

Bài 11.2: Các toán tử truy vấn cơ bản

- ✓ Tổng quan
- ✓ Toán tử lọc dữ liệu
- ✓ Toán tử sắp xếp
- ✓ Toán tử gộp nhóm
- ✓ Toán tử nối
- ✓ Toán tử lựa chọn(phép chiếu)

Tổng quan về các toán tử

- ✓ Bài học này sẽ giới thiệu cho bạn các loại toán tử điển hình thường dùng trong câu truy vấn.
- ✓ Bước đầu tiên trong LINQ là xác định nguồn dữ liệu. Trong một truy vấn LINQ, mệnh đề **from** xuất hiện đầu tiên để giới thiệu nguồn dữ liệu và biến vận hành trên nguồn dữ liệu đó.
- ✓ Biến vận hành hoạt động giống một biến lặp trong vòng lặp foreach ngoại trừ việc không xảy ra quá trình lặp trong biểu thức truy vấn.
- ✓ Khi lệnh truy vấn được thực thi, biến vận hành sẽ tham chiếu tới từng phần tử trong nguồn cấp dữ liệu.
- ✓ Trong quá trình truy vấn ta có thể bổ sung thêm các biến khác nếu cần với mệnh đề let

Toán tử lọc dữ liệu

- ✓ Hầu hết các toán tử truy vấn đều áp dụng một bộ lọc dưới dạng biểu thức kiểu Boolean.
- ✓ Bộ lọc làm cho câu lệnh truy vấn trả về chỉ các phần tử thỏa mãn điều kiện đã đưa ra.
- ✓ Việc lọc dữ liệu được thể hiện qua mệnh đề **where**.
- ✓ Ví dụ: tìm các thành phố bắt đầu bằng chữ 'H':

Ví dụ

```
var cities = new List<string>() {  
    "Hà Nội", "Đà Nẵng", "Hồ Chí Minh", "Bình Dương", "Cà Mau",  
    "Nam Định", "Thanh Hóa", "Hà Tĩnh", "Điện Biên", "Lai Châu",  
    "Thái Nguyên", "Hải Phòng", "Bắc Ninh", "Bắc Giang", "Bến Tre",  
    "Bình Định", "Quảng Nam", "Nghệ An", "Hà Nam"  
};  
// biểu thức truy vấn  
var result = from city in cities  
              where city[0] == 'H'  
              select city;  
// thực thi biểu thức truy vấn và hiển thị kết quả  
foreach (var item in result)  
{  
    Console.WriteLine("==> " + item);  
}
```

C:\WINDOWS\system32\cmd.exe

```
==> Hà Nội  
==> Hồ Chí Minh  
==> Hà Tĩnh  
==> Hải Phòng  
==> Hà Nam
```

Toán tử sắp xếp

- ✓ Dữ liệu trước khi trả về có thể được sắp xếp với mệnh đề **orderby**.
- ✓ Trước hết ta chọn một vài tiêu chí để sắp xếp.
- ✓ Nếu muốn sắp xếp tăng dần ta dùng mệnh đề **ascending**.
- ✓ Ngược lại sử dụng mệnh đề **descending**.

Ví dụ

```
var cities = new List<string>() {  
    "Hà Nội", "Đà Nẵng", "Hồ Chí Minh", "Bình Dương", "Cà Mau",  
    "Nam Định", "Thanh Hóa", "Hà Tĩnh", "Điện Biên", "Lai Châu",  
    "Thái Nguyên", "Hải Phòng", "Bắc Ninh", "Bắc Giang", "Bến Tre",  
    "Bình Định", "Quảng Nam", "Nghệ An", "Hà Nam"  
};  
// biểu thức truy vấn có sắp xếp  
var result = from city in cities  
              where city[0] == 'H'  
              orderby city ascending  
              select city;  
// thực thi biểu thức truy vấn và hiển thị kết quả  
foreach (var item in result)  
{  
    Console.WriteLine("==> " + item);  
}
```

C:\WINDOWS\system32\cmd.exe

```
==> Hà Nam  
==> Hà Nội  
==> Hà Tĩnh  
==> Hải Phòng  
==> Hồ Chí Minh
```

Toán tử gộp nhóm

- ✓ Mệnh đề **group** cho phép ta gom nhóm các kết quả trả về dựa trên một từ khóa mà bạn chỉ định.
- ✓ Ví dụ bạn có thể gom nhóm các sinh viên, khách hàng theo tên thành phố họ sinh sống.

