SHRI MAHAVIR VIDYAMANDIR TRUST
सा विद्या या विमुक्तये

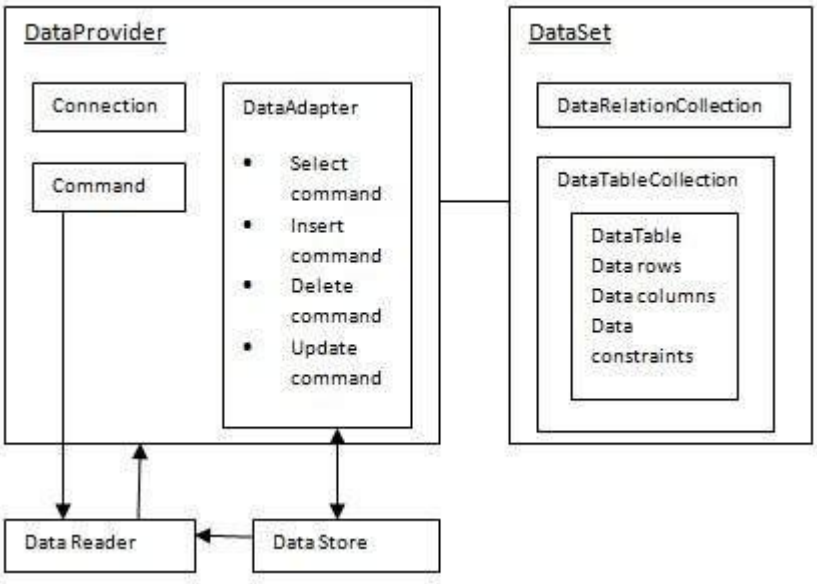
# ASP.NET UNIT - 3

## Unit No 3. Database Access

### 1. Write note on ADO.NET architecture.

Microsoft ActiveX Data Objects.Net (ADO.Net) is a model, a part of the .Net framework that is used by the .Net applications for retrieving, accessing and updating data.

ADO.Net object model is nothing but the structured process flow through various components. The object model can be pictorially described as




**Data Provider**

A data provider is used for connecting to a database, executing commands and retrieving data, storing it in a dataset, reading the retrieved data and updating the database.

The data provider in ADO.Net consists of the following four objects −

| Sr.No. | Objects & Description |
|---|---|
| 1 | **Connection**<br>This component is used to set up a connection with a data source. |
| 2 | **Command**<br>A command is a SQL statement or a stored procedure used to retrieve, insert, delete or modify data in a data source. |

| | |
|---|---|
| 3 | **DataReader**<br>Data reader is used to retrieve data from a data source in a read-only and forward-only mode. |
| 4 | **DataAdapter**<br>This is integral to the working of ADO.Net since data is transferred to and from a database through a data adapter. It retrieves data from a database into a dataset and updates the database. When changes are made to the dataset, the changes in the database are actually done by the data adapter. |

