FPT Education

FPT UNIVERSITY

Student Grading Manage

DBI202 – LECTURER: SONNT5

Lê Việt Anh

HE160318 | IA1604

TABLE OF CONTENTS

INTRODUCE THE ASSIGNMENT

I. DESCRIBE THE ASSIGNMENT

Officially established on September 8, 2006, under the decision of the Prime Minister, FPT University became the first university in Vietnam established by a company with 100% investment capital from the Corporation. FPT. Along with many differences compared to other universities, the University is currently training groups in IT, Economics, Languages, and Applied Arts. Because of the growth and the huge number of students, there is an urgent problem that needs a database to manage student data. Therefore, this assignment is made to strengthen and create a professional management environment for FPT University, as well as acquire and learn more.

II. MANAGEMENT OBJECTIVES

- Student can check their result during and at the end of semester.
- Student can manage their score in each course by seeing assessments in each subject.
- The Lecture can give score to many students in each assessment.
- School can control score of students.

I. DATA ANALYZING

1. Assessment

5 assessment(s)

Category	Type	Part	Weight	Completion Criteria	Duration	CLO	Question Type	No Question	Knowledge and Skill	Grading Guide	Note
Assignment	on-going	1	20.0%	>0	28 slots			N/A	All subjects in syllabus	in class, by teacher	
Lab	on-going	5	10.0%	>0	90'			N/A		in class, by teacher	
Practical Exam	on-going	1	30.0%	>0	90			N/A		In class, by teacher	
Progress test	on-going	2	10.0%	>0	30'		Multiple choices; Marked by Computer or a suitable format	20		in class, by LMS system	Instruction and schedules for Progress Tests must be presented in the Course Implementation Plan approved by director of the campus. Progress test must be taken right after the last lectures of required material. Instructor has responsibility to review the test for students after graded. Progress test must be taken right after the last lectures of required material. Instructor has resposibility to review the test for students after graded.
Final exam	final exam	1	30.0%	4	60			50	All subjects in syllabus	Exam room	The exam questions must be updated or different at least 70% to the previous ones.

This is the component score information table, this table contains the points that throughout a subject in a semester students will be assessed by their Lecture. Thereby determining the Data:

- Category: types of tests to review students.
- Type: Status of Category.
- Part: The number of categories.
- Weight: Percentage of points affects the grade point average in subject.
- Completion Criteria: Condition of points to pass the course.
- Duraion: Time to finish the test.
- Question type: The style of question.
- No Question: The number of questions in one test.
- Knowledge and Skill: The requirement to pass the exam.
- Granding Guilde: Instruction to give score.

- Note: Short explanation of information gived by Lecture.

2. Subjects And Course Of Students

NO.	SUBJECT CODE	SUBJECT NAME	SEMESTER	GROUP	STARTDATE	ENDDATE	AVERAGE MARK	STATUS
1	SSL101c	Academic Skills for University Success	Spring2021					Not Passed
2	SSG103	Communication and In-Group Working Skills	Summer2021					Passed
3	NWC203c	Computer Networking	Summer2021					Passed
4	CEA201	Computer Organization and Architecture	Spring2021					Passed
5	MAD101	Discrete mathematics	Summer2021					Passed
6	JPD113	Elementary Japanese 1-A1.1	Fall2021					Passed
7	CSI104	Introduction to Computer Science	Spring2021					Passed
8	DBI202	Introduction to Databases	Fall2021					Not Passed
9	LUK1	Level 1	Fall2019					Passed
10	LUK2	Level 2	Spring2020					Passed
11	LUK3	Level 3	Spring2020					Passed
12	LUK4	Level 4	Summer2020					Pass (with conditions)
13	LUK5	Level 5	Summer2020					Passed
14	LUK6	Level 6	Fall2020					Passed
15	MAE101	Mathematics for Engineering	Spring2021					Passed
16	GDQP	Military training	Fall2019					Passed
17	PRO192	Object-Oriented Programming	Fall2021					Passed
18	PRO192	Object-Oriented Programming	Fall2021					Not Passed
19	OSG202	Operating Systems	Summer2021					Passed
20	PRF192	Programming Fundamentals	Summer2021					Not Passed
21	PRF192	Programming Fundamentals	Spring2021					Attendance Fail
22	ĐTB102	Traditional musical instrument	Summer2020					Passed
23	VOV114	Vovinam 1	Fall2019					Passed
24	VOV124	Vovinam 2	Summer2020					Passed
25	VOV134	Vovinam 3	Summer2020					Passed

In this table, Students could see the basic information of all cousres that they have been joined in each semester:

- No: Serial
- Subject code: Code of each course.
- Subject name: Name of course.
- Semester: Describe seasons and years.
- Group: Describe the class that student has been joined.
- Start/End Date: The start day and the end day of the course.

- Average Mark: GPA
- Status: Standing (Passed, Not Passed, Passed with conditions)

This table contain many information and could be a lot of attributes. So, this table could not stand alone. Must dived to many tables to avoid 3NF.

3. Student Transcripts

GRADE CATEGORY	GRADE ITEM	WEIGHT	VALUE	COMMENT
Quiz 2	Quiz 2	7.0 %	7.8	
	Total	7.0 %	7.8	
Quiz 1	Quiz 1	8.0 %	7.6	
	Total	8.0 %	7.6	
Activity	Activity	10.0 %	8.5	
	Total	10.0 %	8.5	
Group Assignment	Group Assignment	15.0 %	9	
	Total	15.0 %	9	
Group Project	Group Project	30.0 %	8.3	
	Total	30.0 %	8.3	
Final Exam	Final Exam	30.0 %	8.6	
	Total	30.0 %	8.6	
Final Exam Resit	Final Exam Resit	30.0 %		
	Total	30.0 %		
COURSE TOTAL	**AVERAGE**	**8.4**		
	STATUS	**PASSED**		

In this table, there are many categories that students have taken during and after the course. This student transcripts contains:

- Grade category: Score students have in courses.
- Grade Item: types of tests to review students in courses.
- Weight: Percentage of points affects the grade point average in courses.
- Value: Score stutens get and gived by Lecture.
- Comment: Note of the lecture.

II. IDENTIFY ENTITIES AND ATTRIBUTES

1. Student

Because this database mainly related to students and **Students** entity has many relationships, so I decided to create **Students** entity.

