

Name: Sudhir Singh

Roll No: TEAD23152

Batch: A3

Practical no.7

Title: Database Connectivity: Write a program to implement MySQL/Oracle database connectivity with any front end language to implement Database navigation operations (add, delete, edit etc.)

Program:

```
import java.sql.*;
import java.util.Scanner;

public class EmployeeDatabase {
    private static final String URL = "jdbc:mysql://localhost:3306/testDB";
    private static final String USER = "root";
    private static final String PASSWORD = "sudhir@867@";

    public static Connection connectToMySQL() {
        try {
            // Load MySQL JDBC driver
            Class.forName("com.mysql.cj.jdbc.Driver");
            // Establish connection
            return DriverManager.getConnection(URL, USER, PASSWORD);
        } catch (ClassNotFoundException e) {
            System.out.println("MySQL JDBC Driver not found.");
            return null;
        } catch (SQLException e) {
            System.out.println("Error connecting to database: " + e.getMessage());
            return null;
        }
    }

    public static void addEmployee(String name, int age, String department) {
        String sql = "INSERT INTO employees (name, age, department) VALUES (?, ?, ?)";
        try (Connection conn = connectToMySQL(); PreparedStatement pstmt = conn.prepareStatement(sql)) {
            if (conn == null) return;
            pstmt.setString(1, name);
            pstmt.setInt(2, age);
            pstmt.setString(3, department);
            pstmt.executeUpdate();
        }
    }
}
```

```

        System.out.println("Employee '" + name + "' added successfully.");
    } catch (SQLException e) {
        System.out.println("Failed to add employee: " + e.getMessage());
    }
}

```

```

public static void displayEmployees() {
    String sql = "SELECT * FROM employees";
    try (Connection conn = connectToMySQL(); Statement stmt = conn.createStatement()) {
        if (conn == null) return;
        ResultSet rs = stmt.executeQuery(sql);
        System.out.println("\nEmployees List:");
        while (rs.next()) {
            System.out.printf("ID: %d, Name: %s, Age: %d, Department: %s\n",
                rs.getInt("id"), rs.getString("name"), rs.getInt("age"), rs.getString("department"));
        }
    } catch (SQLException e) {
        System.out.println("Failed to fetch employees: " + e.getMessage());
    }
}

```

```

public static void editEmployee(int employeeId, String newName, Integer newAge, String newDepartment) {
    try (Connection conn = connectToMySQL()) {
        if (conn == null) return;

        if (newName != null) {
            String sql = "UPDATE employees SET name = ? WHERE id = ?";
            try (PreparedStatement pstmt = conn.prepareStatement(sql)) {
                pstmt.setString(1, newName);
                pstmt.setInt(2, employeeId);
                pstmt.executeUpdate();
            }
        }

        if (newAge != null) {
            String sql = "UPDATE employees SET age = ? WHERE id = ?";
            try (PreparedStatement pstmt = conn.prepareStatement(sql)) {
                pstmt.setInt(1, newAge);
                pstmt.setInt(2, employeeId);
            }
        }
    }
}

```

```

        pstmt.executeUpdate();
    }
}

if (newDepartment != null) {
    String sql = "UPDATE employees SET department = ? WHERE id = ?";
    try (PreparedStatement pstmt = conn.prepareStatement(sql)) {
        pstmt.setString(1, newDepartment);
        pstmt.setInt(2, employeeId);
        pstmt.executeUpdate();
    }
}

System.out.println("Employee ID " + employeeId + " updated successfully.");
} catch (SQLException e) {
    System.out.println("Failed to update employee: " + e.getMessage());
}
}

public static void deleteEmployee(int employeeId) {
    String sql = "DELETE FROM employees WHERE id = ?";
    try (Connection conn = connectToMySQL(); PreparedStatement pstmt = conn.prepareStatement(sql)) {
        if (conn == null) return;
        pstmt.setInt(1, employeeId);
        pstmt.executeUpdate();
        System.out.println("Employee ID " + employeeId + " deleted successfully.");
    } catch (SQLException e) {
        System.out.println("Failed to delete employee: " + e.getMessage());
    }
}

public static void showMenu() {
    Scanner scanner = new Scanner(System.in);
    while (true) {
        System.out.println("\n--- Employee Database Operations ---");
        System.out.println("1. Add Employee");
        System.out.println("2. Display All Employees");
        System.out.println("3. Edit Employee");
        System.out.println("4. Delete Employee");
        System.out.println("5. Exit");
    }
}

