

Nguyễn Văn Vương

HE163581 | IA1604

**Student Grading Manage**

DBI202 – Lecturer: sonnt5

1. **Phân thích và thiết kế hệ thống**
2. **Hệ thống quản lí điểm**
   1. **Dữ liệu đầu vào**

* Thông tin sinh viên, thông tin giảng viên, thông tin lớp học, thông tin về điểm
* Các yêu cầu tìm kiếm
  1. **Dữ liệu đầu ra**
* Bảng điểm từng môn học
* Bảng các môn đã học qua các kì
* Bảng chi tiết về các Category của môn học
* Điều kiện để **Pass** các môn:
  + Average mark >=5.0: PASSED
  + Average mark < 5.0: NOT PASSED
  + Progress Tests or Practical Exam = 0: NOT PASSED
  + Final Exam < 4.0: NOT PASSED
  + Số buổi nghỉ > 20% tổng số buổi học: NOT PASSED
* Xếp loại học tập
  + 8.0 -> 8.99: xếp loại sinh viên Giỏi
  + 9.0 -> 10.0: xếp loại sinh viên Xuất sắc
  + 9.0 -> 10.0 và điểm phong trào đạt 10.0: Đạt Cóc vàng.

**2.Xác định và tổng hợp các dữ liệu**

Bảng thông tin các điểm thành phần



**Category:**

Mô tả: Danh sách các đầu điểm

* Quiz1
* Quiz2
* ProgressTest

- Final Exam

**Part:**

Mô tả: Số lượng đầu điểm trên từng hạng mục

**Weight:**

Mô tả: Trọng số của từng đầu điểm

**Completion Criteria:**

Mô tả: Điều kiện thi và điểm tối thiểu để pass

**Duration:**

Mô tả: Lượng thời gian và địa điểm thực hiện đầu điểm

**Question Type:**

Mô tả: Cách để lấy đầu điểm

- Multiple Choice

- Presentation

- Practice

**No Question:**

Mô tả: Số lượng câu hỏi trong đầu điểm

**Knowledge and Skill:**

Mô tả: Kiến thức và kĩ năng cần chuẩn bị cho bài thi lấy đầu điểm

**Grading Guide:**

Mô tả: Người đánh giá và cho điểm

Bảng thống kê thông tin các môn học của một học sinh:



**Subject Code:**

Mô tả: Mã môn học

**Subject Name:**

Mô tả: Tên môn học

**Semester:**

Mô tả: Học kì đã học môn học đó

**Group**

Mô tả: Lớp học mà học sinh tham gia để học môn học

**Start Date:**

Mô tả: Ngày bắt đầu kì học

**End Date:**

Mô tả: Ngày kết thúc kì học

**Average Mark:**

Mô tả: Điểm trung bình của môn học đó

**Status:**

Mô tả: Kết quả đã qua hay chưa qua môn

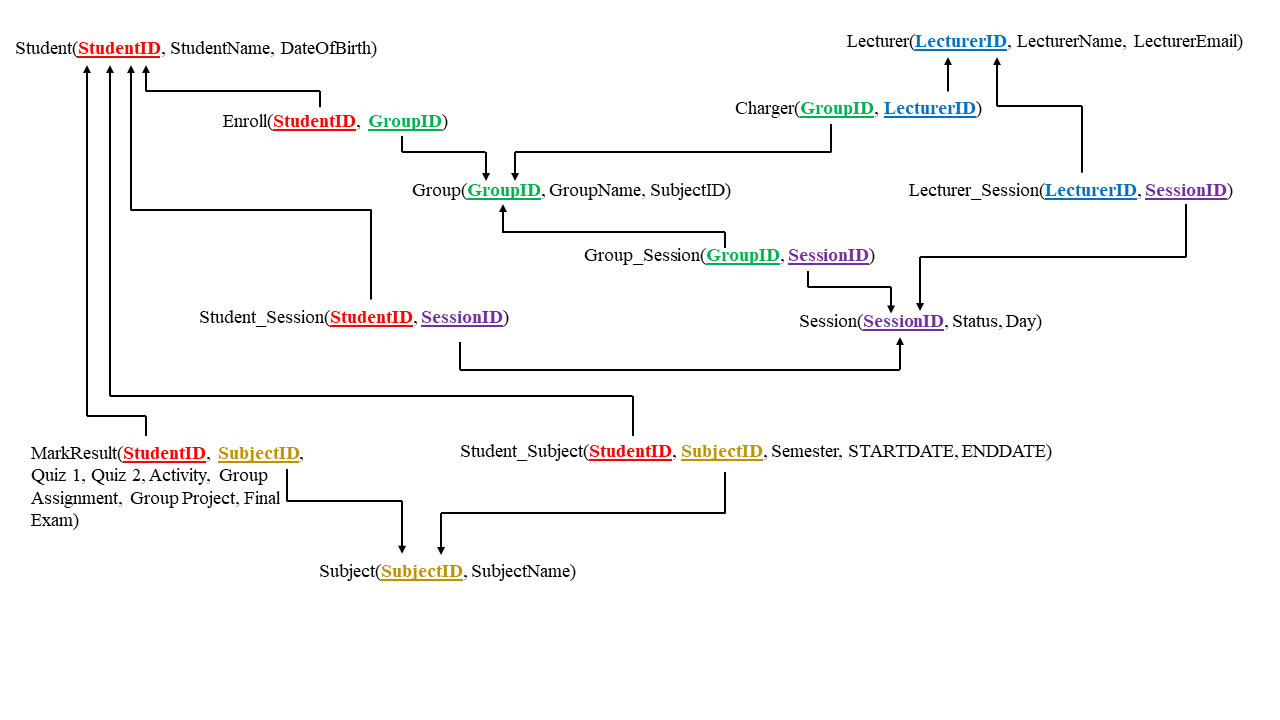
Bảng điểm của sinh viên:



**Grade Category:**

Mô tả:

**3.Phân tích, phác thảo database sang dạng bảng**



Database này sẽ được chia làm 12 bảng, trong đó:

Thực thể 1: Student bao gồm

* **StudentID**
* StudentName
* DateOfBirth
* Gender
* Address.

