

Relational Databases with MySQL Week 10 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:

Week 10 - Week_10_HW_Assignment/src/entity/Food.java - Eclipse IDE

Edit Source Refactor Navigate Search Project Run Window Help

Application [Java Application] C:\Program Files\Java\jdk-11.0.14\bin\javaw.exe (Jul 8, 2022, 5:43:35 PM) [pid: 37192]
Connection successful.

Package Explorer: Lecture, VideoExample, src, (default package), application, application, application, application, Menujava, dao, DBConnect, MemberDao, TeamDao, entity, Memberjava, Teamjava, JRE System Library, Referenced Library, teamDB.sql, Week_10_HW_Assign, JRE System Library, src, application, application, application, application, Menujava, dao, DBConnect, MemberDao, TeamDao, entity, Memberjava, Teamjava, JRE System Library, Referenced Library, foodDB.sql

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

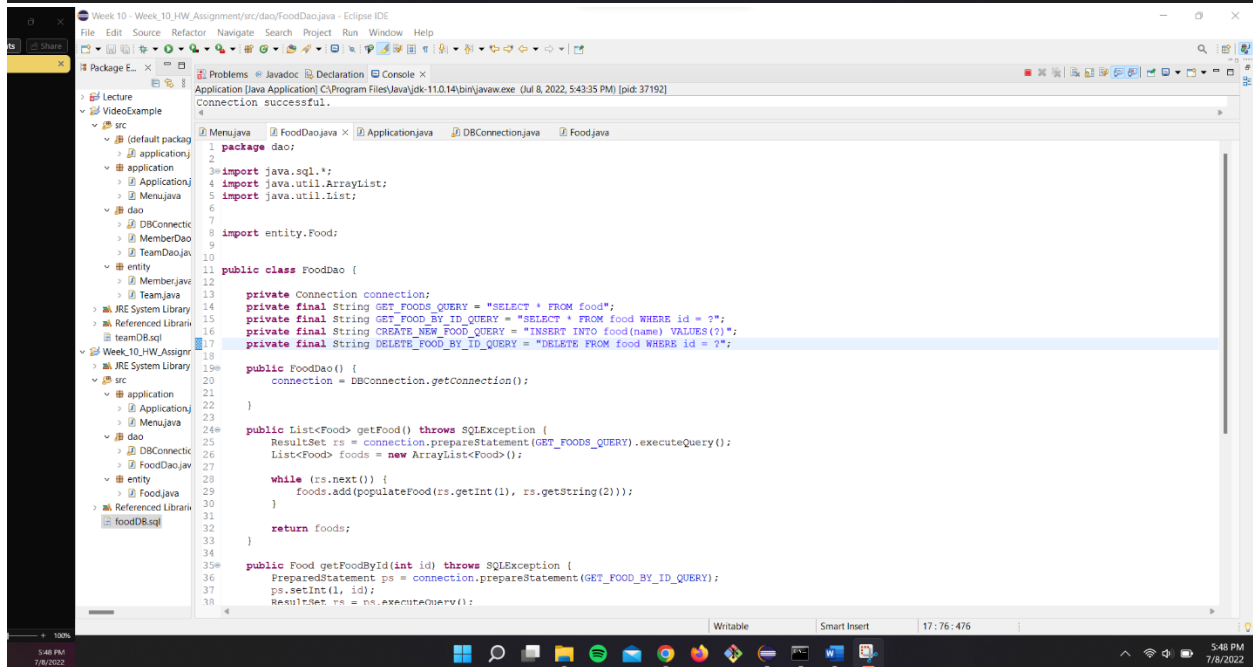
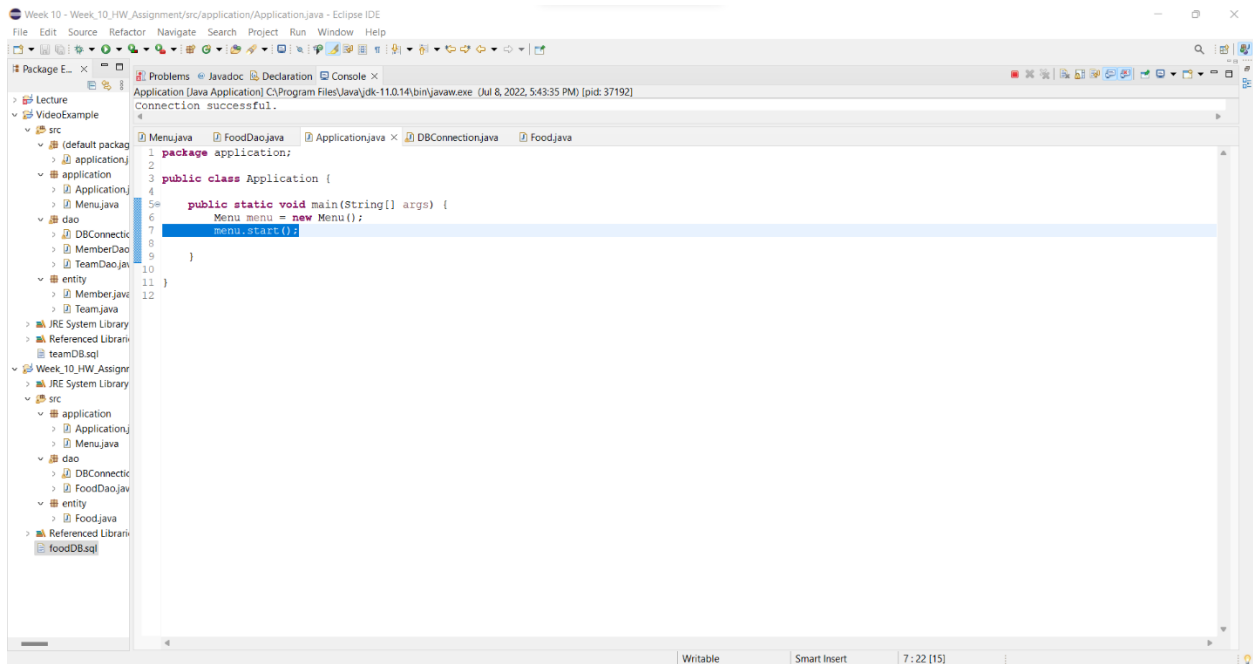
Edit Source Refactor Navigate Search Project Run Window Help

