**Question 1: What is ADO.NET?**

**Answer 1:** ADO stands for Active Data Object and ADO.NET is a set of .NET libraries for ADO.

ADO.NET is a collection of managed libraries used by .NET applications for data source communication using a driver or provider:

* Enterprise applications handle a large amount of data. This data is primarily stored in relational databases, such as Oracle, SQL Server, and Access and so on. These databases use Structured Query Language (SQL) for retrieval of data.
* To access enterprise data from a .NET application, an interface was needed. This interface acts as a bridge between an RDBMS system and a .NET application. ADO.NET is such an interface that is created to connect .NET applications to RDBMS systems.
* In the .NET framework, Microsoft introduced a new version of Active X Data Objects (ADO) called ADO.NET. Any .NET application, either Windows based or web based, can interact with the database using a rich set of classes of the ADO.NET library. Data can be accessed from any database using connected or disconnected architecture.

ADO.NET provides mainly the following two types of architectures:

* Connected Architecture.
* Disconnected Architecture.

**ADO.NET Namespaces**

|  |  |
| --- | --- |
| Namespaces | Description |
| System.Data | Contains the definition for columns, relations, tables, database, rows, views and constraints. |
| System.Data.SqlClient | Contains the classes that are used to connect to a Microsoft SQL Server database such as SqlCommand, SqlConnection, SqlDataAdapter. |
| System.Data.Odbc | Contains classes required to connect to most ODBC drivers. These classes include OdbcCommand,OdbcConnection. |
| System.Data.OracleClient | Contains classes such as OracleConnection,OracleCommand required to connect to an Oracle database. |

See for more detail:

* [Database Programming With ADO.NET](http://www.c-sharpcorner.com/UploadFile/d0a1c8/database-programming-with-ado-net/)

**Q.1(b)Explain ADO.NET in brief.**

ADO.NET is a very important feature of .NET Framework, which is used to work with data that is stored in structured data sources, such as databases and XML files. The following are some of the important features of ADO.NET:

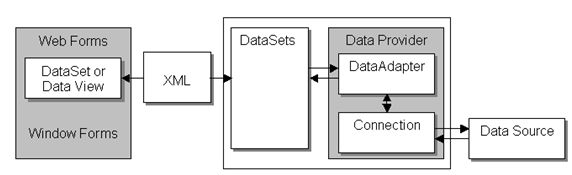
* Contains a number of classes that provide you with various methods and attributes to manage the communication between your application and data source.
* Enables you to access different data sources, such as Microsoft SQL Server, and XML, as per your requirements.
* Provides a rich set of features, such as connection and commands that can be used to develop robust and highly efficient data services in .NET applications.
* Provides various data providers that are specific to databases produced by various vendors. For example, ADO.NET has a separate provider to access data from Oracle databases; whereas, another provider is used to access data from SQL databases.

**Question 2: What are the ADO.NET components?**

**Answer 2:** ADO.NET components categorized in three modes: disconnected, common or shared and the .NET data providers.

The disconnected components build the basic ADO.NET architecture. You can use these components (or classes) with or without data providers. For example, you can use a DataTable object with or without providers and shared or common components are the base classes for data providers. Shared or common components are the base classes for data providers and shared by all data providers. The data provider components are specifically designed to work with different kinds of data sources. For example, ODBC data providers work with ODBC data sources and OleDb data providers work with OLE-DB data sources.

