

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
VIỆN KHOA HỌC KỸ THUẬT BƯU ĐIỆN



BÁO CÁO

Bài Tập Lớn Môn CSDL

Nhóm: Geeks

Đề Tài : Quản lý thông tin cá nhân của sinh viên

Giảng viên hướng dẫn: Phan Lý Huỳnh

Sinh viên thực hiện: 1: Nguyễn Minh Quang: B23DCCC140

2: Nguyễn Ngọc Duy: B23DCCC049

3: Lê Hải An: B23DCCC001

MỤC LỤC

GIỚI THIỆU.....	2
PHẦN I :Mô tả nghiệp vụ	3
1.1. Mô tả yêu cầu bài toán	3
1.2. Mô tả loại người dùng	3
PHẦN II :Thiết kế mô hình liên kết thực thể (ER).....	4
2.1. Mô tả tập thực thể ,thuộc tính	4
2.2. Mối quan hệ qua mô hình ERD	5
PHẦN III : Chuyển mô hình ER sang mô hình quan hệ	6
3.1. Các bước chuyển từ mô hình ER đã thiết kế sang mô hình quan hệ.....	6
3.2. Mô hình quan hệ	7
PHẦN IV :Thiết kế các bảng dữ liệu vật lý.....	8
4.1. Mô tả	8
4.2. Triển khai sơ sở dữ liệu đã thiết kế vào hệ thống vật lý	10
PHẦN V :Truy vấn	14
PHẦN VI :Xây dựng giao diện.....	17
PHẦN VII :Kết luận.....	19
7.1. Đánh giá kết quả vận hành	19

Lời giới thiệu

Song hành với sự phát triển của loài người đó là hệ thống thông tin. Nó luôn đóng vai trò quan trọng, chi phối sâu sắc, nhanh chóng và toàn vẹn đến mọi lĩnh vực của đời sống xã hội. Có thể nói sự phát triển của tin học mang đến cho con người sự tiện dụng, chính xác, nhanh gọn, mang đến cho chúng ta cơ hội nghỉ ngơi, thư giãn.

Trong đó, hệ thống phần mềm quản lý , một mảng của tin học, đóng vai trò quan trọng trong các lĩnh vực hoạt động kinh doanh tài chính , ngân hàng , bưu điện, giáo dục và nhiều loại hình dịch vụ khác nữa.Nó rút ngắn thời gian làm việc, tăng độ chính xác và góp phần tăng giá trị kinh tế.

Trong phạm vi của một bài tập lớn, nhóm em xin trình bày đề tài “***Thiết kế hệ thống quản lý sinh viên***”

Do sự hạn chế về kiến thức cũng như hiểu biết về lĩnh vực quản lý hệ thống , nên bài viết không tránh khỏi thiếu sót. Vì vậy, chúng em mong nhận được sự chỉ bảo của thầy giáo , giúp chúng em có được một bài làm hoàn chỉnh hơn.

I: Mô tả nghiệp vụ

1.1. Mô tả yêu cầu bài toán

❖ *Yêu cầu chức năng :*

-Chức năng quản trị hệ thống: Cho phép người quản trị hệ thống có thể quản lý người sử dụng, phân quyền, quản lý danh mục và vận hành hệ thống.

-Chức năng quản lý thông tin: Cho phép các bộ phận, phòng ban thực hiện cập nhập và quản lý thông tin hoạt động quản lý thông tin hoạt động của đơn vị mình.

-Chức năng tra cứu thông tin: Cho phép người truy cập hệ thống có thể xem các thông tin mà đã được người quản trị phân quyền cho mình.

❖ *Yêu cầu phi chức năng:*

-Giao diện thân thiện, dễ sử dụng.

-Truy xuất nhanh dữ liệu,lưu trữ dữ liệu tốt.

-Tìm kiếm nhanh,thuận tiện.

-Hệ thống bảo mật cao.

-Đáp ứng được nhu cầu nghiệp vụ.

❖ *Yêu cầu miền ứng dụng:*

-Chạy được trên các hệ điều hành khác nhau.

-Hệ quản trị cơ sở dữ liệu tập trung(SQL server).

-Giao diện thiết kế theo một chuẩn nhất định.

1.2. Mô tả loại người dùng

❖ Người dùng được chia thành hai đối tượng chính : Quản lý hệ thống và người truy cập hệ thống để xem thông tin.

❖ Quản lý hệ thống có quyền cập nhập , chỉnh sửa thông tin trong toàn bộ hệ thống với hệ thống “Quản lý sinh viên” người quản lý hệ thống có chức năng cập nhập thông tin về sinh viên như:Họ và tên, ngày sinh, quê quán, ngành học, lớp học, điểm,.....

