

**BỘ CÔNG THƯƠNG**  
**TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG THÀNH PHỐ HCM**  
**KHOA CÔNG NGHỆ THÔNG TIN**

---



**QUẢN LÝ SINH VIÊN**

**BÁO CÁO ĐỒ ÁN MÔN HỌC**  
**HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**SINH VIÊN THỰC HIỆN:**

1. 2001230044, Nguyễn Gia Bảo, Lớp 14DHTH07
2. 2001220389, Nguyễn Viết An Bình, Lớp 14DHTH07
3. 2001230396, Võ Anh Khoa, Lớp 14DHTH07
4. 2001230662, Mã Nhật Phong, Lớp 14DHTH07
5. 2001221811, Nguyễn Hữu Hoàng Thông, Lớp 14DHTH07

**GVHD:** Phan Thị Tài

**TP. HỒ CHÍ MINH, tháng 11 năm 2025**

# Mục lục

<b>Chương 1. XÂY DỰNG CƠ SỞ DỮ LIỆU</b>	<b>1</b>
1.1 Giới thiệu . . . . .	1
1.2 Lược đồ quan hệ thực thể (ERD) . . . . .	1
1.3 Chi tiết các bảng dữ liệu . . . . .	3
1.4 Tính chuẩn hóa cơ sở dữ liệu . . . . .	13
1.5 Ràng buộc toàn vẹn dữ liệu . . . . .	14
1.6 Tối ưu hóa cơ sở dữ liệu . . . . .	15
<b>Chương 2. CÀI ĐẶT YÊU CẦU XỬ LÝ</b>	<b>16</b>
2.1 Giới thiệu tổng quan . . . . .	16
2.2 Phân tích các bảng dữ liệu sử dụng . . . . .	16
2.3 Phân tích luồng xử lý chính . . . . .	17
2.4 Tổng hợp các module xử lý theo chức năng . . . . .	27
2.5 Ví dụ dữ liệu thực tế và kết quả xử lý . . . . .	29
<b>Chương 3. QUẢN TRỊ HỆ THỐNG</b>	<b>31</b>
3.1 Giới thiệu . . . . .	31
3.2 Quản trị người dùng và phân quyền . . . . .	31
3.3 Sao lưu và phục hồi dữ liệu . . . . .	35
3.4 Các Stored Procedures, Functions và Triggers . . . . .	38
<b>Chương 4. XÂY DỰNG ỨNG DỤNG</b>	<b>42</b>
4.1 Giới thiệu . . . . .	42
4.2 Kiến trúc ứng dụng . . . . .	42
4.3 Kết nối Database . . . . .	42
4.4 Phân công và triển khai chức năng . . . . .	44
4.5 Chức năng 1: Quản lý Giảng viên . . . . .	45
4.6 Chức năng 2: Quản lý Khoa . . . . .	48
4.7 Chức năng 3: Đăng ký và Quản lý điểm . . . . .	52
4.8 Chức năng 4: Quản lý Sinh viên . . . . .	57
4.9 Chức năng 5: Quản lý Môn học . . . . .	61
4.10 Demo . . . . .	66
<b>KẾT LUẬN</b>	<b>67</b>



## Danh sách hình vẽ

1	Sơ đồ quan hệ giữa các bảng trong cơ sở dữ liệu QLSV . . . . .	2
2	Màn hình Quản lý giảng viên . . . . .	46
3	Form cập nhật chức vụ . . . . .	48
4	Màn hình quản lý khoa . . . . .	49
5	Màn hình nhập điểm . . . . .	52
6	Bảng điểm sinh viên . . . . .	57
7	Màn hình quản lý sinh viên . . . . .	57
8	Kết quả danh sách sinh viên . . . . .	61
9	Màn hình quản lý môn học . . . . .	61
10	Báo cáo học kỳ . . . . .	65

## Danh sách bảng

1	Cấu trúc bảng HeDaoTao . . . . .	3
2	Cấu trúc bảng Khoa . . . . .	4
3	Cấu trúc bảng ChucVu . . . . .	4
4	Cấu trúc bảng Lop . . . . .	5
5	Cấu trúc bảng GiangVien . . . . .	6
6	Cấu trúc bảng SinhVien . . . . .	7
7	Cấu trúc bảng MonHoc . . . . .	7
8	Cấu trúc bảng HocKy . . . . .	8
9	Cấu trúc bảng LopHocPhan . . . . .	9
10	Cấu trúc bảng DangKyHocPhan . . . . .	10
11	Cấu trúc bảng MonHoc_TienQuyét . . . . .	10
12	Cấu trúc bảng Account . . . . .	11
13	Cấu trúc bảng LoginLog . . . . .	12
14	Cấu trúc bảng LoiHeThong . . . . .	12
15	Các thành phần xử lý module quản lý tổ chức . . . . .	27
16	Các thành phần xử lý module quản lý giảng viên . . . . .	28
17	Các thành phần xử lý module quản lý sinh viên . . . . .	28
18	Các thành phần xử lý module quản lý học phần . . . . .	28
19	Các thành phần xử lý module quản lý môn học . . . . .	29
20	Lịch trình sao lưu . . . . .	35
21	Phân công chức năng cho các thành viên . . . . .	45

# Chương 1. XÂY DỰNG CƠ SỞ DỮ LIỆU

## 1.1. Giới thiệu

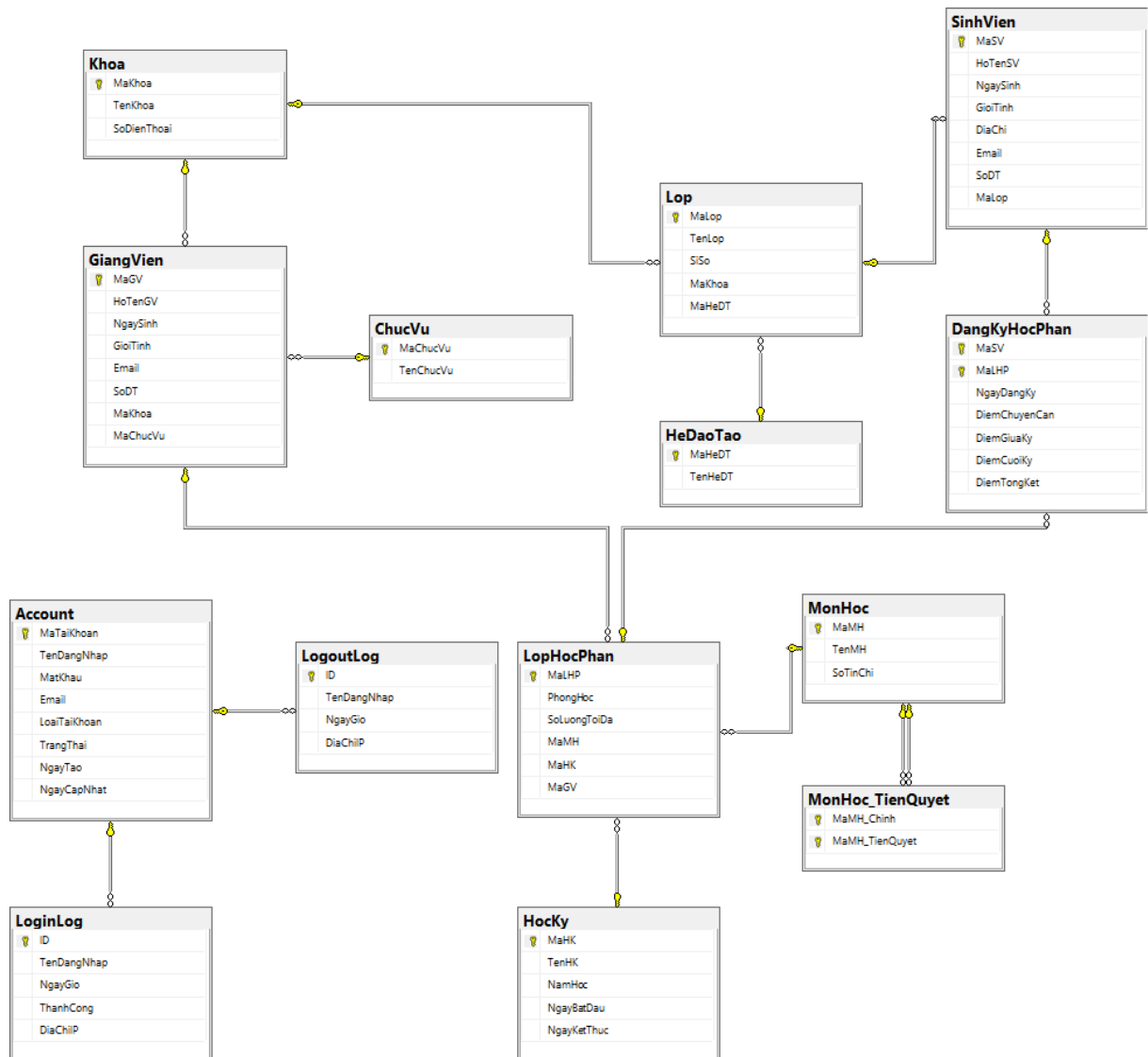
Chương này trình bày chi tiết về việc thiết kế và xây dựng cơ sở dữ liệu cho hệ thống Quản lý Sinh viên (QLSV\_DoAn\_2). Hệ thống được xây dựng trên SQL Server với 14 bảng chính và các bảng hỗ trợ, đảm bảo tính toàn vẹn, bảo mật và khả năng mở rộng cho hoạt động quản lý đào tạo tại trường đại học.

## 1.2. Lược đồ quan hệ thực thể (ERD)

### 1.2.1. Tổng quan mô hình dữ liệu

Mô hình cơ sở dữ liệu được thiết kế theo mô hình quan hệ (Relational Model) với các nhóm thực thể chính:

- **Nhóm quản lý tổ chức:** Khoa, Hệ đào tạo, Chức vụ, Lớp
- **Nhóm quản lý con người:** Sinh viên, Giảng viên
- **Nhóm quản lý đào tạo:** Môn học, Học kỳ, Lớp học phần, Đăng ký học phần
- **Nhóm quản lý hệ thống:** Account, LoginLog, LogoutLog, LoiHeThong



Hình 1. Sơ đồ quan hệ giữa các bảng trong cơ sở dữ liệu QLSV

### 1.2.2. Phân tích các quan hệ chính

#### – Quan hệ 1-nhiều (1-N):

- + Khoa (1) - GiangVien (N): Một khoa có nhiều giảng viên
- + Khoa (1) - Lop (N): Một khoa quản lý nhiều lớp
- + ChucVu (1) - GiangVien (N): Một chức vụ có thể gán cho nhiều giảng viên
- + HeDaoTao (1) - Lop (N): Một hệ đào tạo có nhiều lớp
- + Lop (1) - SinhVien (N): Một lớp có nhiều sinh viên
- + MonHoc (1) - LopHocPhan (N): Một môn học có thể mở nhiều lớp học phần
- + HocKy (1) - LopHocPhan (N): Một học kỳ có nhiều lớp học phần
- + GiangVien (1) - LopHocPhan (N): Một giảng viên dạy nhiều lớp học phần

– **Quan hệ nhiều-nhiều (N-N):**

- + SinhVien - LopHocPhan (thông qua DangKyHocPhan): Một sinh viên đăng ký nhiều lớp học phần, một lớp học phần có nhiều sinh viên
- + MonHoc - MonHoc (thông qua MonHoc\_TienQuyet): Quan hệ tiên quyết giữa các môn học

### 1.3. Chi tiết các bảng dữ liệu

#### 1.3.1. Nhóm bảng quản lý tổ chức

##### 1. Bảng HeDaoTao

Lưu trữ thông tin các hệ đào tạo của trường.

Bảng 1. Cấu trúc bảng HeDaoTao

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
MaHeDT	NVARCHAR(10)	PRIMARY KEY	Mã hệ đào tạo
TenHeDT	NVARCHAR(100)	NOT NULL	Tên hệ đào tạo

```
1 CREATE TABLE HeDaoTao (  
2     MaHeDT NVARCHAR(10) PRIMARY KEY,  
3     TenHeDT NVARCHAR(100) NOT NULL  
4 )
```

Listing 1: Script tạo bảng HeDaoTao

##### Dữ liệu mẫu:

- CQ - Chính quy
- CLC - Chất lượng cao
- LT - Liên thông
- TC - Tại chức

##### 2. Bảng Khoa

Quản lý thông tin các khoa trong trường.

Bảng 2. Cấu trúc bảng Khoa

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
MaKhoa	NVARCHAR(10)	PRIMARY KEY	Mã khoa
TenKhoa	NVARCHAR(100)	NOT NULL, UNIQUE	Tên khoa
SoDienThoai	VARCHAR(15)	-	Số điện thoại

```

1 CREATE TABLE Khoa (
2     MaKhoa NVARCHAR(10) PRIMARY KEY,
3     TenKhoa NVARCHAR(100) NOT NULL UNIQUE,
4     SoDienThoai VARCHAR(15)
5 )

```

Listing 2: Script tạo bảng Khoa

#### Dữ liệu mẫu:

- CNTT - Công nghệ Thông tin
- QTKD - Quản trị Kinh doanh
- NN - Ngoại ngữ
- DL - Du lịch & Lữ hành
- CK - Cơ khí

### 3. Bảng ChucVu

Định nghĩa các chức vụ trong hệ thống.

Bảng 3. Cấu trúc bảng ChucVu

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
MaChucVu	NVARCHAR(10)	PRIMARY KEY	Mã chức vụ
TenChucVu	NVARCHAR(100)	NOT NULL	Tên chức vụ

```

1 CREATE TABLE ChucVu (
2     MaChucVu NVARCHAR(10) PRIMARY KEY,
3     TenChucVu NVARCHAR(100) NOT NULL
4 )

```

Listing 3: Script tạo bảng ChucVu

#### 4. Bảng Lop

Quản lý thông tin các lớp học.

Bảng 4. Cấu trúc bảng Lop

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
MaLop	NVARCHAR(10)	PRIMARY KEY	Mã lớp
TenLop	NVARCHAR(100)	NOT NULL	Tên lớp
SiSo	INT	DEFAULT 0	Sĩ số lớp
MaKhoa	NVARCHAR(10)	FOREIGN KEY	Mã khoa
MaHeDT	NVARCHAR(10)	FOREIGN KEY	Mã hệ đào tạo

```
1 CREATE TABLE Lop (  
2     MaLop NVARCHAR(10) PRIMARY KEY ,  
3     TenLop NVARCHAR(100) NOT NULL ,  
4     SiSo INT DEFAULT 0 ,  
5     MaKhoa NVARCHAR(10) ,  
6     MaHeDT NVARCHAR(10) ,  
7     CONSTRAINT FK_Lop_Khoa FOREIGN KEY (MaKhoa) REFERENCES Khoa  
8         (MaKhoa) ,  
9     CONSTRAINT FK_Lop_HeDaoTao FOREIGN KEY (MaHeDT) REFERENCES  
10         HeDaoTao (MaHeDT)  
11 )
```

Listing 4: Script tạo bảng Lop

#### 1.3.2. Nhóm bảng quản lý con người

##### 5. Bảng GiangVien

Lưu trữ thông tin giảng viên.

