аргументы командной строки и console

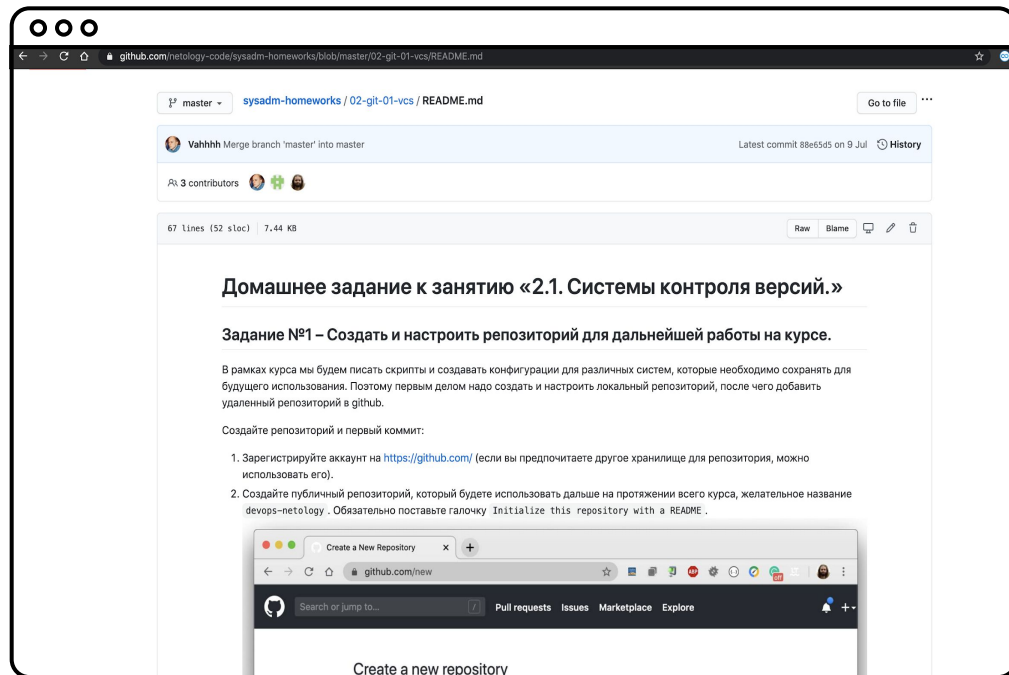
Дмитрий Бронских
Fullstack developer в ООО «Телеком-Нейва»

Лекции будут в формате видеолекций и вебинаров. Это поможет вам влиться в обучение в своём темпе.

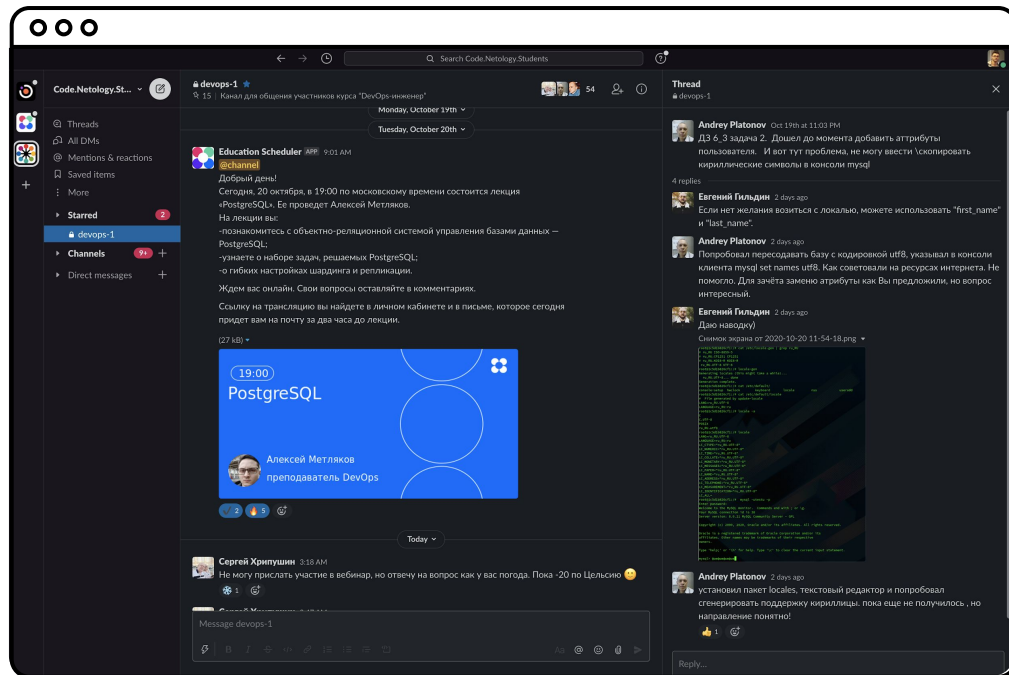
В конце модуля мы разбираем вопросы по изученным темам

