# Relational Databases with MySQL Week 6 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. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

## **Coding Steps:**

This week you will be working together as a team to create a full CRUD application.

Your console CRUD application will need to use a database to store all the application data.

As a team, decide what you want your project to do. Get instructor approval early in the week before beginning development.

You need to have at least 3 entities.

Users should be able to interact via the console (i.e. Scanner(System.in)))

Use git to collaborate.

Everyone will be graded on their individual contributions.

## **Screenshots of Code:**

# **Application:**

```
se IDE
: Run Window Help
+ Q + Q + # Ø + / / / | | | + | | | | | |
             AppetizersDA...
Appetizers
Menu.java
1 package MenuApp;
3 public class Application {
      public static void main(String[] args) {
48
          Menu menu = new Menu();
5
          menu.start();
6
7
      }
8
9 }
```

```
Run Window Help
▼ 🛂 ▼ 😘 ▼ 🔡 🕝 ▼ 🤔 🚀 ▼ 🍇 ▼ 🎘 ▼ 🎘 ▼ 🍪 ▼ 🌣 🐡
Menu.java ⋈ 🚺 AppetizersDA...

    Appetizers.java

                                                          DessertsDAO....
                                                                               DBConnection...
                                                                                                      NumberFo
 package MenuApp;
 3 import java.sql.SQLException;
 4 import java.util.Arrays;
 5 import java.util.List;
 6 import java.util.Scanner;
 8 import dao.AppetizersDAO;
   import dao.DessertsDAO;
10 import dao.EntreesDAO;
11 import entity.Appetizers;
12 import entity.Desserts;
13 import entity.Entrees;
14
15
    public class Menu {
        private AppetizersDAO AppetizersDao= new AppetizersDAO();
16
17
        private EntreesDAO EntreesDao= new EntreesDAO();
18
        private DessertsDAO DessertsDao= new DessertsDAO();
19
        private Scanner scanner = new Scanner(System.in);
20
        private List<String> options = Arrays.asList("Display Appetizer", "Create Appetizers", "Delete Appetizers",
210
                "Display Entrees", "Create Entrees", "Delete Entrees", "Display Desserts", "Create Desserts",
22
23
                "Delete Desserts");
24
25@
        public void start() {
26
            // TODO Auto-generated method stub
            String select = "";
28
            do {
29
                printMenu();
                select = scanner.nextLine();
30
31
                try {
                    if (select.equals("1")) {
32
33
                        displayAppetizers();
34
                    } else if (select.equals("2")) {
35
                        createAppetizers();
36
                    } else if (select.equals("3")) {
37
                        deleteAppetizers();
38
                    } else if (select.equals("4")) {
                        displayEntrees();
39
40
                    } else if (select.equals("5")) {
41
                        createEntrees();
42
                    } else if (select.equals("6")) {
43
                        deleteEntrees();
44
                    } else if (select.equals("7")) {
45
                        displayDesserts();
46
                    } else if (select.equals("8")) {
47
                        createDesserts();
48
                    } else if (select.equals("9")) {
49
                        deleteDesserts();
50
51
                }catch (SQLException e) {
52
                    e.printStackTrace();
53
54
55
56
            } while (!select.equals("-1"));
57
58
59
        private void deleteDesserts() throws SQLException {
60<del>0</del>
61
            // TODO Auto-generated method stub
            System.out.println("Enter dessert id to delete: ");
62
63
            int id =Integer.parseInt(scanner.nextLine());
            DessertsDao.deleteDessert(id);
64
65
66
67
680
        private void createDesserts() throws SQLException {
```

```
private void createDesserts() throws SQLException {
 69
             // TODO Auto-generated method stub
 78
             System.out.println("Enter New Dessert name:");
 71
             String dessertName = scanner.nextLine();
             System.out.println("Enter it's price: $");
 72
             double dessertPrice= scanner.nextDouble();
 73
 74
             DessertsDao.createDessert(dessertName, dessertPrice);
 75
 76
         private void displayDesserts() throws SQLException {
 778
 78
             // TODO Auto-generated method stub
             List<Desserts> Desserts = DessertsDao.getDesserts();
 79
             for (Desserts dessert : Desserts) {
 88
                 System.out.println(dessert.getDId() + ": " + dessert.getDName() +
81
82
                         ": " + dessert.getDPrice());
83
             }
        }
84
 85
         private void deleteEntrees() throws SQLException {
860
87
             // TODO Auto-generated method stub
             System.out.println("Enter Entree id to delete: ");
88
29
             int id =Integer.parseInt(scanner.nextLine());
 98
             EntreesDao.deleteEntree(id);
91
92
93@
         private void createEntrees() throws SQLException {
94
             // TODO Auto-generated method stub
 95
             System.out.println("Enter New Entree name:");
96
             String entreeName = scanner.nextLine();
97
             System.out.println("Enter it's price: $");
98
             double entreePrice= scanner.nextDouble();
99
             EntreesDao.createEntree(entreeName, entreePrice);
100
101
         private void displayEntrees() throws SQLException {
1820
             // TODO Auto-generated method stub
103
194
             List<Entrees> Entrees = EntreesDao.getEntrees();
105
             for (Entrees entree : Entrees) {
106
                 System.out.println(entree.getEId() + ": " + entree.getEName() +
107
                         ": " + entree.getEPrice());
108
199
110
         private void deleteAppetizers() throws SQLException {
1110
112
             // TODO Auto-generated method stub
113
             System.out.println("Enter Appetizer id to delete: ");
114
             int id =Integer.parseInt(scanner.nextLine());
115
             AppetizersDao.deleteAppetizer(id);
        3
116
117
1180
         private void createAppetizers() throws SQLException {
119
             // TODO Auto-generated method stub
120
             System.out.println("Enter New Appetizer name:");
121
             String appetizerName = scanner.nextLine();
122
             System.out.println("Enter it's price: $");
123
             double appetizerPrice= scanner.nextDouble();
             AppetizersDao.createAppetizer(appetizerName, appetizerPrice);
124
125
126
127⊖
         private void displayAppetizers() throws SQLException {
128
             // TODO Auto-generated method stub
129
             List<Appetizers> Appetizers = AppetizersDao.getAppetizers();
             for (Appetizers appetizer : Appetizers) {
130
                 System.out.println(appetizer.getAId() + ": " + appetizer.getAName()
131
132
                         ": " + appetizer.getAPrice());
             }
133
134
135
```

```
136
 137
          private void printMenu() {
 1389
139
              // TODO Auto-generated method stub
              System.out.println("Please Select an Options:");
 140
 141
             for (int i = 0; i < options.size(); i++) {
 142
                  System.out.println(i + 1 + ". " + options.get(i));
 143
 144
          }
 145
 146 }
 147
```

