

HỌ TÊN ĐỖ VĂN CHUNG

LỚP IA1604

Báo cáo Assignment

1. Phân tích đối tượng

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 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	

Bảng Syllabus

- Category: Phân loại các điểm thành phần của từng môn học (tùy theo môn học Category có thể thay đổi)

VD: DBI202 có các Category như: Progress test, Assignment, labs, practice exam, final exam

- Type: Kiểu của Category
- Part: Số phần của từng Category
VD: Progress test có 2 phần
- Weight: trọng số điểm chiếm của từng Category trong một môn học tổng của chúng phải bằng 100%
- Completion Criteria điều kiện điểm của từng Category
VD: Progress test có Completion Criteria là ">0" điều này có nghĩa là số điểm của progress test bắt buộc phải lớn hơn 0.
- Duration: Thời gian làm bài
- Question type: kiểu câu hỏi, cấu trúc của bài thi
- No Question: số lượng câu hỏi, nếu Category không có từ là nó không phải dạng Multiple choice
- Knowledge and Skill: những kiến thức và yêu cầu của bài kiểm tra
- Grading guide: hướng dẫn cách thức làm bài
- Note: một số chú ý

Bảng Academic Transcript

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

Bảng Academic Trascript

- NO số thứ tự
- Subject Code: Mã môn học
- Subject Name: Tên môn học cụ thể
- Semester: kì Học
- Group: Mã lớp học
- StartDate: ngày kì học bắt đầu
- EndDate: Ngày kì học kết thúc
- Average mark Điểm trung bình của từng môn học
- Status trạng thái môn học đã qua môn hoặc bị trượt môn

Bảng Student Transcript

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	

Grade Category: Các bài thi cần phải hoàn thành

Grade Item: Điểm thành phần

Weight: Trọng số điểm chiếm trong một kì học

Value: Kết quả thi học sinh đạt được

Phân tích chung

Mục đích quản lý điểm của các học sinh

Của các học sinh nên chắc chắn chúng ta phải có một bảng để lưu các thông tin của học sinh

Vì quản lý điểm nên chúng ta cần có một bảng để lưu điểm mà học sinh đạt được

Cần một bảng để lưu các thông tin môn học số bài kiểm tra của một môn học trọng số của từng bài test

Rules

Một học sinh có thể học nhiều môn học

Học sinh khi thi cần làm rõ ai thi , thi thì thi môn nào , được bao điểm

Nên các thông tin SID, suID, mark cần được thể hiện trong một bản chính là bảng student_subject

Những Object cần có

- Student: Vì quản lý điểm học sinh lên chúng ta cần một đối tượng Student để lưu các thông tin của học sinh như mã sinh viên, full name giới tính, ngày sinh.
- Subject: Để lưu các thông tin của môn học như mã môn học tên môn học cụ thể.
- Group
- Semester kì học thời gian học
- Result: kết quả thi của các học sinh
- Category để lưu các thông tin của môn học

Chia Object

Student (sID, sName, sGender, DOB)

sID Là mã sinh viên có format VD: HE160136 yêu cầu Mã sinh viên phải là duy nhất trên toàn hệ thống nên sID là primary key

sName tên của sinh viên vì tên sinh viên là tiếng việt có dấu nên kiểu dữ liệu là nvarchar

sGender giới tính của sinh viên kiểu dữ liệu là bit với giá trị là 1 ứng với giới tính nam con 0 ứng với giới tính nữ

DOB: Date of birth Ngày sinh ngày sinh của sinh viên ngày sinh phải được chuẩn hóa theo format

(YYYY-mm--DD) ví dụ 2022-1-14

Subject(sulD, suName)

Bảng Subject lưu các thông tin của các môn học

sulD: mã môn học duy nhất trên toàn hệ thống được set làm primary key trong bảng subject

suName: tên đầy đủ của môn học dựa vào sulD để xác định tên môn học

Group(gID)

gID: Mã lớp học

semester (seID, seName, startDate, endDate)

seID: mã kì học VD: sp2022, su2022, fa2022 với 2 kí tự đầu là mùa kì, con 4 chữ số là năm học

Vì theo cách tổ chức học của trường thì một năm có 3 mùa học chia ra làm 3 kì học ứng với số năm duy nhất nên seID được SET làm khóa chính.

seName: tên kì học VD: Spring 2022, Summer2022, Fall2022 seName phụ thuộc vào seID

startDate: ngày bắt đầu của một kì nó phụ thuộc vào seID.

endDate: ngày kết thúc của một kì nó phụ thuộc vào seID.

category(suID, caName, type, part, Weight, CC, note, duration,grading guide, no question, knowlesges)

Category để lưu trữ các thông tin cụ thể của một môn học.

