

Nguyễn Văn Vương

HE163581 | IA1604

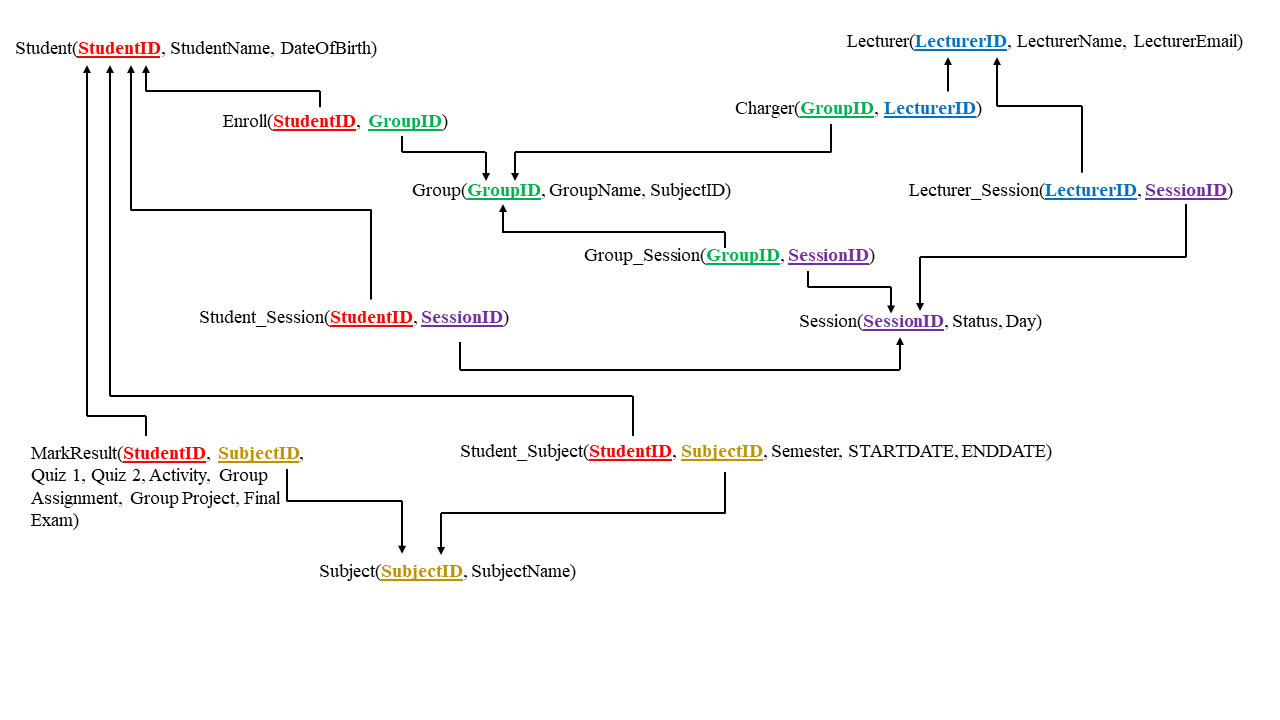
**Student Grading Manage**

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.

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



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

Bảng Student bao gồm: **StudentID**, StudentName, DateOfBirth, Gender, Address.

Bảng Group: **GroupID**, GroupName, SubjectID

Bảng Enroll: Bao gồm **StudentID,** **GroupID**. Đây chính là bảng relation giữa Group và Student

Bảng Lecturer: **LecturerID**, LecturerName, LecturerEmail

Bảng Charge: **GroupID**, **LecturerID** để liên kết 2 bảng Group và Lecturer

Bảng Session: **SessionID**, Status, Day

Bảng Group\_Session: **GroupID**, **SessionID** liên kết bảng Group và Session

Bảng Lecturer\_Session: **LecturerID**, **SessionID** liên kết bảng Lecturer và Session

