

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 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 `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 sportsteams;
2
3 use sportsteams;
4
5 DROP TABLE IF EXISTS sportsteams;
6
7 CREATE TABLE sportsteams (
8     id int(10) NOT NULL auto_increment,
9     name varchar(50) NOT NULL,
10    PRIMARY KEY(id)
11 );
```

```
[mysql> use sportsteams;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

Database changed

```
[mysql> show tables;
```

```
+-----+
| Tables_in_sportsteams |
+-----+
| sportsteams           |
+-----+
1 row in set (0.01 sec)
```

```
[mysql> desc sportsteams;
```

```
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | int           | NO   | PRI | NULL    | auto_increment |
| name  | varchar(50)   | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
1 package application;
2
3 public class Application {
4
5     public static void main(String[] args) {
6
7         Menu menu = new Menu();
8         menu.start();
9     }
10 }
11
12
13
14
```

Writable Smart Insert 11:6:143

```
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.TeamDao;
9 import entity.Team;
10
11 public class Menu {
12
13     private TeamDao teamDao = new TeamDao();
14     private Scanner scanner = new Scanner(System.in);
15     private List<String> options = Arrays.asList(
16         "Display Teams",
17         "Create Team",
18         "Update Team",
19         "Delete Team",
20         "Exit");
21
22     public void start() {
23         String selection = "";
24         do {
25             printMenu();
26             selection = scanner.nextLine();
27             try {
28                 if (selection.equals("1")) {
29                     displayTeams();
30                 } else if (selection.equals("2")) {
31                     createTeam();
32                 } else if (selection.equals("3")) {
33                     updateTeam();
34                 } else if (selection.equals("4")) {
35                     deleteTeam();
36                 }
37             } catch (SQLException e) {
38                 e.printStackTrace();
39             }
40             System.out.println("<< < Press enter to continue >> >");
41             scanner.nextLine();
42         } while (!selection.equals("5"));
43         System.out.println("Goodbye!");
44     }
45
46
47
48
49
50
51
```

Writable Smart Insert 06:1:2060

```
Application.java Menu.java DBConnection.java TeamDao.java Team.java
48 } while (!selection.equals("5"));
49
50 System.out.println("Goodbye!");
51 }
52
53
54 private void printMenu() {
55     System.out.println("Select an Option:\n-----");
56
57     for (int i = 0; i < options.size(); i++) {
58         System.out.println(i + 1 + " " + options.get(i));
59     }
60 }
61
62
63 private void displayTeams() throws SQLException {
64     List<Team> teams = teamDao.getTeams();
65
66     for (Team team : teams) {
67         System.out.println(team.getTeamId() + " : " + team.getName());
68     }
69 }
70
71
72 private void createTeam() throws SQLException {
73     System.out.print("Enter new team name:");
74     String teamName = scanner.nextLine();
75     teamDao.createNewTeam(teamName);
76 }
77
78
79 private void updateTeam() throws SQLException {
80     System.out.print("Enter team id to update:");
81     int id = Integer.parseInt(scanner.nextLine());
82
83     System.out.print("Enter new team name: ");
84     String name = scanner.nextLine();
85
86     teamDao.updateTeam(id, name);
87 }
88
89
90 private void deleteTeam() throws SQLException {
91     System.out.print("Enter team id to delete:");
92     int id = Integer.parseInt(scanner.nextLine());
93     teamDao.deleteTeamById(id);
94 }
95
96
97 }
98
```

