

Работа с базами данных. Часть I

- Отношением R называется подмножество декартового произведения
- Каждая строка в таблице может быть помечена уникальным идентификатором или **первичным ключом**
- Строки из разных таблиц могут быть связаны внешним ключом
- Для организации доступа к данным обычно используется язык запросов SQL
- Обеспечение целостности (ограничения)
 - первичные ключи (primary keys)
 - внешние ключи (foreign keys)
 - ограничения типа UNIQUE, NOT NULL
- Транзакционность:
 - атомарность (все или ничего)
 - согласованность (соответствие предопределенным правилам)
 - изолированность (изолированное выполнение транзакций)
 - долговечность (сохранения изменений, даже в случае падения БД)
- Приведение отношений к более строгой нормальной форме называется **нормализацией**, обратно к более слабой – **денормализацией**.
- Теорема Кодда:
 - relational algebra and query languages are precisely equivalent in expressive power (Реляционная алгебра и SQL – это одно и то же)
- SQL:
 - CREATE TABLE
 - DROP TABLE
 - JOIN
 - INSERT
 - UPDATE
 - DELETE

Работа с базами данных. Часть II

- JDBC – API, определяющий интерфейсы доступа к реляционным бд
- Для работы с конкретной СУБД требуется включить JDBC-драйвер этой СУБД
- → JDBC API (Java) + JDBC драйвер (сторонняя библиотека)
- Задача JDBC API – предоставить набор интерфейсов и классов для выполнения SQL запросов к БД
- Задача JDBC драйвера – преобразовать методы приложения Java в протокол, специфичный для конкретной СУБД
- JDBC-API: Соединения:
 1. Создание соединения
 2. Загрузка класса драйвера в память JVM
 3. Закрытие соединения → try-with-resources
- Запросы → execute
- Параметризованные запросы → в запрос можно передать параметры
- Курсоры → навигация по выборке
- Режим доступа к БД:
 - только чтение
 - чтение и запись
- Тип курсора и режим доступа задаются при создании запроса
- Чтобы представлять отношения в виде классов а не наборов → ORM
- Для этого нужно реализовать:
 - POJO – Plain Old Java Object
 - DAO – Data Access Object
 - create()
 - read()
 - update()
 - delete

- Чтобы не писать похожие DAO для каждого класса, можно использовать ORM библиотеки