

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÁO CÁO CUỐI KÌ
MÔN CƠ SỞ DỮ LIỆU
Chủ đề “Hệ thống quản lý giáo dục”

Giảng viên : Đỗ Văn Hanh

Nhóm bài tập : 08

Lớp : D21PTUD

Sinh viên thực hiện : Nguyễn Thị Quỳnh - B21DCPT196

Trịnh Kiều Trang - B21DCPT037

Trần Quang Thắng - B21DCPT204

Hà Nội, năm 2024

LỜI CẢM ƠN

Lời đầu tiên, trước khi bắt đầu đề tài, nhóm chúng em xin được bày tỏ lòng kính trọng và lòng biết ơn tới Học viện Công nghệ Bưu chính Viễn thông đã tạo điều kiện cho chúng em học tập, tiếp cận môn học này.

Đặc biệt, chúng em muốn gửi lời cảm ơn sâu sắc đến Giảng viên Đỗ Văn Hanh, người đã giảng dạy, tận tình hướng dẫn, động viên và hỗ trợ chúng em trong suốt quá trình thực hiện nghiên cứu này. Sự tận tâm, nhiệt huyết và những kiến thức quý báu của thầy đã giúp nhóm chúng em định hướng, giải quyết những khó khăn, vướng mắc trong quá trình nghiên cứu.

Bộ môn Cơ sở dữ liệu là môn học thú vị, vô cùng bổ ích và có tính thực tế cao. Đảm bảo cung cấp đủ kiến thức, gắn liền với nhu cầu thực tiễn của sinh viên. Nhờ có sự chỉ dẫn tận tình của thầy, chúng em đã có được những hiểu biết sâu sắc hơn về bộ môn này và mở ra cơ hội để phát triển, khám phá khả năng nghiên cứu của bản thân.

Chúng em đã rất cố gắng hoàn thiện báo cáo nghiên cứu này một cách hoàn chỉnh nhất nhưng chắc chắn không thể tránh khỏi những thiếu sót và hạn chế. Nhóm chúng em rất mong nhận được sự góp ý, nhận xét từ phía thầy để những nghiên cứu sau này đạt được hiệu quả tốt hơn.

Chúng em xin chân thành cảm ơn.

Trân trọng,

Nhóm 8

MỤC LỤC

1. Mô tả yêu cầu.....	4
2. Phân tích yêu cầu nghiệp vụ.....	4
a. Thông tin.....	4
b. Yêu cầu chức năng.....	4
• Quản lý thông tin phòng học.....	4
• Quản lý thông tin phòng ban.....	4
• Quản lý khóa học.....	5
• Quản lý giảng viên.....	5
• Quản lý lớp học.....	5
• Quản lý giảng dạy.....	5
• Quản lý sinh viên.....	5
• Quản lý tư vấn học tập.....	5
• Quản lý khung giờ học.....	5
c. Yêu cầu phi chức năng.....	6
3. Quy trình hoạt động.....	6
a. Quy trình kinh doanh.....	6
b. Quy trình của hệ thống.....	6
4. Ràng buộc quan hệ.....	6
5. Lược đồ thực thể liên kết.....	7
a. Phân tích lược đồ thực thể liên kết.....	7
b. Lược đồ thực thể liên kết.....	9
6. Lược đồ dữ liệu quan hệ.....	9
a. Phân tích lược đồ dữ liệu quan hệ.....	9
b. Thiết kế chi tiết các bảng.....	14
c. Lược đồ dữ liệu quan hệ.....	18
7. Tạo bảng và các ràng buộc.....	19
a. Tạo bảng department.....	19
b. Tạo bảng instructor.....	19
c. Tạo bảng student.....	19
d. Tạo bảng course.....	19
e. Tạo bảng classroom.....	20
f. Tạo bảng time_slot.....	20
g. Tạo bảng class.....	20
h. Tạo bảng teaches.....	21

i. Tạo bảng takes.....	21
j. Tạo bảng advisor.....	21
8. Thêm dữ liệu vào bảng.....	22
9. Thực hiện một số truy vấn.....	22
a. Truy vấn đơn giản - GitHub.....	22
b. Truy vấn bằng stored procedure - GitHub.....	27
10. Tạo transaction thực hiện nghiệp vụ thực tế.....	33
a. Xóa sinh viên ra khỏi CSDL.....	33
b. Cập nhật tổng số tín cho học viên.....	34
c. Xóa giảng viên ra khỏi CSDL.....	35

1. Mô tả yêu cầu

Mục đích: Hệ cơ sở dữ liệu phục vụ công tác quản lý một hệ thống giáo dục của Học viện Công nghệ Bưu chính Viễn thông bao gồm thêm sinh viên, thêm học phần vào chương trình đào tạo, xóa sinh viên ra khỏi CSDL, cập nhật điểm học phần chi tiết cho 1 sinh viên,...

2. Phân tích yêu cầu nghiệp vụ

a. Thông tin

Học viện Công nghệ Bưu chính Viễn thông:

- Trường gồm 3 tòa nhà chính phục vụ cho công tác giảng dạy và học: Tòa A1, Tòa A2, Tòa A3. Trong đó tòa A1 là nơi văn phòng làm việc của các bộ phận: phòng giáo vụ, phòng CT&CTSV, phòng Đảng ủy, phòng ban của khoa Đa phương tiện, phòng học, viện trung tâm nghiên cứu CDiT,... Tòa A2 là nơi giảng dạy dạy chính bao gồm giảng đường và các phòng học và là phòng ban của khoa Công nghệ thông tin, khoa Cơ bản, khoa Viễn Thông, khoa kỹ thuật điện tử, khoa Quản trị kinh doanh, khoa An toàn thông tin, khoa Tài chính kế toán,.. Tòa A3 cũng là nơi giảng dạy bao gồm các phòng học và trung tâm thí nghiệm thực hành.
- Gồm 8 khoa: Khoa cơ bản, Khoa Công nghệ thông tin, Khoa Viễn thông, Khoa Kỹ thuật điện tử, Khoa Đa phương tiện, Khoa Quản trị kinh doanh, Khoa An toàn thông tin, Khoa Tài chính kế toán.
- Mỗi khoa thì có các sinh viên các khóa và có các thầy cô giảng viên
- Các sinh viên học tập theo quá trình tích lũy tín chỉ. Mỗi kỳ thì có các môn học với số tín chỉ khác nhau. Một kỳ tối đa 25 tín chỉ và tối thiểu là 16 tín chỉ thì mới được đăng ký học phần
- Gồm 4 kỳ học: Xuân, Hè, Thu, Đông.

b. Yêu cầu chức năng

- Quản lý thông tin phòng học
 - Lưu trữ thông tin về các phòng học bao gồm: thông tin tòa nhà, số phòng, và sức chứa của phòng học.
 - Quản lý không gian học tập, sắp xếp và tối ưu hóa việc sử dụng các phòng học dựa trên sức chứa và vị trí.
- Quản lý thông tin phòng ban
 - Lưu trữ thông tin về các phòng ban hoặc khoa, bao gồm các thông tin: tên phòng ban, tòa nhà và ngân sách.

- Quản lý tài chính và địa điểm của các phòng ban, đảm bảo mỗi khoa có nguồn lực cần thiết để hoạt động.
- Quản lý khóa học
 - Lưu trữ thông tin về các khóa học như mã khóa học, tiêu đề, phòng ban và số tín chỉ.
 - Theo dõi các khóa học được cung cấp bởi các phòng ban khác nhau và quản lý chương trình học của học viện.
- Quản lý giảng viên
 - Lưu trữ thông tin về giảng viên như mã giảng viên, tên, phòng ban và lương.
 - Quản lý thông tin cá nhân và tài chính của giảng viên, đảm bảo rằng mỗi giảng viên thuộc về một phòng ban cụ thể.
- Quản lý lớp học
 - Lưu trữ thông tin về các lớp học như mã khóa học, mã lớp, học kỳ, năm học, tòa nhà, số phòng và mã khung giờ.
 - Quản lý thời gian và địa điểm của các lớp học, đảm bảo rằng các lớp học được tổ chức tại các phòng học phù hợp và trong các khung giờ nhất định.
- Quản lý giảng dạy
 - Lưu trữ thông tin về việc giảng viên dạy các lớp học cụ thể.
 - Quản lý sự phân công giảng dạy của giảng viên, đảm bảo rằng mỗi lớp học có giảng viên phụ trách.
- Quản lý sinh viên
 - Lưu trữ thông tin về sinh viên như mã sinh viên, tên, phòng ban và tổng số tín chỉ đã tích lũy.
 - Quản lý hồ sơ học tập và phòng ban của sinh viên, theo dõi tiến độ học tập của từng sinh viên.
- Quản lý tư vấn học tập
 - Lưu trữ thông tin về mối quan hệ giữa sinh viên và giảng viên tư vấn.
 - Quản lý sự hỗ trợ học tập cho sinh viên, đảm bảo rằng mỗi sinh viên có một giảng viên tư vấn để hỗ trợ trong quá trình học tập.
- Quản lý khung giờ học
 - Lưu trữ thông tin về các khung giờ học bao gồm mã khung giờ, ngày trong tuần, giờ bắt đầu và giờ kết thúc.
 - Quản lý thời gian biểu của các lớp học, đảm bảo rằng các lớp học không bị trùng lịch và được tổ chức trong các khung giờ hợp lý.

c. Yêu cầu phi chức năng

- Bảo mật dữ liệu: Quản lý quyền truy cập vào các dữ liệu quan trọng: thông tin cá nhân sinh viên, thông tin lương giảng viên,...
- Hiệu suất và độ tin cậy: Đảm bảo hệ thống hoạt động một cách ổn định và nhanh chóng, đáp ứng các yêu cầu xử lý dữ liệu trong thời gian ngắn.
- Tích hợp với các hệ thống khác: Có khả năng tích hợp dữ liệu và các quy trình với các hệ thống khác như hệ thống quản lý tài chính, hệ thống quản lý nhân sự, ...

3. Quy trình hoạt động

a. Quy trình kinh doanh

- Đăng ký môn học: Sinh viên đăng ký môn học thông qua hệ thống, hệ thống phân lớp và phân công giảng viên.
- Giảng dạy và học tập: Giảng viên phân công giảng dạy, sinh viên tham gia lớp học, thực hiện các hoạt động học tập và kiểm tra.
- Quản lý tài chính: Quản lý ngân sách của từng khoa/phòng ban, chi tiêu cho giảng viên và các hoạt động học tập.

b. Quy trình của hệ thống

- Quản lý dữ liệu: Bao gồm thêm dữ liệu, xử lý dữ liệu, lưu trữ và truy xuất dữ liệu.
- Xử lý nghiệp vụ: Xử lý các yêu cầu đăng ký môn học, quản lý lớp học, quản lý giảng viên, quản lý sinh viên.

4. Ràng buộc quan hệ

- 1 khoa (department) có nhiều sinh viên (student)
- 1 sinh viên (student) thuộc 1 khoa (department)
- 1 khoa (department) có nhiều giảng viên (instructor)
- 1 giảng viên (instructor) thuộc 1 khoa (department)
- 1 khoa (department) có nhiều khóa học (course)
- 1 phòng học (classroom) có nhiều lớp học (class)
- 1 lớp học (class) học ở 1 phòng (classroom) duy nhất tại một thời điểm
- 1 khung thời gian (time_slot) có thể có nhiều lớp học (class)
- 1 khóa học (course) có nhiều lớp học (class)
- 1 sinh viên (student) có một giảng viên cố vấn (instructor)
- 1 giảng viên (instructor) cố vấn nhiều sinh viên (student)
- 1 lớp (class) có nhiều giảng viên (instrutor)
- 1 sinh viên (student) có thể học nhiều lớp học (class)

- 1 lớp học (class) có thể có nhiều sinh viên đăng ký
- 1 giảng viên (instructor) có thể dạy nhiều lớp
- 1 lớp học (class) chỉ có thể do 1 giảng viên (instructor) dạy

5. Lược đồ thực thể liên kết

a. Phân tích lược đồ thực thể liên kết

- Danh sách các thực thể

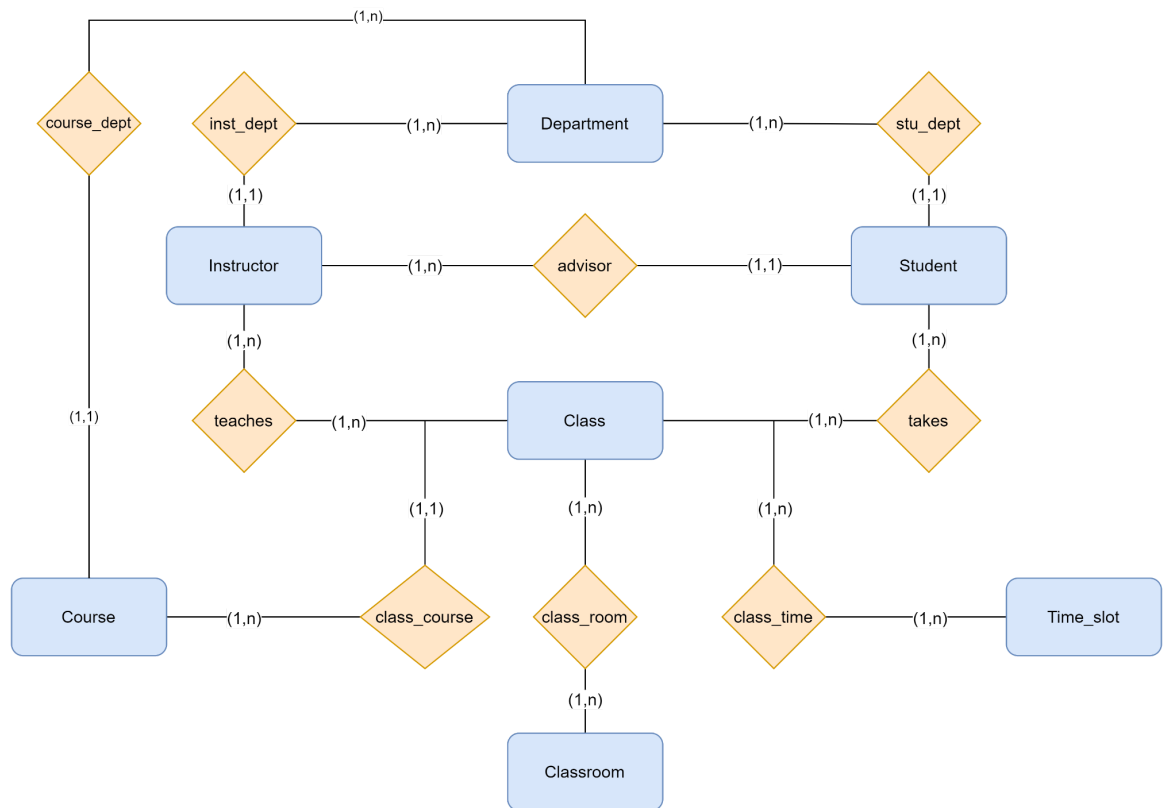
- Department: Đại diện cho các khoa trong trường đại học, chứa thông tin về tên khoa, building và ngân sách.
- Instructor: Đại diện cho giảng viên, bao gồm thông tin như mã giảng viên, tên, và tên khoa mà họ thuộc về.
- Student: Đại diện cho sinh viên, bao gồm thông tin như mã sinh viên, tên, và tên khoa mà họ thuộc về.
- Course: Đại diện cho các khóa học, bao gồm thông tin như mã khóa học, tiêu đề, tên khoa và số tín chỉ.
- Classroom: Đại diện cho các phòng học, bao gồm thông tin như building, room number và sức chứa.
- Time Slot: Đại diện cho các khung thời gian, bao gồm thông tin về ngày, giờ bắt đầu và kết thúc.
- Class: Đại diện cho các lớp học, bao gồm thông tin như mã khóa học, mã lớp học, học kỳ, năm học, building, room number và time slot.

- Danh sách các mối quan hệ

- Khoa (Department) - Giảng viên (Instructor)
 - Quan hệ: Một-Nhiều (One-to-Many)
 - Mô tả: Mỗi khoa có nhiều giảng viên, nhưng mỗi giảng viên chỉ thuộc một khoa.
- Khoa (Department) - Sinh viên (Student)
 - Quan hệ: Một-Nhiều (One-to-Many)
 - Mô tả: Mỗi khoa có nhiều sinh viên, nhưng mỗi sinh viên chỉ thuộc một khoa.
- Khoa (Department) - Khóa học (Course)
 - Quan hệ: Một-Nhiều (One-to-Many)
 - Mô tả: Mỗi khoa có nhiều khóa học, nhưng mỗi khóa học chỉ thuộc một khoa.
- Khóa học (Course) - Lớp học (Class)
 - Quan hệ: Một-Nhiều (One-to-Many)
 - Mô tả: Một khóa học có thể có nhiều lớp học, nhưng mỗi lớp học chỉ thuộc một khóa học.
- Lớp học (Class) - Phòng học (Classroom)
 - Quan hệ: Một-Nhiều (One-to-Many)
 - Mô tả: Một lớp học chỉ có thể diễn ra ở một phòng học.

- Quan hệ: Một-Nhiều (One-to-Many)
- Mô tả: Một lớp học diễn ra tại một phòng học, nhưng một phòng học có thể phục vụ nhiều lớp học.
- Lớp học (Class) - Khung thời gian (Time Slot)
 - Quan hệ: Một-Nhiều (One-to-Many)
 - Mô tả: Một lớp học có một khung thời gian cụ thể, nhưng một khung thời gian có thể phục vụ nhiều lớp học.
- Giảng viên (Instructor) - Lớp học (Class)
 - Quan hệ: Nhiều-Nhiều (Many-to-Many)
 - Mô tả: Một giảng viên có thể giảng dạy nhiều lớp học, và một lớp học có thể có nhiều giảng viên.
 - Thực thể kết nối: Teaches (Giảng dạy)
- Sinh viên (Student) - Lớp học (Class)
 - Quan hệ: Nhiều-Nhiều (Many-to-Many)
 - Mô tả: Một sinh viên có thể theo học nhiều lớp học, và một lớp học có thể có nhiều sinh viên.
 - Thực thể kết nối: Takes (Theo học)
- Sinh viên (Student) - Giảng viên (Instructor)
 - Quan hệ: Một-Nhiều (One-to-Many)
 - Mô tả: Một sinh viên có thể có một giảng viên cố vấn, nhưng một giảng viên có thể cố vấn nhiều sinh viên.
 - Thực thể kết nối: Advisor (Cố vấn)

b. Lược đồ thực thể liên kết



6. Lược đồ dữ liệu quan hệ

a. Phân tích lược đồ dữ liệu quan hệ

- Department:
 - + Nội dung: Khoa/Viện đào tạo
 - + Thuộc tính:
 - dept_name: Tên khoa (VARCHAR(50)).
 - building: Tên tòa nhà mà khoa đặt trụ sở (VARCHAR(5)).
 - budget: Ngân sách của khoa (NUMERIC(12,2)) với ràng buộc kiểm tra giá trị phải lớn hơn 0.
 - + Khóa chính: primary key(dept_name) đảm bảo mỗi khoa là duy nhất.
- Instructor:
 - + Nội dung: Thông tin giảng viên
 - + Thuộc tính:
 - id: Mã giảng viên (VARCHAR(6)).
 - name: Tên giảng viên (VARCHAR(30)) và không được để trống.
 - dept_name: Tên khoa của giảng viên (VARCHAR(50)).

- salary: Lương của giảng viên (NUMERIC(8,2)) với ràng buộc kiểm tra giá trị phải lớn hơn 0.
 - + Khóa chính: primary key (id) đảm bảo mỗi giảng viên là duy nhất.
 - + Khóa ngoại: FOREIGN KEY(dept_name) liên kết với dept_name của bảng department. Khi một khoa bị xóa, giá trị dept_name trong bảng instructor sẽ được đặt thành NULL.
- Student:
 - + Nội dung: Liên kết sinh viên với các lớp học mà họ tham gia
 - + Thuộc tính:
 - ID: Mã sinh viên (kiểu dữ liệu VARCHAR(11)).
 - name: Tên sinh viên (kiểu dữ liệu VARCHAR(30)) và không được để trống.
 - dept_name: Tên khoa của sinh viên (kiểu dữ liệu VARCHAR(50)).
 - tot_cred: Tổng số tín chỉ của sinh viên (kiểu dữ liệu NUMERIC(3,0)) với ràng buộc kiểm tra giá trị phải lớn hơn hoặc bằng 0.
 - + Khóa chính: PRIMARY KEY (ID), đảm bảo mỗi sinh viên là duy nhất.
 - + Khóa ngoại: FOREIGN KEY (dept_name), liên kết với dept_name của bảng department. Khi một khoa bị xóa, giá trị dept_name trong bảng student sẽ được đặt thành NULL.
- Class
 - + Nội dung: Các lớp khóa học
 - + Thuộc tính
 - course_id: Mã khóa học (kiểu dữ liệu VARCHAR(8)).
 - class_id: Mã lớp học (kiểu dữ liệu VARCHAR(8)).
 - semester: Học kỳ (kiểu dữ liệu VARCHAR(6)) với ràng buộc kiểm tra giá trị phải là một trong các giá trị: 'Fall', 'Winter', 'Spring', 'Summer'.
 - year: Năm (kiểu dữ liệu NUMERIC(4,0)) với ràng buộc kiểm tra giá trị phải nằm trong khoảng từ 1952 đến 2100.
 - building: Tên tòa nhà nơi phân học được tổ chức (kiểu dữ liệu VARCHAR(5)).
 - room_number: Số phòng nơi phân học được tổ chức (kiểu dữ liệu VARCHAR(5)).

- time_slot_id: Mã khung thời gian (kiểu dữ liệu VARCHAR(4)).
- + Khóa chính: PRIMARY KEY (course_id, class_id, semester, year), bao gồm bốn cột mã khóa học, mã lớp học, học kì, năm; đảm bảo mỗi phần học là duy nhất.
- + Khóa ngoại:
 - FOREIGN KEY (course_id): Liên kết với course_id của bảng course. Khi một khóa học bị xóa, các lớp học liên quan cũng bị xóa.
 - FOREIGN KEY (building, room_number): Liên kết với building và room_number của bảng classroom. Khi một phòng học bị xóa, giá trị building và room_number trong bảng section sẽ được đặt thành NULL.
- Course
 - + Nội dung: Các khóa học
 - + Thuộc tính:
 - course_id: Mã khóa học (kiểu dữ liệu VARCHAR(8)).
 - title: Tiêu đề khóa học (kiểu dữ liệu VARCHAR(50)).
 - dept_name: Tên khoa cung cấp khóa học (kiểu dữ liệu VARCHAR(50)).
 - credits: Số tín chỉ của khóa học (kiểu dữ liệu NUMERIC(2,0)) với ràng buộc kiểm tra giá trị phải lớn hơn 0.
 - + Khóa chính: PRIMARY KEY (course_id), đảm bảo mỗi khóa học là duy nhất.
- Classroom
 - + Nội dung: Phòng học
 - + Thuộc tính:
 - building: Tên tòa nhà (VARCHAR(5)).
 - room_number: Số phòng (VARCHAR(5)).
 - capacity: Sức chứa của phòng (NUMERIC(3,0)).
 - + Khóa chính: (building, room_number) đảm bảo mỗi phòng học trong mỗi tòa nhà là duy nhất
- Takes
 - + Nội dung: Học tập
 - + Thuộc tính
 - ID: Mã sinh viên (kiểu dữ liệu VARCHAR(11)).

- course_id: Mã khóa học (kiểu dữ liệu VARCHAR(8)).
 - class_id: Mã lớp học (kiểu dữ liệu VARCHAR(8)).
 - semester: Học kỳ (kiểu dữ liệu VARCHAR(6)).
 - year: Năm (kiểu dữ liệu NUMERIC(4,0)).
 - grade: Điểm của sinh viên trong phần học (kiểu dữ liệu NUMERIC(2,1)) .
- + Khóa chính: PRIMARY KEY (ID, course_id, sec_id, semester, year)
- + Khóa ngoại:
 - FOREIGN KEY (course_id, class_id, semester, year): Liên kết với các cột tương ứng của bảng section. Khi một phần học bị xóa, các bản ghi liên quan trong bảng takes cũng bị xóa.
 - FOREIGN KEY (ID): Liên kết với ID của bảng student. Khi một sinh viên bị xóa, các bản ghi liên quan trong bảng takes cũng bị xóa.
- Teaches
 - + Nội dung: Liên kết giảng viên với phần học mà họ giảng dạy.
 - + Thuộc tính:
 - ID: Mã giảng viên (kiểu dữ liệu VARCHAR(5)).
 - course_id: Mã khóa học (kiểu dữ liệu VARCHAR(8)).
 - class_id: Mã phần học (kiểu dữ liệu VARCHAR(8)).
 - semester: Học kỳ (kiểu dữ liệu VARCHAR(6)).
 - year: Năm (kiểu dữ liệu NUMERIC(4,0)).
 - + Khóa chính: PRIMARY KEY (ID, course_id, sec_id, semester, year)
 - + Khóa ngoại:
 - FOREIGN KEY (course_id, class_id, semester, year): Liên kết với các cột tương ứng của bảng section. Khi một phần học bị xóa, các bản ghi liên quan trong bảng teaches cũng bị xóa.
 - FOREIGN KEY (ID): Liên kết với ID của bảng instructor. Khi một giảng viên bị xóa, các bản ghi liên quan trong bảng teaches cũng bị xóa.
- Advisor:
 - + Nội dung: Liên kết sinh viên với giảng viên cố vấn.

- + Thuộc tính:
 - s_ID: đại diện cho mã sinh viên (kiểu dữ liệu VARCHAR(11))
 - i_ID: đại diện cho mã giảng viên (kiểu dữ liệu VARCHAR(5))
- + Khóa chính: PRIMARY KEY (s_ID), đảm bảo mỗi sinh viên chỉ có một cố vấn.
- + Khóa ngoại:
 - FOREIGN KEY (s_ID): Khóa ngoại này tham chiếu đến cột ID trong bảng student. Đảm bảo rằng mọi giá trị trong cột s_ID phải tồn tại trong cột ID của bảng student. Nếu một sinh viên bị xóa khỏi bảng student, bản ghi tương ứng trong bảng advisor cũng sẽ bị xóa theo (vì ràng buộc "on delete cascade")
 - FOREIGN KEY (i_ID): Khóa ngoại này tham chiếu đến cột ID trong bảng instructor. Nếu một giảng viên được tham chiếu bị xóa khỏi bảng instructor, i_ID trong bảng advisor sẽ được đặt thành null, nghĩa là nếu một giảng viên bị xóa, mối quan hệ cố vấn cho sinh viên sẽ được cập nhật để hiển thị rằng hiện tại họ không có cố vấn.
- Time_slot
 - + Nội dung: Thông tin về khung thời gian của các lớp học
 - + Thuộc tính:
 - time_slot_id: Mã khung thời gian (kiểu dữ liệu VARCHAR(4)).
 - day: Ngày trong tuần (kiểu dữ liệu VARCHAR(1)).
 - start_hr: Giờ bắt đầu (kiểu dữ liệu NUMERIC(2)) với ràng buộc kiểm tra giá trị phải từ 0 đến 23.
 - start_min: Phút bắt đầu (kiểu dữ liệu NUMERIC(2)) với ràng buộc kiểm tra giá trị phải từ 0 đến 59.
 - end_hr: Giờ kết thúc (kiểu dữ liệu NUMERIC(2)) với ràng buộc kiểm tra giá trị phải từ 0 đến 23.
 - end_min: Phút kết thúc (kiểu dữ liệu NUMERIC(2)) với ràng buộc kiểm tra giá trị phải từ 0 đến 59.
 - + Khóa chính: PRIMARY KEY (time_slot_id)

b. Thiết kế chi tiết các bảng

Bảng Department

Thuộc tính	Diễn giải	Kiểu	Độ rộng	Định dạng	Ràng buộc
dept_name	Tên Khoa	varchar	50		Khóa chính
building	Tên Tòa Nhà	varchar	5	Ax	
budget	Ngân sách	numeric	12,2		check (budget > 0)

Bảng Instructor

Thuộc tính	Diễn giải	Kiểu	Độ rộng	Định dạng	Ràng buộc
ID	Mã Giảng Viên	varchar	6	GVxxx	Khóa chính
name	Tên Giảng Viên	varchar	30		NOT NULL
dept_name	Tên Khoa	varchar	50		check (salary > 0)
salary	Lương	numeric	8,2		Khóa ngoại

Bảng Student

Thuộc tính	Diễn giải	Kiểu	Độ rộng	Định dạng	Ràng buộc
ID	Mã Sinh Viên	varchar	11	BxxDCxx xxx	Khóa chính
name	Tên Sinh Viên	varchar	30		NOT NULL
dept_name	Tên Khoa	varchar	50		Khóa ngoại
tot_cred	Tín chỉ tích lũy	numeric	3		check (tot_cred >= 0)

Bảng Class

Thuộc tính	Diễn giải	Kiểu	Độ rộng	Định dạng	Ràng buộc
course_id	Mã Khóa Học	varchar	8		Khóa ngoại
class_id	Mã Lớp Học	varchar	8	CLxxx	Khóa chính
semester	Học Kỳ	vachar	6		check (semester in ('Fall', 'Winter', 'Spring', 'Summer'))
year	Năm Học	numeric	4		check (year > 1952 and year < 2100)
building	Tên Tòa Nhà	varchar	5	Ax	Khóa ngoại
room number	Số Phòng	varchar	5		Khóa ngoại
time_slot_ib	Mã khung thời gian	varchar	4	Kx_x	Khóa ngoại

Bảng Course

Thuộc tính	Diễn giải	Kiểu	Độ rộng	Định dạng	Ràng buộc
course_id	Mã Khóa Học	varchar	8		Khóa chính
title	Tiêu Đề Khóa Học	varchar	50		
dept_name	Tên Khoa	vachar	50		Khóa ngoại
credits	Số Tín Chỉ	numeric	2		check (credits > 0)

Bảng Classroom

Thuộc tính	Diễn giải	Kiểu	Độ rộng	Định dạng	Ràng buộc
building	Tên Tòa Nhà	varchar	5	Ax	
room_number	Số Phòng	varchar	5		Khóa chính
capacity	Sức Chứa	numeric	3		

Bảng Takes

Thuộc tính	Diễn giải	Kiểu	Độ rộng	Định dạng	Ràng buộc
ID	Mã Sinh Viên	varchar	11	BxxDCxx xxx	Khóa chính (kết hợp với 'course_id', 'class_id', 'semester', 'year')
course_id	Mã Khóa Học	varchar	8		Khóa chính (kết hợp với 'ID', 'class_id', 'semester', 'year')
class_id	Mã Lớp Học	varchar	8		Khóa chính (kết hợp với 'ID', 'course_id', 'semester', 'year')
semester	Học Kỳ	varchar	6		Khóa chính (kết hợp với 'ID', 'course_id', 'class_id', 'year')
year	Năm	numeric	4		Khóa chính (kết hợp với 'ID', 'course_id', 'class_id', 'semester')
grade	Điểm	numeric	2		

Bảng Teaches

Thuộc tính	Diễn giải	Kiểu	Độ rộng	Định dạng	Ràng buộc
ID	Mã Giảng Viên	varchar	5	GVxxx	Khóa chính (kết hợp với 'course_id', 'class_id', 'semester', 'year')
course_id	Mã Khóa Học	varchar	8		Khóa chính (kết hợp với 'ID', 'class_id', 'semester', 'year')
class_id	Mã Lớp Học	varchar	8		Khóa chính (kết hợp với 'ID', 'course_id', 'semester', 'year')
semester	Học Kỳ	varchar	6		Khóa chính (kết hợp với 'ID', 'course_id', 'class_id', 'year')
year	Năm	numeric	4		Khóa chính (kết hợp với 'ID', 'course_id', 'class_id', 'semester')

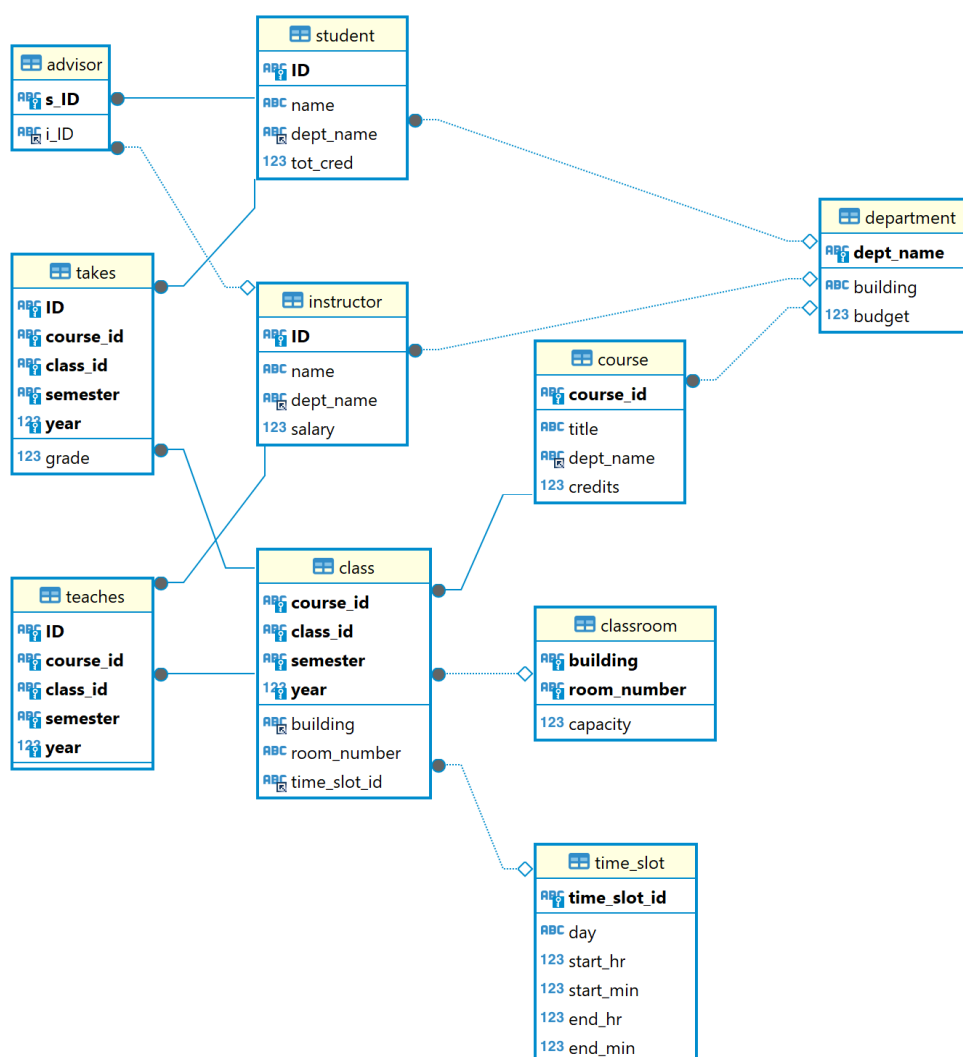
Bảng Advisor

Thuộc tính	Diễn giải	Kiểu	Độ rộng	Định dạng	Ràng buộc
s_ID	Mã Sinh Viên	varchar	11		Khóa chính
i_ID	Mã Giảng Viên	varchar	5		

Bảng Time_slot

Thuộc tính	Diễn giải	Kiểu	Độ rộng	Định dạng	Ràng buộc
time_slot_id	Mã Thời Gian	varchar	4	Kx_x	Khóa chính
day	Ngày	varchar	2		
start_hr	Giờ Bắt Đầu	numeric	2		check (start_hr >= 0 and start_hr < 24)
start_min	Phút Bắt Đầu	numeric	2		check (start_min >= 0 and start_min < 60)
end_hr	Giờ Kết Thúc	numeric	2		check (end_hr >= 0 and end_hr < 24)
end_min	Phút Kết Thúc	numeric	2		check (end_min >= 0 and end_min < 60)

c. Lược đồ dữ liệu quan hệ



7. Tạo bảng và các ràng buộc - [GitHub](#)

a. Tạo bảng department

```
create table department
    (dept_name      varchar(50),
     building       varchar(5),
     budget         numeric(12,2) check (budget > 0),
     primary key (dept_name)
    );
```

b. Tạo bảng instructor

```
create table instructor
    (ID              varchar(6),
     name            varchar(30) not null,
     dept_name       varchar(50),
     salary          numeric(8,2) check (salary > 0),
     primary key (ID),
     foreign key (dept_name) references department (dept_name)
     on delete set null
    );
```

c. Tạo bảng student

```
create table student
    (ID              varchar(11),
     name            varchar(30) not null,
     dept_name       varchar(50),
     tot_cred        numeric(3,0) check (tot_cred >= 0),
     primary key (ID),
     foreign key (dept_name) references department (dept_name)
     on delete set null
    );
```

d. Tạo bảng course

```
create table course
    (course_id       varchar(8),
     title           varchar(50),
     dept_name       varchar(50),
     credits         numeric(2,0) check (credits > 0),
     primary key (course_id),
     foreign key (dept_name) references department (dept_name)
     on delete set null
    );
```

e. Tạo bảng classroom

```
create table classroom
    (building          varchar(5),
     room_number       varchar(5),
     capacity          numeric(3,0),
     primary key (building, room_number)
    );
```

f. Tạo bảng time_slot

```
create table time_slot
    (time_slot_id varchar(4),
     day varchar(1),
     start_hr numeric(2) check (start_hr >= 0 and start_hr < 24),
     start_min numeric(2) check (start_min >= 0 and start_min < 60),
     end_hr numeric(2) check (end_hr >= 0 and end_hr < 24),
     end_min numeric(2) check (end_min >= 0 and end_min < 60),
     primary key (time_slot_id)
    );
```

g. Tạo bảng class

```
create table class
    (course_id          varchar(8),
     class_id           varchar(8),
     semester           varchar(6)
     check (semester in ('Fall', 'Winter', 'Spring', 'Summer')),
     year               numeric(4,0) check (year > 1952 and year < 2100),
     building           varchar(5),
     room_number        varchar(5),
     time_slot_id       varchar(4),
     primary key (course_id, class_id, semester, year),
     foreign key (course_id) references course (course_id)
     on delete cascade,
     foreign key (building, room_number) references classroom (building,
room_number)
     on delete set null,
     foreign key (time_slot_id) references time_slot (time_slot_id)
     on delete set null
    );
```

h. Tạo bảng teaches

```
create table teaches
    (ID                varchar(5),
     course_id         varchar(8),
     class_id          varchar(8),
     semester          varchar(6),
     year              numeric(4,0),
     primary key (ID, course_id, class_id, semester, year),
     foreign key (course_id, class_id, semester, year) references class
     (course_id, class_id, semester, year)
     on delete cascade,
     foreign key (ID) references instructor (ID)
     on delete cascade
    );
```

i. Tạo bảng takes

```
create table takes
    (ID                varchar(11),
     course_id         varchar(8),
     class_id          varchar(8),
     semester          varchar(6),
     year              numeric(4,0),
     grade             numeric(2,1),
     primary key (ID, course_id, class_id, semester, year),
     foreign key (course_id, class_id, semester, year) references class
     (course_id, class_id, semester, year)
     on delete cascade,
     foreign key (ID) references student (ID)
     on delete cascade
    );
```

j. Tạo bảng advisor

```
create table advisor
    (s_ID              varchar(11),
     i_ID              varchar(5),
     primary key (s_ID),
     foreign key (i_ID) references instructor (ID)
     on delete set null,
     foreign key (s_ID) references student (ID)
     on delete cascade
    );
```

8. Thêm dữ liệu vào bảng - [GitHub](#)

9. Thực hiện một số truy vấn

a. Truy vấn đơn giản - [GitHub](#)

-- 1. Sắp xếp danh sách người hướng dẫn có lương từ 5000.00 đến 5500.00 theo thứ tự lương tăng dần

SELECT

*

FROM

instructor

WHERE

salary BETWEEN 5000.00 AND 5500.00

ORDER BY salary ASC;

-- 2. Lấy ra thông tin những phòng ban có số lượng mã khóa học ≥ 20 sắp xếp theo thứ tự tăng dần của mã khóa học

SELECT dept_name, COUNT(course_id)

FROM course

GROUP BY dept_name

HAVING COUNT(course_id) ≥ 20

ORDER BY COUNT(course_id) ASC;

-- 3. Thống kê ngân sách các tòa nhà của phòng ban

SELECT

building,

SUM(budget) TongNgansach,

AVG(budget) TrungbinhNgansach,

MAX(budget) NgansachLonnhhat,

MIN(budget) NgansachNhonhat

FROM

department

GROUP BY building;

-- 4. Danh sách sinh viên đang học lớp 'INT1358'

SELECT s.ID, s.name

FROM student s

JOIN takes t ON s.ID = t.ID

WHERE t.course_id = 'INT1358';

-- 5. *Hiển thị thông tin gồm: ID, tên giảng viên, tiền lương, số lớp dạy với tên khoa là Khoa Đa phương tiện*

```
SELECT i.ID,  
       i.name,  
       i.salary,  
       COUNT(t.course_id) AS num_classes  
FROM instructor i  
LEFT JOIN teaches t ON i.ID = t.ID  
WHERE i.dept_name = 'Khoa Đa phương tiện'  
GROUP BY i.ID, i.name, i.salary;
```

-- 6. *Tìm khoa có mức lương trung bình cao nhất*

```
SELECT  
  dept_name  
FROM  
  instructor  
GROUP BY dept_name  
HAVING AVG(salary) = (  
  SELECT MAX(avg_salary)  
  FROM (  
    SELECT AVG(salary) AS avg_salary  
    FROM instructor  
    GROUP BY dept_name  
  ) AS subquery  
);
```

-- 7. *Tìm tổng số sinh viên (riêng biệt) đã tham gia các phần khóa học do giảng viên có ID GV056 giảng dạy*

```
SELECT  
  COUNT(DISTINCT ID)  
FROM  
  takes  
WHERE  
  EXISTS  
  ( SELECT  
    course_id, class_id, semester, year
```



```

FROM
    teaches
WHERE
    teaches.ID = 'GV056'
    AND takes.course_id = teaches.course_id
    AND takes.class_id = teaches.class_id
    AND takes.semester = teaches.semester
    AND takes.year = teaches.year);

```

-- 8. Sắp xếp số lượng giảng viên trong mỗi bộ môn giảng dạy một khóa học vào học kỳ mùa đông năm 2023 theo thứ tự giảm dần

```

SELECT
    instructor.ID,
    instructor.name,
    instructor.dept_name,
    COUNT(teaches.ID) AS instr_count
FROM
    instructor
JOIN
    teaches ON instructor.ID = teaches.ID
WHERE
    teaches.year = 2023
    AND teaches.semester = 'Winter'
GROUP BY
    instructor.ID,
    instructor.name,
    instructor.dept_name
ORDER BY
    instr_count DESC;

```

-- 9. Tổng số sinh viên khoá 2023

```

SELECT
    COUNT(*) AS total_students
FROM
    student
WHERE
    ID LIKE 'B23%';

```

-- 10. Tìm tất cả các phòng ban có tổng mức lương là lớn hơn mức trung bình của tổng tiền lương ở tất cả các bộ phận.

```
WITH dept_total (dept_name, VALUE) AS
    (SELECT
        dept_name, sum(salary)
        FROM
            instructor
        GROUP BY dept_name),
dept_total_AVG(VALUE) AS
    (SELECT
        AVG(VALUE)
        FROM dept_total)
SELECT
    dept_name
FROM
    dept_total, dept_total_avg
WHERE dept_total.VALUE > dept_total_avg.VALUE;
```

-- 11. Tìm sinh viên của khóa học có điểm nhỏ hơn 4

```
SELECT s.ID, s.name, s.dept_name, t.course_id, t.class_id, t.semester,
t.year, t.grade
FROM student s
JOIN takes t ON s.ID = t.ID
JOIN class c ON t.course_id = c.course_id
                AND t.class_id = c.class_id
                AND t.semester = c.semester
                AND t.year = c.year
WHERE t.grade < 4;
```

-- 12. Hiện thị danh sách học viên mà giảng viên có ID là 'GV005' đang cố vấn theo ID của học viên tăng dần

```
SELECT s.ID, s.name, s.dept_name, s.tot_cred
FROM student s
JOIN advisor a ON s.ID = a.s_ID
```

```
WHERE a.i_ID = 'GV005'
ORDER BY s.ID;
```

-- 13. Hiển thị danh sách các lớp học có điểm trung bình trên 7.0 trong học kỳ mùa đông năm 2023

```
SELECT
    c.course_id,
    c.class_id,
    c.semester,
    c.year,
    AVG(t.grade) AS average_grade
FROM
    class c
JOIN
    takes t ON c.course_id = t.course_id AND c.class_id = t.class_id AND
    c.semester = t.semester AND c.year = t.year
WHERE
    c.semester = 'Winter' AND c.year = 2023
GROUP BY
    c.course_id, c.class_id, c.semester, c.year
HAVING
    AVG(t.grade) > 7.0
ORDER BY
    c.course_id, c.class_id;
```

-- 14. Hiển thị danh sách 10 sinh viên có điểm cao nhất môn 'MUL1454'

```
SELECT s.ID, s.name, AVG(t.grade) AS avg_grade
FROM student s, takes t
WHERE s.ID = t.ID
    AND t.course_id = 'MUL1454'
GROUP BY s.ID, s.name
ORDER BY avg_grade DESC
LIMIT 10;
```

-- 15. Hiển thị danh sách lớp trong học kỳ hè năm 2022 có điểm trung bình học viên trên 7.0

```

SELECT DISTINCT c.course_id, c.class_id, c.semester, c.year,
AVG(t.grade)
FROM class c
JOIN takes t ON c.course_id = t.course_id AND c.class_id = t.class_id
AND c.semester = t.semester AND c.year = t.year
JOIN student s ON t.ID = s.ID
WHERE c.semester = 'Summer' AND c.year = 2022
GROUP BY c.course_id, c.class_id, c.semester, c.year
HAVING AVG(t.grade) > 7.0
ORDER BY AVG(t.grade);

```

b. Truy vấn bằng stored procedure - [GitHub](#)

- Các bước chung để tạo 1 stored procedure
 - Bước 1: Đặt lại DELIMITER (nếu cần): thay đổi ký tự kết thúc lệnh mặc định từ dấu chấm phẩy (;) sang một ký tự khác (như \$\$) để xử lý các khối lệnh
 - Bước 2: Viết câu lệnh CREATE PROCEDURE để định nghĩa tên của stored procedure và các tham số đầu vào/đầu ra.
 - Bước 3: Định nghĩa khối lệnh trong BEGIN ... END, đặt tất cả các câu lệnh SQL cần thiết trong khối BEGIN ... END.
 - Bước 4: Đặt lại DELIMITER về mặc định

● Các truy vấn

-- 1. Xóa một bản ghi cụ thể từ bảng advisor dựa trên hai tham số đầu vào là s_ID và i_ID

```

DELIMITER $$
CREATE PROCEDURE delete_advisor (
IN p_s_ID VARCHAR (11),
IN p_i_ID VARCHAR(5)
)
BEGIN
DELETE FROM advisor
WHERE s_ID = p_s_ID
AND i_ID = p_i_ID;
END $$
DELIMITER ;

```

-- Xóa một bản ghi cụ thể từ bảng advisor, trong đó sinh viên có s_ID là 'B21DCCB022', giảng viên có i_ID là 'GV002'

```
CALL delete_advisor('B21DCCB022', 'GV002');
```

-- 2. Xóa một phòng học cụ thể từ bảng classroom dựa trên các tham số đầu vào là building và room_number

```
DELIMITER $$
CREATE PROCEDURE delete_classroom (
  IN p_building varchar(5),
  IN p_room_number varchar(5))
BEGIN
  DELETE FROM classroom
  WHERE
    building = p_building AND
    room_number = p_room_number;
END$$
DELIMITER ;
-- Xóa phòng 101 ở tòa A1
CALL delete_classroom('A1', '101');
```

-- 3. Xóa các khóa học của một khoa

```
DELIMITER $$
CREATE PROCEDURE delete_course (IN p_deptname
  varchar(50))
BEGIN
  DELETE FROM course
  WHERE
    dept_name = p_deptname;
END$$
DELIMITER ;
-- Xóa các khóa học của “Khoa cơ bản”
CALL delete_course ('Khoa Cơ bản');
```

-- 4. Xóa một khoa

```
DELIMITER $$
CREATE PROCEDURE delete_department (IN p_dept_name
  varchar(50))
BEGIN
  DELETE FROM department
  WHERE
    dept_name = p_dept_name;
END $$
```

```

        DELIMITER ;
-- 5. Xóa một giảng viên dựa vào mã giảng viên
        DELIMITER $$
        CREATE PROCEDURE delete_instructor (IN p_ID varchar(6))
        BEGIN
            DELETE FROM instructor
            WHERE
                ID = p_ID;
        END$$
        DELIMITER ;
-- Xóa giảng viên có mã giảng viên là ('GV001')
        CALL delete_instructor ('GV001');

-- 6. Xóa một lớp học dựa vào mã lớp và mã khóa học
        DELIMITER $$
        CREATE PROCEDURE delete_class (
            p_courseid VARCHAR(8),
            p_classid VARCHAR(8)
        )
        BEGIN
            DELETE FROM class
            WHERE course_id = p_courseid
                AND class_id = p_classid;
        END$$
        DELIMITER ;

-- 7. Xóa một sinh viên dựa vào mã sinh viên
        DELIMITER $$
        CREATE PROCEDURE delete_student (IN p_ID varchar(11))
        BEGIN
            DELETE FROM student
            WHERE
                ID = p_ID;
        END$$
        DELIMITER ;

```

-- 8. Thêm một bản ghi mới vào bảng advisor với các giá trị cho các cột s_ID và i_ID

```
DELIMITER ??
CREATE PROCEDURE insert_advisor (
  IN sID VARCHAR(11),
    IN iID VARCHAR(5)
)
BEGIN
  INSERT INTO advisor (s_ID, i_ID)
    VALUES (sID, iID);
END ??
DELIMITER ;
```

-- chèn một sinh viên có mã sinh viên là 'B21DCPT100' và do
giảng viên có mã 'GV003' cố vấn
CALL insert_advisor ('B21DCPT100', 'GV003');

-- 9. Thêm 1 khoá học mới

```
DELIMITER ??
CREATE PROCEDURE insert_course (
  IN new_courseid INT,
    IN new_title VARCHAR (50),
    IN new_deptname VARCHAR (50),
    IN new_credits INT
)
BEGIN
  INSERT INTO course (course_id, title, dept_name, credits)
    VALUES (new_courseid, new_title, new_deptname,
new_credits);
END ??
DELIMITER ;
```

-- 10. Thêm 1 khoa mới

```
DELIMITER ??
CREATE PROCEDURE insert_department (
  IN new_deptname VARCHAR(50),
    IN new_building VARCHAR(5),
```

```

        IN new_budget DECIMAL(12,2)
    )
BEGIN
    INSERT INTO department (dept_name, building, budget)
    VALUES (new_deptname, new_building, new_budget);
END ??
DELIMITER ;

```

-- 11. *Thêm 1 khung thời gian học mới*

```

DELIMITER ??
CREATE PROCEDURE insert_timeslot (
    IN new_timeslotid VARCHAR(4),
    IN new_day VARCHAR(1),
    IN new_starthr DECIMAL (2,0),
    IN new_startmin DECIMAL (2,0),
    IN new_endhr DECIMAL (2,0),
    IN new_endmin DECIMAL (2,0)
)
BEGIN
    INSERT INTO time_slot (time_slot_id, day, start_hr,
    start_min, end_hr, end_min)
    VALUES (new_timeslotid, new_day, new_starthr,
    new_startmin, new_endhr, new_endmin);
END ??
DELIMITER ;

```

-- 12. *Cập nhật thông tin của 1 cố vấn*

```

DELIMITER ??
CREATE PROCEDURE update_advisor (
    IN new_iID VARCHAR(5),
    IN sID VARCHAR(11)
)
BEGIN
    UPDATE advisor
    SET i_ID = new_iID
    WHERE s_ID = sID;
END ??
DELIMITER ;

```


-- cập nhật thông tin của một cố vấn cho sinh viên có mã sinh viên là 'B21DCCB010'

CALL update_advisor ('GV005','B21DCCB010');

-- 13. Cập nhật sức chứa của một phòng học

DELIMITER ??

CREATE PROCEDURE update_capacity (

IN old_capacity DECIMAL (3,0),

IN new_capacity DECIMAL (3,0)

)

BEGIN

UPDATE classroom

SET capacity = new_capacity

WHERE capacity = old_capacity;

END ??

DELIMITER ;

-- Cập nhật phòng có sức chứa 55 người lên 60 người

CALL update_capacity (55,60);

-- 14. Cập nhật số tín chỉ của sinh viên theo mã sinh viên và khoa

DELIMITER ??

CREATE PROCEDURE update_totcred(

IN p_id VARCHAR(11),

IN p_deptname VARCHAR(50),

IN p_new_totcred DECIMAL(3,0)

)

BEGIN

UPDATE student

SET tot_cred = p_new_totcred

WHERE

id = p_id

AND

dept_name = p_deptname;

END ??

DELIMITER ;

-- Cập nhật số tín chỉ lên 50 cho sinh viên có mã sinh viên là 'B23DCCN017' thuộc 'Khoa Công nghệ thông tin'

```
CALL update_totcred('B23DCCN017', 'Khoa Công nghệ thông tin',  
50);
```

10. Tạo transaction thực hiện nghiệp vụ thực tế - [GitHub](#)

a. Xóa sinh viên ra khỏi CSDL

Các bước thực hiện:

- Bắt đầu transaction.
- Kiểm tra có tồn tại sinh viên này không. Nếu không tồn tại. → Rollback.
- Xóa dữ liệu với ID sinh viên cần xóa trong các bảng student, takes và advisor.
- Thực hiện Commit.

Cài đặt:

```
DELIMITER $$
```

```
CREATE PROCEDURE DELETE_STUDENT(IN p_ID  
VARCHAR(25))
```

```
BEGIN
```

```
START TRANSACTION;
```

```
SELECT @flag:=COUNT(ID) FROM student WHERE ID = p_ID;
```

```
DELETE FROM takes WHERE ID = p_ID;
```

```
COMMIT;
```

```
DELETE FROM advisor WHERE s_ID = p_ID;
```

```
COMMIT;
```

```
DELETE FROM student WHERE ID = p_ID;
```

```
IF (@flag = 0)
```

```
THEN ROLLBACK;
```

```
ELSE COMMIT;
```

```
END IF;
```

```
END$$
```

```
DELIMITER ;
```

b. Cập nhật tổng số tín cho học viên

Các bước thực hiện:

- Bắt đầu transaction.
- Kiểm tra có tồn tại học phần cụ thể như vậy không. Nếu không tồn tại. → Rollback.
- Điểm học phần lớn hơn 4. → Cập nhật tot_cred trong bảng student. → Commit.
- Điểm học phần bằng 4 → Commit.

Cài đặt:

DELIMITER \$\$

```
CREATE PROCEDURE UPDATE_CREDITS(
```

```
IN p_ID VARCHAR(11),
```

```
IN p_course_id VARCHAR(8),
```

```
IN p_class_id VARCHAR(8),
```

```
IN p_semester VARCHAR(6),
```

```
IN p_year DECIMAL(4,0),
```

```
IN p_grade DECIMAL(2,1))
```

```
BEGIN
```

```
DECLARE EXIT HANDLER FOR SQLEXCEPTION
```

```
BEGIN
```

```
    SHOW ERRORS;
```

```
    ROLLBACK;
```

```
END;
```

```
START TRANSACTION;
```

```
SELECT @flag_fi := COUNT(course_id) FROM course WHERE  
course_id = p_course_id;
```

```
SELECT @flag_se := grade FROM takes  
WHERE ID = p_ID AND course_id = p_course_id  
AND class_id = p_class_id AND semester = p_semester AND year =  
p_year;
```

```
SELECT @cred := credits FROM course WHERE course_id =  
p_course_id;
```

```

IF (@flag_fi = 0) THEN
ROLLBACK;
ELSEIF (@flag_se > 4) THEN
UPDATE student SET tot_cred = tot_cred + @cred WHERE ID = p_ID;
COMMIT;
ELSE COMMIT;
END IF;

SELECT @flag_fi, @flag_se;
END$$

DELIMITER ;

```

c. Xóa giảng viên ra khỏi CSDL

Các bước thực hiện:

- Bắt đầu transaction.
- Kiểm tra có tồn tại giảng viên này không. Nếu không tồn tại. → Rollback.
- Xóa dữ liệu với ID giảng viên cần xóa trong các bảng instructor, teaches và advisor.
- Thực hiện Commit.

Cài đặt:

```
DELIMITER $$
```

```

CREATE PROCEDURE DELETE_INSTRUCTOR(IN p_ID
VARCHAR(25))
BEGIN

```

```
START TRANSACTION;
```

```

SELECT @flag:=COUNT(ID) FROM instructor WHERE ID = p_ID;
DELETE FROM teaches WHERE ID = p_ID;
COMMIT;
DELETE FROM advisor WHERE s_ID = p_ID;
COMMIT;
DELETE FROM instructor WHERE ID = p_ID;

```

```
IF (@flag = 0)
THEN ROLLBACK;
ELSE COMMIT;
END IF;
```

```
END$$
```

```
DELIMITER ;
```