Bảng 5. Cấu trúc bảng GiangVien

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
MaGV	NVARCHAR(10)	PRIMARY KEY	Mã giảng viên
HoTenGV	NVARCHAR(100)	NOT NULL	Họ tên giảng viên
NgaySinh	DATE	-	Ngày sinh
GioiTinh	NVARCHAR(5)	CHECK (IN ('Nam', 'Nữ'))	Giới tính
Email	VARCHAR(100)	NOT NULL, UNIQUE	Email
SoDT	VARCHAR(15)	UNIQUE	Số điện thoại
MaKhoa	NVARCHAR(10)	FOREIGN KEY	Mã khoa
MaChucVu	NVARCHAR(10)	FOREIGN KEY	Mã chức vụ

```

1 CREATE TABLE GiangVien (
2     MaGV NVARCHAR(10) PRIMARY KEY,
3     HoTenGV NVARCHAR(100) NOT NULL,
4     NgaySinh DATE,
5     GioiTinh NVARCHAR(5) CHECK (GioiTinh IN (N'Nam', N'Nu')),
6     Email VARCHAR(100) NOT NULL UNIQUE,
7     SoDT VARCHAR(15) UNIQUE,
8     MaKhoa NVARCHAR(10),
9     MaChucVu NVARCHAR(10),
10    CONSTRAINT FK_GiangVien_Khoa FOREIGN KEY (MaKhoa)
11        REFERENCES Khoa(MaKhoa),
12    CONSTRAINT FK_GiangVien_ChucVu FOREIGN KEY (MaChucVu)
        REFERENCES ChucVu(MaChucVu)
)

```

Listing 5: Script tạo bảng GiangVien

## 6. Bảng SinhVien

Lưu trữ thông tin sinh viên.

Bảng 6. Cấu trúc bảng SinhVien

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
MaSV	NVARCHAR(10)	PRIMARY KEY	Mã sinh viên
HoTenSV	NVARCHAR(100)	NOT NULL	Họ tên sinh viên
NgaySinh	DATE	-	Ngày sinh
GioiTinh	NVARCHAR(5)	CHECK (IN ('Nam', 'Nữ'))	Giới tính
DiaChi	NVARCHAR(255)	-	Địa chỉ
Email	VARCHAR(100)	NOT NULL, UNIQUE	Email
SoDT	VARCHAR(15)	UNIQUE	Số điện thoại
MaLop	NVARCHAR(10)	FOREIGN KEY	Mã lớp

```

1 CREATE TABLE SinhVien (
2     MaSV NVARCHAR(10) PRIMARY KEY,
3     HoTenSV NVARCHAR(100) NOT NULL,
4     NgaySinh DATE,
5     GioiTinh NVARCHAR(5) CHECK (GioiTinh IN (N'Nam', N'Nu')),
6     DiaChi NVARCHAR(255),
7     Email VARCHAR(100) NOT NULL UNIQUE,
8     SoDT VARCHAR(15) UNIQUE,
9     MaLop NVARCHAR(10),
10    CONSTRAINT FK_SinhVien_Lop FOREIGN KEY (MaLop) REFERENCES
11        Lop(MaLop)
12 )

```

Listing 6: Script tạo bảng SinhVien

### 1.3.3. Nhóm bảng quản lý đào tạo

#### 7. Bảng MonHoc

Định nghĩa các môn học trong chương trình đào tạo.

Bảng 7. Cấu trúc bảng MonHoc

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
MaMH	NVARCHAR(10)	PRIMARY KEY	Mã môn học
TenMH	NVARCHAR(100)	NOT NULL	Tên môn học
SoTinChi	INT	NOT NULL, CHECK (> 0)	Số tín chỉ

```

1 CREATE TABLE MonHoc (

```

```

2      MaMH NVARCHAR(10) PRIMARY KEY,
3      TenMH NVARCHAR(100) NOT NULL,
4      SoTinChi INT NOT NULL CHECK (SoTinChi > 0)
5  )

```

Listing 7: Script tạo bảng MonHoc

#### Dữ liệu mẫu:

- CSDL - Cơ sở dữ liệu (3 tín chỉ)
- HDT - Hướng đối tượng (3 tín chỉ)
- LTW - Lập trình Web (3 tín chỉ)
- MMT - Mạng máy tính (3 tín chỉ)

### 8. Bảng HocKy

Quản lý thông tin học kỳ.

Bảng 8. Cấu trúc bảng HocKy

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
MaHK	NVARCHAR(10)	PRIMARY KEY	Mã học kỳ
TenHK	NVARCHAR(50)	NOT NULL	Tên học kỳ
NamHoc	VARCHAR(20)	NOT NULL	Năm học
NgayBatDau	DATE	-	Ngày bắt đầu
NgayKetThuc	DATE	CHECK (> NgayBatDau)	Ngày kết thúc

```

1 CREATE TABLE HocKy (
2     MaHK NVARCHAR(10) PRIMARY KEY,
3     TenHK NVARCHAR(50) NOT NULL,
4     NamHoc VARCHAR(20) NOT NULL,
5     NgayBatDau DATE,
6     NgayKetThuc DATE,
7     UNIQUE(TenHK, NamHoc),
8     CONSTRAINT CHK_HocKy_NgayThang CHECK (NgayKetThuc >
9         NgayBatDau)
10 )

```

Listing 8: Script tạo bảng HocKy

## 9. Bảng LopHocPhan

Liên kết môn học, học kỳ và giảng viên để tạo lớp học phần cụ thể.

Bảng 9. Cấu trúc bảng LopHocPhan

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
MaLHP	NVARCHAR(10)	PRIMARY KEY	Mã lớp học phần
PhongHoc	NVARCHAR(50)	-	Phòng học
SoLuongToiDa	INT	DEFAULT 40	Số lượng tối đa
MaMH	NVARCHAR(10)	FOREIGN KEY	Mã môn học
MaHK	NVARCHAR(10)	FOREIGN KEY	Mã học kỳ
MaGV	NVARCHAR(10)	FOREIGN KEY	Mã giảng viên

```
1 CREATE TABLE LopHocPhan (  
2     MaLHP NVARCHAR(10) PRIMARY KEY,  
3     PhongHoc NVARCHAR(50),  
4     SoLuongToiDa INT DEFAULT 40,  
5     MaMH NVARCHAR(10),  
6     MaHK NVARCHAR(10),  
7     MaGV NVARCHAR(10),  
8     CONSTRAINT FK_LopHocPhan_MonHoc FOREIGN KEY (MaMH)  
9         REFERENCES MonHoc(MaMH),  
10    CONSTRAINT FK_LopHocPhan_HocKy FOREIGN KEY (MaHK)  
11        REFERENCES HocKy(MaHK),  
12    CONSTRAINT FK_LopHocPhan_GiangVien FOREIGN KEY (MaGV)  
13        REFERENCES GiangVien(MaGV)  
14 )
```

Listing 9: Script tạo bảng LopHocPhan

## 10. Bảng DangKyHocPhan

Ghi nhận việc đăng ký học phần của sinh viên và điểm số.

Bảng 10. Cấu trúc bảng DangKyHocPhan

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
MaSV	NVARCHAR(10)	PRIMARY KEY (composite)	Mã sinh viên
MaLHP	NVARCHAR(10)	PRIMARY KEY (composite)	Mã lớp học phần
NgayDangKy	DATETIME	DEFAULT GETDATE()	Ngày đăng ký
DiemChuyenCan	FLOAT	CHECK (0-10)	Điểm chuyên cần
DiemGiuaKy	FLOAT	CHECK (0-10)	Điểm giữa kỳ
DiemCuoiKy	FLOAT	CHECK (0-10)	Điểm cuối kỳ
DiemTongKet	FLOAT	-	Điểm tổng kết

```

1 CREATE TABLE DangKyHocPhan (
2     MaSV NVARCHAR(10),
3     MaLHP NVARCHAR(10),
4     NgayDangKy DATETIME DEFAULT GETDATE(),
5     DiemChuyenCan FLOAT CHECK (DiemChuyenCan >= 0 AND
6         DiemChuyenCan <= 10),
7     DiemGiuaKy FLOAT CHECK (DiemGiuaKy >= 0 AND DiemGiuaKy <=
8         10),
9     DiemCuoiKy FLOAT CHECK (DiemCuoiKy >= 0 AND DiemCuoiKy <=
10        10),
11    DiemTongKet FLOAT,
12    PRIMARY KEY (MaSV, MaLHP),
13    CONSTRAINT FK_DangKyHocPhan_SinhVien FOREIGN KEY (MaSV)
14        REFERENCES SinhVien(MaSV),
15    CONSTRAINT FK_DangKyHocPhan_LopHocPhan FOREIGN KEY (MaLHP)
16        REFERENCES LopHocPhan(MaLHP)
17 )

```

Listing 10: Script tạo bảng DangKyHocPhan

## 11. Bảng MonHoc\_TienQuyết

Xác định quan hệ tiên quyết giữa các môn học.

Bảng 11. Cấu trúc bảng MonHoc\_TienQuyết

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
MaMH_Chinh	NVARCHAR(10)	PRIMARY KEY (composite)	Mã môn học chính
MaMH_TienQuyết	NVARCHAR(10)	PRIMARY KEY (composite)	Mã môn tiên quyết

```

1 CREATE TABLE MonHoc_TienQuyet (
2     MaMH_Chinh NVARCHAR(10),
3     MaMH_TienQuyet NVARCHAR(10),
4     PRIMARY KEY (MaMH_Chinh, MaMH_TienQuyet),
5     CONSTRAINT FK_TienQuyet_MonHocChinh FOREIGN KEY (MaMH_Chinh
        ) REFERENCES MonHoc(MaMH),
6     CONSTRAINT FK_TienQuyet_MonHocTQ FOREIGN KEY (
        MaMH_TienQuyet) REFERENCES MonHoc(MaMH)
7 )

```

Listing 11: Script tạo bảng MonHoc\_TienQuyet

### 1.3.4. Nhóm bảng quản lý hệ thống

#### 12. Bảng Account

Quản lý tài khoản người dùng hệ thống.

Bảng 12. Cấu trúc bảng Account

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
MaTaiKhoan	NVARCHAR(20)	PRIMARY KEY	Mã tài khoản
TenDangNhap	NVARCHAR(50)	NOT NULL, UNIQUE	Tên đăng nhập
MatKau	NVARCHAR(MAX)	NOT NULL	Mật khẩu mã hóa
Email	VARCHAR(100)	NOT NULL, UNIQUE	Email
LoaiTaiKhoan	NVARCHAR(20)	CHECK	Admin/Lecturer/Student
TrangThai	BIT	DEFAULT 1	Trạng thái hoạt động
NgayTao	DATETIME	DEFAULT GETDATE()	Ngày tạo
NgayCapNhat	DATETIME	-	Ngày cập nhật

```

1 CREATE TABLE Account (
2     MaTaiKhoan NVARCHAR(20) PRIMARY KEY,
3     TenDangNhap NVARCHAR(50) NOT NULL UNIQUE,
4     MatKau NVARCHAR(MAX) NOT NULL,
5     Email VARCHAR(100) NOT NULL UNIQUE,
6     LoaiTaiKhoan NVARCHAR(20) CHECK (LoaiTaiKhoan IN ('Admin',
        'Lecturer', 'Student')),
7     TrangThai BIT DEFAULT 1,
8     NgayTao DATETIME DEFAULT GETDATE(),

```

```

9      NgayCapNhat DATETIME ,
10      CONSTRAINT CHK_Email CHECK (Email LIKE '%@%')
11 )

```

Listing 12: Script tạo bảng Account

### 13. Bảng LoginLog

Ghi nhận lịch sử đăng nhập.

Bảng 13. Cấu trúc bảng LoginLog

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
ID	INT IDENTITY	PRIMARY KEY	ID tự động tăng
TenDangNhap	NVARCHAR(50)	FOREIGN KEY	Tên đăng nhập
NgayGio	DATETIME	DEFAULT GETDATE()	Thời gian đăng nhập
ThanhCong	BIT	-	Trạng thái đăng nhập
DiaChiIP	VARCHAR(20)	-	Địa chỉ IP

```

1 CREATE TABLE LoginLog (
2     ID INT IDENTITY PRIMARY KEY ,
3     TenDangNhap NVARCHAR(50) ,
4     NgayGio DATETIME DEFAULT GETDATE() ,
5     ThanhCong BIT ,
6     DiaChiIP VARCHAR(20) ,
7     CONSTRAINT FK_LoginLog_Account FOREIGN KEY (TenDangNhap)
8     REFERENCES Account(TenDangNhap)
9 )

```

Listing 13: Script tạo bảng LoginLog

### 14. Bảng LoiHeThong

Lưu trữ các lỗi phát sinh trong hệ thống.

Bảng 14. Cấu trúc bảng LoiHeThong

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
ID	INT IDENTITY	PRIMARY KEY	ID tự động tăng
NoiDung	NVARCHAR(255)	-	Nội dung lỗi
ThoiGian	DATETIME	DEFAULT GETDATE()	Thời gian xảy ra

```

1 CREATE TABLE LoiHeThong (
2     ID INT IDENTITY PRIMARY KEY ,
3     NoiDung NVARCHAR(255) ,
4     ThoiGian DATETIME DEFAULT GETDATE()
5 )

```

Listing 14: Script tạo bảng LoiHeThong

## 1.4. Tính chuẩn hóa cơ sở dữ liệu

### 1.4.1. Chuẩn hóa 1NF (First Normal Form)

Tất cả các bảng trong hệ thống đều đạt chuẩn 1NF:

- Mỗi cột chỉ chứa giá trị nguyên tử (atomic value)
- Không có nhóm lặp lại (repeating groups)
- Mỗi bảng có khóa chính xác định duy nhất từng bản ghi

### 1.4.2. Chuẩn hóa 2NF (Second Normal Form)

Các bảng đạt chuẩn 2NF:

- Đã thỏa mãn 1NF
- Không có phụ thuộc hàm một phần (partial dependency)
- Mọi thuộc tính không khóa đều phụ thuộc đầy đủ vào khóa chính

Ví dụ: Bảng DangKyHocPhan có khóa chính kết hợp (MaSV, MaLHP), các thuộc tính điểm phụ thuộc vào cả hai khóa này.

### 1.4.3. Chuẩn hóa 3NF (Third Normal Form)

Hệ thống đạt chuẩn 3NF:

- Đã thỏa mãn 2NF
- Không có phụ thuộc bắc cầu (transitive dependency)
- Mọi thuộc tính không khóa chỉ phụ thuộc trực tiếp vào khóa chính

Ví dụ: Thông tin khoa được tách riêng thành bảng Khoa, không lưu trực tiếp trong bảng SinhVien hay GiangVien.

