Базы данных
Лекция 1

Введение. Реляционные БД, таблицы и ключи. Нормальные формы

Андрей Каледин

План лекции

1. Правила и структура курса
2. Зачем нужны БД?
3. Различия моделей данных
4. OLTP vs OLAP
5. История развития БД
6. Основы реляционной модели
7. Ключи
8. Нормальные формы

Структура курса

2 блока (10 занятий)

5 занятий про реляционную модель
5 занятий про конкретные NoSQL технологии

Лекции

На лекциях разбираем теоретический материал

Семинары

На семинарах разбираем задачи и практикуемся

Домашние работы

Практические задания по пройденному материалу

Оценка =

10 домашних заданий

10% за каждое

+

Активность на семинаре

20% в конце курса

Больше 10 баллов получить нельзя ☹

Домашние работы

Каждую неделю

ДЗ выдается каждую неделю после семинара

Сдаем на GitHub

ДЗ сдаем в репозиторий через Merge Request из ветки с домашкой в master

Под каждое задание должна быть папка по шаблону homeworkX.

Дедлайн 2 недели

По прошествии 2-х недель с даты выдачи ДЗ не принимается

Коммуникация по курсу

https://t.me/+zFFPlIIp6lcwM2Ri

Кто ведет курс?

Андрей Каледин
Старший разработчик

Лектор и семинарист
1 блока

Мгер Аршакян
Ведущий разработчик

Лектор и семинарист
2 блока

Герман Тамбовцев
Разработчик

Семинарист 1 и 2 блока

Правила на лекциях

- Задавать вопросы в процессе лекции и после нее
- Проявлять уважение друг к другу
- Можно перебивать, если непонятно

Зачем нужны БД?

Информацию нужно:

- Хранить
- Структурировать
- Получать

Зачем нужны БД?

Информацию нужно:

- Хранить
- Структурировать
- Получать

Что хотим от системы хранения информации:

Зачем нужны БД?

Информацию нужно:

- Хранить
- Структурировать
- Получать

Что хотим от системы хранения информации:

- Удобно взаимодействовать

Зачем нужны БД?

Информацию нужно:

- Хранить
- Структурировать
- Получать

Что хотим от системы хранения информации:

- Удобно взаимодействовать
- Быстро и многопоточно записывать и читать

Зачем нужны БД?

Информацию нужно:

- Хранить
- Структурировать
- Получать

Что хотим от системы хранения информации:

- Удобно взаимодействовать
- Быстро и многопоточно записывать и читать
- Надежно хранить

Зачем нужны БД?

Информацию нужно:

- Хранить
- Структурировать
- Получать

Что хотим от системы хранения информации:

- Удобно взаимодействовать
- Быстро и многопоточно записывать и читать
- Надежно хранить
- Уметь работать с большими объемами

Что такое БД и СУБД?

БД - совокупность данных, хранящая некоторую информацию о реальном мире.

Что такое БД и СУБД?

БД - совокупность данных, хранящая некоторую информацию о реальном мире.

Система управления базами данных (СУБД) - это программное обеспечение, управляющее базой данных.

Что такое БД и СУБД?

БД - совокупность данных, хранящая некоторую информацию о реальном мире.

Система управления базами данных (СУБД) - это программное обеспечение, управляющее базой данных.

Примеры СУБД: Oracle Database, PostgreSQL, Clickhouse, Redis

Что такое модель данных?

Модель данных – набор правил хранения и обработки данных. Она описывает структуру БД и правила взаимодействия с ней.

Что такое модель данных?

Модель данных – набор правил хранения и обработки данных. Она описывает структуру БД и правила взаимодействия с ней.

Модель данных	**Сценарии использования**	**Примеры**
Реляционная	Составные запросы и транзакции	PostgreSQL

Что такое модель данных?

Модель данных – набор правил хранения и обработки данных. Она описывает структуру БД и правила взаимодействия с ней.

Модель данных	**Сценарии использования**	**Примеры**
Реляционная	Составные запросы и транзакции	PostgreSQL
Документная	Данные с большой вложенностью	MongoDB

Что такое модель данных?

Модель данных – набор правил хранения и обработки данных. Она описывает структуру БД и правила взаимодействия с ней.