Практика

После каждого занятия у вас будет **практическое задание** — решить 1–2 задачи



Поддержка



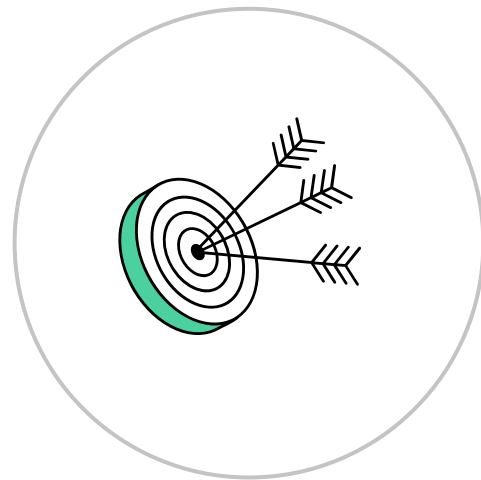
При обучении вас поддержат преподаватели и аспиранты, координатор, сокурсники.

Задавайте вопросы, делитесь обратной связью.

Эта атмосфера поможет вам влиться в IT-сообщество

Цели занятия

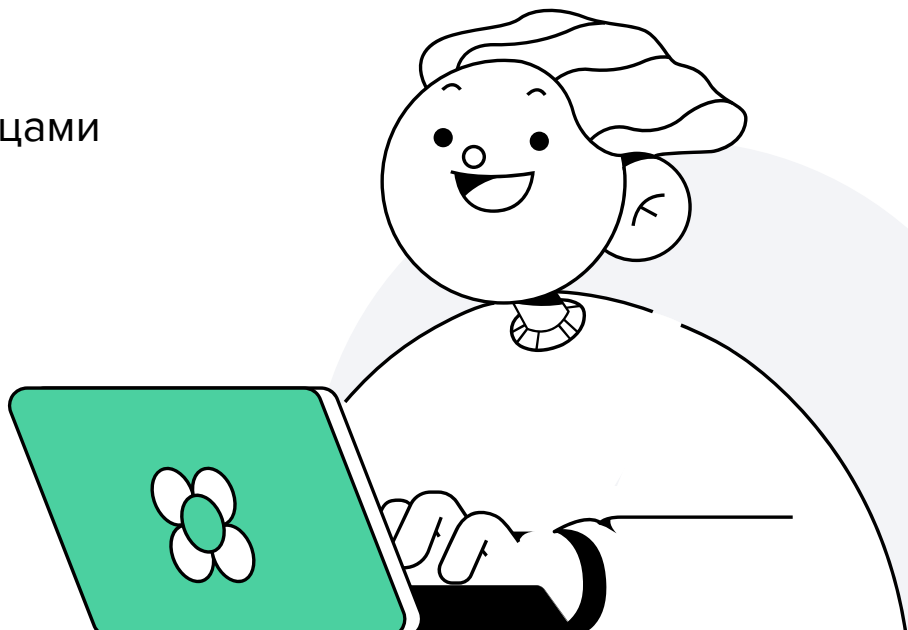
- Познакомимся с понятием базы данных
- Узнаем о типах БД и СУБД
- Спроектируем базу данных интернет-магазина



План занятия

- 1 БД и СУБД
- 2 Типы БД
- 3 Таблицы, атрибуты, кортежи
- 4 Первичный ключ и связи между таблицами
- 5 Нормальные формы
- 6 Домашнее задание

*Нажми на нужный раздел для перехода



БД и СУБД



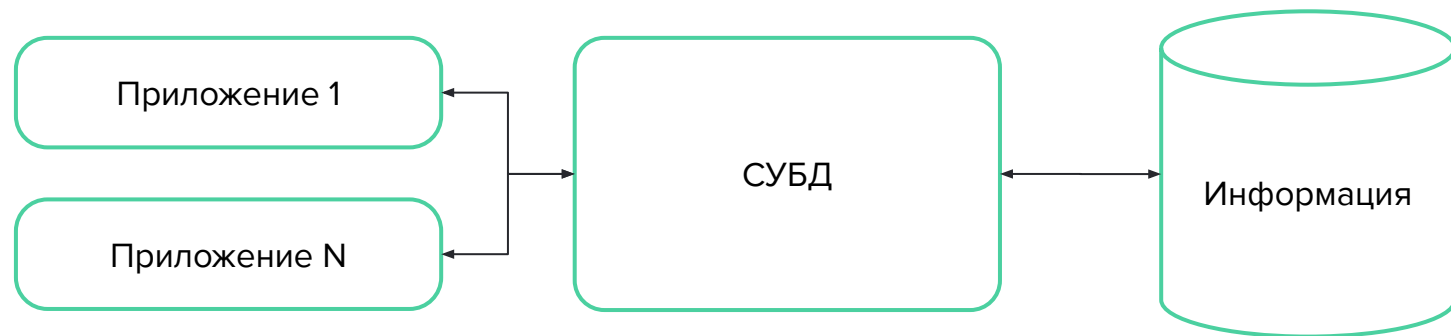
1



База данных — набор взаимосвязанных данных и правила хранения этих данных

Система управления базами данных (СУБД)

Система управления базами данных — комплекс программных средств для управления данными



Аналогия

1

Архив с данными

это **информация**, которую
хотим хранить

2

Архивариус

СУБД, через него идут все
манипуляции с данными,
будь то удаление,
добавление или получение

