ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH

DATABASE SYSTEM

Báo Cáo Assignment 1

Đề Tài 1: TEACHING & LEARNING

GVHD: Phạm Nguyễn Hoàng Nam

SVTH: Bùi Xuân Thông - 1915350

Hoàng Văn Hiếu - 1913328

Nguyễn Minh Phú - 1914659

Tp. Hồ Chí Minh, Tháng 10/2021

Mục lục

1. Overview

Bài tập lớn này sẽ thảo luận về việc thiết kế một hệ thống cơ sở dữ liệu quản lý các lớp học trong hệ thống giáo dục theo quy chế tín chỉ.

2. Requirements Analysis

2.1. Requirement

- Sinh viên (STUDENT) có trạng thái học tập bình thường (ACTIVE STATUS) có thể đăng ký một hoặc nhiều môn học, những sinh viên có trạng thái tạm dừng (PAUSED STATUS) hoặc buộc thôi học (DEPORTED STATUS) sẽ không thể đăng ký bất kỳ môn học nào ở học kỳ đó.

- Một môn học (SUBJECT) được mở và triển khai gồm một hoặc nhiều lớp tùy vào số lượng sinh viên đăng ký ở mỗi học kỳ.

- Mỗi lớp (CLASS) do một hoặc nhiều giảng viên (LECTURER) phụ trách ở các tuần học khác nhau, trong đó có một giảng viên phụ trách chính sẽ ghi điểm cho các sinh viên vào cuối học kỳ.

- Một hoặc nhiều giáo trình (TEXTBOOK) sẽ được sử dụng cho mỗi môn học.

- Mỗi giáo trình có một nhà xuất bản (PUBLISHER), có thể là nhà xuất bản trong nước hoặc ngoài nước, nhiều loại giáo trình có thể được mua chung từ một nhà xuất bản. Nhiều giáo trình thuộc về cùng một lĩnh vực (Specialization) có thể được mua từ các nhà xuất bản khác nhau.

- Mỗi giáo trình được biên soạn bởi một hay nhiều tác giả (AUTHOR) và một tác giả cũng có thể biên soạn một hoặc nhiều giáo trình.

- Mỗi giảng viên hay mỗi sinh viên chỉ thuộc về một khoa nhất định (FACULTY).

- Mỗi khoa quản lý danh sách các môn học thuộc chương trình đào tạo của khoa đó và giảng viên quản lý các môn học và các giáo trình cho môn học đó.

2.2. Constraint

- Mỗi môn học tối đa 3 tín chỉ.

- Mỗi sinh viên chỉ được đăng ký tối đa 18 tín chỉ cho một học kỳ.

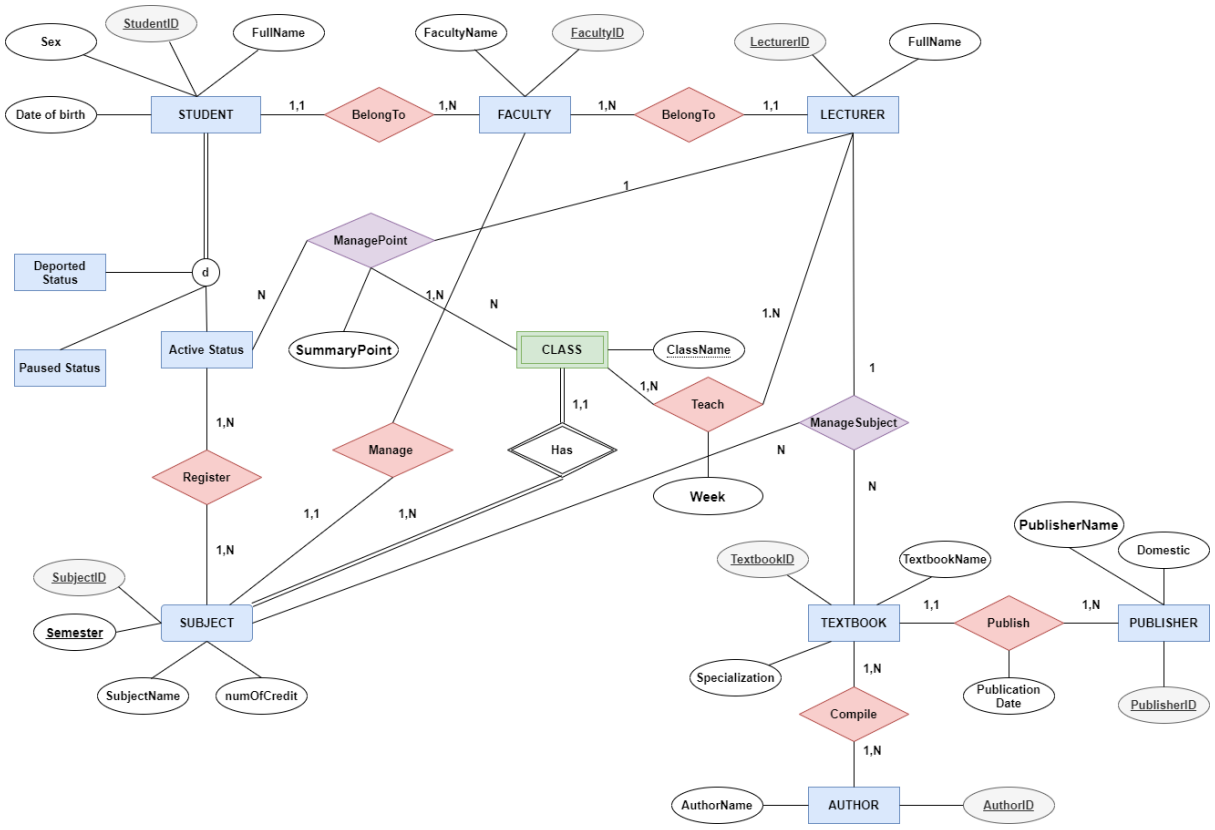
- Mỗi lớp học có tối đa 60 sinh viên đăng ký.

