FINAL REPORT

FPT Education

FPT UNIVERSITY

Student Grading Management Sub-System

Name : Nguyễn Cảnh Thương

Mssv : HE163742

Class : IA1604

Major : Information Asssurance

School : FPT University

Email : thuongnche163742@fpt.edu.vn

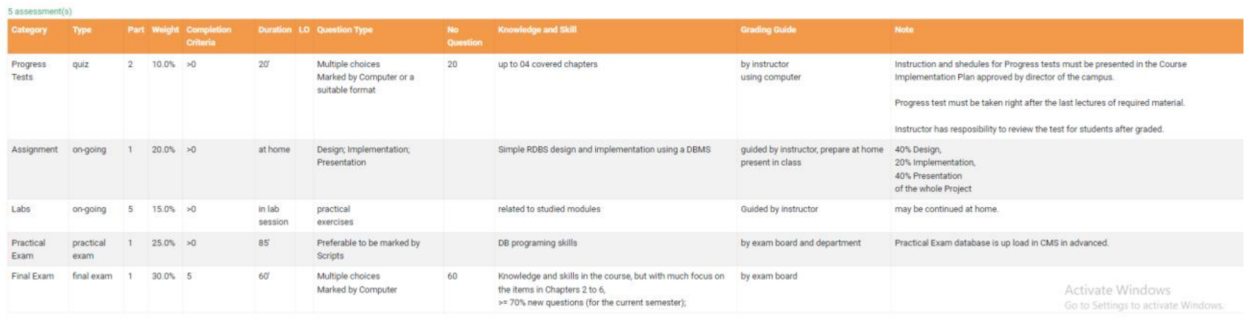
Phone number : 0886019122

[Type here]

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 (hạng mục): Progress Tests , Assignment,Labs,Pe,FE

Type(loại) : Quiz , On-going, PE ,FE

Part(phần): Được làm bao nhiêu lần

Weight(trọng số từng hạng mục)

Completion Criteria(Điều kiện để thi và điểm tối thiểu để pass)

Duration(khoảng thời gian mà cần làm việc trong hạng mục): at home , in lab session

Question Type(Loại câu hỏi) :MTC

No Question(Số câu hỏi)

Knowledge and Skill

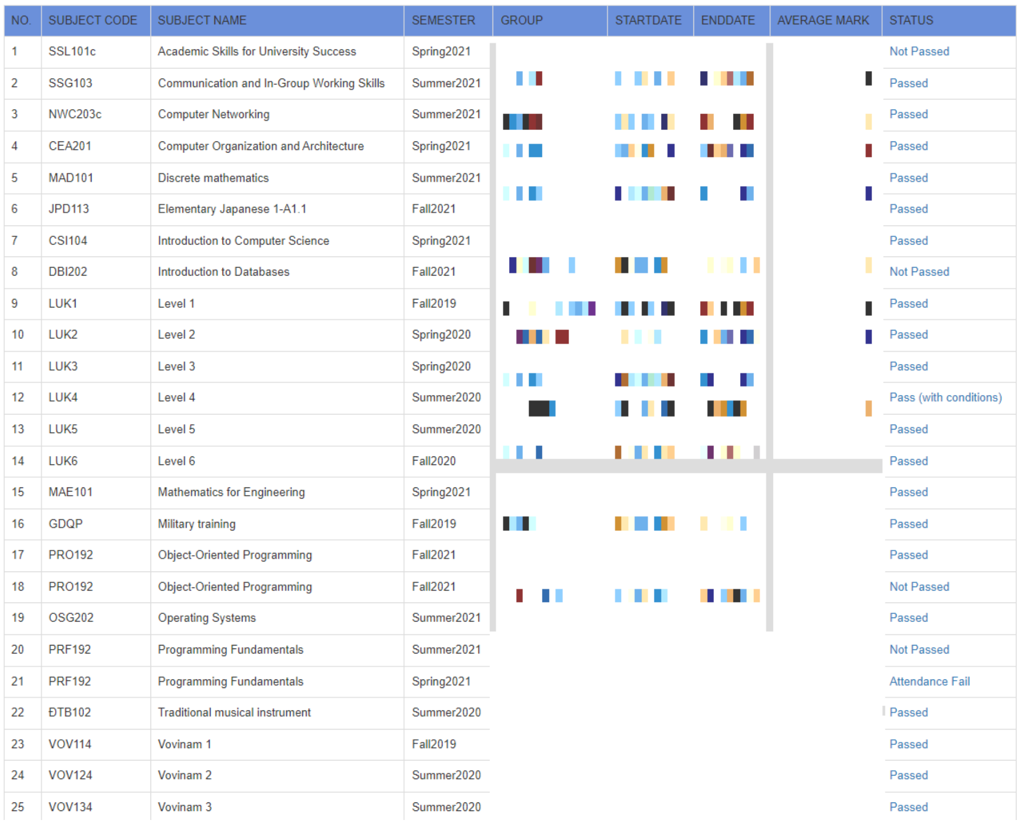
Grading Guide

Note:

[Type here]

Students can check their results at the end of semester as following example:

Table : Subject.(chưa được phân tách)



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	[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					Passed
7	CSI104	Introduction to Computer Science	Spring2021					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]		Passed
12	LUK4	Level 4	Summer2020	[redacted]	[redacted]	[redacted]	[redacted]	Pass (with conditions)
13	LUK5	Level 5	Summer2020					Passed
14	LUK6	Level 6	Fall2020	[redacted]	[redacted]	[redacted]		Passed
15	MAE101	Mathematics for Engineering	Spring2021					Passed
16	GDQP	Military training	Fall2019	[redacted]	[redacted]	[redacted]		Passed
17	PRO192	Object-Oriented Programming	Fall2021					Passed
18	PRO192	Object-Oriented Programming	Fall2021	[redacted]	[redacted]	[redacted]		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

NO: number of subject

Subject code : one subject one code

Subject name : define of subject

Semester : season+year

Group:Lớp học

StartDate:thời điểm bắt đầu môn học

EndDate:thời điểm kết thúc môn học

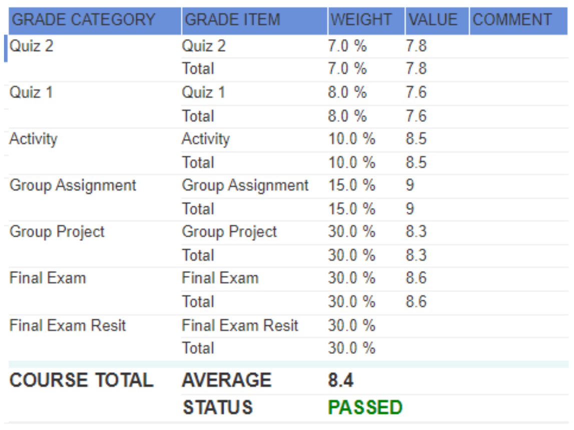
Average mark: điểm trung bình

Status: not passed or passed

Each Subject code, student can check their detailed result of as below example:

Table Result of Mark (chưa được phân tách)

[Type here]



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 (hạng mục) : quiz 2

Grade Item(hạng mục) : thêm 1 row total

Weight: trọng số(cũng có ở bên FML table)

Value: mark.

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,[Completion Criteria],Type)

Grade(SID,AssID,Score,[Date Exam])

Group_Student(Gid,Sid)

Student(Sid,[First name],[Last name],Gender,DOB,Address)

Group(Gid,GName)

Lecturers(LecID,GID,[First Name],[Last Name],Gender)

[Type here]

Course(CourseID,CourseName)

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

Semester(SesID,SesName,StartDate,EndDate)

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.

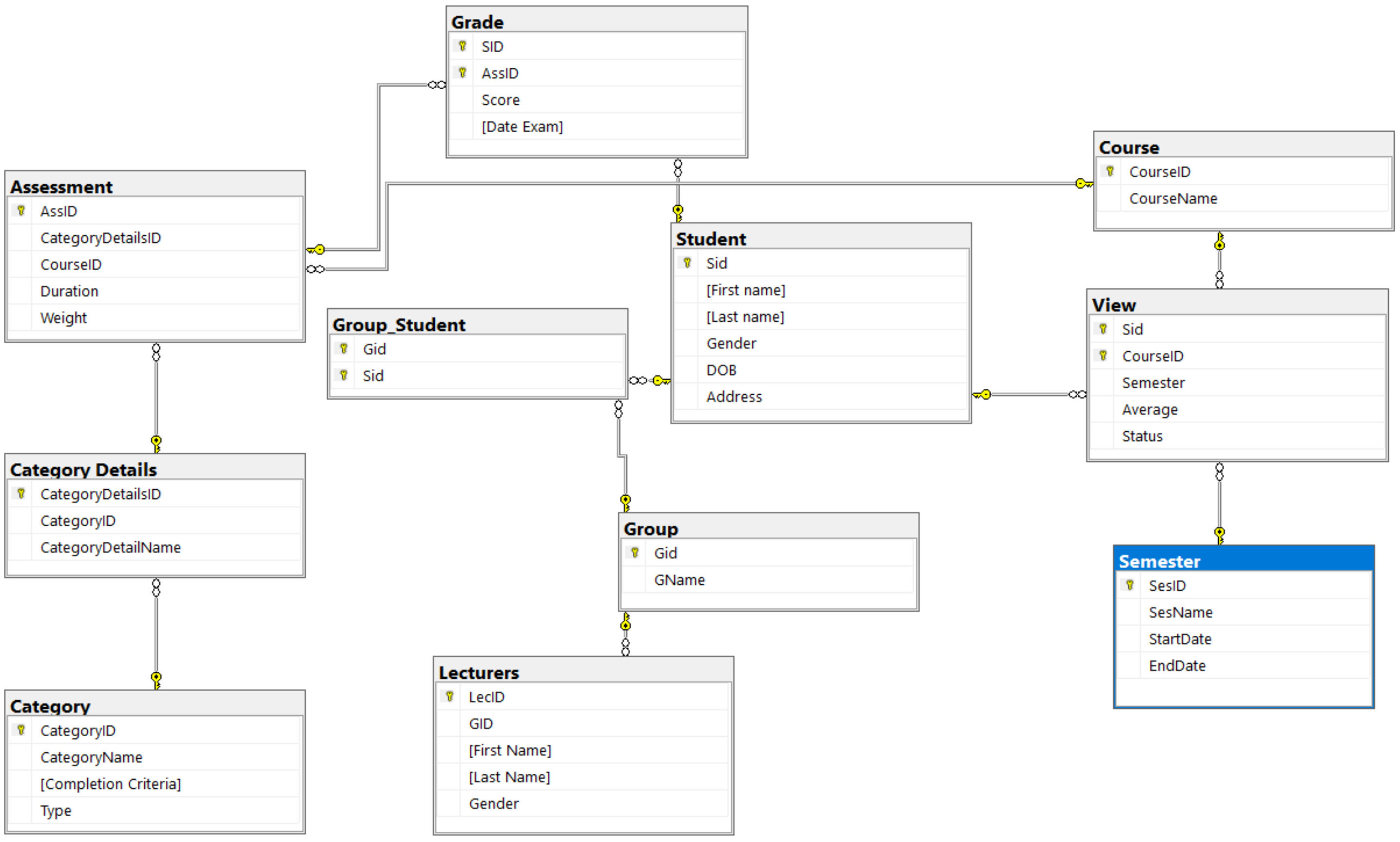
-> Xác Định Quan Hệ Giữa Entity Assessment System Và Entity Courses là quan hệ một nhiều (1-n)

