Relational Databases with MySQL Week 10 Coding 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:

Application.java × Sport.java SportDB.java

```java
package com.promineotech;

import java.sql.SQLException;
import java.util.Scanner;

public class Application {

  public static void main(String[] args) throws SQLException {

    String selection = "";

do {
    SportDB database = new SportDB();
    System.out.println("Please select a menu option:");
    System.out.println("---------------------------");
    System.out.println("1) Display Sports");
    System.out.println("2) Add Sport");
    System.out.println("3) Update Sport");
    System.out.println("4) Delete Sport");
    Scanner input = new Scanner(System.in);

    selection = input.nextLine();

    if (selection.equals("1")) {
      database.getAll();
    } else if (selection.equals("2")) {
      System.out.println("Please enter the name of the sport you'd like to add:");
      String sportName = input.nextLine();
      database.addSport(sportName);
    } else if (selection.equals("3")) {
      System.out.println("Please enter the number of the sport you'd like to change:");
      String idName = input.nextLine();
      System.out.println("Please enter the new name of this sport:");
      String newSportName = input.nextLine();
      database.updateSport(newSportName, idName);
    } else if (selection.equals("4")) {
      System.out.println("Please enter the number of the sport you'd like to delete:");
      String idName = input.nextLine();
      database.deleteSport(idName);
    }


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

  }

}
```

Application.java | Sport.java | SportDB.java

```java
package com.promineotech;

public class Sport {
  private int id;
  private String name;

  public Sport() {

  }

  public String getName() {
    return name;
  }

  public void setName(String name) {
    this.name = name;
  }

  public int getId() {
    return id;
  }

  public void setId(int id) {
    this.id = id;
  }


  @Override
  public String toString() {
    return String.format("[%s] %s", getId(), getName());
  }
}

```

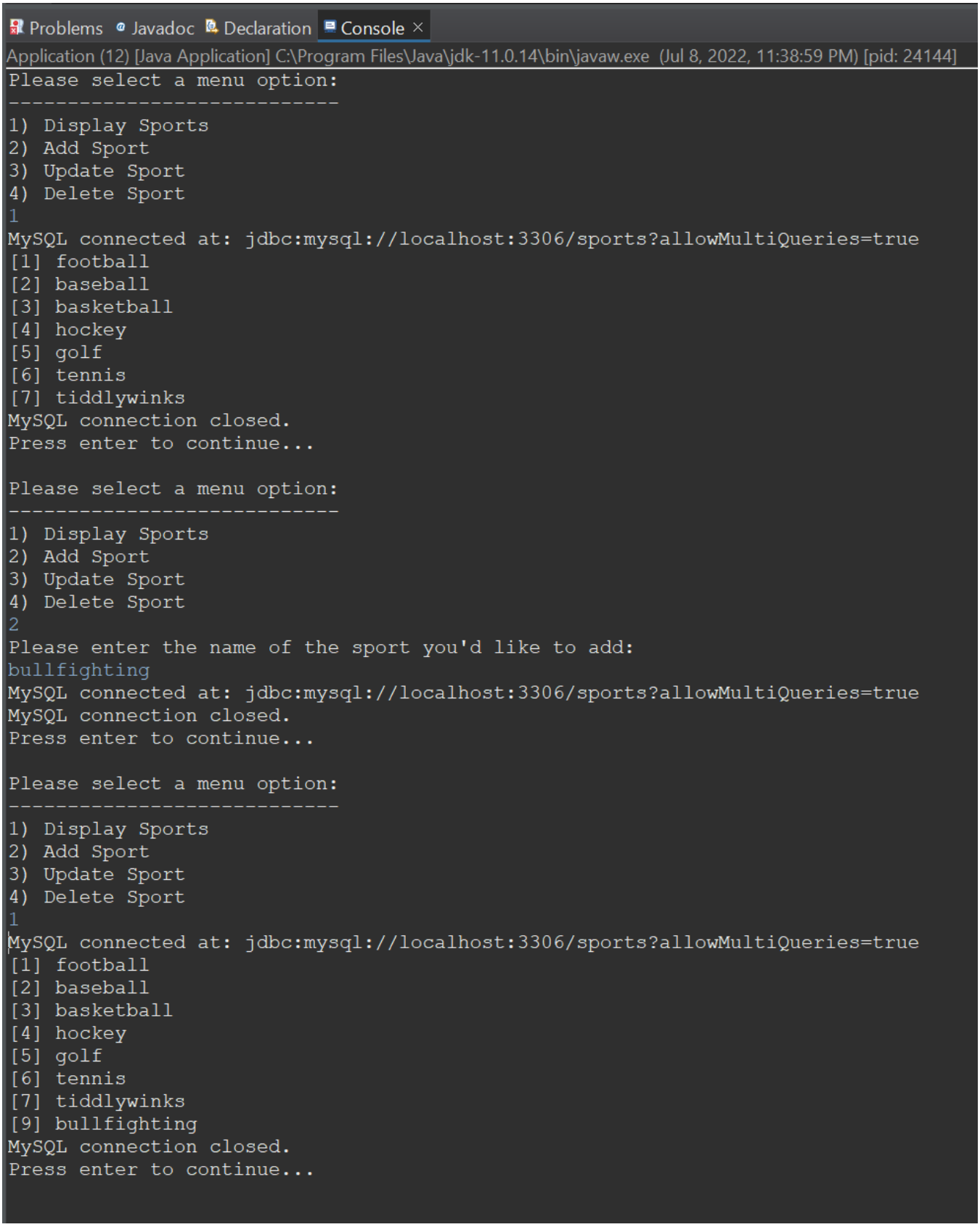
Application.java | Sport.java | SportDB.java

