

**Student Grading Management**

**Sub-System**

DBI202 – Lecture: Sonnt5

Triệu Đức Thành

HE163374 | IA1604

**Lời Nói Đầu**

Mỗi ngày, chúng ta gặp gỡ rất nhiều con người, sự vật, sự việc và tiếp cận rất nhiều thông tin thuộc nhiều lĩnh vực trong cuộc sống. Tuy nhiên, chúng ta không cần phải nhớ hết tất cả các thông tin. Từ vô vàn thông tin đó, chúng ta lọc ra những thông tin cần thiết và hữu ích để lưu lại. Đây là data (dữ liệu).

Tuỳ theo từng nhu cầu và mục đích sử dụng, lượng data lưu lại của mỗi người sẽ khác nhau. Hình thức data vô cùng phong phú: chữ viết, con số, kí hiệu, hình ảnh, âm thanh, …

Database (cơ sở dữ liệu) là một tập hợp những data (dữ liệu) có liên quan với nhau. Database được duy trì dưới dạng một tập hợp các tập tin trong hệ điều hành hay được lưu trữ trong các hệ quản trị cơ sở dữ liệu. (Theo Wikipedia)

Và sau đây là bản báo cáo về quá trình tạo dựng một hệ cơ sở dữ liệu để hệ thống, tính điểm cho sinh viên trường đại học FPT. Bản báo cáo được tạo ra theo yêu cầu của giảng viên bộ môn DBI202 của lớp IA1604.

Em xin gửi lời cảm ơn tới thầy Ngô Tùng Sơn – người đã cung cấp cơ sở kiến thức về Cơ sở dữ liệu (Database) và trực tiếp hướng dẫn để em có thể hoàn thành bản báo cáo này.

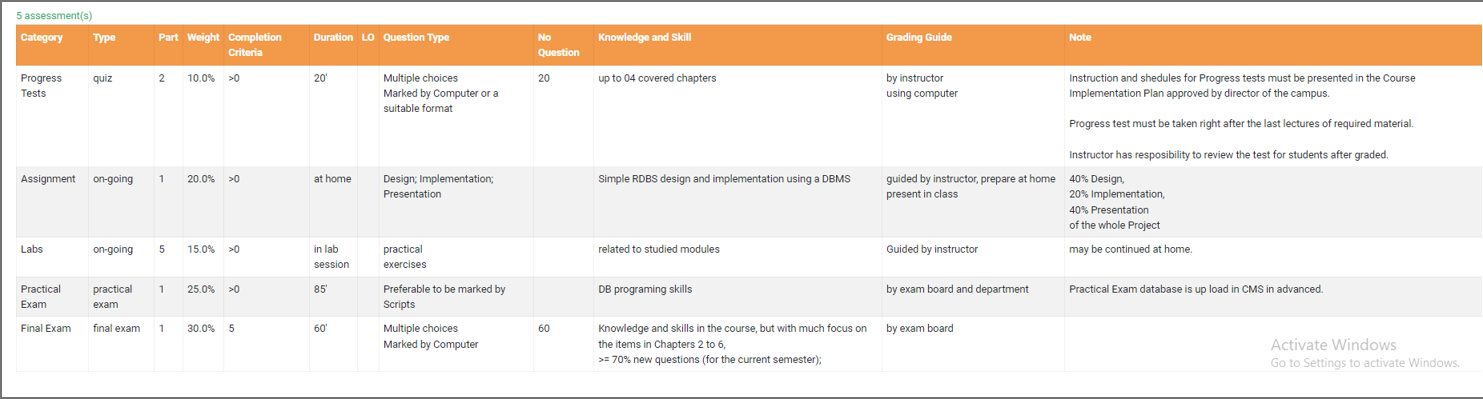
Bản báo cáo bao gồm phân tích đề bài, các bước tạo dựng một database cơ bản một số câu query ứng dụng database, …

**Mục Lục**

1. Phân tích dữ liệu thu thập được từ đề bài
2. Dự đoán các Entities cần thiết trong Database
3. Chuẩn hoá 3NF
4. Tạo Database trên Sql\_Sever
5. Tạo Tables và set khoá chính
6. Tạo các Relation giữa các Table
7. Đưa dữ liệu vào Database

1. Ứng dụng Database (Query, View, store procedure, …)
2. Một số câu query view, trigger, …
3. Mười câu query thể hiện ứng dụng của database
4. Tổng Kết
5. References
6. **Phân tích dữ liệu thu thập được từ đề bài:**

Các Bảng, thực thể đã biết trước (dựa vào đề)

* Điểm thành phần (DBI202): 

+ 5 Loại hạng mục đánh giá: (Progress Tests, Assignment, Labs, Pe, FE).

+ Type: Cách đánh giá: Quiz, quan sát, thi thực hành, PE…

+ Part: Số lần thực hiện.

+ Weight: Trọng số của đầu điểm.

+ Completion Criteria: Điểm tối thiểu cần đạt để đủ chỉ tiêu qua môn.

+ Duration: thời gian thực hiện các hạng mục đánh giá (ở nhà, trong tiết lab).

+ Question Type: Hình thức thực hiện: Multiple choices, presentation, ….

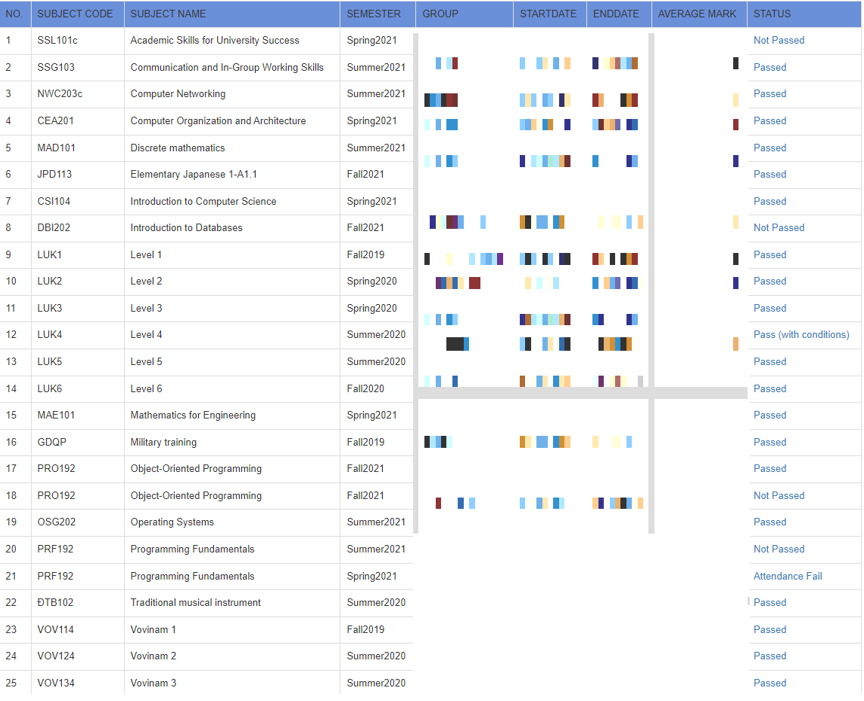
+ No Question: Số lượng câu hỏi.

+ Knowledge and Skill: Các kiến thức và kĩ năng yêu cầu.

+ Grading Guide: Được chấm điểm, đánh giá bởi ai (instructor, computer, exam part, ….)

+ Note: Chú thích ….

* Nhận thấy rằng mỗi 1 môn đều có những loại bài test khác nhau, không dùng chung 1 form hay cơ cấu để tính điểm tất cả các môn => Cần 1 entity là ‘Category’ lưu lại tất cả các loại bài test và các thông số cơ bản để áp dụng cho bài test đó.
* Bảng điểm tổng quát của học sinh:



+ NO: Number of subjects.

+ Subject code – Subject Name: Tên và code của khoá học.

+ Smester: Khoá học diễn ra vào (Mùa + Năm).

+ Group: Lớp học.

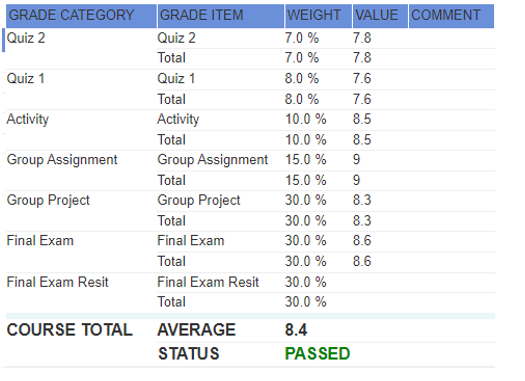
+ StartDate: Thời gian bắt đầu khoá học.

+ Enđate: Thời gian kết thúc khoá học.

+ Average Mark: Điểm trung bình.

+ Status: Pass – Not Pass.

* Thấy rằng bảng này chứa rất nhiều thông tin và có nhiều attributes có thể làm thành khóa. Điều này không tốt cho 1 table. Do vậy một mình bảng này không thể chỉ được thể hiện trong 1 entity được, phải thông qua nhiều các entity nhỏ lẻ để tránh việc vi phạm vào 3NF.
* Bảng điểm chi tiết 1 khoá học:



+ Grade Category: Các loại hạng mục đánh giá

+ Grade Item: các thành phần của hạng mục đó, thêm 1 dòng total

+ Weight: Trọng số hạng mục

+ Value: Điểm

+ Comment: Chú thích của người quản lý điểm (có thể null)

+ Course total: Điểm tổng kết của khoá và Status

* Ở bảng này, ta thấy được các đầu điểm, điểm trung bình và trạng thái của 1 môn học mà 1 sinh viên khi học môn học đó đã đạt được. Cần join nhiều bảng khác như [Category] và một số bảng khác chứa điểm, trọng số, … để tạo thành.

