Database System Manage Mark

STUDENT: NGUYỄN THỊ PHƯƠNG DIỆU

HE161355 – DBI202

LECTURER: NGÔ TÙNG SƠN

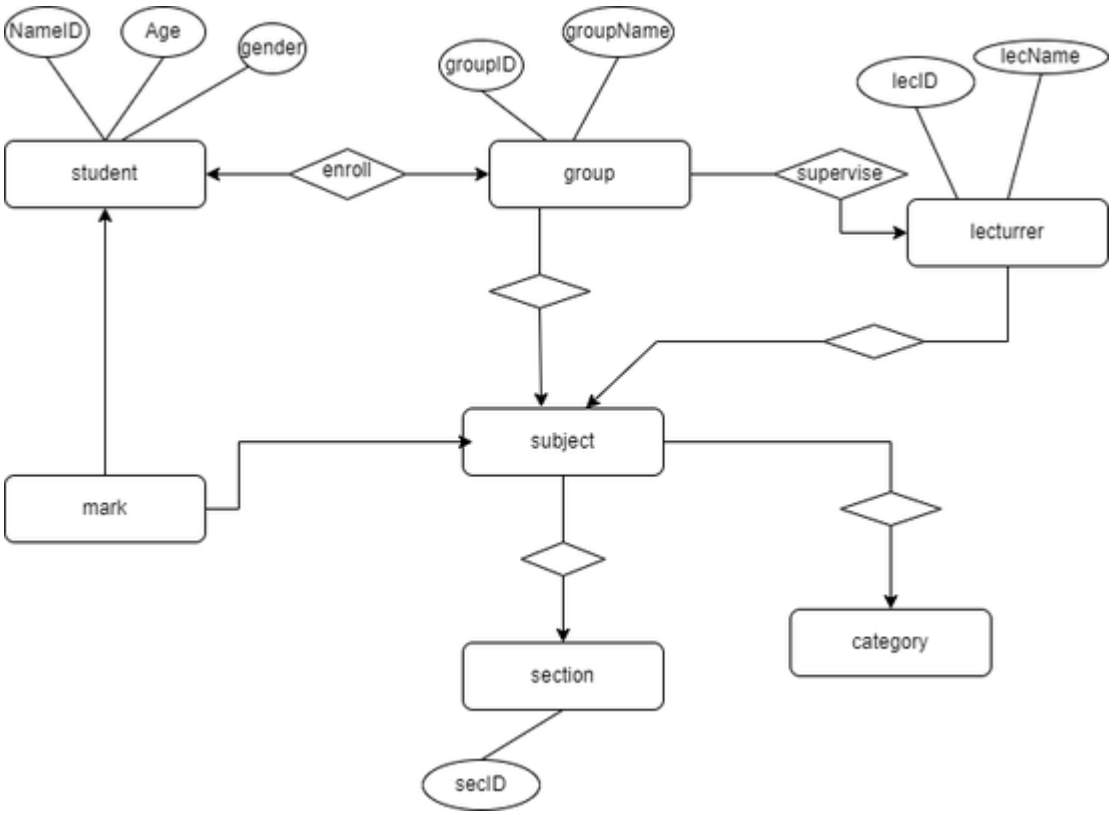
1)Description Database

The database consist of at least 7 tables that have been populated with data: Student, Group, Lecture, Subject, Group_Student, Assessment, Student_Assessment. The Student table include studentID, studentName, gender, dob with primary key is studentID. The group table consist of groupID, groupName, lecID, subID in which groupID is primary key. The lecture table consist of lecID and lecName, primary key is lecID. Subject table include subID and primary key is subID. The Group_Student consist of studentID and groupID in which studentID is primary. Assessment table have 3 attribute which is assID, assName, subID, primary key is subID. The Student_Assessment table have studentID, assID, score with primary key is studentID and assID.