Модель данных	**Сценарии использования**	**Примеры**
Реляционная	Составные запросы и транзакции	PostgreSQL
Документная	Данные с большой вложенностью	MongoDB
Колоночная	Аналитические запросы	Clickhouse

Что такое модель данных?

Модель данных – набор правил хранения и обработки данных. Она описывает структуру БД и правила взаимодействия с ней.

Модель данных	**Сценарии использования**	**Примеры**
Реляционная	Составные запросы и транзакции	PostgreSQL
Документная	Данные с большой вложенностью	MongoDB
Колоночная	Аналитические запросы	Clickhouse
Ключ-значение	Кэширование	Redis

Что такое модель данных?

Модель данных – набор правил хранения и обработки данных. Она описывает структуру БД и правила взаимодействия с ней.

Модель данных	**Сценарии использования**	**Примеры**
Реляционная	Составные запросы и транзакции	PostgreSQL
Документная	Данные с большой вложенностью	MongoDB
Колоночная	Аналитические запросы	Clickhouse
Ключ-значение	Кэширование	Redis
Графовая	Анализ связей между сущностями	Neo4j

OLAP vs OLTP

OLTP (Online Transaction Processing) БД нужны для большого количества коротких и частых операций изменения данных
Сценарий использования: Обработка пользовательских транзакций

OLAP vs OLTP

OLTP (Online Transaction Processing) БД нужны для большого количества коротких и частых операций изменения данных
Сценарий использования: Обработка пользовательских транзакций

OLAP (Online Analytical Processing) БД нужны для анализа больших объемов данных с низкой частотой обновления данных
Сценарий использования: Аналитика продаж за прошедший месяц

OLAP vs OLTP

OLTP (Online Transaction Processing) БД нужны для большого количества коротких и частых операций изменения данных
Сценарий использования: Обработка пользовательских транзакций

OLAP (Online Analytical Processing) БД нужны для анализа больших объемов данных с низкой частотой обновления данных
Сценарий использования: Аналитика продаж за прошедший месяц

Часто работают в связке:
1. OLTP собирает данные
2. Отдельный процесс переносит данные из OLTP-хранилища в OLAP-хранилище
3. OLAP анализирует данные и строит отчеты

Почему не хранить все в текстовом файле?

Почему не хранить все в текстовом файле?

Проблемы хранения данных в текстовых файлах:

1. **Производительность**
   Линейный поиск по файлу, перезапись всего файла при изменении данных

Почему не хранить все в текстовом файле?

Проблемы хранения данных в текстовых файлах:

1. **Производительность**
   Линейный поиск по файлу, перезапись всего файла при изменении данных
2. **Конфликты при многопоточной работе**
   В БД есть механизмы блокировок и транзакций

Почему не хранить все в текстовом файле?

Проблемы хранения данных в текстовых файлах:

1. **Производительность**
   Линейный поиск по файлу, перезапись всего файла при изменении данных
2. **Конфликты при многопоточной работе**
   В БД есть механизмы блокировок и транзакций
3. **Масштабируемость**
   БД загружает только нужные данные. Дополнительно, может быть развернута на нескольких серверах сразу

Почему не хранить все в текстовом файле?

Проблемы хранения данных в текстовых файлах:

1. **Производительность**
   Линейный поиск по файлу, перезапись всего файла при изменении данных
2. **Конфликты при многопоточной работе**
   В БД есть механизмы блокировок и транзакций
3. **Масштабируемость**
   БД загружает только нужные данные. Дополнительно, может быть развернута на нескольких серверах сразу
4. **Контроль за форматом данных**
   БД следит за схемой и типами данных

Почему не хранить все в текстовом файле?

Проблемы хранения данных в текстовых файлах:

1. **Производительность**
   Линейный поиск по файлу, перезапись всего файла при изменении данных
2. **Конфликты при многопоточной работе**
   В БД есть механизмы блокировок и транзакций
3. **Масштабируемость**
   БД загружает только нужные данные. Дополнительно, может быть развернута на нескольких серверах сразу
4. **Контроль за форматом данных**
   БД следит за схемой и типами данных
5. **Нужно писать сложный код для выполнения запросов**
   В БД есть операции пересечений, группировки и т.д. (JOIN, GROUP BY)

Когда нормально хранить данные в файле?

