GAM/IT/2022/F/0079 – J.S.Senarath

Java JDBC Lab Practical using NetBeans IDE 8.2

Java JDBC Lab Practical using NetBeans IDE 8.2

- 1. Set Up MySQL Database
- Install MySQL (if you haven't already) and create a database for the lab.
- Open MySQL Workbench or command line and execute the following SQL commands:

CREATE DATABASE employee db;

USE employee db;

CREATE TABLE employees (

id INT PRIMARY KEY AUTO INCREMENT,

name VARCHAR(100),

position VARCHAR(100),

salary DECIMAL(10, 2)

);

-- Insert some sample data

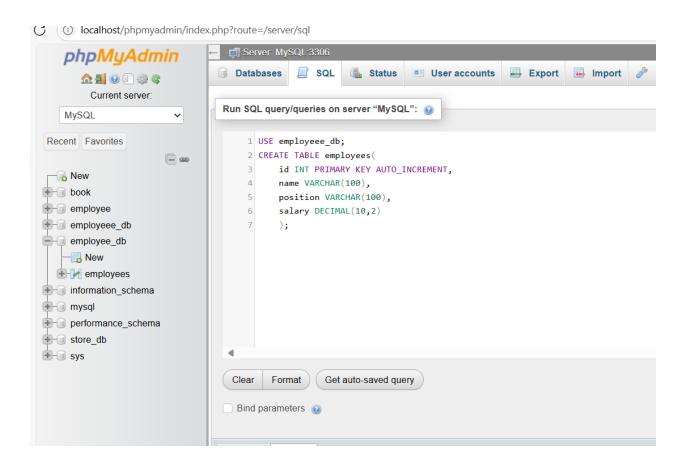
INSERT INTO employees (name, position, salary) VALUES ('John Doe', 'Software Engineer', 75000);

INSERT INTO employees (name, position, salary) VALUES ('Jane Smith', 'HR Manager', 65000);

INSERT INTO employees (name, position, salary) VALUES ('Steve Brown', 'Team Lead', 85000);









```
INSERT INTO `employees`( `name`, `position`, `salary`) VALUES ('[John Doe]','[Software Engineer]','[75000]');
2 INSERT INTO `employees`( `name`, `position`, `salary`) VALUES ('[jane smith]','[HR Manager]','[65000]');
3 INSERT INTO `employees`( `name`, `position`, `salary`) VALUES ('[Steve Brown]','[Team Lead]','[85000]');
            3 [John Doe]
                                             [Software Engineer]
                                                                  0.00
     🥒 Edit
                                                                  0.00
     Delete
                               4 [jane smith]
                                             [HR Manager]
     🥒 Edit 👫 Copy
                     Delete
                               5 [Steve Brown] [Team Lead]
                                                                  0.00
        Check all
                                              👬 Copy
                      With selected:
                                                           Delete
                                                                     Export
    Show all
                Number of rows:
                                25 🗸
                                                     Search this table
                                                                             Sort by key:
                                                                                         None
                                           Filter rows:
```

2. Set Up NetBeans Project

- 1. Open NetBeans IDE 8.2.
- 2. Create a new Java application:
- o Go to File > New Project.
- o Select Java as the project type, and choose Java Application.
- o Name your project JDBCExample.
- 3. Add MySQL JDBC Driver to your project:
- o Right-click on the project in the Projects pane.
- o Select Properties.
- o In the Libraries tab, click Add JAR/Folder.
- o Navigate to the location of your mysql-connector-java-x.x.xx.jar file and add it.

```
dashboard
  dashboard1
                                   4
  IssueBook
                                   5
                                        package jdbcexample;
JDBC
JDBCExample
                                  7 - /**
 Source Packages
                                  8

✓ III jdbcexample

                                         * @author janan
                                  9
      DatabaseConnection.java
                                 10
      Employee.java
                                  11
                                        public class JDBCExample {
      EmployeeDAO.java
                                  12
      13
                                            public static void main(String[] args) {
      Main.java
                                  14
      ■ NewJPanel.java
                                  15
> ia Test Packages
                                  16
Libraries
                                  17
  > 🗐 mysql-connector-j-9.3.0.zip
                                  18
   JDK 17 (Default)
                                  19
 Test Libraries
🍃 login
LOGINN
  mavenproject1
mavenproject2
mavenproject5
  memberdetails
```

3. Establish JDBC Connection

• Create a DatabaseConnection.java class to establish a connection to your database.

