**MẪU THIẾT KẾ HƯỚNG ĐỐI TƯỢNG VÀ ỨNG DỤNG**

**BÁO CÁO ĐỒ ÁN CUỐI KỲ**

**DAM/ORM Framework**

Giáo viên: Nguyễn Minh Huy

Hồ Tuấn Thanh

Nhóm 25 – DDD:

* 1512104 – Trần Tấn Đạt
* 1512092 – Nguyễn Thanh Đại
* 1512100 – Nguyễn Quốc Đạt
* 1512047 – Dương Minh Công

C:\Users\tdqua_000\Dropbox\SS-Slides\DeCuong-CDIO\Template CDIO v4.2\Templates\Hinh anh\LogoTruong.png

Bộ môn Công nghệ phần mềm

Khoa Công nghệ thông tin

Đại học Khoa học tự nhiên TP HCM

**MỤC LỤC**

[1 Thông tin chung 3](#_Toc534492506)

[2 Sơ đồ lớp 3](#_Toc534492507)

[3 Mô tả các lớp và chức năng 5](#_Toc534492508)

[4 Các mẫu thiết kế đã áp dụng 8](#_Toc534492509)

# Thông tin chung

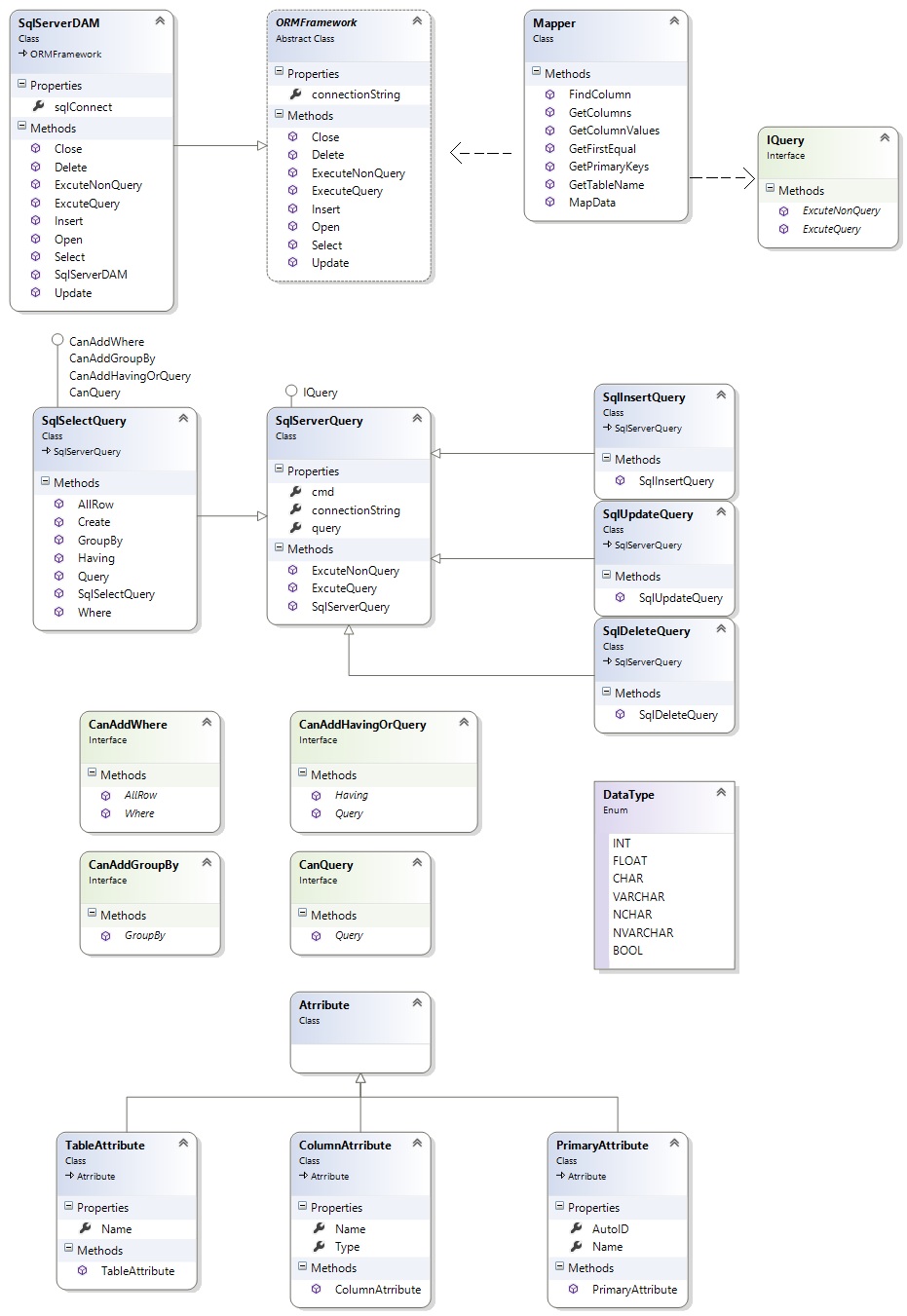
**Thông tin nhóm: Nhóm 25 (DDD)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | MSSV | Họ Tên | Email | Đóng góp |
| 1 | 1512047 | Dương Minh Công | [1512047@student.hcmus.edu.vn](mailto:1512047@student.hcmus.edu.vn) | 15% |
| 2 | 1512092 | Nguyễn Thanh Đại | [1512092@student.hcmus.edu.vn](mailto:1512092@student.hcmus.edu.vn) | 27.5% |
| 3 | 1512100 | Nguyễn Quốc Đạt | [1512100@student.hcmus.edu.vn](mailto:1512100@student.hcmus.edu.vn) | 27.5% |
| 4 | 1512104 | Trần Tấn Đạt | [1512104@student.hcmus.edu.vn](mailto:1512104@student.hcmus.edu.vn) | 30% |

