*Київський національний університет імені Т. Шевченка*

*Факультет комп’ютерних наук та кібернетики*

Паралельні обчислення

Лабораторна робота №6

Звіт

Виконав: студент групи ІПС-31

Кравчук Павло

*Київ-2020*

***Умова завдання:***

Розробити веб-застосунок, що надає користувачу дані про об’єкти заданої предметної області. Інформація про об’єкти зчитується з бази даних, створеної при виконанні лабораторної роботи №2.

***Варіант 8***

***Предметна область: Футбол***

***Об'єкти: Команди, Гравці***

Файл Team.java:

import java.util.ArrayList;  
import java.util.List;  
  
public class Team { //don't use constructor since we're filling elements one by one  
 public int code;  
 public String name;  
 public List<Player> players = new ArrayList<>();  
  
 public String toString() {  
 return name;  
 }  
}

Містить поля ID, назви команди та список гравців зо належать до команди. Також наявний метод приведення до строки що повертає назву.

Файл Player.java:

public class Player {  
 public int code;  
 public String name;  
 public boolean isCaptain;  
 public int salary;  
  
 public String toString() {  
 return name;  
 }  
}

Містить поля ID, імені, статусу капітана та зарплатні гравця Також наявний метод приведення до строки що повертає ім’я.

Файл Football.java:

import java.io.PrintWriter;  
import java.sql.Connection;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.sql.Statement;  
import java.util.ArrayList;  
import java.util.List;  
  
public class Football {  
 private Connection con;  
 private Statement stmt;  
  
 Football(Connection con) throws SQLException {  
 this.con = con;  
 this.stmt = con.createStatement();  
 }  
  
 public void stop() throws SQLException {  
 con.close();  
 }  
  
 public List<Team> getTeams(PrintWriter out) {  
 List<Team> teams = new ArrayList<>();  
 String sql = "SELECT ID\_T, NAME FROM TEAMS";  
 try  
 {  
 ResultSet rs = stmt.executeQuery(sql);  
 while (rs.next())  
 {  
 int id = rs.getInt("ID\_T");  
 String name = rs.getString("NAME");  
 Team team = new Team();  
 team.code = id;  
 team.name = name;  
 teams.add(team);  
 }  
 rs.close();  
  
 for (Team team : teams) {  
 team.players = getTeamPlayers(team.code);  
 }  
 } catch (SQLException e) {  
 out.println(e.getMessage());  
 }  
 return teams;  
 }  
  
 public List<Player> getTeamPlayers(int teamCode) {  
 List<Player> players = new ArrayList<>();  
 String sql = "SELECT \* FROM PLAYERS WHERE ID\_T = "+teamCode;  
 try  
 {  
 ResultSet rs = stmt.executeQuery(sql);  
 while (rs.next())  
 {  
 int id = rs.getInt("ID\_P");  
 String name = rs.getString("NAME");  
 boolean isCaptain = rs.getInt("ISCAPTAIN") == 1;  
 int salary = rs.getInt("SALARY");  
 Player player = new Player();  
 player.code = id;  
 player.name = name;  
 player.isCaptain = isCaptain;  
 player.salary = salary;  
 players.add(player);  
 }  
 rs.close();  
 } catch (SQLException ignore) { }  
 return players;  
 }  
}

Клас, що здійснює запити до СУБД. В методі getTeams отримується список команд, після чого для кожної команди викликається getTeamPlayers який отримує список гравців, що належать до команди з заданим ID.

Файл MyServlet.java:

import com.mysql.cj.jdbc.Driver;  
  
import java.io.IOException;  
import java.io.PrintWriter;  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.SQLException;  
import java.util.List;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
public class MyServlet extends HttpServlet {  
  
 public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {  
 List<Team> teams = null;  
 PrintWriter out = response.getWriter();  
 try {  
 Driver driver = new com.mysql.cj.jdbc.Driver();  
 DriverManager.*registerDriver*(driver);  
 } catch (SQLException e) {  
 out.println(e.getMessage());  
 }  
 String url = "jdbc:mysql://localhost:3306/football?serverTimezone=UTC&useSSL=false";  
 try {  
 Connection con = DriverManager.*getConnection*(url, "root", "5502");  
 Football football = new Football(con);  
 teams = football.getTeams(out);  
 football.stop();  
 } catch (SQLException e) {  
 out.println(e.getMessage());  
 }  
  
 response.setContentType("text/html;charset=UTF-8");  
 try {  
 out.println("<html>");  
 out.println("<head>");  
 out.println("<title>Teams</title>");  
 out.println("</head>");  
 out.println("<body>");  
 out.println("<h1>Teams</h1>");  
  
 for (Team team : teams) {  
 out.println(team.code + " name: " + team.name + " players: <br>");  
 for (Player player : team.players) {  
 out.println("&nbsp;&nbsp;&nbsp;&nbsp;"+ player.code + " - " + player.name + " - " +  
 (player.isCaptain?"1":"0") + " - " + player.salary + "<br>");  
 }  
 }  
  
 out.println("</body>");  
 out.println("</html>");  
 } finally {  
 out.close();  
 }  
 }  
 public String getServletInfo() {  
 return "Teams";  
 }  
}

Основний клас-сервлет, що здійснює підключення до СУБД, з допоміжного класу Football отримує список команд після чого в форматі HTML виводить команди та їх гравців на сторінку.

Файл web.xml:

<?xml version="1.0" encoding="UTF-8"?>  
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"  
 version="4.0">  
 <servlet>  
 <servlet-name>MyServlet</servlet-name>  
 <servlet-class>MyServlet</servlet-class>  
 </servlet>  
 <servlet-mapping>  
 <servlet-name>MyServlet</servlet-name>  
 <url-pattern>/</url-pattern>  
 </servlet-mapping>  
 <session-config>  
 <session-timeout>  
 30  
 </session-timeout>  
 </session-config>  
  
</web-app>