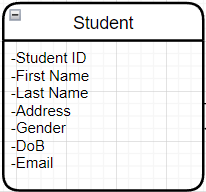
1. **Dự đoán các Entities cần thiết trong Database**
2. **Dự đoán**

* [Students]: Tất cả các bảng trên đều liên quan tới thực thể Students (bảng điểm là bảng điểm của 1 sinh viên, status cũng là của 1 sinh viên, phân loại đầu điểm cũng xem từ môn học mà sinh viên tham gia), mà 1 hệ thống tính điểm thì không chỉ có của riêng 1 sinh viên nào đó. Do vậy, 1 entity Students là chắc chắn phải có trong hệ thống này.
* [Courses]: Một hệ thống tinh điểm thì không thể thiếu các Courses. Sinh viên sẽ tham gia các môn học này, và đầu điểm đánh giá sẽ được dựa theo môn học tương ứng.
* [Lecturers]: Có các môn học thì chúng ta cũng sẽ có các Lecturers tham gia giảng dạy các môn học đó. Giảng viên sẽ trực tiếp cho điểm đánh giá sinh viên theo môn học tương ứng. Note: Lecturers sẽ có 1 attribute là Report phục vụ cho yêu cầu viết query phía sau của đề bài.

* [Classes]: Sinh viên sẽ tham gia các lớp học, giảng viên sẽ giảng dạy các môn học cũng theo các lớp học tương ứng. Do vậy, chúng ta cũng có 1 entitiy là Classes.
* [Groups]: Không thể để tất cả các sinh viên học cùng 1 lớp được. Họ sẽ được phân thành các nhóm và học các lớp theo ngày, giờ khác nhau. Student Grading Management Subsystem 8 Các môn học ứng với từng nhóm cũng có thể được dạy bởi các giảng viên khác nhau. Do vậy, chúng ta cũng sẽ có 1 entity là Groups.
* [Categories]: Một môn học được đánh giá qua nhiều bài test và nhiều đầu điểm khác nhau. Các giảng viên sẽ cho điểm đánh giá sinh viên học môn học tương ứng dựa vào các loại đầu điểm này. Do vậy, cần 1 entity là Categories

1. **Các attribute tương ứng của các entity**

* Student



- Student ID: ID của sinh viên

- First Name: Tên của sinh viên

- Last Name: Họ của sinh viên

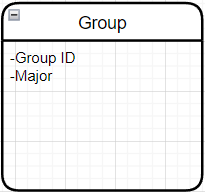
- Address: Địa chỉ

- Gender: Giới tính

- DoB: Ngày Sinh

- Email

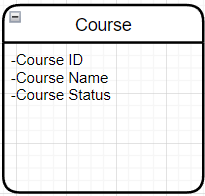
* Group



- GroupID: ID của Group

- Major: Chuyên ngành

* Course

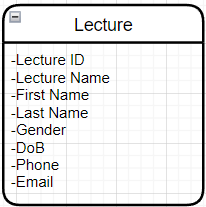


- CourseID: ID khoá học

- Course Name: Tên khoá học

- Course Status: Online - Offline

* Lecture



- LectureID: ID của giáo viên

- First Name: Tên giáo viên

- Last Name: Họ của giáo viên

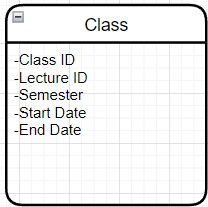
- Gender: Giới tính

- DoB: Ngày sinh

- Phone: Sđt

- Email

* Class



- ClassID: ID lớp học

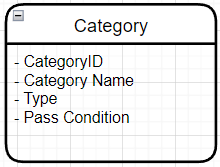
- LectureID: ID giáo viên quản lý Class

- Semester: Học kì

- Start date: Ngày bắt đầu

- End date: Ngày kết thúc

* Category



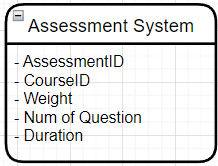
- CategoryID

- Category Name

- Type: Cách thức làm

- Pass Condition: Điệu kiến pass

* Assessment System



- Assessment ID

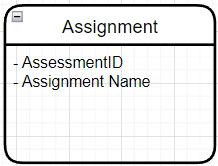
- Course ID: ID khoá học

- Weight: Trọng số

- Num of Question: Số câu hỏi

- Duration: Thời gian làm

* Assignment



- Assignment ID

- Assignment Name

1. **Chuẩn hoá 3NF**

* Xác đinh mối quan hệ giữa các thực thể (1-1, M-M, 1-M, ….)

- Student <-> Group: Một học sinh, sinh viên có thể tham gia nhiều nhóm lớp học và 1 nhóm lớp học cũng có thế có nhiều học sinh, sinh viên.

=> Mối quan hệ M-M

- Student <-> Assessment System: Một học sinh có thể được đánh giá bởi nhiều hệ thống điểm và 1 hệ thống điểm có thể đánh giá được nhiều học sinh

=> Mối quan hệ M-M

- Studen <-> Class: Một học sinh có thể tham gia nhiều lớp học và 1 lớp học có thể chứa nhiều học sinh

=> Mối quan hệ M-M

- Class <-> Lecture: Một lớp chỉ có thể được 1 giáo viên quản lý, và 1 giáo viên có thể quản lý nhiều lớp học.

=> Mối quan hệ 1-M

- Class <-> Group: Một lớp có thể chứa nhiều Group và 1 Group có thể đăng kí nhiều lớp học.

=> Mối quan hệ M-M

- Class <-> Assessment System: Một Class có thể được đánh giá bởi nhiều hệ thống điểm và 1 hệ thống điểm có thể đánh giá được nhiều Class

=> Mối quan hệ M-M

- Class <-> Asignment: Một Class chỉ nhận đúng 1 Assignment và 1 Assignment có thể được giao cho nhiều Class khác nhau

=> Mối quan hệ 1-M

- Course <-> Assessment System: Một Course chỉ có 1 hệ thống đánh giá điểm và 1 hệ thống điểm có thể đánh giá nhiều Course khác nhau.

=> Mối quan hệ 1-M

- Course <-> Assignment: 1 Course chỉ assignment và 1 assignment có thể dùng trong nhiều course  
=> Mối quan hệ 1-M

- Category <-> Assissesment System: Một Category có thể tổng hợp từ nhiều Assignment Systems và 1 Assignment Systems chỉ có thể đưa vào 1 Category duy nhất.

=> Mối quan hệ 1-M

-Lecture <-> Assignment: Một Lecture có thể ra nhiều assignment và một assignment có thể được dùng bởi nhiều Lecture

=> Mối quan hệ M-M

* Phân tách các quan hệ giữa các Entity

- 1-n: Tạo Foriegn Key

- n-n: Tạo 1 bảng mới có khoá chính là khoá chính của 2 bảng còn lại

+ Student <-> Group => Tạo Table Join (StudentID, GroupID)

+ Student <-> Assessment System => Tạo Table Grade (AssessmentID, CourseID, StudentID)

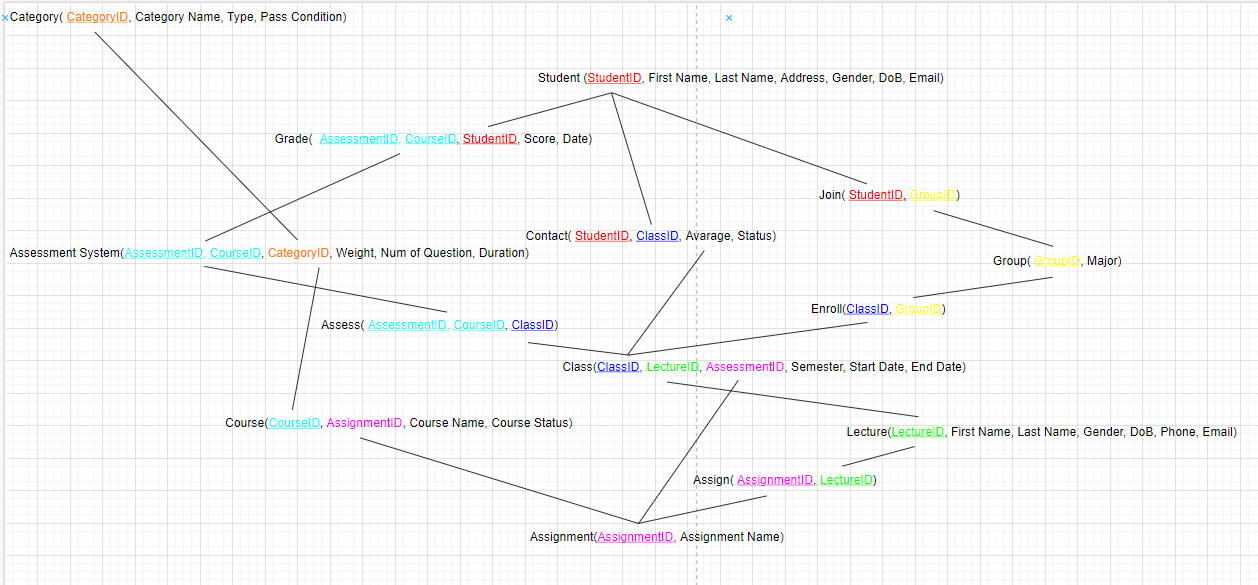
+ Studen <-> Class => Tạo Table Contact (StudentID, ClassID)

+ Class <-> Group => Tạo Table Enroll (ClassID, GroupID)

