



**FPT POLYTECHNIC**



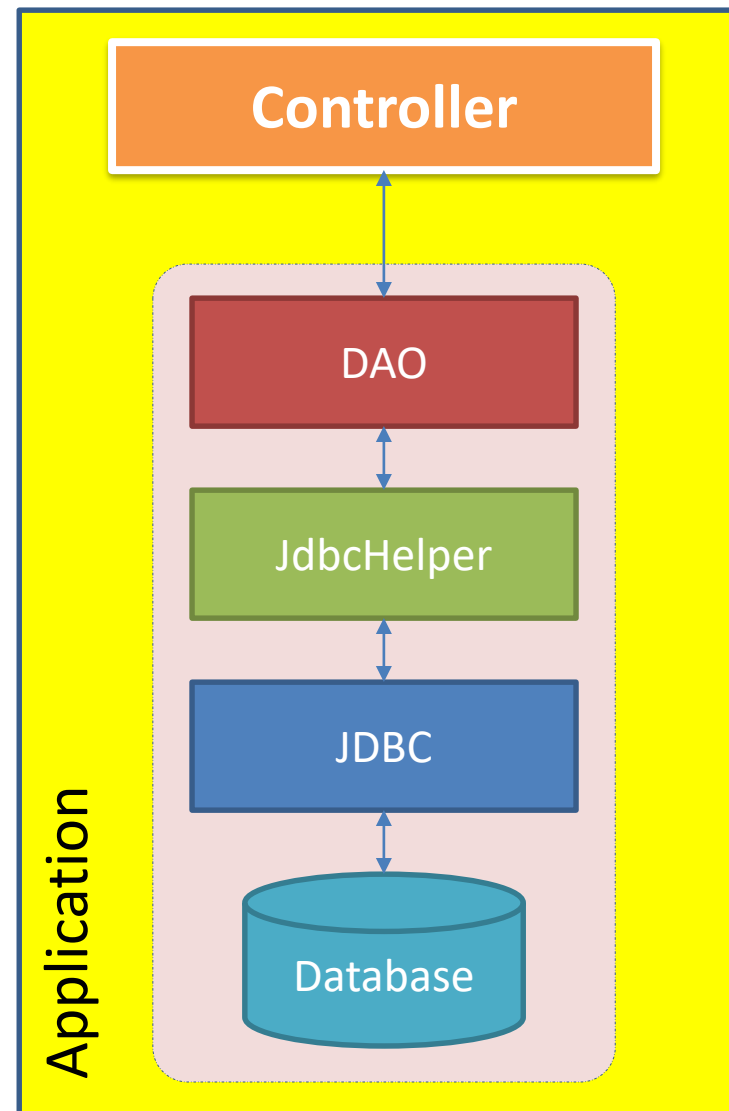
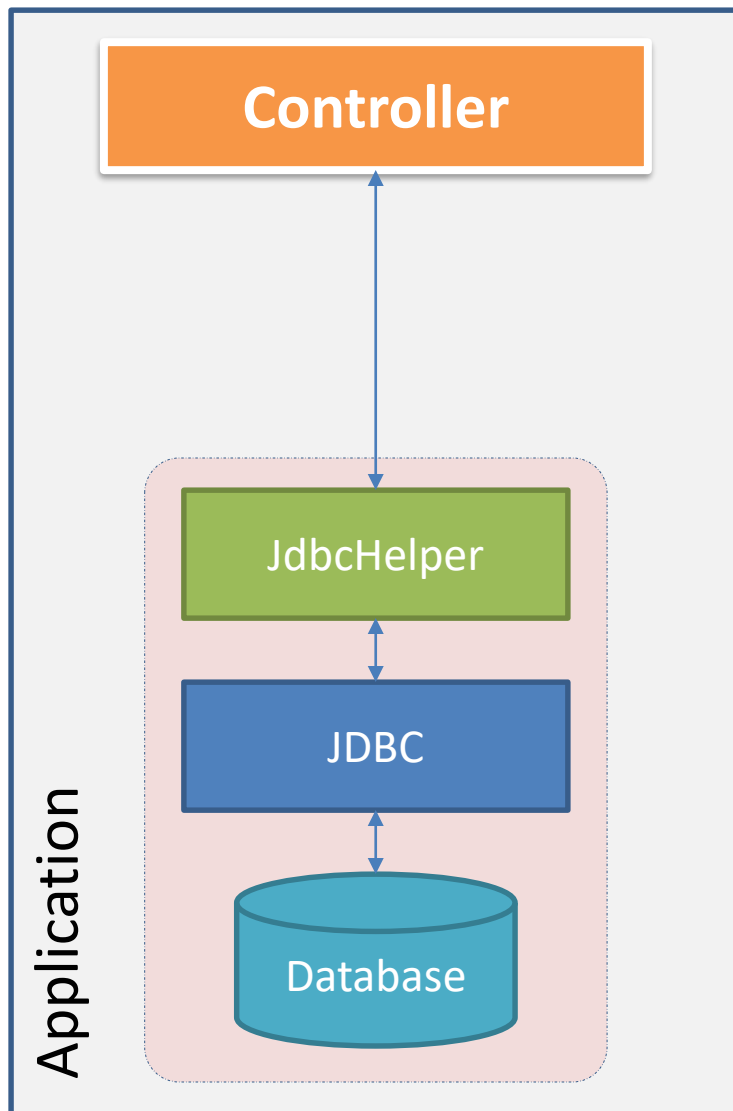