**Mô tả đồ án: Database access management (DAM) Framework**

* Xây dựng DAM/ORM framework, ánh xạ dữ liệu từ database sang lớp đối tượng, giúp cho việc lập trình tương tác với CSDL trở nên dễ dàng hơn.
* Sử dụng reflection và attribute trên ngon ngữ lập trình c# để đọc thông dữ liệu từ CSDL, đọc các thông tin của bảng như tên bảng, tên cột, các dòng dữ liệu.
* Framework hỗ trợ các thao tác cơ bản giống tương tự LINQ, Entity như mở kết nối, đóng kết nối, insert, update, delete, chuyển thao tác trên lớp đối tượng thành SQL.

# Sơ đồ lớp



# Mô tả các lớp và chức năng

#### DAMFramework: Lớp trừu tượng đại diện cho Framework, khai báo các lớp con kế thừa và tự định nghĩa các kết nối với CSDL tương ứng

|  |  |
| --- | --- |
| connectionString | Chuỗi dữ liệu chứa thông tin để kết nối với CSDL |
| Open() | Mở kết nối với CSDL |
| Close() | Đóng kết nối với CSDL |
| Select<T>() | Tạo đối tượng SqlSelectQuery |
| Insert<T>(T obj) | Thêm obj có kiểu dữ liệu T vào CSDL |
| Update<T>(T obj) | Cập nhật đối tượng obj dựa vào danh sách khóa chính đã có. |
| Delete<T>(T obj | Xóa đối tượng obj khỏi CSDL |
| ExcuteQuery<T>(string queryString) | Thực hiện câu lệnh query được lưu trong queryString, trả về 1 List đối tượng |
| ExcuteNonQuery<T>(string queryString) | Thực hiện các câu lệnh như Insert, Update, Delete |

#### SqlServerDAM: Là lớp kế thừa từ lớp DAMFramework, định nghĩa các phương thức, thuộc tính để thao tác với SQL Server. Lớp này có thêm

|  |  |
| --- | --- |
| sqlConnect | Đối tượng kiểu SqlConnection, kết nối với SQL Server |
| SqlServerDAM(string datasource, string database, string username, string password) | Hàm khởi tạo đối tượng với các tham số cơ bản để kết nối với SQL Server |

#### Mapper: Lớp chứa các phương thức để đọc dữ liệu bảng, thông tin về các thành phần của bảng, chuyển đổi dữ liệu từ bảng sang T.

|  |  |
| --- | --- |
| MapData<T>(DataRow dr) | Chuyển dữ liệu từ DataRow sang đối tượng, trả về đôi tượng kiểu T |
| GetTableName<T>() | Trả về tên bảng |
| GetPrimaryKey<T>() | Trả về List PrimaryKey của class T |
| GetColumns<T>() | Trả về List cột của class T |
| GetColumnsValues<T>() | Trả về danh sách cột và giá trị của cột |
| FindColumn(string columnName, List<ColumnAttribute> list) | Trả về Column trong danh sách theo columnName |
| FindColumn(string columnName, Dictionary <ColumnAttribute, object> list) | Trả về Column trong từ điển theo columnName |
| GetFirstEqual(object[] attributes, Type type) | Trả về attribute đầu tiên có cùng kiểu với type. Nếu không có trả về null. |

#### IQuery: Interface quy định các phương thức query của Frame work

|  |  |
| --- | --- |
| ExecuteQuery<T>() | Thực hiện câu lệnh truy vấn. Trả về danh sách đối tượng |
| ExecuteNonQuery() | Thực hiện các câu lệnh không trả về dòng dữ liệu như: Insert, Update, Delete. |

