

Relational Databases with MySQL Week 5 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, with your Java project code, to the repository. Lastly, in the Learning Management System, click the "Add Submission" button and paste the URL to your GitHub repository.

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 `PreparedStatement.executeQuery()` is only for Reading data and `.executeUpdate()` is used for Creating, Updating, and Deleting data.

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

Screenshots of Code:

```
1 • CREATE DATABASE if not exists sports;
2 • USE sports;
3 • DROP TABLE if exists sports;
4 • CREATE TABLE sports(
5     id INT NOT NULL AUTO_INCREMENT,
6     name VARCHAR(40) NOT NULL,
7     PRIMARY KEY(id)
8 );
```

```
1 package application;
2
3 public class Application {
4
5     public static void main(String[] args) {
6         Menu menu = new Menu();
7         menu.start();
8     }
9
10 }
11 |
```

```
1 package dao;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.SQLException;
6
7 public class DBConnection {
8
9     private final static String URL = "jdbc:mysql://localhost:3306/sports";
10    private final static String USERNAME = "root";
11    private final static String PASSWORD = "12345";
12    private static Connection connection;
13    private static DBConnection instance;
14
15    private DBConnection(Connection connection) {
16        this.connection = connection;
17    }
18
19    public static Connection getConnection() {
20        if (instance == null) {
21            try {
22                connection = DriverManager.getConnection(URL, USERNAME, PASSWORD);
23                instance = new DBConnection(connection);
24                System.out.println("Connection successful");
25            } catch (SQLException e) {
26                e.printStackTrace();
27            }
28        }
29        return DBConnection.connection;
30    }
31
32 }
33
```

```
1 package entity;
2
3 public class Sport {
4
5     private int id;
6     private String name;
7
8     public Sport(int id, String name) {
9         this.setId(id);
10        this.setName(name);
11    }
12
13
14    public int getId() {
15        return id;
16    }
17
18    public void setId(int id) {
19        this.id = id;
20    }
21
22    public String getName() {
23        return name;
24    }
25
26    public void setName(String name) {
27        this.name = name;
28    }
29
30 }
31
```

```

1 package dao;
2
3 import java.sql.Connection;
4 import java.sql.PreparedStatement;
5 import java.sql.ResultSet;
6 import java.sql.SQLException;
7 import java.util.ArrayList;
8 import java.util.List;
9
10 import entity.Sport;
11
12 public class SportDao {
13
14     private Connection connection;
15     private final String GET_SPORTS_QUERY = "SELECT * FROM Sports";
16     private final String GET_SPORT_BY_ID_QUERY = "SELECT * FROM sports WHERE id = ?";
17     private final String CREATE_NEW_SPORT_QUERY = "INSERT INTO sports(name) VALUES(?)";
18     private final String UPDATE_SPORT_BY_ID_QUERY = "UPDATE sports SET name = ? WHERE id = ?";
19     private final String DELETE_SPORT_BY_ID_QUERY = "DELETE FROM sports WHERE id = ?";
20
21     public SportDao() {
22
23         connection = DBConnection.getConnection();
24     }

```

```

26     public List<Sport> getSports() throws SQLException{
27         ResultSet rs = connection.prepareStatement(GET_SPORTS_QUERY).executeQuery();
28         List<Sport> sports = new ArrayList<Sport>();
29
30         while (rs.next()) {
31             sports.add(populateSport(rs.getInt(1), rs.getString(2)));
32         }
33
34         return sports;
35     }
36
37     public Sport getSportById(int id) throws SQLException {
38         PreparedStatement ps = connection.prepareStatement(GET_SPORT_BY_ID_QUERY);
39         ps.setInt(1, id);
40         ResultSet rs = ps.executeQuery();
41         rs.next();
42         return populateSport(rs.getInt(1), rs.getString(2));
43     }
44
45     public void createNewSport(String sportName) throws SQLException {
46         PreparedStatement ps = connection.prepareStatement(CREATE_NEW_SPORT_QUERY);
47         ps.setString(1, sportName);
48         ps.executeUpdate();
49     }
50 }
51

```

```

52 public void updateSportByID(int id, String name) throws SQLException {
53     PreparedStatement ps = connection.prepareStatement(UPDATE_SPORT_BY_ID_QUERY);
54     ps.setString(1, name);
55     ps.setInt(2, id);
56     ps.executeUpdate();
57 }
58
59 public void deleteSportByID(int id) throws SQLException {
60     PreparedStatement ps = connection.prepareStatement(DELETE_SPORT_BY_ID_QUERY);
61     ps.setInt(1, id);
62     ps.executeUpdate();
63 }
64
65 private Sport populateSport(int id, String name) {
66     return new Sport(id, name);
67 }
68
69 }
70