3

Посетитель

программа, которой
необходимо производить
операции с данными

Аналогия

1

Архив с данными

это информация, которую
хотим хранить

2

Архивариус

СУБД, через него идут все
манипуляции с данными,
будь то удаление,
добавление или получение

3

Посетитель

программа, которой
необходимо производить
операции с данными

Аналогия

1

Архив с данными

это информация, которую
хотим хранить

2

Архивариус

СУБД, через него идут все
манипуляции с данными,
будь то удаление,
добавление или получение

3

Посетитель

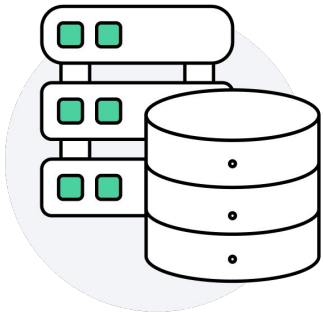
программа, которой
необходимо производить
операции с данными

Типы СУБД

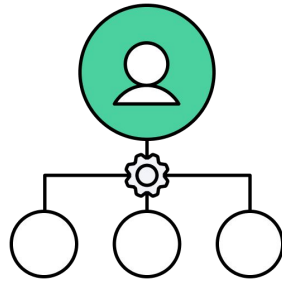


2

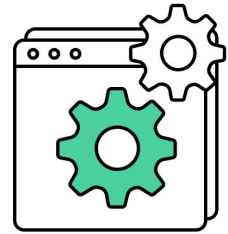
Типы СУБД



Файл-серверные



Клиент-серверные



Встраиваемые

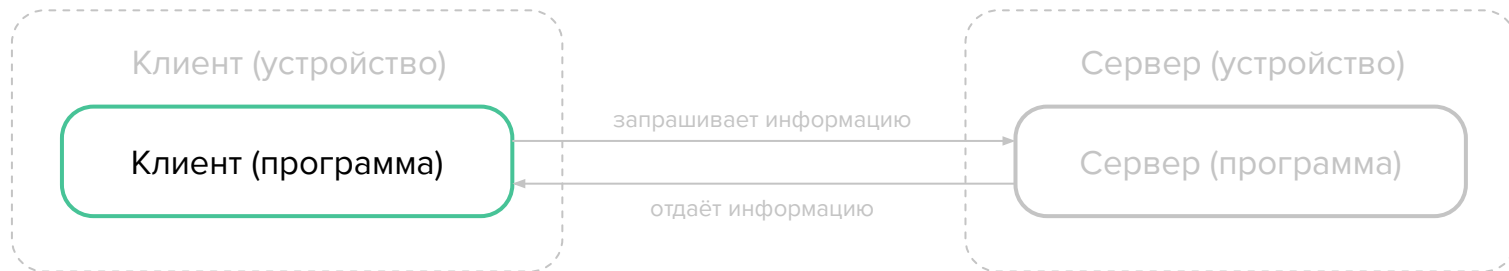
Клиент и сервер

Клиент:

- 1 Программа, которая хочет получить информацию
- 2 Физическое устройство, на котором работает программа-клиент

Сервер:

- 1 Специальная программа, которая даёт информацию
- 2 Физическое устройство, на котором запущена программа-сервер



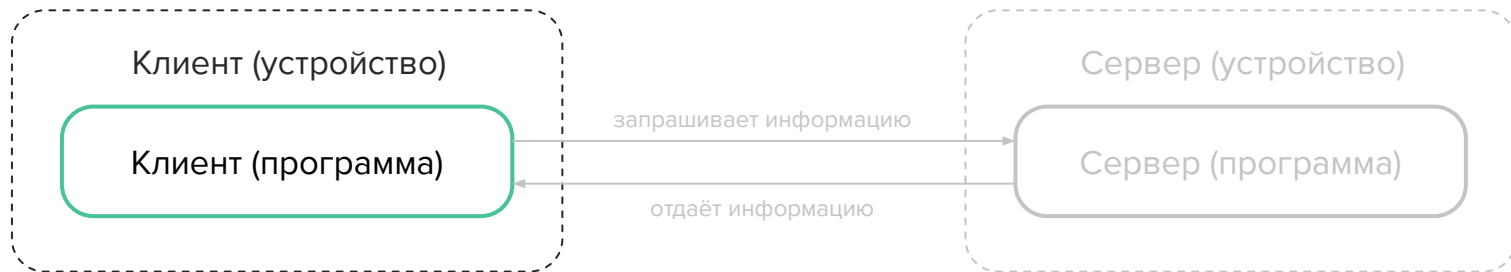
Клиент и сервер

Клиент:

- 1 Программа, которая хочет получить информацию
- 2 Физическое устройство, на котором работает программа-клиент

Сервер:

- 1 Специальная программа, которая даёт информацию
- 2 Физическое устройство, на котором запущена программа-сервер