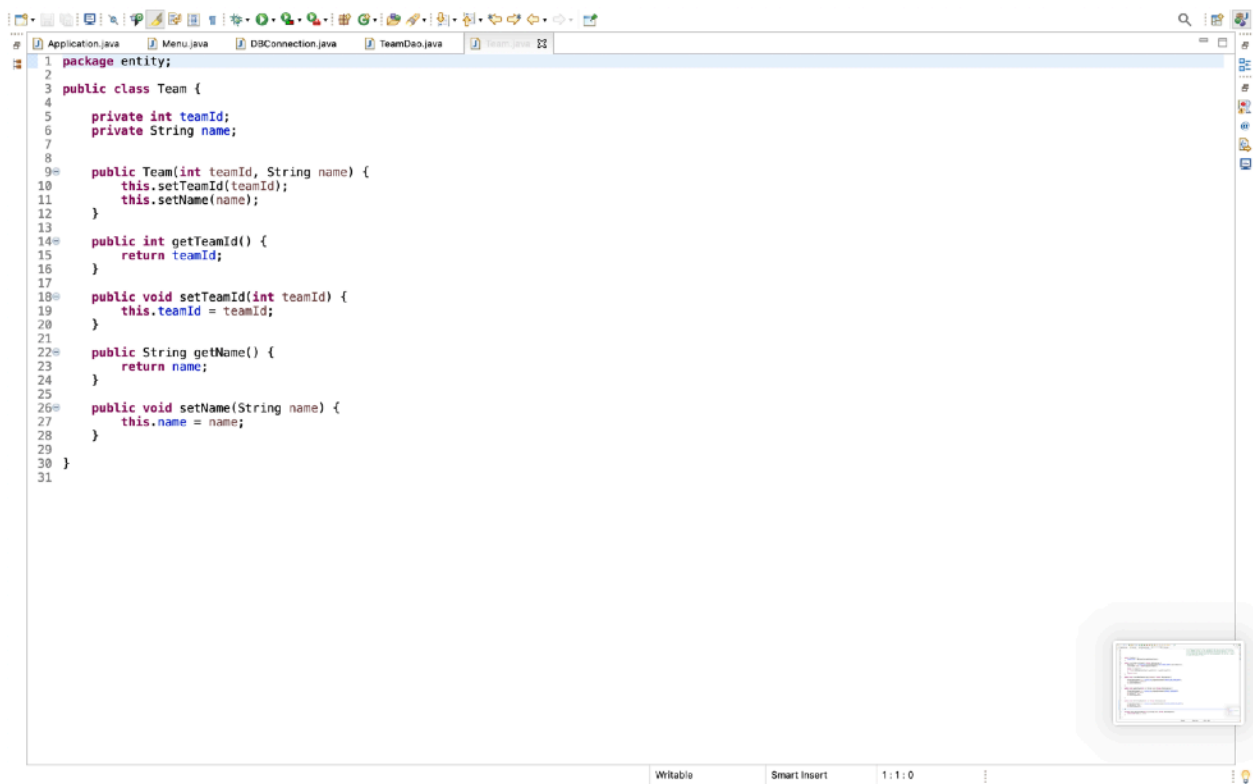
Writable Smart Insert 06 : 1 : 2060

```
Application.java Menu.java DBConnection.java TeamDao.java Team.java
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/sportsteam";
10    private final static String USERNAME = System.getenv("USERNAME");
11    private final static String PASSWORD = System.getenv("PASSWORD");
12    private static Connection connection;
13    private static DBConnection instance;
14
15    @SuppressWarnings("static-access")
16    private DBConnection(Connection connection) {
17        this.connection = connection;
18    }
19
20    public static Connection getConnection() {
21        if (instance == null) {
22            try {
23                connection = DriverManager.getConnection(URL, USERNAME, PASSWORD);
24                instance = new DBConnection(connection);
25                System.out.println("We're In!!\n");
26            } catch (SQLException e) {
27                e.printStackTrace();
28            }
29        }
30        return DBConnection.connection;
31    }
32
33 }
34
35 }
36
37 }
38
```