```

```

1 package application;
2
3 import java.sql.SQLException;
4 import java.util.Arrays;
5 import java.util.List;
6 import java.util.Scanner;
7
8 import dao.SportDao;
9 import entity.Sport;
10
11 public class Menu {
12
13     private SportDao sportDao = new SportDao();
14     private Scanner scanner = new Scanner(System.in);
15     private List<String> options = Arrays.asList(
16         "Display Sports",
17         "Display a Sport",
18         "Create Sport",
19         "Update Sport",
20         "Delete Sport");
21
22     public void start() {
23         String selection = "";
24
25         do {
26             printMenu();
27             selection = scanner.nextLine();
28

```

```

25         do {
26             printMenu();
27             selection = scanner.nextLine();
28         try {
29             if (selection.equals("1")) {
30                 displaySports();
31             } else if (selection.equals("2")) {
32                 displaySport();
33             } else if (selection.equals("3")) {
34                 createSport();
35             } else if (selection.equals("4")) {
36                 updateSport();
37             } else if (selection.equals("5")) {
38                 deleteSport();
39             }
40         } catch (SQLException e) {
41             e.printStackTrace();
42         }
43     }

```

```

44         System.out.println("Press enter to continue");
45         scanner.nextLine();
46     } while (!selection.equals("-1"));
47 }
48
49
50 private void printMenu() {
51     System.out.println("Select an option:\n-----");
52     for (int i = 0; i < options.size(); i++) {
53         System.out.println(i+1+ " " + options.get(i));
54     }
55 }
56
57 private void displaySports() throws SQLException {
58     List<Sport> sports = sportDao.getSports();
59     for (Sport sport : sports) {
60         System.out.println(sport.getId() + ": " + sport.getName());
61     }
62 }
63

```

```

64 private void displaySport() throws SQLException {
65     System.out.print("Enter sport id: ");
66     int id = Integer.parseInt(scanner.nextLine());
67     Sport sport = sportDao.getSportById(id);
68     System.out.println(sport.getId() + ": " + sport.getName());
69 }
70
71 private void createSport() throws SQLException {
72     System.out.print("Enter new sport name:");
73     String sportName = scanner.nextLine();
74     sportDao.createNewSport(sportName);
75 }
76 private void updateSport() throws SQLException {
77     System.out.println("Enter sport id to update:");
78     int id = Integer.parseInt(scanner.nextLine());
79     String name = scanner.nextLine();
80     sportDao.updateSportById(id, name);
81 }
82 private void deleteSport() throws SQLException {
83     System.out.println("Enter sport id to delete:");
84     int id = Integer.parseInt(scanner.nextLine());
85     sportDao.deleteSportById(id);
86 }
87 }
88 }

```

Screenshots of Running Application:

```

Application [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (Oct 31, 2020 11:51:36)
Connection successful
Select an option:
-----
1)Display Sports
2)Display a Sport
3)Create Sport
4)Update Sport
5>Delete Sport

```



```
Problems Javadoc Declaration Console
Application [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (Oct 31, 2020 11:51:36 AM)
Connection successful
Select an option:
-----
1)Display Sports
2)Display a Sport
3)Create Sport
4)Update Sport
5)Delete Sport
1
1: Tennis
Press enter to continue
```

```
Problems Javadoc Declaration Console
Application [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (Oct 31, 2020 11:53:18 AM)
Connection successful
Select an option:
-----
1)Display Sports
2)Display a Sport
3)Create Sport
4)Update Sport
5)Delete Sport
2
Enter sport id: 1
1: Tennis
Press enter to continue
```

```
Application [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (Oct 31, 2020 11:53:44 AM)
Connection successful
Select an option:
-----
1)Display Sports
2)Display a Sport
3)Create Sport
4)Update Sport
5)Delete Sport
3
Enter new sport name:Soccer
Press enter to continue
```

```
3)Create Sport
4)Update Sport
5)Delete Sport
1
1: Tennis
3: Soccer
Press enter to continue
```

```
Problems Javadoc Declaration Console
Application [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (Oct 31, 2020 11:54:50 AM)
Connection successful
Select an option:
-----
1)Display Sports
2)Display a Sport
3)Create Sport
4)Update Sport
5)Delete Sport
4
Enter sport id to update:
3
Football
Press enter to continue
```

```
Press enter to continue

Select an option:
-----
1)Display Sports
2)Display a Sport
3)Create Sport
4)Update Sport
5)Delete Sport
1
1: Tennis
3: Football
Press enter to continue
```

```
Problems Javadoc Declaration Console
Application [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (Oct 31, 2020 11:56:31 AM)
Connection successful
Select an option:
-----
1)Display Sports
2)Display a Sport
3)Create Sport
4)Update Sport
5)Delete Sport
5
Enter sport id to delete:
1
Press enter to continue
```

```
5)Delete Sport  
1  
3: Football  
4: Soccer  
5: Golf  
6: Basketball  
Press enter to continue
```

URL to GitHub Repository:

<https://github.com/lcuevas6/week-5-sql-assignment.git>