Code for DatabaseConnection.java:

```
package jdbcexample;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DatabaseConnection {
    private static final String URL = "jdbc:mysql://localhost:3306/employeee_db"; // Database URL
```

```
private static final String USER = "root"; // Your MySQL username
private static final String PASSWORD = ""; // Your MySQL password

public static Connection getConnection() throws SQLException {
    try {
            // Load the JDBC driver
            Class.forName("com.mysql.cj.jdbc.Driver");
            // Return the database connection
            return DriverManager.getConnection(URL, USER, PASSWORD);
        } catch (ClassNotFoundException | SQLException e) {
            System.out.println("Connection failed: " + e.getMessage());
            throw new SQLException("Failed to establish connection.");
        }
    }
}
```

```
^ Click nois://nonost/systemfilesystem/Templates/Classes/Class.java to edit this template
dashboard1
IssueBook
                                                                                               package jdbcexample;
) JDBC
JDBCExample
                                                                                 7 = import java.sql.Connection;
  Source Packages
                                                                                              import java.sql.DriverManager;

y  

ightharpoonup ji y  

jdbcexample

ightha
                                                                                              import java.sql.SQLException;
            10
              Employee.java
                                                                                              public class DatabaseConnection {
                                                                               11
              EmployeeDAO.java
                                                                               12
              JDBCExample.java
                                                                                                        private static final String URL = "jdbc:mysql://localhost:3306/employeee db"; // Database URL
                                                                               13
              Main.java
                                                                                                        private static final String USER = "root"; // Your MySQL username
                                                                               14
              NewJPanel.java
                                                                                                        private static final String PASSWORD = ""; // Your MySQL password
                                                                               15
  Test Packages
                                                                               16
  Libraries
                                                                               17
                                                                                                        public static Connection getConnection() throws SQLException {
        mysql-connector-j-9.3.0.zip
                                                                                18
       JDK 17 (Default)
                                                                                                                             // Load the JDBC driver
                                                                                19
  Test Libraries
                                                                                20
                                                                                                                             Class.forName(className:"com.mysql.cj.jdbc.Driver");
login
                                                                                21
                                                                                                                             // Return the database connection
▶ LOGINN
                                                                               22
                                                                                                                            return DriverManager.getConnection(url: URL, user: USER, password: PASSWORD);
mavenproject1
                                                                                                                   } catch (ClassNotFoundException | SQLException e) {
                                                                               23
mavenproject2
                                                                               24
                                                                                                                            System.out.println("Connection failed: " + e.getMessage());
mavenproject5
                                                                                25
                                                                                                                             throw new SQLException(reason: "Failed to establish connection.");
memberdetails
                                                                               26
MultiThreadApp2
                                                                               27
  Source Packages
                                                                               28

✓ III multithreadapp2

                                                                                               }
                                                                                29
              MultiThreadApp2.java
              RunnableTask.java
```

4. Perform CRUD Operations

Next, we will create a class called EmployeeDAO.java that contains methods for performing CRUD operations.