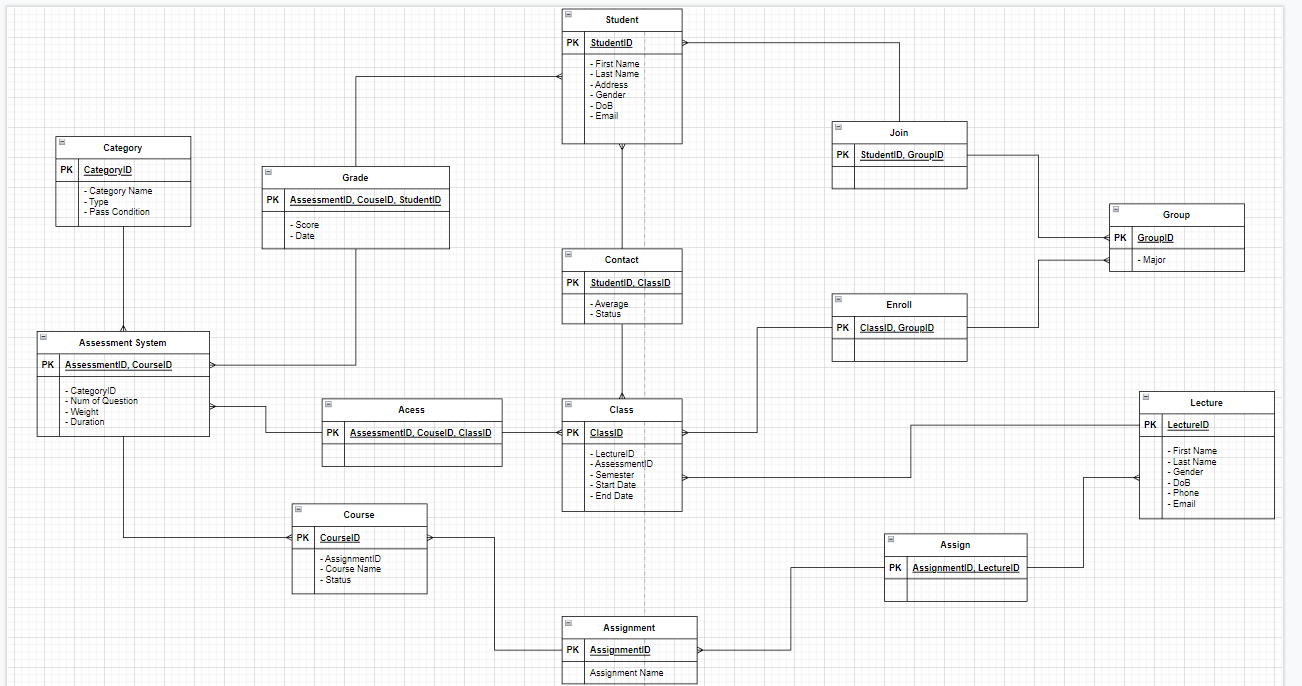
+ Class <-> Assessment System => Tạo Table Assess (AssessmentID, CourseID, ClassID)

+ Lecture <-> Assignment => Tạo Table Assign (AssignmentID, LectureID)

* Chuyển đổi các Entity thành các Table và xác định Primary Key, Foriegn Key, Attributes các Table

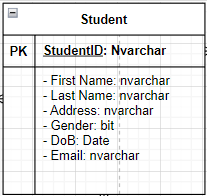


* ERD Diagram

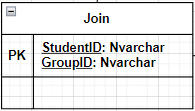


* Định dạng các kiểu dữ liệu cho từng attribute của các Table

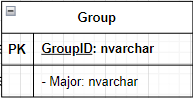
- Student:



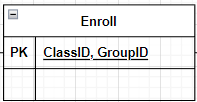
- Join:



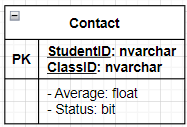
- Group:



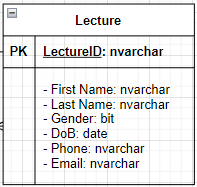
- Enroll:



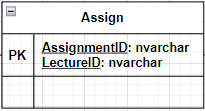
- Contact:



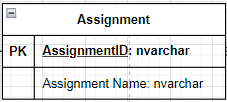
- Lecture:



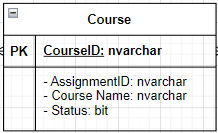
- Assign:



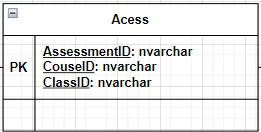
- Assignment:



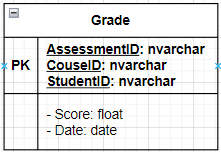
- Course:



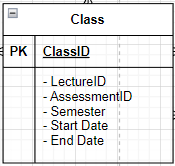
- Acess:



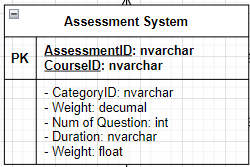
- Grade:



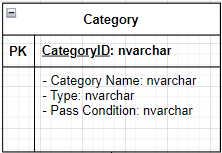
- Class:



- Assessment System:



- Category:



1. **Tạo Database trên Sql Sever**
2. **Tạo các Table và set khoá chính:**

* **Table [Acess]**

**CREATE** **TABLE** **[**dbo**].[**Acess**](**

**[**AssessmentID**]** **[**nvarchar**](**50**)** **NOT** **NULL,**

**[**CourseID**]** **[**nvarchar**](**50**)** **NOT** **NULL,**

**[**ClassID**]** **[**nvarchar**](**50**)** **NOT** **NULL,**

**CONSTRAINT** **[**PK\_Acess**]** **PRIMARY** **KEY** **CLUSTERED**

**(**

**[**AssessmentID**]** **ASC,**

**[**CourseID**]** **ASC,**

**[**ClassID**]** **ASC**

**)WITH** **(**PAD\_INDEX **=** **OFF,** STATISTICS\_NORECOMPUTE **=** **OFF,** IGNORE\_DUP\_KEY **=** **OFF,** ALLOW\_ROW\_LOCKS **=** **ON,** ALLOW\_PAGE\_LOCKS **=** **ON,** OPTIMIZE\_FOR\_SEQUENTIAL\_KEY **=** **OFF)** **ON** **[PRIMARY]**

**)** **ON** **[PRIMARY]**

**GO**

* **Table [Assessment System]**

**CREATE** **TABLE** **[**dbo**].[**Assessment System**](**

**[**AssessmentID**]** **[**nvarchar**](**50**)** **NOT** **NULL,**

**[**CourseID**]** **[**nvarchar**](**50**)** **NOT** **NULL,**

**[**CategoryID**]** **[**nvarchar**](**50**)** **NULL,**

**[**Weight**]** **[**float**]** **NULL,**

**[**Num **of** Question**]** **[**int**]** **NULL,**

**[**Duration**]** **[**nvarchar**](**20**)** **NULL,**

**CONSTRAINT** **[**PK\_Assessment System**]** **PRIMARY** **KEY** **CLUSTERED**

**(**

**[**AssessmentID**]** **ASC,**

**[**CourseID**]** **ASC**

**)WITH** **(**PAD\_INDEX **=** **OFF,** STATISTICS\_NORECOMPUTE **=** **OFF,** IGNORE\_DUP\_KEY **=** **OFF,** ALLOW\_ROW\_LOCKS **=** **ON,** ALLOW\_PAGE\_LOCKS **=** **ON,** OPTIMIZE\_FOR\_SEQUENTIAL\_KEY **=** **OFF)** **ON** **[PRIMARY]**

**)** **ON** **[PRIMARY]**

**GO**

* **Table [Assign]**

**CREATE** **TABLE** **[**dbo**].[**Assign**](**

**[**AssignmentID**]** **[**nvarchar**](**50**)** **NOT** **NULL,**

**[**LectureID**]** **[**nvarchar**](**50**)** **NOT** **NULL,**

**CONSTRAINT** **[**PK\_Assign**]** **PRIMARY** **KEY** **CLUSTERED**

**(**

**[**AssignmentID**]** **ASC,**

**[**LectureID**]** **ASC**

**)WITH** **(**PAD\_INDEX **=** **OFF,** STATISTICS\_NORECOMPUTE **=** **OFF,** IGNORE\_DUP\_KEY **=** **OFF,** ALLOW\_ROW\_LOCKS **=** **ON,** ALLOW\_PAGE\_LOCKS **=** **ON,** OPTIMIZE\_FOR\_SEQUENTIAL\_KEY **=** **OFF)** **ON** **[PRIMARY]**

**)** **ON** **[PRIMARY]**

**GO**

* **Table [Assignment]**

**CREATE** **TABLE** **[**dbo**].[**Assignment**](**

**[**AssignmentID**]** **[**nvarchar**](**50**)** **NOT** **NULL,**

**[**Assignment Name**]** **[**nvarchar**](**100**)** **NULL,**

**CONSTRAINT** **[**PK\_Assignment**]** **PRIMARY** **KEY** **CLUSTERED**

**(**

**[**AssignmentID**]** **ASC**

**)WITH** **(**PAD\_INDEX **=** **OFF,** STATISTICS\_NORECOMPUTE **=** **OFF,** IGNORE\_DUP\_KEY **=** **OFF,** ALLOW\_ROW\_LOCKS **=** **ON,** ALLOW\_PAGE\_LOCKS **=** **ON,** OPTIMIZE\_FOR\_SEQUENTIAL\_KEY **=** **OFF)** **ON** **[PRIMARY]**

**)** **ON** **[PRIMARY]**

**GO**

**SET** ANSI\_NULLS **ON**

**GO**

**SET** QUOTED\_IDENTIFIER **ON**

**GO**

* **Table [Category]**

**CREATE** **TABLE** **[**dbo**].[**Category**](**

**[**CategoryID**]** **[**nvarchar**](**50**)** **NOT** **NULL,**

**[**Category Name**]** **[**nvarchar**](**50**)** **NULL,**

**[Type]** **[**nvarchar**](**20**)** **NULL,**

**[**Pass Condition**]** **[**nvarchar**](**50**)** **NULL,**

**CONSTRAINT** **[**PK\_Category**]** **PRIMARY** **KEY** **CLUSTERED**

**(**

**[**CategoryID**]** **ASC**

**)WITH** **(**PAD\_INDEX **=** **OFF,** STATISTICS\_NORECOMPUTE **=** **OFF,** IGNORE\_DUP\_KEY **=** **OFF,** ALLOW\_ROW\_LOCKS **=** **ON,** ALLOW\_PAGE\_LOCKS **=** **ON,** OPTIMIZE\_FOR\_SEQUENTIAL\_KEY **=** **OFF)** **ON** **[PRIMARY]**