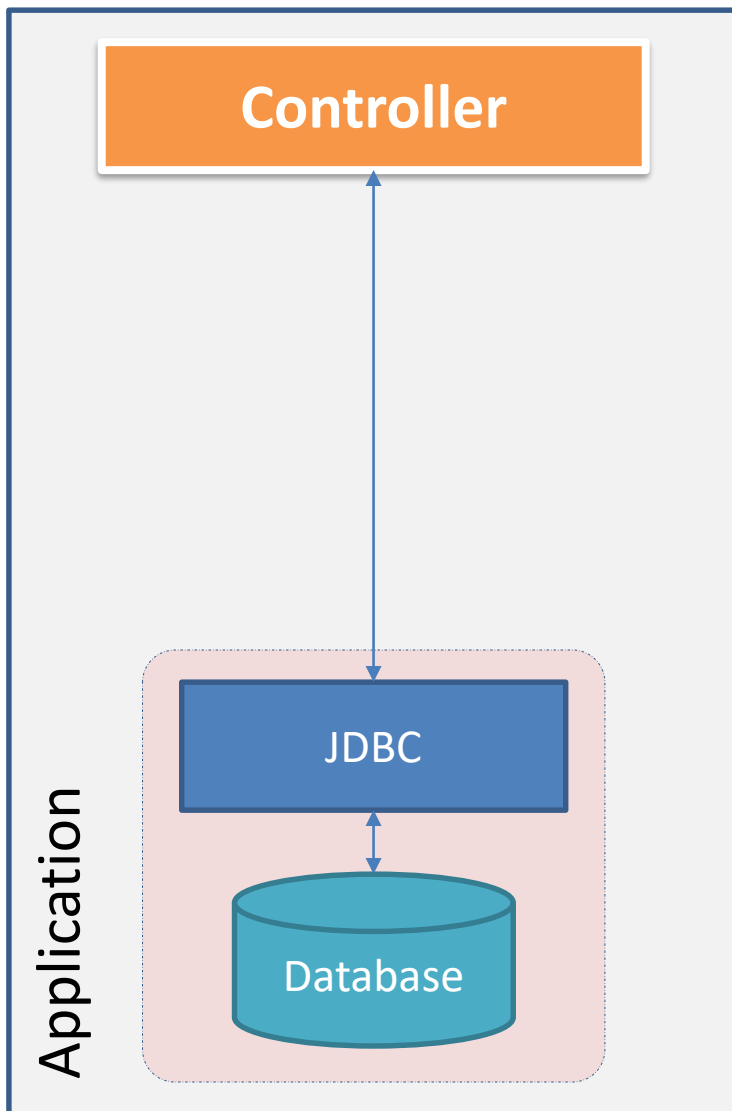
Conceive Design Implement Operate

