

Student Grading Management Subsystem

Giới thiệu

1. Thông tin cá nhân:

- **Họ tên:** Nguyễn Doanh Thịnh.
- **Mã số sinh viên:** HE161890.
- **Lớp:** IA1604.
- **Khoa:** Công nghệ thông tin.
- **Chuyên ngành:** An toàn thông tin.
- **Trường:** Đại học FPT cơ sở Hòa Lạc.
- **Email:** thinhndhe161890@fpt.edu.vn.
- **Số điện thoại liên lạc:** 0976622548.

2. Về bản báo cáo:

- **Tên:** Database lưu trữ dữ liệu hệ thống tính điểm cho sinh viên trường đại học FPT.
- **Nguyên do:** 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 có lấy điểm.

- **Tóm tắt nội dung:** đưa ra các phân tích về đề bài, hình ảnh, các thông tin có thể khai thác từ các hình ảnh, tạo dựng một database cơ bản cho 1 hệ thống tính điểm và test độ ứng dụng khi đưa vào thực tiễn.

Mục lục

I. Phân tích dữ liệu.

1. Ảnh 1: Thống kê các bài test của 1 môn học.
2. Ảnh 2: Status của 1 sinh viên.
3. Ảnh 3: Bảng điểm của 1 môn học.

II. Dự đoán các entities cần thiết.

1. Entities.
2. Relationships.
3. Attributes.

III. Chuẩn hóa 3rd Normal Form đưa ra các table.

I. Phân tích đề bài, các hình ảnh:

1. Ảnh 1: Thống kê các bài test của 1 môn học.

Category	Type	Part	Weight	Completion Criteria	Duration	LO	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 04 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
Labs	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	

- Nhìn sơ qua, có thể thấy được ở ảnh này, có 1 bảng với nội dung là các loại bài test, hoặc đầu điểm mà một sinh viên phải đáp ứng của 1 môn học.
- 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.

- Nhận thấy rằng mỗi 1 môn đều có những loại bài test khác nhau (không phải 1 cơ cấu đánh giá điểm như trong hình được áp dụng cho tất cả các môn).
- Do vậy cần 1 entity là category lưu lại tất cả các loại bài test và các thông số cơ bản mà luôn áp dụng cho bài test đó.

2. Ảnh 2: Status của 1 sinh viên.

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

- Ở bảng này, sinh viên sẽ có thể xem được thông tin cơ bản của toàn bộ các môn học mà mình đã tham gia.
- Bảng bao gồm các cột:
 - No: Số hiệu / 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.

- 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).
- Thấy rằng bảng này chứa rất nhiều thông tin và có nhiều attributes có thể làm thành khóa. Điều này không tốt cho 1 table.
 - Do vậy một mình bảng này không thể chỉ được thể hiện trong 1 entity được, phải thông qua nhiều các entities nhỏ lẻ để tránh việc vi phạm vào 3rd-Normal Form Standard.

3. Ảnh 3: Bảng điểm của 1 môn học.

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		

- Ở bảng này, thấy được các đầu điểm , điểm trung bình và trạng thái của 1 môn học mà 1 sinh viên sau khi học môn học đó đã đạt được.
- 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: Một số lời nhắc hoặc chú ý thì sẽ được note vào cột này.

- Phía cuối sau khi kết thúc bảng điểm, xuất ra điểm trung bình môn và trạng thái. 2 dòng này có thể cũng là 1 attribute của entity.
- Dự đoán: Để thu được bảng này, ta cần join khá nhiều bảng khác lại với nhau, vì riêng nó khó có thể làm 1 entity mà không vi phạm 3rd-Normal Form Standard.

II. Dự đoán các entities cần thiết:

1. Nhận thấy rằng các bảng đã được phân tích bên trên đều có liên quan tới một số các entities xác định:
 - **Students** : Tất cả các bảng trên đều liên quan tới thực thể **Students** (bảng điểm là bảng điểm của 1 sinh viên, status cũng là của 1 sinh viên, phân loại đầu điểm cũng xem từ môn học mà sinh viên tham gia), mà 1 hệ thống tính điểm thì không chỉ có của riêng 1 sinh viên nào đó. Do vậy, 1 entity **Students** là chắc chắn phải có trong hệ thống này.
 - **Courses** : Một hệ thống tính điểm thì không thể thiếu các **Courses** . Sinh viên sẽ tham gia các môn học này, và đầu điểm đánh giá sẽ được dựa theo môn học tương ứng.
 - **Lecturers** : Có các môn học thì chúng ta cũng sẽ có các **Lecturers** tham gia giảng dạy các môn học đó. Giảng viên sẽ trực tiếp cho điểm đánh giá sinh viên theo môn học tương ứng. Note: **Lecturers** sẽ có 1 attribute là **Report** phục vụ cho yêu cầu viết query phía sau của đề bài.
 - **Classes** : Sinh viên sẽ tham gia các lớp học, giảng viên sẽ giảng dạy các môn học cũng theo các lớp học tương ứng. Do vậy, chúng ta cũng có 1 entity là **Classes** .
 - **Groups** : Không thể để tất cả các sinh viên học cùng 1 lớp được. Họ sẽ được phân thành các nhóm và học các lớp theo ngày, giờ khác nhau.