1. Create EmployeeDAO.java for CRUD Operations:

```
assignment
attendence
Book Details
                                   2
3
4
                                          * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
dashboard
                                         * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
dashboard1
IssueBook
JDBC
JDBCExample
Source Packages
/ III idbcexample
    DatabaseConnection.java
Employee inve
                                         * @author janan
                                  10
       Employee.java
                                      import java.sql.*;
                                  11
EmployeeDAO.java

JDBCExample.java
                                  12
                                         import java.util.ArrayList;
    Main.java
NewJPanel.java
                                  13
                                         import java.util.List;
                                  14
                                  15
                                         public class EmployeeDAO {
Test Packages
i Libraries
                                  17
18
                                             // Create an employee
> iii mysql-connector-j-9.3.0.zip
> iii JDK 17 (Default)
                                             public static void addEmployee(String name, String position, double salary) {
                                  19
                                                  String sql = "INSERT INTO employees (name, position, salary) VALUES (?, ?, ?)";
Test Libraries
                                  20
21
login
                                                  try (Connection conn = DatabaseConnection.getConnection();
LOGINN
                                  22
                                                       PreparedStatement stmt = conn.prepareStatement(string:sql)) {
mavenproject1
                                  23
mavenproject2
                                  24
25
                                                      stmt.setString(i: 1, string:name);
mavenproject5
                                                       stmt.setString(i: 2, string:position);
memberdetails
                                  26
                                                      stmt.setDouble(i: 3, d: salary);
MultiThreadApp2
                                  27
Source Packages
                                                       int rowsAffected = stmt.executeUpdate();
                                  28
✓ III multithreadapp2
                                  29
                                                      System.out.println("Employee added successfully. Rows affected: " + rowsAffected);
    Counter.java
MultiThreadApp2.java
                                  30
                                                  } catch (SQLException e) {
                                  №
32
                                                      e.printStackTrace();
     RunnableTask.java
SimpleThread.java
                                  33
    SynchronizedExample.java
ThreadLifecycleExample.java
                                  34
35
                                             // Read all employees
     ThreadPoolExample.java
                                             public static List<Employee> getAllEmployees() {
   List<Employee> employees = new ArrayList<>();
Test Packages
                                                                                                                                                                                          37:23
```

```
(consider the second of the se
                                                                   ...va 🗟 DatabaseConnection.java × 🗟 EmployeeDAO.java × 💰 Employee.java × 🚳 Main.java × 🚳 Xml.java × 🚳 xml.java × 🚳 xml.java
jects × Files
                             Services
assignment
                                                                  attendence
                                                                  36 -
                                                                                      public static List<Employee> getAllEmployees() {
Book Details
                                                                   9
                                                                                            List<Employee> employees = new ArrayList<>();
dashboard
                                                                                              String sql = "SELECT * FROM employees";
                                                                  38
dashboard1
                                                                  39
IssueBook
                                                                  40
                                                                                              try (Connection conn = DatabaseConnection.getConnection();
JDBC
                                                                  41
                                                                                                        Statement stmt = conn.createStatement();
JDBCExample
                                                                                                       ResultSet rs = stmt.executeQuery(string:sql)) {
                                                                  42
 Source Packages
                                                                  43
     √ III jdbcexample
                                                                   44
                                                                                                      while (rs.next()) {
              DatabaseConnection.java
                                                                                                              Employee employee = new Employee(
                                                                  45
              Employee.java
                                                                  46
                                                                                                                      id: rs.getInt(string:"id"),
            EmployeeDAO.java
                                                                  47
                                                                                                                      name: rs.getString(string: "name"),
              JDBCExample.java
                                                                  48
                                                                                                                      position: rs.getString(string: "position"),
              Main.java
                                                                  49
                                                                                                                      salary: rs.getDouble (string: "salary")
              NewJPanel.java
                                                                  50
> 📵 Test Packages
                                                                  51
                                                                                                               employees.add(e: employee);
Libraries
                                                                  52
         mysql-connector-j-9.3.0.zip
                                                                  53
                                                                                              } catch (SQLException e) {
        JDK 17 (Default)
                                                                                                      e.printStackTrace();
                                                                   &
    Test Libraries
                                                                  55
login
                                                                  56
LOGINN
                                                                  57
                                                                                              return employees;
mavenproject1
mavenproject2
                                                                  58
                                                                  59
mavenproject5
                                                                  60
                                                                                      // Update an employee's information
memberdetails
                                                                  61
                                                                                      public static void updateEmployee(int id, String name, String position, double salary) {
MultiThreadApp2
                                                                                              String sql = "UPDATE employees SET name = ?, position = ?, salary = ? WHERE id = ?";
                                                                  62
 63

✓ ■ multithreadapp2

                                                                  64
                                                                                              try (Connection conn = DatabaseConnection.getConnection();
              Counter.java
                                                                                                       PreparedStatement stmt = conn.prepareStatement(string:sql)) {
                                                                  65
              MultiThreadApp2.java
                                                                  66
              RunnableTask.java
                                                                  67
                                                                                                      stmt.setString(i: 1, string:name);
              SimpleThread.iava
                                                                  68
                                                                                                      stmt.setString(i: 2, string:position);
              SynchronizedExample.java
                                                                  69
                                                                                                      stmt.setDouble(i: 3, d: salary);
              ThreadLifecycleExample.java
                                                                  70
                                                                                                      stmt.setInt(i: 4, i1: id);
             ThreadPoolExample.java
                                                                  71
> 🛅 Test Packages
                                                                                                      int rowsAffected = stmt.executeUpdate();
                                                                  72
```

```
cts × Files
                                                                               ...va db DatabaseConnection.java × db EmployeeDAO.java × db Employ
                                Services
assignment
                                                                                               attendence
                                                                                                                            stmt.setInt(i: 4, i1: id);
Book Details
                                                                               71
dashboard
                                                                               72
                                                                                                                           int rowsAffected = stmt.executeUpdate();
dashboard1
                                                                                                                           System.out.println("Employee updated successfully. Rows affected: " |+ rowsAffected);
                                                                               73
IssueBook
                                                                                                                 } catch (SQLException e) {
JDBCExample
                                                                                Q
                                                                                                                           e.printStackTrace();
                                                                               76
77
 Source Packages
   √ 🖺 jdbcexample
                                                                               78
               DatabaseConnection.java
                                                                                                       // Delete an employee
              Employee.java
                                                                               79
                                                                               80
                                                                                                      public static void deleteEmployee(int id) {
EmployeeDAO.java
                                                                               81
                                                                                                                String sql = "DELETE FROM employees WHERE id = ?";
             JDBCExample.java
             Main.java
                                                                                                                 try (Connection conn = DatabaseConnection.getConnection();
             ☐ Test Packages
                                                                                                                             PreparedStatement stmt = conn.prepareStatement(string:sql)) {
 Libraries
                                                                               86
                                                                                                                            stmt.setInt(i: 1, i1: id);
       mysql-connector-j-9.3.0.zip
       JDK 17 (Default)
                                                                                                                           int rowsAffected = stmt.executeUpdate();
                                                                                                                           System.out.println("Employee deleted successfully. Rows affected: " + rowsAffected);
                                                                               88
 Test Libraries
                                                                               89
                                                                                                                } catch (SQLException e) {
LOGINN
                                                                                                                           e.printStackTrace();
                                                                               91
mavenproject1
                                                                               92
mavenproject2
                                                                               93
mavenproject5
memberdetails
MultiThreadApp2
 Source Packages

✓ ■ multithreadapp2
              Counter.java
              MultiThreadApp2.java
             RunnableTask.java
SimpleThread.iava
```

