ĐỒ ÁN CUỐI KÌ

MẪU THIẾT KẾ HƯỚNG ĐỐI TƯỢNG VÀ ỨNG DỤNG

Database Access Management (DAM) Framework

Giáo viên: Nguyễn Minh Huy

Thực hiện: Nhóm 03 - DEST

TP.HCM, ngày 03 tháng 07 năm 2017

**MỤC LỤC**

[I. THÔNG TIN CHUNG 3](#_Toc486881856)

[1. Thông tin nhóm 3](#_Toc486881857)

[2. Sơ lược về đồ án 3](#_Toc486881858)

[II. NỘI DUNG BÁO CÁO 4](#_Toc486881859)

[1. Sơ đồ lớp 4](#_Toc486881860)

[1.1 SCOConnection 5](#_Toc486881861)

[1.2 Mapper 5](#_Toc486881862)

[1.3 SCOSqlConnection 6](#_Toc486881863)

[1.4 SqlMapper 6](#_Toc486881864)

[1.5 IQuery 6](#_Toc486881865)

[1.6 SqlQuery 7](#_Toc486881866)

[1.7 ICanAddWhere 7](#_Toc486881867)

[1.8 ICanAddHavingOrRun 7](#_Toc486881868)

[1.9 ICanAddGroupBy 7](#_Toc486881869)

[1.10 ICanRun 7](#_Toc486881870)

[1.11 SqlSelectQuery 7](#_Toc486881871)

[1.12 SqlInsertQuery 8](#_Toc486881872)

[1.13 SqlUpdateQuery 8](#_Toc486881873)

[1.14 SqlDeleteQuery 8](#_Toc486881874)

[2. Các mẫu thiết kế đã sử dụng 9](#_Toc486881875)

[2.1 Singleton 9](#_Toc486881876)

[2.2 Adapter 10](#_Toc486881877)

[2.3 Builder 11](#_Toc486881878)

[2.4 Template Method 12](#_Toc486881879)