Attribute	Type	Format	Requires	Primary Key
StudentID	int		NOT NULL	*
StudentCode	nvarchar	50	NOT NULL	
Name	nvarchar	50	NOT NULL	
Gender	bit		NULL	
Dob	date		NULL	
Email	nvarchar	50	NULL	

2. Group

Student will have to join to a group to learn courses. Also there are many groups of student, so we'll have **Group** entity.

Attribute	Type	Format	Requires	Primary Key
GroupID	Int		NOT NULL	*
GroupCode	Int		NOT NULL	

3. Subject

A grade management system is certainly indispensable for these subjects. Students will have to go through the course of the subjects from which to evaluate the assessment.

Attribute	Type	Format	Requires	Primary Key
SubjectID	Int		NOT NULL	*
SubjectCode	Nchar	10	NULL	
SubjectName	Nvarchar	50	NULL	

4. Teacher

Teacher will be the person to teach many students and take grade for them. Because teacher will be many relationships so there will be **Teacher** entity.

Attribute	Type	Format	Requires	Primary Key
TeacherID	int		NOT NULL	*
Name	nvarchar	50	NOT NULL	
Gender	bit		NULL	
Dob	Date		NULL	
Email	nvarchar	50	NULL	

5. Semester

Schools set up semesters by year to easily manage students as well as grades and related issues.

Attribute	Type	Format	Requires	Primary Key
SemesterID	Int		NOT NULL	*
SemesterCode	Nchar	15	NULL	

6. Assessment

After identifying all the necessary entity, we going to create **Assessment** entity. This entity contains all grades, assessments are included in a subject and will be assessed by the teacher for each student.

Attribute	Type	Format	Requires	Primary Key
Assid	Int		NOT NULL	*
Name	Nchar	10	NOT NULL	
TypeID	Nvarchar	50	NULL	
Part	Int		NULL	
Weight	Float		NOT NULL	
Completion Criteria	Int		NULL	

Duration	nvarchar	50	NULL	
NoQuestion	Int		NULL	

7. Mark

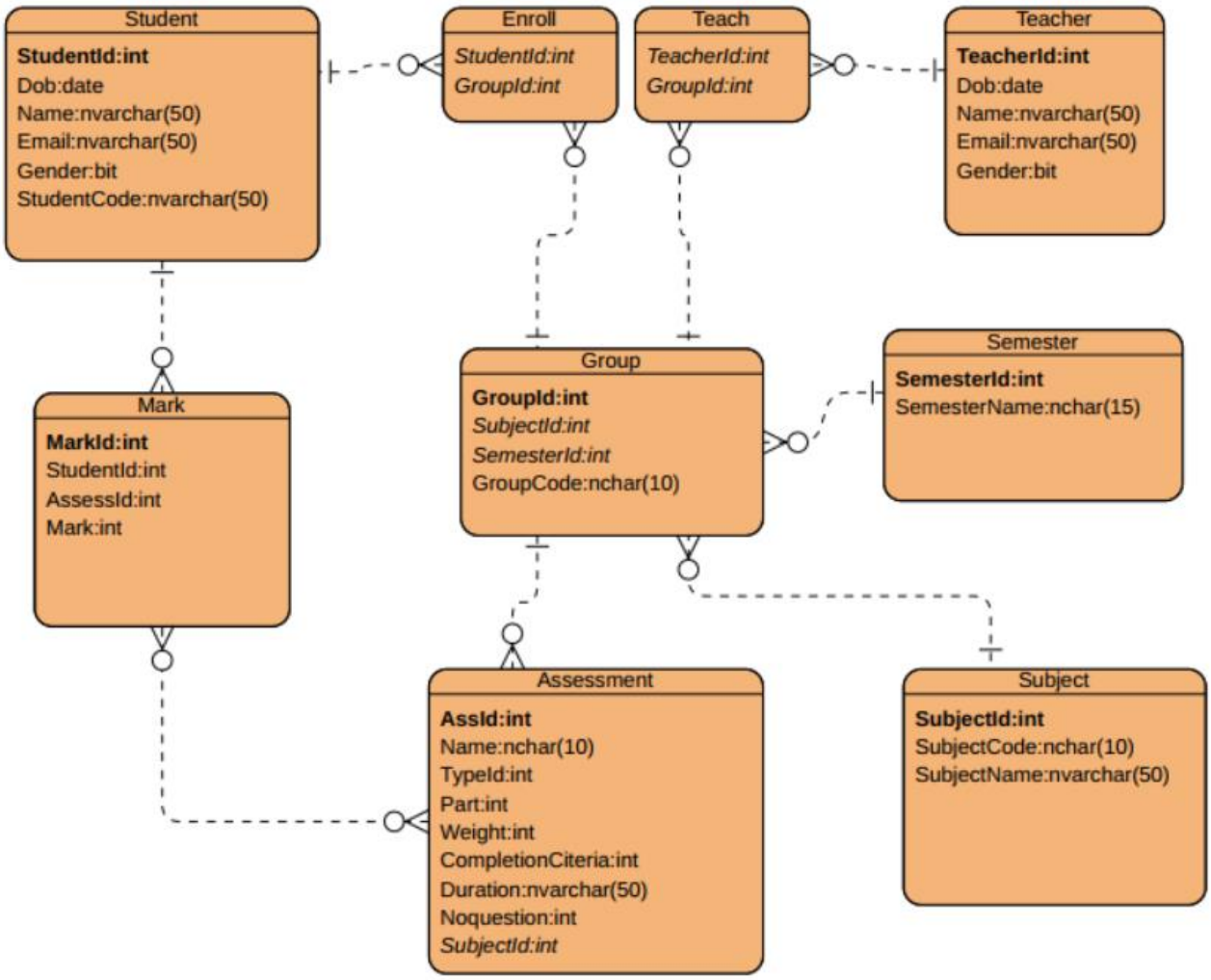
Attribute	Type	Format	Requires	Primary Key
MarkID	Int		NOT NULL	*
Mark	Int		NULL	