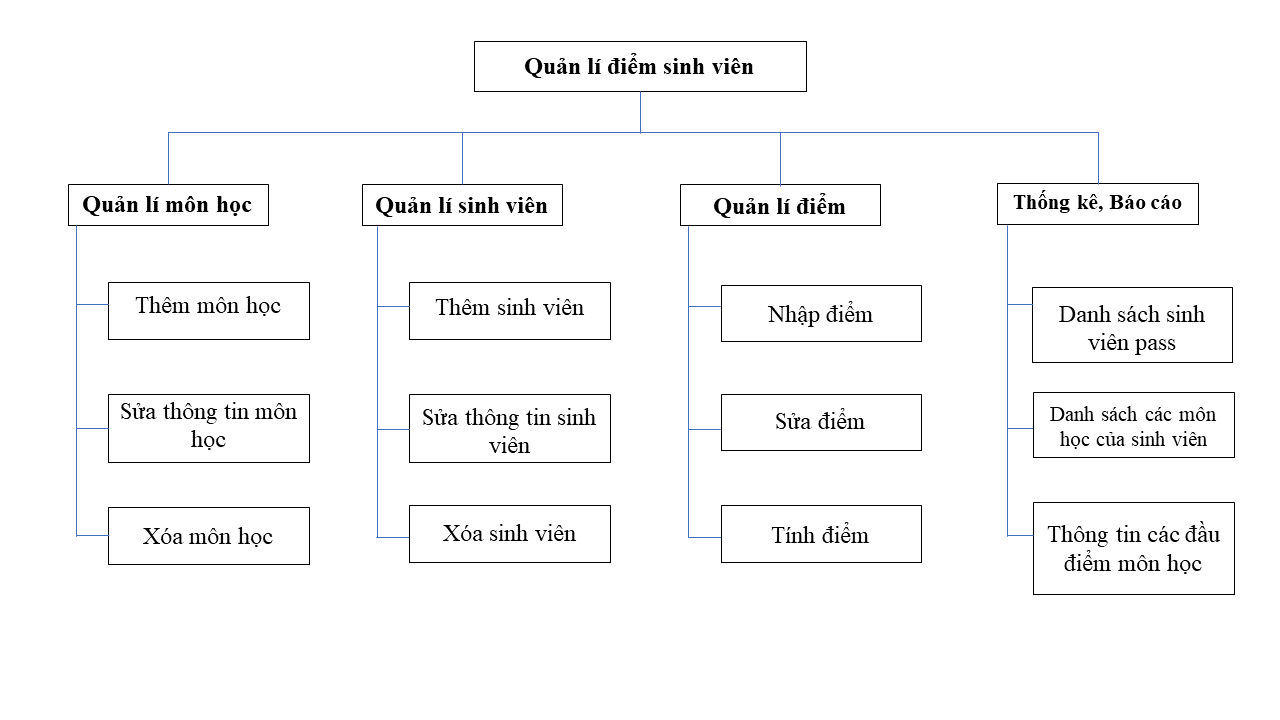
Bảng Student\_Session: **StudentID**, **SessionID** liên kết bảng Student và Sesion

Bảng Subject: **SubjectID**, SubjectName

Bảng MarkResult: **StudentID**, **SubjectID**, Quiz 1, Quiz 2, Activity, Group Assignment, Group Project, Final Exam biểu diễn, truy xuất điểm từng môn của sinh viên

Bảng Student\_Subject: **StudentID**, **SubjectID**, Semester, STARTDATE, ENDDATE

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

+ Cung

* 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

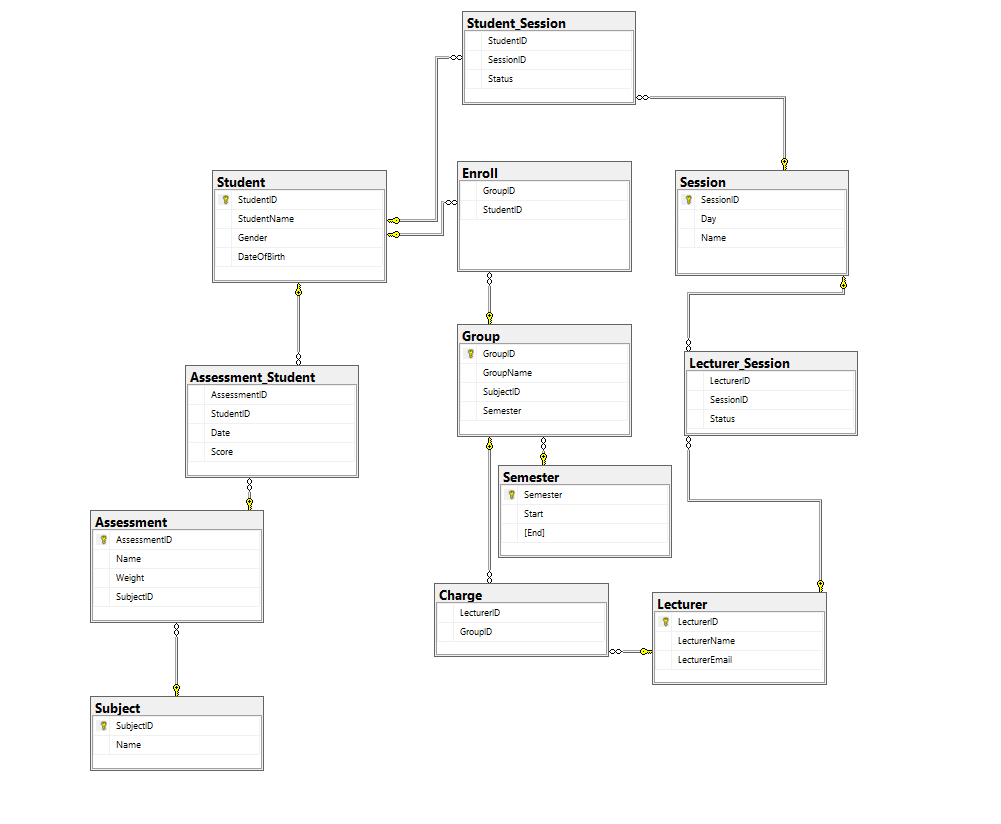
+ Thống kê tỷ lệ sinh viên pass và not pass sau mỗi kì