Figure represents the ADO.NET components model and how they work together:



See for more detail:

* [Understanding ADO.NET Components](http://www.c-sharpcorner.com/uploadfile/mahesh/understanding-ado-net-components/)

**Question 3: What is the differences Between DataReader and DataSet?**

**Answer**

|  |  |  |
| --- | --- | --- |
| No | DataReader | DataSet |
| 1 | Used in a connected architecture | Used in a disconnected architecture. |
| 2 | Provides better performance | Provides lower performance. |
| 3 | DataReader object has read-only access | A DataSet object has read/write access |
| 4 | DataReader object supports a single table based on a single SQL query of one database | A DataSet object supports multiple tables from various databases. |
| 5 | A DataReader object is bound to a single control. | A DataSet object is bound to multiple controls. |
| 6 | A DataReader object has faster access to data. | A DataSet object has slower access to data. |
| 7 | A DataReader object must be manually coded. | A DataSet object is supported by Visual Studio tools. |
| 8 | We can't create a relation in a data reader. | We can create relations in a dataset. |
| 9 | Whereas a DataReader doesn't support data reader communicates with the command object. | A Dataset supports integration with XML Dataset communicates with the Data Adapter only. |
| 10 | DataReader cannot modify data. | A DataSet can modify data. |

See for more detail:

* [Database Programming With ADO.NET](http://www.c-sharpcorner.com/UploadFile/d0a1c8/database-programming-with-ado-net/)

**Question 4: What is SqlCommand Object?**

**Answer :** The SqlCommand carries the SQL statement that needs to be executed on the database. SqlCommand carries the command in the CommandText property and this property will be used when the SqlCommand calls any of its execute methods.

* 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.

The three important methods exposed by the SqlCommand object is shown below:

* ExecuteScalar
* ExecuteNonQuery
* ExecuteReader

**ExecuteScalar** is useful for returning a single value from the database. For example, using this method we can retrieve a sum of sales made by a specific product, total number of records in the employee table, unique id by supplying filtering conditions and so on. Since this method performs faster we do not need to go for the Reader method just to retrieve a single scalar value.

**ExecuteNonQuery** is useful for performing data manipulation on the database. Simply, the ExecuteNonQuery is for executing the DML statements. The return value of the ExecuteNonQuery is an integral value that represents the number of rows affected by the Operation.

**ExecuteReader** is used when we need to retrieve rows and columns of data using the SQL select statements. As the data retrieved is a table of data, ExecuteReader returns SqlDataReader. We should iterate through this object to get the required values.  
  
See for more detail:

* [Working With Command Object in ADO.NET](http://www.c-sharpcorner.com/UploadFile/c5c6e2/working-with-command-object/)

**Question 5: What is the difference between ADO and ADO.NET?**

**Answer:** Difference between ADO and ADO.NET.

|  |  |
| --- | --- |
| ADO | ADO.NET |
| ADO has one main object that is used to reference data, called the RecordSet object. | ADO.NET provides objects that allow you to access data in various ways. The DataSetobject allows you to store the relational model of your [database](http://www.c-sharpcorner.com/UploadFile/puranindia/what-is-ado-net/WhatisADONET.aspx). MARS (Multiple Active Result Sets) is implemented in ADO.NET |
| You can only work on connected manner. This means that when you access data, such as viewing and updating data, it is real-time, with a connection being used all the time. This is barring, of course, you programming special routines to pull all your data into temporary tables.  In connectedmodel you always get refreshed data. | ADO.NET uses data in a disconnected fashion. When you access data, ADO.NET makes a copy of the data using XML. ADO.NET only holds the connection open long enough to either pull down the data or to make any requested updates. This makes ADO.NET efficient to use for [Web applications.](http://www.c-sharpcorner.com/UploadFile/puranindia/what-is-ado-net/WhatisADONET.aspx) It's also decent for desktop applications.  You can work on connected and disconnected manner.br>  In disconnectedmodel you will get old data as you are editing it. Outlook is an example of disconnected model. We work on offline object model and when connection is required it is connected.  Connected object can be used on disconnected object. |
| Whereas ADO allows you to persist records in XML format. | ADO.NET allows you to manipulate your data using XML as the primary means. This is nice when you are working with other business applications and also helps when you are working with firewalls because data is passed as HTML and XML. |
| ADO allows you to create client-side cursors only. | ADO.NET gives you the choice of either using client-side or server-side cursors. In ADO.NET, classes actually handle the work of cursors. The developer has the freedom of choice in [internet](http://www.c-sharpcorner.com/UploadFile/puranindia/what-is-ado-net/WhatisADONET.aspx) development, for creating efficient applications. |

See for more detail:

* [What is ADO.NET?](http://www.c-sharpcorner.com/uploadfile/puranindia/what-is-ado-net/)

**Question 6: What is the DataAdapter Object in ADO.NET?**

**Answer :** A Data Adapter represents a set of data commands and a database connection to fill the dataset and update a SQL Server database.

