

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:

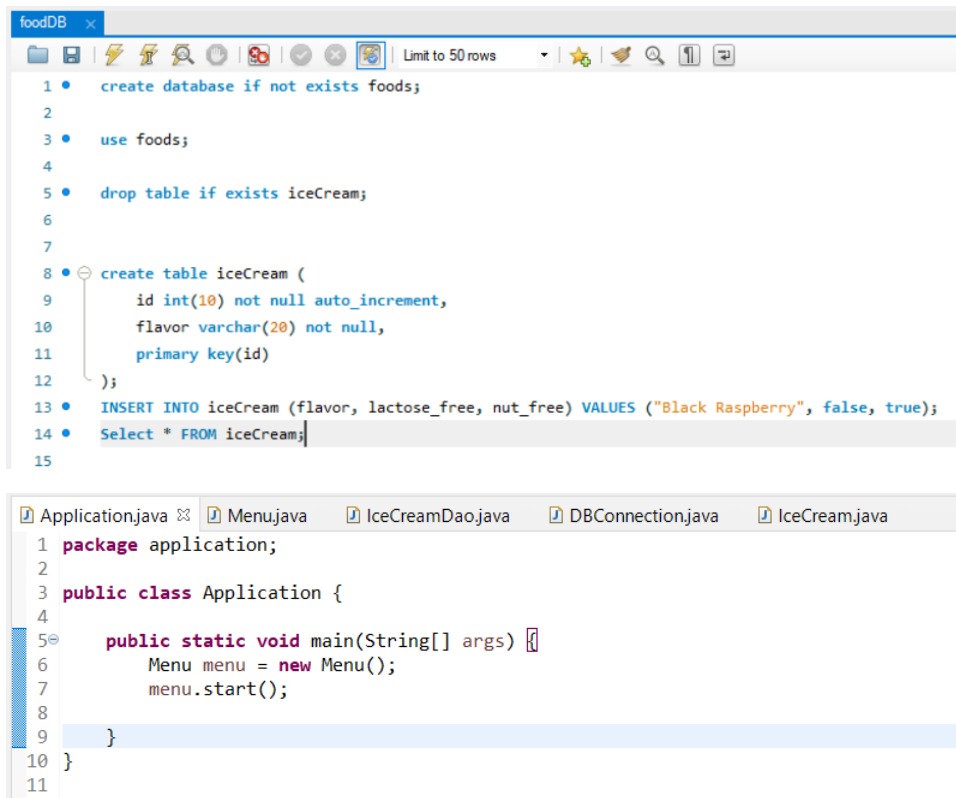
The application does not need to be as complex as the example in the video curriculum.

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 foods;
2
3 • use foods;
4
5 • drop table if exists iceCream;
6
7
8 • create table iceCream (
9     id int(10) not null auto_increment,
10    flavor varchar(20) not null,
11    primary key(id)
12 );
13 • INSERT INTO iceCream (flavor, lactose_free, nut_free) VALUES ("Black Raspberry", false, true);
14 • Select * FROM iceCream;
15
```

```
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
```

```

Application.java Menu.java IceCreamDao.java DBConnection.java IceCream.java
1 package application;
2
3 import java.sql.SQLException;
4
5 public class Menu {
6
7     private IceCreamDao iceCreamDao = new IceCreamDao();
8     private Scanner scanner = new Scanner(System.in);
9     private List<String> options = Arrays.asList("Display Flavors", "Display a single Flavor", "Create a new Flavor", "Delete a Flavor", "Update a Flavor");
10
11     public void start() {
12         String selection = "";
13
14         do {
15             printMenu();
16             selection = scanner.nextLine();
17
18             try {
19                 if (selection.equals("1")) {
20                     displayFlavors();
21                 } else if (selection.equals("2")) {
22                     displayFlavor();
23                 } else if (selection.equals("3")) {
24                     createFlavor();
25                 } else if (selection.equals("4")) {
26                     deleteFlavor();
27                 } else if (selection.equals("5")) {
28                     updateFlavor();
29                 }
30             } catch (SQLException e) {
31                 e.printStackTrace();
32             }
33
34             System.out.println("Press enter to continue...");
35             scanner.nextLine();
36         } while (!selection.equals("-1"));
37         System.out.println("Enjoy your ice cream ♥");
38     }
39 }

```

