**Thông tin nhóm 17:**

1. **Nguyễn Văn Nguyện – 13520567**
2. **Huỳnh Ngọc Thắng – 13520775**
3. **Trần Huy Thịnh – 13520830**
4. **Lê Hoàng Sinh – 13520703**

**Đề tài: TÌM HIỂU VỀ LINQ**

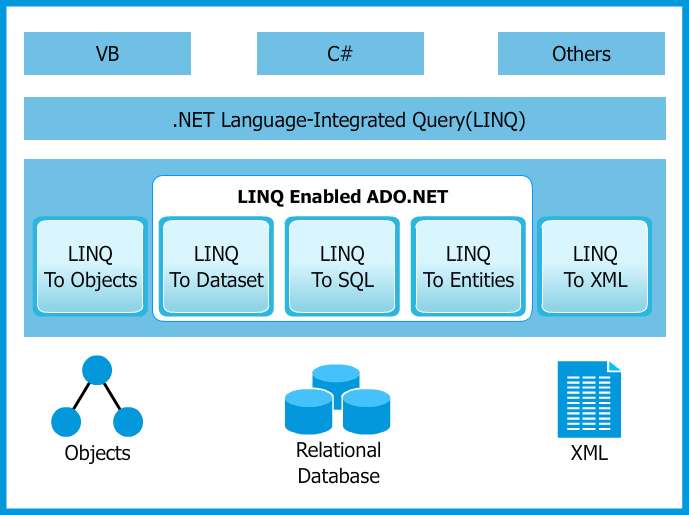
**(Dự kiến)**

1. **TỔNG QUAN VỀ LINQ**
   1. **Định nghĩa , hoàn cảnh ra đời**
      1. ***Định nghĩa***

* Ngôn ngữ tích hợp truy vấn (Language Integrated Query: LINQ) được Microsoft thêm vào .NET Framework để làm việc với dữ liệu (ví dụ: danh sách đối tượng trong bộ nhớ, database, XML) theo cách đơn giản và trực quan nhất. LINQ cung cấp tầng lập trình trừu tượng giữa các ngôn ngữ .NET với dữ liệu.
* LINQ cho phép các lập trình viên truy vấn dữ liệu sử dụng cú pháp tương tự ngôn ngữ C# hoặc VB.NET, không phụ thuộc vào dữ liệu nguồn. Điều đó có nghĩa là một lập trình viên có thể làm việc được với dữ liệu là database hoặc XML cho dù người đó không biết ngôn ngữ T-SQL hoặc cách truy vấn XPath, XQuery để lấy dữ liệu trong XML.
  + 1. ***Hoàn cảnh ra đời***
* Khi sự phát triển của công nghệ lập trình hướng đối tượng (OO) đến điểm ổn định, mọi developer đều rất quen với ngôn ngữ lập trình hướng đối tượng. Tuy nhiên trong các ngôn ngữ truy vấn dữ liệu hiện nay như SQL, Xpath,XQuery rất khác biệt so với ngôn ngữ hướng đối tượng gây khó khăn cho việc xử lý dữ liệu.
* Trong 1 dự án thì có nhiều ngôn ngữ truy vấn dữ liệu. Lập trình viên muốn nắm được hết các ngôn ngữ này là điều phức tạp vì chúng khác xa nhau
* Để giải quyết sự phức tạp và khó khăn đó, Linq đã ra đời.
  1. **Cú pháp**
* *Lambda (Method)*
* var longWords = words.Where( w => w.length > 10);
* Dim longWords = words.Where(Function(w) w.length > 10)
* *Query (Comprehension)*
* var longwords = from w in words where w.length > 10;
* Dim longwords = from w in words where w.length > 10
  1. **Loại**
* LinQ to Objects: làm việc với dữ liệu trên bộ nhớ (thường là các danh sách kế thừa interface IEnumerable) sử dụng cú pháp ngôn ngữ LINQ.
* LinQ to XML (XLINQ): làm việc với dữ liệu XML sử dụng cú pháp ngôn ngữ LINQ.
* LinQ to Dataset: Một kỹ thuật cho phép sử dụng LINQ trong một ứng dụng dựa trên DataSet mà không cần viết lại sử dụng LINQ to SQL.
* LinQ to SQL (DLINQ):  Làm việc với SQL Server.
* LinQ to Entities: làm việc với Entity Framework.

