FINAL REPORT

FPT Education

FPT UNIVERSITY

Student Grading Manage

Name :Vũ Đức Huy

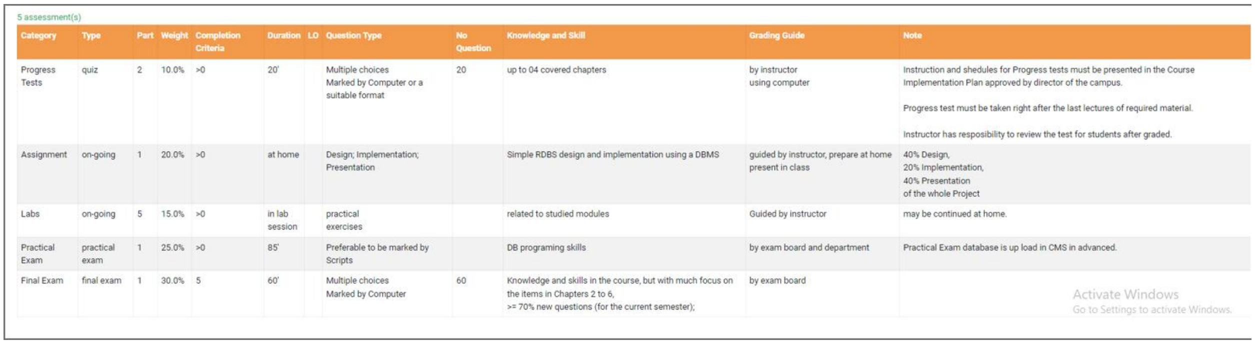
Mssv: HE160866

Class:IA1604

Student Grading Management Sub-System

For each subject that attended by the student, the lecture will give score to the assessment to each of their assessment. Below figure shows an Example of the assessments for course DBI202.

Table FML(Chưa được phân tách)



5 assessment(s)

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 shedules 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 resposibility 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 programing 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	

+ Category: Tên các đầu điểm thành phần của 1 môn học

+ Type: Kiểu kiểm tra của các đầu điểm.

+ Part: Số phần điểm thành phần nhỏ trong điểm thành phần.

+Weight: Trọng số điểm thành phần chiếm bao nhiêu % trên 100% điểm.

+ Completion Criteria: Điều kiện tiên quyết của điểm thành phần cần đạt được.

+Duration: Thời gian làm việc cho mỗi thành phần điểm.

+Question Type: Các kiểu câu hỏi có trong hạng mục kiểm tra.

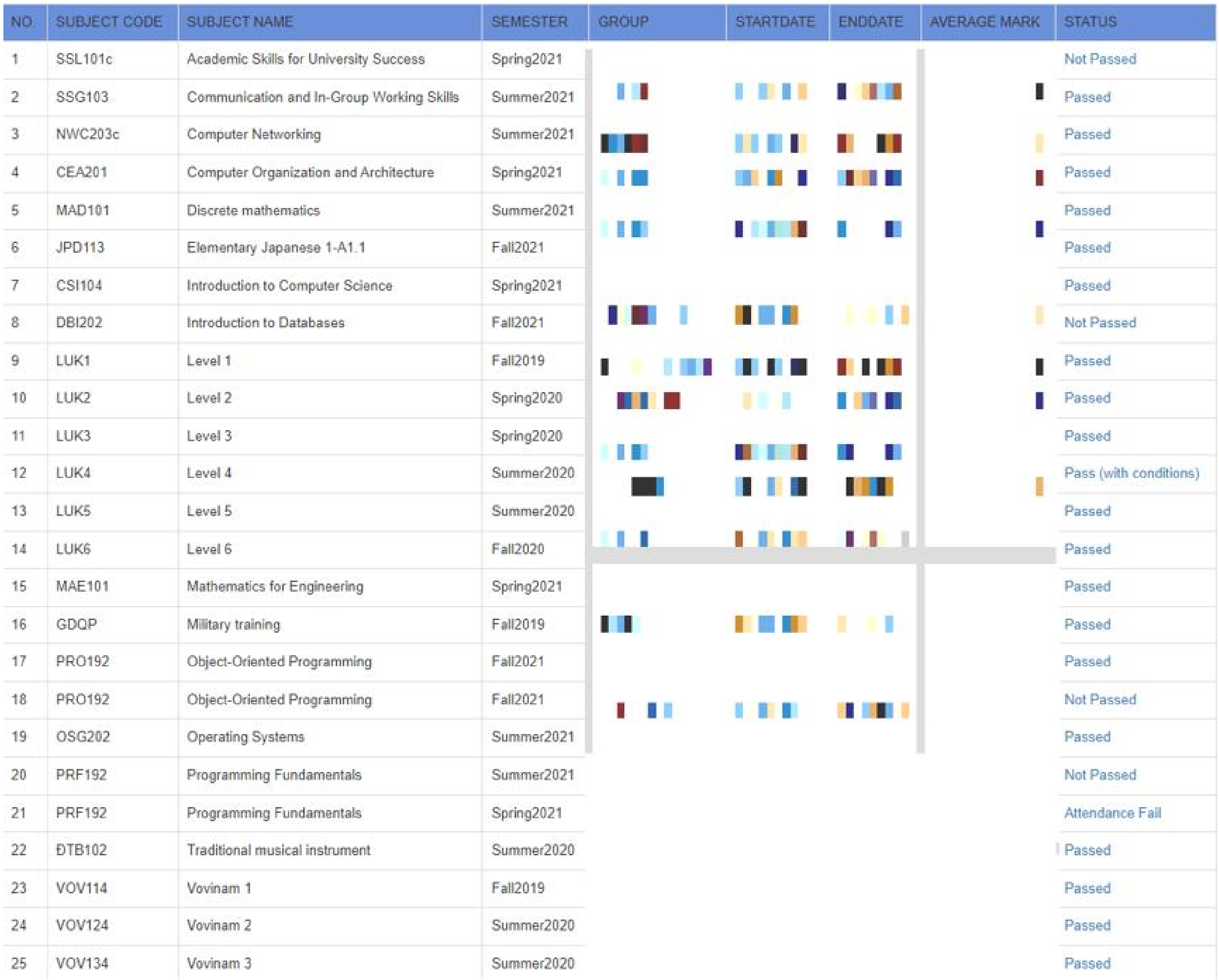
+No Question: Số lượng câu hỏi.

+Knowledge and Skill: Kiến thức và kỹ năng cần có để làm kiểm tra hạng mục.

+Grading Guide: Người chịu trách nhiệm chấm điểm cho hạng mục kiểm tra.

+Note: Một số ghi chú cần thiết.

Thông tin trạng thái môn học của sinh viên



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

+NO : 1 số tự tăng biểu thị STT

+ Subject code: Mã môn học.

+Subject name: Tên môn học.

+ SEMESTER: kỳ học diễn ra môn học.

+GROUP: Lớp học môn học đó.

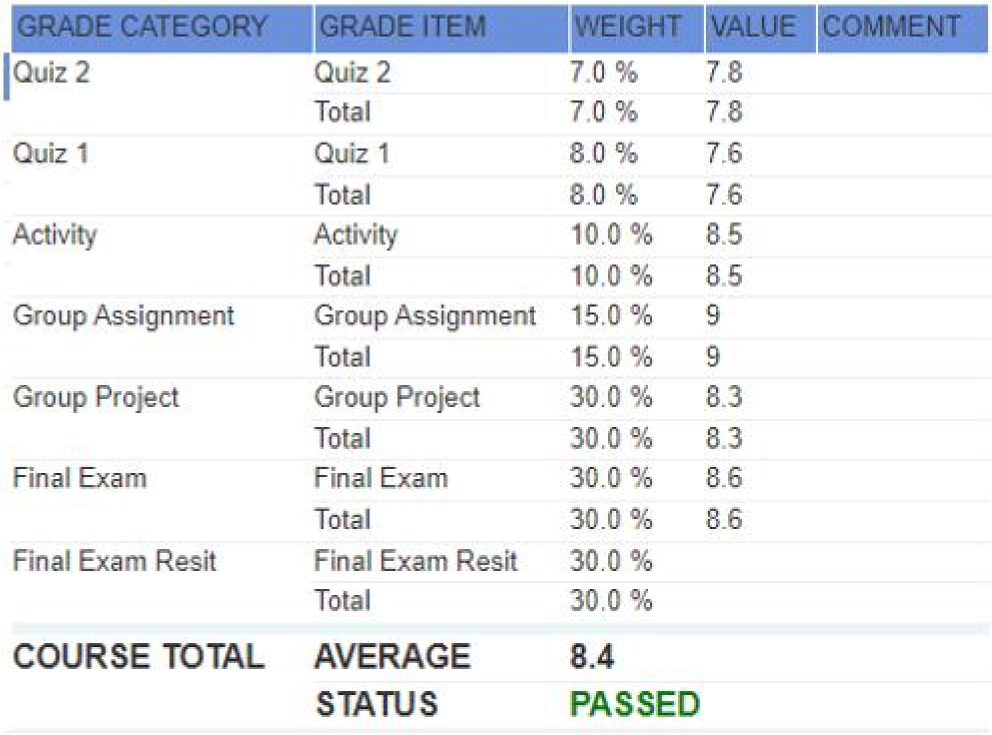
+StartDate: Ngày bắt đầu môn học.