Thực thể 2: Group bao gồm

* **GroupID**
* GroupName
* SubjectID
* Semester

Thực thể 3: Enroll Bao gồm

* **StudentID**
* **GroupID**

Đây chính là bảng relation giữa Group và Student

Thực thể 4: Lecturerbao gồm

* **LecturerID**
* LecturerName
* LecturerEmail

Thực thể 5: Charge bao gồm

* **GroupID**
* **LecturerID**

để liên kết 2 bảng Group và Lecturer

Thực thể 6: Session bao gồm

* **SessionID**
* Status
* Day

Thực thể 7: Lecturer\_Session bao gồm

* **LecturerID**
* **SessionID**

liên kết bảng Lecturer và Session

Thực thể 8: Student\_Session bao gồm

* **StudentID**
* **SessionID**

liên kết bảng Student và Sesion

Thực thể 9: Subject bao gồm

* **SubjectID**
* SubjectName

Thực thể 10: Semester bao gồm

* **Semester**
* Start
* End

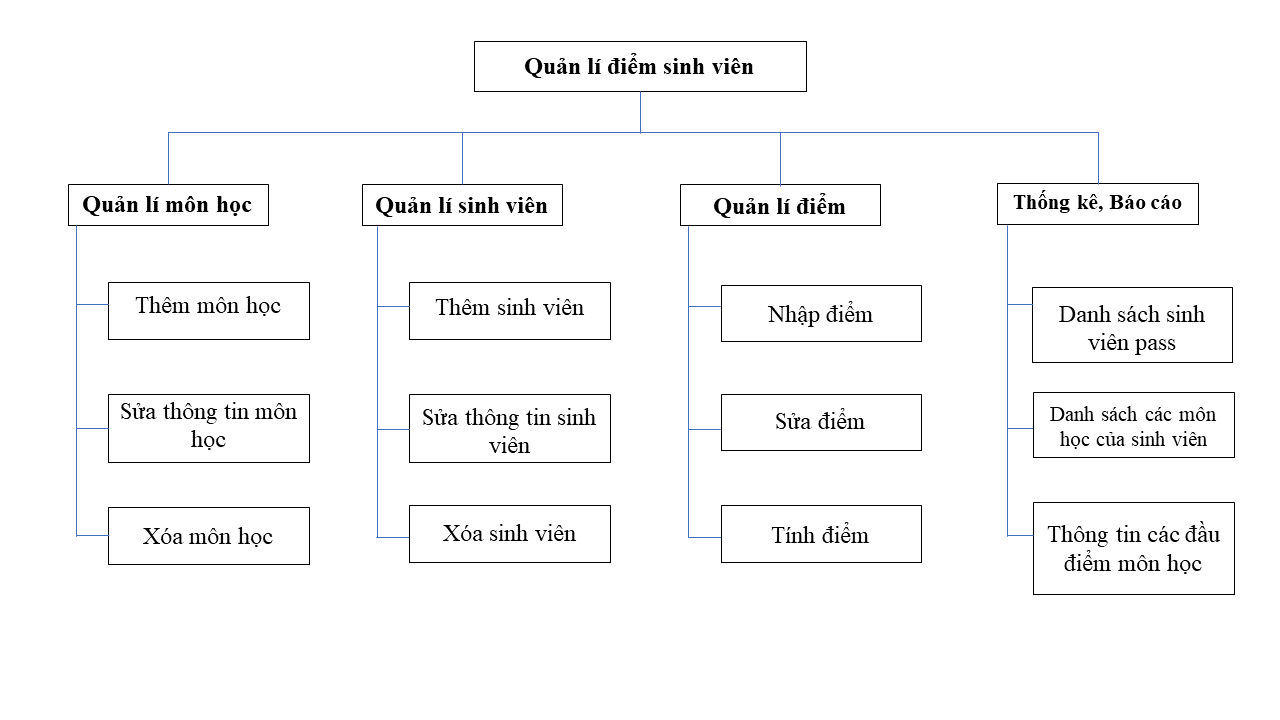
Thực thể 11: Assessment bao gồm

* **AssessmentID**
* Name
* Weight
* SubjectID

Thực thể 12: Assessment\_Student bao gồm

* **AssessmentID**
* **Student**
* Date
* Score

1. **Biểu đồ phân cấp chức năng**



Các chức năng cụ thể:

* Quản lý môn học:

+ Có các chức năng thêm, sửa, xóa các môn học theo yêu cầu

+ Cung cấp các thông tin về môn học trong mỗi kì học cho nhà trường, giáo viên và sinh viên khi có yêu cầu

* Quản lí sinh viên

+ Có các chức năng thêm, sửa xóa thông tin sinh viên theo yêu cầu của nhà trường

+ Cung cấp thông tin về sinh viên cho nhà trường, giáo viên theo yêu

* Quản lí điểm

+ Có chức năng cập nhật điểm cho sinh viên sau mỗi kì học

+ Thực hiện phúc khảo điểm cho sinh viên khi sinh viên yêu cầu

+ Báo cáo, thống kê điểm của sinh viên cho nhà trường, giáo viên và sinh viên khi có yêu cầu

