ADO.NET Connected Architecture

Q1. What are the objects used to execute SQL queries or Stored Procedures against the data Source?

Ans: Command Object and Connection Object

Q2. After Execution of a query where will the result stored?

Ans: DataReader Object

Q3. What is the method that we use to retrieve the rows from the table?

Ans: DataReader.read()

Q4. What is the purpose of connection in connected architecture?

Ans: Connection class used to establish the connection between front end and back end.

Explanation:In Connection Oriented Data Access Architecture the application makes a connection to the Data Source and then interact with it through SQL requests using the same connection. In these cases the application stays connected to the database system even when it is not using any Database Operations.

Q5. What is the flow of connection in connected architecture?

Ans: Open Connection

Execute Command

Process Data in Reader

Close Reader

Close Connection

Explanation:In connection-oriented architecture the Database gets connected to the front end then the command passes the query to the server from back end and on the server the result which has been generated. The result which has been generated will be read by DataReader.

Q6. What are the classes that contains ADO.NET?

A)

Connection

Command

DataReader

DataAdapter

DataSet

Q7. What is DataSet Object?

A) It is set to be collection of data with a tabular column representation.

Q8. What is Connection Oriented Data Access Architecture?

EXPLANATION:

In Connection Oriented Data Access Architecture the application makes a connection to the Data Source and then interact with it through SQL requests using the same connection. In these cases the application stays connected to the database system even when it is not using any Database Operations.

ADO.Net solves this problem by introduces a new component called Dataset. The DataSet is the central component in the ADO.NET Disconnected Data Access Architecture. A DataSet is an in-memory data store that can hold multiple tables at the same time. DataSets only hold data and do not interact with a Data Source. One of the key characteristics of the DataSet is that it has no knowledge of the underlying Data Source that might have been used to populate it.

DataSet ds = new DataSet();

In Connection Oriented Data Access, when you read data from a database by using a DataReader object, an open connection must be maintained between your application and the Data Source. Unlike the DataReader, the DataSet is not connected directly to a Data Source through a Connection object when you populate it. It is the DataAdapter that manages connections between Data Source and Dataset by fill the data from Data Source to the Dataset and giving a disconnected behavior to the Dataset. The DataAdapter acts as a bridge between the Connected and Disconnected Objects.

SqlDataAdapter adapter = new SqlDataAdapter("sql", "connection");

DataSet ds = new DataSet();

adapter.Fill(ds, "Src Table");

By keeping connections open for only a minimum period of time, ADO .NET conserves system resources and provides maximum security for databases and also has less impact on system performance.

Q 9. DATAREADER IN CONNECTED ARCHITECTURE

EXPLANATION: You can use the ADO.NET DataReader to retrieve a read-only, forward-only stream of data from a database. Results are returned as the query executes, and are stored in the network buffer on the client until you request them using the Read method of the DataReader. Using the DataReader can increase application performance both by retrieving

data as soon as it is available, and (by default) storing only one row at a time in memory, reducing system overhead.

Q10. What are Stored Procedures?

Ans. A stored procedure is a reusable sub routine stored in a database. SQL Server compiles stored procedures which makes it more efficient to use. Therefore , rather than dynamically building queries in the code, we can always take advantage of the reusability and performance benefits of stored procedures. A stored procedure can be called by simply passing the stored procedure name followed by parameter arguments as an SQL statement. But you can use the Parameters collection of the ADO.NET Command object which enables you to more explicitly define stored procedure parameters as well as to access output parameters and return values. To call a stored procedure, set the CommandType of the Command object to StoredProcedure.

Q11. How to execute Stored Procedures?

Ans. A Stored Procedure written in SQL server can be called and executed through ADO.NET

The SQL command object is used to execute Stored Procedurs

Example: sql Command cmd = new sqlCommand("Query_Stud",con);

cmd.CommandType = CommandType.Stored Procedure;

Q12. What are the various SQL Command constructors used in a connected Architecture?

Ans.- There are 4 types of SQL Command Constructors used .They are as follows :

a. SQLCommand()

b. SQLCommand(String CommandText)

c. SQLCommand(String CommandText,SqlConnection con)

d. SQLCommand(String CommandText, SqlConnection con,SqlTransaction trans)

SQLCommand() creates a new SqlCommand with no properties set and no associated connection.

SQLCommand(String CommandText) creates a new SqlCommand with the CommandText property set but no associated connection.

SQLCommand(String CommandText,SqlConnection con) creates a new SqlCommand with the CommandText property set and the specified SqlConnection as the Connection property.