III. IDENTIFY RELATIONSHIPS

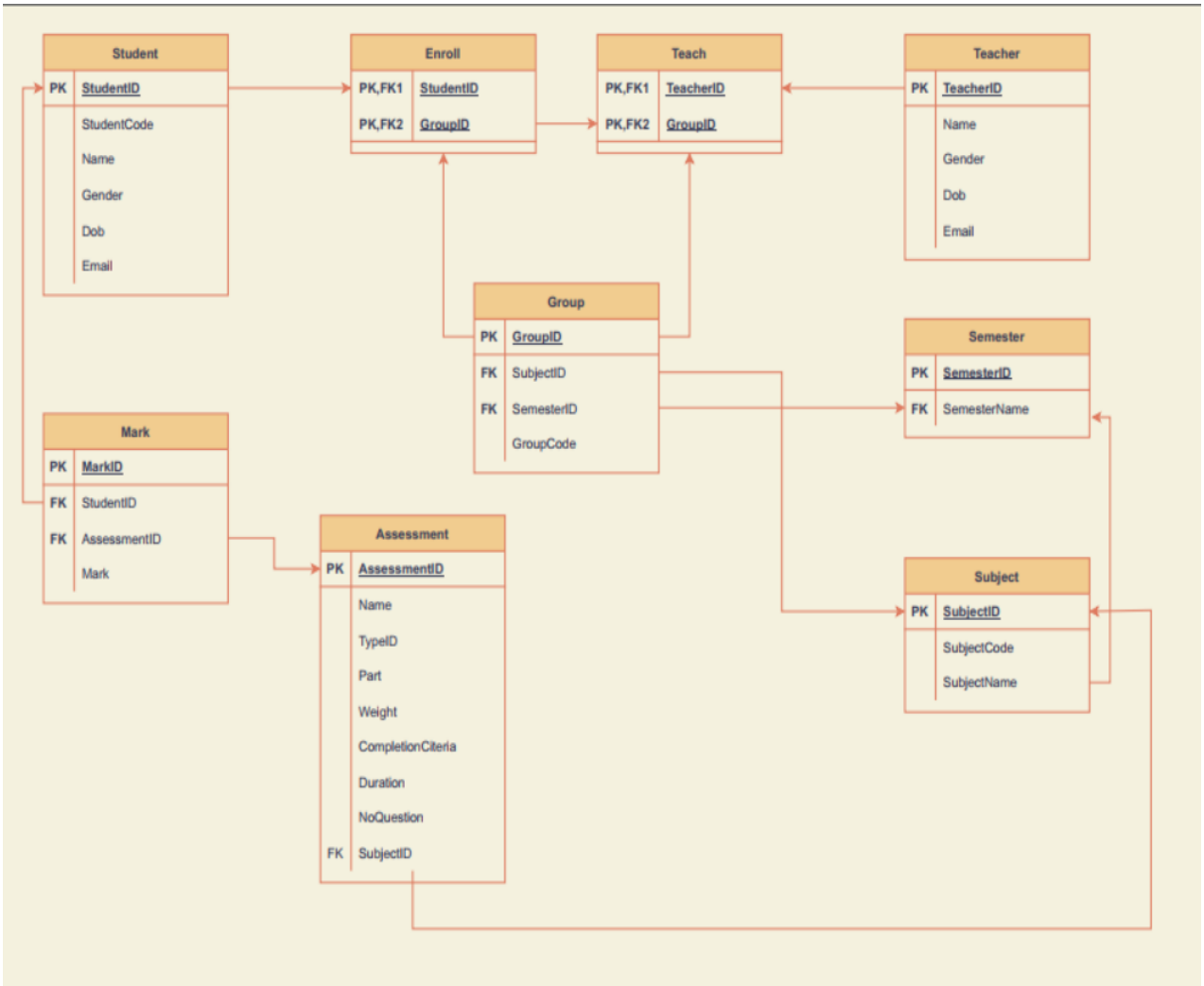
- For **Student** vs **Group**, students will join many different groups based on semester, study, and significance. And each class will also be attended by many students. Therefore, it is possible to define the relationship between these two entities as **N-N**.
- For **Student** vs **Mark**, for each student, there will be many marks that the student has obtained over the term and based on the number of subjects. And vice versa, many points of the subject will be assigned to students. So we will give these 2 entities a 1-N relationship.
- For **Teacher** vs **Group**, teacher is responsible for teaching a group of students. But in a school, there will be many such groups of students, so teachers will have to manage many groups. That's why we'll set these 2 entities a N-N relationship.
- For **Semester** vs **Group**, each semester will have many groups of student join. So, realtionship between **Semester** vs **Group** is 1-N.
- For **Assessment** vs **Subject**, assessment includes many conditions for assessing students in a subject. It is easy to see that a subject has only one assessment so we will put the condition 1-1 here.
- For **Subject** vs **Group**, each group will learn many subjects and one subject is studied by many groups, we will establish a 1-N relationship.
- For **Mark** vs **Assessment**, because mark are the values that will evaluate students according to each assessment. But each assessment has many different marks for students according to different levels and types. So, we will establish a 1-N relationship.

IV. ENTITY RELATIONSHIP DIAGRAM

After identifying entities, attributes and their relationships, we could summary them to the ERD below



V. RELATIONAL DIAGRAM



VI. CREATE DATABASE AND TABLES

1. Create database

```
CREATE DATABASE [Assignment_DBI]
```

2. Create **Assessment** table

```
CREATE TABLE [dbo].[Assessment](
    [AssId] [int] IDENTITY(1,1) NOT NULL PRIMARY
KEY,
    [Name] [nchar](15) NOT NULL,
    [Type] [nvarchar](15) NULL,
    [Part] [int] NOT NULL,
    [Weight] [float] NOT NULL,
    [CompletionCiteria] [int] NULL,
    [Duration] [nvarchar](50) NULL,
    [Noquestion] [int] NULL,
    [SubjectId] [int] NULL,
    FOREIGN KEY(SubjectId) REFERENCES
Student(SubjectId)
)
```

3. Create **Group** table

```
CREATE TABLE [dbo].[GroupS](
	[GroupId] [int] IDENTITY(1,1) NOT NULL PRIMARY
  KEY,
	[SubjectId] [int] NOT NULL,
	[SemesterId] [int] NOT NULL,
	[GroupCode] [nchar](10) NULL,
	FOREIGN KEY (SemesterId) REFERENCES
  Semester(SemesterId),
	FOREIGN KEY (SubjectId) REFERENCES
  Subject(SubjectId)
)
```

4. Create **Mark** table

```
CREATE TABLE [dbo].[Mark](
    [MarkId] [int] IDENTITY(1,1) NOT NULL PRIMARY
KEY,
    [StudentId] [int] NULL,
    [AsssessId] [int] NULL,
    [Mark] [int] NULL,
    FOREIGN KEY (StudentId) REFERENCES
Student(StudentId),
```

```
    FOREIGN KEY (AsssessId) REFERENCES
Assessment(AsssessId)
)
```