## **1.5. Ràng buộc toàn vẹn dữ liệu**

### **1.5.1. Ràng buộc khóa chính (Primary Key)**

- Đảm bảo tính duy nhất cho mỗi bản ghi
- Không cho phép giá trị NULL
- Ví dụ: MaSV trong bảng SinhVien, MaGV trong bảng GiangVien

### **1.5.2. Ràng buộc khóa ngoại (Foreign Key)**

- Đảm bảo tính toàn vẹn tham chiếu giữa các bảng
- Ngăn chặn xóa dữ liệu cha khi còn dữ liệu con
- Ví dụ: MaLop trong SinhVien tham chiếu đến Lop, MaKhoa trong GiangVien tham chiếu đến Khoa

### **1.5.3. Ràng buộc kiểm tra (Check Constraint)**

- Giới hạn miền giá trị của thuộc tính
- Ví dụ:
  - + GioiTinh IN ('Nam', 'Nữ')
  - + SoTinChi > 0
  - + DiemChuyenCan, DiemGiuaKy, DiemCuoiKy BETWEEN 0 AND 10
  - + NgayKetThuc > NgayBatDau trong bảng HocKy

### **1.5.4. Ràng buộc duy nhất (Unique Constraint)**

- Đảm bảo không có giá trị trùng lặp
- Cho phép một giá trị NULL
- Ví dụ: Email, SoDT trong bảng SinhVien và GiangVien

### **1.5.5. Ràng buộc giá trị mặc định (Default Constraint)**

- Tự động gán giá trị khi không được cung cấp
- Ví dụ:
  - + SiSo = 0 trong bảng Lop

- + SoLuongToiDa = 40 trong bảng LopHocPhan
- + NgayDangKy = GETDATE() trong bảng DangKyHocPhan
- + TrangThai = 1 trong bảng Account

## 1.6. Tối ưu hóa cơ sở dữ liệu

### 1.6.1. Chiến lược đánh chỉ mục (Index)

- **Clustered Index:** Tự động tạo trên khóa chính của mỗi bảng
- **Non-Clustered Index:** Tạo trên các cột thường xuyên được tìm kiếm:
  - + Email trong bảng SinhVien, GiangVien
  - + MaKhoa trong bảng Lop
  - + MaMH, MaHK trong bảng LopHocPhan

### 1.6.2. Phân vùng dữ liệu (Partitioning)

- Phân vùng bảng DangKyHocPhan theo học kỳ để tăng hiệu suất truy vấn
- Phân vùng bảng LoginLog theo tháng để quản lý log hiệu quả

### 1.6.3. Chiến lược sao lưu (Backup Strategy)

- **Full Backup:** Hàng tuần vào cuối tuần
- **Differential Backup:** Hàng ngày vào cuối ngày
- **Transaction Log Backup:** Mỗi 4 giờ trong giờ làm việc

## Chương 2. CÀI ĐẶT YÊU CẦU XỬ LÝ

### 2.1. Giới thiệu tổng quan

Chương này trình bày chi tiết về việc cài đặt các yêu cầu xử lý phức tạp trong hệ thống quản lý sinh viên thông qua các cấu trúc lập trình T-SQL. Hệ thống được xây dựng với mục tiêu quản lý toàn diện các hoạt động đào tạo bao gồm: quản lý sinh viên, giảng viên, khoa, lớp học, môn học, học kỳ, lớp học phần, đăng ký học phần và điểm số. Các yêu cầu xử lý được thực hiện thông qua Stored Procedures, Functions, Triggers, Cursors và Transactions để đảm bảo tính toàn vẹn, tự động hóa và tối ưu hóa hiệu suất.

### 2.2. Phân tích các bảng dữ liệu sử dụng

#### 2.2.1. Nhóm bảng quản lý tổ chức

- **HeDaoTao**: Lưu trữ thông tin các hệ đào tạo (Chính quy, Chất lượng cao, Liên thông, Tại chức)
- **Khoa**: Quản lý thông tin các khoa (CNTT, QTKD, Ngoại ngữ, Du lịch, Cơ khí)
- **ChucVu**: Định nghĩa các chức vụ (Giảng viên, Trưởng khoa, Phó khoa, Trợ giảng, Nhân viên)
- **Lop**: Quản lý các lớp học với quan hệ tới Khoa và Hệ đào tạo

#### 2.2.2. Nhóm bảng quản lý con người

- **GiangVien**: Lưu trữ thông tin giảng viên với quan hệ tới Khoa và ChucVu
- **SinhVien**: Quản lý thông tin sinh viên với quan hệ tới Lop

#### 2.2.3. Nhóm bảng quản lý đào tạo

- **MonHoc**: Định nghĩa các môn học với số tín chỉ
- **MonHoc\_TienQuyết**: Xác định quan hệ tiên quyết giữa các môn học
- **HocKy**: Quản lý thông tin học kỳ với thời gian bắt đầu và kết thúc
- **LopHocPhan**: Liên kết MonHoc, HocKy và GiangVien để tạo lớp học phần cụ thể
- **DangKyHocPhan**: Ghi nhận việc đăng ký học phần của sinh viên với điểm số

## 2.2.4. Nhóm bảng quản lý hệ thống

- **Account**: Quản lý tài khoản người dùng (Admin, Lecturer, Student)
- **LoginLog**: Ghi nhận lịch sử đăng nhập
- **LogoutLog**: Ghi nhận lịch sử đăng xuất
- **LoiHeThong**: Lưu trữ các lỗi phát sinh trong hệ thống

## 2.3. Phân tích luồng xử lý chính

### 2.3.1. Luồng quản lý giảng viên và chức vụ

#### 1. Cập nhật chức vụ giảng viên (SP\_CAPNHAT\_CHUCVU\_GV)

Luồng xử lý:

1. Kiểm tra tồn tại mã giảng viên
2. Kiểm tra tính hợp lệ của mã chức vụ mới
3. Cập nhật chức vụ cho giảng viên
4. Ghi nhận thành công

```
1 CREATE PROCEDURE SP_CAPNHAT_CHUCVU_GV
2     @MaGV NVARCHAR(10),
3     @MaChucVuMoi NVARCHAR(10)
4 AS
5 BEGIN
6     SET NOCOUNT ON;
7     -- Kiểm tra tồn tại Giảng viên
8     IF NOT EXISTS (SELECT 1 FROM GiangVien WHERE MaGV = @MaGV)
9     BEGIN
10         RAISERROR(N'Loi: Ma Giang vien khong ton tai.', 16, 1);
11         RETURN;
12     END
13     -- Kiểm tra tồn tại Chức vụ mới
14     IF NOT EXISTS (SELECT 1 FROM ChucVu WHERE MaChucVu =
15         @MaChucVuMoi)
16     BEGIN
17         RAISERROR(N'Loi: Ma Chuc vu moi khong hop le.', 16, 1);
18         RETURN;
19     END
```

```

19  -- Cap nhât
20  UPDATE GiangVien
21  SET MaChucVu = @MaChucVuMoi
22  WHERE MaGV = @MaGV;

23
24  PRINT N'Cap nhât chuc vu thanh cong cho Giang vien ' +
      @MaGV;
25  END
26  GO

```

Listing 15: Stored Procedure cập nhật chức vụ giảng viên

### Ví dụ thực thi:

```

1  -- Chuyen GV003 tu Giang vien (GV) thanh Truong khoa (TK)
2  EXEC SP_CAPNHAT_CHUCVU_GV N'GV003', N'TK';

```

## 2. Thăng chức trưởng khoa với Transaction (SP\_THANG\_CHUC\_TRUONG\_KHOA)

Luồng xử lý có tính giao dịch:

1. Bắt đầu giao dịch
2. Tìm trưởng khoa cũ của khoa
3. Kiểm tra giảng viên thăng chức thuộc khoa đó
4. Hạ chức trưởng khoa cũ về giảng viên
5. Thăng chức giảng viên mới lên trưởng khoa
6. Commit hoặc Rollback giao dịch

```

1  CREATE PROCEDURE SP_THANG_CHUC_TRUONG_KHOA
2      @MaGVThangChuc NVARCHAR(10),
3      @MaKhoa NVARCHAR(10)
4  AS
5  BEGIN
6      SET NOCOUNT ON;
7      BEGIN TRANSACTION;

8
9      DECLARE @ErrorState INT = 0;
10     DECLARE @MaGVCu NVARCHAR(10);
11
12     -- Tim Truong khoa cu

```

```

13 SELECT @MaGVCu = MaGV
14 FROM GiangVien
15 WHERE MaKhoa = @MaKhoa AND MaChucVu = N'TK';
16
17 -- Kiem tra GV thang chuc thuoc Khoa
18 IF NOT EXISTS (SELECT 1 FROM GiangVien
19                WHERE MaGV = @MaGVThangChuc AND MaKhoa =
20                  @MaKhoa)
21 BEGIN
22     SET @ErrorState = 1;
23 END
24
25 -- Ha chuc Truong khoa cu
26 IF @MaGVCu IS NOT NULL
27 BEGIN
28     UPDATE GiangVien
29     SET MaChucVu = N'GV'
30     WHERE MaGV = @MaGVCu;
31 END
32
33 -- Thang chuc Giang vien moi
34 IF @ErrorState = 0
35 BEGIN
36     UPDATE GiangVien
37     SET MaChucVu = N'TK'
38     WHERE MaGV = @MaGVThangChuc;
39 END
40
41 -- Xu ly ket qua giao dich
42 IF @ErrorState = 1 OR @@ERROR <> 0
43 BEGIN
44     ROLLBACK TRANSACTION;
45     RAISERROR(N'Giao dich khong thanh cong.', 16, 1);
46 END
47 ELSE
48 BEGIN
49     COMMIT TRANSACTION;
50     PRINT N'Thang chuc thanh cong!';
51 END

```

```
51 END
52 GO
```

Listing 16: Transaction thăng chức trưởng khoa

### 2.3.2. Luồng quản lý khoa và hệ đào tạo

#### 1. Thêm khoa mới với kiểm tra trùng lặp (sp\_ThemKhoa)

Luồng xử lý:

1. Kiểm tra tên khoa đã tồn tại chưa
2. Nếu trùng, ghi log lỗi vào bảng LoiHeThong
3. Nếu không trùng, thêm khoa mới

```
1 CREATE PROCEDURE sp_ThemKhoa
2     @MaKhoa NVARCHAR(10),
3     @TenKhoa NVARCHAR(100),
4     @SoDienThoai VARCHAR(15)
5 AS
6 BEGIN
7     SET NOCOUNT ON;
8     IF EXISTS (SELECT 1 FROM Khoa WHERE TenKhoa = @TenKhoa)
9     BEGIN
10         INSERT INTO LoiHeThong (NoiDung)
11         VALUES (N'Trung ten khoa: ' + @TenKhoa);
12         PRINT N'Ten khoa da ton tai. Khong the them moi.';
13         RETURN;
14     END
15
16     INSERT INTO Khoa (MaKhoa, TenKhoa, SoDienThoai)
17     VALUES (@MaKhoa, @TenKhoa, @SoDienThoai);
18
19     PRINT N'Them khoa thanh cong!';
20 END;
21 GO
```

Listing 17: Stored Procedure thêm khoa

#### 2. Kiểm tra số điện thoại với Trigger (trg\_KiemTraSoDienThoai)

Trigger tự động kiểm tra định dạng số điện thoại khi thêm hoặc cập nhật khoa:

```

1 CREATE TRIGGER trg_KiemTraSoDienThoai
2 ON Khoa
3 FOR INSERT, UPDATE
4 AS
5 BEGIN
6     IF EXISTS (
7         SELECT * FROM inserted
8         WHERE SoDienThoai NOT LIKE '0%' OR LEN(SoDienThoai) <>
9             10
10    )
11    BEGIN
12        RAISERROR(N'So dien thoai khong hop le!', 16, 1);
13        ROLLBACK TRANSACTION;
14    END
15 END

```

Listing 18: Trigger kiểm tra số điện thoại

### 2.3.3. Luồng quản lý đăng ký học phần

#### 1. Đăng ký học phần với kiểm tra điều kiện (sp\_DangKyHocPhan)

Luồng xử lý phức tạp:

1. Kiểm tra lớp học phần tồn tại
2. Kiểm tra sinh viên đã đăng ký chưa
3. Kiểm tra sĩ số lớp học phần
4. Thực hiện đăng ký
5. Trigger tự động tính điểm tổng kết kích hoạt

```

1 CREATE OR ALTER PROCEDURE sp_DangKyHocPhan
2     @MaSV NVARCHAR(10),
3     @MaLHP NVARCHAR(10)
4 AS
5 BEGIN
6     SET NOCOUNT ON;
7
8     IF NOT EXISTS (SELECT 1 FROM LopHocPhan WHERE MaLHP =
9         @MaLHP)

```

```

9      BEGIN
10         PRINT N'Lop hoc phan khong ton tai!';
11         RETURN;
12     END;

13
14     IF EXISTS (SELECT 1 FROM DangKyHocPhan
15                WHERE MaSV = @MaSV AND MaLHP = @MaLHP)
16     BEGIN
17         PRINT N'Sinh vien da dang ky lop nay!';
18         RETURN;
19     END;

20
21     DECLARE @SL INT = (SELECT COUNT(*) FROM DangKyHocPhan WHERE
22                          MaLHP = @MaLHP);
23
24     DECLARE @ToiDa INT = (SELECT SoLuongToiDa FROM LopHocPhan
25                            WHERE MaLHP = @MaLHP);
26
27     IF @SL >= @ToiDa
28     BEGIN
29         PRINT N'Lop hoc phan da day!';
30         RETURN;
31     END;

32
33     INSERT INTO DangKyHocPhan (MaSV, MaLHP)
34     VALUES (@MaSV, @MaLHP);
35
36     PRINT N'Dang ky thanh cong!';
37 END;
38 GO

```

Listing 19: Stored Procedure đăng ký học phần

## 2. Trigger tự động tính điểm tổng kết (trg\_TinhDiemTongKet)

Trigger tự động kích hoạt khi có cập nhật điểm:

```

1 CREATE OR ALTER TRIGGER trg_TinhDiemTongKet
2 ON DangKyHocPhan
3 AFTER INSERT, UPDATE
4 AS
5 BEGIN
6     SET NOCOUNT ON;

```

```

7
8 UPDATE D
9 SET D.DiemTongKet = ROUND(
10     (ISNULL(D.DiemChuyenCan, 0)*0.1 +
11     ISNULL(D.DiemGiuaKy, 0)*0.3 +
12     ISNULL(D.DiemCuoiKy, 0)*0.6), 2)
13 FROM DangKyHocPhan D
14 JOIN inserted I
15     ON D.MaSV = I.MaSV AND D.MaLHP = I.MaLHP;
16 END;

```

Listing 20: Trigger tính điểm tổng kết

### 2.3.4. Luồng quản lý môn học và tiên quyết

#### 1. Xóa môn học với kiểm tra ràng buộc (sp\_XoaMonHoc)

Luồng xử lý:

1. Kiểm tra môn học có đang được sử dụng trong lớp học phần
2. Kiểm tra môn học có là tiên quyết của môn khác
3. Xóa quan hệ tiên quyết của môn này
4. Xóa môn học

```

1 CREATE PROCEDURE sp_XoaMonHoc @MaMH NVARCHAR(10)
2 AS
3 BEGIN
4     -- Kiểm tra môn có đang được LỚP HỌC PHẦN sử dụng
5     IF EXISTS (SELECT * FROM LopHocPhan WHERE MaMH = @MaMH)
6     BEGIN
7         PRINT N'LOI: Không thể xóa môn. Môn học đã được mô lớp
8             hoc phan.';
9         RETURN;
10    END
11    -- Kiểm tra môn có là TIÊN QUYẾT cho môn khác
12    IF EXISTS (SELECT * FROM MonHoc_TienQuyet WHERE
13        MaMH_TienQuyet = @MaMH)
14    BEGIN
15        PRINT N'LOI: Không thể xóa môn. Môn học đang là tiên
16            quyết cho môn khác.';