SQLCommand(String CommandText, SqlConnection con,SqlTransaction trans) creates a new SqlCommand with the CommandText property set, the specified SqlConnection as the Connection property and SQLTransaction to for the command to execute in.

Q13. What are the functions of the command classes OracleCommand and SqlCommand?

Ans. The command classes (OracleCommand and SqlCommand) are used to execute commands on a data source across a connection. Once a connection is established, a command can be used to execute a SQL statement or stored procedure. The SQLCommand represents a TransactSQL statement or stored procedure to execute against a SQL Server database. The SQLCommand class is part of the System.Data.SQLClient namespace. When an instance of SqlCommand is created, the read/write properties are set to their initial values.

Q14. What is a DataReader object?

The DataReader object helps in retrieving the data from a database in a forward-only, read-only mode. The base class for all the DataReader objects is the DbDataReader class.The DataReader object is returned as a result of calling the ExecuteReader() method of the Command object. The DataReader object enables faster retrieval of data from databases and enhances the performance of .NET applications by providing rapid data access speed. However, it is less preferred as compared to the DataAdapter object because the DataReader object needs an Open connection till it completes reading all the rows of the specified table.

Q15. What is the use of the Connection object?

The Connection object is used to connect your application to a specific data source by providing the required authentication information in connection string. The connection object is used according to the type of the data source. For example, the OleDbConnection object is used with an OLE-DB provider and the SqlConnection object is used with an MS SQL Server.

CONNECTED ARCHITECTURE SUMMARY

Connected architecture is a method of connecting the application with a database, in our case, with the MS SQL Server database. It requires a network connection to function properly. This is made possible using providers, one of which is SqlClient, which we use as a namespace in the start of our code. We also use a namespace called System.Data.Configuration, as this helps in configuration of the SQL Commands to be used.

We use an SQL Connection class, using which we can establish connection with the database. After this is done, any commands necessary may be executed. The connection class helps in establishing connection with the front and back end. It is used for connecting the database with the ADO .net application currently in use. The application then connects to the data source, and

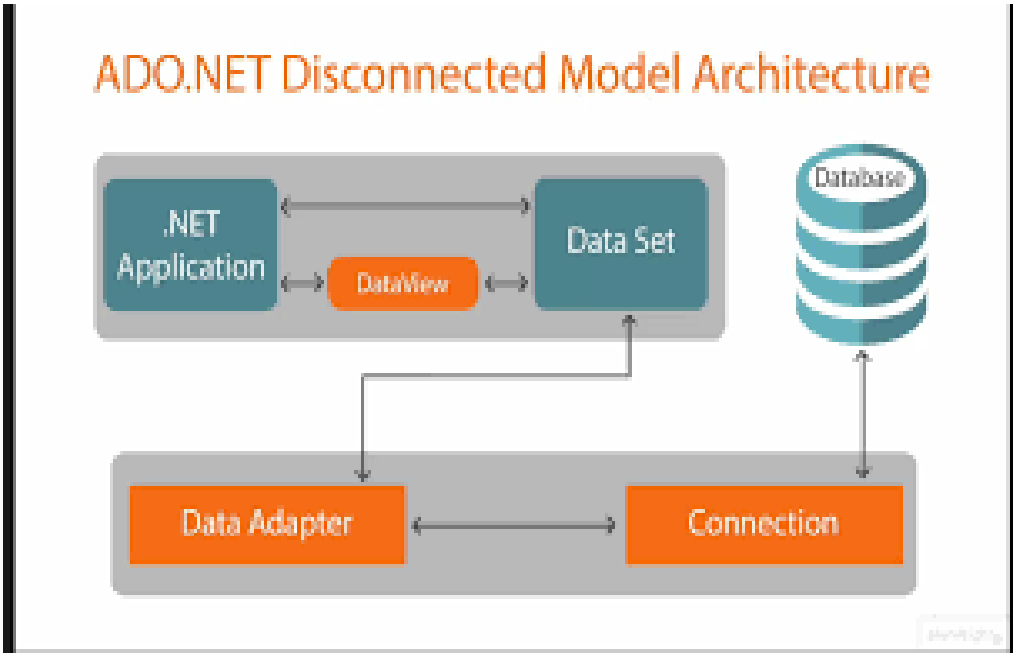
stays connected even when the application is not being used. The Connection is when constantly make trips to the database for any CRUD (Create, Read, Update and Delete) operation you wish to do. This creates more traffic to the database but is normally much faster as you should be doing smaller transactions. There are several elements in the connected architecture model. They are- Connection, Command, DataSet, DataAdapter and DataReader. Each has a separate function. As mentioned earlier, Connection helps in connecting the front and back end. It ensures that the changes are made in both places in a timely fashion. The SQLCommand helps in querying the type of interaction we want to perform with a database. For example, we can do select, insert, modify, and delete commands on rows of data in a database table. It has several types of its own, from upto creating basic SQL commands to adding associated connections. Commands may be creating using one of the command constructors for the .NET Framework data provider you are working with. Constructors can take optional arguments. Another element used is the DataSet object. The DataSet contains the copy of the data we requested through the SQL statement. The DataSet is a memory-resident representation of data that provides a consistent relational programming model regardless of the data source. It contains the data table collections and the data reader collections. The tables in a DataSet may be filled or populated using a SQL DataAdapter class, which is another element of the Connected Architecture model. Using the SQLAdpater, it is possible to build and fill each DataTable in a DataSet with data from a data source. The purpose of the DataAdapter is embedded in its name: It performs the activities necessary to get the data from the data source on the server into the database that's held in the DataSet. To do that, the DataAdapter lets us specify the commands that should be carried out to retrieve and update data. The object provides four properties that allow us to control how updates are made to the server: SelectCommand, UpdateCommand, InsertCommand, and DeleteCommand. The four properties are set to Command objects that are used when data is manipulated.For instance, when we call the DataAdapter's Fill method to retrieve data from a data source and pour it into a DataSet, the Command object in the SelectCommand property is used. DataReader is the part of the connected architecture that is used for accessing data from the data store and is one of the two mechanisms that ADO.NET provides. This object provides a read only, forward only, high performance mechanism to retrieve data from a data store as a data stream, while staying connected with the data source. The DataReader provides an unbuffered stream of data that allows procedural logic to efficiently process results from a data source sequentially. The Read method of the DataReader object may be used to obtain a row from the results of the query. Access to each column of the returned row may be done by passing the name or ordinal reference of the column to the DataReader. The DataReader is a good choice when retrieving large amounts of data because the data is not cached in memory. We should always call the Close method when we have finished using the DataReader object. If our Command contains output parameters or return values, they will not be available until the DataReader is closed.

