ADO .NET:

The full form of ADO is ActiveX Data Object. 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.

ADO.NET is an object-oriented set of libraries that allows you to interact with data sources.

ADO.NET Data Providers

Provider Name	API prefix	Data Source Description
ODBC Data Provider	Odbc	Data Sources with an ODBC interface. Normally older data bases.
OleDb Data Provider	OleDb	Data Sources that expose an OleDb interface, i.e. Access or Excel.
Oracle Data Provider	Oracle	For Oracle Databases.
SQL Data Provider	Sql	For interacting with Microsoft SQL Server.
Borland Data Provider	Bdp	Generic access to many databases such as Interbase, SQL Server, IBM DB2, and Oracle.

4 ADO.NET OBJECTS:

• **Connection Object:**The connection helps identify the database server, the database name, user name, password, and other parameters that are required for connecting to the data base. A connection object is used by command objects so they will know which database to execute the command on.

• **Command Object:** Use a command object to send SQL statements to the database. A command object uses a connection object to figure out which database to communicate with.

• **DataReaderObject** : The data reader object allows you to obtain the results of a SELECT statement from a command object. For performance reasons, the data returned from a data reader is a fast forward-only stream of data.

- **DataSet Object**: DataSet objects are in-memory representations of data. They contain multiple Datatable objects, which contain columns and rows, just like normal database tables. You can even define relations between tables to create parent-child relationships. The DataSet is an object that is used by all of the Data Providers, which is why it does not have a Data Provider specific prefix.•
- **DataAdapterObject** : The data adapter fills a DataSet object when reading the data and writes in a single batch when persisting changes back to the database. A data adapter contains a reference to the connection object and opens and closes the connection automatically when reading from or writing to the database.

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

Connected Architecture of ADO.NET

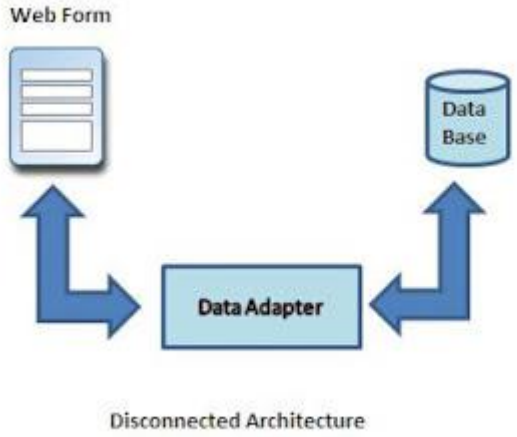
The architecture of ADO.net, in which connection must be opened to access the data retrieved from database is called as connected architecture. Connected architecture was built on the classes connection, command, datareader and transaction.

Connected architecture is when you constantly make trips to the database for any CRUD (Create, Read, Update and Delete) operation you wish to do. This creates more traffic to the database but is normally much faster as you should be doing smaller transactions.

Connected Architecture

Disconnected Architecture in ADO.NET

The architecture of ADO.net in which data retrieved from database can be accessed even when connection to database was closed is called as disconnected architecture. Disconnected architecture of ADO.net was built on classes connection, dataadapter, commandbuilder and dataset and dataview.
Disconnected architecture is a method of retrieving a record set from the database and storing it giving you the ability to do many CRUD (Create, Read, Update and Delete) operations on the data in memory, then it can be re-synchronized with the database when reconnecting. A method of using disconnected architecture is using a Dataset.



Disconnected Architecture

DataReader is Connected Architecture since it keeps the connection open until all rows are fetched one by one
DataSet is DisConnected Architecture since all the records are brought at once and there is no need to keep the connection alive
Difference between Connected and disconnected architecture

Connected	Disconnected
It is connection oriented.	It is dis_connection oriented.
Datareader	DataSet
Connected methods gives faster performance	Disconnected get low in speed and performance.
connected can hold the data of single table	disconnected can hold multiple tables of data
connected you need to use a read only forward only data reader	disconnected you cannot
Data Reader can't persist the data	Data Set can persist the data
It is Read only, we can't update the data.	We can update data

DataGridView

The DataGridView control is designed to be a complete solution for displaying tabular data with Windows Forms. The DataGridView control is highly configurable and extensible, and it provides many properties, methods, and events to customize its appearance and behaviour.

The DataGridView control makes it easy to define the basic appearance of cells and the display formatting of cell values. The cell is the fundamental unit of interaction for the DataGridView. All cells derive from the DataGridViewCell base class. Each cell within the DataGridView control can have its own style, such as text format, background color, foreground color, and font. Typically, however, multiple cells will share particular style characteristics. The data type for the cell's Value property by default is of type Object.

The GridView control displays the values of a data source in a table. Each column represents a field, while each row represents a record. The GridView control supports the following features:

- Binding to data source controls, such as SqlDataSource.
- Built-in sort capabilities.
- Built-in update and delete capabilities.
- Built-in paging capabilities.
- Built-in row selection capabilities.
- Programmatic access to the GridView object model to dynamically set properties, handle events, and so on.
- Multiple key fields.
- Multiple data fields for the hyperlink columns.
- Customizable appearance through themes and styles.

SQL Parameter

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.

Using parameterized queries is a three-step process:

1. Construct the SqlCommand command string with parameters.
2. Declare a SqlParameter object, assigning values as appropriate.
3. Assign the SqlParameter object to the SqlCommand object's Parameters property.

```
// 1. declare command object with parameter
    SqlCommandcmd = newSqlCommand("select * from Customers
where city = @City", conn);
// 2. define parameters used in command object
    SqlParameterparam  = new SqlParameter();
    param.ParameterName = "@City";
    param.Value         = inputCity;
// 3. add new parameter to command object
    cmd.Parameters.Add(param);
```

Data Adapter Object

- 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

Provider-Specific Data Adapter classes

Data Source

- A Data Adapter object accesses data in a disconnected mode. Its object contains a reference to a connection object.
- It is designed in a way that it implicitly opens and closes the connection whenever required.
- It maintains the data in a DataSet object. The user can read the data if required from the dataset and write back the changes in a single batch to the database. Additionally, the Data Adapter contains a command object reference for SELECT, INSERT, UPDATE, and DELETE operations on the data objects and a data source.
- 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 it before populating the data. If subsequent calls to the method for refreshing the data are required then the primary key information should be present.
- **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 appriopriate INSERT, UPDATE, and DELETE statements.
  A Data Adapter object is formed between a disconnected ADO.NET object and a data source.

- **Difference Between DataSet and Data Reader**

No	Data Reader	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