

## MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- Xây dựng ứng dụng CRUD với JDBC

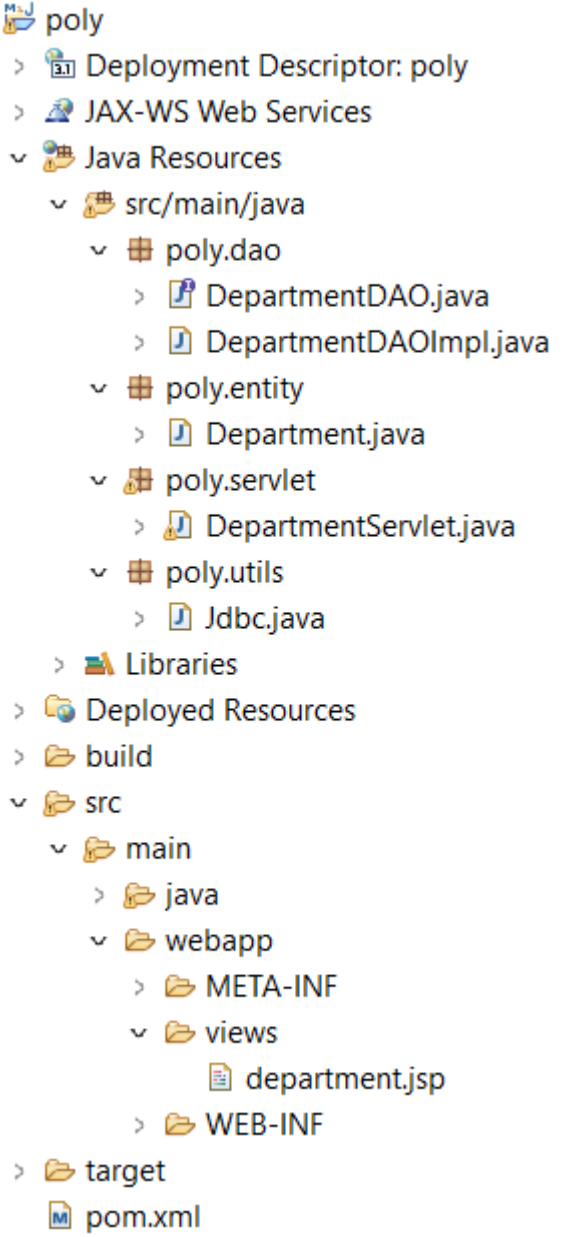
## YÊU CẦU

Xây dựng ứng dụng CRUD (xem, thêm, sửa, xóa) phòng ban Department sử dụng JDBC theo mô hình MVC như mô tả sau