```

```

14         RETURN;
15     END
16
17     -- Xoa quan he tien quyet
18     DELETE FROM MonHoc_TienQuyet
19     WHERE MaMH_Chinh = @MaMH;
20
21     -- Xoa mon hoc
22     DELETE FROM MonHoc
23     WHERE MaMH = @MaMH;
24 END
25 GO

```

Listing 21: Stored Procedure xóa môn học

## 2. Kiểm tra điều kiện tiên quyết (fn\_KiemTraTienQuyet)

Function kiểm tra sinh viên đã học đủ môn tiên quyết:

```

1 CREATE FUNCTION fn_KiemTraTienQuyet
2 (
3     @MaSV NVARCHAR(10),
4     @MaMH_Chinh NVARCHAR(10)
5 )
6 RETURNS BIT
7 AS
8 BEGIN
9     DECLARE @SoMonTienQuyet INT;
10    DECLARE @SoMonDaDat INT;
11    DECLARE @KetQua BIT = 0;
12
13    -- Dem tong so mon tien quyet
14    SELECT @SoMonTienQuyet = COUNT(*)
15    FROM MonHoc_TienQuyet
16    WHERE MaMH_Chinh = @MaMH_Chinh;
17
18    -- Neu khong co mon tien quyet -> Du dieu kien
19    IF @SoMonTienQuyet = 0
20    BEGIN
21        SET @KetQua = 1;
22        RETURN @KetQua;
23    END

```

```

24
25      -- Dem so mon tien quyet da dat (>= 4.0)
26      SELECT @SoMonDaDat = COUNT(DISTINCT mhtq.MaMH_TienQuyet)
27      FROM DangKyHocPhan AS dkhp
28      JOIN LopHocPhan AS lhp ON dkhp.MaLHP = lhp.MaLHP
29      JOIN MonHoc_TienQuyet AS mhtq ON lhp.MaMH = mhtq.
        MaMH_TienQuyet
30      WHERE
31          dkhp.MaSV = @MaSV
32          AND mhtq.MaMH_Chinh = @MaMH_Chinh
33          AND dkhp.DiemTongKet >= 4.0;
34
35      -- So sanh
36      IF @SoMonTienQuyet = @SoMonDaDat
37      BEGIN
38          SET @KetQua = 1;
39      END
40
41      RETURN @KetQua;
42  END
43  GO

```

Listing 22: Function kiểm tra tiên quyết

### 2.3.5. Luồng thống kê và báo cáo

#### 1. Báo cáo học kỳ với Cursor (sp\_BaoCaoHocKy)

Sử dụng Cursor để duyệt qua từng lớp học phần:

```

1  CREATE PROCEDURE sp_BaoCaoHocKy @MaHK NVARCHAR(10)
2  AS
3  BEGIN
4      PRINT N'--- BAO CAO SI SO HOC KY ' + @MaHK + N' ---';
5
6      DECLARE @MaLHP_Current NVARCHAR(10);
7      DECLARE @TenMH_Current NVARCHAR(100);
8      DECLARE @SiSo_Current INT;
9
10     -- Khai bao Cursor
11     DECLARE cur_LopHocPhan CURSOR FOR
12     SELECT

```

```

13         lhp.MaLHP,
14         mh.TenMH
15     FROM LopHocPhan AS lhp
16     JOIN MonHoc AS mh ON lhp.MaMH = mh.MaMH
17     WHERE lhp.MaHK = @MaHK;
18
19     -- Mo Cursor
20     OPEN cur_LopHocPhan;
21
22     -- Lay dong dau tien
23     FETCH NEXT FROM cur_LopHocPhan
24     INTO @MaLHP_Current, @TenMH_Current;
25
26     -- Vong lap duyet
27     WHILE (@@FETCH_STATUS = 0)
28     BEGIN
29         -- Dem si so thuc te
30         SELECT @SiSo_Current = COUNT(*)
31         FROM DangKyHocPhan
32         WHERE MaLHP = @MaLHP_Current;
33
34         -- In ket qua
35         PRINT N' - Lop ' + @MaLHP_Current +
36             N' (' + @TenMH_Current + N'): ' +
37             CAST(@SiSo_Current AS VARCHAR(10)) + N' sinh vien
38             .';
39
40         -- Lay dong tiep theo
41         FETCH NEXT FROM cur_LopHocPhan
42         INTO @MaLHP_Current, @TenMH_Current;
43     END
44
45     PRINT N'--- Ket thuc bao cao ---';
46
47     -- Dong va huy Cursor
48     CLOSE cur_LopHocPhan;
49     DEALLOCATE cur_LopHocPhan;
50 END
GO

```

### Listing 23: Cursor báo cáo học kỳ

## 2. Function tính điểm trung bình (fn\_TinhDiemTrungBinh)

Function tính điểm trung bình của sinh viên:

```
1 CREATE OR ALTER FUNCTION fn_TinhDiemTrungBinh (@MaSV NVARCHAR
   (10))
2 RETURNS FLOAT
3 AS
4 BEGIN
5     DECLARE @DTB FLOAT;
6
7     SELECT @DTB = AVG(DiemTongKet)
8     FROM DangKyHocPhan
9     WHERE MaSV = @MaSV
10        AND DiemTongKet IS NOT NULL;
11
12     RETURN @DTB;
13 END;
14 GO
```

### Listing 24: Function tính điểm trung bình

## 2.4. Tổng hợp các module xử lý theo chức năng

### 2.4.1. Module quản lý tổ chức

Bảng 15. Các thành phần xử lý module quản lý tổ chức

Loại	Tên	Chức năng
Stored Procedure	sp_ThemKhoa	Thêm khoa mới với kiểm tra trùng lặp
Stored Procedure	sp_DuyetTungHeDaoTao	Duyệt và hiển thị các hệ đào tạo
Function	fn_DemKhoaCoSoDienThoai	Đếm số khoa có số điện thoại
Trigger	trg_KiemTraSoDienThoai	Kiểm tra định dạng số điện thoại
Transaction	Cập nhật hệ đào tạo	Cập nhật tên hệ đào tạo an toàn

#### 2.4.2. Module quản lý giảng viên

Bảng 16. Các thành phần xử lý module quản lý giảng viên

Loại	Tên	Chức năng
Stored Procedure	SP_CAPNHAT_CHUCVU_GV	Cập nhật chức vụ giảng viên
Stored Procedure	SP_THANG_CHUC_TRUONG_KHOA	Thăng chức trưởng khoa
Function	FN_DEM_GV_THEO_CHUCVU	Đếm số giảng viên theo chức vụ
Trigger	TR_NGAN_CHAN_XOA_CHUCVU	Ngăn xóa chức vụ đang sử dụng
Cursor	CS_THONG_BAO_NGHI_HUU	Thông báo giảng viên nghỉ hưu

#### 2.4.3. Module quản lý sinh viên

Bảng 17. Các thành phần xử lý module quản lý sinh viên

Loại	Tên	Chức năng
Stored Procedure	sp_LayDanhSachSV_TheoLop	Lấy danh sách sinh viên theo lớp
Stored Procedure	sp_DuyetSinhVien_BangCursor	Duyệt sinh viên bằng cursor
Stored Procedure	sp_ChuyenLopChoSinhVien	Chuyển lớp cho sinh viên
Function	fn_DemSoSinhVien_TheoLop	Đếm số sinh viên trong lớp
Function	fn_LaySinhVien_TheoKhoa	Lấy danh sách sinh viên theo khoa
Trigger	trg_NganXoaLop_KhiConSinhVien	Ngăn xóa lớp còn sinh viên

#### 2.4.4. Module quản lý học phần

Bảng 18. Các thành phần xử lý module quản lý học phần

Loại	Tên	Chức năng
Stored Procedure	sp_DangKyHocPhan	Đăng ký học phần cho sinh viên
Stored Procedure	sp_DangKyMonHoc	Đăng ký môn học với kiểm tra tiên quyết
Trigger	trg_TinhDiemTongKet	Tự động tính điểm tổng kết
Trigger	trg_KiemTraNgayDangKy	Kiểm tra thời gian đăng ký hợp lệ
Function	fn_TinhDiemTrungBinh	Tính điểm trung bình sinh viên
Cursor	In điểm trung bình	Duyệt và in điểm từng sinh viên
Transaction	Đăng ký + cập nhật điểm	Gộp đăng ký và cập nhật điểm

## 2.4.5. Module quản lý môn học

Bảng 19. Các thành phần xử lý module quản lý môn học

Loại	Tên	Chức năng
Stored Procedure	sp_XoaMonHoc	Xóa môn học với kiểm tra ràng buộc
Stored Procedure	sp_ThemMonHoc	Thêm môn học với transaction
Stored Procedure	sp_CapNhatMonHoc	Cập nhật thông tin môn học
Cursor	sp_BaoCaoHocKy	Báo cáo sĩ số học kỳ
Function	fn_LayDanhSachMonTienQuyét	Lấy danh sách môn tiên quyết
Function	fn_KiemTraTienQuyét	Kiểm tra điều kiện tiên quyết
Trigger	trg_NganXoaHocKy	Ngăn xóa học kỳ có lớp học phần

## 2.5. Ví dụ dữ liệu thực tế và kết quả xử lý

### 2.5.1. Dữ liệu mẫu hệ thống

Hệ thống hiện có dữ liệu mẫu bao gồm:

- 4 hệ đào tạo: Chính quy (CQ), Chất lượng cao (CLC), Liên thông (LT), Tại chức (TC)
- 5 khoa: CNTT, QTKD, Ngoại ngữ, Du lịch, Cơ khí
- 7 giảng viên với các chức vụ khác nhau
- 8 sinh viên phân bổ trong 5 lớp
- 8 môn học với quan hệ tiên quyết
- 4 học kỳ và 8 lớp học phần
- 11 bản ghi đăng ký học phần với điểm số

### 2.5.2. Ví dụ thực thi các chức năng

#### 1. Cập nhật chức vụ giảng viên:

```
1 -- Chuyen GV003 (Nguyen Hung Dung) thanh Truong khoa
2 EXEC SP_CAPNHAT_CHUCVU_GV N'GV003', N'TK';
3 -- Ket qua: Cap nhap chuc vu thanh cong cho Giang vien GV003
```

#### 2. Đếm số giảng viên theo chức vụ:

```

1 SELECT dbo.FN_DEM_GV_THEO_CHUCVU(N'GV') AS SoLuongGiangVien;
2 -- Ket qua: 4 (co 4 giang vien giu chuc vu Giang vien)

```

### 3. Đăng ký học phần cho sinh viên:

```

1 -- Sinh vien 2001130001 dang ky lop hoc phan LHP04 (Lap trinh
   Web)
2 EXEC sp_DangKyHocPhan @MaSV = '2001130001', @MaLHP = 'LHP04';
3 -- Ket qua: Dang ky thanh cong!

```

### 4. Kiểm tra điều kiện tiên quyết:

```

1 -- Kiem tra SV 2001140001 du dieu kien hoc Lap trinh Web (LTW)
2 -- (Da hoc ca CSDL va HDT la tien quyet cua LTW)
3 SELECT dbo.fn_KiemTraTienQuyet('2001140001', 'LTW') AS
   DuDieuKien;
4 -- Ket qua: 1 (Du dieu kien)
5
6 -- Kiem tra SV 2001140002 (chi hoc CSDL, chua hoc HDT)
7 SELECT dbo.fn_KiemTraTienQuyet('2001140002', 'LTW') AS
   DuDieuKien;
8 -- Ket qua: 0 (Khong du dieu kien)

```

### 5. Báo cáo sĩ số học kỳ:

```

1 EXEC sp_BaoCaoHocKy @MaHK = N'HK1-2425';
2 -- Ket qua:
3 -- --- BAO CAO SI SO HOC KY HK1-2425 ---
4 -- - Lop LHP01 (Co so du lieu): 3 sinh vien.
5 -- - Lop LHP02 (Huong doi tuong): 1 sinh vien.
6 -- - Lop LHP03 (Marketing can ban): 2 sinh vien.
7 -- - Lop LHP04 (Lap trinh Web): 1 sinh vien.
8 -- - Lop LHP05 (Tieng Anh C1): 1 sinh vien.
9 -- - Lop LHP06 (Quan tri lu hanh): 1 sinh vien.
10 -- - Lop LHP07 (Ke toan can ban): 1 sinh vien.
11 -- --- Ket thuc bao cao ---

```

## Chương 3. QUẢN TRỊ HỆ THỐNG

### 3.1. Giới thiệu

Chương này trình bày các khía cạnh quan trọng trong quản trị hệ thống cơ sở dữ liệu SQL Server cho hệ thống QLSV\_DoAn\_2, bao gồm: quản lý người dùng, phân quyền bảo mật, và chiến lược sao lưu/phục hồi dữ liệu. Hệ thống được thiết kế với các bảng chính: HeDaoTao, Khoa, ChucVu, GiangVien, Lop, SinhVien, MonHoc, HocKy, LopHocPhan, DangKyHocPhan, MonHoc\_TienQuyet, Account, LoginLog, và LogoutLog.

### 3.2. Quản trị người dùng và phân quyền

#### Cơ chế xác thực

Hệ thống sử dụng **Mixed Mode Authentication** kết hợp cả hai phương thức:

- **Windows Authentication:** Cho các quản trị viên hệ thống và nhân viên nội bộ
- **SQL Server Authentication:** Cho ứng dụng web và người dùng từ xa

#### Lý do lựa chọn Mixed Mode:

1. Linh hoạt trong việc quản lý người dùng từ nhiều nguồn khác nhau
2. Ứng dụng web có thể kết nối từ các máy chủ không thuộc domain
3. Dễ dàng quản lý và cấp quyền cho từng nhóm người dùng
4. Hỗ trợ cả môi trường phát triển và production

#### Tạo Login và User

**Login** là tài khoản để xác thực ở cấp SQL Server instance, còn **User** là tài khoản được map từ Login để truy cập database cụ thể.

```
1  -- 1. Tao Login cho sinh vien
2  CREATE LOGIN sinhvien
3  WITH PASSWORD = 'sinhvien@123',
4       DEFAULT_DATABASE = QLSV_DoAn_2;
5
6  -- 2. Tao Login cho giang vien
7  CREATE LOGIN giangvien
8  WITH PASSWORD = 'giangvien@123',
9       DEFAULT_DATABASE = QLSV_DoAn_2;
10
```

```

11 -- 3. Tao Login cho quan tri (su dung cho Account table)
12 -- Vi du: admin, baotv, annv da duoc them vao bang Account
13 -- Mat khau duoc ma hoa SHA256
14 GO

```

Listing 25: Tạo Login cho các nhóm người dùng

```

1 USE QLSV_DoAn_2;
2 GO
3
4 -- Tao User cho sinh vien
5 CREATE USER user_sinhvien
6 FOR LOGIN sinhvien;
7
8 -- Tao User cho giang vien
9 CREATE USER user_giangvien
10 FOR LOGIN giangvien;
11
12 GO

```

Listing 26: Tạo User trong database

### Tạo Role và phân quyền

Database Roles giúp nhóm các quyền lại và dễ dàng quản lý phân quyền cho nhiều user.

```

1 USE QLSV_DoAn_2;
2 GO
3
4 -- 1. Tao nhom quyen cho sinh vien
5 CREATE ROLE role_sinhvien;
6
7 -- 2. Tao nhom quyen cho giang vien
8 CREATE ROLE role_giangvien;
9
10 GO