Ngoài những loại được kể trên thì LinQ còn có một loại tên PLINQ – hỗ trợ sử dụng LINQ trong lập trình song song của Microsoft

* 1. **Kiến trúc Linq**



* 1. *LinQ to Objects*
     1. Cung cấp các khả năng truy vấn đến bất cứ đối tượng nào trong bộ nhớ C# như mảng, danh sách, ... hoặc các đối tượng kiểu IEnumerable Collection
     2. IEnumerable dùng để duyệt (một chiều) các phần tử trong danh sách. Nếu danh sách của bạn có hỗ trợ IEnumerable thì bạn có thể sử dụng foreach để duyệt
  2. *LinQ to Dataset*
     1. Cung cấp một cách nhanh chóng và dễ dàng hơn để truy vấn dữ liệu được lưu giữ trong đối tượng DataSet.
     2. Dataset trong ADO.NET là một bước phát triển lớn trong việc phát triển ứng dụng cơ sở dữ liệu đa hệ. Khi lấy và chỉnh sửa dữ liệu, duy trì liên tục kết nối với Data Source trong khi chờ user yêu cầu thì rõ ràng là tốn tài nguyên máy rất nhiều.
     3. Dataset giúp ích ở đây rất lớn. Vì Dataset cho phép lưu trữ dữ liệu và chỉnh sửa tại “Local cache” hay gọi là Offline Mode. Có thể xem xét và xử lí thông tin trong khi kết nối. Sau khi chỉnh sửa và xem xong thì tạo 1 kết nối và update dữ liệu từ local vào Data source
     4. Dữ liệu dataset được lưu trữ dưới dạng một Collection các Tables và bạn cần phải xử lí thông qua các lớp DataTable(DataRow và DataColunm)
     5. Cho phép người lập trình sử dụng Datasets như một nguồn dữ liệu bình thường bằng các cú pháp truy vấn cơ bản của Linq
  3. *LinQ to SQL*
     1. LINQ to SQL là một phiên bản hiện thực hóa của O/RM (object relational mapping) có bên trong .NET Framework 3.5, nó cho phép bạn mô hình hóa một cơ sở dữ liệu dùng các lớp .NET. Sau đó bạn có thể truy vấn cơ sở dữ liệu dùng LINQ, cũng như cập nhật/thêm/xóa dữ liệu từ đó.
     2. LINQ to SQL hỗ trợ đầy đủ transaction, view và các stored procedure (SP). Nó cũng cung cấp một cách dễ dàng để thêm khả năng kiểm tra tính hợp lệ của dữ liệu và các quy tắc vào trong mô hình dữ liệu của bạn.
  4. *LinQ to Entities*
     1. Entity Framework như giản đồ của các dữ liệu được lưu trữ trong cơ sở dữ liệu và trình bày giản đồ khái niệm ứng dụng của nó. Giản đồ này được ánh xạ từ cơ sỡ dữ liệu.
     2. LINQ to Entities cho phép chúng ta có thể truy vấn các thực thể bên trong Entity Framework.
     3. Entity Framework là cơ sở dữ liệu độc lập và xây dựng dựa trên một mô hình chuẩn ADO.NET cho phép truy cập vào cơ sở dữ liệu của bên thứ ba thông qua việc sử dụng ADO.NET.
  5. *LinQ to XML*
     1. Cho phép truy vấn, thay đổi cũng như việc lưu trữ sau khi thay đổi các tài liệu XML. Việc sử dụng LINQ để làm việc với các tài liệu XML sẽ có những thuận lợi và nhiều ưu điểm hơn so với sử dụng các lớp XmlDocument, XPath và XQuery.
  6. **Biểu thức truy vấn**

Biểu thức truy vấn được cung cấp từ phiên bản C# 3.0 dùng để tạo các truy vấn ngắn gọn và dễ hiểu. Trình biên dịch sẽ chuyển các biểu thức truy vấn về dạng phương thức mở rộng lúc biên dịch.

