

ADO.NET Interview Questions And Answers



C# Curator



Updated date May 21 2019



707k



0



25



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](#)

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](#)

Question 3: How can you define the DataSet structure?

Answer 3: A DataSet object falls in disconnected components series. The DataSet consists of a collection of tables, rows, columns and relationships.

DataSet contains a collection of DataTables and the DataTable contains a collection of DataRows, DataRelations, and DataColumns. A DataTable maps to a table in the database. The previous DataSet contains a DataTable that maps to the Orders table because you filled it with a SELECT query executed on the Order table.

See for more detail:

- [Data Set and Data View objects in ADO.NET](#)

Question 4: What is Connection Pooling in ADO.NET?

Answer 4: Connection pooling is the ability of reusing your connection to the database. This means if you enable Connection pooling in the connection object, actually you enable the re-use of the connection to more than one user.

ADO.NET uses a technique called connection pooling, which minimizes the cost of repeatedly opening and closing connections. Connection pooling reuses existing active connections with the same connection string instead of creating new connections when a request is made to the database. It involves the use of a connection manager that is responsible for maintaining a list, or pool, of available connections for a given connection string. Several pools exist if different connection strings ask for connection pooling.

Example of Pooling:

```
01. connection.ConnectionString = sqlConnectionString + "Connection Timeout=30;Connection Lifet.  
02. //Open connection
```

A Connection String in the Web.Config file with connection pooling option:

```
01. <connectionStrings>  
02.   <clear />  
03.   <add name="sqlConnectionString" connectionString="Data Source=mysqlServer;Initial Cata  
04. </connectionStrings>
```

SQL Server connection string pooling attributes:

- **Connection Lifetime:** Length of time in seconds after creation after which a connection is destroyed. The default is 0, indicating that connection will have the maximum timeout.
- **Connection Reset:** Specifies whether the connection is reset when removed from the pool. The default is true.
- **Enlist:** Specifies whether the connection is automatically enlisted in the current transaction context of the creation thread if that transaction context exists. The default is true.
- **Load Balance Timeout:** Length of time in seconds that a connection can remain idle in a connection pool before being removed.
- **Max Pool Size:** Maximum number of connections allowed in the pool. The default is 100.
- **Min Pool Size:** Minimum number of connections maintained in the pool. The default is 0.
- **Pooling:** When true, the connection is drawn from the appropriate pool, or if necessary, created and added to the appropriate pool. The default is true.

See for more detail:

- [Connection Pooling \(ADO.NET\)](#)

Question 5: 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](#)

Question 6: What is SqlCommand Object?

Answer 6: 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](#)

Question 7: 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 . 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 . 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 development, for creating efficient applications.

See for more detail:

- [What is ADO.NET?](#)

Question 8: What is the DataAdapter Object in ADO.NET?

Answer 8: 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:

```

01. SqlDataAdapter da=new SqlDataAdapter("Select * from
02. Employee", con);
03. da.Fill(ds,"Emp");
04. bldr =new SqlCommandBuilder(da);
05. dataGridView1.DataSource = ds.Tables["Emp"];

```

See for more detail:

- [Database Programming With ADO.NET](#)

Question 9: 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

```

01. DataTable dt = new DataTable();
02. DataColumn col =new DataColumn();
03. Dt.columns.Add(col2);
04. DataRow row = dt.newRow();

```

See for more detail:

- [Database Programming With ADO.NET](#)

Question 10: Describe the System.Data Namespace Class?

Answer: The three general namespaces are System.Data, System.Data.Common and System.Data.SqlTypes. Some of the provider-specific namespaces are System.Data.OleDb, Microsoft.Data.Odbc and System.Data.SqlClient.

The System.Data namespace defines classes that you can use with all the data providers or without data providers at all. This namespace also defines interfaces that are base classes for the data provider classes. The following figure shows the System.Data namespace class hierarchy.