* Thống kê, báo cáo

+ Thống kê danh sách sinh viên pass từng môn học sau mỗi khì học

+ Thống kê tỷ lệ sinh viên pass và not pass sau mỗi kì

+ Thống kê danh sách môn học của từng kì

+ Thông kê danh sách điểm của từng sinh viên

+ Thống kê các sinh viên theo từng kì học

**3.Chuẩn hóa thuộc tính các Attribute trên từng bảng**

1.Table Student

|  |  |
| --- | --- |
| Attributes | Type |
| StudentID | NVARCHAR |
| StudentName | NVARCHAR |
| DateOfBirth | DATE |

2.Table Group

|  |  |
| --- | --- |
| Attributes | Type |
| GroupID | NVARCHAR |
| GroupName | NVARCHAR |
| SubjectID | NVARCHAR |
| Semester | NVARCHAR |

3.Table Enroll

|  |  |
| --- | --- |
| Attributes | Type |
| StudentID | NVARCHAR |
| GroupID | NVARCHAR |

4.Table Semester

|  |  |
| --- | --- |
| Attributes | Type |
| Semester | NVARCHAR |
| Start | DATE |
| End | DATE |

5.Table Lecturer

|  |  |
| --- | --- |
| Attributes | Type |
| Lecturer | NVARCHAR |
| LecturerName | NVARCHAR |
| LecturerEmail | NVARCHAR |

6.Table Charge

|  |  |
| --- | --- |
| Attributes | Type |
| GroupID | NVARCHAR |
| LecturerID | NVARCHAR |

7.Table Session

|  |  |
| --- | --- |
| Attributes | Type |
| SessionID | NVARCHAR |
| Status | BIT |
| Day | DATE |

8.Table Student\_Session

|  |  |
| --- | --- |
| Attributes | Type |
| StudentID | NVARCHAR |
| SessionID | NVARCHAR |

9.Table Lecturer\_Session

|  |  |
| --- | --- |
| Attributes | Type |
| LecturerID | NVARCHAR |
| SessionID | NVARCHAR |

10.Table Assessment

|  |  |
| --- | --- |
| Attributes | Type |
| AssessmentID | NVARCHAR |
| Name | NVARCHAR |
| Weight | FLOAT |
| SubjectID | NVARCHAR |