Id: <input type="text"/> Name: <input type="text"/> Description: <input type="text"/> <div> <input type="button" value="Create"/> <input type="button" value="Update"/> <input type="button" value="Delete"/> <input type="button" value="Reset"/> </div> <table border="1"> <thead> <tr> <th>No.</th> <th>Id</th> <th>Name</th> <th>Description</th> <th></th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> <td><a href="#">Edit</a></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>					No.	Id	Name	Description						<a href="#">Edit</a>																<table border="1"> <thead> <tr> <th>Sự kiện</th> <th>Điều khiển sự kiện</th> </tr> </thead> <tbody> <tr> <td>Page.Open</td> <td> <ul style="list-style-type: none"> <li>• Hiện thị bảng</li> <li>• Xóa trắng form</li> </ul> </td> </tr> <tr> <td>Create.Click</td> <td> <ul style="list-style-type: none"> <li>• Thêm mới</li> <li>• Hiện thị lại bảng</li> <li>• Xóa trắng form</li> </ul> </td> </tr> <tr> <td>Update.Click</td> <td> <ul style="list-style-type: none"> <li>• Cập nhật</li> <li>• Hiện thị lại bảng</li> <li>• Giữ nguyên dữ liệu trên form</li> </ul> </td> </tr> <tr> <td>Delete.Click</td> <td> <ul style="list-style-type: none"> <li>• Xóa mục theo id trên form</li> <li>• Hiện thị lại bảng</li> <li>• Xóa trắng form</li> </ul> </td> </tr> <tr> <td>Reset.Click</td> <td> <ul style="list-style-type: none"> <li>• Hiện thị lại bảng</li> <li>• Xóa trắng form</li> </ul> </td> </tr> <tr> <td>Edit.Click</td> <td> <ul style="list-style-type: none"> <li>• Hiện thị mục chọn lên form</li> <li>• Hiện thị lại bảng</li> </ul> </td> </tr> </tbody> </table>	Sự kiện	Điều khiển sự kiện	Page.Open	<ul style="list-style-type: none"> <li>• Hiện thị bảng</li> <li>• Xóa trắng form</li> </ul>	Create.Click	<ul style="list-style-type: none"> <li>• Thêm mới</li> <li>• Hiện thị lại bảng</li> <li>• Xóa trắng form</li> </ul>	Update.Click	<ul style="list-style-type: none"> <li>• Cập nhật</li> <li>• Hiện thị lại bảng</li> <li>• Giữ nguyên dữ liệu trên form</li> </ul>	Delete.Click	<ul style="list-style-type: none"> <li>• Xóa mục theo id trên form</li> <li>• Hiện thị lại bảng</li> <li>• Xóa trắng form</li> </ul>	Reset.Click	<ul style="list-style-type: none"> <li>• Hiện thị lại bảng</li> <li>• Xóa trắng form</li> </ul>	Edit.Click	<ul style="list-style-type: none"> <li>• Hiện thị mục chọn lên form</li> <li>• Hiện thị lại bảng</li> </ul>
No.	Id	Name	Description																																									
				<a href="#">Edit</a>																																								
Sự kiện	Điều khiển sự kiện																																											
Page.Open	<ul style="list-style-type: none"> <li>• Hiện thị bảng</li> <li>• Xóa trắng form</li> </ul>																																											
Create.Click	<ul style="list-style-type: none"> <li>• Thêm mới</li> <li>• Hiện thị lại bảng</li> <li>• Xóa trắng form</li> </ul>																																											
Update.Click	<ul style="list-style-type: none"> <li>• Cập nhật</li> <li>• Hiện thị lại bảng</li> <li>• Giữ nguyên dữ liệu trên form</li> </ul>																																											
Delete.Click	<ul style="list-style-type: none"> <li>• Xóa mục theo id trên form</li> <li>• Hiện thị lại bảng</li> <li>• Xóa trắng form</li> </ul>																																											
Reset.Click	<ul style="list-style-type: none"> <li>• Hiện thị lại bảng</li> <li>• Xóa trắng form</li> </ul>																																											
Edit.Click	<ul style="list-style-type: none"> <li>• Hiện thị mục chọn lên form</li> <li>• Hiện thị lại bảng</li> </ul>																																											

## PHẦN I: TỔ CHỨC DỰ ÁN VÀ XÂY DỰNG DAO

### BÀI 1: TỔ CHỨC VÀ CẤU HÌNH DỰ ÁN

<p>Tạo dự án web động bao gồm các thành phần được tổ chức như hình bên.</p> <ul style="list-style-type: none"> <li>✓ Pom.xml: khai báo các thư viện phụ thuộc (JSTL, SQL Server Driver, BeanUtils)</li> <li>✓ Department.jsp: giao diện ứng dụng</li> <li>✓ DepartmentServlet: Servlet điều khiển request</li> <li>✓ Department: Java Bean mô tả cấu trúc dữ liệu phòng</li> <li>✓ DepartmentDAO và DepartmentDAOImpl: interface và class cung cấp các phương thức truy vấn và thao tác dữ liệu bảng Departments</li> <li>✓ Jdbc: thư viện tiện ích cung cấp các phương thức truy vấn và thao tác dữ liệu chung chung (làm việc với SQL)</li> </ul>	
---	---

Cấu hình pom.xml

```
<dependency>
  <groupId>com.microsoft.sqlserver</groupId>
  <artifactId>mssql-jdbc</artifactId>
  <version>12.8.0.jre8</version>
</dependency>
<dependency>
```

```
<groupId>commons-beanutils</groupId>
<artifactId>commons-beanutils</artifactId>
<version>1.9.3</version>
</dependency>
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>jstl</artifactId>
  <version>1.2</version>
</dependency>
```