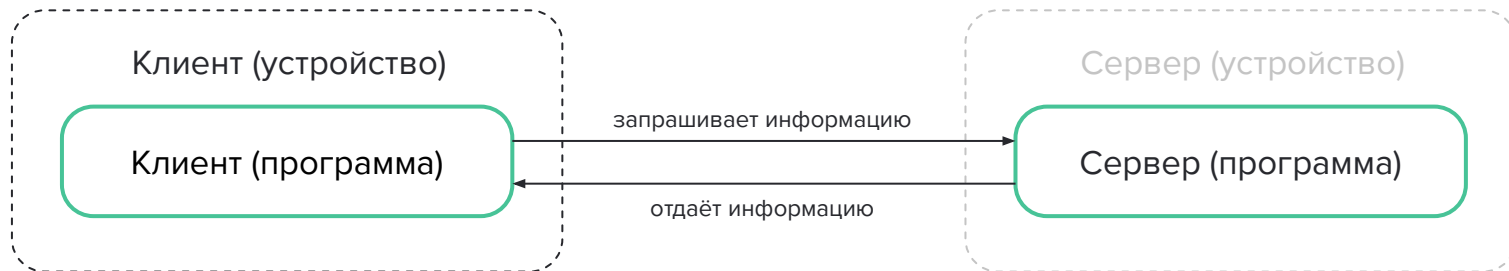
Клиент и сервер

Клиент:

- 1 Программа, которая хочет получить информацию
- 2 Физическое устройство, на котором работает программа-клиент

Сервер:

- 1 Специальная программа, которая даёт информацию
- 2 Физическое устройство, на котором запущена программа-сервер



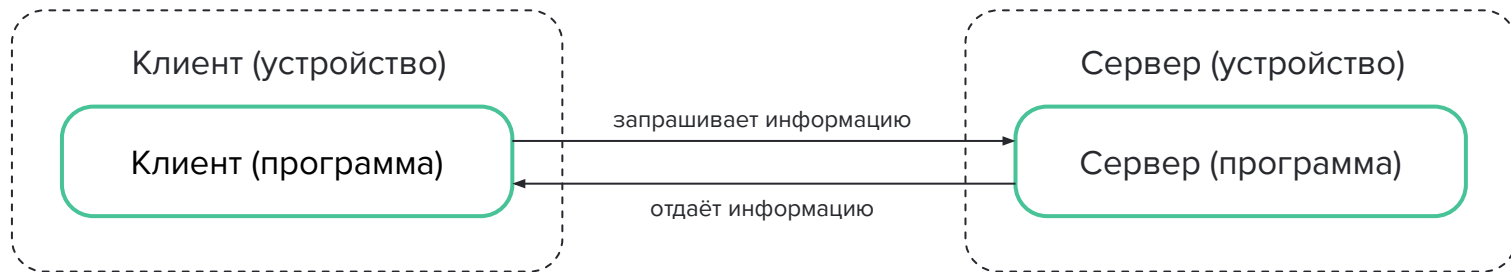
Клиент и сервер

Клиент:

- 1 Программа, которая хочет получить информацию
- 2 Физическое устройство, на котором работает программа-клиент

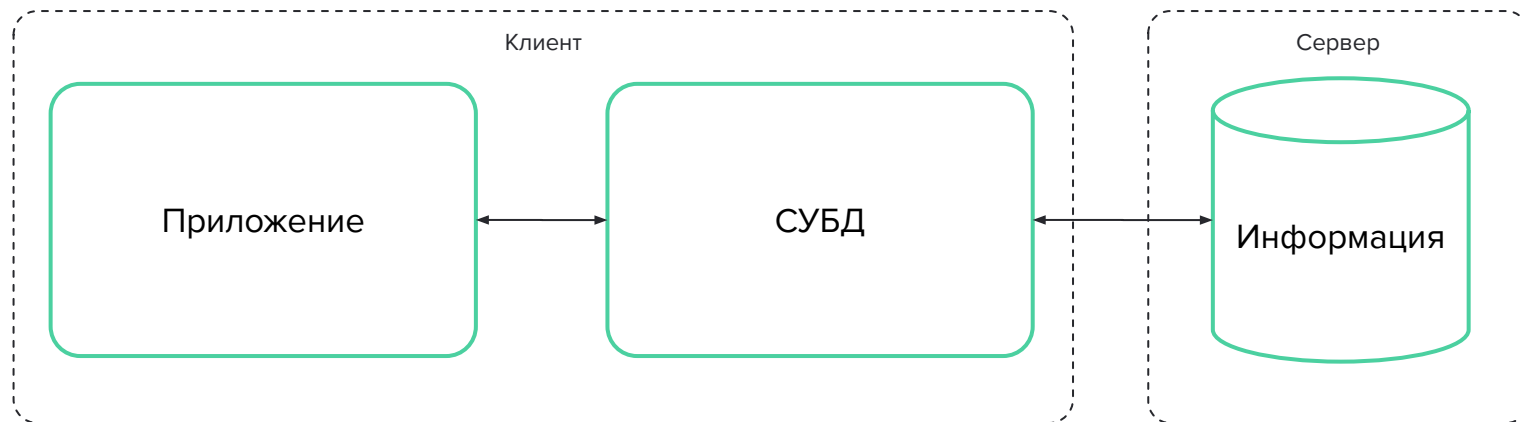
Сервер:

- 1 Специальная программа, которая даёт информацию
- 2 Физическое устройство, на котором запущена программа-сервер



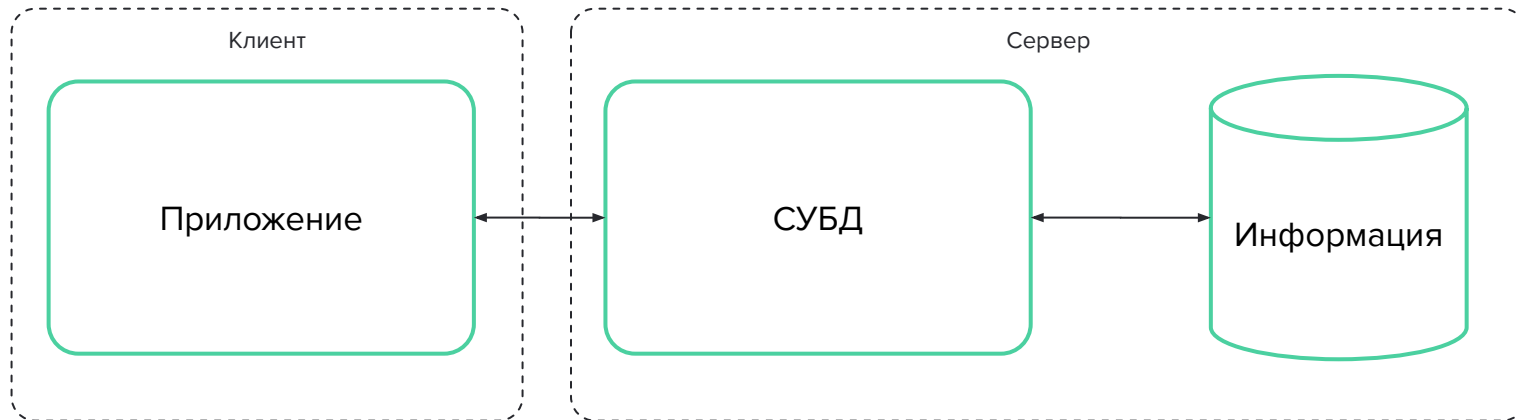
Файл-серверные СУБД

Файлы с информацией хранятся на сервере, а СУБД на клиенте



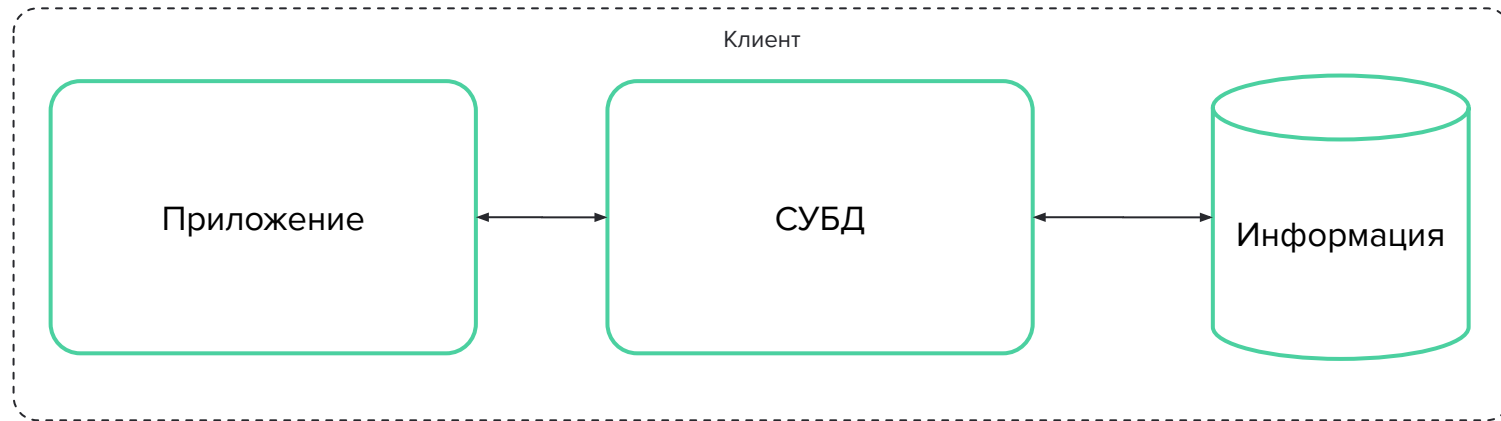
Клиент-серверные СУБД

Файлы с информацией и СУБД находятся на сервере, а клиент обращается за информацией через легковесную вспомогательную программу



Встраиваемые СУБД

Файлы и СУБД хранятся на клиенте



Сравнение типов СУБД

| Тип СУБД | Плюсы | Минусы |
|------------------|---|---|
| Файл-серверные | Сервер может быть обычным файловым хранилищем | <ul style="list-style-type: none">• Плохо параллелятся действия от разных клиентов• Требуется установка СУБД на каждом клиенте |
| Клиент-серверные | <ul style="list-style-type: none">• На клиенте не надо устанавливать СУБД• Хорошо параллелятся действия от разных клиентов | Сервер должен быть достаточно производительным => дорого |
| Встраиваемые | Не нужен сервер | Подходит только для локального хранения |

