



# Урок 14.

## Работа с базой данных в Java

# Agenda

- Реляционные базы данных
- Язык запросов SQL
- Работа с базой данных через JDBC
- Введение в JPA
- Работа с базой данных с использованием Hibernate и Spring
- Транзакции в Java

# Реляционные базы данных

**Реляционная база** набор данных в виде таблиц состоящих из строк и столбцов

- Строки должны быть уникальны. Обычно используют главный (primary) ключ
- Колонки не должны повторяться
- Связь между таблицами осуществляется с помощью внешних (foreign) ключей

см 12 правил Кодда

# Язык запросов SQL

**SQL** (structured query language) - язык программирования, используемый для создания, модификации, управления и выборки данных из реляционной базы

insert deny select  
drop create  
truncate revoke  
update delete  
alter grant

# Реляционные базы данных

Наиболее популярные базы данных:

- MySQL (MariaDB)
- PostgreSQL
- SQLite
- Oracle
- DB2
- H2
- Derby
- etc.

# Работа с базой данных через JDBC

**JDBC** (Java database connectivity) - стандарт взаимодействия с базой данных входящий в состав J2SE

- находится в java.sql
- не зависит от конкретной базы данных (если использовать только стандарт SQL)
- можно подключаться к любой базе по URL

# Работа с базой данных через JDBC

**JDBC** (Java database connectivity) - стандарт взаимодействия с базой данных входящий в состав J2SE

- Connection - подключение к базе
- Statement - обычное sql выражение
- PreparedStatement - sql выражение с параметрами
- CallableStatement - вызов хранимой процедуры
  
- ResultSet - результат sql выражения

# Введение в JPA 2.0

**JPA (Java Persistence API)** - API, предоставляющий возможность сохранять Java-объекты в базу данных.

Введен с версии 5, текущая версия 2.1

- расположен в `javax.persistence`
- базируется на понятии `@Entity`
- позволяет использовать JPQL (Java persistence query lang.)
- позволяет абстрагироваться от запросов с помощью Criteria API



# Введение в JPA 2.0

**JPA (Java Persistence API)** - API, предоставляющий возможность сохранять Java-объекты в базу данных. ORM в мире Java.

Введен с версии 5, текущая версия 2.1

- расположен в `javax.persistence`
- базируется на понятии `@Entity`
- позволяет использовать JPQL (Java persistence query lang.)
- позволяет абстрагироваться от запросов с помощью Criteria API

# Введение в JPA 2.0

**Entity** ключевое понятие в JPA.

Должно удовлетворять следующим условиям

- Содержать хотя бы одно поле @Id
- Иметь конструктор без аргументов
- Не иметь final полей
- Не может быть final
- Не может быть вложенным

# Введение в JPA 2.0

**JPA** это прежде всего спецификация, интерфейсы и абстракции

Основные реализации (вендоры)

- OpenJPA
- Hibernate
- EclipseLink
- Batoo

# Введение в JPA 2.0

**JPA** это прежде всего спецификация, интерфейсы и абстракции

Основные реализации (вендоры)

- OpenJPA
- **Hibernate**
- EclipseLink
- Batoo

# Введение в JPA 2.0

## Порядок работы с JPA в Spring

- настраивается источник данных (data source)
- настраивается менеджер сущностей (entity manager)
- настраивается менеджер транзакций (transaction manager)

# Транзакции в Java

**Транзакции базы данных** единицы работы с СУБД, которые обрабатываются как единое целое.

- Атомарность
- Согласованность
- Изолированность
- Надежность

# Home Work

1. Добавить возможность регистрации пользователей в сетевой чат
  - Пользователь должен ввести логин, пароль, имя и фамилию, возраст при первой регистрации
  - Пользователь должен ввести логин/пароль, чтобы войти в чат
2. Добавить возможность удаления пользователя. Для этого нужно ввести обязательно его логин и пароль.
3. Добавить возможность получения данных любого пользователя путем ввода его логина