Finally, when it comes to the process flow of the Connected Architecture, the flow is as follows :-

- Open Connection
- Execute Command
- Process Data in Reader
- Close Reader
- Close Connection

Disconnected Architecture in ADO.NET

The architecture of ADO.net in which data retrieved from database can be accessed even when connection to database was closed is called as disconnected architecture. Disconnected architecture of ADO.net was built on classes connection, dataadapter, commandbuilder and dataset and dataview. The data is modified independently & changes are merged



Connection: Connection object is used to establish a connection to database and connectionit self will not transfer any data.

DataAdapter: DataAdapter is used to transfer the data between database and dataset. It has commands like select, insert, update and delete. Select command is used to retrieve data from database and insert, update and delete commands are used to send changes to the data in dataset to database. It needs a connection to transfer the data.

CommandBuilder: by default dataadapter contains only the select command and it doesn'tcontain insert, update and delete commands. To create insert, update and delete commands for the dataadapter, commandbuilder is used. It is used only to create these commands for the dataadapter and has no other purpose.

DataSet: Dataset is used to store the data retrieved from database by dataadapter and make it available for .net application.
To fill data in to dataset **fill()** method of dataadapter is used and has the following syntax.

da.Fill(ds,"tableName");

When fill method was called, dataadapter will open a connection to database, executes select command, stores the data retrieved by select command in to dataset and immediately closes the connection.

As connection to database was closed, any changes to the data in dataset will not be directly sent to the database and will be made only in the dataset. To send changes made to data in dataset to the database, **Update()** method of the dataadapter is used that has the following syntax.

da.Update(ds,"tablename");

When Update method was called, dataadapter will again open the connection to database, executes insert, update and delete commands to send changes in dataset to database and immediately closes the connection. As connection is opened only when it is required and will be automatically closed when it was not required, this architecture is called disconnected architecture.

A dataset can contain data in multiple tables.

DataView: DataView is a view of table available in DataSet. It is used to find a record, sort the records and filter the records. By using dataview, you can also perform insert, update and delete as in case of a DataSet.

1. What is SQL Data Adapter?

The SqlDataAdapter uses the Fill method , which changes the data in the DataSet to

match the data in the data source, and the Update method, which changes the data

in the data source to match the data in the DataSet, using the appropriate Transact-

SQL statements against the data source. SqlDataAdapter fills a DataSet , it creates the necessary tables and columns for the returned data if they do not already exist.