A Data Adapter contains a set of data commands and a database connection to fill the dataset and update a SQL Server database. Data Adapters form the bridge between a data source and a dataset.

Data Adapters are designed depending on the specific data source. The following table shows the Data Adapter classes with their data source.

|  |  |
| --- | --- |
| Provider-Specific Data Adapter classes | Data Source |
| SqlDataAdapter | SQL Server |
| OledbDataAdapter | OLE DB provider |
| OdbcDataAdapter | ODBC driver |
| OracleDataAdapter | Oracle |

A Data Adapter supports mainly the following two methods:

* **Fill ():** The Fill method populates a dataset or a data table object with data from the database. It retrieves rows from the data source using the SELECT statement specified by an associated select command property.  
    
  The Fill method leaves the connection in the same state as it encountered before populating the data.
* **Update ():** The Update method commits the changes back to the database. It also analyzes the RowState of each record in the DataSet and calls the appropriate INSERT, UPDATE, and DELETE statements.

**Example:**

1. SqlDataAdapter da=**new** SqlDataAdapter("Select \* from
2. Employee", con);
3. da.Fill(ds,"Emp");
4. bldr =**new** SqlCommandBuilder(da);
5. dataGridView1.DataSource = ds.Tables["Emp"];

**Question 7: Use of DataSet object in ADO.NET?**

**Answer**

* It is used in a disconnected architecture.
* Provides lower performance. A DataSet object has read/write access.
* A DataSet object supports multiple tables from various databases.
* A DataSet object is bound to multiple controls.
* A DataSet object has slower access to data.
* A DataSet object is supported by Visual Studio tools.
* We can create relations in a dataset.
* A Dataset supports integration with XML.
* A DataSet communicates with the Data Adapter only.
* A DataSet can modify data.

A DataSet is a collection of DataTable and DataRelations. Each DataTable is a collection of DataColumn, DataRows and Constraints.

**Example**

1. DataTable dt = **new** DataTable();
2. DataColumn col =**new** DataColumn();
3. Dt.columns.Add(col2);
4. DataRow row = dt.newRow();

**Question 8: What is DataTable in ADO.NET?**

**Answer**

* DataTable represents a single table in a database.
* In this show row and column.
* DataSet is a collection of data tables.
* In this store data record.

DataTable representation in .aspx.cs code,

1. **protected** **void** BinddataTable()
2. {
3. SqlConnection con = **new** SqlConnection("your database connection string");
4. con.Open();
5. SqlCommand cmd = **new** SqlCommand("Write your query or procedure", con);
6. SqlDataAdapter da = **new** SqlDataAdapter(cmd);
7. DataTable dt = **new** DataTable();
8. da.Fill(dt);
9. grid.DataSource = dt;
10. grid.DataBind();
11. }

See for more detail:

* [ADO.NET Technique With ASP.NET](http://www.c-sharpcorner.com/UploadFile/b926a6/ado-net-technique-with-Asp-Net/)

**Question 9: What is the DataReader in ADO.Net?**

**Answer**

* DataReader holds only one table at a time.
* It only provides read only access mode and cannot write data.
* It is not required local storage to data store.
* Holds one row at a time.
* Uses less memory.
* DataReader do not maintain relation.