```

```
System.out.print("Enter your choice: ");
```

```
String choice = scanner.nextLine();
```

```
switch (choice) {
```

```
    case "1":
```

```
        System.out.print("Enter employee name: ");
```

```
        String name = scanner.nextLine();
```

```
        System.out.print("Enter employee age: ");
```

```
        int age = Integer.parseInt(scanner.nextLine());
```

```
        System.out.print("Enter employee department: ");
```

```
        String department = scanner.nextLine();
```

```
        addEmployee(name, age, department);
```

```
        break;
```

```
    case "2":
```

```
        displayEmployees();
```

```
        break;
```

```
    case "3":
```

```
        System.out.print("Enter employee ID to edit: ");
```

```
        int editId = Integer.parseInt(scanner.nextLine());
```

```
        System.out.print("Enter new name (leave blank to keep existing): ");
```

```
        String newName = scanner.nextLine();
```

```
        System.out.print("Enter new age (leave blank to keep existing): ");
```

```
        String newAgeStr = scanner.nextLine();
```

```
        System.out.print("Enter new department (leave blank to keep existing): ");
```

```
        String newDepartment = scanner.nextLine();
```

```
        Integer newAge = newAgeStr.isEmpty() ? null : Integer.parseInt(newAgeStr);
```

```
        editEmployee(editId, newName.isEmpty() ? null : newName, newAge, newDepartment.isEmpty() ? null :  
newDepartment);
```

```
        break;
```

```
    case "4":
```

```
        System.out.print("Enter employee ID to delete: ");
```

```
        int deleteId = Integer.parseInt(scanner.nextLine());
```

```
        deleteEmployee(deleteId);
```

```
        break;
```

```
    case "5":
```

```

        System.out.println("Exiting program.");
        scanner.close();
        return;

    default:
        System.out.println("Invalid choice, please try again.");
    }
}

}

public static void main(String[] args) {
    showMenu();
}
}

```

/*Folder structure:

/user/sudhi/

```

├── EmployeeDB/
│   ├── EmployeeDatabase.java
│   └── mysql-connector-j-9.4.0/
│       └── mysql-connector-j-9.4.0.jar

```

***/**

OUTPUT

```

PS C:\Users\sudhi> mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.40 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE testDB;
Query OK, 1 row affected (0.07 sec)

mysql> USE testDB;
Database changed
mysql> CREATE TABLE employees (
  ->   id INT AUTO_INCREMENT PRIMARY KEY,
  ->   name VARCHAR(50),
  ->   age INT,
  ->   department VARCHAR(50)
  -> );
Query OK, 0 rows affected (0.07 sec)

```

```
mysql> INSERT INTO employees (name, age, department) VALUES
-> ('Sudhir', 21, 'AI&DS'),
-> ('Rahul', 23, 'Computer Science'),
-> ('Saish', 22, 'Information Technology'),
-> ('Ayyub', 25, 'E&TC');
```

```
Query OK, 4 rows affected (0.03 sec)
Records: 4 Duplicates: 0 Warnings: 0
```

```
mysql> SELECT * FROM employees;
```

id	name	age	department
1	Sudhir	21	AI&DS
2	Rahul	23	Computer Science
3	Saish	22	Information Technology
4	Ayyub	25	E&TC

```
4 rows in set (0.04 sec)
```

```
mysql> exit
```

```
Bye
```

```
PS C:\Users\sudhi\EmployeeDB> javac -cp ".;mysql-connector-j-9.4.0.jar" EmployeeDatabase.java
PS C:\Users\sudhi\EmployeeDB> java -cp ".;mysql-connector-j-9.4.0.jar" EmployeeDatabase
```

```
--- Employee Database Operations ---
```

1. Add Employee
2. Display All Employees
3. Edit Employee
4. Delete Employee
5. Exit

```
Enter your choice: 1
```

```
Enter employee name: Harsh
```

```
Enter employee age: 20
```

```
Enter employee department: AI&DS
```

```
Employee 'Harsh' added successfully.
```

```
--- Employee Database Operations ---
```

1. Add Employee
2. Display All Employees
3. Edit Employee
4. Delete Employee
5. Exit

```
Enter your choice: 2
```

```
Employees List:
```

```
ID: 1, Name: Sudhir, Age: 21, Department: AI&DS
ID: 2, Name: Rahul, Age: 23, Department: Computer Science
ID: 3, Name: Saish, Age: 22, Department: Information Technology
ID: 4, Name: Ayyub, Age: 25, Department: E&TC
ID: 5, Name: Harsh, Age: 20, Department: AI&DS
```

```
--- Employee Database Operations ---
1. Add Employee
2. Display All Employees
3. Edit Employee
4. Delete Employee
5. Exit
Enter your choice: 3
Enter employee ID to edit: 3
Enter new name (leave blank to keep existing): Saishh
Enter new age (leave blank to keep existing):
Enter new department (leave blank to keep existing): AI&DS
Employee ID 3 updated successfully.

--- Employee Database Operations ---
1. Add Employee
2. Display All Employees
3. Edit Employee
4. Delete Employee
5. Exit
Enter your choice: 4
Enter employee ID to delete: 5
Employee ID 5 deleted successfully.

--- Employee Database Operations ---
1. Add Employee
2. Display All Employees
3. Edit Employee
4. Delete Employee
5. Exit
Enter your choice: 2

Employees List:
ID: 1, Name: Sudhir, Age: 21, Department: AI&DS
ID: 2, Name: Rahul, Age: 23, Department: Computer Science
ID: 3, Name: Saishh, Age: 22, Department: AI&DS
ID: 4, Name: Ayyub, Age: 25, Department: E&TC

--- Employee Database Operations ---
1. Add Employee
2. Display All Employees
3. Edit Employee
4. Delete Employee
5. Exit
Enter your choice: 5
Exiting program.
PS C:\Users\sudhi\EmployeeDB> |
```