Ví dụ

```
var students = new List<Student>()
{
    new Student("B25DCCN100", "Trần Trung Đức", "Hà Nội"),
    new Student("B25DCCN105", "Lê Khả Ngân", "Nam Định"),
    new Student("B25DCCN103", "Hoàng Thanh Mai", "Đà Nẵng"),
    new Student("B25DCCN109", "Nguyễn Thúy Quyên", "Hà Nội"),
    new Student("B25DCCN106", "Nông Tiến Mạnh", "Thái Nguyên"),
    new Student("B25DCCN101", "Hồ Hoài Anh", "Hà Nội"),
    new Student("B25DCCN104", "Trương Thanh Thức", "Hồ Chí Minh"),
};
// biểu thức truy vấn có sắp xếp
var result = from student in students
              group student by student.Address;
// thực thi biểu thức truy vấn và hiển thị kết quả
foreach (var group in result)
{
    Console.WriteLine($"==> {group.Key}:");
    foreach (var student in group)
    {
        Console.WriteLine(student);
    }
}
```

==> Hà Nội:

Student[B25DCCN100, Trần Trung Đức, Hà Nội]

Student[B25DCCN109, Nguyễn Thúy Quyên, Hà Nội]

Student[B25DCCN101, Hồ Hoài Anh, Hà Nội]

==> Nam Định:

Student[B25DCCN105, Lê Khả Ngân, Nam Định]

==> Đà Nẵng:

Student[B25DCCN103, Hoàng Thanh Mai, Đà Nẵng]

==> Thái Nguyên:

Student[B25DCCN106, Nông Tiến Mạnh, Thái Nguyên]

==> Hồ Chí Minh:

Student[B25DCCN104, Trương Thanh Thức, Hồ Chí Minh]

Toán tử nối

- ✓ Mệnh đề **join** cho phép ta tạo liên kết giữa các tập đầu vào không được mô hình hóa tường minh trong nguồn dữ liệu.
- ✓ Mệnh đề **join** luôn hoạt động dựa trên tập hợp các đối tượng thay vì trực tiếp trong CSDL.
- ✓ Thường ta ít khi phải nối hai tập hợp vì các biểu diễn dữ liệu trong đối tượng C# thường chứa tập hợp các đối tượng tham chiếu đến khóa ngoại của nó và cho phép truy cập thông qua tham chiếu.
- ✓ Ví dụ một Customer object có thể chứa một List<Order> các đơn hàng. Từ đó có thể truy cập qua cú pháp : *from order in Customer.Orders...*

Ví dụ

```
{
    new Student("B25DCCN100", "Trần Trung Đức", "Hà Nội"),
    new Student("B25DCCN105", "Lê Khả Ngân", "Nam Định"),
    new Student("B25DCCN103", "Hoàng Thanh Mai", "Đà Nẵng"),
    new Student("B25DCCN109", "Nguyễn Thúy Quyên", "Hà Nội"),
    new Student("B25DCCN106", "Nguyễn Tiến Mạnh", "Thái Nguyên"),
    new Student("B25DCCN101", "Hồ Hoài Anh", "Hà Nội"),
    new Student("B25DCCN104", "Trương Thanh Thức", "Hồ Chí Minh"),
};

var registers = new List<RegisteredSubject>() {
    new RegisteredSubject("B25DCCN100", "Java", DateTime.Now),
    new RegisteredSubject("B25DCCN100", "C#", DateTime.Now),
    new RegisteredSubject("B25DCCN100", "SQL", DateTime.Now),
    new RegisteredSubject("B25DCCN105", "Android", DateTime.Now),
    new RegisteredSubject("B25DCCN104", "C#", DateTime.Now),
    new RegisteredSubject("B25DCCN101", "C#", DateTime.Now),
};
// biểu thức truy vấn có sắp xếp
var result = from student in students
              join register in registers
              on student.Id equals register.StudentId
              select new { student.Id, student.FullName, register.SubjectName };
// thực thi biểu thức truy vấn và hiển thị kết quả
foreach (var item in result)
{
    Console.WriteLine($"{item.Id} - {item.FullName}: {item.SubjectName}");
}
```