1. **THÔNG TIN CHUNG**
2. **Thông tin nhóm**

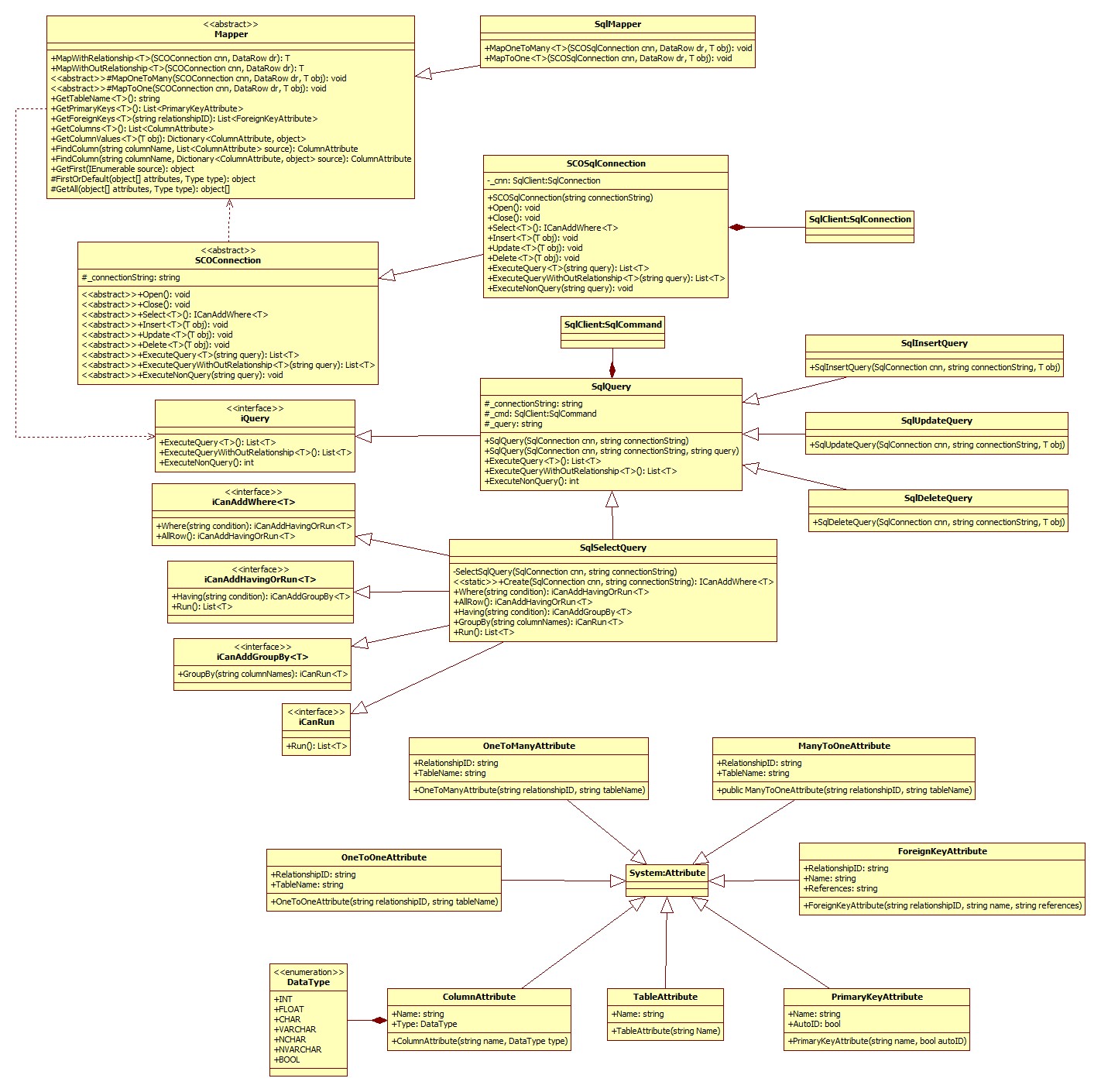
* STT: 03
* Tên nhóm: DEST
* Danh sách thành viên:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | MSSV | Họ và tên | SĐT |
| 1 | 1312211 | Lê Thế Hoàng | 0167 944 7332 |
| 2 | 1312215 | Nguyễn Huy Hoàng |  |
| 3 | 1312254 | Phạm Hoàng Huy | 0128 4884 900 |
| 4 | 1312635 | Nguyễn Đức Trung | 0987 300 571 |

1. **Sơ lược về đồ án**

* Xây dựng framework hỗ trợ ánh xạ dữ liệu quan hệ thành dữ liệu đối tượng, giúp cho việc lập trình tương tác CSDL trở nên dễ dàng hơn.
* Framework hỗ trợ các thao tác cơ bản như: kết nối, đóng kết nối CSDL, insert, update, delete, select dữ liệu.
* Đồng thời, xây dựng các lớp cơ sở để truy xuất dữ liệu bảng dưới dạng đối tượng. Thể hiện quan hệ giữa các bảng bằng quan hệ của các đối tượng.
* Nhóm đã xây dựng framework với tên gọi SCO Framework, được viết bằng ngôn ngữ lập trình C#, sử dụng reflection, attribute, generics để đáp ứng những yêu cầu nêu trên. Chi tiết được trình bày ở phần II.

1. **NỘI DUNG BÁO CÁO**
2. **Sơ đồ lớp**



1. SCOConnection

Là một lớp trừu tượng đại diện cho kết nối CSDL, khai báo các phương thức trừu tượng để các lớp con tự định nghĩa.