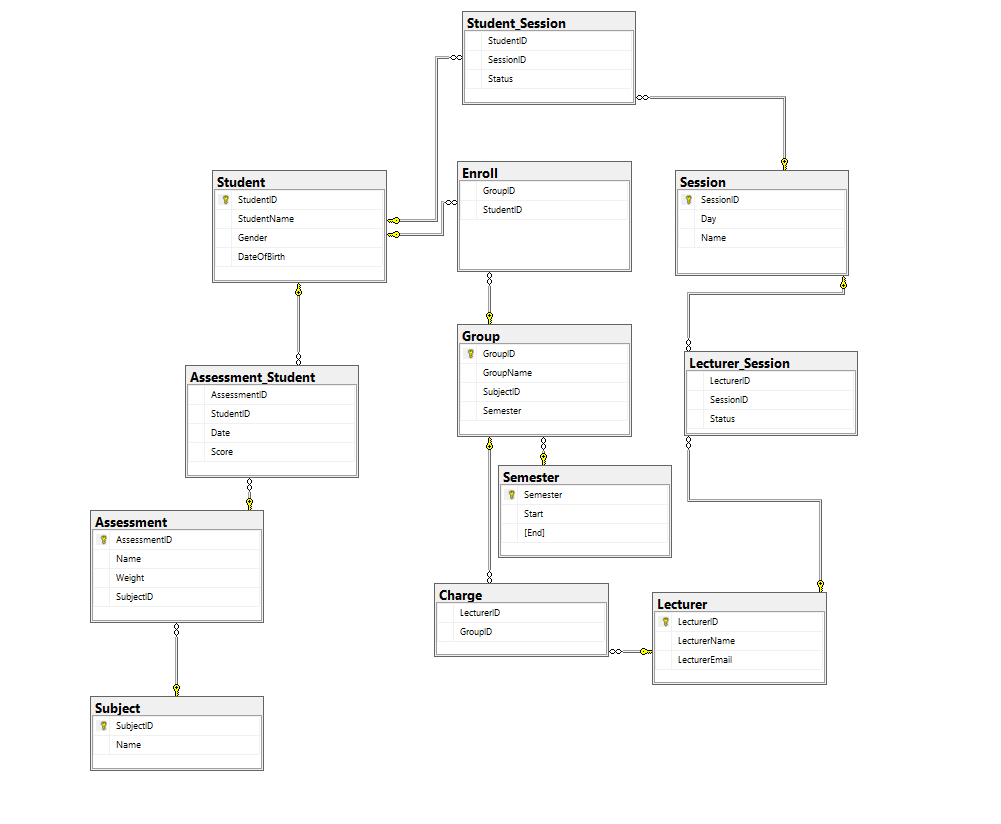
11.Table Assessment\_Student

|  |  |
| --- | --- |
| Attributes | Type |
| StudentID | NVARCHAR |
| StudentID | NVARCHAR |
| Date | DATE |
| Score | FLOAT |

12.Table Subject

|  |  |
| --- | --- |
| Attributes | Type |
| SubjectID | NVARCHAR |
| SubjectName | NVARCHAR |

1. **THIẾT KẾ CƠ SỞ DỮ LIỆU**
2. **Thiết kế cơ sở dữ liệu**



**Biểu đồ cấu trúc dữ liệu theo mối quan hệ**

* Danh sách các thuộc tính:
* Danh sách các thuộc tính lặp:
* Danh sách các phụ thuộc hàm

1. **Một số truy vấn SQL**

Create Database DBI202\_Assignment\_Vuong

USE [DBI202\_Assignment\_Vuong]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

