

QUẢN LÝ ĐIỂM

MỤC LỤC

I.DỰ ĐOÁN ENTITIES



- | | |
|----|---|
| A. | Các đối tượng cơ bản chắc chắn có trong hệ thống điểm |
| B. | Xác định 1 số yêu cầu đặc biệt của đề bài |
| 1. | Requirement1 (pic 1) |
| 2. | Requirement2 (pic 2) |
| 3. | Requirement3 (pic 3) |
| C. | Các đối tượng cơ bản chắc chắn có trong hệ thống điểm |

II.ENTITY RELATIONSHIP DIAGRAM



III. CHUẨN HÓA



IV. MODULE DIAGRAM



V. PHÂN TÍCH MODULE DIAGRAM



VI. TẠO DATABASE VÀ INSERT DỮ LIỆU



VIII. QUERIES



I.Xác Định Object có thể Trong DataBase.

A.CÁC ĐỐI TƯỢNG CƠ BẢN CHẮC CHẮN 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
- Assetment : 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.

B.Xác định Một Số Yêu cầu đặc biệt của đề bài.

[Type here]

1. REQUIREMENT 1 (PIC 1)

5 assessment(s)

Category	Type	Part	Weight	Completion Criteria	Duration	L.O.	Question Type	No Question	Knowledge and Skill	Grading Guide	Note
Progress Tests	quiz	2	10.0%	>0	20'		Multiple choices Marked by Computer or a suitable format	20	up to 94 covered chapters	by instructor using computer	Instruction and schedules for Progress tests must be presented in the Course Implementation Plan approved by director of the campus. Progress test must be taken right after the last lectures of required material. Instructor has responsibility to review the test for students after graded.
Assignment	on-going	1	20.0%	>0	at home		Design, implementation, Presentation		Simple RDBS design and implementation using a DBMS	guided by instructor, prepare at home present in class	40% Design, 20% Implementation, 40% Presentation of the whole Project
Lab	on-going	5	15.0%	>0	in lab session		practical exercises		related to studied modules	Guided by instructor	may be continued at home.
Practical Exam	practical exam	1	25.0%	>0	85'		Preferable to be marked by Scripts		DB programming skills	by exam board and department	Practical Exam database is up load in CMS in advanced.
Final Exam	final exam	1	30.0%	5	60'		Multiple choices Marked by Computer	60	Knowledge and skills in the course, but with much focus on the items in Chapters 2 to 6, >= 70% new questions (for the current semester);	by exam board	

Activate Windows
Go to Settings to activate Windows.

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:

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

[Type here]

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)

NO.	SUBJECT CODE	SUBJECT NAME	SEMESTER	GROUP	STARTDATE	ENDDATE	AVERAGE MARK	STATUS
1	SSL101c	Academic Skills for University Success	Spring2021					Not Passed
2	SSG103	Communication and In-Group Working Skills	Summer2021					Passed
3	NWC203c	Computer Networking	Summer2021					Passed
4	CEA201	Computer Organization and Architecture	Spring2021					Passed
5	MAD101	Discrete mathematics	Summer2021					Passed
6	JPD113	Elementary Japanese 1-A1.1	Fall2021					Passed
7	CSI104	Introduction to Computer Science	Spring2021					Passed
8	DBI202	Introduction to Databases	Fall2021					Not Passed
9	LUK1	Level 1	Fall2019					Passed
10	LUK2	Level 2	Spring2020					Passed
11	LUK3	Level 3	Spring2020					Passed
12	LUK4	Level 4	Summer2020					Pass (with conditions)
13	LUK5	Level 5	Summer2020					Passed
14	LUK6	Level 6	Fall2020					Passed
15	MAE101	Mathematics for Engineering	Spring2021					Passed
16	GDQP	Military training	Fall2019					Passed
17	PRO192	Object-Oriented Programming	Fall2021					Passed
18	PRO192	Object-Oriented Programming	Fall2021					Not Passed
19	OSG202	Operating Systems	Summer2021					Passed
20	PRF192	Programming Fundamentals	Summer2021					Not Passed
21	PRF192	Programming Fundamentals	Spring2021					Attendance Fail
22	ĐTB102	Traditional musical instrument	Summer2020					Passed
23	VOV114	Vovinam 1	Fall2019					Passed
24	VOV124	Vovinam 2	Summer2020					Passed
25	VOV134	Vovinam 3	Summer2020					Passed

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

Bảng bao gồm các cột:

- ❖ No: Số thứ tự
- ❖ Subject Code: Mã số của môn học.
- ❖ Subject Name: Tên của môn học.
- ❖ Semester: Kì học mà sinh viên đã tham gia môn học tương ứng.
- ❖ Group: Nhóm sinh viên mà sinh viên này đã tham gia học cùng môn học tương ứng.