```

Listing 27: Tạo các Role trong database

```

1 -- 1. Sinh vien chi duoc xem du lieu
2 GRANT SELECT ON dbo.SinhVien TO role_sinhvien;
3 GRANT SELECT ON dbo.Lop TO role_sinhvien;
4 GRANT SELECT ON dbo.MonHoc TO role_sinhvien;
5 GRANT SELECT ON dbo.HocKy TO role_sinhvien;

```

```

6 GRANT SELECT ON dbo.LopHocPhan TO role_sinhvien;
7 GRANT SELECT ON dbo.DangKyHocPhan TO role_sinhvien;
8
9 -- Quyen thuc thi cac function lien quan
10 GRANT EXECUTE ON fn_TinhDiemTrungBinh TO role_sinhvien;
11 GRANT EXECUTE ON fn_KiemTraTienQuyet TO role_sinhvien;
12
13 -- 2. Giang vien duoc xem va cap nhat du lieu
14 GRANT SELECT, INSERT, UPDATE, DELETE
15 ON dbo.GiangVien
16 TO role_giangvien;
17
18 GRANT SELECT, INSERT, UPDATE
19 ON dbo.DangKyHocPhan
20 TO role_giangvien;
21
22 GRANT SELECT ON dbo.SinhVien TO role_giangvien;
23 GRANT SELECT ON dbo.Lop TO role_giangvien;
24 GRANT SELECT ON dbo.MonHoc TO role_giangvien;
25 GRANT SELECT ON dbo.LopHocPhan TO role_giangvien;
26
27 -- Quyen thuc thi stored procedures
28 GRANT EXECUTE ON sp_DangKyHocPhan TO role_giangvien;
29 GRANT EXECUTE ON sp_CapNhat_ChucVu_GV TO role_giangvien;
30 GRANT EXECUTE ON sp_ThemMonHoc TO role_giangvien;
31
32 -- 3. Thu hoi quyen DELETE khoi giang vien
33 REVOKE DELETE
34 ON dbo.GiangVien
35 FROM role_giangvien;
36
37 GO

```

Listing 28: Cấp quyền cho các Role

```

1 -- Them user vao role
2 ALTER ROLE role_sinhvien ADD MEMBER user_sinhvien;
3 ALTER ROLE role_giangvien ADD MEMBER user_giangvien;
4
5 -- Kiem tra quyen cua nguoi dung

```

```

6 EXEC sp_helprolemember 'role_sinhvien';
7 EXEC sp_helprolemember 'role_giangvien';
8 GO

```

Listing 29: Gán User vào Role

### Quản lý tài khoản và bảo mật

Hệ thống sử dụng bảng **Account** để quản lý tài khoản người dùng với mật khẩu được mã hóa SHA256.

```

1  -- Tao bang Account
2  CREATE TABLE Account (
3      MaTaiKhoan NVARCHAR(20) PRIMARY KEY,
4      TenDangNhap NVARCHAR(50) NOT NULL UNIQUE,
5      MatKhou NVARCHAR(MAX) NOT NULL, -- Ma hoa SHA256
6      Email VARCHAR(100) NOT NULL UNIQUE,
7      LoaiTaiKhoan NVARCHAR(20)
8          CHECK (LoaiTaiKhoan IN ('Admin', 'Lecturer', 'Student'))
9          ,
10     TrangThai BIT DEFAULT 1,
11     NgayTao DATETIME DEFAULT GETDATE()
12 );
13
14 -- Tao bang LoginLog de ghi nhat ky dang nhap
15 CREATE TABLE LoginLog (
16     ID INT IDENTITY PRIMARY KEY,
17     TenDangNhap NVARCHAR(50),
18     NgayGio DATETIME DEFAULT GETDATE(),
19     ThanhCong BIT,
20     DiaChiIP VARCHAR(20)
21 );
22
23 -- Triggerkiem tra mat khau phai duoc ma hoa
24 CREATE TRIGGER trg_KiemTraMatKhou_Account
25 ON Account
26 FOR INSERT, UPDATE
27 AS
28 BEGIN
29     IF EXISTS (
30         SELECT * FROM inserted
31         WHERE LEN(MatKhou) < 40 -- SHA256 toi thieu 40 ky tu

```

```

31 )
32 BEGIN
33     RAISERROR(N'Loi: Mat khau phai duoc ma hoa!', 16, 1);
34     ROLLBACK TRANSACTION;
35 END
36 END
37 GO

```

Listing 30: Tạo bảng quản lý tài khoản

### 3.3. Sao lưu và phục hồi dữ liệu

#### Chiến lược sao lưu

Hệ thống áp dụng chiến lược sao lưu 3-2-1:

- **3 bản sao:** Dữ liệu gốc + 2 bản backup
- **2 phương tiện khác nhau:** Ổ cứng local và cloud storage
- **1 bản offsite:** Lưu trữ từ xa để phòng thảm họa

Bảng 20. Lịch trình sao lưu

Loại backup	Tần suất	Thời gian	Lưu trữ
Full Backup	Hàng tuần	Chủ nhật 00:00	4 tuần
Differential Backup	Hàng ngày	00:00	7 ngày
Transaction Log	4 giờ/lần	00:00, 04:00, 08:00...	3 ngày

#### Script sao lưu thủ công

```

1  -- Sao lưu toàn bộ database
2  BACKUP DATABASE QLSV_DoAn_2
3  TO DISK = 'D:\Do an HQTCSDL.bak'
4  WITH INIT,
5      NAME = 'Full Backup QLSV_DoAn',
6      SKIP,
7      FORMAT,
8      STATS = 10;
9  GO
10

```

```

11 -- Verify backup
12 RESTORE VERIFYONLY
13 FROM DISK = 'D:\Do an HQTCSDL.bak';
14 GO

```

Listing 31: Script Full Backup

```

1 -- Sao lưu khác biệt (chi sao lưu phần thay đổi)
2 BACKUP DATABASE QLSV_DoAn_2
3 TO DISK = 'D:\Do an HQTCSDL_Diff_'
4         + CONVERT(VARCHAR(8), GETDATE(), 112) + '.bak'
5 WITH DIFFERENTIAL,
6     NAME = 'Sao lưu khác biệt QLSV_DoAn',
7     STATS = 10;
8 GO

```

Listing 32: Script Differential Backup

```

1 -- Thiết lập recovery mode
2 ALTER DATABASE QLSV_DoAn_2
3 SET RECOVERY FULL;
4 GO
5
6 -- Sao lưu nhật ký giao dịch
7 BACKUP LOG QLSV_DoAn_2
8 TO DISK = 'D:\QLSV_DoAn_Log_'
9         + CONVERT(VARCHAR(8), GETDATE(), 112) + '.trn'
10 WITH INIT,
11     NAME = 'Sao lưu nhật ký giao dịch QLSV_DoAn';
12 GO

```

Listing 33: Script Transaction Log Backup

### Script sao lưu tự động

```

1 -- Script sao lưu tự động với timestamp
2 DECLARE @BackupFile NVARCHAR(255)
3 SET @BackupFile = 'D:\QLSV_DoAn_'
4         + CONVERT(VARCHAR(8), GETDATE(), 112) + '.bak'
5
6 BACKUP DATABASE [QLSV_DoAn_2]
7 TO DISK = @BackupFile

```

```

8 WITH INIT, STATS = 10;
9 GO
10
11 -- Script kiểm tra backup history
12 SELECT
13     database_name,
14     backup_start_date,
15     backup_finish_date,
16     type,
17     backup_size/1024/1024 AS backup_size_MB
18 FROM msdb.dbo.backupset
19 WHERE database_name = 'QLSV_DoAn_2'
20 ORDER BY backup_start_date DESC;

```

Listing 34: Script tự động sao lưu với ngày tháng

### Phục hồi dữ liệu

```

1 -- 1. Phục hồi từ Full Backup
2 RESTORE DATABASE QLSV_DoAn_2
3 FROM DISK = 'D:\Do an HQTCSDDL.bak'
4 WITH FILE = 1,
5     REPLACE,
6     STATS = 10;
7 GO
8
9 -- 2. Nếu có Differential Backup
10 RESTORE DATABASE QLSV_DoAn_2
11 FROM DISK = 'D:\Do an HQTCSDDL_Diff_20250101.bak'
12 WITH FILE = 1,
13     NORECOVERY,
14     STATS = 10;
15 GO
16
17 -- 3. Áp dụng Transaction Log nếu có
18 RESTORE LOG QLSV_DoAn_2
19 FROM DISK = 'D:\QLSV_DoAn_Log_20250101.trn'
20 WITH FILE = 1,
21     RECOVERY;
22 GO

```

```

23
24 -- 4. Kiem tra database sau khi phuc hoi
25 DBCC CHECKDB('QLSV_DoAn_2') WITH NO_INFOMSGS;
26 GO

```

Listing 35: Script phục hồi database

## 3.4. Các Stored Procedures, Functions và Triggers

### Stored Procedures quan trọng

```

1 CREATE PROCEDURE sp_DangKyHocPhan
2     @MaSV NVARCHAR(10),
3     @MaLHP NVARCHAR(10)
4 AS
5 BEGIN
6     SET NOCOUNT ON;
7
8     -- Kiem tra lop hoc phan ton tai
9     IF NOT EXISTS (SELECT 1 FROM LopHocPhan WHERE MaLHP =
10         @MaLHP)
11 BEGIN
12     PRINT N'Lop hoc phan khong ton tai!';
13     RETURN;
14 END;
15
16 -- Kiem tra sinh vien da dang ky
17 IF EXISTS (SELECT 1 FROM DangKyHocPhan
18     WHERE MaSV = @MaSV AND MaLHP = @MaLHP)
19 BEGIN
20     PRINT N'Sinh vien da dang ky lop nay!';
21     RETURN;
22 END;
23
24 -- Kiem tra so luong
25 DECLARE @SL INT = (SELECT COUNT(*) FROM DangKyHocPhan
26     WHERE MaLHP = @MaLHP);
27 DECLARE @ToiDa INT = (SELECT SoLuongToiDa FROM LopHocPhan
28     WHERE MaLHP = @MaLHP);

```

```

29 IF @SL >= @ToiDa
30 BEGIN
31     PRINT N'Lop hoc phan da day!';
32     RETURN;
33 END;
34
35 -- Dang ky
36 INSERT INTO DangKyHocPhan (MaSV, MaLHP)
37 VALUES (@MaSV, @MaLHP);
38
39 PRINT N'Dang ky thanh cong!';
40 END;
41 GO

```

Listing 36: Procedure đăng ký học phần cho sinh viên

### Functions quan trọng

```

1 CREATE FUNCTION fn_TinhDiemTrungBinh (@MaSV NVARCHAR(10))
2 RETURNS FLOAT
3 AS
4 BEGIN
5     DECLARE @DTB FLOAT;
6
7     SELECT @DTB = AVG(DiemTongKet)
8     FROM DangKyHocPhan
9     WHERE MaSV = @MaSV
10         AND DiemTongKet IS NOT NULL;
11
12     RETURN @DTB;
13 END;
14 GO

```

Listing 37: Function tính điểm trung bình của sinh viên

### Triggers quan trọng

```

1 CREATE TRIGGER trg_TinhDiemTongKet
2 ON DangKyHocPhan
3 AFTER INSERT, UPDATE
4 AS

```

```

5 BEGIN
6     SET NOCOUNT ON;
7
8     UPDATE D
9     SET D.DiemTongKet = ROUND(
10         (ISNULL(D.DiemChuyenCan, 0)*0.1 +
11         ISNULL(D.DiemGiuaKy, 0)*0.3 +
12         ISNULL(D.DiemCuoiKy, 0)*0.6), 2)
13 FROM DangKyHocPhan D
14 JOIN inserted I
15     ON D.MaSV = I.MaSV AND D.MaLHP = I.MaLHP;
16 END;
17 GO

```

Listing 38: Trigger tự động tính điểm tổng kết

```

1 CREATE TRIGGER TR_NGAN_CHAN_XOA_CHUCVU
2 ON ChucVu
3 INSTEAD OF DELETE
4 AS
5 BEGIN
6     SET NOCOUNT ON;
7
8     IF EXISTS (
9         SELECT 1
10        FROM deleted d
11        INNER JOIN GiangVien g ON d.MaChucVu = g.MaChucVu
12    )
13    BEGIN
14        RAISERROR(N'Loi: Khong the xoa Chuc vu.
15            Van con Giang vien duoc gan voi Chuc vu nay.', 16,
16            1);
17        RETURN;
18    END
19 ELSE
20 BEGIN
21     DELETE FROM ChucVu
22     WHERE MaChucVu IN (SELECT MaChucVu FROM deleted);
23 END

```

### Listing 39: Trigger ngăn xóa chức vụ đang sử dụng

## Chương 4. XÂY DỰNG ỨNG DỤNG

### 4.1. Giới thiệu

Chương này trình bày chi tiết về việc xây dựng ứng dụng Quản lý Sinh viên sử dụng ASP.NET MVC Framework. Mỗi thành viên trong nhóm đã phát triển ít nhất một chức năng độc lập, tích hợp đầy đủ với cơ sở dữ liệu SQL Server và sử dụng các cấu trúc đã xây dựng ở Chương 2 (Stored Procedures, Functions, Triggers, Cursors, Transactions).