2. What does the Update and Fill method of Data Adapter do?

The DataAdapter's Fill and Update method automatically opens the connection,executes, and closes the connection

3. What is Data Set?

The DataSet is a memory-resident representation of data that provides consistent relational programming model regardless of the source of the data it contains. A DataSet represents a complete set of data including the tables that contain columns, rows and constrain the data, as well as the relationshipsbetween the tables. It is disconnected from the data source.

4. What are the various members exposed by Data Set?

DataSetName - Gets or sets the name of the current DataSet

EnforceConstraints – Gets or sets a value indicating whether constraint rules are followed when attempting any update operation

HasErrors – Indicates whether there are errors in any of the rows in any of the tables of this DataSet

Relations – Retrieves the collection of relations that link tables and allow navigation from parent tables to child tables

Tables - Retrieves the collection of tables contained in the DataSet.

AcceptChanges() – Commits all the changes made to this DataSet since it was loaded or the last time AcceptChanges was called

Clear()- clears the DataSet of any data by removing all rows in all tables.

GetChanges()- Retrieves a copy of the DataSet containing all changes made to it since it was last loaded, or since AcceptChanges was called.

HasChanges()- Retrieves a value indicating whether the DataSet has changes, including new, deleted, or modified rows

RejectChanges()- Rolls back all the changes made to the DataSet since it was created, or since the last time DataSet.AcceptChanges was called

Merge() – Merges this DataSet with a specified DataSet.

Reset()- Resets the DataSet to its original state. Subclasses should override Reset to restore a DataSet to its original state.

5. When do we use Data Set?

DataSet can be used in the application whenever discrete table of results is being navigated and also if the application manipulates data from different sources like a relational database or XML file. Additionally, caching the data using the DataSet improves performance for tasks such as Sorting, Filtering.

6. What is Typed DataSet?

It is a collection of classes that are inherit from DataSet, DataTable and DataRow Classes with additional properties, methods and events based on DataSet Schema. It allows developer wmn values, access parent and child records, and helps us find rows and handle null column values.

7. What is Untyped DataSet?

It doesn't have corresponding built-in-schema. It has tables, columns and so on which are exposed as collection.

DataSet's structure can be exported after creating tables and elements manually.

8. When do you use DataSet?

DataSet can be used in the application whenever discrete table of results is being navigated and also if the application manipulates data from different sources like a relational database or XML file.

Additionally, caching the data using the DataSet improves performance for tasks such as Sorting, Filtering.

To include type checking at design time use Typed DataSet. Apart from this in .Net environment statement completion is also supported when you work with Typed DataSet.

9. Method to refresh Data in DataSet?How?

DataAdapter.Fill method. This method matches new rows based on primary keys and applies corresponding changes from the server, if a Primary Key is defined for the DataTable.

If the Primary Key is not defined, then leads to duplicate rows.

10. Members of Data Table?

*ChildRelations

*Columns

*Constraints

*DataSet

*ParentRelations

*PrimaryKey

*Rows

*Tablename

*NewRow()

*Select().

11. Features of DataColumn?

*Type of Data to store

*Length of the column

*Whether the data can be modified

*Whether the data is unique

*Can accept null values

*Automated values

*Values calculated based on the expressions.

12. Purpose of the DataRow?

It maintains RowState property that is used by ADO.NET to track the changes that have been made to a DataRow and allows to identify the rows which are modified.

13. What is RowState?

It is a propety to track changes that have been made to DataRow while it is disconnected to be updated back to the data source. It has to versions :

a) Original version - contains values that are loaded into the row.

b) Current version - it contains values that are modified.

14. What is foreignKey constrain?

It enforces rules how update and deletes to related tables are propagated and they follow:

*Cascade - Deletes or updates related rows

*SetNull - Set values in related rows to DBNull

*SetDefault - Set values in related rows to the default value.

*None - Take no action on related rows.