- Mỗi môn học có tối đa 3 giáo trình chính do giảng viên phụ trách chính chỉ định.

- Mỗi giáo trình chính có thời gian kể từ lúc xuất bản không được quá 10 năm.

3. Conceptual Design

3.1. Entity Relationship Diagram



Entity-Relationship Diagram (ERD)

3.2. Entity Table

Kiểu thực thể	Mô tả
STUDENT	Lưu trữ thông tin của tất cả sinh viên, được chia làm 3 loại dựa theo theo trạng thái học tập của sinh viên đó: bình thường (Active), ngừng học (Paused) và buộc thôi học (Deported).
LECTURER	Lưu trữ thông tin của tất cả giảng viên trong trường.

FACULTY	Lưu trữ thông tin các khoa của trường. Các khoa có vai trò quản lý các sinh viên và giảng viên thuộc về khoa đó.
SUBJECT	Lưu trữ thông tin tất cả các môn học thuộc chương trình đào tạo của các khoa.
TEXTBOOK	Lưu trữ thông tin tất cả giáo trình mà các môn học sử dụng.
PUBLISHER	Lưu trữ thông tin các nhà xuất bản của các giáo trình.
AUTHOR	Lưu trữ thông tin của các tác giả đã biên soạn giáo trình
CLASS (WEAK ENTITY)	Lưu trữ thông tin các lớp được mở trong một học kỳ của tất cả các môn học.

3.3. Relationship Table

Quan hệ	Mô tả
BelongTo	Thể hiện quan hệ giữa : - Khoa và sinh viên (FACULTY – STUDENT) - Khoa và giảng viên (FACULTY- LECTURER) Thông qua quan hệ này có thể xác định sinh viên và giảng viên thuộc về khoa nào hay khoa đó có những sinh viên hay giảng viên nào.
Register	Thể hiện quan hệ giữa sinh viên có trạng thái học tập bình thường và môn học (ACTIVE STATUS – SUBJECT). Thông qua quan hệ này có thể xác định sinh viên đăng ký những môn học nào.
Manage	Thể hiện quan hệ giữa khoa và môn học (FACULTY – SUBJECT) Thông qua đó, các khoa có thể quản lý các môn học thuộc chương trình đào tạo của mình.

Teach	Thể hiện quan hệ giữa giảng viên và lớp (LECTURER – CLASS) Thông qua đó, có thể xác định được các giảng viên dạy cho từng lớp.
Compile	Thể hiện quan hệ giữa tác giả và giáo trình (AUTHOR – TEXTBOOK) Thông qua đó, có thể xác định được người đã biên soạn cho giáo trình mà chúng ta sử dụng.
Publish	Thể hiện quan hệ giữa nhà xuất bản và giáo trình (PUBLISHER – TEXTBOOK) Thông qua đó, có thể xác định được nhà xuất bản của các giáo trình.
ManagePoint	Thể hiện quan hệ giữa ba kiểu thực thể: giảng viên chính, sinh viên và lớp học (LECTURER – STUDENT – CLASS). Thông qua đó, giảng viên chính của các lớp có thể ghi điểm cho sinh viên của mình vào cuối học kỳ.
ManageSubject	Thể hiện quan hệ giữa ba kiểu thực thể : giảng viên, môn học và giáo trình (LECTURER – SUBJECT – TEXTBOOK) Thông qua đó, giảng viên chỉ định những giáo trình nào được sử dụng cho môn học.
Has	Thể hiện quan hệ phụ thuộc của kiểu thực thể yếu lớp học (CLASS) phụ thuộc vào kiểu thực thể mạnh là môn học (SUBJECTS). Thông qua đó có thể xác định được lớp học được mở cho môn học nào.