Các môn học ứng với từng nhóm cũng có thể được dạy bởi các giảng viên khác nhau. Do vậy, chúng ta cũng sẽ có 1 entity là **Groups**.

- **Categories**: Một môn học được đánh giá qua nhiều bài test và nhiều đầu điểm khác nhau. Các giảng viên sẽ cho điểm đánh giá sinh viên học môn học tương ứng dựa vào các loại đầu điểm này. Do vậy, cần 1 entity là **Categories**.

2. Các entities đã dự đoán bên trên khả năng cao sẽ có một số relationships sau:

- Đối với **Students** và **Groups**, các sinh viên sẽ tham gia vào các nhóm khác nhau khi dựa trên kì học, môn học và chuyên ngành, các nhóm cũng sẽ chứa nhiều sinh viên cùng các đặc điểm. Do vậy, ta đặt 2 entities này vào 1 quan hệ N—N.
- Đối với **Groups** và **Classes**, các nhóm sẽ được phân vào các lớp học với giảng viên và môn học tương ứng với kì học hiện tại của nhóm sinh viên đó. Một nhóm có thể tham gia nhiều lớp học, và 1 lớp cũng có thể có nhiều nhóm tham gia. Do vậy, ta sẽ đặt 2 entities này vào 1 quan hệ N—N.
- Đối với **Courses** và **Classes**, một lớp sẽ được phân 1 môn học và một môn học thì lại được phân vào nhiều lớp khác nhau. Do vậy, 2 entities này sẽ được đặt vào quan hệ **Courses** 1—N **Classes**.
- Đối với **Courses** và **Categories**, một môn học thì được đánh giá qua nhiều loại đầu điểm, và một loại đầu điểm thì cũng được dùng để đánh giá nhiều môn học. Do vậy, 2 entities này sẽ được đặt vào quan hệ **Courses** N—N **Categories**.
- Đối với **Lecturers** và **Classes**, tương tự như với **Courses**, một lớp thì cũng sẽ chỉ có 1 giảng viên tham gia giảng dạy chính (không tính dạy thay vì khi xét về cuối kì thì không xét đến giảng viên dạy thay) và một

giảng viên cũng được phân vào nhiều lớp để giảng dạy. Do vậy, 2 entities này cũng sẽ được đặt vào quan hệ **Lecturers** 1—N **Classes**.

- Đối với **Lecturers** và **Courses**, một giảng viên có thể dạy được nhiều môn học và một môn học cũng có thể được dạy bởi nhiều giảng viên. Phụ thuộc vào kì học, 1 giảng viên có thể được phân công dạy 1 môn ở kì này, nhưng có thể không phải dạy môn đó ở kì khác. 2 entities này cũng sẽ được đặt vào 1 quan hệ N—N.
- Đối với **Lecturers** và **Students**, giảng viên sẽ giảng dạy các môn học theo các lớp, các sinh viên sẽ tham gia vào các lớp này ứng với các nhóm được phân vào. 1 kì học, sinh viên học 5 môn có thể có tối đa 5 giảng viên giảng dạy, 1 giảng viên dạy nhiều lớp cũng gồm nhiều sinh viên theo học
- Đối với **Students** và **Courses**, học sinh khi tham gia vào các lớp theo nhóm thì cũng có quan hệ với môn học thông qua các lớp đó. Trong 1 kì, sinh viên được trường phân cho 5 lớp ứng với 5 môn, nhưng sinh viên cũng có thể đăng kí thêm hoặc đăng kí tạm hoãn. Một môn học cũng sẽ được dạy cho rất nhiều sinh viên.
- Đối với **Groups** và **Courses**, các nhóm được phân vào các lớp, và các lớp lại được giao cho các môn học riêng. Do vậy các nhóm cũng có quan hệ với môn học thông qua các lớp. trong 1 kì, các nhóm sẽ được phân cho học nhiều môn, và 1 môn cũng được đem đi dạy cho nhiều nhóm sinh viên.
- Đối với **Students** và **Classes**, sinh viên sẽ được phân vào các nhóm và các nhóm sẽ được phân vào học ở các lớp khác nhau ở mỗi kì, sinh viên sẽ có quan hệ với lớp thông qua các nhóm. Trong 1 kì, sinh viên có thể tham gia nhiều lớp, và 1 lớp cũng được phân cho nhiều nhóm có nhiều sinh viên.
- Đối với **Lecturers** và **Groups**, các giảng viên được giao cho vào các lớp học, các nhóm sinh viên học lại được phân vào học ở các lớp. Do vậy, giảng viên và các nhóm sinh viên cũng có quan hệ với các giảng viên

thông qua các lớp. Trong 1 kì, 1 nhóm sinh viên được nhiều giảng viên dạy, 1 giảng viên cũng giảng dạy cho nhiều nhóm sinh viên.

- Đối với **Lecturers** và **Lecturers**, các bộ môn sẽ có những giảng viên làm trưởng môn của môn học đó, những giảng viên dạy môn này đều phải báo cáo tính hình giảng dạy, điểm trắc của sinh viên cho trưởng môn. Do vậy, ca có quan hệ **Lecturers** 1—1 **Lecturers**

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

- **Students**: Các attributes của **Students** chính là các thông tin cơ bản của 1 sinh viên:
 - Mã số sinh viên được cấp bởi nhà trường cho 1 sinh viên.
 - Địa chỉ email mà nhà trường 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 kì học mà sinh viên tham gia.
 - Những nhóm mà sinh viên tham gia ở mỗi kì.
 - Những môn mà sinh viên học mỗi kì.
 - Loại đầu điểm ở mỗi môn học.
 - Giảng viên thực hiện giảng dạy ở môn học tương ứng.
 - Điểm của sinh viên tại mỗi đầu điểm.
- **Groups**: Các attributes của **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 sinh viên của 1 nhóm.
 - Kỳ học mà nhóm đó tham gia học tập.
 - Lớp học theo kì.
 - Môn học mà nhóm đó tham gia.
 - Điểm trung bình của môn học tương ứng.
 - Giảng viên đã giảng dạy 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.

- Kì 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 cấp.
 - Địa chỉ email mà nhà trường cấp cho giảng viên.
 - Tên gọi của 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 các tính hình giảng dạy cho trưởng bộ môn.
 - Kì học mà giảng viên phải tham gia giảng dạy.
- **Categories** : Các attributes của **Categories** 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 bì 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 kĩ năng cần thiết để có thể làm được bài test.
- Hướng dẫn cho điểm cho giảng viên.
- Chú thích của giảng viên (nếu có).

III. Chuẩn hóa 3rd Normal Form đưa ra các tables:

1. Students:

- Nhận thấy 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
 - Khi đưa vào bảng, sẽ hình thành các multiple values attributes.
 - Còn chưa đạt chuẩn 1st.
 - Phải tách hết những attributes này ra, đưa vào entities hoặc relationships khác.
- Sau khi tách bỏ hết, ta được bảng **Students** với các cột:
 - Mssv: 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.
 - First Name: Tên gọi của sinh viên.

- Last Name: Họ và tên đệm của sinh viên.
- Gender: Giới tính của sinh viên.
- DoB: Ngày tháng năm sinh của sinh viên.
- Sau khi tách hết các multiple values attributes, bảng **Students** 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.
 - Bảng **Students** đã đạt chuẩn 3rd Normal Form.

2. Groups:

- Nhận thấy 1 nhóm thì có nhiều sinh viên tham gia vào, 1 nhóm cũng tham gia nhiều kì học. Ở 1 kì học thì 1 nhóm cũng tham gia vào nhiều lớp học (5 lớp nếu theo sắp xếp của nhà trường), và nhiều lớp thì lại có nhiều môn và nhiều giảng viên.
 - Khi đưa vào bảng thì các attributes này sẽ có phụ thuộc bắc cầu với nhau.
 - Chưa đạt chuẩn 2nd.
 - Các attributes này cũng là các multiple values attributes.
 - Chưa đạt cả chuẩn 1st.
 - Phải tách hết những attributes này ra, đưa vào entities hoặc relationships khác.
- Sau khi tách bỏ hết, bảng **Groups** sẽ có các cột sau:
 - **GrID**: Mã nhận diện của 1 nhóm. Mã này cũng là primary key của bảng.
 - Major: Chuyên ngành chính của gần như toàn bộ sinh viên được xếp vào 1 nhóm.