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)

Diagram of Grading Management :



[Type here]

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

[Type here]

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

[Type here]

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

[Type here]

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	

[Type here]

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 Lecturers

Attributes	Data Type	Requires	Key

LecID	Varchar	Not null	Primary
GID	Varchar	Not null	Primary_Foreign Key
First Name	Varchar	Not null	
Last Name	Varchar	Not null	
Gender	bit	Not null	

9.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	

10.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	

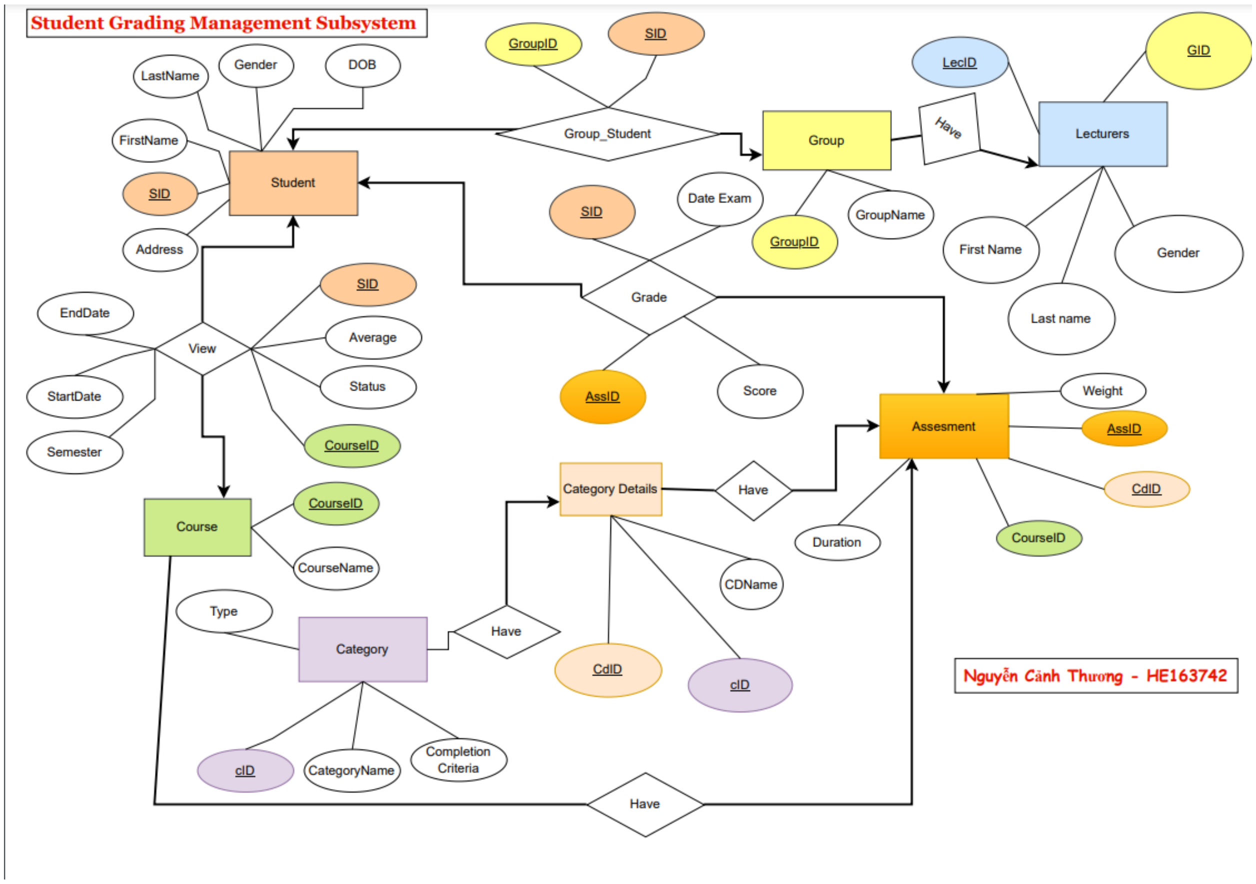
11.Table View

[Type here]

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

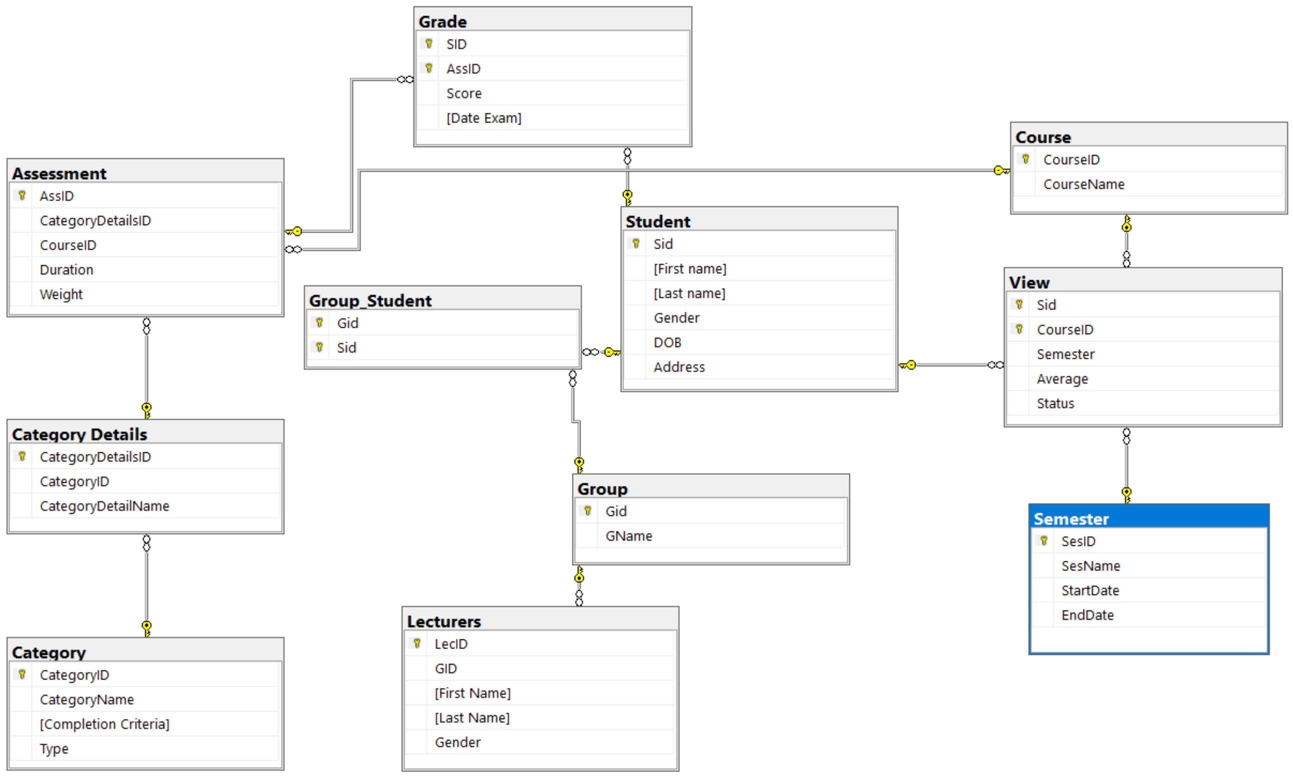


[Type here]

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 Catego
	[CourseID] [varchar](10) NOT NULL,
	[Duration] [nvarchar](30) NOT NULL,
)
```

[Type here]

```sql
CREATE TABLE [dbo].[Category](
	[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,
)
```

```
CREATE DATABASE AND CODE :

USE [Grading  Management ]
GO
INSERT [dbo].[Category] ([CategoryID], [CategoryName], [Completion Criteria],
INSERT [dbo].[Category] ([CategoryID], [CategoryName], [Completion Criteria],
INSERT [dbo].[Category] ([CategoryID], [CategoryName], [Completion Criteria],
```

[Type here]

