Data access in Visual Basic:

(1) Direct Access Object (DAO)

The DAO was introduced in **VB3.0**. It connects to a database. This technology allows accessing and manipulating **local databases**. It is used for small database like **Ms Access, Dbase, and FoxPro etc**.The main drawback of this technology is that it is not designed to access **remote databases.**

(2) Remote Data Object (RDO)

The RDO was introduced in **VB4.0**.it is designed for accessing remote databases. It is useful for accessing data from **relational database** such as MS SQL and Oracle .It does not access **desktop database.**

(3) Active X Data Object (ADO)

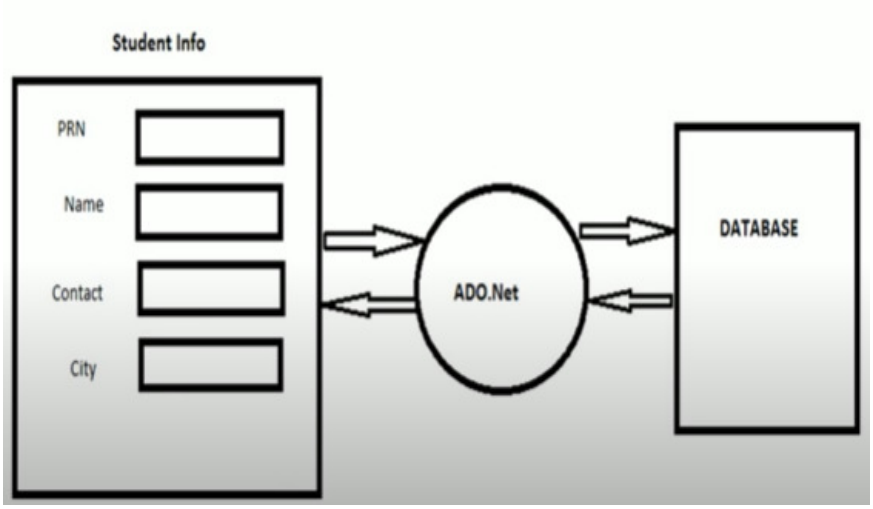
The ADO was introduced in **VB6.0**.It is Microsoft newest data access technology and provide access to almost any data stored in different format. It enables accessing data from **relational and non relational databases**. It works with **connected architecture**. Which means when you access the data from data source, such as viewing or updating data, ADO **recordset** is **keeping connection** with the data source.

(4) ADO.NET

ADO.NET is a database technology of .NET Framework used to connect **application system (Frontend Application) and database server (Backend Application).** It is a part of the .NET Framework. It consists of a **set of classes used to handle data access**. It uses **XML** to store and transfer data among applications, which is not only an industry standard but also provide fast access of data for desktop and distributed applications. It works with **disconnected architecture.**

ADO.NET

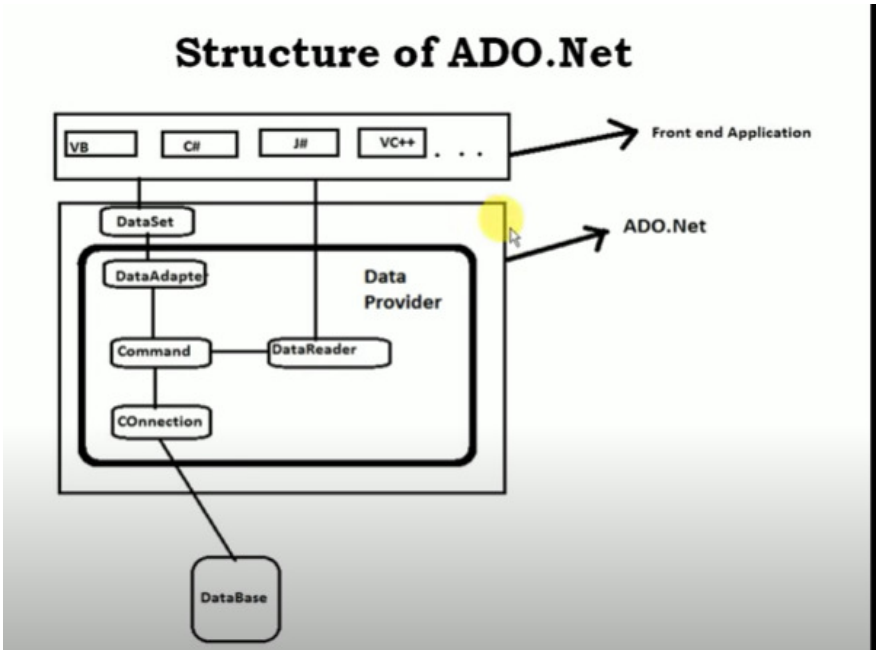
Ado.Net is a technology which works between **database and Frontend Application**. It is used to access database. For Example