|  |  |
| --- | --- |
| Thuộc tính | |
| connectionString | Lưu chuỗi kết nối CSDL. |
| Phương thức | |
| Open() | Mở kết nối CSDL. |
| Close() | Đóng kết nối CSDL. |
| Select<T>() | Tạo đối tượng SqlSelectQuery. |
| Insert<T>(T obj) | Thêm đối tượng obj kiểu T vào CSDL như một dòng dữ liệu trong bảng. |
| Update<T>(T obj) | Cập nhập dòng dữ liệu tương ứng với đối tượng obj. |
| Delete<T>(T obj) | Xóa dòng dữ liệu tương ứng với đối tượng obj. |
| ExecuteQuery<T>(string q) | Thực hiện câu lệnh select được lưu trong chuỗi q. Trả về danh sách đối tượng (nếu có) bao gồm dữ liệu thể hiện quan hệ của các đối tượng (nếu có). |
| ExecuteQueryWithOut  Relationship<T>(string q) | Thực hiện câu lệnh select được lưu trong chuỗi q. Trả về danh sách đối tượng (nếu có) không bao gồm dữ liệu thể hiện quan hệ của các đối tượng. |
| ExecuteNonQuery(string q) | Thực hiện các câu lệnh không trả về dòng dữ liệu như: Insert, Update, Delete. Trả về số lượng dòng thay đổi trong CSDL. |

1. Mapper

Là lớp trừu tượng định nghĩa các phương thức giúp chuyển đổi dữ liệu dạng bảng sang đối tượng. Đồng thời chứa các phương thức làm việc với reflector, attribute như GetTableName, GetPrimaryKeys, GetColumns,…

|  |  |
| --- | --- |
| Phương thức | |
| MapWithRelationship<T>  (SCOConnection cnn, DataRow dr) | Chuyển đổi dữ liệu từ DataRow sang đối tượng, bao gồm dữ liệu thể hiện quan hệ của các đối tượng. Trả về object kiểu T. |
| MapWithOUTRelationship  <T>(SCOConnection cnn, DataRow dr) | Chuyển đổi dữ liệu từ DataRow sang đối tượng, không bao gồm dữ liệu thể hiện quan hệ của các đối tượng. Trả về object kiểu T. |
| MapOneToMany() | Lấy phần dữ liệu thể hiện quan hệ 1-N trong mỗi đối tượng (nếu có). |
| MapToOne() | Lấy phần dữ liệu thể hiện quan hệ \*-1 (1-1 hoặc N-1) trong mỗi đối tượng (nếu có). |
| GetTableName<T>() | Trả về tên bảng (Name) được lưu trong TableAttribute của đối tượng. |
| GetPrimaryKeys<T>() | Trả về danh sách khóa chính (List<PrimaryKeyAttribute>) của class T. |
| GetForeignKeys<T>() | Trả về danh sách khóa ngoại (List<ForeignKeyAttribute>) của class T. |
| GetColumns<T>() | Trả về danh sách cột (List<ColumnAttribute) của class T. |
| GetColumnValues<T>() | Trả về danh sách cột kèm giá trị tương ứng từng cột của đối tượng kiểu T. |
| FindColumn(string columnName, List<ColumnAttribute> source) | Trả về ColumnAttribute trong danh sách theo tên. |
| FindColumn(string columnName, Dictionary  <ColumnAttribute, object> source) | Trả về ColumnAttribute trong danh sách theo tên. |
| GetFirst(IEnumerable source) | Trả về phần tử đầu tiên trong danh sách. |
| FirstOrDefault(object[] attributes, Type type) | Trả về attribute đầu tiên có cùng kiểu với tham số truyền vào. Nếu không có trả về null. |
| GetAll(object[] attributes, Type type) | Trả về danh sách attribute có cùng kiểu với tham số truyền vào. |

1. SCOSqlConnection

Là lớp đối tượng kế thừa từ lớp SCOConnection, định nghĩa lại các phương thức trong lớp cha để xử lý với CSDL SQL Server. Ngoài các thuộc tính, phương thức từ lớp cha, SqlQuery còn có:

|  |  |
| --- | --- |
| Thuộc tính | |
| cnn | Đối tượng SqlClient:SqlConnection. |
| Phương thức | |
| SCOSqlConnection(string connectionString) | Hàm khởi tạo với chuỗi connectionString. |

1. SqlMapper

Là lớp con của Mapper, kế thừa các phương thức của lớp cha và định nghĩa lại 2 phương thức abstract là MapOneToMany và MapToOne ứng với CSDL SQL Server.

