

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:

```
App.java x Menu.java DBConnection.java Users_dao.java Weights_dao.java User.java Weight.java
1 package application;
2
3 public class App {
4     public static void main(String[] args) {
5         Menu menu = new Menu();
6         menu.start();
7     }
8 }
9
```

```
App.java Menu.java x DBConnection.java Users_dao.java Weights_dao.java User.java Weight.java
1 package application;
2
3 import java.sql.SQLException;
4
5
6
7
8
9
10
11 public class Menu {
12
13     private Scanner scanner = new Scanner(System.in);
14     private String[] menuOpts = {"Enter name", "Enter weight and date", "Update weight", "Delete user"};
15     private Users_dao usersDao = new Users_dao();
16     private Weights_dao weightsDao = new Weights_dao();
17
18     private void printMenu() {
19         System.out.println("-----");
20         for (int i=0; i < menuOpts.length; i++) {
21             System.out.println((i+1) + " " + menuOpts[i]);
22         }
23     }
24
25     private void enterName() throws SQLException{
26         System.out.println("Enter Name");
27         String name = scanner.nextLine();
28         usersDao.enterNewName (name);
29     }
30
31     private void enterWeight() throws SQLException{
32         String measureDate = "";
33         System.out.println("User ID: ");
34         int userId = scanner.nextInt();
35         do {
36             System.out.print("Enter Weigh In Date");
37             measureDate= scanner.nextLine();
38             if(measureDate.isEmpty()) {
39                 System.out.println("\nWeigh in date cannot be empty. Please try again...");
40             }
41         }while(measureDate.isEmpty());
42
43         System.out.println("Enter Weight: ");
44         String weight = scanner.nextLine();
45
46
```

```
App.java  Menu.java X DBConnection.java  Users_dao.java  Weights_dao.java  User.java  Weight.java
46
47     weightsDao.enterNewWeight (userId, measureDate, weight);
48 }
49
50 private void updateWeight()throws SQLException {
51     System.out.println("Enter ID of user to update log: ");
52     String nl = scanner.nextLine() ;
53     Integer id = null;
54     try {
55         id = Integer.parseInt(nl);
56     }catch (NumberFormatException e) {
57         System.out.println("Please enter a number!");
58         return;
59     }
60     if (id != null) {
61         System.out.println("Enter weight");
62         String weight = scanner.nextLine();
63         if(!weight.isEmpty()) {
64             weightsDao.updateWeight (id,weight);
65         }
66     }
67 }
68
69 private void deleteUser() throws SQLException{
70     System.out.println("Enter ID to delete: ");
71     int id = Integer.parseInt(scanner.nextLine());
72     usersDao.deleteUserById(id);
73 }
74
```

```

App.java Menu.java x DBConnection.java Users_dao.java Weights_dao.java User.java Weight.java
73     }
74
75     public void start() {
76         String selection = "";
77
78         do {
79             printMenu();
80             selection = scanner.nextLine();
81
82             try {
83                 switch(selection) {
84                     case "1": enterName();
85                             break;
86                     case "2": enterWeight();
87                             break;
88                     case "3": updateWeight();
89                             break;
90                     case "4": deleteUser();
91                             break;
92                 }
93             }
94             catch (SQLException e) {
95                 e.printStackTrace();
96                 end();
97             }
98             if (!selection.equals("-1")) {
99                 System.out.println("Press enter to continue...");
100                 scanner.nextLine();
101             }
102
103         } while (!selection.equals("-1"));
104
105         end();
106     }
107     public void end() {
108         System.out.println("Goodbye!");
109         scanner.close();
110         usersDao.close();
111         weightsDao.close();
112     }
113 }
114