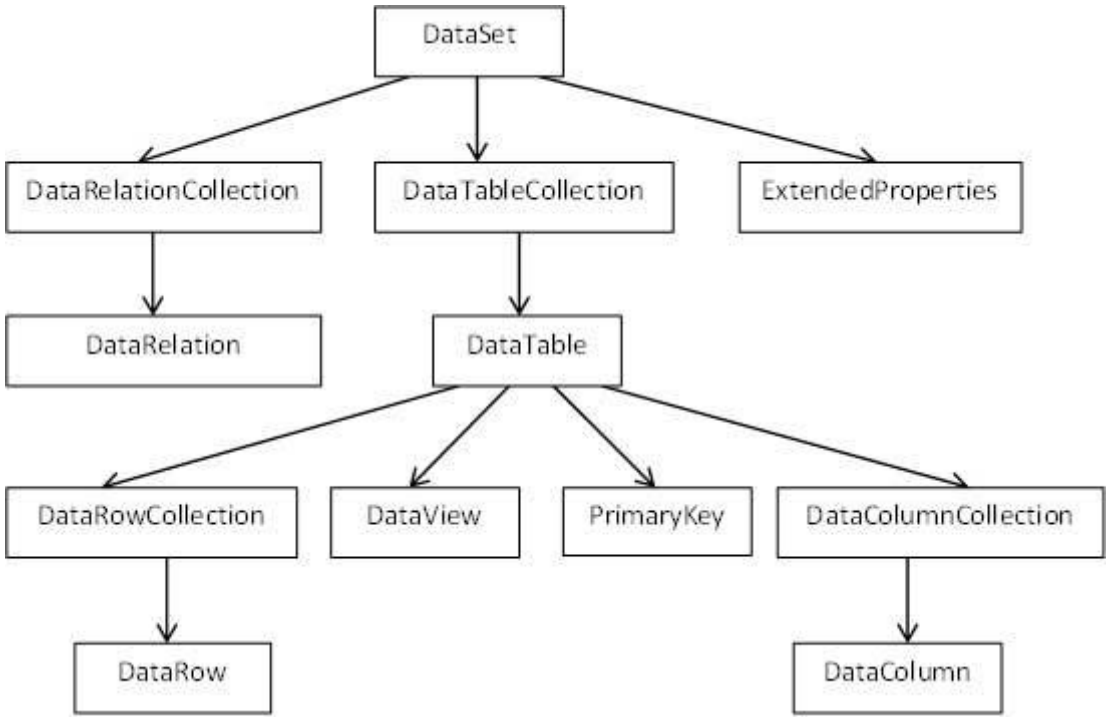
There are following different types of data providers included in ADO.Net

- The .Net Framework data provider for SQL Server - provides access to Microsoft SQL Server.
- The .Net Framework data provider for OLE DB - provides access to data sources exposed by using OLE DB.
- The .Net Framework data provider for ODBC - provides access to data sources exposed by ODBC.
- The .Net Framework data provider for Oracle - provides access to Oracle data source.
- The EntityClient provider - enables accessing data through Entity Data Model (EDM) applications.

## DataSet

**DataSet** is an in-memory representation of data. It is a disconnected, cached set of records that are retrieved from a database. When a connection is established with the database, the data adapter creates a dataset and stores data in it. After the data is retrieved and stored in a dataset, the connection with the database is closed. This is called the 'disconnected architecture'. The dataset works as a virtual database containing tables, rows, and columns.

The following diagram shows the dataset object model −



The DataSet class is present in the **System.Data** namespace. The following table describes all the components of DataSet −

| Sr.No. | Components & Description |
|---|---|
| 1 | **DataTableCollection :** It contains all the tables retrieved from the data source. |
| 2 | **DataRelationCollection :** It contains relationships and the links between tables in a data set. |
| 3 | **ExtendedProperties :** It contains additional information, like the SQL statement for retrieving data, time of retrieval, etc. |
| 4 | **DataTable :** It represents a table in the DataTableCollection of a dataset. It consists of the DataRow and DataColumn objects. The DataTable objects are case-sensitive. |
| 5 | **DataRelation :** It represents a relationship in the DataRelationshipCollection of the dataset. It is used to relate two DataTable objects to each other through the DataColumn objects. |
| 6 | **DataRowCollection :** It contains all the rows in a DataTable. |
| 7 | **DataView :** It represents a fixed customized view of a DataTable for sorting, filtering, searching, editing and navigation. |
| 8 | **PrimaryKey :** It represents the column that uniquely identifies a row in a DataTable. |
| 9 | **DataRow :** It represents a row in the DataTable. The DataRow object and its properties and methods are used to retrieve, evaluate, insert, delete, and update values in the DataTable. The NewRow method is used to create a new row and the Add method adds a row to the table. |
| 10 | **DataColumnCollection :** It represents all the columns in a DataTable. |
| 11 | **DataColumn :** It consists of the number of columns that comprise a DataTable. |

## Connecting to a Database

The .Net Framework provides two types of Connection classes −

- **SqlConnection** − designed for connecting to Microsoft SQL Server.
- **OleDbConnection** − designed for connecting to a wide range of databases, like Microsoft Access and Oracle.

