Practical 10: JDBC

Name: Sutariya Savankumar

Roll no: MA065

1. **Develop a java application to accept employee information and store in Employee(empid, name, dept and salary) table. Also provide following functionality.**
   a. **Add new employee**
   b. **Update employee details (base on empid)**
   c. **Delete employee**
   d. **Display all employee**
   e. **Display maximum and minimum salary employee details**
   f. **Display employee information according to their department(Group by)**

Code

```
import java.sql.*;
public class EmployeeDBMS {
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost:3306/ma065";
    static final String USER = "root";
    static final String PASS = "savan";
    public static void main(String[] args) {
        Connection conn = null;
        Statement stmt = null;
        ResultSet rs = null;
        try {
            Class.forName(JDBC_DRIVER);
            conn = DriverManager.getConnection(DB_URL, USER, PASS);
            stmt = conn.createStatement();
       String sql = "CREATE TABLE IF NOT EXISTS Employee " +"(empid INTEGER not NULL, "
              +" name VARCHAR(255), " +" dept VARCHAR(255), "+" salary INTEGER, "
              + "PRIMARY KEY ( empid ))";
stmt.executeUpdate(sql);
stmt.close();
while (true) {
    System.out.println("\nEmployee Management System");
    System.out.println(" ---------------------- ");
    System.out.println("1. Add new employee");
    System.out.println("2. Update employee details");
    System.out.println("3. Delete employee");
    System.out.println("4. Display all employee");
    System.out.println("5. Display maximum and minimum salary employee details");
    System.out.println("6. Display employee information according to their
department(Group by)");
    System.out.println("0. Exit");
    int option = Integer.parseInt(System.console().readLine("Enter option: "));
    switch (option) {
        case 0:
            System.out.println("Goodbye!");
            System.exit(0);
            break;
        case 1:
            int empid = Integer.parseInt(System.console().readLine("Enter employee ID:
"));
            String name = System.console().readLine("Enter employee name: ");
            String dept = System.console().readLine("Enter employee department: ");
            intsalary = Integer.parseInt(System.console().readLine("Enter employee
salary: "));

            sql = "INSERT INTO Employee "+ "VALUES (" + empid + ", '" + name + "', '" +
dept + "', " + salary + ")";
            stmt = conn.createStatement();
            stmt.executeUpdate(sql);
            System.out.println("Employee added successfully!");
            stmt.close();
            break;
        case 2:
            empid = Integer.parseInt(System.console().readLine("Enter employee ID: "));
            name = System.console().readLine("Enter employee name: ");
            dept = System.console().readLine("Enter employee department: ");
            salary = Integer.parseInt(System.console().readLine("Enter employee salary:
"));
            sql = "UPDATE Employee " + "SET name = '" + name + "', dept = '" + dept +
"', salary = " + salary + "WHERE empid = " + empid;
            stmt = conn.createStatement();
            int rowsUpdated = pra stmt.executeUpdate(sql);
            if (rowsUpdated > 0) {
                System.out.println("Employee details updated successfully!");
            } else {
                System.out.println("No employee found with ID " + empid);}
            stmt.close();
            break;
        case 3:
            empid = Integer.parseInt(System.console().readLine("Enter employee ID: "));
            sql = "DELETE FROM Employee " + "WHERE empid = " + empid;
            stmt = conn.createStatement();
            rowsUpdated = stmt.executeUpdate(sql);
            if (rowsUpdated > 0) {
                System.out.println("Employee deleted successfully!");
            } else {
                System.out.println("No employee found with ID " + empid);}
            stmt.close();
            break;
        case 4:
            sql = "SELECT * FROM Employee";
            stmt = conn.createStatement();
            rs = stmt.executeQuery(sql);
            System.out.println("\nAll Employees:");
            System.out.println(" -------------- ");
            while (rs.next()) {
                System.out.println("ID: " + rs.getInt("empid") + ", Name: " +
rs.getString("name") + ", Department: " + rs.getString("dept") + ", Salary: " +
rs.getInt("salary"));
            }
            rs.close();
            stmt.close();
            break;
        case 5:
            sql = "SELECT * FROM Employee "+ "WHERE salary = (SELECT MAX(salary) FROM
Employee)";
            stmt = conn.createStatement();
            rs = stmt.executeQuery(sql);
            System.out.println("\nEmployee with Maximum Salary:");
            System.out.println(" -------------- ");
            while (rs.next()) {
                System.out.println("ID: " + rs.getInt("empid") + ", Name: " +
rs.getString("name") + ", Department: " + rs.getString("dept") + ", Salary: " +
rs.getInt("salary"));
            }
            rs.close();
            sql = "SELECT * FROM Employee "+ "WHERE salary = (SELECT MIN(salary) FROM
Employee)";
            stmt = conn.createStatement();
            rs = stmt.executeQuery(sql);
            System.out.println("\nEmployee with Minimum Salary:");
            System.out.println(" -------------- ");
            while (rs.next()) {
```

```
        System.out.println("ID: " + rs.getInt("empid") + ", Name: " + 
rs.getString("name") + ", Department: " + rs.getString("dept") + ", Salary: " + 
rs.getInt("salary"));
      }
      rs.close();
      stmt.close();
      break;
    case 6:
      sql = "SELECT dept, COUNT(*) AS count, AVG(salary) AS average_salary FROM 
Employee GROUP BY dept ";
      stmt = conn.createStatement(); rs = stmt.executeQuery(sql);
      System.out.println("\nEmployee Information Grouped by Department:");
      System.out.println(" -------------- ");
      while (rs.next()) {
        System.out.println("Department: " + rs.getString("dept") + ", Number of 
Employees: " + rs.getInt("count") + ", Average Salary: " + 
rs.getInt("average_salary"));
      }
      rs.close();
      stmt.close();
      break;
    default:
      System.out.println("Invalid option. Please try again.");
      break;
  }
}
    } catch (SQLException se) {
      se.printStackTrace();
    } catch (Exception e) {
      e.printStackTrace();
    } finally {
      try {
        if (stmt != null)
          stmt.close();
      } catch (SQLException se2) {
      }
      try {
        if (conn != null)
          conn.close();
      } catch (SQLException se) {
        se.printStackTrace();
      }
    }
  }
}
```