5. Create **Semester** table

```
CREATE TABLE [dbo].[Semester](
    [SemesterId] [int] IDENTITY(1,1) NOT NULL
PRIMARY KEY,
    [SemesterName] [nchar](15) NULL
)
```

6. Create **Student** table

```
CREATE TABLE [dbo].[Student](
    [StudentId] [int] IDENTITY(1,1) NOT NULL
PRIMARY KEY,
    [Dob] [date] NULL,
    [Name] [nvarchar](50) NOT NULL
    [Email] [nvarchar](50) NULL,
    [Gender] [bit] NULL,
    [StudentCode] [nvarchar](50) NOT NULL
)
```

7. Create **Subject** table

```
CREATE TABLE [dbo].[Subject](
    [SubjectId] [int] IDENTITY(1,1) NOT NULL
PRIMARY KEY,
    [SubjectCode] [nchar](10) NULL,
    [SubjectName] [nvarchar](50) NULL
)
```

8. Create **Teacher** table

```
CREATE TABLE [dbo].[Teacher](
    [TeacherId] [int] IDENTITY(1,1) NOT NULL
PRIMARY KEY,
    [Dob] [date] NULL,
    [Name] [nvarchar](50) NOT NULL,
    [Email] [nvarchar](50) NOT NULL,
    [Gender] [bit] NULL
)
```

9. Create **Teach** table

```
CREATE TABLE [dbo].[Teach](
    [TeacherId] [int] NOT NULL,
    [GroupId] [int] NULL,
    PRIMARY KEY (TeacherId, GroupId),
```

```
    FOREIGN KEY (TeacherId) REFERENCES
Teacher(TeacherId),
    FOREIGN KEY (GroupId) REFERENCES
[Group](GroupId)
)
```

10. Create **Enroll** table

```
CREATE TABLE [dbo].[Enroll](
    [StudentId] [int] NOT NULL,
    [GroupId] [int] NOT NULL,
    PRIMARY KEY (StudentId, GroupId),
    FOREIGN KEY (StudentId) REFERENCES
Student(StudentId),
    FOREIGN KEY (GroupId) REFERENCES
[Group](GroupId)
)
```

VII. INSERT DATA

1. Assessment

```
INSERT [dbo].[Assessment] ([AssId], [Name], [Type], [Part],
[Weight], [CompletionCiteria], [Duration], [Noquestion],
[SubjectId])
 VALUES (1, N'Progess Test', N'quiz', 2, 5, 0, N'20m', 20, 1)
INSERT [dbo].[Assessment] ([AssId], [Name], [Type], [Part],
[Weight], [CompletionCiteria], [Duration], [Noquestion],
[SubjectId])
 VALUES (2, N'Assignment     ', N'on-going', 1, 20, 0, N'at home',
0, 1)
INSERT [dbo].[Assessment] ([AssId], [Name], [Type], [Part],
[Weight], [CompletionCiteria], [Duration], [Noquestion],
[SubjectId])
 VALUES (3, N'Labs           ', N'on-going', 5, 3, 0, N'in lab
session', 0, 1)
INSERT [dbo].[Assessment] ([AssId], [Name], [Type], [Part],
[Weight], [CompletionCiteria], [Duration], [Noquestion],
[SubjectId])
 VALUES (4, N'Practical Exam ', N'practical exam', 1, 25, 0,
N'85m', 0, 1)
INSERT [dbo].[Assessment] ([AssId], [Name], [Type], [Part],
[Weight], [CompletionCiteria], [Duration], [Noquestion],
[SubjectId])
 VALUES (5, N'Final Exam     ', N'final exam', 1, 30, 5, N'in lab
session', 60, 1)
INSERT [dbo].[Assessment] ([AssId], [Name], [Type], [Part],
[Weight], [CompletionCiteria], [Duration], [Noquestion],
[SubjectId])
VALUES (6, N'Progess Test   ', N'quiz', 2, 10, 0, N'20m', 50, 3)
```

```
INSERT [dbo].[Assessment] ([AssId], [Name], [Type], [Part],
[Weight], [CompletionCiteria], [Duration], [Noquestion],
[SubjectId])
 VALUES (7, N'Mid-term       ', N'Mid-term', 1, 30, 0, N'60m', 50,
3)
INSERT [dbo].[Assessment] ([AssId], [Name], [Type], [Part],
[Weight], [CompletionCiteria], [Duration], [Noquestion],
[SubjectId])
 VALUES (8, N'Participation  ', N'Participation', 1, 10, 0,
N'along semester', 0, 3)
INSERT [dbo].[Assessment] ([AssId], [Name], [Type], [Part],
[Weight], [CompletionCiteria], [Duration], [Noquestion],
[SubjectId])
 VALUES (9, N'Final Exam     ', N'final exam', 1, 40, 5,
N'multiple choices', 60, 3)
INSERT [dbo].[Assessment] ([AssId], [Name], [Type], [Part],
[Weight], [CompletionCiteria], [Duration], [Noquestion],
[SubjectId])
 VALUES (10, N'Progess Test   ', N'quiz', 2, 5, 0, N'20m', 20, 2)
INSERT [dbo].[Assessment] ([AssId], [Name], [Type], [Part],
[Weight], [CompletionCiteria], [Duration], [Noquestion],
[SubjectId])
 VALUES (11, N'Assignment     ', N'on-going', 1, 20, 0, N'at
home', 0, 2)
INSERT [dbo].[Assessment] ([AssId], [Name], [Type], [Part],
[Weight], [CompletionCiteria], [Duration], [Noquestion],
[SubjectId])
 VALUES (12, N'Labs           ', N'on-going', 5, 3, 0, N'in lab
session', 0, 2)
INSERT [dbo].[Assessment] ([AssId], [Name], [Type], [Part],
[Weight], [CompletionCiteria], [Duration], [Noquestion],
[SubjectId])
 VALUES (13, N'Practical Exam ', N'practical exam', 1, 25, 0,
N'85m', 0, 2)
```

```
INSERT [dbo].[Assessment] ([AssId], [Name], [Type], [Part],
[Weight], [CompletionCiteria], [Duration], [Noquestion],
[SubjectId])
 VALUES (14, N'Final Exam     ', N'final exam', 1, 30, 5,
N'Multiple choices', 60, 2)
INSERT [dbo].[Assessment] ([AssId], [Name], [Type], [Part],
[Weight], [CompletionCiteria], [Duration], [Noquestion],
[SubjectId])
 VALUES (15, N'Progess Test   ', N'quiz', 2, 5, 0, N'20m', 20, 4)
INSERT [dbo].[Assessment] ([AssId], [Name], [Type], [Part],
[Weight], [CompletionCiteria], [Duration], [Noquestion],
[SubjectId])
 VALUES (16, N'Assignment     ', N'on-going', 1, 20, 0, N'at
home', 0, 4)
INSERT [dbo].[Assessment] ([AssId], [Name], [Type], [Part],
[Weight], [CompletionCiteria], [Duration], [Noquestion],
[SubjectId])
 VALUES (17, N'Labs           ', N'on-going', 5, 3, 0, N'in lab
session', 0, 4)
INSERT [dbo].[Assessment] ([AssId], [Name], [Type], [Part],
[Weight], [CompletionCiteria], [Duration], [Noquestion],
[SubjectId])
 VALUES (18, N'Practical Exam ', N'practical exam', 1, 25, 0,
N'85m', 0, 4)
INSERT [dbo].[Assessment] ([AssId], [Name], [Type], [Part],
[Weight], [CompletionCiteria], [Duration], [Noquestion],
[SubjectId])
 VALUES (19, N'Final Exam     ', N'final exam', 1, 30, 5,
N'Multiple choices', 60, 4)
```