## 2. Difference between Data Reader and Data Adapter. Explain Data Reader in details.

| Sr | Data Adapter | Data Reader |
|---|---|---|
| 1 | DataAdapter follows connectionless-oriented architecture which simply means you need not necessarily be connected to a data-source | DataReader is a connection-oriented architecture which means it needs an active connection to a data-source for it to operate. |
| 2 | DataAdapter is an intermediate layer/ middleware which acts a bridge between the DataSet and a Database | DataReader provides forward-only, read-only access to data using a server-side cursor (*simply put it is ued to read the data*). |
| 3 | Using DataSet we can manipulate and update a DataSet's contents while disconnected from the Datasource and send any modified data back for processing using a related DataAdapter | DataReader can only read data from a Database & cannot modify it. |
| 4 | DataAdapter object is used to read the data from database and populates that data to DataSet | DataReader simply reads the data using the Read() method. |
| 5 | DataAdapter is comparatively slower then Data Reaser | DataReader can increase application performance both by retrieving data as soon as it is available, and (by default) storing only one row at a time in memory, reducing system overhead. |

### Data Reader

The **DataReader** object in ADO.NET allows you to retrieve data from database in read-only and forward-only mode. It means you can only read and display data but can't update or delete data. If you want to make modification in retrieved data you need to use DataAdapter instead of DataReader.

A data reader provides an easy way for the programmer to read data from a database as if it were coming from a stream. The DataReader is the solution for forward streaming data through ADO.NET. The data reader is also called a firehose cursor or forward read-only cursor because it moves forward through the data. The data reader not only allows you to move forward through

each record of database, but it also enables you to parse the data from each column. The DataReader class represents a data reader in ADO.NET.

When you want to only display information or search result, you can use DataReader. **There are various advantages of using DataReader** like:

**1.** The retrieved data is stored in the network buffer in the client and then the client can read data using Read method. As data gets stored in the client network buffer it increases application performance significantly.

**2.** By default **DataReader** stores only one row at a time in memory. It reduces system overhead.

| 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 |

| METHOD | 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. |

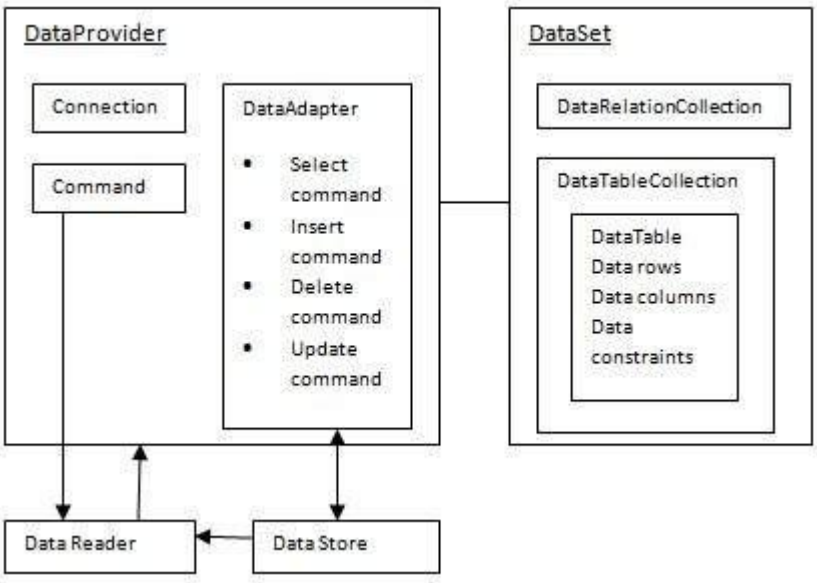
**Example**

```
Dim dr As OleDbDataReader
Dim cmd As New OleDbCommand
cmd.CommandText = "Select * from EmpMst”
cmd.Connection = cn
dr = cmd.ExecuteReader
While dr.Read()
    txtfname.Text = dr.Item("Fname")
    txtlname.Text = dr.Item("Lname")
    txtcity.Text = dr.Item("city")
    txtstate.Text = dr.Item("State")
End While
```

```
dr.Close()
```

## 3. Write note on connected mode architecture of ADO.NET.

ADO.Net object model is nothing but the structured process flow through various components. The object model can be pictorially described as –




The data residing in a data store or database is retrieved through the **data provider**. Various components of the data provider retrieve data for the application and update data.

