<p style="text-align: center;"><u>Unit IV</u>: Data access in VB.NET</p>

Introduction

Data access in VB.NET involves retrieving, manipulating, and managing data from various data sources such as databases, XML files, or web services. This process is essential for developing data-driven applications where information needs to be stored, retrieved, and presented to users.

Architecture of ADO.Net:

   - ADO.NET: Stands for ActiveX Data Objects for .NET. It is a part of the .NET Framework designed to provide data access services. ADO.NET includes classes that allow developers to interact with data sources such as databases and XML files.

   - Architecture: ADO.NET follows a disconnected architecture where data is retrieved from the data source, manipulated in-memory, and then changes are propagated back to the data source as necessary.
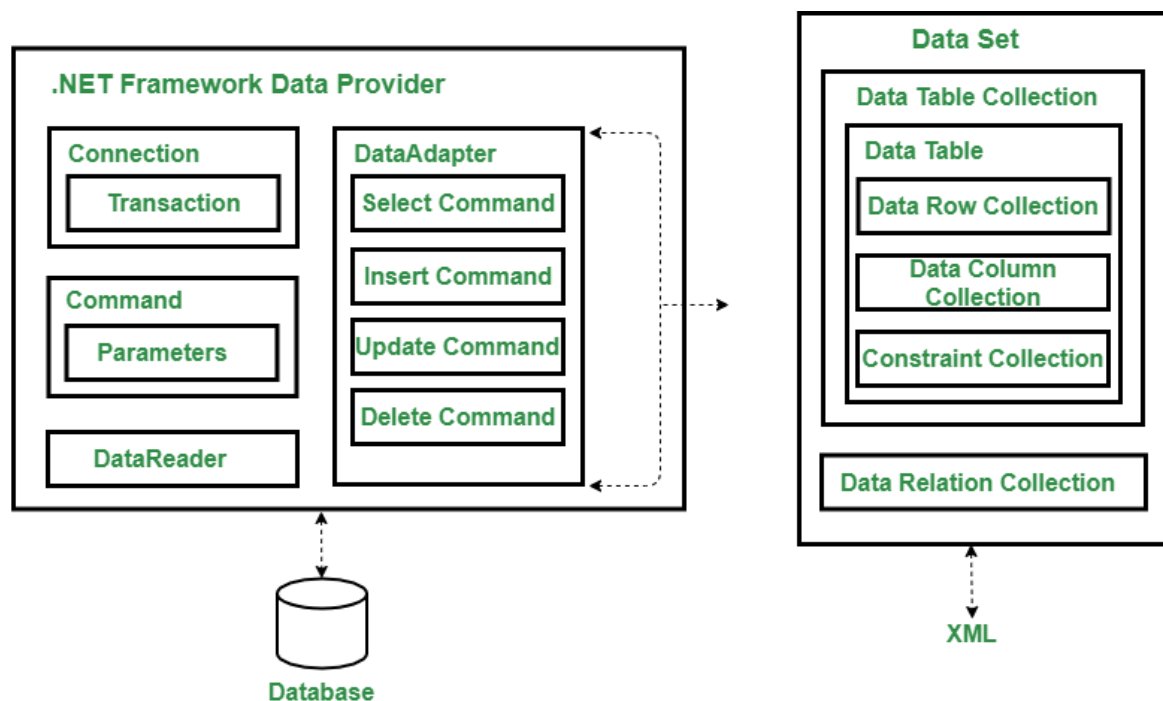
Accessing Data with Server Explorer:

   - Server Explorer in Visual Studio provides a convenient interface to interact with data sources such as databases. You can add connections to databases, view database objects, execute queries, and perform other data-related tasks directly from within Visual Studio.

Data Providers:

- ADO.NET includes various data providers for connecting to different types of data sources. The two primary data providers are:

  - SQL Server Data Provider: Used for connecting to Microsoft SQL Server databases.

  - OLE DB Data Provider: Offers connectivity to a wide range of data sources using OLE DB.

Connections, Data Reader, Data Adapters, and Datasets:

- Connections: ADO.NET provides the `SqlConnection` class for establishing connections to a database. Connections are opened using the `Open()` method and closed using the `Close()` method.

- Data Reader: The `SqlDataReader` class allows forward-only, read-only access to data retrieved from a database. It is used for retrieving large datasets efficiently when data needs to be read sequentially.

- Data Adapters: Data adapters, such as `SqlDataAdapter`, serve as a bridge between a dataset and a data source. They fill datasets with data retrieved from the database and update the database with changes made to the dataset.

- Datasets: A dataset is an in-memory representation of data retrieved from a database. It consists of tables, rows, and columns. Datasets can be manipulated independently of the database and can store multiple result sets from different queries.

Creating a New Data Connection:

- To create a new data connection in Visual Studio, you can use the Server Explorer or the Data Source Configuration Wizard. You need to specify the type of data source (e.g., SQL Server, OLE DB), connection details (server name, authentication method), and optionally, the database name.

Creating a Dataset:

- To create a dataset programmatically, you can define a new instance of the `DataSet` class. You can then add tables, define table schemas, and populate the dataset with data retrieved from a data source using data adapters.

Data Grid Control and Displaying Data in Data Grid:

- The `DataGridView` control in VB.NET provides a convenient way to display and manipulate tabular data. You can bind the DataGridView to a dataset or other data sources, allowing users to view and interact with data in a grid format.

- Syntax Example:

```vb
Dim connectionString As String = "Data Source=myServerAddress;Initial Catalog=myDataBase;User Id=myUsername;Password=myPassword;"
Dim queryString As String = "SELECT * FROM Customers"

Using connection As New SqlConnection(connectionString)
    Dim adapter As New SqlDataAdapter(queryString, connection)
    Dim dataSet As New DataSet()

    adapter.Fill(dataSet, "Customers")

    ' Data is now available in the dataSet.Tables("Customers") table
End Using
```