2. Enroll

```
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (1, 1)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (2, 1)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (3, 1)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (4, 1)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (5, 1)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (6, 1)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (7, 1)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (8, 1)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (9, 1)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (10, 1)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (11, 2)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (12, 2)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (13, 2)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (4, 2)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (5, 2)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (6, 2)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (7, 2)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (8, 2)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (9, 2)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (10, 2)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (15, 3)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (16, 3)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (11, 3)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (12, 3)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (13, 3)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (14, 3)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (7, 3)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (8, 3)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (9, 3)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (10, 3)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (15, 4)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (16, 4)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (11, 4)
```

```
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (12, 4)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (13, 4)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (14, 4)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (17, 4)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (18, 4)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (9, 4)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (5, 4)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (17, 5)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (18, 5)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (15, 5)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (16, 5)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (1, 5)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (2, 5)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (3, 5)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (4, 5)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (9, 5)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (5, 5)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (17, 6)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (18, 6)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (15, 6)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (16, 6)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (13, 6)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (12, 6)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (11, 6)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (4, 6)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (9, 6)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (5, 6)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (17, 7)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (18, 7)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (15, 7)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (16, 7)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (13, 7)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (12, 7)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (11, 7)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (14, 7)
```

```
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (10, 7)
INSERT [dbo].[Enroll] ([StudentId], [GroupId]) VALUES (5, 7)
```

3. Group

```
INSERT [dbo].[GroupS] ([GroupId], [SubjectId], [SemesterId],
[GroupCode]) VALUES (1, 1, 2, N'SE1607')
INSERT [dbo].[GroupS] ([GroupId], [SubjectId], [SemesterId],
[GroupCode]) VALUES (2, 2, 3, N'SE1506')
INSERT [dbo].[GroupS] ([GroupId], [SubjectId], [SemesterId],
[GroupCode]) VALUES (3, 3, 4, N'IOT1513')
INSERT [dbo].[GroupS] ([GroupId], [SubjectId], [SemesterId],
[GroupCode]) VALUES (4, 4, 5, N'SE1506')
INSERT [dbo].[GroupS] ([GroupId], [SubjectId], [SemesterId],
[GroupCode]) VALUES (5, 5, 6, N'SE1617')
INSERT [dbo].[GroupS] ([GroupId], [SubjectId], [SemesterId],
[GroupCode]) VALUES (6, 6, 7, N'JP1513')
INSERT [dbo].[GroupS] ([GroupId], [SubjectId], [SemesterId],
[GroupCode]) VALUES (7, 7, 8, N'AI1603')
INSERT [dbo].[GroupS] ([GroupId], [SubjectId], [SemesterId],
[GroupCode]) VALUES (8, 8, 9, N'AI1605')
INSERT [dbo].[GroupS] ([GroupId], [SubjectId], [SemesterId],
[GroupCode]) VALUES (9, 9, 10, N'HS1515')
INSERT [dbo].[GroupS] ([GroupId], [SubjectId], [SemesterId],
[GroupCode]) VALUES (10, 10, 2, N'HS1535')
```

4. Mark

```
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (1, 1, 1, 5)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (2, 1, 1, 7)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (3, 1, 2, 6)
```

```
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (4, 1, 3, 5)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (5, 1, 3, 6)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (6, 1, 3, 7)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (7, 1, 3, 4)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (8, 1, 3, 8)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (9, 1, 4, 8)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (10, 1, 5, 6)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (11, 2, 1, 7)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (12, 2, 1, 8)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (13, 2, 2, 9)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (14, 2, 3, 9)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (15, 2, 3, 8)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (16, 2, 3, 7)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (17, 2, 3, 7)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (18, 2, 3, 6)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (19, 2, 4, 5)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (20, 2, 5, 7)
```

```
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (21, 3, 1, 6)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (22, 3, 1, 6)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (23, 3, 2, 6)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (24, 3, 3, 9)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (25, 3, 3, 5)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (26, 3, 3, 7)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (27, 3, 3, 8)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (28, 3, 3, 6)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (29, 3, 4, 5)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (30, 3, 5, 9)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (31, 4, 1, 7)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (32, 4, 1, 7)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (33, 4, 2, 7)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (34, 4, 3, 7)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (35, 4, 3, 8)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (36, 4, 3, 8)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (37, 4, 3, 8)
```

```
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (38, 4, 3, 5)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (39, 4, 4, 7)
INSERT [dbo].[Mark] ([MarkId], [StudentId], [AsssessId], [Mark])
VALUES (40, 4, 5, 9)
```

5. Semester

```
INSERT [dbo].[Semester] ([SemesterId], [SemesterName]) VALUES (1,
N'Summer2019')
INSERT [dbo].[Semester] ([SemesterId], [SemesterName]) VALUES (2,
N'Summer2020')
INSERT [dbo].[Semester] ([SemesterId], [SemesterName]) VALUES (3,
N'Summer2021')
INSERT [dbo].[Semester] ([SemesterId], [SemesterName]) VALUES (4,
N'Summer2022')
INSERT [dbo].[Semester] ([SemesterId], [SemesterName]) VALUES (5,
N'Spring2019')
INSERT [dbo].[Semester] ([SemesterId], [SemesterName]) VALUES (6,
N'Spring2020')
INSERT [dbo].[Semester] ([SemesterId], [SemesterName]) VALUES (7,
N'Spring2021')
INSERT [dbo].[Semester] ([SemesterId], [SemesterName]) VALUES (8,
N'Spring2022')
INSERT [dbo].[Semester] ([SemesterId], [SemesterName]) VALUES (9,
N'Fall2019')
INSERT [dbo].[Semester] ([SemesterId], [SemesterName]) VALUES (10,
N'Fall2020')
INSERT [dbo].[Semester] ([SemesterId], [SemesterName]) VALUES (11,
N'Fall2021')
```