- **Настройки приложения** (часто в формате JSON или YAML)
- **Локальные данные** для небольших pet-проектов
- **Временное хранение данных** (кэширование, экспорт данных для ручной обработки)

Как развивались СУБД?

До 1960-х годов:
Хранение данных в файлах

Как развивались СУБД?

До 1960-х годов:
Хранение данных в файлах

1960-е – 1970-е:
Появление иерархической и сетевой модели

Как развивались СУБД?

До 1960-х годов:
Хранение данных в файлах

1960-е – 1970-е:
Появление иерархической и сетевой модели

1970-е:
Появление концепции реляционной СУБД и SQL

Как развивались СУБД?

До 1960-х годов:
Хранение данных в файлах

1960-е – 1970-е:
Появление иерархической и сетевой модели

1970-е:
Появление концепции реляционной СУБД и SQL

1970-е – 1990-е:
Распространение SQL БД, появление Oracle DB и PostgreSQL

Как развивались СУБД?

До 1960-х годов:
Хранение данных в файлах

1960-е – 1970-е:
Появление иерархической и сетевой модели

1970-е:
Появление концепции реляционной СУБД и SQL

1970-е – 1990-е:
Распространение SQL БД, появление Oracle DB и PostgreSQL

2000-е – настоящее время:
NoSQL и BigData. Появление колоночных, документных и других типов

Что сейчас хайп?

- **Мультимодельные БД** (реляционные + JSONB в одной БД)

Что сейчас хайп?

- **Мультимодельные БД** (реляционные + JSONB в одной БД)
- **Распределенные БД** (Google Spanner)

Что сейчас хайп?

- **Мультимодельные БД** (реляционные + JSONB в одной БД)
- **Распределенные БД** (Google Spanner)
- **Специальные БД для ML**

Что сейчас хайп?

- **Мультимодельные БД** (реляционные + JSONB в одной БД)
- **Распределенные БД** (Google Spanner)
- **Специальные БД для ML**
- **Упрощение аналитической инфраструктуры** (объединение ETL-процессов и аналитики в DataLake)

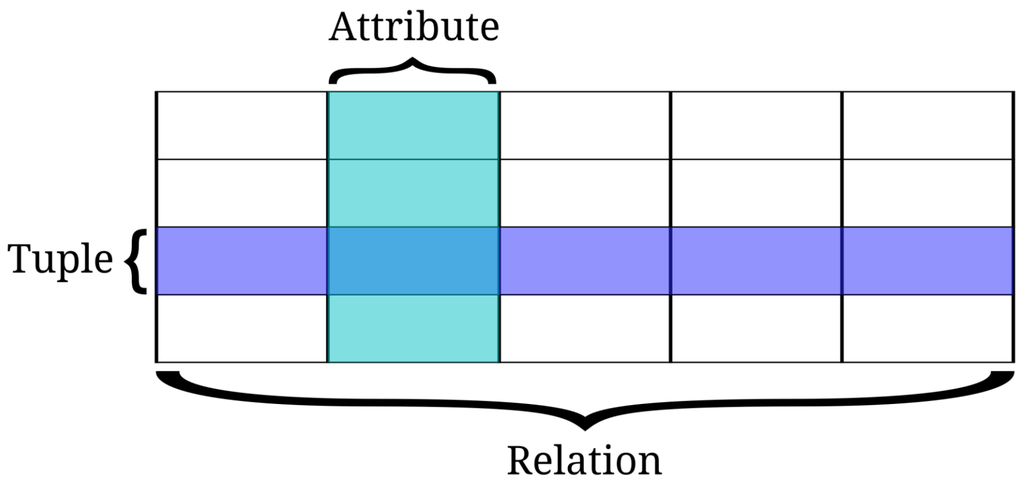
Реляционная модель

Relation – отношение

Принципы:

- Данные в БД – наборы отношений
- Существуют ограничения целостности
- Над данными можно производить действия (вставка, изменение, удаление и т.д)

Сущности реляционной модели



Данные организованы в таблицы

Отношение (Relation) – таблица
Кортеж (Tuple, Строка) – запись в таблице, представляющая конкретный объект
Столбец (Attribute, Поле) – отдельная характеристика объекта

