JDBC

1) **Design a Java program to create a simple employee management system using JDBC and MySQL Connector/J. The program should allow users to perform the following operations**:

   a) **Add a new employee: The user can enter details like employee ID, name, department, and salary, and the program should add the employee to the database.**
   b) **Update employee details: The user can update the name, department, or salary of an existing employee based on their employee ID.**
   c) **Delete an employee: The user can delete an employee from the database based on their employee ID.**
   d) **Display all employees: The program should retrieve and display a list of all employees and their details from the database.**
   e) **Requirements:**
      i) **Use JDBC and MySQL Connector/J to connect to the MySQL database and perform CRUD (Create, Read, Update, Delete) operations.**
      ii) **Implement exception handling to handle possible errors during database interactions.**
      iii) **Provide a user-friendly console interface for the user to interact with the employee management system.**
      iv) **Cover Java topics such as classes, methods, user input and output (I/O), and exception handling.**
   f) **Note: Before running the program, make sure you have MySQL installed, create a database named "employee_management," and a table named "employees" with columns: "id" (INT, PRIMARY KEY), "name" (VARCHAR), "department" (VARCHAR), and "salary" (DOUBLE).**

Program:
Main.java

```
import employeemodel.Employee;

public class Main {
public static void main(String[] args) {
EmployeeService employeeService = new EmployeeService();
employeeService.start();
```

```
}
}
```

EmployeeDAO:

```
package employeDAO;

import employeemodel.Employee;
public class EmployeeDAO {
// JDBC connection details
private static final String JDBC_URL ="jdbc:mysql://localhost:3306/test";
private static final String USERNAME = "root";
private static final String PASSWORD = "";

// SQL queries
private static final String ADD_EMPLOYEE_SQL = "INSERT INTO employees (id, name,
department, salary) VALUES (?, ?, ?, ?)";
private static final String UPDATE_EMPLOYEE_SQL = "UPDATE employees SET name=?,
department=?, salary=? WHERE id=?";
private static final String DELETE_EMPLOYEE_SQL = "DELETE FROM employees WHERE id=?";
private static final String GET_ALL_EMPLOYEES_SQL = "SELECT * FROM employees";

// Methods
public void addEmployee(Employee employee) {
try (Connection connection = DriverManager.getConnection(JDBC_URL, USERNAME, PASSWORD);
PreparedStatement preparedStatement = connection.prepareStatement(ADD_EMPLOYEE_SQL)) {
preparedStatement.setInt(1, employee.getId());
preparedStatement.setString(2, employee.getName());
preparedStatement.setString(3, employee.getDepartment());
preparedStatement.setDouble(4, employee.getSalary());

preparedStatement.executeUpdate();
} catch (SQLException e) {
e.printStackTrace();
}
}
public void updateEmployee(Employee employee) {
try (Connection connection = DriverManager.getConnection(JDBC_URL, USERNAME, PASSWORD);
PreparedStatement preparedStatement = connection.prepareStatement(UPDATE_EMPLOYEE_SQL)) {
preparedStatement.setString(1, employee.getName());
preparedStatement.setString(2, employee.getDepartment());
preparedStatement.setDouble(3, employee.getSalary());
preparedStatement.setInt(4, employee.getId());

preparedStatement.executeUpdate();
} catch (SQLException e) {
e.printStackTrace();
}
}
public void deleteEmployee(int id) {
try (Connection connection = DriverManager.getConnection(JDBC_URL, USERNAME, PASSWORD);
PreparedStatement preparedStatement = connection.prepareStatement(DELETE_EMPLOYEE_SQL)) {
preparedStatement.setInt(1, id);

preparedStatement.executeUpdate();
} catch (SQLException e) {
e.printStackTrace();
}
}

public List<Employee> getAllEmployees() {
List<Employee> employees = new ArrayList<>();
try (Connection connection = DriverManager.getConnection(JDBC_URL, USERNAME, PASSWORD);
```

```
Statement statement = connection.createStatement();
ResultSet resultSet = statement.executeQuery(GET_ALL_EMPLOYEES_SQL)) {
while (resultSet.next()) {
int id = resultSet.getInt("id");
String name = resultSet.getString("name");
String department = resultSet.getString("department");
double salary = resultSet.getDouble("salary");
Employee employee = new Employee(id, name, department, salary);
employees.add(employee);
}
} catch (SQLException e) {
e.printStackTrace();
}
return employees;
}
}
```