```
INSERT [dbo].[Category] ([CategoryID], [CategoryName], [Completion Criteria],
INSERT [dbo].[Category] ([CategoryID], [CategoryName], [Completion Criteria],
INSERT [dbo].[Category] ([CategoryID], [CategoryName], [Completion Criteria],
INSERT [dbo].[Category] ([CategoryID], [CategoryName], [Completion Criteria],
INSERT [dbo].[Category] ([CategoryID], [CategoryName], [Completion Criteria],
INSERT [dbo].[Category] ([CategoryID], [CategoryName], [Completion Criteria],
INSERT [dbo].[Category] ([CategoryID], [CategoryName], [Completion Criteria],
INSERT [dbo].[Category] ([CategoryID], [CategoryName], [Completion Criteria],
INSERT [dbo].[Category] ([CategoryID], [CategoryName], [Completion Criteria],
INSERT [dbo].[Category] ([CategoryID], [CategoryName], [Completion Criteria],
INSERT [dbo].[Category] ([CategoryID], [CategoryName], [Completion Criteria],
INSERT [dbo].[Category] ([CategoryID], [CategoryName], [Completion Criteria],
GO
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
INSERT [dbo].[Category Details] ([CategoryDetailsID], [CategoryID], [Category
GO
INSERT [dbo].[Student] ([Sid], [First name], [Last name], [Gender], [DOB], [A
AS Date), N'Nghe An')
INSERT [dbo].[Student] ([Sid], [First name], [Last name], [Gender], [DOB], [A
AS Date), N'Thai Binh')
INSERT [dbo].[Student] ([Sid], [First name], [Last name], [Gender], [DOB], [A
Date), N'Ha Giang')
INSERT [dbo].[Student] ([Sid], [First name], [Last name], [Gender], [DOB], [A
Date), N'Thanh Hoa')
INSERT [dbo].[Student] ([Sid], [First name], [Last name], [Gender], [DOB], [A
AS Date), N'Hai Phong')
INSERT [dbo].[Student] ([Sid], [First name], [Last name], [Gender], [DOB], [A
Date), N'Son Tay')
INSERT [dbo].[Student] ([Sid], [First name], [Last name], [Gender], [DOB], [A
Date), N'Ha Noi')
INSERT [dbo].[Student] ([Sid], [First name], [Last name], [Gender], [DOB], [A
AS Date), N'Nam Dinh')
INSERT [dbo].[Student] ([Sid], [First name], [Last name], [Gender], [DOB], [A
Date), N'Quang Binh')
```

```
INSERT [dbo].[Student] ([Sid], [First name], [Last name], [Gender], [DOB], [A
12-10' AS Date), N'Ha Tinh')
INSERT [dbo].[Student] ([Sid], [First name], [Last name], [Gender], [DOB], [A
AS Date), N'Ninh Binh')
INSERT [dbo].[Student] ([Sid], [First name], [Last name], [Gender], [DOB], [A
Date), N'Bac Giang')
INSERT [dbo].[Student] ([Sid], [First name], [Last name], [Gender], [DOB], [A
AS Date), N'Hung Yen')
INSERT [dbo].[Student] ([Sid], [First name], [Last name], [Gender], [DOB], [A
Date), N'Phu Tho')
INSERT [dbo].[Student] ([Sid], [First name], [Last name], [Gender], [DOB], [A
AS Date), N'Hoa Binh')
GO
INSERT [dbo].[Group] ([Gid], [GName]) VALUES (N'IA1', N'IA1604')
INSERT [dbo].[Group] ([Gid], [GName]) VALUES (N'SE1', N'SE1636')
INSERT [dbo].[Group] ([Gid], [GName]) VALUES (N'SE2', N'SE1647')
GO
INSERT [dbo].[Group_Student] ([Gid], [Sid]) VALUES (N'IA1', N'HE163750')
INSERT [dbo].[Group_Student] ([Gid], [Sid]) VALUES (N'IA1', N'HE163751')
INSERT [dbo].[Group_Student] ([Gid], [Sid]) VALUES (N'IA1', N'HE163752')
INSERT [dbo].[Group_Student] ([Gid], [Sid]) VALUES (N'IA1', N'HE163753')
INSERT [dbo].[Group_Student] ([Gid], [Sid]) VALUES (N'IA1', N'HE163754')
INSERT [dbo].[Group_Student] ([Gid], [Sid]) VALUES (N'SE1', N'HE163740')
INSERT [dbo].[Group_Student] ([Gid], [Sid]) VALUES (N'SE1', N'HE163741')
INSERT [dbo].[Group_Student] ([Gid], [Sid]) VALUES (N'SE1', N'HE163742')
INSERT [dbo].[Group_Student] ([Gid], [Sid]) VALUES (N'SE1', N'HE163743')
INSERT [dbo].[Group_Student] ([Gid], [Sid]) VALUES (N'SE1', N'HE163744')
INSERT [dbo].[Group_Student] ([Gid], [Sid]) VALUES (N'SE2', N'HE163745')
INSERT [dbo].[Group_Student] ([Gid], [Sid]) VALUES (N'SE2', N'HE163746')
INSERT [dbo].[Group_Student] ([Gid], [Sid]) VALUES (N'SE2', N'HE163747')
INSERT [dbo].[Group_Student] ([Gid], [Sid]) VALUES (N'SE2', N'HE163748')
INSERT [dbo].[Group_Student] ([Gid], [Sid]) VALUES (N'SE2', N'HE163749')
GO
INSERT [dbo].[Course] ([CourseID], [CourseName]) VALUES (N'CSI104', N'Introdu
INSERT [dbo].[Course] ([CourseID], [CourseName]) VALUES (N'MAD101', N'Discret
INSERT [dbo].[Course] ([CourseID], [CourseName]) VALUES (N'MAE101', N'Mathema
INSERT [dbo].[Course] ([CourseID], [CourseName]) VALUES (N'PRF192', N'Program
INSERT [dbo].[Course] ([CourseID], [CourseName]) VALUES (N'SSL101c', N'Academ
GO
```

```
INSERT [dbo].[Assessment] ([AssID], [CategoryDetailsID], [CourseID], [Duratio
INSERT [dbo].[Assessment] ([AssID], [CategoryDetailsID], [CourseID], [Duratio
INSERT [dbo].[Assessment] ([AssID], [CategoryDetailsID], [CourseID], [Duratio
INSERT [dbo].[Assessment] ([AssID], [CategoryDetailsID], [CourseID], [Duratio
INSERT [dbo].[Assessment] ([AssID], [CategoryDetailsID], [CourseID], [Duratio
INSERT [dbo].[Assessment] ([AssID], [CategoryDetailsID], [CourseID], [Duratio
INSERT [dbo].[Assessment] ([AssID], [CategoryDetailsID], [CourseID], [Duratio
INSERT [dbo].[Assessment] ([AssID], [CategoryDetailsID], [CourseID], [Duratio
INSERT [dbo].[Assessment] ([AssID], [CategoryDetailsID], [CourseID], [Duratio
INSERT [dbo].[Assessment] ([AssID], [CategoryDetailsID], [CourseID], [Duratio
INSERT [dbo].[Assessment] ([AssID], [CategoryDetailsID], [CourseID], [Duratio
INSERT [dbo].[Assessment] ([AssID], [CategoryDetailsID], [CourseID], [Duratio
INSERT [dbo].[Assessment] ([AssID], [CategoryDetailsID], [CourseID], [Duratio
INSERT [dbo].[Assessment] ([AssID], [CategoryDetailsID], [CourseID], [Duratio
INSERT [dbo].[Assessment] ([AssID], [CategoryDetailsID], [CourseID], [Duratio
INSERT [dbo].[Assessment] ([AssID], [CategoryDetailsID], [CourseID], [Duratio
INSERT [dbo].[Assessment] ([AssID], [CategoryDetailsID], [CourseID], [Duratio
INSERT [dbo].[Assessment] ([AssID], [CategoryDetailsID], [CourseID], [Duratio
INSERT [dbo].[Assessment] ([AssID], [CategoryDetailsID], [CourseID], [Duratio
INSERT [dbo].[Assessment] ([AssID], [CategoryDetailsID], [CourseID], [Duratio
INSERT [dbo].[Assessment] ([AssID], [CategoryDetailsID], [CourseID], [Duratio
INSERT [dbo].[Assessment] ([AssID], [CategoryDetailsID], [CourseID], [Duratio
INSERT [dbo].[Assessment] ([AssID], [CategoryDetailsID], [CourseID], [Duratio
INSERT [dbo].[Assessment] ([AssID], [CategoryDetailsID], [CourseID], [Duratio
INSERT [dbo].[Assessment] ([AssID], [CategoryDetailsID], [CourseID], [Duratio
INSERT [dbo].[Assessment] ([AssID], [CategoryDetailsID], [CourseID], [Duratio
GO
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
GO
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16374
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16375
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16375
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16375
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16375
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16375
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16375
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16375
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16375
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16375
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16375
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16375
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16375
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16375
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16375
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16375
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16375
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16375
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16375
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16375
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16375
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16375
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16375
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16375
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16375
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16375
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16375
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16375
INSERT [dbo].[Grade] ([SID], [AssID], [Score], [Date Exam]) VALUES (N'HE16375
GO
INSERT [dbo].[Semester] ([SesID], [SesName], [StartDate], [EndDate]) VALUES (
Date))
INSERT [dbo].[Semester] ([SesID], [SesName], [StartDate], [EndDate]) VALUES (
Date))
INSERT [dbo].[Semester] ([SesID], [SesName], [StartDate], [EndDate]) VALUES (
Date))
INSERT [dbo].[Semester] ([SesID], [SesName], [StartDate], [EndDate]) VALUES (
Date))
GO
INSERT [dbo].[View] ([Sid], [CourseID], [Semester], [Average], [Status]) VALU
INSERT [dbo].[View] ([Sid], [CourseID], [Semester], [Average], [Status]) VALU
INSERT [dbo].[View] ([Sid], [CourseID], [Semester], [Average], [Status]) VALU
INSERT [dbo].[View] ([Sid], [CourseID], [Semester], [Average], [Status]) VALU
INSERT [dbo].[View] ([Sid], [CourseID], [Semester], [Average], [Status]) VALU
INSERT [dbo].[View] ([Sid], [CourseID], [Semester], [Average], [Status]) VALU
INSERT [dbo].[View] ([Sid], [CourseID], [Semester], [Average], [Status]) VALU
INSERT [dbo].[View] ([Sid], [CourseID], [Semester], [Average], [Status]) VALU
INSERT [dbo].[View] ([Sid], [CourseID], [Semester], [Average], [Status]) VALU
INSERT [dbo].[View] ([Sid], [CourseID], [Semester], [Average], [Status]) VALU
INSERT [dbo].[View] ([Sid], [CourseID], [Semester], [Average], [Status]) VALU
GO
INSERT [dbo].[Lecturers] ([LecID], [GID], [First Name], [Last Name], [Gender]
INSERT [dbo].[Lecturers] ([LecID], [GID], [First Name], [Last Name], [Gender]
INSERT [dbo].[Lecturers] ([LecID], [GID], [First Name], [Last Name], [Gender]
GO
```

Some code to completion the Assignment

```
-- 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
     INNER JOIN Grade g on a.AssID = g.AssID ) tbl1 WHERE CourseID = @courseI
     UPDATE [View] SET Average = @TOTAL WHERE CourseID = @courseID and 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, [Comple
    INNER JOIN Assessment a on g.AssID = a.AssID
    INNER JOIN [Category Details] cd on cd.CategoryDetailsID = a.CategoryDet
    INNER JOIN Category c on c.CategoryID = cd.CategoryID  GROUP BY CourseID
    ) 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] F
            (
                SELECT g.sID, a.CourseID, c.CategoryID, AVG(Score) as sub_
                FROM Grade g
                INNER JOIN Assessment a on g.AssID = a.AssID
                INNER JOIN [Category Details] cd on cd.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] = @s
            IF @score <= @scoreMin
                BEGIN
                    set @flag = 1;
                    break;
                END
            FETCH NEXT FROM check_pass_cursor INTO @courseID, @sid, @catego
        END
    CLOSE check_pass_cursor;
    DEALLOCATE check_pass_cursor;
    return @flag;
END

GO
```

[Type here]

```
-- 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
     IF @average1 > 5 AND dbo.check_pass(@courseID,@sID) = 0
     UPDATE [View] SET [Status] = 'PASSED' WHERE  CourseID = @courseID  and
     ELSE
     UPDATE [View] SET [Status] = 'NOT PASSED' WHERE  CourseID = @courseID  a
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
INNER JOIN Assessment a on g.AssID = a.AssID
INNER JOIN [Category Details] cd on cd.CategoryDetailsID = a.CategoryDetailsI
INNER JOIN Category c on c.CategoryID = cd.CategoryID  GROUP BY CourseID, sID
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, [Comple
     INNER JOIN Assessment a on g.AssID = a.AssID
     INNER JOIN [Category Details] cd on cd.CategoryDetailsID = a.CategoryDet
     INNER JOIN Category c on c.CategoryID = cd.CategoryID  GROUP BY CourseID
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 C
              ) as tbl1 group by sID, CourseID
IF @AVG <> @average
BEGIN
     PRINT 'Conflict input data'
```

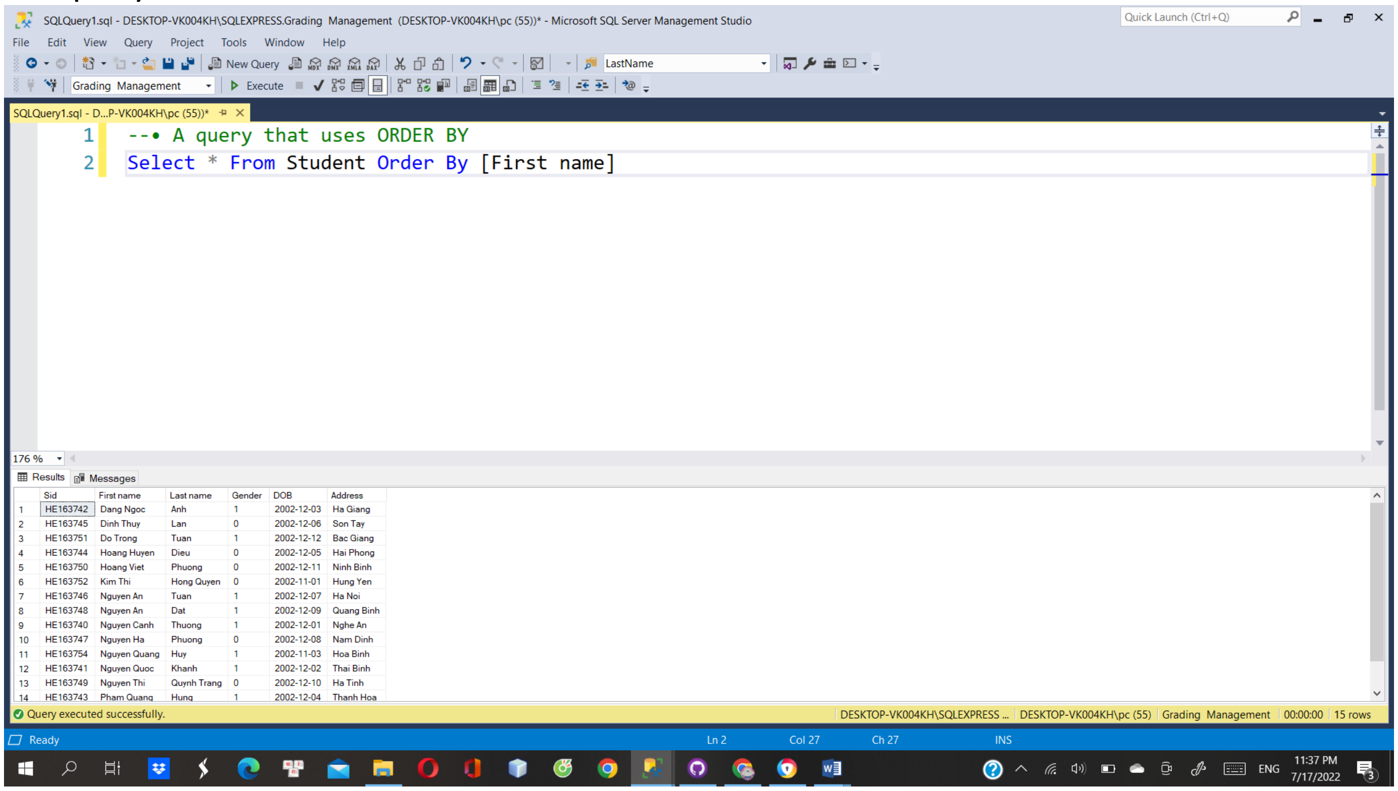
[Type here]

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

UPDATE [View] SET Average = 6.5, [Status] = 'PASSED', Semester ='Fall21' WHEF

SELECT * FROM [View]
```

[Type here]

10 Query

• A query that uses ORDER BY31



SQLQuery1.sql - DESKTOP-VK004KH\SQLEXPRESS.Grading Management (DESKTOP-VK004KH\pc (55))* - Microsoft SQL Server Management Studio

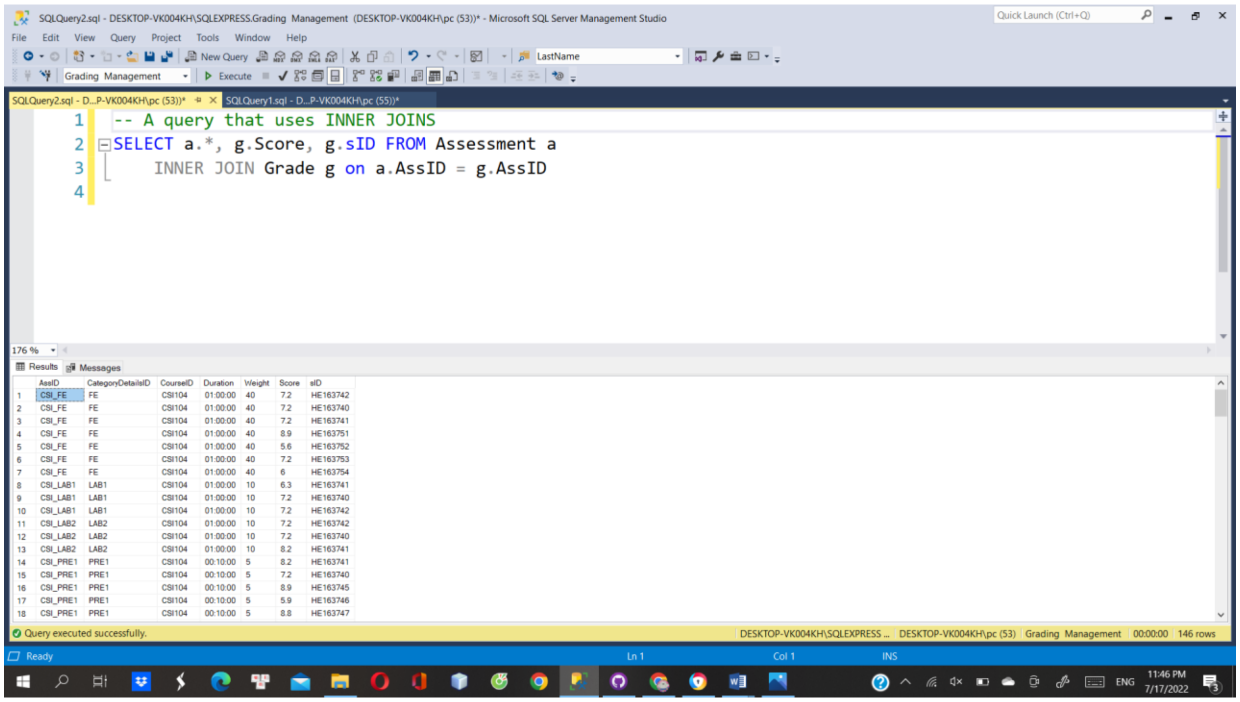
```sql
--• A query that uses ORDER BY
Select * From Student Order By [First name]
```

	Sid	First name	Last name	Gender	DOB	Address
1	HE163742	Dang Ngoc	Anh	1	2002-12-03	Ha Giang
2	HE163745	Dinh Thuy	Lan	0	2002-12-06	Son Tay
3	HE163751	Do Trong	Tuan	1	2002-12-12	Bac Giang
4	HE163744	Hoang Huyen	Dieu	0	2002-12-05	Hai Phong
5	HE163750	Hoang Viet	Phuong	0	2002-12-11	Ninh Binh
6	HE163752	Kim Thi	Hong Quyen	0	2002-11-01	Hung Yen
7	HE163746	Nguyen An	Tuan	1	2002-12-07	Ha Noi
8	HE163748	Nguyen An	Dat	1	2002-12-09	Quang Binh
9	HE163740	Nguyen Canh	Thuong	1	2002-12-01	Nghe An
10	HE163747	Nguyen Ha	Phuong	0	2002-12-08	Nam Dinh
11	HE163754	Nguyen Quang	Huy	1	2002-11-03	Hoa Binh
12	HE163741	Nguyen Quoc	Khanh	1	2002-12-02	Thai Binh
13	HE163749	Nguyen Thi	Quynh Trang	0	2002-12-10	Ha Tinh
14	HE163743	Pham Quang	Hung	1	2002-12-04	Thanh Hoa

Query executed successfully. DESKTOP-VK004KH\SQLEXPRESS ... DESKTOP-VK004KH\pc (55) Grading Management 00:00:00 15 rows

[Type here]

• A query that uses INNER JOINS


SQLQuery2.sql - DESKTOP-VK004KH\SQLEXPRESS.Grading Management (DESKTOP-VK004KH\pc (53))* - Microsoft SQL Server Management Studio

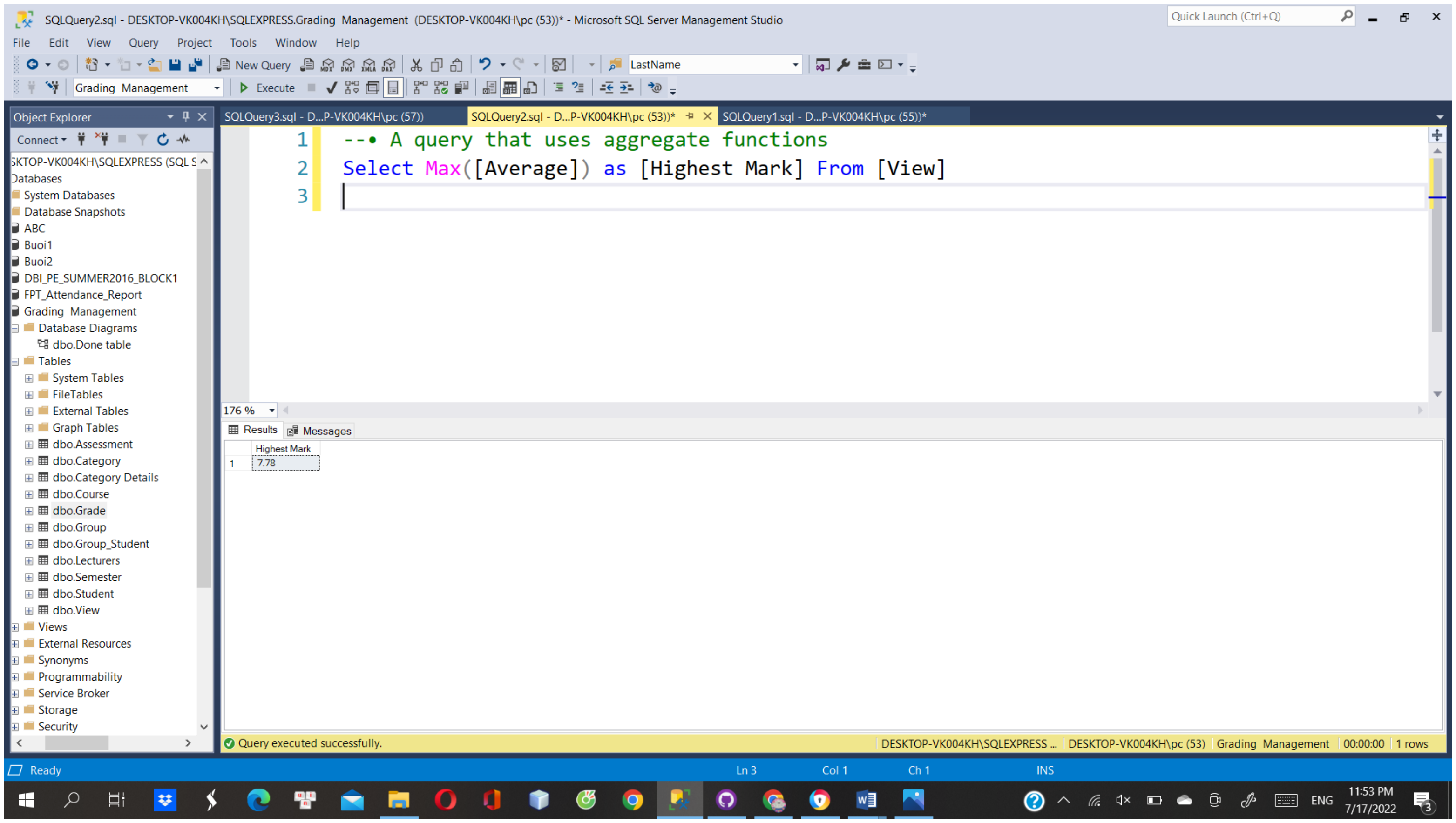
```sql
-- A query that uses INNER JOINS
SELECT a.*, g.Score, g.sID FROM Assessment a
    INNER JOIN Grade g on a.AssID = g.AssID
```

	AssID	CategoryDetailsID	CourseID	Duration	Weight	Score	sID
1	CSI_FE	FE	CSI104	01:00:00	40	7.2	HE163742
2	CSI_FE	FE	CSI104	01:00:00	40	7.2	HE163740
3	CSI_FE	FE	CSI104	01:00:00	40	7.2	HE163741
4	CSI_FE	FE	CSI104	01:00:00	40	8.9	HE163751
5	CSI_FE	FE	CSI104	01:00:00	40	5.6	HE163752
6	CSI_FE	FE	CSI104	01:00:00	40	7.2	HE163753
7	CSI_FE	FE	CSI104	01:00:00	40	6	HE163754
8	CSI_LAB1	LAB1	CSI104	01:00:00	10	6.3	HE163741
9	CSI_LAB1	LAB1	CSI104	01:00:00	10	7.2	HE163740
10	CSI_LAB1	LAB1	CSI104	01:00:00	10	7.2	HE163742
11	CSI_LAB2	LAB2	CSI104	01:00:00	10	7.2	HE163742
12	CSI_LAB2	LAB2	CSI104	01:00:00	10	7.2	HE163740
13	CSI_LAB2	LAB2	CSI104	01:00:00	10	8.2	HE163741
14	CSI_PRE1	PRE1	CSI104	00:10:00	5	8.2	HE163741
15	CSI_PRE1	PRE1	CSI104	00:10:00	5	7.2	HE163740
16	CSI_PRE1	PRE1	CSI104	00:10:00	5	8.9	HE163745
17	CSI_PRE1	PRE1	CSI104	00:10:00	5	5.9	HE163746
18	CSI_PRE1	PRE1	CSI104	00:10:00	5	8.8	HE163747

Query executed successfully. DESKTOP-VK004KH\SQLEXPRESS ... DESKTOP-VK004KH\pc (53) Grading Management 00:00:00 146 rows

• A query that uses aggregate functions32

[Type here]



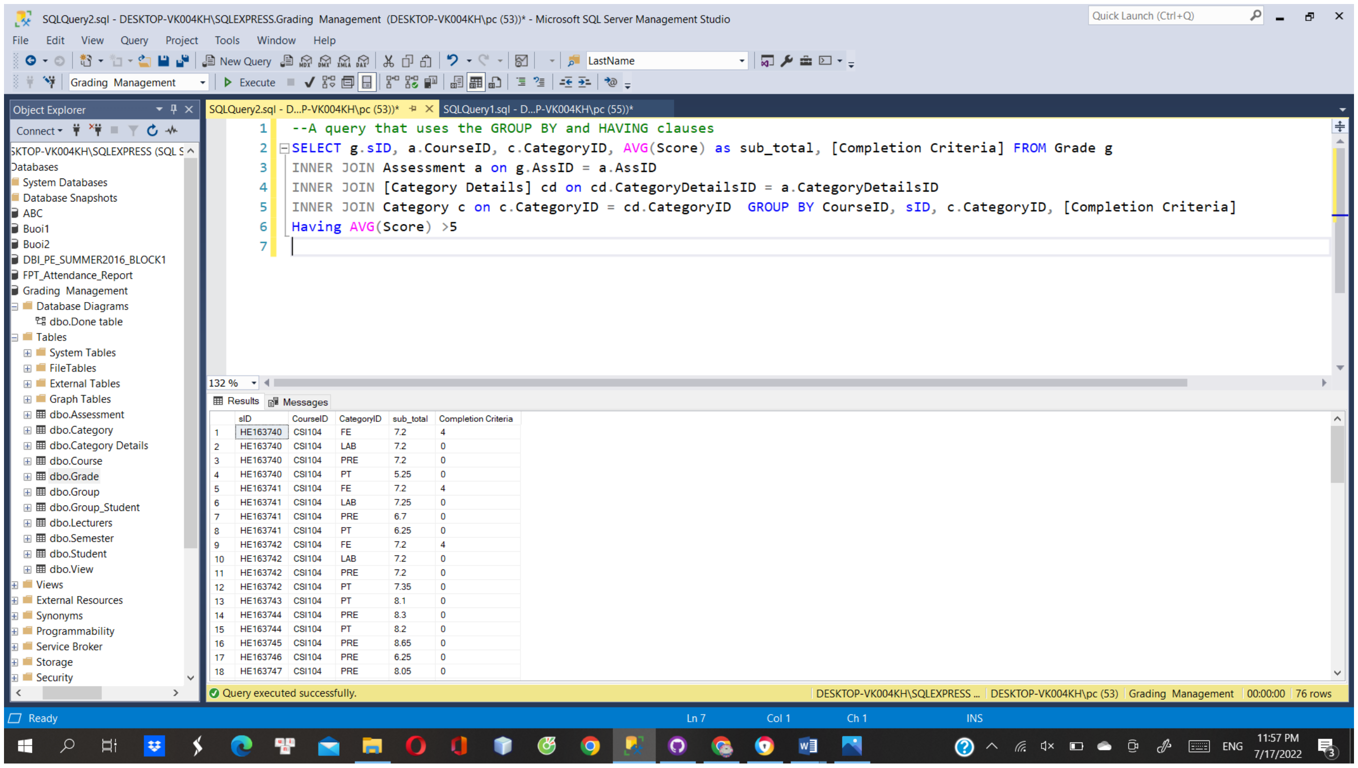
```
--• A query that uses aggregate functions
Select Max([Average]) as [Highest Mark] From [View]
```

	Highest Mark
1	7.78

Query executed successfully.

[Type here]

- A query that uses the GROUP BY and HAVING clauses



SQLQuery2.sql - DESKTOP-VK004KH\SQLEXPRESS.Grading Management (DESKTOP-VK004KH\pc (53))* - Microsoft SQL Server Management Studio

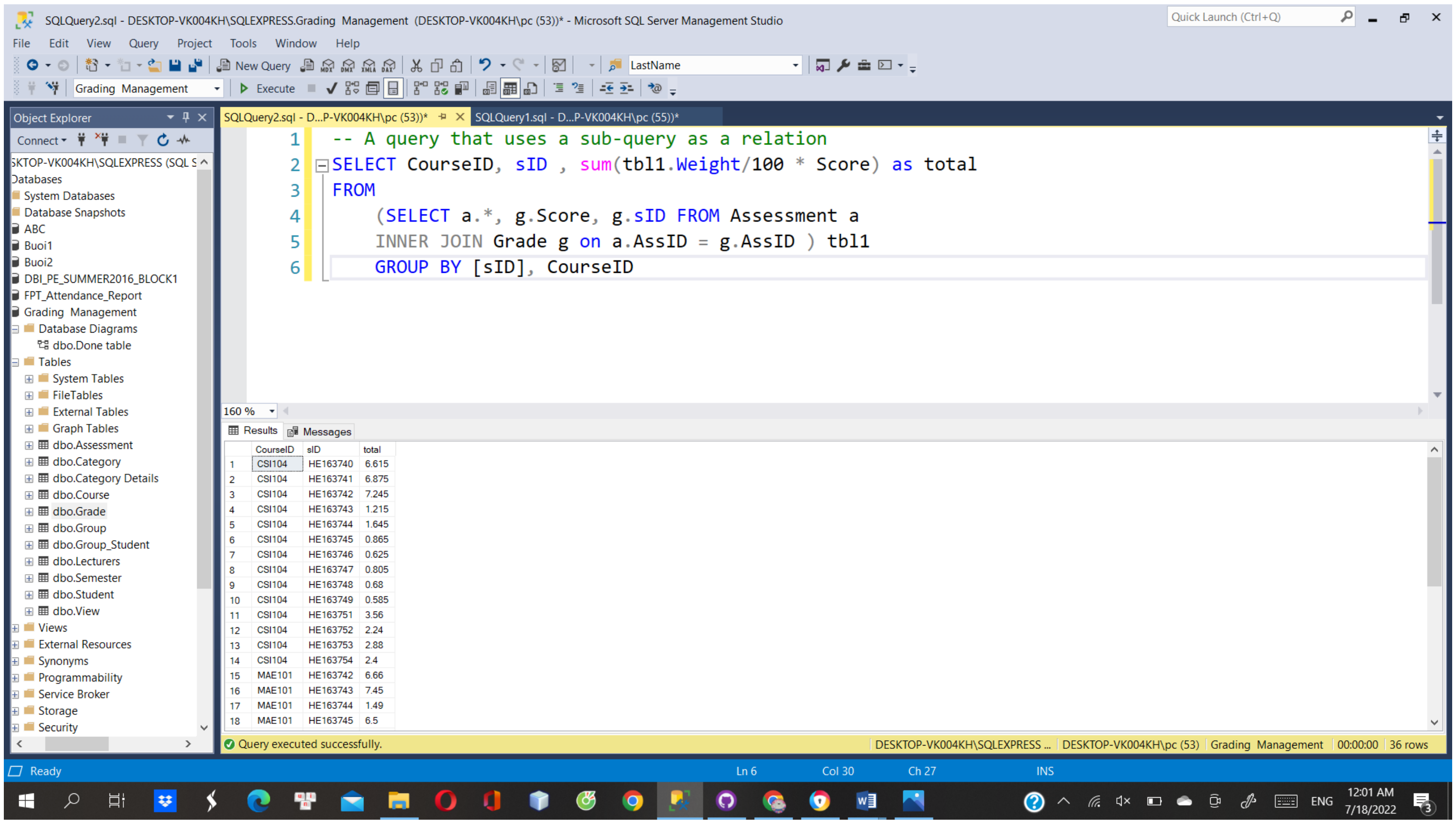
```sql
--A query that uses the GROUP BY and HAVING clauses
SELECT g.sID, a.CourseID, c.CategoryID, AVG(Score) as sub_total, [Completion Criteria] FROM Grade g
INNER JOIN Assessment a on g.AssID = a.AssID
INNER JOIN [Category Details] cd on cd.CategoryDetailsID = a.CategoryDetailsID
INNER JOIN Category c on c.CategoryID = cd.CategoryID  GROUP BY CourseID, sID, c.CategoryID, [Completion Criteria]
Having AVG(Score) >5
```

	sID	CourseID	CategoryID	sub_total	Completion Criteria
1	HE163740	CSI104	FE	7.2	4
2	HE163740	CSI104	LAB	7.2	0
3	HE163740	CSI104	PRE	7.2	0
4	HE163740	CSI104	PT	5.25	0
5	HE163741	CSI104	FE	7.2	4
6	HE163741	CSI104	LAB	7.25	0
7	HE163741	CSI104	PRE	6.7	0
8	HE163741	CSI104	PT	6.25	0
9	HE163742	CSI104	FE	7.2	4
10	HE163742	CSI104	LAB	7.2	0
11	HE163742	CSI104	PRE	7.2	0
12	HE163742	CSI104	PT	7.35	0
13	HE163743	CSI104	PT	8.1	0
14	HE163744	CSI104	PRE	8.3	0
15	HE163744	CSI104	PT	8.2	0
16	HE163745	CSI104	PRE	8.65	0
17	HE163746	CSI104	PRE	6.25	0
18	HE163747	CSI104	PRE	8.05	0

Query executed successfully. | DESKTOP-VK004KH\SQLEXPRESS ... | DESKTOP-VK004KH\pc (53) | Grading Management | 00:00:00 | 76 rows

[Type here]

• A query that uses a sub-query as a relation35



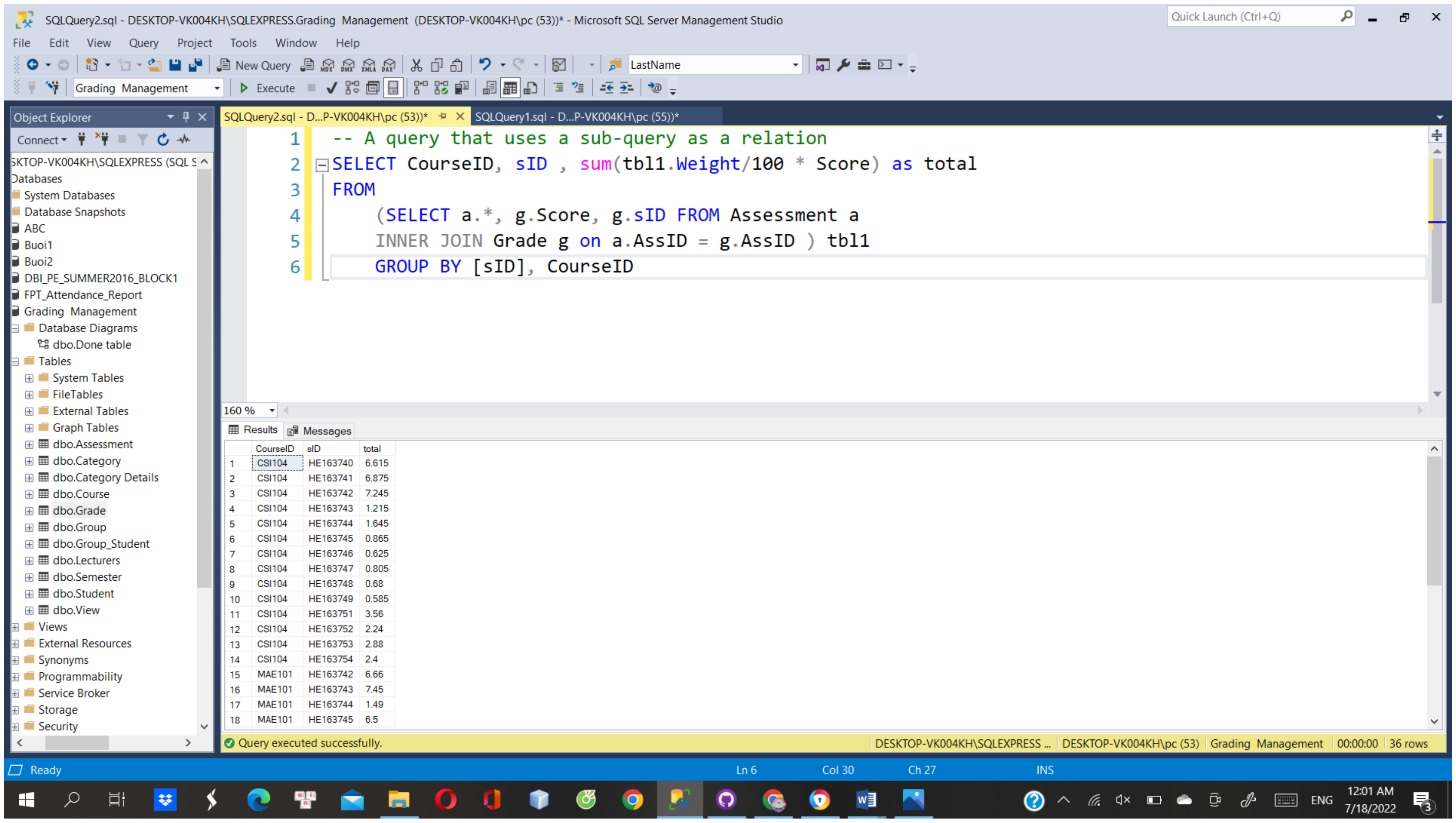
SQLQuery2.sql - DESKTOP-VK004KH\SQLEXPRESS.Grading Management (DESKTOP-VK004KH\pc (53))* - Microsoft SQL Server Management Studio

```
-- A query that uses a sub-query as a relation
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
```

	CourseID	sID	total
1	CSI104	HE163740	6.615
2	CSI104	HE163741	6.875
3	CSI104	HE163742	7.245
4	CSI104	HE163743	1.215
5	CSI104	HE163744	1.645
6	CSI104	HE163745	0.865
7	CSI104	HE163746	0.625
8	CSI104	HE163747	0.805
9	CSI104	HE163748	0.68
10	CSI104	HE163749	0.585
11	CSI104	HE163751	3.56
12	CSI104	HE163752	2.24
13	CSI104	HE163753	2.88
14	CSI104	HE163754	2.4
15	MAE101	HE163742	6.66
16	MAE101	HE163743	7.45
17	MAE101	HE163744	1.49
18	MAE101	HE163745	6.5

Query executed successfully. DESKTOP-VK004KH\SQLEXPRESS ... DESKTOP-VK004KH\pc (53) Grading Management 00:00:00 36 rows

• A query that uses a sub-query as a relation36



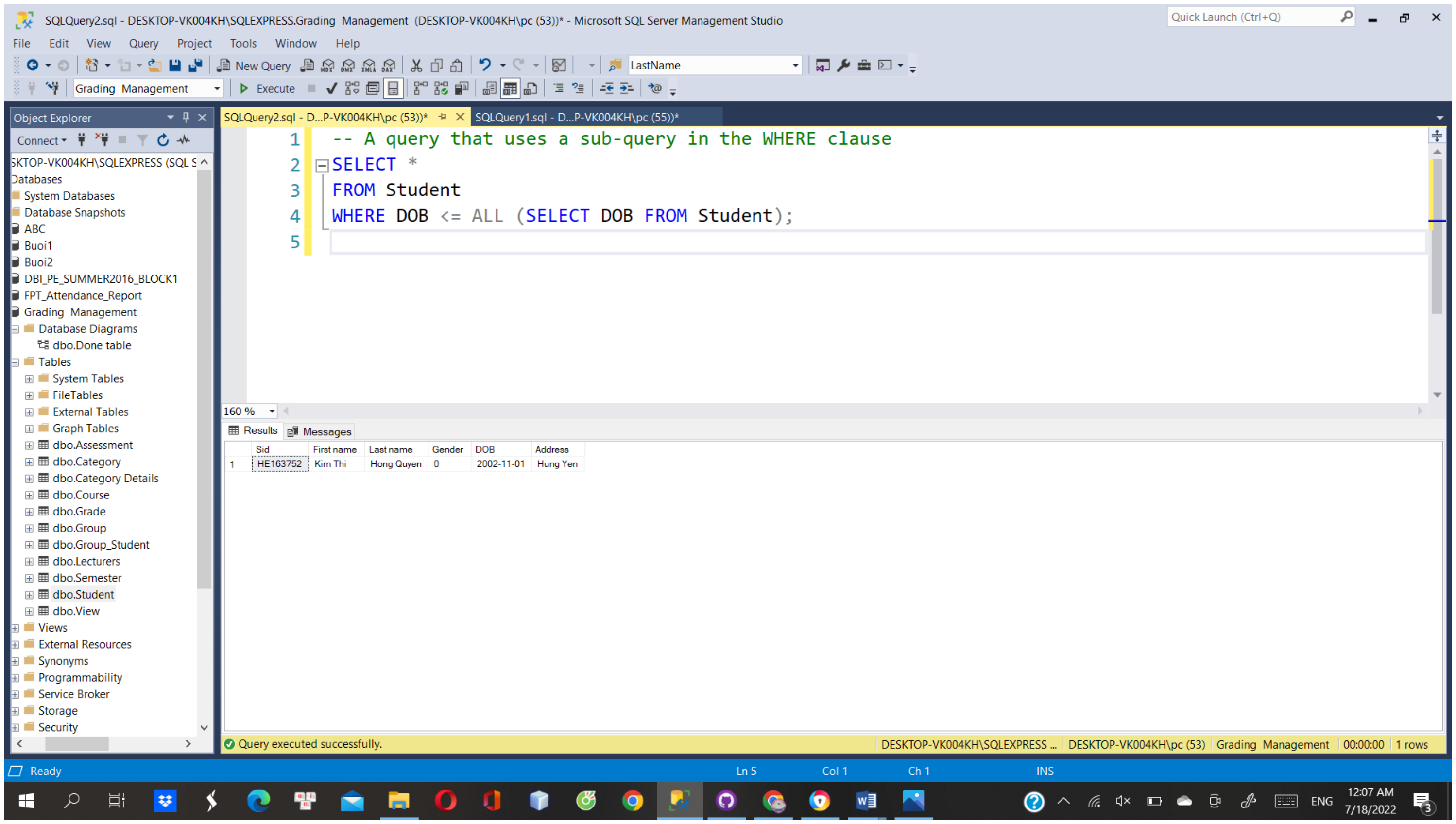
```sql
-- A query that uses a sub-query as a relation
SELECT CourseID, sID , sum(tbl1.Weight/100 * Score) as total
FROM
    (SELECT a.*, g.Score, g.sID FROM Assessment a
    INNER JOIN Grade g on a.AssID = g.AssID ) tbl1
    GROUP BY [sID], CourseID
```

	CourseID	sID	total
1	CSI104	HE163740	6.615
2	CSI104	HE163741	6.875
3	CSI104	HE163742	7.245
4	CSI104	HE163743	1.215
5	CSI104	HE163744	1.645
6	CSI104	HE163745	0.865
7	CSI104	HE163746	0.625
8	CSI104	HE163747	0.805
9	CSI104	HE163748	0.68
10	CSI104	HE163749	0.585
11	CSI104	HE163751	3.56
12	CSI104	HE163752	2.24
13	CSI104	HE163753	2.88
14	CSI104	HE163754	2.4
15	MAE101	HE163742	6.66
16	MAE101	HE163743	7.45
17	MAE101	HE163744	1.49
18	MAE101	HE163745	6.5

Query executed successfully. DESKTOP-VK004KH\SQLEXPRESS ... DESKTOP-VK004KH\pc (53) Grading Management 00:00:00 36 rows

[Type here]

o A query that uses a sub-query in the WHERE clause



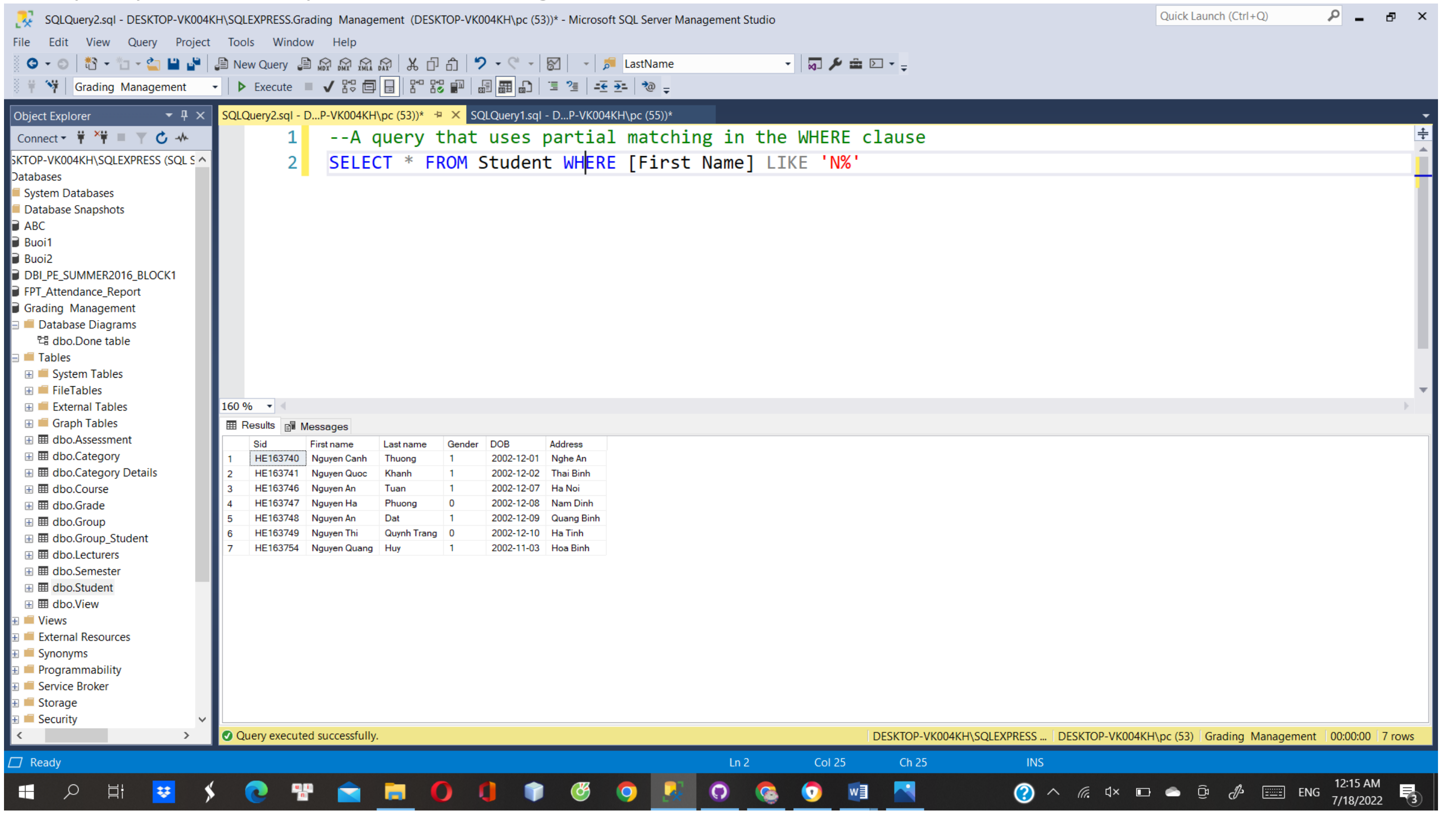
```
-- A query that uses a sub-query in the WHERE clause
SELECT *
FROM Student
WHERE DOB <= ALL (SELECT DOB FROM Student);
```

```
   Sid        First name  Last name   Gender  DOB         Address
1  HE163752   Kim Thi     Hong Quyen  0       2002-11-01  Hung Yen
```

Query executed successfully.

[Type here]

- A query that uses partial matching in the WHERE clause38



```sql
--A query that uses partial matching in the WHERE clause
SELECT * FROM Student WHERE [First Name] LIKE 'N%'
```

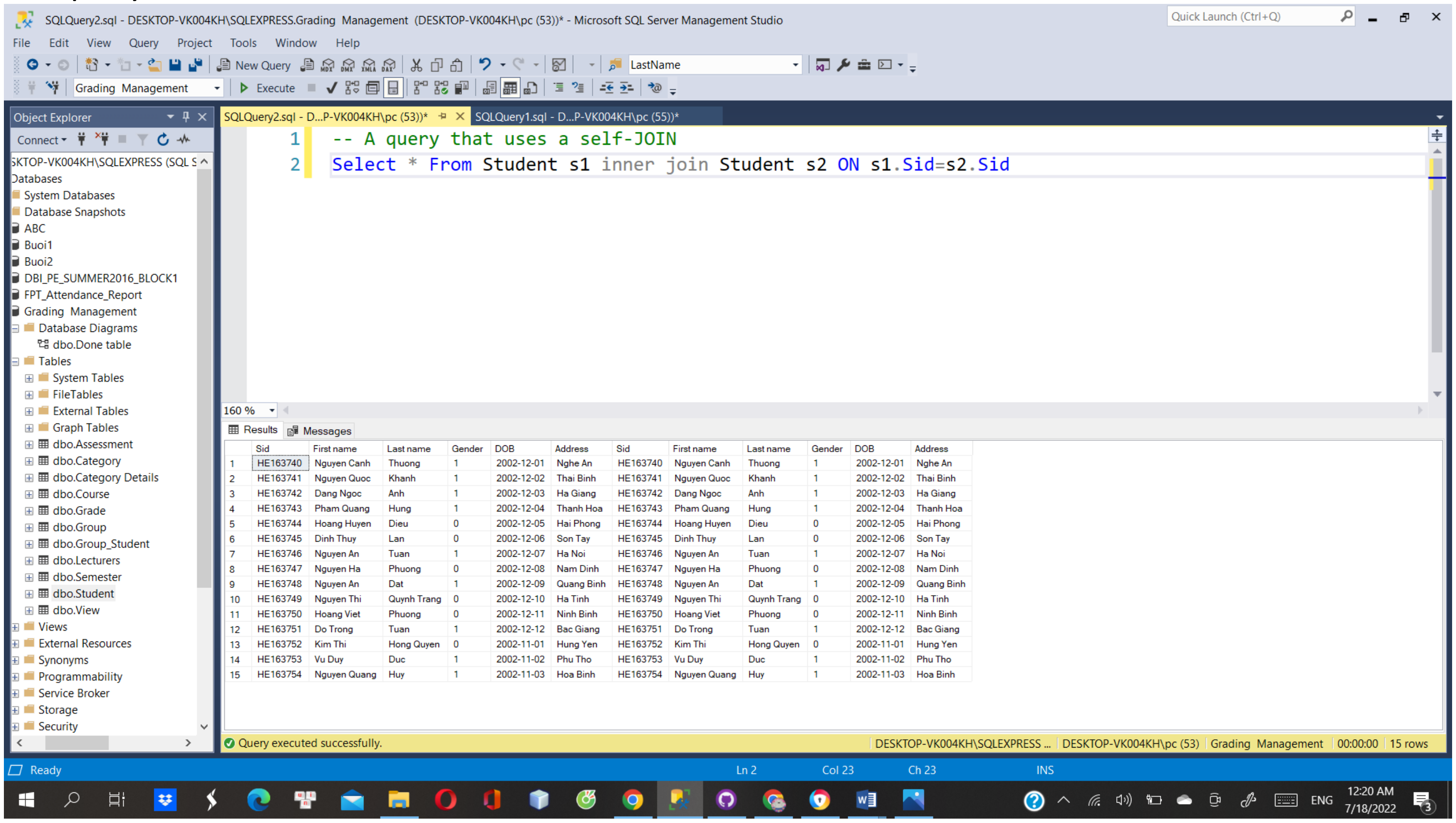
	Sid	First name	Last name	Gender	DOB	Address
1	HE163740	Nguyen Canh	Thuong	1	2002-12-01	Nghe An
2	HE163741	Nguyen Quoc	Khanh	1	2002-12-02	Thai Binh
3	HE163746	Nguyen An	Tuan	1	2002-12-07	Ha Noi
4	HE163747	Nguyen Ha	Phuong	0	2002-12-08	Nam Dinh
5	HE163748	Nguyen An	Dat	1	2002-12-09	Quang Binh
6	HE163749	Nguyen Thi	Quynh Trang	0	2002-12-10	Ha Tinh
7	HE163754	Nguyen Quang	Huy	1	2002-11-03	Hoa Binh

Query executed successfully.

[Type here]

s

• A query that uses a self-JOIN



```sql
-- A query that uses a self-JOIN
Select * From Student s1 inner join Student s2 ON s1.Sid=s2.Sid
```

	Sid	First name	Last name	Gender	DOB	Address	Sid	First name	Last name	Gender	DOB	Address
1	HE163740	Nguyen Canh	Thuong	1	2002-12-01	Nghe An	HE163740	Nguyen Canh	Thuong	1	2002-12-01	Nghe An
2	HE163741	Nguyen Quoc	Khanh	1	2002-12-02	Thai Binh	HE163741	Nguyen Quoc	Khanh	1	2002-12-02	Thai Binh
3	HE163742	Dang Ngoc	Anh	1	2002-12-03	Ha Giang	HE163742	Dang Ngoc	Anh	1	2002-12-03	Ha Giang
4	HE163743	Pham Quang	Hung	1	2002-12-04	Thanh Hoa	HE163743	Pham Quang	Hung	1	2002-12-04	Thanh Hoa
5	HE163744	Hoang Huyen	Dieu	0	2002-12-05	Hai Phong	HE163744	Hoang Huyen	Dieu	0	2002-12-05	Hai Phong
6	HE163745	Dinh Thuy	Lan	0	2002-12-06	Son Tay	HE163745	Dinh Thuy	Lan	0	2002-12-06	Son Tay
7	HE163746	Nguyen An	Tuan	1	2002-12-07	Ha Noi	HE163746	Nguyen An	Tuan	1	2002-12-07	Ha Noi
8	HE163747	Nguyen Ha	Phuong	0	2002-12-08	Nam Dinh	HE163747	Nguyen Ha	Phuong	0	2002-12-08	Nam Dinh
9	HE163748	Nguyen An	Dat	1	2002-12-09	Quang Binh	HE163748	Nguyen An	Dat	1	2002-12-09	Quang Binh
10	HE163749	Nguyen Thi	Quynh Trang	0	2002-12-10	Ha Tinh	HE163749	Nguyen Thi	Quynh Trang	0	2002-12-10	Ha Tinh
11	HE163750	Hoang Viet	Phuong	0	2002-12-11	Ninh Binh	HE163750	Hoang Viet	Phuong	0	2002-12-11	Ninh Binh
12	HE163751	Do Trong	Tuan	1	2002-12-12	Bac Giang	HE163751	Do Trong	Tuan	1	2002-12-12	Bac Giang
13	HE163752	Kim Thi	Hong Quyen	0	2002-11-01	Hung Yen	HE163752	Kim Thi	Hong Quyen	0	2002-11-01	Hung Yen
14	HE163753	Vu Duy	Duc	1	2002-11-02	Phu Tho	HE163753	Vu Duy	Duc	1	2002-11-02	Phu Tho
15	HE163754	Nguyen Quang	Huy	1	2002-11-03	Hoa Binh	HE163754	Nguyen Quang	Huy	1	2002-11-03	Hoa Binh

Query executed successfully. | DESKTOP-VK004KH\SQLEXPRESS ... | DESKTOP-VK004KH\pc (53) | Grading Management | 00:00:00 | 15 rows

[Type here]

The Trigger , store procedure, and the index should be added (explain why you make it)

```
TRIGGER
--I created that trigger to warn about adding, correcting, deleting, w
Create 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 = @sI
             ) 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 =
BEGIN
     PRINT 'Incorrect Status'
     ROLLBACK TRAN
END
```

[Type here]

```
UPDATE [View] SET Average = 6.5, [Status] = 'PASSED', Semester ='Fall21' WHERE sID =

SELECT * FROM [View]
--Procedure : Average of component scores
Create PROC select_sub_total
AS
BEGIN
	SELECT g.sID, a.CourseID, c.CategoryID, AVG(Score) as sub_total, [Completion Criter
	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,
END
GO

EXEC select_sub_total

  -  Index :Helps to find information faster
CREATE INDEX Stu_Name ON Student([Last Name], [First Name])
CREATE INDEX Lec_Name ON Lecturers([Last Name], [First Name])
SELECT * FROM Student WHERE [Last Name] = N'Thuong' AND [First Name] =
```