## **Entities:**

# **Appetizers**

```
package entity;
  public class Appetizers {
3
      private int AId;
      private String AName;
      private double APrice;
6
      public Appetizers(int AId, String AName, double APrice) {
80
           this.setAId(AId);
9
0
           this.setAName(AName);
1
           this.setAPrice(APrice);
.2
3
40
      public int getAId() {
.5
          return AId;
.6
7
      public void setAId(int aId) {
80
.9
          AId = aId;
9
1
28
       public String getAName() {
13
          return AName;
4
15
160
      public void setAName(String aName) {
17
          AName = aName;
.8
19
188
       public double getAPrice() {
1
           return APrice;
12
       }
13
40
      public void setAPrice(double aPrice) {
5
          APrice = aPrice;
6
17
8 }
```

## **Entrees**

```
AppetizersDAO, java
                         Appetizers.java
                                              ☑ Entrees.java ☒
1 package entity;
  3
    public class Entrees {
        private int EId;
        private String EName;
  5
  6
        private double EPrice;
  7
  80
        public Entrees(int EId, String EName, double EPrice) {
            this.setEId(EId);
 9
 10
            this.setEName(EName);
 11
            this.setEPrice(EPrice);
        }
 12
 13
 140
        public int getEId() {
            return EId;
 16
 17
 180
        public void setEId(int eId) {
 19
            EId = eId;
 20
 21
        public String getEName() {
 229
 23
            return EName;
 24
 25
 260
        public void setEName(String eName) {
 27
            EName = eName;
 28
 29
        public double getEPrice() {
 30⊖
            return EPrice;
 31
 32
 33
        public void setEPrice(double ePrice) {
 340
 35
            EPrice = ePrice;
 36
 37 }
 38
```