Output

```
Employee Management System
 ----------------------
1. Add new employee
2. Update employee details
3. Delete employee
4. Display all employee
5. Display maximum and minimum salary employee details
6. Display employee information according to their department(Group by)
0. Exit
Enter option: 1
Enter employee ID: 1
Enter employee name: A
Enter employee department: IT
Enter employee salary: 10000
Employee added successfully!

Employee Management System
 ----------------------
1. Add new employee
2. Update employee details
3. Delete employee
4. Display all employee
5. Display maximum and minimum salary employee details
6. Display employee information according to their department(Group by)
0. Exit
Enter option: 1
Enter employee ID: 2
Enter employee name: B
Enter employee department: IT
Enter employee salary: 20000
Employee added successfully!

Employee Management System
 ----------------------
1. Add new employee
2. Update employee details
3. Delete employee
4. Display all employee
5. Display maximum and minimum salary employee details
6. Display employee information according to their department(Group by)
0. Exit
Enter option: 1
Enter employee ID: 3
Enter employee name: C
Enter employee department: HR
Enter employee salary: 30000
Employee added successfully!

Employee Management System
 ----------------------
1. Add new employee
2. Update employee details
3. Delete employee
4. Display all employee
5. Display maximum and minimum salary employee details
6. Display employee information according to their department(Group by)
0. Exit
Enter option: 4

All Employees:
 --------------
ID: 1, Name: A, Department: IT, Salary: 10000
ID: 2, Name: B, Department: IT, Salary: 20000
ID: 3, Name: C, Department: HR, Salary: 30000

Employee Management System
 ----------------------
1. Add new employee
2. Update employee details
3. Delete employee
4. Display all employee
5. Display maximum and minimum salary employee details
6. Display employee information according to their department(Group by)
0. Exit
Enter option: 0
Exiting...
```