

Databases, SQL, JDBC, Hibernate.

Anastasiya Solodkaya, Denis Stepulenok

LevelUP

10 марта 2017 г.

- 1 Базы данных
- 2 Реляционные базы данных
- 3 Проектирование базы данных
- 4 JDBC
- 5 JPA, Hibernate

Что такое база данных?

- Совокупность элементов
- Файл тоже можно считать базой данных

СУБД

СУБД

Система управления базами данных. Обычно именно это подразумевают под словами "База данных". Однако это - средства, язык, инструменты, но не сами данные.

Какие базы данных бывают?

- Реляционные
- Иерархические
- Объектно-ориентированные
- И прочие...

Что такое NoSQL?

Примеры СУБД

Oracle, MS SQL, My SQL, PostgreSQL, SQLite, Sybase, FoxPro, MS Access, Apache Cassandra, MongoDB, HBase...

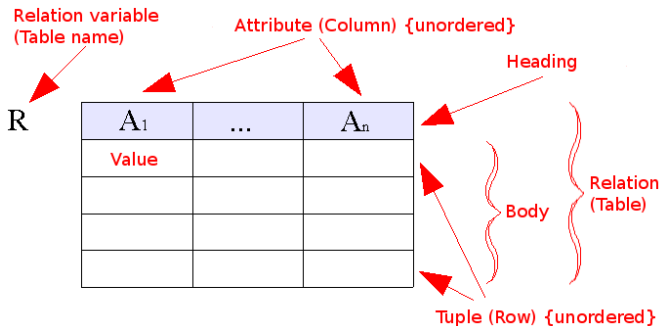
- 1 Базы данных
- 2 **Реляционные базы данных**
- 3 Проектирование базы данных
- 4 JDBC
- 5 JPA, Hibernate

Реляционные базы данных и РСУБД (RDBMS)

- На основе реляционной модели данных. Алгебра отношений.
- Содержит таблицы ("отношения") и строки ("кортежи")
- Oracle DB, MS SQL, MySQL, PostgreSQL, SQLite, MS Access, ...

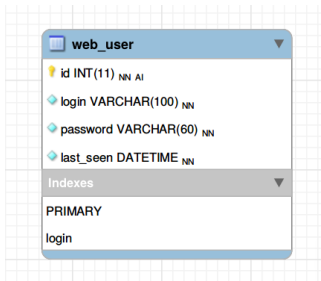
Реляционная модель

- Таблица содержит сущности одного типа (представлены строками)
- Таблица имеет атрибуты (названия столбцов)
- Строка содержит значение атрибута (значение столбца)



Пример таблицы

id	login	password	last_seen
1	nastya	sdfgdfg	2017-03-10 17:18:20
2	ekaterina	rgf	2017-03-10 17:18:30
3	pavel	213sdfd	2017-03-10 17:18:36



```
create table web_user (
  id int auto_increment primary key,
  login varchar(100) not null unique,
  password varchar(60) not null,
  last_seen datetime not null default now()
)
```

SQL

- Structured Query Language
- **DDL** - Data Definition Language
- **DML** - Data Manipulation Language
- **DCL** - Data Control Language
- **TCL** - Transaction Control Language

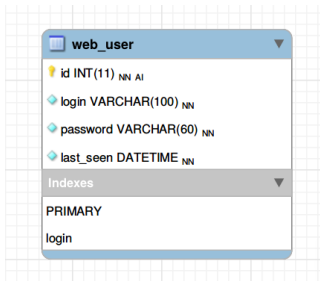
DDL

Позволяет описать схему данных

- create database...
- create table...
- drop table...
- alter table...
- rename table...
- и другие

Пример таблицы

id	login	password	last_seen
1	nastya	sdfgdfg	2017-03-10 17:18:20
2	ekaterina	rgf	2017-03-10 17:18:30
3	pavel	213sdfd	2017-03-10 17:18:36



```
create table web_user (
  id int auto_increment primary key,
  login varchar(100) not null unique,
  password varchar(60) not null,
  last_seen datetime not null default now()
)
```

DML

Позволяет манипулировать данными. CRUD:

- **create** - insert into table...
- **read** - select from table...
- **update** - update table...
- **delete** - delete from table...

- 1 Базы данных
- 2 Реляционные базы данных
- 3 Проектирование базы данных
- 4 JDBC
- 5 JPA, Hibernate

Как спроектировать базу данных?

- Определить область знаний
- Определить основные сущности и взаимоотношения между ними
- Возможно нарисовать диаграмму (ER-диаграмму)
- Проверить свою базу данных на целостность (нормализация)
- Создать скрипт для создания базы
- Возможно создать скрипт для предварительного заполнения базы

1 + 2 - Определить область знаний и основные сущности

Разрабатываем базу для хранения информации об обучении студентов в небюджетном учреждении.