## **Desserts**

```
AppetizersDAO.java
                         Appetizers.java

☑ Entrees.java

1 package entity;
  3 public class Desserts {
        private int DId;
         private String DName;
  6
         private double DPrice;
  7
        public Desserts(int DId, String DName, double DPrice) {
            this.setDId(DId);
            this.setDName(DName);
 10
            this.setDPrice(DPrice);
 11
 12
         }
 13
         public int getDId() {
 149
 15
            return DId;
 16
 17
         public void setDId(int dId) {
 18⊕
            DId = dId;
 19
 20
 21
 22⊖
         public String getDName() {
 23
            return DName;
 24
         }
 25
         public void setDName(String dName) {
 26⊕
 27
            DName = dName;
 28
 29
         public double getDPrice() {
 30⊖
            return DPrice;
 31
 32
 33
         public void setDPrice(double dPrice) {
 340
            DPrice = dPrice;
 36
         }
 37 }
 38
```

# DAO:

# **Appetizers**

```
☑ Menu,java
☑ AppetizersDA... ※ ☑ Appetizers,java
☑ DessertsDAO....
☑ Application....
☑ DBConne
   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.tist;
 10 import entity.Appetizers;
11
18@
19
20
                   public AppetizersDAO() {
    connection = DBConnection.getConnection();
                                                                  Section Connection day. DBConnection.getConnection()
public List<Appetizers> getApp
   ResultSet rs = connection.
   List<Appetizers> desserts
                                                                                                                                               Query();
                         while(rs.next()) {
    desserts.add(populateAppetizers(rs.getInt(1), rs.getString(2), rs.getDouble(3)));
                         }
return desserts;
                  private Appetizers populateAppetizers(int id, String name, double price) {
    return new Appetizers(id, name, price);
                   public void createAppetizer(String aName, double dPrice) throws SQLException {
    PreparedStatement ps = connection.prepareStatement(CREATE_NEW_Appetizer_QUERY);
    ps.setString(1, aName);
    ps.setDouble(2, dPrice);
    ps.executeUpdate();
                   public void deleteAppetizer (int id) throws SQLException {
    PreparedStatement ps = connection.prepareStatement(DELETE_Appetizer_BY_ID);
    ps.executeUpdate();
    ps.executeUpdate();
                   }
```

#### Entrees

```
J Menu.java
                 AppetizersDA...
                                       IntreesDAO.java ⋈ In Application....
                                                                                 DBConnection...
                                                                                                        JE
 1 package dao;
 30 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;
 10 import entity.Entrees;
 11
12 public class EntreesDAO {
        private Connection connection;
        private final String DISPLAY_Entrees_QUERY = "SELECT * FROM Entrees";
 14
 15
        private final String CREATE_NEW_Entree_QUERY ="INSERT INTO Entrees(dishName, Price) VALUES(?,?)";
        private final String DELETE_Entree_BY_ID = "DELETE FROM Entrees WHERE ID = ?";
 16
 17
            public EntreesDAO() {
 180
 19
                 connection = DBConnection.getConnection();
            }
 20
 21
 220
            public List<Entrees> getEntrees() throws SQLException {
 23
                 ResultSet rs = connection.prepareStatement(DISPLAY Entrees QUERY).executeQuery();
 24
                List<Entrees> desserts = new ArrayList<Entrees>();
                 while(rs.next()) {
 26
                     desserts.add(populateEntrees(rs.getInt(1), rs.getString(2), rs.getDouble(3)));
 28
                 return desserts;
 38
 31
        }
 32
 33@
             private Entrees populateEntrees(int id, String name, double price) {
 34
                     return new Entrees(id, name, price);
 35
             public void createEntree(String eName, double ePrice) throws SQLException {
 36⊖
 37
                PreparedStatement ps = connection.prepareStatement(CREATE_NEW_Entree_QUERY);
 38
                 ps.setString(1, eName);
 39
                ps.setDouble(2, ePrice);
 40
                ps.executeUpdate();
 41
 42
 43⊕
            public void deleteEntree (int id) throws SQLException {
 44
                PreparedStatement ps = connection.prepareStatement(DELETE_Entree_BY_ID);
 45
                ps.setInt(1, id);
 46
                ps.executeUpdate();
 47
 48
            }
49
50 }
```