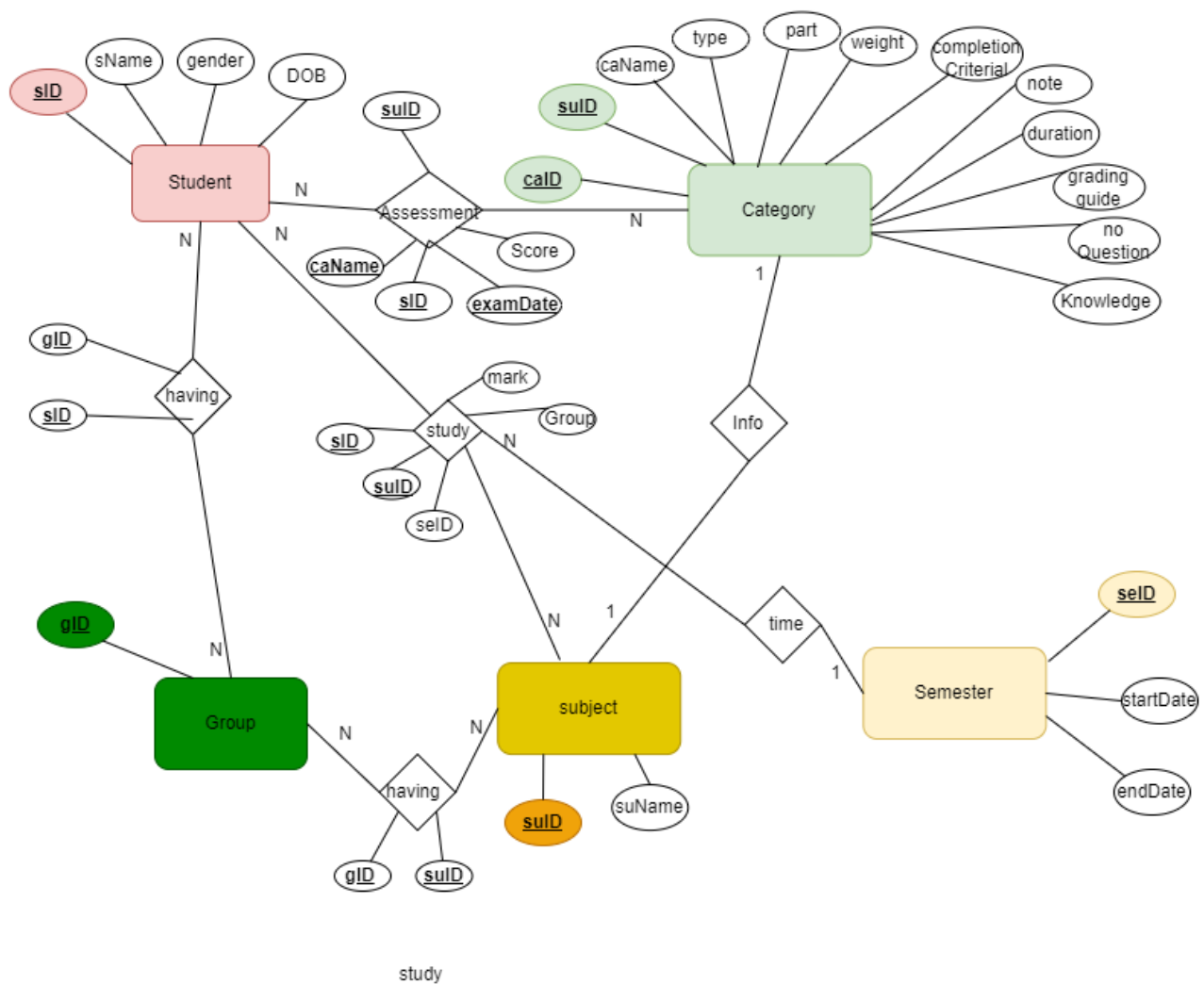
Name: tên thành phần môn học nó bao gồm những bài test, presentation để đánh giá sinh viên trong cả quá trình học tập

