

HỆ THỐNG PHỤ GIÚP QUẢN LÝ ĐIỂM SINH VIÊN

I. Mô tả hệ thống.

1. Nhiệm vụ cơ bản :

Theo dõi quá trình học tập của sinh viên, đảm bảo không xảy ra sai sót trong quá trình nhập điểm và in điểm. Hệ thống này giúp phòng đào tạo theo dõi, đánh giá kết quả học tập của sinh viên một cách nhanh nhất và chính xác nhất, giảm đáng kể công sức, giấy tờ, sổ sách lưu trữ.

Ngoài việc quản lý điểm hệ thống cũng sẽ giúp tra cứu một vài thông tin cơ bản như : tên, tuổi, ngày sinh

2. Hệ thống này gồm có 9 bảng như sau

1. Students.

- Gồm 10 sinh viên.
- Các trường : StudentID, name, gender, Dob.

2. Subject

- Gồm 7 môn học.
- Các trường: SubjectID , name.

3. Semester

- Gồm có 2 kỳ học.
- Các trường : Semester ID, name.

4. Assessment

- Miêu tả điểm thành phần môn học.
- Các trường : AssessmentID , name, weight, StudentID

5. Group

- Gồm 2 group
- Các trường : GroupID, name.

6. Student_Group

- Cho biết sinh viên tương ứng với từng group
- Các trường : StudentID, GroupID

7. Student_Assessment

- Cho biết điểm tương ứng sinh viên
- Các trường : StudentID, AssessmentID, date, Score