Structure of Ado.Net

It has two components:

- Data Providers
- Dataset



Data Providers:

It is responsible for **providing and maintaining the connection to the database.** We can use following data provider in Ado.Net

- Oledb (for Access Database)
- Sqlclient(for sqlserver Database)
- Oracle (for oracle Database)
- Odbc (for odbc Databse)

Data provider consist following classes (Ado.Net Core Object)

- **Connection class**: used to establish connection
- **Command class**: used to execute sql query
- **DataReader Class**: used to read the result set
- **DataAdapter Class**: interface between dataset and database.

Oledb Dataprovider consist following classes

- OledbConnection
- OledbCommand
- OledbDataReader
- OledbDataAdapter

SqlClient Dataprovider consist following classes

- sqlConnection
- sqlCommand
- sqlDataReader
- sqlDataAdapter

And same for remaining DataProvider.

(1) Connection Object:

- This is the object that allows you to **establish a connection** with the data source.
- The connection object helps to identify the **database server**, the **database name**, **user name, Password** and other parameters those are required for connection to the database

Properties

ConnectionString	It stores the connection string that is passed to the connection object at the time of creating its object
Database	It stores the name of the database to which you need to connect
State	It returns the state of the connection i.e **IsClose or IsOpen**
ConnectionTimeOut	Gets the time to wait while trying to establish a connection before terminating the attempt and generating an error.

Methods

Open	It opens the connection
Close	It closes the connection
BeginTransaction	It creates the transaction object

(2) Command Object:

- It uses the connection object to **execute SQL queries**.
- The queries can be in the Form of **Inline text, Stored Procedures or direct Table access.**
- If a select query is issued, the result set it returns is usually stored in either a **DataSet or a DataReader** object.

Properties:

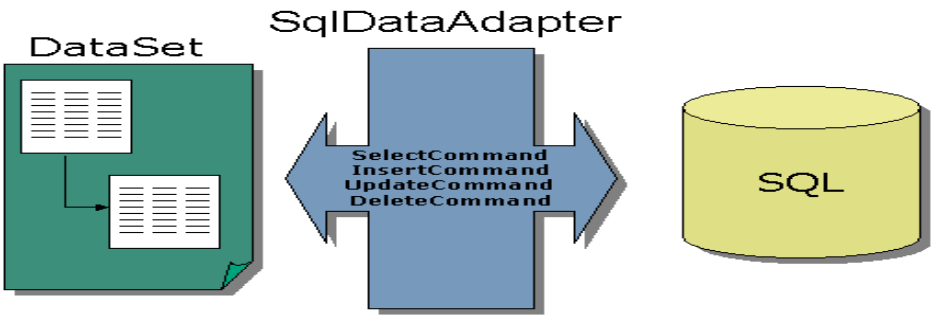
Property	**Description**
Connection	To set connection object
CommandText	It specifies SQL Statement or the name of the Stored Procedure.
CommandType	This property indicates how the CommandText property should be interpreted. The possible values are: • Text (T-SQL Statement) • StoredProcedure (Stored Procedure Name) • TableDirect
ConnectionTimeOut	Gets the time to wait while trying to establish a connection before terminating the attempt and generating an error.

Methods

Method	**Description**
ExecuteNonQuery	Used to execute an SQL statement and **returns the no. of rows affected** by the given statement. Return type of this method is **int**
ExecuteScalar	Used to execute an SQL statement and **return a single value**.
ExecuteReader	Used to execute a select a statement and return the rows returned by the select statement as a DataReader.

(3) DataAdapter:

- It acts as a **bridge between the DataSet and the database.** This helps the Dataset to contain data from multiple databases or other data source.
- It has commands like Select (retrieve data from the database) and Insert, Update and Delete (Used to send changes to the data in dataset to database).



Properties

Property	Description
SelectCommand	Used to hold a command that retrieve data from the data source.
InsertCommand	Used to hold a command that insert data into the data source.
UpdateCommand	Used to hold a command that updates data to the data source.
DeleteCommand	Used to hold a command that delete data from the data source.
CommandType	This property indicates how the CommandText property should be interpreted. The possible values are: • Text (T-SQL Statement) • StoredProcedure (Stored Procedure Name) • TableDirect

Methods

Method	Description
Fill	It is used to populate a dataset object with the data that the DataAdapter object retrieve from the data and store using its SelectCommand.
Update	It is used to update the database according to the changes that are made in the Dataset

(4) DataReader:

- It is used to read data from database. It reads data in **read-forward only manner.**
- It is working on **connected mode**.
- It is an alternative to the Dataset and DataAdapter combination.
- It is faster than DataSet.
- The **Command.ExecuteReader** method creates and returns a DataReader object

Properties

Property	Description
FieldCount	It is used to get the number of columns in the current row.
HasRows	It is used to get a value that indicates whether the SqlDataReader contains one or more rows.
IsClosed	It is used to retrieve a boolean value that indicates whether the specified SqlDataReader instance has been closed
Item	It is used to get the value of the specified column
RecordsAffected	It is used to get the number of rows changed, inserted or deleted by execution of the Transact-SQL statement

Methods:

Method	Description
Read()	It is used to read record from the SQL Server database
Close()	It is used to closes the SqlDataReader object
GetName(index)	It is used to get the name of the specified column
GetSchemaTable()	It is used to get a DataTable that describes the column metadata of the SqlDataReader
GetValue(Index)	It is used to get the value of the specified column
GetValues	It is used to populate an array of objects
NextResult()	It is used to get the next result, when reading the results of SQL statements

Example: Login Form, Display Data According to ID

DataSet:

- It is **disconnected architecture** that means there is no need to active connections during
- work with dataset.
- It is a collection of DataTables and Relation between tables. It is in memory representation
- of Data. It can be considered as a local copy of the portions of the database. It is persisted
- in memory and it can be manipulated and updated independent of the database.
- When the use of this dataset is finished, changes can be made back to the central the
- database for updating.
- Dataset class resides in **System. Data** Namespace.
- **Example: Dispdata()**

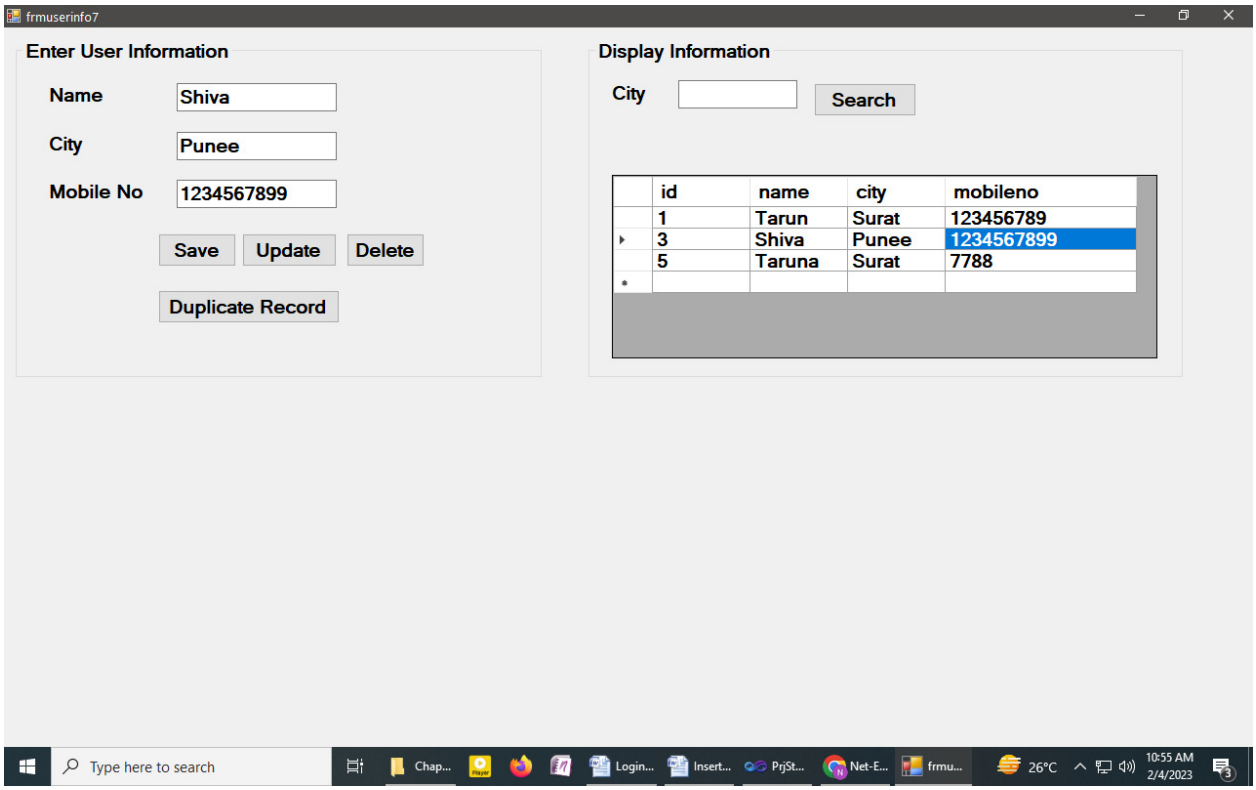
```
Try
        cmd.CommandText = "select * from tblstudent order by id"
        Dim da As New SqlDataAdapter(cmd)
        ds = New DataSet
        da.Fill(ds, "tblstudent")
        DataGridView1.DataSource = ds.Tables("tblstudent")
  Catch ex As Exception
        MessageBox.Show(ex.Message)
  End Try
```

