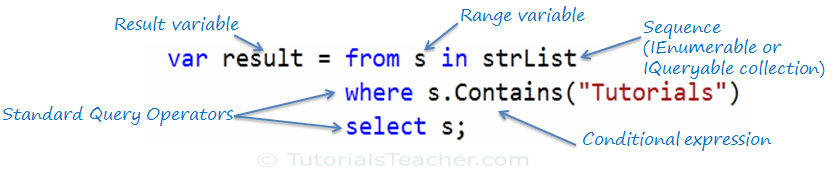
var booktitles = from b in books

select b.Title// FROM clause must appear before SELECT in LINQ.

Is equivalent to the following SQL query:

SELECT Title from Books



var orderByResult = from s in studentList

orderby s.StudentName

select s;

var orderByDescendingResult = from s in studentList

orderby s.StudentName descending

select s;

**What are the types of LINQ?**

* LINQ to Objects
* LINQ to XML
* LINQ to Dataset
* LINQ to SQL
* LINQ to Entities

**Which *interface* implements the standard query operators in LINQ?**

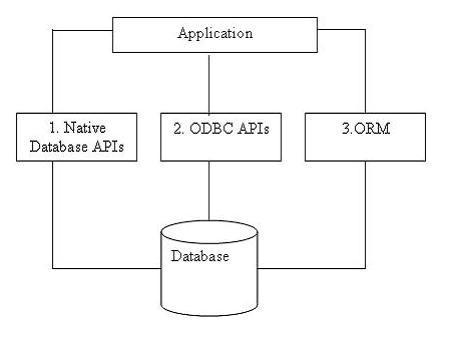
The standard query operators implement the **IEnumerable<T>** or the **IQueryable<T>** interface in C# and the IEnumerable(Of T) or the IQueryable(Of T) interface in Visual Basic.

it implements the IEnumerable(T) interface in Visual Basic and the IEnumerable<T> interface is implemented in C#.

**What are lambda expressions in LINQ?**

A lambda expression is a function without a name that calculates and returns a single value. All lambda expressions use the lambda operator **=>,** which read as goes to. The left side of the lambda operator specifies the input parameters and the right side holds the expression or statement block.

Input => expression or statement block.



LINQ to SQL, ADO.NET **SqlClient** adapters are used to communicate with real SQL Server databases.

**LINQ to SQL:**

When the application runs, LINQ to SQL translates the language-integrated queries in the object model into SQL and sends them to the database for execution. When the database returns the results, LINQ to SQL translates them back to objects that you can work with in your own programming language.

**DataContext** class acts as a bridge between SQL Server database and the LINQ to SQL.

**1.Comparing LINQ to SQL with LINQ to Objects:**

Followings are some key differences between **LINQ to SQL** and **LINQ to Objects**:

* LINQ to SQL needs a **Data Context** object. The Data Context object is the bridge between LINQ and the database. LINQ to Objects **doesn’t** need any intermediate LINQ provider or API.
* LINQ to SQL returns data of type IQueryable<T> while LINQ to Objects returns data of type IEnumerable<T>.
* LINQ to SQL is translated to SQL by way of Expression Trees, which allow them to be evaluated as a single unit and translated to the appropriate and optimal SQL statements. LINQ to Objects does not need to be translated.
* LINQ to SQL is translated to SQL calls and executed on the specified database while LINQ to Objects is executed in the local machine memory.

DataContext object is constructed in LINQ to SQL using :-

NorthwindDataContext db = new NorthwindDataContext();

## 2.LINQ to Entities:

## LINQ to Entities applications requires EntityClient mapping provider.

LINQ to Entities applications work against a **conceptual data model (EDM**).

If you want to work against a conceptual data model, use LINQ to Entities

 It is the same as what LINQ to SQL can do, though LINQ to Entities supports more features than LINQ to SQL, like multiple-table inheritance, and it supports many other mainstream RDBMS databases besides Microsoft SQL Server, like Oracle, DB2, and MySQL.

LINQ to SQL allows developers to generate .NET classes that represent data. Rather than mapping to a conceptual data model, these generated classes map directly to database tables, views, Stored Procedures, and user defined functions.

| **Features** | **LINQ to SQL** | **LINQ to Entities** |
| --- | --- | --- |
| Conceptual Data Model | No | Yes |
| Storage Schema | No | Yes |
| Mapping Schema | No | Yes |
| New Data Access Provider | No | Yes |
| Non-SQL Server Database Support | No | Yes |
| Direct Database Connection | Yes | No |
| Language Extensions Support | Yes | Yes |
| Stored Procedures | Yes | Yes |
| Single-table Inheritance | Yes | Yes |
| Multiple-table Inheritance | No | Yes |
| Single Entity from Multiple Tables | No | Yes |
| Lazy Loading Support | Yes | Yes |

**3.LINQ to DataSet:**

DataSet object by using LINQ to DataSet, you must first populate the dataset How can you do this?

You can load the data into the dataset by using different methods, such as:

* Using the DataAdapter class
* Using LINQ to SQL

**What are the different implementations of LINQ?**

The different implementations of LINQ are:

* **LINQ to SQL** - Refers to a component of.NET Framework version 3.5 that provides a run-time infrastructure to manage relational data as objects.
* **LINQ to DataSet** - Refers to a component that makes it easier and faster to query over data cached in a DataSet object.
* **LINQ to XML** - Provides an in-memory XML programming interface.
* **LINQ to Objects** - Refers to the use of LINQ queries with any IEnumerable or IEnumerable(T) collection directly, without the use of an intermediate LINQ provider or API, such as LINQ to SQL or LINQ to XML.

The **AsParallel** extension method is required to run a parallel query in PLINQ.