***Chú ý:****Cách sắp xếp các từ khóa của biểu thức truy vấn khác với SQL. Cách sắp xếp From-Where-Select cho phép Visual Studio hỗ trợ đầy đủ Intellisense khi viết mã. Điều này giúp bạn viết mã nhanh và chính xác.*

**5.**1 Tạo biến cục bộ

**5.**2 Mở rộng truy vấn

**5.**3

* 1. **Phương pháp mở rộng**
  2. **So sánh Linq và ADO.NET**
  3. **So sánh Linq và SQL**
  4. **Sự khác biệt giữa Linq và Stored Procedure**

**Stored Procedure** có tốc độ thực thi nhanh nhất vì nó gần với dữ liệu trên CSDL nhất.

**LINQ** là thư viện cho phép truy vấn trên nhiều kiểu dữ liệu khác nhau: Object collections, XML, LINQ to SQL (truy vấn cơ sở dữ liệu MS-SQL), LINQ to Dataset, LINQ to Entity (ADO.net Entity Framework), LINQ to SharePoint, LINQ to XML, LINQ to nHIbernate, LINQ to XYZ, LINQ to Azure SQL Service… Vì vậy tính mở rộng rất cao.

Tóm lại:

SP chạy nhanh, nhưng tính mở rộng kém, khả năng bảo trì kém.

Linq thì là một cú pháp lập trình chung nên tính mở rộng rất cao. Nhưng tốc độ không bằng SP.

**So sánh về việc tính ứng dụng của Linq và SP**

Khi chúng ta viết ứng dụng Client-Server, khá thuần về công nghệ, ít có sự tương tác với nhiều nguồn dữ liệu khác nhau. Ứng dụng logic có khả năng bảo trì được trong giới hạn cho phép. Phần cứng máy chủ hạn chế, viết Stored Proc để tối ưu tốc độ.

Nên viết LINQ khi hệ thống phức tạp, việc bảo trì theo thời gian là rất lớn, Viết LINQ hợp lý sẽ chuyển nhiều logic từ server xử lý lên business layer. Giờ đây xu hướng, chúng ta không viết LINQ to SQL thuần túy nữa, mà chúng ta nên sử dụng ADO.net Entity Framework, EF4.

* 1. **Những điều cần cho Linq**
  2. **Ưu điểm của Linq**
* *Chỉ cần nhớ một ngôn ngữ truy vấn*

Đây là lợi thế đầu tiên mà LINQ cung cấp cho các lập trình viên. Một khi bạn đã học được các toán tử truy vấn chuẩn của LINQ được cung cấp trong ngôn ngữ C# hoặc VB.NET, bạn có thể truy cập bất kỳ dữ liệu nào mà LINQ hỗ trợ mà không cần phải học cú pháp ngôn ngữ truy vấn dữ liệu đó

* *Kiểm tra kiểu dữ liệu và tên lúc biên dịch*

Các truy vấn LINQ được kiểm tra kiểu dữ liệu và tên lúc biên dịch, giảm thiểu các lỗi xuất hiện lúc thực thi. Nhiều ngôn ngữ-T-SQL chẳng hạn, nhúng câu truy vấn vào chuỗi. Điều này làm trình biên dịch khó phát hiện ra lỗi và lỗi có thể xảy ra lúc thực thi. Nhiều lỗi về kiểu dữ liệu và thiếu kiểu dữ liệu cho các trường dữ liệu sẽ được phát hiện lúc biên dịch và sẽ được chỉnh sửa tại thời điểm đó