**)** **ON** **[PRIMARY]**

**GO**

* **Table [Class]**

**CREATE** **TABLE** **[**dbo**].[Class](**

**[**ClassID**]** **[**nvarchar**](**50**)** **NOT** **NULL,**

**[**LectureID**]** **[**nvarchar**](**50**)** **NULL,**

**[**AssessmentID**]** **[**nvarchar**](**50**)** **NULL,**

**[**Semester**]** **[**nvarchar**](**20**)** **NULL,**

**[Start** Date**]** **[**date**]** **NULL,**

**[End** Date**]** **[**date**]** **NULL,**

**CONSTRAINT** **[**PK\_Class**]** **PRIMARY** **KEY** **CLUSTERED**

**(**

**[**ClassID**]** **ASC**

**)WITH** **(**PAD\_INDEX **=** **OFF,** STATISTICS\_NORECOMPUTE **=** **OFF,** IGNORE\_DUP\_KEY **=** **OFF,** ALLOW\_ROW\_LOCKS **=** **ON,** ALLOW\_PAGE\_LOCKS **=** **ON,** OPTIMIZE\_FOR\_SEQUENTIAL\_KEY **=** **OFF)** **ON** **[PRIMARY]**

**)** **ON** **[PRIMARY]**

**GO**

* **Table [Contact]**

**CREATE** **TABLE** **[**dbo**].[**Contact**](**

**[**StudentID**]** **[**nvarchar**](**50**)** **NOT** **NULL,**

**[**ClassID**]** **[**nvarchar**](**50**)** **NOT** **NULL,**

**[**Average**]** **[**float**]** **NULL,**

**[**PASS**]** **[**bit**]** **NULL,**

**CONSTRAINT** **[**PK\_Contact**]** **PRIMARY** **KEY** **CLUSTERED**

**(**

**[**StudentID**]** **ASC,**

**[**ClassID**]** **ASC**

**)WITH** **(**PAD\_INDEX **=** **OFF,** STATISTICS\_NORECOMPUTE **=** **OFF,** IGNORE\_DUP\_KEY **=** **OFF,** ALLOW\_ROW\_LOCKS **=** **ON,** ALLOW\_PAGE\_LOCKS **=** **ON,** OPTIMIZE\_FOR\_SEQUENTIAL\_KEY **=** **OFF)** **ON** **[PRIMARY]**

**)** **ON** **[PRIMARY]**

**GO**

* **Table [Course]**

**CREATE** **TABLE** **[**dbo**].[**Course**](**

**[**CourseID**]** **[**nvarchar**](**50**)** **NOT** **NULL,**

**[**AssignmentID**]** **[**nvarchar**](**50**)** **NULL,**

**[**Course Name**]** **[**nvarchar**](**100**)** **NULL,**

**[Online]** **[**bit**]** **NULL,**

**CONSTRAINT** **[**PK\_Course**]** **PRIMARY** **KEY** **CLUSTERED**

**(**

**[**CourseID**]** **ASC**

**)WITH** **(**PAD\_INDEX **=** **OFF,** STATISTICS\_NORECOMPUTE **=** **OFF,** IGNORE\_DUP\_KEY **=** **OFF,** ALLOW\_ROW\_LOCKS **=** **ON,** ALLOW\_PAGE\_LOCKS **=** **ON,** OPTIMIZE\_FOR\_SEQUENTIAL\_KEY **=** **OFF)** **ON** **[PRIMARY]**

**)** **ON** **[PRIMARY]**

**GO**

* **Table [Enroll]**

**CREATE** **TABLE** **[**dbo**].[**Enroll**](**

**[**ClassID**]** **[**nvarchar**](**50**)** **NOT** **NULL,**

**[**GroupID**]** **[**nvarchar**](**50**)** **NOT** **NULL,**

**CONSTRAINT** **[**PK\_Enroll**]** **PRIMARY** **KEY** **CLUSTERED**

**(**

**[**ClassID**]** **ASC,**

**[**GroupID**]** **ASC**

**)WITH** **(**PAD\_INDEX **=** **OFF,** STATISTICS\_NORECOMPUTE **=** **OFF,** IGNORE\_DUP\_KEY **=** **OFF,** ALLOW\_ROW\_LOCKS **=** **ON,** ALLOW\_PAGE\_LOCKS **=** **ON,** OPTIMIZE\_FOR\_SEQUENTIAL\_KEY **=** **OFF)** **ON** **[PRIMARY]**

**)** **ON** **[PRIMARY]**

**GO**

* **Table [Grade]**

**CREATE** **TABLE** **[**dbo**].[**Grade**](**

**[**AssessmmentID**]** **[**nvarchar**](**50**)** **NOT** **NULL,**

**[**CourseID**]** **[**nvarchar**](**50**)** **NOT** **NULL,**

**[**StudentID**]** **[**nvarchar**](**50**)** **NOT** **NULL,**

**[**Score**]** **[**float**]** **NULL,**

**[**Date**]** **[**date**]** **NULL,**

**CONSTRAINT** **[**PK\_Grade**]** **PRIMARY** **KEY** **CLUSTERED**

**(**

**[**AssessmmentID**]** **ASC,**

**[**CourseID**]** **ASC,**

**[**StudentID**]** **ASC**

**)WITH** **(**PAD\_INDEX **=** **OFF,** STATISTICS\_NORECOMPUTE **=** **OFF,** IGNORE\_DUP\_KEY **=** **OFF,** ALLOW\_ROW\_LOCKS **=** **ON,** ALLOW\_PAGE\_LOCKS **=** **ON,** OPTIMIZE\_FOR\_SEQUENTIAL\_KEY **=** **OFF)** **ON** **[PRIMARY]**

**)** **ON** **[PRIMARY]**

**GO**

* **Table [Group]**

**CREATE** **TABLE** **[**dbo**].[Group](**

**[**GroupID**]** **[**nvarchar**](**50**)** **NOT** **NULL,**

**[**Major**]** **[**nvarchar**](**30**)** **NULL,**

**CONSTRAINT** **[**PK\_Group**]** **PRIMARY** **KEY** **CLUSTERED**

**(**

**[**GroupID**]** **ASC**

**)WITH** **(**PAD\_INDEX **=** **OFF,** STATISTICS\_NORECOMPUTE **=** **OFF,** IGNORE\_DUP\_KEY **=** **OFF,** ALLOW\_ROW\_LOCKS **=** **ON,** ALLOW\_PAGE\_LOCKS **=** **ON,** OPTIMIZE\_FOR\_SEQUENTIAL\_KEY **=** **OFF)** **ON** **[PRIMARY]**

**)** **ON** **[PRIMARY]**

**GO**

* **Table [Join]**

**CREATE** **TABLE** **[**dbo**].[Join](**

**[**StudentID**]** **[**nvarchar**](**50**)** **NOT** **NULL,**

**[**GroupID**]** **[**nvarchar**](**50**)** **NOT** **NULL,**

**CONSTRAINT** **[**PK\_Join**]** **PRIMARY** **KEY** **CLUSTERED**

**(**

**[**StudentID**]** **ASC,**

**[**GroupID**]** **ASC**

**)WITH** **(**PAD\_INDEX **=** **OFF,** STATISTICS\_NORECOMPUTE **=** **OFF,** IGNORE\_DUP\_KEY **=** **OFF,** ALLOW\_ROW\_LOCKS **=** **ON,** ALLOW\_PAGE\_LOCKS **=** **ON,** OPTIMIZE\_FOR\_SEQUENTIAL\_KEY **=** **OFF)** **ON** **[PRIMARY]**

**)** **ON** **[PRIMARY]**

**GO**

* **Table [Lecture]**

**CREATE** **TABLE** **[**dbo**].[**Lecture**](**

**[**LectureID**]** **[**nvarchar**](**50**)** **NOT** **NULL,**

**[First** Name**]** **[**nvarchar**](**10**)** **NULL,**

**[Last** Name**]** **[**nvarchar**](**10**)** **NULL,**

**[**Gender**]** **[**bit**]** **NULL,**

**[**DoB**]** **[**date**]** **NULL,**

**[**Phone**]** **[**nvarchar**](**20**)** **NULL,**

**[**Emall**]** **[**nvarchar**](**200**)** **NULL,**

**CONSTRAINT** **[**PK\_Lecture**]** **PRIMARY** **KEY** **CLUSTERED**

**(**

**[**LectureID**]** **ASC**

**)WITH** **(**PAD\_INDEX **=** **OFF,** STATISTICS\_NORECOMPUTE **=** **OFF,** IGNORE\_DUP\_KEY **=** **OFF,** ALLOW\_ROW\_LOCKS **=** **ON,** ALLOW\_PAGE\_LOCKS **=** **ON,** OPTIMIZE\_FOR\_SEQUENTIAL\_KEY **=** **OFF)** **ON** **[PRIMARY]**

**)** **ON** **[PRIMARY]**

**GO**

* **Table [Student]**

**CREATE** **TABLE** **[**dbo**].[**Student**](**

**[**StudentID**]** **[**nvarchar**](**50**)** **NOT** **NULL,**

**[First** Name**]** **[**nvarchar**](**10**)** **NULL,**

**[Last** Name**]** **[**nvarchar**](**10**)** **NULL,**

**[**Address**]** **[**nvarchar**](**200**)** **NULL,**

**[**Gender**]** **[**bit**]** **NULL,**

**[**DoB**]** **[**date**]** **NULL,**

**[**Emaill**]** **[**nvarchar**](**100**)** **NULL,**

**CONSTRAINT** **[**PK\_Student**]** **PRIMARY** **KEY** **CLUSTERED**

**(**

**[**StudentID**]** **ASC**

**)WITH** **(**PAD\_INDEX **=** **OFF,** STATISTICS\_NORECOMPUTE **=** **OFF,** IGNORE\_DUP\_KEY **=** **OFF,** ALLOW\_ROW\_LOCKS **=** **ON,** ALLOW\_PAGE\_LOCKS **=** **ON,** OPTIMIZE\_FOR\_SEQUENTIAL\_KEY **=** **OFF)** **ON** **[PRIMARY]**