## **LẬP TRÌNH CSDL**

**GIẢNG VIÊN: NGUYỄN NGHIỆM**

[www.poly.edu.vn](http://www.poly.edu.vn)

# MỨC ĐỘ TÁI SỬ DỤNG TRONG LẬP TRÌNH JDBC





**FPT POLYTECHNIC**



Conceive Design Implement Operate

## **ÁP DỤNG DAO MODEL**

**GIẢNG VIÊN: NGUYỄN NGHIỆM**

[www.poly.edu.vn](http://www.poly.edu.vn)

- ⊙ Kết thúc môn học này, bạn có khả năng
  - ⊙ Giải thích được mô hình DAO
  - ⊙ Xây dựng được các lớp thực thể (**entity** class)
  - ⊙ Tổ chức được sơ đồ phân cấp kế thừa DAO
  - ⊙ Xây dựng được các **DAO** làm việc với các bảng



# MÔ HÌNH QUẢN LÝ NHÂN VIÊN SỬ DỤNG DAO

QUẢN LÝ NHÂN VIÊN

QUẢN LÝ NHÂN VIÊN QUẢN TRỊ

CẬP NHẬT DANH SÁCH

Mã nhân viên

Mật khẩu

Xác nhận mật khẩu

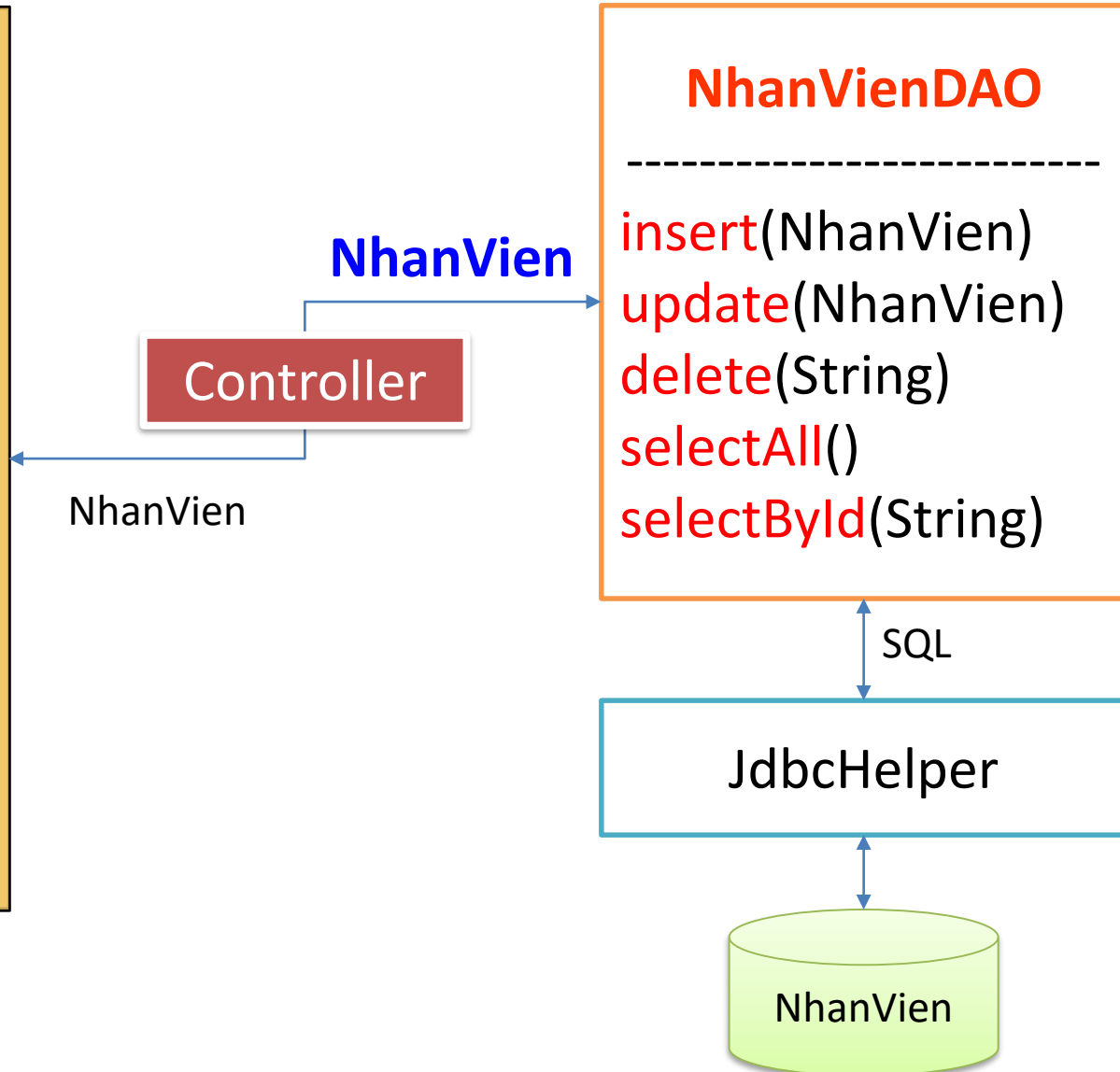
Họ và tên

Vai trò

☐ Trưởng phòng ☒ Nhân viên

Thêm Sửa Xóa Mới

|< << >> >|





# XÂY DỰNG ENTITY CLASS

---

# XÂY DỰNG ENTITY CLASS

	Column Name	Data Type
🔑	MaNH	nchar(7)
	HoTen	nvarchar(50)
	NgaySinh	date
	GioiTinh	bit
	DienThoai	nvarchar(50)
	Email	nvarchar(50)
	GhiChu	nvarchar(MAX)
	MaNV	nvar
	NgayDK	date

Entity Class

CHAR/VARCHAR/TEXT => **String**  
 DATE => **java.util.Date**  
 BIT => **Boolean/boolean**  
 FLOAT => **Double/double**  
 INT => **Integer/int**  
 BIGINT => **Long/long**  
 BINARY/VARBINARY => **byte[]**

NgnoiHoc

-----  
**String** maNH  
 String hoTen  
**Date** ngaySinh  
**boolean** GioiTinh  
 String dienThoai  
 String email  
 String ghiChu  
 String maNV  
 Date ngayDK

-----  
 getters/setters

-----  