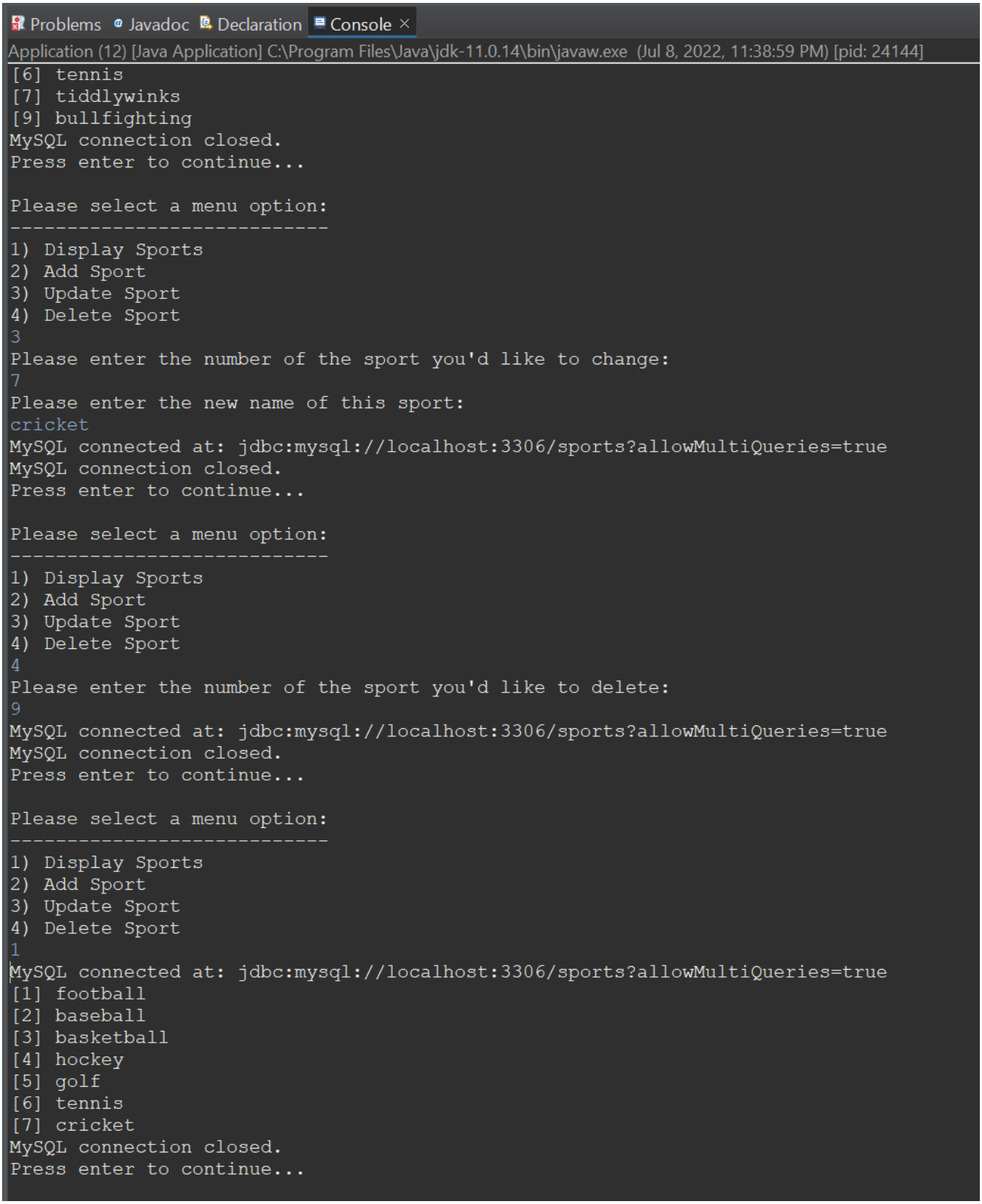
```java
package com.promineotech;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class SportDB {
  private String connectionString = "jdbc:mysql://localhost:3306/sports?allowMultiQueries=true";
  private String username = "root";
  private String password = "password";

  public void getAll() {
    try {
      // Open Connection
      Connection connection = DriverManager.getConnection(connectionString, username, password);
      System.out.println("MySQL connected at: " + connectionString);

      //Prepare SQL Statement
      String sql = "SELECT * FROM sports;";
      PreparedStatement statement = connection.prepareStatement(sql);
      //Bind parameters

      //Execute SQL Statement / Open Reader
      ResultSet rs = statement.executeQuery();

      //Read Row by Row
      while(rs.next()) {
        String id = rs.getString("id");
        String name = rs.getString("name");
        System.out.printf("[%s] %s%n", id, name);
      }
      //Close connection
      connection.close();
      System.out.println("MySQL connection closed.");
    } catch (SQLException e) {
      System.out.println("Database connection error:" + e.getMessage());
      e.printStackTrace();
    }
  }

public void addSport(String sportName) throws SQLException {
  try {
    // Open Connection
    Connection connection = DriverManager.getConnection(connectionString, username, password);
    System.out.println("MySQL connected at: " + connectionString);

    //Prepare SQL Statement
    String sql = "INSERT INTO sports (name) VALUES (?);";
    PreparedStatement statement = connection.prepareStatement(sql);
    //Bind parameters
    statement.setString(1, sportName);
    //Execute SQL Statement / Open Reader
    statement.executeUpdate();

