

Task: Employee Table Operations

Name: Om Ingle Batch: 3rd Year IT Repository: [OmIngle82/JDBC-Assessments](#)

1. Class & Method Explanation

- **EmployeeCRUD Class:** This class handles the database connection and executes the full lifecycle of SQL operations (Create, Read, Update, Delete) on the employee table.
- **createSQL:** Defines the table structure with columns id, name, email, city, and contact. It uses IF NOT EXISTS to ensure the program runs smoothly even if the table was created in a previous run.
- **insertSQL:** Inserts a sample record (Virat Kohli, ID 18) into the table to verify that data entry works correctly.
- **updateSQL:** Modifies the existing record by changing the city from 'Delhi' to 'Mumbai' for the employee with ID 18.
- **deleteSQL:** Removes the record with ID 18 to clean up the database after the operations are verified.
- **stmt.executeUpdate():** This single method is used for all these operations (CREATE, INSERT, UPDATE, DELETE) as they modify the database state rather than returning a result set.

2. Console Output Screenshot

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure under "DB_Operations [JDBC-Assessments main]".
- EmployeeCRUD.java:** The code is as follows:

```
1 import java.sql.Connection;
2 import java.sql.DriverManager;
3 import java.sql.SQLException;
4 import java.sql.Statement;
5
6 public class EmployeeCRUD {
7     public static void main(String[] args) {
8
9         // Mandatory Output
10        System.out.println("Name: Om Ingle");
11        System.out.println("Batch: 3rd Year IT");
12        System.out.println("-----");
13
14        String url = "jdbc:mysql://localhost:3306/test_db";
15        String user = "root";
16        String password = "password";
17
18        Connection conn = null;
19        Statement stmt = null;
20
21        try {
22            conn = DriverManager.getConnection(url, user, password);
23            stmt = conn.createStatement();
24
25            // Create Table
26            String createTableQuery = "CREATE TABLE IF NOT EXISTS employee (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(255), email VARCHAR(255), city VARCHAR(255), contact VARCHAR(255));";
27            stmt.executeUpdate(createTableQuery);
28
29            // Insert Record
30            String insertQuery = "INSERT INTO employee (name, email, city, contact) VALUES ('Virat Kohli', 'viratkohli@example.com', 'Delhi', '9898989898');";
31            stmt.executeUpdate(insertQuery);
32
33            // Update Record
34            String updateQuery = "UPDATE employee SET city = 'Mumbai' WHERE id = 18";
35            stmt.executeUpdate(updateQuery);
36
37            // Delete Record
38            String deleteQuery = "DELETE FROM employee WHERE id = 18";
39            stmt.executeUpdate(deleteQuery);
40
41        } catch (SQLException e) {
42            e.printStackTrace();
43        } finally {
44            if (stmt != null) {
45                try {
46                    stmt.close();
47                } catch (SQLException e) {
48                    e.printStackTrace();
49                }
50            }
51            if (conn != null) {
52                try {
53                    conn.close();
54                } catch (SQLException e) {
55                    e.printStackTrace();
56                }
57            }
58        }
59    }
60}
```

- Console:** Displays the output of the program execution.

```
Batch: 3rd Year IT
-----
Database Connected.
Table 'employee' Ready.
Inserted: Virat Kohli (ID 18)
Updated: Virat Kohli moved to Mumbai.
Deleted: Record ID 18 removed.
Resources Closed.
```