Сущности реляционной модели

- **Первичный ключ (Primary Key, PK)** – уникальный идентификатор записи в таблице

Сущности реляционной модели

- **Первичный ключ (Primary Key, PK)** – уникальный идентификатор записи в таблице
- **Внешний ключ (Foreign Key, FK)** – ссылка на PK из другой таблицы (для установления связи между записями в разных таблицах)

Сущности реляционной модели

- **Первичный ключ (Primary Key, PK)** – уникальный идентификатор записи в таблице
- **Внешний ключ (Foreign Key, FK)** – ссылка на PK из другой таблицы (для установления связи между записями в разных таблицах)
- **Индекс (Index)** – дополнительная структура для ускорения поиска по таблице

Сущности реляционной модели

- **Первичный ключ (Primary Key, PK)** – уникальный идентификатор записи в таблице
- **Внешний ключ (Foreign Key, FK)** – ссылка на PK из другой таблицы (для установления связи между записями в разных таблицах)
- **Индекс (Index)** – дополнительная структура для ускорения поиска по таблице
- **Ограничения (Constraints)** – правила для обеспечения согласованности данных. Примеры: UNIQUE, NOT NULL

Нормальные формы

Нормальные формы – наборы правил для проектирования структуры реляционной БД, которые позволяют избежать избыточности данных и аномалий при манипуляциях с данными.

Нормальные формы

Нормальные формы – наборы правил для проектирования структуры реляционной БД, которые позволяют избежать избыточности данных и аномалий при манипуляциях с данными.

Виды нормальных форм:

1. Первая нормальная форма
2. Вторая нормальная форма
3. Третья нормальная форма
4. Нормальная форма Бойса-Кодда 
5. Четвертая нормальная форма

Первая нормальная форма (1NF)

Требования:

1. Столбцы атомарны
2. Нет повторяющихся строк

Неправильно:

Student	Courses
Bob	Math, Computer Science

Первая нормальная форма (1NF)

Требования:

1. Столбцы атомарны
2. Нет повторяющихся строк

Неправильно:

Student	Courses
Bob	Math, Computer Science

Правильно:

Student	Courses
Bob	Math
Bob	Computer Science

Вторая нормальная форма (2NF)

Требования:

1. Таблица в 1NF
2. Все атрибуты зависят от целого первичного ключа, а не от части

Вторая нормальная форма (2NF)

Неправильно:

Title	**Format**	Author	Price
Fifty Shades Of Grey	E-book	E. L. James	120
Fifty Shades Of Grey	Paper	E. L. James	150

PK = (Title, Format)

Вторая нормальная форма (2NF)

Неправильно:

Title	**Format**	Author	Price
Fifty Shades Of Grey	E-book	E. L. James	120
Fifty Shades Of Grey	Paper	E. L. James	150

PK = (Title, Format)

Правильно:

Title	**Format**	Price
Fifty Shades Of Grey	E-book	120
Fifty Shades Of Grey	Paper	150

PK = (Title, Format)

Title	Author
Fifty Shades Of Grey	E. L. James
Fifty Shades Of Grey	E. L. James

PK = (Title)

Третья нормальная форма (3NF)

Требования:

1. Таблица в 2NF
2. Ни один атрибут не зависит от других неключевых атрибутов

Неправильно:

ID	Name	Department	Department Location
413	Bob Smith	Engineering	82 Victoria Street London

Третья нормальная форма (3NF)

Требования:

1. Таблица в 2NF
2. Ни один атрибут не зависит от других неключевых атрибутов

Неправильно:

ID	Name	Department	Department Location
413	Bob Smith	Engineering	82 Victoria Street London

Правильно:

ID	Name	Department
413	Bob Smith	Engineering

Department	Department Location
Engineering	82 Victoria Street London

Нормальная форма Бойса-Кодда (BCNF / 3.5NF)

Требования:

1. Таблица в 3NF
2. Не существует зависимости X -> Y, где X – не является суперключом

Неправильно:

Course	Lecturer	Student
Math	B. Smith	Mike
Math	B. Smith	Alice
Physics	Dr. Cube	Mike

Условие: каждый курс читает только один преподаватель

Проблема: для смены лектора придется проверять всю таблицу

Нормальная форма Бойса-Кодда (BCNF / 3.5NF)