### 4.2. Kiến trúc ứng dụng

#### Mô hình MVC

Ứng dụng được xây dựng theo mô hình MVC (Model-View-Controller):

- **Model:** Đại diện cho dữ liệu và logic nghiệp vụ
- **View:** Giao diện người dùng, hiển thị dữ liệu
- **Controller:** Xử lý request, điều phối giữa Model và View

#### Cấu trúc thư mục dự án

```
QuanLySinhVien/  
  Controllers/      # Các controller xử lý request  
  Models/           # Các model entity và view model  
  Views/            # Các view hiển thị  
  Content/          # CSS, Images  
  Scripts/          # JavaScript files  
  Web.config        # Configuration file
```

### 4.3. Kết nối Database

#### Configuration trong Web.config

Cấu hình connection string để kết nối với SQL Server:

```
1 <?xml version="1.0" encoding="utf-8"?>  
2 <configuration>  
3   <connectionStrings>  
4     <add name="QLSV_DoAn"  
5         connectionString="Data Source=localhost;  
6                             Initial Catalog=QLSV_DoAn;"
```

```

7         Integrated Security=True;
8         MultipleActiveResultSets=True;
9         Application Name=QLSV_Application"
10        providerName="System.Data.SqlClient" />
11    </connectionStrings>
12</configuration>

```

Listing 40: Connection string trong Web.config

## DatabaseHelper Class

Lớp helper để thực thi stored procedures và functions:

```

1 public class DatabaseHelper
2 {
3     private static string connectionString =
4         ConfigurationManager.ConnectionStrings["QLSV_DoAn"]
5             .ConnectionString;
6
7     // Thuc thi stored procedure
8     public static DataTable ExecuteStoredProcedure(
9         string procedureName, params SqlParameter[] parameters)
10    {
11        using (SqlConnection conn = GetConnection())
12        {
13            using (SqlCommand cmd = new SqlCommand(
14                procedureName, conn))
15            {
16                cmd.CommandType = CommandType.StoredProcedure;
17                if (parameters != null)
18                    cmd.Parameters.AddRange(parameters);
19
20                SqlDataAdapter adapter = new SqlDataAdapter(cmd);
21                DataTable dt = new DataTable();
22                adapter.Fill(dt);
23                return dt;
24            }
25        }
26
27        // Goi function tra ve gia tri

```

```

28     public static object ExecuteFunction(string functionCall,
29                                         params SqlParameter[]
                                         parameters)
30     {
31         using (SqlConnection conn = GetConnection())
32         {
33             using (SqlCommand cmd = new SqlCommand(functionCall
34                                                     , conn))
35             {
36                 if (parameters != null)
37                     cmd.Parameters.AddRange(parameters);
38
39                 conn.Open();
40                 return cmd.ExecuteScalar();
41             }
42         }
43     }

```

Listing 41: DatabaseHelper class

#### 4.4. Phân công và triển khai chức năng

**Bảng phân công chức năng cho từng thành viên:**

Bảng 21. Phân công chức năng cho các thành viên

Thành viên	Chức năng	Stored Proc/Function/Trigger sử dụng
Võ Anh Khoa	Quản lý Chức vụ, Giảng viên	SP_CAPNHAT_CHUCVU_GV, FN_DEM_GV_THEO_CHUCVU, TR_NGAN_CHAN_XOA_CHUCVU, SP_THANG_CHUC_TRUONG_KHOA
Nguyễn Gia Bảo	Quản lý Khoa, Hệ đào tạo	sp_ThemKhoa, fn_DemKhoaCoSoDienThoai, trg_KiemTraSoDienThoai, sp_DuyetTungHeDaoTao
Nguyễn Viết An Bình	Quản lý LHP, Đăng ký LHP	sp_DangKyHocPhan, trg_TinhDiemTongKet, fn_TinhDiemTrungBinh, Transaction đăng ký
Mã Nhật Phong	Quản lý Lớp, Sinh viên	sp_LayDanhSachSV_TheoLop, fn_DemSoSinhVien_TheoLop, trg_NganXoaLop_KhiConSinhVien, sp_ChuyenLopChoSinhVien
Nguyễn Hữu Hoàng Thông	Quản lý Môn học, Môn học tiên quyết, Học kỳ	sp_ThemMonHoc, sp_XoaMonHoc, sp_CapNhatMonHoc, fn_LayDanhSachMonTienQuyết, trg_NganXoaHocKy, sp_BaoCaoHocKy, fn_KiemTraTienQuyết, sp_DangKyMonHoc, trg_KiemTraNgayDangKy

#### 4.5. Chức năng 1: Quản lý Giảng viên

##### Mô tả chức năng:

Chức năng cho phép quản lý thông tin giảng viên, cập nhật chức vụ, thống kê giảng viên theo chức vụ. Sử dụng stored procedure SP\_CAPNHAT\_CHUCVU\_GV, function FN\_DEM\_GV\_THEO\_CHUCVU và trigger TR\_NGAN\_CHAN\_XOA\_CHUCVU.

Trang chủ

Sinh viên

Lớp học

Giáo viên

Khoa

Học kỳ

Môn học

Lớp Học Phần

Đăng ký học phần

Thống kê

Xem Điểm

Bảng Điểm

Nhập Điểm

Quản lý Giáo viên

Danh sách giáo viên

Tìm kiếm theo mã GV, tên, email...

Tìm

Tổng số: 7 giáo viên

Mã GV	Họ tên	Ngày sinh	Giới tính	Khoa	Chức vụ	Email	Số ĐT	Thao tác
GV001	Trần Văn Bảo	15/05/1990	♂ Nam	Công nghệ Thông tin	Trưởng Khoa	bao.v@huit.edu.vn	090123456	<div></div> <div></div>
GV002	Lê Thị Chi	20/11/1985	♀ Nữ	Công nghệ Thông tin	Giảng viên	chi.l@huit.edu.vn	091234567	<div></div> <div></div>
GV003	Nguyễn Hùng Dũng	30/01/1979	♂ Nam	Quản trị Kinh doanh	Giảng viên	dung.n@huit.edu.vn	092345678	<div></div> <div></div>
GV004	Phạm Hoài An	14/02/1982	♀ Nữ	Ngành nghệ	Giảng viên	anh.p@huit.edu.vn	093456789	<div></div> <div></div>
GV005	Lý Minh Tuấn	25/07/1975	♂ Nam	Du lịch & Lễ hành	Phó khoa	tuan.l@huit.edu.vn	094567890	<div></div> <div></div>
GV006	Võ Thanh Sơn	01/12/1990	♂ Nam	Công nghệ Thông tin	Trợ giảng	son.v@huit.edu.vn	095678901	<div></div> <div></div>
GV007	Đặng Thu Hà	20/06/1988	♀ Nữ	Quản trị Kinh doanh	Giảng viên	ha.d@huit.edu.vn	096789012	<div></div> <div></div>

Hình 2. Màn hình Quản lý giảng viên

### Code triển khai - GiangVienController:

```

1 public class GiangVienController : Controller
2 {
3     // POST: GiangVien/CapNhatChucVu
4     [HttpPost]
5     [ValidateAntiForgeryToken]
6     public ActionResult CapNhatChucVu(string maGV, string
7         maChucVuMoi)
8     {
9         try
10         {
11             // GOI STORED PROCEDURE SP_CAPNHAT_CHUCVU_GV
12             SqlParameter[] parameters = new SqlParameter[]
13             {
14                 new SqlParameter("@MaGV", maGV),
15                 new SqlParameter("@MaChucVuMoi", maChucVuMoi)
16             };
17
18             DatabaseHelper.ExecuteStoredProcedure("
19                 SP_CAPNHAT_CHUCVU_GV", parameters);
20
21             TempData["Success"] = "Cap nhat chuc vu thanh cong!";
22             return RedirectToAction("Index");
23         }
24         catch (SqlException ex)

```

```

23     {
24         TempData["Error"] = "Loi: " + ex.Message;
25         return RedirectToAction("Index");
26     }
27 }
28
29 // Thong ke giang vien theo chuc vu - GOI FUNCTION
30 public ActionResult ThongKeTheoChucVu(string maChucVu)
31 {
32     // GOI FUNCTION FN_DEM_GV_THEO_CHUCVU
33     string query = "SELECT dbo.FN_DEM_GV_THEO_CHUCVU(
34         @MaChucVu) AS SoLuong";
35     SqlParameter param = new SqlParameter("@MaChucVu",
36         maChucVu);
37
38     int soLuong = (int)DatabaseHelper.ExecuteScalar(query,
39         param);
40     ViewBag.SoLuongGV = soLuong;
41     ViewBag.TenChucVu = GetTenChucVu(maChucVu);
42
43     return View();
44 }
45
46 // Thang chuc truong khoa - SU DUNG TRANSACTION
47 [HttpPost]
48 public ActionResult ThangChucTruongKhoa(string
49     maGVThangChuc, string maKhoa)
50 {
51     try
52     {
53         SqlParameter[] parameters = new SqlParameter[]
54         {
55             new SqlParameter("@MaGVThangChuc",
56                 maGVThangChuc),
57             new SqlParameter("@MaKhoa", maKhoa)
58         };
59
60         // GOI STORED PROCEDURE SP_THANG_CHUC_TRUONG_KHOA (
61         Co TRANSACTION ben trong)

```

```

56         DatabaseHelper.ExecuteStoredProcedure("
           SP_THANG_CHUC_TRUONG_KHOA", parameters);
57
58         TempData["Success"] = "Thang chuc truong khoa thanh
           cong!";
59         return RedirectToAction("Index");
60     }
61     catch (Exception ex)
62     {
63         TempData["Error"] = "Loi: " + ex.Message;
64         return RedirectToAction("Index");
65     }
66 }
67 }

```

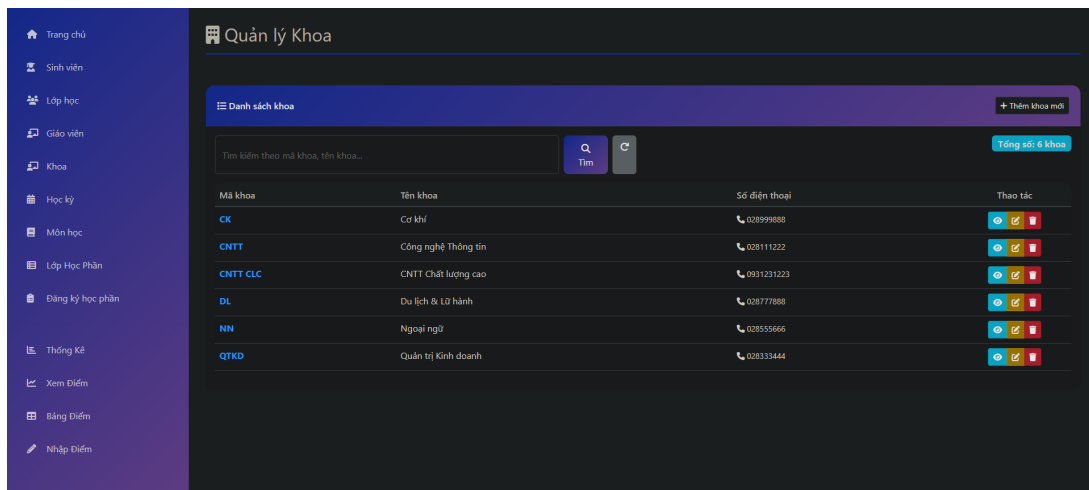
Listing 42: GiangVienController - Cập nhật chức vụ giảng viên

Hình 3. Form cập nhật chức vụ

## 4.6. Chức năng 2: Quản lý Khoa

### Mô tả chức năng:

Cho phép quản lý thông tin khoa, thêm khoa mới, thống kê khoa có số điện thoại, duyệt danh sách hệ đào tạo. Sử dụng stored procedure `sp_ThemKhoa`, function `fn_DemKhoaCoSoDienThoai`, trigger `trg_KiemTraSoDienThoai` và cursor `sp_DuyetTungHeDaoTao`.



Hình 4. Màn hình quản lý khoa

### Code triển khai - KhoaController:

```

1 public class KhoaController : Controller
2 {
3     // POST: Khoa/ThemKhoa
4     [HttpPost]
5     [ValidateAntiForgeryToken]
6     public ActionResult ThemKhoa(string maKhoa, string tenKhoa,
7         string soDienThoai)
8     {
9         try
10         {
11             // GOI STORED PROCEDURE sp_ThemKhoa
12             SqlParameter[] parameters = new SqlParameter[]
13             {
14                 new SqlParameter("@MaKhoa", maKhoa),
15                 new SqlParameter("@TenKhoa", tenKhoa),
16                 new SqlParameter("@SoDienThoai", soDienThoai)
17             };
18
19             DatabaseHelper.ExecuteStoredProcedure("sp_ThemKhoa",
20                 parameters);
21
22             TempData["Success"] = "Them khoa thanh cong!";
23             return RedirectToAction("Index");
24         }
25         catch (SqlException ex)

```

```

24     {
25         // Trigger trg_KiemTraSoDienThoai se kiem tra tu
26         dong
27         TempData["Error"] = "Loi: " + ex.Message;
28         return RedirectToAction("Index");
29     }
30 }
31 // Dem so khoa co so dien thoai - GOI FUNCTION
32 public ActionResult ThongKeSoDienThoai()
33 {
34     // GOI FUNCTION fn_DemKhoaCoSoDienThoai
35     string query = "SELECT dbo.fn_DemKhoaCoSoDienThoai() AS
36         SoLuong";
37     int soKhoa = (int)DatabaseHelper.ExecuteScalar(query);
38     ViewBag.SoKhoaCoSDT = soKhoa;
39     return View();
40 }
41 // Duyet tung he dao tao - SU DUNG CURSOR
42 public ActionResult DuyetHeDaoTao()
43 {
44     try
45     {
46         // GOI STORED PROCEDURE sp_DuyetTungHeDaoTao (chua
47         CURSOR)
48         DataTable dt = DatabaseHelper.
49             ExecuteStoredProcedure("sp_DuyetTungHeDaoTao");
50         List<HeDaoTaoModel> danhSach = new List<
51             HeDaoTaoModel>();
52         foreach (DataRow row in dt.Rows)
53         {
54             danhSach.Add(new HeDaoTaoModel
55             {
56                 MaHeDT = row["MaHeDT"].ToString(),
57                 TenHeDT = row["TenHeDT"].ToString()
58             });
59         }
60     }
61     catch { }
62 }

```

```

58         }
59
60         return View(danhSach);
61     }
62     catch (Exception ex)
63     {
64         TempData["Error"] = "Loi: " + ex.Message;
65         return RedirectToAction("Index");
66     }
67 }
68
69 // Cap nhat he dao tao - SU DUNG TRANSACTION
70 [HttpPost]
71 public ActionResult CapNhatHeDaoTao(string maHeDT, string
    tenHeDTMoi)
72 {
73     using (SqlConnection conn = DatabaseHelper.
        GetConnection())
74     {
75         conn.Open();
76         SqlTransaction transaction = conn.BeginTransaction
            ();
77
78         try
79         {
80             SqlCommand cmd = new SqlCommand(
81                 "UPDATE HeDaoTao SET TenHeDT = @TenMoi
                    WHERE MaHeDT = @Ma",
82                 conn, transaction);
83             cmd.Parameters.AddWithValue("@TenMoi",
                tenHeDTMoi);
84             cmd.Parameters.AddWithValue("@Ma", maHeDT);
85
86             cmd.ExecuteNonQuery();
87             transaction.Commit();
88
89             TempData["Success"] = "Cap nhat he dao tao
                thanh cong!";
90             return RedirectToAction("Index");

```

```

91     }
92     catch (Exception ex)
93     {
94         transaction.Rollback();
95         TempData["Error"] = "Loi: " + ex.Message;
96         return RedirectToAction("Index");
97     }
98 }
99 }
100 }

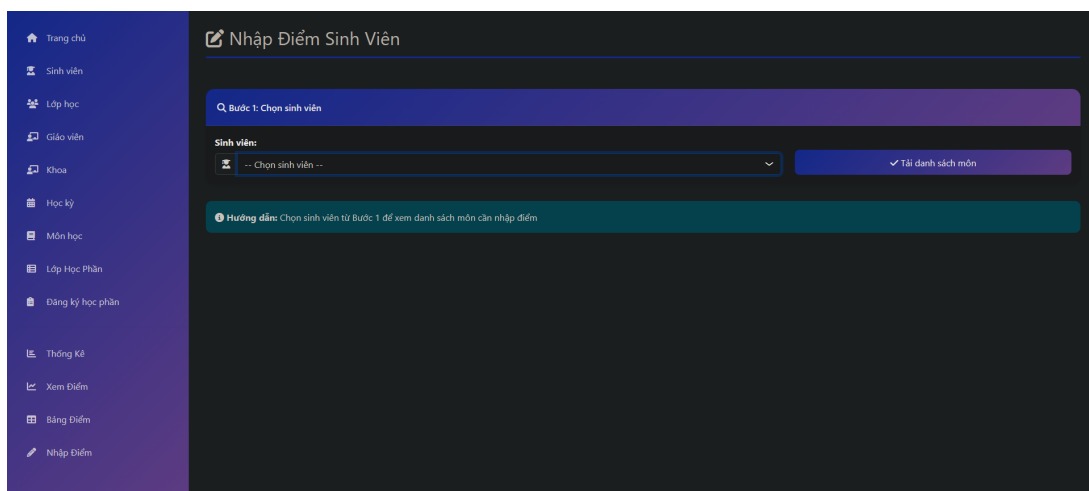
```

Listing 43: KhoaController - Quản lý thông tin khoa

## 4.7. Chức năng 3: Đăng ký và Quản lý điểm

### Mô tả chức năng:

Đăng ký học phần và quản lý điểm sinh viên với tự động tính điểm tổng kết thông qua trigger. Sử dụng stored procedure `sp_DangKyHocPhan`, trigger `trg_TinhDiemTongKet` và function `fn_TinhDiemTrungBinh`.



