**ĐỒ ÁN CUỐI KỲ: DAM/ORM FRAMWORK**

MÔN: MẪU THIẾT KẾ HƯỚNG ĐỐI TƯỢNG & ỨNG DỤNG

GIÁO VIÊN: NGUYỄN MINH HUY

NHÓM 03 - DEST

HƯƠNG DẪN SỬ DỤNG

* BƯỚC 1

Download framework ở đường link sau: <https://drive.google.com/open?id=0B6d3S9rbty-cR1pTSjRXSTFCR1U>

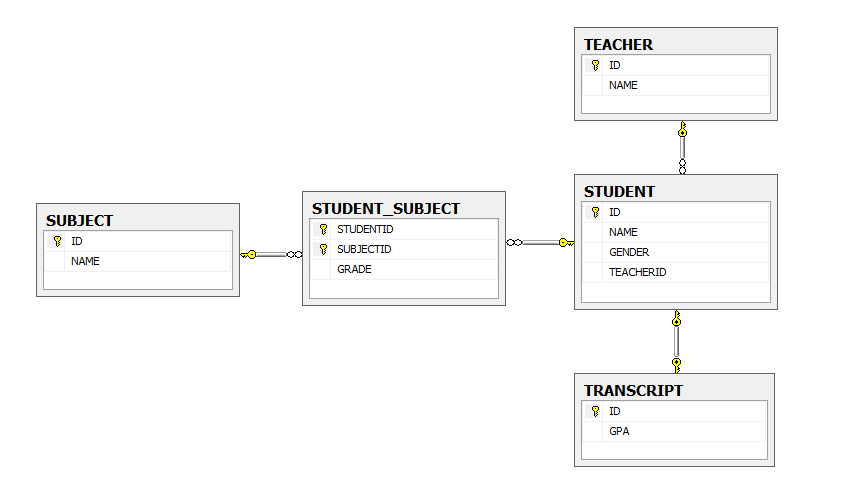
* BƯỚC 2

Tạo project và add reference file SCOFramework.dll vừa tải về.

* BƯỚC 3

Khai báo các lớp đối tượng biểu diễn quan hệ của các bảng. Để hiểu rõ hơn. Ta cùng xét 1 ví dụ:

Giả sử có có CSDL như sau:

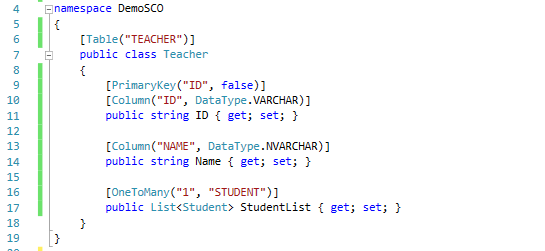


Tiếp theo, ta khai báo các lớp đối tượng tương ứng với các bảng trên và quan hệ giữa chúng.

