

1. Explain what is LINQ? Why is it required?

Language Integrated Query or LINQ is the collection of standard query operators which provides query facilities into .NET framework language like C#, VB.NET.

LINQ is required as it bridges the gap between the world of data and world of objects.

2. What are the types of LINQ?

LINQ to Objects

LINQ to XML

LINQ to Dataset

LINQ to SQL

LINQ to Entities

3. Explain how LINQ is useful than Stored Procedures?

Debugging: It is difficult to debug a stored procedure but as LINQ is part of .NET, visual studios debugger can be used to debug the queries

Deployment: For stored procedure, additional script should be provided but with LINQ everything gets compiled into single DLL hence deployment becomes easy

Type Safety: LINQ is type safe, so queries errors are type checked at compile time

4. List out the three main components of LINQ? Explain what is the extension of the file, when LINQ to SQL is used?

Three main components of LINQ are

Standard Query Operators

Language Extensions

LINQ Providers

The extension of the file used is .dbml

5. Define what is Where clause and Let clause?

Where clause: It allows adding some conditional filters to the query.

Let clause: It allows defining a variable and assigning it a value calculated from the data values.

6. Explain why SELECT clause comes after FROM clause in LINQ?

With other programming language and C#, LINQ is used, it requires all the variables to be declared first. "FROM" clause of LINQ query defines the range or conditions to select records. So, FROM clause must appear before SELECT in LINQ.

7. Explain what is the use of System.XML.Xlinq.dll?

System.Data.Linq.dll provides the functionality to work with LINQ to SQL

8. Explain what is lambda expressions in LINQ?

Lambda expression is referred as a unique function use to form delegates or expression tree types, where right side is the output and left side is the input to the method. For writing LINQ queries particularly, Lambda expression is used.

9. Explain how LINQ with databases can be used?

LINQ supports XML, SQL, Dataset and Objects. Through LINQ to objects or LINQ to Datasets one can use LINQ with other databases. The objects and datasets take care of database particular operations, and LINQ only needs to deal with those objects and not the database operations directly.

10. Explain what is the difference between Skip() and SkipWhile() extension method?

Skip() : It will take an integer argument and from the given IEnumerable it skips the top n numbers

SkipWhile() : It will continue to skip the elements as far as the input condition is true. It will return all remaining elements if the condition is false

11. In LINQ how will you find the index of the element using where () with Lambda Expressions?

In order to find the index of the element using where () with the lambda expression

Where (i, ix) => i == ix);

12. Explain what is the difference between Statement Lambda and Expression Lambda?

Expression Lambdas are extensively used in the construction of Expression Trees

To create expression trees statement lambdas cannot be used

13. Mention what is the role of DataContext classes in LINQ?

DataContext class acts as a bridge between SQL Server database and the LINQ to SQL. For accessing the database and also for changing the data in the database, it contains connections string and the functions.

14. Explain what are LINQ query expressions?

Query expression is nothing but an LINQ query. It is a combination of query clauses that identifies the data sources for a query. It contains information for sorting, filtering, grouping or joining to apply to the source data. It determines what information should be retrieved from the data source.CV.

15. Explain what are compiled queries?

In compiled LINQ queries, the plan is cached in a static class and static class is a global cache.

Rather than preparing the query plan from scratch, LINQ prepares plan using static class object.

16. Explain how standard query operators useful in LINQ?

Standard Query Operators useful in LINQ are

Get a total count of elements in the collection

Order the results of a collection

Grouping

Computing average

Joining two collections based on matching keys

Filter the results

17. Explain what is the purpose of LINQ providers in LINQ?

LINQ providers are set of classes that take an LINQ query which generates method that executes an equivalent query against a particular data source.

18. Explain what is “LINQ to Objects”?

When LINQ queries any IEnumerable(Of T) collection or IEnumerable directly without the use of an intermediate LINQ provider or API such as LINQ to SQL or LINQ to XML is referred as “LINQ to Objects.”

19. Explain how you can differentiate between Conversion Operator “ToDictionary” and “IEnumerable” of LINQ?

To solve the conversion type problems “IEnumerable” and “ToDictionary” conversion operator are used.

“ToDictionary” conversion operator is the instance of Dictionary (k, T). The “keySelector” predicate recognizes the key of each item, while “elementSelector”, is used to extract each single item, if it is given.

Extension method on “IEnumerable” is.AsEnumerable. AsEnumerable simply returns the source sequence as an object of type IEnumerable <T>.

20. What are the advantages of using LINQ in Dataset?

The advantages of LINQ are as follows -

The primary purpose of using LINQ is to retrieve complex query in a dataset.

LINQ is used to combine the values from two different data set.

It is also used to fetch a unique value from the data set.

LINQ gives a more precise way than SQL query of querying the dataset.

LINQ also provides more functionality as compared with ADO.NET.

21. What are the different ways to write LINQ query syntax?

There are two common ways to write a LINQ query to data sources.

Query syntax or Expression syntax

Method syntax or method Extension Syntax

22. What is the Query syntax?

Query syntax is similar to Structured Query Language for the database. It is described within the C# or VB code.

Syntax

from <range variable> in <IEnumerable<T> or IQueryable<T> Collection>

<specific Query Operators> <lambda expression>

<select or groupBy operator> <result formation>

Few important points about Query syntax are as follows.