 constructors

## ChuyenDe

---

String maCD  
String tenCD  
double hocPhi  
int thoiLuong  
String hinh  
String moTa

## KhoaHoc

---

int maKH  
String maCD  
double hocPhi  
int thoiLuong  
Date ngayKG  
String ghiChu  
String maNV  
Date ngayTao

## NgnoiHoc

---

String maNH  
String hoTen  
Date ngaySinh  
boolean GioiTinh  
String dienThoai  
String email  
String ghiChu  
String maNV  
Date ngayDK

## NhanVien

---

String maNV  
String hoTen  
String matKhau  
boolean vaiTro

## HocVien

---

int maHV  
int maKH  
String maNH  
double diem







XÂY DỰNG DAO

---

**EduSysDAO**<EntityType, KeyType>

-----

- + void **insert**(EntityType)
- + void **update**(EntityType)
- + void **delete**(KeyType)
- + List<EntityType> **selectAll**()
- + EntityType **selectById**(KeyType)
- List<EntityType> **selectBySql**(String, Object...)

Abstract  
Class

extends

**ChuyenDeDAO**

**KhoaHocDAO**

**NgnoiHocDAO**

**HocVienDAO**

**NhanVienDAO**

```
abstract public class EduSysDAO<EntityType, KeyType> {  
    abstract public void insert(EntityType entity);  
    abstract public void update(EntityType entity);  
    abstract public void delete(KeyType id);  
    abstract public EntityType selectById(KeyType id);  
    abstract public List<EntityType> selectAll();  
    abstract protected List<EntityType> selectBySql(String sql, Object...args);  
}
```

Ở mức này không biết được Entity và Key cụ thể nên tổng quát hóa (generic). Nó sẽ được cụ thể hóa ở lớp con