**)** **ON** **[PRIMARY]**

1. **Tạo các Relations giữa các Table**

* **[Acess] – [Assessment System]**

**ALTER** **TABLE** **[**dbo**].[**Acess**]** **WITH** **CHECK** **ADD** **CONSTRAINT** **[**FK\_Acess\_Assessment System**]** **FOREIGN** **KEY([**AssessmentID**],** **[**CourseID**])**

**REFERENCES** **[**dbo**].[**Assessment System**]** **([**AssessmentID**],** **[**CourseID**])**

**GO**

**ALTER** **TABLE** **[**dbo**].[**Acess**]** **CHECK** **CONSTRAINT** **[**FK\_Acess\_Assessment System**]**

**GO**

* **[Acess] – [Class]**

**ALTER** **TABLE** **[**dbo**].[**Acess**]** **WITH** **CHECK** **ADD** **CONSTRAINT** **[**FK\_Acess\_Class**]** **FOREIGN** **KEY([**ClassID**])**

**REFERENCES** **[**dbo**].[Class]** **([**ClassID**])**

**GO**

**ALTER** **TABLE** **[**dbo**].[**Acess**]** **CHECK** **CONSTRAINT** **[**FK\_Acess\_Class**]**

**GO**

* **[Assessment System] – [Category]**

**ALTER** **TABLE** **[**dbo**].[**Assessment System**]** **WITH** **CHECK** **ADD** **CONSTRAINT** **[**FK\_Assessment System\_Category**]** **FOREIGN** **KEY([**CategoryID**])**

**REFERENCES** **[**dbo**].[**Category**]** **([**CategoryID**])**

**GO**

**ALTER** **TABLE** **[**dbo**].[**Assessment System**]** **CHECK** **CONSTRAINT** **[**FK\_Assessment System\_Category**]**

**GO**

* **[Assessment System] – [Course]**

**ALTER** **TABLE** **[**dbo**].[**Assessment System**]** **WITH** **CHECK** **ADD** **CONSTRAINT** **[**FK\_Assessment System\_Course**]** **FOREIGN** **KEY([**CourseID**])**

**REFERENCES** **[**dbo**].[**Course**]** **([**CourseID**])**

**GO**

**ALTER** **TABLE** **[**dbo**].[**Assessment System**]** **CHECK** **CONSTRAINT** **[**FK\_Assessment System\_Course**]**

**GO**

* **[Assign] – [Assignment]**

**ALTER** **TABLE** **[**dbo**].[**Assign**]** **WITH** **CHECK** **ADD** **CONSTRAINT** **[**FK\_Assign\_Assignment**]** **FOREIGN** **KEY([**AssignmentID**])**

**REFERENCES** **[**dbo**].[**Assignment**]** **([**AssignmentID**])**

**GO**

**ALTER** **TABLE** **[**dbo**].[**Assign**]** **CHECK** **CONSTRAINT** **[**FK\_Assign\_Assignment**]**

**GO**

* **[Assign] – [Lecture]**

**ALTER** **TABLE** **[**dbo**].[**Assign**]** **WITH** **CHECK** **ADD** **CONSTRAINT** **[**FK\_Assign\_Lecture**]** **FOREIGN** **KEY([**LectureID**])**

**REFERENCES** **[**dbo**].[**Lecture**]** **([**LectureID**])**

**GO**

**ALTER** **TABLE** **[**dbo**].[**Assign**]** **CHECK** **CONSTRAINT** **[**FK\_Assign\_Lecture**]**

**GO**

* **[Class] – [Assignment]**

**ALTER** **TABLE** **[**dbo**].[Class]** **WITH** **CHECK** **ADD** **CONSTRAINT** **[**FK\_Class\_Assignment**]** **FOREIGN** **KEY([**AssessmentID**])**

**REFERENCES** **[**dbo**].[**Assignment**]** **([**AssignmentID**])**

**GO**

**ALTER** **TABLE** **[**dbo**].[Class]** **CHECK** **CONSTRAINT** **[**FK\_Class\_Assignment**]**

**GO**

* **[Class] – [Lecture]**

**ALTER** **TABLE** **[**dbo**].[Class]** **WITH** **CHECK** **ADD** **CONSTRAINT** **[**FK\_Class\_Lecture**]** **FOREIGN** **KEY([**LectureID**])**

**REFERENCES** **[**dbo**].[**Lecture**]** **([**LectureID**])**

**GO**

**ALTER** **TABLE** **[**dbo**].[Class]** **CHECK** **CONSTRAINT** **[**FK\_Class\_Lecture**]**

**GO**

* **[Contact] - [Class]**

**ALTER** **TABLE** **[**dbo**].[**Contact**]** **WITH** **CHECK** **ADD** **CONSTRAINT** **[**FK\_Contact\_Class**]** **FOREIGN** **KEY([**ClassID**])**

**REFERENCES** **[**dbo**].[Class]** **([**ClassID**])**

**GO**

**ALTER** **TABLE** **[**dbo**].[**Contact**]** **CHECK** **CONSTRAINT** **[**FK\_Contact\_Class**]**

**GO**

* **[Contact] – [Student]**

**ALTER** **TABLE** **[**dbo**].[**Contact**]** **WITH** **CHECK** **ADD** **CONSTRAINT** **[**FK\_Contact\_Student**]** **FOREIGN** **KEY([**StudentID**])**

**REFERENCES** **[**dbo**].[**Student**]** **([**StudentID**])**

**GO**

**ALTER** **TABLE** **[**dbo**].[**Contact**]** **CHECK** **CONSTRAINT** **[**FK\_Contact\_Student**]**

**GO**

* **[Course] – [Assignment]**

**ALTER** **TABLE** **[**dbo**].[**Course**]** **WITH** **CHECK** **ADD** **CONSTRAINT** **[**FK\_Course\_Assignment**]** **FOREIGN** **KEY([**AssignmentID**])**

**REFERENCES** **[**dbo**].[**Assignment**]** **([**AssignmentID**])**

**GO**

**ALTER** **TABLE** **[**dbo**].[**Course**]** **CHECK** **CONSTRAINT** **[**FK\_Course\_Assignment**]**

**GO**

* **[Enroll] – [Class]**

**ALTER** **TABLE** **[**dbo**].[**Enroll**]** **WITH** **CHECK** **ADD** **CONSTRAINT** **[**FK\_Enroll\_Class**]** **FOREIGN** **KEY([**ClassID**])**

**REFERENCES** **[**dbo**].[Class]** **([**ClassID**])**

**GO**

**ALTER** **TABLE** **[**dbo**].[**Enroll**]** **CHECK** **CONSTRAINT** **[**FK\_Enroll\_Class**]**

**GO**

* **[Enroll] – [Group]**

**ALTER** **TABLE** **[**dbo**].[**Enroll**]** **WITH** **CHECK** **ADD** **CONSTRAINT** **[**FK\_Enroll\_Group**]** **FOREIGN** **KEY([**GroupID**])**

**REFERENCES** **[**dbo**].[Group]** **([**GroupID**])**

**GO**

**ALTER** **TABLE** **[**dbo**].[**Enroll**]** **CHECK** **CONSTRAINT** **[**FK\_Enroll\_Group**]**

**GO**

* **[Grade] – [Assessment System]**

**ALTER** **TABLE** **[**dbo**].[**Grade**]** **WITH** **CHECK** **ADD** **CONSTRAINT** **[**FK\_Grade\_Assessment System**]** **FOREIGN** **KEY([**AssessmmentID**],** **[**CourseID**])**

**REFERENCES** **[**dbo**].[**Assessment System**]** **([**AssessmentID**],** **[**CourseID**])**

**GO**

**ALTER** **TABLE** **[**dbo**].[**Grade**]** **CHECK** **CONSTRAINT** **[**FK\_Grade\_Assessment System**]**

**GO**

* **[Grade] – [Student]**

**ALTER** **TABLE** **[**dbo**].[**Grade**]** **WITH** **CHECK** **ADD** **CONSTRAINT** **[**FK\_Grade\_Student**]** **FOREIGN** **KEY([**StudentID**])**

**REFERENCES** **[**dbo**].[**Student**]** **([**StudentID**])**

**GO**

**ALTER** **TABLE** **[**dbo**].[**Grade**]** **CHECK** **CONSTRAINT** **[**FK\_Grade\_Student**]**

**GO**

* **[Join] – [Group]**

**ALTER** **TABLE** **[**dbo**].[Join]** **WITH** **CHECK** **ADD** **CONSTRAINT** **[**FK\_Join\_Group**]** **FOREIGN** **KEY([**GroupID**])**

