LINQ – Language Integrated Query

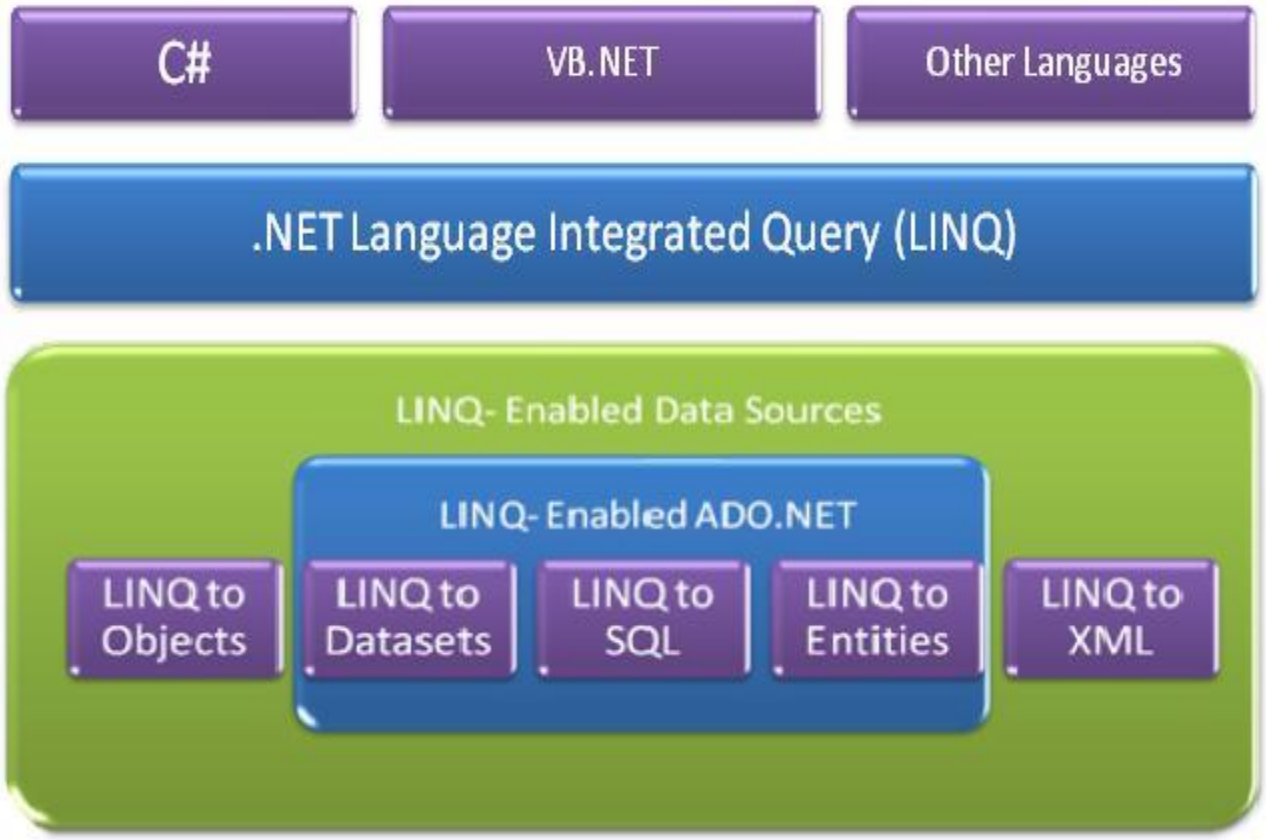
LINQ Basics – What, Where

- LINQ is a data source agnostic way of querying
- Integrated into the programming language
- Provides compile time syntax checking
- It is also applicable to in memory data sources

Data Access before LINQ

The data source determines the tools and APIs that needs to be used

LINQ architecture



Data Source

- Data source must be an object that supports the IEnumerable<T> interface
- Or a derived interface such as IQueryable<T>
- If data source is not a queryable type the LINQ provider must represent it as such.

