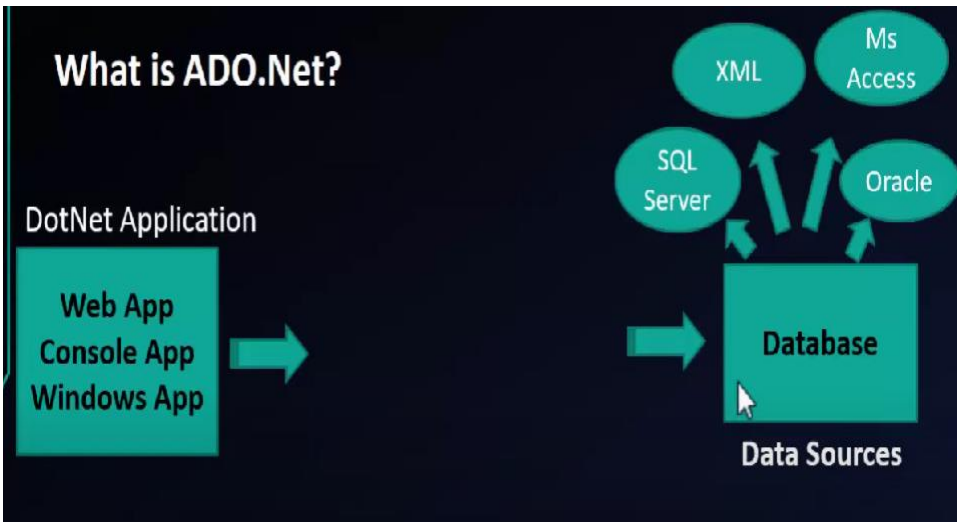
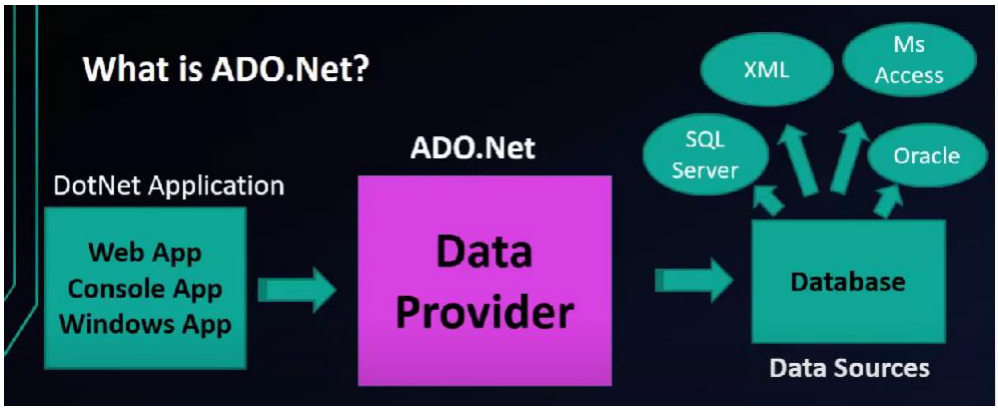
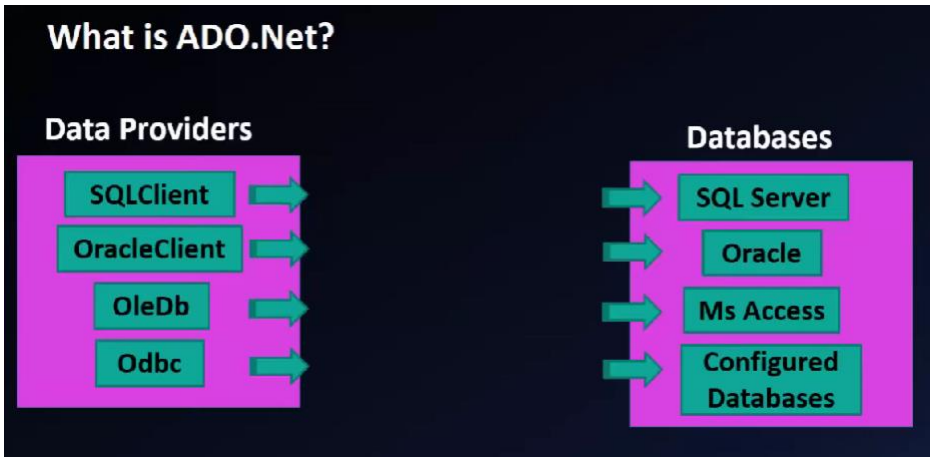