6. Student

```
INSERT [dbo].[Student] ([StudentId], [Dob], [Name], [Email],
[Gender], [StudentCode]) VALUES (1, CAST(N'2001-05-06' AS Date),
N'Le Quang Manh', N'manh@gmail.com', 1, N'HE163617')
INSERT [dbo].[Student] ([StudentId], [Dob], [Name], [Email],
[Gender], [StudentCode]) VALUES (2, CAST(N'2002-07-06' AS Date),
N'Le Hong Quan', N'quan@gmail.com', 1, N'HE153617')
INSERT [dbo].[Student] ([StudentId], [Dob], [Name], [Email],
[Gender], [StudentCode]) VALUES (3, CAST(N'2002-07-06' AS Date),
N'Doan Hai Manh', N'manh@gmail.com', 1, N'HE143617')
INSERT [dbo].[Student] ([StudentId], [Dob], [Name], [Email],
[Gender], [StudentCode]) VALUES (4, CAST(N'2003-05-08' AS Date),
N'Trinh Quang Thang', N'manh@gmail.com', 1, N'HE173617')
INSERT [dbo].[Student] ([StudentId], [Dob], [Name], [Email],
[Gender], [StudentCode]) VALUES (5, CAST(N'2002-10-06' AS Date),
N'Dinh Tien Manh', N'manh@gmail.com', 1, N'HE164619')
INSERT [dbo].[Student] ([StudentId], [Dob], [Name], [Email],
[Gender], [StudentCode]) VALUES (6, CAST(N'2002-04-09' AS Date),
N'Le Quang Trung', N'manh@gmail.com', 1, N'HE123613')
INSERT [dbo].[Student] ([StudentId], [Dob], [Name], [Email],
[Gender], [StudentCode]) VALUES (7, CAST(N'2001-05-06' AS Date),
N' Quang Manh', N'manh@gmail.com', 1, N'HE163617')
INSERT [dbo].[Student] ([StudentId], [Dob], [Name], [Email],
[Gender], [StudentCode]) VALUES (8, CAST(N'2002-03-03' AS Date),
N'Le Hong ', N'quan@gmail.com', 1, N'HE153635')
INSERT [dbo].[Student] ([StudentId], [Dob], [Name], [Email],
[Gender], [StudentCode]) VALUES (9, CAST(N'2002-07-07' AS Date),
N'Doan Hai ', N'hai@gmail.com', 1, N'HE143435')
INSERT [dbo].[Student] ([StudentId], [Dob], [Name], [Email],
[Gender], [StudentCode]) VALUES (10, CAST(N'2003-08-08' AS Date),
N'Trinh  Thang', N'Thang@gmail.com', 1, N'HE171234')
INSERT [dbo].[Student] ([StudentId], [Dob], [Name], [Email],
[Gender], [StudentCode]) VALUES (11, CAST(N'2002-10-10' AS Date),
N'Dinh  Manh', N'Manh@gmail.com', 1, N'HE164456')
```

```
INSERT [dbo].[Student] ([StudentId], [Dob], [Name], [Email],
[Gender], [StudentCode]) VALUES (12, CAST(N'2002-09-09' AS Date),
N'Le  Trung', N'Trung@gmail.com', 1, N'HE145632')
INSERT [dbo].[Student] ([StudentId], [Dob], [Name], [Email],
[Gender], [StudentCode]) VALUES (13, CAST(N'2001-01-06' AS Date),
N' Quang Manh Hang', N'Hang@gmail.com', 0, N'HE153617')
INSERT [dbo].[Student] ([StudentId], [Dob], [Name], [Email],
[Gender], [StudentCode]) VALUES (14, CAST(N'2001-03-03' AS Date),
N'Le Hong Trung', N'Trung@gmail.com', 0, N'HE153638')
INSERT [dbo].[Student] ([StudentId], [Dob], [Name], [Email],
[Gender], [StudentCode]) VALUES (15, CAST(N'2003-07-07' AS Date),
N'Doan Hai An', N'An@gmail.com', 0, N'HE143439')
INSERT [dbo].[Student] ([StudentId], [Dob], [Name], [Email],
[Gender], [StudentCode]) VALUES (16, CAST(N'2002-08-08' AS Date),
N'Trinh Thang Nhe', N'Nhe@gmail.com', 0, N'HE171238')
INSERT [dbo].[Student] ([StudentId], [Dob], [Name], [Email],
[Gender], [StudentCode]) VALUES (17, CAST(N'2001-10-10' AS Date),
N'Dinh  Manh Phong', N'Phong@gmail.com', 0, N'HE164457')
INSERT [dbo].[Student] ([StudentId], [Dob], [Name], [Email],
[Gender], [StudentCode]) VALUES (18, CAST(N'2002-03-09' AS Date),
N'Le  Trung Hung', N'Hung@gmail.com', 0, N'HE145633')
```

7. Subject

```
INSERT [dbo].[Subject] ([SubjectId], [SubjectCode], [SubjectName])
VALUES (1, N'WED201c', N'Web Design')
INSERT [dbo].[Subject] ([SubjectId], [SubjectCode], [SubjectName])
VALUES (2, N'2DP49', N'Digital project 2D')
INSERT [dbo].[Subject] ([SubjectId], [SubjectCode], [SubjectName])
VALUES (3, N'JPD113', N'Elementary Japanese 1-A1.1')
INSERT [dbo].[Subject] ([SubjectId], [SubjectCode], [SubjectName])
VALUES (4, N'IOT102', N'Internet of Things')
INSERT [dbo].[Subject] ([SubjectId], [SubjectCode], [SubjectName])
VALUES (5, N'ACC101', N'Principles of Accounting ')
```

```
INSERT [dbo].[Subject] ([SubjectId], [SubjectCode], [SubjectName])
VALUES (6, N'SWE201c', N'Introduction to Software Engineering')
INSERT [dbo].[Subject] ([SubjectId], [SubjectCode], [SubjectName])
VALUES (7, N'SWP391', N'Application development project')
INSERT [dbo].[Subject] ([SubjectId], [SubjectCode], [SubjectName])
VALUES (8, N'CSD201', N'Data Structures and Algorithms ')
INSERT [dbo].[Subject] ([SubjectId], [SubjectCode], [SubjectName])
VALUES (9, N'PRE201c', N'Excel Skills for Business ')
INSERT [dbo].[Subject] ([SubjectId], [SubjectCode], [SubjectName])
VALUES (10, N'SWT301', N'Software Testing')
INSERT [dbo].[Subject] ([SubjectId], [SubjectCode], [SubjectName])
VALUES (11, N'JPD123', N'Elementary Japanese 1-A1.2')
```