```

Application.java Menu.java IceCreamDao.java DBConnection.java IceCream.java
44 while (selection.equals("-1")),
45     System.out.println("Enjoy your ice cream ♥");
46 }
47
48
49 private void printMenu() {
50     System.out.println("Select an Option:\n-----");
51     for (int i = 0; i < options.size(); i++) {
52         System.out.println(" " + (i + 1) + " " + options.get(i));
53     }
54     System.out.println("-1) Abort Ice Cream Mission");
55 }
56
57 private void displayFlavors() throws SQLException {
58     List<IceCream> flavors = iceCreamDao.getIceCreams();
59     for (IceCream flavor : flavors) {
60         System.out.println(flavor.getId() + " : " + flavor.getFlavor());
61     }
62 }
63
64 private void displayFlavor() throws SQLException {
65     System.out.print("Enter flavor number: ");
66     int id = Integer.parseInt(scanner.nextLine());
67     IceCream flavor = iceCreamDao.getIceCreamById(id);
68     System.out.println(flavor.getFlavor());
69 }
70
71 private void createFlavor() throws SQLException {
72     System.out.print("Enter new flavor: ");
73     String newIceCream = scanner.nextLine();
74
75     iceCreamDao.createNewIceCream(newIceCream);
76     System.out.println("New Ice Cream added, enjoy ♥");
77 }
78
79 private void deleteFlavor() throws SQLException {
80     System.out.println("Enter Ice Cream number to remove: ");
81     int id = Integer.parseInt(scanner.nextLine());
82     iceCreamDao.deleteIceCreamById(id);
83 }

```

```

11     }
12 }
13
14 private void displayFlavor() throws SQLException {
15     System.out.print("Enter flavor number: ");
16     int id = Integer.parseInt(scanner.nextLine());
17     IceCream flavor = iceCreamDao.getIceCreamById(id);
18     System.out.println(flavor.getFlavor());
19 }
20
21 private void createFlavor() throws SQLException {
22     System.out.print("Enter new flavor: ");
23     String newIceCream = scanner.nextLine();
24
25     iceCreamDao.createNewIceCream(newIceCream);
26     System.out.println("New Ice Cream added, enjoy ♥");
27 }
28
29 private void deleteFlavor() throws SQLException {
30     System.out.println("Enter Ice Cream number to remove: ");
31     int id = Integer.parseInt(scanner.nextLine());
32     iceCreamDao.deleteIceCreamById(id);
33     System.out.println("Poof: successfully removed");
34
35 }
36
37 private void updateFlavor() throws SQLException {
38     System.out.println("Enter Ice Cream number to update: ");
39     int id = Integer.parseInt(scanner.nextLine());
40     IceCream flavor = iceCreamDao.getIceCreamById(id);
41     System.out.println(flavor.getFlavor());
42     System.out.print("Which will now change to: ");
43     String newIceCream = scanner.nextLine();
44     iceCreamDao.updateIceCreamById(id, newIceCream);
45     System.out.println("Ice Cream has been updated");
46
47 }
48 }
49 }

```

```

Application.java  Menu.java  IceCreamDao.java  DBConnection.java  IceCream.java
1 package dao;
2
3 import java.sql.Connection;
4
11 public class IceCreamDao {
12
13     private static final String CREATE_NEW_ICECREAM_QUERY = "INSERT INTO iceCream(flavor) VALUES(?)";
14     private static final String DELETE_ICECREAM_BY_ID_QUERY = "DELETE FROM iceCream WHERE id = ?";
15     private static final String UPDATE_ICECREAM_BY_ID_QUERY = "UPDATE iceCream SET flavor = ? WHERE id = ?";
16     private Connection connection;
17     // private FlavorsDao flavorsDao;
18     private final String GET_ICECREAM_QUERY = "SELECT * FROM iceCream";
19     private final String GET_ICECREAM_BY_ID_QUERY = "SELECT * FROM iceCream WHERE id = ?";
20     // private final String CREATE_NEW_FLAVORS_QUERY = "INSERT INTO flavors(name) VALUES(?)";
21     // private final String DELETE_FLAVORS_BY_ID_QUERY = "DELETE FROM flavors WHERE id = ?";
22
23     public IceCreamDao() {
24         connection = DBConnection.getConnection();
25     }
26
27     public List<IceCream> getIceCreams() throws SQLException {
28         ResultSet rs = connection.prepareStatement(GET_ICECREAM_QUERY).executeQuery();
29         List<IceCream> iceCream = new ArrayList<IceCream>();
30
31         while (rs.next()) {
32             iceCream.add(populateIceCream(rs.getInt(1), rs.getString(2)));
33         }
34
35         return iceCream;
36     }
37
38     public IceCream getIceCreamById(int id) throws SQLException {
39         PreparedStatement ps = connection.prepareStatement(GET_ICECREAM_BY_ID_QUERY);
40         ps.setInt(1, id);
41         ResultSet rs = ps.executeQuery();
42         rs.next();
43         return populateIceCream(rs.getInt(1), rs.getString(2));
44     }
45 }

```

```

45     }
46
47
48     public void createNewIceCream(String flavor) throws SQLException {
49         PreparedStatement ps = connection.prepareStatement(CREATE_NEW_ICECREAM_QUERY);
50         ps.setString(1, flavor);
51         ps.executeUpdate();
52     }
53
54     public void deleteIceCreamById(int id) throws SQLException {
55         PreparedStatement ps = connection.prepareStatement(DELETE_ICECREAM_BY_ID_QUERY);
56         ps.setInt(1, id);
57         ps.executeUpdate();
58     }
59
60     //update
61     public void updateIceCreamById(int id, String newIceCream) throws SQLException {
62         PreparedStatement ps = connection.prepareStatement(UPDATE_ICECREAM_BY_ID_QUERY);
63         ps.setInt(2, id);
64         ps.setString(1, newIceCream);
65         ps.executeUpdate();
66     }
67
68     private IceCream populateIceCream(int id, String name) throws SQLException {
69         return new IceCream(id, name);
70     }
71
72     public List<IceCream> getFlavors() {
73         // TODO Auto-generated method stub
74         return null;
75     }
76 }
77
78

```

Application.java Menu.java IceCreamDao.java DBConnection.java IceCream.java