* *Dễ đọc mã*

Cú pháp của LINQ rất đơn giản do nó đã xóa bỏ đi các các đoạn mã vòng lặp, sắp xếp, gom nhóm, điều kiện phức tạp

* *Trên 50 toán tử truy vấn chuẩn*

Các toán tử truy vấn chuẩn được xây dựng sẵn cho phép dễ dàng thực hiện các công việc như gom nhóm, sắp xếp, liên kết, tập hợp, lọc, hoặc lấy dữ liệu.

|  |  |
| --- | --- |
| **Loại toán tử** | **Tên các toán tử truy vấn chuẩn** |
| Aggregation | Aggregate, Average, Count, LongCount, Max, Min, Sum |
| Conversion | AsEnumerable, Cast, OfType, ToArray, ToDictionary, ToList, ToLookup |
| Element | DefaultIfEmpty, ElementAt, ElementAtOrDefault, First,  FirstOrDefault, Last, LastOrDefault, Single, SingleOrDefault |
| Equality | SequenceEqual |
| Generation | Empty, Range, Repeat |
| Grouping | GroupBy, ToLookup |
| Joining | GroupJoin, Join |
| Merging | Zip |
| Ordering | OrderBy, ThenBy, OrderByDescending, ThenByDescending, Reverse |
| Projection | Select, SelectMany |
| Partitioning | Skip, SkipWhile, Take, TakeWhile |
| Quantifiers | All, Any, Contains |
| Restriction | Distinct, Where |
| Set | Concat, Except, Intersect, Union |

Nhiều toán tử truy vấn có thể được tìm thấy trong các ngôn ngữ truy vấn cơ sở dữ liệu. Nếu bạn đã từng làm việc với các ngôn ngữ truy vấn dữ liệu thì bạn có thể dễ dàng phán đoán ra chức năng của các toán tử đó. Tuy nhiên cũng có vài toàn tử hoàn toàn mới, được thêm vào để làm việc với dữ liệu để làm đơn giản hóa các đoạn mã truyền thống vốn phức tạp và dài dòng vào trong 1 dòng lệnh.

* *Kiến trúc mở và có thể mở rộng*

LINQ được thiết kế có khả năng mở rộng. Có nghĩa là có thể thêm vào các toán tử khi cần thiết

1. **MÔI TRƯỜNG CÀI ĐẶT**
2. **LINQ – OPERATOR**
   1. Filter
   2. Join
   3. Projection
   4. Sorting
   5. Grouping
   6. Conversions (Chuyển đổi trong Linq)
   7. Aggregation , Generation
   8. Set Operations
   9. Equality
   10. Element Operators
3. **LINQ TO SQL**
   1. Giới thiệu Linq to SQL
   2. Cách sử dụng Linq to SQL
   3. Truy vấn với Linq to SQL
   4. Insert, Update, Delete với Linq to SQL
4. **LINQ TO ỌBJECT**
   1. Giới thiệu
   2. Query in Memory Collections
5. **LINQ TO DATASET**
   1. Giới thiệu
   2. Truy vấn Dataset sử dụng Linq to Dataset
6. **LINQ TO XML**
   1. Giới thiệu Linq to XML
   2. Đọc file XML sử dụng Linq
   3. Thêm mới Node
   4. Xóa Node
7. **LINQ TO ENTITIES QUERY CREATION AND EXCUTION PROCESS**
   1. Giới thiệu
   2. Query Creation and Excution Process
   3. Ví dụ ADD, UPDATE, DELETE sử dụng Linq với Entity Model
8. **LINQ – LAMBDA EXPRESSTION**
9. **LINQ DATASOURCE CONTROL**
   1. Giới thiệu
   2. LinqDatasource Control
   3. Cách INSERT, UPDATE, DELETE data trong ASP.NET sử dụng Linq
10. **DEMO**