1. IQuery

Là interface quy định các phương thức chung nhất của một câu truy vấn.

|  |  |
| --- | --- |
| Phương thức | |
| ExecuteQuery<T>() | Thực hiện câu lệnh truy vấn. Trả về danh sách đối tượng (nếu có) bao gồm dữ liệu thể hiện quan hệ của các đối tượng (nếu có). |
| ExecuteQueryWithOut  Relationship<T>() | Thực hiện câu lệnh truy vấn. Trả về danh sách đối tượng (nếu có) không bao gồm dữ liệu thể hiện quan hệ của các đối tượng. |
| ExecuteNonQuery() | Thực hiện các câu lệnh không trả về dòng dữ liệu như: Insert, Update, Delete. Trả về số lượng dòng thay đổi trong CSDL. |

1. SqlQuery

Cài đặt các phương thức kế thừa từ interface IQuery để cụ thể hóa xử lí cho CSDL SQL Server. Ngoài các thuộc tính, phương thức từ lớp cha, SqlQuery còn có:

|  |  |
| --- | --- |
| Thuộc tính | |
| connectionString | Lưu chuỗi kết nối CSDL. |
| cmd | Đối tượng SqlClient:Command để thực hiện các câu truy vấn SQL Server. |
| query | Câu truy vấn. |
| Phương thức | |
| SqlQuery(SqlConnection cnn, string connectionString) | Hàm khởi tạo với SqlConnection và ConnectionString. |
| SqlQuery(SqlConnection cnn, string connectionString, string query) | Hàm khởi tạo với SqlConnection, ConnectionString và Query. |

1. ICanAddWhere

Là interface cho phép thêm điều kiện Where vào câu truy vấn để chỉ điều kiện select hoặc AllRow() để lấy tất cả các dòng trong bảng.

1. ICanAddHavingOrRun

Là interface cho phép thêm điều kiện Having vào câu truy vấn sau Where hoặc có thể Run() để chạy câu truy vấn.

1. ICanAddGroupBy

Là interface cho phép thêm câu lệnh GroupBy vào câu truy vấn sau khi vừa thêm điều kiện Having.

1. ICanRun

Là interface cho phép thêm Run() câu truy vấn sau khi đã thêm GroupBy.

1. SqlSelectQuery

Là một lớp đại diện cho câu truy vấn Select, kế thừa lớp SqlQuery. Ngoài ra, SqlSelectQuery còn kế thừa các interface IcanAddWhere, ICanAddHavingOrRun, IcanAddGroupBy, IcanRun giúp người dùng có thể tạo được câu truy vấn Select một cách trực quan, dễ dàng hơn mà không cần phải nhớ chính xác cấu trúc một câu lệnh Select phức tạp.

|  |  |
| --- | --- |
| Phương thức | |
| <<private>> SelectSqlQuery(SqlConnection cnn, string connectionString) | Phương thức khởi tạo được giấu đi bằng cách quy định tầm vực là private. |
| <<static>> Create(SqlConnection cnn, string connectionString): ICanAddWhere<T> | Hàm static Create để khởi tạo đối tượng. Hàm này sẽ gọi phương thức khởi tạo. |
| Where(string condition): iCanAddHavingOrRun<T> | Thêm điều kiện cho câu lệnh Select. |
| AllRow(): iCanAddHavingOrRun<T> | Select tất cả các dòng, không chứa câu lệnh Where. |
| Having(string condition): iCanAddGroupBy<T> | Thêm Having cho câu lệnh Select. |
| GroupBy(string columnNames): iCanRun<T> | Thêm GroupBy cho câu lệnh Select. |
| Run(): List<T> | Run câu truy vấn. Trả về danh sách đối tượng được select. |

1. SqlInsertQuery

Là một lớp đại diện cho câu truy vấn Insert, kế thừa lớp SqlQuery. Tự động phát sinh câu lệnh SQL Insert tương ứng với đối tượng được truyền vào.