DataReader representation in .aspx.cs code,

1. **protected** **void** Bind()
2. {
3. SqlConnection con = **new** SqlConnection("your database connection string ");
4. con.Open();
5. SqlCommand cmd = **new** SqlCommand("Write your query or procedure ", con);
6. SqlDataReader dr = cmd.ExecuteReader();
7. grid.DataSource = dr;
8. grid.DataBind();
9. }

**The DataReader properties**

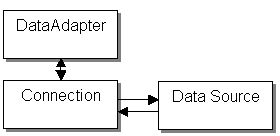
|  |  |
| --- | --- |
| Property | Description |
| Depth | Indicates the depth of nesting for row |
| FieldCount | Returns number of columns in a row |
| IsClosed | Indicates whether a data reader is closed |
| Item | Gets the value of a column in native format |
| RecordsAffected | Number of row affected after a transaction |

**The DataReader methods**

|  |  |
| --- | --- |
| Property | Description |
| Close | Closes a DataRaeder object. |
| Read | Reads next record in the data reader. |
| NextResult | Advances the data reader to the next result during batch transactions. |
| Getxxx | There are dozens of Getxxx methods. These methods read a specific data type value from a column. For example. GetChar will return a column value as a character and GetString as a string. |

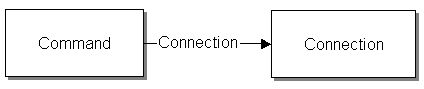
**Question 10: What is the Connection object in ADO.NET?**

**Answer :** A Connection object sits between a data source and a DataAdapter (via Command). You need to define a data provider and a data source when you create a connection. With these two, you can also specify the user ID and password depending on the type of data source. Figure 3-3 shows the relationship between a connection, a data source, and a data adapter.



**Figure:** The relationship between connection, data Adapter, and a data source.

Connection can also be connected to a Command object to execute SQL queries, which can be used to retrieve, add, update and delete data to a data source. Figure 2 shows the relationship between the Command and Connection objects.



**Data provider connection classes**

|  |  |
| --- | --- |
| Data provider | Connection classes |
| OleDb | OleDbConnection |
| Sql | SqlConnection |
| ODBC | OdbcConnection |

See for more detail:

* [Connection Object in ADO.NET](http://www.c-sharpcorner.com/uploadfile/mahesh/connection-object-in-ado-net/)

**Question 11: Describe the DataView in ADO.NET?**

**Answer:** A DataView enables you to create different views of the data stored in a DataTable, a capability that is often used in data binding applications. Using a DataView, you can expose the data in a table with different sort orders, and you can filter the data by row state or based on a filter expression. A DataView provides a dynamic view of data whose content, ordering, and membership reflect changes to the underlying DataTable as they occur. This is different from the Select method of the DataTable, which returns a DataRow array from a table per particular filter and/or sort order and whose content reflects changes to the underlying table, but whose membership and ordering remain static. The dynamic capabilities of the DataView make it ideal for data-binding applications.

**How we can create a DataView**

There are two ways to create a DataView. You can use the DataView constructor, or you can create a reference to the DefaultView property of the DataTable. The DataView constructor can be empty, or will also take either a DataTable as a single argument, or a DataTable along with filter criteria, sort criteria, and a row state filter.

1. DataView custDV = **new** DataView(customerDS.Tables["Customers"],
2. "Country = 'USA'",
3. "ContactName",
4. DataViewRowState.CurrentRows);
5. DataView custDV = customerDS.Tables["Customers"].DefaultView;

See for more detail:

* [DataView in C#](http://www.c-sharpcorner.com/UploadFile/jgodel/DataSetsOverview0611302005002923AM/DataSetsOverview06.aspx)

**Question 12: What is ExecuteScalar method in ADO.NET?**

**Answer:**

**ExecuteScalar Method**

The ExecuteScalar method of the SqlCommand object is useful for retrieving a single value from the database. In our example, we need to retrieve the total number of records in the Titles table of the Pubs database. Since the total number of records is a single scalar value, the Execute Scalar method is used. The following is the code and its explanation:

1. **private** **void** frmSqlCommand\_Load(**object** sender, EventArgs e)
2. {
3. //Sample 03: Open Database Connection
4. String con\_string = Properties.Settings.Default.ConStrPubs;
5. pubs\_db\_connection = **new** SqlConnection(con\_string);
6. pubs\_db\_connection.Open();
7. //Sample 04: Form the Command Object
8. SqlCommand cmd = **new** SqlCommand();
9. cmd.CommandText = "Select Count(\*) as Count from Titles";
10. cmd.Connection = pubs\_db\_connection;
11. //Sample 05: Execute the Command & retrive scalar value
12. lblTotal.Text = System.Convert.ToString(cmd.ExecuteScalar());
13. }

See for more detail:

* [Understanding SqlCommand and DataReader in ADO.NET](http://www.c-sharpcorner.com/UploadFile/6897bc/understanding-sqlcommand-and-datareader/)

**Question 13: What are the methods of DataSet?**

**Answer:** It is used in disconnected architecture. It represent records in the form of Database table (Row and Column) format. It stores record of one or more tables.

1. SqlDataAdapter da;
2. DataSet ds;
3. **string** strconn = "Data Source=YourServerName;Initial Catalog=EMP;Integrated Security=True";
4. **private** **void** Form1\_Load(**object** sender, EventArgs e)
5. {
6. da = **new** SqlDataAdapter("select \* from userdet", strconn);
7. ds = **new** System.Data.DataSet();
8. da.Fill(ds);
9. dataGridView1.DataSource = ds.Tables[0];
10. }

**Methods of DataSet:**

* **AcceptChanges():** This method saves changes which are made with records in a DataSet.
* **Clear():** This method clears (removes) all rows from DataSet.
* **Clone():** The clone method copy the structure of DataSet. Means it copy only schema not full records of DataSet.
* **Copy():** It copies the whole records with structure of DataSet.
* **RejectChanges():** This method discard changes which is made with DataSet and set the DataSet to previous stage (which was at first).
* **HasChanges():** This method return boolean value to show whether record of DataSet has changed or not. It returns true if any changes has made and false if no other changes made.
* **GetChanges():** This method keep copy of those record, which is changed or modified.

**Question 14 : What you understand by ExecuteNonQuery Method?**

**Answer :** The ExecuteNonQuery method is used to execute the command and return the number of rows affected.

The ExecuteNonQuery method cannot be used to return the result set.

**Snippets working with ExecuteNonQuery**

1. **public** **void** CallExecuteNonQuery()
2. {
3. SqlConnection conn = **new** SqlConnection();
4. conn.ConnectionString = ConfigurationManager.ConnectionStrings["connString"].ConnectionString;
5. **try**
6. {
7. SqlCommand cmd = **new** SqlCommand();
8. cmd.Connection = conn;
9. cmd.CommandText = "DELETE FROM EMP WHERE DEPTNO = 40";
10. cmd.CommandType = CommandType.Text;
11. conn.Open();
12. Int32 RowsAffected = cmd.ExecuteNonQuery();
13. MessageBox.Show(RowsAffected + " rows affected", "Message");
14. cmd.Dispose();
15. conn.Dispose();
16. }
17. **catch**(Exception ex)
18. {
19. MessageBox.Show(ex.Message);

**Question 15: What are the important classes in ADO.NET?**

**Answer:** ADO.NET is a set of classes (a framework) to interact with data sources such as databases and XML files. ADO is the acronym for ActiveX Data Object. It allows us to connect to underlying data or databases. It has classes and methods to retrieve and manipulate data.

The following are a few of the .NET applications that use ADO.NET to connect to a database, execute commands and retrieve data from the database.

* ASP.NET Web Applications
* Console Applications
* Windows Applications.

Important classes in ADO.NET

1. Connection Class
2. Command Class
3. DataReader Class
4. DataAdaptor Class
5. DataSet.Class

**How to Connect to a Database using ADO.NET:**

To create a connection, you must be familiar with connection strings. A connection string is required as a parameter to SQLConnection. A ConnectionString is a string variable (not case sensitive).

