



FPT POLYTECHNIC



Conceive Design Implement Operate

ỨNG DỤNG MÔ HÌNH CRUD VÀ JDBC

LẬP TRÌNH JAVA #3 (P7.2)

www.poly.edu.vn

- ☐ Xây dựng thư viện tiện ích Jdbc
- ☐ Thiết kế và xây dựng DAO
- ☐ Tích hợp mã DAO vào Servlet



```

public class Jdbc {
    private static String driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";
    private static String dburl = "jdbc:sqlserver://localhost;database=HRM";
    private static String username = "sa";
    private static String password = "*****";
    static {
        try {
            Class.forName(driver); // Nạp 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 {...}
    /**Truy vấn dữ liệu*/
    public static ResultSet executeQuery(String sql, Object... values) throws SQLException {...}
}

```

```

String sql = "INSERT|UPDATE|DELETE...?...";
Object[] values = {...};
int rows = Jdbc.executeUpdate(sql, values);

String sql = "SELECT...?...";
Object[] values = {...};
ResultSet rs = Jdbc.executeQuery(sql, values);

```

/**Thao tác dữ liệu*/

```
public static int executeUpdate(String sql, Object... values) throws SQLException {  
    Connection connection = getConnection();  
    PreparedStatement statement = connection.prepareStatement(sql);  
    for (int i = 0; i < values.length; i++) {  
        statement.setObject(i + 1, values[i]);  
    }  
    return statement.executeUpdate();  
}
```

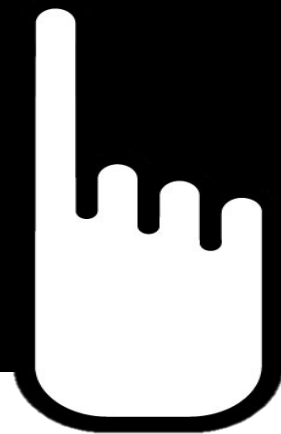
/**Truy vấn dữ liệu*/

```
public static ResultSet executeQuery(String sql, Object... values) throws SQLException {  
    Connection connection = getConnection();  
    PreparedStatement statement = connection.prepareStatement(sql);  
    for (int i = 0; i < values.length; i++) {  
        statement.setObject(i + 1, values[i]);  
    }  
    return statement.executeQuery();  
}
```



DEMO

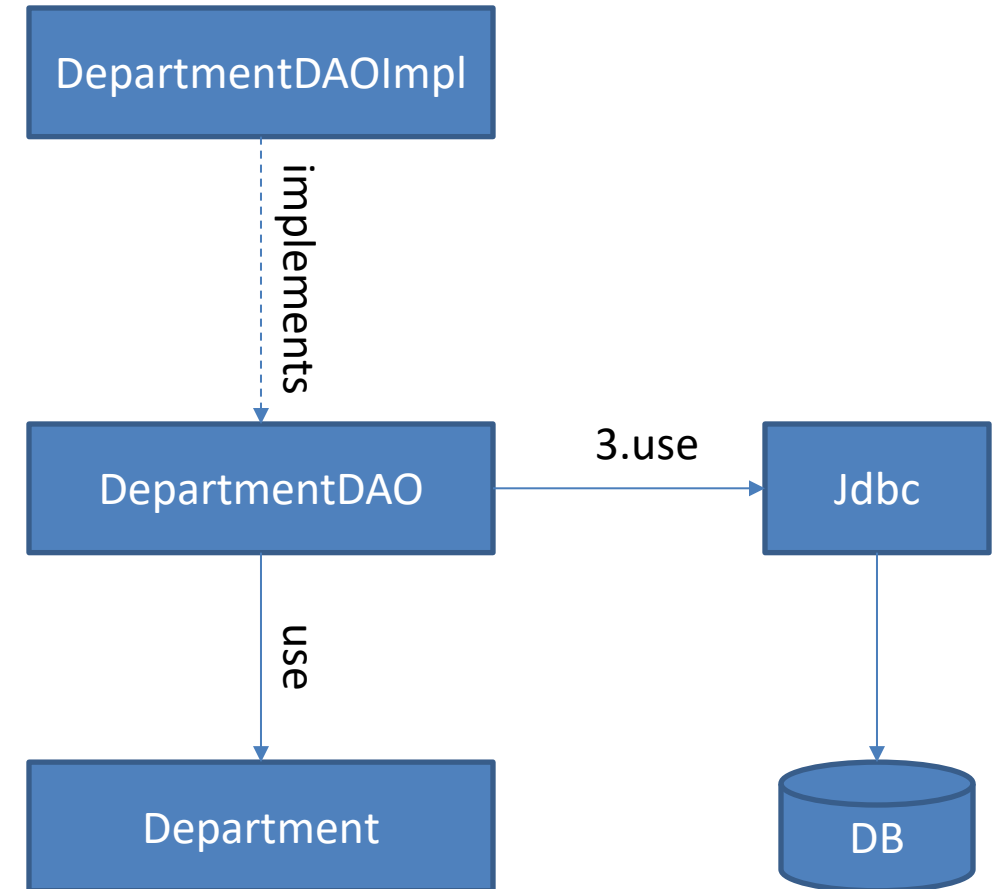
HOÀN THIỆN VÀ TEST JDBC VỚI
DEPARTMENTS/EMPLOYEES



INTERFACE DEPARTMENTDAO

```
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);  
}
```

- ❑ Khai báo các phương thức cần thiết làm việc với bảng Departments



```

public class DepartmentDAOImpl implements DepartmentDAO {
    @Override
    public List<Department> findAll() {
        String sql = "SELECT * FROM Departments";
    }
    @Override
    public Department findById(String id) {
        String sql = "SELECT * FROM Departments WHERE Id=?";
    }
    @Override
    public void create(Department entity) {
        String sql = "INSERT INTO Departments(Id, Name, Description) VALUES(?, ?, ?)";
    }
    @Override
    public void update(Department entity) {
        String sql = "UPDATE Departments SET Name=?, Description=? WHERE Id=?";
    }
    @Override
    public void deleteById(String id) {
        String sql = "DELETE FROM Departments WHERE Id=?";
    }
}

```

@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);  
    }  
}
```

- ❑ Bổ sung dòng code tạo đối tượng DAO sau đây vào đầu phương thức service()
 - ❖ `DepartmentDAO dao = new DepartmentDAOImpl();`
- ❑ Gọi các phương thức truy xuất dữ liệu của **dao** tại các vị trí phù hợp của service() theo hướng dẫn sau:

Path	Code
path.contains("create")	<code>dao.create(form);</code> <code>form = new Department();</code>
path.contains("update")	<code>dao.update(form);</code>
path.contains("delete")	<code>dao.deleteById(form.getId());</code> <code>form = new Department();</code>
path.contains("edit")	<code>form = dao.findById(id);</code>
new ArrayList<>()	<code>dao.findAll()</code>

```
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();
}
```



DEMO

HOÀN THIỆN VÀ CHẠY ỨNG DỤNG CRUD



- ✓ Xây dựng thư viện tiện ích Jdbc
- ✓ Thiết kế và xây dựng DAO
- ✓ Tích hợp mã DAO vào Servlet





Cảm ơn