--Tao Bang Assessment--

CREATE TABLE [dbo].[Assessment](

[AssessmentID] [nvarchar](50) NOT NULL,

[Name] [nvarchar](50) NOT NULL,

[Weight] [float] NOT NULL,

[SubjectID] [nvarchar](50) NOT NULL,

CONSTRAINT [PK\_Assessment] PRIMARY KEY CLUSTERED

(

[AssessmentID] 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

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Assessment\_Student](

[AssessmentID] [nvarchar](50) NOT NULL,

[StudentID] [nvarchar](50) NOT NULL,

[Date] [date] NOT NULL,

[Score] [float] NOT NULL

) ON [PRIMARY]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Charge](

[LecturerID] [nvarchar](50) NOT NULL,

[GroupID] [nvarchar](50) NOT NULL

) ON [PRIMARY]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Enroll](

[GroupID] [nvarchar](50) NOT NULL,

[StudentID] [nvarchar](50) NOT NULL

) ON [PRIMARY]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Group](

[GroupID] [nvarchar](50) NOT NULL,

[GroupName] [nvarchar](50) NOT NULL,

[SubjectID] [nvarchar](50) NOT NULL,

[Semester] [nvarchar](50) 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

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Lecturer](

[LecturerID] [nvarchar](50) NOT NULL,

[LecturerName] [nvarchar](50) NOT NULL,

[LecturerEmail] [nvarchar](50) NOT NULL,

CONSTRAINT [PK\_Lecturer] PRIMARY KEY CLUSTERED

(

[LecturerID] 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

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Lecturer\_Session](

[LecturerID] [nvarchar](50) NOT NULL,

[SessionID] [nvarchar](50) NOT NULL,

[Status] [bit] NOT NULL

) ON [PRIMARY]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Semester](

[Semester] [nvarchar](50) NOT NULL,

[Start] [date] NOT NULL,

[End] [date] NOT NULL,

CONSTRAINT [PK\_Semester] PRIMARY KEY CLUSTERED

(

[Semester] 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

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Session](

[SessionID] [nvarchar](50) NOT NULL,

[Day] [date] NOT NULL,

[Name] [nvarchar](50) NULL,

CONSTRAINT [PK\_Session] PRIMARY KEY CLUSTERED

(

[SessionID] 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

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Student](

[StudentID] [nvarchar](50) NOT NULL,

[StudentName] [nvarchar](50) NOT NULL,

[Gender] [bit] NOT NULL,

[DateOfBirth] [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

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Student\_Session](

[StudentID] [nvarchar](50) NOT NULL,

[SessionID] [nvarchar](50) NOT NULL,

[Status] [bit] NOT NULL

) ON [PRIMARY]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Subject](

[SubjectID] [nvarchar](50) NOT NULL,

[Name] [nvarchar](50) NOT NULL,

CONSTRAINT [PK\_Subject] PRIMARY KEY CLUSTERED

(

[SubjectID] 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

ALTER TABLE [dbo].[Assessment] WITH CHECK ADD CONSTRAINT [FK\_Assessment\_Subject] FOREIGN KEY([SubjectID])

REFERENCES [dbo].[Subject] ([SubjectID])

GO

ALTER TABLE [dbo].[Assessment] CHECK CONSTRAINT [FK\_Assessment\_Subject]

GO

ALTER TABLE [dbo].[Assessment\_Student] WITH CHECK ADD CONSTRAINT [FK\_Assessment\_Student\_Assessment] FOREIGN KEY([AssessmentID])

REFERENCES [dbo].[Assessment] ([AssessmentID])

GO

ALTER TABLE [dbo].[Assessment\_Student] CHECK CONSTRAINT [FK\_Assessment\_Student\_Assessment]

GO

ALTER TABLE [dbo].[Assessment\_Student] WITH CHECK ADD CONSTRAINT [FK\_Assessment\_Student\_Student] FOREIGN KEY([StudentID])

REFERENCES [dbo].[Student] ([StudentID])

GO

ALTER TABLE [dbo].[Assessment\_Student] CHECK CONSTRAINT [FK\_Assessment\_Student\_Student]

GO

ALTER TABLE [dbo].[Charge] WITH CHECK ADD CONSTRAINT [FK\_Charge\_Group] FOREIGN KEY([GroupID])

REFERENCES [dbo].[Group] ([GroupID])

GO

ALTER TABLE [dbo].[Charge] CHECK CONSTRAINT [FK\_Charge\_Group]

GO

ALTER TABLE [dbo].[Charge] WITH CHECK ADD CONSTRAINT [FK\_Charge\_Lecturer] FOREIGN KEY([LecturerID])

REFERENCES [dbo].[Lecturer] ([LecturerID])

GO

ALTER TABLE [dbo].[Charge] CHECK CONSTRAINT [FK\_Charge\_Lecturer]

GO

ALTER TABLE [dbo].[Enroll] WITH CHECK ADD CONSTRAINT [FK\_Enroll\_Group] FOREIGN KEY([GroupID])

REFERENCES [dbo].[Group] ([GroupID])

GO

ALTER TABLE [dbo].[Enroll] CHECK CONSTRAINT [FK\_Enroll\_Group]

GO

ALTER TABLE [dbo].[Enroll] WITH CHECK ADD CONSTRAINT [FK\_Enroll\_Student] FOREIGN KEY([StudentID])

REFERENCES [dbo].[Student] ([StudentID])

GO

ALTER TABLE [dbo].[Enroll] CHECK CONSTRAINT [FK\_Enroll\_Student]

GO

ALTER TABLE [dbo].[Group] WITH CHECK ADD CONSTRAINT [FK\_Group\_Semester] FOREIGN KEY([Semester])

REFERENCES [dbo].[Semester] ([Semester])

GO

ALTER TABLE [dbo].[Group] CHECK CONSTRAINT [FK\_Group\_Semester]

GO

ALTER TABLE [dbo].[Lecturer\_Session] WITH CHECK ADD CONSTRAINT [FK\_Lecturer\_Session\_Lecturer] FOREIGN KEY([LecturerID])