This contains key and value pairs, like provider, server, database, userid and word as in the following:

*Server="nameof the server or IP Address of the server"  
Database="name of the database"  
userid="user name who has permission to work with database"  
word="the word of userid"*

**Question 16: What are the two fundamental objects in ADO.NET?**

**Answer:** There are the two fundamental objects in ADO.NET:

* Data reader
* Data set

**DataReader Class:**

The DataReader class object allows you to read the data returned by a SELECT command by a simple forward-only and read-only cursor. It requires a live connection with the data source and provides a very efficient way of looping and consuming all parts of the result set. The object of the DataReader cannot be directly instantiated. Instead you must call the ExecuteReader method of the Command object and close the connection when you are done using the DataReader otherwise the connection remains alive until it is explicitly closed.

**DataReader with ExecuteReader() Method:**

1. //Open connection
2. Conn.Open();
3. sdr = sc.ExecuteReader(CommandBehavior.CloseConnection);
4. //Get all records
5. **while**(sdr.Read())
6. {
7. textBox1.AppendText(sdr.GetValue(0) + "\t" + sdr.GetValue(1));
8. textBox1.AppendText("\n");
9. }

**DataSet class:**

A DataSet is a disconnected architecture technology. It contains zero or more tables and relationships. When you work with a dataset, the data in the data source is not touched at all. Instead all the changes are made locally to the dataset in memory. In the following example you will see how to retrieve data from a SQL Server table and use it to fill in a DataTable object in the DataSet.

1. **private** **void** Form1\_Load(**object** sender, EventArgs e)
2. {
3. //Connection String
4. **string** conString = "Data Source=localhost;Database=AdventureWorksLT2008;Integrated Security=SSPI";
5. // Add Connection string to SqlConnection
6. SqlConnection Conn = **new** SqlConnection(conString);
7. **string** query = "select \* from SalesLT.Customer";
8. //Command Class definition
9. SqlCommand sc = **new** SqlCommand(query, Conn);
10. // Data Adapter definition
11. SqlDataAdapter sda = **new** SqlDataAdapter();
12. sda.SelectCommand = sc;
13. //data Set definition
14. DataSet ds = **new** DataSet();
15. // filling the result set in data table
16. sda.Fill(ds, "SalesLT.Customer");
17. //output in data grid
18. dataGridView1.DataSource = ds.Tables["SalesLT.Customer"];
19. }

**Question 17: What is DataAdapter and its property?**

**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.

1. // Data Adapter definition
2. SqlDataAdapter sda = **new** SqlDataAdapter(sc);

These are the commonly used properties offered by the SqlDataAdapter class as in the following:

|  |  |
| --- | --- |
| **Property** | **Description** |
| 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. |

See for more detail:

* [Overview of ADO.NET Architecture](http://www.c-sharpcorner.com/UploadFile/ajyadav123/ado-net-architecture/)

**Question 18: Which namespaces are used for data access?**

**Answer:** ADO.NET is a collection of managed libraries used by .NET applications for data source communication using a driver or provider.

ADO.NET provides libraries for the datasource communication under the following namespaces.

1. system.Data
2. system.Data.OleDb
3. system.Data.SqlClient
4. system.Data.OracleClient
5. system.Data.Odbc
6. **System.Data:** This namespace is used for holding and managing data on a client machine.
7. **System.Data.OleDb:** This namespace can communicate with any data source like files, databases, indexing servers and so on using the “**OleDb**” Provider.
8. **System.Data.SqlClient:** This namespace can communicate with “**SQL Server**” database only using SqlClient Providers.
9. **System.Data.OracleClient:** This namespace can communicate with an “**Oracle**” database only using OracleClient Providers.
10. **System.Data.ODBC**: This namespace contains the same set of classes as the following:  
      
    * Connection
    * Command
    * DataReader
    * DataAdaptar
    * CommandBuilder
    * Parameter

**Question 19: Explain the properties and methods of Command Object.**

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

1. The Command Object uses the connection object to execute SQL queries.
2. The queries can be in the form of Inline text, Stored Procedures or direct Table access.
3. An important feature of Command object is that it can be used to execute queries and Stored Procedures with Parameters.
4. If a select query is issued, the result set it returns is usually stored in either a DataSet or a DataReader object.