- Number of Students: Số sinh viên đã tham gia nhóm này.
- 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.
→ Bảng **Groups** đã đạt chuẩn 3rd Normal Form.

3. Join:

- Đây là 1 bảng mới được tạo ra như là hệ quả của việc tách multiple values attributes từ 2 bảng trên.
- Một số attributes đã tách ra từ 2 bảng trên sẽ được thêm vào bảng này để mô tả quan hệ **Students** N—N **Groups**.
- Gồm các cột:
 - Mssv: Mã số sinh viên.
 - GrID: Mã nhóm mà sinh viên tương ứng đã tham gia.
- 2 cột này cũng tạo thành primary key của bảng **Join**.
- Do bảng chỉ có 2 cột và 2 cột này là primary key nên bảng này cũng đạt chuẩn 3rd Normal Form.

4. Lecturers:

- Nhận thấy 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.
→ Các attributes này cũng là các multiple values attributes.

→ Chưa đạt cả chuẩn 1st.

→ Phải tách hết những attributes này ra, đưa vào entities hoặc relationships khác.

- Sau khi tách bỏ hết, bảng **Lecturers** sẽ có các cột sau:
 - **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.
 - DoB: Ngày tháng năm sinh của giảng viên.
 - Report: Mã số nhận diện của giảng viên trưởng bộ môn của giảng viên tương ứng.
- Sau khi tách hết các multiple values attributes, bảng **Lecturers** giờ chỉ còn 1 primary key là **LecID**, không có non-key attributes nào phụ thuộc 1 phần vào key, không có non-key attributes nào phụ thuộc bắc cầu vào key.
 - Bảng **Lecturers** đã đạt chuẩn 3rd Normal Form.

5. Courses:

- Nhận thấy 1 môn học thì cần rất nhiều đầu điểm để đánh giá. 1 môn thì cũng được giảng dạy bởi nhiều giảng viên và được phân vào nhiều lớp.
 - Khi đưa vào bảng, sẽ hình thành các multiple values attributes.
 - Còn chưa đạt chuẩn 1st.

→ Phải tách hết những attributes này ra, đưa vào entities hoặc relationships khác.

- Sau khi tách bỏ hết, bảng **Courses** sẽ có các cột sau:
 - **CouID** : 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.
- Bảng **Courses** đã đạt chuẩn 3rd Normal Form.

6. Assignment:

- Bảng này lên kết **Courses** với **Lecturers** lại với nhau theo đúng quan hệ N—N
- 1 môn học thì do nhiều giảng viên dạy và 1 giảng viên cũng có thể được phân công dạy nhiều môn. Do vậy ta tạo 1 bảng phân công để thể hiện rõ những giảng viên nào được phân công dạy môn nào.
- Ở mỗi kì học, có thể giảng viên này được phân công môn này nhưng ở kì khác thì lại được phân công môn khác. Để linh hoạt trong việc tái sử dụng những phân công thì không để **Assignment** thành 1 bảng relationship được mà chuyển thành 1 entity.
- Các attributes của entity phân công này gồm có
 - Mã nhận diện 1 phân công, mã này sẽ được sử dụng ở các kì học và lớp học, thuận tiện cho việc xếp luôn cả giảng viên và môn học vào các lớp theo kì.
 - Môn học mà phân công này chỉ định.

- Giảng viên được phân công dạy môn học tương ứng.
- Khi đưa vào bảng thì **Assignment** sẽ có các cột như sau:
 - AsnID: Mã nhận diện 1 phân công, cột này chính là primary key của bảng.
 - Major: Môn học mà phân công này chỉ định.
 - Lecturer: Giảng viên được phân công dạy môn học tương ứng.
- 2 cột **Major** và **Lecturer** là 1 candidate key của bảng **Assignment**.
- Bảng này cũng toàn các key nên bản thân nó cũng đã đạt chuẩn 3rd Normal Form.