See for more detail:

- [ADO .NET Class Hierarchy](#)

Question 11: 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,

```

01. protected void BinddataTable()
02. {
03.     SqlConnection con = new SqlConnection("your database connection string");
04.     con.Open();
05.     SqlCommand cmd = new SqlCommand("Write your query or procedure", con);
06.     SqlDataAdapter da = new SqlDataAdapter(cmd);
07.     DataTable dt = new DataTable();
08.     da.Fill(dt);
09.     grid.DataSource = dt;
10.     grid.DataBind();
11. }

```

See for more detail:

- [ADO.NET Technique With ASP.NET](#)

Question 12: 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,

```

01. protected void Bind()
02. {
03.     SqlConnection con = new SqlConnection("your database connection string ");
04.     con.Open();
05.     SqlCommand cmd = new SqlCommand("Write your query or procedure ", con);
06.     SqlDataReader dr = cmd.ExecuteReader();
07.     grid.DataSource = dr;
08.     grid.DataBind();
09. }

```

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.

See for more detail:

- [DataReader in ADO.NET](#)

Question 13: What is the SqlCommandBuilder?

Answer: CommandBuilder helps you to generate update, delete, and insert commands on a single database table for a data adapter. Similar to other objects, each data provider has a command builder class. The OleDbCommandBuilder,

SqlCommandBuilder, and OleDbCommandBuilder classes represent the CommonBuilder object in the OleDb, Sql, and ODBC data providers.

Creating a Command Builder Object:

Creating a CommonedBuider object is pretty simply. You pass a DataAdapter as an argument of the CommandBuilder constructor. For example,

```
01. // Create a command builder object
02. SqlCommandBuilder builder = new SqlCommandBuilder(adapter);
```

See for more detail:

- [CommandBuilder in ADO.NET](#)

Question 14: 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](#)

Question 15: 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.

```
01. DataView custDV = new DataView(customerDS.Tables["Customers"],
02. "Country = 'USA'",
03. "ContactName",
04. DataRowState.CurrentRows);
05. DataView custDV = customerDS.Tables["Customers"].DefaultView;
```

See for more detail:

- [DataView in C#](#)

Question 16: 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:

```
01. private void frmSqlCommand_Load(object sender, EventArgs e)
02. {
03.     //Sample 03: Open Database Connection
04.     String con_string = Properties.Settings.Default.ConStrPubs;
05.     pubs db connection = new SqlConnection(con_string);
06.     pubs db connection.Open();
07.     //Sample 04: Form the Command Object
08.     SqlCommand cmd = new SqlCommand();
09.     cmd.CommandText = "Select Count(*) as Count from Titles";
10.     cmd.Connection = pubs db connection;
11.     //Sample 05: Execute the Command & retrieve scalar value
12.     lblTotal.Text = System.Convert.ToString(cmd.ExecuteScalar());
13. }
```

See for more detail:

- [Understanding SqlCommand and DataReader in ADO.NET](#)

Question 17: 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.

```
01. SqlDataAdapter da;
02. DataSet ds;
03. string strconn = "Data Source=YourServerName;Initial Catalog=EMP;Integrated Security=True";
04. private void Form1_Load(object sender, EventArgs e)
05. {
06.     da = new SqlDataAdapter("select * from userdet", strconn);
07.     ds = new System.Data.DataSet();
08.     da.Fill(ds);
09.     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.

See for more detail:

- [Working With DataSet and It's Methods in ADO.NET](#)

Question 18: What are Parameters in ADO.NET?

Answer: Parameters in conjunction with SelectCommand to help you Select data for the DataSet.

The OleDbType describes the type information for the parameter. It consists of everything from strings to Global Unique Identifiers (GUIDs). SQL data provider has SqlDbType, and ODBC data provider has an ODBC type. These type names and definitions differ depending upon the provider you're using. For example, the Money type is the same in ODBC and Sqldata providers, but is called Currency in OleDb data providers.

The OLEDB Parameter Class properties

Property	Description
DbType	Represents the DbType of the parameter.
Direction	Represents the direction of a parameter. A parameter can be input-only, output-only, bi-directional, or a stored procedure.
IsNullable	Represents whether a parameter accepts null values.
OleDbType	Represents the OleDbType of the parameter.
ParameterName	Represents the name of the parameter.
Precision	Represents the maximum number of digits used to represent the Value property.
Scale	Represents the decimal places to which Value is resolved.
Size	Represents the maximum size in bytes a column can store.
SourceColumn	Represents the source column mapped to the DataSet.
SourceVersion	Represents the DataRowVersion.
Value	Represents the Value of the parameter.

Creating a parameter

