

Quản Lý Điểm

Student Grading Management Subsystem

Giới thiệu

1. Thông tin cá nhân:

- **Họ tên:** Trịnh Tùng Sơn.
- **Mã số sinh viên:** HE163856.
- **Lớp:** IA1604.
- **Khoa:** Công nghệ thông tin.
- **Chuyên ngành:** An toàn thông tin.
- **Trường:** Đại học FPT cơ sở Hòa Lạc.
- **Email:** Sontthe163856 @fpt.edu.vn.

Số điện thoại liên lạc: 0832066696

2. Về bản báo cáo:

Tên: Database **Student Grading Management Subsystem** cho sinh viên trường đại học FPT.

Bản báo cáo này được tạo ra theo yêu cầu của giảng viên thực hiện giảng dạy môn DBI202 của lớp IA1604 như một bài Progress Test .

Tóm tắt nội dung Bản báo cáo: Từ những dữ liệu mà đề bài cung cấp tóm tắt, phân tích và tạo ra một database **Student Grading Management Subsystem** và kiểm tra độ ứng dụng vào thực tế.

I. PHÂN TÍCH ĐỀ BÀI VÀ Xác Định Object có thể Trong DataBase.

A. Phân tích, Xác định Một Số Yêu cầu đặc biệt của đề bài.

1.Requirement 1 (Pic 1): BẢNG THÔNG TIN ĐIỂM THÀNH PHẦN

2.REQUIREMENT 2(Pic 2): BẢNG MÃ MÔN, KHÓA HỌC, TRẠNG THÁI HỌC TẬP CỦA SINH VIÊN

3.REQUIREMENT 3(Pic 3): BẢNG ĐIỂM CỦA SINH VIÊN

B .Các đối tượng cơ bản sẽ có trong hệ thống điểm :

C.Xác định Mỗi Quan Hệ Giữa Các Đối tượng

D.Từ những dữ liệu trên, các attributes ứng với từng entities được dự đoán

II.Chuẩn hóa:

1. Student
2. Group
3. Student _ Group (Enroll)
4. Lecturer
5. Subject
6. Assessment
7. Assessment _ Subject (Include)
8. Class
9. Semester
- 10.Class_ Semester

11. Grade

12. Category

III. Diagram

IV. Chuyển Đổi

Phân Tách Các Quan Hệ < n-n > Và Tạo Table

1. STUDENTS VÀ GROUPS

2. CLASSES VÀ Semester

3. Assessment và Subject

V. Phân tích MD và dữ liệu sẽ được add vào database:

1. **Student**

2. Lecturers

3. Classes

4. Subject

5. Group

6. Group_Student

7. Semester

8. Class_Student

9. Assessment

10. Category

11. Assessment_Subject

12. Grade

VI. Tạo Database và insert dữ liệu:

1. Tạo Database và các bảng:
2. Insert dữ liệu

VII. Query

1. Query dùng ORDER BY.
2. Query dùng INNER JOIN.
3. Query dùng Aggregate function.
4. Query dùng GROUP BY và HAVING.
5. Query sử dụng sub-query như 1 quan hệ.
6. Query sử dụng sub-query ở mệnh đề WHERE:
7. Query sử dụng partial matching trong mệnh đề WHERE:
8. Query sử dụng self-join:
9. Query ngẫu nhiên 1:
10. Query ngẫu nhiên 2:

I .Phân tích đề bài và Xác Định Object có thể Trong DataBase.

A.Phân tích, Xác định Một Số Yêu cầu đặc biệt của đề bài.

1.Requirement 1 (Pic 1)

RQ1: Giảng viên sẽ cho điểm từng assessment của môn học học đó (trong hình là các Assesment của môn DBI202) .

Bảng bao gồm các cột :

- v Category: Loại bài test / đầu điểm.
- v Type: Kiểu bài test.
- v Part: Bao gồm bao nhiêu phần.
- v Weight: Trọng số của số điểm bài test đó ảnh hưởng lên điểm trung bình môn học.
- v Completion Criteria: Điều kiện pass môn.
- v Duration: Thời lượng của bài test.
- v Question Type: Loại câu hỏi xuất hiện.
- v No Question: Số lượng câu hỏi 1 bài.
- v Knowledge and Skill: Kiến thức và kĩ năng cần thiết để có thể làm được bài test.
- v Grading Guide: Hướng dẫn cho điểm.
- v Note: Chú thích của giáo viên.

RQ1 có được đáp ứng bởi table (entity) Assesment đã được phân tích ở phần A. Đồng thời từ hình trên ta cũng xác định được các Attribute của bảng Assessment như Category , Type...

Tuy nhiên trong hệ thống đánh giá điểm này có sự khác biệt giữa các Assessment giống nhau trong các Subject khác nhau.

Vd như Progress test trong môn DBI có Type là 'quiz' , Weight là 10 % ,Duration là '20 phút 'và No question (số lượng câu hỏi) là '20' .

Tuy nhiên Progress test trong môn MAD sẽ là multiple choice trên EOS , Weight là '15%', Duration là '30 phút 'và No question (số lượng câu hỏi) là '15' .

Do đó những thuộc tính này sẽ nằm trong bảng Subject_ Assessment được tách từ mối quan hệ N-N giữa 2 bảng Subject và Assessment .

Và để nhằm đạt chuẩn 2NF ta cũng cần tách Type của Assessment_ Subject ra 1 bảng mới . Ở đây sẽ để là Category.

2.Requirement 2 (Pic 2)

RQ2: Sinh viên có thể kiểm tra kết quả của mình vào cuối kỳ học (vd như hình 2).

Bảng bao gồm các cột:

v No: Số thứ tự

v Subject Code: Mã số của môn học.

v Subject Name: Tên của môn học.

v Semester: Kì học mà sinh viên đã tham gia môn học tương ứng.

v Group: Nhóm sinh viên mà sinh viên này đã tham gia học cùng môn học tương ứng.

- v Start Date: Ngày bắt đầu của môn học tương ứng.
- v End Date: Ngày kết thúc của môn học tương ứng.
- v Average Mark: Điểm trung bình của sinh viên ở môn học tương ứng.
- v Status: Trạng thái của sinh viên ở môn học tương ứng (passed or not passed).

những môn học mà Sinh viên đó đã tham gia trong mỗi kỳ học và những thông tin liên quan. Tuy nhiên nó sẽ không được coi là 1 Entity mà cần tách thành các bảng có mối quan hệ với nhau như : Subject, Semester, Group để nhằm chuẩn hóa dữ liệu tối thiểu là 3NF (vd như ta có thể dễ dàng thấy Subject Name sẽ chỉ phụ thuộc vào Subject Cod; Start Date và End Date sẽ chỉ phụ thuộc vào Group) . Ngoài ra để lấy được Average Mark và Status ta cũng cần join với bảng Grade (và Attendance nếu có).

3.Requirement 3 (Pic 3)

RQ3 : Với mỗi môn học , Student có thể kiểm tra chi tiết kết quả từng Assesment

Bảng gồm 5 cột:

- v Grade Category: Các loại đầu điểm mà sinh viên phải có trong môn học này.
- v Grade Item: Từng phần của mỗi loại đầu điểm.
- v Weight: Trọng số của từng đầu điểm đối với môn học này.
- v Value: Điểm số sinh viên đạt được ứng với mỗi bài test.
- v Comment: ghi chú nếu có

Ta có trong hình 3 mỗi record là 1 đầu điểm của Sinh viên trong môn học cụ thể .và cuối cùng là 2 thông tin về điểm trung bình và trạng thái về tiến độ môn học .

Tất cả thông tin trong bảng này có thể lấy được từ 3 bảng Subject (Course) , Grade , và Assesment .

B .Các đối tượng cơ bản sẽ có trong hệ thống điểm :

Student : Hệ thống đánh giá điểm với mục tiêu chủ đạo là lưu trữ điểm số của tất cả sinh viên trong suốt quá trình học của 1 cơ sở giáo dục . Vì vậy SINH VIÊN (Student) sẽ là đối tượng chủ đạo, bắt buộc trong hệ thống này.

Lecturer : Để đánh giá được điểm số của sinh viên thì chắc chắn cần có giảng viên , là những người trực tiếp giảng dạy và cho điểm sinh viên , vì thế GIẢNG VIÊN (Lecture) sẽ là đối tượng cần thiết để dễ dàng quản lý , khắc phục sai sót nếu có về sau.

Subject : Trong quá trình học tập, đào tạo, một sinh viên cần trải qua nhiều môn học , và để hoàn thành chương trình đào tạo thì sinh viên cần đạt được ngưỡng điểm nhất định đối với những môn học trong giáo trình .đồng thời điểm trung bình của các môn học cũng là thước đo để đánh giá quá trình học của sinh viên đó. Vì vậy MÔN HỌC (Course or Subject) cũng sẽ là 1 đối tượng chính trong hệ thống điểm.

Class : Vì số lượng sinh viên lớn nên sẽ được chia nhỏ vào các lớp học với từng môn học ,đồng thời giảng viên cũng sẽ được phân công dạy các lớp đó.Do đó CLASS cũng sẽ là 1 đối tượng cần thiết để dễ dàng lưu trữ quản lý thông tin trong quá trình học và dạy của sinh viên và giảng viên

Assesment : Trong các Course, để đánh giá tiến độ hoàn thiện của 1 sinh viên thì thường sẽ chia ra làm bài kiểm tra (đầu điểm) theo từng giai đoạn học . Các Assesment chính là cơ sở để lưu trữ thông tin về các bài kiểm tra đó. Đồng thời cũng là để tính điểm trung bình của môn học tương ứng

Semester : Một đối tượng khác để giúp chia nhỏ quá trình đánh giá điểm . Mỗi học kỳ 1 Sinh viên sẽ học và thi 1 số môn học , và các môn học có thể thêm , thay đổi trong mỗi học kỳ. Đồng thời một học kỳ cũng là thời gian tồn tại của 1 lớp học.

Grade : Đối tượng trọng tâm trong hệ thống đánh giá điểm của 1 trường , Grade lưu giữ điểm của từng học sinh với từng bài kiểm tra (Assesment) đã tham gia. Đây sẽ là bảng trung tâm của hệ thống và lưu trữ nhiều bản ghi nhất.