8. Teacher

```
INSERT [dbo].[Teacher] ([TeacherId], [Dob], [Name], [Email],
[Gender]) VALUES (1, CAST(N'1990-04-03' AS Date), N'Haint',
N'hai@gmail.com', 0)
INSERT [dbo].[Teacher] ([TeacherId], [Dob], [Name], [Email],
[Gender]) VALUES (2, CAST(N'1985-05-03' AS Date), N'Trunglt',
N'trung@gmail.com', 1)
INSERT [dbo].[Teacher] ([TeacherId], [Dob], [Name], [Email],
[Gender]) VALUES (3, CAST(N'1980-08-03' AS Date), N'Dangva',
N'dang@gmail.com', 1)
INSERT [dbo].[Teacher] ([TeacherId], [Dob], [Name], [Email],
[Gender]) VALUES (4, CAST(N'1980-08-05' AS Date), N'Chilt',
N'chi@gmail.com', 0)
```

9. Teach

```
INSERT [dbo].[Teach] ([TeacherId], [GroupId]) VALUES (1, 1)
INSERT [dbo].[Teach] ([TeacherId], [GroupId]) VALUES (2, 2)
```

```
INSERT [dbo].[Teach] ([TeacherId], [GroupId]) VALUES (3, 3)
INSERT [dbo].[Teach] ([TeacherId], [GroupId]) VALUES (4, 4)
INSERT [dbo].[Teach] ([TeacherId], [GroupId]) VALUES (1, 5)
INSERT [dbo].[Teach] ([TeacherId], [GroupId]) VALUES (2, 6)
INSERT [dbo].[Teach] ([TeacherId], [GroupId]) VALUES (3, 7)
INSERT [dbo].[Teach] ([TeacherId], [GroupId]) VALUES (1, 2)
INSERT [dbo].[Teach] ([TeacherId], [GroupId]) VALUES (4, 8)
INSERT [dbo].[Teach] ([TeacherId], [GroupId]) VALUES (1, 9)
INSERT [dbo].[Teach] ([TeacherId], [GroupId]) VALUES (2, 10)
```

VIII. QUERIES

1. Query 1

```
select s.*, g.GroupCode from Student s, Enroll e, GroupS g
where s.StudentId = e.StudentId and g.GroupId = e.GroupId and g.GroupCode = 'SE1607'
order by s.Dob
```

⇨ This query used ORDER BY to review all students in one group.

⇨ Result:

	StudentId	Dob	Name	Email	Gender	StudentCode	GroupCode
1	1	2001-05-06	Le Quang Manh	manh@gmail.com	1	HE163617	SE1607
2	2	2002-07-06	Le Hong Quan	quan@gmail.com	1	HE153617	SE1607
3	3	2002-07-06	Doan Hai Manh	manh@gmail.com	1	HE143617	SE1607
4	4	2003-05-08	Trinh Quang Thang	manh@gmail.com	1	HE173617	SE1607
5	5	2002-10-06	Dinh Tien Manh	manh@gmail.com	1	HE164619	SE1607
6	6	2002-04-09	Le Quang Trung	manh@gmail.com	1	HE123613	SE1607
7	7	2001-05-06	Quang Manh	manh@gmail.com	1	HE163617	SE1607
8	8	2002-03-03	Le Hong	quan@gmail.com	1	HE153635	SE1607
9	9	2002-07-07	Doan Hai	hai@gmail.com	1	HE143435	SE1607
10	10	2003-08-08	Trinh Thang	Thang@gmail.com	1	HE171234	SE1607

2. Query 2

```
select g.*,s.SemesterName from GroupS g INNER JOIN Semester s
where g.SemesterId = s.SemesterId
```

⇨ This query used INNER JOIN to see semester of group.

⇨ Each of semester will have many groups that students joined. Use this query will have school manage groups are having in semseter. Also, it's easier to keep students informed.

⇨ Result:

	GroupId	SubjectId	SemesterId	GroupCode	SemesterName
1	1	1	2	SE1607	Summer2020
2	2	2	3	SE1506	Summer2021
3	3	3	4	IOT1513	Summer2022
4	4	4	5	SE1506	Spring2019
5	5	5	6	SE1617	Spring2020
6	6	6	7	JP1513	Spring2021
7	7	7	8	AI1603	Spring2022
8	8	8	9	AI1605	Fall2019
9	9	9	10	HS1515	Fall2020
10	10	10	2	HS1535	Summer2020

3. Query 3

```
select t.*, g.GroupCode from Teacher t, Teach e, GroupS g
where t.TeacherId = e.TeacherId and g.GroupId = e.GroupId and g.GroupCode = 'SE1506'
```

⇨ Knowing the information of our teachers is very important when student study in school. Use this query will let's student see all information made public by school, so that students could prepare better for lessons and subjects.

⇨ Result:

	TeacherId	Dob	Name	Email	Gender	GroupCode
1	2	1985-05-03	TrungIt	trung@gmail.com	1	SE1506
2	4	1980-08-05	ChiIt	chi@gmail.com	0	SE1506
3	1	1990-04-03	HaiNt	hai@gmail.com	0	SE1506

4. Query 4

```
select * from Assessment a, [Subject] s
where a.SubjectId = s.SubjectId and s.SubjectId =1
```

⇨ In school, each subject will have differences requirements for students to follow. Knowing this will help students study better in that subject. Query above will help students to do this.