```
this.oleDbDeleteCommand2.Parameters.Add(new System.Data.OleDb.OleDbParameter("ContactName",  
System.Data.OleDb.OleDbType.char, 30, System.Data.ParameterDirection.Input, false, ((System.Byte)(0)),((System.Byte)(0)),  
"Contact Name", System.Data.DataRowVersion.Original, null));
```

See for more detail:

- [Parameters in ADO.NET](#)

Question 19: What is the difference between DataSet and DataReader?

Answer:

DataReader

1. The ADO.NET DataReader is used to retrieve read-only (cannot update data back to datasource) and forward-only (cannot read backward/random) data from a database.
2. Using the DataReader increases application performance and reduces system overheads. This is due to one row at a time is stored in memory.
3. You create a DataReader by calling Command.ExecuteReader after creating an instance of the Command object.
4. This is a connected architecture: The data is available as long as the connection with database exists.
5. You need to open and close the connection manually in code.

DataSet

1. The DataSet is a in-memory representation of data.
2. It can be used with multiple data sources. That is a single DataSet and can hold the data from different data sources holding data from different databases/tables.
3. The DataSet represents a complete set of data including related tables, constraints, and relationships among the tables.

4. The DataSet can also persist and reload its contents as XML and its schema as XML Schema definition language (XSD) schema.
5. The DataAdapter acts as a bridge between a DataSet and a data source for retrieving and saving data.
6. The DataAdapter helps mapping the data in the DataSet to match the data in the data source.
7. Also, upon an update of dataset, it allows changing the data in the data source to match the data in the DataSet.
8. No need to manually open and close connection in code.
9. Hence, point (8) says that it is a disconnected architecture. Fill the data in DataSet and that's it. No connection existence required.

See for more detail:

- [DataReader Vs DataSet?](#)

Question 20 : What you understand by ExecuteNonQuery Method?

Answer 20: 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

```
01. public void CallExecuteNonQuery()  
02. {  
03.     SqlConnection conn = new SqlConnection();  
04.     conn.ConnectionString = ConfigurationManager.ConnectionStrings["connString"].Connect  
05.     try  
06.     {  
07.         SqlCommand cmd = new SqlCommand();  
08.         cmd.Connection = conn;  
09.         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);  
20.     }  
21. }
```

See for more detail:

- [Working With Command Object in ADO.NET](#)

Question 21: What do you understand by DataRelation class?

Answer: The DataRelation is a class of disconnected architecture in the .NET framework. It is found in the System.Data namespace. It represents a relationship between database tables and correlates tables on the basis of matching column.

```
01. DataRelation drel;  
02. drel = new DataRelation("All", ds.Tables[0].Columns[0], ds.Tables[1].Columns[0]);
```

See for more detail:

- [DataRelation class in ADO.NET](#)

Question 22: How can create a SqlConnection?

Answer: SqlConnection class is used to establish the connection between front end and back end.

Syntax:

SqlConnection obj=new SqlConnection("Integrated Security=true;Initial Catalog=Table_Name;Data Source=.");-- for Windows authentication

SqlConnection obj=new SqlConnection("user id= sa ; Password=sa123;server=.;database=name"); --Sql server Authentication
See for more detail:

- [Overview Of ADO.NET](#)

Question 23: 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 SqlConnection 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"

See for more detail:

- [Basics of ADO.NET](#)

Question 24: What do you understand by SqlTransaction class in ADO.NET?

Answer: The SqlTransaction class is an important class of .NET Framework. It ensures that a body of code will affect a Database or kept the same as previous (Rollback).

At first we should know about its two most important method which will be used here. They are given below.

- **Commit():** It commits the transaction. It save changes made in Database during transaction. In simple term we can also say that it shows the end of transaction at that time.
- **Rollback():** It is used to rollback the transaction. It set the database in previous stage which was, before the begin of transaction.

See for more detail:

- [Working with the SqlTransaction Class in ADO.NET](#)

Question 25: 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:

```
01. //Open connection
02. Conn.Open();
03. sdr = sc.ExecuteReader(CommandBehavior.CloseConnection);
04. //Get all records
05. while(sdr.Read())
06. {
07.     textBox1.AppendText(sdr.GetValue(0) + "\t" + sdr.GetValue(1));
08.     textBox1.AppendText("\n");
09. }
```

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.

```
01. private void Form1_Load(object sender, EventArgs e)
02. {
03.     //Connection String
04.     string conString = "Data Source=localhost;Database=AdventureWorksLT2008;Integrated S
05.     // Add Connection string to SqlConnection
06.     SqlConnection Conn = new SqlConnection(conString);
07.     string query = "select * from SalesLT.Customer";
08.     //Command Class definition
09.     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. }
```

See for more detail:

- [Overview of ADO.NET Architecture](#)

Question 26: 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.

```
01. // Data Adapter definition
02. 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](#)

Question 27: 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

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

See for more detail:

- [Introduction To ADO.NET](#)

Question 28: 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	Description
----------	---------	-------------

	Access	
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](#)

Question 29: 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.

```
01. | SqlDataReader reader = cmd.ExecuteReader(CommandBehavior.CloseConnection);
```

- [Working With Command Object in ADO.NET](#)

Question 30: Explain the ExecuteScalar method in ADO.NET?

Answer:

The ExecuteScalar Method in SqlCommandObject returns the first column of the first row after executing the query against the Data Source.

1. If the result set contain more than one column or rows, it takes only the first column of the first row. All other values are ignored.
2. If the result set is empty it will return null.

```
01. | Int32 TotalSalary = Convert.ToInt32(cmd.ExecuteScalar());
```

See for more detail:

- [Working With Command Object in ADO.NET](#)

Question 31: Explain the ExecuteXmlReader?

Answer: The execute reader method is flexible when we need the result set in the form of an XML document. The ExecuteXmlReader methods returns an instance of XmlReader class.

Example: `XmlReader xmlreader = cmd.ExecuteXmlReader();`
`XmlDocument xdoc = new XmlDocument();`

Using the XmlDocument class we load the XmlReader object and save it to the File System using the Save method.

See for more detail:

- [Working With Command Object in ADO.NET](#)

Question 32: What method in the OleDbDataAdapter class populates a dataset with records?

Answer: The Fill method populates a dataset with records.

```
01. private void button1_Click(object sender, EventArgs e)
02. {
03.     SqlConnection con;
04.     SqlDataAdapter adapter;
05.     DataSet ds = new DataSet();
06.     try
07.     {
08.         //create connection object
09.         con = new SqlConnection("connetionString");
10.         //create query string(SELECT QUERY)
11.         String query = "SELECT * FROM SOMETABLE WHERE...";
12.         con.Open();
13.         //Adapter bind to query and connection object
14.         adapter = new SqlDataAdapter(query, con);
15.         //fill the dataset
16.         adapter.Fill(ds);
17.     }
18.     catch(Exception ex)
19.     {
20.         con.Close();
21.     }
22. }
```

Your DataSet contains the result table of SELECT query. You can play around the table and its data.

- [SqlDataAdapter Fill Method](#)

Question 33: Explain the OleDbDataAdapter Command Properties with Examples?