Типы баз данных

Реляционные — это БД, в которых информация строго структурирована и связана с другой информацией жёсткими правилами

Пример:

Microsoft Access

PostgreSQL

Microsoft SQL

SQLite

MySQL

Базы данных SQL

Популярные СУБД SQL, которые чаще всего используются:



Типы баз данных

Нереляционные (NoSQL) — это БД, в которых жёсткие ограничения не требуются ни по структуре, ни по связи между информацией

Пример:

Redis

MongoDB

Cassandra

Базы данных NoSQL

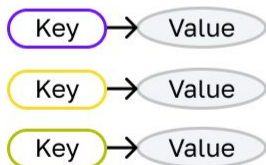
Популярные NoSQL СУБД, которые чаще всего используются:



Типы нереляционных баз данных

«Ключ-значение»

- Redis
- Berkley DB
- MemcacheDB



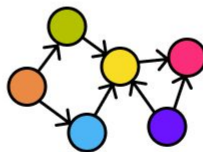
Колоночные

- Cassandra
- HBase



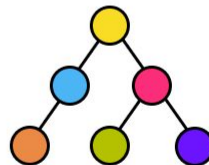
Графовые

- Neo4j
- OrientDB



Документо-ориентированные

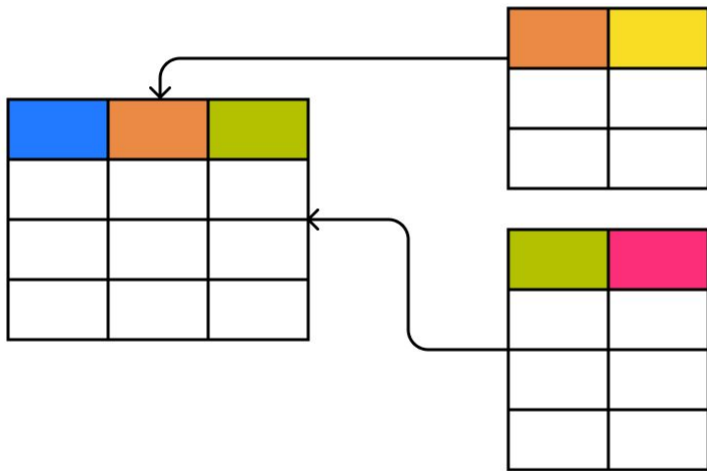
- MongoDB
- CouchDB



Сравнение SQL и NoSQL

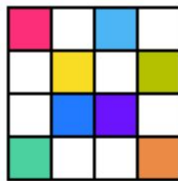
SQL

Регуляционные СУБД

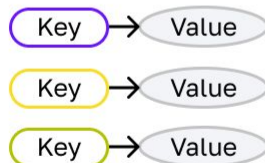


NoSQL

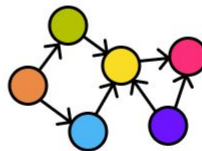
Колоночные



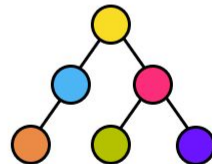
«Ключ-значение»



Графовые



Документо-ориентированные



Таблицы, атрибуты, кортежи

3

Реляционные базы данных

Сущность — описываемый объект

Например, мы хотим рассмотреть успеваемость студентов на курсе



Реляционные базы данных

Отношение — таблица

| id | name | гра |
|----|------|------|
| 1 | Егор | 4.82 |
| 2 | Егор | 4.11 |
| 3 | Егор | 3.88 |

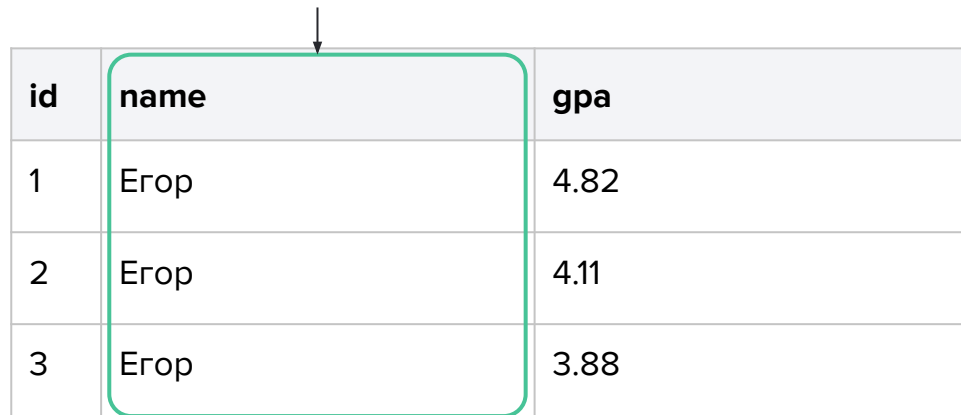
Пример отношения «Успеваемость студентов»