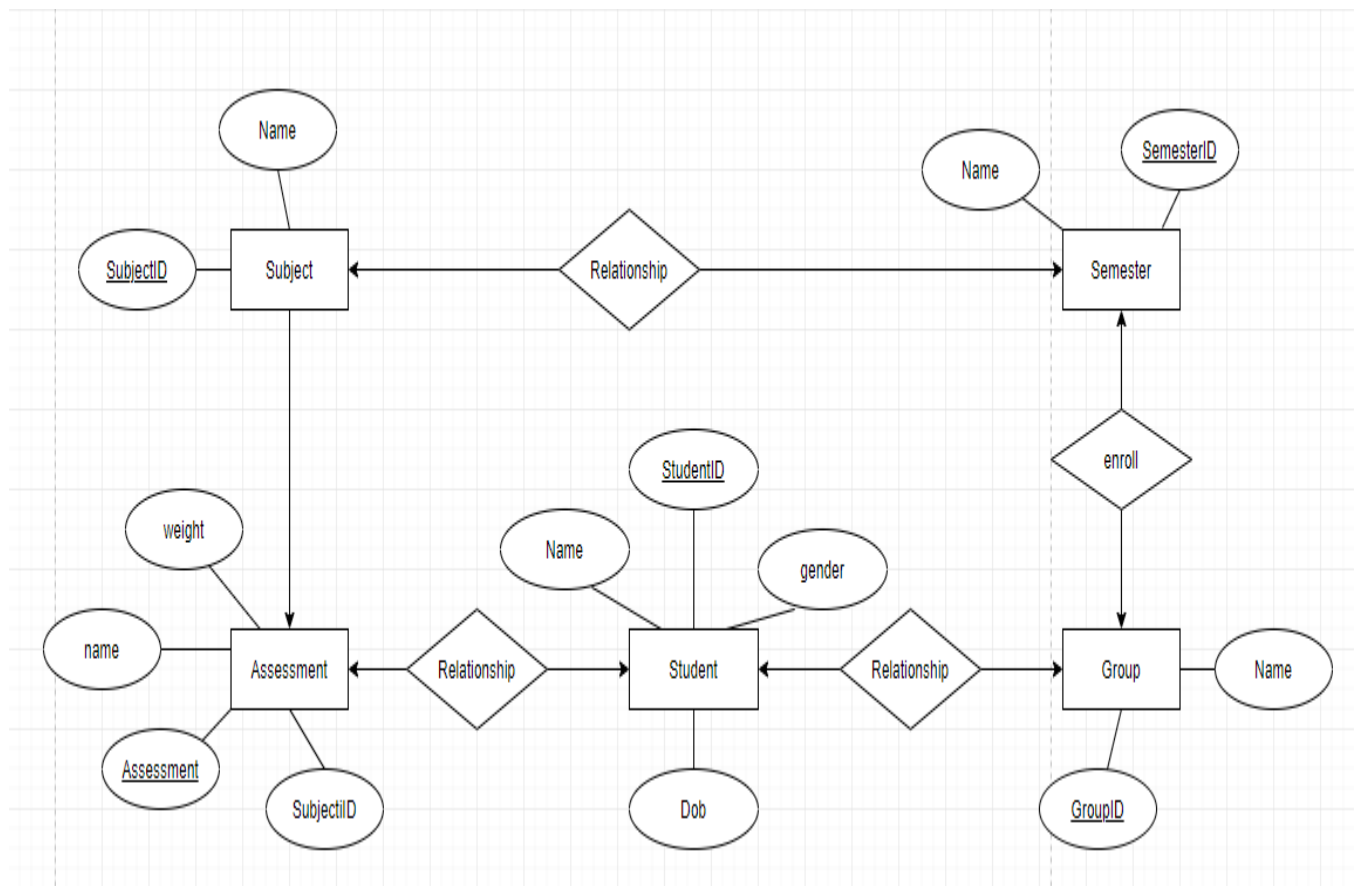
8. Semester_Subject

- Cho biết môn học của từng kỳ học.
- Các trường : SemesterID, SubjectID

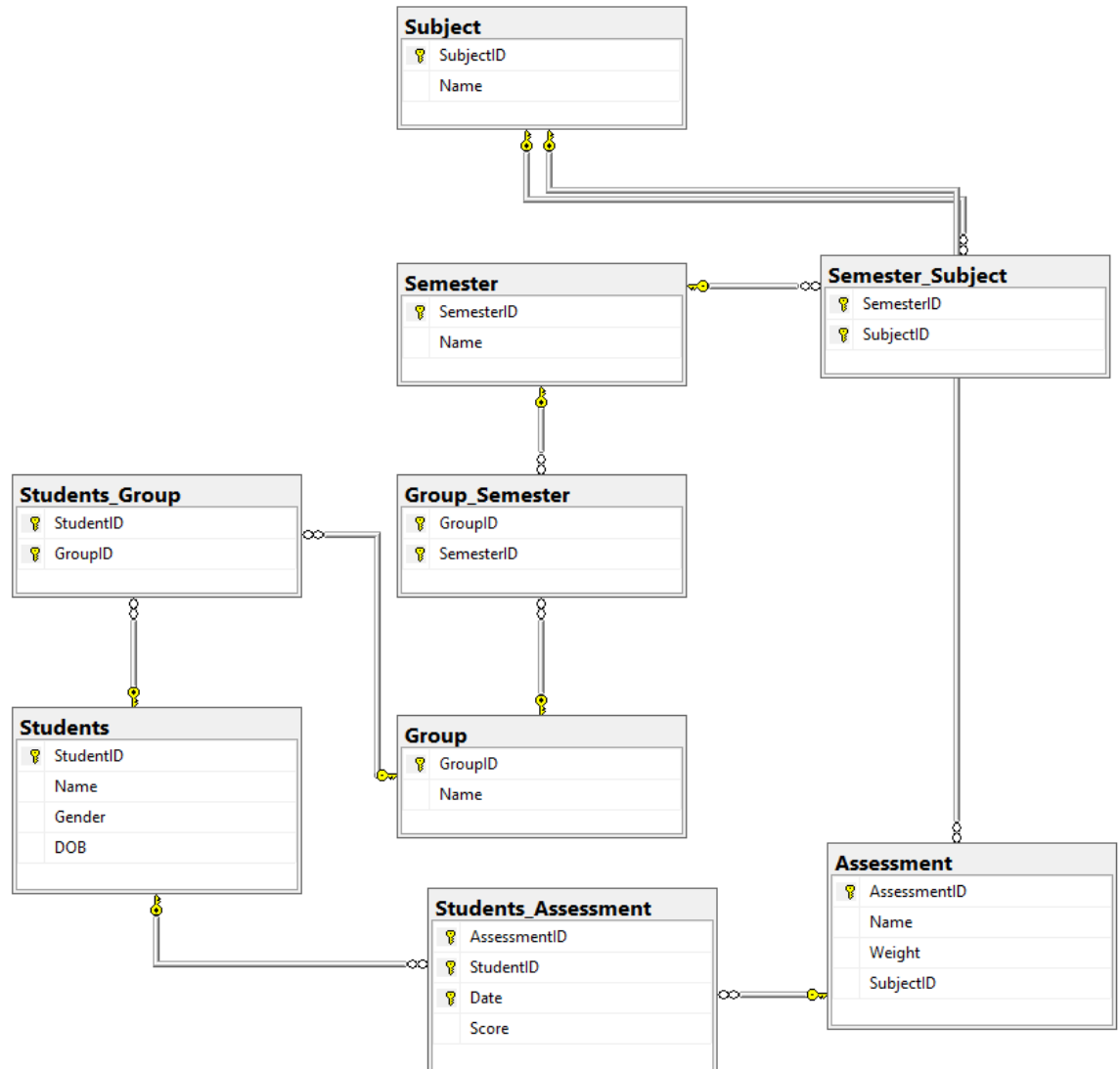
9. Group_Semester

- Cho biết kỳ học có những lớp học nào
- Gồm các trường : GroupID, SemesterID

II . ERD mô tả cơ sở dữ liệu.



III. Lược đồ quan hệ suy ra từ ERD.



IV. Câu Lệnh tạo bảng

```
Create table Students(  
    StudentID int not null primary key,  
    [Name] varchar(150) not null,  
    Gender varchar(150) not null,  
    DOB date not null  
);
```

```
Create table Subject(  
    SubjectID int not null primary key,  
    [Name] varchar(150) not null,  
);
```

```
Create table Semester(  
    SemesterID int not null primary key,  
    [Name] varchar(150) not null  
);
```

```
Create table Semester_Subject(  
    SemesterID int not null,  
    SubjectID int not null,  
    primary key (SemesterID,SubjectID),  
    FOREIGN KEY (SemesterID) REFERENCES [Semester](SemesterID),  
    FOREIGN KEY (SubjectID) REFERENCES [Subject] (SubjectID)  
);
```

```
Create table [Group](  