```

1 package dao;
2
3 import java.sql.Connection;
4
5
6 public class DBConnection {
7
8
9     private final static String URL = "jdbc:mysql://localhost:3306/foods";
10    private final static String USERNAME = "root";
11    private final static String PASSWORD = "password";
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

```



```
Application.java Menu.java IceCreamDao.java DBConnection.java IceCream.java
1 package entities;
2
3 public class IceCream {
4
5     private int id;
6     private String flavor;
7
8     public IceCream(int id, String flavor) {
9         this.setId(id);
10        this.setFlavor(flavor);
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 getFlavor() {
23        return flavor;
24    }
25
26    public void setFlavor(String flavor) {
27        this.flavor = flavor;
28    }
29
30
31 }
32
```

Screenshots of Running Application:

```
Console ⓘ
Application (4) [Java Application] C:\Users\trepv\Downloads\ec
Connection successful
Select an Option:
-----
1) Display Flavors
2) Display a single Flavor
3) Create a new Flavor
4) Delete a Flavor
5) Update a Flavor
-1) Abort Ice Cream Mission
1
1: Black Raspberry
3: Mint Chocolate Chip
4: Monkeybread
Press enter to continue...

Select an Option:
-----
1) Display Flavors
2) Display a single Flavor
3) Create a new Flavor
4) Delete a Flavor
5) Update a Flavor
-1) Abort Ice Cream Mission
2
Enter flavor number: 4
Monkeybread
Press enter to continue...

Select an Option:
-----
1) Display Flavors
2) Display a single Flavor
3) Create a new Flavor
4) Delete a Flavor
5) Update a Flavor
-1) Abort Ice Cream Mission
```

```
Console
Application (4) [Java Application] C:\Users\trepv\Do
3) Create a new Flavor
4) Delete a Flavor
5) Update a Flavor
-1) Abort Ice Cream Mission
2
Enter flavor number: 4
Monkeybread
Press enter to continue...

Select an Option:
-----
1) Display Flavors
2) Display a single Flavor
3) Create a new Flavor
4) Delete a Flavor
5) Update a Flavor
-1) Abort Ice Cream Mission
3
Enter new flavor: Vanilla
New Ice Cream added, enjoy ♥
Press enter to continue...

Select an Option:
-----
1) Display Flavors
2) Display a single Flavor
3) Create a new Flavor
4) Delete a Flavor
5) Update a Flavor
-1) Abort Ice Cream Mission
1
1: Black Raspberry
3: Mint Chocolate Chip
4: Monkeybread
6: Vanilla
Press enter to continue...
<
```



```
Application (4) [Java Application] C:\Users\trepv\Download
5) Update a Flavor
-1) Abort Ice Cream Mission
1
1: Black Raspberry
3: Mint Chocolate Chip
4: Monkeybread
6: Vanilla
Press enter to continue...
4
Select an Option:
-----
1) Display Flavors
2) Display a single Flavor
3) Create a new Flavor
4) Delete a Flavor
5) Update a Flavor
-1) Abort Ice Cream Mission
4
Enter Ice Cream number to remove:
4
Poof: successfully removed
Press enter to continue...

Select an Option:
-----
1) Display Flavors
2) Display a single Flavor
3) Create a new Flavor
4) Delete a Flavor
5) Update a Flavor
-1) Abort Ice Cream Mission
1
1: Black Raspberry
3: Mint Chocolate Chip
6: Vanilla
Press enter to continue...
<
```

Application (4) [Java Application] C:\Users\trepv\Downloa

-1) Abort Ice Cream Mission

1

1: Black Raspberry

3: Mint Chocolate Chip

6: Vanilla

Press enter to continue...

5

Select an Option:

1) Display Flavors

2) Display a single Flavor

3) Create a new Flavor

4) Delete a Flavor

5) Update a Flavor

-1) Abort Ice Cream Mission

5

Enter Ice Cream number to update:

6

Vanilla

Which will now change to: Neopolitan

Ice Cream has been updated

Press enter to continue...

Select an Option:

1) Display Flavors

2) Display a single Flavor

3) Create a new Flavor

4) Delete a Flavor

5) Update a Flavor

-1) Abort Ice Cream Mission

1

1: Black Raspberry

3: Mint Chocolate Chip

6: Neopolitan

Press enter to continue...

URL to GitHub Repository:

<https://github.com/trepvox/Mysql-week-5-coding-assignment>