Answer:

Property	Example
SelectCommand	cmd.SelectCommand.CommandText = "SELECT * FROM Orders ORDER BY Price"
DeleteCommand	TheDataSetCommand.DeleteCommand.CommandText = "DELETE FROM orders WHERE LastName = 'Smith' ";
InsertCommand	TheDataSetCommand.InsertCommand.CommandText = "INSERT INTO Orders VALUE (25, 'Widget1', 'smith')";
UpdateCommand	TheDataSetCommand.UpdateCommand.CommandText = "UPDATE Orders SET ZipCode = '34956' WHERE OrderNum = 14";

The OLEDB Data Adapter methods

Method	Description
Fill	This method fills data records from a DataAdapter to a DataSet object.
FillSchema	This method adds a DataTable to a DataSet.
GetFillParameters	This method retrieves parameters that are used when a SELECT statement is executed.
Update	This method stores data from a data set to the data source.

See for more detail:

- [DataAdapter Properties and Methods in ADO.NET](#)

Question 34: Explain the clear() method of DataSet?

Answer: This method clears (removes) all rows from DataSet.

Create a button, set it's text as Clear and write the following code in the click event.

```
01. private void btnClear_Click(object sender, EventArgs e)
02. {
03.     ds.Clear();
04. }
05. Run the application.
```

Output:

Click the "clear" button. You will note that all rows are clear as in the following screenshot,

See for more detail:

- [Working With DataSet and It's Methods in ADO.NET](#)

Question 35: What is clone() method of DataSet?

Answer: The clone method copy the structure of DataSet. Means it copy only schema not full records of DataSet.

Take another DataGridView and one button in your project and write the following code on button click event.

```
01. private void btnClone_Click(object sender, EventArgs e)
02. {
03.     DataSet daset = ds.Clone();
04.     dataGridView2.DataSource = daset.Tables[0];
05. }
```

Now run the application.

Output

Click the "**Clone**" button. The following figure shows that only structure has copied.

See for more detail:

- [Working With DataSet and It's Methods in ADO.NET](#)

Question 36: What is the Copy() method of DataSet?

Answer: Copy the whole records with structure of DataSet.

Take a button and set it's text as copy and write the following code on click event.

```
01. private void btnCopy_Click(object sender, EventArgs e)
02. {
03.     DataSet daset = ds.Copy();
04.     dataGridView2.DataSource = daset.Tables[0];
05. }
```


Now run the application.

Output

Click the copy button.

See for more detail:

- [Working With DataSet and It's Methods in ADO.NET](#)

Question 37: What is the HasChanges() method of DataSet?

Answer: This method return boolean value to show whether record of DataSet has changed or not. It returns true if any changes made and false if no changes performed.

Take a button and set it's text as "HasChanges" and write the following code on button click.

```
01. private void btnHasChanges_Click(object sender, EventArgs e)
02. {
03.     if(ds.HasChanges())
04.     {
05.         MessageBox.Show("Changes Has Made");
06.     }
07.     if(!ds.HasChanges())
08.     {
09.         MessageBox.Show("No Change");
10.     }
11. }
```

Run the application.

Output

Now click at "HasChanges" button after doing some changes in dataset record.

Output

See for more detail:

- [Working With DataSet and It's Methods in ADO.NET](#)

Question 38: 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
<code>BeginTransaction</code>	Begins database transaction.
<code>ChangeDatabase</code>	Changes databases for an open connection.
<code>Close</code>	Closes an opened connection.
<code>CreateCommand</code>	Creates and return a Command object depends on the data providers. For example, <code>OleDbConnection</code> returns <code>OleDbCommand</code> , and <code>SqlConnection</code> returns <code>SqlCommand</code> .
<code>Open</code>	Open a new connection.
<code>ReleaseObjectPool</code>	Represents that the connection pooling can be cleared when the provider is released. Available only for <code>OleDb</code> data providers.

- [Connecting to the Database in ADO.NET](#)

Question 39: What is the preferred method for executing SQL commands that contain parameters?

Answer: The preferred method is to use the `SqlParameter` and `oleDbParameter` objects, as detailed in the section "Using Parameters with the command Object."

The `SqlParameter` class is found in the **"System.Data.SqlClient"** namespace. It is a class of a connected architecture of .NET framework. It represents parameters. To work with the `SqlParameter` class we should have a database.

See for more detail:

- [Working With the SqlParameter Class in ADO.NET](#)

Question 40: Hpw DataSet objects in ADO.NET replace the ADO Recordset object?