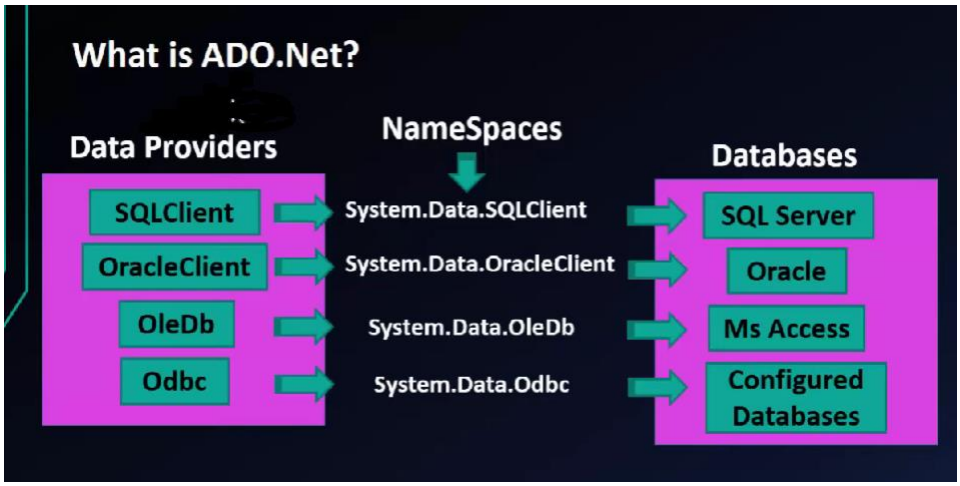
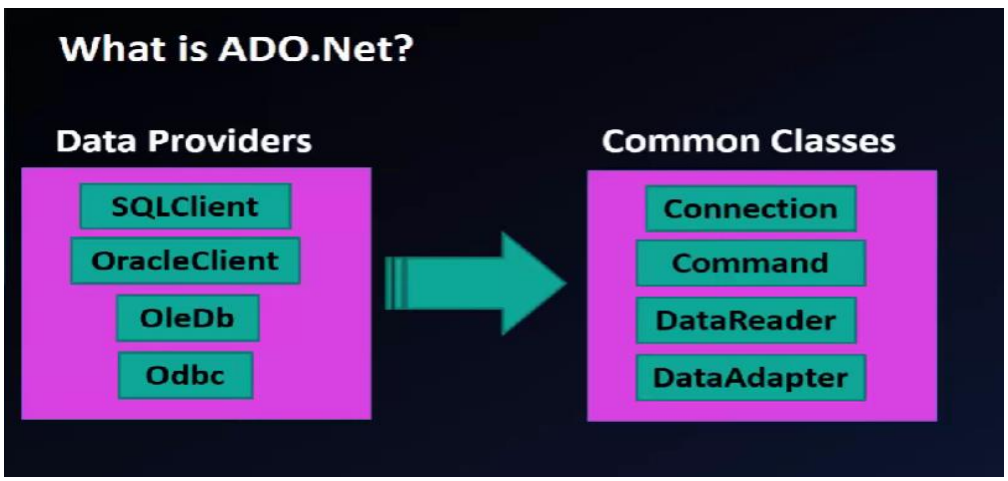
ADO.NET

