

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.

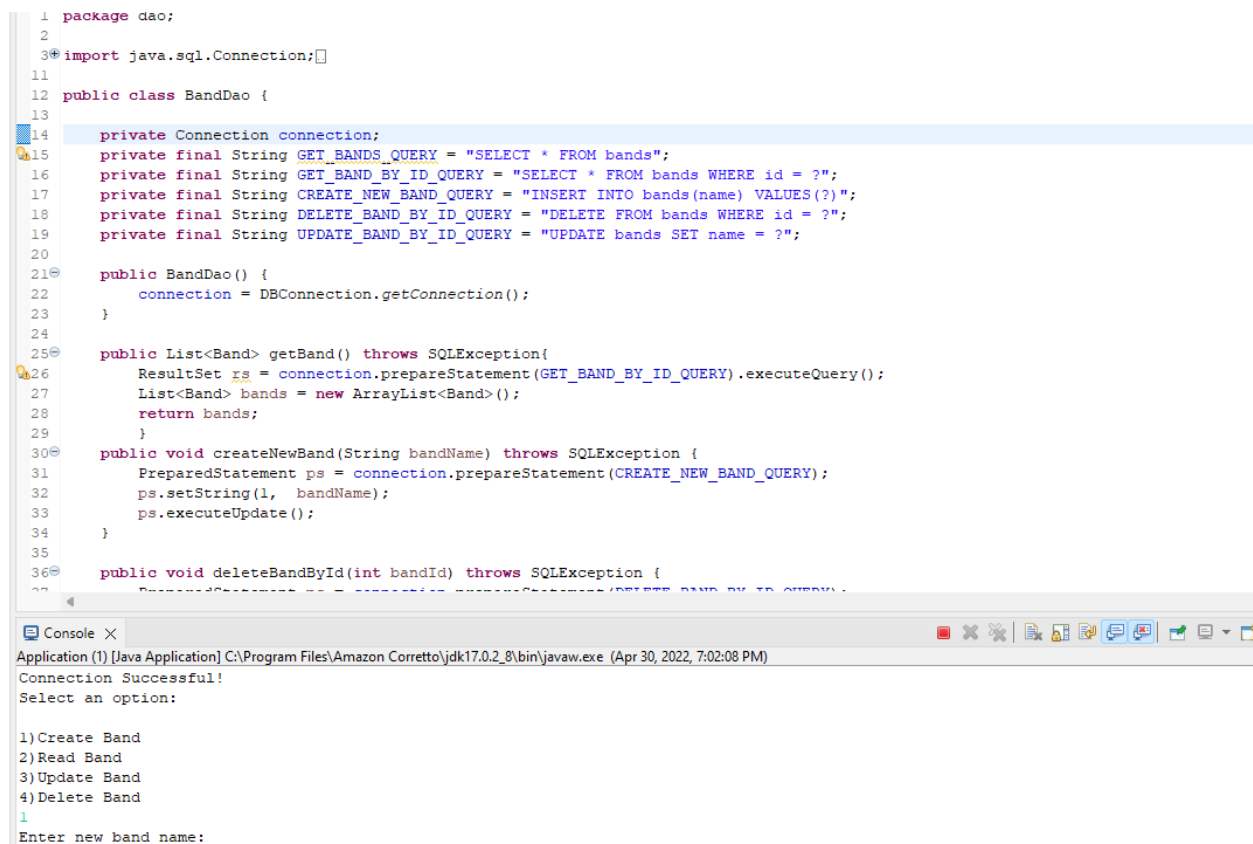
You need an option for each of the CRUD operations (Create, Read, Update, and Delete).

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:

Screenshots of Running Application:



```
1 package dao;
2
3 import java.sql.Connection;
4
11
12 public class BandDao {
13
14     private Connection connection;
15     private final String GET_BANDS_QUERY = "SELECT * FROM bands";
16     private final String GET_BAND_BY_ID_QUERY = "SELECT * FROM bands WHERE id = ?";
17     private final String CREATE_NEW_BAND_QUERY = "INSERT INTO bands(name) VALUES(?)";
18     private final String DELETE_BAND_BY_ID_QUERY = "DELETE FROM bands WHERE id = ?";
19     private final String UPDATE_BAND_BY_ID_QUERY = "UPDATE bands SET name = ?";
20
21     public BandDao() {
22         connection = DBConnection.getConnection();
23     }
24
25     public List<Band> getBand() throws SQLException {
26         ResultSet rs = connection.prepareStatement(GET_BAND_BY_ID_QUERY).executeQuery();
27         List<Band> bands = new ArrayList<Band>();
28         return bands;
29     }
30
31     public void createNewBand(String bandName) throws SQLException {
32         PreparedStatement ps = connection.prepareStatement(CREATE_NEW_BAND_QUERY);
33         ps.setString(1, bandName);
34         ps.executeUpdate();
35     }
36
37     public void deleteBandById(int bandId) throws SQLException {
38         PreparedStatement ps = connection.prepareStatement(DELETE_BAND_BY_ID_QUERY);
39         ps.setInt(1, bandId);
40         ps.executeUpdate();
41     }
42 }
```

Console X

Application (1) [Java Application] C:\Program Files\Amazon Corretto\jdk17.0.2_8\bin\javaw.exe (Apr 30, 2022, 7:02:08 PM)

Connection Successful!

Select an option:

- 1) Create Band
- 2) Read Band
- 3) Update Band
- 4) Delete Band

1

Enter new band name:

URL to GitHub Repository:

<https://github.com/mttsgr/Week10-Assignment>