**Student Grading Management Sub-System**

Bùi VIỆT ANH

HE160929 – DBI202 Slot 3

LECTURER: NGÔ TÙNG SƠN

1) DESCRIPTION DATABASE

-A student can join many groups and a group can have many students

-A subject can have many groups, but a group can only participate in one subject

-A subject will have multiple assessments and each assessment can only be used for one subject

\*) Things that can be done with this database:

- Display information of student order by StudentId

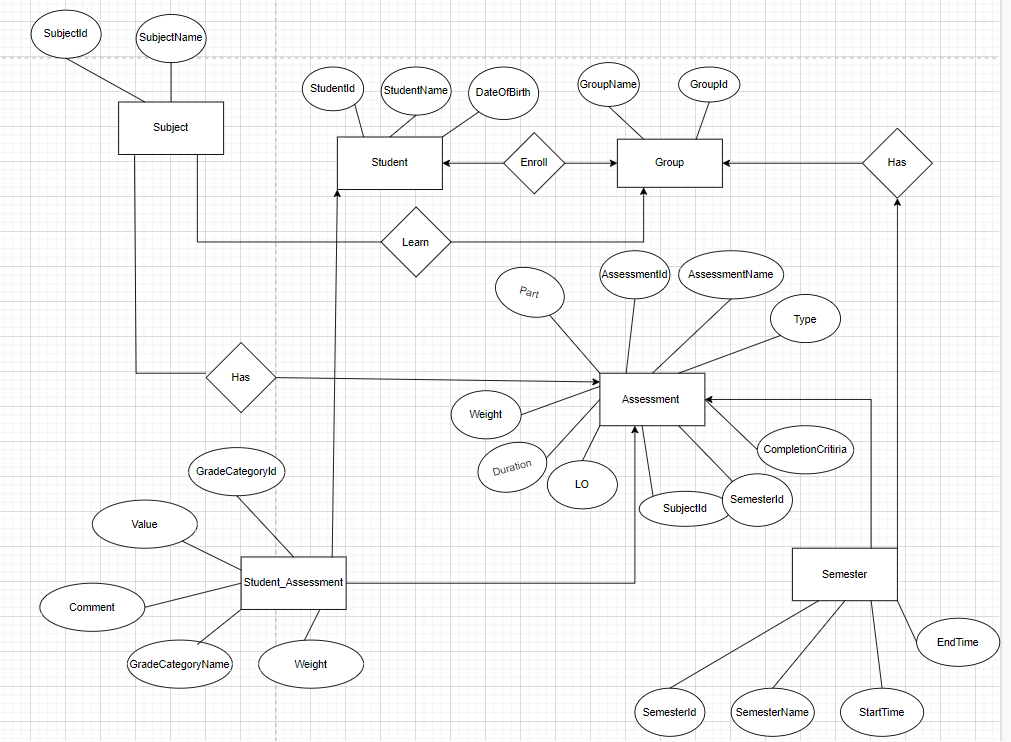
- Display information of student have like assessmentId