[Type here]

- ❖ Start Date: Ngày bắt đầu của môn học tương ứng.
- ❖ End Date: Ngày kết thúc của môn học tương ứng.
- ❖ Average Mark: Điểm trung bình của sinh viên ở môn học tương ứng.
- ❖ 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)

GRADE CATEGORY	GRADE ITEM	WEIGHT	VALUE	COMMENT
Quiz 2	Quiz 2	7.0 %	7.8	
	Total	7.0 %	7.8	
Quiz 1	Quiz 1	8.0 %	7.6	
	Total	8.0 %	7.6	
Activity	Activity	10.0 %	8.5	
	Total	10.0 %	8.5	
Group Assignment	Group Assignment	15.0 %	9	
	Total	15.0 %	9	
Group Project	Group Project	30.0 %	8.3	
	Total	30.0 %	8.3	
Final Exam	Final Exam	30.0 %	8.6	
	Total	30.0 %	8.6	
Final Exam Resit	Final Exam Resit	30.0 %		
	Total	30.0 %		
COURSE TOTAL	AVERAGE	8.4		
	STATUS	PASSED		

RQ3: Với mỗi môn học, Student có thể kiểm tra chi tiết kết quả từng Assesment
Bảng gồm 5 cột:

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

❖ 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 .

C.XÁC ĐỊNH MỐI QUAN HỆ GIỮA CÁC ĐỐI TƯỢNG

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

- Student \leftrightarrow 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 \leftrightarrow 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 \leftrightarrow 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 \leftrightarrow 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 \leftrightarrow 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 \leftrightarrow 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 mỗi quan hệ giữa Subject và Assessment là một quan hệ N – N .
- Student \leftrightarrow Grade: Mỗi quan hệ chủ đạo của hệ thống đánh giá điểm chính là Điểm và Sinh viên . (Ban đầu em để Grade là bảng được tách ra từ mỗi quan hệ N – N giữa Student và Assessment , nhưng sau vì thấy 1 bài kiểm tra vd như progress test trong 1 số môn học sẽ có 2 bài . Vì nếu để là pt1 , pt2 thì không hợp lí vì như bài lab tùy môn sẽ có rất nhiều lab , sẽ không tối ưu nếu để lab15 mà chỉ có 1 môn dùng. Vì vậy e sẽ để tất cả pt1 , pt2 là pt và như thế trong bảng Grade sẽ có 2 record là student1 , pt tương ứng vs pt1 và pt2. Tuy nhiên như vậy thì AssesID sẽ ko thể là 1 phân khóa chính trong bảng Grade □ Chuyển Grade từ bảng tách ra từ mỗi quan hệ nhiều nhiều thành một đối tượng -Entity và có GradeID là khóa chính) 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 \leftrightarrow Assessment và Grade \leftrightarrow Subject (Ban đầu e để Grade sẽ có mỗi quan hệ 1 – N với lần lượt cả Assessment và Subject Như đã nói ở phần B – RQ1 .Với việc một Assessment vd progress test -pt sẽ khác nhau phụ thuộc vào mỗi môn học , và mỗi môn học sẽ có những Assessment khác nhau nên nếu để Grade quan hệ 1—N vs cả Assessment và Subject sẽ có khả năng nhầm lẫn khi 1 record trong Grade sẽ có 1 điểm số của Assessment nhưng Assessment đó lại ko nằm trong môn học tương ứng) Để đảm bảo hạn chế lỗi khi insert dữ liệu, Grade sẽ tạo mỗi quan hệ trực tiếp với bảng Assessment _ Subject được tách ra từ mỗi 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 \leftrightarrow Category. Như đã phân tích ở Phần B – RQ1. Chúng ta cần tách attribute Type ra 1 bảng riêng. (Ban đầu e để Category là

1 Entity có quan hệ N – N với Assessment , tuy nhiên về sau khi chuyển Grade tạo quan hệ trực tiếp với Assessment _ Subject , đồng thời cũng bởi Assessment trong Subject khác nhau sẽ có kiểu Type khác nhau, nghĩa là Assessment phụ thuộc rất lớn vào Subject thì e quyết định chuyển luôn Category tạo quan hệ trực tiếp với Assessment _ Subject thay vì Assessment và nó sẽ chuyển thành quan hệ 1—N) 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.
- Kỳ học mà nhóm đã tham gia học tập.
- Lớp học theo kỳ.
- Các môn học mà nhóm đã tham gia.
- 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.