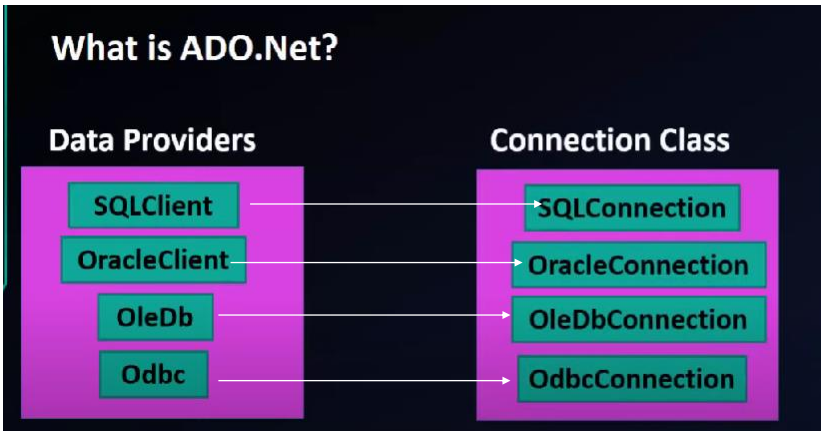

What is ADO.Net?
XML
Ms Access
SQL Server
Oracle
DotNet Application
Web App
Console App
Windows App
Database
Data Sources


What is ADO.Net?
DotNet Application
Web App
Console App
Windows App
ADO.Net
Data Provider
XML
Ms Access
SQL Server
Oracle
Database
Data Sources


What is ADO.Net?
Data Providers
SQLClient
OracleClient
OleDb
Odbc
Databases
SQL Server
Oracle
Ms Access
Configured Databases


What is ADO.Net?
Data Providers
NameSpaces
Databases
SQLClient
OracleClient
OleDb
Odbc
System.Data.SQLClient
System.Data.OracleClient
System.Data.OleDb
System.Data.Odbc
SQL Server
Oracle
Ms Access
Configured Databases


What is ADO.Net?
Data Providers
Common Classes
SQLClient
OracleClient
OleDb
Odbc
Connection
Command
DataReader
DataAdapter



What is ADO.Net?

- ADO.NET provides a bridge between the front end controls and the back end database.
- ADO.NET is a module of .Net Framework which is used to establish connection between application and data sources.
- Data sources can be such as **SQL Server** and **XML**.
- ADO.NET consists of classes that can be used to **connect**, **retrieve**, **insert** and **delete** data.
- All the ADO.NET classes are located into **System.Data.dll** and integrated with XML classes located into **System.Xml.dll.**

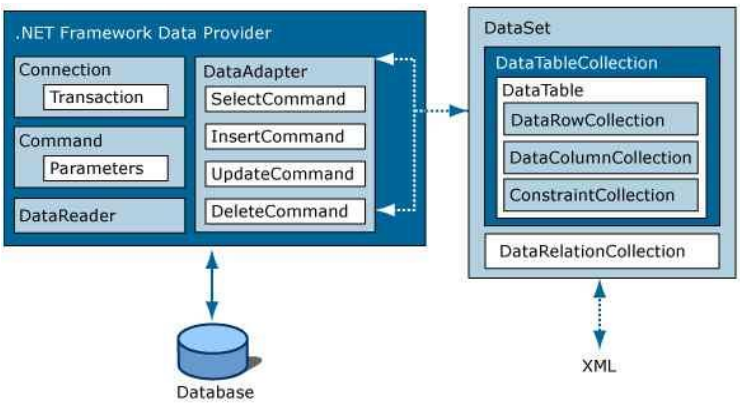
What is ADO.Net?

- ADO.NET stands for **ActiveX Data Object.**
- It is a database access technology created by **Microsoft** as part of its **.NET framework** that can access any kind of data source.
- It's a set of **object-oriented classes** that provides a rich set of data components to create high-performance, reliable and scalable database applications.

As you know when we create software or website or any other application which is connected to the database we need to make connection to the database to get data. Here you can get data in two different ways.

Either it can be connected architecture where you go and connect to the database and get data or disconnected architecture where you connect to the database first time and get all data in an object and use it if required. You can perform any action like insert, update, and search on this.

