Министерство образования и науки Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение**

**высшего профессионального образования**

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

**(ВлГУ)**

**Кафедра информационных систем и программной инженерии**

**Лабораторная работа №1**

**по дисциплине:**

**«Распределенные программные системы»**

**Тема работы: «Работа с реляционной СУБД средствами JDBC API»**

**Выполнил**:

Студент гр. ИСТ-119

Солдатов А.Г. Графков А.А.

Студент гр. ПРИ-119  
Лебедева В.М.

**Приняла**:

Проскурина Г.В.

Владимир 2021 г.

**Цель работы:**

Изучить средства JDBC API для работы с реляционными СУБД и получить навыки их практического использования

**Выполнение:**

Создадим базу данных, соответствующую предметной области курсового проекта и свяжем её с приложением.

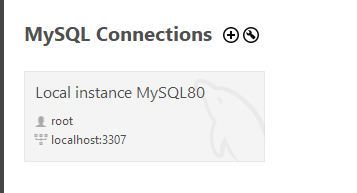


Рисунок 1. MySQL сервер

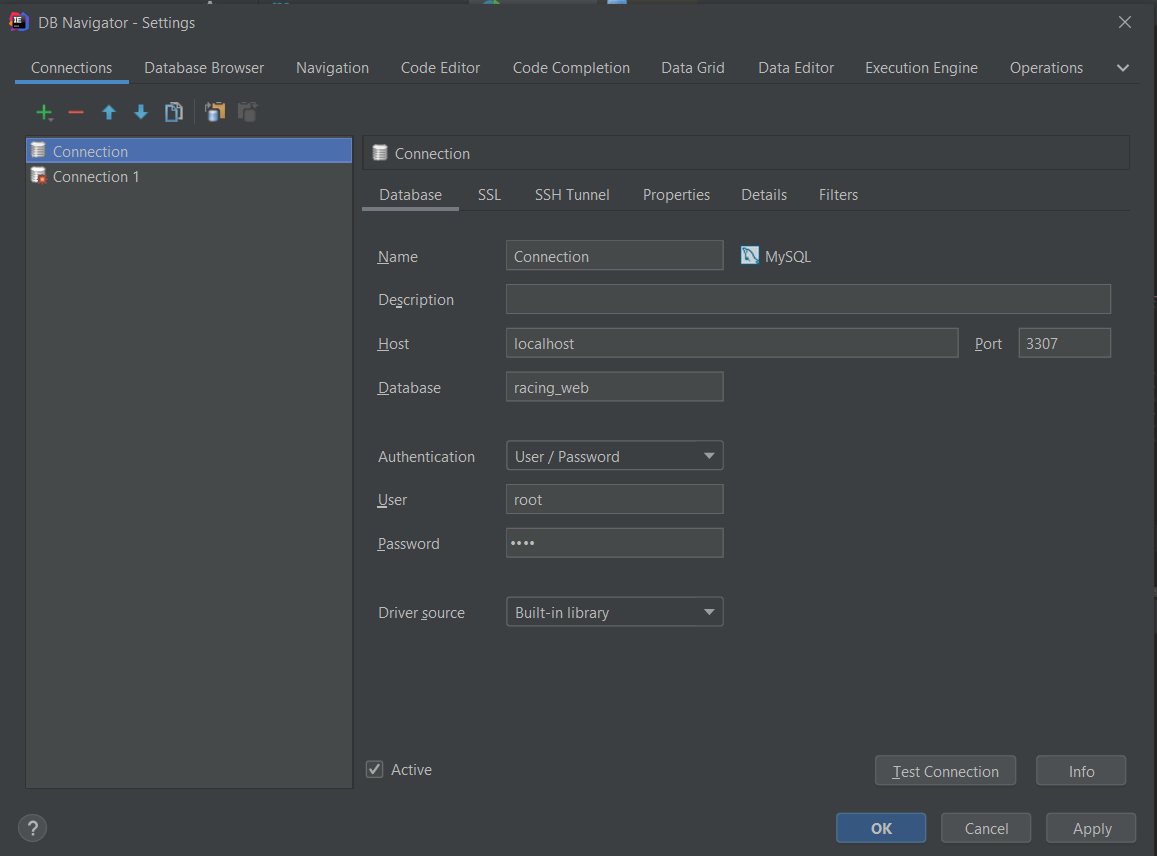


Рисунок 2. Установление связи

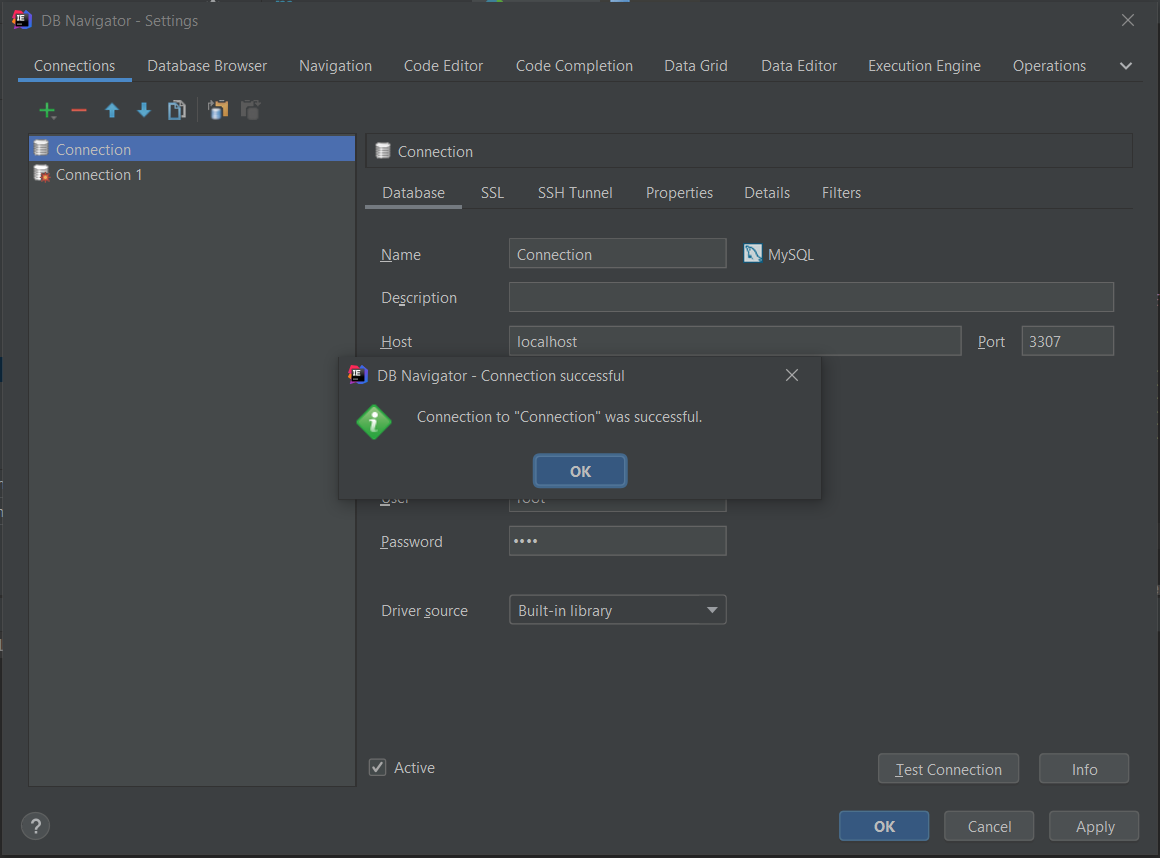


Рисунок 3. Успешное соединение

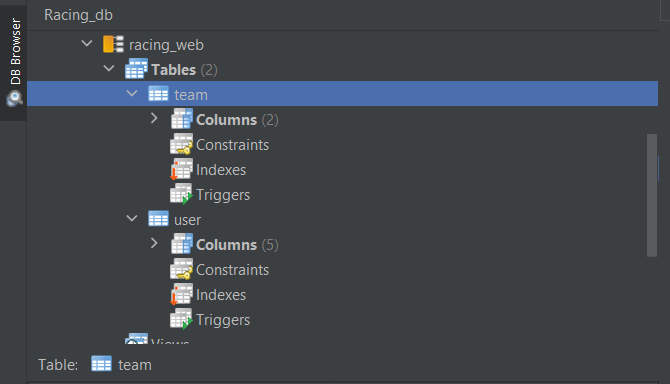


Рисунок 4. Доступ к базе через IDE

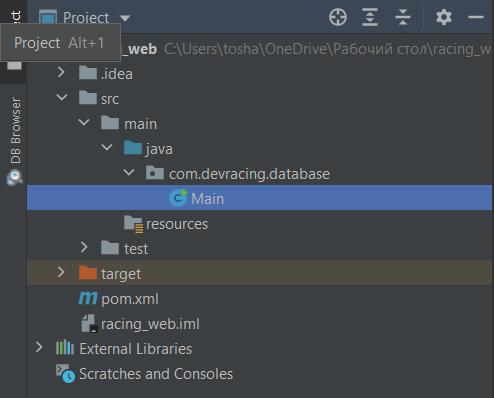


Рисунок 5. Структура maven-проекта

**Pom.xml**

<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>org.example</groupId>  
 <artifactId>racing\_web</artifactId>  
 <version>1.0-SNAPSHOT</version>  
  
