Lab sheet -03

```
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);
//Code for DatabaseConnection.java:
packagejdbcexample;
importjava.sql.Connection;
importjava.sql.DriverManager;
importjava.sql.SQLException;
/**
 *
 * @author student
 */
public class DatabaseConnection {
private static final String URL ="jdbc:mysql://localhost:3306/employee_db"; // Database URL
private static final String USER = "root";
private static final String PASSWORD = "";
```

```
public static Connection getConnection() throws SQLException {

try {

Class.forName("com.mysql.cj.jdbc.Driver");

returnDriverManager.getConnection(URL, USER, PASSWORD);
}
catch (ClassNotFoundException | SQLException e) {
System.out.println("Connection failed:" + e.getMessage());
throw new SQLException("Failed to establish connection.");
   }
 }
}
```

1. Open NetBeans IDE 8.2.

2. Create a new Java application:

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

//Code for EmployeeDAO.java:

```
packagejdbcexample;
importjava.sql.Connection;
importjava.sql.DriverManager;
importjava.sql.SQLException;
/**
```

```
 *
 * @author student
 */
public class DatabaseConnection {
private static final String URL ="jdbc:mysql://localhost:3306/employee_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 {
Class.forName("com.mysql.cj.jdbc.Driver");
returnDriverManager.getConnection(URL, USER, PASSWORD);
}
catch (ClassNotFoundException | SQLException e) {
System.out.println("Connection failed:" + e.getMessage());
throw new SQLException("Failed to establish connection.");
   }
 }
}
//Code for EmployeeDAO.java:
packagejdbcexample;
importjava.sql.*;
importjava.util.ArrayList;
importjava.util.List;
/**
 *
 * @author student
 */
public class EmployeeDAO {
public static void addEmployee(String name, String position, double salary) {
```