#### **Desserts**

```
AppetizersDA...
J) Menu.java

    DessertsDAO.... 
    □ EntreesDAO.java

                                                                                         Application....
                                                                                                               J Entrees java
  1 package dao;
 3@ import java.sql.Connection;
 4 import java.sql.PreparedStatement;
 5 import java.sql.ResultSet;
    import java.sql.SQLException;
     import java.util.ArrayList;
 8 import java.util.List;
 10 import entity.Desserts;
 12 public class DessertsDAO {
 13
         private Connection connection;
         private final String DISPLAY_DESSERTS_QUERY = "SELECT * FROM Desserts";
         private final String CREATE_NEW_Dessert_QUERY ="INSERT INTO Desserts(dishName,Price) VALUES(?,?)";
 15
         private final String DELETE_Dessert_BY_ID = "DELETE FROM Desserts WHERE ID = ?";
 16
 17
 188
              public DessertsDAO() {
                  connection = DBConnection.getConnection();
 19
 20
 21
             public List<Desserts> getDesserts() throws SQLException {
 220
                 ResultSet rs = connection.prepareStatement(DISPLAY_DESSERTS_QUERY).executeQuery();
List<Desserts> desserts = new ArrayList<Des
 23
24
                                                                 String dao.DessertsDAO.DISPLAY_DESSERTS_QUERY
 25
 26
                  while(rs.next()) [
                      desserts.add(populateDesserts(rs.getInt
 28
 29
                  return desserts;
                                                                                                         Press 'F2' for focu
 30
 31
 32
 33⊕
             private Desserts populateDesserts(int id, String name, double price) {
 34
                      return new Desserts(id, name, price);
 35
              public void createDessert(String dName, double dprice) throws SQLException {
 360
                  PreparedStatement ps = connection.prepareStatement(CREATE_NEW_Dessert_QUERY);
 37
 38
                  ps.setString(1, dName);
 39
                  ps.setDouble(2, dprice);
 40
                  ps.executeUpdate();
 41
 42
             public void deleteDessert (int id) throws SQLException {
    PreparedStatement ps = connection.prepareStatement(DELETE_Dessert_BY_ID);
 43@
 44
 45
                  ps.setInt(1, id);
46
                  ps.executeUpdate();
             }
47
48
49 }
```

## **DBConnection**

```
1 package dao;
3⊕ import java.sql.Connection;
7 public class DBConnection {
      private final static String connectionString = "jdbc:mysql://localhost:3306/Food";
      private static DBConnection instance;
3
      private static Connection connection;
1
      private DBConnection(Connection connection) {
30
         this.connection = connection;
78
      public static Connection getConnection() {
33
         if (instance == null) {
             try {
3
                 connection = DriverManager.getConnection(connectionString, "root", "Black426!");
                 instance = new DBConnection(connection);
1
                System.out.println("Success!");
             } catch (SQLException e) {
3
                System.out.println("Error!");
                 e.printStackTrace();
         }
3
         return DBConnection.connection;
      }
3
1 }
2
```