+EndDate: Ngày kết thúc môn học.

+Average mark: Điểm trung bình của môn học.

+Status: Trạng thái của môn học đó (passed or not passed).

Bảng Điểm Chi Tiết Của Sinh Viên:



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: Tên điểm thành phần

+ GRADE ITEM : Tên điểm thành phần nhỏ trong điểm thành phần.

+WEIGHT: Trọng số của điểm thành phần.

+Value: giá trị điểm thành phần đạt được.

+Comment: Ghi chú cho các điểm thành phần khi có vấn đề.

In the system analyse , I can see that the Student Grading Management Sub-System have built about many main entities **: Assesment ; Grade ; Student ; Category ; Lecturers ; View ; Semester** . Especially , Subject is the best important in Database . In addition , there are some entity : **Group_Student ;Group; Course ; Category Details...**

First step , we should analyse more attribute in many entities :

Assessment (AssID,CategoryDetailsID,CourseID,Duration,Weight)

Category Details(CategoryDetailsID,CategoryID,CategoryDetailName)

Category(CategoryID,CategoryName,CompletionCriteria,Type)

Grade(SID,AssID,Score,Date Exam)

Group_Student(Gid,Sid)

Student(Sid,FirstName,LastName,Gender,DOB,Address)

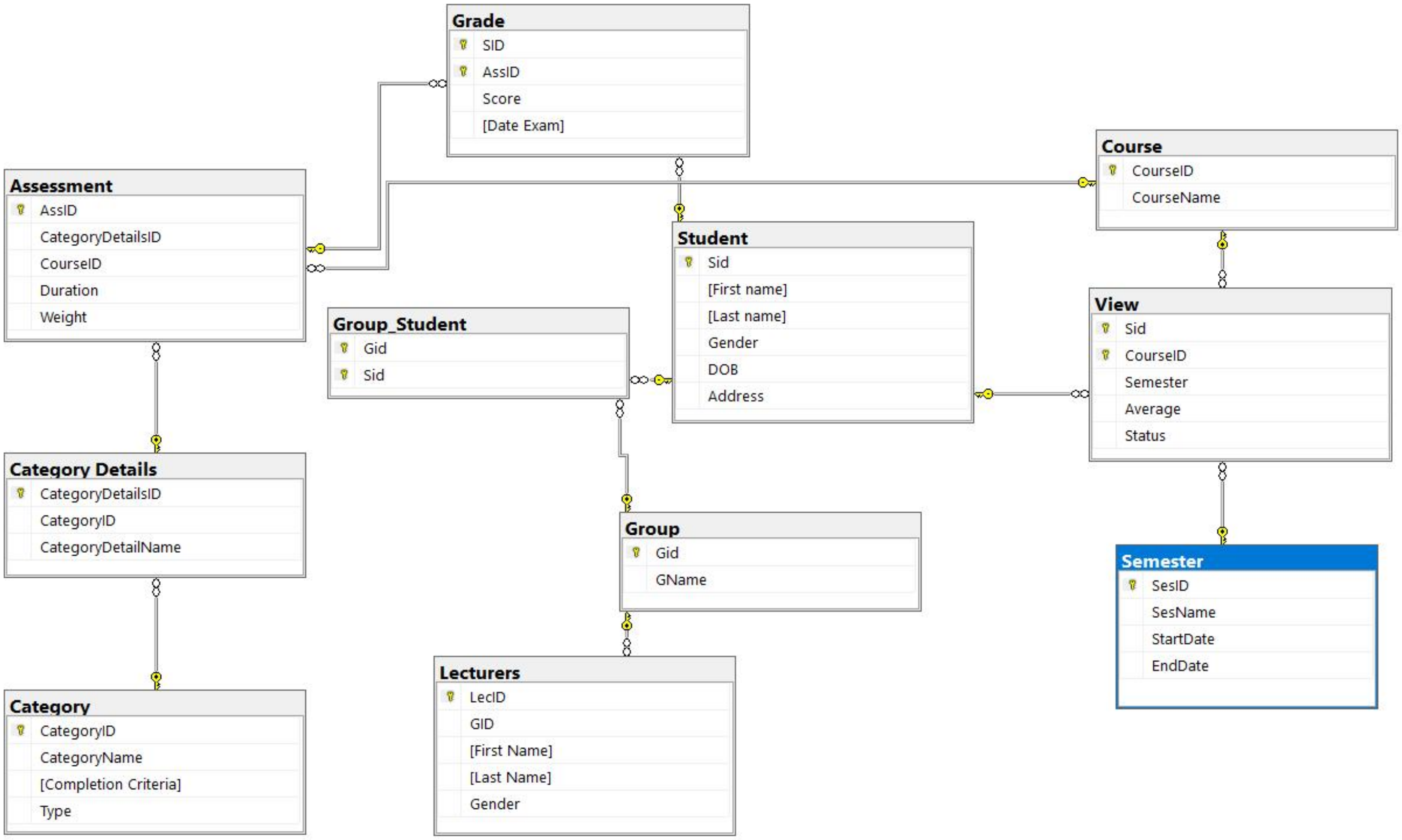
Group(Gid,GName)

Course(CourseID,CourseName)

View(Sid,CourseID,Semester,Average,Status)

Semester(SesID,SesName,StartDate,EndDate)

Diagram of Grading Management :



Phân Chia Các Entities Và Relationships

Entity Students <-> Entity Groups

Mô Tả: Một Student có thể đăng kí học nhiều Group Và 1 Group có thể có nhiều Student đăng kí học.

-> Xác Định Quan Hệ Giữa Entity Students Và Entity Groups là quan hệ nhiều nhiều (n-n)

Entity Student <-> Entity Assessment System

Mô Tả: Một Student có thể có nhiều hệ thống đánh giá các đầu điểm và 1 Assessment System có thể phụ trách đầu điểm của nhiều Students.

-> Xác Định Quan Hệ Giữa Entity Students Và Entity Assessment System là quan hệ nhiều nhiều (n-n)

Entity Category <-> Entity CategoryDetail

Mô Tả: Một Category có thể có nhiều phần nhỏ trong CategoryDetail

-> Xác Định Quan Hệ Giữa Entity Category Và Entity CategoryDetai là quan hệ nhiều nhiều (1-n)

Entity Courses <-> Entity Assessment System

Mô Tả: Một Course chỉ có thể có duy nhất 1 hệ thống đánh giá các đầu điểm và 1 Assessment System có thể là hệ thống đánh giá của nhiều Courses.

-> Xác Định Quan Hệ Giữa Entity Assessment System Và Entity Courses là quan hệ một nhiều (1-n)

Entity CategoryDetail <-> Entity Assignment Systems

Mô Tả: Một CategoryDetail có thể tổng hợp từ nhiều Assignment Systems và 1 Assignment Systems chỉ có thể đưa vào 1 Category duy nhất.