- Можно выделить преподавателя, студента, преподаваемый курс, текущий трек курса
- Каждый преподаватель может вести несколько курсов, как и курс может иметь несколько преподавателей. ФИО преподавателей не могут быть одинаковыми.
- Каждый студент может посещать несколько треков. У каждого студента есть свой счет, на который зачисляется оплата обучения. К этому счету так же привязаны паспортные данные.
- Трек курса связан с одним преподавателем и его может посещать несколько студентов, которые по этому курсу будут иметь оценки.

Ограничения в базе данных

- Ограничения на связи ("каждый преподаватель может вести несколько курсов")
- Ограничения на уникальность ("ФИО преподавателей не могут быть одинаковыми", "паспортные данные студента")
- Ограничения на значения (оценки студента)

Ключи и констрейнты

- **Primary key** (например, уникальный идентификатор преподавателя, сгенерированный базой данных). Он всегда NOT NULL + UNIQUE. Может быть составным.
- **Foreign key** (например, преподаватель текущего трека). Ссылка на **primary key** другой таблицы. Позволяет управлять политикой удаления и обновления записей. Что делать с треками, если преподавателя удалили из базы данных?
- **Not Null, Unique, Set, Enum**, и т.д.

Нормализация БД

- 7 "нормальных" форм свойств
- **1NF** - атрибут не дублируется в строке
- **2NF** - + каждый неключевой атрибут неприводимо зависит от ключа
- И так далее...

Разрабатываем базу

Вместе разрабатываем базу

Манипуляция данными

```
insert into users(login, password) values ('Petya',  
      'dsfgdft54');  
  
delete from users where login='Petya';  
  
update users set password='123' where login='Petya';  
  
select * from users;  
  
select password, last_seen from users where login='Petya';
```

- 1 Базы данных
- 2 Реляционные базы данных
- 3 Проектирование базы данных
- 4 JDBC
- 5 JPA, Hibernate

JDBC

- Java DataBase Connectivity
- Стандарт для работы с базами данных
- Для практической работы используются "драйвера"
- `java.sql`

Основные классы

- `java.sql.Connection`
- `java.sql.Statement` - для запросов
- `java.sql.PreparedStatement` - для запросов с параметрами
- `java.sql.CallableStatement` - для вызова хранимых процедур
- `java.sql.ResultSet` - по сути - результат запроса

JDBC

```
import java.sql.*;

public class JDBCdemo {
    public static void main(String[] args) throws
        ClassNotFoundException, SQLException {
        Driver driver = new org.postgresql.Driver();
        String userName = "postgres";
        String password = "123";
        Connection con = DriverManager.getConnection(
            "jdbc:postgresql://localhost:5432/postgres",
            userName, password);
        Statement query = con.createStatement();
        ResultSet resultSet = query.executeQuery("SELECT *
            FROM \"user\" ORDER BY NAME");
        while (resultSet.next()) {
            System.out.println(resultSet.getString("name"));
        }
        resultSet.close();
    }
}
```

JDBC

Метаданные ResultSet

```
resultSet = query.executeQuery("SELECT * FROM \"user\"  
    ORDER BY NAME");  
ResultSetMetaData rsmd = resultSet.getMetaData();  
while (resultSet.next()) {  
    for (int i = 1; i <= rsmd.getColumnCount(); i++) {  
        System.out.println(rsmd.getColumnName(i) + " = " +  
            resultSet.getString(i));  
    }  
}  
resultSet.close();
```

- 1 Базы данных
- 2 Реляционные базы данных
- 3 Проектирование базы данных
- 4 JDBC
- 5 JPA, Hibernate

Hibernate

- **ORM** - Object-Relational Mapping
- **JPA** - Java Persistence API. JQL
- **Hibernate** - Библиотека для решения задач ORM. HQL, JPA.

Hibernate

- Аннотации - @Table, @Entity, @Id, @Column, @NamedQuery, @OneToOne ...
- **SessionFactory, Session, Query**
- Можно сконфигурировать с помощью xml-файла или property-файла (класть в **resources**)

Загрузка конфигурации

```
Configuration cfg = new Configuration();  
cfg.configure("hibernate.cfg.xml");  
SessionFactory factory = cfg.buildSessionFactory();  
Session session = factory.openSession();
```

Простая конфигурация

```
<hibernate-configuration>
  <session-factory>
    <property name="hibernate.connection.driver_class">
      org.postgresql.Driver
    </property>
    <property name="hibernate.connection.url">
      jdbc:postgresql://localhost:5432/levelp
    </property>
    <property name="hibernate.connection.username">
      levelp
    </property>
    <property name="hibernate.connection.password">
      levelp
    </property>
```


Простая конфигурация

```
<property name="hibernate.dialect">
    org.hibernate.dialect.PostgreSQLDialect
</property>
<property name="show_sql">
    true
</property>
<property name="hibernate.hbm2ddl.auto">
    create-drop
</property>
<mapping class="com.levelp.model.User"/>
<mapping class="com.levelp.model.UserData"/>
</session-factory>
</hibernate-configuration>
```