Требования:

1. Таблица в 3NF
2. Не существует зависимости X -> Y, где X – не является суперключом

Неправильно:

Course	Lecturer	Student
Math	B. Smith	Mike
Math	B. Smith	Alice
Physics	Dr. Cube	Mike

Условие: каждый курс читает только один преподаватель

Проблема: для смены лектора придется проверять всю таблицу

Правильно:

Course	Lecturer
Physics	Dr. Cube
Math	B. Smith

Student	Course
Mike	Math
Alice	Math
Mike	Physics

Четвертая нормальная форма (4NF)

Требования:

1. Таблица в BCNF
2. Отсутствуют многозначные зависимости

Неправильно:

Course	Lecturer	Student
Math	B. Smith	Mike
Math	B. Smith	Alice
Physics	Dr. Cube	Mike

Проблема: если курс может вести несколько преподавателей, то придется дублировать записи с этим курсом

Четвертая нормальная форма (4NF)

Требования:

1. Таблица в BCNF
2. Отсутствуют многозначные зависимости

Неправильно:

Course	Lecturer	Student
Math	B. Smith	Mike
Math	B. Smith	Alice
Physics	Dr. Cube	Mike

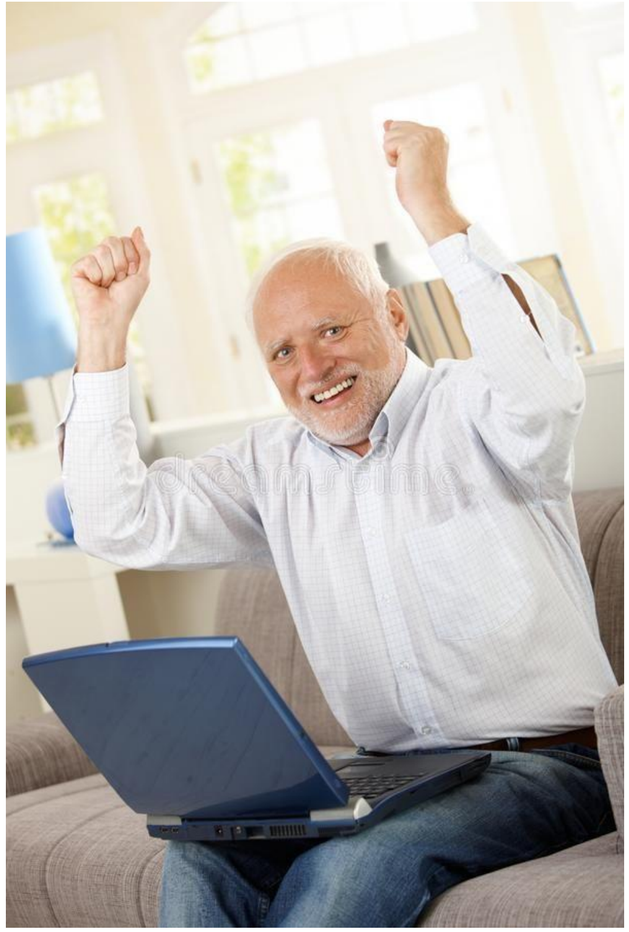
Проблема: если курс может вести несколько преподавателей, то придется дублировать записи с этим курсом

Правильно:

Course	Lecturer
Physics	Dr. Cube
Math	B. Smith
Math	Dr. Dre

Student	Course
Mike	Math
Alice	Math
Mike	Physics

Это было непросто, но мы справились!



Что мы сегодня узнали?

1. Поняли, из чего будет состоять курс
2. Разобрались, зачем нам нужны БД и почему данные в файле – не всегда лучший выбор
3. Узнали про виды моделей данных и разницу между OLTP и OLAP
4. Посмотрели за историческим развитием СУБД
5. Погрузились в основы реляционной модели
6. Узнали, что существуют разные нормальные формы и примерно поняли, чем они отличаются

Что будет на следующей лекции?

1. Начнем говорить про SQL
2. Разберем основные функции и синтаксис
3. DDL vs DML vs DTL
4. Рассмотрим агрегации, оконные функции и join’ы

Спасибо за внимание!