```

```
App.java Menu.java DBConnection.java X Users_dao.java Weights_dao.java User.java Weight.java
1 package dao;
2
3 import java.sql.Connection;
4
5
6
7 public final class DBConnection {
8
9     private final static String URL = "jdbc:mysql://localhost:3306/weightTracker";
10    private final static String USERNAME = "root";
11    private final static String PASSWORD = "Walleye#2021";
12
13    private static DBConnection instance = new DBConnection();
14    private static Connection connection;
15
16    private DBConnection() {}
17
18    public static DBConnection getInstance() {
19        return instance;
20    }
21
22    public Connection getConnection() {
23        if( connection == null ) {
24            try {
25                connection = DriverManager.getConnection(URL, USERNAME, PASSWORD );
26                System.out.println("Connection Successful!");
27            } catch (SQLException e) {
28                e.printStackTrace();
29            }
30        }
31        return connection;
32    }
33
34    public void closeConnection() {
35        try {
36            connection.close();
37        } catch (Exception e) {
38            e.printStackTrace();
39        } finally {
40            connection = null;
41        }
42    }
43
44 }
```

```
App.java Menu.java DBConnection.java Users_dao.java X Weights_dao.java User.java Weight.java
1 package dao;
2
3 import java.sql.Connection;
4
5
6
7
8
9
10 public class Users_dao {
11     private Connection connection;
12     private final String ADD_NAME = "INSERT INTO users (name) VALUES (?)";
13     private final String DELETE_USER = "DELETE from users WHERE id = ?";
14
15     public Users_dao() {
16         connection = DBConnection.getInstance().getConnection();
17     }
18
19     public void enterNewName(String name) throws SQLException{
20         PreparedStatement ps = connection.prepareStatement(ADD_NAME);
21         ps.setString(1, name);
22         ps.executeUpdate();
23     }
24
25     public void deleteUserById(int id) throws SQLException{
26         PreparedStatement ps = connection.prepareStatement(DELETE_USER);
27         ps.setInt(1, id);
28         ps.executeUpdate();
29     }
30     public void close() {
31         DBConnection.getInstance().closeConnection();
32     }
33
34 }
35
```

```
App.java  Menu.java  DBConnection.java  Users_dao.java  Weights_dao.java x  User.java  Weight.java

1 package dao;
2
3 import java.sql.Connection;
4
5
6
7 public class Weights_dao {
8     private Connection connection;
9     private final String ADD_WEIGHT = "INSERT INTO weights (user_id, measureDate, weight) VALUES (?, ?, ?)";
10    private final String UPDATE_ENTRY = "UPDATE weights SET weight=? WHERE id=?";
11
12    public Weights_dao() {
13        connection = DBConnection.getInstance().getConnection();
14    }
15
16    public void enterNewWeight(int userId, String measureDate, String weight) throws SQLException{
17        PreparedStatement ps = connection.prepareStatement(ADD_WEIGHT);
18        ps.setInt(1, userId);
19        ps.setString(2, measureDate);
20        ps.setString(3, weight);
21        ps.executeUpdate();
22    }
23
24    public void updateWeight(int id, String weight) throws SQLException{
25        PreparedStatement ps = connection.prepareStatement(UPDATE_ENTRY);
26        ps.setInt(1, id);
27        ps.setString(2, weight);
28        ps.executeUpdate();
29    }
30
31    public void close() {
32        DBConnection.getInstance().closeConnection();
33    }
34
35
36
37 }
38
```

```
App.java  Menu.java  DBConnection.java  Users_dao.java  Weights_dao.java  User.java  Weight.java
1 package entity;
2
3 public class User {
4     private int id;
5     private String name;
6
7     public User (int id, String name) {
8         this.setid (id);
9         this.setName (name);
10    }
11
12    public int getid() {
13        return id;
14    }
15
16    public void setid(int id) {
17        this.id = id;
18    }
19
20    public String getName() {
21        return name;
22    }
23
24    public void setName(String name) {
25        this.name = name;
26    }
27
28
29 }
```



```
App.java Menu.java DBConnection.java Users_dao.java Weights_dao.java User.java Weight.java x
1 package entity;
2
3 public class Weight {
4     private int id;
5     private int userId;
6     private String measureDate;
7     private String weight;
8
9     public Weight (int id, int userId, String measureDate, String weight) {
10         this.setWeightsId (id);
11         this.setUserId (userId);
12         this.setMeasureDate (measureDate);
13         this.setWeight (weight);
14     }
15
16
17     public int getWeightsId() {
18         return id;
19     }
20
21     public void setWeightsId(int id) {
22         this.id = id;
23     }
24
25     public int getUserId() {
26         return userId;
27     }
28
29     public void setUserId(int userId) {
30         this.userId = userId;
31     }
32
33     public String getMeasureDate() {
34         return measureDate;
35     }
36
37     public void setMeasureDate(String measureDate) {
38         this.measureDate = measureDate;
39     }
40
```

```
40
41     public String getWeight() {
42         return weight;
43     }
44
45     public void setWeight(String weight) {
46         this.weight = weight;
47     }
48
49 }
50
51
```

Screenshots of Running Application:

```
Connection Successful!
-----
1)Enter name
2)Enter weight and date
3)Update weight
4)Delete user
1
Enter Name
Jeff
Press enter to continue...
```

```
Connection Successful!
-----
1)Enter name
2)Enter weight and date
3)Update weight
4)Delete user
1
Enter Name
Jeff
Press enter to continue...

|-----
1)Enter name
2)Enter weight and date
3)Update weight
4)Delete user
```

Press enter to continue...

- 1)Enter name
- 2)Enter weight and date
- 3)Update weight
- 4)Delete user

2

User ID:

1

Enter Weigh In Date

Weigh in date cannot be empty. Please try again...

Enter Weigh In Date03/06/2021|

- 1)Enter name
- 2)Enter weight and date
- 3)Update weight
- 4)Delete user

2

User ID:

1

Enter Weigh In Date

Weigh in date cannot be empty. Please try again...

Enter Weigh In Date03/06/2021

Enter Weight:

250

Press enter to continue...

```
-----  
1)Enter name  
2)Enter weight and date  
3)Update weight  
4)Delete user  
3  
Enter ID of user to update log:  
1  
Enter weight  
320  
Press enter to continue...
```

```
-----  
1)Enter name  
2)Enter weight and date  
3)Update weight  
4)Delete user  
4  
Enter ID to delete:  
1  
Press enter to continue...
```

URL to GitHub Repository: <https://github.com/tramtran09/SQL-Week-6>