It starts with FROM keyword and ends with SELECT or GROUP BY keyword.

It is same as SQL (Structured Query Language) syntax.

It uses some different operators like joining, grouping, sorting, filtering operators to construct the desired result.

It uses implicitly typed variable to hold the result of the LINQ query.

23. Explain LINQ method syntax and give an example?

The LINQ method syntax or fluent syntax uses some additional methods included in the Enumerable or Queryable static class.

Method Syntax is the same as calling extension method.

LINQ Method Syntax is also known as fluent syntax because it facilitates a series of extension methods call.

Implicitly typed variable can be used to hold the result of the LINQ query

Syntax

```
// string collection
```

```
ICollection<string> stringList = new List<string>() {
    "Java Developer",
    ".Net Developer",
    "Graphics Designer",
    "Mean Stack Developer" ,
    "Java"};
```

```
// LINQ Query Syntax
```

```
var result = stringList.Where(s => s.Contains("Tutorials"));
```

Example: C# Program for Method Syntax

```
using System;
```

```

using System.Linq;
using System.Collections.Generic;
public class Program
{
    public static void Main()
    {
        // string collection
        IList<string> stringList = new List<string>() {
            "Java Developer",
            ".Net Developer",
            "Graphics Designer",
            "Mean Stack Developer" ,
            "Java"
        };

        // LINQ Method Syntax
        var result = stringList.Where(s => s.Contains("Developer"));
        foreach (var str in result)
        {
            Console.WriteLine(str);
        }
    }
}

```

Output:

```

Java Developer
.Net Developer
Mean Stack Developer

```

24. Explain LINQ to SQL?

LINQ to SQL is a part of ADO.NET technologies. It manages the relational data as an object. LINQ to SQL converts the language integrated query in the object to SQL and forward them to the database for execution. When the database responses, the result LINQ to SQL translate them back to objects.

LINQ to SQL supports user-defined functions and stored procedure in the database.

25. Explain LINQ to XML?

LINQ to XML provides the in-built document reform capabilities of the DOM (Document Object Model) and supports LINQ Queries. Using it, we can modify query, navigate, and save the changes of an XML document. It allows us to write queries to retrieve and navigate a collection of elements and attributes. It is quite similar to XPath and XQuery.

26. What is Expression Tree in LINQ?

Lambda expressions are extensively used in Expression Tree construction. An Expression Tree represents code in a tree-like format, where each node acts as an impression. Expression trees can be converted into compiled code and run it.

In .NET framework, Expression class is used to create expression tree through the API. Expression Trees API also support assignment and some control flow expression such as conditional blocks, loops, and try-catch blocks. By using the API, we can generate expression trees that are more complex than those that can be created from lambda expressions.

27. Elaborate the PLINQ?

The term PLINQ stands for parallel LINQ. It's a parallel implementation of LINQ to objects. It supports parallel programming, and it is so closely related to the task parallel library. It facilitates

with some queries to automatically take advantage of multiple processors. PLINQ is capable of increasing the speed of LINQ to Objects queries by using all available cores on the host computer more efficiently.

Syntax of PLINQ

Following is the syntax of using PLINQ to increase the performance of LINQ queries in c#, vb.net.

C# Code

```
IEnumerable<int> rvals = Enumerable.Range(1, 100000000);
var output1 = rvals.AsParallel().Where(x => x % 12345678 == 0).Select(x => x);
```

VB.NET Code

```
Dim rvals As IEnumerable(Of Integer) = Enumerable.Range(1, 100000000)
Dim output1 = rvals.AsParallel().Where(Function(x) x Mod 12345678 = 0).[Select](Function(x) x)
```

If you notice the given syntax, I have used Parallel method with LINQ queries to increase performance of LINQ queries.

28. Explain why SELECT clause comes after FROM clause in LINQ?

LINQ requires all the variables to be declared first. The "FROM" clause of LINQ query defines the condition or range to select records so FROM clause must act before SELECT in LINQ query.

29. What are Anonymous Types?

Anonymous types are run-time compiler generated by the compiler. There is no need to specify the name while creating the Anonymous compiler yet we can write properties names and their values. The compiler creates these properties and assigns values to them at runtime.

```
var v = new { PropertyFirst = "first value", PropertySecond = "second value" };
Console.WriteLine(k.PropertyFirst);
```

Anonymous class is helpful in LINQ queries. It saves the intermediate result while performing queries.

There are some restrictions on Anonymous types as well:

- Anonymous types cannot implement interfaces.
- Anonymous types can not specify any methods.
- We cannot define static members.
- All defined properties must be initialized.
- We can only define public fields.

30. What is the difference between First() and FirstOrDefault() selector methods in LINQ?

The First() method always expects at least one element in the result set. If there isn't any element in the result, First() returns an exception. While FirstOrDefault() is compatible with a result set having 0 elements, it does not throw any exception.

31. What is the difference between N-tier and N-layer architecture?

N-Tier and N-Layer are two different concepts. These terms are generally used during the design of the application architecture. N-Tier stands for the actual n system components of your application. On the other hand, N-Layers stands for the internal architecture of your component.

The main advantages of the layered architectural style are as follows.

Abstraction

Isolation
Manageability
Performance
Reusability
Testability

The main advantages of tier architecture styles are as follows.

Maintainability
Scalability
Flexibility
Availability