-> Xác Định Quan Hệ Giữa Entity CategoryDetail Và Entity Assignment Systems là quan hệ một nhiều (1-n)

Chuẩn Hóa Thuộc Tính Các Attribute Trên Từng Bảng :

1. Table Assement

Attributes	Date Type
AssID	Varchar
CategoryDetailsID	Varchar
CourseID	Varchar
Duration	Nvarchar
Weight	float

2.Table Category

Attributes	Data Type
CategoryID	Varchar
CategoryName	nvarchar
[Completion Criterial]	Varchar
Type	Nvarchar

3.Table Category Details

Attributes	Data Type
CategoryDetailsID	Varchar
CategoryID	Varchar
CategoryDetailName	Nvarchar

4.Table Course

Attributes	Data Type
CourseID	Varchar
CourseName	Varchar

5.Table Grade

Attributes	Data Type
SID	Char
AssID	Varchar
Score	Float
Date Exam	Date

6.Table Group

Attributes	Data Type
Gid	Varchar
Gname	Nvarchar

7.Table Group_Student

Attributes	Data Type
Gid	Varchar
Sid	Char

8.Table Lecturers

Attributes	Data Type
LecID	Varchar
GID	Varchar
First Name	Varchar
Last Name	Varchar
Gender	bit

9.Table Semester

Attributes	Data Type
SesID	Varchar
SesName	Varchar
StartDate	Date
EndDate	Date

10.Table Student

Attributes	Data Type
Sid	Char
First name	Nvarchar
Last name	Nvarchar
Gender	Bit
DOB	Date
Address	Nvarchar

11.Table View

Attributes	Data Type
Sid	Char
CourseID	Varchar
Semester	Varchar
Average	Float
Status	Varchar

Xác Định Primary Key, Foriegn Key, Attributes Các TABLES :

1. Table Assement

Attributes	Date Type	Requires	Key
AssID	Varchar	Not null	Primary Key
CategoryDetailsID	Varchar	Not null	
CourseID	Varchar	Not null	
Duration	Nvarchar	Not null	
Weight	float	Not null	

2.Table Category

Attributes	Data Type	Requires	Key
CategoryID	Varchar	Not null	Primary Key
CategoryName	nvarchar	Not null	

[Completion Criterial]	Varchar	Not null	
Type	Nvarchar	Not null	

3.Table Category Details

Attributes	Data Type	Requires	Key
CategoryDetailsID	Varchar	Not null	Primary Key
CategoryID	Varchar	Not null	Primary_Foreign Key
CategoryDetailName	Nvarchar	Not null	

4.Table Course

Attributes	Data Type	Requires	Key
CourseID	Varchar	Not null	Primary Key
CourseName	Varchar	Not null	

5.Table Grade

Attributes	Data Type	Requires	Key
SID	Char	Not null	Primary_Foreign Key
AssID	Varchar	Not null	Primary_Foreign Key
Score	Float	Not null	
Date Exam	Date	Null	

6.Table Group

Attributes	Data Type	Requires	Key
Gid	Varchar	Not null	Primary_Key
Gname	Nvarchar	Not null	

7.Table Group_Student

Attributes	Data Type	Requires	Key
Gid	Varchar	Not null	Primary_Foreign Key
Sid	Char	Not null	Primary_Foreign Key

8.Table Semester

Attributes	Data Type	Requires	Key
SesID	Varchar	Not null	Primary Key
SesName	Varchar	Not null	
StartDate	Date	Not null	
EndDate	Date	Not null	

9.Table Student

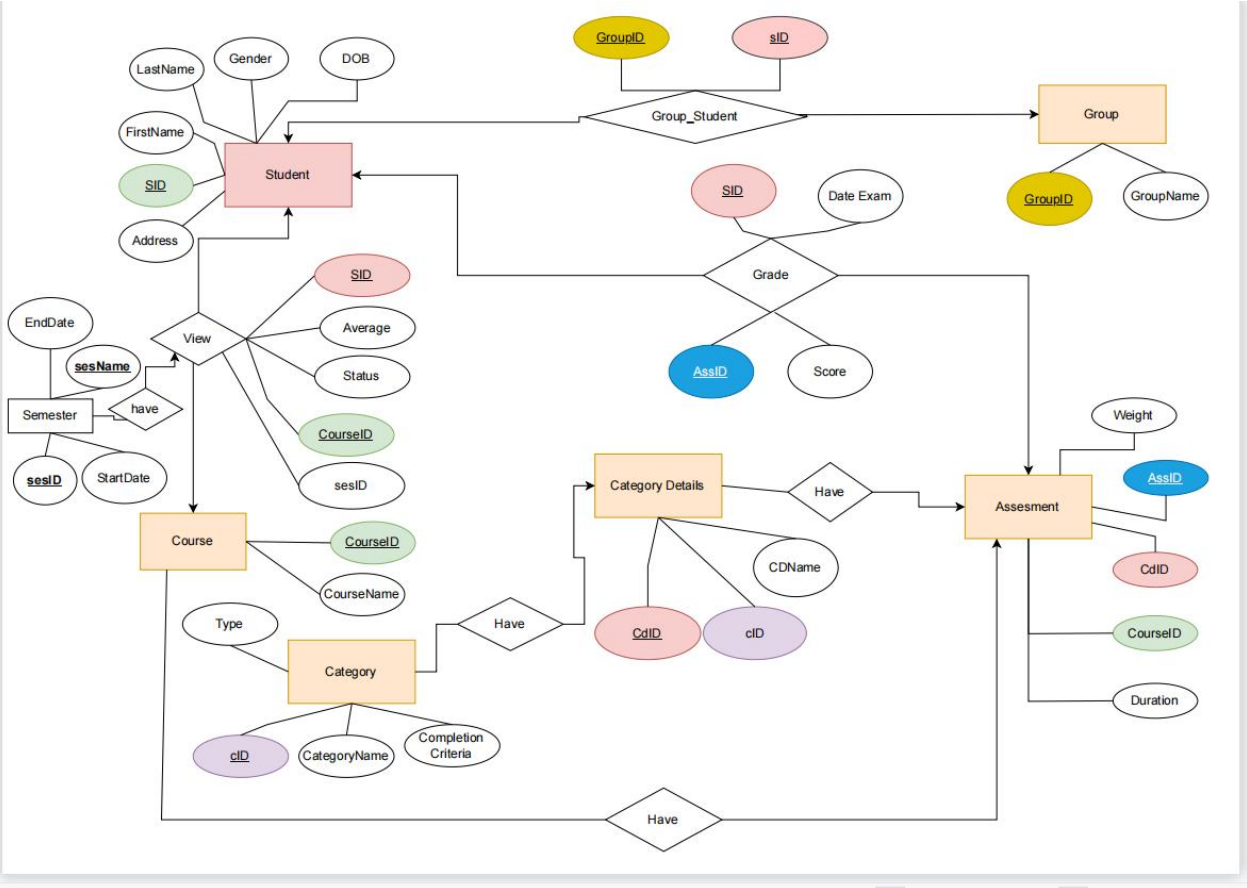
Attributes	Data Type	Requires	Key
Sid	Char	Not null	Primary Key
First name	Nvarchar	Not null	
Last name	Nvarchar	Not null	
Gender	Bit	Not null	
DOB	Date	Not null	
Address	Nvarchar	Not null	

10.Table View

Attributes	Data Type	Requires	Key
Sid	Char	Not null	Primary Key
CourseID	Varchar	Not null	Foregin Key
Semester	Varchar	Not null	
Average	Float	Not null	
Status	Varchar	Not null	