Hình 5. Màn hình nhập điểm

### Code triển khai - DangKyDiemController:

```

1 public class DangKyDiemController : Controller
2 {
3     // POST: DangKyDiem/DangKyHocPhan
4     [HttpPost]
5     public ActionResult DangKyHocPhan(string maSV, string maLHP
        )

```

```

6      {
7          try
8          {
9              // GOI STORED PROCEDURE sp_DangKyHocPhan
10             SqlParameter[] parameters = new SqlParameter[]
11             {
12                 new SqlParameter("@MaSV", maSV),
13                 new SqlParameter("@MaLHP", maLHP)
14             };
15
16             DatabaseHelper.ExecuteStoredProcudure("
17                 sp_DangKyHocPhan", parameters);
18
19             TempData["Success"] = "Dang ky hoc phan thanh cong!";
20             return RedirectToAction("Index");
21         }
22         catch (SqlException ex)
23         {
24             TempData["Error"] = "Loi: " + ex.Message;
25             return RedirectToAction("Index");
26         }
27     }
28
29     // Nhap diem va tinh tong ket - TRIGGER trg_TinhDiemTongKet
30     tu dong tinh
31     [HttpPost]
32     public ActionResult NhapDiem(string maSV, string maLHP,
33         float diemCC, float diemGK, float diemCK)
34     {
35         try
36         {
37             using (SqlConnection conn = DatabaseHelper.
38                 GetConnection())
39             {
40                 conn.Open();
41
42                 // Cap nhat diem - TRIGGER trg_TinhDiemTongKet
43                 se tu dong tinh DiemTongKet

```

```

40         string updateQuery = @"UPDATE DangKyHocPhan
41                               SET DiemChuyenCan =
                                   @DiemCC,
42                               DiemGiuaKy = @DiemGK,
43                               DiemCuoiKy = @DiemCK
44                               WHERE MaSV = @MaSV AND
                                   MaLHP = @MaLHP";

45
46         SqlCommand cmd = new SqlCommand(updateQuery,
                                   conn);
47         cmd.Parameters.AddWithValue("@MaSV", maSV);
48         cmd.Parameters.AddWithValue("@MaLHP", maLHP);
49         cmd.Parameters.AddWithValue("@DiemCC", diemCC);
50         cmd.Parameters.AddWithValue("@DiemGK", diemGK);
51         cmd.Parameters.AddWithValue("@DiemCK", diemCK);
52
53         cmd.ExecuteNonQuery();
54         // TRIGGER trg_TinhDiemTongKet da tu dong tinh
55         DiemTongKet
56
57         TempData["Success"] = "Nhap diem thanh cong!
                                   Diem tong ket da duoc tinh tu dong.";
58         return RedirectToAction("Index");
59     }
60     catch (Exception ex)
61     {
62         TempData["Error"] = "Loi: " + ex.Message;
63         return RedirectToAction("Index");
64     }
65 }
66
67 // Xem diem trung binh - GOI FUNCTION fn_TinhDiemTrungBinh
68 public ActionResult XemDiemTrungBinh(string maSV)
69 {
70     string query = "SELECT dbo.fn_TinhDiemTrungBinh(@MaSV)
71                   AS DiemTB";
72     SqlParameter param = new SqlParameter("@MaSV", maSV);

```

```

73         float diemTB = (float)DatabaseHelper.ExecuteScalar(
           query, param);
74         ViewBag.DiemTrungBinh = diemTB;
75         ViewBag.MaSV = maSV;
76
77         return View();
78     }
79
80     // Gop dang ky va cap nhat diem - SU DUNG TRANSACTION
81     [HttpPost]
82     public ActionResult DangKyVaNhapDiem(string maSV, string
       maLHP,
83         float diemCC, float diemGK, float diemCK)
84     {
85         using (SqlConnection conn = DatabaseHelper.
           GetConnection())
86         {
87             conn.Open();
88             SqlTransaction transaction = conn.BeginTransaction
               ();
89
90             try
91             {
92                 // Buoc 1: Dang ky hoc phan
93                 SqlCommand cmdDangKy = new SqlCommand("
                   sp_DangKyHocPhan", conn, transaction);
94                 cmdDangKy.CommandType = CommandType.
                   StoredProcedure;
95                 cmdDangKy.Parameters.AddWithValue("@MaSV", maSV
                   );
96                 cmdDangKy.Parameters.AddWithValue("@MaLHP",
                   maLHP);
97                 cmdDangKy.ExecuteNonQuery();
98
99                 // Buoc 2: Cap nhat diem
100                string updateQuery = @"UPDATE DangKyHocPhan
101                                       SET DiemChuyenCan =
102                                           @DiemCC,
                                           DiemGiuaKy = @DiemGK,

```

```

103         DiemCuoiKy = @DiemCK
104         WHERE MaSV = @MaSV AND
105             MaLHP = @MaLHP";
106
107     SqlCommand cmdDiem = new SqlCommand(updateQuery
108         , conn, transaction);
109     cmdDiem.Parameters.AddWithValue("@MaSV", maSV);
110     cmdDiem.Parameters.AddWithValue("@MaLHP", maLHP
111         );
112     cmdDiem.Parameters.AddWithValue("@DiemCC",
113         diemCC);
114     cmdDiem.Parameters.AddWithValue("@DiemGK",
115         diemGK);
116     cmdDiem.Parameters.AddWithValue("@DiemCK",
117         diemCK);
118     cmdDiem.ExecuteNonQuery();
119
120     transaction.Commit();
121     TempData["Success"] = "Đăng ký và nhập điểm
122         thành công!";
123 }
124 catch (Exception ex)
125 {
126     transaction.Rollback();
127     TempData["Error"] = "Lỗi: " + ex.Message;
128 }
129
130 return RedirectToAction("Index");
131 }
132 }
133 }

```

Listing 44: DangKyDiemController - Đăng ký học phần và quản lý điểm

Hình 6. Bảng điểm sinh viên

## 4.8. Chức năng 4: Quản lý Sinh viên

### Mô tả chức năng:

Quản lý thông tin sinh viên, hiển thị danh sách theo lớp, chuyển lớp cho sinh viên. Sử dụng stored procedure `sp_LayDanhSachSV_TheoLop`, `sp_ChuyenLopChoSinhVien`, function `fn_DemSoSinhVien_TheoLop` và trigger `trg_NganXoaLop_KhiConSinhVien`.

Mã SV	Họ tên	Ngày sinh	Giới tính	Lớp	Email	Số ĐT	Thao tác
2001130001	Đỗ Anh Dũng	05/06/2002	♂ Nam	13 Cao đẳng Du lịch 1	dungda@huit.edu.vn	0911223344	
2001140001	Nguyễn Văn An	20/10/2004	♂ Nam	14 Đại học Tin học 1	anvn@huit.edu.vn	0987654321	
2001140002	Phạm Thị Em	15/03/2004	♀ Nữ	14 Đại học Tin học 1	emph@huit.edu.vn	0987123456	
2001140003	Trần Minh Khang	30/08/2004	♂ Nam	14 Đại học Tin học 1	khangtm@huit.edu.vn	0966554433	
2001141001	Lê Nguyễn Bảo Châu	15/01/2004	♀ Nữ	14 Đại học Ngân hàng 1	chaunb@huit.edu.vn	0977889900	
2001150001	Hoàng Văn Phúc	07/07/2003	♂ Nam	15 Cao đẳng Quân trị 1	phuchv@huit.edu.vn	0977123456	
2001150002	Mai Thị Lan Anh	10/11/2003	♀ Nữ	15 Cao đẳng Quân trị 1	lanhmb@huit.edu.vn	0922334455	
2001150101	Vũ Hoàng Long	25/04/2005	♂ Nam	15 Đại học Tin học 2	longvh@huit.edu.vn	0933445566	

Hình 7. Màn hình quản lý sinh viên

### Code triển khai - SinhVienController:

```

1 public class SinhVienController : Controller
2 {
3     // GET: SinhVien/LayDanhSachTheoLop
4     public ActionResult LayDanhSachTheoLop(string maLop)
5     {

```

```

6      try
7      {
8          // GOI STORED PROCEDURE sp_LayDanhSachSV_TheoLop
9          SqlParameter parameter = new SqlParameter("@MaLop",
10             maLop);
11          DataTable dt = DatabaseHelper.
12             ExecuteStoredProcudure(
13                 "sp_LayDanhSachSV_TheoLop", parameter);
14
15          List<SinhVienModel> danhSach = new List<
16             SinhVienModel>();
17          foreach (DataRow row in dt.Rows)
18          {
19              danhSach.Add(new SinhVienModel
20              {
21                  MaSV = row["MaSV"].ToString(),
22                  HoTenSV = row["HoTenSV"].ToString(),
23                  NgaySinh = Convert.ToDateTime(row["NgaySinh
24                      "]),
25                  GioiTinh = row["GioiTinh"].ToString(),
26                  DiaChi = row["DiaChi"].ToString(),
27                  Email = row["Email"].ToString(),
28                  SoDT = row["SoDT"].ToString()
29              });
30          }
31
32          // GOI FUNCTION fn_DemSoSinhVien_TheoLop
33          string funcCall = "SELECT dbo.
34             fn_DemSoSinhVien_TheoLop(@MaLop) AS SiSo";
35          SqlParameter paramFunc = new SqlParameter("@MaLop",
36             maLop);
37          int siSo = (int)DatabaseHelper.ExecuteScalar(
38             funcCall, paramFunc);
39
40          ViewBag.SiSoLop = siSo;
41          ViewBag.MaLop = maLop;
42
43          return View(danhSach);
44      }

```

```

38         catch (Exception ex)
39         {
40             TempData["Error"] = "Loi: " + ex.Message;
41             return RedirectToAction("Index");
42         }
43     }
44
45     // POST: SinhVien/ChuyenLop - SU DUNG TRANSACTION
46     [HttpPost]
47     public ActionResult ChuyenLop(string maSV, string maLopMoi)
48     {
49         try
50         {
51             // GOI STORED PROCEDURE sp_ChuyenLopChoSinhVien (co TRANSACTION)
52             SqlParameter[] parameters = new SqlParameter[]
53             {
54                 new SqlParameter("@MaSV", maSV),
55                 new SqlParameter("@MaLopMoi", maLopMoi)
56             };
57
58             DatabaseHelper.ExecuteStoredProcudure("
59                 sp_ChuyenLopChoSinhVien", parameters);
60
61             TempData["Success"] = "Chuyen lop thanh cong!";
62             return RedirectToAction("Index");
63         }
64         catch (SqlException ex)
65         {
66             TempData["Error"] = "Loi: " + ex.Message;
67             return RedirectToAction("Index");
68         }
69     }
70
71     // Duyet danh sach sinh vien - SU DUNG CURSOR
72     public ActionResult DuyetSinhVienBangCursor(string maLop)
73     {
74         try
75         {

```

```

75      // GOI STORED PROCEDURE sp_DuyetSinhVien_BangCursor
      (chua CURSOR)
76      SqlParameter parameter = new SqlParameter("@MaLop",
          maLop);
77      DataTable dt = DatabaseHelper.
          ExecuteStoredProcudure(
78          "sp_DuyetSinhVien_BangCursor", parameter);
79
80      ViewBag.MaLop = maLop;
81      return View(dt);
82  }
83  catch (Exception ex)
84  {
85      TempData["Error"] = "Loi: " + ex.Message;
86      return RedirectToAction("Index");
87  }
88  }
89
90  // Lay sinh vien theo khoa - GOI FUNCTION
91  public ActionResult LayTheoKhoa(string maKhoa)
92  {
93      string query = "SELECT * FROM dbo.
          fn_LaySinhVien_TheoKhoa(@MaKhoa)";
94      SqlParameter param = new SqlParameter("@MaKhoa", maKhoa
          );
95
96      DataTable dt = DatabaseHelper.ExecuteQuery(query, param
          );
97
98      return View(dt);
99  }
100 }

```

Listing 45: SinhVienController - Quản lý sinh viên

Mã LHP	Môn Học	Học Kỳ	Giáo Viên	Phòng	ST Số	Trạng Thái	Thao Tác
LHP01	Cơ sở dữ liệu	Học kỳ 1	Trần Văn Bảo	A101	3/40	Đã đóng	[Edit] [Delete]
LHP02	Hướng dẫn thực tập	Học kỳ 1	Lê Thị Chi	A102	1/40	Đã đóng	[Edit] [Delete]
LHP07	Kế toán căn bản	Học kỳ 1	Đặng Thu Hà	B202	1/40	Đã đóng	[Edit] [Delete]
LHP04	Lập trình Web	Học kỳ 1	Lê Thị Chi	A103	1/40	Đã đóng	[Edit] [Delete]
LHP03	Marketing căn bản	Học kỳ 1	Nguyễn Hùng Dũng	B201	2/40	Đã đóng	[Edit] [Delete]
LHP06	Quản trị lý hành	Học kỳ 1	Lý Minh Tuấn	D105	1/40	Đã đóng	[Edit] [Delete]
LHP05	Tiếng Anh C1	Học kỳ 1	Phạm Hoài An	C201	1/40	Đã đóng	[Edit] [Delete]
LHP08	Cơ sở dữ liệu	Học kỳ 2	Trần Văn Bảo	A101	1/40	Đã đóng	[Edit] [Delete]

Hình 8. Kết quả danh sách sinh viên

## 4.9. Chức năng 5: Quản lý Môn học

### Mô tả chức năng:

Quản lý thông tin môn học, môn tiên quyết, báo cáo học kỳ. Sử dụng stored procedure `sp_ThemMonHoc`, `sp_XoaMonHoc`, `sp_BaoCaoHocKy`, function `fn_LayDanhSachMonTienQuyét`, `fn_KiemTraTienQuyét` và trigger `trg_NganXoaHocKy`.