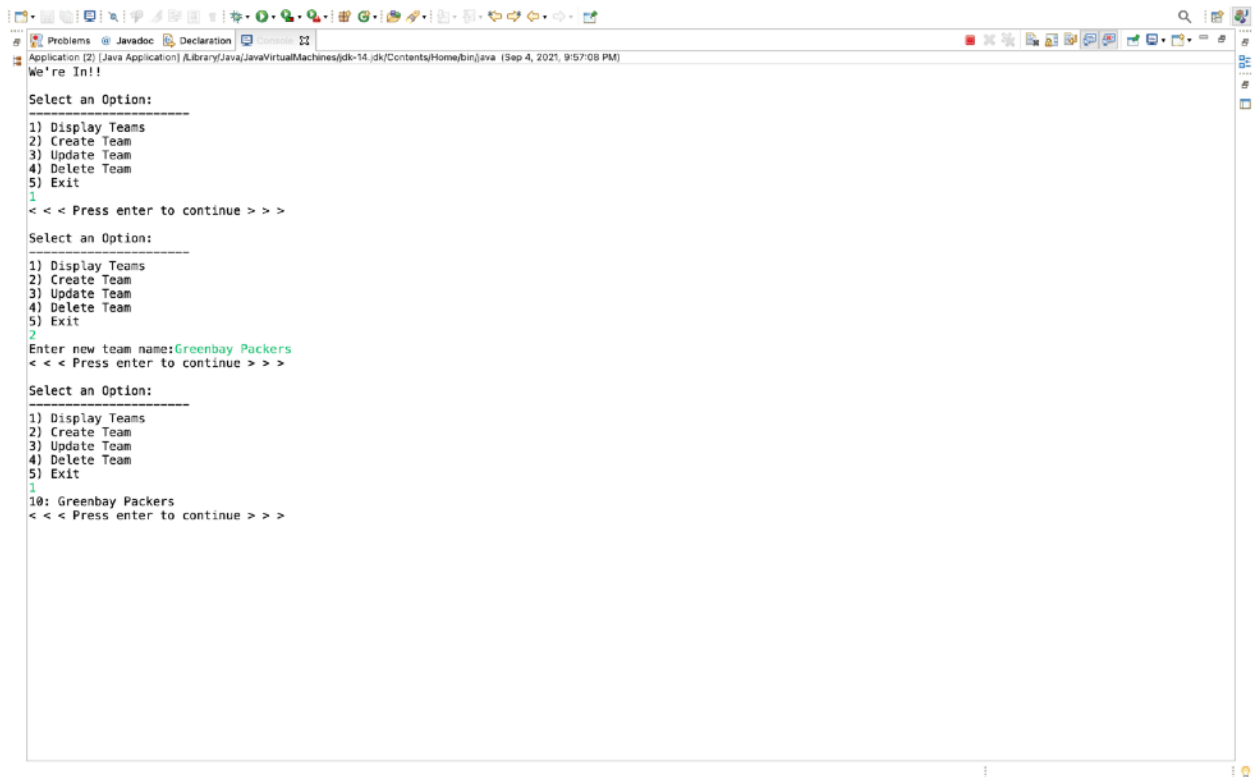
Writable Smart Insert 11 : 70 : 346

```
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.Team;
11
12 public class TeamDao {
13
14     private Connection connection;
15     private final String GET_TEAMS_QUERY = "SELECT * FROM sportsteams";
16     private final String CREATE_NEW_TEAM_QUERY = "INSERT INTO sportsteams(name) VALUES(?)";
17     private final String UPDATE_TEAM_QUERY = "UPDATE sportsteams SET name = ? WHERE id = ?";
18     private final String DELETE_TEAM_BY_ID_QUERY = "DELETE FROM sportsteams WHERE id = ?";
19
20     // --- This kept throwing a "com.mysql.cj.jdbc.exceptions.
21     // MySQLDataTruncation:Data truncation: Truncated incorrect DOUBLE
22     // value:" exception and I could not get it to work, there are
23     // no DOUBLE values in the database. Any advice would be welcome,
24     // I researched solutions online that were over my head.
25     // This was the hardest part of the assignment for me but I didn't
26     // want to leave it out.
27
28     public TeamDao() {
29         connection = DBConnection.getConnection();
30     }
31
32     public List<Team> getTeams() throws SQLException {
33         ResultSet rs = connection.prepareStatement(GET_TEAMS_QUERY).executeQuery();
34         List<Team> teams = new ArrayList<Team>();
35
36         while (rs.next()) {
37             teams.add(populateTeam(rs.getInt(1), rs.getString(2)));
38         }
39
40         return teams;
41     }
42
43     public void createNewTeam(String teamName) throws SQLException {
44         PreparedStatement ps = connection.prepareStatement(CREATE_NEW_TEAM_QUERY);
45         ps.setString(1, teamName);
46         ps.executeUpdate();
47     }
48
49     public void updateTeam(int id, String name) throws SQLException {
50         PreparedStatement ps = connection.prepareStatement(UPDATE_TEAM_QUERY);
51         ps.setString(2, name);