EmployeeService:

```
package employeeservice;

import employeemodel.Employee;
import employeeDAO.EmployeeDAO;
import java.util.List;
import java.util.Scanner;
public class EmployeeService {
private final Scanner scanner;
private final EmployeeDAO employeeDAO;

public EmployeeService() {
scanner = new Scanner(System.in);
employeeDAO = new EmployeeDAO();
}

public void start() {
boolean exit = false;
while (!exit) {
System.out.println("\nEmployee Management System");
System.out.println("1. Add a new employee");
System.out.println("2. Update employee details");
System.out.println("3. Delete an employee");
System.out.println("4. Display all employees");
System.out.println("5. Exit");
System.out.print("Enter your choice: ");
int choice = scanner.nextInt();
scanner.nextLine(); // Consume newline character
switch (choice) {
case 1:
addEmployee();
break;
case 2:
updateEmployee();
break;
case 3:
deleteEmployee();
break;
case 4:
displayAllEmployees();
break;
case 5:
exit = true;
break;
```

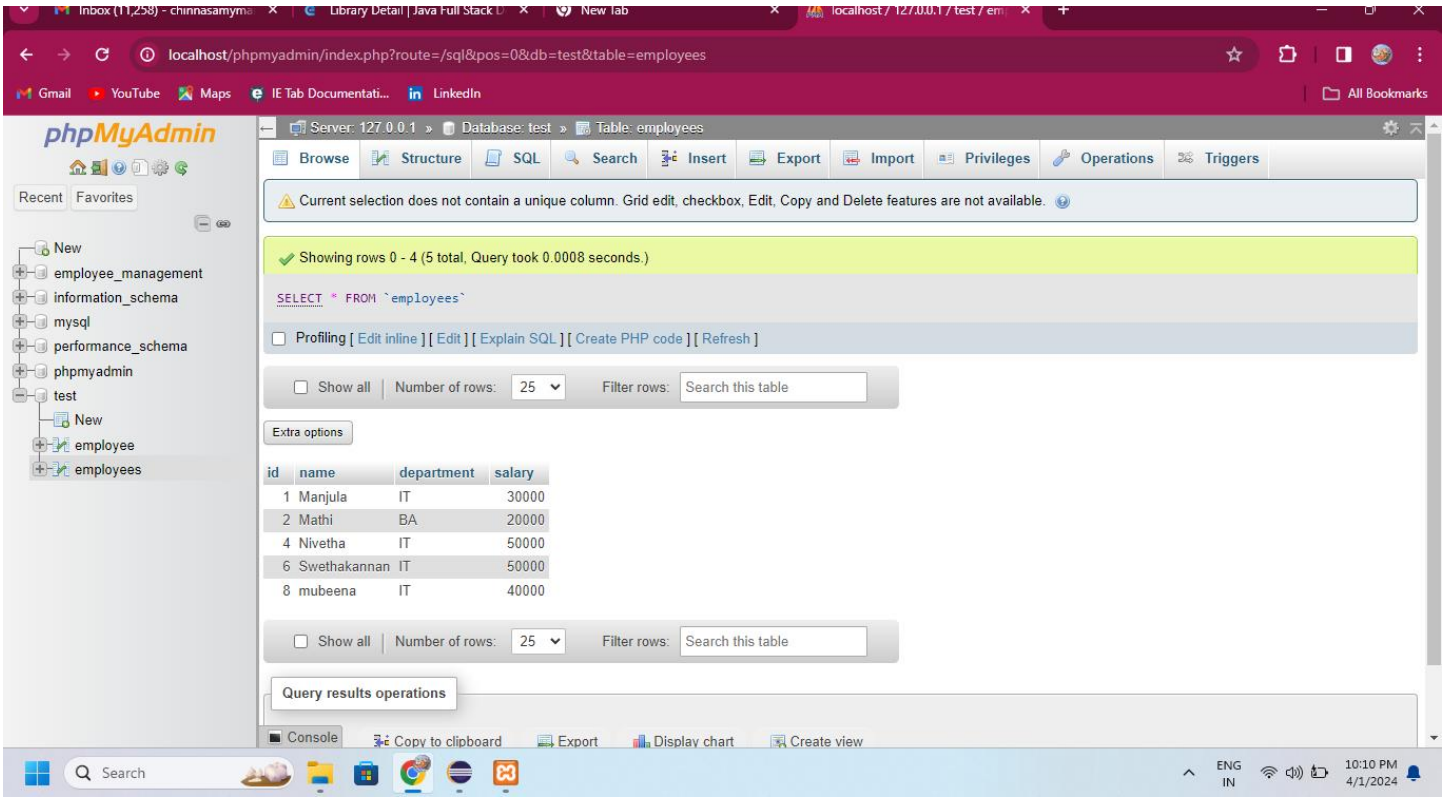
```java
default:
System.out.println("Invalid choice. Please enter a number between 1 and 5.");
}
}
}

private void addEmployee() {
System.out.println("\nEnter employee details:");
System.out.print("Employee ID: ");
int id = scanner.nextInt();
scanner.nextLine(); // Consume newline character
System.out.print("Name: ");
String name = scanner.nextLine();
System.out.print("Department: ");
String department = scanner.nextLine();
System.out.print("Salary: ");
double salary = scanner.nextDouble();

Employee employee = new Employee(id, name, department, salary);
employeeDAO.addEmployee(employee);
System.out.println("Employee added successfully.");
}

private void updateEmployee() {
System.out.println("\nEnter employee details to update:");
System.out.print("Employee ID: ");
int id = scanner.nextInt();
scanner.nextLine(); // Consume newline character
System.out.print("Name: ");
String name = scanner.nextLine();
System.out.print("Department: ");
String department = scanner.nextLine();
System.out.print("Salary: ");
double salary = scanner.nextDouble();
Employee employee = new Employee(id, name, department, salary);
employeeDAO.updateEmployee(employee);
System.out.println("Employee details updated successfully.");
}

private void deleteEmployee() {
System.out.println("\nEnter employee ID to delete:");
int id = scanner.nextInt();
scanner.nextLine(); // Consume newline character

employeeDAO.deleteEmployee(id);
System.out.println("Employee deleted successfully.");
}

private void displayAllEmployees() {
List<Employee> employees = employeeDAO.getAllEmployees();
System.out.println("\nAll Employees:");
System.out.println("ID\tName\tDepartment\tSalary");
for (Employee employee : employees) {
System.out.println(employee.getId() + "\t" + employee.getName() + "\t" +
employee.getDepartment() + "\t" + employee.getSalary());
}
}
}
```