1. SqlUpdateQuery

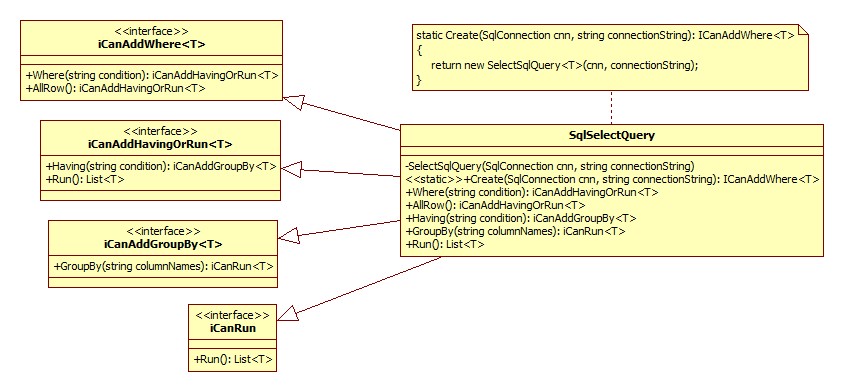
Là một lớp đại diện cho câu truy vấn Update, kế thừa lớp SqlQuery. Tự động phát sinh câu lệnh SQL Update tương ứng với đối tượng được truyền vào.

1. SqlDeleteQuery

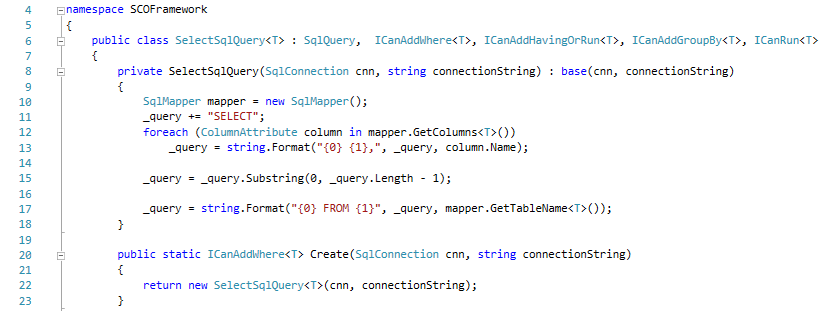
Là một lớp đại diện cho câu truy vấn Delete, kế thừa lớp SqlQuery. Tự động phát sinh câu lệnh SQL Delete tương ứng với đối tượng được truyền vào.