REFERENCES [dbo].[Lecturer] ([LecturerID])

GO

ALTER TABLE [dbo].[Lecturer\_Session] CHECK CONSTRAINT [FK\_Lecturer\_Session\_Lecturer]

GO

ALTER TABLE [dbo].[Lecturer\_Session] WITH CHECK ADD CONSTRAINT [FK\_Lecturer\_Session\_Session] FOREIGN KEY([SessionID])

REFERENCES [dbo].[Session] ([SessionID])

GO

ALTER TABLE [dbo].[Lecturer\_Session] CHECK CONSTRAINT [FK\_Lecturer\_Session\_Session]

GO

ALTER TABLE [dbo].[Student\_Session] WITH CHECK ADD CONSTRAINT [FK\_Student\_Session\_Session] FOREIGN KEY([SessionID])

REFERENCES [dbo].[Session] ([SessionID])

GO

ALTER TABLE [dbo].[Student\_Session] CHECK CONSTRAINT [FK\_Student\_Session\_Session]

GO

ALTER TABLE [dbo].[Student\_Session] WITH CHECK ADD CONSTRAINT [FK\_Student\_Session\_Student] FOREIGN KEY([StudentID])

REFERENCES [dbo].[Student] ([StudentID])

GO

ALTER TABLE [dbo].[Student\_Session] CHECK CONSTRAINT [FK\_Student\_Session\_Student]

GO

**--Các câu lệnh--**

--1. Truy vấn điểm từng môn của sinh viên--

SELECT \*, (Case WHEN tb1.Score >5 THEN 'PASS'

ELSE 'NOT PASS'

END) as [STATUS]

FROM (SELECT s.StudentID, s.StudentName,a.SubjectID,SUM(ass.Score\* a.Weight) as [Score]

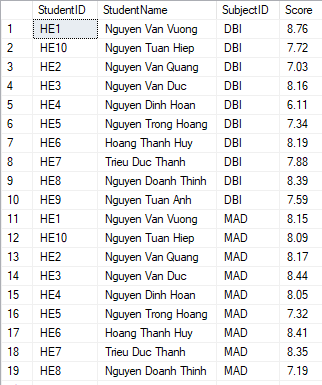
FROM Student as s

INNER JOIN Assessment\_Student as ass ON s.StudentID = ass.StudentID

INNER JOIN Assessment as a ON ass.AssessmentID = a.AssessmentID

INNER JOIN [Subject] as su ON a.SubjectID = su.SubjectID

GROUP BY s.StudentID,s.StudentName,a.SubjectID) as tb1



--2. Tính GPA của các sinh viên--

SELECT tb4.StudentID,AVG(tb4.Score) as [GPA]

FROM (SELECT tb3.StudentID,tb3.SubjectID,SUM(tb3.score \* tb3.weight) as [Score]

FROM (SELECT tb2.StudentID,tb2.SubjectID,ISNULL(a1.score, 0) as [score],at1.weight

FROM (SELECT StudentID, ass.SubjectID, ass.AssessmentID

FROM (SELECT st.StudentID,su.SubjectID

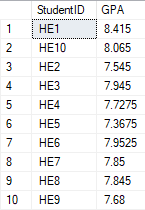
FROM Student as st, Assessment as su GROUP BY st.StudentID,su.SubjectID) as tb1 INNER JOIN Assessment as ass ON tb1.SubjectID = ass.SubjectID)

as tb2 INNER JOIN Assessment as at1 on at1.SubjectID = tb2.SubjectID and at1.AssessmentID = tb2.AssessmentID

LEFT JOIN Assessment\_Student as a1 ON tb2.AssessmentID = a1.AssessmentID AND tb2.StudentID = a1.StudentID ) as tb3

