Practical-16

AIM: - Explore functionalities of LINQ.

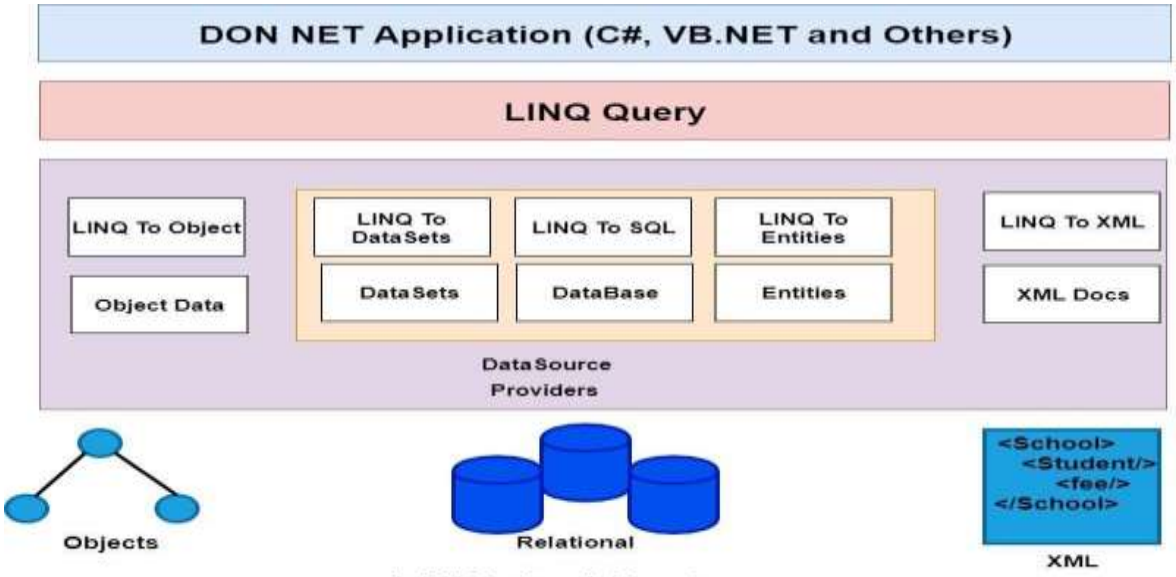
What is LINQ

The full form of LINQ is 'Language Integrated Query,' and introduced in .NET Framework 3.5 to query the data from different sources of data such as collections, generics, XML documents, ADO.NET Datasets, SQL, Web Services, etc. in C# and VB.NET.

LINQ provides the rich, standardized query syntax in a .NET programming language such as C# and VB.NET, which allows the developers to interact with any data sources.

In C# or VB.NET, LINQ functionality can be achieved by importing the System. LINQ namespace in our application. Generally, the LINQ contains a set of extension methods which allows us to query the source of data object directly in our code based on the requirement.

LINQ Architecture



LINQ Architecture

This is the architecture of the LINQ, as we know that we can develop the application in .NET. Similarly, the LINQ programming can return the any above mentioned .NET programming language. Between the actual LINQ query and the underlying data source, there is another component also present, known as LINQ Provider.

Provider

The responsibility of the LINQ provider is to convert the LINQ Query into a format so that the data source can understand it.

Need of LINQ

LINQ is simpler, ordered, and higher-level than SQL. When we want to use Querying Database, in most cases, LINQ is a more productive query language than SQL.

Also, we have the benefits of IntelliSense as the LINQ query is written in behind the code. LINQ has full type checking at compile time so that we can catch any error in compile time itself. In C# or VB.Net to write the query in LINQ is more fun.

Advantages of LINQ

In our applications, the benefits of LINQ are:

1. We do not need to learn new query language syntaxes for different sources of data because it provides the standard query syntax for the various data sources.
2. In LINQ, we have to write the Less code in comparison to the traditional approach. With the use of LINQ, we can minimize the code.
3. LINQ provides the compile-time error checking as well as intelligence support in Visual Studio. This powerful feature helps us to avoid run-time errors.
4. LINQ provides a lot of built-in methods that we can be used to perform the different operations such as filtering, ordering, grouping, etc. which makes our work easy.
5. The query of LINQ can be reused.

Disadvantages of LINQ

Disadvantages of LINQ are:

1. With the use of LINQ, it's very difficult to write a complex query like SQL.
2. It was written in the code, and we cannot make use of the Cache Execution plan, which is the SQL feature as we do in the stored procedure.
3. If the query is not written correctly, then the performance will be degraded.
4. If we make some changes to our queries, then we need to recompile the application and need to redeploy the dll to the server.

LINQ to SQL CRUD operations:

The LINQ to SQL has the facility to maintain the changes whatever we do with the objects like adding, removing, or updating the items in the collection of the object till we submit the changes

by using the method SubmitChanges() . After submitting the changes, the LINQ to SQL will translate our actions to SQL and submit the changes to the database.

LINQ to SQL Insert Operations

Here is the syntax of using the LINQ to SQL Insert operations query in C# to insert the data in the database.

```
EmployeeDBDataContext db = new
EmployeeDBDataContext();   Employee Detail emp = new
EmployeeDetail();   emp.EmpName = txtname.Text;
emp.Location = txtlocation.Text;
db.EmployeeDetails.InsertOnSubmit(emp);
db.SubmitChanges();
```

LINQ to SQL Update Operations

Here is the syntax of using the LINQ to SQL update operation query to update the data in the database.

```
EmployeeDBDataContext db = new EmployeeDBDataContext();

EmployeeDetail emp = new EmployeeDetail();
 emp = db.EmployeeDetails.Single(x => x.EmpId ==
 empid);   emp.EmpName = txtname.Text;   emp.Location =
 txtlocation.Text;   emp.Gender = txtgender.Text;
 db.SubmitChanges();
```

LINQ to SQL Delete Operations

Here is the syntax of using the LINQ to SQL delete operations in C# to delete the data in the database.

```
EmployeeDBDataContext db = new EmployeeDBDataContext();

EmployeeDetail emp = new EmployeeDetail();
 emp = db.EmployeeDetails.Single(x => x.EmpId == empid);
 db.EmployeeDetails.DeleteOnSubmit(emp);
 db.SubmitChanges();
```