Database_Diagram

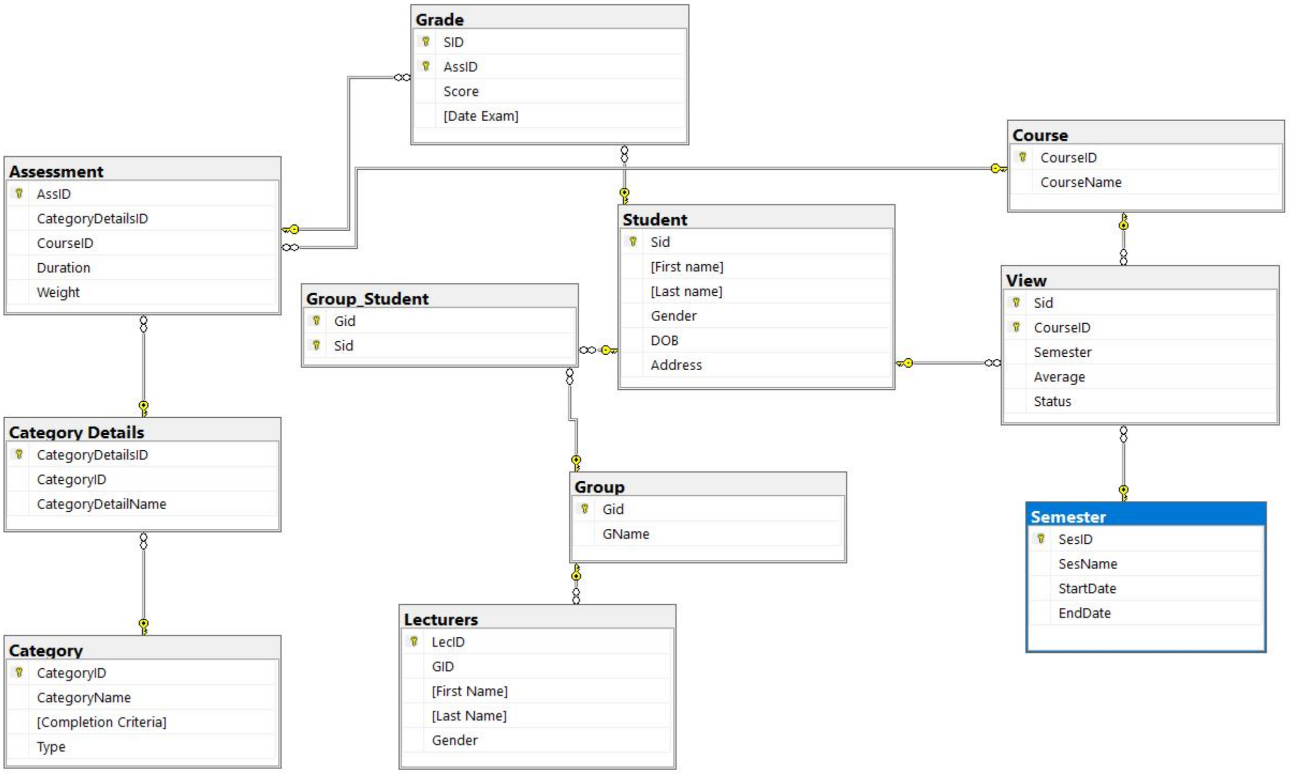
Hình Ảnh Và Mô Tả



Creat Table And Attributes

Code sql

Image + Results:



```
CREATE TABLE [dbo].[Assessment](
     [AssID] [varchar](10) NOT NULL,
       [CategoryDetailsID] [varchar](10) NOT NULL foreign key
references Category Details(CategoryDetailsID),
     [CourseID] [varchar](10) NOT NULL,
     [Duration] [nvarchar](30) NOT NULL,
)
CREATE TABLE [dbo].[Category](
```

```sql
	[CategoryID] [varchar](10) NOT NULL,
	[CategoryName] [nvarchar](50) NOT NULL,
	[Completion Criteria] [varchar](10) NOT NULL,
	[Type] [nvarchar](50) NOT NULL,
)
CREATE TABLE [dbo].[Category Details](
	[CategoryDetailsID] [varchar](10) NOT NULL,
	[CategoryID] [varchar](10) NOT NULL,
	[CategoryDetailName] [nvarchar](50) NOT NULL,
)
CREATE TABLE [dbo].[Course](
	[CourseID] [varchar](10) NOT NULL,
	[CourseName] [varchar](50) NOT NULL,
)
CREATE TABLE [dbo].[Grade](
	[SID] [char](8) NOT NULL,
	[AssID] [varchar](10) NOT NULL,
	[Score] [float] NOT NULL,
	[Date Exam] [date] NULL,
)
CREATE TABLE [dbo].[Group](
	[Gid] [varchar](10) NOT NULL,
	[GName] [nvarchar](50) NOT NULL,
)
CREATE TABLE [dbo].[Group_Student](
	[Gid] [varchar](10) NOT NULL,
	[Sid] [char](8) NOT NULL,
)
CREATE TABLE [dbo].[Lecturers](
	[LecID] [varchar](10) NOT NULL,
	[GID] [varchar](10) NOT NULL,
	[First Name] [varchar](50) NOT NULL,
	[Last Name] [varchar](50) NOT NULL,
	[Gender] [bit] NOT NULL,
)
CREATE TABLE [dbo].[Semester](
	[SesID] [varchar](10) NOT NULL,
	[SesName] [varchar](50) NOT NULL,
	[StartDate] [date] NOT NULL,
	[EndDate] [date] NOT NULL,
)
CREATE TABLE [dbo].[Student](
    [Sid] [char](8) NOT NULL,
    [First name] [nvarchar](50) NOT NULL,
    [Last name] [nvarchar](50) NOT NULL,
    [Gender] [bit] NOT NULL,
    [DOB] [date] NOT NULL,
    [Address] [nvarchar](150) NULL,
)
CREATE TABLE [dbo].[View](
    [Sid] [char](8) NOT NULL,
    [CourseID] [varchar](10) NOT NULL,
    [Semester] [varchar](10) NOT NULL,
    [Average] [float] NULL,
    [Status] [varchar](20) NULL,
)
```