GROUP BY tb3.StudentID,tb3.SubjectID) as tb4

GROUP BY tb4.StudentID



--3. Danh sách các môn học trong các kì--

SELECT [Subject].SubjectID,[Subject].[Name],[Group].Semester,

[Group].GroupName,Semester.[Start],Semester.[End] FROM [Subject]

INNER JOIN Assessment ON [Subject].SubjectID = Assessment.SubjectID

INNER JOIN Assessment\_Student ON Assessment.AssessmentID = Assessment\_Student.AssessmentID

INNER JOIN Student ON Assessment\_Student.StudentID = Student.StudentID

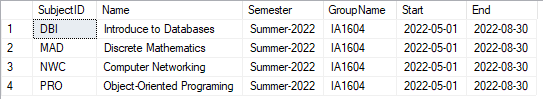
INNER JOIN Enroll ON Student.StudentID = Enroll.StudentID

INNER JOIN [Group] ON Enroll.GroupID=[Group].GroupID

INNER JOIN Semester ON [Group].Semester = Semester.Semester

GROUP BY [Subject].SubjectID,[Subject].[Name],[Group].Semester,[Group].GroupName,

Semester.[Start],Semester.[End]



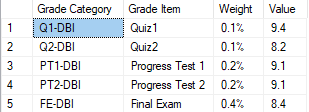
--4. Điểm thành phần của từng môn học, Ví dụ môn DBI của sinh viên HE1--

SELECT ass.AssessmentID as [Grade Category], a.[Name] as [Grade Item],CAST(a.[Weight] as nvarchar) +'%' as [Weight], ass.Score as [Value]

FROM Assessment as a

INNER JOIN Assessment\_Student ass ON ass.AssessmentID = a.AssessmentID

WHERE ass.StudentID = 'HE1' AND a.SubjectID = 'DBI'



--5. Hiển thị danh sách sinh viên của một lớp học sắp xếp theo tên sinh viên--

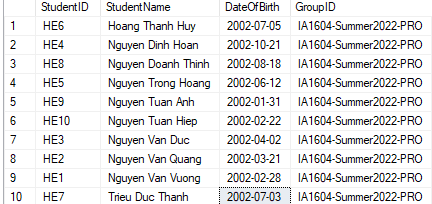
SELECT s.StudentID,s.StudentName,s.DateOfBirth,g.GroupID FROM Student as s

INNER JOIN Enroll as e ON s.StudentID = e.StudentID

INNER JOIN [Group] as g On e.GroupID = g.GroupID

WHERE g.GroupID = 'IA1604-Summer2022-PRO'

ORDER BY StudentName



--6. Danh sách các môn học của một giáo viên, Ví dụ: Lecturer: KhuongPD--

SELECT l.LecturerID,l.LecturerName,g.GroupID,g.SubjectID

FROM [Group] as g

INNER JOIN Charge as c ON g.GroupID = c.GroupID

INNER JOIN Lecturer as l on c.LecturerID = l.LecturerID

WHERE l.LecturerID = 'KhuongPD'



--7. Kiểm tra điều kiện điểm danh của sinh viên theo mã số sinh viên,

VD: Sinh viên có mã số HE2--

SELECT tb1.StudentID,tb1.StudentName,tb1.[Name],(CASE WHEN tb1.Attend >=24 THEN 'PASS'

ELSE 'NOT PASS'

END) as [Condition]

FROM (SELECT s.StudentID,s.StudentName,se.[Name],COUNT(ss.[Status]) as [Attend]

FROM Student as s

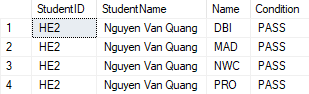
INNER JOIN [Student\_Session] as ss ON s.StudentID=ss.StudentID

INNER JOIN [Session] as se ON ss.SessionID = se.SessionID

WHERE ss.[Status] = 1

GROUP BY s.StudentID,s.StudentName,se.[Name]) as tb1

WHERE tb1.StudentID = 'HE2'



8. Danh sách điểm của sinh viên theo môn