LINQ

1. Định nghĩa về LinQ:

Để giảm gánh nặng thao tác trên nhiều ngôn ngữ khác nhau và cải thiện năng suất lập trình, Microsoft đã phát triển giải pháp tích hợp dữ liệu cho .NET Framework có tên gọi là LINQ (Language Integrated Query), đây là thư viện mở rộng cho các ngôn ngữ lập trình C# và Visual Basic.NET (có thể mở rộng cho các ngôn ngữ khác) cung cấp khả năng truy vấn trực tiếp dữ liệu Object, CSDL và XML.

LINQ là một tập hợp các thành phần mở rộng cho phép viết các câu truy vấn dữ liệu ngay trong một ngôn ngữ lập trình, như C# hoặc VB.NET. Khi tạo một đối tượng LINQ thì Visual Studio sẽ tự động sinh ra các lớp có các thành phần tương ứng với CSDL của chúng ta. Khi muốn truy vấn, làm việc với CSDL ta chỉ việc gọi và truy xuất các hàm, thủ tục tương ứng của LINQ mà không cần quan tâm đến các câu lệnh SQL thông thường.

Điểm mạnh (chưa chắc về độ mạnh, nhưng hay) của LINQ là “viết truy vấn cho rất nhiều các đối tượng dữ liệu”. Từ CSDL, XML Data Object … thậm chí là viết truy vấn cho một biến mảng đã tạo ra trước đó. Vì vậy mới có các khái niệm LinQ to SQL, LinQ to XML,..

Tuy nhiên so với mô hình Entity (Entity Framework), LINQ có yếu điểm là chậm và thiếu nhất quán.

LINQ có từ bản .NET 3.5, vậy nên tối thiểu chương trình của bạn phải chạy trên nền tảng này.

Visual Studio 2008, hoặc các phiên bản Express của nó là các bộ công cụ phát triển tiêu biểu cho ứng dụng dùng LINQ.

2. Truy vấn với LinQ

2.1 Cú pháp truy vấn LINQ

Cú pháp truy vấn tương tự như truy vấn SQL (ngôn ngữ truy vấn có cấu trúc) cho cơ sở dữ liệu. Nó có cú pháp trong mã C# hoặc VB.NET như sau:

|  |
| --- |
| from <range variable> in <Ienumerable<T> or Iqueryable<T> Collection>  <Standard Query Operators> <lambda expression>  <select ỏ groupBy operator> <result formation> |

Cú pháp truy vấn LINQ bắt đầu từ khóa from và kết thúc bằng từ khóa select. Điều này ngược với cú pháp truy vấn SQL bắt đầu bằng từ khóa select.

Lý do bắt đầu bằng từ khóa from là IntelliSense sẽ biết trước được kiểu dữ liệu để hỗ trợ bạn gõ biểu thức truy vấn nhanh chóng.

Ví dụ dưới đây minh họa một biểu thức truy vấn LINQ trả về các chuỗi có chứa từ "Tutorials":

|  |
| --- |
| Ilist<string> stringList = new List<string>()  {  “C# tutorial”,  “VB.NET tutorials”,  “Learn C++”,  “MVC Tutorials”,  “Java”  }  var result = from s in stringList  Where s.Contains(“Tutorials”  Select s;  )  foreach(var item in result)  {  Console.WriteLine(item);  } |

Cú pháp truy vấn bắt đầu bằng mệnh đề from theo sau là biến range. Từ khóa from có cấu trúc from s in strList có nghĩa là từ mỗi phần tử trong danh sách. Nó tương tự như một vòng lặp foreach

Sau mệnh đề from, bạn có thể sử dụng các toán tử truy vấn tiêu chuẩn khác nhau để lọc, gom nhóm, nối các phần tử của danh sách. Có khoảng 50 toán tử truy vấn tiêu chuẩn có sẵn trong LINQ.

Cú pháp truy vấn LINQ luôn kết thúc bằng mệnh đề select hoặc group by. Mệnh đề select được sử dụng để định hình dữ liệu. Bạn có thể chọn toàn bộ đối tượng hoặc chỉ một soos thuộc tính của nó để trả về.

Trong ví dụ dưới đây, tôi sử dụng cú pháp truy vấn LINQ để tìm ra các sinh viên có độ tuổi từ 13 đến 19 từ danh sách sinh viên.

|  |
| --- |
| Ilist<Student> studentList = new List<Student>()  {  New Student() { StudentID = 1, StudentName = “John”, Age = 13},  New Student() { StudentID = 2, StudentName = “Moin”, Age = 21},  New Student() { StudentID = 3, StudentName = “Bill”, Age = 18},  New Student() { StudentID = 4, StudentName = “Ram”, Age = 20},  New Student() { StudentID = 5, StudentName = “Ron”, Age = 15},  };  var result = (from s in studentList where s.Age >12 && s.Age <20 select s).ToList(); |

2.2 Những điểm cần nhớ của cú pháp truy vấn LINQ

Cú pháp truy vấn giống như cú pháp SQL.

Cú pháp truy vấn bắt đầu bằng mệnh đề from và có kết thúc bằng mệnh đề select hoặc group by.

Sử dụng các thành phần khác như lọc, nối, nhóm, sắp xếp toán tử để xây dựng kết quả mong muốn.

Biến định kiểu ngầm định – var có thể được sử dụng để lưu trữ kết quả của truy vấn LINQ.

2.3 Cú pháp phương thức LINQ

Cú pháp phương thức LINQ sử dụng các phương thức mở rộng có trong hai lớp tĩnh là Enumerable hoặc Queryable, tương tự như cách bạn sẽ gọi các phương thức mở rọng của một lớp bất kỳ.

Ví dụ dưới đây minh họa một truy vấn sử dunjg cú pháp phương thức LINQ trả về những chuỗi có chứa một từ “Tutorials”.

|  |
| --- |
| Ilist<string> stringList = new List<string>()  {  “C# tutorials”,  “VB.NET tutorials”,  “Learn C++”,  “MVC tutorials”,  “Java”  };  var result = stringList.where(s => s.Contains(“Tutorilas”)).ToList(); |

Ví dụ sau đây minh họa cách sử dụng cú pháp phương thức LINQ với danh sách Ienumerable<T>.

|  |
| --- |
| Ilist<Student> studentList = new List<Student>()  {  New Student() { StudentID = 1, StudentName = “John”, Age = 13},  New Student() { StudentID = 2, StudentName = “Moin”, Age = 21},  New Student() { StudentID = 3, StudentName = “Bill”, Age = 18},  New Student() { StudentID = 4, StudentName = “Ram”, Age = 20},  New Student() { StudentID = 5, StudentName = “Ron”, Age = 15},  };  Var result = studentList.where(s => s.Age > 12 && s.Age <20).ToList(); |

2.4 Những điểm cần nhớ của cú pháp phương thức LINQ.

Cú pháp phương thức LINQ giống như gọi phương thức mở rộng.

Cú pháp phương thức LINQ còn gọi là cú pháp thông thạo vì nó cho phép thực hiện các phương thức mở rộng.

Biến được định kiểu ngầm đinh – var có thể được sử dụng để lưu trữ kết quả truy vấn LINQ.