# **Screenshots of Running Application:**

```
Application [Java Application] ( Application [Java Application] C:\e
 Please Select an Options: Please Select an Options:
 Application [Java Application] C:\e 1. Display Appetizer 2. Create Appetizers 3. Delete Appetizers 3. Delete Appetizers 4. Display Appetizers 4. Display Appetizers 5. Create Entrees 5. Create Entrees 6. Delete Entrees 6. Delete Entrees 7. Display Desserts 7. Display
                                                                                                                                                                                        7. Display Desserts
8. Create Desserts
9. Delete Desserts
  5. Create Entrees
                                                                                              Create Desserts
  6. Delete Entrees
                                                                                               9. Delete Desserts
  7. Display Desserts
   8. Create Desserts
                                                                                                                                                                                                Enter New Dessert name:
                                                                                                Enter New Entree name:
   9. Delete Desserts
                                                                                                                                                                                                 Cake
                                                                                                   steak
                                                                                                                                                                                                Enter it's price: $
                                                                                                  Enter it's price: $
   Enter New Appetizer name:
                                                                                                   9.99
                                                                                                  Please Select an Options: Please Select an Options:
   Fries
                                                                                                1. Display Appetizer
2. Create Appetizers
3. Delete Appetizers
4. Display Entrees
5. Create Entrees
6. Delete Entrees
6. Delete Entrees
7. Display Appetizer
7. Create Appetizer
8. Display Appetizer
8. Display Appetizer
9. Create Appetizer
9. Create Appetizer
9. Create Appetizer
9. Display Appetizer
9. Display
  Enter it's price: $
   1.99
 Please Select an Options:

1. Display Appetizer

2. Create Appetizers

3. Delete Appetizers

4. Display Entrees

5. Create Entrees

4. Display Entrees

6. Delete Entrees
  Please Select an Options:
                                                                                           6. Delete Entrees
6. Delete Entrees
7. Display Desserts
8. Create Desserts
9. Delete Desserts
9. Delete Desserts
9. Delete Desserts
  5. Create Entrees
8. Create Desserts
9. Delete Desserts
9. Delete Desserts
9. Delete Desserts
1. Display Appetizer
1. Display Appetizer
2. Create Appetizers
3. Delete Appetizers
4. Display Entrees
5. Create Entrees
6. Delete Entrees
9. Delete Desserts
Please Select an Options:
1. Display Appetizer
2. Create Appetizers
3. Delete Appetizers
4. Display Entrees
5. Create Entrees
6. Delete Entrees
6. Delete Entrees
                                                                                          5. Create Entrees
6. Delete Entrees
7. Display Desserts
8. Create Desserts
9. Delete Desserts
                                                                                                                                                                                    6. Delete Entrees
7. Display Desserts
8. Create Desserts
   5. Create Entrees
   6. Delete Entrees
                                                                                                                                                                                               9. Delete Desserts
   7. Display Desserts
   8. Create Desserts
                                                                                                                                                                                                 1: Cake: 7.99
   9. Delete Desserts
                                                                                                  1: steak: 9.99
                                                                                                  1: steak: 9.99
Please Select an Options:
1. Display Appetizer
 3: Fries: 1.99
Please Select an Options:
1. Display Appetizer
2. Create Appetizers
3. Delete Appetizers
3. Delete Appetizers
4. Display Entrees
5. Create Entrees
6. Delete Entrees
7. Display Appetizer
2. Create Appetizers
3. Delete Appetizers
4. Display Entrees
5. Create Entrees
6. Delete Entrees
7. Display Descents
  3: Fries: 1.99
                                                                                               Display Desserts
                                                                                                                                                                                                8. Create Desserts
  6. Delete Entrees
                                                                                              8. Create Desserts
  Display Desserts
                                                                                                                                                                                                 9. Delete Desserts
                                                                                                  9. Delete Desserts
   8. Create Desserts
   9. Delete Desserts
                                                                                                                                                                                                 Enter dessert id to delete:
                                                                                                   Enter Entree id to delete:
 Enter Appears

Please Select an Options:

Display Appetizer

Appetizers
  Enter Appetizer id to delete: 1
                                                                                                   Please Select an Options: Please Select an Options: 1. Display Appetizer
                                                                                                                                                                                                   Please Select an Options:
                                                                                                   1. Display Appetizer 2. Create Appetizers
                                                                                                   2. Create Appetizers
                                                                                                                                                                                              Delete Appetizers
                                                                                                   3. Delete Appetizers
                                                                                                                                                                                               4. Display Entrees
5. Create Entrees
6. Delete Entrees
  3. Delete Appetizers
                                                                                                  4. Display Entrees
                                                                                              5. Create Entrees

    Display Entrees

   5. Create Entrees
                                                                           7. Display Desserts
8. Create Dessert
                                                                                              Delete Entrees
                                                                                                                                                                                                7. Display Desserts
   6. Delete Entrees
                                                                                                                                                                                               8. Create Desserts
  Display Desserts
                                                                                                  8. Create Desserts
                                                                                                                                                                                                 9 Delete Desserts
  8. Create Desserts
                                                                                            9. Delete Desserts
```

# URL to GitHub Repository:

https://github.com/marlon214/SQLWeek6/