C.XÁC ĐỊNH MỐI QUAN HỆ GIỮA CÁC ĐỐI TƯỢNG

(Giải thích sự thay đổi của 1 số bảng và quan hệ)

§ Student là Group : các sinh viên sẽ tham gia vào các Group khác nhau dựa trên mỗi môn học , mỗi kỳ học và chuyên ngành của Sinh viên đó , có nghĩa là 1 Sinh viên sẽ tham gia nhiều Group . Và ngược lại chắc chắn 1 Group cũng sẽ có nhiều Sinh viên . Do vậy quan hệ giữa Student và Group là N – N.

§ Group là Class : Ở đây do yêu cầu đặc biệt trong hệ thống của FPT, Một lớp học với mã lớp cố định (vd: IA1604) Có thể tồn tại trong nhiều kỳ học . Tuy nhiên IA1604 trong những môn học khác nhau lại có danh sách Sinh viên khác nhau , Giảng viên khác nhau . Và danh sách Sinh viên của IA1604

trong mỗi môn học đó chính là Group . Và 1 bảng mới được sinh ra để lưu giữ tên lớp chung (IA1604, v.v..) gọi là Class. Việc gộp chung nhiều Group dưới tên 1 Class này thoạt nhìn có vẻ vô nghĩa và làm phức tạp hơn cho hệ thống . Tuy nhiên nó lại mang ý nghĩa trong việc quản lý Sinh viên . Bởi vì những Group trong Class đó đều có điểm chung về chuyên ngành , và thường trong những Group đó . Và vì 1 class sẽ có nhiều group nhưng 1 Group chỉ có thể nằm dưới tên 1 Class . Vậy nên quan hệ giữa Group và Class là quan hệ 1 – N-

§ Class là Semester : Mỗi Class sẽ đại diện cho chuyên ngành và niên khóa của những Sinh viên trong kỳ đầu tiên mà Class được tạo ra . Và nó sẽ duy trì có thể là đến khi không còn Sinh viên nào trong lớp đó nữa . Vì vậy 1 Class sẽ có thể tồn tại trong nhiều kỳ học , và ngược lại , chắc chắn trong 1 kỳ học sẽ có nhiều class như thế. Do đó quan hệ giữa Class và Semester là N – N

§ Group là Subject : Như đã phân tích ở trên , mỗi Group trong cùng 1 Class khác nhau dựa trên môn học (Subject) khác nhau . Điều đó nghĩa là Group phụ thuộc rất lớn vào Subject và mỗi Group tồn tại chỉ để nhằm học 1 Subject và có 1 Giảng viên riêng. Còn ở chiều ngược lại , 1 Subject sẽ có nhiều lớp học môn đó . Vì vậy quan hệ giữa Group và Subject là quan hệ 1 – N

§ Group là Lecturer : Mỗi Group sẽ tồn tại để cho những Sinh viên trong Group đó học 1 môn , và nó sẽ được giảng dạy bởi 1 Lecturer . Sẽ chỉ có 1 Giảng viên chịu trách nhiệm và đánh giá điểm cho Sinh viên trong 1 Group . Tuy nhiên 1 Giảng viên có thể giảng dạy nhiều lớp trong suốt quá trình làm việc , với 1 hay nhiều môn học . Do đó quan hệ giữa Group và Lecturer là quan hệ 1 – N .

§ Subject là Assessment : Như đã phân tích ở phần A . Mỗi Subject sẽ có nhiều Assessment hay nhiều bài test để đánh giá tiến độ của Sinh viên tham gia môn học đó . Đồng..... thời một đầu điểm Assessment có thể nằm trong nhiều môn học (Subject) vd điển hình là final exam (FE) sẽ là đầu điểm

bắt buộc cho hầu hết các môn học . Vì vậy mối quan hệ giữa Subject và Assessment là một quan hệ N – N .

§ Student là Grade : Mối quan hệ chủ đạo của hệ thống đánh giá điểm chính là Điểm và Sinh viên Theo đó , 1 sinh viên chắc chắn sẽ có nhiều đầu điểm với nhiều môn học khác nhau và mỗi bản ghi trong Grade chỉ là điểm của 1 Sinh viên . Suy ra quan hệ giữa Student và Grade là 1– N .

§ Grade là Assessment và Grade là Subject Để đảm bảo hạn chế lỗi khi insert dữ liệu , Grade sẽ tạo mối quan hệ trực tiếp với bảng Assessment _ Subject được tách ra từ mối quan hệ N – N của 2 bảng đó. Và vì 1 bản ghi trong Grade sẽ chỉ lưu giữ được 1 đầu điểm của 1 môn của 1 hs , trong khi 1 Assessment -Subject sẽ xuất hiện trong nhiều record của Grade . Suy ra quan hệ giữa Grade và Assessment- _ Subject sẽ là 1—N .

§ Assessment _ Subject là Category . Như đã phân tích ở Phần B – RQ1. Chúng ta cần tách attribute Type ra 1 bảng riêng. Bảng Category này sẽ lưu trữ các Type của Assessment , và đặc biệt khi tạo mối quan hệ trực tiếp với Assessment _ Subject thì mỗi Assessment trong Subject cố định sẽ chỉ có 1 Type duy nhất . Chiều ngược lại chắc chắn 1 Type sẽ xuất hiện trong nhiều Assessment . Vì vậy quan hệ giữa Assessment _ Subject và Category sẽ là quan hệ 1 – N .

D. Từ những dữ liệu trên, các attributes ứng với từng entities được dự đoán như sau:

Student : Các attributes của tables Student chính là các thông tin cơ bản của 1 sinh viên:

- Mã số sinh viên được cấp bởi nhà trường cho 1 sinh viên nhất định.
- Địa chỉ email nhà trường cung cấp cho sinh viên.
- Tên gọi của sinh viên.
- Họ và tên đệm của sinh viên.
- Giới tính của sinh viên.
- Ngày tháng năm sinh của sinh viên.
- Những kỳ học mà sinh viên tham gia học tập.
- Những nhóm mà sinh viên tham gia ở mỗi kì.
- Những môn mà sinh viên học trong mỗi kì.
- Loại đầu điểm ở mỗi môn học.
- Giảng viên thực hiện giảng dạy ở các môn học tương ứng.
- Điểm của sinh viên tại mỗi đầu điểm.

Groups : Các attributes của tables Groups là các thông tin cơ bản của 1 nhóm:

- Mã nhận diện của 1 nhóm.
- Sinh viên đã tham gia nhóm đó.
- Chuyên ngành chính của nhóm sinh viên đó.
- Số lượng sinh viên của 1 nhóm.
- Kỳ học mà nhóm đã tham gia học tập.
- Lớp học theo kỳ.
- Các môn học mà nhóm đã tham gia.
- Điểm trung bình của môn học tương ứng.
- Giảng viên đã giảng dạy các môn học tương ứng

Classes : Các attributes của Classes là các thông tin cơ bản của 1 lớp học:

Mã nhận diện 1 lớp học.

Các nhóm học sinh đã tham gia học lớp đó.

Môn học được giảng dạy ở lớp học đó.

Giảng viên tham gia giảng dạy ở lớp học đó.

Điểm trung bình của cả nhóm tham gia học môn học tương ứng.

Courses : Các attributes của Courses là các thông tin cơ bản của 1 môn học:

-Mã nhận diện 1 môn học.

- Các loại bài test, các đầu điểm mà một môn học cần phải có.
- Các giáo viên sẽ tham gia dạy môn học này.
- Các lớp học mà được phân cho giảng dạy môn học này.
- Kỳ học mà lớp học này được phân vào.

Lecturers : Các attributes của Lecturers là các thông tin cơ bản của 1 giảng viên:

- Mã số giảng viên được nhà trường cung cấp.
- Địa chỉ email mà nhà trường cấp cho giảng viên.
- Tên gọi của các giảng viên.
- Họ và tên đệm của giảng viên.
- Giới tính của giảng viên.
- Ngày tháng năm sinh của giảng viên.
- Các lớp mà giảng viên tham gia giảng dạy.
- Các môn học mà giảng viên tham gia giảng dạy.
- Trưởng bộ môn mà giảng viên giảng dạy, giảng viên sẽ phải báo cáo tình hình giảng dạy cho trưởng bộ môn.
- Kỳ học mà giảng viên phải tham gia giảng dạy.

Grade: Các attributes của tables Grade là các thông tin cơ bản của 1 loại đầu điểm.

- Mã nhận diện.
- Tên của đầu điểm, hoặc bài test mà sinh viên cần thực hiện.
- Loại bài test (on-going, at home, ...).
- Loại đầu điểm tương ứng cần mấy phần.
- Trọng số mà đầu điểm này ảnh hưởng đến điểm trung bình 1 môn.
- Điều kiện pass môn.
- Thời lượng của bài test.
- Loại câu hỏi xuất hiện.
- Số lượng câu hỏi 1 bài.
- Kiến thức và kỹ năng cần thiết để có thể làm được bài test.
- Hướng dẫn cho điểm cho giảng viên.
- Chú thích của giảng viên (nếu có).

Semester: Các attributes của Tables Semester là các thông tin cơ bản của 1 Kỳ học

- Mã số cho các kỳ học trong năm
- Tên của các kỳ trong năm mà sinh viên theo học
- Các lớp học tập trong kỳ học

-Điểm thành phần và tổng kết của sinh viên trong kỳ học

-Giáo viên dạy trong kỳ học đấy

II. Chuẩn hóa .

1. Student :

+ 1 học sinh có thể tham gia vào nhiều nhóm khác nhau, nhiều môn học khác nhau, nhiều kì khác nhau, nhiều lớp khác nhau và được nhiều giáo viên giảng dạy . Nếu đưa vào 1 bảng sẽ xuất hiện những ô đa trị (không phải nguyên tử) à ko đạt 1NF à tách các thuộc tính và tạo FK vs các bảng như Group , Grade , ...

+ Tách những attribute với multiple part như Full Name thành các thuộc tính nhỏ hơn : First Name và Last Name.

Sau khi tách ta được bảng Student với các thuộc tính :

- **StuID** : Mã số sinh viên được cấp bởi nhà trường cho 1 sinh viên. Đây sẽ chính là Primary Key của bảng này.
- Email: Địa chỉ email mà nhà trường cấp cho sinh viên.
- Phone : Số điện thoại của Sinh viên
- First Name: Tên gọi của sinh viên.
- Last Name: Họ và tên đệm của sinh viên.
- Gender: Giới tính của sinh viên.