3.4. Attribute table

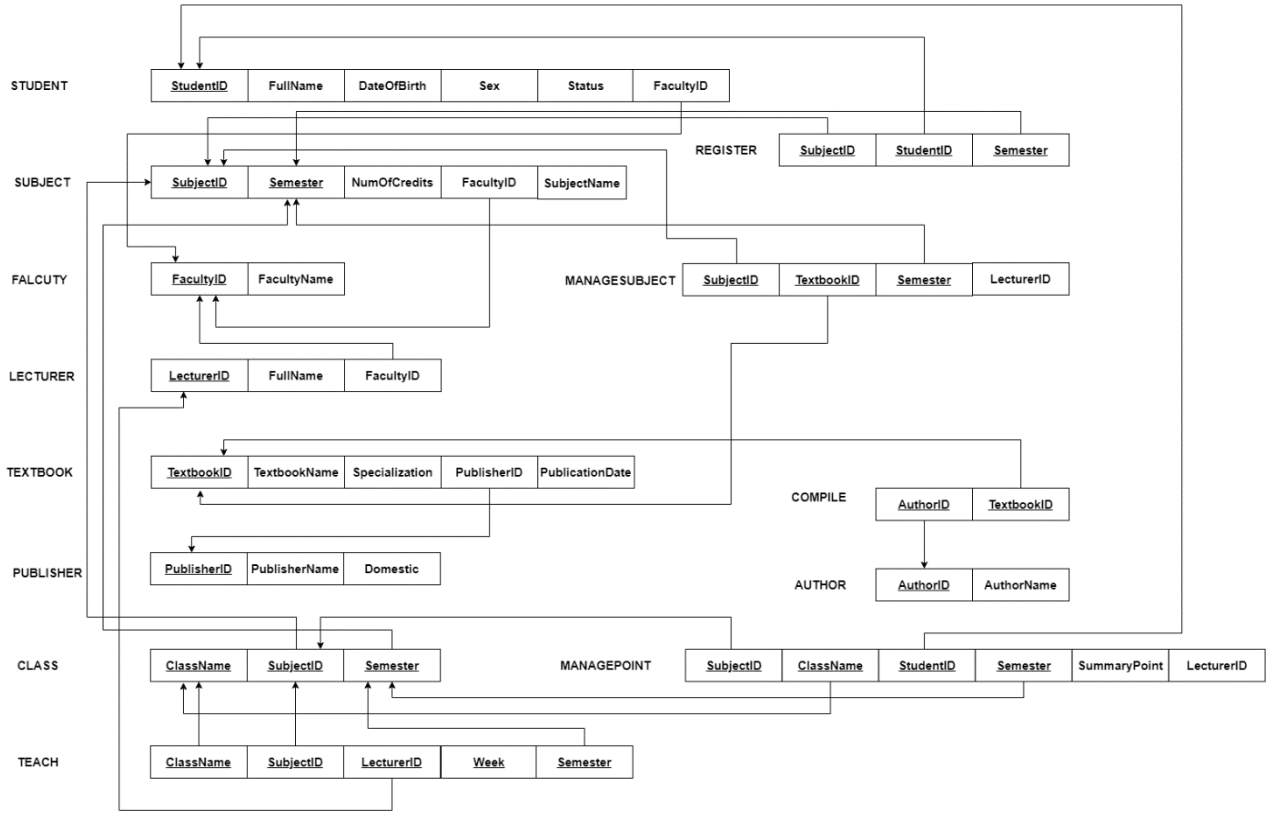
Thực thể	Thuộc tính	Khóa chính

(Entity)	(Attribute)	(Primary Key)
STUDENT	- StudentID: mã số sinh viên - FullName: họ và tên - DateOfBirth: ngày sinh - Sex: giới tính	StudentID
LECTURER	- LecturerID: mã số giảng viên - FullName: họ và tên	LecturerID
FACULTY	- FacultyID: mã khoa - FacultyName: tên của khoa	FacultyID
SUBJECT	- SubjectID: mã môn học - SubjectName: tên môn học - NumberOfCredit: số tín chỉ	SubjectID
AUTHOR	- AuthorID: mã số tác giả - AuthorName: tên tác giả	AuthorID
TEXTBOOK	- TextbookID: mã giáo trình - TextbookName: tên giáo trình - Specialization: lĩnh vực, chuyên ngành	TextbookID
PUBLISHER	- PublisherID: mã nhà xuất bản - PublisherName: tên nhà xuất bản - Domestic: trong nước	PublisherID
CLASS (WEAK ENTITY)	- ClassName: tên lớp (khóa riêng phần phụ thuộc vào kiểu thực thể mạnh Subject)	ClassName; SubjectID

4. Logic Design

4.1.Relational Schema

Sau khi ánh xạ:

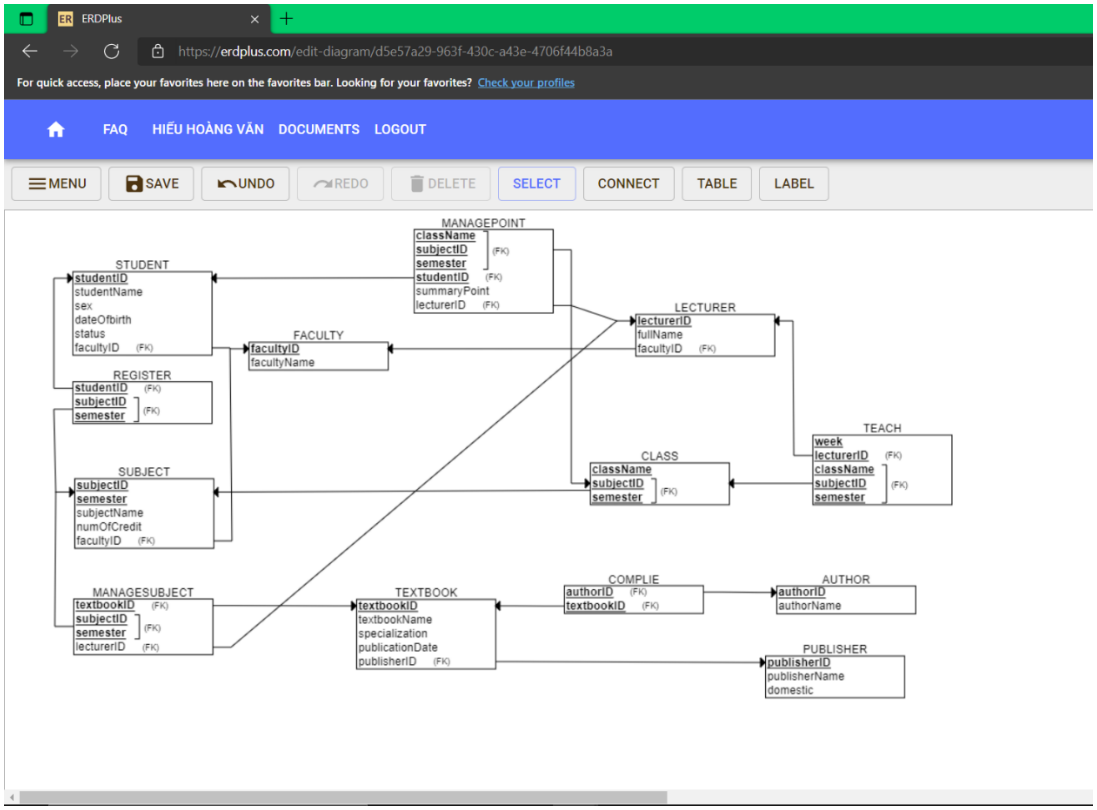


Relational Schema

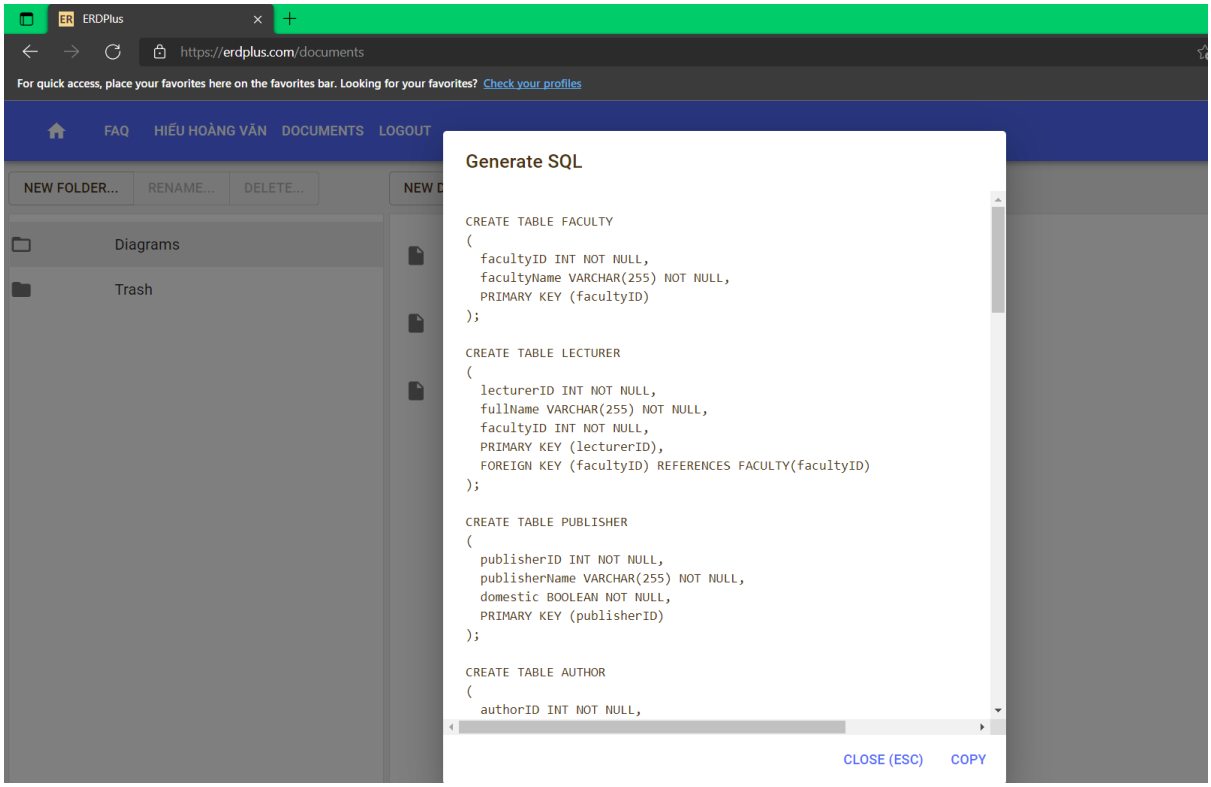
4.2.Sử dụng công cụ để thiết kế Relational schema