**REFERENCES** **[**dbo**].[Group]** **([**GroupID**])**

**GO**

**ALTER** **TABLE** **[**dbo**].[Join]** **CHECK** **CONSTRAINT** **[**FK\_Join\_Group**]**

**GO**

* **[Join] – [Student]**

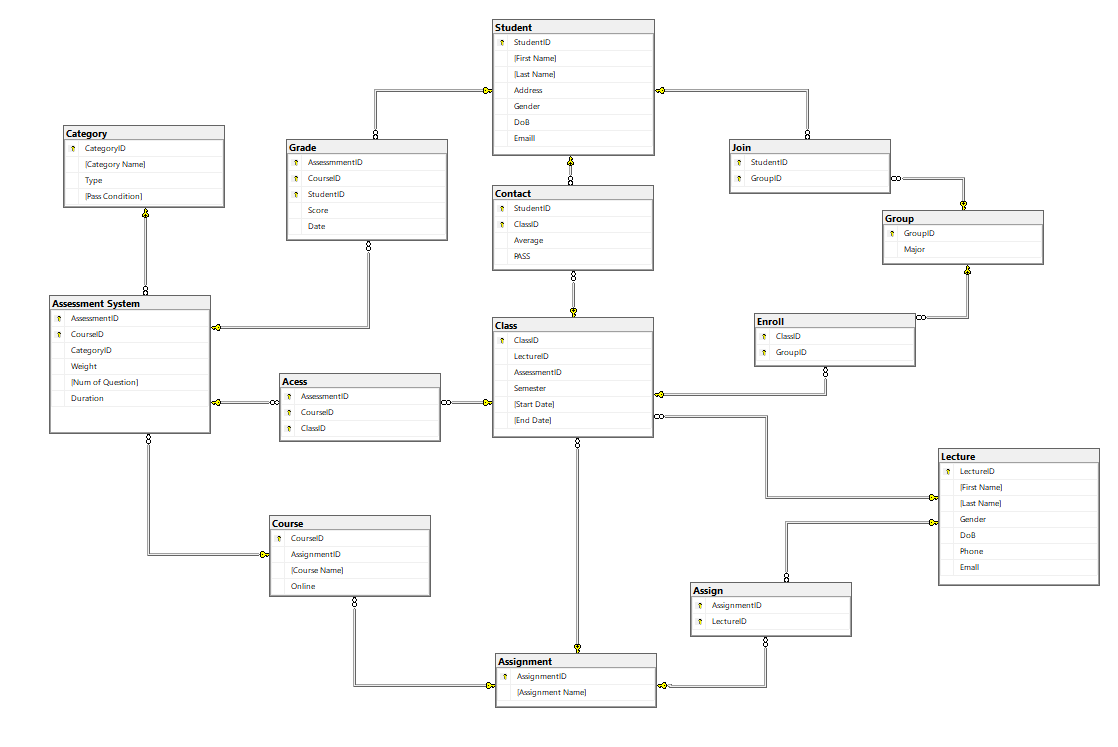
**ALTER** **TABLE** **[**dbo**].[Join]** **WITH** **CHECK** **ADD** **CONSTRAINT** **[**FK\_Join\_Student**]** **FOREIGN** **KEY([**StudentID**])**

**REFERENCES** **[**dbo**].[**Student**]** **([**StudentID**])**

**GO**

**ALTER** **TABLE** **[**dbo**].[Join]** **CHECK** **CONSTRAINT** **[**FK\_Join\_Student**] GO**

* **Database Diagram:**



1. **Đưa Dữ Liệu Vào DataBase**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Student** | | | | | | |
| **StudentID** | **First Name** | **Last Name** | **Address** | **Gender** | **DoB** | **Emaill** |
| HE000 | Hoyt | Schiller | 45511 Jordy Union | 1 | 2002-12-10 | Kendall@florian.biz |
| HE111 | Lauriane | Mueller | 444 Violette Ridges | 1 | 2001-03-28 | Krista@dasia.io |
| HE222 | Afton | Johnson | 989 Krajcik Road | 1 | 2002-12-27 | Blake@blair.com |
| HE333 | Dalton | Metz | 2485 Lind | 0 | 2001-07-22 | Abdiel@frankie.co.uk |
| HE444 | Minnie | Cummerata | 6844 Denesik Route | 0 | 2002-08-14 | Elias@christop.org |
| HE555 | Raleigh | Heller | 9218 Zemlak Cliff | 0 | 2001-02-16 | Kaley@emmy.us |
| HE666 | Annabell | Kuvalis | 1970 Vandervort Cliff | 1 | 2001-11-01 | Bernard@earnest.biz |
| HE777 | Israel | Koch | 18846 Beier Cape | 1 | 2002-02-28 | Kellie.Bechtelar@wilton.biz |
| HE888 | Craig | Anna | 19370 Sibyl Circles | 0 | 2002-07-01 | Margarita\_Davis@neil.us |
| HE999 | Coralie | Purdy | 586 Johnathon Extension | 1 | 2003-05-25 | Vernie@mylene.net |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Lecture** | | | | | | |
| **LectureID** | **First Name** | **Last Name** | **Gender** | **DoB** | **Phone** | **Emall** |
| AnNT7 | Verlie | Senger | 1 | 1975-08-22 | 671-775-2067 x7563 | Federico@eloisa.io |
| DucHM29 | Kiera | Jones | 1 | 1975-05-31 | 1-784-533-6624 | Ryann@jaden.us |
| NangNTH | Zena | Rosenbaum | 0 | 1980-04-12 | 558.872.9878 x401 | Camron@maci.info |
| SonNT5 | Myrna | Denesik | 1 | 1980-02-23 | 414.279.1654 | Cheyenne@kitty.net |
| VanDT | Adelia | Anna | 0 | 1985-01-07 | (080)406-2418 x7085 | Sonya\_Legros@cicero.me |

|  |  |
| --- | --- |
| **GROUP** | |
| **GroupID** | **Major** |
| AI1604 | Artificial Intelligence |
| IA1604 | Information Assurance |
| SE1604 | Software Engineering |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CLASS** | | | | | |
| **ClassID** | **LectureID** | **AssessmentID** | **Semester** | **Start Date** | **End Date** |
| 1 | SonNT5 | 2 | SPR22 | 2022-01-05 | 2022-03-25 |
| 10 | DucHM29 | 5 | SU22 | 2022-05-09 | 2022-07-20 |
| 2 | AnNT7 | 4 | SPR22 | 2022-01-05 | 2022-03-25 |
| 3 | NangNTH | 2 | SPR22 | 2022-01-05 | 2022-03-25 |
| 4 | VanDT | 3 | SPR22 | 2022-01-05 | 2022-03-25 |
| 5 | DucHM29 | 5 | SPR22 | 2022-01-05 | 2022-03-25 |
| 6 | SonNT5 | 1 | SU22 | 2022-05-09 | 2022-07-20 |
| 7 | AnNT7 | 4 | SU22 | 2022-05-09 | 2022-07-20 |
| 8 | NangNTH | 2 | SU22 | 2022-05-09 | 2022-07-20 |
| 9 | VanDT | 3 | SU22 | 2022-05-09 | 2022-07-20 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Contact** | | | |
| **StudentID** | **ClassID** | **Average** | **PASS** |
| HE000 | 2 | 8.65 | 1 |
| HE000 | 3 | NULL | 1 |
| HE111 | 1 | 8.1 | 1 |

|  |  |
| --- | --- |
| **JOIN** | |
| **StudentID** | **GroupID** |
| HE000 | AI1604 |
| HE000 | IA1604 |
| HE111 | AI1604 |
| HE111 | IA1604 |
| HE222 | SE1604 |
| HE333 | IA1604 |
| HE444 | SE1604 |
| HE555 | AI1604 |
| HE666 | AI1604 |
| HE777 | SE1604 |
| HE888 | IA1604 |
| HE999 | SE1604 |

|  |  |
| --- | --- |
| **Enroll** | |
| **ClassID** | **GroupID** |
| 1 | AI1604 |
| 10 | IA1604 |
| 2 | IA1604 |
| 3 | SE1604 |
| 4 | AI1604 |
| 5 | SE1604 |
| 6 | IA1604 |
| 7 | SE1604 |
| 8 | IA1604 |
| 9 | AI1604 |

|  |  |
| --- | --- |
| **Assignment** | |
| **AssignmentID** | **Assignment Name** |
| 1 | DBI202 |
| 2 | LAB211 |
| 3 | JPD113 |
| 4 | CSD202 |
| 5 | IAO202 |

|  |  |
| --- | --- |
| **Assign** | |
| **AssignmentID** | **LectureID** |
| 1 | SonNT5 |
| 2 | NangNTH |
| 2 | SonNT5 |
| 3 | VanDT |
| 4 | AnNT7 |
| 5 | DucHM29 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Course** | | | |
| **CourseID** | **AssignmentID** | **Course Name** | **Online** |
| CSD202 | 4 | Data Structures and Algorithms | 0 |
| DBI202 | 1 | Introduction to Databases | 0 |
| IAO202 | 5 | Introduction to Information Assurance | 1 |
| JPD113 | 3 | Elementary Japanese 1-A1.1 | 0 |
| LAB211 | 2 | OOP with Java Lab | 0 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Assessment System** | | | | | |
| **AssessmentID** | **CourseID** | **CategoryID** | **Weight** | **Num of Question** | **Duration** |
| 1 | CSD202 | Lab1 | 0.05 | NULL | 45 mins |
| 10 | DBI202 | As2 | 0.1 | NULL | 45 mins |
| 11 | DBI202 | PE | 0.4 | 10 | 90 mins |
| 12 | DBI202 | FE | 0.3 | 50 | 90 mins |
| 13 | LAB211 | LOC | 1 | NULL | NULL |
| 2 | CSD202 | Lab2 | 0.05 | NULL | 60 mins |
| 3 | CSD202 | Pt1 | 0.1 | 25 | 45 mins |
| 4 | CSD202 | Pt2 | 0.1 | 25 | 45 mins |
| 5 | CSD202 | PE | 0.4 | 3 | 90 mins |
| 6 | CSD202 | FE | 0.3 | 50 | 90 mins |
| 7 | DBI202 | Lab1 | 0.05 | NULL | 45 mins |
| 8 | DBI202 | Lab2 | 0.05 | NULL | 45 mins |
| 9 | DBI202 | As1 | 0.1 | NULL | 45 mins |