Sau khi tách hết các multiple values attributes, bảng Student giờ chỉ còn 1 primary key là **Mssv**, không có non-key attributes nào phụ thuộc 1 phần vào key, không có non-key attributes nào phụ thuộc bắc cầu vào key

2. Group

+1 nhóm thì có nhiều sinh viên tham gia vào, Và nhiều nhóm có thể chung 1 class, Từ group chúng ta có thể suy ra được tên class

è Cần tách các thuộc tính và tạo FK vs bảng Student và Class để đạt chuẩn 2NF.

+ Sau khi tách bỏ hết, bảng Groups sẽ có các cột sau:

- **GID** : Mã nhận diện của 1 nhóm. Mã này cũng là primary key của bảng.
- LecID : Mã Giảng viên dạy lớp đó
- SubID: Môn học của Group đó
- ClassID: Mã lớp của Group đó
- StartDate: Ngày bắt đầu
- EndDate: Ngày kết thúc

+ Sau khi tách hết các multiple values attributes, bảng Groups giờ chỉ còn 1 primary key là GrID , không có non-key attributes nào phụ thuộc 1 phần vào key, không có non-key attributes nào phụ thuộc bắc cầu vào key. Sẽ đạt chuẩn 3NF

3. Student _ Group (Enroll)

Vì quan hệ giữa 2 bảng Student và Group là N – N nên để chuẩn hóa ta cần tạo thêm 1 bảng mới liên kết 2 bảng với nhau.

Bảng enroll sẽ bao gồm các thuộc tính:

- **StuID** : Mã sinh viên , khóa ngoại tới StuID trong bảng Student
- **GID** : Mã Group , khóa ngoại tới GID trong bảng Group

2 thuộc tính này cũng chính là khóa chính của bảng enroll . Và bảng này cũng sẽ thỏa mãn 3NF.

4. Lecturer

+ một giảng viên có thể dạy nhiều môn học, một môn học lại được xếp vào nhiều lớp, một lớp lại được phân vào nhiều kì. → Khi đưa vào bảng thì các attributes này sẽ có phụ thuộc bắc cầu với nhau.

→ Chưa đạt chuẩn 2nd. Cần tách các thuộc tính này ra và tạo FK với các bảng liên quan.

+Sau khi tách bản Lecturer sẽ có các thuộc tính :

- **LecID** : Mã số mà nhà trường cấp cho mỗi giảng viên. Mã số này là độc nhất của mỗi giảng viên, do vậy, nó cũng chính là primary key của bảng.
- Email: Địa chỉ email mà nhà trường cấp cho giảng viên.
- First Name: Tên gọi của giảng viên.
- Last Name: Họ và tên đệm của giảng viên.
- Gender: Giới tính của giảng viên.
- Phone : Điện thoại liên lạc

5. Subject

+ 1 môn học thì cần rất nhiều đầu điểm để đánh giá. 1 môn thì cũng được học bởi nhiều Group .

Để tránh đa trị và đạt chuẩn 3NF , cần tách các thuộc tính và tạo FK với các bảng liên quan.

+ Bảng sau khi tách sẽ có các thuộc tính :

- SubID : Mã của môn học, mã này cũng là unique đối với mỗi môn.
- Name: Tên của môn học tương ứng.

Bỏ hết attributes liên quan đến đầu điểm hay giảng viên và class vì chúng ta sẽ thiết kế bảng riêng cho chúng. Như vậy, thì sau khi tách hết các multiple values attributes, bảng Courses giờ chỉ còn 1 primary key là CouID , không có non-key attributes nào phụ thuộc 1 phần vào key, không có non-key attributes nào phụ thuộc bắc cầu vào key.

6. Assessment

+ Sau khi tách các multiple values attributes liên quan đến các đầu điểm đánh giá môn học từ bảng Subject thì ta sẽ đặt những attributes đó vào bảng Assessment này.

+ Mỗi Assessment trong từng môn học sẽ khác nhau . Nên để chuẩn hóa ta cần tách các thuộc tính và tạo FK với các bảng liên quan.

+ Sau khi tách các thuộc tính của Assessment :

AssesID : Mã bài test;

AssesName : Tên của đầu điểm .

+ Bảng Assessment chỉ còn 1 PK và 1 attribute phụ thuộc hoàn toàn vào PK à đạt chuẩn 3NF

7. Assessment _ Subject (Include)

Vì quan hệ giữa 2 bảng Assessment và Subject là N – N nên để chuẩn hóa ta cần tạo thêm 1 bảng mới liên kết 2 bảng với nhau.

Bảng include sẽ bao gồm các thuộc tính:

- **AssesID** : Mã đầu điểm , khóa ngoại tới AssesID trong bảng Assessment
- **SubID** : Mã môn học , khóa ngoại tới SubID trong bảng Subject
- **Weight** : Trọng số, tỉ lệ phần trăm mà đầu điểm chiếm trong môn học đó
- **Duration** : Giới hạn thời gian làm bài.
- **No question**: Số câu hỏi.

- Criteria : Điểm số tối thiểu để qua môn

2 thuộc tính AssesID và SubID cũng chính là khóa chính của bảng enroll . Và bảng này cũng sẽ thỏa mãn 3NF

8. Class

+ 1 lớp thì sẽ có thể có nhiều nhóm được xếp cho tham gia vào và do vậy cũng kéo theo việc 1 lớp có nhiều điểm trung bình từ các nhóm sinh viên khác nhau.

è Khi đưa vào bảng thì các attributes này sẽ có phụ thuộc bắc cầu với nhau.

è Chưa đạt chuẩn 2NF, Cần tách các thuộc tính phụ thuộc và tạo FK với các bảng liên quan.

+ Bảng Class sau khi tách :

- **ClassID**: Mã lớp
- ClassName: Tên lớp

+ Bảng Class chỉ còn 1 PK và 1 attribute phụ thuộc hoàn toàn vào PK à đạt chuẩn 3NF

9. Semester

+ Vì 1 kỳ có thể có nhiều class nên để chuẩn hóa ta cũng cần tách các thuộc tính và tạo FK với các bảng liên quan.

+ Thuộc tính của Semester:

SemID : Mã học kỳ

SemName: Tên đầy đủ

+ Bảng Semester chỉ còn 1 PK và 1 attribute phụ thuộc hoàn toàn vào PK à đạt chuẩn 3NF

10. Class_ Semester

Vì quan hệ giữa 2 bảng Class và Semester là N – N nên để chuẩn hóa ta cần tạo thêm 1 bảng mới liên kết 2 bảng với nhau.

Bảng Class_ Semester sẽ bao gồm các thuộc tính:

- **ClassID** : Mã lớp , khóa ngoại tới ClassID trong bảng Class
- **SemID** : Mã học kỳ , khóa ngoại tới SemID trong bảng Semester

2 thuộc tính này cũng chính là khóa chính của bảng. Và do đó bảng này cũng sẽ thỏa mãn 3NF

11. Grade

+ Sau khi đã hoàn thành bảng Assessment_ Subject , ta đã có đầy đủ mọi thứ cần thiết để cho điểm đánh giá cho sinh viên. Do vậy bảng Assessment System sẽ có 1 liên kết với bảng Students .

+ Bảng Grade được chuẩn hóa có dạng :

GradeID : Số thứ tự bản ghi, là duy nhất PK

AssesID và SubID tạo FK đến bảng Assessment _ Subject

LecID : Mã Giảng viên cho điểm

StuID : Mã Sinh viên nhận điểm

Mark : Điểm nhận được.

12. Category

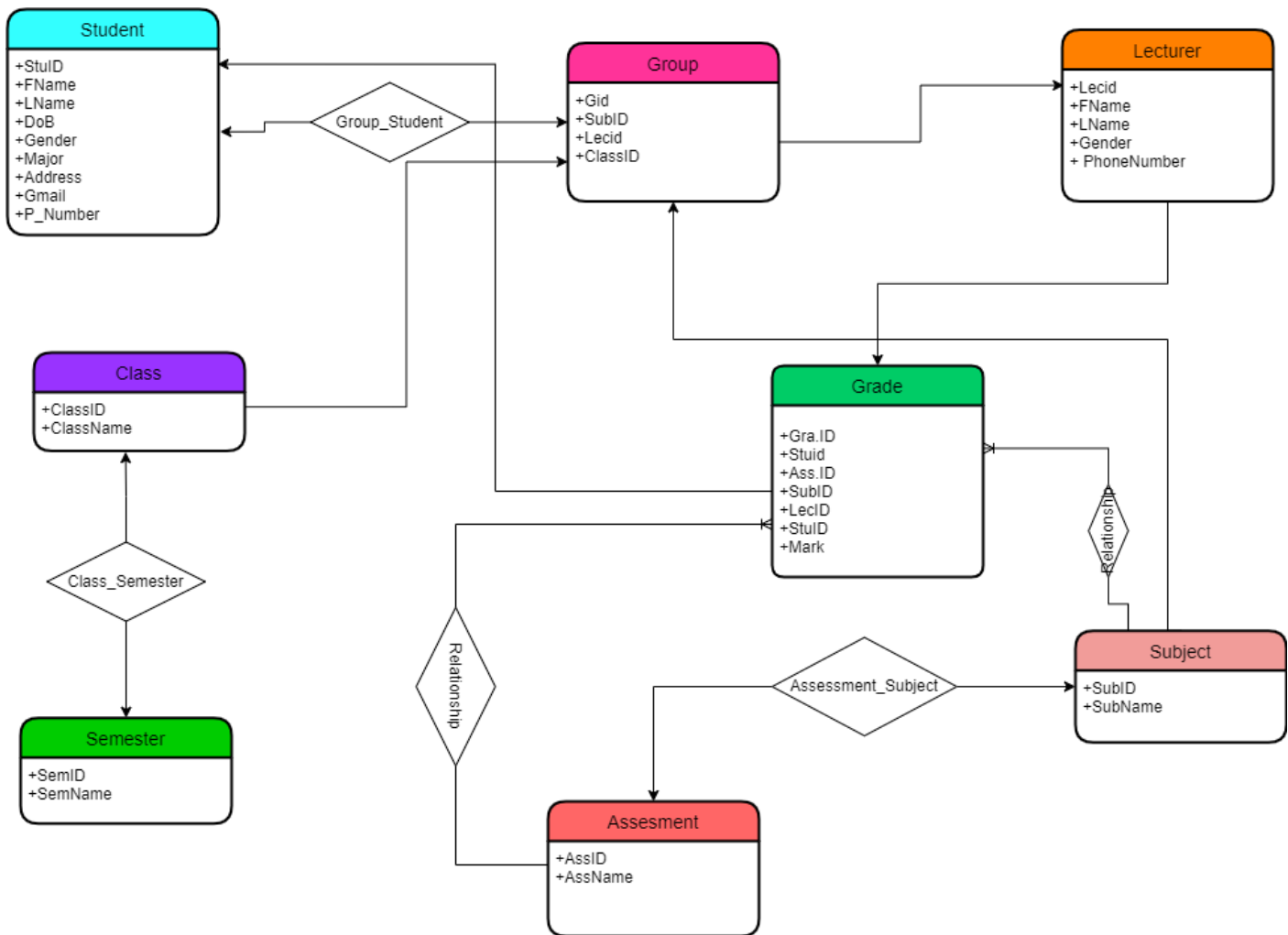
+ Category là hệ quả của việc tách Type khỏi Assessment .

