**Môn: Mẫu thiết kế hướng đối tượng & ứng dụng**

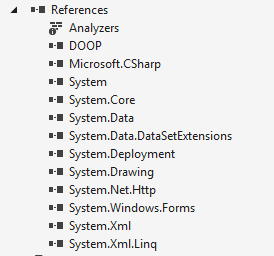
Đồ án: Database access management Framework

Giáo viên: Hồ Tuấn Thanh

Nhóm thực hiện: 23 - DOOP

# USER GUILD

## Tạo project và thêm reference (DOOP) vào project.



## Khai báo các class đối tượng tương ứng với các bảng trong database: Ví dụ ta có database như sau: Khai báo các lớp đối tượng: 1. Lớp KHO: 2. Lớp NGUYENLIEU: 3. Lớp GIANGUYENLIEU: 4. Lớp MONAN: Lưu ý: Những class trên đều sử dụng namespace DOOP\_FRAMEWORK.PropertySQL. - Khai báo quan hệ giữa các lớp: [TenQuanHe(idQuanHe, TenBangQuanHe) + Quan hệ 1-1: Đối tượng NGUYENLIEU chứa duy nhất 1 đối tượng GIANGUYENLIEU nên ta khai báo Attribute [OneOne("2", "GIANGUYENLIEU")] và [ForeignKey("2", "ID", "ID")] (khai báo trường trỏ đến) ở lớp NGUYENLIEU. Nếu muốn đối tượng GIANGUYENLIEU chứa NGUYENLIEU thì ta làm ngược lại. + Quan hệ 1-N: Đối tượng KHO chứa nhiều đối tượng NGUYENLIEU nên ta khai báo Attribute [OneMany("1", "NGUYENLIEU")] trước List đối tượng NGUYENLIEU, đồng thời khai báo khóa ngoại [ForeignKey("1", "KHO", "ID")] ở lớp NGUYENLIEU trỏ đến trường ID của KHO. + Quan hệ N-1: Nhiều đối tượng NGUYENLIEU thuộc cùng 1 đối tượng KHO nên ta khai báo Attribute [ManyOne("1", "KHO")], đồng thời khai báo khóa ngoại [ForeignKey("1", "KHO", "ID")] trỏ đến trường ID của KHO. + Quan hệ N-N: Ta có thể quy về quan hệ 1-N giữa 2 lớp với nhau, tức là 1 đối tượng của lớp này chứa nhiều đối tượng của lớp kia và ngược lại.

## Tạo giao diện quản lý và sử dụng các chức năng mà framework hỗ trợ Giao diện quản lý món ăn Giao diện quản lý Nguyên liệu: Giao diện quản lý nguyên liệu

|  |  |
| --- | --- |
|  | Danh sách các chức năng mà framework hỗ trợ: |

## Khởi tạo kết nối: aConnection connection = new Connection(connectionString);

1. Open(): Mở kết nối:  
   **connection.Open();**
2. Close(): Đóng kết nối:   
   **connection.Close();**
3. EQueryNoneRelationship<T>(string query): Thực hiện câu truy vấn, và trả về dữ liệu của bảng dưới dạng đối tượng.  
    **List<NGUYENLIEU> nguyenlieus = connection.** **EQueryNoneRelationship<NGUYENLIEU>("SELECT \* FROM NGUYENLIEU");**
4. EQuery<T>(string query): Thực hiện câu truy vấn, trả về dữ liệu của bảng dưới dạng đối tượng kèm theo các dữ liệu đối tượng quan hệ: onetoone, onetomany, manytoone.  
    **List<NGUYENLIEU> nguyenlieus = connection.EQuery<NGUYENLIEU>("SELECT \* FROM NGUYENLIEU");**
5. ENoneQuery(string query): Thực hiện câu lệnh không trả về dữ liệu (insert, update, delete) mà trả về kết quả thực thi (số dòng thay đổi trong table).  
    **connection.ENoneQuery<NGUYENLIEU>("DELETE NGUYENLIEU WHERE ID = '1'");**
6. Select<T>().Where(string condition).Having(string condition).GroupBy(string columns).Run(): (theo đúng thứ tự) Trả về dữ liệu như EQuery<T>(string query).  
    **List<NGUYENLIEU> nguyenlieus = connection.Select<NGUYENLIEU>().Where("ID LIKE N'%" + txtkeyword.Text + "%'").Run()**
7. Insert<T>(T obj): thêm 1 dòng dữ liệu dưới dạng đối tượng vào bảng trong CSDL.  
    **NGUYENLIEU s = new NGUYENLIEU();  
    s.ID = ‘nguyenlieun’;  
    s.Name = ‘tennguyenlieun’;  
    s.IDKHO = 123;  
    connection.Insert(s);**
8. Update<T>(T obj): cập nhật 1 dòng dữ liệu dưới dạng đối tượng trong CSDL.  
    **NGUYENLIEU s = new NGUYENLIEU();  
    s.ID = ‘nguyenlieun’;  
    s.Name = ‘tennguyenlieun’;  
    s.IDKHO = 123;  
    connection.Update(s);**
9. Delete<T>(T obj): xóa 1 dòng dữ liệu dưới dạng đối tượng ra khỏi bảng trong CSDL.  
    **NGUYENLIEU s = new NGUYENLIEU();  
    s.ID = ‘nguyenlieun’;  
    s.Name = ‘tennguyenlieun’;  
    s.IDKHO = 123;**  
    **connection.Delete(s);**