- ❑ Các *XYZDAO* con phải cài đặt mã cho các phương thức của *EduSysDAO*
- ❑ *selectBySql()* chỉ cho DAO con kế thừa mà không cho phép sử dụng bên ngoài

```
public class NhanVienDAO extends EduSysDAO<NhanVien, String> {  
    String INSERT_SQL = "INSERT INTO NhanVien(MaNV, MatKhai, HoTen, VaiTro) VALUES (?, ?, ?, ?)";  
    String UPDATE_SQL = "UPDATE NhanVien SET MatKhai = ?, HoTen = ?, VaiTro = ? WHERE MaNV = ?";  
    String DELETE_SQL = "DELETE FROM NhanVien WHERE MaNV = ?";  
    String SELECT_ALL_SQL = "SELECT * FROM NhanVien";  
    String SELECT_BY_ID_SQL = "SELECT * FROM NhanVien WHERE MaNV = ?";  
    @Override  
    public void insert(NhanVien entity) {...}  
    @Override  
    public void update(NhanVien entity) {...}  
    @Override  
    public void delete(String id) {...}  
    @Override  
    public NhanVien selectById(String id) {...}  
    @Override  
    public List<NhanVien> selectAll() {...}  
    @Override  
    protected List<NhanVien> selectBySql(String sql, Object... args) {...}  
}
```

**Cụ thể hóa:**

**EntityType = NhanVien**

**KeyType = String**

*Khai báo các câu lệnh SQL cần thiết  
được sử dụng trong các phương thức*

```
@Override
```

```
public void insert(NhanVien entity) {  
    JdbcHelper.update(INSERT_SQL,  
        entity.getMaNV(), entity.getHoTen(), entity.getMatKhau(), entity.isVaiTro());  
}
```

```
@Override
```

```
public void update(NhanVien entity) {  
    JdbcHelper.update(UPDATE_SQL,  
        entity.getHoTen(), entity.getMatKhau(), entity.isVaiTro(), entity.getMaNV());  
}
```

```
@Override
```

```
public void delete(String id) {  
    JdbcHelper.update(DELETE_SQL, id);  
}
```

@Override

```
protected List<NhanVien> selectBySql(String sql, Object... args) {  
    List<NhanVien> list = new ArrayList<NhanVien>();  
    try {  
        ResultSet rs = JdbcHelper.query(sql, args);  
        while(rs.next()) {  
            NhanVien entity = new NhanVien();  
            entity.setMaNV(rs.getString("MaNV"));  
            entity.setHoTen(rs.getString("HoTen"));  
            entity.setMatKhau(rs.getString("MatKhau"));  
            entity.setVaiTro(rs.getBoolean("VaiTro"));  
            list.add(entity);  
        }  
        rs.getStatement().getConnection().close();  
        return list;  
    } catch (Exception e) {  
        throw new RuntimeException(e);  
    }  
}
```

@Override

```
public NhanVien selectById(String id) {  
    List<NhanVien> list = this.selectById(SELECT_BY_ID_SQL, id);  
    if(list.isEmpty()) {  
        return null;  
    }  
    return list.get(0);  
}
```

@Override

```
public List<NhanVien> selectAll() {  
    return this.selectAll(SELECT_ALL_SQL);  
}
```

@Override

```
protected List<NhanVien> selectBySql(String sql, Object... args) {...}
```