+ Sau khi chuẩn hóa thuộc tính của Category có:

TypeID: Mã loại hình kiểm tra.

Description: Mô tả

III. Diagram:



IV. Chuyển Đổi

Phân Tách Các Quan Hệ < n-n > Và Tạo Table

1. STUDENTS VÀ GROUPS

Quan hệ nhiều nhiều được biểu hiện qua "Group_Student"

-> Tách và Tạo TABLE

TABLE_STUDENTS <-> TABLE_Group_Student(GrID, StudentID) <-> TABLE_GROUPS.

2. CLASSES VÀ Semester

Quan hệ nhiều nhiều được biểu hiện qua "Class_Semester"

-> Tách và Tạo TABLE

TABLE_CLASSES <-> TABLE_Class_Semester(ClassID, SemID) <-> TABLE_Semester

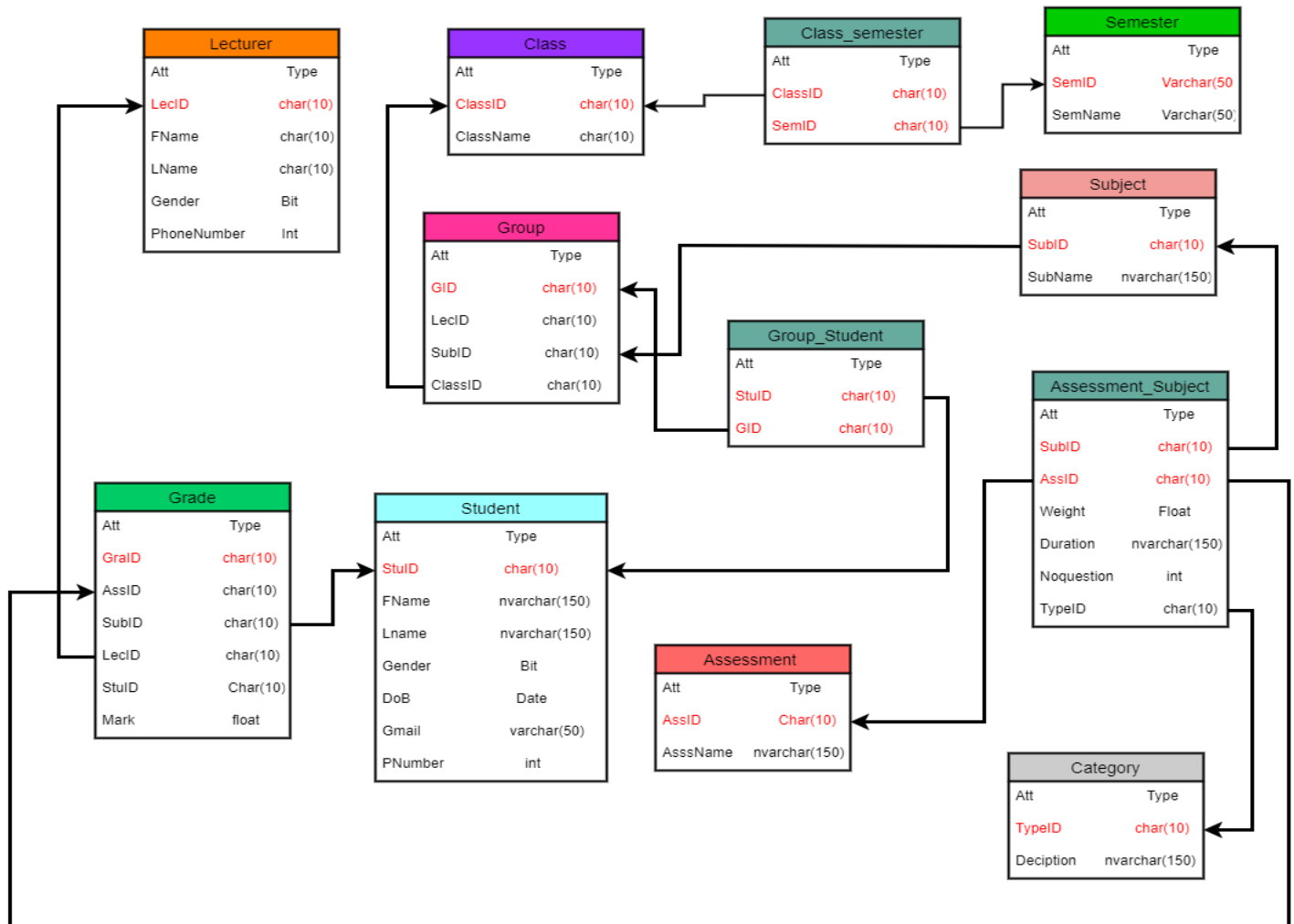
3.Assessment và Subject

Quan hệ nhiều nhiều được biểu hiện qua "Assessment_Subject"

-> Tách và Tạo TABLE

TABLE_Assessment<-> TABLE_Assessment_Subject(SuID, AssID) <-> TABLE_Subject

FAP STUDENT GRADING MANAGEMENT SUBSYSTEMS - ENTITY RELATIONSHIP DIAGRAM



VI.

1.Student

Student	
PK	<u>StuID</u>
	FName
	LName
	Major
	Gender

❖ StuID(Primary key) :

- + Type: varchar(50)
- + Syntax: HE/HA/HS + YX

Trong đó :

- HE/HA/HS là mã ứng với các khối ngành chính (CNTT,Kinh tế,Ngôn ngữ).
- Y thể hiện niên khóa của sv đó tương ứng với năm nhập học
- X là mã số(stt) của sv khi nhập học(không giới hạn mã số phụ thuộc vào số lượng tuyển sinh của mỗi năm).

+Note: not null

Eg: HE161665, HA151596, HS181652

❖ FName(First name):

+Type: NVARCHAR(50)

+Họ của sv(có thể viết tiếng Việt có dấu)

+Note: not null

❖ LName(Last name):

+Type: NVARCHAR(50)

+Tên đệm + tên sv(có thể viết tiếng Việt có dấu)

+Note: not null

❖ Major: Chuyên ngành của sv(IA, AI, SE,...)

❖ Gender:

+Type: bit - 1(nam), 0(nữ)

+Note: not null

❖ PhoneNumber:

+Type: varchar(10) - giới hạn sđt có 10 chữ số

+Note: có thể null

2.Lecturer

Lecturer	
PK	<u>LecID</u>
	FName
	LName
	Gender
	PhoneNumber

- ❖ LecID(Primary key):
 - Type: VARCHAR(50)
 - Note: not null
 - Eg: Sonnt5, Annt79, DuchM29,.....
- ❖ FName(First name):
 - Type: NVARCHAR(50)
 - Họ của giảng viên(có thể viết tiếng Việt có dấu)
 - Note: not null
- ❖ LName(Last name):
 - Type: NVARCHAR(50)
 - Tên đệm + tên giảng viên(có thể viết tiếng Việt có dấu)
 - Note: not null
- ❖ Gender:
 - Type: bit - 1(nam), 0(nữ)
 - Note: not null
- ❖ PhoneNumber:
 - +Type: varchar(10) - giới hạn sđt có 10 chữ số

+Note: có thể null

3.Class

Class	
PK	<u>ClassID</u>
	ClassName

❖ ClassID: Mã lớp học

- Type: VARCHAR(50)
- Syntax: XYZ

Trong đó:

- X: mã ngành(IA, IS, AI...)
- Y: khóa học(16,15,17..)
- Z: stt lớp(04,03,05,...)
- Eg: IA1604, AI1604, IS1706

❖ ClassName: tên của lớp học cũng như mã ngành

- Type: varchar(150)
- Ví dụ:
 - IA(Information Assurance)
 - SE(Software Engineering)

4.Subject

Subject	
PK	<u>SubID</u>
	SubName

❖ SubID(Primary key): Đây là mã ký hiệu của môn học của một lớp trong một kỳ

- Type: VARCHAR(50)
- dữ liệu có dạng : xxxyyy
 - Trong đó xxx là mã ký hiệu viết tắt của môn học
 - yyy là mã môn học

Ví dụ : CSD201 có ký hiệu là CSD và mã môn là 201

❖ SubName: Tên đầy đủ của môn học có chứa mã ký hiệu SubID

- Type: VARCHAR(150)
- Ví dụ :
 - DBI202 (Database Introduction)
 - JPD113(Elementary Japanese 1-A1.1)

5.Group

Group	
PK	<u>GID</u>
	LecID
	ClassID
	SubID

❖ GID(Primary key):

- Type: VARCHAR(50)
- Syntax: tên lớp + tên môn học(mã môn)
- Eg: IA1604-CSD, IA1604-DBI, AI1601-Lab

❖ LecID():

- Mã giảng viên theo từng môn
- Type: VARCHAR(50)
- Foreign key từ bảng Lecturer

❖ SubID(Foreign key):

- Type: VARCHAR(50)
- Foreign key từ bảng Subject

6.Group_Student

Group_Student	
PK,FK1	<u>StuID</u>
PK,FK2	<u>GID</u>

là bảng mới được tạo ra như là hệ quả của việc tách multiple values attributes từ 2 bảng Group và Student

❖ StuID(Primary key,Foreign key):

- Type: VARCHAR(50)
- Foreign key từ bảng Student

❖ GID(Primary key,Foreign key):

- Type: VARCHAR(50)
- Foreign key từ bảng group

7.Semester

Semester	
PK	<u>SemID</u>
	SemName

❖ SemID(Primary key):

➤ Type: VARCHAR(50)

➤ Syntax: XY

■ X: là kí hiệu của kỳ học(SPR,SUM,FALL) tương ứng với mùa diễn ra kỳ học trong năm đó

■ Y: là 2 số cuối năm học (22, 21,...)