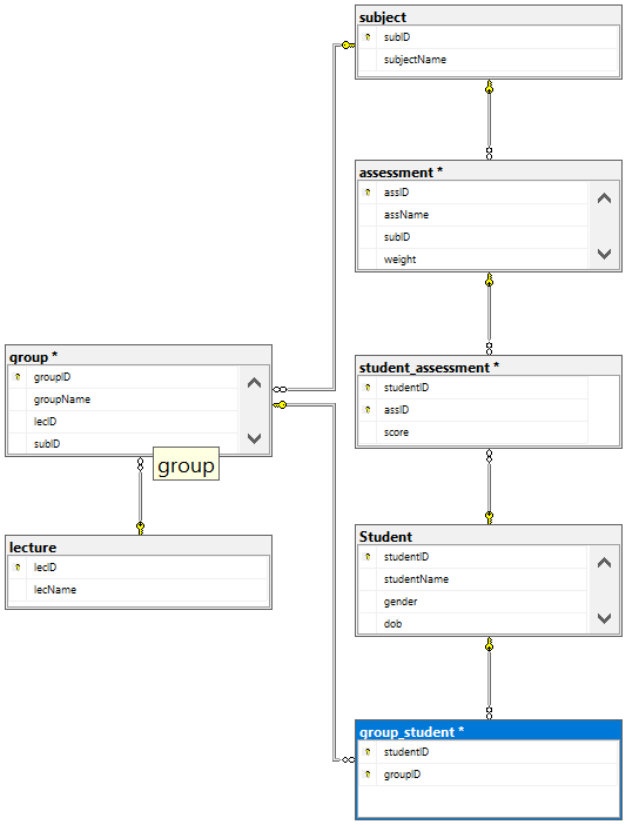
In this database, we can:

- sort students order by StudentID of a class
- Display name and score of student
- Calculate GPA of a student in one semester
- Find student has highest FE score DBI202 of a major in 1 semester
- Caculate GPA of a student.
- Grade students by grades .

2)Relationship Diagram



3)Relational Schema



4)Create Table:

```
create table Student(
        studentID int NOT NULL PRIMARY KEY,
        studentName Nvarchar(50) NOT NULL,
        gender BIT not NULL,
        dob  DATE not null
        )

create TABlE lecture(
        lecID int not null primary key,
        lecName nvarchar(50) not null
        )

create table [subject](
```

```
	subID int NOT NULL PRIMARY KEY,
	subjectName varchar(50) not null
	)

CREATE TABLE [group](
	groupID int NOT NULL PRIMARY KEY,
	groupName varchar(50) not null,
	lecID int not null,
	subID int not null,
	foreign key (lecID) references lecture(lecID),
	foreign key (subID) references [subject](subID)

	)

CREATE TABLE group_student(
	studentID int not null ,
	groupID int not null ,
	PRIMARY KEY(studentID, groupID),
	foreign key (studentID) references Student(studentID),
	foreign key (groupID) references [group](groupID)
)




create table assessment(
	assID int not null primary key,
	assName varchar(50) not null,
	subID int not null,
	[weight] int not null,
```

```
        foreign key (subID) references subject(subID)
        )

create table student_assessment(
        studentID int not null ,
        assID int not null ,
        score float not null ,
        PRIMARY KEY(studentID, assID),
        foreign key ( studentID) references student(studentID),
        foreign key (assID) references assessment(assID)
        )
```

5) Query Statement

```
--- săp xep sinh vien theo studenID
select *from Student order by studentID

-- truy van ra ten va diem cua sinh vien
select s.studentName, sa.score
from Student s Inner join student_assessment sa on s.studentID = sa.studentID

----tinh trung binh diem mon cua moi sinh vien
select sa.studentID,s.studentName, SUM(sa.score * a.[weight]/100) as [AVG]
from assessment a inner join student_assessment sa on a.assID = sa.assID
                                        inner join Student s on sa.studentID = s.studentID
GROUP BY sa.studentID, a.assID, s.studentName
---tinh gpa cua tung sinh vien
select tb1.studentID,tb1.studentName, AVG([AVG]) AS [GPA]
from(select sa.studentID,s.studentName, SUM(sa.score * a.[weight]/100) as [AVG]
from assessment a inner join student_assessment sa on a.assID = sa.assID
                                        inner join Student s on sa.studentID = s.studentID
```

```
GROUP BY sa.studentID, a.assID, s.studentName) tb1

group by tb1.studentID, tb1.studentName

---

----tim sinh vien co diem fe mon dbi cao nhat

SELECT Top 1 s.studentID, s.studentName,su.subjectName, a.assName, sa.Score

FROM Student s        inner join student_assessment sa ON s.studentID = sa.studentID

            inner JOIN subject su on sa.studentID = s.studentID

                                    inner JOIN Assessment a ON a.assID = sa.assID

WHERE a.assName = 'FE' AND su.subjectName = 'DBI202'

ORDER BY sa.Score DESC


---xep loai sinh vien theo score
SELECT  s.StudentName, s.studentID, AVG(score),
   CASE
      WHEN AVG(score) <= 5 THEN 'trung binh'
      WHEN AVG(score) >= 7 AND AVG(Score) < 8 THEN 'kha'
                  ELSE ' gioi'
   END
FROM student_assessment sa INNER JOIN Student s on s.StudentID = sa.studentID
GROUP BY s.studentID, s.studentName
```