Реляционные базы данных

→ **Атрибут (или поле)** — столбец

→ **Запись (или кортеж)** — строка

Атрибут



| id | name | gpa |
|----|------|------|
| 1 | Егор | 4.82 |
| 2 | Егор | 4.11 |
| 3 | Егор | 3.88 |

Пример отношения «Успеваемость студентов»

Реляционные базы данных

→ Атрибут (или поле) — столбец

→ **Запись (или кортеж)** — строка

Атрибут
↓

| id | name | gpa |
|----|------|------|
| 1 | Егор | 4.82 |
| 2 | Егор | 4.11 |
| 3 | Егор | 3.88 |

Кортеж →

Пример отношения «Успеваемость студентов»

Первичный ключ и связи между таблицами



4

Primary key

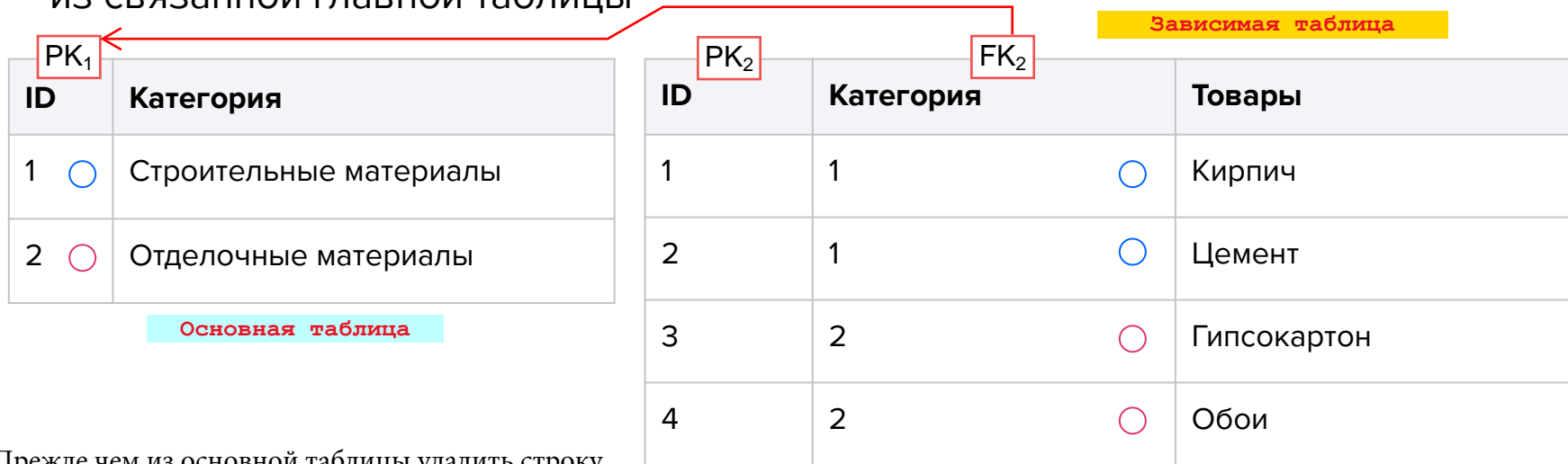
Primary key (первичный ключ) — столбец или набор столбцов, которые уникальным образом идентифицируют строку в рамках всей таблицы:

| ID PK | Категория |
|-------|------------------------|
| 1 | Строительные материалы |
| 2 | Отделочные материалы |

! Наличие первичного ключа — необязательное требование для таблицы, но в большинстве случаев он будет

Foreign key

- Внешние ключи позволяют установить связи между таблицами
- Внешний ключ устанавливается для столбцов из зависимой (подчинённой) таблицы и указывает на один из столбцов из главной таблицы
- Как правило, внешний ключ указывает на первичный ключ из связанной главной таблицы



Прежде чем из основной таблицы удалить строку, необходимо удалить все зависимые значения в зависимой таблице!

Нормальные формы



5

Нормальные формы

Нормальная форма — требование, предъявляемое к структуре таблиц в теории реляционных баз данных для устранения из базы избыточных функциональных зависимостей между атрибутами

! **Цель нормализации** — исключить избыточное дублирование данных, которое является причиной аномалий, возникающих при добавлении, редактировании и удалении кортежей

Первая нормальная форма (1НФ)

Переменная отношения находится **в первой нормальной форме (1НФ)**, когда в любом допустимом значении отношения каждый его кортеж (строка) содержит только одно значение для каждого из атрибутов

| Категория | Товары |
|------------------------|---------------------------|
| Строительные материалы | Кирпич, цемент, гвозди |
| Отделочные материалы | Гипсокартон, обои, краска |

**Не нормализованная
форма!**

| Категория | Товары |
|------------------------|-------------|
| Строительные материалы | Кирпич |
| Строительные материалы | Цемент |
| Строительные материалы | Гвозди |
| Отделочные материалы | Гипсокартон |
| Отделочные материалы | Обои |
| Отделочные материалы | Краска |

1НФ

Вторая нормальная форма (2НФ)