Answer : DataSet is good for ADO.NET objects to replace the ADO Recordset object:

- DataSet can hold multiple tables at a time.
- It allows data access to easily read or write data operations in / from the database.
- DataSet data stores in local system.
- It holds multiple rows at a time.
- It uses more memory.
- DataSet maintain relation.
- It bind data from the database.

DataSet representation in .aspx.cs code,

```

01. protected void BindDataSet()
02. {
03.     SqlConnection con = new SqlConnection("your database connection string ");
04.     con.Open();
05.     SqlCommand cmd = new SqlCommand("Write your query or procedure ", con);
06.     SqlDataAdapter da = new SqlDataAdapter(cmd);
07.     DataSet ds = new DataSet();
08.     da.Fill(ds);
09.     grid.DataSource = ds;
10.     grid.DataBind();
11. }

```

See for more detail:

- [ADO.NET Technique With ASP.NET](#)

Question 41: What is Transactions and Concurrency in ADO.NET?

Answer: Transactions: ADO.NET provides a transaction class that represents a transaction. All data providers provide their own version of the transaction class. The IDbTransaction interface implements the basic functionality of the transaction class. All data provider-specific classes implement this namespace.

Figure shows some of the classes that implement IDbTransaction.



Methods of the Transaction Class

Method	Description
Commit	Commits the transaction to the database
Rollback	Rollbacks a transaction to the previous database state
Begin(IsolationLevel)	Begins a nested database transaction passing the isolation level

Concurrency in ADO.NET

The ADO.NET model assumes that the optimistic concurrency is the default concurrency because of its disconnected nature of data. A user reads data in a data through a data adapter, and data is available to user as a local copy of the data. The server database is available to all other users.

Another way of handling optimistic concurrency that you may be familiar with is by checking to see if a timestamp on the data source row has changed or the row version number has changed on the row being updated.

Pessimistic locking on the database isn't really supported by the data providers because the connection to the database is not kept open, so you must perform all locking with business logic on the DataSet.

- [Transactions and Concurrency in ADO.NET](#)

Question 42: 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.

See for more detail:

- [Introduction to ADO.Net](#)

Question 43: How can we Create and Manage Connections In ADO.NET?

Answer : Creating a Connection object

The connection component of a dataprovider establishes a connection with a database. To connect to Microsoft SQL Server, you use the SQL connection class. The following are the commonly used properties and methods of the SqlConnection class.

ConnectionString: It provides information, such as database name and user credentials for database access and so on.

Open(): Opens the connection for accessing the database.

Close(): Closes the connection to the database.

For Example:

```
01. // Creating object of SqlConnection Class.
02. SqlConnection cn = new SqlConnection();
03. //Creating connection string to sample database.
04. cn.ConnectionString = "Data source=.; Initial Catalog=Sample; User Id=sa; Password=facul";
```

The connection string provides the information that defines the connection to the database.

- **Data Source:** Specifies the provider name or your server name.
- **Initial Catalog:** Specifies the name of the database.
- **User Id and Password:** Provide the username and password of your database server.

```
01. Open the Connection
02.
03. // Creating object of SqlConnection Class.
04. SqlConnection cn = new SqlConnection();
05. //Creating connection string to sample database.
06. cn.ConnectionString = "Data source=.; Initial Catalog=Sample; User Id=sa; Password=facul";
07. cn.Open(); // it open the connection to database server..
```

Close the Connection

```
01. // Creating object of SqlConnection Class.
02. SqlConnection cn = new SqlConnection();
03. //Creating connection string to sample database.
04. cn.ConnectionString = "Data source=.; Initial Catalog=Sample; User Id=sa; Password=facul";
05. cn.Open(); // it open the connection to database server..
06. //Creating sqlcommand class object
07. SqlCommand cmd = new SqlCommand("Select * from tblEmployees", cn);
08. SqlDataReader dr = cmd.ExecuteReader();//Executing query
09. cn.Close();//Closing the connection
```

See for more detail:

- [Introduction to ADO.Net](#)

Question 44: 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](#)

Question 45: 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](#)

Question 46: Explain about Data Access Object or DAO?