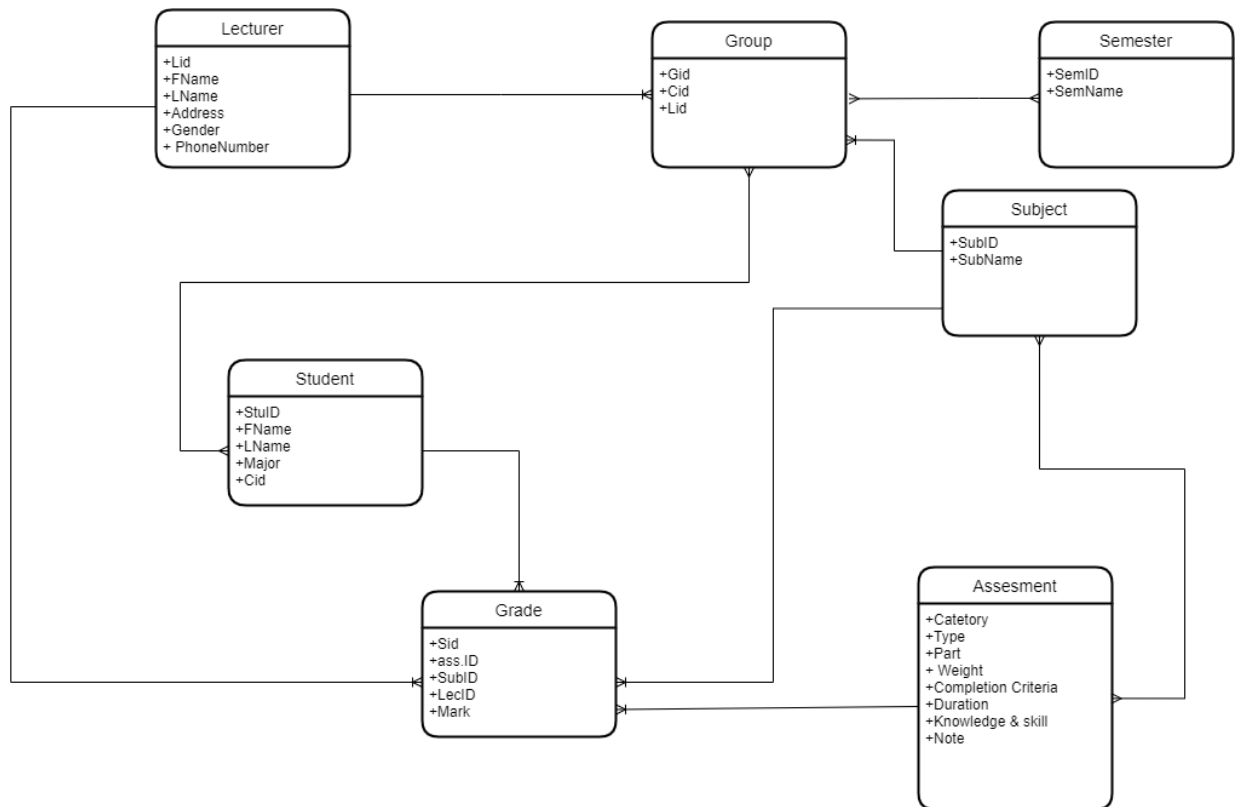
Grade: Các attributes của tables Grade là các thông tin cơ bản của 1 loại đầu điểm.

- 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. Diagram:



III. 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) → ko đạt 1NF → 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

→ 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
→ 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.