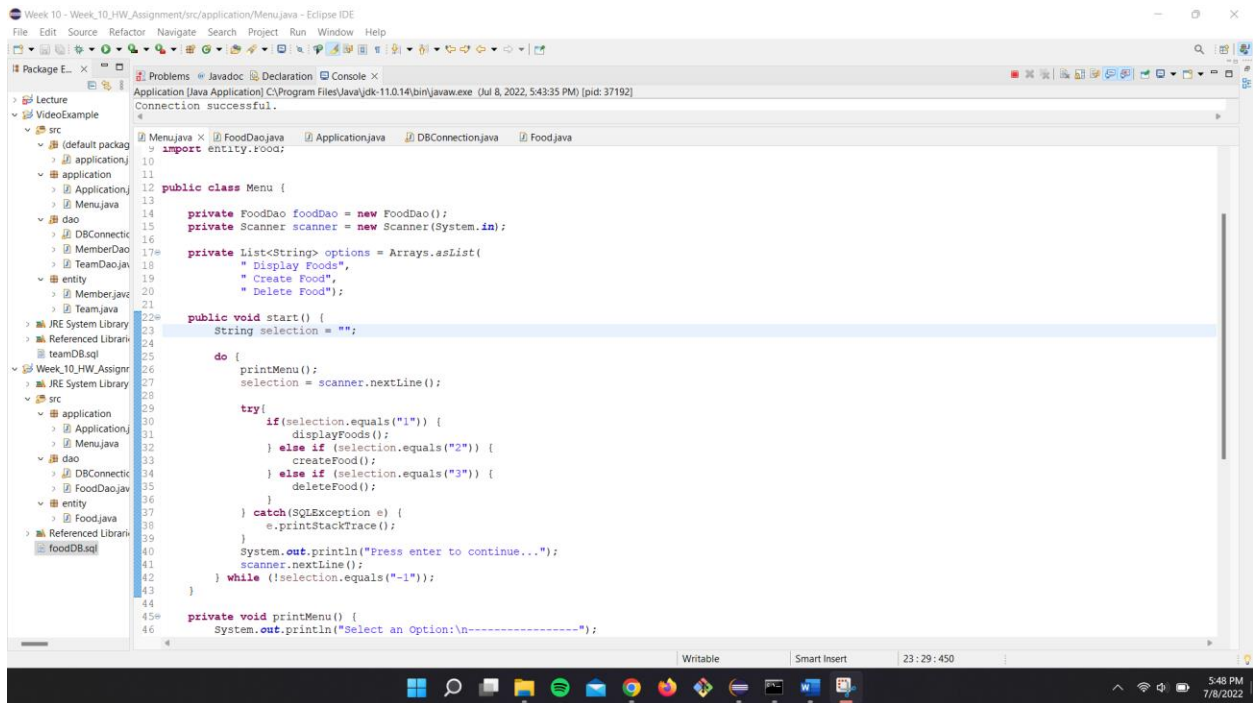
Application [Java Application] C:\Program Files\Java\jdk-11.0.14\bin\javaw.exe (Jul 8, 2022, 5:43:35 PM) [pid: 37192]
Connection successful.