ĐƯA DỮ LIỆU VÀO DATABASE

Code sql

```
USE [Grading Management System]
GO
INSERT [dbo].[Student] ([sID], [FirstName], [LastName], [Gender],
[DOB], [Address]) VALUES (N'HA123456', N'Nguyễn Như', N'Quỳnh', 0,
CAST(N'2003-10-24' AS Date), N'Tuyên Quang')
INSERT [dbo].[Student] ([sID], [FirstName], [LastName], [Gender],
[DOB], [Address]) VALUES (N'HE111111', N'Vũ Đức', N'Huy', 1,
CAST(N'2002-02-02' AS Date), N'Thái Bình')
INSERT [dbo].[Student] ([sID], [FirstName], [LastName], [Gender],
[DOB], [Address]) VALUES (N'HE111112', N'Vũ Đức', N'Hải', 1,
CAST(N'2002-03-02' AS Date), N'Hà Nội')
INSERT [dbo].[Student] ([sID], [FirstName], [LastName], [Gender],
[DOB], [Address]) VALUES (N'HE111113', N'Nguyễn Đức', N'Huy', 1,
CAST(N'2002-04-02' AS Date), N'Thái Bình')
INSERT [dbo].[Student] ([sID], [FirstName], [LastName], [Gender],
[DOB], [Address]) VALUES (N'HE111114', N'Trần Công', N'Hoan', 1,
CAST(N'2002-05-03' AS Date), N'Hải Phòng')
INSERT [dbo].[Student] ([sID], [FirstName], [LastName], [Gender],
[DOB], [Address]) VALUES (N'HE111115', N'Trần Thị', N'Nhu', 0,
CAST(N'2002-07-03' AS Date), N'Hà Giang')
INSERT [dbo].[Student] ([sID], [FirstName], [LastName], [Gender],
[DOB], [Address]) VALUES (N'HE111116', N'Lê Nguyễn Hân', N'Du', 0,
CAST(N'2002-12-01' AS Date), N'Thái Bình')
INSERT [dbo].[Student] ([sID], [FirstName], [LastName], [Gender],
[DOB], [Address]) VALUES (N'HE111117', N'Đặng Thị Mai', N'Trinh', 0,
CAST(N'2002-02-02' AS Date), N'Thái Bình')
INSERT [dbo].[Student] ([sID], [FirstName], [LastName], [Gender],
[DOB], [Address]) VALUES (N'HE111118', N'Trần Đình', N'Cương', 1,
CAST(N'2002-11-19' AS Date), N'Thái Nguyên')
INSERT [dbo].[Student] ([sID], [FirstName], [LastName], [Gender],
[DOB], [Address]) VALUES (N'HE111119', N'Vũ Văn ', N'Hải', 1,
CAST(N'2002-10-10' AS Date), N'Hà Giang')
INSERT [dbo].[Student] ([sID], [FirstName], [LastName], [Gender],
[DOB], [Address]) VALUES (N'HS123444', N'Nguyễn Cảnh ', N'Thương',
1, CAST(N'2003-11-11' AS Date), N'Hà Giang')
GO
INSERT [dbo].[Course] ([CourseID], [CourseName]) VALUES (N'CSI104',
N'Introduction to Computer Science ')
INSERT [dbo].[Course] ([CourseID], [CourseName]) VALUES (N'MAD101',
N'Discrete mathematics')
INSERT [dbo].[Course] ([CourseID], [CourseName]) VALUES (N'MAE101',
N'Mathematics for Engineering')
INSERT [dbo].[Course] ([CourseID], [CourseName]) VALUES (N'PRF192',
N'Programming Fundamentals ')
INSERT [dbo].[Course] ([CourseID], [CourseName]) VALUES (N'SSL101c',
N'Academic Skills for University Success ')
GO
INSERT [dbo].[Semester] ([SesID], [SesName], [StartDate], [EndDate])
VALUES (N'FALL21', N'Fall 2021', CAST(N'2021-09-06' AS Date),
CAST(N'2021-11-30' AS Date))
INSERT [dbo].[Semester] ([SesID], [SesName], [StartDate], [EndDate])
VALUES (N'SP21', N'Spring 2021', CAST(N'2021-01-06' AS Date),
CAST(N'2021-03-25' AS Date))
INSERT [dbo].[Semester] ([SesID], [SesName], [StartDate], [EndDate])
VALUES (N'SP22', N'Spring 2022', CAST(N'2022-01-05' AS Date),
CAST(N'2022-03-25' AS Date))
INSERT [dbo].[Semester] ([SesID], [SesName], [StartDate], [EndDate])
VALUES (N'SU21', N'Summer 2021', CAST(N'2021-05-06' AS Date),
CAST(N'2021-07-02' AS Date))
GO
INSERT [dbo].[View] ([sID], [CourseID], [Semester], [Average],
[Status]) VALUES (N'HE111111', N'MAE101', N'FALL21',
9.2399997711181641, N'PASSED')
```

```
INSERT [dbo].[View] ([sID], [CourseID], [Semester], [Average],
[Status]) VALUES (N'HE111111', N'PRF192', N'FALL21',
9.3891000747680664, N'PASSED')
INSERT [dbo].[View] ([sID], [CourseID], [Semester], [Average],
[Status]) VALUES (N'HE111112', N'MAE101', N'FALL21', 8, N'PASSED')
INSERT [dbo].[View] ([sID], [CourseID], [Semester], [Average],
[Status]) VALUES (N'HE111112', N'PRF192', N'FALL21',
7.4520001411437988, N'NOT PASSED')
INSERT [dbo].[View] ([sID], [CourseID], [Semester], [Average],
[Status]) VALUES (N'HE111113', N'MAE101', N'FALL21',
8.3999996185302734, N'PASSED')
INSERT [dbo].[View] ([sID], [CourseID], [Semester], [Average],
[Status]) VALUES (N'HE111113', N'PRF192', N'FALL21',
7.2080001831054688, N'PASSED')
INSERT [dbo].[View] ([sID], [CourseID], [Semester], [Average],
[Status]) VALUES (N'HE111114', N'MAE101', N'FALL21', 7.5, N'PASSED')
INSERT [dbo].[View] ([sID], [CourseID], [Semester], [Average],
[Status]) VALUES (N'HE111114', N'PRF192', N'FALL21',
6.8369998931884766, N'PASSED')
GO
INSERT [dbo].[Group] ([Gid], [gName]) VALUES (N'IA1', N'IA1604')
INSERT [dbo].[Group] ([Gid], [gName]) VALUES (N'IA2', N'IA1605')
INSERT [dbo].[Group] ([Gid], [gName]) VALUES (N'SE1', N'SE1647')
INSERT [dbo].[Group] ([Gid], [gName]) VALUES (N'SE2', N'SE1621')
INSERT [dbo].[Group] ([Gid], [gName]) VALUES (N'SE3', N'SE1649')
GO
INSERT [dbo].[Group_Student] ([gID], [sID]) VALUES (N'IA1',
N'HE111117')
INSERT [dbo].[Group_Student] ([gID], [sID]) VALUES (N'IA1',
N'HE111119')
INSERT [dbo].[Group_Student] ([gID], [sID]) VALUES (N'IA2',
N'HA123456')
INSERT [dbo].[Group_Student] ([gID], [sID]) VALUES (N'IA2',
N'HS123444')
INSERT [dbo].[Group_Student] ([gID], [sID]) VALUES (N'SE1',
N'HE111111')
INSERT [dbo].[Group_Student] ([gID], [sID]) VALUES (N'SE1',
N'HE111112')
INSERT [dbo].[Group_Student] ([gID], [sID]) VALUES (N'SE1',
N'HE111113')
INSERT [dbo].[Group_Student] ([gID], [sID]) VALUES (N'SE1',
N'HE111119')
```

```
INSERT [dbo].[Group_Student] ([gID], [sID]) VALUES (N'SE2',
N'HE111111')
INSERT [dbo].[Group_Student] ([gID], [sID]) VALUES (N'SE2',
N'HE111114')
INSERT [dbo].[Group_Student] ([gID], [sID]) VALUES (N'SE2',
N'HE111116')
GO
INSERT [dbo].[Category] ([CategoryID], [CategoryName],
[CompletionCriteria], [Type]) VALUES (N'ACT', N'Activity', 0, N'on-
going')
INSERT [dbo].[Category] ([CategoryID], [CategoryName],
[CompletionCriteria], [Type]) VALUES (N'ASS', N'Assignment', 0,
N'on-going')
INSERT [dbo].[Category] ([CategoryID], [CategoryName],
[CompletionCriteria], [Type]) VALUES (N'FE', N'Final exam', 4,
N'final exam')
INSERT [dbo].[Category] ([CategoryID], [CategoryName],
[CompletionCriteria], [Type]) VALUES (N'GA', N'Group asm', 0, N'on-
going')
INSERT [dbo].[Category] ([CategoryID], [CategoryName],
[CompletionCriteria], [Type]) VALUES (N'GP', N'Group Project', 0,
N'on-going')
INSERT [dbo].[Category] ([CategoryID], [CategoryName],
[CompletionCriteria], [Type]) VALUES (N'LAB', N'Lab', 0, N'on-
going')
INSERT [dbo].[Category] ([CategoryID], [CategoryName],
[CompletionCriteria], [Type]) VALUES (N'ME', N'Mid-term test', 0,
N'on-going')
INSERT [dbo].[Category] ([CategoryID], [CategoryName],
[CompletionCriteria], [Type]) VALUES (N'PA', N'Participation', 0,
N'on-going')
INSERT [dbo].[Category] ([CategoryID], [CategoryName],
[CompletionCriteria], [Type]) VALUES (N'PE', N'Practice Exam', 0,
N'practical exam')
INSERT [dbo].[Category] ([CategoryID], [CategoryName],
[CompletionCriteria], [Type]) VALUES (N'PEc', N'Practice exam', 4,
N'final exam')
INSERT [dbo].[Category] ([CategoryID], [CategoryName],
[CompletionCriteria], [Type]) VALUES (N'PRE', N'Group presentation',
0, N'on-going')
INSERT [dbo].[Category] ([CategoryID], [CategoryName],
[CompletionCriteria], [Type]) VALUES (N'PT', N'Progress test', 0,
N'quiz')
INSERT [dbo].[Category] ([CategoryID], [CategoryName],
[CompletionCriteria], [Type]) VALUES (N'QUIZ', N'Quiz', 0, N'on-
going')
INSERT [dbo].[Category] ([CategoryID], [CategoryName],
[CompletionCriteria], [Type]) VALUES (N'TE', N'Theory Exam', 4,
N'final exam')
INSERT [dbo].[Category] ([CategoryID], [CategoryName],
[CompletionCriteria], [Type]) VALUES (N'WS', N'Workshop', 0, N'on-
going     ')
GO
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'ACT1', N'ACT', N'Activity 1')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'ACT2', N'ACT', N'Activity 2')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'ACT3', N'ACT', N'Activity 3')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'ASS', N'ASS', N'Assignment')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'ASS1', N'ASS', N'Assignment 1')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'ASS2', N'ASS', N'Assignment 2')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'ASS3', N'ASS', N'Assignment 3')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'ASS4', N'ASS', N'Assignment 4')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'ASS5', N'ASS', N'Assignment 5')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'ASS6', N'ASS', N'Assignment 6')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'ASS7', N'ASS', N'Assignment 7')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'ASS8', N'ASS', N'Assignment 8')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'FE', N'FE', N'Final exam')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'GA1', N'GA', N'Group asm 1')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'GA2', N'GA', N'Group asm 2')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
```

```
VALUES (N'GP1', N'GP', N'Group Project 1')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'GP2', N'GP', N'Group Project 2')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'GP3', N'GP', N'Group Project 3')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'LAB1', N'LAB', N'Lab 1')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'LAB2', N'LAB', N'Lab 2')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'LAB3', N'LAB', N'Lab 3')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'LAB4', N'LAB', N'Lab 4')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'LAB5', N'LAB', N'Lab 5')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'LAB6', N'LAB', N'Lab 6')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'ME', N'ME', N'Mid-term test')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'PA', N'PA', N'Participation')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'PE', N'PE', N'Practice Exam')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'PEc', N'PEc', N'Practice Exam')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'PRE1', N'PRE', N'Presentation 1')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'PRE2', N'PRE', N'Presentation 2')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'PRE3', N'PRE', N'Presentation 3')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'PRE4', N'PRE', N'Presentation 4')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'PRE5', N'PRE', N'Presentation 5')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'PT1', N'PT', N'Progress Test 1')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'PT2', N'PT', N'Progress Test 2')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'PT3', N'PT', N'Progress Test 3')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
```

```