Differences between DataSet and DataReader

DataSet	DataReader
It is disconnected object and can provide access to data even when connection to Database was closed.	It is connected object and can not provide access to data when connection to database Was closed.
It can store data from multiple tables	It can store data from only one table.
It allows insert, update and delete on data	It is read only and it doesn't allow insert, Update and delete on data.
It allows navigation between record either Forward or backward.	It allows only forward navigation that also Only to immediate next record.
It can contain multiple records.	It can contain only one record at a time.

All the data of a dataset will be on client	All the data of a DataReader will be on server and one record at a time is retrieved and stored in DataReader when you call the Read() method of datareader.

Design of the form



Source Code:

```
Imports System.Data.SqlClient
Public Class frmuserinfo7
    Dim con As New SqlConnection
    Dim cmd As New SqlCommand
    Dim intid As Integer
    Private Sub frmuserinfo7_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Try
            con.ConnectionString = "Data
Source=.\SQLEXPRESS;AttachDbFilename=F:\NNM\Sutex_Data\SYBCA\Vb.Net\2019\Proj
ect\PrjStudent\PrjStudent\dbstudent7.mdf;Integrated Security=True;User
Instance=True"
            cmd.Connection = con
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
        dispdata()
    End Sub
    'datepicker1.value.Tostring("yyyy/MM/dd")
    Private Sub btnsave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnsave.Click
```

```
        Try
            cmd.CommandText = "insert into tblstudent values('" &
txtname.Text & "','" & txtcity.Text & "','" & txtmobileno.Text & "')"
            con.Open()
            cmd.ExecuteNonQuery()
            con.Close()
            MsgBox("Record is inserted")
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
        dispdata()
        clearcontrols()
    End Sub
    Sub dispdata()
        Try
            cmd.CommandText = "select * from tblstudent order by id"
            Dim dt As New DataTable
            Dim da As New SqlDataAdapter(cmd)
            da.Fill(dt)
            DataGridView1.DataSource = dt
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End Sub

    Private Sub DataGridView1_CellClick(ByVal sender As Object, ByVal e As
System.Windows.Forms.DataGridViewCellEventArgs) Handles
DataGridView1.CellClick
        Try
            intid = DataGridView1.Rows(e.RowIndex).Cells(0).Value
            txtname.Text = DataGridView1.Rows(e.RowIndex).Cells(1).Value
            txtcity.Text = DataGridView1.Rows(e.RowIndex).Cells(2).Value
            txtmobileno.Text = DataGridView1.Rows(e.RowIndex).Cells(3).Value
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try

    End Sub

    Private Sub btndelete_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btndelete.Click
        Try
            cmd.CommandText = "delete from tblstudent where id=" & intid & ""
            con.Open()
            cmd.ExecuteNonQuery()
            con.Close()
            MsgBox("Record is deleted")
        Catch ex As Exception
            MsgBox(ex.Message)
```

```
        End Try
        dispdata()
        clearcontrols()
    End Sub

    Private Sub btnupdate_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnupdate.Click
        Try
            cmd.CommandText = "update tblstudent set name='" & txtname.Text &
"',city='" & txtcity.Text & "',mobileno='" & txtmobileno.Text & "' where id="
& intid & ""
            con.Open()
            cmd.ExecuteNonQuery()
            con.Close()
            MsgBox("Record is updated")
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
        dispdata()
        clearcontrols()
    End Sub
    Sub clearcontrols()
        txtname.Text = ""
        txtcity.Text = ""
        txtmobileno.Text = ""
    End Sub

    Private Sub btnsearch_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnsearch.Click
        Try
            cmd.CommandText = "select * from tblstudent where city='" &
txtsearchcity.Text & "'"
            Dim dt As New DataTable
            Dim da As New SqlDataAdapter(cmd)
            da.Fill(dt)
            DataGridView1.DataSource = dt
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End Sub

    Private Sub btndup_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btndup.Click
        Try
            cmd.CommandText = "select id from tblstudent where name='" &
txtname.Text & "'"
            con.Open()
            Dim intdupid As Integer = cmd.ExecuteScalar()
            If intdupid > 0 Then
```

```
                MsgBox("user is already exists")
            End If
            con.Close()
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End Sub
End Class
```