**Associated Properties of SqlCommand class**

|  |  |  |
| --- | --- | --- |
| Property | Type of Access | Description |
| Connection | Read/Write | The SqlConnection object that is used by the command object to execute SQL queries or Stored Procedure. |
| CommandText | Read/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 |
| CommandTimeout | Read/Write | This property indicates the time to wait when executing a particular command.  **Default Time for Execution of Command is 30 Seconds.**  The Command is aborted after it times out and an exception is thrown. |

Now, let us have a look at various execute methods that can be called from a Command Object.

|  |  |
| --- | --- |
| Property | Description |
| ExecuteNonQuery | This method executes the command specifies and returns the number of rows affected. |
| ExecuteReader | The ExecuteReader method executes the command specified and returns an instance of instance of SqlDataReader class. |
| ExecuteScalar | This method executes the command specified and returns the first column of first row of the result set. The remaining rows and column are ignored |
| ExecuteXMLReader | This method executes the command specified and returns an instance of XmlReader class. This method can be used to return the result set in the form of an XML document |

See for more detail:

* [Working With Command Object in ADO.NET](http://www.c-sharpcorner.com/UploadFile/c5c6e2/working-with-command-object/)

**Question 20: Explain the ExecuteReader method**

1. The DataReader object is a forward-only and read-only cursor.
2. It requires a live connection to the data source.
3. The DataReader object cannot be directly instantiated. Instead, we must call the ExecuteReader() method of the command object to obtain a valid DataReader object.
4. SqlDataReader reader = cmd.ExecuteReader(CommandBehavior.CloseConnection);

## Question 21: What are the Connection object properties and Connection class members?

**Answer:** The Connection class has a connection string that opens a connection to the database. The connection string will vary depending upon the provider used. The connection strings typically contain a group of property-value pair to describe how to connect to a database. For an OleDbConnection, you have properties such as Provider and DataSource.

|  |  |
| --- | --- |
| Property | Description |
| ConnectionString | Represent the connection string. |
| ConnectionTimeOut | Waiting time while establishing a connection. |
| DataBase | Name of the current database. |
| DataSource | Location of the file name of the data source. |
| Provider | Name of the OLE DB provider. This property is not available for Sql and ODBC data providers. |
| State | Current state of the connection of type ConnectionState. (Table 5-17 describes the ConnectionState). |
| PacketSize | Size of network packets. Available to only Sql data providers. |
| ServerVersion | SQL server version. Available to only Sql data providers. |
| WorkStationId | Database client ID. Available to only Sql data providers. |

The connection can have different states such as open, closed, connecting, and so on. The ConnectionType enumeration defines the members of the ConnectionState.

The connection Class Members

|  |  |
| --- | --- |
| **Method** | **Description** |
| BeginTransaction | Begins database transaction. |
| ChangeDatabase | Changes databases for an open connection. |
| Close | Closes an opened connection. |
| CreateCommand | Creates and return a Command object depends on the data providers. For example, OleDb Connection returns OleDbCommand, and SqlConnection returns SqlCommand. |
| Open | Open a new connection. |
| ReleaseObjectPool | Represents that the connection pooling can be cleared when the provider is released. Available only for Ole Db data providers. |

**Question 22: What is the ADO.NET Data provider?**

**Answer:** There are four .NET data providers available.

1. **SQL Server:** It is used to work specifically with Microsoft SQL Server. It exists in a namespace within the System.Data.SqlClient.
2. **OLE DB:** It is used to work with the OLEDB provider. The System.Data.dll assembly implements the OLEDB .NET framework data provider in the System.Data.OleDb namespace.
3. **ODBC:** To use this type of provider, you must use an ODBC driver. The System.Data.ODBC.dll assembly implements the ODBC .NET framework data provider. This assembly is not part of the Visual Studio .NET installation.
4. **Oracle:** The System.Data.OracleClient.dll assembly implements the Oracle .NET framework data provider in the System.Data.OracleClient namespace. The Oracle client software must be installed on the system before you can use the provider to connect to an Oracle data source.

**Question 23: What is disconnected data?**

**Answer:** A data representation, such a DataSet, that doesn't require a continuous database connection. Working with disconnected data:

The data in DataSet is disconnected from database. Once you fetch the results of a query into a DataSet using a DataAdapter object, there is no longer a connection between DataSet and database. Changes you make to the contents of the DataSet will not affect the database. If other users modify data in database that corresponds to the data in DataSet, you will not see those changes in your DataSet.

Working with disconnected data structures definitely has its benefits. The first major benefit of working with disconnected data is that it does not require a live connection to your database. Once you've fetched the results of your query into a DataSet object, you can close the connection to your database and continue to work with the data in your DataSet.

Disconnected data structures such as DataSets are also helpful when you build multi-tiered applications. If your application uses business objects running on a middle-tier server to access database, business object needs to pass disconnected data structures to client application. The DataSet object is designed for use in such situations. You can pass the contents of a DataSet from one component to another. The component that receives the data can work with the information as a DataSet (if the component is built using the Microsoft .NET Framework) or as an XML document.

See for more detail:

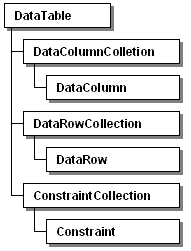
* [DataSets in Microsoft .NET](http://www.c-sharpcorner.com/UploadFile/raghavnayak/DataSetsIn.NET12032005003647AM/DataSetsIn.NET.aspx)

**Question 24: Explain the DataTable and Relationship between the DataTable, the DataRow, and the DataColumn**.

**Answer:** A DataTable object represents a database table. A data table is a collection of columns and rows. The DataRow object represents a table row, and the DataColumn object represents a column of the table.

The Columns property of the DataTable object represents the DataColumnCollection, which is a collection of DataColumn objects in a DataTable. You use a DataRow object to add data to a data table. TheDataRowCollection object represents a collection of rows of a DataTable object, which can be accessed by its Rows property.

Figure shows the relationship between the DataTable, DataRow, and DataColumn.



See for more detail:

* [DataTable, DataColumn and DataRow classes](http://www.c-sharpcorner.com/uploadfile/mahesh/datatable-datacolumn-and-datarow-classes/)

## Question 25: What are the advantages of ADO.NET?

**Answer: Advantages of ADO.NET**

ADO.NET offers several advantages over previous Microsoft data access technologies, including ADO. Few advantages are listed below:

**Single Object-oriented API**

ADO.NET provides a single object-oriented set of classes. There are different data providers to work with different data sources but the programming model for all these data providers work in the same way.

**Managed Code**

The ADO.NET classes are managed classes. CLR takes care of language independency and automatic resource management.

**Deployment**

Microsoft uses MDAC (Microsoft Data Access Component), which is used as ActiveX component in .NET Framework (X is extensible component, when X is written after a term means extensible). .NET components takes care of deployment which was difficult than the previous technologies used in deployment.

**XML Support**

ADO.NET data is cached and transferred in XML (EXtensible Markup Language) format. XML provide fast access of data for desktop and distributed applications.

**Performance and scalability**

Performance and scalability are two major factors while developing web-based application and services. Disconnected cached data in XML help in performance and scalability.

.

**Q26:Explain Generic Features of ADO.Net ?**

* ADO.Net provides in built classes to connect with Databases like Oracle, MySQL, SQL Server, MS Access etc.
* Provides in built classes to do data manipulation operations like Insert, Update, Delete and Select data.
* Provides data providers for specific databases for efficient interactions with DB. Example - ODP.Net provider for Oracle.
* Tight integration with XML
* Provides functionality to combine data from different data sources
* Disconnected Data architecture for better performance