#### SqlServerQuery: Kế thừa từ IQuery cài đặt cụ thể cho SQL Server

|  |  |
| --- | --- |
| connectionString | Chuỗi kết nối SQL Server |
| cmd | Đối tượng SqlCommand để thực hiện các câu truy vấn SQL Server |
| query | Chuỗi truy vấn |
| SqlQuery(SqlConnection sqlConnect, string connectionString) | Hàm khởi tạo với sqlConnect và ConnectionString. |
| SqlQuery(SqlConnection sqlConnect, string connectionString, string query) | Hàm khởi tạo với sqlConnect, connectionString và query. |

#### SqlSelectQuery: Tạo câu truy vấn Select, kế thừa các interface CanAddWhere, CanAddHavingOrQuery, CanAddGroupBy, CanQuery

|  |  |
| --- | --- |
| private SqlSelectQuery(SqlConnection sqlConnect, string | Phương thức khởi tạo |
| static Create(SqlConnection cnn, string connectionString): CanAddWhere<T> | Hàm static Create để khởi tạo đối tượng |
| Where(string condition): CanAddHavingOrQuery<T> | Điều kiện where cho câu lệnh select |
| AllRow(): CanAddHavingOrQuery<T> | Select tất cả các dòng |
| Having(string condition): CanAddGroupBy<T> | Điều kiện Having cho câu lệnh select |
| GroupBy(string columnNames): CanQuery<T> | Điều kiện GroupBy cho câu lệnh select |
| Query (): List<T> | Thực hiện câu truy vấn. Trả về danh sách đối tượng |

#### SqlInsertQuery: Phát sinh câu lệnh SQL Insert tương ứng với đối tượng được truyền vào

#### SqlUpdateQuery: Phát sinh câu lệnh SQL Update tương ứng với đối tượng được truyền vào

#### SqlDeleteQuery: Phát sinh câu lệnh SQL Delete tương ứng với đối tượng được truyền vào

#### CanAddWhere: là interface cho phép thêm điều kiện Where vào câu truy vấn Select

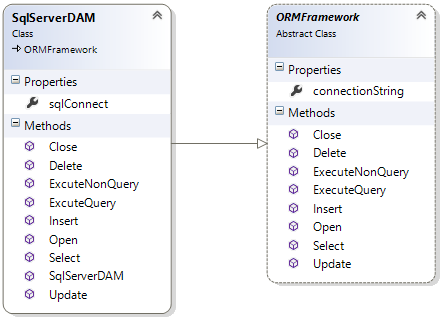
#### CanAddHavingOrQuery: là interface cho phép thêm điều kiện Having vào câu truy vấn sau khi thêm điều kiện Where hoặc có thể gọi Query để thực hiện truy vấn

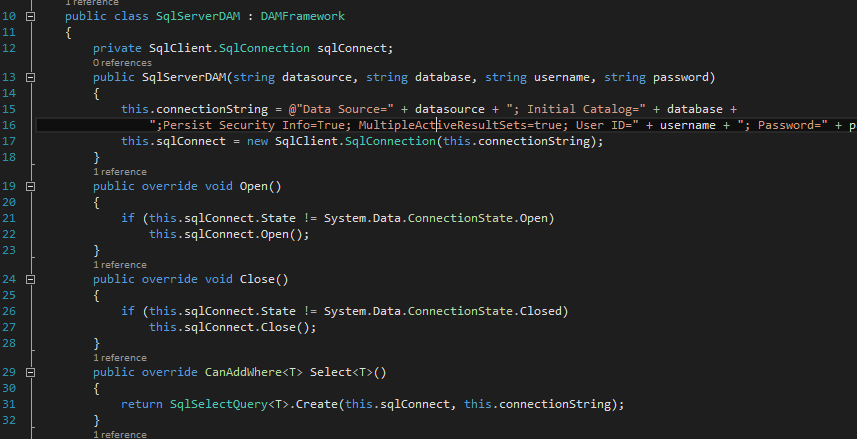
#### CanAddGroupBy: là interface cho phép thêm điều kiện GroupBy vào câu truy vấn sau khi thêm điều kiện Having