Sau khi tìm hiểu các công cụ dùng để thiết kế sơ đồ quan hệ của các cơ sở dữ liệu như StartUML, Vertabelo, ERWin, RationalRose… nhóm quyết định chọn công cụ ERDplus để thiết kế.

4.2.1.Relational schema



4.2.2.Code sql được tạo ra từ công cụ ERDplus



5.Normalization

Chuẩn hoá là quá trình tách bảng thành các bảng nhỏ hơn dựa vào các phụ thuộc hàm. Các dạng chuẩn là các chỉ dẫn để thiết kế các bảng trong CSDL.

Mục đích của chuẩn hoá là loại bỏ các dư thừa dữ liệu và các lỗi khi thao tác dư thừa và các lỗi khi thao tác dữ liệu (Insert, Delete, Update). Nhưng chuẩn hoá làm tăng thời gian truy vấn.

5.1.Dạng chuẩn 1NF

Một bảng (quan hệ) được gọi là ở dạng chuẩn 1NF nếu và chỉ nếu toàn bộ các miền giá trị của các cột có mặt trong bảng (quan hệ) đều chỉ chứa các giá trị nguyên tử (nguyên tố), nghĩa là mọi thuộc tính trong bảng quan hệ đó đều là thuộc tính đơn trị và không có bất kỳ thuộc tính nào là thuộc tính hỗn hợp.

1NF là yêu cầu tối thiểu của một mô hình dữ liệu quan hệ để nó có thể được triển khai trong ngôn ngữ truy vấn có cấu trúc (SQL). Trong mô hình Teaching-Learning đang thiết kế , không có vi phạm dạng chuẩn 1NF.

5.2.Dạng chuẩn 2NF

Định nghĩa một quan hệ ở dạng chuẩn 2NF nếu quan hệ đó:

- Là 1NF
- Các thuộc tính không khoá phải phụ thuộc hàm đầy đủ vào khoá chính

Một quan hệ ở dạng chuẩn 2NF nếu thoả mãn 1 trong các điều kiện sau: Khoá chính chỉ gồm một thuộc tính Bảng không có các thuộc tính không khoá Tất cả các thuộc tính không khoá phụ thuộc hoàn toàn vào tập các thuộc tính khoá chính

Trong mô hình đang thiết kế, không có vi phạm chuẩn 2NF

5.3.Dạng chuẩn 3NF

Một quan hệ ở dạng chuẩn 3NF nếu quan hệ đó:

- Là 2NF
- Các thuộc tính không khoá phải phụ thuộc trực tiếp vào khoá chính, nghĩa là không có phụ thuộc bắc cầu của các thuộc tính không phải nguyên tố.

Một quan hệ ở dạng 3NF nếu có ít nhất một trong các điều kiện sau đây trong mọi phụ thuộc hàm không tầm thường X -> Y

- X là một siêu khóa.

- Y là thuộc tính nguyên tố (mỗi phần tử của Y là một phần của khóa ứng viên nào đó).

Sau khi hoàn thiện, các mối quan hệ trong mô hình thiết kế đều đáp ứng các điều kiện của dạng chuẩn 3NF

5.4.Dạng chuẩn BCNF (Boyce Codd Normal Form)

Một quan hệ ở dạng chuẩn BCNF nếu quan hệ đó:

- Là 3NF
- Không có thuộc tính khoá mà phụ thuộc hàm vào thuộc tính không khoá.

Một quan hệ thỏa mãn BCNF nếu trong mọi phụ thuộc hàm không tầm thường X -> Y, X là siêu khóa.

Nhìn chung, mô hình mà nhóm thiết kế đã đáp ứng đầy đủ 4 dạng chuẩn hóa về yêu cầu dữ liệu.

6. Physical Design

6.1. Introduce Database Management System (DBMS)

Trong phạm vi môn học, sử dụng công cụ MySQL để tạo bảng dữ liệu.

6.1.1. Giới thiệu tổng quan về MySQL

MySQL là hệ quản trị cơ sở dữ liệu mã nguồn mở phổ biến hàng đầu trên thế giới và đặc biệt được ưa chuộng trong quá trình xây dựng, phát triển ứng dụng. Đây là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có khả năng thay đổi mô hình sử dụng phù hợp với điều kiện công việc khả chuyển.

Với tốc độ và tính bảo mật cao, MySQL thích hợp với các ứng dụng có truy cập cơ sở dữ liệu trên internet.