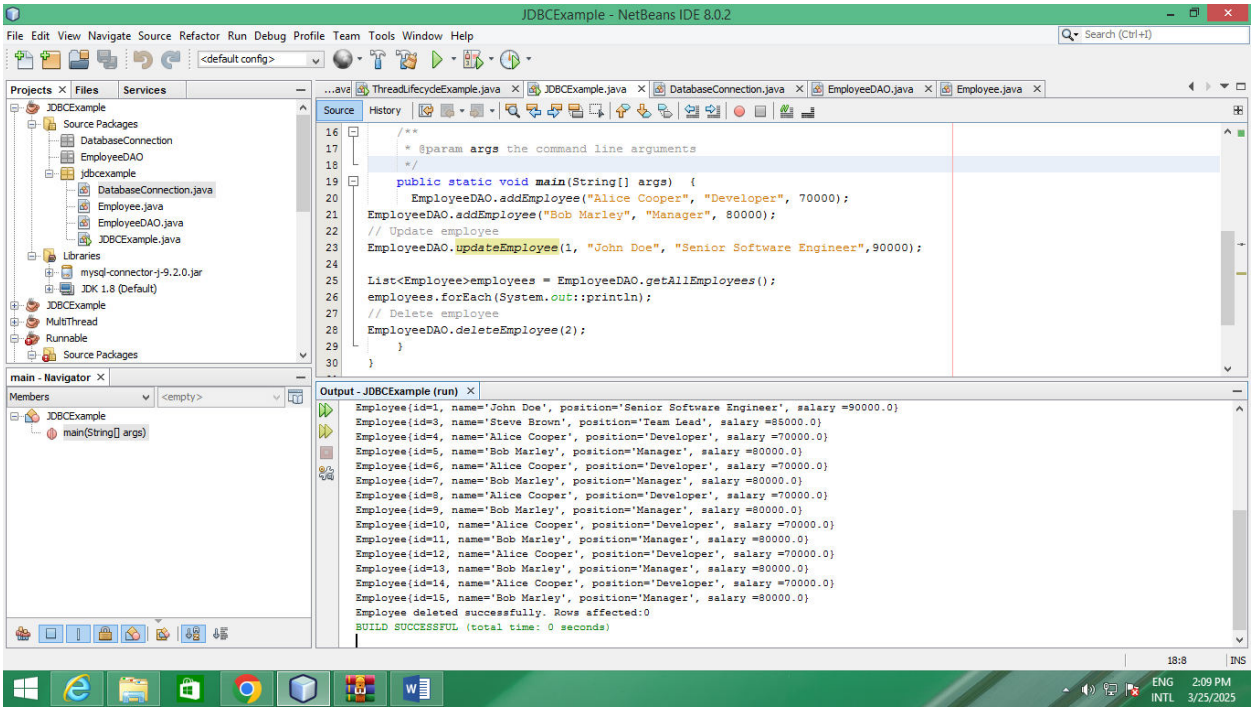
```
String sql = "INSERT INTO employees (name, position, salary) VALUES(?, ?, ?)";
try (Connection conn = DatabaseConnection.getConnection();
PreparedStatementstmt = conn.prepareStatement(sql)) {
stmt.setString(1, name);
stmt.setString(2, position);
stmt.setDouble(3, salary);
introwsAffected = stmt.executeUpdate();
System.out.println("Employee added successfully. Rows affected:" + rowsAffected);
} catch (SQLException e) {
e.printStackTrace();
}
}
// Read all employees
public static List<Employee>getAllEmployees() {
List<Employee> employees = new ArrayList<>();
String sql = "SELECT * FROM employees";
try (Connection conn = DatabaseConnection.getConnection();
Statement stmt = conn.createStatement();
ResultSetrs = stmt.executeQuery(sql)) {
while (rs.next()) {
Employee employee = new Employee(

rs.getInt("id"),
rs.getString("name"),
rs.getString("position"),
rs.getDouble("salary")
);
employees.add(employee);
}
```

```
} catch (SQLException e) {
e.printStackTrace();
}
return employees;
}
// Update an employee  information
public static void updateEmployee(int id, String name, String position,
double salary) {
String sql = "UPDATE employees (name , position , salary)VALUES(?,?,?)";

try (Connection conn = DatabaseConnection.getConnection();
PreparedStatementstmt = conn.prepareStatement(sql)) {
stmt.setString(1, name);
stmt.setString(2, position);
stmt.setDouble(3, salary);
stmt.setInt(4, id);
introwsAffected = stmt.executeUpdate();
System.out.println("Employee updated successfully. Rows affected:" + rowsAffected);
} catch (SQLException e) {
e.printStackTrace();
}
}
// Delete an employee
public static void deleteEmployee(int id) {
String sql = "DELETE FROM employees WHERE id = ?";
try (Connection conn = DatabaseConnection.getConnection();
PreparedStatementstmt = conn.prepareStatement(sql)) {
stmt.setInt(1, id);
introwsAffected = stmt.executeUpdate();
```

```
System.out.println("Employee deleted successfully. Rows affected:" + rowsAffected);
} catch (SQLException e) {
e.printStackTrace();
}
}
}
//Code for Employee.java:
public class Employee {

privateint id;
private String name;
private String position;
private double salary;
public Employee(int id, String name, String position, double salary) {
this.id = id;
this.name = name;
this.position = position;
this.salary = salary;
}
// Getters and setters
publicintgetId() { return id; }
public void setId(int id) { this.id = id; }
public String getName() { return name; }
public void setName(String name) { this.name = name; }
public String getPosition() { return position; }
public void setPosition(String position) { this.position = position; }
public double getSalary() { return salary; }
public void setSalary(double salary) { this.salary = salary; }
@Override
```

```
public String toString() {
return "Employee{id=" + id + ", name='" + name + "', position='" +position + "', salary =" + salary + '}';
}
}
//Code for JDBCExample.java:
packagejdbcexample;

importjava.util.List;

/**
 *
 * @author student
 */
public class JDBCExample {

   /**
    * @paramargs the command line arguments
    */
public static void main(String[] args)  {
EmployeeDAO.addEmployee("Alice Cooper", "Developer", 70000);
EmployeeDAO.addEmployee("Bob Marley", "Manager", 80000);
EmployeeDAO.updateEmployee(1, "John Doe", "Senior Software Engineer",90000);

List<Employee>employees = EmployeeDAO.getAllEmployees();
employees.forEach(System.out::println);
EmployeeDAO.deleteEmployee(2);
   }}
```

OUT PUT



DATABASE UPDATE