52         ps.setInt(1, id);
53     }
54
55     public void deleteTeamById(int id) throws SQLException {
56         PreparedStatement ps = connection.prepareStatement(DELETE_TEAM_BY_ID_QUERY);
57         ps.setInt(1, id);
58         ps.executeUpdate();
59     }
60
61     private Team populateTeam(int id, String name) throws SQLException {
62         return new Team(id, name);
63     }
64 }
```

```
19 // value:" exception and I could not get it to work, there are
20 // no DOUBLE values in the database. Any advice would be welcome,
21 // I researched solutions online that were over my head.
22 // This was the hardest part of the assignment for me but I didn't
23 // want to leave it out.
24
25 public TeamDao() {
26     connection = DBConnection.getConnection();
27 }
28
29 public List<Team> getTeams() throws SQLException {
30     ResultSet rs = connection.prepareStatement(GET_TEAMS_QUERY).executeQuery();
31     List<Team> teams = new ArrayList<Team>();
32
33     while (rs.next()) {
34         teams.add(populateTeam(rs.getInt(1), rs.getString(2)));
35     }
36
37     return teams;
38 }
39
40 public void createNewTeam(String teamName) throws SQLException {
41     PreparedStatement ps = connection.prepareStatement(CREATE_NEW_TEAM_QUERY);
42     ps.setString(1, teamName);
43     ps.executeUpdate();
44 }
45
46 public void updateTeam(int id, String name) throws SQLException {
47     PreparedStatement ps = connection.prepareStatement(UPDATE_TEAM_QUERY);
48     ps.setString(2, name);
49     ps.setInt(1, id);
50     ps.executeUpdate();
51 }
52
53 public void deleteTeamById(int id) throws SQLException {
54     PreparedStatement ps = connection.prepareStatement(DELETE_TEAM_BY_ID_QUERY);
55     ps.setInt(1, id);
56     ps.executeUpdate();
57 }
58
59 private Team populateTeam(int id, String name) throws SQLException {
60     return new Team(id, name);
61 }
62 }
```