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

→ 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 → 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 → 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

[Type here]

+ 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

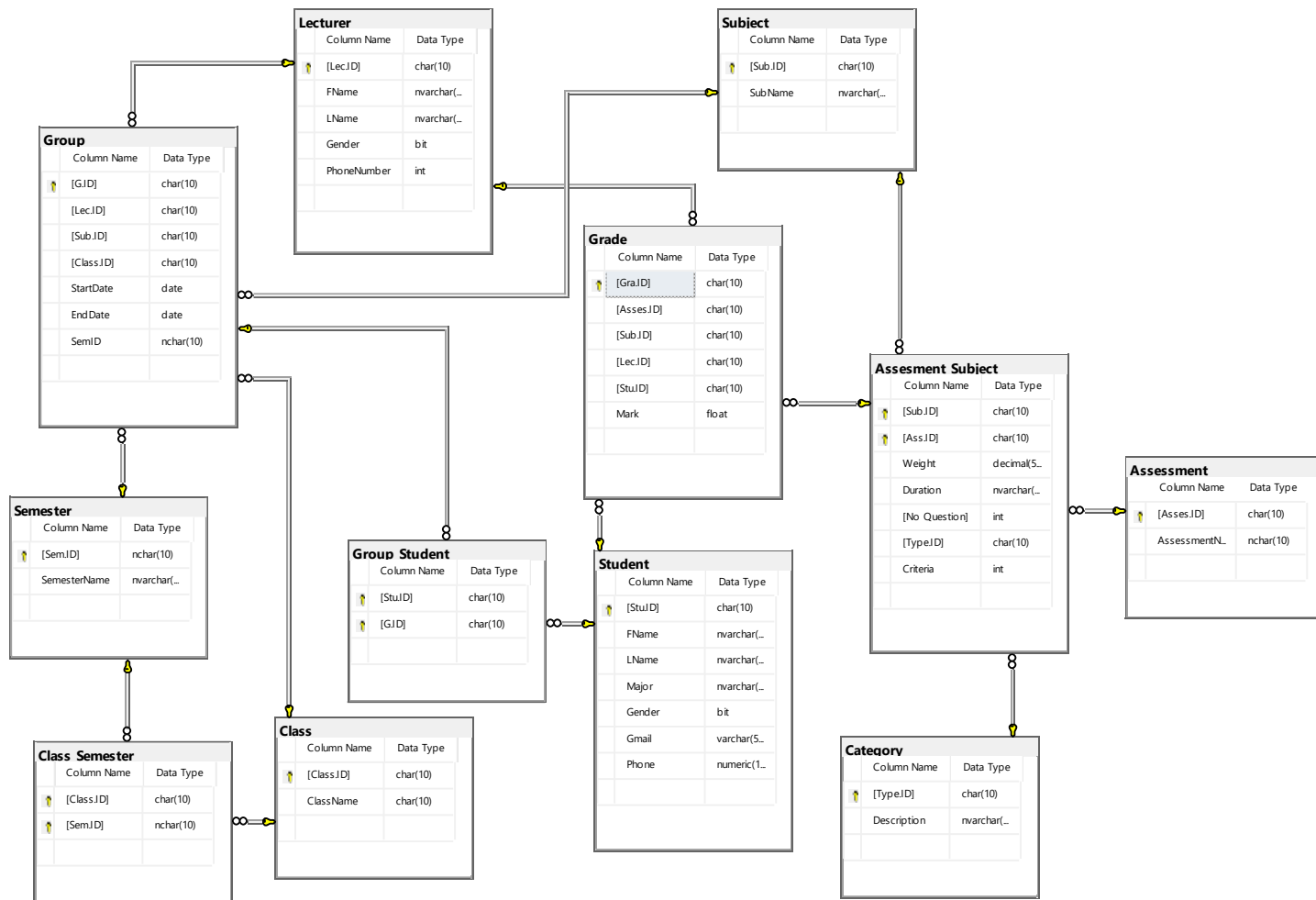
+ Category là hệ quả của việc tách Type khỏi Assessment .

+ Sau khi chuẩn hóa thuộc tính của Category có:

TypeID: Mã loại hình kiểm tra.

Description: Mô tả

IV. MODULE DIAGRAM



IV. Phân tích diagram

1.Student

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

- ❖ StuID(Primary key) :
- ❖ Type: varchar(50)

[Type here]

❖ 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 sdt 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

[Type here]

- ❖ SubID:
 - Type: VARCHAR(50)
 - Foreign key từ bảng Subjec
- ❖ 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. Student

```
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,
);
```

2. Lecturer

```
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,
);
```

3. Subject

```
CREATE TABLE [Subject](
  SubID VARCHAR(50) NOT NULL PRIMARY KEY,
  SubName VARCHAR(150) NOT NULL,
```

);

4.Semester

```
CREATE TABLE Semester(
SemID VARCHAR(50) NOT NULL PRIMARY KEY,
SemName VARCHAR(150) NOT NULL,
);
```