## BÀI 2: XÂY DỰNG JDBC VÀ DAO

### 2.1 Xây dựng lớp tiện ích Jdbc cho phép làm việc với SQL hoặc lời gọi PROC.

```
public class Jdbc {
    static String driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";
    static String dburl = "jdbc:sqlserver://localhost;database=HRM";
    static String username = "sa";
    static String password = "123456";

    static {
        try { // nạp driver
            Class.forName(driver);
        } catch (ClassNotFoundException e) {
            throw new RuntimeException(e);
        }
    }

    /**Mở kết nối*/
    public static Connection getConnection() throws SQLException {
        return DriverManager.getConnection(dburl, username, password);
    }

    /**Thao tác dữ liệu*/
    public static int executeUpdate(String sql, Object... values) throws SQLException {
        PreparedStatement stmt = Jdbc.createPreStmt(sql, values);
        return stmt.executeUpdate();
    }

    /**Truy vấn dữ liệu*/
    public static ResultSet executeQuery(String sql, Object... values) throws
    SQLException {
        PreparedStatement stmt = Jdbc.createPreStmt(sql, values);
        return stmt.executeQuery();
    }

    /**Tạo PreparedStatement làm việc với SQL hoặc PROC*/
    private static PreparedStatement createPreStmt(String sql, Object... values) throws
    SQLException {
```

```

        Connection connection = Jdbc.getConnection();
        PreparedStatement stmt = null;
        if(sql.trim().startsWith("{")) {
            stmt = connection.prepareCall(sql);
        }
        stmt = connection.prepareStatement(sql);
        for (int i = 0; i < values.length; i++) {
            stmt.setObject(i + 1, values[i]);
        }
        return stmt;
    }
}

```

2.2 Xây dựng interface DepartmentDAO khai báo các phương thức truy xuất dữ liệu bảng Department.

```

public interface DepartmentDAO {
    /**Truy vấn tất cả*/
    List<Department> findAll();
    /**Truy vấn theo mã*/
    Department findById(String id);
    /**Thêm mới*/
    void create(Department item);
    /**Cập nhật*/
    void update(Department item);
    /**Xóa theo mã*/
    void deleteById(String id);
}

```

2.3 Tạo lớp DepartmentDAOImpl hiện thực hóa các phương thức khai báo trong DepartmentDAO.

```

public class DepartmentDAOImpl implements DepartmentDAO {
    @Override
    public List<Department> findAll() {
        String sql = "SELECT * FROM Departments";
        try {
            List<Department> entities = new ArrayList<>();
            Object[] values = {};
            ResultSet resultSet = Jdbc.executeQuery(sql, values);
            while (resultSet.next()) {
                Department entity = new Department();
                entity.setId(resultSet.getString("Id"));
                entity.setName(resultSet.getString("Name"));
                entity.setDescription(resultSet.getString("Description"));
            }
        }
    }
}

```

```

        entities.add(entity);
    }
    return entities;
} catch (Exception e) {
    throw new RuntimeException(e);
}
}

@Override
public Department findById(String id) {
    String sql = "SELECT * FROM Departments WHERE Id=?";
    try {
        Object[] values = { id };
        ResultSet resultSet = Jdbc.executeQuery(sql, values);
        if (resultSet.next()) {
            Department entity = new Department();
            entity.setId(resultSet.getString("Id"));
            entity.setName(resultSet.getString("Name"));
            entity.setDescription(resultSet.getString("Description"));
            return entity;
        }
    } catch (Exception e) {
        throw new RuntimeException(e);
    }
    throw new RuntimeException("Not found");
}

@Override
public void create(Department entity) {
    String sql = "INSERT INTO Departments(Id, Name, Description) VALUES(?, ?, ?)";
    try {
        Object[] values = {
            entity.getId(),
            entity.getName(),
            entity.getDescription()
        };
        Jdbc.executeUpdate(sql, values);
    } catch (Exception e) {
        throw new RuntimeException(e);
    }
}

@Override
public void update(Department entity) {