6.1.2. Ưu điểm và nhược điểm của MySQL

Ưu điểm:

- Sử dụng dễ dàng : công cụ dễ sử dụng và hoạt động trên nhiều hệ điều hành
- Tính bảo mật cao: phù hợp với các ứng dụng có truy cập cơ sở dữ liệu trên internet.
- Đa tính năng: có thể hỗ trợ hàng loạt các chức năng SQL từ hệ quản trị cơ sở dữ liệu quan hệ trực tiếp và cả gián tiếp.
- Khả năng mở rộng và mạnh mẽ : có khả năng xử lý khối dữ liệu lớn và có thể mở rộng
- Tương thích trên nhiều hệ điều hành: cung cấp phương tiện mà các máy khách có thể chạy trên cùng một máy tính với máy chủ hoặc trên một máy tính khác
- Cho phép khôi phục: cho phép các transaction được khôi phục, cam kết và phục hồi sự cố.

Nhược điểm:

MySQL bị hạn chế dung lượng, cụ thể, khi số bản ghi của người dùng lớn dần, sẽ gây khó khăn cho việc truy xuất dữ liệu, khiến người dùng cần áp dụng nhiều biện pháp để tăng tốc độ chia sẻ dữ liệu như chia tải database ra nhiều server, hoặc tạo cache MySQL.

So với Microsoft SQL Server hay Oracle, độ bảo mật của MySQL chưa cao bằng. Và quá trình Restore cũng có phần chậm hơn

6.1.3. Sử dụng MYSQL vào bài tập lớn

MySQL là sự lựa chọn thông dụng, tích hợp đầy đủ các tiện ích, dễ sử dụng. Công cụ này dễ sử dụng, lại còn hoạt động được ở nhiều hệ điều hành. MySQL được sử dụng với mục đích nhằm bổ trợ NodeJs, PHP, Perl, và nhiều ngôn ngữ khác. Và cuối cùng, công cụ này có phiên bản được sử dụng hoàn toàn miễn phí.

MySQL cũng có các nhược điểm, tuy nhiên trong phạm vi của môn học thì MySQL là hệ cơ sở dữ liệu hoàn toàn với những yêu cầu đã đề ra. Chính vì thế, chúng em quyết định sử dụng MySQL để lập trình hệ cơ sở dữ liệu.

6.2. Tạo bảng bằng công cụ MySQL

6.2.1. FACULTY

```
create schema school;
use school;

CREATE TABLE IF NOT EXISTS FACULTY(
   facultyID INT NOT NULL,
   facultyName VARCHAR (255) NOT NULL,

   PRIMARY KEY (facultyID)
);
```

6.2.2. LECTURER

```
CREATE TABLE IF NOT EXISTS LECTURER(
   lecturerID INT NOT NULL,
   fullName VARCHAR (255) NOT NULL,
   facultyID INT NOT NULL,

   PRIMARY KEY (lecturerID),
   FOREIGN KEY (facultyID)
      REFERENCES FACULTY (facultyID)
      ON DELETE RESTRICT
      ON UPDATE CASCADE
);
```

6.2.3. STUDENT

```
CREATE TABLE IF NOT EXISTS STUDENT (
   studentID INT NOT NULL ,
   fullName VARCHAR (255) NOT NULL ,
   dateOfBirth date,
   sex VARCHAR(10) DEFAULT NULL,
   status VARCHAR(10) DEFAULT NULL,
   facultyID INT NOT NULL,

   PRIMARY KEY (studentID),
   FOREIGN KEY (facultyID)
      REFERENCES FACULTY (facultyID)
      ON DELETE RESTRICT
      ON UPDATE CASCADE
);
```

6.2.4. SUBJECT

```
CREATE TABLE IF NOT EXISTS SUBJECT(
   subjectID CHAR (6) NOT NULL,
   subjectName VARCHAR (255) NOT NULL,
   facultyID INT NOT NULL ,
   numOfCredits INT NOT NULL CHECK (numOfCredits > 0 AND numOfCredits <
4),
   semester INT NOT NULL,

   PRIMARY KEY (subjectID, semester),
   FOREIGN KEY (facultyID)
      REFERENCES FACULTY (facultyID)
      ON DELETE RESTRICT
      ON UPDATE CASCADE
);
```

6.2.5. CLASS

```
   CREATE TABLE IF NOT EXISTS CLASS(
      className VARCHAR (255) NOT NULL,
      subjectID CHAR (6) NOT NULL,
      semester INT NOT NULL,

      PRIMARY KEY (className, subjectID, semester),
      FOREIGN KEY (subjectID, semester)
      REFERENCES SUBJECT (subjectID, semester)
      ON DELETE RESTRICT
      ON UPDATE CASCADE
   );
```

6.2.6. PUBLISHER

```
   CREATE TABLE IF NOT EXISTS PUBLISHER(
      publisherID INT NOT NULL,
      publisherName VARCHAR (255) NOT NULL,
      domestic BOOLEAN DEFAULT FALSE,

      PRIMARY KEY (publisherID)

   );
```