+*Quản lý sinh viên* : Hệ thống sẽ cập nhập danh sách sinh viên có sẵn phòng đào tạo, thêm sửa thông tin sinh viên và xóa sinh viên trong trường hợp sinh viên bị buộc thôi học hoặc nguyên nhân nào đó.

+*Quản lý môn học* :

- Thêm môn học: Thêm môn học cho sinh viên sau khi có được danh sách các môn học trong học kỳ của sinh viên . Trong trường hợp sinh viên học lại hoặc có thêm môn học khác , bộ môn phải thêm vào và hệ thống quản lý cũng phải thêm môn học cho sinh viên để quản lý điểm môn học đó.
- Xóa môn học: Nếu môn học bị thêm nhầm cho sinh viên , hệ thống có thể xóa đi.

+*Theo dõi xử lý điểm* : Là quá trình cốt lõi của hệ thống nó bao gồm

- Nhập điểm : Hệ thống sẽ nhận điểm từ giáo viên bộ môn và nhập điểm vào hệ thống.
- Sửa điểm : Sau khi báo điểm cho sinh viên nếu có sai sót có thể sửa lại.

❖ Người truy cập hệ thống : Là những người cần lấy thông tin cần thiết từ hệ thống và đối tượng này khi truy cập vào hệ thống để xem thông tin cần có tài khoản .Đối tượng này chỉ có quyền truy cập xem thông tin chứ không thể truy cập, sửa đổi thông tin trong hệ thống .

Phần II: Thiết kế mô hình liên kết thực thể (ER)

2.1. Các thực thể:

a :Sinh viên (Students):

Mã sinh viên (PK) , Họ và tên, Ngày sinh, Giới tính, Địa chỉ, Số điện thoại, Email.

b :Lớp (Classes):

Mã lớp (PK), Tên lớp, Niên khóa.

c : Khoa (Department):

Mã khoa (PK), Tên khoa.

c :Học phần (Courses):

Mã học phần (PK), Tên học phần, Số tín chỉ, Giảng viên, Thời gian, Phòng học.

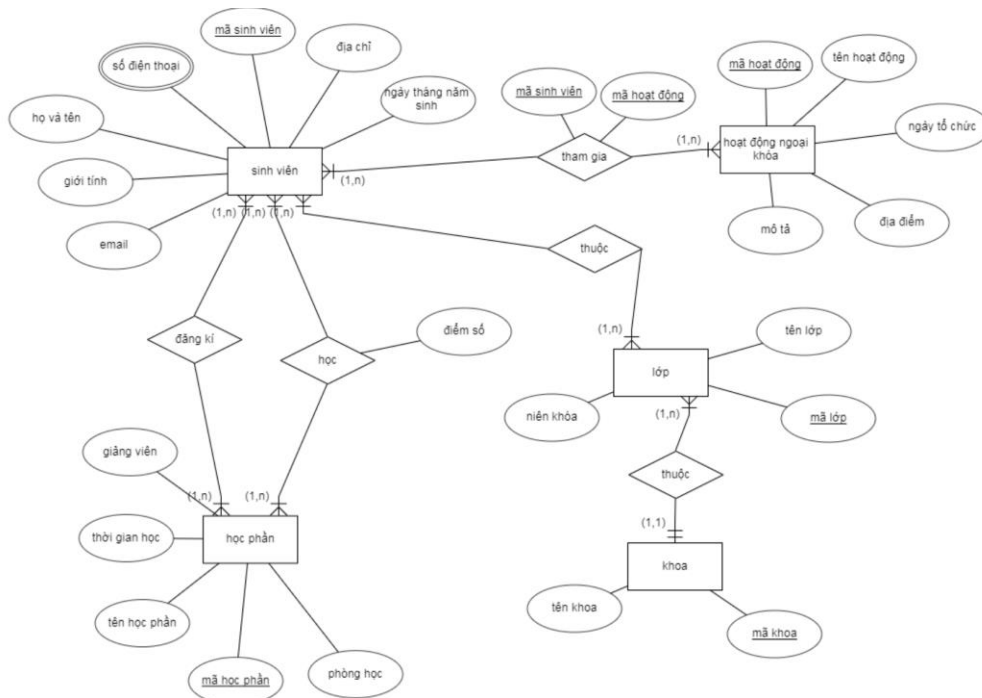
e :Điểm học phần (Course Grades):

Điểm học phần (PK), Mã sinh viên (liên kết với thực thể Sinh viên),
Mã học phần (liên kết với thực thể Học phần),

f :Hoạt động ngoại khóa (Extracurricular Activities):

Mã hoạt động (PK), Tên hoạt động, Mô tả, Thời gian tổ chức, Địa điểm.