You can use two C# objects to achieve this, first one is **DataSet** and other one is **DataReader**.



- **DataSet**

  It is a type of disconnected architecture. Disconnected architecture means, you don't need to connect always when you want to get data from the database.

- You can get data from dataset; basically DataSet is a collection of datatables. We can store the database table, view data in the DataSet and can also store the xml value in dataset and get it if required.

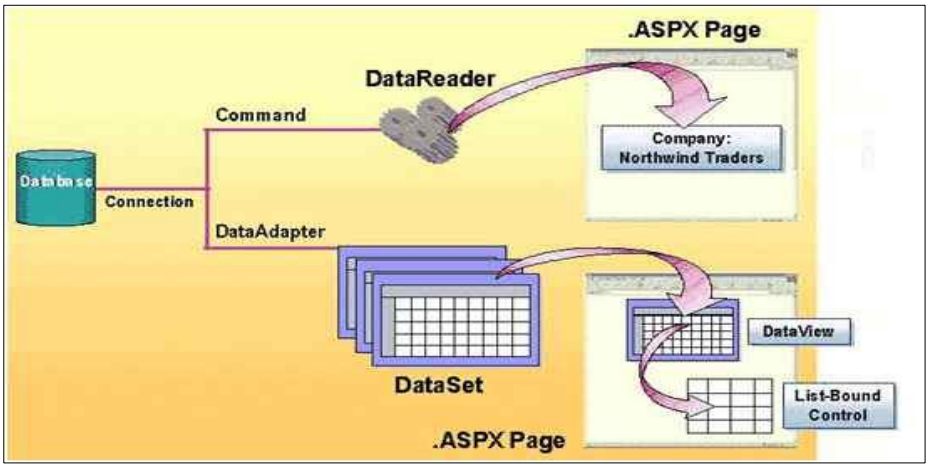
- To achieve this you need to use **DataAdapter** which work as a mediator between Database and **DataSet**.

DataReader

- It is a connected architecture, which means when you require data from the database you need to connect with database and fetch the data from there.

- You can use if you need updated data from the database in a faster manner.

- DataReader is Read/Forward only that means we can only get the data using this but we cannot update or insert the data into the database. It fetches the record one by one.



- List controls include the ListBox, DropDownList, CheckBoxList, RadioButtonList, and BulletedList. Each of these controls can be data bound to a data source. They use one field in the data source as the display text and can optionally use a second field as the value of an item.

- A DataView provides various views of the data stored in a DataTable. 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. That is we can customize the views of data from a DataTable.

Types of Architecture
Ado.net is both connection-oriented as well as disconnection oriented. Depending upon the functionality of an application, we can make it connection-oriented or disconnection oriented. We can even use both the modes together in a single application.

1.Connected Architecture

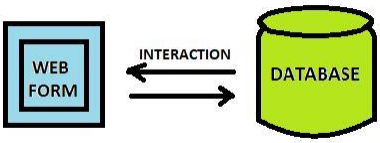
•As the name suggests, connected architecture refers to the fact that the connection is established for the full time between the database and application. For e.g. we make a program in C# that is connected with the database for the full time, so that will be connected architecture.

•Connected architecture is forward only and read-only. This means the connected mode will work only in one particular direction i.e. forward and that too for read-only purpose. Application issues query then read back results and process them.

•For connected architecture, we mainly use the object of the DataReader class.

•DataReader is used to retrieve the data from the database and it also ensures that the connection is maintained for the complete interval of time.

•In connected architecture, the application is directly



DataReader in Connected architecture

- DataReader class is used to read the data from the database. It works in forward only and reads the only mode and requires the connection for the complete time. That is why we use it in connected architecture.
- The forward only feature makes it an efficient way to read data. Thus we can say, DataReader is connection-oriented and requires an active connection while reading the data.