+ 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

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

--Tao bang Student—

Create Table Student(

StudentID Nvarchar(50) Primary key NOT NULL,

StudentName Nvarchar(50) NOT NULL,

Gender Bit NOT NULL,

DateOfBirth Date NOT NULL

)

--Tao bang Assessment—

Create Table Assessment(

AssessmentID NVARCHAR(50) Primary key not null,

Name Nvarchar(50) NOT NULL,

Weight Float NOT NULL,

SubjectID NVARCHAR(15) NOT NULL

)

--Tao bang Subject—

CREATE TABLE Subject(

SubjectID NVARCHAR(15) PRIMARY KEY NOT NULL,

Name NVARCHAR(50) NOT NULL

CONSTRAINT FK\_Assessment FOREIGN KEY (SubjectID) REFERENCE Assessment(SubjectID)

)

--Tao bang Assessment\_Student--

CREATE TABLE Assessment\_Student(

AssessmentID NVARCHAR(50) NOT NULL FOREIGN KEY Assessment(AssessmentID),

StudentID NVARCHAR(50) NOT NULL FOREIGN KEY Student(StudentID),

Date DATE NOT NULL,

Score FLOAT NOT NULL

)

--Tao bang Group—

CREATE TABLE Group(

GroupID NVARCHAR(50) PRIMARY KEY NOT NULL,

GroupName NVARCHAR(50) NOT NULL,

SubjectID NVARCHAR(15) NOT NULL,

Semester NVARCHAR(15) NOT NULL

)

--Tao bang Enroll

CREATE TABLE Enroll(

GroupID NVARCHAR(50) NOT NULL FOREIGN KEY Group(GroupID),

StudentID NVARCHAR(50) NOT NULL FOREIGN KEY Student(StudentID)

)

--Tao bang Lecturer--

CREATE TABLE Lecturer(

LecturerID NVARCHAR(50) PRIMARY KEY NOT NULL,

LecturerName NVARCHAR(50) NOT NULL,

LecturerEmail NVARCHAR(50) NOT NULL

)

--Tao bang Charge—

CREATE TABLE Charge(

LecturerID NVARCHAR(50) NOT NULL FOREGIN KEY Lecturer(LecturerID),

GroupID NVARCHAR(50) NOT NULL FOREIGN KEY Group(GroupID)

)

--Tao bang Session—

CREATE TABLE Session(

SessionID NVARCHAR(50) PRIMARY KEY NOT NULL,

Day DATE NOT NULL,

Name NVARCHAR(50) NOT NULL

)

--Tao bang Student\_Session--

CREATE TABLE Student\_Session(

StudentID NVARCHAR(50) NOT NULL FOREIGN KEY Student(StudentID),

SessionID INT NOT NULL FOREIGN KEY Session(SessionID),

Status Bit

)

--Tao bang Lecturer\_Session—

CREATE TABLE Lecturer\_Session(

LecturerID NVARCHAR(50) NOT NULL FOREIGN KEY Lecturer(LecturerID),

SessionID NVARCHAR(50) NOT NULL FOREIGN KEY Session(SessionID),

Status Bit

)

**--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]

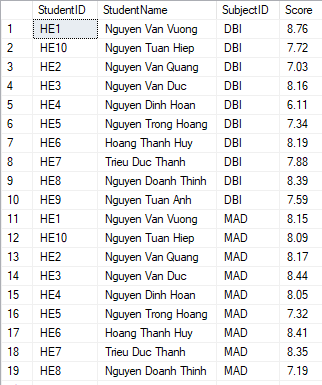
FROM Student as s

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

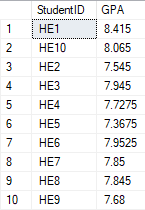
FROM Student as st, Assessment as su GROUP BY st.StudentID,su.SubjectID) as tb1 INNER JOIN Assessment as ass ON tb1.SubjectID = ass.SubjectID)

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

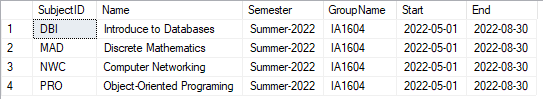
INNER JOIN Enroll ON Student.StudentID = Enroll.StudentID

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]



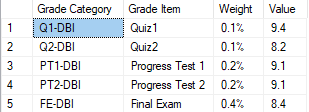
--4. Điểm thành phần của từng môn học, Ví dụ môn DBI của sinh viên HE1--

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--

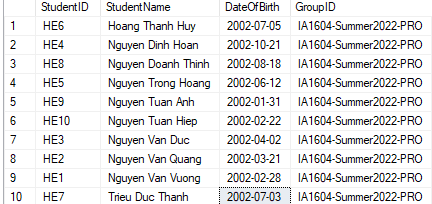
SELECT s.StudentID,s.StudentName,s.DateOfBirth,g.GroupID FROM Student as s

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

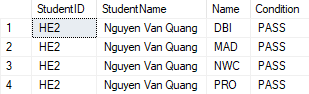
INNER JOIN [Student\_Session] as ss ON s.StudentID=ss.StudentID

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