2.2 Mối quan hệ qua mô hình ERD



Phần III: Các bước biến đổi từ mô hình ER sang mô hình quan hệ

3.1. Các bước chuyển từ mô hình ER thiết kế sang mô hình quan hệ

3.1.1 Xác định các thực thể và thuộc tính của chúng

- **Thực thể mạnh (Strong Entity):** Tạo một bảng cho mỗi thực thể mạnh. Mỗi thuộc tính của thực thể trở thành một cột trong bảng.
- **Khóa chính (Primary Key):** Chọn một hoặc nhiều thuộc tính làm khóa chính cho bảng.
- **Thực thể yếu (Weak Entity):** Tạo một bảng cho mỗi thực thể yếu. Thực thể yếu phụ thuộc vào thực thể mạnh thông qua một mối quan hệ.
- **Khóa chính:** Kết hợp khóa chính của thực thể mạnh và các thuộc tính phân biệt của thực thể yếu.

3.1.2. Xác định các mối quan hệ (Relationships)

a. Mối quan hệ 1-1 (One-to-One Relationship):

- Tạo bảng cho mỗi thực thể để liên kết hai bảng. Hoặc kết hợp cả hai bảng thành một bảng nếu mối quan hệ là toàn phần (total participation).

b. Mối quan hệ 1-n (One-to-Many Relationship):

- Thêm khóa chính của thực thể “một” vào bảng của thực thể “nhiều” như một khóa ngoại để liên kết các bảng.

c. Mối quan hệ n-n (Many-to-Many Relationship):

- Tạo một bảng trung gian để đại diện cho mối quan hệ. Bảng này chứa khóa chính của cả hai thực thể tham gia mối quan hệ làm khóa ngoại, và thường kết hợp các khóa này để tạo thành khóa chính của bảng trung gian.

3.1.3. Xác định các thuộc tính đa trị (Multivalued Attributes)

- Tạo một bảng riêng biệt cho mỗi thuộc tính đa trị.
- Khóa chính: Kết hợp khóa chính của thực thể ban đầu và thuộc tính đa trị này.

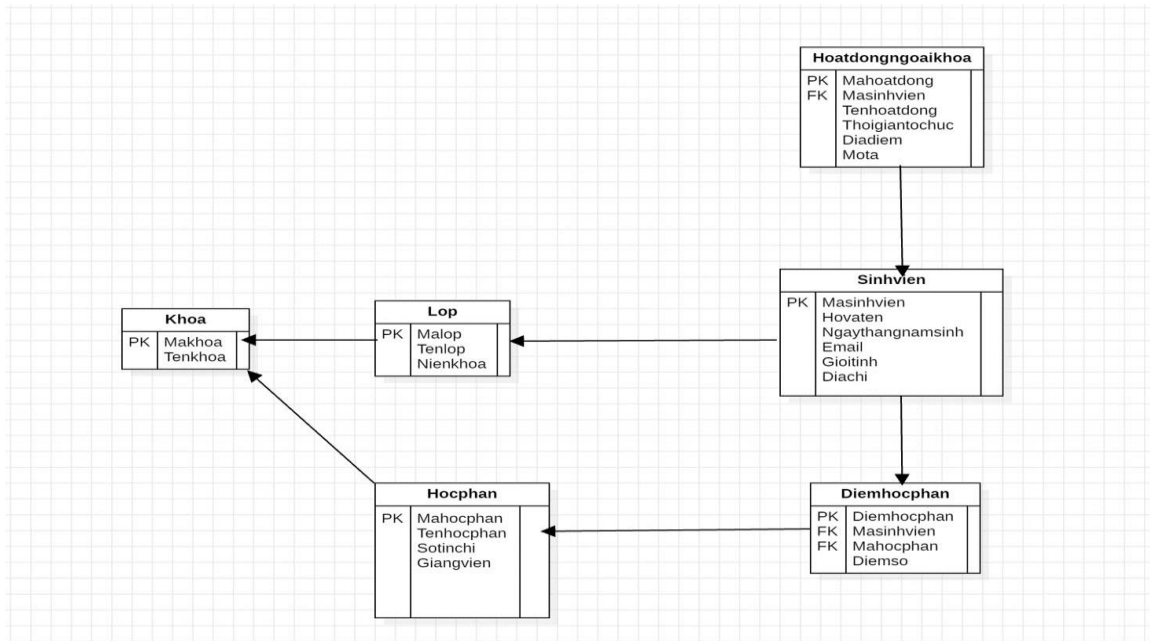
3.1.4. Xử lý các thuộc tính phức tạp (Composite Attributes)

- Phân rã các thuộc tính phức tạp thành các thuộc tính đơn lẻ và xử lý chúng như các thuộc tính thông thường của thực thể.