VALUES (N'QUIZ', N'QUIZ', N'Quiz')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'TE', N'TE', N'Theory Exam')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'WS1', N'WS', N'Workshop 1')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'WS2', N'WS', N'Workshop 2')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'WS3', N'WS', N'Workshop 3')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'WS4', N'WS', N'Workshop 4')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'WS5', N'WS', N'Workshop 5')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'WS6', N'WS', N'Workshop 6')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'WS7', N'WS', N'Workshop 7')
INSERT [dbo].[CategoryDetails] ([cdID], [CategoryID], [CDName])
VALUES (N'WS8', N'WS', N'Workshop 8')
GO
INSERT [dbo].[Assesment] ([AssID], [CDID], [CourseID], [Duration],
[Weight]) VALUES (N'CSI_FE', N'FE', N'CSI104', CAST(N'01:00:00' AS
Time), 40)
INSERT [dbo].[Assesment] ([AssID], [CDID], [CourseID], [Duration],
[Weight]) VALUES (N'CSI_LAB1', N'LAB1', N'CSI104', CAST(N'01:00:00'
AS Time), 10)
INSERT [dbo].[Assesment] ([AssID], [CDID], [CourseID], [Duration],
[Weight]) VALUES (N'CSI_LAB2', N'LAB2', N'CSI104', CAST(N'01:00:00'
AS Time), 10)
INSERT [dbo].[Assesment] ([AssID], [CDID], [CourseID], [Duration],
[Weight]) VALUES (N'CSI_PRE1', N'PRE1', N'CSI104', CAST(N'00:10:00'
AS Time), 5)
INSERT [dbo].[Assesment] ([AssID], [CDID], [CourseID], [Duration],
[Weight]) VALUES (N'CSI_PRE2', N'PRE2', N'CSI104', CAST(N'00:10:00'
AS Time), 5)
INSERT [dbo].[Assesment] ([AssID], [CDID], [CourseID], [Duration],
[Weight]) VALUES (N'CSI_PT1', N'PT1', N'CSI104', CAST(N'01:00:00' AS
Time), 15)
INSERT [dbo].[Assesment] ([AssID], [CDID], [CourseID], [Duration],
[Weight]) VALUES (N'CSI_PT2', N'PT2', N'CSI104', CAST(N'01:00:00' AS
Time), 15)
INSERT [dbo].[Assesment] ([AssID], [CDID], [CourseID], [Duration],
[Weight]) VALUES (N'MAE_ASS1', N'ASS1', N'MAE101', CAST(N'00:20:00'
AS Time), 10)
INSERT [dbo].[Assesment] ([AssID], [CDID], [CourseID], [Duration],
[Weight]) VALUES (N'MAE_ASS2', N'ASS2', N'MAE101', CAST(N'00:20:00'
AS Time), 10)
INSERT [dbo].[Assesment] ([AssID], [CDID], [CourseID], [Duration],
[Weight]) VALUES (N'MAE_ASS3', N'ASS3', N'MAE101', CAST(N'00:20:00'
AS Time), 10)
INSERT [dbo].[Assesment] ([AssID], [CDID], [CourseID], [Duration],
[Weight]) VALUES (N'MAE_FE', N'FE', N'MAE101', CAST(N'01:00:00' AS
Time), 40)
INSERT [dbo].[Assesment] ([AssID], [CDID], [CourseID], [Duration],
[Weight]) VALUES (N'MAE_PT1', N'PT1', N'MAE101', CAST(N'00:20:00' AS
Time), 10)
INSERT [dbo].[Assesment] ([AssID], [CDID], [CourseID], [Duration],
[Weight]) VALUES (N'MAE_PT2', N'PT2', N'MAE101', CAST(N'00:20:00' AS
Time), 10)
INSERT [dbo].[Assesment] ([AssID], [CDID], [CourseID], [Duration],
[Weight]) VALUES (N'MAE_PT3', N'PT3', N'MAE101', CAST(N'00:20:00' AS
Time), 10)
INSERT [dbo].[Assesment] ([AssID], [CDID], [CourseID], [Duration],
[Weight]) VALUES (N'PRF_ASS', N'ASS', N'PRF192', CAST(N'01:00:00' AS
Time), 10)
INSERT [dbo].[Assesment] ([AssID], [CDID], [CourseID], [Duration],
[Weight]) VALUES (N'PRF_FE', N'FE', N'PRF192', CAST(N'01:00:00' AS
Time), 30)
INSERT [dbo].[Assesment] ([AssID], [CDID], [CourseID], [Duration],
[Weight]) VALUES (N'PRF_PE', N'PE', N'PRF192', CAST(N'01:30:00' AS
Time), 40)
INSERT [dbo].[Assesment] ([AssID], [CDID], [CourseID], [Duration],
[Weight]) VALUES (N'PRF_PT1', N'PT1', N'PRF192', CAST(N'00:25:00' AS
Time), 5)
INSERT [dbo].[Assesment] ([AssID], [CDID], [CourseID], [Duration],
[Weight]) VALUES (N'PRF_PT2', N'PT2', N'PRF192', CAST(N'00:25:00' AS
Time), 5)
INSERT [dbo].[Assesment] ([AssID], [CDID], [CourseID], [Duration],
[Weight]) VALUES (N'PRF_WS1', N'WS1', N'PRF192', CAST(N'02:00:00' AS
Time), 1.3)
INSERT [dbo].[Assesment] ([AssID], [CDID], [CourseID], [Duration],
[Weight]) VALUES (N'PRF_WS2', N'WS2', N'PRF192', CAST(N'02:00:00' AS
Time), 1.3)
INSERT [dbo].[Assesment] ([AssID], [CDID], [CourseID], [Duration],
[Weight]) VALUES (N'PRF_WS3', N'WS3', N'PRF192', CAST(N'02:00:00' AS
Time), 1.3)
INSERT [dbo].[Assesment] ([AssID], [CDID], [CourseID], [Duration],
[Weight]) VALUES (N'PRF_WS4', N'WS4', N'PRF192', CAST(N'02:00:00' AS
Time), 1.3)
INSERT [dbo].[Assesment] ([AssID], [CDID], [CourseID], [Duration],
[Weight]) VALUES (N'PRF_WS5', N'WS5', N'PRF192', CAST(N'02:00:00' AS
Time), 1.3)
INSERT [dbo].[Assesment] ([AssID], [CDID], [CourseID], [Duration],
[Weight]) VALUES (N'PRF_WS6', N'WS6', N'PRF192', CAST(N'02:00:00' AS
Time), 1.3)
INSERT [dbo].[Assesment] ([AssID], [CDID], [CourseID], [Duration],
[Weight]) VALUES (N'PRF_WS7', N'WS7', N'PRF192', CAST(N'02:00:00' AS
Time), 1.3)
INSERT [dbo].[Assesment] ([AssID], [CDID], [CourseID], [Duration],
[Weight]) VALUES (N'PRF_WS8', N'WS8', N'PRF192', CAST(N'02:00:00' AS
Time), 1.3)
GO
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111111', N'MAE_ASS1', 9.5, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111112', N'MAE_ASS1', 9, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111113', N'MAE_ASS1', 9, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111114', N'MAE_ASS1', 9, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111111', N'MAE_ASS2', 10, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111112', N'MAE_ASS2', 1, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111113', N'MAE_ASS2', 9, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111114', N'MAE_ASS2', 8, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111111', N'MAE_ASS3', 10, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111112', N'MAE_ASS3', 9, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111113', N'MAE_ASS3', 9, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111114', N'MAE_ASS3', 8, NULL)
```

```
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111111', N'MAE_FE', 8.4, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111112', N'MAE_FE', 8, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111113', N'MAE_FE', 8, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111114', N'MAE_FE', 7, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111111', N'MAE_PT1', 10, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111112', N'MAE_PT1', 10, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111113', N'MAE_PT1', 9, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111114', N'MAE_PT1', 8, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111111', N'MAE_PT2', 10, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111112', N'MAE_PT2', 9, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111113', N'MAE_PT2', 7, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111114', N'MAE_PT2', 8, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111111', N'MAE_PT3', 9.3, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111112', N'MAE_PT3', 10, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111113', N'MAE_PT3', 9, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111114', N'MAE_PT3', 6, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111111', N'PRF_ASS', 9.5, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111112', N'PRF_ASS', 0, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111113', N'PRF_ASS', 9, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111114', N'PRF_ASS', 10, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111111', N'PRF_FE', 9, NULL)
```

```
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111112', N'PRF_FE', 9, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111113', N'PRF_FE', 6, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111114', N'PRF_FE', 9, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111111', N'PRF_PE', 10, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111112', N'PRF_PE', 8, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111113', N'PRF_PE', 7, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111114', N'PRF_PE', 4, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111111', N'PRF_PT1', 8.6, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111112', N'PRF_PT1', 8, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111113', N'PRF_PT1', 8, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111114', N'PRF_PT1', 9, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111111', N'PRF_PT2', 7.8, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111112', N'PRF_PT2', 9, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111113', N'PRF_PT2', 9, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111114', N'PRF_PT2', 9, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111111', N'PRF_WS1', 10, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111112', N'PRF_WS1', 10, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111113', N'PRF_WS1', 10, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111114', N'PRF_WS1', 7, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111111', N'PRF_WS2', 10, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111112', N'PRF_WS2', 9, NULL)
```

```
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111113', N'PRF_WS2', 10, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111114', N'PRF_WS2', 8, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111111', N'PRF_WS3', 8.5, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111112', N'PRF_WS3', 8, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111113', N'PRF_WS3', 9, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111114', N'PRF_WS3', 7, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111111', N'PRF_WS4', 7, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111112', N'PRF_WS4', 7, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111113', N'PRF_WS4', 6, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111114', N'PRF_WS4', 5, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111111', N'PRF_WS5', 10, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111112', N'PRF_WS5', 6, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111113', N'PRF_WS5', 7, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111114', N'PRF_WS5', 4, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111111', N'PRF_WS6', 7, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111112', N'PRF_WS6', 5, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111113', N'PRF_WS6', 9, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111114', N'PRF_WS6', 4, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111111', N'PRF_WS7', 8.2, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111112', N'PRF_WS7', 4, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111113', N'PRF_WS7', 7, NULL)
```

```
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111114', N'PRF_WS7', 6, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111111', N'PRF_WS8', 10, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111112', N'PRF_WS8', 5, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111113', N'PRF_WS8', 8, NULL)
INSERT [dbo].[Grade] ([sID], [AssID], [Score], [DateExam]) VALUES
(N'HE111114', N'PRF_WS8', 8, NULL)
GO
```