Переменная отношения находится **во второй нормальной форме**, когда соблюдаются все условия первой нормальной формы и каждый неключевой атрибут неприводимо (функционально полно) зависит от её потенциального ключа

| Категория | Дата акции | Скидка | Товары |
|------------------------|------------|--------|-------------|
| Строительные материалы | 21.11.2020 | 10% | Кирпич |
| Строительные материалы | 21.11.2020 | 10% | Цемент |
| Отделочные материалы | 25.12.2020 | 20% | Гипсокартон |
| Отделочные материалы | 25.12.2020 | 20% | Обои |

1НФ

| Категория | Товары |
|------------------------|-------------|
| Строительные материалы | Кирпич |
| Строительные материалы | Цемент |
| Отделочные материалы | Гипсокартон |
| Отделочные материалы | Обои |

| Категория | Дата акции | Скидка |
|------------------------|------------|--------|
| Строительные материалы | 21.11.2020 | 10% |
| Отделочные материалы | 25.12.2020 | 20% |

2НФ

Третья нормальная форма (3НФ)

Переменная отношения находится **в третьей нормальной форме** тогда и только тогда, когда соблюдаются все условия второй нормальной формы и отсутствуют функциональные зависимости неключевых атрибутов от ключевых

Привести к 3НФ — значит выделить из таблицы первичный ключ, который будет уникально идентифицировать запись в таблице

| PK | | FK ₂ | | |
|-----------------|------------------------|-----------------|-----------|-------------|
| ID | Категория | ID | Категория | Товары |
| 1 | Строительные материалы | PK ₂ | 1 | Кирпич |
| 2 | Отделочные материалы | | 1 | Цемент |
| FK ₁ | | 3 | 2 | Гипсокартон |
| PK ₁ | | 4 | 2 | Обои |
| ID | Категория | Дата акции | Скидка | |
| 1 | 1 | 21.11.2020 | 10% | |
| 2 | 2 | 25.12.2020 | 20% | |

| Категория | Товары |
|------------------------|---------------------------|
| Строительные материалы | Кирпич, цемент, гвозди |
| Отделочные материалы | Гипсокартон, обои, краска |

Остальные нормальные формы

- Нормальная форма Бойса — Кодда (НФБК)
- Четвёртая нормальная форма (4НФ)
- Пятая нормальная форма (5НФ)
- Шестая нормальная форма (6НФ)

Плюсы и минусы нормализации

Плюсы

- ⊕ Декомпозиция информации
- ⊕ Строгое хранение данных без возможности хранить дублированную и противоречивую информацию

Минусы

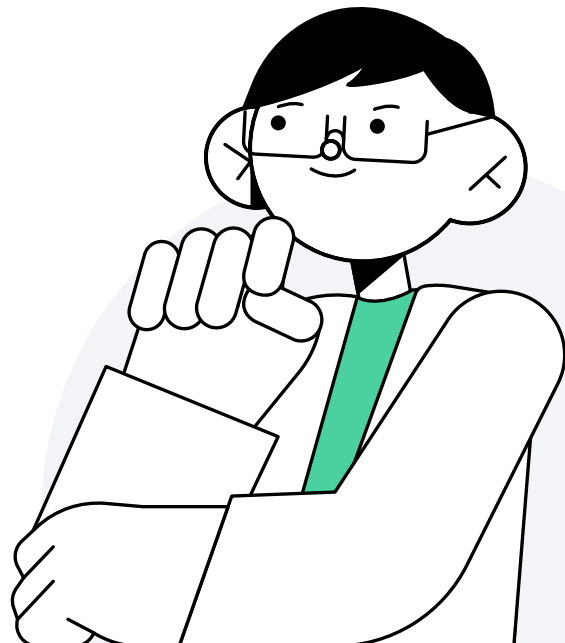
- ⊖ Время на приведение к нормальным формам
- ⊖ Накладные расходы при извлечении информации на объединение таблиц

Итоги



Итоги

- 1 Познакомились с понятием базы данных
- 2 Узнали о типах БД и СУБД
- 3 Спроектировали базу данных интернет-магазина

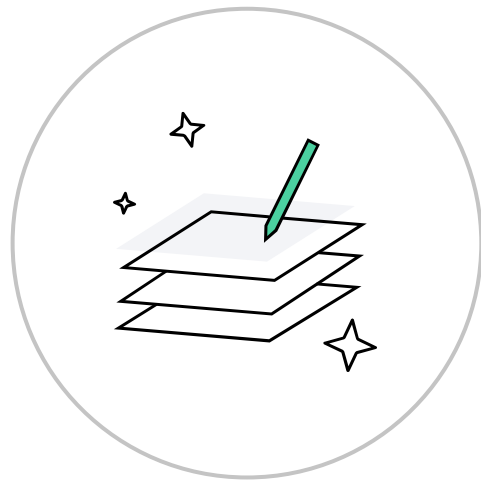


Домашнее задание

Давайте посмотрим ваше домашнее задание.

- 1 Вопросы о домашней работе задавайте в чате группы
- 2 Задачи можно сдавать по частям
- 3 Зачёт по домашней работе проставляется после того, как приняты **все задачи**

Выполните тест по пройденному материалу



Задавайте вопросы и пишите отзыв о лекции

Владислав Хорев
Ведущий программист в компании Andersen