  Example – XML

  LINQ to XML loads the XML into a queryable XElement type

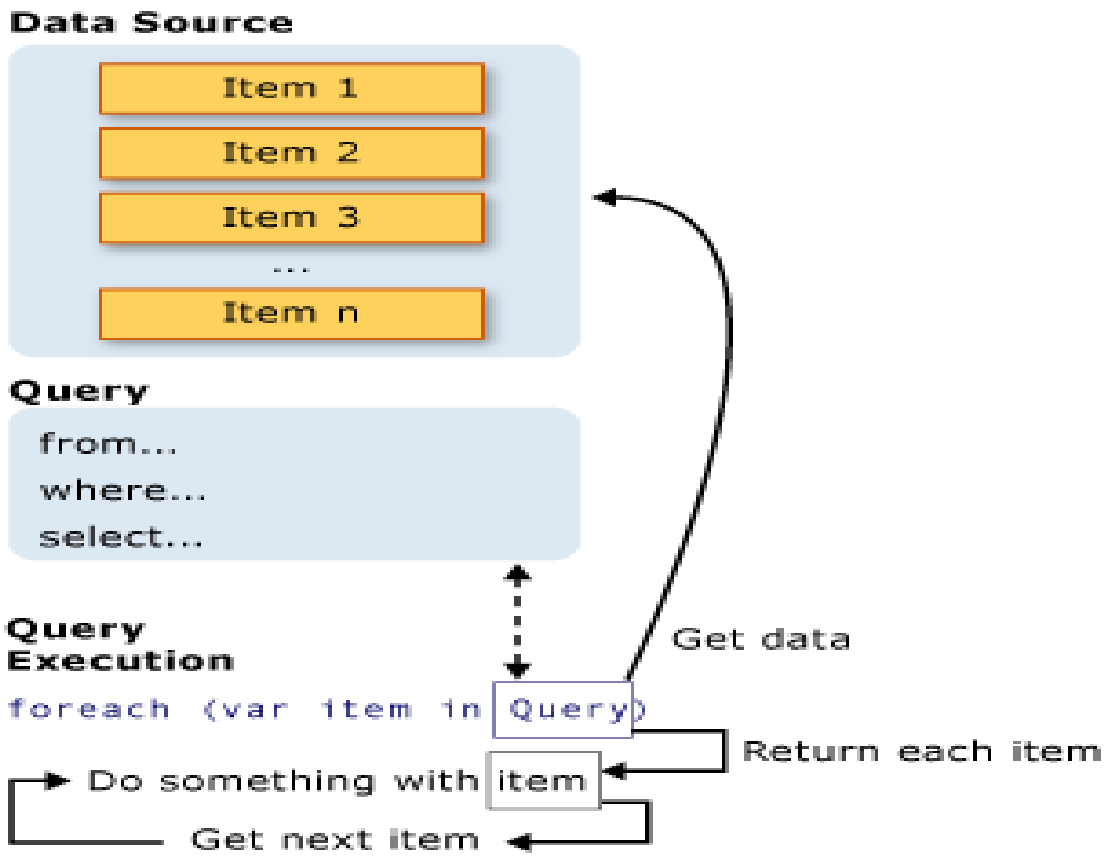
LINQ Providers

- LINQ is extensible
- Additions can be made to the query operators
- LINQ Provider – software that implements the IQueryProvider and IQueryable interfaces for a particular datastore
- LINQ Providers allow LINQ queries against any data store.

LINQ Queries

- Obtain a Data Source
- Create the query
- Execute the query

LINQ Query Execution



LINQ Standard Query Operators

- Standard Query operators are defined as extension methods on the static Enumerable and Queryable classes of System.Linq namespace
- Arrays, Lists and Collection Objects implement IEnumerable<T> interface
- Objects implementing IQueryable<T> are backed by LINQ providers

Standard Query Operators Classification

- Filtering
- Projecting
- Joining
- Ordering
- Grouping
- Conversions
- Sets
- Aggregation
- Quantifiers
- Generation
- Elements

Query Keywords

- from clause – datasource and range variable
- where clause - filter
- select clause – type of values that will be produced when query is executed. Query must end with select or group clause

Query Keywords

- group clause - The group clause returns a sequence of IGrouping<TKey, TElement> objects that contain zero or more items that match the key value for the group.
- into - The into contextual keyword can be used to create a temporary identifier to store the results of a group, join or select clause into a new identifier

Query Keywords

- orderby clause
- join clause
  - Inner join
  - Group join
  - Left o

Query Keywords

- let clause – stores results of subqueries
- ascending, descending
- on
- equals
- by
- in

Deferred Vs Immediate Execution

- Deferred or lazy operators offer a deferred execution of the query
- Immediate or greedy operators execute the query immediately

Deferred Vs Immediate Execution - Thumb rule

- If a query returns a single result then it will be immediately executed
- If a query returns IEnumerable<T> then most of the time it is deferred execution
- Hover on the var variable in Visual Studio to determine the return type

LINQ and Generic Types

- LINQ Queries are based on Generic Types
- IEnumerable<T> enables the Generic classes to be enumerated
- LINQ query variables are typed as IEnumerable<T> or IQueryable<T>

LINQ And Non Generic Types

- Range variable must be explicitly specified when querying non generic IEnumerable collections like ArrayList

Lambda Expressions

- It is an anonymous type used to create delegates or expression tree types
- Useful in writing LINQ queries
- => is called Lambda operator
- x => x * x specifies a parameter that is named x and returns the value of x squared

LINQ And Strings

- LINQ can be used to query and transform strings and collection of strings
- It can be used along with the string functions

LINQ to XML

- LINQ to XML provides an in memory XML programming interface
- It can be compared to an updated, redesigned Document Object Model
- Ability to use query results as parameters to XElement and XAttribute object constructors

LINQ to XML Uses

- Load XML from files or streams.
- Serialize XML to files or streams.
- Create XML from scratch by using functional construction.
- Query XML using XPath-like axes.

LINQ to XML Uses

- Manipulate the in-memory XML tree by using methods such as Add, Remove, ReplaceWith, and SetValue.
- Validate XML trees using XSD (XML Schema Definition).
- Use a combination of these features to transform XML trees from one shape into another

LINQ to XML Vs DOM

- Work directly with XML elements instead of loading the Document Object Model that is memory intensive
- LINQ to XML simplifies handling of XML namespaces
- Static Method for loading of XML
- Supports XPathNavigator
- Supports XSD