❖ SemName:

➤ Type: VARCHAR(150)

➤ Là tên đầy đủ của kỳ học + năm diễn ra kỳ học (Summer 2022, Fall 2021, Spring 2019,...)

8.Class_Semester

Class_Semester	
PK,FK1	<u>ClassID</u>
PK,FK2	<u>SemID</u>

là bảng mới được tạo ra như là hệ quả của việc tách multiple values attributes từ 2 bảng Class và Semester

❖ ClassID(Primary key,Foreign key):

➤ Type: VARCHAR(50)

➤ Foreign key từ bảng Class

❖ SemID(Primary key,Foreign key):

- Type: VARCHAR(50)
- Foreign key từ bảng Semester

9.Assessment

Assesment	
PK	<u>AssID</u>
	AssName

- ❖ AssID(Primary key): là ký hiệu các đầu điểm của từng môn học. Mã này là unique đối với mỗi hệ đánh giá và cũng chính là primary key phân định các attributes khác của bảng
 - Type: VARCHAR(50)
 - Eg: PT, ASS,Labs, PE, FE.....
- ❖ AssName: Tên của các đầu điểm cho từng môn học.Để giải thích rõ hơn về AssID
 - Type: VARCHAR(50)
 - Eg: Progress Test, Assignment, Labs, Practical exam, Final exam...

10.Category

Category	
PK	<u>TypeID</u>
	Description

- ❖ TypeID(Primary key):
 - Type: VARCHAR(50)
 - Thể hiện hình thức bài thi của từng đầu điểm

trong từng môn học

➤ Eg: Quiz, On-going, Final exam, practical exam,...

❖ Description:

➤ Type: VARCHAR(150)

11.Assessment_Subject

Assesment_Subject	
PK,FK1	<u>SubID</u>
PK,FK2	<u>AssID</u>
	Weight
	Duration
	TypeID
	No Questions

là bảng mới được tạo ra như là hệ quả của việc tách multiple values attributes từ 2 bảng Assessment và Subject

❖ SubID(Primary key,Foreign key) :

- Type: VARCHAR(50)
- Foreign key từ bảng Subject

❖ AssID

- Type: VARCHAR(50)
- Foreign key từ bảng Assessment

❖ Weight

- Type: VARCHAR(50)
- Thể hiện tỉ lệ của mỗi đầu điểm trong từng môn học

❖ Duration

- Type: VARCHAR(50)
- Thể hiện thời lượng làm bài thi của từng đầu điểm

❖ TypeID(Foreign key):

- Type: VARCHAR(50)
- Foreign key từ bảng Category

12. Grade

Grade	
PK	<u>GradeID</u>
	StuID
	LecID
	SubID
	AssID
	Mark

❖ GradeID(Primary key):

- Type: VARCHAR(50)
-

❖ StuID:

- Type: VARCHAR(50)
- Foreign key từ bảng Student

❖ LecID:

- Type: VARCHAR(50)
- Foreign key từ bảng Lecturer

❖ SubID:

- Type: VARCHAR(50)
- Foreign key từ bảng Subject

❖ AssID:

- Type: VARCHAR(50)
- Foreign key từ bảng Assessment

❖ Mark:

- Type: float
- Thể hiện điểm của từng bài kiểm tra trong từng môn học của sinh viên

VI. **Tạo Database và insert dữ liệu:**

1.Tạo Database và các bảng:

```
CREATE TABLE Student(  
StuID VARCHAR(50) NOT NULL PRIMARY KEY,  
FName NVARCHAR(50) NOT NULL,  
LName NVARCHAR(50) NOT NULL,  
Major VARCHAR(50) NOT NULL,  
Gender bit NOT NULL,
```

PhoneNumber VARCHAR(50) NOT NULL,

);

--

CREATE TABLE Lecturer(

LID VARCHAR(50) NOT NULL PRIMARY KEY,

FName NVARCHAR(50) NOT NULL,

LName NVARCHAR(50) NOT NULL,

[Address] NVARCHAR(150) NOT NULL,

Gender BIT NOT NULL,

PhoneNumber int NOT NULL,

);

--

CREATE TABLE [Subject](

SubID VARCHAR(50) NOT NULL PRIMARY KEY,

SubName VARCHAR(150) NOT NULL,

);

--

CREATE TABLE Semester(

SemID VARCHAR(50) NOT NULL PRIMARY KEY,

SemName VARCHAR(150) NOT NULL,

);

CREATE TABLE Class(

ClassID VARCHAR(50) NOT NULL PRIMARY KEY ,

ClassName VARCHAR(50) NOT NULL,

);

drop table CLASS_Semester

CREATE TABLE CLASS_Semester(

ClassID VARCHAR(50) NOT NULL ,

SemID VARCHAR(50) NOT NULL,

CONSTRAINT pk_CLASS_Semester PRIMARY KEY (ClassID,SemID),

CONSTRAINT fk_CLASS_Semester1 FOREIGN KEY (ClassID) REFERENCES Class(ClassID),

CONSTRAINT fk_CLASS_Semester2 FOREIGN KEY (SemID) REFERENCES Semester(SemID),

);

--

Drop table [Group]

CREATE TABLE [Group](

GID VARCHAR(50) NOT NULL PRIMARY KEY ,

LID VARCHAR(50) NOT NULL,

SubID VARCHAR(50) NOT NULL,

ClassID VARCHAR(50) NOT NULL,

```
CONSTRAINT fk_Group FOREIGN KEY (LID) REFERENCES Lecturer(LID),

CONSTRAINT fk_Group1 FOREIGN KEY (SubID) REFERENCES [Subject](SubID),

CONSTRAINT fk_group2 FOREIGN KEY (ClassID) REFERENCES Class(ClassID),

);
```

```
--
```

```
CREATE TABLE Group_Student(

StuID VARCHAR(50) NOT NULL,

GID VARCHAR(50) NOT NULL,


CONSTRAINT pk_Group_Student PRIMARY KEY (StuID, GID),

CONSTRAINT fk_Student1 FOREIGN KEY (StuID) REFERENCES Student(StuID),

CONSTRAINT fk_Group1 FOREIGN KEY (GID) REFERENCES [Group](GID),

);
```

```
drop table Group_Student
```

```
--
```

```
CREATE TABLE Assessment(

AssID VARCHAR(50) NOT NULL PRIMARY KEY,

[Type] VARCHAR(150) NOT NULL,

Part INT NOT NULL,

[Weight] INT NOT NULL,

[Completion Criteria] INT NOT NULL,
```


[Knowledge and Skill] VARCHAR(150) NOT NULL,

NOTE VARCHAR,

);

--

CREATE TABLE Assessment_Subject(

SubID VARCHAR(50) NOT NULL,

AssID VARCHAR(50) NOT NULL,

[Weight] INT NOT NULL,

TypeID varchar(50),

Duration VARCHAR NOT NULL,

CONSTRAINT pk_Assessment_Subject PRIMARY KEY (SubID, AssID),

CONSTRAINT fk_Subject FOREIGN KEY (SubID) REFERENCES [Subject](SubID),

CONSTRAINT fk_Assesment FOREIGN KEY (AssID) REFERENCES Assessment(AssID),

);

--

CREATE TABLE Grade(

StuID VARCHAR(50) NOT NULL,

SubID VARCHAR(50) NOT NULL,

AssID VARCHAR(50) NOT NULL,

LID VARCHAR(50) NOT NULL,

Mark FLOAT NOT NULL,

CONSTRAINT Check_Mark CHECK (Mark >= 0 and Mark <= 10),

CONSTRAINT pk_Grade PRIMARY KEY (StuID, SubID),

CONSTRAINT fk_Grade1 FOREIGN KEY (StuID) REFERENCES Student(StuID),

CONSTRAINT fk_Grade2 FOREIGN KEY (SubID) REFERENCES [Subject](SubID),

CONSTRAINT fk_Grade3 FOREIGN KEY (AssID) REFERENCES Assesment(AssID),

CONSTRAINT fk_Grade4 FOREIGN KEY (LID) REFERENCES Lecturer(LID),

);

CREATE TABLE Student_Class(

StuID VARCHAR(50) NOT NULL,

GID VARCHAR(50) NOT NULL,

CONSTRAINT pk_Student_Class PRIMARY KEY (StuID, GID),

CONSTRAINT fk_Student_Class FOREIGN KEY (StuID) REFERENCES Student(StuID),

CONSTRAINT fk_Group1 FOREIGN KEY (GID) REFERENCES [Group](GID),

);

CREATE table Category(

TypeID **varchar**(50) **PRIMARY KEY**,

Description **varchar**(150),

);

2.Insert dữ liệu

```
INSERT [dbo].[Assesment_Subject] ([SubID], [AssID], [Weight], [TypeID], [Duration]) VALUES  
(N'CSD201', N'ASM', 0.2, N'On-going', N'at home')
```

```
INSERT [dbo].[Assesment_Subject] ([SubID], [AssID], [Weight], [TypeID], [Duration]) VALUES  
(N'CSD201', N'FE', 0.3, N'FE', N'60''')
```

```
INSERT [dbo].[Assesment_Subject] ([SubID], [AssID], [Weight], [TypeID], [Duration]) VALUES  
(N'CSD201', N'Labs', 0.1, N'On-going', N'at home')
```

```
INSERT [dbo].[Assesment_Subject] ([SubID], [AssID], [Weight], [TypeID], [Duration]) VALUES  
(N'CSD201', N'PE', 0.3, N'PE', N'85''')
```

```
INSERT [dbo].[Assesment_Subject] ([SubID], [AssID], [Weight], [TypeID], [Duration]) VALUES  
(N'CSD201', N'PT', 0.1, N'Quiz', N'20''')
```

```
INSERT [dbo].[Assesment_Subject] ([SubID], [AssID], [Weight], [TypeID], [Duration]) VALUES  
(N'DBI202', N'ASM', 0.2, N'On-going', N'at home')
```

