An Introduction to Language INtegrated Query

By
Praseed Pai K.T.

(Presentation for “Microsoft Community Tech Days”
@ Technopark, Trivandrum on 24th October, 2009)

What is LINQ ?

LINQ is a Query subsystem integrated into the C# and VB.net programming languages to query data in a uniform manner against Object Hierarchies , Databases , XML and any kind of structured data with a provider (say Email , News etc).

Why bother ?

LINQ looks like a poor cousin of SQL on the surface. Beneath the Hype there is indeed substance.

Well Query , then What else ?

One can mutate the underlying data source as well. The data source is updatable !

Why LINQ ?

- Programming Languages are Procedural
- Query Languages are declarative
- The State of the Art before LINQ was Query creation using “stringification”.
- The method is not type safe
- Solution :- Make Query System integrated with the Language Type mechanism.

An Example Query

```
SQLAccess sq = new SQLAccess(DbConfig.GetConStr("MAINDB"));
        string qry = "select pas.s_desc,total from "+
           "(select j_code ,sum( case  j_drcr "+
                          "when 'DR' then "+
                           "j_amount "+
                           "else "+
                           "-j_amount end) as total from JournalDetail" +
                           " group by j_code) test , "+
               "FaSubgroup pas where test.j_code = pas.s_code";
        DataSet ds = sq.Execute(qry);
        return ds.Tables[0];
 }
```

Anatomy of a Query Language

- Relational Algebra (E F CODD)
- Cartesian Product
- Projection
- Rename
- Filter (Where)
- SET OPERATIONS (union,intersect,difference)
- JOIN
- Outer Join

A DEMO

Querying a relational database using LINQ.

LINQ – Three Syntax

- Lambda Syntax
- Comprehension Syntax
- Mixed Mode Syntax

Demos

- Hello World - A LINQ program using Lambda , Comprehension and Mixed Mode Syntax.
- A LINQ program which demonstrates the use of Sub Query using Lambda , Comprehension and Mixed Mode Syntax.

Lambda – What the heck is it ?

- Lambda Syntax is based on Lambda Calculus
- Lambda Calculus was invented way before a physical computer was engineered
- Alonzo Church , was trying to solve Hilbert's 10th problem
- All Functional Programming Languages are based on Lambda Calculus
- Some good examples are Scheme , F#
- Scheme uses untyped Lambda Calculus
- F# uses Typed Lambda Calculus (ML , OCCAML LINEAGE)

Alonzo Church – The Inventor of Lambda Calculus

Demos

- Examples of Lambda

A case study

MAPREDUCE

Lambda Function in Scheme

```
( map (lambda(m) ( * m m ) ) '( 1 2 3 4 ) )



(define sqr  ( lambda(m) (* m m )) )


( map sqr '(4 5))
```

Map/Reduce in scheme

```
( define  ( custom_map f x )
     ( cond (( null? x ) '() )
           (else ( cons (f (car x ))
                        (custom_map f ( cdr x ))))))

( define ( reduce f x v ) ( cond ((null? x ) v)
                          (else (f (car x ) (reduce f
                                (cdr x ) v )))))

(reduce ( lambda(a b ) ( + a b ) )
(custom_map (lambda(a) ( * a a ) ) '(1 2 3)) 0 )

(reduce ( lambda(a b ) ( * a b ) )
(custom_map (lambda(a) ( * a a ) ) '(1 2 3)) 2 )
```

Map in C#

```
 public static IEnumerable<T>
          Maps<T>(this IEnumerable<T> x, Func<T, T> f)
{
          List<T> n = new List<T>();

          foreach (T t in x)
          {
              n.Add(f(t));

          }

          return n;
}
```

Reduce in C#

```
public static
  T Reduce<T>(this IEnumerable<T> x,
           Func<T, T, T> f, T init)
      {
           T s = init;
           foreach (T t in x)
           {
               s = f(s, t);


           }
           return s;
  }
```

LINQ as ORM

- **The Problem of persistence**
- **The World is sticking with Relational Paradigm**
- **Applications are designed in an Object Oriented manner**
- **Top Down Design vs Bottum Up design**
- **How to brige the Gap ?**
- **ORM – if it is the solution , how do i go about it ?**
- **ORM – what the heck is it ?**
- **Hibernate and Nhibernate**
- **LINQ can update data as well**

LINQ as XML Processor

- The “great” parsing game
- DOM VS SAX
- XmlDocument solution from MS
- Stringification is the strategy
- A good Alternative is LINQ to XML
- One can use the same LINQ syntax to Query XML

MultiCore Programming

- **The underlying functional programming constructs of LINQ**
- **CLOSURE – a great invention**
- **Regular Expression and Relational database – The Closure connection**
- **Outer Variable Capture**
- **Potential for a good stateless programming model**
- **Parellel LinQ (PLINQ)**
- **Task Parellel Library (TPL)**
- **The MultiCore computing**

Q&A