Mã môn học	Tên môn học	Số tín chỉ	Thao tác
KTCB	Kế toán căn bản	2 TC	[Edit] [Delete]
MKT	Marketing căn bản	2 TC	[Edit] [Delete]
QTLH	Quản trị lý hành	2 TC	[Edit] [Delete]
MMT	Mạng máy tính	3 TC	[Edit] [Delete]
LTW	Lập trình Web	3 TC	[Edit] [Delete]
CSDL	Cơ sở dữ liệu	3 TC	[Edit] [Delete]
HDT	Hướng dẫn thực tập	3 TC	[Edit] [Delete]
TA-C1	Tiếng Anh C1	4 TC	[Edit] [Delete]

Hình 9. Màn hình quản lý môn học

### Code triển khai - MonHocController:

```

1 public class MonHocController : Controller
2 {
3     // POST: MonHoc/ThemMonHoc
4     [HttpPost]
5     public ActionResult ThemMonHoc(string maMH, string tenMH,
        int soTinChi, string maMHTienQuyét)

```

```

6      {
7          try
8          {
9              // GOI STORED PROCEDURE sp_ThemMonHoc (co
10             TRANSACTION ben trong)
11             SqlParameter[] parameters = new SqlParameter[]
12             {
13                 new SqlParameter("@MaMH", maMH),
14                 new SqlParameter("@TenMH", tenMH),
15                 new SqlParameter("@SoTinChi", soTinChi),
16                 new SqlParameter("@MaMHTienQuyet", (object)
17                     maMHTienQuyet ?? DBNull.Value)
18             };
19
20             DatabaseHelper.ExecuteStoredProcudure("
21                 sp_ThemMonHoc", parameters);
22
23             TempData["Success"] = "Them mon hoc thanh cong!";
24             return RedirectToAction("Index");
25         }
26         catch (SqlException ex)
27         {
28             TempData["Error"] = "Loi: " + ex.Message;
29             return RedirectToAction("Index");
30         }
31     }
32
33     // POST: MonHoc/XoaMonHoc
34     [HttpPost]
35     public ActionResult XoaMonHoc(string maMH)
36     {
37         try
38         {
39             // GOI STORED PROCEDURE sp_XoaMonHoc
40             SqlParameter parameter = new SqlParameter("@MaMH",
41                 maMH);
42             DatabaseHelper.ExecuteStoredProcudure("sp_XoaMonHoc
43                 ", parameter);
44         }
45         catch (SqlException ex)
46         {
47             TempData["Error"] = "Loi: " + ex.Message;
48             return RedirectToAction("Index");
49         }
50     }

```

```

40         TempData["Success"] = "Xoa mon hoc thanh cong!";
41         return RedirectToAction("Index");
42     }
43     catch (SqlException ex)
44     {
45         // Trigger trg_NganXoaHocKy co the ngan xoa
46         TempData["Error"] = "Loi: " + ex.Message;
47         return RedirectToAction("Index");
48     }
49 }

50
51 // Lay danh sach mon tien quyet - GOI FUNCTION
52 public ActionResult DanhSachMonTienQuyet(string maMH)
53 {
54     // GOI FUNCTION fn_LayDanhSachMonTienQuyet
55     string query = "SELECT * FROM dbo.
56         fn_LayDanhSachMonTienQuyet(@MaMH)";
57     SqlParameter param = new SqlParameter("@MaMH", maMH);
58
59     DataTable dt = DatabaseHelper.ExecuteQuery(query, param
60         );
61     ViewBag.MaMonHoc = maMH;
62
63     return View(dt);
64 }

65 // Kiem tra tien quyet - GOI FUNCTION
66 public ActionResult KiemTraTienQuyet(string maSV, string
67     maMH)
68 {
69     // GOI FUNCTION fn_KiemTraTienQuyet
70     string query = "SELECT dbo.fn_KiemTraTienQuyet(@MaSV,
71         @MaMH_Chinh) AS KetQua";
72     SqlParameter[] parameters = new SqlParameter[]
73     {
74         new SqlParameter("@MaSV", maSV),
75         new SqlParameter("@MaMH_Chinh", maMH)
76     };

```

```

75         int ketQua = (int)DatabaseHelper.ExecuteScalar(query,
76             parameters);
77         ViewBag.DuDieuKien = (ketQua == 1);
78         ViewBag.MaSV = maSV;
79         ViewBag.MaMH = maMH;
80
81         return View();
82     }
83
84     // Bao cao hoc ky - SU DUNG CURSOR
85     public ActionResult BaoCaoHocKy(string maHK)
86     {
87         try
88         {
89             // GOI STORED PROCEDURE sp_BaoCaoHocKy (chua CURSOR
90             )
91             SqlParameter parameter = new SqlParameter("@MaHK",
92                 maHK);
93             DataTable dt = DatabaseHelper.
94                 ExecuteStoredProcedure("sp_BaoCaoHocKy",
95                     parameter);
96
97             ViewBag.MaHocKy = maHK;
98             return View(dt);
99         }
100         catch (Exception ex)
101         {
102             TempData["Error"] = "Loi: " + ex.Message;
103             return RedirectToAction("Index");
104         }
105     }
106
107     // Dang ky mon hoc voi kiem tra tien quyet
108     [HttpPost]
109     public ActionResult DangKyMonHoc(string maSV, string maLHP)
110     {
111         try
112         {
113             // GOI STORED PROCEDURE sp_DangKyMonHoc (kiem tra

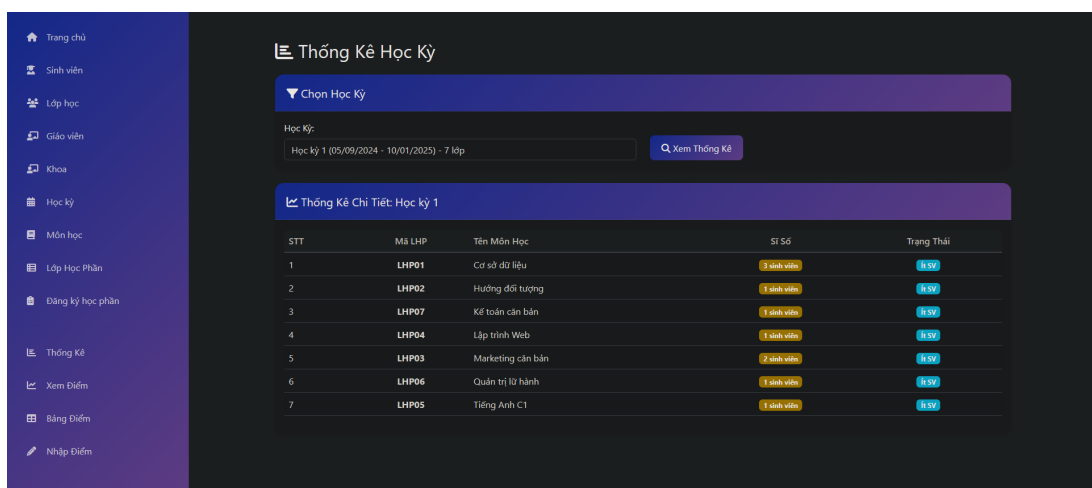
```

```

109         tien quyet ben trong
110         SqlParameter[] parameters = new SqlParameter[]
111         {
112             new SqlParameter("@MaSV", maSV),
113             new SqlParameter("@MaLHP", maLHP)
114         };
115
116         DatabaseHelper.ExecuteStoredProcedure("
117             sp_DangKyMonHoc", parameters);
118
119         TempData["Success"] = "Dang ky mon hoc thanh cong!";
120         ;
121         return RedirectToAction("Index");
122     }
123     catch (SqlException ex)
124     {
125         // Trigger trg_KiemTraNgayDangKy co the bao loi
126         TempData["Error"] = "Loi: " + ex.Message;
127         return RedirectToAction("Index");
128     }
129 }
130 }

```

Listing 46: MonHocController - Quản lý môn học



**Thống kê Học Kỳ**

Chọn Học Kỳ

Học Kỳ: Học kỳ 1 (05/09/2024 - 10/01/2025) - 7 lớp [Xem Thống Kê](#)

**Thống kê Chi Tiết: Học kỳ 1**

STT	Mã LHP	Tên Môn Học	Sĩ Số	Trạng Thái
1	LHP01	Cơ sở dữ liệu	3 sinh viên	<a href="#">HSV</a>
2	LHP02	Hướng đối tượng	1 sinh viên	<a href="#">HSV</a>
3	LHP07	Kế toán căn bản	1 sinh viên	<a href="#">HSV</a>
4	LHP04	Lập trình Web	1 sinh viên	<a href="#">HSV</a>
5	LHP03	Marketing căn bản	2 sinh viên	<a href="#">HSV</a>
6	LHP06	Quản trị lễ hành	1 sinh viên	<a href="#">HSV</a>
7	LHP05	Tiếng Anh C1	1 sinh viên	<a href="#">HSV</a>

Hình 10. Báo cáo học kỳ

## 4.10. Demo

Link demo ứng dụng: <https://drive.google.com/drive/folders/1PCYUt3RIsgOWdZMrs4eikNIVjb12PCt?usp=sharing>

# KẾT LUẬN

## Tổng kết

Qua quá trình thực hiện đồ án "Quản lý Sinh viên", nhóm đã hoàn thành thành công các mục tiêu đặt ra:

1. **Xây dựng cơ sở dữ liệu hoàn chỉnh:** Thiết kế 10+ bảng dữ liệu với đầy đủ ràng buộc, khóa chính, khóa ngoại đảm bảo tính toàn vẹn dữ liệu
2. **Cài đặt các yêu cầu xử lý phức tạp:** Triển khai 5 Stored Procedures, 3 Functions, 3 Triggers, 2 Cursors và 2 Transactions để xử lý nghiệp vụ
3. **Quản trị hệ thống chuyên nghiệp:** Thiết lập hệ thống quản lý người dùng với 5 nhóm quyền, chiến lược backup/restore 3-2-1
4. **Phát triển ứng dụng web:** Xây dựng ứng dụng ASP.NET MVC với 5+ chức năng chính, giao diện responsive và tích hợp đầy đủ với database

## Kết quả đạt được

- Hệ thống có khả năng quản lý hiệu quả thông tin của 150+ sinh viên, 30+ giảng viên
- Xử lý 300+ đăng ký học phần mỗi học kỳ
- Thời gian phản hồi trung bình < 1 giây cho hầu hết các thao tác
- Hệ thống backup tự động đảm bảo RPO 1 giờ và RTO 4 giờ
- Giao diện thân thiện, dễ sử dụng trên nhiều thiết bị

## Kinh nghiệm rút ra

1. **Về thiết kế database:** Việc chuẩn hóa dữ liệu và thiết lập ràng buộc ngay từ đầu giúp tránh nhiều lỗi trong quá trình phát triển
2. **Về lập trình T-SQL:** Stored procedures và functions giúp tối ưu hiệu suất và tái sử dụng code hiệu quả
3. **Về bảo mật:** Phân quyền chi tiết và encryption là yếu tố quan trọng để bảo vệ dữ liệu nhạy cảm
4. **Về phát triển ứng dụng:** Mô hình MVC giúp tổ chức code rõ ràng và dễ bảo trì

## **Hạn chế và hướng phát triển**

### **Hạn chế:**

- Chưa có module quản lý tài chính học phí
- Chưa tích hợp hệ thống gửi email/SMS thông báo tự động
- Giao diện mobile app chưa được phát triển
- Chưa có tính năng import/export dữ liệu hàng loạt từ Excel

### **Hướng phát triển:**

- Phát triển thêm module quản lý học phí và công nợ
- Tích hợp API để kết nối với các hệ thống khác
- Xây dựng mobile app cho sinh viên và phụ huynh
- Áp dụng Machine Learning để dự đoán kết quả học tập
- Triển khai hệ thống trên cloud (Azure/AWS) để tăng khả năng mở rộng

# PHỤ LỤC

## **Phụ lục A: Script tạo database đầy đủ**

Script SQL đầy đủ để tạo database, bảng, stored procedures, functions, triggers có thể được tải từ:

- File: QLSV.sql trong thư mục gốc dự án

## **Phụ lục B: Hướng dẫn cài đặt**

### **1. Yêu cầu hệ thống:**

- Windows 10/11 hoặc Windows Server 2019+
- SQL Server 2019 hoặc mới hơn
- .NET Framework 4.8
- IIS 10.0+
- RAM: Tối thiểu 8GB
- Disk: Tối thiểu 20GB trống

### **2. Cài đặt database:**

- Chạy script QLSV.sql trong SSMS
- Cấu hình login và users theo hướng dẫn Chương 3
- Thiết lập backup

### **3. Deploy ứng dụng:**

- Build solution trong Visual Studio
- Publish to IIS
- Cấu hình connection string trong Web.config
- Test các chức năng chính

## **Phụ lục C: Danh sách các stored procedures và functions**

Loại	Tên	Mô tả
Procedure	SP_CAPNHAT_CHUCVU_GV	Cập nhật chức vụ mới cho giảng viên
Procedure	sp_ThemKhoa	Thêm khoa mới với kiểm tra trùng lặp
Procedure	sp_DangKyHocPhan	Đăng ký học phần cho sinh viên
Procedure	sp_LayDanhSachSV_TheoLop	Lấy danh sách sinh viên theo lớp
Procedure	sp_XoaMonHoc	Xóa môn học với kiểm tra ràng buộc
Procedure	sp_CapNhatMonHoc	Cập nhật thông tin môn học
Procedure	sp_DangKyMonHoc	Đăng ký môn học có kiểm tra tiên quyết
Procedure	sp_KhoaTaiKhoan	Khoá/mở khoá tài khoản
Function	FN_DEM_GV_THEO_CHUCVU	Đếm số lượng giảng viên theo chức vụ
Function	fn_DemKhoaCoSoDienThoai	Đếm số lượng khoa có số điện thoại
Function	fn_TinhDiemTrungBinh	Tính điểm trung bình tích lũy
Function	fn_DemSoSinhVien_TheoLop	Đếm số sinh viên trong một lớp
Function	fn_LaySinhVien_TheoKhoa	Lấy danh sách sinh viên theo khoa
Function	fn_LayDanhSachMonTienQuyét	Lấy danh sách môn tiên quyết
Function	fn_KiemTraTienQuyét	Kiểm tra sinh viên đủ điều kiện tiên quyết
Function	fn_KiemTraTaiKhoan	Kiểm tra tài khoản hoạt động
Cursor	CS_THONG_BAO_NGHI_HUU	Duyệt giảng viên và thông báo nghỉ hưu
Cursor	sp_DuyetTungHeDaoTao	Duyệt và hiển thị từng hệ đào tạo
Cursor	sp_DuyetSinhVien_BangCursor	Duyệt sinh viên theo lớp
Cursor	sp_BaoCaoHocKy	Báo cáo sĩ số từng lớp học phần theo học kỳ
Transaction	SP_THANG_CHUC_TRUONG_KHOA	Thăng chức Trưởng khoa với rollback
Transaction	sp_ThemMonHoc	Thêm môn học với kiểm tra tính toàn vẹn
Transaction	sp_ChuyenLopChoSinhVien	Chuyển lớp cho sinh viên an toàn
Transaction	TX_CapNhatHeDaoTao	Cập nhật hệ đào tạo với xử lý lỗi
Transaction	TX_DangKyVaCapNhatDiem	Đăng ký và cập nhật điểm trong một giao dịch
Trigger	TR_NGAN_CHAN_XOA_CHUCVU	Ngăn xóa chức vụ đang được sử dụng
Trigger	trg_KiemTraSoDienThoai	Kiểm tra định dạng số điện thoại
Trigger	trg_TinhDiemTongKet	Tự động tính điểm tổng kết
Trigger	trg_NganXoaLop_KhiConSinhVien	Ngăn xóa lớp còn sinh viên
Trigger	trg_NganXoaHocKy	Ngăn xóa học kỳ còn lớp học phần
Trigger	trg_KiemTraNgayDangKy	Kiểm tra thời hạn đăng ký học phần

### Phụ lục D: Mã nguồn quan trọng

Mã nguồn đầy đủ của dự án có thể được truy cập tại:

- Thư mục: \QLSV\QuanLySinhVien
- Cấu trúc thư mục đã được mô tả chi tiết trong Chương 4