```

```
String sql = "UPDATE Departments SET Name=?, Description=? WHERE
Id=?";

    try {
        Object[] values = {
            entity.getName(),
            entity.getDescription(),
            entity.getId()
        };
        jdbc.executeUpdate(sql, values);
    } catch (Exception e) {
        throw new RuntimeException(e);
    }
}

@Override
public void deleteById(String id) {
    String sql = "DELETE FROM Departments WHERE Id=?";
    try {
        Object[] values = { id };
        jdbc.executeUpdate(sql, values);
    } catch (Exception e) {
        throw new RuntimeException(e);
    }
}
}
```

## PHẦN II: XÂY DỰNG ỨNG DỤNG CRUD

### BÀI 3: XÂY DỰNG GIAO DIỆN

Viết mã cho trang department.jsp như sau:

```
<%@ page pageEncoding="utf-8"%>
<%@ taglib uri="http://java.sun.com/jstl/fmt_rt" prefix="fmt"%>
<%@ taglib uri="http://java.sun.com/jstl/core_rt" prefix="c"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Department Management</title>
</head>
<body>
    <c:url var="path" value="/department" />
    <!-- FORM -->
    <form method="post">
```

```

        <label>Id:</label><br>
        <input name="id" value="${item.id}"><br>
        <label>Name:</label><br>
        <input name="name" value="${item.name}"><br>
        <label>Description:</label><br>
        <textarea name="description" rows="3">${item.description}</textarea>
        <hr>
        <button formaction="${path}/create">Create</button>
        <button formaction="${path}/update">Update</button>
        <button formaction="${path}/delete">Delete</button>
        <button formaction="${path}/reset">Reset</button>
    </form>
    <hr>
    <!-- TABLE -->
    <table border="1" style="width: 100%">
        <thead>
            <tr>
                <th>No.</th>
                <th>Id</th>
                <th>Name</th>
                <th>Description</th>
                <th></th>
            </tr>
        </thead>
        <tbody>
            <c:forEach var="d" items="${list}" varStatus="vs">
                <tr>
                    <td>${vs.count}</td>
                    <td>${d.id}</td>
                    <td>${d.name}</td>
                    <td>${d.description}</td>
                    <td><a href="${path}/edit/${d.id}">Edit</a></td>
                </tr>
            </c:forEach>
        </tbody>
    </table>
</body>
</html>

```

#### BÀI 4: XÂY DỰNG SERVLET

Viết mã cho servlet DepartmentServlet như sau

```

@WebServlet({
    "/department/index",
    "/department/edit/*",

```

```
        "/department/create",
        "/department/update",
        "/department/delete",
        "/department/reset"
    })
    public class DepartmentServlet extends HttpServlet {
        @Override
        protected void service(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
            Department form = new Department();
            try {
                BeanUtils.populate(form, req.getParameterMap());
            } catch (IllegalAccessException | InvocationTargetException e) {
                e.printStackTrace();
            }
            DepartmentDAO dao = new DepartmentDAOImpl();
            String path = req.getServletPath();
            if (path.contains("edit")) {
                String id = req.getPathInfo().substring(1);
                form = dao.findById(id);
            } else if (path.contains("create")) {
                dao.create(form);
                form = new Department();
            } else if (path.contains("update")) {
                dao.update(form);
            } else if (path.contains("delete")) {
                dao.deleteById(form.getId());
                form = new Department();
            } else {
                form = new Department();
            }
            req.setAttribute("item", form);
            List<Department> list = dao.findAll();
            req.setAttribute("list", list);
            req.getRequestDispatcher("/department/index.jsp").forward(req, resp);
        }
    }
}
```

### BÀI 5: ỨNG DỤNG CRUD NÂNG CAO

Xây dựng ứng dụng CRUD để quản lý bảng Employees trong CSDL HRM.

Chú ý:

- ✓ Kiểu dữ liệu của các cột
- ✓ Upload file hình



- ✓ Khóa ngoại (đổ dữ liệu vào Select Box)