Disconnected Architecture

- Disconnected architecture refers to the mode of architecture in Ado.net where the connectivity between the database and application is not maintained for the full time.
- Connectivity within this mode is established only to read the data from the database and finally to update the data within the database.
- This means during the processing of the application, we need data so that data is fetched from the database and kept in temporary tables.
- After that whenever data is required, it is fetched from the temporary tables. And finally, when the operations were completed, the connection was established to update the data within the database from the temporary tables.
- In this mode, application issues query then retrieves and store results for processing. For this purpose, we use objects of SqlDataAdapter and DataSet classes.
- In disconnected architecture, a Dataset is used for retrieving data from the database.
- This way there is no need to establish a connection for the full time because DataSet acts as temporary storage. All the operations can be performed on the data using the Dataset and finally modified at the database.

DataAdapter in Disconnected architecture

DataAdapter class acts as an interface between application and database. It provides the data to the Dataset which helps the user to perform the operations and finally the modifications are done in the Dataset which is passed to the DataAdapter which updates the database. DataAdapter takes the decision for the establishment and termination of the connection.

DataAdapter is required for connectivity with the database. DataAdapter established a connection with the database and fetches the data from the database and fill it into the Dataset. And finally, when the task is completed it takes the data from the DataSet and updates it into the database by again establishing the connection.
It can be said that DataAdapter acts as a mediator between the application and database which allows the interaction in disconnected architecture.

Features of ADO.NET :
The following are the features of ADO.NET –

- **Interoperability-**

We know that XML documents are text-based formats. So, one can edit and edit XML documents using standard text-editing tools. ADO.NET uses XML in all data exchanges and for internal representation of data.

- **Maintainability –**
  ADO.NET is built around the idea of separation of data logic and user interface. It means that we can create our application in independent layers.

- **Performance –**
  It uses disconnected data architecture which is easy to scale as it reduces the load on the database. Everything is handled on the client-side, so it improves performance.

- **Scalability –**
  It means meeting the needs of the growing number of clients, which degrading performance. As it uses disconnected data access, applications do not retain database lock connections for a longer time. Thus, it accommodates scalability by encouraging programmers to conserve limited resources and allow users to access data simultaneously.

Differences between DataSet and DataReader

DataSet	DataReader
DataSet object can contain multiple rowsets from the same data source as well as from the relationships between them.	DataReader provides forward-only and read-only access to data.
Dataset is a disconnected architecture.	Datareader is connected architecture.
Dataset can persist data.	Datareader can not persist data.
A DataSet is well suited for data that needs to be retrieved from multiple tables.	It has live connection while reading data
DatsSet is slower than DataReader ,Due to overhead.	Speed performance is better in DataReader

Data Reader	Dataset	Data Adapter
datareader(connection oriented) is a read only and forward only record set which will have the data retrieved based on the select statement. We can't do DML operations through DataReader.	dataset(disconnected) model which we can do all the DML operations.	Data Adapter is Bridge between Database and Dataset. Works in Disconnected Mode
Using Data Reader only one value can be accessed from the database	using Dataset you can access a any number of values from the database.	**Data adapter** acts as a bridge between data source and dataset and it is a multipurpose object i.e., we can insert, update, delete throught data adapter
Datareader is connected to the data source	dataset is connected to the XML.	It fetches all the required data from database and pass it to Dataset.
DataReader is conncetion oriented u can find the changes in the database reflected in data reader immediately	dataset is pure disconnected one you will find the changes only when u refresh the dataset using dataadapter	
The **Data Reader** provides a fast, forward-only, read-only stream of data from a database.	The **Dataset** provides an in-memory relational representation of data, a complete set of data that includes the tables that contain, order, and constrain the data, as well as the relationships between the tables	
Data Reader can't persist the data	Data Set can persist the data	

https://www.c-sharpcorner.com/UploadFile/8a67c0/connected-vs-disconnected-architecture-in-C-Sharp/

https://tutorialslink.com/Articles/Connected-and-Disconnected-architecture-in-Adonet/1679

http://asp.net-informations.com/ado.net/ado-architecture.htm