 <properties>  
 <maven.compiler.source>8</maven.compiler.source>  
 <maven.compiler.target>8</maven.compiler.target>  
 </properties>  
  
 <!--MySQL java connector-->  
 <dependencies>  
 <dependency>  
 <groupId>mysql</groupId>  
 <artifactId>mysql-connector-java</artifactId>  
 <version>8.0.27</version>  
 </dependency>  
 </dependencies>  
  
  
</project>

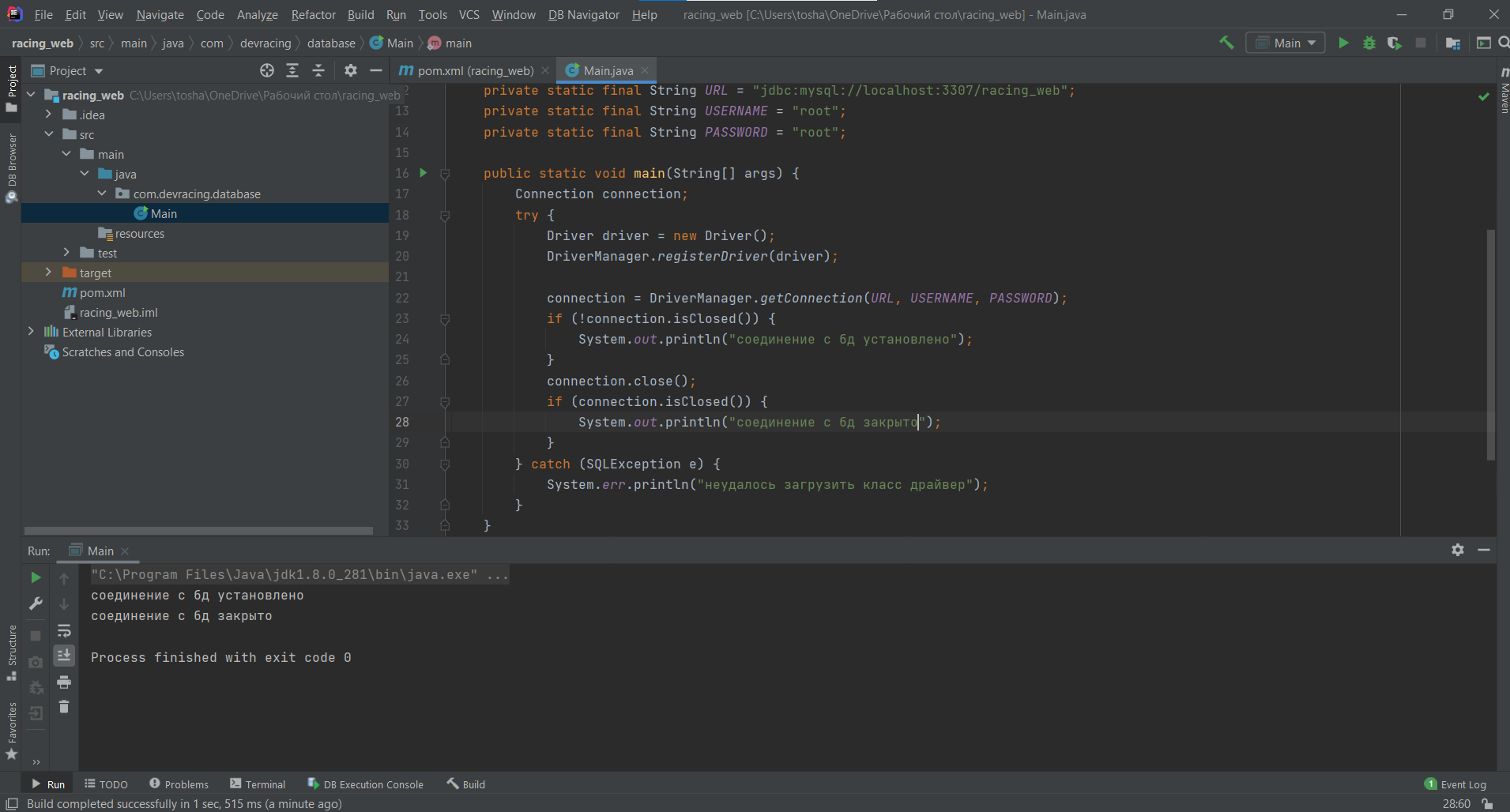


Рисунок 6. Проверка соединения

Код программы:

package com.devracing.database;  
import com.mysql.cj.jdbc.Driver;  
import com.mysql.cj.xdevapi.Result;  
import java.sql.\*;  
  
  
public class Main {  
  
 private static final String *URL* = "jdbc:mysql://localhost:3307/racing\_web";  
 private static final String *USERNAME* = "root";  
 private static final String *PASSWORD* = "root";  
  