7. Classes:

- Nhận thấy rằng 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 2nd.
 - Các attributes này cũng là các multiple values attributes.
 - Chưa đạt cả chuẩn 1st.
 - Phải tách hết những attributes này ra, đưa vào entities hoặc relationships khác.
- Ngoài ra, như đã nói ở trên, chúng ta sẽ sử dụng 1 quan hệ **Classes** N —1 **Assignments**, để có thể lấy được môn học và giảng viên dạy môn học đó phân vào các lớp. 1 lớp thì chỉ có thể có 1 phân công được bổ nhiệm vào, và 1 phân công thì được giao vào nhiều lớp khác nhau.
- Sau khi tách bỏ hết, bảng **Classes** sẽ có các cột sau:

- **ClassID**: Mã số nhận diện một lớp học. Mã này cũng là unique đối với mỗi lớp học. Đây cũng chính là primary key của bảng **Classes**.
- **AsnID**: Mã phân công được giao cho lớp tương ứng.
- **Semester**: Kì học mà lớp này có được phân cho các nhóm sinh viên.
- **Start Date**: Ngày bắt đầu của kì học tương ứng.
- **End Date**: Ngày kết thúc của kì học tương ứng.
- **AsnID** và **Semester** cũng là candidate key của bảng **Classes**
- Sau khi tách hết các multiple values attributes, bảng **Classes** giờ chỉ còn 1 primary key là **ClassID**, không có non-key attributes nào phụ thuộc 1 phần vào key, không có non-key attributes nào phụ thuộc bắc cầu vào key.
→ Bảng **Classes** đã đạt chuẩn 3rd Normal Form.

8. Enroll:

- Bây giờ, để biết nhóm sinh viên nào được giao vào lớp nào tại kì học nào, chúng ta cần 1 bảng thể hiện quan hệ giữa 2 entities **Classes** N—N **Groups**.
- Các multiple values attributes mà ta tách ra từ 2 bảng trên cũng sẽ được thể hiện ở đây.
- Gồm các cột:
 - **ClassID**: Mã nhận diện lớp.
 - **GrID**: Mã nhận diện nhóm được phân vào học lớp tương ứng.
- 2 cột này cũng tạo thành primary key của bảng **Enroll**.
- Do bảng chỉ có 2 cột và 2 cột này là primary key nên bảng này cũng đạt chuẩn 3rd Normal Form.

9. Categories:

- Sau khi tách các multiple values attributes liên quan đến các đầu điểm đánh giá môn học từ bảng `Courses` thì ta sẽ đặt những attributes đó vào bảng `Categories` này.
- Tuy nhiên, tùy vào từng môn học mà việc phân loại này có thể khác nhau. Ví dụ: 1 đầu điểm ở 1 môn học này có trọng số 0.05 nhưng ở một môn khác thì sẽ có trọng số 0.1, thời gian làm các bài test ứng với từng môn học cũng khác nhau tùy theo đặc trưng yêu cầu của môn học.
 - Khi đưa vào bảng, sẽ hình thành các multiple values attributes.
 - Còn chưa đạt chuẩn 1st.
 - Phải tách hết những attributes này ra, đưa vào entities hoặc relationships khác.
- Sau khi tách bỏ hết, ta được bảng `Categories` với các cột:
 - CatID: Mã nhận diện loại điểm. Đây chính là primary key của bảng này.
 - Name: Tên gọi của bài test hoặc loại đầu điểm.
 - Type: Hình thức kiểm tra, đánh giá.
 - Completion Criteria: Điều kiện pass bài test.
- Bảng `Categories` giờ chỉ còn 1 primary key là `ClassID`, không có non-key attributes nào phụ thuộc 1 phần vào key, không có non-key attributes nào phụ thuộc bắc cầu vào key.
 - Bảng `Categories` đã đạt chuẩn 3rd Normal Form.

10. Assessment System:

- Từng môn học sẽ có hệ thống đánh giá riêng phù hợp theo các tiêu chí của khung chương trình. Môn học có thể có các loại đầu điểm khác nhau, thời lượng làm bài khác nhau, số câu hỏi cũng khác nhau, trọng số cũng có thể khác nhau.
- **Assessment System** phản ánh quan hệ của 2 entities **Courses** N—N **Categories** nên nó sẽ được đặt vào 1 bảng thể hiện được quan hệ này.
- Tuy nhiên, ta nhận thấy rằng **Assessment System** sẽ còn liên quan nhiều đến phần chấm điểm cho sinh viên của giảng viên và việc xem điểm số của mình của sinh viên. Do vậy ta đặt **Assessment System** làm 1 entity hoàn chỉnh.
- Bảng này sẽ chứa các attributes của việc đánh giá 1 môn học dựa theo loại đầu điểm tương ứng. Do vậy, bảng này sẽ gồm các cột:
 -