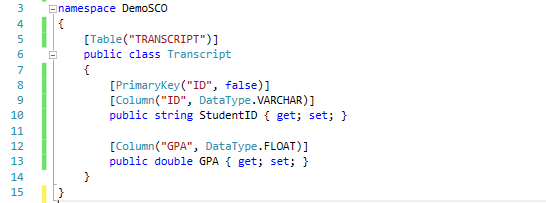
1. Lớp Student



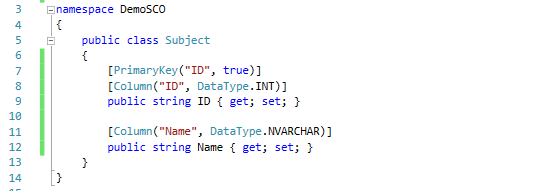
1. Lớp Teacher



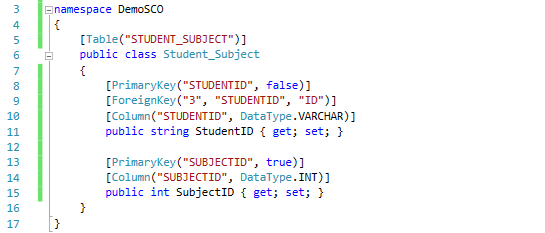
1. Lớp Transcript



1. Lớp Subject



1. Lớp Student\_Subject

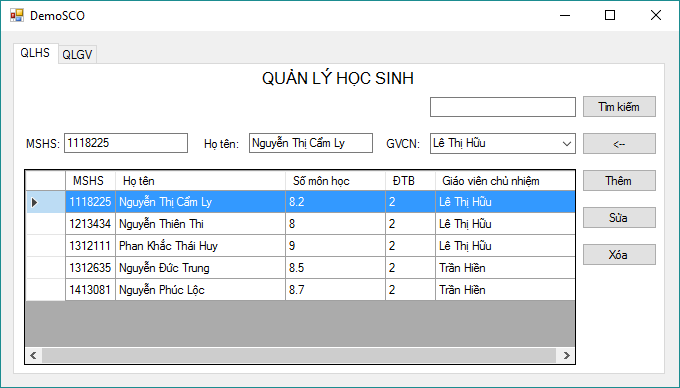


**Lưu ý khi khai báo quan hệ giữa các đối tượng:**

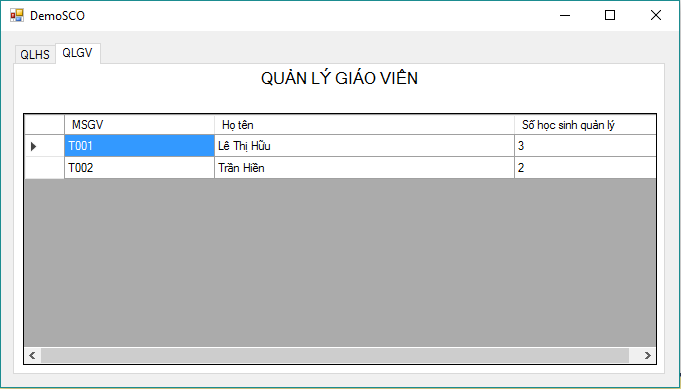
* Tất cả các Attribute dùng để khai báo quan hệ giữa các đối tượng đều gồm có ID quan hệ (RelationshipID) và tên bảng có cùng quan hệ với nó.
* Với quan hệ 1 – N: Đối tượng Teacher chứa N đối tượng Student. Vì thế, ta khai báo attribute OneToMany trước thuộc tính kiểu List<Student> trong lớp Teacher. Đồng thời khai báo khóa ngoại ForeingKeyAttribute ở lớp Student.
* Với quan hệ 1 – 1: Mỗi đối tượng Student tương ứng với 1 đối tượng Transcript. Vì thế, ta khai báo attribute OneToOne trước thuộc tính kiểu Transcript và khóa ngoại ForeingKeyAttribute ở lớp Student (hoặc muốn Transcript chứa Student thì khai báo trong Transcript).
* Với quan hệ N – 1: Nhiều đối tượng Student cùng chung 1 đối tượng kiểu Teacher. Vì thế, ta khai báo attribute ManyToOne trước thuộc tính kiểu Teacher trong lớp Student. Đồng thời khai báo khóa ngoại ForeingKeyAttribute ở lớp Student.
* Với quan hệ N – N: Ta thấy 1 đối tượng Student có nhiều đối tượng Subject và người lại. Mỗi đối tượng Subject cũng bao gồm nhiều Student. Cũng như trong CSDL, ta tạo thêm 1 đối tượng trung gian (bảng trung gian) Student\_Subject. Từ đó đưa quan hệ N – N thành các quan hệ 1 – N, N – 1.
* BƯỚC 4

Sử dụng các hàm/chức năng mà FrameWork hỗ trợ.

Quay trở lại ví dụ như đã nêu ở bước 3. Ta cần xây dựng 1 chương trình nhỏ quản lý học sinh, giáo viên như sau:



Liệt kê danh sách sinh viên cùng thông tin cá nhân. Cho phép tìm, thêm, sửa, xóa.



Liệt kê danh sách Giáo viên chủ nhiệm cùng số học sinh trong lớp

Ta có các phương thức sau trong lớp SCOSqlConnection:

1. Khởi tạo:

SCOConnection connection = new SCOSqlConnection(ConfigurationManager.ConnectionStrings["ConnectionString"].ToString());

1. Open(): Mở kết nối

connection.Open();

1. Close(): Đóng kết nối

connection.Close();

1. ExecuteQuery<T>(string queryString): Thực hiện câu truy vấn, trả về danh sách đối tượng kiểu T ứng với số dòng dữ liệu trả về.

List<Student> students = connection.ExecuteQuery<Student>("SELECT \* FROM STUDENT");

1. ExecuteQueryWithOutRelationship<T>(string queryString): Tương tự câu lệnh ExecuteQuery<T> nhưng bỏ qua các quan hệ OneToMany, ManyToOne, OneToOne mà chỉ lấy dữ liệu từ các cột có trong bảng (ColumnAttribute).
2. ExecuteNonQuery(string queryString): Thực hiện các câu lệnh không trả về dòng dữ liệu (insert, update, delete) mà trả về số dòng thay đổi trong CSDL.

connection.ExecuteQuery<Student>("DELETE STUDENT WHERE ID = '1'");

1. Select<T>().Where(string condition).Having(string condition).GroupBy(string columns).Run(): Select dữ liệu. Kết quả trả về giống với ExecuteQuery<T>.

List<Student> students = connection.Select<Student>().AllRow().Run();

1. Insert(T object): Insert đối tượng vào CSDL.

Student s = new Student();

s.ID = "1312635";

s.Name = "Trung Nguyen";

connection.Insert(s);

1. Update(T object): Update đối tượng vào CSDL.

List<Student> students = connection.Select<Student>().AllRow().Run();

students[0].Name = "Trung Nguyen";

connection.Update(students[0]);

1. Delete(T object): Xóa đối tượng khỏi CSDL.

List<Student> students = connection.Select<Student>().AllRow().Run();

connection.Delete(students[0]);

*Chi tiết source code demo vui lòng tham khảo ..\ 4. Others\Demo*

**CHÚC CÁC BẠN THÀNH CÔNG!**