5. Create Employee.java Class

Create a simple Employee.java POJO (Plain Old Java Object) to represent employee data. Code for Employee.java:

```
* @author janan
            10
            11
                  public class Employee {
            12
            13
                      private int id;
                      private String name;
            14
                      private String position;
            15
tion.java
            16
                      private double salary;
            18 📮
                      public Employee (int id, String name, String position, double salary) {
            19
                          this.id = id;
            20
                          this.name = name;
                          this.position = position;
            21
                          this.salary = salary;
            22
            23
            25
                      // Getters and setters
            26
                      public int getId() { return id; }
            27 -
                      public void setId(int id) { this.id = id; }
            28
            29 -
                      public String getName() { return name; }
            30 🖃
                      public void setName(String name) { this.name = name; }
            32 -
                      public String getPosition() { return position; }
            33 -
                      public void setPosition(String position) { this.position = position; }
            34
            35 -
                      public double getSalary() { return salary; }
            36 🖃
                      public void setSalary(double salary) { this.salary = salary; }
2.java
            37
                      @Override

    □

                          return "Employee (id=" + id + ", name='" + name + "', position='" + position + "', salary=" + salary + ')';
xample.java
            41
ple.java
            42
```

6. Test the Application

Create a Main.java class to test the CRUD operations.

Code for Main.java:

```
* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
      * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template */
4
     package jdbcexample;
6 - import java.util.List;
8
     public class Main {
         public static void main(String[] args) {
11
            // Add employees
12
              EmployeeDAO.addEmployee(name: "Alice Cooper", position: "Developer", salary:70000);
             EmployeeDAO.addEmployee(name: "Bob Marley", position: "Manager", salary:80000);
13
14
15
             // Update employee
16
             EmployeeDAO.updateEmployee(id: 1, name: "John Doe", position: "Senior Software Engineer", salary: 90000);
17
             // Get all employees
18
19
             List<Employee> employees = EmployeeDAO.getAllEmployees();
20
              employees.forEach(System.out::println);
21
22
              // Delete employee
23
             EmployeeDAO.deleteEmployee(id: 2);
25
      }
```