    GroupID int not null primary key,  
    [Name] varchar(150) not null,  
);
```

```
Create table Students_Group(  
    StudentID int not null,  
    GroupID int not null,  
    primary key(StudentID, GroupID),  
    FOREIGN KEY (StudentID) REFERENCES [Students](StudentID),  
    FOREIGN KEY (GroupID) REFERENCES [Group] (GroupID)  
);
```

```
Create table Assessment(  
    AssessmentID int not null primary key,  
    [Name] varchar(150) not null,  
    [Weight] int not null,  
    SubjectID INT FOREIGN KEY REFERENCES [Subject](SubjectID)  
);
```

```
Create table Students_Assessment(  
    AssessmentID INT FOREIGN KEY REFERENCES  
[Assessment](AssessmentID),  
    StudentID int FOREIGN KEY REFERENCES Students(StudentID),  
    [Date] date not null,  
    Score int not null,  
    primary key (AssessmentID,StudentID,[Date])  
);
```

```
Create table Group_Semester(  
    GroupID int not null,  
    SemesterID int not null,  
    primary key ( GroupID, SemesterID),  
    FOREIGN KEY (GroupID) REFERENCES [Group](GroupID),  
    FOREIGN KEY (SemesterID) REFERENCES [Semester] (SemesterID)  
);
```