|  |  |  |
| --- | --- | --- |
| **Acess** | | |
| **AssessmentID** | **CourseID** | **ClassID** |
| 1 | CSD202 | 2 |
| 10 | DBI202 | 1 |
| 11 | DBI202 | 1 |
| 12 | DBI202 | 1 |
| 13 | LAB211 | 8 |
| 2 | CSD202 | 2 |
| 3 | CSD202 | 2 |
| 4 | CSD202 | 2 |
| 5 | CSD202 | 2 |
| 6 | CSD202 | 2 |
| 7 | DBI202 | 1 |
| 8 | DBI202 | 1 |
| 9 | DBI202 | 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Grade** | | | | |
| **AssessmmentID** | **CourseID** | **StudentID** | **Score** | **Date** |
| 1 | CSD202 | HE000 | 7 | 2022-01-20 |
| 10 | DBI202 | HE111 | 9 | 2022-03-01 |
| 11 | DBI202 | HE111 | 9 | 2022-03-15 |
| 12 | DBI202 | HE111 | 6 | 2022-03-20 |
| 13 | LAB211 | HE000 | 750 | NULL |
| 2 | CSD202 | HE000 | 8 | 2022-01-30 |
| 3 | CSD202 | HE000 | 9 | 2022-02-10 |
| 4 | CSD202 | HE000 | 7 | 2022-03-01 |
| 5 | CSD202 | HE000 | 9 | 2022-03-15 |
| 6 | CSD202 | HE000 | 9 | 2022-03-20 |
| 7 | DBI202 | HE111 | 10 | 2022-01-20 |
| 8 | DBI202 | HE111 | 10 | 2022-01-30 |
| 9 | DBI202 | HE111 | 8 | 2022-02-10 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | | | |
| CategoryID | Category Name | Type | Pass Condition |
| As1 | Assignment 1 | On-going | >750 |
| As2 | Assignment 2 | On-going | >0 |
| FE | Final exam | Final exam | >4 |
| Lab1 | Lab 1 | On-going | >0 |
| Lab2 | Lab 2 | On-going | >0 |
| LOC | LOC | On-going | >0 |
| PE | Practical exam | Practical exam | >0 |
| Pt1 | Progress test 1 | Quiz | >0 |
| Pt2 | Progress test 2 | Quiz | >0 |
| Pt3 | Progress test 3 | Quiz | >0 |

1. **Ứng dụng Database**

* **View, Index, Procedure, Trigger**

**1. Index:**

-- Index

**CREATE** **INDEX** **[**StuName**]** **ON** Student**([First** Name**],** **[Last** Name**])**

**2. View: Xem những môn sinh viên HE000 đã học và trạng thái**

-- View StudentID = 'He0000'

**DROP** **VIEW** view\_summary

**CREATE** **VIEW** view\_summary **AS**

**SELECT** Acess**.**CourseID**,** cl**.**Semester**,** Enroll**.**GroupID**,** cl**.[Start** Date**],** cl**.[End** Date**],** co**.**Average**,** co**.**PASS

**FROM**

Student st **INNER** **JOIN** Contact co **ON** st**.**StudentID **=** co**.**StudentID

**INNER** **JOIN** **Class** cl **ON** cl**.**ClassID **=** co**.**ClassID

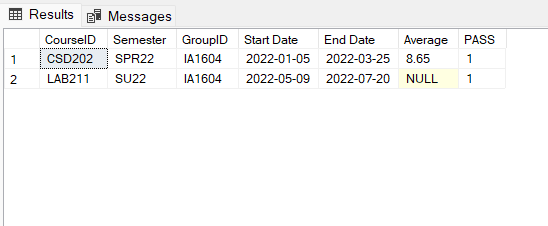
**INNER** **JOIN** Enroll **ON** Enroll**.**ClassID **=** cl**.**ClassID

**LEFT** **JOIN** Acess **ON** cl**.**ClassID **=** Acess**.**ClassID

**WHERE** st**.**StudentID **=** 'HE000'

**GROUP** **BY** CourseID**,** Semester**,** GroupID**,**cl**.[Start** Date**],** cl**.[End** Date**],** co**.**Average**,** co**.**PASS

**SELECT** **\*** **FROM** view\_summary



**3. Stored Procedure: Đếm số sinh viên tham gia group IA1604**

-- STORED PROCEDURE count student join GROUP SE1604

**DROP** **PROCEDURE** Count\_Student

**CREATE** **PROCEDURE** Count\_Student

@GR NVARCHAR**(**50**),**

@**COUNT** INT **OUT**

**AS**

**BEGIN**

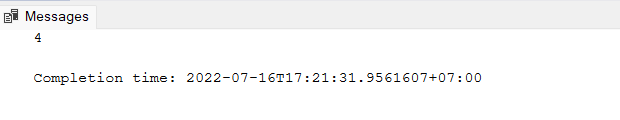
**SELECT** @**COUNT** **=** **COUNT(**StudentID**)** **FROM** **[Join]** **WHERE** GroupID **=** @GR

**END**

**DECLARE** @Num INT **=** 0**;**

**EXEC** Count\_Student @GR **=** 'IA1604'**,** @**COUNT** **=** @Num **OUT**

**PRINT** @Num**;**



=> Có 4 Sinh viên tham gia

**4. Trigger: Làm cho đầu vào của điểm trung bình trong Table [Contact] không lớn hơn 10**

-- TRIGGER –

**DROP** **TRIGGER** Tr\_update\_AVG

**CREATE** **TRIGGER** Tr\_update\_AVG **ON** Contact

**FOR** **UPDATE,** **INSERT** **AS**

**IF** **(SELECT** Average **FROM** Contact**)**

**>** 10

**BEGIN**

**PRINT** N'Wrong Input, Average can greater than 10'

**ROLLBACK** **TRAN**

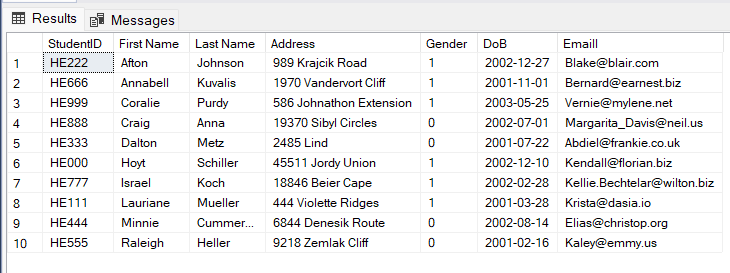
**END**

* **Mười câu Query:**

1. **A query that uses ORDER BY: Sắp xếp danh sách học sinh theo First Name:**

-- A query that uses ORDER BY:

**SELECT** **\*** **FROM** Student **ORDER** **BY** **[First** Name**]**



1. **A query that uses INNER JOINS: Xem các lớp và Assigment mà Sinh Viên tham gia:**

-- A query that uses INNER JOINS: Xem các lớp và Assigment mà Sinh Viên tham gia

**SELECT** Student**.**StudentID**,** **[First** Name**]+**' '**+[Last** Name**]** **AS** **[Full** Name**],**

**Class.**ClassID**,** AssignmentID**,** Semester**,** **[Start** Date**],** **[End** Date**]**

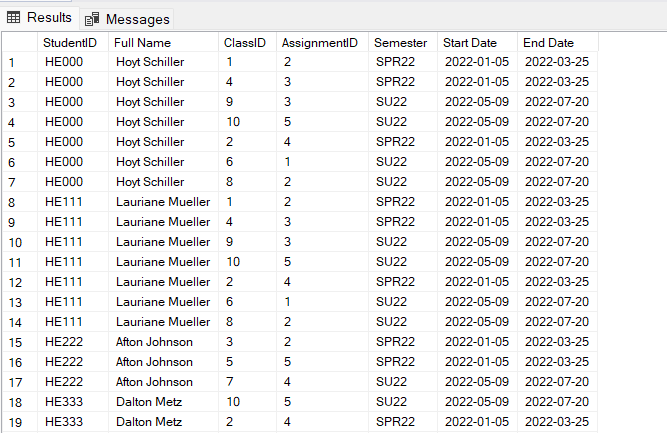
**FROM**

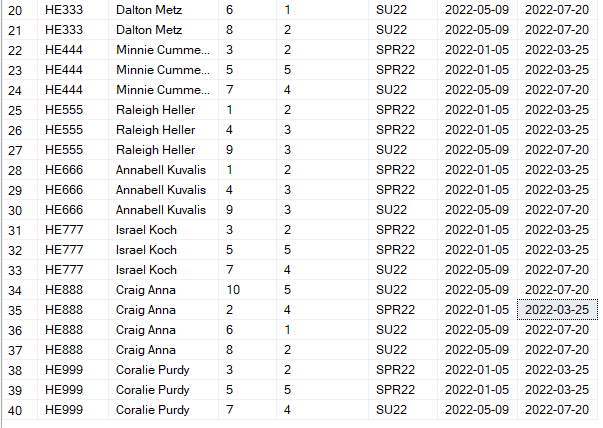
Student **INNER** **JOIN** **[Join]** **ON** Student**.**StudentID **=** **[Join].**StudentID

**INNER** **JOIN** Enroll **ON** Enroll**.**GroupID **=** **[Join].**GroupID

**INNER** **JOIN** **Class** **ON** Enroll**.**ClassID **=** **Class.**ClassID

**ORDER** **BY** Student**.**StudentID

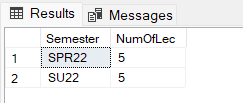




1. **A query that uses aggregate functions: Đếm số giáo viên dạy học trong các Kỳ**

-- A query that uses aggregate functions: Đếm số giáo viên dạy học trong các Kỳ

**SELECT** Semester**,** **COUNT(**LectureID**)** **AS** 'NumOfLec' **FROM** **Class** **GROUP** **BY** Semester



1. **A query that uses the GROUP BY and HAVING clauses: Đếm số giáo viên dạy học trong kỳ 'SPR22':**

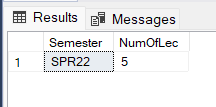
-- A query that uses the GROUP BY and HAVING clauses: Đếm số giáo viên dạy học trong kỳ 'SPR22'