Answer: Data Access Object (DAO) enabled programmers to access local databases in the Microsoft Jet Database Engine format, which were primarily Indexed Sequential Access Method (ISAM) files. After DAO came RDO and then ActiveX Data Objects (ADO). These data access technologies were designed for a client / server paradigm. However the tendency of distributed computing forced the development of a new technology to solve the problems of data manipulation on a n-tier architecture. ADO.NET is the evolution of ADO and its components have been designed to function properly on a n-tier architecture.

Summary of the evolution of the database objects to access data from Microsoft:

Name	Brief	Description
DAO	Data Access Objects	The first object-oriented interface that exposed the Microsoft Jet database engine that allowed developers using Visual Basic to directly connect to Access tables and other databases using ODBC. It is ideal for small databases in local deployments and single-system applications.
RDO	Remote Data Objects	An object-oriented data access interface to ODBC combined with the easy functionality of DAO allowing an interface to almost all of ODBC's low power and flexibility. RDO can't access Jet or ISAM databases in an efficient way. RDO provides the objects, properties, and methods needed to access the more complex aspects of stored procedures and complex resultsets.
ADO	Microsoft ActiveX Data Objects	ADO is the successor to DAO/RDO. ADO is the consolidation of almost all the functionality of DAO and RDO. ADO mostly includes RDO-style functionality to interact with OLE DB data sources, plus remoting and DHTML technology.
ADO MD	Microsoft ActiveX Data Objects Multidimensional	Provides easy access to multidimensional data from languages such as Microsoft Visual Basic and Microsoft Visual C++. ADO MD extends Microsoft ActiveX Data Objects (ADO) to include objects specific to multidimensional data, such as the CubeDef and Cellset objects. To work with ADO MD, the provider must be a multidimensional data provider (MDP) as defined by the OLE DB for OLAP specification. MDPs present data in multidimensional views as opposed to tabular data providers (TDPs) that present data in tabular views. With ADO MD you can browse multidimensional schema, query a cube, and retrieve the results.
ADOX	Microsoft ActiveX Data Objects	Extensions for Data Definition Language and Security Is an extension to the ADO objects and programming model. ADOX includes objects for schema creation and modification, as well as security. Because it is an object-based approach to schema manipulation, you can write code that will work against various data sources regardless of differences in their native syntaxes.ADOX is a companion library to the core ADO objects. It exposes additional objects for creating, modifying, and deleting schema objects, such as tables and procedures. It also includes security objects to maintain users and groups and to grant and revoke permissions on objects.
RDS	Remote Data Service	You can move data from a server to a client application or Web page, manipulate the data on the client, and return updates to the server in a single round trip.
		.NET ADO.NET is entirely based on XML. ADO.NET provides consistent access to data sources such as Microsoft SQL Server, as well as data sources exposed via OLE DB and XML. Data-sharing

ADO.NET	Microsoft ActiveX Data Objects	<p>consumer applications can use ADO.NET to connect to these data sources and retrieve, manipulate, and update data.</p> <p>ADO.NET cleanly factors data access from data manipulation into discrete components that can be used separately or in tandem. ADO.NET includes .NET data providers for connecting to a database, executing commands, and retrieving results. Those results are either processed directly, or placed in an ADO.NET DataSet object in order to be exposed to the user in an ad-hoc manner, combined with data from multiple sources, or remoted between tiers. The ADO.NET DataSet object can also be used independently of a .NET data provider to manage data local to the application or sourced from XML.</p> <p>The ADO.NET classes are found in System.Data.dll, and are integrated with the XML classes found in System.Xml.dll. When compiling code that uses the System.Data namespace, reference both System.Data.dll and System.Xml.dll. Compared with ADO there is no Recordset object.</p> <p>In ADO.NET there are four classes that read and write data from data sources:</p> <ol style="list-style-type: none"> 1. Connection .- Connect to data source 2. Command .- Execute stored procedures 3. DataAdapter .- Connects DataSet to database 4. DataReader .- Forward/only, read/only cursor
---------	--------------------------------------	--

See for more detail:

- [From DAO to ADO.NET](#)

Question 47: 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.