3.1.5. Xử lý các khóa ngoại (Foreign Keys)

- Đảm bảo rằng tất cả các mối quan hệ đã được xác định đều có các khóa ngoại phù hợp trong các bảng tương ứng.

3.2. Mô hình lược đồ quan hệ:



Phần IV : Thiết kế các bảng dữ liệu vật lý

4.1. Mô tả bảng dữ liệu vật lý

a. Sinh viên (Students):

- Mã sinh viên (PK): MaSV - Kiểu dữ liệu: INT, miền dữ liệu: số nguyên dương, khóa chính, không được để trống.
- Họ và tên: HoTen - Kiểu dữ liệu: VARCHAR(100), không được để trống.
- Ngày sinh: NgaySinh - Kiểu dữ liệu: DATE, không được để trống.
- Giới tính: GioiTinh - Kiểu dữ liệu: CHAR(1), giá trị: 'M' hoặc 'F' (nam hoặc nữ), không được để trống.
- Địa chỉ: DiaChi - Kiểu dữ liệu: VARCHAR(255).
- Số điện thoại: SoDienThoai - Kiểu dữ liệu: VARCHAR(15).
- Email: Email - Kiểu dữ liệu: VARCHAR(100).
- Mã lớp (FK): MaLop - Kiểu dữ liệu: INT, liên kết với MaLop trong bảng Lớp.

b. Lớp (Classes):

-Mã lớp (PK): MaLop - Kiểu dữ liệu: INT, miền dữ liệu: số nguyên dương, khóa chính, không được để trống.

-Tên lớp: TenLop - Kiểu dữ liệu: VARCHAR(100), không được để trống.

-Niên khóa: NienKhoa - Kiểu dữ liệu: VARCHAR(10), ví dụ: '2023-2024'.

Mã khoa (FK): MaKhoa - Kiểu dữ liệu: INT, liên kết với MaKhoa trong bảng Khoa.

c. Khoa (Department):

-Mã khoa (PK): MaKhoa - Kiểu dữ liệu: INT, miền dữ liệu: số nguyên dương, khóa chính, không được để trống.

-Tên khoa: TenKhoa - Kiểu dữ liệu: VARCHAR(100), không được để trống.

d. Học phần (Courses):

-Mã học phần (PK): MaHP - Kiểu dữ liệu: INT, miền dữ liệu: số nguyên dương, khóa chính, không được để trống.

-Tên học phần: TenHP - Kiểu dữ liệu: VARCHAR(100), không được để trống.

-Số tín chỉ: SoTinChi - Kiểu dữ liệu: INT, miền dữ liệu: số nguyên dương.

-Giảng viên: GiangVien - Kiểu dữ liệu: VARCHAR(100).

-Thời gian: ThoiGian - Kiểu dữ liệu: TIME.

-Phòng học: PhongHoc - Kiểu dữ liệu: VARCHAR(10).

e. Điểm học phần (Course Grades):

-Điểm học phần (PK): DiemHP - Kiểu dữ liệu: FLOAT, không được để trống.

-Mã sinh viên (FK): MaSV - Kiểu dữ liệu: INT, liên kết với MaSV trong bảng Sinh viên.

-Mã học phần (FK): MaHP - Kiểu dữ liệu: INT, liên kết với MaHP trong bảng Học phần.

f. Hoạt động ngoại khóa (Extracurricular Activities):

- Mã hoạt động (PK): MaHoatDong - Kiểu dữ liệu: INT, miền dữ liệu: số nguyên dương, khóa chính, không được để trống.
- Tên hoạt động: TenHoatDong - Kiểu dữ liệu: VARCHAR(100), không được để trống.
- Mô tả: MoTa - Kiểu dữ liệu: TEXT.
- Thời gian tổ chức: ThoiGianToChuc - Kiểu dữ liệu: DATE, không được để trống.
- Địa điểm: DiaDiem - Kiểu dữ liệu: VARCHAR(100).

Các mối quan hệ:

- Mối quan hệ giữa các bảng được thiết lập thông qua các trường khóa ngoại (FK) như đã được nêu ở trên.
- Khi thiết lập mối quan hệ, hãy đảm bảo các ràng buộc toàn vẹn dữ liệu (constraints) được thiết lập để đảm bảo tính toàn vẹn của dữ liệu:
- Ràng buộc khóa ngoại (FOREIGN KEY) để đảm bảo dữ liệu trong trường khóa ngoại tồn tại trong bảng mà nó liên kết tới.
- Ràng buộc không cho phép giá trị NULL (NOT NULL) để đảm bảo dữ liệu quan trọng không bị bỏ trống.
- Ràng buộc kiểm soát miền dữ liệu (CHECK) nếu cần để đảm bảo giá trị dữ liệu nằm trong phạm vi cho phép.

4.2. Thiết kế các bảng dữ liệu mức vật lý

a. Tạo cơ sở dữ liệu:

```
CREATE DATABASE QuanLySinhVien;
```

Nếu đã có cơ sở dữ liệu, để sử dụng sử dụng câu lệnh

```
USE QuanLySinhVien;
```

b. Tạo các bảng:

Bảng Sinh Viên (SinhVien):

```
CREATE TABLE SinhVien (
    MaSV INT PRIMARY KEY,
    HoTen VARCHAR(100) NOT NULL,
    NgaySinh DATE NOT NULL,
    GioiTinh CHAR(1) CHECK (GioiTinh IN ('M', 'F')) NOT NULL,
    DiaChi VARCHAR(255),
    SoDienThoai VARCHAR(15),
    Email VARCHAR(100),
    MaLop INT,
    FOREIGN KEY (MaLop) REFERENCES Lop(MaLop)
);
```

Bảng Lớp (Lop):

```
CREATE TABLE Lop (
    MaLop INT PRIMARY KEY,
    TenLop VARCHAR(100) NOT NULL,
    NienKhoa VARCHAR(10) NOT NULL,
    MaKhoa INT,
    FOREIGN KEY (MaKhoa) REFERENCES Khoa(MaKhoa)
);
```

Bảng Khoa (Khoa):

```
CREATE TABLE Khoa (
    MaKhoa INT PRIMARY KEY,
    TenKhoa VARCHAR(100) NOT NULL
);
```

Bảng Học Phần (HocPhan):

```
CREATE TABLE HocPhan (
    MaHP INT PRIMARY KEY,
    TenHP VARCHAR(100) NOT NULL,
    SoTinChi INT NOT NULL,
    GiangVien VARCHAR(100),
    ThoiGian TIME,
    PhongHoc VARCHAR(10)
);
```

Bảng điểm học phần (HocPhan):

```
CREATE TABLE DiemHocPhan (
    MaSV INT,
    MaHP INT,
    DiemHP FLOAT NOT NULL,
    PRIMARY KEY (MaSV, MaHP),
    FOREIGN KEY (MaSV) REFERENCES SinhVien(MaSV),
    FOREIGN KEY (MaHP) REFERENCES HocPhan(MaHP)
);
```

Bảng Hoạt động ngoại khóa (HoatDongNgoaiKhoa) :

```
CREATE TABLE HoatDongNgoaiKhoa (
    MaHoatDong INT PRIMARY KEY,
    TenHoatDong VARCHAR(100) NOT NULL,
    MoTa TEXT,
    ThoiGianToChuc DATE NOT NULL,
    DiaDiem VARCHAR(100)
);
```

Bảng sinh viên tham gia hoạt động (SinhVien_HoatDongNgoaiKhoa):

```
CREATE TABLE SinhVien_HoatDongNgoaiKhoa (
    MaSV INT,
    MaHoatDong INT,
    CONSTRAINT FK_SinhVien FOREIGN KEY (MaSV) REFERENCES SinhVien(MaSV),
    CONSTRAINT FK_HoatDongNgoaiKhoa FOREIGN KEY (MaHoatDong) REFERENCES HoatDongNgoaiKhoa(MaHoatDong),
    PRIMARY KEY (MaSV, MaHoatDong)
);
```

c. Nhập các dữ liệu mẫu

```

INSERT INTO SinhVien (MaSV, HoTen, NgaySinh, GioiTinh, DiaChi, SoDienThoai, Email, MaLop)
VALUES
  (1, 'Nguyen Van A', '2000-01-15', 'M', '123 Main St', '0912345678', 'a@example.com', 101),
  (2, 'Tran Thi B', '2001-02-25', 'F', '456 Elm St', '0923456789', 'b@example.com', 101),
  (3, 'Le Van C', '2000-05-10', 'M', '789 Oak St', '0934567890', 'c@example.com', 102),
  ....
  (15, 'Pham Thi P', '2000-09-30', 'F', '1500 Maple St', '0967890123', 'p@example.com', 105);