 //? - это переменные в SQL-запросе  
 private static final String *INSERT\_FIRST*="insert into team (ID, Title, Motorsport) values (?, ?, ?)";  
 private static final String *INSERT\_SECOND*="insert into team (ID, Title, Motorsport) values (?, ?, ?)";  
 private static final String *INSERT\_THIRD*="insert into team (ID, Title, Motorsport) values (?, ?, ?)";  
 private static final String *GET\_ALL*="select \* from team";  
 private static final String *UPDATE\_TEAM*="update team set Title = ? where id = ?";  
 private static final String *DELETE*="delete from team where id = ?";  
  
 public static void main(String[] args) {  
 Connection connection = null;  
 PreparedStatement preparedStatement = null;  
  
 try {  
 Driver driver = new Driver();  
 DriverManager.*registerDriver*(driver);  
 connection = DriverManager.*getConnection*(*URL*, *USERNAME*, *PASSWORD*);  
 preparedStatement = connection.prepareStatement(*INSERT\_FIRST*);  
 preparedStatement.setInt(1, 1);  
 preparedStatement.setString(2, "Red Bull");  
 preparedStatement.setString(3, "Formula-1");  
 preparedStatement.execute();  
  
 preparedStatement = connection.prepareStatement(*INSERT\_SECOND*);  
 preparedStatement.setInt(1, 2);  
 preparedStatement.setString(2, "Scuderia Vedrarri");  
 preparedStatement.setString(3, "Formula-1");  
 preparedStatement.execute();  
  
 preparedStatement = connection.prepareStatement(*INSERT\_THIRD*);  
 preparedStatement.setInt(1, 3);  
 preparedStatement.setString(2, "Mercedes AMG Petronas");  
 preparedStatement.setString(3, "Formula-1");  
 preparedStatement.execute();  
  
 preparedStatement = connection.prepareStatement(*UPDATE\_TEAM*);  
 preparedStatement.setString(1, "Scuderia Ferrari");  
 preparedStatement.setInt(2, 2);  
 preparedStatement.executeUpdate();  
  
 preparedStatement = connection.prepareStatement(*DELETE*);  
 preparedStatement.setInt(1, 3);  
 preparedStatement.executeUpdate();  
  
 connection.close();  
 if(!connection.isClosed()) {  
 System.*out*.println("не закрыт");  
 }  
  
 } catch (SQLException e) {  
 //System.err.println("неудалось загрузить класс драйвер");  
 e.printStackTrace();  
 }  
 }  
}

Вывод в консоль осуществляется с помощью цикла:

preparedStatement = connection.prepareStatement(*GET\_ALL*);  
  
ResultSet res = preparedStatement.executeQuery();  
  
while (res.next()) {  
 int id = res.getInt("ID");  
 String title = res.getString("Title");  
 String motorsport = res.getString("Motorsport");  
 System.*out*.println("id: "+id+", Title: "+title+", Motorsport: "+motorsport);  
}

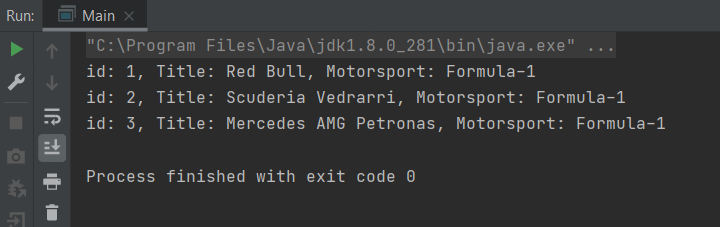


Рисунок 7. Результат создания и заполнения таблиц

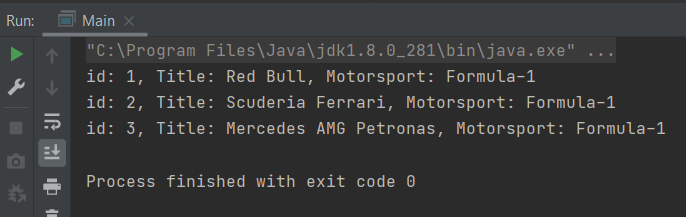


Рисунок 8. Результат обновления таблиц

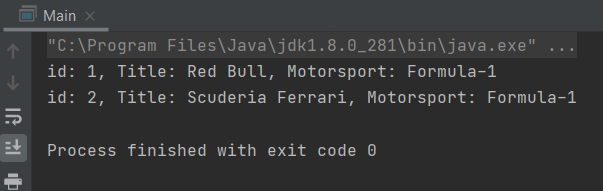


Рисунок 9. Результат удаления

**Вывод:**

В результате проделанной работы были изучены средства JDBC API для работы с реляционными СУБД и получены навыки их практического использования