INSERT [dbo].[Assesment_Subject] ([SubID], [AssID], [Weight], [TypeID], [Duration]) VALUES
(N'DBI202', N'FE', 0.3, N'FE', N'60''')

INSERT [dbo].[Assesment_Subject] ([SubID], [AssID], [Weight], [TypeID], [Duration]) VALUES
(N'DBI202', N'Labs', 0.1, N'On-going', N'at home')

INSERT [dbo].[Assesment_Subject] ([SubID], [AssID], [Weight], [TypeID], [Duration]) VALUES
(N'DBI202', N'PE', 0.3, N'PE', N'85''')

INSERT [dbo].[Assesment_Subject] ([SubID], [AssID], [Weight], [TypeID], [Duration]) VALUES
(N'DBI202', N'PT', 0.1, N'On-going', N'20''')

INSERT [dbo].[Assesment_Subject] ([SubID], [AssID], [Weight], [TypeID], [Duration]) VALUES
(N'IAO202', N'FE', 0.3, N'FE', N'85''')

INSERT [dbo].[Assesment_Subject] ([SubID], [AssID], [Weight], [TypeID], [Duration]) VALUES
(N'IAO202', N'Labs', 0.4, N'On-going', N'at home')

INSERT [dbo].[Assesment_Subject] ([SubID], [AssID], [Weight], [TypeID], [Duration]) VALUES
(N'IAO202', N'Parti', 0.1, N'On-going', N'at home')

INSERT [dbo].[Assesment_Subject] ([SubID], [AssID], [Weight], [TypeID], [Duration]) VALUES
(N'IAO202', N'PT', 0.2, N'Quiz', N'30''')

INSERT [dbo].[Assesment_Subject] ([SubID], [AssID], [Weight], [TypeID], [Duration]) VALUES
(N'JPD113', N'FE', 0.3, N'FE', N'85''')

```
INSERT [dbo].[Assesment_Subject] ([SubID], [AssID], [Weight], [TypeID], [Duration]) VALUES  
(N'JPD113', N'MT', 0.3, N'On-going', N'30''')
```

```
INSERT [dbo].[Assesment_Subject] ([SubID], [AssID], [Weight], [TypeID], [Duration]) VALUES  
(N'JPD113', N'Parti', 0.1, N'On-going', N'at home')
```

```
INSERT [dbo].[Assesment_Subject] ([SubID], [AssID], [Weight], [TypeID], [Duration]) VALUES  
(N'JPD113', N'PT', 0.1, N'Quiz', N'20''')
```

```
INSERT [dbo].[Assessment] ([AssID], [Completion Criteria], [Knowledge and Skill], [NOTE])  
VALUES (N'ASM', N'>0', N'Simple', NULL)
```

```
INSERT [dbo].[Assessment] ([AssID], [Completion Criteria], [Knowledge and Skill], [NOTE])  
VALUES (N'FE', N'>4', N'dsajfsdj', NULL)
```

```
INSERT [dbo].[Assessment] ([AssID], [Completion Criteria], [Knowledge and Skill], [NOTE])  
VALUES (N'Labs', N'>0', N'sdfkasdf', NULL)
```

```
INSERT [dbo].[Assessment] ([AssID], [Completion Criteria], [Knowledge and Skill], [NOTE])  
VALUES (N'MT', N'>0', N'sadfd', NULL)
```

```
INSERT [dbo].[Assessment] ([AssID], [Completion Criteria], [Knowledge and Skill], [NOTE])  
VALUES (N'Parti', N'>0', N'adsjf', NULL)
```

```
INSERT [dbo].[Assessment] ([AssID], [Completion Criteria], [Knowledge and Skill], [NOTE])  
VALUES (N'PE', N'>0', N'sadjfksldjf', NULL)
```

```
INSERT [dbo].[Assessment] ([AssID], [Completion Criteria], [Knowledge and Skill], [NOTE])  
VALUES (N'PT', N'>0', N'up to 04 covered chapters', NULL)
```

```
INSERT [dbo].[Category] ([TypeID], [Decription]) VALUES (N'FE', N'Final Exam')
```

```
INSERT [dbo].[Category] ([TypeID], [Decription]) VALUES (N'Labs', N'OOP with Java Lab')
```

```
INSERT [dbo].[Category] ([TypeID], [Decription]) VALUES (N'On-going', N'Progress Test')
```

```
INSERT [dbo].[Category] ([TypeID], [Decription]) VALUES (N'PE', N'Practice Exam')
```

```
INSERT [dbo].[Category] ([TypeID], [Decription]) VALUES (N'Quiz', N'Quiz')
```

```
INSERT [dbo].[Class] ([ClassID], [ClassName]) VALUES (N'IA1604', N'IA1604  
)
```

```
INSERT [dbo].[Class] ([ClassID], [ClassName]) VALUES (N'IA1605', N'IA1605  
)
```

```
INSERT [dbo].[Class] ([ClassID], [ClassName]) VALUES (N'SE1623', N'SE1623  
)
```

```
INSERT [dbo].[Class] ([ClassID], [ClassName]) VALUES (N'SE1653', N'SE1653  
)
```

```
INSERT [dbo].[CLASS_Semester] ([ClassID], [SemID]) VALUES (N'IA1604', N'FALL21')
```

INSERT [dbo].[CLASS_Semester] ([ClassID], [SemID]) VALUES (N'IA1604', N'SUM22')

INSERT [dbo].[CLASS_Semester] ([ClassID], [SemID]) VALUES (N'IA1605', N'FALL22')

INSERT [dbo].[CLASS_Semester] ([ClassID], [SemID]) VALUES (N'IA1605', N'SUM21')

INSERT [dbo].[CLASS_Semester] ([ClassID], [SemID]) VALUES (N'SE1623', N'FALL22')

INSERT [dbo].[CLASS_Semester] ([ClassID], [SemID]) VALUES (N'SE1623', N'SPR22')

INSERT [dbo].[CLASS_Semester] ([ClassID], [SemID]) VALUES (N'SE1653', N'SPR21')

INSERT [dbo].[CLASS_Semester] ([ClassID], [SemID]) VALUES (N'SE1653', N'SUM22')

INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (1,
N'HE130425', N'DBI202', N'PT', N'Sonnt5', 7)

INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (2,
N'HE130425', N'DBI202', N'ASM', N'Sonnt5', 6)

INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (3,
N'HE130425', N'DBI202', N'Labs', N'Sonnt5', 8)

INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (4,
N'HE130425', N'DBI202', N'PE', N'Sonnt5', 6)

```
INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (5,  
N'HE130425', N'DBI202', N'FE', N'Sonnt5', 5)
```

```
INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (6,  
N'HE130608', N'DBI202', N'ASM', N'Sonnt5', 7)
```

```
INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (7,  
N'HE130608', N'DBI202', N'Labs', N'Sonnt5', 7)
```

```
INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (8,  
N'HE130608', N'DBI202', N'PE', N'Sonnt5', 7)
```

```
INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (9,  
N'HE130608', N'DBI202', N'FE', N'Sonnt5', 8)
```

```
INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (10,  
N'HE160136', N'DBI202', N'PT', N'Sonnt5', 8)
```

```
INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (11,  
N'HE160136', N'DBI202', N'ASM', N'Sonnt5', 7)
```

```
INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (12,  
N'HE160136', N'CSD201', N'PT', N'Annt79', 9)
```

```
INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (13,  
N'HE160136', N'CSD201', N'ASM', N'Annt79', 10)
```


INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (15, N'HE130608', N'CSD201', N'PT', N'Annt79', 10)

INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (16, N'HE160307', N'JPD113', N'PT', N'VanDT', 9)

INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (17, N'HE130872', N'DBI202', N'PT', N'Sonnt5', 6)

INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (18, N'HE130872', N'DBI202', N'ASM', N'Sonnt5', 7)

INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (19, N'HE130872', N'DBI202', N'Labs', N'Sonnt5', 6)

INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (20, N'HE130872', N'DBI202', N'FE', N'Sonnt5', 8)

INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (21, N'HE160307', N'DBI202', N'PT', N'Sonnt5', 8)

INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (22, N'HE160307', N'CSD201', N'PT', N'Annt79', 10)

INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (23, N'HE160307', N'CSD201', N'ASM', N'Annt79', 10)

INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (24, N'HE160307', N'DBI202', N'ASM', N'Annt79', 9.5)

INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (25, N'HE160307', N'IAO202', N'PT', N'DucHM29', 10)

INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (26, N'HE160307', N'IAO202', N'Labs', N'DucHM29', 10)

INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (27, N'HE160136', N'DBI202', N'Labs', N'Sonnt5', 8)

INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (28, N'HE160136', N'DBI202', N'ASM', N'Sonnt5', 8)

INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (29, N'HE160136', N'DBI202', N'PE', N'Sonnt5', 7)

INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (30, N'HE160136', N'DBI202', N'FE', N'Sonnt5', 8)

INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (31, N'HE130872', N'DBI202', N'PE', N'Sonnt5', 7)

INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (32, N'HE160307', N'DBI202', N'Labs', N'Sonnt5', 8)

```
INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES (33,  
N'HE160307', N'DBI202', N'PE', N'Sonnt5', 7)
```

```
INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES  
(34, N'HE160307', N'DBI202', N'FE', N'Sonnt5', 8)
```

```
INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES  
(35, N'HE160307', N'CSD201', N'Labs', N'Annt79', 9)
```

```
INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES  
(36, N'HE160307', N'CSD201', N'PE', N'Annt79', 7)
```

```
INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES  
(37, N'HE160307', N'CSD201', N'FE', N'Annt79', 6)
```

```
INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES  
(38, N'HE160307', N'IAO202', N'FE', N'DucHM29', 8)
```

```
INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES  
(39, N'HE160307', N'IAO202', N'Parti', N'DucHM29', 9)
```

```
INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES  
(40, N'HE160307', N'JPD113', N'Parti', N'VanDT', 8)
```

```
INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES  
(41, N'HE160307', N'JPD113', N'MT', N'VanDT', 7)
```

```
INSERT [dbo].[Grade] ([GradeID], [StuID], [SubID], [AssID], [LID], [Mark]) VALUES  
(42, N'HE160307', N'JPD113', N'FE', N'VanDT', 8)
```

```
INSERT [dbo].[Group] ([GID], [LID], [SubID], [ClassID], [SemID]) VALUES  
(N'IA1604-CSD', N'Annt79', N'CSD201', N'IA1604', N'SUM22')
```

```
INSERT [dbo].[Group] ([GID], [LID], [SubID], [ClassID], [SemID]) VALUES  
(N'IA1604-DBI', N'Sonnt5', N'DBI202', N'IA1604', N'SUM22')
```

```

INSERT [dbo].[Group] ([GID], [LID], [SubID], [ClassID], [SemID]) VALUES
(N'IA1604-IAO', N'DucHM29', N'IAO202', N'IA1604', N'SUM22')
INSERT [dbo].[Group] ([GID], [LID], [SubID], [ClassID], [SemID]) VALUES
(N'IA1605_CSD', N'TienTD17', N'CSD201', N'IA1605', N'SUM22')
INSERT [dbo].[Group] ([GID], [LID], [SubID], [ClassID], [SemID]) VALUES
(N'IA1605_DBI', N'KhuongPD', N'DBI202', N'IA1605', N'FALL22')
INSERT [dbo].[Group] ([GID], [LID], [SubID], [ClassID], [SemID]) VALUES
(N'SE1623_LAB211', N'NangNTH', N'LAB211', N'SE1623', N'SUM21')
INSERT [dbo].[Group] ([GID], [LID], [SubID], [ClassID], [SemID]) VALUES
(N'SE1653_JPD', N'VanDT', N'JPD113', N'SE1653', N'SUM21')
INSERT [dbo].[Lecturer] ([LID], [FName], [LName], [Address], [Gender],
[PhoneNumber]) VALUES (N'Annt79', N'Nguyễn', N'Tân Ân', N'', 1, NULL)
INSERT [dbo].[Lecturer] ([LID], [FName], [LName], [Address], [Gender],
[PhoneNumber]) VALUES (N'DucHM29', N'Hoàng', N'Mạnh Đức', N'', 1, NULL)
INSERT [dbo].[Lecturer] ([LID], [FName], [LName], [Address], [Gender],
[PhoneNumber]) VALUES (N'Hoakt', N'Khuất', N'Thị Hoa', N'', 0, NULL)
INSERT [dbo].[Lecturer] ([LID], [FName], [LName], [Address], [Gender],
[PhoneNumber]) VALUES (N'KhuongPD', N'Phùng', N'Duy Khương', N'', 1,
NULL)
INSERT [dbo].[Lecturer] ([LID], [FName], [LName], [Address], [Gender],
[PhoneNumber]) VALUES (N'NangNTH', N'Nguyễn', N'Thị Hải Năng', N'', 0, NULL)
INSERT [dbo].[Lecturer] ([LID], [FName], [LName], [Address], [Gender],
[PhoneNumber]) VALUES (N'Sonnt5', N'Ngô', N'Tùng Sơn', N'', 1, NULL)
INSERT [dbo].[Lecturer] ([LID], [FName], [LName], [Address], [Gender],
[PhoneNumber]) VALUES (N'TienTD17', N'Tạ', N'Đình Tiến', N'', 1, NULL)

```

```

INSERT [dbo].[Lecturer] ([LID], [FName], [LName], [Address], [Gender],
[PhoneNumber]) VALUES (N'TungH9', N'Hoàng', N'Tùng', N'a', 1, NULL)
INSERT [dbo].[Lecturer] ([LID], [FName], [LName], [Address], [Gender],
[PhoneNumber]) VALUES (N'VanDT', N'Đỗ', N'Thị Vân', N'', 0, NULL)
GO
INSERT [dbo].[Semester] ([SemID], [SemName], [Year]) VALUES (N'FALL21',
N'Fall202', NULL)
INSERT [dbo].[Semester] ([SemID], [SemName], [Year]) VALUES (N'FALL22',
N'Fall2022', NULL)
INSERT [dbo].[Semester] ([SemID], [SemName], [Year]) VALUES (N'SPR21',
N'Spring2021', NULL)
INSERT [dbo].[Semester] ([SemID], [SemName], [Year]) VALUES (N'SPR22',
N'Spring2022', NULL)
INSERT [dbo].[Semester] ([SemID], [SemName], [Year]) VALUES (N'SUM21',
N'Summer2021', NULL)
INSERT [dbo].[Semester] ([SemID], [SemName], [Year]) VALUES (N'SUM22',
N'Summer2022', NULL)
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'    SE05692', N'Nguyễn ', N'Hữu Đức', N'SE', N'0163888245')
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE130025', N'Nguyễn ', N'Xuân Hải', N'SE', N'0163888254')
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE130139', N'Đào ', N'Hoài Nam', N'SE', N'0163885687')
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE130425', N'Vũ', N'Quốc Phong', N'SE', N'0163845687')

```

```
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE130608', N'Dương', N'Tùng Lâm', N'SE', N'0163457521')
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE130872', N'Nguyễn', N'Tiến Đạt', N'SE', N'0165787548')
```

```
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE130919', N'Phan ', N'Việt Bách', N'SE', N'0163845873')
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE130920', N'Nguyễn ', N'Văn Linh', N'SE', N'0163875466')
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE13100', N'Nguyễn ', N'Văn Thanh', N'SE', N'0164853421')
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE140272', N'Bùi ', N'Duy Hải', N'SE', N'0163884567')
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE140399', N'Hồ ', N'Quốc Bảo', N'SE', NULL)
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE140585', N'Nguyễn', N'Hải Long', N'SE', NULL)
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE140797', N'Nguyễn', N'Hoài Nam', N'SE', NULL)
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE141153', N'Lê ', N'Thị Tiểu Mai', N'SE', NULL)
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE150837', N'Đinh ', N'Công Lập', N'SE', NULL)
```

```

INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE151170', N'Trinh ', N'Minh Hùng', N'SE', NULL)
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE153561', N'Mai', N'Hoàng Anh', N'SE', NULL)
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE153622', N'Nguyễn ', N'Duy Hưng', N'SE', NULL)
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE160136', N'Đỗ', N'Văn Chung', N'IA', NULL)
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE160307', N'Nguyễn ', N'Văn Đức', N'IA', NULL)
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE160318', N'Lê', N'Việt Anh', N'IA', NULL)
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE160594', N'Hà', N'Mạnh Đạt', N'IA', NULL)
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE160666', N'Tạ', N'Việt Nam', N'IA', NULL)
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE160866', N'Vũ ', N'Đức Huy', N'SE', NULL)
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE161106', N'Châu', N'Ngọc Đức', N'IA', NULL)
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE161116', N'Đỗ ', N'Đức Anh', N'IA', NULL)
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE161198', N'Nguyễn ', N'Nhật Huy', N'IA', NULL)

```

```

INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE161260', N'Trần ', N'Công Doãn Đạt', N'SE', NULL)
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE161665', N'Trần', N'Thị Thanh Tâm', N'IA', NULL)
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE161753', N'Nguyễn', N'Thành Đạt', N'AI', NULL)
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE161777', N'Trần', N'Hải Hà', N'IA', NULL)
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE161803', N'Đỗ', N'Minh Quân', N'IA', NULL)
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE161844', N'Nguyễn ', N'Ngọc Dương', N'SE', NULL)
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE161890', N'Nguyễn ', N'Doanh Thịnh', N'IA', NULL)
GO
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE163135', N'Nguyễn', N'Tùng Dương', N'SE', NULL)
GO
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE163238', N'Vũ', N'Tuấn Anh', N'SE', NULL)
GO
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE163252', N'Phạm', N'Thị Ngọc Mai', N'SE', NULL)
GO

```



```

INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE163374', N'Triệu ', N'Đức Thành', N'IA', NULL)
GO
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE163469', N'Vũ', N'Đức Minh', N'SE', NULL)
GO
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE163474', N'Lê ', N'Khắc Minh Hiếu', N'SE', NULL)
GO
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE163581', N'Nguyễn', N'Văn Vương', N'IA', NULL)
GO
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE163589', N'Trương', N'Hoàng Minh', N'SE', NULL)
GO
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE163682', N'Đào', N'Vũ Hiệp ', N'IA', NULL)
GO
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE163693', N'Đào', N'Vũ Hiệp ', N'IA', NULL)
GO
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])
VALUES (N'HE163742', N'Nguyễn', N'Cảnh Thương', N'IA', NULL)
GO

```

```
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])  
VALUES (N'HE163767', N'Nguyễn ', N'Quang Huy', N'SE', NULL)
```

GO

```
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])  
VALUES (N'HE163790', N'Nguyễn ', N'Nhật Nam', N'IA', NULL)
```

GO

```
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])  
VALUES (N'HE163833', N'Nguyễn', N'Văn Quang', N'SE', NULL)
```

GO

```
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])  
VALUES (N'HE163856', N'Trinh ', N'Tùng Sơn', N'IA', NULL)
```

GO

```
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])  
VALUES (N'HE163868', N'Phạm', N'Hồng Đăng', N'IA', NULL)
```