Query Requirements And Results

```
-- update average
go
DECLARE @courseID varchar(10);
DECLARE @sID char(8);
DECLARE update_total_cursor CURSOR FOR
SELECT CourseID, sID FROM [View];
OPEN update_total_cursor;
FETCH NEXT FROM update_total_cursor INTO @courseiD, @sID
WHILE @@FETCH_STATUS = 0
BEGIN
    DECLARE @total float;
    SELECT @total = sum(tbl1.Weight/100 * Score)  FROM
    (SELECT a.*, g.Score, g.sID FROM Assessment a
```

```sql
     INNER JOIN Grade g on a.AssID = g.AssID ) tbl1 WHERE CourseID =
@courseID and sid = @sid;
     UPDATE [View] SET Average = @TOTAL WHERE CourseID = @courseID
and sid = @sid;
     FETCH NEXT FROM update_total_cursor INTO @courseID, @sid
END
CLOSE update_total_cursor;
DEALLOCATE update_total_cursor;

-- UPDATE STATUS
go
-- Hàm check điều kiện xem có điểm thành phần nào không đủ điều kiện
không?.
CREATE FUNCTION check_pass(@courseID varchar(10), @sID char(8))
RETURNS int
AS
BEGIN
     DECLARE @flag int;
     DECLARE @categoryID varchar(10);
     SET @flag = 0;
     DECLARE check_pass_cursor CURSOR FOR
     SELECT [sID],CourseID, CategoryID FROM
     (
     SELECT g.sID, a.CourseID, c.CategoryID, AVG(Score) as sub_total,
[Completion Criteria] FROM Grade g
     INNER JOIN Assessment a on g.AssID = a.AssID
     INNER JOIN [Category Details] cd on cd.CategoryDetailsID =
a.CategoryDetailsID
     INNER JOIN Category c on c.CategoryID = cd.CategoryID  GROUP BY
CourseID, sID, c.CategoryID, [Completion Criteria]
     ) as tbl1 WHERE  CourseID = @courseID and [sID] = @sID ;
     OPEN check_pass_cursor;
     FETCH NEXT FROM check_pass_cursor INTO @sID, @courseID,
@categoryID
     WHILE @@FETCH_STATUS = 0
          BEGIN
               DECLARE @score fLOAT;
               DECLARE @scoreMin FLOAT
               SELECT @score = sub_total , @scoreMin = [Completion
Criteria] FROM
               (
                    SELECT g.sID, a.CourseID, c.CategoryID,
AVG(Score) as sub_total, [Completion Criteria]
                    FROM Grade g
                    INNER JOIN Assessment a on g.AssID = a.AssID
                    INNER JOIN [Category Details] cd on
cd.CategoryDetailsID = a.CategoryDetailsID
                    INNER JOIN Category c on c.CategoryID =
cd.CategoryID
                    GROUP BY CourseID, sID, c.CategoryID, [Completion
Criteria]
               ) as tbl1 WHERE tbl1.CourseID = @courseID  AND
tbl1.[sID] = @sID AND  CategoryID = @categoryID
               IF @score <= @scoreMin
                    BEGIN
                         set @flag = 1;
                         break;
                    END
               FETCH NEXT FROM check_pass_cursor INTO @courseID,
@sid, @categoryID
          END
     CLOSE check_pass_cursor;
     DEALLOCATE check_pass_cursor;
     return @flag;
END

GO
```

```
-- Stored Procedure update status is passed or not passed.
CREATE PROC update_status_pass
     @courseID varchar(10),
     @sID char(8)
AS
BEGIN
     DECLARE @average1 FLOAT;
     SELECT @average1 = Average FROM [View] WHERE  CourseID =
@courseID  and  [sID] = @sID ;
     IF @average1 > 5 AND dbo.check_pass(@courseID,@sID) = 0
     UPDATE [View] SET [Status] = 'PASSED' WHERE  CourseID =
@courseID  and  [sID] = @sID ;
     ELSE
     UPDATE [View] SET [Status] = 'NOT PASSED' WHERE  CourseID =
@courseID  and  [sID] = @sID ;
END

-- cusor update status while have average.
go
DECLARE @courseID varchar(10);
DECLARE @sID char(8);
DECLARE update_status_cursor1 CURSOR FOR
SELECT CourseID, [sID] FROM [View];
OPEN update_status_cursor1;
FETCH NEXT FROM update_status_cursor1 INTO @courseiD, @sID
WHILE @@FETCH_STATUS = 0
BEGIN
     EXEC update_status_pass @CourseID, @sID
     FETCH NEXT FROM update_status_cursor1 INTO @courseID, @sid
END
CLOSE update_status_cursor1;
DEALLOCATE update_status_cursor1;


--test
SELECT g.sID, a.CourseID, c.CategoryID, AVG(Score) as sub_total,
[Completion Criteria] FROM Grade g
INNER JOIN Assessment a on g.AssID = a.AssID
INNER JOIN [Category Details] cd on cd.CategoryDetailsID =
a.CategoryDetailsID
INNER JOIN Category c on c.CategoryID = cd.CategoryID  GROUP BY
CourseID, sID, c.CategoryID, [Completion Criteria]
Having AVG(Score) >5
-- test
GO
-- procedure calculator sub_total
CREATE PROC select_sub_total
AS
BEGIN
     SELECT g.sID, a.CourseID, c.CategoryID, AVG(Score) as sub_total,
[Completion Criteria] FROM Grade g
     INNER JOIN Assessment a on g.AssID = a.AssID
     INNER JOIN [Category Details] cd on cd.CategoryDetailsID =
a.CategoryDetailsID
     INNER JOIN Category c on c.CategoryID = cd.CategoryID  GROUP BY
CourseID, sID, c.CategoryID, [Completion Criteria]
END
GO

EXEC select_sub_total



-- TRIGGER WHILE INPUT DATA AVERAGE OR STATUS--
GO
Drop TRIGGER View_Average ON [View]
AFTER INSERT, UPDATE
AS
DECLARE @AVG FLOAT;
DECLARE @courseID VARCHAR(10);
DECLARE @ses varchar(10)
DECLARE @sID char(8);
DECLARE @average FLOAT;
DECLARE @status VARCHAR(20);

SELECT @sID = sID, @courseID = CourseID, @ses = Semester,
         @average = Average, @status = [Status]
FROM inserted;
SELECT @AVG = sum(tbl1.Weight/100 * Score) FROM
         (SELECT a.*, g.Score, g.sID FROM Assessment a
             INNER JOIN Grade g on a.AssID = g.AssID  WHERE sID =
@sID and CourseID = @courseID
              ) as tbl1 group by sID, CourseID
IF @AVG <> @average
BEGIN
     PRINT 'Conflict input data'
     ROLLBACK TRAN
END
ELSE IF (NOT @status = 'PASSED') AND (NOT @status = 'NOT PASSED')
BEGIN
     PRINT 'Status must be passed or not passed'
     ROLLBACK TRAN
END
ELSE IF (@AVG <= 5 AND @status = 'PASSED') OR (@AVG > 5 AND @status
= 'NOT PASSED')
```

```
BEGIN
     PRINT 'Incorrect Status'
     ROLLBACK TRAN
END

UPDATE [View] SET Average = 6.5, [Status] = 'PASSED', Semester
='Fall21' WHERE sID = 'HE163745' AND CourseID = 'MAE101'

SELECT * FROM [View]
```