V. 10 truy vấn dựa trên cơ sở dữ liệu

1. Sử dụng **ORDER BY**

SELECT * FROM Students **ORDER BY** dob DESC

	StudentID	Name	Gender	DOB
1	6	Bùi Mạnh Tuấn	Nam	2002-10-11
2	5	Vũ Thị Thanh Thảo	Nữ	2002-09-11
3	10	Nguyễn Đức Thắng	Nam	2002-07-11
4	3	Nguyễn Việt Thắng	Nam	2002-07-05
5	9	Nguyễn Thị Phương Diệu	Nữ	2002-05-10
6	2	Hoàng Thế Anh	Nam	2002-05-07
7	7	Chu Hoàng Lan Anh	Nữ	2002-05-01
8	8	Phùng Quý Quốc	Nam	2002-04-03
9	4	Nguyễn Quang Minh	Nam	2002-03-02
10	1	Nguyễn Hải Dương	Nam	2002-01-02

Sắp xếp lại danh sách học sinh theo ngày tháng năm sinh

2. Sử dụng **AGGREGATE FUNCTIONS**

```
SELECT s.Name, su.Name as [Subject], SUM((Score * Weight/100)) as [Total]
FROM
    Subject su INNER JOIN Assessment a ON su.SubjectID = a.SubjectID
    INNER JOIN Students_Assessment sa ON a.AssessmentID =
        sa.AssessmentID
    INNER JOIN Students s ON s.StudentID = sa.StudentID
GROUP BY s.Name, su.Name
```

	Name	Subject	Total
1	Bùi Mạnh Tuấn	CSD201	6,15
2	Chu Hoàng Lan Anh	CSD201	6,45
3	Hoàng Thế Anh	CSD201	6,1
4	Nguyễn Đức Thắng	CSD201	8,8
5	Nguyễn Hải Dương	CSD201	6,4
6	Nguyễn Quang Minh	CSD201	7,05
7	Nguyễn Thị Phương Diệu	CSD201	7,425
8	Nguyễn Việt Thắng	CSD201	6,5
9	Phùng Quý Quốc	CSD201	7,25
10	Vũ Thị Thanh Thảo	CSD201	6,225

Điểm trung bình môn của sinh viên

3. Sử dụng **GROUP BY** và **HAVING**

```
SELECT s.Name, su.Name as [Subject], SUM((Score * Weight/100)) as [Total]
FROM
    Subject su INNER JOIN Assessment a ON su.SubjectID = a.SubjectID
    INNER JOIN Students_Assessment sa ON a.AssessmentID =
        sa.AssessmentID
    INNER JOIN Students s ON s.StudentID = sa.StudentID
GROUP BY s.Name, su.Name
HAVING SUM((Score * Weight/100)) > 6
```

	Name	Subject	Total
1	Bùi Mạnh Tuấn	CSD201	6,15
2	Chu Hoàng Lan Anh	CSD201	6,45
3	Hoàng Thế Anh	CSD201	6,1
4	Nguyễn Đức Thắng	CSD201	8,8
5	Nguyễn Hải Dương	CSD201	6,4
6	Nguyễn Quang Minh	CSD201	7,05
7	Nguyễn Thị Phương Diệu	CSD201	7,425
8	Nguyễn Việt Thắng	CSD201	6,5
9	Phùng Quý Quốc	CSD201	7,25
10	Vũ Thị Thanh Thảo	CSD201	6,325
11	Bùi Mạnh Tuấn	DBI202	7
12	Chu Hoàng Lan Anh	DBI202	7,025

Sinh viên có điểm trung bình môn > 6

4. Sử dụng **INNER JOIN**

```
SELECT s.Name, sa.Date
FROM Students s INNER JOIN Students_Assessment sa ON s.StudentID =
sa.StudentID
```