1. **Các mẫu thiết kế đã sử dụng**
2. Singleton

* Sơ đồ lớp



* Đoạn code sử dụng mẫu

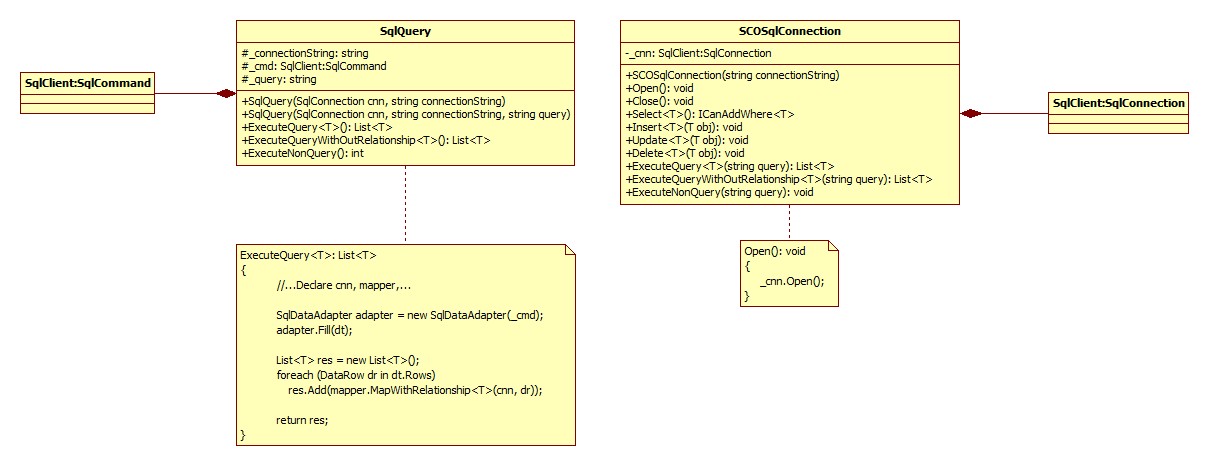


* Ý nghĩa

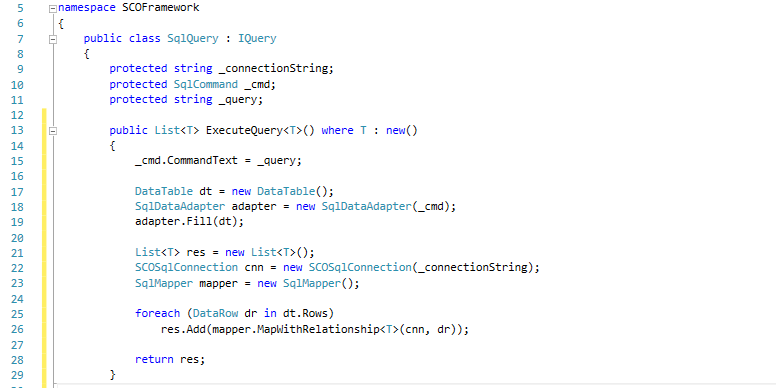
Ngăn chặn khởi tạo đối tượng SelectSqlQuery từ bên ngoài lớp. Cho phép tạo đối tượng thông qua phương thức static Create(). Trả về đối tượng theo interface ICanAddWhere<T> thay vì SelectSqlQuery.