10 Querry

```
-- SELECT ra total use ORDER BY, INNER JOIN, HAVING, Sub-query
SELECT CourseID, sID , sum(tbl1.Weight/100 * Score) as total
FROM
     (SELECT a.*, g.Score, g.sID FROM Assessment a
     INNER JOIN Grade g on a.AssID = g.AssID ) tbl1
GROUP BY [sID], CourseID
HAVING sum(tbl1.Weight/100 * Score) > 7 ORDER BY [sID]

-- Count number student.
SELECT Count(*) FROM Student
--A query that uses a sub-query in the WHERE clause to display
student max age in all student
SELECT *
FROM Student st
WHERE DOB <= ALL (SELECT DOB FROM Student);

-- A query that uses partial matching in the WHERE clause FIND
student [First Name] have characteristic 'V'
SELECT * FROM Student WHERE [First Name] LIKE 'V%'

-- A query that uses a self-JOIN to display student same city
SELECT  s2.Address,     s1.[First Name] + ' ' +s1.Last Name] as
fullName,
```

```sql
s2.[First Name] + ' ' +s2.Last Name] as fullName
FROM Student s1
INNER JOIN Student s2 ON s1.sID > s2.sID  AND s1.Address =
s2.Address



-- Stored Procedure update status is passed or not passed.
CREATE PROC update_status_pass
    @courseID varchar(10),
    @sID char(8)
AS
BEGIN
    DECLARE @average1 FLOAT;
    SELECT @average1 = Average FROM [View] WHERE  CourseID =
@courseID  and  [sID] = @sID ;
    IF @average1 > 5 AND dbo.check_pass(@courseID,@sID) = 0
    UPDATE [View] SET [Status] = 'PASSED' WHERE  CourseID =
@courseID  and  [sID] = @sID ;
    ELSE
    UPDATE [View] SET [Status] = 'NOT PASSED' WHERE  CourseID =
@courseID  and  [sID] = @sID ;
END

-- cusor update status while have average.
go
DECLARE @courseID varchar(10);
DECLARE @sID char(8);
DECLARE update_status_cursor1 CURSOR FOR
SELECT CourseID, [sID] FROM [View];
OPEN update_status_cursor1;
FETCH NEXT FROM update_status_cursor1 INTO @courseiD, @sID
WHILE @@FETCH_STATUS = 0
BEGIN
    EXEC update_status_pass @CourseID, @sID
    FETCH NEXT FROM update_status_cursor1 INTO @courseID, @sid
END
CLOSE update_status_cursor1;
DEALLOCATE update_status_cursor1;


--test
```

```sql
SELECT g.sID, a.CourseID, c.CategoryID, AVG(Score) as sub_total,
[Completion Criteria] FROM Grade g
INNER JOIN Assessment a on g.AssID = a.AssID
INNER JOIN [Category Details] cd on cd.CategoryDetailsID =
a.CategoryDetailsID
INNER JOIN Category c on c.CategoryID = cd.CategoryID  GROUP BY
CourseID, sID, c.CategoryID, [Completion Criteria]
Having AVG(Score) >5
-- test
GO
-- procedure calculator sub_total
CREATE PROC select_sub_total
AS
BEGIN
     SELECT g.sID, a.CourseID, c.CategoryID, AVG(Score) as sub_total,
[Completion Criteria] FROM Grade g
     INNER JOIN Assessment a on g.AssID = a.AssID
     INNER JOIN [Category Details] cd on cd.CategoryDetailsID =
a.CategoryDetailsID
     INNER JOIN Category c on c.CategoryID = cd.CategoryID  GROUP BY
CourseID, sID, c.CategoryID, [Completion Criteria]
END
GO

EXEC select_sub_total



-- TRIGGER WHILE INPUT DATA AVERAGE OR STATUS--
GO
Drop TRIGGER View_Average ON [View]
AFTER INSERT, UPDATE
AS
DECLARE @AVG FLOAT;
DECLARE @courseID VARCHAR(10);
DECLARE @ses varchar(10)
DECLARE @sID char(8);
DECLARE @average FLOAT;
DECLARE @status VARCHAR(20);

SELECT @sID = sID, @courseID = CourseID, @ses = Semester,
         @average = Average, @status = [Status]
```

```
FROM inserted;
SELECT @AVG = sum(tbl1.Weight/100 * Score) FROM
          (SELECT a.*, g.Score, g.sID FROM Assessment a
              INNER JOIN Grade g on a.AssID = g.AssID  WHERE sID =
@sID and CourseID = @courseID
              ) as tbl1 group by sID, CourseID
IF @AVG <> @average
BEGIN
     PRINT 'Conflict input data'
     ROLLBACK TRAN
END
ELSE IF (NOT @status = 'PASSED') AND (NOT @status = 'NOT PASSED')
BEGIN
     PRINT 'Status must be passed or not passed'
     ROLLBACK TRAN
END
ELSE IF (@AVG <= 5 AND @status = 'PASSED') OR ((@AVG > 5 AND @status
= 'NOT PASSED')
BEGIN
     PRINT 'Incorrect Status'
     ROLLBACK TRAN
END

UPDATE [View] SET Average = 6.5, [Status] = 'PASSED', Semester
='Fall21' WHERE sID = 'HE163745' AND CourseID = 'MAE101'

SELECT * FROM [View]

-- Index:
Khi tìm tên sinh viên ta sẽ tìm từ đầu đến cuối bảng ghi và so sánh
điều này sẽ gây tốn performance. Chính vì thế ta sẽ lưu bảng student
thành dạng tree bằng cách sử dụng INDEX. Điều này sẽ giúp tìm kiếm
nhanh hơn bằng cách tìm kiếm thông thường
CREATE INDEX student_index ON Student (LastName, FirstName)
```