An application accesses data either through a dataset or a data reader.

- **Datasets** store data in a **disconnected** cache and the application retrieves data from it.
- **Datareaders** provide data to the application in a read-only and forward-only mode using **connected data model**
- DataReades are used to **efficiently retrieve a forward-only stream of data from a database**.
- DataReaders are appropriate when the need is to **simply display the result set**, as only one record at a time is ever present in memory.
- The **DataReader** provides a high-performance stream of data from the data source.

  **ExecuteReader** : ExecuteReader used for getting the query results as a DataReader object. It is readonly forward only retrieval of records and it uses select command to read through the table from the first to the last.

```
Dim reader As SqlDataReader
reader = Command.ExecuteReader()
While reader.Read()
    MsgBox(reader.Item(0))
End While
reader.Close()
```

**ExecuteScaler** : ExecuteScalar() in SqlCommand Object is used for get a single value from Database after its execution. It executes SQL statements or Stored Procedure and returned a scalar value on first column of first row in the Result Set. If the Result Set contains more than one columns or rows , it takes only the first column of first row, all other values will ignore. If the Result Set is empty it will return a Null reference.
It is very useful to use with aggregate functions like Count(*) or Sum() etc. When compare to ExecuteReader() , ExecuteScalar() uses fewer System resources.

```
Dim retValue As Integer
Command = New SqlCommand(Sql, Connection)
retValue = Command.ExecuteScaler()
```

**ExecuteNonQuery :** ExecuteNonQuery used for executing queries that does not return any data. It is used to execute the sql statements like update, insert, delete etc. ExecuteNonQuery executes the command and returns the number of rows affected.

```
Dim retValue As Integer
Command = New SqlCommand(Sql, Connection)
retValue = Command.ExecuteNonQuery()
```

### Data Provider

A data provider is used for connecting to a database, executing commands and retrieving data, storing it in a dataset, reading the retrieved data and updating the database.

The data provider in ADO.Net consists of the following four objects –

| Sr.No. | Objects & Description |
|---|---|
| 1 | **Connection :** This component is used to set up a connection with a data source. |
| 2 | **Command :** A command is a SQL statement or a stored procedure used to retrieve, insert, delete or modify data in a data source. |
| 3 | **DataReader :** Data reader is used to retrieve data from a data source in a read-only and forward-only mode. |

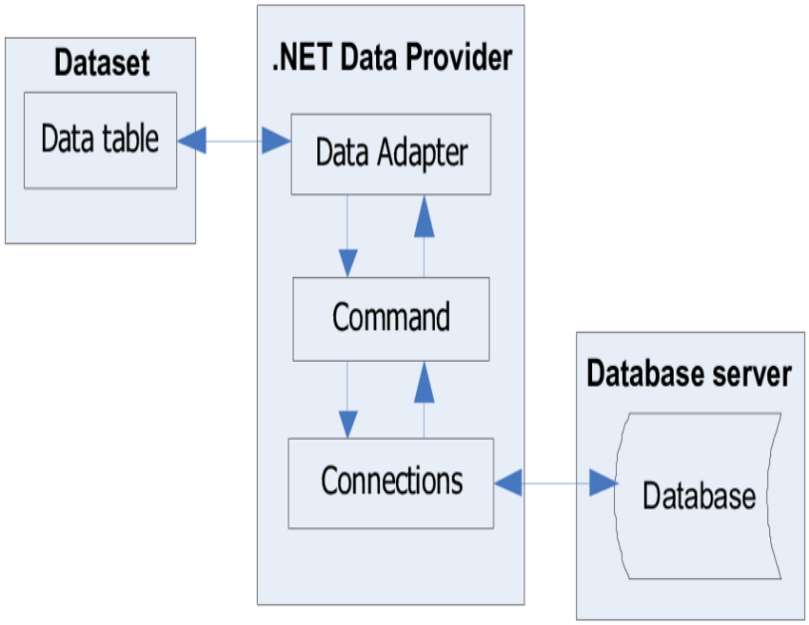
| | |
|---|---|
| 4 | **DataAdapter :** This is integral to the working of ADO.Net since data is transferred to and from a database through a data adapter. It retrieves data from a database into a dataset and updates the database. When changes are made to the dataset, the changes in the database are actually done by the data adapter. |