6.2.7. TEXTBOOK

```
   CREATE TABLE IF NOT EXISTS TEXTBOOK(
      textbookID INT NOT NULL,
      textbookName VARCHAR (255) NOT NULL,
      specialization VARCHAR (255) NOT NULL,
```

```
    publisherID INT NOT NULL,
    publishcationDate DATE,

    PRIMARY KEY (textbookID),
    FOREIGN KEY (publisherID)
      REFERENCES PUBLISHER (publisherID)
      ON DELETE RESTRICT
      ON UPDATE CASCADE
);
```

6.2.8. AUTHOR

```
CREATE TABLE IF NOT EXISTS AUTHOR(
    authorID INT NOT NULL,
    authorName VARCHAR (255) NOT NULL,
    PRIMARY KEY (authorID)
);
```

6.2.9. TEACH

```
CREATE TABLE IF NOT EXISTS TEACH(
    className VARCHAR(255) NOT NULL,
    subjectID CHAR(6) NOT NULL,
    lecturerID INT NOT NULL,
    week INT NOT NULL,
    semester INT NOT NULL,

    PRIMARY KEY (className, subjectID, week, lecturerID, semester),
    FOREIGN KEY (className)
      REFERENCES CLASS (className)
      ON DELETE RESTRICT
      ON UPDATE CASCADE,
    FOREIGN KEY (subjectID, semester)
      REFERENCES CLASS (subjectID, semester)
      ON DELETE RESTRICT
      ON UPDATE CASCADE,
    FOREIGN KEY (lecturerID)
      REFERENCES LECTURER (lecturerID)
      ON DELETE RESTRICT
      ON UPDATE CASCADE
);
```

6.2.10. REGISTER

```
CREATE TABLE IF NOT EXISTS REGISTER (
```

```
    subjectID CHAR(6) NOT NULL,
    studentID INT NOT NULL,
    semester INT NOT NULL,

    PRIMARY KEY (subjectID , studentID, semester),
    FOREIGN KEY (subjectID, semester)
       REFERENCES SUBJECT (subjectID, semester)
       ON DELETE RESTRICT
       ON UPDATE CASCADE,
    FOREIGN KEY (studentID)
       REFERENCES STUDENT (studentID)
       ON DELETE RESTRICT
       ON UPDATE CASCADE
);
```

6.2.11. MANAGESUBJECT

```
CREATE TABLE IF NOT EXISTS MANAGESUBJECT(
    subjectID CHAR(6) NOT NULL,
    textbookID INT NOT NULL,
    lecturerID INT NOT NULL,
    semester INT NOT NULL,

    PRIMARY KEY (subjectID, textbookID, semester),
    FOREIGN KEY (subjectID, semester)
       REFERENCES SUBJECT (subjectID, semester)
       ON DELETE RESTRICT
       ON UPDATE CASCADE ,
    FOREIGN KEY (textbookID)
       REFERENCES TEXTBOOK (textbookID)
       ON DELETE RESTRICT
       ON UPDATE CASCADE
);
```

6.2.12. COMPILE

```
CREATE TABLE IF NOT EXISTS COMPILE(
    authorID INT NOT NULL,
    textBookID INT NOT NULL,

    PRIMARY KEY (authorID, textBookID),
    FOREIGN KEY (authorID)
       REFERENCES AUTHOR (authorID)
       ON DELETE RESTRICT
       ON UPDATE CASCADE ,
    FOREIGN KEY (textBookID)
       REFERENCES TEXTBOOK (textBookID)
```

```
        ON DELETE RESTRICT
        ON UPDATE CASCADE
    );
```

6.2.13. MANAGEPOINT

```
  CREATE TABLE IF NOT EXISTS MANAGEPOINT (
    subjectID CHAR(6) NOT NULL,
    className VARCHAR(255) NOT NULL,
    studentID INT NOT NULL,
    lecturerID INT NOT NULL,
    summaryPoINT FLOAT(1) CHECK ((summaryPoINT >= 0 AND
summaryPoINT <= 10) OR summaryPoINT IS NULL ),
    semester INT NOT NULL,

    PRIMARY KEY (subjectID, className, studentID, semester),
    FOREIGN KEY (subjectID, semester)
      REFERENCES CLASS (subjectID, semester)
      ON DELETE RESTRICT
      ON UPDATE CASCADE ,
    FOREIGN KEY (className)
      REFERENCES CLASS (className)
      ON DELETE RESTRICT
      ON UPDATE CASCADE,
    FOREIGN KEY (studentID)
      REFERENCES STUDENT (studentID)
      ON DELETE RESTRICT
      ON UPDATE CASCADE
  );
```