**SELECT** Semester**,** **COUNT(**LectureID**)** **AS** 'NumOfLec'

**FROM** **Class**

**GROUP** **BY** Semester

**HAVING** Semester**=**'SPR22'



1. **A query that uses a sub-query as a relation: Xem các lớp và Assigment mà Sinh Viên tham gia trong kì SPR22:**

-- A query that uses a sub-query as a relation: Xem các lớp và Assigment mà Sinh Viên tham gia trong kì SPR22

**SELECT** st**.**StudentID**,** st**.[Full** Name**],** **Class.**ClassID**,** AssignmentID**,** Semester**,** **[Start** Date**],** **[End** Date**]**

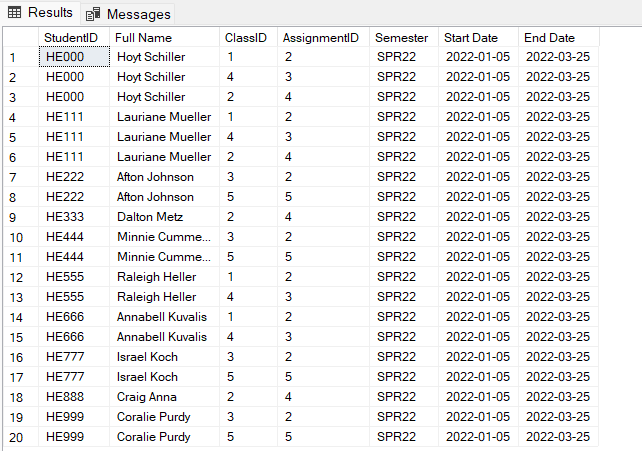
**FROM**

**(SELECT** **[First** Name**]+**' '**+[Last** Name**]** **as** **[Full** Name**],** StudentID

**FROM** Student**)** **as** st **INNER** **JOIN** **[Join]** **ON** st**.**StudentID **=** **[Join].**StudentID

**INNER** **JOIN** Enroll **ON** Enroll**.**GroupID **=** **[Join].**GroupID

**INNER** **JOIN** **Class** **ON** Enroll**.**ClassID **=** **Class.**ClassID



1. **A query that uses a sub-query in the WHERE clause: xem trạng thái của sinh viên ở lớp học có ID = '2':**

-- A query that uses a sub-query in the WHERE clause: xem trạng thái của sinh viên ở lớp học có ID = 2

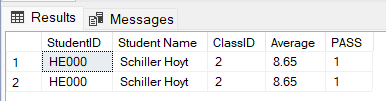
**SELECT** J**.**StudentID**,** **CONCAT([Last** Name**],** ' '**,** **[First** Name**])** **AS** **[**Student Name**],** ClassID**,** Average**,** **[**PASS**]**

**FROM**

Student **INNER** **JOIN** **[Join]** **AS** J **ON** Student**.**StudentID **=** J**.**StudentID

**INNER** **JOIN** **[**Contact**]** **AS** C **ON** J**.**StudentID **=** C**.**StudentID

**WHERE** ClassID **=** **(SELECT** ClassID **FROM** **Class** **WHERE** ClassID **=** '2'**)**



1. **query that uses a self-JOIN: Tên của những Sinh viên lớn tuổi nhất:**

-- A query that uses a self-JOIN: Tên của những Sinh viên lớn tuổi nhất

**SELECT** st**.[First** Name**]+**' '**+** st**.[Last** Name**]** **as** **[Full** Name**]**

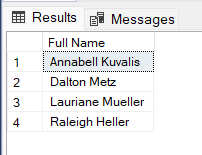
**FROM** Student st

**EXCEPT**

**SELECT** a**.[First** Name**]+** ' '**+** a**.[Last** Name**]**

**FROM** Student a**,** Student b

**WHERE** YEAR**(**a**.**DoB**)** **>** YEAR**(**b**.**DoB**)**



1. **A query that uses a sub-query in the WHERE clause: xem trạng thái của sinh viên ở lớp học có ID = '2':**

-- A query that uses partial matching in the WHERE clause: xem trạng thái của sinh viên ở lớp học có ID = 2

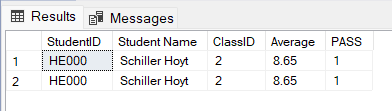
**SELECT** J**.**StudentID**,** **CONCAT([Last** Name**],** ' '**,** **[First** Name**])** **AS** **[**Student Name**],** ClassID**,** Average**,** **[**PASS**]**

**FROM**

Student **INNER** **JOIN** **[Join]** **AS** J **ON** Student**.**StudentID **=** J**.**StudentID

**INNER** **JOIN** **[**Contact**]** **AS** C **ON** J**.**StudentID **=** C**.**StudentID

**WHERE** ClassID **LIKE** '2'



1. **Xem thông tin khoá học 'CSD202' của Sinh Viên HE000:**

-- Xem thông tin khoá học 'CSD202' của Sinh Viên HE000

**SELECT** Acess**.**CourseID**,** cl**.**Semester**,** Enroll**.**GroupID**,** cl**.[Start** Date**],** cl**.[End** Date**],** co**.**Average**,** co**.**PASS

**FROM**

Student st **INNER** **JOIN** Contact co **ON** st**.**StudentID **=** co**.**StudentID

**INNER** **JOIN** **Class** cl **ON** cl**.**ClassID **=** co**.**ClassID

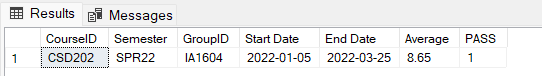
**INNER** **JOIN** Enroll **ON** Enroll**.**ClassID **=** cl**.**ClassID

**LEFT** **JOIN** Acess **ON** cl**.**ClassID **=** Acess**.**ClassID

**WHERE** st**.**StudentID **=** 'HE000'

**GROUP** **BY** CourseID**,** Semester**,** GroupID**,**cl**.[Start** Date**],** cl**.[End** Date**],** co**.**Average**,** co**.**PASS

**HAVING** CourseID**=**'CSD202'



1. **Xem các đầu điểm của môn học CSD202:**

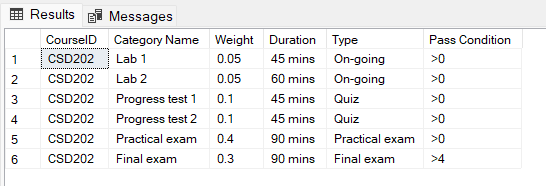
-- Xem các đầu điểm của một môn học (CSD202)

**SELECT** CourseID**,** **[**Category Name**],** **[**Weight**],** Duration**,** **[Type],** **[**Pass Condition**]**

**FROM**

**[**Assessment System**]** ass **INNER** **JOIN** Category c **ON** ass**.**CategoryID **=** c**.**CategoryID

**WHERE** ass**.**CourseID **=** 'CSD202'

****

1. **Tổng kết**

Trong nhiều năm gần đây, thuật ngữ Cơ sở dữ liệu – Database đã trở nên quen thuộc trong nhiều lĩnh vực. Các ứng dụng tin học vào quản lý ngày càng nhiều và đa dạng, hầu hết các lĩnh vực kinh tế, xã hội… đều đã ứng dụng các thành tựu mới của tin học vào phục vụ công tác chuyên môn của mình. Chính vì lẽ đó mà ngày càng nhiều người quan tâm đến thiết kế, xây dựng và ứng dụng cơ sở dữ liệu (CSDL).

CSDL và công nghệ CSDL đã có những tác động to lớn trong việc phát triển sử dụng máy tính. Có thể nói rằng CSDL ảnh hưởng đến tất cả các nơi có sử dụng máy tính như: Kinh doanh, giáo dục, thư viện, …

Bởi nhiều ưu điểm tuyệt vời mà CSDL mang lại, chúng ta có thể thấy mọi trường học đều có một hệ thống CSDL đồ sộ nắm giữ toàn bộ thông tin học sinh, giáo viên, môn học, …

Và điển hình là hệ thống chấm điểm cho học sinh, sinh viên.

Báo cáo trên đã cho thấy quy trình tạo 1 hệ CSDL từ đề bài cho trước quá các bước: Phân tích dữ liệu thu thập được từ đề bài, dự đoán các Entities cần thiết trong Database (khảo sát). Chuẩn hoá 3NF, tạo Database trên Sql\_Sever, truyền dữ liệu vào Database (thiết kế). Kiểm thử, Tiến hành chạy thử các ứng dụng, …

1. **References**

[**https://luatduonggia.vn/co-so-du-lieu-la-gi-vai-tro-phan-loai-va-tam-quan-trong-cua-database/**](https://luatduonggia.vn/co-so-du-lieu-la-gi-vai-tro-phan-loai-va-tam-quan-trong-cua-database/)

[**https://toploigiai.vn/quy-trinh-xay-dung-co-so-du-lieu-la**](https://toploigiai.vn/quy-trinh-xay-dung-co-so-du-lieu-la)

[**https://en.wikipedia.org/wiki/Database**](https://en.wikipedia.org/wiki/Database)

[**https://drive.google.com/file/d/1praVpGRwnhohwQZzO99AQYLAXBg5ZZTk/view?usp=sharing**](https://drive.google.com/file/d/1praVpGRwnhohwQZzO99AQYLAXBg5ZZTk/view?usp=sharing)

[**https://github.com/Neo-Izumi/DBI202-Assignment**](https://github.com/Neo-Izumi/DBI202-Assignment)

[**https://github.com/thajnh30cm/DBI202\_ASG**](https://github.com/thajnh30cm/DBI202_ASG)

[**https://drive.google.com/file/d/1fYYstuuk56-\_XmR7ZlzFhG\_21ysAumtm/view?usp=sharing**](https://drive.google.com/file/d/1fYYstuuk56-_XmR7ZlzFhG_21ysAumtm/view?usp=sharing)

[**https://www.w3schools.com/sql/**](https://www.w3schools.com/sql/)