GO

```
INSERT [dbo].[Student] ([StuID], [FName], [LName], [Major], [phoneNumber])  
VALUES (N'HE163963', N'Võ ', N'Viết Mạnh', N'IA', NULL)
```

```
INSERT Student([StuID], [FName], [LName], [Major], [phoneNumber]) VALUES  
(N' SE05600', N'A ', N'Nguyen Huu ', N'SE', N'01638245')
```

```
INSERT Student([StuID], [FName], [LName], [Major], [phoneNumber]) VALUES  
(N' SE05601', N'B ', N'Nguyen Huu ', N'SE', N'015648245')
```

```
INSERT Student([StuID], [FName], [LName], [Major], [phoneNumber]) VALUES  
(N' IA05602', N'C ', N'Nguyen Huu ', N'SE', N'0163458245')
```

```

INSERT Student([StuID], [FName], [LName], [Major], [phoneNumber]) VALUES
(N' SE05603', N'D ', N'Nguyen Huu ', N'SE', N'0164568245')
INSERT Student([StuID], [FName], [LName], [Major], [phoneNumber]) VALUES
(N' SE05604', N'E ', N'Nguyen Huu ', N'SE', N'0645368245')
INSERT Student([StuID], [FName], [LName], [Major], [phoneNumber]) VALUES
(N' IA05605', N'F ', N'Nguyen Huu ', N'SE', N'0145488245')
INSERT Student([StuID], [FName], [LName], [Major], [phoneNumber]) VALUES
(N' SE05606', N'G ', N'Nguyen Huu ', N'SE', N'0164533245')
INSERT Student([StuID], [FName], [LName], [Major], [phoneNumber]) VALUES
(N' SE05607', N'H ', N'Nguyen Huu ', N'SE', N'0164548245')
INSERT Student([StuID], [FName], [LName], [Major], [phoneNumber]) VALUES
(N' IA05608', N'I ', N'Nguyen Huu ', N'SE', N'0164545245')
INSERT Student([StuID], [FName], [LName], [Major], [phoneNumber]) VALUES
(N' SE05609', N'K ', N'Nguyen Huu ', N'SE', N'0333888245')
INSERT Student([StuID], [FName], [LName], [Major], [phoneNumber]) VALUES
(N' IA05610', N'L ', N'Nguyen Huu ', N'SE', N'0264564556')
INSERT Student([StuID], [FName], [LName], [Major], [phoneNumber]) VALUES
(N' SE05611', N'M ', N'Nguyen Huu ', N'SE', N'0456432455')
INSERT Student([StuID], [FName], [LName], [Major], [phoneNumber]) VALUES
(N' IA05612', N'N ', N'Nguyen Huu ', N'SE', N'0456453453')
INSERT Student([StuID], [FName], [LName], [Major], [phoneNumber]) VALUES
(N' SE05613', N'O ', N'Nguyen Huu ', N'SE', N'0486435345')
INSERT Student([StuID], [FName], [LName], [Major], [phoneNumber]) VALUES
(N' SE05614', N'P ', N'Nguyen Huu ', N'SE', N'0434856434')

```

```
INSERT Student([StuID], [FName], [LName], [Major], [phoneNumber]) VALUES  
(N' SE05615', N'Q ', N'Nguyen Huu ', N'SE', N'0456478374')
```

```
INSERT Student([StuID], [FName], [LName], [Major], [phoneNumber]) VALUES  
(N' SE05616', N'R ', N'Nguyen Huu ', N'SE', N'0348674848')
```

```
INSERT Student([StuID], [FName], [LName], [Major], [phoneNumber]) VALUES  
(N' SE05617', N'S ', N'Nguyen Huu ', N'SE', N'0148345355')
```

```
INSERT Student([StuID], [FName], [LName], [Major], [phoneNumber]) VALUES  
(N' SE05618', N'T ', N'Nguyen Huu ', N'SE', N'0163888245')
```

```
INSERT Student([StuID], [FName], [LName], [Major], [phoneNumber]) VALUES  
(N' SE05619', N'U ', N'Nguyen Huu ', N'SE', N'0163888245')
```

GO

```
INSERT [dbo].[Student_Group] ([StuID], [GID]) VALUES (N' SE05692', N'IA1604-  
CSD')
```

GO

```
INSERT [dbo].[Student_Group] ([StuID], [GID]) VALUES (N' SE05692',  
N'SE1653_JPD')
```

GO

```
INSERT [dbo].[Student_Group] ([StuID], [GID]) VALUES (N'HE130025', N'IA1604-  
CSD')
```

GO

```
INSERT [dbo].[Student_Group] ([StuID], [GID]) VALUES (N'HE130025',  
N'SE1653_JPD')
```

GO

```
INSERT [dbo].[Student_Group] ([StuID], [GID]) VALUES (N'HE130139', N'IA1604-  
CSD')  
GO  
INSERT [dbo].[Student_Group] ([StuID], [GID]) VALUES (N'HE130139',  
N'SE1623_LAB211')  
GO  
INSERT [dbo].[Student_Group] ([StuID], [GID]) VALUES (N'HE130425', N'IA1604-  
CSD')  
GO  
INSERT [dbo].[Student_Group] ([StuID], [GID]) VALUES (N'HE130872',  
N'SE1623_LAB211')  
GO  
INSERT [dbo].[Student_Group] ([StuID], [GID]) VALUES (N'HE140797', N'IA1604-  
IAO')  
GO  
INSERT [dbo].[Student_Group] ([StuID], [GID]) VALUES (N'HE141153', N'IA1604-  
IAO')  
GO  
INSERT [dbo].[Student_Group] ([StuID], [GID]) VALUES (N'HE150837', N'IA1604-  
IAO')  
GO  
INSERT [dbo].[Student_Group] ([StuID], [GID]) VALUES (N'HE160136', N'IA1604-  
CSD')  
GO
```

```
INSERT [dbo].[Student_Group] ([StuID], [GID]) VALUES (N'HE160136', N'IA1604-DBI')
```

```
GO
```

```
INSERT [dbo].[Student_Group] ([StuID], [GID]) VALUES (N'HE160307', N'IA1604-CSD')
```

```
GO
```

```
INSERT [dbo].[Student_Group] ([StuID], [GID]) VALUES (N'HE160307', N'IA1604-DBI')
```

```
GO
```

```
INSERT [dbo].[Student_Group] ([StuID], [GID]) VALUES (N'HE160318', N'IA1604-CSD')
```

```
GO
```

```
INSERT [dbo].[Student_Group] ([StuID], [GID]) VALUES (N'HE160318', N'IA1604-DBI')
```

```
GO
```

```
INSERT [dbo].[Student_Group] ([StuID], [GID]) VALUES (N'HE160594', N'IA1604-CSD')
```

```
GO
```

```
INSERT [dbo].[Student_Group] ([StuID], [GID]) VALUES (N'HE160594', N'IA1604-DBI')
```

```
GO
```

```
INSERT [dbo].[Student_Group] ([StuID], [GID]) VALUES (N'HE160666', N'IA1604-CSD')
```

```
GO
```

```
INSERT [dbo].[Student_Group] ([StuID], [GID]) VALUES (N'HE160666', N'IA1604-DBI')
GO
INSERT [dbo].[Subject] ([SubID], [SubName]) VALUES (N'CSD201', N'Data Structures and Algorithms')
INSERT [dbo].[Subject] ([SubID], [SubName]) VALUES (N'DBI202', N'Introduction to Databases')
INSERT [dbo].[Subject] ([SubID], [SubName]) VALUES (N'IAO202', N'Introduction to Information Assurance')
INSERT [dbo].[Subject] ([SubID], [SubName]) VALUES (N'JPD113', N'Elementary Japanese 1-A1.1')
INSERT [dbo].[Subject] ([SubID], [SubName]) VALUES (N'LAB211', N'OOP with Java Lab')
INSERT [dbo].[Subject] ([SubID], [SubName]) VALUES (N'MAD101', N'Discrete mathematics')
INSERT [dbo].[Subject] ([SubID], [SubName]) VALUES (N'NWC204', N'Computer Networking')
INSERT [dbo].[Subject] ([SubID], [SubName]) VALUES (N'OSG202', N'Operating Systems')
INSERT [dbo].[Subject] ([SubID], [SubName]) VALUES (N'PRO192', N'Object-Oriented Programming')
INSERT [dbo].[Subject] ([SubID], [SubName]) VALUES (N'SSG104', N'Communication and In-Group Working Skills')
```

VII. Query

1. Query dùng ORDER BY.

```
Select * from Student order by Major
```

Query này sẽ sắp xếp các sinh viên theo chuyên ngành .Kết quả trả ra gồm các column của table Student.

2. Query dùng INNER JOIN.

```
Select st.StuID, (FName + ' '+LName) as FullName, g.SubID, g.AssID, Mark from  
Student st inner join Grade g on st.StuID = g.StuID  
inner join Assesment_Subject sub on g.SubID =  
sub.SubID and sub.AssID = g.AssID  
where Mark > 9 AND g.SubID = 'CSD201' AND g.AssID = 'PT'
```

Đây là query để lấy ra các sinh viên có điểm pt môn CSD > 9

3. Query dùng Aggregate function.

```
Select cl.ClassID, COUNT (cs.SemID) as [Number of semester] from  
Class cl inner join Class_Semester cs on cl.ClassID = cs.ClassID  
inner join Semester se on cs.SemID = se.SemID  
group by cl.ClassID
```

Query này để đếm và trả ra số kỳ học mà lớp đấy đã trải qua. Kết Quả gồm 2 column mã lớp học và số kỳ đã trải qua của lớp đó.

4. Query dùng GROUP BY và HAVING.

```
Select st.StuID, g.SubID, COUNT (Mark) from  
Grade g inner join Student st on g.StuID = st.StuID  
inner join Assesment_Subject asu on g.AssID = asu.AssID and g.SubID =  
asu.SubID  
group by st.StuID, g.SubID  
having COUNT (Mark) > 1
```

Đếm số sinh viên đã làm nhiều hơn một bài kiểm tra của mỗi môn học

5. Query sử dụng sub-query như 1 quan hệ.

```
select * from
```



```
(select AVG(mark) as AverageMark from Grade where SubID = 'DBI202' and AssID = 'PT' )
as avg_DBI,Grade as g
where g.Mark > avg_DBI.AverageMark and g.SubID = 'DBI202' and g.AssID = 'PT'
```

Đưa ra những sv có số điểm 1 bài kiểm tra lớn hơn điểm trung bình của tất cả (Môn DBI202)

6. Query sử dụng sub-query ở mệnh đề WHERE:

```
select (lt.FName + ' ' +lt.LName)as FullName, gr.ClassID, gr.SubID,gr.SemID from
  [Group] gr inner join Lecturer lt on gr.LID = lt.LID
  --inner join Class cl on cl.ClassID = gr.ClassID
where SemID like '%21'
```

Đưa ra tên gv , lớp và môn học mà gv đó dạy trong năm 2021

7. Query sử dụng partial matching trong mệnh đề WHERE:

8. Query sử dụng self-join:

9. Query ngẫu nhiên 1:

10. Query ngẫu nhiên 2:

