

STUDENT GRADING MANAGEMENT SUBSYSTEM – FLM

Giới thiệu:

1. Thông tin cá nhân

- **Họ và tên:** Phạm Hồng Đăng
- **Mssv:** HE163868
- **Lớp:** IA1604
- **Chuyên ngành:** An toàn thông tin
- **Trường:** đại học FPT cơ sở Hòa Lạc
- **Email:** dangphhe163868@fpt.edu.vn
- **Sdt:** 0985402002

2. Báo cáo

- **Tên báo cáo:** Database lưu trữ dữ liệu hệ thống tính điểm cho sinh viên trường đại học FPT
- **Yêu cầu:** theo yêu cầu của giảng viên giảng dạy môn DBI202 lớp IA1604 thì bản báo cáo này được tạo nên như 1 bài Progress Test có lấy điểm
- **Nội dung:** đưa ra các phân tích về đề bài, hình ảnh cũng như các thông tin có thể khai thác từ các hình ảnh, tạo nên 1 database cơ bản cho 1 hệ thống tính điểm

Phân tích:

Bảng thông tin các điểm thành phần

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 Implementation Progress test Instructor has
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 F
Labs	on-going	5	15.0%	>0	in lab session		practical exercises		related to studied modules	Guided by instructor	may be contin
Practical Exam	practical exam	1	25.0%	>0	85'		Preferable to be marked by Scripts		DB programming skills	by exam board and department	Practical Exam
Final Exam	final exam	1	30.0%	5	60'		Multiple choices Marked by Computer	60	Knowledge and skills in the course, but with much focus on the items in Chapters 2 to 6, >= 70% new questions (for the current semester);	by exam board	

- Bảng này chứa các nội dung là các loại bài test, đầu điểm mà một sinh viên phải đáp ứng của 1 môn học.
- Bảng bao gồm:
 - **Category (Hạng mục):** danh sách, tên các đầu điểm
 - Progress tests
 - Assignment
 - Labs
 - Practical Exam (PE)
 - Final Exam (FE)
 - **Type (loại):** trạng thái và loại hình kiểm tra
 - Quiz
 - On-going
 - Practical Exam (PE)
 - Final Exam (FE)
 - **Part:** Số lượng các đầu điểm trên từng hạng mục
 - **Weight:** Trọng số đầu điểm trên từng hạng mục
 - **Completion Criteria:** Điều kiện để thi và điểm tối thiểu để pass
 - **Duration:** Khoảng thời gian cần làm việc trong hạng mục
 - At home
 - In lab session
 - **Question Type (loại câu hỏi)**
 - Multichoices
 - Practices
 - Design and Present
 - Scripts
 - **No Question (số câu hỏi)**
 - **Knowledge and Skill**
 - Kỹ năng thực hành
 - Kiến thức tập trung, quan trọng
 - **Grading Guide**
 - **Note:** Tiêu chí, tỉ trọng và hướng dẫn cho điểm trên các hạng mục

Bảng môn và Khóa học

NO.	SUBJECT CODE	SUBJECT NAME	SEMESTER	GROUP	STARTDATE	ENDDATE	AVERAGE MARK	STATUS
1	SSL101c	Academic Skills for University Success	Spring2021					Not Passed
2	SSG103	Communication and In-Group Working Skills	Summer2021					Passed
3	NWC203c	Computer Networking	Summer2021					Passed
4	CEA201	Computer Organization and Architecture	Spring2021					Passed
5	MAD101	Discrete mathematics	Summer2021					Passed
6	JPD113	Elementary Japanese 1-A1.1	Fall2021					Passed
7	CSI104	Introduction to Computer Science	Spring2021					Passed
8	DBI202	Introduction to Databases	Fall2021					Not Passed
9	LUK1	Level 1	Fall2019					Passed
10	LUK2	Level 2	Spring2020					Passed
11	LUK3	Level 3	Spring2020					Passed
12	LUK4	Level 4	Summer2020					Pass (with conditions)
13	LUK5	Level 5	Summer2020					Passed
14	LUK6	Level 6	Fall2020					Passed
15	MAE101	Mathematics for Engineering	Spring2021					Passed
16	GDQP	Military training	Fall2019					Passed
17	PRO192	Object-Oriented Programming	Fall2021					Passed
18	PRO192	Object-Oriented Programming	Fall2021					Not Passed
19	OSG202	Operating Systems	Summer2021					Passed
20	PRF192	Programming Fundamentals	Summer2021					Not Passed
21	PRF192	Programming Fundamentals	Spring2021					Attendance Fail
22	DTB102	Traditional musical instrument	Summer2020					Passed
23	VOV114	Vovinam 1	Fall2019					Passed
24	VOV124	Vovinam 2	Summer2020					Passed
25	VOV134	Vovinam 3	Summer2020					Passed

- Ở bảng này, sinh viên có thể xem được thông tin cơ bản của toàn bộ các môn học mà mình tham gia
- Bảng bao gồm:
 - No: Số lượng môn học, khóa học
 - Subject code: Mã code tương ứng trên từng khóa học
 - Subject name: Định nghĩa và khái quát chung về môn học
 - Semester: Thời gian và kì học
 - Seasons
 - Years
 - Group: Tên lớp học theo khóa và kì học
 - Start Date: Thời điểm bắt đầu môn học
 - End Date: Thời điểm kết thúc môn học
 - Average Mark: Điểm trung bình
 - Status: Trạng thái nộp bài của sinh viên

- Not passed
- Passed
- Passed with conditions

Bảng điểm của 1 sinh viên trong 1 môn học

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

- Ở bảng này, thấy được các đầu điểm, điểm trung bình và trạng thái của 1 môn học mà 1 sinh viên sau khi học môn đó đã đạt được
- Bảng bao gồm:
 - Grade category (Hạng mục): Danh sách, tên các đầu điểm
 - Progress Tests (Quiz)
 - Assignment (Person or Group)
 - Labs
 - Practical Exam (PE)
 - Final Exam (FE)
 - Final Exam Resit (2ndFE)
 - Grade Item (Hạng mục): Tên và tổng đầu điểm
 - Item
 - Total
 - Weight: Tỷ trọng điểm thành phần (%)
 - Value: Điểm sinh viên đạt được trên thang điểm quy chuẩn
 - Comment: Một số lời nhắc hoặc chú ý sẽ được note vào cột này

Xác định các thực thể và thông tin các thuộc tính

1. Thực thể: Student

- StudentID
- Email
- First Name
- Last Name
- Gender
- Date of Birth (DoB)

2. Thực thể: Group

- GroupID
- Group Name
- Number of Student

3. Thực thể: Lecture

- LectureID
- Email
- First Name
- Last Name
- Gender
- Date of Birth
- Report

4. Thực thể: Course

- CourseID
- Course Name

5. Thực thể: Assignment

- AssignID
- Assign Name
- LectureID

6. Thực thể: Class

- ClassID
- AssignID
- Semester
- Start Date
- End Date

7. Thực thể: Assessment System

- AssID
- CatelD
- CourseID
- Number of Quest
- Duration
- weight

8. Thực thể: Categories

- CatelD
- Name
- Type
- Completion Criteria

Phân chia các Thực thể và Relationships

1. Thực thể Student <-> Group

- Các Student sẽ tham gia vào các Group khác nhau khi dựa trên kì học, môn học và chuyên ngành, nên các nhóm sẽ chứa nhiều sinh viên cùng địa điểm
- Xác định quan hệ giữa thực thể Student và thực thể Group là mối quan hệ nhiều - nhiều (n-n)

2. Thực thể Group và Class

- Các Group sẽ được phân chia vào các Class với giảng viên và môn học tương ứng với kì học hiện tại của Group sinh viên đó, nên một Group có thể tham gia nhiều Class và 1 Class cũng có thể có nhiều Group tham gia
- Xác định quan hệ giữa thực thể Group và thực thể Class là mối quan hệ nhiều -nhiều (n-n)

3. Thực thể Course và Categories

- Một Course thì được đánh giá qua nhiều loại đầu điểm khác nhau và một loại đầu điểm cũng dùng để đánh giá nhiều môn học khác nhau
- Xác định quan hệ giữa thực thể Course và Categories là mối quan hệ nhiều - nhiều (n-n)

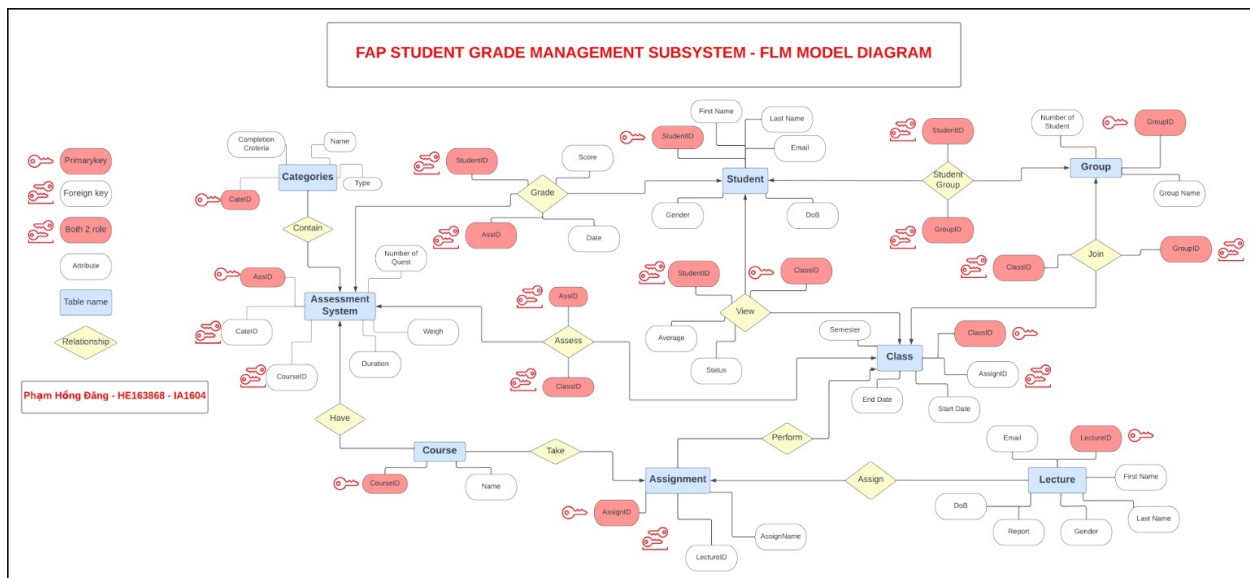
4. Thực thể Lecture và Class

- Một Class thì cũng sẽ chỉ có 1 Lecture tham gia giảng dạy chính (Tại 1 thời điểm) và một giảng viên cũng được phân vào nhiều lớp khác nhau để giảng dạy
- Xác định quan hệ giữa thực thể Lecture và Class là mối quan hệ một - nhiều lecture 1 – n Class

5. Thực thể Lecture và Course

- Một lecture có thể dạy được nhiều Course và một Course cũng có thể được dạy bởi nhiều giảng viên, phụ thuộc vào kì học, 1 giảng viên có thể được dạy 1 môn đó ở kì này nhưng cũng có thể phải dạy môn đó ở kì khác
 - Xác định mối quan hệ giữa thực thể Lecture và Course là mối quan hệ nhiều nhiều
6. Thực thể lecture và Assignment
- Một Assignment có thể có được từ nhiều Lecture và 1 Lecture chỉ có thể đưa ra 1 Assignment
 - Xác định mối quan hệ giữa thực thể Assignment và thực thể Lecture là quan hệ một - nhiều
7. Thực thể Categories và Assignment System
- Một Category có thể tổng hợp từ nhiều Assignment System và 1 Assignment System chỉ có thể đưa vào 1 Category duy nhất
 - Xác định mối quan hệ giữa thực thể Categories và thực thể Assignment System là mối quan hệ một – nhiều
8. Thực thể Course và Assignment
- Một Course chỉ có thể có đúng 1 Assignment và 1 Assignment có thể lấy từ nhiều Course
 - Xác định mối quan hệ giữa thực thể Course và thực thể Assignment là mối quan hệ một – nhiều
9. Thực thể Class và Assignment
- Một Class chỉ nhận đúng 1 Assignment và 1 Assignment có thể được giao cho nhiều các Class khác nhau
 - Xác định mối quan hệ giữa thực thể Class và Assignment là mối quan hệ một – nhiều
10. Thực thể Course và Assesment System
- Một Course chỉ có thể có duy nhất 1 hệ thống đánh giá các đầu điểm và 1 Assesment System có thể là hệ thống đánh giá nhiều Course
 - Xác định mối quan hệ giữa thực thể Assesment System và Course là mối quan hệ một – nhiều
11. Thực thể Student và Assesment System
- Một Student có thể có nhiều hệ thống đánh giá các đầu điểm và 1 Assesment System có thể phụ trách đầu điểm của nhiều Student
 - Xác định mối quan hệ giữa thực thể Student và Assesment System là mối quan hệ nhiều nhiều

Entity Relationship Diagram



Chuyển đổi

Phân tách các quan hệ (n - n) và tạo table

1. Student và Group

- Quan hệ nhiều nhiều thể hiện qua Relationship: Student Group
- Tách và tạo table: Student Group (GroupID, StudentID)
- Table Student <-> Table Student Group <-> Group

2. Class và Group

- Quan hệ nhiều nhiều thể hiện qua Relationship: Join
- Table Class <-> Table Join (ClassID, GroupID) <-> Table Group

3. Class và Assesment System

- Quan hệ nhiều nhiều thể qua Relationship: Assess
- Table Class <-> Table Assess (ClassID, AssID) <-> Assesment System

4. Student và Assesment System

- Quan hệ nhiều nhiều được thể hiện qua Relationship: Grade
- Table Student <-> Table Grade(AssID, StudentID, Score, Date) <-> Assesment System

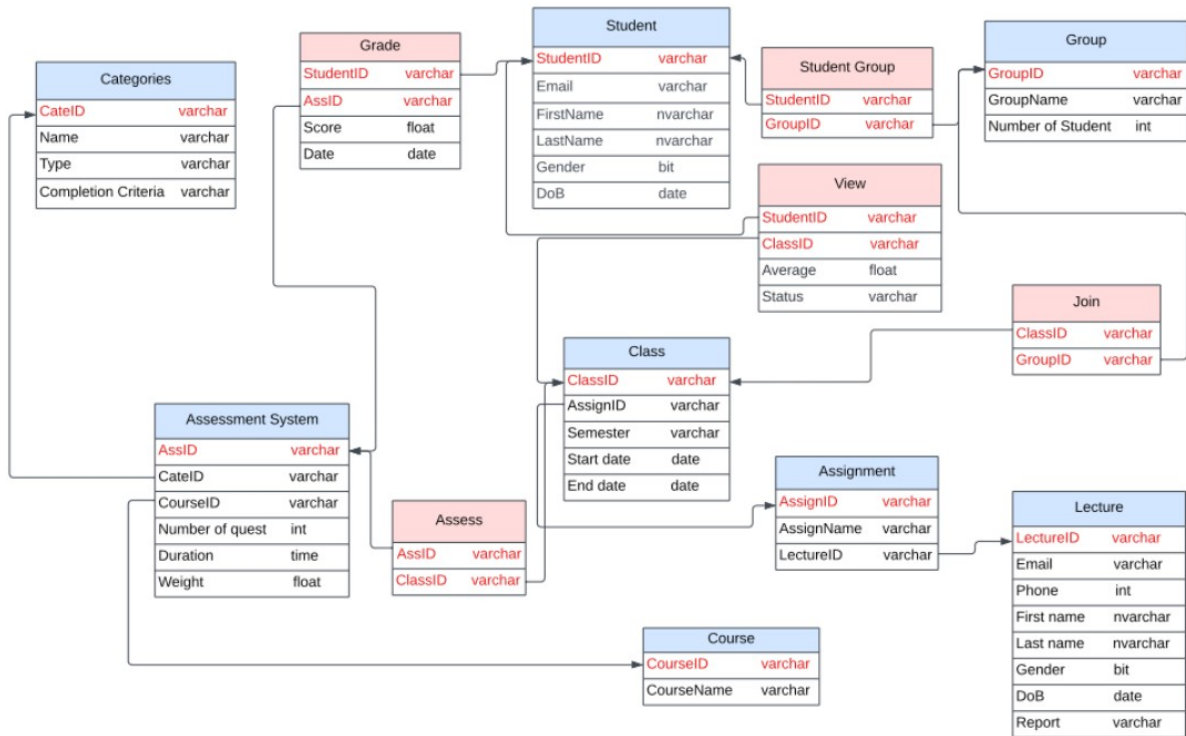
5. Student và Class

- Quan hệ nhiều nhiều được thể hiện qua relationship: View
- Table Student <-> Table View (ClassID, StudentID, Average, Status) <-> Table Class

Module Diagram

FAP STUDENT GRADING MANAGEMENT SUBSYSTEM - ENTITY RELATIONSHIP DIAGRAM

Phạm Hồng Đăng - HE163868 - IA1604 - 7/10/2022



Chuẩn hóa phân tích thuộc tính các Atribute trên từng bảng

1. Student

Student	
StudentID	varchar
Email	varchar
FirstName	nvarchar
LastName	nvarchar
Gender	bit
DoB	date

1. StudentID là primary key của bảng Student
 - Kiểu dữ liệu varchar (50) có dạng AABBBCCC
 - AA là kí hiệu khoa, ngành học
 - BB là khóa học
 - CCCC là số hiệu sinh viên
 - VD: HE163868 – khoa công nghệ thông tin, khóa 16, số hiệu 3868
2. Email
 - Kiểu dữ liệu varchar (50)
 - VD: dangphhe163868@fpt.edu.vn
3. First Name
 - Kiểu dữ liệu nvarchar (50) để có thể viết dấu tiếng việt
4. Last Name
 - Kiểu dữ liệu nvarchar (50) để có thể viết dấu tiếng việt
5. Gender
 - Kiểu dữ liệu Bit, 1 cho nam, 0 cho nữ
6. DoB
 - Kiểu dữ liệu Date
2. Group

Group	
GroupID	varchar
GroupName	varchar
Number of Student	int

- GroupID là primary key của bảng Group
 - Dữ liệu ở dạng varchar (50) VD: IA1604: IA là ngành học Information Assurance, 16 là khóa học, 04 là stt
3. Student Group

Student Group	
StudentID	varchar
GroupID	varchar

- StudentID tương ứng với StudentID của Student

- GroupID tương ứng với GroupID của Group

4. Lecture

Lecture	
LectureID	varchar
Email	varchar
Phone	int
First name	nvarchar
Last name	nvarchar
Gender	bit
DoB	date
Report	varchar

- LectureID là primary key của bảng Lecture
- Dữ liệu có dạng varchar (50) VD: sonnt5 viết tắt tên – họ
- Email có dạng dữ liệu varchar (50) VD: sonnt5@fe.edu.vn
- First Name có dạng dữ liệu nvarchar cho viết tiếng việt
- Last Name có dạng dữ liệu nvarchar cho viết tiếng việt
- Gender Bit 1 cho nam, 0 cho nữ
- DoB có dạng Date
- Report có dạng varchar

5. Course

Course	
CourseID	varchar
CourseName	varchar

- CourseID là primary key của bảng này
- Kiểu dữ liệu varchar (50) VD: DBI202 DBI: tên môn, 202 là mã số
- Course Name kiểu dữ liệu varchar (50)

6. Assignment

Assignment	
AssignID	varchar
AssignName	varchar
LectureID	varchar

- AssignID là primary key của bảng này
- Kiểu dữ liệu varchar (50) VD S5DBI Tên S, số 5, mã môn DBI
- AssignName kiểu dữ liệu varchar (50)
- LectureID tương ứng với LectureID của bảng Lecture

7. Class

Class	
ClassID	varchar
AssignID	varchar
Semester	varchar
Start date	date
End date	date

- ClassID là primary key của bảng này
- Kiểu dữ liệu varchar (50) VD SP22S5DBI – kì học SP22, AssignID là S5DBI
- AssignID tương ứng với AssignID của bảng Assignment
- Semester varchar(50)
- Start date kiểu date
- End date kiểu date

8. Join

Join	
ClassID	varchar
GroupID	varchar

- ClassID tương ứng với ClassID của table Class
- GroupID tương ứng với GroupID của table Group

9. Categories

Categories	
CatID	varchar
Name	varchar
Type	varchar
Completion Criteria	varchar

- CatID là primary key của bảng Categories
- Kiểu dữ liệu varchar (50) VD AS-Assignment
- Type kiểu varchar (50)
- Completion Criteria varchar(50)

10. Assesment System

Assessment System	
AssID	varchar
CatID	varchar
CourseID	varchar
Number of quest	int
Duration	time
Weight	float

- AssID là primary key của bảng Assessment System
- Kiểu dữ liệu varchar (50) Vd DBIAS – môn DBI – đầu điểm là Assignment
- CatID tương ứng với CatID của table Categories
- CourseID tương ứng với CourseID của table Course
- Number of Question kiểu int
- Duration kiểu là varchar (50)
- Weight kiểu là float

11. Grade

Grade	
StudentID	varchar
AssID	varchar
Score	float
Date	date

- StudentID tương ứng với StudentID của bảng Student
- AssID tương ứng với AssID của Assessment System
- Score kiểu float
- Date kiểu date

12. View

View	
StudentID	varchar
ClassID	varchar
Average	float
Status	varchar

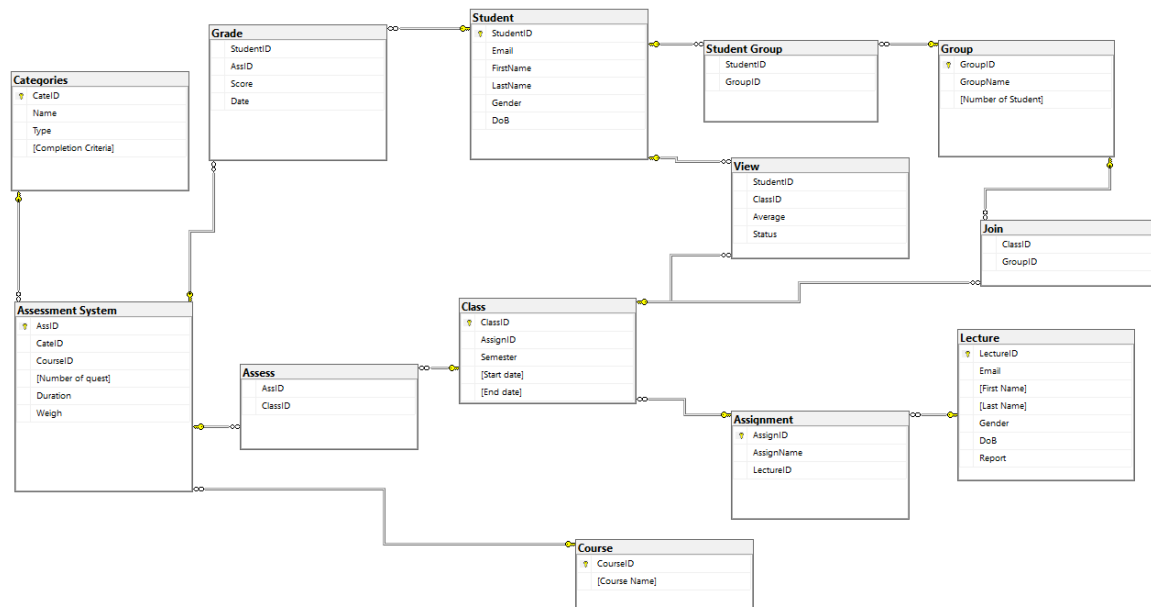
- StudentID tương ứng với StudentID của bảng Student
- ClassID tương ứng với ClassID của bảng Class
- Average kiểu float
- Status kiểu varchar (50)

13. Assess

Assess	
AssID	varchar
ClassID	varchar

- AssID tương ứng với AssID của bảng Assessment System
- ClassID tương ứng với ClassID của bảng Class

Triển khai Database Diagram trên SQL Server



Code tạo bảng

```

USE [Assignment]
GO
/***** Object: Table [dbo].[Assess]    Script Date: 7/15/2022 10:13:31 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Assess](
    [AssID] [varchar](50) NOT NULL,
    [ClassID] [varchar](50) NOT NULL
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Assessment System]    Script Date: 7/15/2022 10:13:31 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Assessment System](
    [AssID] [varchar](50) NOT NULL,
    [CateID] [varchar](50) NOT NULL,
    [CourseID] [varchar](50) NOT NULL,
    [Number of quest] [int] NOT NULL,
    [Duration] [time](7) NOT NULL,
    [Weight] [float] NOT NULL,
    CONSTRAINT [PK_Assessment System] PRIMARY KEY CLUSTERED
(
    [AssID] ASC

```

```

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Assignment]    Script Date: 7/15/2022 10:13:31 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Assignment](
    [AssignID] [varchar](50) NOT NULL,
    [AssignName] [varchar](50) NOT NULL,
    [LectureID] [varchar](50) NOT NULL,
    CONSTRAINT [PK_Assignment] PRIMARY KEY CLUSTERED
(
    [AssignID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Categories]    Script Date: 7/15/2022 10:13:31 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Categories](
    [CateID] [varchar](50) NOT NULL,
    [Name] [varchar](50) NOT NULL,
    [Type] [varchar](50) NOT NULL,
    [Completion Criteria] [varchar](50) NOT NULL,
    CONSTRAINT [PK_Categories] PRIMARY KEY CLUSTERED
(
    [CateID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Class]    Script Date: 7/15/2022 10:13:31 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Class](
    [ClassID] [varchar](50) NOT NULL,
    [AssignID] [varchar](50) NOT NULL,
    [Semester] [varchar](50) NOT NULL,
    [Start date] [date] NOT NULL,
    [End date] [date] NOT NULL,
    CONSTRAINT [PK_Class] PRIMARY KEY CLUSTERED
(
    [ClassID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]

```



```

GO
/***** Object: Table [dbo].[Course]    Script Date: 7/15/2022 10:13:31 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Course](
    [CourseID] [varchar](50) NOT NULL,
    [Course Name] [varchar](50) NULL,
    CONSTRAINT [PK_Course] PRIMARY KEY CLUSTERED
(
    [CourseID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Grade]    Script Date: 7/15/2022 10:13:31 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Grade](
    [StudentID] [varchar](10) NOT NULL,
    [AssID] [varchar](50) NOT NULL,
    [Score] [float] NOT NULL,
    [Date] [date] NOT NULL
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Group]    Script Date: 7/15/2022 10:13:31 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Group](
    [GroupID] [varchar](50) NOT NULL,
    [GroupName] [varchar](50) NOT NULL,
    [Number of Student] [int] NOT NULL,
    CONSTRAINT [PK_Group] PRIMARY KEY CLUSTERED
(
    [GroupID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Join]    Script Date: 7/15/2022 10:13:31 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Join](
    [ClassID] [varchar](50) NOT NULL,
    [GroupID] [varchar](50) NOT NULL
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Lecture]    Script Date: 7/15/2022 10:13:31 AM *****/
SET ANSI_NULLS ON

```

```

GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Lecture](
    [LectureID] [varchar](50) NOT NULL,
    [Email] [varchar](50) NOT NULL,
    [Phone] [int] NOT NULL,
    [First Name] [nvarchar](50) NOT NULL,
    [Last Name] [nvarchar](50) NOT NULL,
    [Gender] [bit] NOT NULL,
    [DoB] [date] NOT NULL,
    [Report] [varchar](50) NOT NULL,
    CONSTRAINT [PK_Lecture] PRIMARY KEY CLUSTERED
(
    [LectureID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Student]    Script Date: 7/15/2022 10:13:31 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Student](
    [StudentID] [varchar](10) NOT NULL,
    [Email] [varchar](50) NOT NULL,
    [FirstName] [nvarchar](50) NOT NULL,
    [LastName] [nvarchar](50) NOT NULL,
    [Gender] [bit] NOT NULL,
    [DoB] [date] NOT NULL,
    CONSTRAINT [PK_Student] PRIMARY KEY CLUSTERED
(
    [StudentID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Student Group]    Script Date: 7/15/2022 10:13:31 AM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Student Group](
    [StudentID] [varchar](10) NOT NULL,
    [GroupID] [varchar](50) NOT NULL
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[View]    Script Date: 7/15/2022 10:13:31 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[View](
    [StudentID] [varchar](10) NOT NULL,

```

```

        [ClassID] [varchar](50) NOT NULL,
        [Average] [float] NOT NULL,
        [Status] [varchar](50) NOT NULL
    ) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Assess] WITH CHECK ADD CONSTRAINT [FK_Assess_Assessment System]
FOREIGN KEY([AssID])
REFERENCES [dbo].[Assessment System] ([AssID])
GO
ALTER TABLE [dbo].[Assess] CHECK CONSTRAINT [FK_Assess_Assessment System]
GO
ALTER TABLE [dbo].[Assess] WITH CHECK ADD CONSTRAINT [FK_Assess_Class] FOREIGN
KEY([ClassID])
REFERENCES [dbo].[Class] ([ClassID])
GO
ALTER TABLE [dbo].[Assess] CHECK CONSTRAINT [FK_Assess_Class]
GO
ALTER TABLE [dbo].[Assessment System] WITH CHECK ADD CONSTRAINT [FK_Assessment
System_Categories] FOREIGN KEY([CateID])
REFERENCES [dbo].[Categories] ([CateID])
GO
ALTER TABLE [dbo].[Assessment System] CHECK CONSTRAINT [FK_Assessment System_Categories]
GO
ALTER TABLE [dbo].[Assessment System] WITH CHECK ADD CONSTRAINT [FK_Assessment
System_Course] FOREIGN KEY([CourseID])
REFERENCES [dbo].[Course] ([CourseID])
GO
ALTER TABLE [dbo].[Assessment System] CHECK CONSTRAINT [FK_Assessment System_Course]
GO
ALTER TABLE [dbo].[Assignment] WITH CHECK ADD CONSTRAINT [FK_Assignment_Lecture]
FOREIGN KEY([LectureID])
REFERENCES [dbo].[Lecture] ([LectureID])
GO
ALTER TABLE [dbo].[Assignment] CHECK CONSTRAINT [FK_Assignment_Lecture]
GO
ALTER TABLE [dbo].[Class] WITH CHECK ADD CONSTRAINT [FK_Class_Assignment] FOREIGN
KEY([AssignID])
REFERENCES [dbo].[Assignment] ([AssignID])
GO
ALTER TABLE [dbo].[Class] CHECK CONSTRAINT [FK_Class_Assignment]
GO
ALTER TABLE [dbo].[Grade] WITH CHECK ADD CONSTRAINT [FK_Grade_Assessment System]
FOREIGN KEY([AssID])
REFERENCES [dbo].[Assessment System] ([AssID])
GO
ALTER TABLE [dbo].[Grade] CHECK CONSTRAINT [FK_Grade_Assessment System]
GO
ALTER TABLE [dbo].[Grade] WITH CHECK ADD CONSTRAINT [FK_Grade_Student] FOREIGN
KEY([StudentID])
REFERENCES [dbo].[Student] ([StudentID])
GO
ALTER TABLE [dbo].[Grade] CHECK CONSTRAINT [FK_Grade_Student]
GO
ALTER TABLE [dbo].[Join] WITH CHECK ADD CONSTRAINT [FK_Join_Class] FOREIGN
KEY([ClassID])
REFERENCES [dbo].[Class] ([ClassID])
GO
ALTER TABLE [dbo].[Join] CHECK CONSTRAINT [FK_Join_Class]

```

```

GO
ALTER TABLE [dbo].[Join] WITH CHECK ADD CONSTRAINT [FK_Join_Group] FOREIGN
KEY([GroupID])
REFERENCES [dbo].[Group] ([GroupID])
GO
ALTER TABLE [dbo].[Join] CHECK CONSTRAINT [FK_Join_Group]
GO
ALTER TABLE [dbo].[Student Group] WITH CHECK ADD CONSTRAINT [FK_Student Group_Group]
FOREIGN KEY([GroupID])
REFERENCES [dbo].[Group] ([GroupID])
GO
ALTER TABLE [dbo].[Student Group] CHECK CONSTRAINT [FK_Student Group_Group]
GO
ALTER TABLE [dbo].[Student Group] WITH CHECK ADD CONSTRAINT [FK_Student Group_Student]
FOREIGN KEY([StudentID])
REFERENCES [dbo].[Student] ([StudentID])
GO
ALTER TABLE [dbo].[Student Group] CHECK CONSTRAINT [FK_Student Group_Student]
GO
ALTER TABLE [dbo].[View] WITH CHECK ADD CONSTRAINT [FK_View_Class] FOREIGN
KEY([ClassID])
REFERENCES [dbo].[Class] ([ClassID])
GO
ALTER TABLE [dbo].[View] CHECK CONSTRAINT [FK_View_Class]
GO
ALTER TABLE [dbo].[View] WITH CHECK ADD CONSTRAINT [FK_View_Student] FOREIGN
KEY([StudentID])
REFERENCES [dbo].[Student] ([StudentID])
GO
ALTER TABLE [dbo].[View] CHECK CONSTRAINT [FK_View_Student]
GO

```

Insert dữ liệu vào các bảng

Student

```

INSERT Student(StudentID, Email, FirstName, LastName, Gender, [DoB]) VALUES ('HE00027',
'ant@fpt.edu.vn', 'A', N'Nguyễn Thị', 0, CAST('02-01-2002' AS DATE));
INSERT Student(StudentID, Email, FirstName, LastName, Gender, DoB) VALUES ('HE00028',
'bnt@fpt.edu.vn', 'B', N'Nguyễn Thị', 0, CAST('02-02-2002' AS DATE));
INSERT Student(StudentID, Email, FirstName, LastName, Gender, DoB) VALUES ('HE00029',
'cnt@fpt.edu.vn', 'C', N'Nguyễn Thị', 0, CAST('02-03-2002' AS DATE));
INSERT Student(StudentID, Email, FirstName, LastName, Gender, DoB) VALUES ('HE00030',
'dnt@fpt.edu.vn', 'D', N'Nguyễn Thị', 0, CAST('02-04-2002' AS DATE));
INSERT Student(StudentID, Email, FirstName, LastName, Gender, DoB) VALUES ('HE00031',
'ent@fpt.edu.vn', 'E', N'Nguyễn Thị', 0, CAST('02-05-2002' AS DATE));
INSERT Student(StudentID, Email, FirstName, LastName, Gender, DoB) VALUES ('HE00032',
'fnt@fpt.edu.vn', 'F', N'Nguyễn Thị', 0, CAST('02-06-2002' AS DATE));
INSERT Student(StudentID, Email, FirstName, LastName, Gender, DoB) VALUES ('HE00033',
'gnt@fpt.edu.vn', 'G', N'Nguyễn Thị', 0, CAST('02-07-2002' AS DATE));
INSERT Student(StudentID, Email, FirstName, LastName, Gender, DoB) VALUES ('HE00034',
'hnt@fpt.edu.vn', 'H', N'Nguyễn Thị', 0, CAST('02-08-2002' AS DATE));
INSERT Student(StudentID, Email, FirstName, LastName, Gender, DoB) VALUES ('HE00035',
'int@fpt.edu.vn', 'I', N'Nguyễn Thị', 0, CAST('02-09-2002' AS DATE));

```

```

INSERT Student(StudentID, Email, FirstName, LastName, Gender, DoB) VALUES ('HE00036',
'knt@fpt.edu.vn', 'K', N'Nguyễn Thị', 0, CAST('02-11-2002' AS DATE));
INSERT Student(StudentID, Email, FirstName, LastName, Gender, DoB) VALUES ('HE00037',
'int@fpt.edu.vn', 'L', N'Nguyễn Thị', 0, CAST('02-12-2002' AS DATE));
INSERT Student(StudentID, Email, FirstName, LastName, Gender, DoB) VALUES ('HE00038',
'mnt@fpt.edu.vn', 'M', N'Nguyễn Thị', 0, CAST('02-13-2002' AS DATE));
INSERT Student(StudentID, Email, FirstName, LastName, Gender, DoB) VALUES ('HE00039',
'nnt@fpt.edu.vn', 'N', N'Nguyễn Thị', 0, CAST('02-14-2002' AS DATE));
INSERT Student(StudentID, Email, FirstName, LastName, Gender, DoB) VALUES ('HE00040',
'ont@fpt.edu.vn', 'O', N'Nguyễn Thị', 0, CAST('02-15-2002' AS DATE));
INSERT Student(StudentID, Email, FirstName, LastName, Gender, DoB) VALUES ('HE00041',
'pnt@fpt.edu.vn', 'P', N'Nguyễn Thị', 0, CAST('02-16-2002' AS DATE));
INSERT Student(StudentID, Email, FirstName, LastName, Gender, DoB) VALUES ('HE00042',
'qnt@fpt.edu.vn', 'Q', N'Nguyễn Thị', 0, CAST('02-17-2002' AS DATE));
INSERT Student(StudentID, Email, FirstName, LastName, Gender, DoB) VALUES ('HE00043',
'rnt@fpt.edu.vn', 'R', N'Nguyễn Thị', 0, CAST('02-18-2002' AS DATE));
INSERT Student(StudentID, Email, FirstName, LastName, Gender, DoB) VALUES ('HE00044',
'snt@fpt.edu.vn', 'S', N'Nguyễn Thị', 0, CAST('02-19-2002' AS DATE));
INSERT Student(StudentID, Email, FirstName, LastName, Gender, DoB) VALUES ('HE00045',
'tnt@fpt.edu.vn', 'T', N'Nguyễn Thị', 0, CAST('02-20-2002' AS DATE));
INSERT Student(StudentID, Email, FirstName, LastName, Gender, DoB) VALUES ('HE00046',
'unt@fpt.edu.vn', 'U', N'Nguyễn Thị', 0, CAST('02-21-2002' AS DATE));
INSERT Student(StudentID, Email, FirstName, LastName, Gender, DoB) VALUES ('HE00047',
'vnt@fpt.edu.vn', 'V', N'Nguyễn Thị', 0, CAST('02-22-2002' AS DATE));
INSERT Student(StudentID, Email, FirstName, LastName, Gender, DoB) VALUES ('HE00048',
'wnt@fpt.edu.vn', 'W', N'Nguyễn Thị', 0, CAST('02-23-2002' AS DATE));
INSERT Student(StudentID, Email, FirstName, LastName, Gender, DoB) VALUES ('HE00049',
'xnt@fpt.edu.vn', 'X', N'Nguyễn Thị', 0, CAST('02-24-2002' AS DATE));
INSERT Student(StudentID, Email, FirstName, LastName, Gender, DoB) VALUES ('HE00050',
'ynt@fpt.edu.vn', 'Y', N'Nguyễn Thị', 0, CAST('02-25-2002' AS DATE));

```

Course

```

INSERT [dbo].[Course] (CourseID, [Course Name]) VALUES (N'CEA201', N'Computer
Organization and Architecture')
INSERT [dbo].[Course] (CourseID, [Course Name]) VALUES (N'CSD201', N'Data Structures and
Algorithms')
INSERT [dbo].[Course] (CourseID, [Course Name]) VALUES (N'CSI104', N'Introduction to
Computer Science')
INSERT [dbo].[Course] (CourseID, [Course Name]) VALUES (N'DBI202', N'Introduction to
Databases')
INSERT [dbo].[Course] (CourseID, [Course Name]) VALUES (N'IA0202', N'Introduction to
Information Assurance')
INSERT [dbo].[Course] (CourseID, [Course Name]) VALUES (N'JPD113', N'Elementary Japanese
1')
INSERT [dbo].[Course] (CourseID, [Course Name]) VALUES (N'LAB211', N'OOP with Java Lab')
INSERT [dbo].[Course] (CourseID, [Course Name]) VALUES (N'MAD101', N'Discrete
mathematics')
INSERT [dbo].[Course] (CourseID, [Course Name]) VALUES (N'MAE101', N'Mathematics for
Engineering')
INSERT [dbo].[Course] (CourseID, [Course Name]) VALUES (N'NWC204', N'Computer
Networking')
INSERT [dbo].[Course] (CourseID, [Course Name]) VALUES (N'OSG202', N'Operating Systems')
INSERT [dbo].[Course] (CourseID, [Course Name]) VALUES (N'PRF192', N'Programming
Fundamentals')

```

```

INSERT [dbo].[Course] (CourseID, [Course Name]) VALUES (N'PRO192', N'Object-Oriented Programming');
INSERT [dbo].[Course] (CourseID, [Course Name]) VALUES (N'SSG104', N'Communication and In-Group Working Skills');
INSERT [dbo].[Course] (CourseID, [Course Name]) VALUES (N'SSL101c', N'Academic Skills for University Success');

```

Group

```

INSERT [Group](GroupID, GroupName,[Number of Student]) VALUES ('IA1604', 'Information Assurance',15);
INSERT [Group](GroupID, GroupName,[Number of Student]) VALUES ('AI1605', 'Artificial Intelligence',15);
INSERT [Group](GroupID, GroupName,[Number of Student]) VALUES ('SE1606', 'Software Engineering',20);
INSERT [Group](GroupID, GroupName,[Number of Student]) VALUES ('IOT1607', 'Internet of Things',15);
INSERT [Group](GroupID, GroupName,[Number of Student]) VALUES ('IA1608', 'Information Assurance',10);
INSERT [Group](GroupID, GroupName,[Number of Student]) VALUES ('SE1609', 'Software Engineering',20);

```

Student Group

```

INSERT [Student Group](StudentID, GroupID) VALUES ('HE00001', 'IA1604');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00002', 'IA1604');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00003', 'IA1604');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00004', 'IA1604');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00005', 'IA1604');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00006', 'IA1604');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00007', 'IA1604');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00008', 'IA1604');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00009', 'IA1604');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00010', 'IA1604');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00011', 'IA1604');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00012', 'IA1604');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00013', 'IA1604');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00014', 'IA1604');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00015', 'IA1604');

INSERT [Student Group](StudentID, GroupID) VALUES ('HE00016', 'AI1605');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00017', 'AI1605');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00018', 'AI1605');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00019', 'AI1605');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00020', 'AI1605');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00021', 'AI1605');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00022', 'AI1605');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00023', 'AI1605');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00024', 'AI1605');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00025', 'AI1605');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00026', 'AI1605');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00027', 'AI1605');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00028', 'AI1605');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00029', 'AI1605');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00030', 'AI1605');

```

```

INSERT [Student Group](StudentID, GroupID) VALUES ('HE00031', 'SE1606');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00032', 'SE1606');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00033', 'SE1606');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00034', 'SE1606');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00035', 'SE1606');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00036', 'SE1606');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00037', 'SE1606');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00038', 'SE1606');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00039', 'SE1606');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00040', 'SE1606');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00041', 'SE1606');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00042', 'SE1606');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00043', 'SE1606');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00044', 'SE1606');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00045', 'SE1606');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00046', 'SE1606');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00047', 'SE1606');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00048', 'SE1606');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00049', 'SE1606');
INSERT [Student Group](StudentID, GroupID) VALUES ('HE00050', 'SE1606');

```

Lecture

```

INSERT Lecture(LectureID, Email, [First Name], [Last Name], Gender, [DoB], Report) VALUES
('GVA', 'gva@fe.edu.vn', 'A', N'Giảng Viên', 1, CAST('01-01-1985' AS date), NULL);
INSERT Lecture(LectureID, Email, [First Name], [Last Name], Gender, [DoB], Report) VALUES
('GVB', 'gvb@fe.edu.vn', 'B', N'Giảng Viên', 0, CAST('01-02-1985' AS date), 'GVA');
INSERT Lecture(LectureID, Email, [First Name], [Last Name], Gender, [DoB], Report) VALUES
('GVC', 'gvc@fe.edu.vn', 'C', N'Giảng Viên', 1, CAST('01-03-1985' AS date), 'GVA');
INSERT Lecture(LectureID, Email, [First Name], [Last Name], Gender, [DoB], Report) VALUES
('GVD', 'gvd@fe.edu.vn', 'D', N'Giảng Viên', 0, CAST('01-04-1985' AS date), 'GVA');
INSERT Lecture(LectureID, Email, [First Name], [Last Name], Gender, [DoB], Report) VALUES
('GVE', 'gve@fe.edu.vn', 'E', N'Giảng Viên', 1, CAST('01-05-1985' AS date), 'GVA');
INSERT Lecture(LectureID, Email, [First Name], [Last Name], Gender, [DoB], Report) VALUES
('GVF', 'gvf@fe.edu.vn', 'F', N'Giảng Viên', 0, CAST('01-06-1985' AS date), NULL);
INSERT Lecture(LectureID, Email, [First Name], [Last Name], Gender, [DoB], Report) VALUES
('GVG', 'gvg@fe.edu.vn', 'G', N'Giảng Viên', 1, CAST('01-07-1985' AS date), 'GVF');
INSERT Lecture(LectureID, Email, [First Name], [Last Name], Gender, [DoB], Report) VALUES
('GVH', 'gvh@fe.edu.vn', 'H', N'Giảng Viên', 0, CAST('01-08-1985' AS date), NULL);
INSERT Lecture(LectureID, Email, [First Name], [Last Name], Gender, [DoB], Report) VALUES
('GVI', 'gvi@fe.edu.vn', 'I', N'Giảng Viên', 1, CAST('01-09-1985' AS date), 'GVH');
INSERT Lecture(LectureID, Email, [First Name], [Last Name], Gender, [DoB], Report) VALUES
('GVJ', 'gvj@fe.edu.vn', 'J', N'Giảng Viên', 0, CAST('01-10-1985' AS date), 'GVH');
INSERT Lecture(LectureID, Email, [First Name], [Last Name], Gender, [DoB], Report) VALUES
('GVK', 'gvk@fe.edu.vn', 'K', N'Giảng Viên', 1, CAST('01-11-1985' AS date), 'GVH');

```

Assignment

```

INSERT Assignment(AssignID, AssignName, LectureID) VALUES ('ACSD', 'CSD201', 'GVA');
INSERT Assignment(AssignID, AssignName, LectureID) VALUES ('BCSD', 'CSD201', 'GVB');
INSERT Assignment(AssignID, AssignName, LectureID) VALUES ('ECSD', 'CSD201', 'GVE');
INSERT Assignment(AssignID, AssignName, LectureID) VALUES ('ADBI', 'DBI202', 'GVA');
INSERT Assignment(AssignID, AssignName, LectureID) VALUES ('DDBI', 'DBI202', 'GVD');
INSERT Assignment(AssignID, AssignName, LectureID) VALUES ('EDBI', 'DBI202', 'GVE');

```



```

INSERT Assignment(AssignID, AssignName, LectureID) VALUES ('BIAO', 'IAO202', 'GVB');
INSERT Assignment(AssignID, AssignName, LectureID) VALUES ('CLAB', 'LAB211', 'GVC');
INSERT Assignment(AssignID, AssignName, LectureID) VALUES ('DLAB', 'LAB211', 'GVD');

INSERT Assignment(AssignID, AssignName, LectureID) VALUES ('FJPD', 'JPD113', 'GVF');
INSERT Assignment(AssignID, AssignName, LectureID) VALUES ('GJPD', 'JPD113', 'GVG');

INSERT Assignment(AssignID, AssignName, LectureID) VALUES ('HCEA', 'CEA201', 'GVH');
INSERT Assignment(AssignID, AssignName, LectureID) VALUES ('JCEA', 'CEA201', 'GVJ');
INSERT Assignment(AssignID, AssignName, LectureID) VALUES ('KCEA', 'CEA201', 'GVK');
INSERT Assignment(AssignID, AssignName, LectureID) VALUES ('KCSI', 'CSI104', 'GVK');
INSERT Assignment(AssignID, AssignName, LectureID) VALUES ('ICSI', 'CSI104', 'GVI');
INSERT Assignment(AssignID, AssignName, LectureID) VALUES ('IMAE', 'MAE101', 'GVI');
INSERT Assignment(AssignID, AssignName, LectureID) VALUES ('JMAE', 'MAE101', 'GVJ');
INSERT Assignment(AssignID, AssignName, LectureID) VALUES ('HPRF', 'PRF192', 'GVH');
INSERT Assignment(AssignID, AssignName, LectureID) VALUES ('ISSL', 'SSL101c', 'GVI');

INSERT Assignment(AssignID, AssignName, LectureID) VALUES ('JMAD', 'MAD101', 'GVJ');
INSERT Assignment(AssignID, AssignName, LectureID) VALUES ('BNWC', 'NWC204', 'GVB');
INSERT Assignment(AssignID, AssignName, LectureID) VALUES ('HOSG', 'OSG202', 'GVH');
INSERT Assignment(AssignID, AssignName, LectureID) VALUES ('FSSG', 'SSG104', 'GVF');
INSERT Assignment(AssignID, AssignName, LectureID) VALUES ('GSSG', 'SSG104', 'GVG');
INSERT Assignment(AssignID, AssignName, LectureID) VALUES ('APRO', 'PRO192', 'GVA');

```

Class (1 phần của Class)

```

INSERT Class(ClassID, AssignID, Semester, [Start date], [End date]) VALUES ('SP22ACSD',
'ACSD', 'SPRING 2022', CAST('01-01-2022' AS date), CAST('04-01-2022' AS date));
INSERT Class(ClassID, AssignID, Semester, [Start date], [End date]) VALUES ('SP22BCSD',
'BCSD', 'SPRING 2022', CAST('01-01-2022' AS date), CAST('04-01-2022' AS date));
INSERT Class(ClassID, AssignID, Semester, [Start date], [End date]) VALUES ('SP22ECSD',
'ECSD', 'SPRING 2022', CAST('01-01-2022' AS date), CAST('04-01-2022' AS date));
INSERT Class(ClassID, AssignID, Semester, [Start date], [End date]) VALUES ('SP22ADBI',
'ADBI', 'SPRING 2022', CAST('01-01-2022' AS date), CAST('04-01-2022' AS date));
INSERT Class(ClassID, AssignID, Semester, [Start date], [End date]) VALUES ('SP22DBI',
'DDBI', 'SPRING 2022', CAST('01-01-2022' AS date), CAST('04-01-2022' AS date));
INSERT Class(ClassID, AssignID, Semester, [Start date], [End date]) VALUES ('SP22BIAO',
'BIAO', 'SPRING 2022', CAST('01-01-2022' AS date), CAST('04-01-2022' AS date));
INSERT Class(ClassID, AssignID, Semester, [Start date], [End date]) VALUES ('SP22CLAB',
'CLAB', 'SPRING 2022', CAST('01-01-2022' AS date), CAST('04-01-2022' AS date));
INSERT Class(ClassID, AssignID, Semester, [Start date], [End date]) VALUES ('SP22DLAB',
'DLAB', 'SPRING 2022', CAST('01-01-2022' AS date), CAST('04-01-2022' AS date));
INSERT Class(ClassID, AssignID, Semester, [Start date], [End date]) VALUES ('SP22FJPD',
'FJPD', 'SPRING 2022', CAST('01-01-2022' AS date), CAST('04-01-2022' AS date));
INSERT Class(ClassID, AssignID, Semester, [Start date], [End date]) VALUES ('SP22GJPD',
'GJPD', 'SPRING 2022', CAST('01-01-2022' AS date), CAST('04-01-2022' AS date));

```

Join (1 phần của Join)

```

INSERT [Join](ClassID, GroupID) VALUES ('SU21KCSI', 'IA1604')
INSERT [Join](ClassID, GroupID) VALUES ('SU21IMAE', 'IA1604')
INSERT [Join](ClassID, GroupID) VALUES ('SU21JCEA', 'IA1604')
INSERT [Join](ClassID, GroupID) VALUES ('SU21ISSL', 'IA1604')
INSERT [Join](ClassID, GroupID) VALUES ('SU21HPRF', 'IA1604')
INSERT [Join](ClassID, GroupID) VALUES ('FA21HOSG', 'IA1604')
INSERT [Join](ClassID, GroupID) VALUES ('FA21BNWC', 'IA1604')

```



```

INSERT [Join](ClassID, GroupID) VALUES ('SU21FJPD', 'IA1604')
INSERT [Join](ClassID, GroupID) VALUES ('FA21APRO', 'IA1604')
INSERT [Join](ClassID, GroupID) VALUES ('FA21JMAD', 'IA1604')
INSERT [Join](ClassID, GroupID) VALUES ('SP22BCSD', 'IA1604')
INSERT [Join](ClassID, GroupID) VALUES ('SP22ADBI', 'IA1604')
INSERT [Join](ClassID, GroupID) VALUES ('SP22FJPD', 'IA1604')
INSERT [Join](ClassID, GroupID) VALUES ('SP22BIAO', 'IA1604')
INSERT [Join](ClassID, GroupID) VALUES ('SP22CLAB', 'IA1604')

```

Categories

```

INSERT Categories(CateID, [Name], [Type], [Completion Criteria]) VALUES ('AS',
'Assignment ', 'On-Going', '>0');
INSERT Categories(CateID, [Name], [Type], [Completion Criteria]) VALUES ('AS1',
'Assignment 1', 'On-Going', '>0');
INSERT Categories(CateID, [Name], [Type], [Completion Criteria]) VALUES ('AS2',
'Assignment 2', 'On-Going', '>0');
INSERT Categories(CateID, [Name], [Type], [Completion Criteria]) VALUES ('LAB', 'Lab',
'On-Going', '>0');
INSERT Categories(CateID, [Name], [Type], [Completion Criteria]) VALUES ('LAB1', 'Lab 1',
'On-Going', '>0');
INSERT Categories(CateID, [Name], [Type], [Completion Criteria]) VALUES ('LAB2', 'Lab 2',
'On-Going', '>0');
INSERT Categories(CateID, [Name], [Type], [Completion Criteria]) VALUES ('LOC', 'Line of
Code', 'On-Going', '>=750');
INSERT Categories(CateID, [Name], [Type], [Completion Criteria]) VALUES ('PT', 'Progress
Test', 'On-Going', '>0');
INSERT Categories(CateID, [Name], [Type], [Completion Criteria]) VALUES ('PT1', 'Progress
Test 1', 'On-Going', '>0');
INSERT Categories(CateID, [Name], [Type], [Completion Criteria]) VALUES ('PT2', 'Progress
Test 2', 'On-Going', '>0');
INSERT Categories(CateID, [Name], [Type], [Completion Criteria]) VALUES ('PE', 'Practical
Exam', 'On-Going', '>0');
INSERT Categories(CateID, [Name], [Type], [Completion Criteria]) VALUES ('FE', 'Final
Exam', 'Final Exam', '>4');
INSERT Categories(CateID, [Name], [Type], [Completion Criteria]) VALUES ('FER', 'Final
Exam Retake', 'Final Exam', '>4');

```

Assessment System (1 phần của Assessment System)

```

INSERT [Assessment System](AssID, CateID, CourseID, [Number of quest], Duration, [Weigh])
VALUES ('CEAAS1', 'AS1', 'CEA201', 20, '30 mins', 0.1);
INSERT [Assessment System](AssID, CateID, CourseID, [Number of quest], Duration, [Weigh])
VALUES ('CEAAS2', 'AS2', 'CEA201', 20, '30 mins', 0.1);
INSERT [Assessment System](AssID, CateID, CourseID, [Number of quest], Duration, [Weigh])
VALUES ('CEAPT1', 'PT1', 'CEA201', 30, '60 mins', 0.2);
INSERT [Assessment System](AssID, CateID, CourseID, [Number of quest], Duration, [Weigh])
VALUES ('CEAPT2', 'PT2', 'CEA201', 30, '60 mins', 0.2);
INSERT [Assessment System](AssID, CateID, CourseID, [Number of quest], Duration, [Weigh])
VALUES ('CEAFE', 'FE', 'CEA201', 50, '90 mins', 0.4);

```

```

INSERT [Assessment System](AssID, CateID, CourseID, [Number of quest], Duration, [Weigh])
VALUES ('CSDLAB1', 'LAB1', 'CSD201', 10, '30 mins', 0.05);
INSERT [Assessment System](AssID, CateID, CourseID, [Number of quest], Duration, [Weigh])
VALUES ('CSDLAB2', 'LAB2', 'CSD201', 10, '30 mins', 0.05);
INSERT [Assessment System](AssID, CateID, CourseID, [Number of quest], Duration, [Weigh])
VALUES ('CSDAS', 'AS', 'CSD201', 25, '30 mins', 0.1);
INSERT [Assessment System](AssID, CateID, CourseID, [Number of quest], Duration, [Weigh])
VALUES ('CSDPT', 'PT', 'CSD201', 35, '60 mins', 0.1);
INSERT [Assessment System](AssID, CateID, CourseID, [Number of quest], Duration, [Weigh])
VALUES ('CSDPE', 'PE', 'CSD201', 15, '90 mins', 0.3);
INSERT [Assessment System](AssID, CateID, CourseID, [Number of quest], Duration, [Weigh])
VALUES ('CSDFE', 'FE', 'CSD201', 50, '90 mins', 0.4);

```

Assess (1 phần của Assess)

```

INSERT Assess(AssID, ClassID) VALUES ('PROAS', 'FA21APRO')
INSERT Assess(AssID, ClassID) VALUES ('PROLAB1', 'FA21APRO')
INSERT Assess(AssID, ClassID) VALUES ('PROLAB2', 'FA21APRO')
INSERT Assess(AssID, ClassID) VALUES ('PROPT', 'FA21APRO')
INSERT Assess(AssID, ClassID) VALUES ('PROPE', 'FA21APRO')
INSERT Assess(AssID, ClassID) VALUES ('PROFE', 'FA21APRO')

```

Grade (1 phần của Grade)

```

INSERT Grade(StudentID, AssID, Score, [Date]) VALUES ('HE00001', 'CEAAS1', 6, CAST('2021-08-01' AS date))
INSERT Grade(StudentID, AssID, Score, [Date]) VALUES ('HE00001', 'CEAAS2', 5, CAST('2021-08-01' AS date))
INSERT Grade(StudentID, AssID, Score, [Date]) VALUES ('HE00001', 'CEAFE', 7, CAST('2021-08-01' AS date))
INSERT Grade(StudentID, AssID, Score, [Date]) VALUES ('HE00001', 'CEAPT1', 8.6, CAST('2021-08-01' AS date))
INSERT Grade(StudentID, AssID, Score, [Date]) VALUES ('HE00001', 'CEAPT2', 1.2, CAST('2021-08-01' AS date))
INSERT Grade(StudentID, AssID, Score, [Date]) VALUES ('HE00001', 'CSIAS1', 7, CAST('2021-08-01' AS date))
INSERT Grade(StudentID, AssID, Score, [Date]) VALUES ('HE00001', 'CSIAS2', 9, CAST('2021-08-01' AS date))
INSERT Grade(StudentID, AssID, Score, [Date]) VALUES ('HE00001', 'CSIFE', 10, CAST('2021-08-01' AS date))
INSERT Grade(StudentID, AssID, Score, [Date]) VALUES ('HE00001', 'CSILAB1', 10, CAST('2021-08-01' AS date))
INSERT Grade(StudentID, AssID, Score, [Date]) VALUES ('HE00001', 'CSILAB2', 0.5, CAST('2021-08-01' AS date))
INSERT Grade(StudentID, AssID, Score, [Date]) VALUES ('HE00001', 'CSIPT1', 8, CAST('2021-08-01' AS date))
INSERT Grade(StudentID, AssID, Score, [Date]) VALUES ('HE00001', 'CSIPT2', 9, CAST('2021-08-01' AS date))
INSERT Grade(StudentID, AssID, Score, [Date]) VALUES ('HE00001', 'JPDAS1', 2.6, CAST('2021-08-01' AS date))
INSERT Grade(StudentID, AssID, Score, [Date]) VALUES ('HE00001', 'JPDAS2', 7.4, CAST('2021-08-01' AS date))
INSERT Grade(StudentID, AssID, Score, [Date]) VALUES ('HE00001', 'JPDFE', 8.4, CAST('2021-08-01' AS date))

```

```

INSERT Grade(StudentID, AssID, Score, [Date]) VALUES ('HE00001', 'JPDLAB1', 5,
CAST('2021-08-01' AS date))
INSERT Grade(StudentID, AssID, Score, [Date]) VALUES ('HE00001', 'JPDLAB2', 6,
CAST('2021-08-01' AS date))
INSERT Grade(StudentID, AssID, Score, [Date]) VALUES ('HE00001', 'JPDPT1', 8, CAST('2021-
08-01' AS date))
INSERT Grade(StudentID, AssID, Score, [Date]) VALUES ('HE00001', 'JPDPT2', 9, CAST('2021-
08-01' AS date))
INSERT Grade(StudentID, AssID, Score, [Date]) VALUES ('HE00001', 'MAEAS1', 10,
CAST('2021-08-01' AS date))

```

Sử dụng Index, View, Trigger và Store Procedure

1. Index

The screenshot shows a SQL Server query window with the following code:

```

1 select * from Student
2 create index Student_name on Student([LastName], [FirstName])
3
4 select * from Lecture
5 create index Lecture_name on Lecture([Last Name], [First Name])
6
7 select * from Student where [LastName] = N'Nguyễn Thị' and [FirstName] = N'B'

```

Below the query window, the 'Results' tab is active, displaying a table with the following data:

	StudentID	Email	FirstName	LastName	Gender	DoB
1	HE00028	bnt@fpt.edu.vn	B	Nguyễn Thị	0	2002-02-02

- Do không thể nhớ được hết StudentID và LectureID nên thao tác tìm, sắp xếp, sửa đổi chủ yếu sẽ do tên
- Dữ liệu của ta gồm 2 phần First Name và Last Name nên khi tìm phải where 2 cột lại với nhau khá mất thời gian
- Nên tạo luôn 1 index cho 2 cột này ở 2 table Student và lecture thì khi nhập tên vào tìm kiếm, database sẽ chỉ cần dùng index để tìm ra luôn

Sử dụng Queries:

1. Query dùng Order By:

```
16
17 --
18 Select * from Student order by [FirstName], [LastName]
```

95 %

Results Messages

	StudentID	Email	FirstName	LastName	Gender	DoB
1	HE00027	ant@fpt.edu.vn	A	Nguyễn Thị	0	2002-02-01
2	HE00001	anv@fpt.edu.vn	A	Nguyễn Văn	1	2002-01-01
3	HE00028	bnt@fpt.edu.vn	B	Nguyễn Thị	0	2002-02-02
4	HE00002	bnv@fpt.edu.vn	B	Nguyễn Văn	1	2002-01-02
5	HE00029	cnt@fpt.edu.vn	C	Nguyễn Thị	0	2002-02-03

- Query này cho ta xem toàn bộ sinh viên đã đăng kí vào trường

2. Query dùng inner join

```
21
22 --Inner join
23 select Student.StudentID, [LastName], [FirstName], Class.ClassID, AssignID, Semester, [Start Date], [End Date]
24 from Student inner join [Student Group] on Student.StudentID = [Student Group].StudentID
25         inner join [Join] on [Join].GroupID = [Student Group].GroupID
26         inner join Class on [Join].ClassID = Class.ClassID
```

95 %

Results Messages

	StudentID	LastName	FirstName	ClassID	AssignID	Semester	Start Date	End Date
1	HE00001	Nguyễn Văn	A	SU21KCSI	KCSI	SUMMER 2021	2021-05-01	2021-08-01
2	HE00001	Nguyễn Văn	A	SU21IMAE	IMAE	SUMMER 2021	2021-05-01	2021-08-01
3	HE00001	Nguyễn Văn	A	SU21JCEA	JCEA	SUMMER 2021	2021-05-01	2021-08-01
4	HE00001	Nguyễn Văn	A	SU21ISSL	ISSL	SUMMER 2021	2021-05-01	2021-08-01
5	HE00001	Nguyễn Văn	A	SU21UPDE	UPDE	SUMMER 2021	2021-05-01	2021-08-01

- Query sẽ cho ra 1 bảng gồm 8 cột và là sinh viên, lớp học tham gia ở các kì