⇨ Result:

	AssId	Name	Type	Part	Weight	CompletionCiteria	Duration	Noquestion	SubjectId	SubjectId	SubjectCode	SubjectName
1	1	Progess Test	quiz	2	5	0	20m	20	1	1	WED201c	Web Design
2	2	Assignment	on-going	1	20	0	at home	0	1	1	WED201c	Web Design
3	3	Labs	on-going	5	3	0	in lab session	0	1	1	WED201c	Web Design
4	4	Practical Exam	practical exam	1	25	0	85m	0	1	1	WED201c	Web Design
5	5	Final Exam	final exam	1	30	5	in lab session	60	1	1	WED201c	Web Design

5. Query 5

```
select t.Name, g.* from Teacher t, Teach e, GroupS g
where t.TeacherId = e.TeacherId and g.GroupId = e.GroupId and t.TeacherId =1
```

⇨ School can use this query to manage their teacher by knowing all classes that the teacher is teaching.

⇨ Result:

	Name	GroupId	SubjectId	SemesterId	GroupCode
1	Haint	1	1	2	SE1607
2	Haint	5	5	6	SE1617
3	Haint	2	2	3	SE1506
4	Haint	9	9	10	HS1515

6. Query 6

```
select t.Name, m.*, a.[Name] from Mark m , Student t, Assessment a
where t.StudentId = m.StudentId and m.AsssessId = a.AssId and t.StudentId =1
```

⇨ If students demand to see every mark they have been gave, this query is very usefull.

⇨ Result:

	Name	MarkId	StudentId	AsssessId	Mark	Name
1	Le Quang Manh	1	1	1	5	Progess Test
2	Le Quang Manh	2	1	1	7	Progess Test
3	Le Quang Manh	3	1	2	6	Assignment
4	Le Quang Manh	4	1	3	5	Labs
5	Le Quang Manh	5	1	3	6	Labs
6	Le Quang Manh	6	1	3	7	Labs
7	Le Quang Manh	7	1	3	4	Labs
8	Le Quang Manh	8	1	3	8	Labs
9	Le Quang Manh	9	1	4	8	Practical Exam
10	Le Quang Manh	10	1	5	6	Final Exam

7. Query 7

```
select t.Name, sum(m.Mark * a.[Weight] /100) from Mark m , Student t, Assessment a
where t.StudentId = m.StudentId and m.AsssessId = a.AssId and t.StudentId =1
group by t.[Name]
```

⇨ Average mark (avg) is very important for students and school. Avg can evaluate students through the scale of exellent, good, average, .etc or PASS/ NOT PASS conditions. Query 7 allow students know their avg and also help school manage their students.

⇨ Result:

	Name	(No column name)
1	Le Quang Manh	6,5

8. Query 8

```
select g.*, s.SubjectCode,s.SubjectName from GroupS g, [Subject] s
where g.SubjectId = s.SubjectId
```

⇨ In school, it will be difficult to manage group. It required many queries to control them. Query above is an example. It's help manager to see subjects that have been studied by group.

⇨ Result:

	GroupId	SubjectId	SemesterId	GroupCode	SubjectCode	SubjectName
1	1	1	2	SE1607	WED201c	Web Design
2	2	2	3	SE1506	2DP49	Digital project 2D
3	3	3	4	IOT1513	JPD113	Elementary Japanese 1-A1.1
4	4	4	5	SE1506	IOT102	Internet of Things
5	5	5	6	SE1617	ACC101	Principles of Accounting
6	6	6	7	JP1513	SWE201c	Introduction to Software Engineering
7	7	7	8	AI1603	SWP391	Application development project
8	8	8	9	AI1605	CSD201	Data Structures and Algorithms
9	9	9	10	HS1515	PRE201c	Excel Skills for Business
10	10	10	2	HS1535	SWT301	Software Testing

9. Query 9

```
select a.*, s.SubjectName from Assessment a, [Subject] s
where a.SubjectId = s.SubjectId and s.SubjectId = 3
```

⇨ This query allows students to see the assessment of one subject.

⇨ Result:

	AssId	Name	Type	Part	Weight	CompletionCiteria	Duration	Noquestion	SubjectId	SubjectName
1	6	Progess Test	quiz	2	10	0	20m	50	3	Elementary Japanese 1-A1.1
2	7	Mid-term	Mid-term	1	30	0	60m	50	3	Elementary Japanese 1-A1.1
3	8	Participation	Participation	1	10	0	along semester	0	3	Elementary Japanese 1-A1.1
4	9	Final Exam	final exam	1	40	5	multiple choices	60	3	Elementary Japanese 1-A1.1

10. Query 10

```
select stu.Name, stu.StudentCode, sub.SubjectCode,sub.SubjectName from [Subject] sub,
  (select s.*, g.* from Student s, Enroll e, GroupS g
  where s.StudentId = e.StudentId and g.GroupId = e.GroupId) as stu
where sub.SubjectId = stu.SubjectId
```

⇨ To manage students as I mentioned before need a lot of queries. These queries are very important because if it's wrong, will make the management broke and cause consequences for both school and students. Query 10 above allow school see subjects of students.

⇨ Result:

	Name	StudentCode	SubjectCode	SubjectName
1	Le Quang Manh	HE163617	WED201c	Web Design
2	Le Hong Quan	HE153617	WED201c	Web Design
3	Doan Hai Manh	HE143617	WED201c	Web Design
4	Trinh Quang Thang	HE173617	WED201c	Web Design
5	Dinh Tien Manh	HE164619	WED201c	Web Design
6	Le Quang Trung	HE123613	WED201c	Web Design
7	Quang Manh	HE163617	WED201c	Web Design
8	Le Hong	HE153635	WED201c	Web Design
9	Doan Hai	HE143435	WED201c	Web Design
10	Trinh Thang	HE171234	WED201c	Web Design
11	Dinh Manh	HE164456	2DP49	Digital project 2D
12	Le Trung	HE145632	2DP49	Digital project 2D
13	Quang Manh Hang	HE153617	2DP49	Digital project 2D
14	Trinh Quang Thang	HE173617	2DP49	Digital project 2D
15	Dinh Tien Manh	HE164619	2DP49	Digital project 2D
16	Le Quang Trung	HE123613	2DP49	Digital project 2D
17	Quang Manh	HE163617	2DP49	Digital project 2D
18	Le Hong	HE153635	2DP49	Digital project 2D
19	Doan Hai	HE143435	2DP49	Digital project 2D
20	Trinh Thang	HE171234	2DP49	Digital project 2D
21	Doan Hai An	HE143439	JPD113	Elementary Jap...
22	Trinh Thang Nhe	HE171238	JPD113	Elementary Jap...

CONCLUSION

The database serves two objects: students and schools. In this report, we have briefly introduced the steps of forming a complete database system and its applications. After analysis, we can realize that the database is still sketchy and can be applied to small systems. This database is created for assignment and learning purposes.

Thank you for following up here.