6.3. Table Relationships

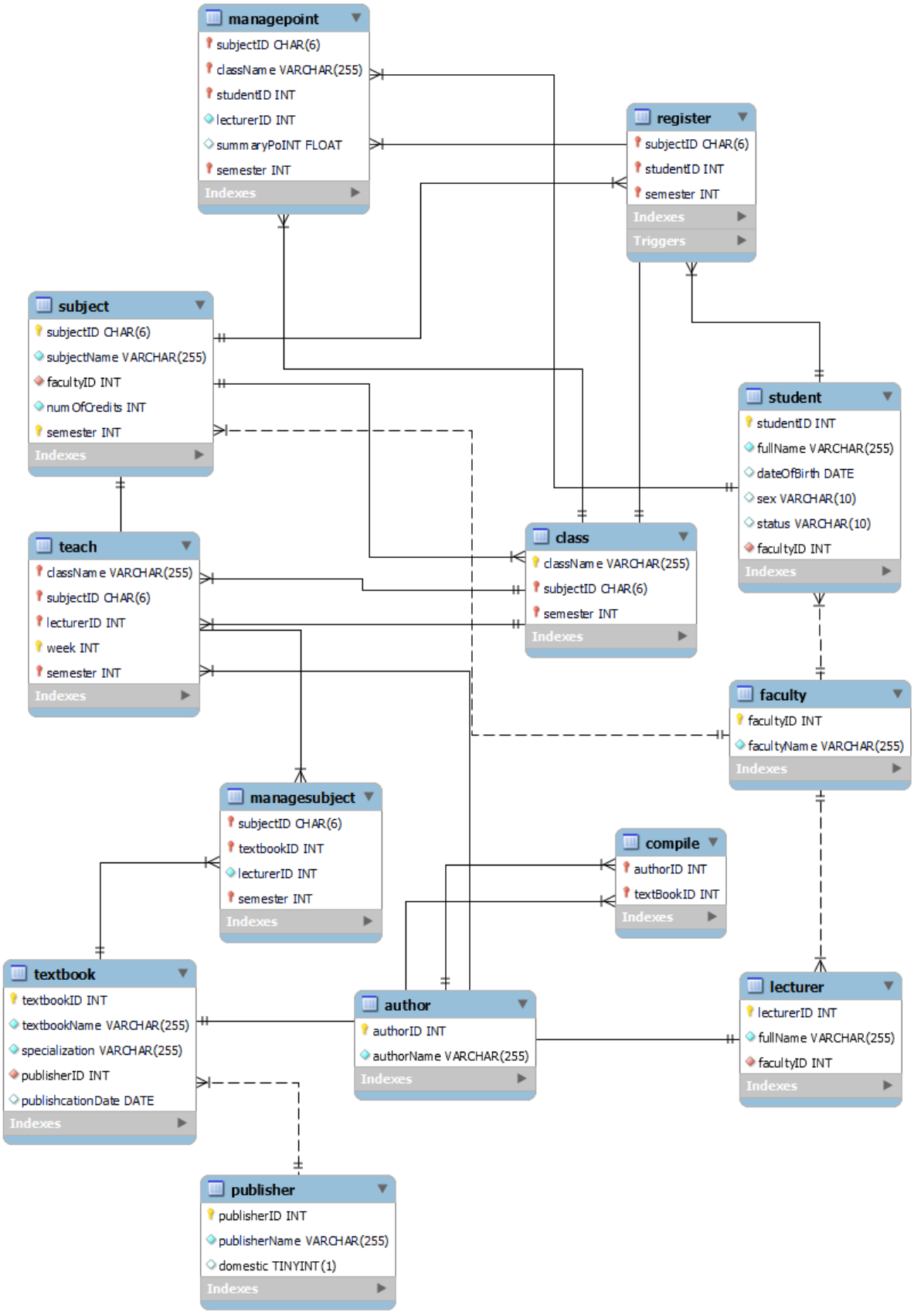


Table Relationship in MySQL

6.4. Insert Data

5.4.1. Code

```
/*Dump data into Table Student and Faculty*/

INSERT INTO `FACULTY` (`facultyID`, `facultyName`)
VALUES (106, 'Khoa hoc va ki thuat may tinh'),
      (113, 'Ki thuat hoa hoc'),
      (115, 'Quan li cong nghiep');

INSERT INTO `Student` (`studentID`, `fullName`, `dateOfBirth`, `sex`, `status`, `facultyID`)
VALUES (1914262, 'Nguyen Ngoc Trinh', '2000-10-04', 'Female', 'Active', 106),
      (1921345, 'Luong Huu Phuoc', '2001-10-03', 'Male' , 'Active', 113),
      (1922415, 'Nhan Phuc Vinh', '2001-03-03', 'Male' , 'Active', 115);
```

6.4.2. Result

Result Grid | Filter Rows: | Edit: | Export/Import:

facultyID	facultyName
106	Khoa hoc va ki thuat may tinh
113	Ki thuat hoa hoc
115	Quan li cong nghiep
NULL	NULL

FACULTY Table

Result Grid | Filter Rows: | Edit: | Export/Import:

studentID	fullName	dateOfBirth	sex	status	facultyID
1914262	Nguyen Ngoc Trinh	2000-10-04	Female	Active	106
1914562	Nguyen Ngoc Trinh	2000-10-04	Female	Active	106
1921345	Luong Huu Phuoc	2001-10-03	Male	Active	113
1922415	Nhan Phuc Vinh	2001-03-03	Male	Active	115
NULL	NULL	NULL	NULL	NULL	NULL

STUDENT Table