XÂY DỰNG THÔNG KÊ DAO

---

```
CREATE PROC sp_BangDiem(@MaKH INT)
AS BEGIN
    SELECT                                {CALL sp_BangDiem(?)}
        nh.MaNH,
        nh.HoTen,
        hv.Diem
    FROM HocVien hv
        JOIN NguoiHoc nh ON nh.MaNH=hv.MaNH
    WHERE hv.MaKH = @MaKH
    ORDER BY hv.Diem DESC
END
```

```
CREATE PROC sp_DiemChuyenDe
AS BEGIN
    SELECT                                {CALL sp_DiemChuyenDe}
        TenCD ChuyenDe,
        COUNT(MaHV) SoHV,
        MIN(Diem) ThapNhat,
        MAX(Diem) CaoNhat,
        AVG(Diem) TrungBinh
    FROM KhoaHoc kh
        JOIN HocVien hv ON kh.MaKH=hv.MaKH
        JOIN ChuyenDe cd ON cd.MaCD=kh.MaCD
    GROUP BY TenCD
END
```

```
CREATE PROC sp_LuongNguoiHoc
AS BEGIN
    SELECT                                {CALL sp_LuongNguoiHoc}
        YEAR(NgayDK) Nam,
        COUNT(*) SoLuong,
        MIN(NgayDK) DauTien,
        MAX(NgayDK) CuoiCung
    FROM NguoiHoc
    GROUP BY YEAR(NgayDK)
END
```

```
CREATE PROC sp_DoanhThu(@Year INT)
AS BEGIN
    SELECT                                {CALL sp_DoanhThu}
        TenCD ChuyenDe,
        COUNT(DISTINCT kh.MaKH) SoKH,
        COUNT(hv.MaHV) SoHV,
        SUM(kh.HocPhi) DoanhThu,
        MIN(kh.HocPhi) ThapNhat,
        MAX(kh.HocPhi) CaoNhat,
        AVG(kh.HocPhi) TrungBinh
    FROM KhoaHoc kh
        JOIN HocVien hv ON kh.MaKH=hv.MaKH
        JOIN ChuyenDe cd ON cd.MaCD=kh.MaCD
    WHERE YEAR(NgayKG) = @Year
    GROUP BY TenCD
END
```

```
public class ThongKeDAO {  
    private List<Object[]> getListOfArray(String sql, String[] cols, Object...args) {...17 lines }  
  
    public List<Object[]> getBangDiem(Integer makh) {...5 lines }  
  
    public List<Object[]> getLuongNguoiHoc() {...5 lines }  
  
    public List<Object[]> getDiemChuyenDe() {...5 lines }  
  
    public List<Object[]> getDoanhThu(int nam) {...5 lines }  
}
```

- ❑ ***getListOfArray()*** chỉ được phục vụ cho các phương thức còn lại. Nó dựa vào tên PROC, mảng các cột để đọc dữ liệu trên mỗi bản ghi.

```
public List<Object[]> getBangDiem(Integer makh){
    String sql = "{CALL sp_BangDiem (?)}";
    String[] cols = {"MaNH", "HoTen", "Diem"};
    return this.getListOfArray(sql, cols, makh);
}

public List<Object[]> getLuongNguoiHoc(){
    String sql = "{CALL sp_LuongNguoiHoc}";
    String[] cols = {"Nam", "SoLuong", "DauTien", "CuoiCung"};
    return this.getListOfArray(sql, cols);
}

public List<Object[]> getDiemChuyenDe(){
    String sql = "{CALL sp_DiemChuyenDe}";
    String[] cols = {"ChuyenDe", "SoHV", "ThapNhat", "CaoNhat", "TrungBinh"};
    return this.getListOfArray(sql, cols);
}

public List<Object[]> getDoanhThu(int nam){
    String sql = "{CALL sp_DoanhThu (?)}";
    String[] cols = {"ChuyenDe", "SoKH", "SoHV", "DoanhThu", "ThapNhat", "CaoNhat", "TrungBinh"};
    return this.getListOfArray(sql, cols, nam);
}
```

## PHƯƠNG THỨC GETLISTOFARRAY()

```
private List<Object[]> getListOfArray(String sql, String[] cols, Object...args){
```

```
    try {
```

```
        List<Object[]> list=new ArrayList<>();
```

```
        ResultSet rs = XJdbc.query(sql, args);
```

```
        while(rs.next()){
```

```
            Object[] vals = new Object[cols.length];
```

```
            for(int i=0; i<cols.length; i++){
```

```
                vals[i] = rs.getObject(cols[i]);
```

```
            }
```

```
            list.add(vals);
```

```
        }
```

```
        rs.getStatement().getConnection().close();
```

```
        return list;
```

```
    } catch (Exception e) {
```

```
        throw new RuntimeException(e);
```

```
    }
```

```
}
```

{CALL **sp\_BangDiem** (?)};

{"MaNH", "HoTen", "Diem"}

**105**

{CALL sp\_BangDiem (?)}

{"Nam", "SoLuong", "DauTien", "CuoiCung"}



- ☑ Mô hình DAO
- ☑ Xây dựng Entity Class
- ☑ Phân cấp thừa kế DAO
- ☑ Xây dựng EduSysDAO
- ☑ Xây dựng ThongKeDAO







**Cảm ơn**