15. What is the difference between data reader and data set?

DATA READER:

1. Used in a connected architecture

2. Provides better performance

3. DataReader object has read-only access

4. DataReader object supports a single table based on a single SQL query of one database.

5. A DataReader object is bound to a single control.

6. A DataReader object has faster access to data.

7. A DataReader object must be manually coded.

8. We can't create a relation in a data reader.

9. Whereas a DataReader doesn't support data reader communicates with the command object.

10. DataReader cannot modify data.

DATASET

1. Used in a disconnected architecture.

2. Provides lower performance.

3. A DataSet object has read/write access.

4. A DataSet object supports multiple tables from various databases.

5. A DataSet object is bound to multiple controls.

6. A DataSet object has slower access to data.

7. A DataSet object is supported by Visual Studio tools.

8. We can create relations in a dataset.

9. A Dataset supports integration with XML Dataset communicates with the Data Adapter only.

10. A DataSet can modify data.

16. Disadvantages of disconnected architecture?

a.Data is not always uptodate.

b.Change conflicts can occur and must be changed.

17. What is a command object?

The command object is one of the basic components of ADO .NET.

The Command Object uses the connection object to execute SQL queries.

The queries can be in the form of Inline text, Stored Procedures or direct Table access.

An important feature of Command object is that it can be used to execute queries and Stored Procedures with Parameters.

If a select query is issued, the result set it returns is usually stored in either a DataSet or a DataReader object.

Connection Read/Write The SqlConnection object that is used by the command object to execute SQL queries or Stored Procedure.

CommandTextRead/Write Represents the T-SQL Statement or the name of the Stored Procedure.

CommandType Read/Write This property indicates how the CommandText property should be interpreted. The possible values are:

1. Text (T-SQL Statement)

2. StoredProcedure (Stored Procedure Name)

3. TableDirect.

18. What is DataAdapter.

Answer: A DataAdapter bridges the gap between the disconnected DataTable objects and the physical data source.

The SqlDataAdapter is capable of executing a SELECT, DELETE and UPDATE statement on a data source as well as extracting

input from the result set into a DataTable object.

The SqlDataAdapter class provides a method called Fill() to copy the result set into the DataTable.

19. What are the properties of data adapter?

SelectCommand This command executed to fill in a Data Table with the result set.

InsertCommand Executed to insert a new row to the SQL database.

UpdateCommand	Executed to update an existing record on the SQL database.
DeleteCommand	Executed to delete an existing record on the SQL database.

20. Describe the Disconnected Architecture of Ado.net's Data Access Model?

ADO.NET maintains a disconnected database access model, which means, the application never remains connected constantly to the data source.

Any changes and operations done on the data are saved in a local copy (dataset) that acts as a data source.

Whenever, the connection to the server is re-established, these changes are sent back to the server,

in which these changes are saved in the actual database or data source.

LINQ & EF

1. What is LINQ , when it was introduced and what are the advantages of LINQ?

LINQ Stands for Language Integrated Query.

It was introduced from .NET framework 3.5 and continued as a part of .Net framework 4.0 and 4.5.

- Integrate Better with programming languages
- Improves productivity through intellisense in Visual studio
- Interact with data as objects.
- By creating a common querying language in LINQ, the developer is free from having to master independent data programmability technologies (for example: Xpath, Xquery, T/SQL). Instead, LINQ offers the developer a consistent way to query data.

++

2.Explain the layer of LINQ Architecture.

1. The top most layer consist of language extension which show all the languages which can use LINQ including C# , VB and other language which are the part of .Net Framework
2. The second layer consist of .NET LINQ
3. The third layer consist of LINQ Enabled Data source which specifies different data source on which LINQ can be used. LINQ can be use with Objects, SQL, Dataset and XML.

++

3. What are different flavors of LINQ?

LINQ to Objects

LINQ to ADO.NET

LINQ to XML

++

4.What is PLINQ?

Parallel LINQ (PLINQ) is a parallel implementation of the LINQ pattern. The primary difference is that PLINQ attempts to make full use of all the processors on the system. It does this by partitioning the data source into segments, and then executing the query on each segment on separate worker threads in parallel on multiple processors. In many cases, parallel execution means that the query runs significantly faster. Through parallel execution, PLINQ can achieve significant performance improvements over legacy code for certain kinds of queries, often just by adding the (AsParallel) query operation to the data source.

++

5.In which Namespaces API for different flavours of LINQ is defined?

Components	namespaces
LINQ to Objects	System.Linq namespaces inside System.Core.dll
LINQ to ADO.NET	System.Data.Linq namespace
LINQ to XML	System.XML.Linq namespace

++

6.What are the basic action required for LINQ query operations ?

All LINQ Query consists of three distinct actions

1.Obtain the data source

2.Create the query

3.Execute the query

++

7.write the syntax of a Basic LINQ Query.

from <range variable> in <collection>

<filter, joining, grouping, aggregate operators etc><Lambda expression>

<select or groupBy operator><formulate the result>

Every LINQ query begins with a from clause ends with a select query or group. In

between can be use filtering , joining , grouping etc on the data.