type: kiểu đánh giá có thể là đánh giá trong quá trình học tập, các bài test

Part số các bài đánh giá của mỗi category vì mỗi bài test có thể có một bài hoặc nhiều bài nhỏ

Weight trong số của mỗi bài test sẽ có những bài quan trọng có thể đánh giá sinh viên chiếm trọng số cao còn những chứa ít kiến thức chiếm trọng số thấp, yêu cầu quan trọng đối với weight là tổng của chúng phải là 100% không được ít hơn hoặc nhỏ hơn.

ERD DIAGRAM Student Grading Management Sub-System



Phân tích các quan hệ giữa các table

- Student – group

Bảng Student và group có quan hệ nhiều nhiều bởi vì một học sinh có thể học nhiều lớp khác nhau không bắt buộc một học sinh chỉ được học một lớp duy nhất, còn một group có thể gồm có nhiều sinh viên khác nhau nên quan hệ của chúng là nhiều

- Group – subject

Một lớp học có thể học nhiều môn cũng như môn subject cũng có thể được học bởi nhiều lớp

Nên quan hệ của hai bảng này là nhiều nhiều

Vì quan hệ nhiều nhiều nên chúng ta cần tạo thêm bảng Group_Subject

Group_Subject(gID, suID) bảng này cho biết lớp học nào thì học môn nào.

- subject – Category

Mỗi subject chỉ có một category duy nhất bởi vì với mỗi môn học khác nhau có các category khác nhau

VD: với DBI202 chúng ta có các category:

- Progress test
- Labs
- Assignment
- Practice Exam
- Final exam

Nhưng đối với môn MAE101 thì chúng ta có các category như:

- Assignment
- Progress test
- Final Exam

Như vậy có thể thấy category của từng môn là khác nhau không chỉ vậy weight của từng môn cũng khác nhau nên một môn chỉ được một category duy nhất và mỗi category cũng chỉ ứng với một môn nên quan hệ của bảng Category – subject là một – một.

Vì hai bảng có quan hệ 1-1 nên khóa trong bảng category và subject sẽ có Attribute suID

Trong bảng category Primary sẽ là suID và caName vì khi và chỉ khi hai khóa này làm khóa chính chúng ta mới có thể xác định toàn bộ các attribute còn lại

- Student – Category

Quan hệ nhiều nhiều

Vì một student có thể thi nhiều môn và một môn có thể được nhiều sinh viên thi

Hai bảng này có quan hệ nhiều nhiều nên chúng ta sẽ tạo một bảng mới có tên là Assessment

Bảng Assessment để lưu kết quả các bài test mà student đã tham gia

Assessment (suID, caName, sID, examDate, score)

Trong bảng Assessment primary chính là suID, caName, sID, examDate

suID, caName, sID, examDate để xác định môn thi, loại bài thi, người thi, ngày thi

- Student – Subject

Quan hệ nhiều nhiều

Vì Một học sinh có thể học nhiều môn học và một môn học cũng có thể được học bởi nhiều học sinh

Từ quan hệ trên sinh ra bảng Student_Subject

Student_Subject (sID, suID, group, seID, mark, status)

sID và suID sẽ làm khóa chính còn lại là lưu thêm một số thông tin như tên group,

seID kì học ứng với học sinh và môn học vì kì học và môn học của các học sinh là khác nhau