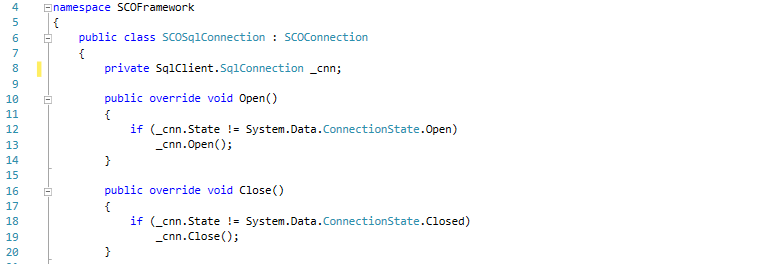
1. Adapter

* Sơ đồ lớp



* Đoạn code sử dụng mẫu



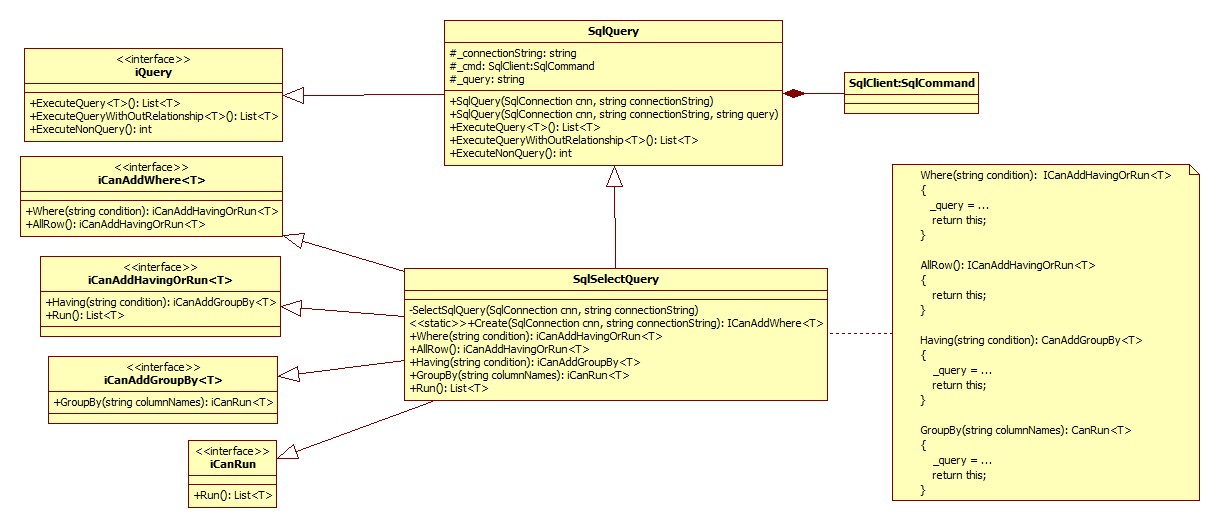


* Ý nghĩa

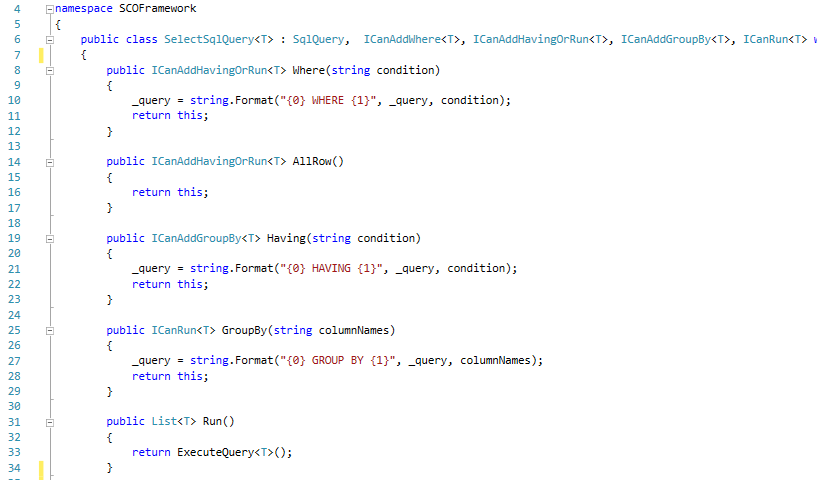
Sử dụng các lớp sẵn có (SqlCommand, SqlConnection), chuyển đổi giao diện để tương thích với những lớp khác trong chương trình.

1. Builder

* Sơ đồ lớp



* Đoạn code sử dụng mẫu

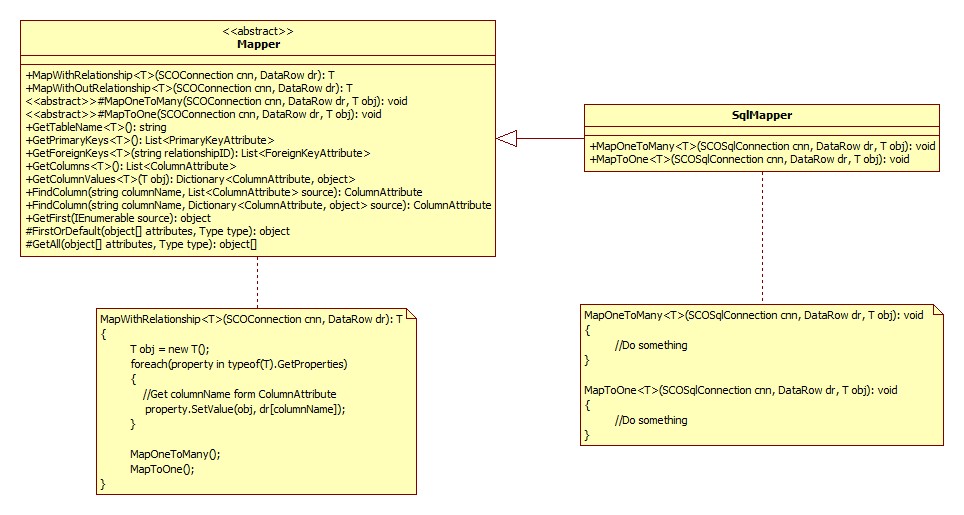


* Ý nghĩa

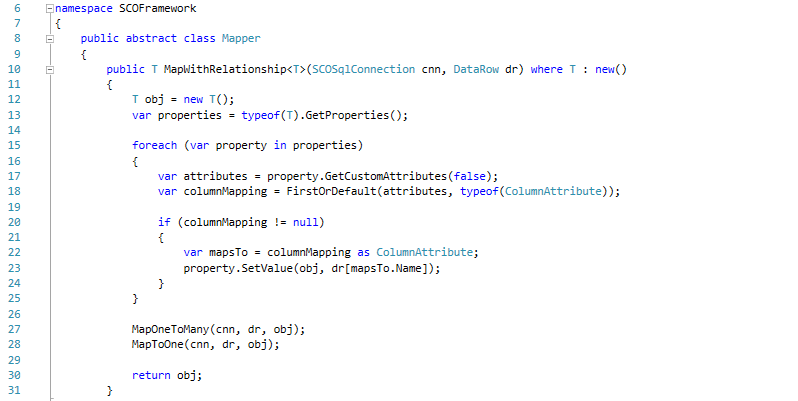
Một câu query SQL khá phức tạp, bao gồm nhiều phần: Select, From, Where, Having, GroupBy,… Việc sử dụng mẫu Builder có thể giúp người dùng không cần phải nhớ chính xác cú pháp của 1 câu query SQL để khởi tạo đối tượng. Ngược lại, đối tượng được khởi tạo một cách dễ dàng qua từng bước (công đoạn), giảm thiểu lỗi cú pháp.

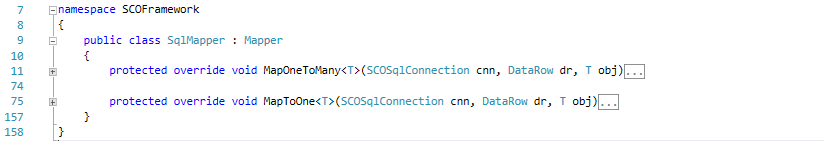
1. Template Method

* Sơ đồ lớp



* Đoạn code sử dụng mẫu



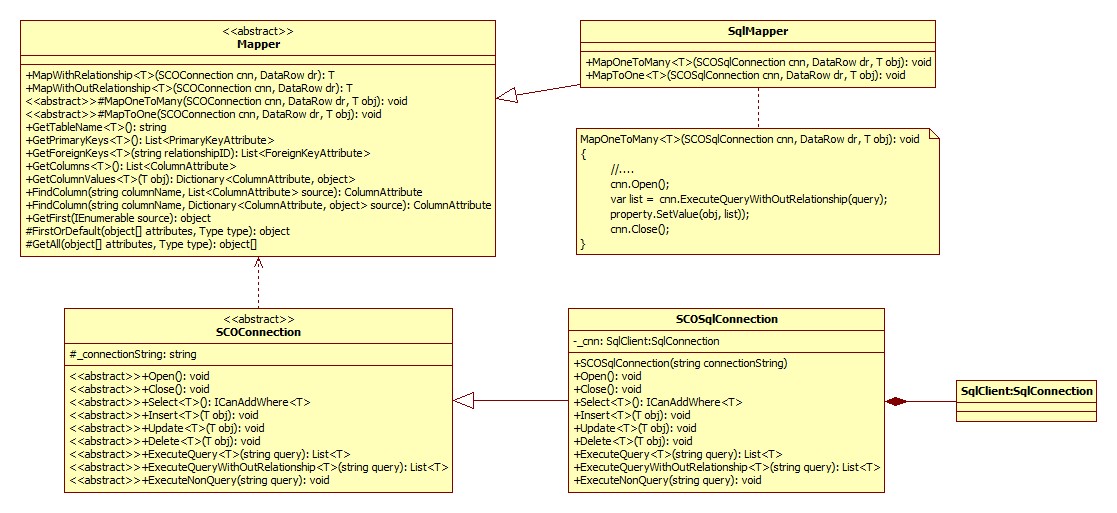


* Ý nghĩa

Định nghĩa một phương thức trong đó có các bước chung cần thực hiện. Một số bước sẽ phụ thuộc vào từng loại lớp con cụ thể. Nhằm giảm sự trùng lắp mã nguồn.

1. Strategy

* Sơ đồ lớp



* Đoạn code sử dụng mẫu



* Ý nghĩa

Trừu tượng hóa xử lý của phương thức MapOneToMany. Cho phép lựa chọn thuật toán trong thời gian thực thi (run-time) phụ thuộc vào đối tượng SCOSqlConnection.

Phương thúc ExecuteQueryWithOutRelationship được định nghĩa khác nhau trong từng class con của SCOSqlConnection.