```

Application.java | Sport.java | SportDB.java

```
56
57      //Read Row by Row
58
59      //Close connection
60      connection.close();
61      System.out.println("MySQL connection closed.");
62    } catch (SQLException e) {
63      System.out.println("Database connection error:" + e.getMessage());
64      e.printStackTrace();
65    }
66  }
67
68  public void updateSport(String idName, String newSportName) throws SQLException {
69  try {
70    // Open Connection
71    Connection connection = DriverManager.getConnection(connectionString, username, password);
72    System.out.println("MySQL connected at: " + connectionString);
73
74    //Prepare SQL Statement
75    String sql = "UPDATE sports SET name = ? WHERE id = ?;";
76    PreparedStatement statement = connection.prepareStatement(sql);
77    //Bind parameters
78    statement.setString(2, newSportName);
79    statement.setString(1, idName);
80    //Execute SQL Statement / Open Reader
81    statement.executeUpdate();
82
83    //Read Row by Row
84
85    //Close connection
86    connection.close();
87    System.out.println("MySQL connection closed.");
88  } catch (SQLException e) {
89    System.out.println("Database connection error:" + e.getMessage());
90    e.printStackTrace();
91  }
92  }
93
94  public void deleteSport(String idName) throws SQLException {
95  try {
96    // Open Connection
97    Connection connection = DriverManager.getConnection(connectionString, username, password);
98    System.out.println("MySQL connected at: " + connectionString);
99
100   //Prepare SQL Statement
101   String sql = "DELETE FROM sports WHERE id = ?;";
102   PreparedStatement statement = connection.prepareStatement(sql);
103   //Bind parameters
104   statement.setString(1, idName);
105   //Execute SQL Statement / Open Reader
106   statement.executeUpdate();
107
108   //Read Row by Row
109
110   //Close connection
111   connection.close();
112   System.out.println("MySQL connection closed.");
113 } catch (SQLException e) {
114   System.out.println("Database connection error:" + e.getMessage());
115   e.printStackTrace();
116 }
117 }
118
119 }
120
```

Screenshots of Running Application:





URL to GitHub Repository:

https://github.com/jprengaman/BESD-Week10