Package Explorer: Lecture, VideoExample, src, (default package), application, application, application, application, Menujava, dao, DBConnect, MemberDao, TeamDao, entity, Memberjava, Teamjava, JRE System Library, Referenced Library, teamDB.sql, Week_10_HW_Assign, JRE System Library, src, application, application, application, application, Menujava, dao, DBConnect, MemberDao, TeamDao, entity, Memberjava, Teamjava, JRE System Library, Referenced Library, foodDB.sql

```
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/food";
10    private final static String USERNAME = "root";
11    private final static String PASSWORD = "Acmlan#1";
12
13    private static Connection connection;
14    private static DBConnection instance;
15
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("Connection successful.");
26            } catch (SQLException e) {
27                e.printStackTrace();
28            }
29        }
30        return DBConnection.connection;
31    }
32 }
33
34
```

Writable Smart Insert 12:5:318





Screenshots of Running Application:

The screenshot shows an IDE with a console window at the top and a source code editor at the bottom. The console window displays the output of a Java application, including menu prompts, user input, and database connection status. The source code editor shows the implementation of a `Menu` class, which includes an import statement for `entity.Food` and a private field `foodDao` initialized with a new `FoodDao` object.

```
Application [Java Application] C:\Program Files\Java\jdk-11.0.14\bin\javaw.exe (Jul 8, 2022, 5:43:35 PM) [pid: 37192]
Connection successful.
Select an Option:
-----
1 Display Foods
2 Create Food
3 Delete Food
1
1: fries
2: chocolate
Press enter to continue...
Select an Option:
-----
1 Display Foods
2 Create Food
3 Delete Food
2
Enter new food name: nothing
Press enter to continue...
Select an Option:
-----
1 Display Foods
2 Create Food
3 Delete Food
1
1: fries
2: chocolate
4: nothing
Press enter to continue...
Select an Option:
-----
1 Display Foods
2 Create Food
3 Delete Food
3
Enter food id to delete: 3
Press enter to continue...
Select an Option:
-----
1 Display Foods
2 Create Food
3 Delete Food
1
1: fries
2: chocolate
4: nothing
Press enter to continue...
Select an Option:
-----
1 Display Foods
2 Create Food
3 Delete Food
3
Enter food id to delete: 4
Press enter to continue...
Select an Option:
-----
1 Display Foods
2 Create Food
3 Delete Food
1
1: fries
2: chocolate
Press enter to continue...
```

```
Menu.java x FoodDao.java Application.java DBConnection.java Food.java
9 import entity.Food;
10
11
12 public class Menu {
13
14     private FoodDao foodDao = new FoodDao();
15 }
```

URL to GitHub Repository: