

## **BÁO CÁO CHI TIẾT BÀI TẬP LỚN 1**

### **I. Trích dẫn đặc tả yêu cầu ứng dụng**

#### **Đề tài 1: Learning/ Teaching**

Mô tả: Xem xét quản lý các lớp học trong hệ thống giáo dục theo quy chế tín chỉ.

Trong hệ thống giáo dục theo quy chế tín chỉ, sinh viên có thể đăng ký một hoặc nhiều môn học ở mỗi học kỳ nếu ở trạng thái học tập bình thường, không thể đăng ký môn học ở một học kỳ nếu ở trạng thái tạm dừng hoặc buộc thôi học ở học kỳ đó. Trong một học kỳ, một môn học được mở và triển khai gồm một hoặc nhiều lớp. Mỗi lớp do một hoặc nhiều giảng viên phụ trách ở các tuần học khác nhau trong cùng một học kỳ và do một giảng viên phụ trách chính để ghi điểm cho các sinh viên vào cuối học kỳ. Mỗi lớp không có quá 60 sinh viên đăng ký. Giả sử mỗi môn học có số tín chỉ từ 1 đến 3. Khi này, mỗi sinh viên không được đăng ký quá 18 tín chỉ cho mỗi học kỳ.

Cho mỗi môn học, một hoặc nhiều giáo trình được sử dụng. Một giáo trình có thể là giáo trình chính cho nhiều hơn một môn học khác. Mỗi giảng viên phụ trách chính môn học ở mỗi học kỳ sẽ chỉ định từ 1 đến 3 giáo trình chính được dùng cho môn học ở tất cả các lớp được mở ở học kỳ đó. Lưu ý có quy định rằng các giáo trình chính phải được xuất bản không quá 10 năm. Một giáo trình được xuất bản bởi một nhà xuất bản trong hoặc ngoài nước. Nhiều giáo trình được mua chung từ một nhà xuất bản. Nhiều giáo trình thuộc về cùng một lĩnh vực có thể được mua từ các nhà xuất bản khác. Mỗi giáo trình được biên soạn bởi một hay nhiều tác giả và ngược lại, mỗi tác giả biên soạn một hay nhiều giáo trình.

Chương trình ứng dụng được phát triển từ đề tài này dự kiến được dùng bởi:

**(i). Phòng Đào Tạo**

**(ii). Khoa quản lý chương trình đào tạo**

**(iii). Giảng viên**

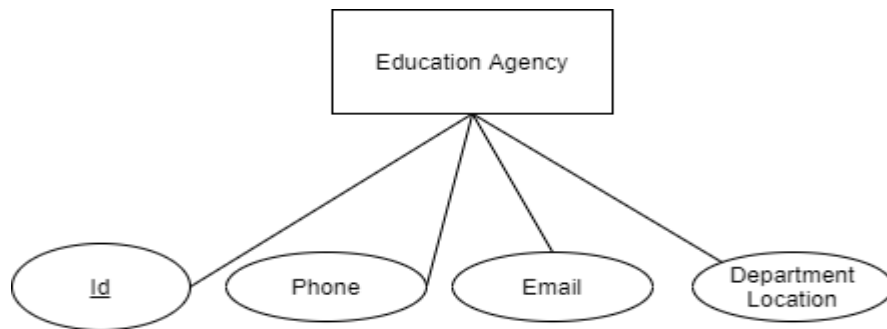
**(iv). Sinh viên**

Nhân sự ở Phòng Đào Tạo, Khoa quản lý chương trình đào tạo, và Giảng viên đều là nhân viên của đơn vị giáo dục yêu cầu phát triển chương trình ứng dụng này. Phòng Đào Tạo có thể thao tác (xem và cập nhật (bao gồm thêm, xóa, sửa) đối với dữ liệu về các lớp được mở cho đăng ký của tất cả các môn học. Khoa quản lý chương trình đào tạo có thể thao tác đối với dữ liệu về các môn học và phân công giảng viên phụ trách các lớp cho các môn học do khoa quản lý. Giảng viên có thể cập nhật giáo trình cho các môn học do mình phụ trách và xem các dữ liệu về các môn học, lớp học, và sinh viên do mình phụ trách. Sinh viên có thể cập nhật các đăng ký lớp học đối với các môn học và xem các dữ liệu về các môn học, lớp học, giáo trình liên quan cho các môn học mà mình đã đăng ký được.

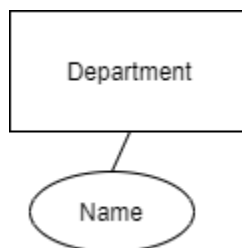
## II. Báo cáo chi tiết

### A1. Liệt kê các kiểu thực thể mạnh và thuộc tính

- *Phòng ban* : kiểu thực thể tổng quát hóa của khoa quản lý và phòng đào tạo.
  - Thuộc tính: Id, Phone, Email, Department Location
  - Khóa chính: Id
  - Miền trị:
    - Id: varchar(10)
    - Phone: varchar(10)
    - Email: varchar(50)
    - Department Location: varchar(10)

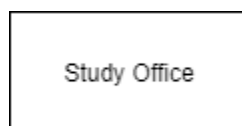


- *Khoa quản lý*:



- Thuộc tính: Name
- Miền trị:
  - Name: varchar(50)

- *Phòng đào tạo*:

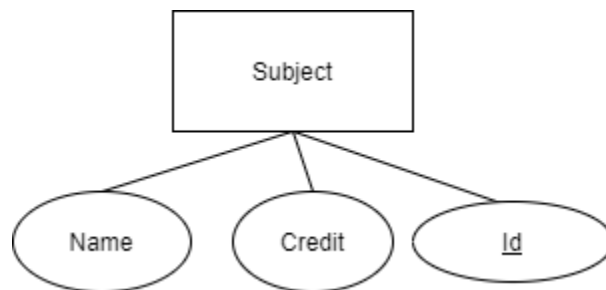


Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

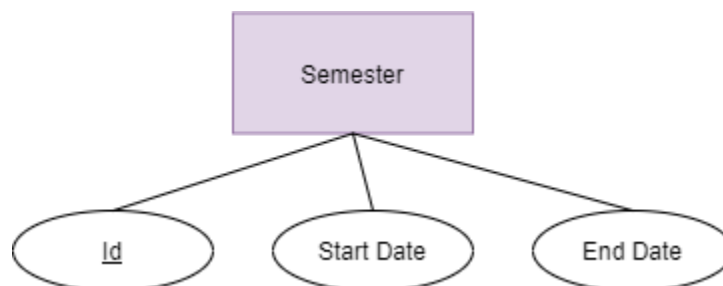
Học kỳ: **1**, Năm học: **2020-2021**

Giảng viên: **TS. Võ Thị Ngọc Châu**

- *Môn học*: lưu trữ thông tin về môn học có trong hệ thống hiện tại, các môn học có thể được mở (hoặc không) bởi một khoa ở mỗi học kỳ.
  - Thuộc tính: Id, Name, Credit
  - Khóa chính: Id
  - Miền trị:
    - Id: varchar(10)
    - Name: varchar(50)
    - Credit (số tín chỉ) : Int[1..3]



- *Học kỳ*: lưu trữ thông tin các học kỳ được trường mở.
  - Thuộc tính: Id, Start Date, End Date
  - Khóa chính: Id
  - Miền trị:
    - Id: varchar(10)
    - Start Date: date
    - End Date: date (**DATEDIFF(DAY, startDate, endDate) > 0**)

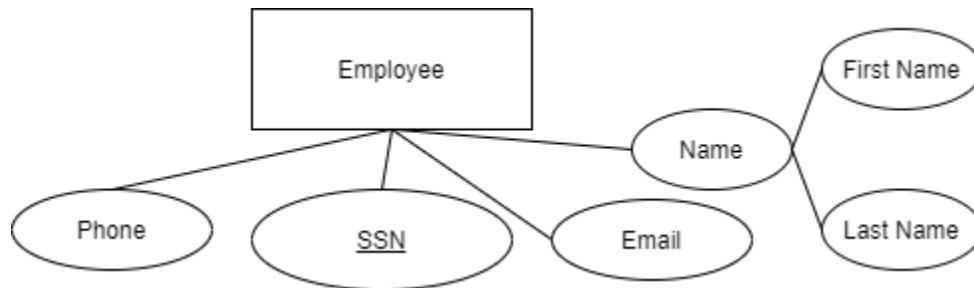


Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

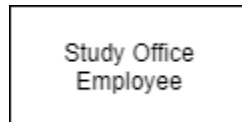
Học kỳ: **1**, Năm học: **2020-2021**

Giảng viên: **TS. Võ Thị Ngọc Châu**

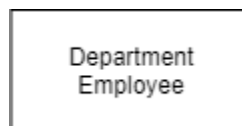
- *Nhân viên*: là tổng quát hóa của cán bộ công chức, nhân viên của nhà trường gồm: cán bộ phòng đào tạo, cán bộ khoa quản lý, giáo viên.
  - Thuộc tính: SSN, Phone, Email, Name (thuộc tính phức kết hợp từ First Name, Last Name)
  - Khóa chính: SSN
  - Miền trị:
    - SSN: varchar(10)
    - Phone: varchar(10)
    - Email: varchar(50)
    - First Name: varchar(50)
    - Last Name: varchar(50)



- *Nhân viên phòng đào tạo*:



- *Nhân viên khoa quản lý*:

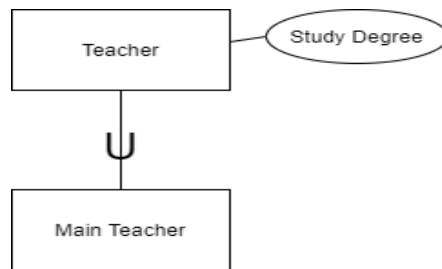


- *Giáo viên*: có thêm thuộc tính học vị. Để làm rõ vai trò của giáo viên đối với môn học và lớp học, giáo viên có thêm chuyên biệt hóa giáo viên chính.
  - Thuộc tính: Study Degree
  - Miền trị:
    - Study Degree: varchar(50)

Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

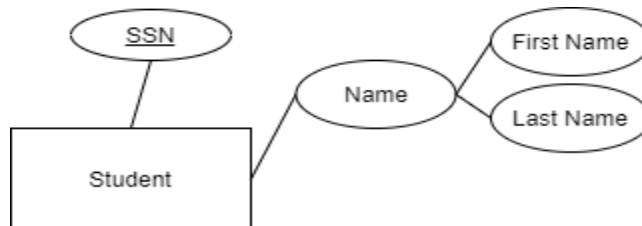
Học kỳ: **1**, Năm học: **2020-2021**

Giảng viên: **TS. Võ Thị Ngọc Châu**



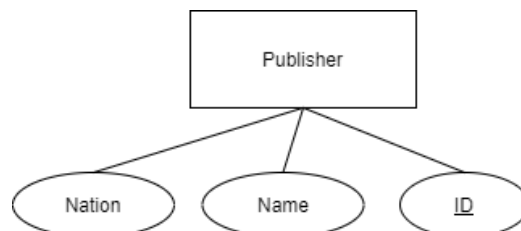
- *Sinh viên*: lưu trữ thông tin của tất cả các sinh viên đang học tại trường. Ở phần thực thể sinh viên nhóm không đề trạng thái học tập của sinh viên là thuộc tính đa trị để biểu diễn trạng thái học tập của sinh viên theo học kỳ vì có khả năng sẽ không kiểm soát được việc trong cơ sở dữ liệu tồn tại nhiều trạng thái học tập của cùng một sinh viên tại cùng một học kỳ. Trạng thái học tập sẽ được thiết kế thành quan hệ và đề cập ở phần tiếp theo.

- Thuộc tính: SSN, Name (thuộc tính phức kết hợp từ First Name, Last Name)
- Khóa chính: SSN
- Miền trị:
  - SSN: varchar(10)
  - First Name: varchar(50)
  - Last Name: varchar(50)



- *Nhà xuất bản*: lưu trữ thông tin về nhà xuất bản của các giáo trình dùng trong các môn học.

- Thuộc tính: Id, Nation, Name
- Khóa chính: Id
- Miền trị:
  - Id: varchar(10)
  - Nation: varchar(50)
  - Name: varchar(50)

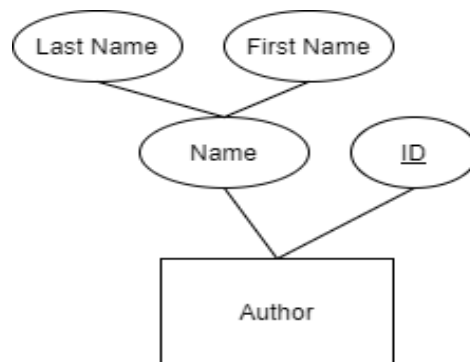


Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

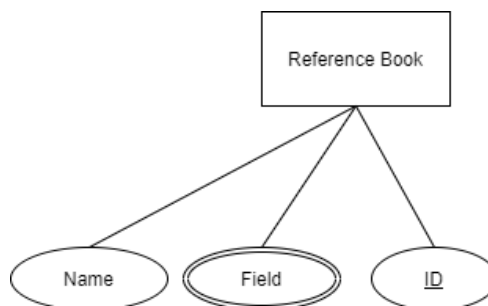
Học kỳ: **1**, Năm học: **2020-2021**

Giảng viên: **TS. Võ Thị Ngọc Châu**

- *Tác giả*: lưu trữ thông tin về tác giả của các giáo trình.
  - Thuộc tính: Id, Name (thuộc tính phức kết hợp từ First Name, Last Name)
  - Khóa chính: Id
  - Miền trị:
    - Id: varchar(10),
    - First Name: varchar(50)
    - Last Name: varchar(50)

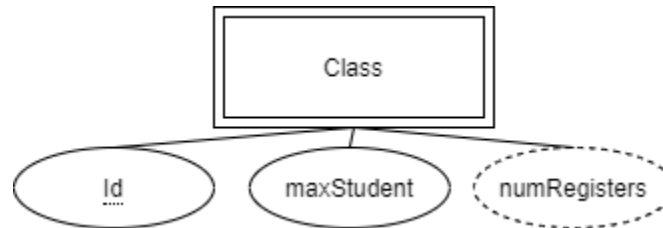


- *Giáo trình*: Lưu trữ thông tin về giáo trình được sử dụng trong các môn học
  - Thuộc tính: Id, Name, Field (thuộc tính đa trị)
  - Khóa chính: Id
  - Miền trị:
    - Id: varchar(10)
    - Name: varchar(50)
    - Field: varchar(50)

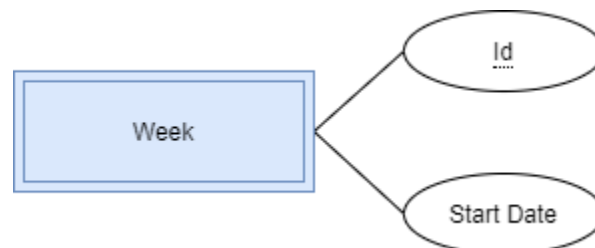


## A2. Liệt kê các kiểu thực thể yếu và thuộc tính

- *Lớp*: ứng với mỗi môn học được mở trong mỗi học kỳ, các lớp của môn này sẽ được thành lập. Vì vậy lớp sẽ được định danh bằng môn học được mở trong một kỳ xác định.
  - Thuộc tính: Id, Max Student, numRegisters ( thuộc tính dẫn xuất, biểu diễn tổng số sinh viên đã đăng ký lớp)
  - Khóa riêng phần: Id
  - Miền trị:
    - Id: varchar(10)
    - Max Student: Int ( theo yêu cầu đề bài tất cả Max Student của lớp đều là 60)



- *Tuần học*: Tuần học là thực thể yếu được định danh theo từng học kỳ. Nhóm giả định một học kỳ có 15 tuần học và có số thứ tự lần lượt từ 1 đến 15.
  - Thuộc tính: Id, Start Date
  - Khóa riêng phần: Id
  - Miền trị:
    - Id: Int[1..15]
    - Start Date: date



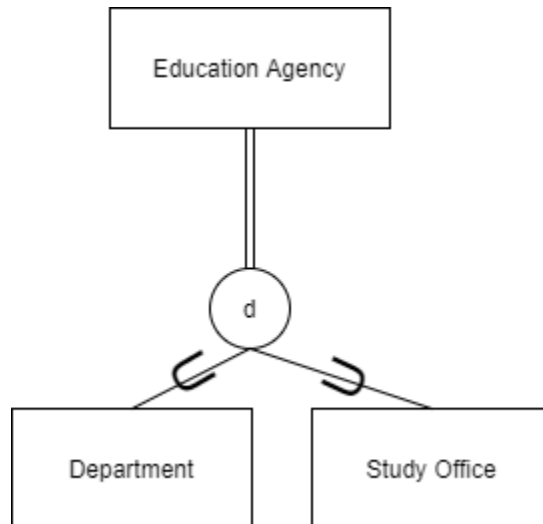
Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

Học kỳ: **1**, Năm học: **2020-2021**

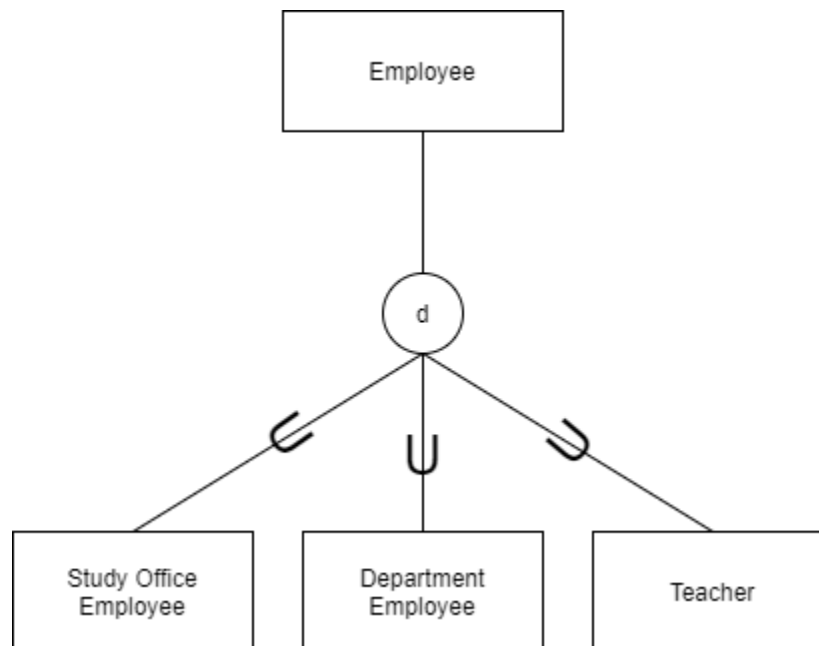
Giảng viên: **TS. Võ Thị Ngọc Châu**

### A3. Các kiểu mối liên kết

- Trường có các phòng ban là văn phòng khoa hoặc phòng đào tạo.



- Các nhân viên có thể là cán bộ phòng đào tạo, cán bộ văn phòng khoa hoặc giáo viên.



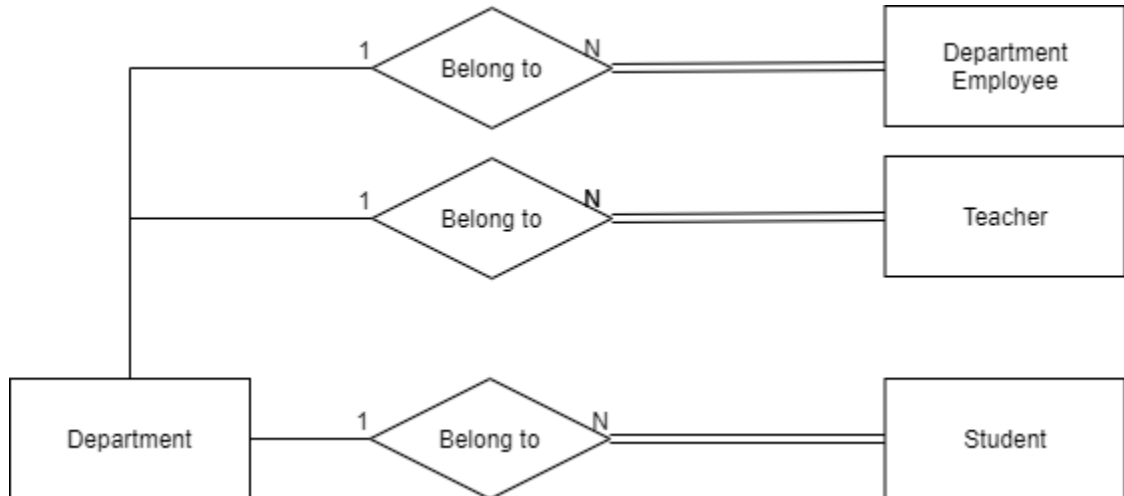


Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

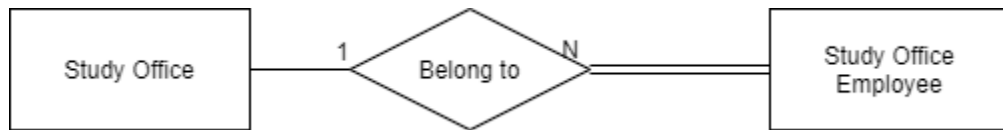
Học kỳ: **1**, Năm học: **2020-2021**

Giảng viên: **TS. Võ Thị Ngọc Châu**

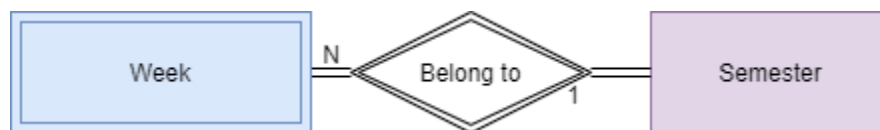
- Mỗi sinh viên, giảng viên, cán bộ văn phòng khoa đều phải thuộc một khoa quản lý. Một khoa quản lý có thể có nhiều (hoặc không có) sinh viên, giảng viên, cán bộ văn phòng khoa.



- Mỗi cán bộ phòng đào tạo đều phải thuộc phòng đào tạo.



- Mỗi tuần học được định danh bằng học kỳ. Mỗi học kỳ đều phải có những tuần học.

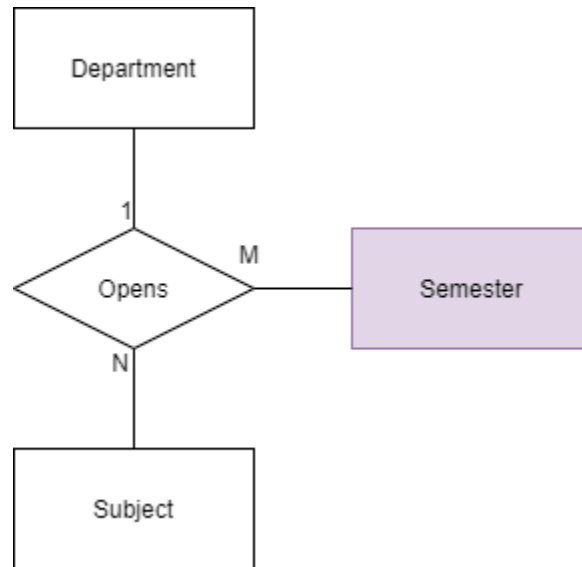


Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

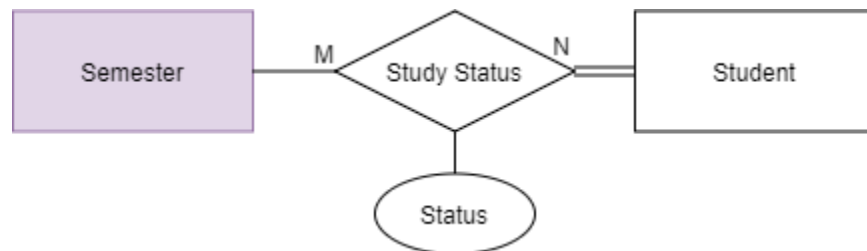
Học kỳ: **1**, Năm học: **2020-2021**

Giảng viên: **TS. Võ Thị Ngọc Châu**

- Tại mỗi học kỳ, khoa quản lý có thể mở nhiều môn học và mỗi kỳ một môn học chỉ được mở bởi một khoa quản lý:



- Tình trạng học tập của sinh viên được ghi nhận cho mỗi học kỳ và là thuộc tính của quan hệ giữa sinh viên và học kỳ. Tình trạng học tập sẽ thuộc tập {Bình thường, Tạm dừng, Thôi học} :

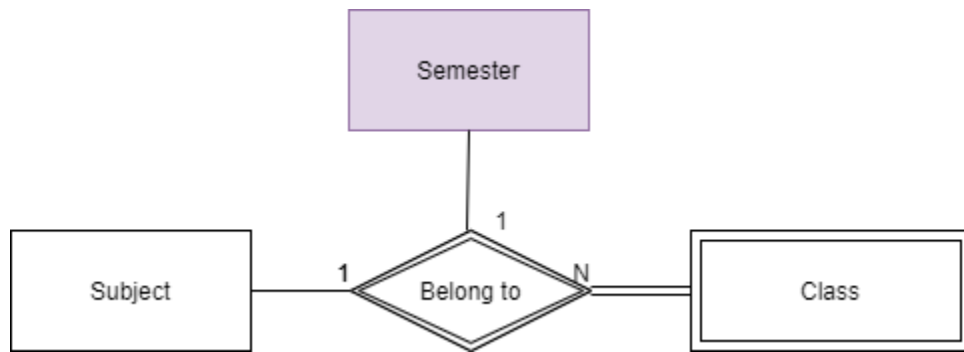


Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

Học kỳ: **1**, Năm học: **2020-2021**

Giảng viên: **TS. Võ Thị Ngọc Châu**

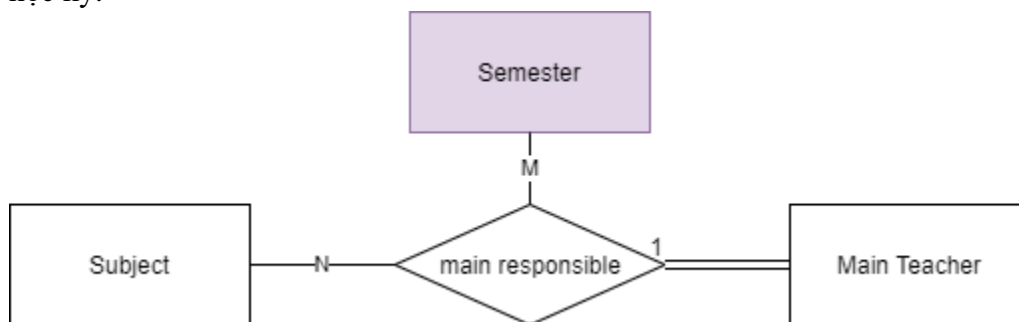
- Các lớp học sẽ được mở ứng với mỗi môn học được mở trong một học kỳ xác định.



- Sinh viên đăng ký môn học bằng cách đăng ký học một trong những lớp học ứng với môn học được mở trong học kỳ.



- Trong thiết kế này, nhóm giả sử giáo viên chính sẽ là người phụ trách chính môn học cho mỗi học kỳ xác định. Giáo viên chính sẽ chọn ra các giáo trình được sử dụng cho các lớp học. Mỗi môn học tại mỗi học kỳ sẽ có một giáo viên phụ trách chính. Một giáo viên có thể phụ trách chính một môn ở nhiều kỳ và phụ trách chính nhiều môn ở một học kỳ.

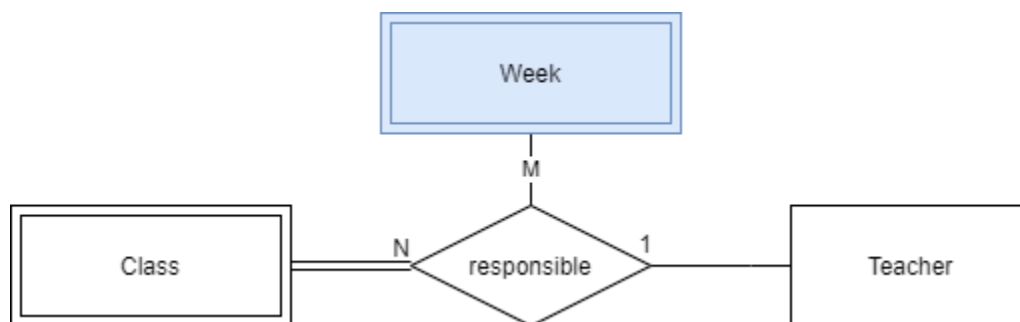


- Các giáo viên được phân công phụ trách lớp ở từng tuần học xác định bởi phòng đào tạo. Mỗi lớp chỉ có một giáo viên phụ trách cho mỗi tuần. Mỗi giáo viên có thể phụ trách nhiều lớp trong một tuần và một lớp trong nhiều tuần, và lớp học luôn phải có sự phụ trách của giáo viên.

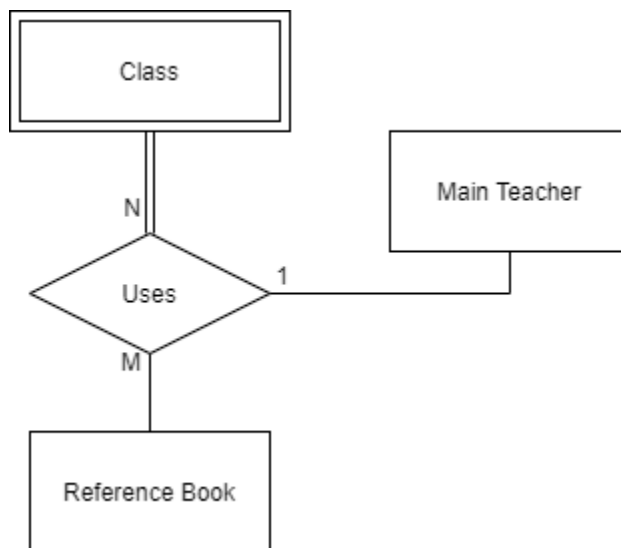
Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

Học kỳ: **1**, Năm học: **2020-2021**

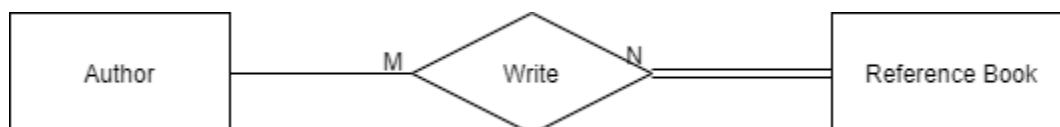
Giảng viên: **TS. Võ Thị Ngọc Châu**



- Mỗi giảng viên phụ trách chính môn học ở mỗi học kỳ sẽ chỉ định từ 1 đến 3 giáo trình chính được dùng cho môn học ở tất cả các lớp được mở ở học kỳ đó.



- Quan hệ giữa tác giả và giáo trình.

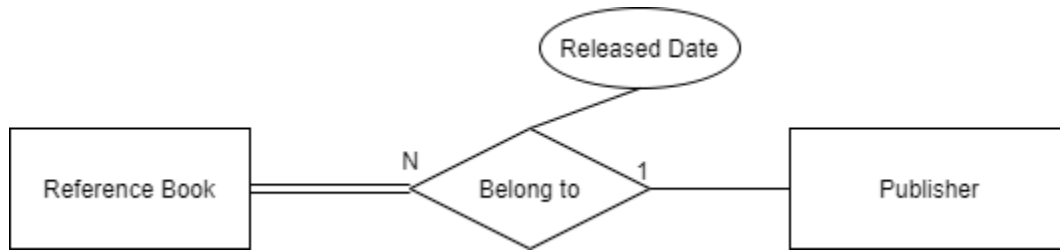


Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

Học kỳ: **1**, Năm học: **2020-2021**

Giảng viên: **TS. Võ Thị Ngọc Châu**

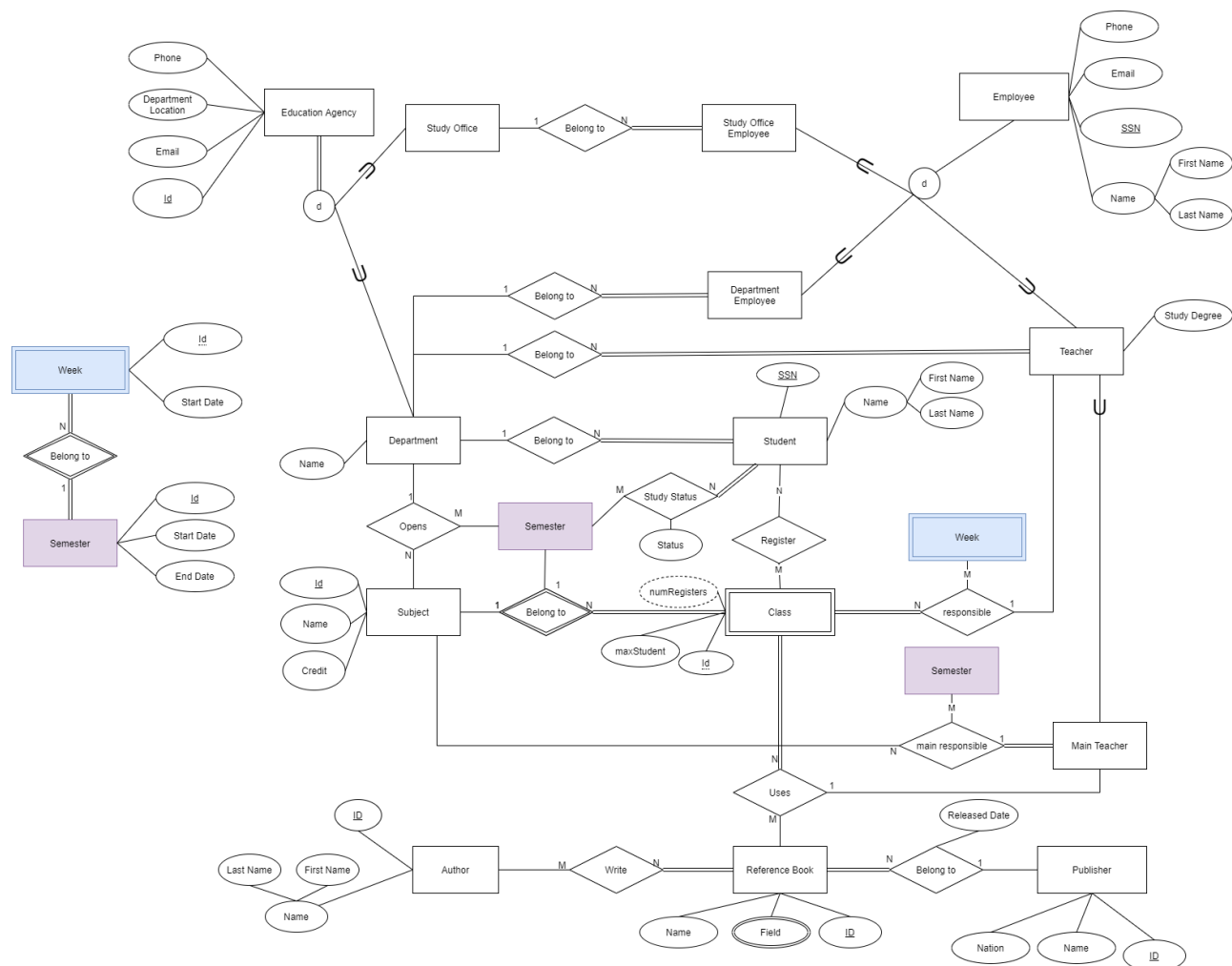
- Mỗi giáo trình đều có một nhà xuất bản. Năm xuất bản của giáo trình là thuộc tính của quan hệ và cách hiện tại không quá 10 năm.



#### A4. Các ràng buộc

- Sinh viên không thể đăng ký môn học nếu tình trạng môn học là tạm dừng hoặc buộc thôi học.
- Một lớp không có quá 60 sinh viên đăng ký.
- Tổng số tín chỉ đăng ký một học kỳ của sinh viên là 18.
- Mỗi lớp học sử dụng 1 đến 3 giáo trình.
- Giáo trình chỉ được xuất bản trong vòng 10 năm trở lại.
- Số tín chỉ của một môn học từ 1 đến 3.

### A5. Lược đồ thực thể - mối liên kết



### A6. Ánh xạ cơ sở dữ liệu quan hệ

**EducationAgency**(id,phone,departmentLocation,email)

**StudyOffice**(id)

Foreign key: id to EducationAgency.id

**Department**(id,name)

Foreign key: id to EducationAgency.id

Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

Học kỳ: **1**, Năm học: **2020-2021**

Giảng viên: **TS. Võ Thị Ngọc Châu**

**Employee**(ssn, phone, email, firstName, lastName)

**StudyOfficeEmployee**(ssn, soId)

Foreign key: ssn to Employee.ssn

Foreign key: soId to StudyOffice.id NOT NULL

**DepartmentEmployee**(ssn, dId)

Foreign key: ssn to Employee.ssn

Foreign key: dId to Department.id NOT NULL

**Teacher**(ssn, dId, studyDegree)

Foreign key: ssn to Employee.ssn

Foreign key: dId to Department.id NOT NULL

**MainTeacher**(ssn)

Foreign key: ssn to Teacher.ssn

**Student**(ssn, firstName, lastName, dId)

Foreign key: dId to Department.id NOT NULL

**StudyStatus**(sid, semesterId, status)

Foreign key: sid to Student.ssn, semesterId to Semester.id

Check for status in ('normal', 'pause', 'stop')

Check total participation of Student

**Subject**(id, name, credit)

Check for credit between [1, 3]

Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

Học kỳ: **1**, Năm học: **2020-2021**

Giảng viên: **TS. Võ Thị Ngọc Châu**

**Semester**(id, startDate, endDate)

Check for startDate < endDate

**RererenceBook**(id, name, releasedDate)

Check for releasedDate to now less than 10 years

**Author**(ssn, firstName, lastName)

**Publisher**(id, name, nation)

**Class**(id, Semester\_id, Subject\_id, maxStudent)

Foreign key: Semester\_id to Semester.id, Subject\_id to Subject.id

Check for maxStudent = 60

Check for number of registers <= maxStudent

**Opens**(Semester\_id, Subject\_id, Department\_id)

Foreign key: Semester\_id to Semester.id,

Subject\_id to Subject.id,

Department\_id to Department.id

**Uses**(Semester\_id, Class\_id, Subject\_id, ReferenceBook\_id, MainTeacher\_ssn)

Foreign key: MainTeacher\_ssn to MainTeacher.ssn,

ReferenceBook\_id to ReferenceBook.id

Semester\_id, Class\_id, Subject\_id to Class.id, Class.Semester\_id, Class.  
Subject\_id

Check for Total participation of Class

**Write**(Author\_ssn, Book\_id)

Foreign key: Author\_ssn to Author.ssn,



Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

Học kỳ: **1**, Năm học: **2020-2021**

Giảng viên: **TS. Võ Thị Ngọc Châu**

Book\_id to ReferenceBook.id

Check for Total participation of ReferenceBook

**Field**(Book\_id, Field)

Foreign key: Book\_id to Book.id

**ReferenceBook**(id, ..., Publisher\_id)

Foreign key: Publisher\_id to Publisher.id

Not Null: Publisher\_id

**Register**(Student\_id, Semester\_id, Subject\_id, Class\_id)

Foreign key: Student\_id to Student.ssn,

(Semester\_id, Subject\_id, Class\_id) to Class.(Semester\_id, Subject\_id, id)

**MainResponsible**(Semester\_id, Subject\_id, MainTeacher\_ssn)

Foreign key: MainTeacher\_ssn to MainTeacher.ssn, Semester\_id to Semester.id, Subject\_id to Subject.id

Check for total participation of MainTeacher

**Week**(id, Semster\_id, startDate)

Foreign key: Semester\_id to Semester.id

**Responsible**(Semester\_id, Subject\_id, Class\_id, Week\_id, Week\_Semester\_id, Teacher\_ssn)

Foreign key: (Semester\_id, Subject\_id, Class\_id) to Class.(Semester\_id, Subject\_id, id)

(Week\_id, Week\_Semester\_id) to Week.(id, Semester\_id)

Teacher\_ssn to Teacher.ssn

Check for Week\_Semester\_id = Semester\_id

Check for total participation of Class

Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

Học kỳ: **1**, Năm học: **2020-2021**

Giảng viên: **TS. Võ Thị Ngọc Châu**

## A7. Hiện thực cơ sở dữ liệu trên SQL Server

```
IF db_id('Class_Registration') IS NULL
    CREATE DATABASE Class_Registration

GO

/*Create Education Agency*/
CREATE TABLE Class_Registration.dbo.EducationAgency(
    id varchar(10) PRIMARY KEY,
    phone varchar(10),
    departmentLocation varchar(10),
    email varchar(50),
);

/*Create Study Office*/
CREATE TABLE Class_Registration.dbo.StudyOffice(
    id varchar(10) PRIMARY KEY,
    FOREIGN KEY (id) REFERENCES Class_Registration.dbo.EducationAgency(id) ON DELETE
CASCADE
);

/*Create Department*/
CREATE TABLE Class_Registration.dbo.Department(
    id varchar(10) PRIMARY KEY,
    [name] varchar(50),
    FOREIGN KEY (id) REFERENCES Class_Registration.dbo.EducationAgency(id) ON DELETE
CASCADE
);

/*Create Employee*/
CREATE TABLE Class_Registration.dbo.Employee(
    ssn varchar(10) PRIMARY KEY,
    phone varchar(10),
    email varchar(50),
    firstName varchar(50),
    lastName varchar(50),
);

/*CREATE Study Office Employee*/
CREATE TABLE Class_Registration.dbo.StudyOfficeEmployee(
    ssn varchar(10) PRIMARY KEY,
    soId varchar(10) NOT NULL,
    FOREIGN KEY (soId) REFERENCES Class_Registration.dbo.StudyOffice(id) ON DELETE
CASCADE,
    FOREIGN KEY (ssn) REFERENCES Class_Registration.dbo.Employee(ssn) ON DELETE
CASCADE,
);

/*Create Department Employee*/
CREATE TABLE Class_Registration.dbo.DepartmentEmployee(
    ssn varchar(10) PRIMARY KEY,
    dId varchar(10) NOT NULL,
    FOREIGN KEY (dId) REFERENCES Class_Registration.dbo.Department(id) ON DELETE
CASCADE,
```

Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

Học kỳ: **1**, Năm học: **2020-2021**

Giảng viên: **TS. Võ Thị Ngọc Châu**

```
        FOREIGN KEY (ssn) REFERENCES Class_Registration.dbo.Employee(ssn) ON DELETE
CASCADE,
    );

/*Create Teacher*/
CREATE TABLE Class_Registration.dbo.Teacher(
    ssn varchar(10) PRIMARY KEY,
    studyDegree varchar(50),
    dId varchar(10) NOT NULL,
    FOREIGN KEY (dId) REFERENCES Class_Registration.dbo.Department(id) ON DELETE
CASCADE,
    FOREIGN KEY (ssn) REFERENCES Class_Registration.dbo.Employee(ssn) ON DELETE
CASCADE
);

/*Create MainTeacher*/
CREATE TABLE Class_Registration.dbo.MainTeacher(
    ssn VARCHAR(10) PRIMARY KEY,
    FOREIGN KEY (ssn) REFERENCES Class_Registration.dbo.Teacher(ssn) ON DELETE CASCADE
);

/*Create Student*/
CREATE TABLE Class_Registration.dbo.Student(
    ssn varchar(10) PRIMARY KEY,
    firstName varchar(50),
    lastName varchar(50),
);

/*Create Semester*/
CREATE TABLE Class_Registration.dbo.Semester(
    id varchar(10) PRIMARY KEY,
    startDate date,
    endDate date,
    CHECK (DATEDIFF(DAY, startDate, endDate) > 0)
);

CREATE TABLE Class_Registration.dbo.StudyStatus(
    [sid] varchar(10),
    semesterId varchar(10),
    [status] varchar(50),
    CHECK ([status] in ('normal', 'pause', 'stop')),
    FOREIGN KEY ([sid]) REFERENCES Class_Registration.dbo.Student(ssn) ON DELETE
CASCADE,
    FOREIGN KEY (semesterId) REFERENCES Class_Registration.dbo.Semester(id) ON DELETE
CASCADE,
    PRIMARY KEY ([sid], semesterId),
);

/*Create Subject*/
CREATE TABLE Class_Registration.dbo.[Subject](
    id varchar(10) PRIMARY KEY,
    [name] varchar(50),
    credit INT CHECK (credit >=1 AND credit <=3),
);
```

Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

Học kỳ: **1**, Năm học: **2020-2021**

Giảng viên: **TS. Võ Thị Ngọc Châu**

```
/*Create ReferenceBook*/
CREATE TABLE Class_Registration.dbo.ReferenceBook(
    id varchar(10) PRIMARY KEY,
    [name] varchar(50),
    releasedDate DATE CHECK(DATEDIFF(YEAR, releasedDate, GETDATE()) <= 10),
);

/*Create Author*/
CREATE TABLE Class_Registration.dbo.Author(
    ssn varchar(10) PRIMARY KEY,
    firstName varchar(50),
    lastName varchar(50),
);

/*Create Publisher*/
CREATE TABLE Class_Registration.dbo.Publisher(
    id varchar(10) PRIMARY KEY,
    [name] varchar(50),
    nation varchar(50),
);

/*CREATE Class*/
CREATE TABLE Class_Registration.dbo.Class(
    id varchar(10),
    maxStudent INT CHECK (maxStudent <= 60),
    Semester_id varchar(10),
    Subject_id varchar(10),
    FOREIGN KEY (Semester_id) REFERENCES Class_Registration.dbo.Semester(id) ON
DELETE CASCADE,
    FOREIGN KEY (Subject_id) REFERENCES Class_Registration.dbo.[Subject](id) ON DELETE
CASCADE,
    PRIMARY KEY (Semester_id, Subject_id, id)
);

ALTER TABLE Class_Registration.dbo.Student
ADD
    dId varchar(10) NOT NULL,
    FOREIGN KEY (dId) REFERENCES Class_Registration.dbo.Department(id) ON DELETE
CASCADE;

--Relationship: Opens
CREATE TABLE Class_Registration.dbo.Opens(
    Semester_id varchar(10),
    Subject_id varchar(10),
    Department_id varchar(10),
    FOREIGN KEY (Semester_id) REFERENCES Class_Registration.dbo.Semester(id),
    FOREIGN KEY (Subject_id) REFERENCES Class_Registration.dbo.[Subject](id),
    FOREIGN KEY (Department_id) REFERENCES Class_Registration.dbo.Department(id),
    PRIMARY KEY (Semester_id, Subject_id)
);

--Relationship: Uses
CREATE TABLE Class_Registration.dbo.Uses(
    Semester_id varchar(10),
    Subject_id varchar(10),
    Class_id varchar(10),
```

Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

Học kỳ: **1**, Năm học: **2020-2021**

Giảng viên: **TS. Võ Thị Ngọc Châu**

```
        MainTeacher_ssn VARCHAR(10),
        ReferenceBook_id VARCHAR(10),
        FOREIGN KEY (MainTeacher_ssn) REFERENCES Class_Registration.dbo.MainTeacher(ssn),
        FOREIGN KEY (ReferenceBook_id) REFERENCES
Class_Registration.dbo.ReferenceBook(id),
        FOREIGN KEY (Semester_id, Subject_id, Class_id) REFERENCES
Class_Registration.dbo.Class(Semester_id, Subject_id, id),
        PRIMARY KEY (Semester_id, Class_id, Subject_id, ReferenceBook_id)
);

--Relationship: Write
CREATE TABLE Class_Registration.dbo.Write(
    Author_ssn VARCHAR(10),
    Book_id VARCHAR(10),
    FOREIGN KEY (Author_ssn) REFERENCES Class_Registration.dbo.Author(ssn),
    FOREIGN KEY (Book_id) REFERENCES Class_Registration.dbo.ReferenceBook(id),
    PRIMARY KEY (Author_ssn, Book_id)
);

--Multi-value attribute: Field
CREATE TABLE Class_Registration.dbo.Field(
    Book_id VARCHAR(10),
    Field VARCHAR(50),
    FOREIGN KEY (Book_id) REFERENCES Class_Registration.dbo.ReferenceBook(id),
    PRIMARY KEY (Book_id, Field)
);

--Relationship: Belong to (Reference Book <-> Publisher)
ALTER TABLE Class_Registration.dbo.ReferenceBook
ADD
    Publisher_id VARCHAR(10) NOT NULL,
    FOREIGN KEY (Publisher_id) REFERENCES Class_Registration.dbo.Publisher(id) ON
DELETE CASCADE

--Relationship: Register
CREATE TABLE Class_Registration.dbo.Register(
    Student_id VARCHAR(10),
    Class_id VARCHAR(10),
    Semester_id VARCHAR(10),
    Subject_id VARCHAR(10),
    FOREIGN KEY (Student_id) REFERENCES Class_Registration.dbo.Student(ssn),
    FOREIGN KEY (Semester_id, Subject_id, Class_id) REFERENCES
Class_Registration.dbo.Class(Semester_id, Subject_id, id),
    PRIMARY KEY (Semester_id, Subject_id, Class_id, Student_id)
);

--Relationship: main responsible
CREATE TABLE Class_Registration.dbo.MainResponsible(
    Semester_id varchar(10),
    Subject_id varchar(10),
    MainTeacher_ssn varchar(10),
    FOREIGN KEY (MainTeacher_ssn) REFERENCES Class_Registration.dbo.MainTeacher(ssn),
    FOREIGN KEY (Semester_id) REFERENCES Class_Registration.dbo.Semester(id),
    FOREIGN KEY (Subject_id) REFERENCES Class_Registration.dbo.Subject(id),
    PRIMARY KEY (Semester_id, Subject_id)
);
```

Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

Học kỳ: **1**, Năm học: **2020-2021**

Giảng viên: **TS. Võ Thị Ngọc Châu**

```
--Weak Entity: Week
CREATE TABLE Class_Registration.dbo.[Week](
    id int,
    Semester_id varchar(10),
    startDate date,
    FOREIGN KEY (Semester_id) REFERENCES Class_Registration.dbo.Semester(id),
    PRIMARY KEY (id, Semester_id)
);

--Relationship: Responsible
CREATE TABLE Class_Registration.dbo.Responsible(
    Semester_id varchar(10),
    Subject_id varchar(10),
    Class_id varchar(10),
    Week_id int,
    Week_Semester_id varchar(10),
    CHECK(Week_Semester_id = Semester_id),
    Teacher_ssn varchar(10),
    FOREIGN KEY (Semester_id, Subject_id, Class_id) REFERENCES
Class_Registration.dbo.Class(Semester_id, Subject_id, id),
    FOREIGN KEY (Week_id, Week_Semester_id) REFERENCES
Class_Registration.dbo.[Week](id, Semester_id),
    FOREIGN KEY (Teacher_ssn) REFERENCES Class_Registration.dbo.Teacher(ssn),
    PRIMARY KEY (Semester_id, Subject_id, Class_id, Week_id)
);
```

#### A8. Phân chia nhóm người dùng và phân quyền

- Hệ thống người dùng bao gồm nhân viên phòng đào tạo, nhân viên văn phòng khoa, giáo viên và sinh viên.

```
USE Class_Registration

--Create student account
CREATE LOGIN student WITH PASSWORD = '1'

--Create study office employee account
CREATE LOGIN soemployee WITH PASSWORD = '1'

--Create department employee account
CREATE LOGIN deemployee WITH PASSWORD = '1'

--Create teacher account
CREATE LOGIN teacher WITH PASSWORD = '1'

--Create student user
CREATE USER student FOR LOGIN student

--Create study office employee user
CREATE USER soemployee FOR LOGIN soemployee
```

Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

Học kỳ: **1**, Năm học: **2020-2021**

Giảng viên: **TS. Võ Thị Ngọc Châu**

```
--Create department employee user  
CREATE USER deemployee FOR LOGIN deemployee
```

```
--Create teacher user  
CREATE USER teacher FOR LOGIN teacher
```

```
--Create so role  
CREATE ROLE students
```

```
--Create students role  
CREATE ROLE soemployees
```

```
--Create de role  
CREATE ROLE deemployees
```

```
--Create teacher role  
CREATE ROLE teachers
```

```
--Add student  
ALTER ROLE students  
    ADD MEMBER student;
```

```
--Add so  
ALTER ROLE soemployees  
    ADD MEMBER soemployee;
```

```
--Add de  
ALTER ROLE deemployees  
    ADD MEMBER deemployee;
```

```
--Add teacher  
ALTER ROLE teachers  
    ADD MEMBER teacher;
```

- Sinh viên chỉ được quyền chỉnh sửa đăng ký môn học

```
--students permission  
GRANT ALL ON OBJECT::Register TO students  
GRANT SELECT ON OBJECT::[Subject] TO students  
GRANT SELECT ON OBJECT::Class TO students  
GRANT SELECT ON OBJECT::Opens TO students  
GRANT SELECT ON OBJECT::Responsible TO students  
GRANT SELECT ON OBJECT::StudyStatus TO students  
GRANT SELECT ON OBJECT::Uses TO students  
GRANT SELECT ON OBJECT::ReferenceBook TO students  
GRANT SELECT ON OBJECT::Author TO students  
GRANT SELECT ON OBJECT::Field TO students  
GRANT SELECT ON OBJECT::Teacher TO students  
GRANT SELECT ON OBJECT::MainTeacher TO students  
GRANT SELECT ON OBJECT::Semester TO students
```

- Giảng viên được chỉnh sửa dữ liệu về giáo trình

```
--teachers permission  
GRANT ALL ON OBJECT::Uses TO teachers  
GRANT SELECT ON SCHEMA::dbo TO teachers
```

Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

Học kỳ: **1**, Năm học: **2020-2021**

Giảng viên: **TS. Võ Thị Ngọc Châu**

- Nhân viên văn phòng khoa chỉnh sửa môn học được mở, giáo viên phụ trách chính, giáo viên phụ trách từng tuần

```
--department employee permission
GRANT ALL ON OBJECT::Opens TO deemployees
GRANT ALL ON OBJECT::Responsible TO deemployees
GRANT ALL ON OBJECT::MainResponsible TO deemployees
GRANT SELECT ON SCHEMA::dbo TO deemployees
```

- Nhân viên phòng đào tạo chỉnh sửa đăng ký môn học, lớp học, tình trạng học tập của sinh viên, thông tin sinh viên.

```
--study office employee permission
GRANT ALL ON OBJECT::Register TO soemployees
GRANT ALL ON OBJECT::Class TO soemployees
GRANT ALL ON OBJECT::StudyStatus TO soemployees
GRANT ALL ON OBJECT::Student TO soemployees
GRANT SELECT ON SCHEMA::dbo TO soemployees
```

## A9. Các store procedure cho từng usecase

```
-- (i)
PDT
--
*****
*****

--(i.1). Cap nhat dang ki mon hoc cua cac lop.
-- Them
CREATE PROCEDURE addRegister(
    @newStudentId AS varchar(10),
    @newClassId AS varchar(10),
    @newSemesterId AS varchar(10),
    @newSubjectId AS varchar(10)
)
AS
BEGIN
    INSERT INTO Register
    VALUES
        (@newStudentId, @newClassId, @newSemesterId, @newSubjectId)
END;

-- Xoa
GO
CREATE PROCEDURE removeRegister(
    @oldStudentId AS varchar(10),
    @oldClassId AS varchar(10),
    @oldSemesterId AS varchar(10),
    @oldSubjectId AS varchar(10)
)
AS
BEGIN
```



Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

Học kỳ: **1**, Năm học: **2020-2021**

Giảng viên: **TS. Võ Thị Ngọc Châu**

```
DELETE FROM Register
WHERE
    Student_id = @oldStudentId
    AND Class_id = @oldClassId
    AND Semester_id = @oldSemesterId
    AND Subject_id = @oldSubjectId

END;

-- Sua
GO
CREATE PROCEDURE updateRegister(
    @oldStudentId AS varchar(10),
    @oldClassId AS varchar(10),
    @oldSemesterId AS varchar(10),
    @oldSubjectId AS varchar(10),

    @newStudentId AS varchar(10),
    @newClassId AS varchar(10),
    @newSemesterId AS varchar(10),
    @newSubjectId AS varchar(10)
)
AS
BEGIN
    UPDATE Register
    SET
        Student_id = @newStudentId,
        Class_id = @newClassId,
        Semester_id = @newSemesterId,
        Subject_id = @newSubjectId
    WHERE
        Student_id = @oldStudentId
        AND Class_id = @oldClassId
        AND Semester_id = @oldSemesterId
        AND Subject_id = @oldSubjectId
END;

GO
--(i.2). Xem danh sach lop da duoc dang ky boi mot sinh vien o mot hoc ky.
CREATE PROCEDURE registeredClass(
    @studentId AS varchar(10),
    @semesterId AS varchar(10)
)
AS
BEGIN
    SELECT Class_id Ma_lop_hoc, Subject_id Ma_mon_hoc
    FROM REGISTER
    WHERE Student_id = @studentId
        AND Semester_id = @semesterId
END;

GO
--(i.3). Xem danh sach lop duoc phu trach boi mot giang vien o mot hoc ky.
CREATE PROCEDURE responsibleClass(
    @teacherSsn AS varchar(10),
    @semesterId AS varchar(10)
)
```

Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

Học kỳ: **1**, Năm học: **2020-2021**

Giảng viên: **TS. Võ Thị Ngọc Châu**

```
)
AS
BEGIN
    SELECT DISTINCT Class_id Ma_lop_hoc, Subject_id Ma_mon_hoc
    FROM Responsible
    WHERE Teacher_ssn = @teacherSsn
        AND Semester_id = @semesterId;
END;

GO
--(i.4). Xem danh sách môn học được đăng ký ở mỗi học kỳ ở mỗi khoa.
CREATE PROCEDURE listClass(
    @departmentId AS varchar(10),
    @semesterId AS varchar(10)
)
AS
BEGIN
    SELECT Subject_id Ma_mon_hoc
    FROM Opens
    WHERE Department_id = @departmentId AND Semester_id = @semesterId
END;

GO
--(i.5). Xem danh sách sinh viên đang ký ở mỗi lớp ở mỗi học kỳ ở mỗi khoa.
CREATE PROCEDURE listStudent
AS
BEGIN
    SELECT Department_id Ma_khoa, c.Semester_id Ma_hoc_ky,
        s.ssn, firstName Ten, lastName Ho

    FROM Student s, Register, Class c, Opens o
    WHERE s.ssn = Student_id
        AND Class_id = c.id
        AND c.Semester_id = o.Semester_id
        AND c.Subject_id = o.Subject_id
    GROUP BY Department_id, c.Semester_id, s.ssn, firstName, lastName
END;

GO
--(i.6). Xem danh sách giảng viên phụ trách ở mỗi lớp ở mỗi học kỳ ở mỗi khoa.
CREATE PROCEDURE listTeacher
AS
BEGIN
    SELECT Department_id Ma_Khoa, c.Semester_id Ma_hoc_ky, Class_id Ma_lop,
        Teacher_ssn SSN, firstName Ten, lastName Ho

    FROM Responsible, Employee, Class c, Opens o
    WHERE Teacher_ssn = ssn
        AND Class_id = c.id
        AND c.Subject_id = o.Subject_id
    GROUP BY Department_id, c.Semester_id, Class_id, Teacher_ssn, firstName, lastName
END;

GO
--(i.7). Xem các giao trình được chỉ định cho mỗi môn học ở mỗi học kỳ ở mỗi khoa
CREATE PROCEDURE listReferenceBook(
```

Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

Học kỳ: **1**, Năm học: **2020-2021**

Giảng viên: **TS. Võ Thị Ngọc Châu**

```
        @semesterId AS varchar(10),
        @subjectId AS varchar(10)
    )
AS
BEGIN
    SELECT ReferenceBook_id Ma_sach
    FROM Uses
    WHERE Semester_id = @semesterId AND Subject_id = @subjectId
END;

GO
--(i.8). Xem tong so mon hoc duoc dang ky o moi hoc ky o moi khoa.
CREATE PROCEDURE numOfSubjects
AS
BEGIN
    SELECT Department_id Ma_khoa, Semester_id Ma_hoc_ky,
           COUNT(Subject_id) So_mon_hoc
    FROM Opens o
    GROUP BY Department_id , Semester_id
END;

GO
--(i.9). Xem tong so lop duoc mo o moi hoc ky o moi khoa.
CREATE PROCEDURE numOfClassess
AS
BEGIN
    SELECT Department_id Ma_khoa, o.Semester_id Ma_hoc_ky,
           COUNT(c.id) So_lop_hoc
    FROM Opens o, Class c
    WHERE o.Subject_id = c.Subject_id
    GROUP BY Department_id , o.Semester_id
END;

GO
--(i.10). Xem tong so sinh vien dang ky o moi lop cua mot mon hoc o mot hoc ky.
CREATE PROCEDURE numOfStudents_class_sem
AS
BEGIN
    SELECT c.Semester_id Ma_hoc_ky, c.id Ma_lop_hoc,
           COUNT(r.Student_id) Tong_sinh_vien
    FROM Register r, Class c
    WHERE r.Class_id = c.id
    GROUP BY c.Semester_id, c.id
END;

GO
--(i.11). Xem so sinh vien dang ky o moi mon hoc o mot hoc ky.
CREATE PROCEDURE numOfStudents_sub_sem
AS
BEGIN
    SELECT c.Semester_id Ma_hoc_ky, c.Subject_id Ma_mon_hoc,
           COUNT(r.Student_id) Tong_sinh_vien
    FROM Register r, Class c
    WHERE r.Class_id = c.id
    GROUP BY c.Semester_id, c.Subject_id
END;
```

Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

Học kỳ: **1**, Năm học: **2020-2021**

Giảng viên: **TS. Võ Thị Ngọc Châu**

```
GO
--(i.12). Xem tong so sinh vien dang ky o moi mon hoc o moi khoa.
CREATE PROCEDURE numOfStudents_sub_dep
AS
BEGIN
    SELECT o.Department_id Ma_khoa, c.Subject_id Ma_mon_hoc,
           COUNT(r.Student_id) Tong_sinh_vien
    FROM Register r, Class c, Opens o
    WHERE r.Class_id = c.id
           AND c.Subject_id = o.Subject_id
    GROUP BY o.Department_id, c.Subject_id
END;
--
Khoa
--
*****
*****

--ii.1: Cap nhat danh sach mon hoc duoc mo truoc dau moi hoc ky.
GO
CREATE PROCEDURE UpdateSubject(
    @semesterId AS varchar(10),
    @departmentId AS VARCHAR(10),
    @subjectId AS VARCHAR(10)
)
AS
BEGIN
    INSERT INTO dbo.Opens VALUES(@semesterId, @subjectId, @departmentId)
END;
--ii.2: Cap nhat danh sach giang vien phu trach moi lop hoc duoc mo truoc dau moi hoc ky.
GO

CREATE PROCEDURE UpdateTeacherOfClass(
    @teacherSsn AS varchar(10),
    @classId AS VARCHAR(10),
    @subjectId AS VARCHAR(10),
    @semesterId AS VARCHAR(10),
    @weekId AS VARCHAR(10)
)
AS
BEGIN
    INSERT INTO dbo.Responsible VALUES(@semesterId, @subjectId, @classId, @weekId,
    @semesterId, @teacherSsn)
END;
--ii.3: Xem danh sach mon hoc o mot hoc ky.
GO
CREATE PROCEDURE SubjectOnSemester(
    @semesterId AS varchar(10),
    @departmentId AS VARCHAR(10)
)
AS
BEGIN
    SELECT Id Ma_mon
    FROM dbo.Subject JOIN dbo.Opens ON Opens.Subject_id = Subject.id
    WHERE dbo.Opens.Semester_id = @semesterId
```

Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

Học kỳ: **1**, Năm học: **2020-2021**

Giảng viên: **TS. Võ Thị Ngọc Châu**

END;

GO

--ii.4: Xem danh sách giảng viên ở một học kỳ.

```
CREATE PROCEDURE TeacherOnSemester(  
    @semesterId AS VARCHAR(10),  
    @departmentId AS VARCHAR(10)  
)  
AS  
BEGIN  
    SELECT DISTINCT Teacher_ssn Ma_giang_vien  
    FROM dbo.Responsible JOIN dbo.Teacher ON Teacher.ssn = Responsible.Teacher_ssn  
    WHERE Semester_id = @semesterId AND dId = @departmentId  
END;
```

GO

--ii.5: Xem danh sách lớp được phụ trách bởi một giảng viên ở một học kỳ.

```
CREATE PROCEDURE ClassOfTeacher(  
    @teacherSsn AS VARCHAR(10),  
    @semesterId AS VARCHAR(10)  
)  
AS  
BEGIN  
    SELECT DISTINCT Class_id Ma_lop, Semester_id Ma_hoc_ky, Subject_id Ma_mon  
    FROM dbo.Responsible  
    WHERE Semester_id = @semesterId AND Teacher_ssn = @teacherSsn  
END;
```

GO

--ii.6: Xem danh sách giảng viên phụ trách ở mọi lớp ở một học kỳ.

```
CREATE PROCEDURE TeacherOfClass(  
    @semesterId AS VARCHAR(10),  
    @departmentId AS VARCHAR(10)  
)  
AS  
BEGIN  
    SELECT DISTINCT Class_id Ma_lop, dbo.Responsible.Semester_id Ma_hoc_ky,  
    dbo.Responsible.Subject_id Ma_mon, Teacher_ssn Ma_giang_vien  
    FROM dbo.Responsible JOIN dbo.Opens ON Opens.Semester_id = Responsible.Semester_id  
    AND Opens.Subject_id = Responsible.Subject_id  
    WHERE dbo.Responsible.Semester_id = @semesterId AND Department_id = @departmentId  
END;
```

GO

--ii.7: Xem các giao trình được chỉ định cho mọi môn học ở một học kỳ.

```
CREATE PROCEDURE BookOfSubject(  
    @semesterId AS VARCHAR(10),  
    @departmentId AS VARCHAR(10)  
)  
AS  
BEGIN  
    SELECT DISTINCT Subject_id Ma_mon, ReferenceBook_id Ma_giao_trinh  
    FROM dbo.Uses  
    WHERE Subject_id IN (SELECT Subject_id FROM dbo.Opens WHERE Semester_id =  
    @semesterId AND Department_id = @departmentId)  
END;
```

Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

Học kỳ: **1**, Năm học: **2020-2021**

Giảng viên: **TS. Võ Thị Ngọc Châu**

```
GO
--ii.8: Xem danh sách sinh viên đang ký cho mỗi lớp ở một học kỳ.
CREATE PROCEDURE StudentOfClass(
    @semesterId AS VARCHAR(10),
    @departmentId AS VARCHAR(10)
)
AS
BEGIN
    SELECT Class_id Ma_lop, dbo.Register.Subject_id Ma_mon, Student_id Ma_sinh_vien
    FROM dbo.Register JOIN dbo.Opens ON Opens.Semester_id = Register.Semester_id AND
    Opens.Subject_id = Register.Subject_id
    WHERE dbo.Register.Semester_id = @semesterId AND Department_id = @departmentId
END;

GO
--ii.9: Xem tổng số sinh viên đang ký ở một học kỳ.
CREATE PROCEDURE NumStudentOfSemester(
    @semesterId AS VARCHAR(10),
    @departmentId AS VARCHAR(10)
)
AS
BEGIN
    SELECT COUNT(DISTINCT Student_id) Tong_sinh_vien
    FROM dbo.Register JOIN dbo.Student ON Student.ssn = Register.Student_id
    WHERE Semester_id = @semesterId AND dId = @departmentId
END;

GO
--ii.10: Xem tổng số lớp được mở ở một học kỳ.
CREATE PROCEDURE NumClassOfSemester(
    @semesterId AS VARCHAR(10),
    @departmentId AS VARCHAR(10)
)
AS
BEGIN
    SELECT COUNT(*) Tong_lop
    FROM dbo.Class JOIN dbo.Opens ON Opens.Semester_id = Class.Semester_id AND
    Opens.Subject_id = Class.Subject_id
    WHERE dbo.Class.Semester_id = @semesterId AND Department_id = @departmentId
END;

--ii.11: Xem những môn có nhiều giảng viên cùng phụ trách nhất ở một học kỳ.
GO
CREATE PROCEDURE SubjectHavingMaxTeacher(
    @semesterId AS VARCHAR(10),
    @departmentId AS VARCHAR(10)
)
AS
BEGIN
    SELECT Subject_id
    FROM (SELECT dbo.Responsible.Subject_id, COUNT(DISTINCT Teacher_ssn) AS Num_Of_Tea
    FROM dbo.Responsible JOIN dbo.Opens ON Opens.Subject_id =
    Responsible.Subject_id AND Opens.Semester_id = Responsible.Semester_id
    WHERE dbo.Responsible.Semester_id = @semesterId AND Department_id =
    @departmentId
```

Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

Học kỳ: **1**, Năm học: **2020-2021**

Giảng viên: **TS. Võ Thị Ngọc Châu**

```
GROUP BY dbo.Responsible.Subject_id) a
WHERE a.Num_Of_Tea = (SELECT MAX(Num_Of_Tea)
FROM (SELECT dbo.Responsible.Subject_id,
COUNT(DISTINCT Teacher_ssn) AS Num_Of_Tea
FROM dbo.Responsible JOIN
dbo.Opens ON Opens.Subject_id = Responsible.Subject_id AND Opens.Semester_id =
Responsible.Semester_id
WHERE
dbo.Responsible.Semester_id = @semesterId AND Department_id = @departmentId
GROUP BY
dbo.Responsible.Subject_id) b)
END;

GO
--ii.12: Xem số sinh viên đang ký trung bình trong 3 năm gần nhất cho một môn học ở một
học kỳ.
CREATE PROCEDURE AvgNumStudent(
    @subjectId AS VARCHAR(10)
)
AS
BEGIN
    SELECT AVG(a.So_sinh_vien)
    FROM (SELECT 100 Semester_id, dbo.Semester.startDate Ngay_bat_dau, COUNT(DISTINCT
Student_id) So_sinh_vien
FROM dbo.Register JOIN dbo.Semester ON Semester_id = dbo.Semester.id
WHERE Subject_id = @subjectId AND DATEDIFF(YEAR, startDate,
GETDATE()) <= 3
GROUP BY Semester_id, startDate) a
END;

-- (iii)
Giang vien
--
*****
*****
-- (iii.1). Cập nhật giao trình chính cho môn học do mình phụ trách.
GO
CREATE PROCEDURE UpdateReferenceBook(
    @teacherSsn AS varchar(10),
    @subjectId AS varchar(10),
    @bookId AS varchar(10),
    @semesterId AS varchar(10),
    @classId AS varchar(10)
)
AS
BEGIN
    IF (SELECT COUNT(*) FROM Uses WHERE Subject_id = @subjectId) < 3
        INSERT INTO Uses
        (Subject_id, Semester_id, Class_id, ReferenceBook_id, MainTeacher_ssn)
        VALUES (@subjectId, @semesterId, @classId, @bookId, @teacherSsn)
    ELSE
        RAISERROR('This subject can not have more than 3 reference books', -1, -1);
END;

-- (iii.2). Xem danh sách lớp học của mỗi môn học do mình phụ trách ở một học kỳ.
GO
CREATE PROCEDURE responsibleClasses
```

Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

Học kỳ: **1**, Năm học: **2020-2021**

Giảng viên: **TS. Võ Thị Ngọc Châu**

```
        (@teacherSsn AS varchar(10),
        @semesterId AS varchar(10)
    )
AS
BEGIN
    SELECT Subject_id Ma_mon_hoc, Class_id Ma_lop_hoc
    FROM Responsible
    WHERE Teacher_ssn = @teacherSsn AND Semester_id = @semesterId
    GROUP BY Subject_id, Class_id
END;

-- (iii.3). Xem danh sach sinh vien cua moi lop hoc do minh phu trach o mot hoc ky.
GO
CREATE PROCEDURE studentOfResopnsibleClass
    (@teacherSsn AS varchar(10),
    @semesterId AS varchar(10)
    )
AS
BEGIN
    SELECT rp.Class_id Ma_lop_hoc, Student_id Ma_sinh_vien, firstName Ten, lastName Ho
    FROM Responsible rp, Register rg, Student st
    WHERE rp.Class_id = rg.Class_id
        AND rg.Student_id = st.ssn
        AND Teacher_ssn = @teacherSsn
        AND rp.Semester_id = @semesterId
    GROUP BY rp.Class_id, Student_id, firstName, lastName
END;

-- (iii.4). Xem danh sach mon hoc va giao trinh chinh cho moi mon hoc do minh phu trach o
mot hoc ky.
GO
CREATE PROCEDURE referenceBookOfResponsibleSubject
    (@teacherSsn AS varchar(10),
    @semesterId AS varchar(10)
    )
AS
BEGIN
    SELECT rp.Class_id Ma_lop_hoc, Student_id Ma_sinh_vien, firstName Ten, lastName Ho
    FROM Responsible rp, Register rg, Student st
    WHERE rp.Class_id = rg.Class_id
        AND rg.Student_id = st.ssn
        AND Teacher_ssn = @teacherSsn
        AND rp.Semester_id = @semesterId
    GROUP BY rp.Class_id, Student_id, firstName, lastName
END;

-- (iii.5). Xem tong so sinh vien cua moi lop hoc do minh phu trach o mot hoc ky.
GO
CREATE PROCEDURE numOfStudents_ofResponsiblesClass
    (@teacherSsn AS varchar(10),
    @semesterId AS varchar(10)
    )
AS
BEGIN
    SELECT rp.Class_id Ma_lop_hoc, COUNT(*) tong_so_sinh_vien
    FROM Responsible rp, Register rg
```



Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

Học kỳ: **1**, Năm học: **2020-2021**

Giảng viên: **TS. Võ Thị Ngọc Châu**

```
WHERE rp.Class_id = rg.Class_id
      AND Teacher_ssn = @teacherSsn
      AND rp.Semester_id = @semesterId
GROUP BY rp.Class_id
END;

-- (iii.6). Xem so lop hoc do minh phu trach o moi hoc ky trong 3 nam lien tiep gan day
nhat.
GO
CREATE PROCEDURE numOfResponsibleClass_3RecentYear
(@teacherSsn AS varchar(10))
AS
BEGIN
SELECT s.id Ma_hoc_ky, COUNT(*) Tong_so_lop
FROM Responsible r, Semester s
WHERE r.Semester_id = s.id
      AND (DATEDIFF(YEAR, s.startDate, GETDATE())) <= 3
      AND Teacher_ssn = @teacherSsn
GROUP BY s.id
END;

-- (iii.7). Xem 5 lop hoc co so sinh vien cao nhat ma giang vien tung phu trach.
GO
CREATE PROCEDURE top5Class_mostStudent
(@teacherSsn AS varchar(10))
AS
BEGIN
SELECT TOP(5) rp.Class_id Ma_lop_hoc--, COUNT(*) tong_so_sinh_vien
FROM Responsible rp, Register rg
WHERE rp.Class_id = rg.Class_id
      AND Teacher_ssn = @teacherSsn
GROUP BY rp.Class_id
ORDER BY COUNT(*) DESC
END;

-- (iii.8). Xem 5 hoc ky co so lop nhieu nhat ma giang vien tung phu trach.
GO
CREATE PROCEDURE top5Semester_mostClass
(@teacherSsn AS varchar(10))
AS
BEGIN
SELECT TOP(5) Semester_id Ma_hoc_ky--, COUNT(*) tong_so_sinh_vien
FROM Responsible
WHERE Teacher_ssn = @teacherSsn
GROUP BY Semester_id
ORDER BY COUNT(*) DESC
END;

--
Sinh vien
--
*****
*****

--iv.1: Dang ky mon hoc o hoc ky duoc dang ky.
GO
CREATE PROCEDURE RegisterSubject(
@studentId AS VARCHAR(10),
```

Giảng viên: **TS. Võ Thị Ngọc Châu**

END;

Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

Học kỳ: **1**, Năm học: **2020-2021**

Giảng viên: **TS. Võ Thị Ngọc Châu**

--iv.5: Xem danh sách lớp học của mỗi môn học mà mình đang ký có nhiều hơn 1 giảng viên phụ trách ở một học kỳ.

GO

```
CREATE PROCEDURE ClassOfSubjectMoreThan1Teacher(
    @studentId AS VARCHAR(10),
    @semesterId AS VARCHAR(10)
)
AS
BEGIN
    SELECT Subject_id Ma_mon, Class_id Ma_lop
    FROM dbo.Responsible
    WHERE Subject_id IN (SELECT DISTINCT Register.Subject_id Ma_mon
                        FROM dbo.Register
                        WHERE Student_id = @studentId AND
                        Semester_id = @semesterId)
    GROUP BY Subject_id, Class_id
    HAVING COUNT(DISTINCT Teacher_ssn) > 1
END;
```

--iv.6: Xem tổng số tín chỉ đã đăng ký được ở một học kỳ.

GO

```
CREATE PROCEDURE SumCredit(
    @studentId AS VARCHAR(10),
    @semesterId AS VARCHAR(10)
)
AS
BEGIN
    SELECT SUM(DISTINCT credit)
    FROM dbo.Register JOIN dbo.Subject ON Subject_id = id
    WHERE Student_id = @studentId AND Semester_id = @semesterId
END;
```

--iv.7: Xem tổng số môn học đã đăng ký được ở một học kỳ.

GO

```
CREATE PROCEDURE SumSubject(
    @studentId AS VARCHAR(10),
    @semesterId AS VARCHAR(10)
)
AS
BEGIN
    SELECT COUNT(DISTINCT Subject_id)
    FROM dbo.Register
    WHERE Student_id = @studentId AND Semester_id = @semesterId
END;
```

--iv.8: Xem 3 học kỳ có số tổng số tín chỉ cao nhất mà mình đã từng đăng ký.

GO

```
CREATE PROCEDURE First3MaxCredit(
    @studentId AS VARCHAR(10)
)
AS
BEGIN
    SELECT TOP 3 Semester_id, SUM(DISTINCT credit)
    FROM dbo.Register JOIN dbo.Subject ON Subject_id = id
    WHERE Student_id = @studentId
    GROUP BY Semester_id
```

Môn: **Hệ Cơ Sở Dữ Liệu (CO2013)**

Học kỳ: **1**, Năm học: **2020-2021**

Giảng viên: **TS. Võ Thị Ngọc Châu**

```
ORDER BY SUM(DISTINCT credit)  
END;
```