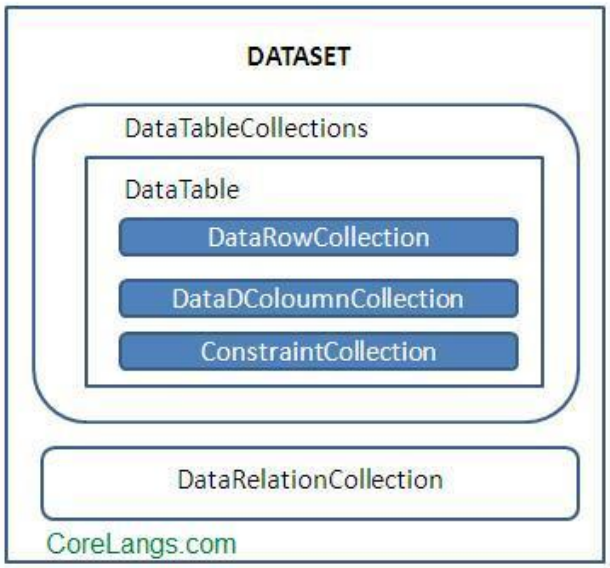
**There are following different types of data providers included in ADO.Net**

- The .Net Framework data provider for SQL Server - provides access to Microsoft SQL Server.
- The .Net Framework data provider for OLE DB - provides access to data sources exposed by using OLE DB.

## 4. Write note on Disconnected mode architecture of ADO.NET.

**DataSet** is an in-memory representation of data. It is a **disconnected**, cached set of records that are retrieved from a database. When a connection is established with the database, the data adapter creates a dataset and stores data in it. After the data is retrieved and stored in a dataset, the connection with the database is closed. This is called the 'disconnected architecture'. The dataset works as a virtual database containing tables, rows, and columns.






**DataSet** provides a disconnected representation of result sets from the Data Source, and it is completely independent from the Data Source. DataSet provides much greater flexibility when dealing with related Result Sets.

**DataSet** contains rows, columns,primary keys, constraints, and relations with other DataTable objects. It consists of a collection of DataTable objects that you can relate to each other with DataRelation objects. The DataAdapter Object provides a bridge between the DataSet and the Data Source.

The DataSet class is present in the **System.Data** namespace. The following table describes all the components of DataSet −

| Sr.No. | Components & Description |
|---|---|
| 1 | **DataTableCollection :** It contains all the tables retrieved from the data source. |
| 2 | **DataRelationCollection :** It contains relationships and the links between tables in a data set. |
| 3 | **ExtendedProperties :** It contains additional information, like the SQL statement for retrieving data, time of retrieval, etc. |
| 4 | **DataTable :** It represents a table in the DataTableCollection of a dataset. It consists of the DataRow and DataColumn objects. The DataTable objects are case-sensitive. |
| 5 | **DataRelation :** It represents a relationship in the DataRelationshipCollection of the dataset. It is used to relate two DataTable objects to each other through the DataColumn objects. |
| 6 | **DataRowCollection :** It contains all the rows in a DataTable. |
| 7 | **DataView :** It represents a fixed customized view of a DataTable for sorting, filtering, searching, editing and navigation. |
| 8 | **PrimaryKey :** It represents the column that uniquely identifies a row in a DataTable. |

| | |
|---|---|
| 9 | **DataRow :** It represents a row in the DataTable. The DataRow object and its properties and methods are used to retrieve, evaluate, insert, delete, and update values in the DataTable. The NewRow method is used to create a new row and the Add method adds a row to the table. |
| 10 | **DataColumnCollection :** It represents all the columns in a DataTable. |
| 11 | **DataColumn :** It consists of the number of columns that comprise a DataTable. |