5.Class

```
CREATE TABLE Class(
ClassID VARCHAR(50) NOT NULL PRIMARY KEY ,
ClassName VARCHAR(50) NOT NULL,
);
```

6.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),
);
```

7. 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),
);

```

8. Group Student

```

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),
);

```

9. Assessment

```

CREATE TABLE Assessment(
AssID VARCHAR(50) NOT NULL PRIMARY KEY,
[Completion Criteria] INT NOT NULL,
[Knowledge and Skill] VARCHAR(150) NOT NULL,
NOTE VARCHAR,
);

```

10. Assessment Subject

```

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_Assesment_Subject PRIMARY KEY
(SubID, AssID),
CONSTRAINT fk_Subject FOREIGN KEY (SubID)
REFERENCES [Subject](SubID),
CONSTRAINT fk_Assesment FOREIGN KEY (AssID)
REFERENCES Assesment(AssID),
);

```

11. Grade

```

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),
);

```

12. Category

```

CREATE table Category(
TypeID varchar(50) PRIMARY KEY,
[Description] varchar(150),
);

```

13. Insert data:

[Type here]

VII, Queries

1. Order by

-Sắp xếp sinh viên theo chuyên ngành

SELECT * FROM Student ORDER BY Major

-Result

	StuID	FName	LName	Major	phoneNumber
1	HE161753	Nguyễn	Thành Đạt	AI	NULL
2	HE161777	Trần	Hải Hà	IA	NULL
3	HE161803	Đỗ	Minh Quân	IA	NULL
4	HE161890	Nguyễn	Doanh Thịnh	IA	NULL
5	HE160136	Đỗ	Văn Chung	IA	NULL
6	HE160307	Nguyễn	Văn Đức	IA	NULL
7	HE160318	Lê	Việt Anh	IA	NULL
8	HE160594	Hà	Mạnh Đạt	IA	NULL
9	HE160666	Tạ	Việt Nam	IA	NULL
10	HE161106	Châu	Ngọc Đức	IA	NULL
11	HE161116	Đỗ	Đức Anh	IA	NULL
12	HE161198	Nguyễn	Nhật Huy	IA	NULL
13	HE163374	Triệu	Đức Thành	IA	NULL
14	HE163581	Nguyễn	Văn Vương	IA	NULL
15	HE163682	Đào	Vũ Hiệp	IA	NULL
16	HE163693	Đào	Vũ Hiệp	IA	NULL
17	HE163742	Nguyễn	Cảnh Thương	IA	NULL
18	HE163790	Nguyễn	Nhật Nam	IA	NULL
19	HE163856	Trịnh	Tùng Sơn	IA	NULL
20	HE163868	Phạm	Hồng Đăng	IA	NULL
21	HE163963	Võ	Việt Mạnh	IA	NULL
22	HE161665	Trần	Thị Thanh T...	IA	NULL
23	HE163833	Nguyễn	Văn Quang	SE	NULL
24	HE163767	Nguyễn	Quang Huy	SE	NULL
25	HE163589	Trương	Hoàng Minh	SE	NULL
26	HE163469	Vũ	Đức Minh	SE	NULL
27	HE163474	Lê	Khắc Minh ...	SE	NULL
28	HE161260	Trần	Công Doãn ...	SE	NULL
29	HE160866	Vũ	Đức Huy	SE	NULL
30	HE163135	Nguyễn	Tùng Dương	SE	NULL
31	HE163238	Vũ	Tuấn Anh	SE	NULL
32	HE163252	Phạm	Thị Ngọc Mai	SE	NULL
33	HE161844	Nguyễn	Ngọc Dương	SE	NULL
34	SE05692	Nguyễn	Hữu Đức	SE	0163888245

35	HE130025	Nguyễn	Xuân Hải	SE	0163888254
36	HE130139	Đào	Hoài Nam	SE	0163885687
37	HE130425	Vũ	Quốc Phong	SE	0163845687
38	HE130608	Dương	Tùng Lâm	SE	0163457521
39	HE130872	Nguyễn	Tiến Đạt	SE	0165787548
40	HE130919	Phan	Việt Bách	SE	0163845873
41	HE130920	Nguyễn	Văn Linh	SE	0163875466
42	HE13100	Nguyễn	Văn Thanh	SE	0164853421
43	HE140272	Bùi	Duy Hải	SE	0163884567
44	HE140399	Hồ	Quốc Bảo	SE	NULL
45	HE140585	Nguyễn	Hải Long	SE	NULL
46	HE140797	Nguyễn	Hoài Nam	SE	NULL
47	HE141153	Lê	Thị Tiểu Mai	SE	NULL
48	HE150837	Đinh	Công Lập	SE	NULL
49	HE151170	Trịnh	Minh Hùng	SE	NULL
50	HE153561	Mai	Hoàng Anh	SE	NULL
51	HE153622	Nguyễn	Duy Hùng	SE	NULL