mark điểm trung bình của từng môn học ứng với một học sinh cụ thể

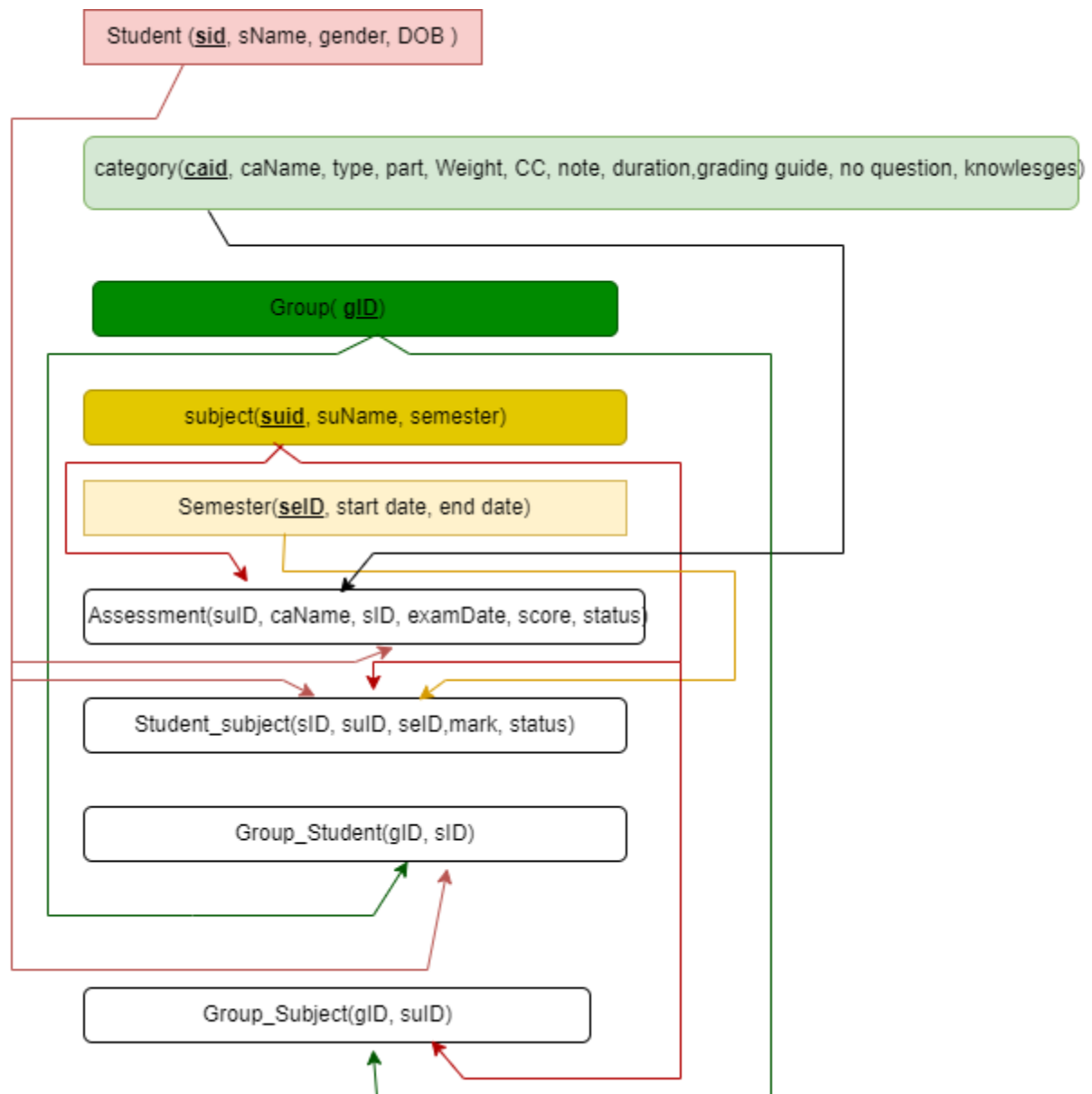
status trạng thái môn học của học sinh có thể là passed hoặc not passed.