EmployeeModel:

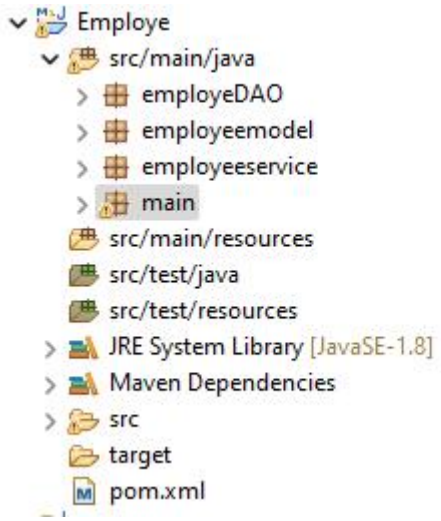
```java
package employeemodel;
public class Employee {
private int id;
private String name;
private String department;
private double salary;

// Constructors, getters, and setters
public Employee(int id, String name, String department, double salary) {
this.setId(id);
this.setName(name);
this.setDepartment(department);
this.setSalary(salary);
}
public int getId() {
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 getDepartment() {
return department;
}

public void setDepartment(String department) {
this.department = department;
}

public double getSalary() {
return salary;
}
public void setSalary(double salary) {
this.salary = salary;
}

// Getters and setters
//...
}
```

Output:

1. **Mysql Database:**



2. **Maven Project for Employee Management**



3. Adding & Updating Employee Details in Database

```
eclipse-workspace - Employe/src/main/java/main/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Console
Main (4) [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe (Apr 1, 2024, 10:13:04 PM) [pid: 7456]

Employee Management System
1. Add a new employee
2. Update employee details
3. Delete an employee
4. Display all employees
5. Exit
Enter your choice: 1

Enter employee details:
Employee ID: 11
Name: Parasu
Department: Support
Salary: 40000
Employee added successfully.

Employee Management System
1. Add a new employee
2. Update employee details
3. Delete an employee
4. Display all employees
5. Exit
Enter your choice: 2

Enter employee details to update:
Employee ID: 12
Name: Parasu
Department: Manager
Salary: 50000
Employee details updated successfully.

Employee Management System
1. Add a new employee
```

4. Before Deleting the employee details to show the existing DB details

```
eclipse-workspace - Employe/src/main/java/main/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Console
Main (4) [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe (Apr 1, 2024, 10:13:04 PM) [pid: 7456]
Employee Management System
1. Add a new employee
2. Update employee details
3. Delete an employee
4. Display all employees
5. Exit
Enter your choice: 4

All Employees:
ID      Name    Department      Salary
1       Manjula IT      30000.0
2       Mathi   BA      20000.0
4       Nivetha IT      50000.0
6       Swethakannan    IT      50000.0
8       mubeena IT      40000.0
11      Parasu  Support 40000.0

Employee Management System
1. Add a new employee
2. Update employee details
3. Delete an employee
4. Display all employees
5. Exit
Enter your choice: 3

Enter employee ID to delete:
6
Employee deleted successfully.
```

// Here employee Parasu details added along with the existing employee details

5. Deletion employee deatails & Display after deletion:
(Swethakannan name deleted in below figure)