-Display test scores in ascending order of a class

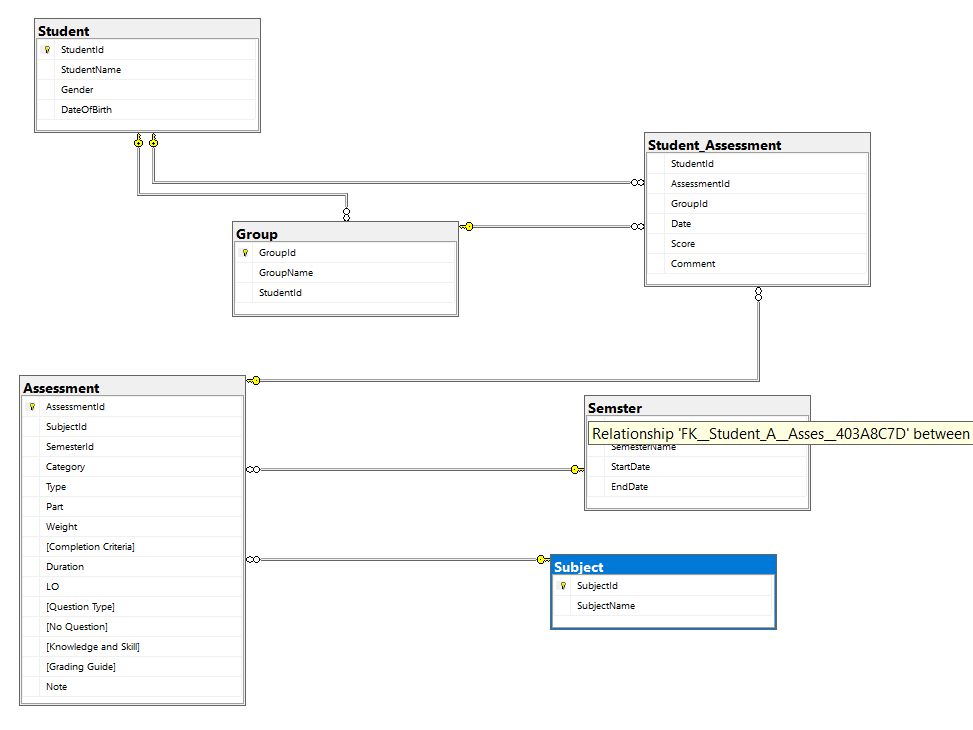
- Display number of score in a class

- Display infomation all score of a student

**2) ENTITY RELATIONSHIP DIAGRAM**

****

**3) RELATIONAL SCHEMA**



**4) THE SET OF DATABASE STATEMENTS USED TO CREATE THE TABLES**

CREATE TABLE Student(

StudentId INT PRIMARY KEY NOT NULL,

StudentName VARCHAR(50) NOT NULL,

Gender bit NOT NULL,

DateOfBirth DATE NOT NULL

)

CREATE TABLE [Subject](

SubjectId INT PRIMARY KEY NOT NULL,

SubjectName VARCHAR(50) NOT NULL

)

CREATE TABLE [Group](

GroupId INT PRIMARY KEY NOT NULL,

GroupName VARCHAR(20) NOT NULL,

StudentId INT,

FOREIGN KEY (StudentId) REFERENCES Student(StudentId),

)

CREATE TABLE Semster(

SemesterId INT PRIMARY KEY NOT NULL,

SemesterName varchar(20) NOT NULL,

StartDate DATE NOT NULL,

EndDate DATE NOT NULL

)

CREATE TABLE Assessment(

AssessmentId INT PRIMARY KEY NOT NULL,

AssessmentName VARCHAR(50),

SubjectId INT NOT NULL,

SemesterId INT NOT NULL,

Category VARCHAR(50) NOT NULL,

[Type] VARCHAR(10) NOT NULL,

Part INT NOT NULL,

[Weight] FLOAT NOT NULL,

[Completion Criteria] VARCHAR(10) NOT NULL,

Duration VARCHAR(15) NOT NULL,

LO INT,

[Question Type] TEXT NOT NULL,

[No Question] INT,

[Knowledge and Skill] VARCHAR(150) NOT NULL,

[Grading Guide] TEXT NOT NULL,

Note VARCHAR(2000),

FOREIGN KEY (SubjectId) REFERENCES Subject (SubjectId),

FOREIGN KEY (SemesterId) REFERENCES Semster (SemesterId),

);

CREATE TABLE Student\_Assessment(

StudentId INT NOT NULL,

AssessmentId INT NOT NULL,

GroupId INT NOT NULL,

[Date] DATE NOT NULL,

Score FLOAT NOT NULL,

Comment TEXT ,

FOREIGN KEY (StudentId) REFERENCES Student (StudentId),

FOREIGN KEY (AssessmentId) REFERENCES Assessment(AssessmentId),

FOREIGN KEY (GroupId) REFERENCES [Group] (GroupId)

);

5) QUERY

- Display information of student order by StudentId

SELECT \* FROM Student

ORDER BY StudentId

- Display information of student have like assessmentId

SELECT \* FROM Student st INNER JOIN Student\_Assessment stm ON st.StudentId = stm.StudentId

WHERE stm.AssessmentId = 1

ORDER BY Score

- Display number of score in a class

SELECT COUNT(StudentId) AS [Number Count], Score FROM Student\_Assessment

WHERE GroupId = 1647

GROUP BY Score

HAVING COUNT(StudentId)<8

- Display infomation number of score in a class

SELECT \* FROM Student\_Assessment Sa INNER JOIN [Group] G ON Sa.GroupId = G.GroupId

WHERE Score IN ( SELECT Score FROM Student\_Assessment WHERE Score >=5 AND GroupName = 'SE1647')

- Display infomation all score of a student

SELECT S.StudentId, S.StudentName, A.Category, A.Weight, G.GroupName, Sa.Score, Sa.Date, Sj.SubjectName FROM [Group] G INNER JOIN Student S ON G.StudentId = S.StudentId

INNER JOIN Student\_Assessment Sa ON S.StudentId = Sa.StudentId

INNER JOIN Assessment A ON Sa.AssessmentId = A.AssessmentId

INNER JOIN Subject Sj ON Sj.SubjectId = A.SubjectId