- Student_Subject – Semester

Vấn đề: tại sao bảng Semester không liên kết với bảng subject

Giải thích: vì nếu liên kết với bảng subject thì mỗi môn chỉ được học trong một kì mà không được học trong các kì khác vì vậy nên semester cần liên kết với Student_Subject để chỉ ra cụ thể sinh viên nào học môn nào và trong kì nào

Chuyển hóa từ ERD Diagram sang mô hình Table



Tạo các bảng trên sql

Create bảng Student

```
CREATE TABLE Student(  
sID nvarchar(50) PRIMARY KEY not null,  
sName nvarchar(50),  
gender bit default(1),  
DOB date  
)
```

Create bảng Category

```
CREATE TABLE Category(  
[suID] [nvarchar](50) NOT NULL,  
[caName] [nvarchar](50) NOT NULL,  
[Type] [nvarchar](50) NULL,  
[Part] [int] NULL,  
[Weigth] [float] NULL,  
[completionCriterial] [nvarchar](50) NULL,  
[duration] [int] NULL,  
[Question_Type] [nvarchar](150) NULL,  
[No_Question] [nvarchar](150) NULL,  
[Knowledge_and_Skill] [nvarchar](150) NULL,  
[Grading Guide] [nvarchar](150) NULL,  
[Note] [nvarchar](150) NULL,  
constraint pk PRIMARY KEY (caName, suID),  
constraint fk_category FOREIGN KEY(suID) REFERENCES subject(suID)  
)
```

Constraint pk gán caName và suID làm khóa chính

Constraint fk_category khóa ngoại tham chiếu đến bảng subject bởi khóa suID

Create bảng Group

```
CREATE TABLE [Group](  
gID nvarchar(50) primary Key not null,  
)
```

Create bảng subject

```
CREATE TABLE subject(  
suID nvarchar(50) primary key not null,  
suName nvarchar(50),  
)
```

Create table Semester

```
CREATE TABLE Semester(  
seID nvarchar(50) PRIMARY KEY NOT NULL,  
startDate nvarchar(50),  
endDate nvarchar(50)  
)
```

Create table Assessment

```
CREATE TABLE Assessment(  
suID nvarchar(50),  
caID nvarchar(50),  
sID nvarchar(50),  
examDate date,  
score float  
constraint pk_A PRIMARY KEY(suID, caID,sID, examDate),  
constraint fk_A FOREIGN KEY (suID,caID) REFERENCES Category(suID,caID),  
constraint fk2_A FOREIGN key (sID) REFERENCES Student(sID)  
)
```

Constraint pk_A nhận 4 attribute làm khóa chính

Constraint fa_A lấy khóa ngoại từ bảng catefory

Constraint fk2_A lấy khóa ngoại từ bảng Student

Create table Group_subject

```
CREATE TABLE Group_Subjet( --study  
gID nvarChar(50),  
suID nvarchar(50)  
constraint pk_Group_Subjet PRIMARY KEY(gID,suID)  
constraint fk1 FOREIGN KEY (gID) REFERENCES [Group](gID),  
constraint fk2 FOREIGN KEY (suID) REFERENCES [Subject](suID)  
)
```

Create table Group_Student

```
CREATE TABLE Group_Student(  
gID nvarchar(50) not null,  
sID nvarchar(50) not null,  
constraint pk_GS PRIMARY KEY (gID, sID),  
constraint fk_GS FOREIGN KEY (gID) REFERENCES [Group](gID),  
constraint fk1_GS FOREIGN KEY (sID) REFERENCES [Student](sID)  
)
```

Create Student_Subject

```
CREATE TABLE Student_Subject(  
sID nvarchar(50) not null,  
suID nvarchar(50)not null,  
seID nvarchar (50),  
mark float,  
[group ] nvarchar(50),  
[status] nvarchar(50)  
  
constraint checkStatus CHECK(status = 'passed'or status = 'passed (with condition)'or  
status = ' not passed'),  
constraint pk_Student_Subject PRIMARY KEY (sID, suID),  
constraint fk_Student_Subject1 FOREIGN KEY (sID) REFERENCES Student(sID),  
constraint fk_Student_Subject2 FOREIGN KEY (suID) REFERENCES Subject(suID),  
constraint fk_Student_Subject3 FOREIGN KEY (seID) REFERENCES Semester(seID)  
)
```

Constraint checkStatus dùng để check status đầu vào chỉ được một trong các kiểu

Passed, not passed, passed (with condition)