- [What is ADO.NET?](#)

Question 48: What is GetChanges() method in ADO.NET?

Answer: The GetChanges method of DataSet can be used to retrieve the rows that have been modified since the last time DataSet was filled, saved or updated. The GetChanges method returns a DataSet object with modified rows.

The GetChanges method can take either no argument or one argument of type DataRowState. The DataRowState enumeration defines the DataRow state, which can be used to filter a DataSet based on the types of rows.

DataRowState members

Member	Description
Added	Add added rows to a DataRowCollection of a DataSet and AcceptChanges has not been called.
Deleted	All the deleted rows.
Detached	Rows were created but not added to the row collection. Either waiting for the addition or have removed from the collection.
Modified	Modified rows and AcceptChanges has not been called.
Unchanged	Unchanged rows since last AcceptChanges was called.

See for more detail:

- [The GetChanges of DataSet and DiffGrams](#)

Question 49: How can you access the data from DataReader?

Answer: DataReader is a class that holds data as rows and columns to access data from the DataReader.

It provides the following methods:

1. GetName(int ColIndex)

The return type of this method is a string, it returns the name of the column for the given index position.

2. Read()

Moves the Record Pointer from the current location to the next row and returns a Boolean status that tells whether the row to which we have moved contains data in it or not, that will be true if present or false if not present.

3. GetValue(int Colindex)

Returns a column's value from the row to which the pointer was pointing by specifying the column index position.

4. NextResult ()

Moves the record pointer from the current table to the next table if a table exists and returns true else returns false.

Features of Data Reader

1. Provides faster access to data from a Data Source, since it is connection oriented.
2. It can hold multiple tables at a time. To load multiple tables into a DataReader pass multiple select statements as the argument to the command separated by a colon (;).

For example:

```
01. Command cmd=new Command("Select * From Student ; Select * From Mark ", Con);
02. Data Reader dr= cmd.ExecuteReader();
```

See for more detail:

- [Introduction To ADO.NET: Part 2](#)

Question 50: What is BindingSource class in ADO.NET?

Answer: The BindingSource class is used to simplify data binding as well as various operations on records. It has different methods like AddNew(), MoveFirst(), MovePrevious(), MoveNext(), etc which provide easier way for adding new row, moving to first record, moving to previous record, moving to next record and many other operations without writing code for them.

See for more detail:

- [Using the BindingSource Class in ADO.NET](#)

More Interview Questions

- [Azure Interview Questions](#)
- [ASP.NET MVC Interview Questions](#)
- [C# interview questions](#)
- [ASP.NET interview questions](#)
- [Bootstrap interview questions](#)
- [Html 5 interview questions](#)
- [WCF interview questions and answers](#)
- [WPF interview questions and answers](#)
- [CSS interview questions and answers](#)
- [Angularjs interview questions](#)
- [SQL Server interview questions](#)
- [Interview question on .NET framework or clr](#)
- [Javascript interview questions](#)
- [Interview questions for 2 year experience in SQL and C#](#)
- [Important .NET interview questions and answers](#)
- [Software Testing Interview Questions/](#)
- [Dot.NET interview questions for experienced and fresher](#)
- [SQL interview questions](#)
- [C# interview questions and answers](#)
- [jQuery interview question and answer with practices part 2](#)
- [Python interview questions](#)

[ADO.NET Interview Questions](#)

[Most Asked ADO.NET Interview Questions](#)



[About Us](#) [Contact Us](#) [Privacy Policy](#) [Terms](#) [Media Kit](#) [Sitemap](#) [Report a Bug](#) [FAQ](#) [Partners](#)

[C# Tutorials](#) [Common Interview Questions](#) [Stories](#) [Consultants](#) [Ideas](#) [Certifications](#)

©2020 C# Corner. All contents are copyright of their authors.