```

```

INSERT INTO Lop (MaLop, TenLop, NienKhoa, MaKhoa)
VALUES
  (101, 'Lop 1', '2023-2024', 1),
  (102, 'Lop 2', '2023-2024', 2),
  ....
  (115, 'Lop 15', '2023-2024', 3);

```

```

INSERT INTO Khoa (MaKhoa, TenKhoa)
VALUES
  (1, 'Khoa Toan'),
  (2, 'Khoa Ly'),
  ....
  (15, 'Khoa Tin Hoc');

```

```

INSERT INTO HocPhan (MaHP, TenHP, SoTinChi, GiangVien, ThoiGian, PhongHoc)
VALUES
  (101, 'Toan A1', 3, 'Nguyen Giang', '08:00:00', 'P1'),
  (102, 'Ly A2', 4, 'Le Giang', '09:30:00', 'P2'),
  ...
  (115, 'Tin Hoc C1', 3, 'Tran Giang', '13:00:00', 'P15');

```

```

INSERT INTO DiemHocPhan (MaSV, MaHP, DiemHP)
VALUES
  (1, 101, 8.5),
  (2, 102, 7.0),
  ...
  (15, 115, 9.0);

```

```

INSERT INTO HoatDongNgoaiKhoa (MaHoatDong, TenHoatDong, MoTa, ThoiGianToChuc, DiaDiem)
VALUES
  (1, 'Bong Da', 'Giai bong da nam', '2024-05-10', 'San Bong A'),
  (2, 'Am Nhac', 'Cuoc thi hat', '2024-05-15', 'Hoi Truong B'),
  ...
  (15, 'Tennis', 'Giai tennis nam', '2024-06-05', 'San Tennis D');

```

```

USE QUANLYSINHVIEN

```

```

INSERT INTO SinhVien_HoatDongNgoaiKhoa (MaSV, MaHoatDong)
VALUES
  (1, 101),
  (2, 103),
  ...
  (15, 101));

```

Phần V: Truy vấn

a. INSERT

```

-- Thêm sinh viên vào bảng SinhVien
INSERT INTO SinhVien (MaSV, HoTen, NgaySinh, GioiTinh, DiaChi, SoDienThoai, Email, MaLop)
VALUES ('20', 'Nguyen Van Quan', '2000-01-01', 'Nam', 'Ha Noi', '0123456789', 'nguyenvana@gmail.com', '101');

-- Thêm môn học vào bảng HocPhan
INSERT INTO HocPhan (MaHocPhan, TenHocPhan, SoTinChi, GiangVien, ThoiGian, PhongHoc)
VALUES (444, 'Toan Cao Cap', 3, 'GiangLe', '15:30:00', 'P401');

-- Thêm điểm học phần vào bảng DiemHocPhan
INSERT INTO DiemHocPhan (MaSV, MaHocPhan, DiemHP)
VALUES ('16', '222', 8.5);

-- Thêm lớp vào bảng Lop
INSERT INTO Lop (MaLop, TenLop, NienKhoa, MaKhoa)
VALUES ('102', 'CNTT1', '2020-2023', '4');

-- Thêm khoa vào bảng Khoa
INSERT INTO Khoa (MaKhoa, TenKhoa)
VALUES ('4', 'Cong Nghe Thong Tin');

```

b. DELETE

```

-- Xóa sinh viên khỏi bảng SinhVien
DELETE FROM SinhVien
WHERE MaSV = '12';

-- Xóa môn học khỏi bảng HocPhan
DELETE FROM HocPhan
WHERE MaHocPhan = '111';

-- Xóa điểm học phần khỏi bảng DiemHocPhan
DELETE FROM DiemHocPhan
WHERE MaSV = '10' AND MaHocPhan = '201';

-- Xóa lớp khỏi bảng Lop
DELETE FROM Lop
WHERE MaLop = '101';

-- Xóa khoa khỏi bảng Khoa
DELETE FROM Khoa
WHERE MaKhoa = '2';

```

c. UPDATE

```

-- Cập nhật thông tin sinh viên
UPDATE SinhVien
SET HoTen = 'Nguyen Van B', DiaChi = 'Hai Phong'
WHERE MaSV = 1;

-- Cập nhật thông tin môn học
UPDATE HocPhan
SET SoTinChi = 4
WHERE MaHocPhan = 222;

-- Cập nhật điểm học phần
UPDATE DiemHocPhan
SET DiemHP = 9.0
WHERE MaSV = 1 AND MaHocPhan = 111;

-- Cập nhật thông tin lớp
UPDATE Lop
SET TenLop = 'CNTT2'
WHERE MaLop = 103;

-- Cập nhật thông tin khoa
UPDATE Khoa
SET TenKhoa = 'Khoa Hoc May Tinh'
WHERE MaKhoa = 1;

```

d. SELECT

- Lấy dữ liệu từ một bảng :
 - Lấy thông tin tất cả sinh viên


```
SELECT * FROM SinhVien;
```
 - Lấy thông tin tất cả môn học


```
SELECT * FROM HocPhan;
```
 - Lấy thông tin tất cả điểm học phần


```
SELECT * FROM DiemHocPhan;
```
 - Lấy thông tin tất cả lớp


```
SELECT * FROM Lop;
```
 - Lấy thông tin tất cả khoa


```
SELECT * FROM Khoa;
```


- Lấy dữ liệu từ 2 bảng:

```
-- Lấy thông tin sinh viên và lớp của họ
= SELECT sv.MaSV, sv.HoTen, lop.TenLop
  FROM SinhVien sv
  JOIN Lop lop ON sv.MaLop = lop.MaLop;

-- Lấy thông tin điểm học phần và tên môn học
= SELECT dhp.MaSV, dhp.DiemHP, hp.TenHocPhan
  FROM DiemHocPhan dhp
  JOIN HocPhan hp ON dhp.MaHocPhan = hp.MaHocPhan;

-- Lấy thông tin sinh viên và khoa của họ
= SELECT sv.MaSV, sv.HoTen, k.TenKhoa
  FROM SinhVien sv
  JOIN Lop lop ON sv.MaLop = lop.MaLop
  JOIN Khoa k ON lop.MaKhoa = k.MaKhoa;

-- Lấy thông tin lớp và khoa của họ
= SELECT lop.MaLop, lop.TenLop, k.TenKhoa
  FROM Lop lop
  JOIN Khoa k ON lop.MaKhoa = k.MaKhoa;
```

- Lấy dữ liệu từ 3 bảng có liên quan và sử dụng GROUP BY, HAVING

```

JOIN Lop lop ON sv.MaLop = lop.MaLop
JOIN DiemHocPhan dhp ON sv.MaSV = dhp.MaSV
GROUP BY lop.TenLop, sv.MaSV
HAVING AVG(dhp.DiemHP) > 7.0;

-- Lấy tổng số tín chỉ sinh viên đã đăng ký theo từng khoa
SELECT k.TenKhoa, sv.MaSV, SUM(hp.SoTinChi) AS TongSoTinChi
FROM SinhVien sv
JOIN Lop lop ON sv.MaLop = lop.MaLop
JOIN Khoa k ON lop.MaKhoa = k.MaKhoa
JOIN DiemHocPhan dhp ON sv.MaSV = dhp.MaSV
JOIN HocPhan hp ON dhp.MaHocPhan = hp.MaHocPhan
GROUP BY k.TenKhoa, sv.MaSV;

-- Lấy danh sách sinh viên và số môn học đã đăng ký
SELECT sv.MaSV, sv.HoTen, COUNT(dhp.MaHocPhan) AS SoMonHoc
FROM SinhVien sv
JOIN DiemHocPhan dhp ON sv.MaSV = dhp.MaSV
GROUP BY sv.MaSV, sv.HoTen;

-- Lấy danh sách sinh viên, lớp và khoa của họ
SELECT sv.MaSV, sv.HoTen, lop.TenLop, k.TenKhoa
FROM SinhVien sv
JOIN Lop lop ON sv.MaLop = lop.MaLop
JOIN Khoa k ON lop.MaKhoa = k.MaKhoa
GROUP BY sv.MaSV, sv.HoTen, lop.TenLop, k.TenKhoa;

-- Lấy điểm trung bình của sinh viên theo từng khoa
SELECT k.TenKhoa, sv.MaSV, AVG(dhp.DiemHP) AS DiemTrungBinh
FROM SinhVien sv
JOIN Lop lop ON sv.MaLop = lop.MaLop
JOIN Khoa k ON lop.MaKhoa = k.MaKhoa
JOIN DiemHocPhan dhp ON sv.MaSV = dhp.MaSV
GROUP BY k.TenKhoa, sv.MaSV;

```

Phần VI : Xây dựng giao diện GUI

- Tạo một giao diện đơn giản để người dùng có thể thao tác với các tác vụ như xem thông tin của sinh viên, học phần,... thêm, sửa đổi và xóa đi các thông tin của cơ sở dữ liệu một cách thuận tiện hơn bằng python.
- Sử dụng :

- Pyodbc để kết nối từ python với database.

```
server='DESKTOP-HRTE7HR\\SQLEXPRESS'
database='QUANLYSINHVIEN'
username=''
password=''

Connection_String=f'DRIVER={{SQL SERVER}};SERVER={server};DATABASE={database};UID={username};password={password}'
conn = pyodbc.connect(Connection_String)
```

- Tkinter để xây dựng giao diện

Quản lý Sinh viên

Mã sinh viên:	<input type="text"/>	Bảng thông tin sinh viên
Họ tên:	<input type="text"/>	Bảng thông tin lớp
Ngày sinh:	<input type="text"/>	Bảng thông tin khoa
Giới tính:	<input type="text"/>	Bảng thông tin học phần
Địa chỉ:	<input type="text"/>	Bảng thông tin điểm học phần
Số điện thoại	<input type="text"/>	Bảng thông tin hoạt động ngoại khóa
Email:	<input type="text"/>	Thêm sinh viên
Mã lớp	<input type="text"/>	Cập nhật thông tin sinh viên
Tên lớp	<input type="text"/>	Xóa sinh viên
Niên khóa	<input type="text"/>	Thêm lớp
Mã khoa	<input type="text"/>	Cập nhật thông tin lớp
Tên Khoa	<input type="text"/>	Xóa lớp
Mã học phần:	<input type="text"/>	Thêm khoa
Tên học phần	<input type="text"/>	Cập nhật thông tin khoa
Số tín chỉ :	<input type="text"/>	Xóa khoa
Giảng viên	<input type="text"/>	Thêm Học phần
Thời gian	<input type="text"/>	Cập nhật Học phần
Phòng học	<input type="text"/>	Xóa Học phần
Điểm học phần	<input type="text"/>	Thêm điểm học phần
Mã hoạt động	<input type="text"/>	Cập nhật điểm học phần
Tên hoạt động	<input type="text"/>	Xóa điểm học phần
Mô tả	<input type="text"/>	Thêm hoạt động
Thời gian tổ chức	<input type="text"/>	Cập nhật hoạt động
Địa điểm	<input type="text"/>	Xóa hoạt động
Thêm sinh viên tham gia hoạt động		Bảng thông tin sinh viên tham gia hoạt động
		Xóa sinh viên tham gia hoạt động

- Giao diện cung cấp các chức năng:
 - Hiện ra các thông tin của các bảng của database như thông tin sinh viên, lớp, khoa, học phần, điểm học phần, hoạt động ngoại khóa.
 - Thao tác thêm, cập nhật, xóa các thông tin như sinh viên, lớp khoa, học phần,

Phần VII: Kết luận

7.1 Đánh giá kết quả.

a. Kết quả đạt được.

- Hệ thống quản lý sinh viên có thể vận hành với mô hình nhỏ.
- Tạo cơ sở dữ liệu quản lý sinh viên.
- Xây dựng được thành công các chức năng cơ bản của hệ thống, chức năng cập nhập thông tin , chức năng thêm xóa thông tin , sửa đổi thông tin đăng nhập.

b. Kết quả chưa đạt được.

- Hệ thống chưa được phát triển tối ưu nên có thể chưa cung cấp đủ các chức năng mà sinh viên và nhân viên quản lý mong đợi, chưa thực hiện được các chức năng phức tạp.

c. Kết luận.

- Bài báo cáo về “*Quản lý thông tin cá nhân của sinh viên*” được xây dựng với ngôn ngữ Python kết hợp với Hệ quản trị cơ sở dữ liệu SQL.
- Tuy nhiên do chưa có kinh nghiệm chưa sâu nên đề tài của chúng em khó tránh khỏi những thiếu sót.
- Chúng em mong nhận được ý kiến đóng góp của các thầy cô và các bạn để đề tài của chúng em hoàn thiện hơn và có thể đi vào thực tế.

LỜI CẢM ƠN

Nhóm chúng em xin chân thành cảm ơn *Thầy Phan Lý Huỳnh* đã tận tình hướng dẫn và chỉ bảo chúng em trong suốt quá trình học để chúng em làm bài tập lớn này.

Chúng em xin chân thành cảm ơn !