Sample data trong các bảng

Data bảng Student

Results					Messages				
	sID	sName	gender	DOB					
1	HE134553	Nguyễn Hồng Vân	0	1995-06-16					
2	HE150112	Yên Công Bằng	1	1996-11-12					
3	HE157932	Chu Cẩm Yến	0	2000-07-11					
4	HE160136	Đỗ Văn Chung	1	2002-03-30					
5	HE160233	Triệu Duy Cường	1	1997-05-27					
6	HE167443	Nguyễn Minh Hùng	1	2000-10-05					
7	HS132332	Phùng Thụy Linh	0	1998-08-19					
8	HS152662	Đào Nhật Minh	1	2001-01-01					

Results	Messages										
suID	caName	Type	Part	Weight	completionCriteria	duration	Question_Type	No_Question			
1	CEA201	Exercises	on-going	4	30	>0	Option 1: At home Option2: (For Constructivism Approach ...	Option 1: Developing Assembly program Option 2 (For Constructivism Approach ...	Option 1: Depends on chapters Option 2 (For Constructivism Approach ...		
2	CEA201	Final Exam	final exam	1	40	4	60'	Multiple choice	50		
3	CEA201	Final Exam rest	final exam	1	40	4	60'	Multiple choice	50		
4	CEA201	Assignment	on-going	2	30	>0	Option 1: At home Option2: (For Constructivism Approach ...	Option 1: Developing Assembly program Option 2 (For Constructivism Approach ...	Option 1: 2 Option 2 (For Constructivism Approach ...		
5	DBI202	Assignment	on-going	1	20	>0	28 slots	NULL	N/A		
6	DBI202	lab	on-going	5	10	>0	90'	NULL	N/A		
7	DBI202	Practice Exam	on-going	1	30	>0	90	NULL	N/A		
8	DBI202	Progress Test	on-going	2	10	>0	30'	Multiple choices; Marked by Computer or a suitable format ...	20		
9	DBI202	Final Exam rest	final exam	1	30	4	NULL	NULL	50		
10	DBI202	Final Exam	final exam	1	30	4	NULL	NULL	50		
11	IOT202	Final Exam	final exam	1	40	4	60'	Computer Gradable	50		
12	IOT202	Final Exam rest	final exam	1	40	4	60'	Computer Gradable	50		
13	IOT202	Progress test	on-going	2	10	>0	30'	Private	N/A		
14	IOT202	Activate learning	on-going	1	10	>0	From start to finish in schedule	Private	N/A		
15	IOT202	Project	on-going	1	30	>0	From start to finish in schedule	Group	N/A		
16	IOT202	Presentation	on-going	1	10	>0	NULL	Group	N/A		
17	MAE101	Assignment	on-going	3	30	>0	OP1: at home // OP2: (For Constructivism Approach ...	OP1: Open // OP2: (For Constructivism Approach ...	OP1: At least 3/chapter // OP2: (For Constructivism Approach ...		
18	MAE101	Progress Test	on-going	3	30	>0	OP1: Test 1: 40Test 2&3: 30'	OP1: MC or a suitable format // OP2: (For Constructivism Approach ...	OP1: Test 1: 20 Test 2&3: 15+ At least 5% ...		

	gID
1	IA1601
2	IA1604
3	SE1532
4	SE1655

	suID	caName	sID	examDate	score
1	CEA201	Assignment	HE134553	2022-06-11	5
2	CEA201	Assignment	HE150112	2022-12-12	5
3	CEA201	Assignment	HE157932	2022-09-06	2
4	CEA201	Assignment	HE160136	2022-12-15	6
5	CEA201	Assignment	HE167443	2022-11-07	6
6	CEA201	Assignment	HS132332	2022-12-25	5
7	CEA201	Exercises	HE134553	2022-06-10	8
8	CEA201	Exercises	HE150112	2022-10-15	9
9	CEA201	Exercises	HE157932	2022-11-16	4
10	CEA201	Exercises	HE160136	2022-12-10	6
11	CEA201	Exercises	HE167443	2022-11-17	2
12	CEA201	Exercises	HS132332	2022-11-09	9
13	CEA201	Final Exam	HE134553	2022-06-15	6
14	CEA201	Final Exam	HE150112	2022-10-06	9
15	CEA201	Final Exam	HE157932	2022-11-11	9