Ví dụ

```
{
    new Student("B25DCCN100", "Trần Trung Đức", "Hà Nội"),
    new Student("B25DCCN105", "Lê Khả Ngân", "Nam Định"),
    new Student("B25DCCN103", "Hoàng Thanh Mai", "Đà Nẵng"),
    new Student("B25DCCN109", "Nguyễn Thúy Quyên", "Hà Nội"),
    new Student("B25DCCN106", "Nông Tiến Mạnh", "Thái Nguyên"),
    new Student("B25DCCN101", "Hồ Hoài Anh", "Hà Nội"),
    new Student("B25DCCN104", "Trương Thanh Thức", "Hồ Chí Minh"),
};

var registers = new List<RegisteredSubject>() {
    new RegisteredSubject("B25DCCN100", "Java", DateTime.Now),
    new RegisteredSubject("B25DCCN100", "C#", DateTime.Now),
    new RegisteredSubject("B25DCCN100", "SQL", DateTime.Now),
    new RegisteredSubject("B25DCCN105", "Android", DateTime.Now),
    new RegisteredSubject("B25DCCN104", "C#", DateTime.Now),
    new RegisteredSubject("B25DCCN101", "C#", DateTime.Now),
};
// biểu thức truy vấn có sắp xếp
var result = from student in students
              join register in registers
              on student.Id equals register.StudentId
              select new { student.Id, student.FullName, register.SubjectName };
// thực thi biểu thức truy vấn và hiển thị kết quả
foreach (var item in result)
{
    Console.WriteLine($"{item.Id} - {item.FullName}: {item.SubjectName}");
}
```

```
B25DCCN100 - Trần Trung Đức: Java
B25DCCN100 - Trần Trung Đức: C#
B25DCCN100 - Trần Trung Đức: SQL
B25DCCN105 - Lê Khả Ngân: Android
B25DCCN101 - Hồ Hoài Anh: C#
B25DCCN104 - Trương Thanh Thức: C#
```

Toán tử lựa chọn

- ✓ Mệnh đề **select** cho phép ta lấy kết quả của một câu truy vấn và xác định các thành phần sẽ bao gồm trong kết quả trả về.
- ✓ Ta có thể chọn 1 hoặc nhiều giá trị trong số các thuộc tính cũ đã có.
- ✓ Ta có thể tạo ra các giá trị mới, đối tượng mới từ dữ liệu đã có.

```
var students = new List<Student>()
{
    new Student("B25DCCN100", "Trần Trung Đức", "Hà Nội"),
    new Student("B25DCCN105", "Lê Khả Ngân", "Nam Định"),
    new Student("B25DCCN103", "Hoàng Thanh Mai", "Đà Nẵng"),
    new Student("B25DCCN109", "Nguyễn Thúy Quyên", "Hà Nội"),
    new Student("B25DCCN106", "Nông Tiến Mạnh", "Thái Nguyên"),
    new Student("B25DCCN101", "Hồ Hoài Anh", "Hà Nội"),
    new Student("B25DCCN104", "Trương Thanh Thức", "Hồ Chí Minh")
};
// câu lệnh truy vấn
var studentNameQuery = from student in students
                        select student.FullName;
// thực thi và hiện kết quả
Console.WriteLine("==> Danh sách sinh viên:");
foreach (var item in studentNameQuery)
{
    Console.WriteLine(item);
}
```



Nội dung tiếp theo

Chuyển đổi dữ liệu trong LINQ