Relational Databases with MySQL Week 10 Assignment

Points possible: 70

Category	Criteria	% of Grade
Functionality	Does the code work?	25
Organization	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
Creativity	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
Completeness	All requirements of the assignment are complete.	25

Instructions: In Eclipse, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document to the repository. Additionally, push an .sql file with all your queries and your Java project code to the same repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

Coding Steps:

In this week's coding activity, you will create a menu driven application backed by a MySQL database.

To start, choose one item that you like. It could be vehicles, sports, foods, etc....

Create a new Java project in Eclipse.

Create a SQL script in the project to create a database with one table. The table should be the item you picked.

Write a Java menu driven application that allows you to perform all four CRUD operations on your table.

Tips:

The application does not need to be as complex as the example in the video curriculum.

You need an option for each of the CRUD operations (Create, Read, Update, and Delete).

Remember that PreparedStatment.executeQuery() is only for Reading data and .executeUpdate() is used for Creating, Updating, and Deleting data.

Remember that both parameters on PreparedStatements and the ResultSet columns are based on indexes that start with 1, not 0.

Screenshots of Code:

App.java × Menu.java DBConnection.java sportsDao.java sport.java

```java
package Application;

public class App {

    public static void main(String[] args) {
        Menu menu = new Menu();
        menu.start();

    }

}

```

```java
package Dao;

import java.sql.Connection;


public class sportsDao {


    private Connection connection;
    private final String GET_SPORTS_QUERY = "SELECT * FROM sports";
    private final String GET_SPORTS_BY_ID_QUERY = "SELECT * FROM SPORTS WHERE id = ?";
    private final String CREATE_NEW_SPORTS_QUERY = "INSERT INTO SPORTS(name) VALUES(?)";
    private final String DELETE_SPORTS_BY_ID_QUERY = "DELETE FROM sports WHERE id = ?";

    public sportsDao() {
        connection = DBConnection.getConnection();

    }

    public List<sport> getSports() throws SQLException {
        ResultSet rs = connection.prepareStatement(GET_SPORTS_QUERY).executeQuery();
        List<sport> sports = new ArrayList<sport>();

        while (rs.next()) {
            sports.add(populateSport(rs.getInt(1), rs.getString(2)));
        }

        return sports;
    }

    public sport getSportById(int id) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(GET_SPORTS_BY_ID_QUERY);
        ps.setInt(1, id);
        ResultSet rs = ps.executeQuery();
        rs.next();
        return populateSport(rs.getInt(1), rs.getString(2));
    }

    public void createNewSport(String sportName) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(CREATE_NEW_SPORTS_QUERY);
        ps.setString(1, sportName);
        ps.executeUpdate();
    }

    public void deleteSportById(int id) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(DELETE_SPORTS_BY_ID_QUERY);
        ps.setInt(1, id);
        ps.executeUpdate();
    }

    private sport populateSport(int id, String name) throws SQLException {
        return new sport();
    }


}

```

App.java Menu.java DBConnection.java × sportsDao.java sport.java

```java
package Dao;

import java.sql.Connection;


public class DBConnection {


    private final static String URL = "jdbc:mysql://localhost:3306/sports";
    private final static String USERNAME = "root";
    private final static String PASSWORD = "Nalabear09!";
    private static Connection connection;
    private static DBConnection instance;

    private DBConnection(Connection connection) {
        this.connection = connection;
    }

    public static Connection getConnection() {
        if (instance == null) {
            try {
                connection = DriverManager.getConnection(URL, USERNAME, PASSWORD);
                instance = new DBConnection(connection);
                System.out.println("Connection Successful.");
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
        return DBConnection.connection;


    }


}

```

App.java | Menu.java × | DBConnection.java | sportsDao.java | sport.java

```java
package Application;

import java.sql.SQLException;

public class Menu {


    private sportsDao sportsDao = new sportsDao();
    private Scanner scanner = new Scanner(System.in);
    private List<String> options = Arrays.asList(
                    "Display Sports",
                    "Display a Sport",
                    "Create Sport",
                    "Delete Sport");



    public void start() {
        String selection = "";

        do {

            printMenu();
            selection = scanner.nextLine();

            try {

                if (selection.equals("1")) {
                    displaySports();
                } else if (selection.equals("2")) {
                    displaySport();
                } else if (selection.equals("3")) {
                    createSport();
                }   else if (selection.equals("4")) {
                    deleteSport();
                }
            } catch (SQLException e) {
                e.printStackTrace();
            }



            System.out.println("Press enter to continue...");
            scanner.nextLine();
        } while (!selection.equals("-1"));
    }

    private void printMenu() {
        System.out.println("Select an Option:\n----------------------------");
        for (int i= 0; i < options.size(); i++) {
            System.out.println(i + 1 + ")" + options.get(i));
        }
    }

    private void displaySports() throws SQLException {
        List<sport> sports = sportsDao.getSports();
        for (sport sport : sports) {
            System.out.println(sport.getSportId() + ": " + sport.getName());
        }
    }
```

```java
            System.out.println("Press enter to continue...");
            scanner.nextLine();
        } while (!selection.equals("-1"));
    }

    private void printMenu() {
        System.out.println("Select an Option:\n----------------------------");
        for (int i= 0; i < options.size(); i++) {
            System.out.println(i + 1 + ")" + options.get(i));
        }
    }

    private void displaySports() throws SQLException {
        List<sport> sports = sportsDao.getSports();
        for (sport sport : sports) {
            System.out.println(sport.getSportId() + ": " + sport.getName());
        }
    }

    private void displaySport() throws SQLException {
        System.out.print("Enter Sport id: ");
        int id = Integer.parseInt(scanner.nextLine());
        sport sport = sportsDao.getSportById(id);
        System.out.println(sport.getSportId() + ": " + sport.getName());


    }

    private void createSport() throws SQLException {
        System.out.print("Enter new team name:");
        String teamname = scanner.nextLine();
        sportsDao.createNewSport(sportType);
    }

    private void deleteSport() throws SQLException {
        System.out.print("Enter sport id to delete:");
        int id = Integer.parseInt(scanner.nextLine());
        sportsDao.deleteSportById(id);
    }


}
```

App.java Menu.java DBConnection.java sportsDao.java sport.java

```java
package Entity;

public class sport {

    private int sportId;
    private String sportType;




    public void Sport(int sportId, String sportType) {
        this.setSportId(sportId);
        this.setSportType(sportType);


    }
    public int getSportId() {
        return sportId;
    }
    public void setSportId(int sportId) {
        this.sportId = sportId;
    }
    public String getSportType() {
        return sportType;
    }
    public void setSportType(String sportType) {
        this.sportType = sportType;
    }


}
```

Screenshots of Running Application:

Javadoc Declaration Console

App (1) [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (Apr 30, 2022, 11:36:59 PM)

```
java.sql.SQLException: No suitable driver found for jdbc:mysql://localhost:3306/sports
        at java.sql/java.sql.DriverManager.getConnection(DriverManager.java:706)
        at java.sql/java.sql.DriverManager.getConnection(DriverManager.java:229)
        at Dao.DBConnection.getConnection(DBConnection.java:24)
        at Dao.sportsDao.<init>(sportsDao.java:23)
        at Application.Menu.<init>(Menu.java:14)
        at Application.App.main(App.java:6)
Select an Option:
----------------------------
1)Display Sports
2)Display a Sport
3)Create Sport
4)Delete Sport
```

URL to GitHub Repository:

https://github.com/Framos22/MySQL-Asisignment-4