Data bảng subject

	suID	suName
1	CEA201	Computer Organization and Architecture
2	DBI202	Introduction to Databases
3	IOT202	Internet of Things
4	MAE101	Mathematics for Engineering

Tính toán điểm của từng môn

Cách tính toán điểm của từng môn

Bảng cách lấy tổng các tích của điểm nhân với hệ số điểm

SQL QUERY:

```
SELECT a.suID, sID, sum(c.Weight*a.score/100) as total FROM Assessment a
INNER JOIN Category c ON a.caName = c.caName and a.suID=c.suID and a.suID=c.suID
group by a.suID, sID
```

Để tính điểm trung bình của từng môn chúng ta cần lấy score của từng category xong nhân với weigh và chia cho 100

Vì score ở assessment và weight nằm ở category nên chúng ta cần join hai bảng lại với nhau

Và cộng tất các các category trong một môn học

Kết quả của câu QUERY trên như ảnh sau:

	suID	sID	total
1	CEA201	HE134553	6.3
2	DBI202	HE134553	6.1
3	IOT202	HE134553	7
4	CEA201	HE150112	7.8
5	MAE101	HE150112	7.1
6	CEA201	HE157932	5.4
7	DBI202	HE157932	5.5
8	IOT202	HE157932	6
9	MAE101	HE157932	7.6
10	CEA201	HE160136	5.2
11	DBI202	HE160136	3.5
12	IOT202	HE160136	6.3
13	DBI202	HE160233	6.3
14	IOT202	HE160233	4
15	MAE101	HE160233	3.2
16	CEA201	HE167443	6
17	DBI202	HE167443	4.4
18	IOT202	HE167443	5
19	CEA201	HS132332	7.4

Kết quả bao gồm các được group by suID, sName

Nhập điểm vào bảng Student_subject để nhập điểm vào chúng ta cần sử dụng Cursor trong Để

Nhập vào từng record

```

DECLARE @suID nvarchar(50);
DECLARE @sID nvarchar(50);
DECLARE update_mark_cursor CURSOR FOR
SELECT suID, sID FROM Assessment;
OPEN update_mark_cursor;
FETCH NEXT FROM update_mark_cursor INTO @suID, @sID--, @caName
WHILE @@FETCH_STATUS = 0
BEGIN
    DECLARE @TOTAL FLOAT;
    SELECT @TOTAL = sum(c.Weight*a.score/100) FROM Assessment a
INNER JOIN Category c ON a.caName = c.caName and a.suID=c.suID
group by a.suID, sID HAVING a.suID = @suID and a.sID= @sID;
    UPDATE Student_Subject SET mark = @TOTAL WHERE suID = @suID and sID = @sID;
    FETCH NEXT FROM update_mark_cursor INTO @suID, @sID
END
CLOSE update_mark_cursor;
DEALLOCATE update_mark_cursor;

```

Cách sử dụng cursor để update data

Bước 1 khai báo hai biến @suID, @sID

Bước 2 khai báo cursor trở tới Select suID, sID From Assessment

Bước tạo vòng while lấy điểm trung bình gán vào biến @total và update vào mark trong bảng Student_subject

Một số câu query trên database

1 Tìm học sinh có GPA lớn nhất

```
SELECT a.sID, AVG(a.total) as GPA FROM
(SELECT a.suID, sID, sum(c.Weight*a.score/100) as total FROM Assessment a
INNER JOIN Category c ON a.caName = c.caName and a.suID=c.suID and a.suID=c.suID
group by a.suID, sID) a
group by sID Having AVG(a.total) =

(Select MAX(b.GPA) FROM
(SELECT a.sID, AVG(a.total) as GPA FROM
(SELECT a.suID, sID, sum(c.Weight*a.score/100) as total FROM Assessment a
INNER JOIN Category c ON a.caName = c.caName and a.suID=c.suID and a.suID=c.suID
group by a.suID, sID) a
group by sID )b)
```