#### CanQuery: là interface cho phép thêm Query vào câu truy vấn sau khi thêm điều kiện GroupBy

# Các mẫu thiết kế đã áp dụng

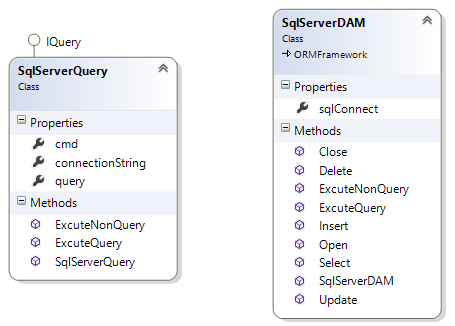
#### Template Method

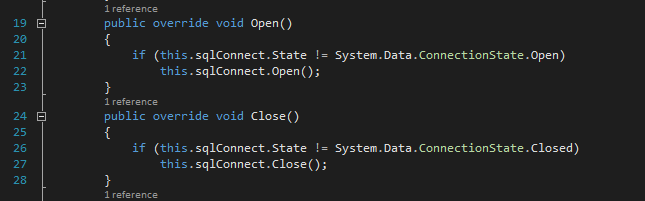




* Ý nghĩa: Kế thừa từ lớp ảo ORMFramework, các phương thức được định nghĩa trước, có thể mở rộng framework bằng cách cài đặt các CSDL (MySql, SqlLite) khác theo một khung có sẵn.

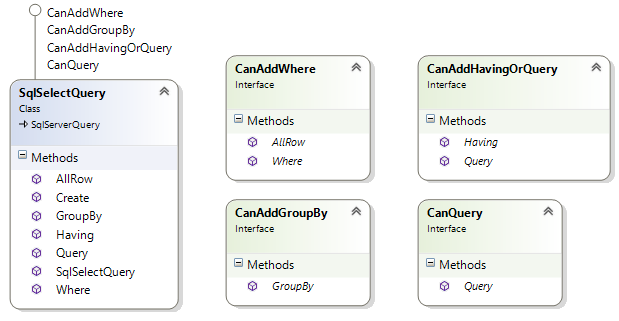
#### Adapter

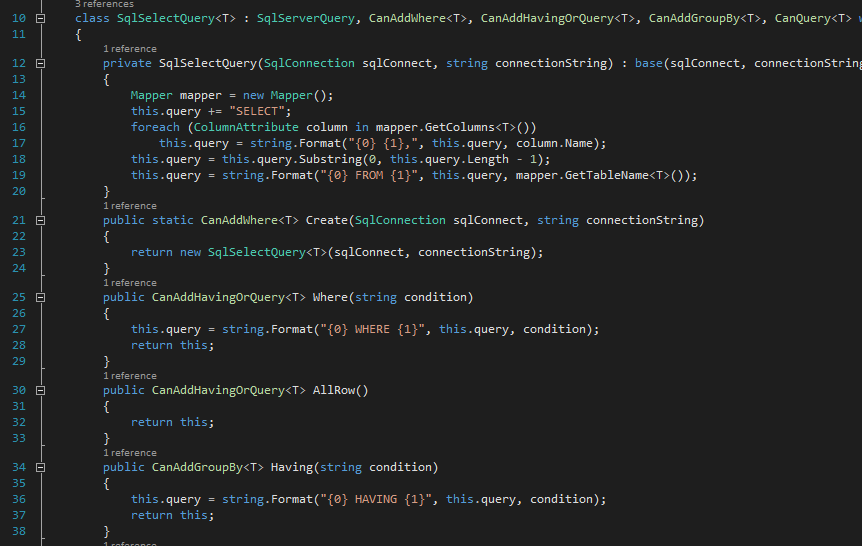




* Ý nghĩa: Chuyển đổi các phương thức trong các lớp (SqlConnection) có sẵn thành giao diện mới để các lớp khác có thể sử dụng được

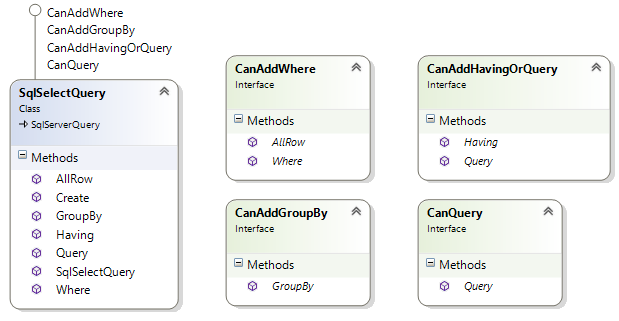
#### Builder

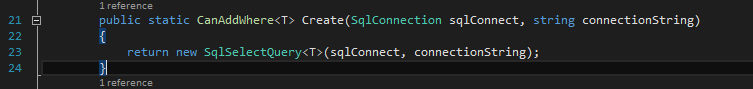




* Ý nghĩa: Sử dụng mẫu Builder cho phép người sử dụng dễ dàng tạo một câu truy vấn dựa theo những quy định có sẵn của các interface.

#### Singleton





* Ý nghĩa: Hàm khởi tạo của lớp SqlSelectQuery là private. Việc cài đặt static cho phép tạo đối tượng này chỉ thông qua hàm Create, không được phép khởi tạo trực tiếp từ bên ngoài.