2. Inner join

- Tên của những sv có điểm pt csd201 > 9

```
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'
```

-Result

	StuID	FullName	SubID	AssID	Mark
1	HE130608	Dương Tùng Lâm	CSD201	PT	10
2	HE160307	Nguyễn Văn Đức	CSD201	PT	10

3. Aggregate functions

- Đưa ra số kỳ học đã trải qua của 1 lớp

```
SELECT cl.ClassID, COUNT (cs.SemID) AS [Number of semester]
FROM
```

[Type here]

```

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

```

- Result

	ClassID	Number of semester
1	IA1604	2
2	IA1605	2
3	SE1623	2
4	SE1653	2

4. GROUP BY and HAVING clauses

- Đưa ra mã sv đã làm nhiều hơn 1 bài ktra với mỗi môn học

```

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

```

-Result

	StuID	SubID	(No column name)
1	HE160136	CSD201	2
2	HE160307	CSD201	5
3	HE130425	DBI202	5
4	HE130608	DBI202	4
5	HE130872	DBI202	5
6	HE160136	DBI202	6
7	HE160307	DBI202	5
8	HE160307	IAO202	4
9	HE160307	JPD113	4

5.A sub-query as a relation

- Đưa ra những sv có số điểm 1 bài kiểm tra lớn hơn điểm trung bình của tất cả, ở đây là PT của DBI202

```

SELECT * FROM
(SELECT AVG(mark) as AverageMark FROM Grade
WHERE

```

[Type here]

```

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'

```

-Result

	AverageMark	GradeID	StuID	SubID	AssID	LID	Mark
1	7.25	10	HE160136	DBI202	PT	Sonnt5	8
2	7.25	21	HE160307	DBI202	PT	Sonnt5	8

5. A sub-query in the WHERE clause

- Đưa ra những sinh viên trong những lớp đã học nhiều hơn 1 kỳ

```

SELECT DISTINCT (st.FName + ' ' + st.LName) AS FullName
FROM
    Student st INNER JOIN Student_Group sg ON st.StuID = sg.StuID
    INNER JOIN [Group] gr ON gr.GID = sg.GID
WHERE
    gr.ClassID IN
    ( SELECT cl.ClassID FROM Class cl inner join CLASS_Semester cs
      ON cl.ClassID = cs.ClassID
    group by cl.ClassID
    having COUNT (cs.ClassID) >1)

```

	FullName
1	Đào Hoài Nam
2	Đinh Công Lập
3	Đỗ Văn Chung
4	Hà Mạnh Đạt
5	Lê Thị Tiểu Mai
6	Lê Việt Anh
7	Nguyễn Hữu Đức
8	Nguyễn Văn Đức
9	Nguyễn Xuân Hải
10	Nguyễn Hoài Nam
11	Nguyễn Tiến Đạt
12	Tạ Việt Nam
13	Vũ Quốc Phong

6. A query that uses partial matching in the WHERE clause

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

```
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'
```

-Result

	FullName	ClassID	SubID	SemID
1	Nguyễn Thị Hải Năng	SE1623	LAB211	SUM21
2	Đỗ Thị Vân	SE1653	JPD113	SUM21

7. Self join

- Đưa ra thông tin của những sv có điểm cao nhất trong bài kiểm tra PT môn CSD

```
SELECT * FROM Grade
WHERE SubID = 'CSD201' AND AssID = 'PT' AND StuID NOT IN
(SELECT g1.StuID FROM Grade g1 INNER JOIN Grade g2
ON
g1.Mark < g2.Mark AND g1.SubID = g2.SubID and g1.AssID = g2.AssID
WHERE
g1.SubID = 'CSD201' AND g1.AssID = 'PT')
```

-Result

	GradeID	StuID	SubID	AssID	LID	Mark
1	15	HE130608	CSD201	PT	Annt79	10
2	22	HE160307	CSD201	PT	Annt79	10