++

8.What are the basic query operator in LINQ?

Filtering Operators

Sorting Operators

projection Operators

Aggregation

Concatenation operator

Grouping Operators

Conversions

++

9. What does filtering operator do explain with example.

It filter a sequence based on a predicate function Suppose that you have to list the names and cities of customers from Italy. To filter a set of items, you can use the Where operator, which is also called a "restriction operator" because it restricts a set of items.

++

10.What are projection operator and what is difference between them?

Select Operator

• The Select operator performs a projection over a sequence.

• When the object returned by Select is enumerated, it enumerates the source sequence. Subsequently, it yields the results of evaluating the select or function for each element.

The first argument of the selector function represents the element to process.

The second argument, if present, represents the zero-based index of the element within the source sequence.

SelectMany Operator

The SelectMany operator performs a projection over a sequence. This operator is similar to Select because it allows us to define the elements that have to be picked up from a sequence. The difference is in the return type.

With the IEnumerable<S> type returned by the selector parameter of SelectMany, it is possible to concatenate many projection operations together. This can be done either on different sequences or starting from the result of a previous query.

++

11.What are sorting operators and describe them

OrderBy and OrderBy Descending

the OrderBy sort the data in ascending order and OrderByDescending operator inverts the ordering.

ThenBy and ThenByDescending

OrderBy allows us to specify only one ordering key. Hence we have to use either ThenBy or ThenByDescending to concatenate ordering-key values.

Reverse

This method simply returns a new sequence with elements in a reverse ordering of the source sequence.REVERSE simply reverses the items in the collection.

++

12.What does group by operator do?

the GroupBy operator groups elements of a sequence based on a given selector function.

++

13.What is concatenation operator do and write the syntax

This operator concatenates two sequences. The resulting IEnumerable<T> type is the concatenation of the first and second sequences specified as a parameter.

number and morenumbers are two different sequences

var query=numbers.Concat(morenumbers);

++

14.What are set opeators and describe them

Distinct:

• This operator is similar to the DISTINCT keyword used in SQL. It eliminates the duplicates from a sequence.

• When the code processes the query, it enumerates the element of the sequence, and stores into an IEnumerable<T> type. It is done for each element mthat has not been previously stored.

• The Distinct operator selects unique values from the sequence.

Except:

• This Operator Produces the set difference of two sequences by using the default equality comparer to compare values.

Intersect:

• This operator returns a sequence made by common elements of two different sequences.

• The Intersect Method is used to retrieve common elements in two sequences

Union:

• This operator returns a new sequence formed by uniting the two different sequences.

++

15.What are Types of Conversion operator and explain them.

• This operator produces a new IEnumerable<T> type composed of only the element of the specified type

OfType:

• The OfType Searches for the specified type T in the Sequence

ToArray:

• This operator returns an array composed of the elements of the source sequence.

SUMMARY

LINQ OPERATIONS

- LINQ stands for Language Integrated query which was introduced from .Net framework 3.5.
- It bridges the gap between the world of objects and the world of data.
- LINQ enables you to query the data from within the .Net programming language.

Benefits of using LINQ:

1. Working with the data in a consistent way regardless of the type of data
2. Interact with data as objects.
3. Integrate better with programming languages.
4. Improve productivity through intellisense in visual studio.

LINQ queries use set of query keywords built in to the .Net framework.Commonly used keywords are :

- from/in
- where
- orderby
- select
- groupby

All LINQ query operations require the following three distinct actions

1. Obtain the data source
2. Create the query
3. Execute the query

Every LINQ query begins with a from clause ends with a select query or group. In between can be use filtering , joining , grouping etc on the data.

Basic Query Operators

Filtering Operators

i) Where :- Filter value based on a predicate function

Projection Operators

i) Select :- The operator projects value on basis of a transform function

Sorting Operators

i) OrderBy :- The operator sort values in an ascending order

ii) OrderByDescending :- The operator sort values in a descending order

iii) ThenBy:- Executes a secondary sorting in an ascending order

iv) ThenByDescending :- Executes a secondary sorting in a descending order

v) Reverse :- Performs a reversal of the order of the elements in a collection

Aggregation

i) Average :- Average value of collection of value is calculated

ii) Count :- Counts the number of entries

iii) Max :- Find out the maximum value within a collection

iv) Min :- Find out the Minimum value within a collection

v) Sum :- Find out the sum of value within a collection