Assignment 8 – 31481

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

Solution-

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

public class MySQLDatabaseOperations {

  private static final String DB_URL = "jdbc:mysql://localhost:3306/te31481_db";
  private static final String DB_USER = "root";
  private static final String DB_PASSWORD = "te31481";

  public void insertRecord(String name, String department, double salary) {
    try (Connection connection = DriverManager.getConnection(DB_URL, DB_USER,
DB_PASSWORD)) {
      String query = "INSERT INTO employees (name, department, salary) VALUES (?, ?, ?)";
      PreparedStatement preparedStatement = connection.prepareStatement(query);
      preparedStatement.setString(1, name);
      preparedStatement.setString(2, department);
      preparedStatement.setDouble(3, salary);
      int rowsInserted = preparedStatement.executeUpdate();
      if (rowsInserted > 0) {
        System.out.println("A new employee was inserted successfully!");
      }
    } catch (SQLException e) {
      e.printStackTrace();
    }
  }

  public void updateRecord(int id, String department, double salary) {
```

```
    try (Connection connection = DriverManager.getConnection(DB_URL, DB_USER,
DB_PASSWORD)) {
        String query = "UPDATE employees SET department = ?, salary = ? WHERE id = ?";
        PreparedStatement preparedStatement = connection.prepareStatement(query);
        preparedStatement.setString(1, department);
        preparedStatement.setDouble(2, salary);
        preparedStatement.setInt(3, id);
        int rowsUpdated = preparedStatement.executeUpdate();
        if (rowsUpdated > 0) {
            System.out.println("Employee's record updated successfully!");
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
  }

  public void deleteRecord(int id) {
    try (Connection connection = DriverManager.getConnection(DB_URL, DB_USER,
DB_PASSWORD)) {
        String query = "DELETE FROM employees WHERE id = ?";
        PreparedStatement preparedStatement = connection.prepareStatement(query);
        preparedStatement.setInt(1, id);
        int rowsDeleted = preparedStatement.executeUpdate();
        if (rowsDeleted > 0) {
            System.out.println("Employee's record deleted successfully!");
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
  }

  public void viewRecords() {
```

```
    try (Connection connection = DriverManager.getConnection(DB_URL, DB_USER,
DB_PASSWORD)) {
      String query = "SELECT * FROM employees";
      Statement statement = connection.createStatement();
      ResultSet resultSet = statement.executeQuery(query);
      while (resultSet.next()) {
        System.out.println("ID: " + resultSet.getInt("id"));
        System.out.println("Name: " + resultSet.getString("name"));
        System.out.println("Department: " + resultSet.getString("department"));
        System.out.println("Salary: " + resultSet.getDouble("salary"));
        System.out.println("------------------------------");
      }
    } catch (SQLException e) {
      e.printStackTrace();
    }
  }

  public static void main(String[] args) {
    MySQLDatabaseOperations dbOperations = new MySQLDatabaseOperations();
    Scanner scanner = new Scanner(System.in);
    int choice = -1;

    while (choice != 5) {
      try {
        System.out.println("\nMenu:");
        System.out.println("1. Insert a record");
        System.out.println("2. Update a record");
        System.out.println("3. Delete a record");
        System.out.println("4. View all records");
        System.out.println("5. Exit");
        System.out.print("Enter your choice: ");
```

```
if (scanner.hasNextInt()) {
    choice = scanner.nextInt();
} else {
    System.out.println("Invalid input. Please enter a number between 1 and 5.");
    scanner.next();
    continue;
}

switch (choice) {
    case 1:
        System.out.print("Enter employee name: ");
        scanner.nextLine();  // Consume newline
        String name = scanner.nextLine();
        System.out.print("Enter employee department: ");
        String department = scanner.nextLine();
        System.out.print("Enter employee salary: ");
        double salary = scanner.nextDouble();
        dbOperations.insertRecord(name, department, salary);
        break;

    case 2:
        System.out.print("Enter employee ID to update: ");
        int updateId = scanner.nextInt();
        scanner.nextLine();  // Consume newline
        System.out.print("Enter new department: ");
        String newDepartment = scanner.nextLine();
        System.out.print("Enter new salary: ");
        double newSalary = scanner.nextDouble();
        dbOperations.updateRecord(updateId, newDepartment, newSalary);
        break;
```

```
            case 3:
              System.out.print("Enter employee ID to delete: ");
              int deleteId = scanner.nextInt();
              dbOperations.deleteRecord(deleteId);
              break;

            case 4:
              dbOperations.viewRecords();
              break;

            case 5:
              System.out.println("Exiting...");
              break;

            default:
              System.out.println("Invalid choice. Please select a number between 1 and 5.");
          }
        } catch (Exception e) {
          System.out.println("Error: " + e.getMessage());
          scanner.next();  // Consume the invalid input
        }
      }
      scanner.close();
    }
}
```

Sql create Table:

CREATE TABLE employees (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(50) NOT NULL, department VARCHAR(50) NOT NULL, salary DECIMAL(10, 2));