sID	GPA
HE150112	7.45

2.Tìm học sinh có GPA nhỏ nhất

```
SELECT a.sID, AVG(a.total) as GPA FROM
(SELECT a.suID, sID, sum(c.Weight*a.score/100) as total FROM Assessment a
INNER JOIN Category c ON a.caName = c.caName and a.suID=c.suID and a.suID=c.suID
group by a.suID, sID) a
group by sID Having AVG(a.total) =

(Select MIN(b.GPA) FROM
(SELECT a.sID, AVG(a.total) as GPA FROM
(SELECT a.suID, sID, sum(c.Weight*a.score/100) as total FROM Assessment a
INNER JOIN Category c ON a.caName = c.caName and a.suID=c.suID and a.suID=c.suID
group by a.suID, sID) a
group by sID )b)
```

sID	GPA
HE160233	4.5

4 tìm những môn học mà học sinh chưa học

```
SELECT suID, sID FROM Student_Subject
except
SELECT suID, sID FROM Assessment Group by suID, sID
```

5 hiển thị bảng Syllabus

```
SELECT TOP (1000) [suID]
      ,[caName]
      ,[Type]
      ,[Part]
      ,[Weight]
      ,[completionCriterial]
      ,[duration]
      ,[Question_Type]
      ,[No_Question]
      ,[Knowledge_and_Skill]
      ,[Grading Guide]
      ,[Note]
FROM [MarkManage].[dbo].[Category]
```

6. Hiển thị bảng Academic transcript

```
SELECT su.suID, su.suName, se.seName, gs.gID, se.startDate, se.endDate, ss.mark, ss.status FROM
Student_Subject ss
INNER join Semester se ON ss.seID = se.seID
INNER JOIN subject su ON ss.suID = su.suID
INNER JOIN Student s ON ss.sID = s.sID
INNER JOIN Group_Student gs ON s.sID= gs.sID where ss.sID = 'HE160136'
order by su.suName
```

Kết quả

	suID	suName	seName	glD	startDate	endDate	mark	status
1	CEA201	Computer Organization and Architecture	Summer2022	IA1601	2022-4-2	2022-9-1	5.2	passed
2	CEA201	Computer Organization and Architecture	Summer2022	IA1604	2022-4-2	2022-9-1	5.2	passed
3	IOT202	Internet of Things	Fall2022	IA1601	2022-9-2	2023-1-1	6.3	passed
4	IOT202	Internet of Things	Fall2022	IA1604	2022-9-2	2023-1-1	6.3	passed
5	DBI202	Introduction to Databases	Summer2022	IA1601	2022-4-2	2022-9-1	3.5	not Passed
6	DBI202	Introduction to Databases	Summer2022	IA1604	2022-4-2	2022-9-1	3.5	not Passed
7	MAE101	Mathematics for Engineering	Spring2023	IA1601	2023-1-2	2023-4-1	NULL	NULL
8	MAE101	Mathematics for Engineering	Spring2023	IA1604	2023-1-2	2023-4-1	NULL	NULL

7. Hiển thị bảng Student transcript

```
SELECT * FROM Assessment WHERE sID = 'HE160136' and suID = 'DBI202'
```

	suID	caName	sID	examDate	score
1	DBI202	Assignment	HE160136	2022-09-06	5
2	DBI202	Final Exam	HE160136	2022-12-09	1
3	DBI202	lab	HE160136	2022-10-30	3
4	DBI202	Practice Exam	HE160136	2022-10-30	3
5	DBI202	Progress Test	HE160136	2022-11-24	10

8. tìm học sinh mà đã học tất cả các môn

```
SELECT sID, Count(suID) as total FROM Student_Subject where status is not null Group BY sID HaVING COUNT(suID) = (SELECT COUNT(*)FROM subject)
```

sID	totalSubject
HE157932	4

9 lớp có nhiều sinh viên nhất

```
SELECT gID, COUNT(sID) FROM Group_Student group by gID HAVING count(sID) = (SELECT  
max(a.numStudent) FROM (SELECT gID, COUNT(sID) as numStudent FROM Group_Student group by  
gID)a)
```

gID	NumberStudent
IA1604	4