	Name	Date
1	Nguyễn Hải Dương	2022-01-02
2	Hoàng Thế Anh	2022-01-02
3	Nguyễn Việt Thắng	2022-01-02
4	Nguyễn Quang Minh	2022-01-02
5	Vũ Thị Thanh Thảo	2022-01-02
6	Bùi Mạnh Tuấn	2022-01-02
7	Chu Hoàng Lan Anh	2022-01-02
8	Phùng Quý Quốc	2022-01-02

Sinh viên đã thi vào những ngày nào

5. Partial matching in the WHERE clause.

```
SELECT *  
FROM Students s  
WHERE s.Name LIKE 'B%'
```

	StudentID	Name	Gender	DOB
1	6	Bùi Mạnh Tuấn	Nam	2002-10-11

sinh viên có tên bắt đầu bằng chữ 'T'

6.

```
SELECT s.Name, sa.Date  
FROM Students s INNER JOIN Students_Assessment sa ON s.StudentID =  
sa.StudentID  
WHERE sa.Date = '2022-01-02'  
GROUP BY s.Name , sa.Date
```

	Name	Date
1	Bùi Mạnh Tuấn	2022-01-02
2	Chu Hoàng Lan Anh	2022-01-02
3	Hoàng Thế Anh	2022-01-02
4	Nguyễn Đức Thắng	2022-01-02
5	Nguyễn Hải Dương	2022-01-02
6	Nguyễn Quang Minh	2022-01-02
7	Nguyễn Thị Phương Diệu	2022-01-02
8	Nguyễn Việt Thắng	2022-01-02
9	Phùng Quý Quốc	2022-01-02
10	Vũ Thị Thanh Thảo	2022-01-02

Sinh viên thi vào 2022-01-02

7. Sử dụng a sub-query in the WHERE clause

```
SELECT s.Name, su.Name  
  
FROM Subject su INNER JOIN Assessment a ON su.SubjectID = a.SubjectID  
INNER JOIN Students_Assessment sa ON  
a.AssessmentID = sa.AssessmentID  
INNER JOIN Students s ON s.StudentID = sa.StudentID  
WHERE Score = 0 or (a.Name = 'Final exam' and Score < 4)  
GROUP BY s.Name, su.Name  
HAVING sum((Score* Weight/100))/7<5
```


	Name	Name
1	Chu Hoàng Lan Anh	OSG202
2	Hoàng Thế Anh	OSG202

Sinh viên trượt môn (1 đầu điểm bằng 0 hoặc trung bình môn < 5 hoặc điểm cuối kỳ < 4)

8.

```
go
create or ALTER trigger trigger1
on Students
after insert
As
declare @count_svnam nvarchar (9)
set @count_svnam = ( select count(Gender) from Students where
Gender='Nam')
if (@count_svnam> 10)
begin
print 'Sinh vien Nam ko duoc vuot qua 10'
rollback tran
end

select * from Students
```

```
Insert into Students values (11, ' Nguyễn Văn Tùng ', 'Nam', '2002-05-04');
Insert into Students values (12, ' Nguyễn Văn Cường', 'Nam', '2002-05-04');
Insert into Students values (13, ' Nguyễn Hoàng Nam ', 'Nam', '2002-05-04');
Insert into Students values (14, ' Nguyễn Tiến Đạt ', 'Nam', '2002-05-04');
```

```
Sinh vien Nam ko duoc vuot qua 10
Msg 3609, Level 16, State 1, Line 57
The transaction ended in the trigger. The batch has been aborted.

Completion time: 2022-07-18T10:43:35.2730403+07:00
```

99 Store Procedure

```
go
CREATE PROCEDURE uspStudentList
AS
BEGIN
    SELECT
        StudentID,
        Name
    FROM
        Students
    ORDER BY
        StudentID;
END;
```