Screenshots of Running Application:



```
Application [2] [Java Application] (Library\Java\JavaVirtualMachines\jdk-14_jdk\Contents\Home\bin\java (Sep 4, 2021, 9:58:43 PM)
We're In!!

Select an Option:
-----
1) Display Teams
2) Create Team
3) Update Team
4) Delete Team
5) Exit
1
10: Greenbay Packers
< < Press enter to continue > >

Select an Option:
-----
1) Display Teams
2) Create Team
3) Update Team
4) Delete Team
5) Exit
4
Enter team id to delete:10
< < Press enter to continue > >

Select an Option:
-----
1) Display Teams
2) Create Team
3) Update Team
4) Delete Team
5) Exit
1
< < Press enter to continue > >
```

```
Application [2] [Java Application] (Library\Java\JavaVirtualMachines\jdk-14_jdk\Contents\Home\bin\java (Sep 4, 2021, 9:59:12 PM)
We're In!!

Select an Option:
-----
1) Display Teams
2) Create Team
3) Update Team
4) Delete Team
5) Exit
1
< < Press enter to continue > >

Select an Option:
-----
1) Display Teams
2) Create Team
3) Update Team
4) Delete Team
5) Exit
2
Enter new team name:Denver Broncos
< < Press enter to continue > >

Select an Option:
-----
1) Display Teams
2) Create Team
3) Update Team
4) Delete Team
5) Exit
1
11: Denver Broncos
< < Press enter to continue > >

Select an Option:
-----
1) Display Teams
2) Create Team
3) Update Team
4) Delete Team
5) Exit
3
Enter team id to update:11
Enter new team name: Broncos
com.mysql.cj.jdbc.exceptions.MySQLDataTruncation: Data truncation: Truncated incorrect DOUBLE value: 'Broncos'
< < Press enter to continue > >
at mysql.connector.java@8.0.26/com.mysql.cj.jdbc.exceptions.SQLExceptionsMapping.translateException(SQLExceptionsMapping.java:104)
at mysql.connector.java@8.0.26/com.mysql.cj.jdbc.ClientPreparedStatement.executeInternal(ClientPreparedStatement.java:953)
at mysql.connector.java@8.0.26/com.mysql.cj.jdbc.ClientPreparedStatement.executeUpdateInternal(ClientPreparedStatement.java:1092)
at mysql.connector.java@8.0.26/com.mysql.cj.jdbc.ClientPreparedStatement.executeUpdateInternal(ClientPreparedStatement.java:1040)
```

```

# Problems JavaDoc Declaration
<terminated> Application (2) [Java Application] [Library\Java\JavaVirtualMachines\jdk-14.jdk\Contents\Home\bin\java (Sep 4, 2021, 9:59:12 PM - 10:00:10 PM)]
2
Enter new team name: Denver Broncos
< < < Press enter to continue > > >

Select an Option:
-----
1) Display Teams
2) Create Team
3) Update Team
4) Delete Team
5) Exit
1
11: Denver Broncos
< < < Press enter to continue > > >

Select an Option:
-----
1) Display Teams
2) Create Team
3) Update Team
4) Delete Team
5) Exit
3
Enter team id to update: 11
Enter new team name: Broncos
com.mysql.cj.jdbc.exceptions.MysqlDataTruncation: Data truncation: Truncated incorrect DOUBLE value: 'Broncos'
< < < Press enter to continue > > >
    at mysql.connector.java@8.0.26/com.mysql.cj.jdbc.exceptions.SQLExceptionsMapping.translateException(SQLExceptionsMapping.java:104)
    at mysql.connector.java@8.0.26/com.mysql.cj.jdbc.ClientPreparedStatement.executeInternal(ClientPreparedStatement.java:953)
    at mysql.connector.java@8.0.26/com.mysql.cj.jdbc.ClientPreparedStatement.executeUpdateInternal(ClientPreparedStatement.java:1092)
    at mysql.connector.java@8.0.26/com.mysql.cj.jdbc.ClientPreparedStatement.executeUpdateInternal(ClientPreparedStatement.java:1040)
    at mysql.connector.java@8.0.26/com.mysql.cj.jdbc.ClientPreparedStatement.executeLargeUpdate(ClientPreparedStatement.java:1350)
    at mysql.connector.java@8.0.26/com.mysql.cj.jdbc.ClientPreparedStatement.executeUpdate(ClientPreparedStatement.java:1025)
    at dao.TeamDao.updateTeam(TeamDao.java:52)
    at application.Menu.updateTeam(Menu.java:87)
    at application.Menu.start(Menu.java:38)
    at application.Application.main(Application.java:9)

Select an Option:
-----
1) Display Teams
2) Create Team
3) Update Team
4) Delete Team
5) Exit
5
< < < Press enter to continue > > >

Goodbye!

```

URL to GitHub Repository:

<https://github.com/johnbarts/mySQLweek5>