Lab-8

Implement JDBC Concept in your domain.

Minimum two tables based on your domain you need to create it.

Minimum 10 records add it in your table.

CRUD Operation preform in your domain based example.

```
import java.sql.*;

import java.util.ArrayList;

import java.util.Arrays;



public class App {

private Statement statement;



public void createTable(Connection con) {

String myTableName = "CREATE TABLE discordBotDetail ("

+ " id int(11),"

+ "botName VARCHAR(10),"

+ "date DATE,"

+ "count INT(64))";



try {

statement = con.createStatement();

// The next line has the issue

statement.executeUpdate(myTableName);
```

```
System.out.println("Table Created");

} catch (SQLException e) {

System.out.println("An error has occurred on Table Creation");

}



}



public void insertRecords(Connection con) {

// ArrayList<String> id = new ArrayList<>(Arrays.asList("1", "2", "3",
"4", "5",

// "6", "7", "8", "9", "10"));

try {

PreparedStatement preparedStatement = con

.prepareStatement("INSERT INTO discordBotDetail (id, botName, date,count)
VALUES (?, ?, ?, ?)");

java.sql.Date currentDate = new java.sql.Date(System.currentTimeMillis());

preparedStatement.setString(1, "12");

preparedStatement.setString(2, "i-stay");

preparedStatement.setDate(3, currentDate);

preparedStatement.setString(4, "16");

preparedStatement.executeUpdate();


} catch (SQLException e) {

System.out.println("An error has occurred on record Creation");

}

}
```

```
public void readRecords(Connection con) {

try {

PreparedStatement insert = con.prepareStatement("SELECT * FROM
discordBotDetail");

ResultSet res = insert.executeQuery();

while (res.next()) {

int id = res.getInt("id");

String botName = res.getString("botName");

java.sql.Date date = res.getDate("date");

int count = res.getInt("count");

System.out.println("ID: " + id + ", BotName: " + botName + ", Date: " +
date + ", Count: " + count);

}



res.close();

insert.close();

} catch (Exception e) {

System.out.println("An error has occurred on record Creation");

}

}



public void updateRecords(Connection con, String botName, int newCount) {

try {

PreparedStatement preparedStatement = con.prepareStatement(

"UPDATE discordBotDetail SET count = ? WHERE botName = ?");

preparedStatement.setInt(1, newCount);
```

```
preparedStatement.setString(2, botName);
preparedStatement.executeUpdate();
System.out.println("Record updated successfully");
} catch (SQLException e) {
System.out.println("An error occurred during the update operation");
e.printStackTrace();
}
}

public void deleteRecords(Connection con, String botName) {
try {
PreparedStatement preparedStatement = con.prepareStatement(
"DELETE FROM discordBotDetail WHERE botName = ?");
preparedStatement.setString(1, botName);
preparedStatement.executeUpdate();
System.out.println("Record deleted successfully");
} catch (SQLException e) {
System.out.println("An error occurred during the delete operation");
e.printStackTrace();
}
}

public Connection createConnection() {
String host = "jdbc:mariadb://localhost:3306/discord";
```

```
String username = "root";

String password = "root";

Connection con = null;

try {

con = DriverManager.getConnection(host, username, password);

System.out.println("Connected to MySQL database");

} catch (SQLException e) {

System.out.println("Failed to connect to MySQL database");

e.printStackTrace();

}

return con;



}



public static void main(String[] args) {



App db = new App();

Connection con = db.createConnection();

App t1 = new App();

// t1.createTable(con);

t1.insertRecords(con);

t1.readRecords(con);

t1.updateRecords(con, "i-stay", 20);

t1.deleteRecords(con, "i-stay");

}
```

```
}
```