5.1. Visual Database Tools

5.2. ADO .NET Object Model

5.3. ADO .NET Programming

5.1. Visual Database Tools

- The Visual Database Tools are a combination of design tools you can use to work with a data source.
- You can use them to create queries, design or modify a database structure, or update data.
- The tools are Database Diagram Designer, Table Designer, and Query and View Designer.
- The database is design, it allows you to create database object such as tables, columns, keys, indexes, relationships, constraints and views.

Visual database tools provide 3 designers to create these objects:

I. Database diagram
II. Table designer
III. Query and View designer

i. Database Diagram

Database diagram tool allows user to design and visualize a database to which we are connected. This diagram contains the tables along with the relationship and primary key.

ii. Table designer

It provides an entire window to design the individual table. It also allow user to change an existing database by changing, adding, duplicating and deleting table. It is used to add or remove columns, specify its data type, size, description, keys, constraints and relationships.

iii. Query and View designer

It helps user to create and maintain the data retrieval and data vmanipulation.When user designs any query,view,functions,or stored procedure,the designer is made up of panes.

❖ ADO.NET

- ADO.NET has the ability to separate data access mechanisms,data manipulation mechanisms and data connectivity mechanisms.
- ADO.NET ia a set of classes that allow application to read and write information in database.
- ADO.NET can be used by any .NET language.
- We need to add System.Data namespace for work with ADO.NET.
- Ado.Net is a technology which works between access database Frontend Application. It is used to access database.

5.2. ADO.NET Object Model

- ADO.NET is an object-oriented set of libraries that allows you to interact with data sources.
- The data source is a database,but it could also be a text file,an Excel spread-sheet,or an XML file.
- There are many different types of databases available such asMicrosoftSQLSever,MicrosoftAccess,Oracle,BorlandInterbase,IBMDB2 etc.

Connected & Disconnected Data(Architecture)

The data access with ADO.NET consists of two parts:

(1) Data Provider

(2) DataSet

(1) Data Provider

- The Data Provider is responsible for providing and maintaining the connection to the database.

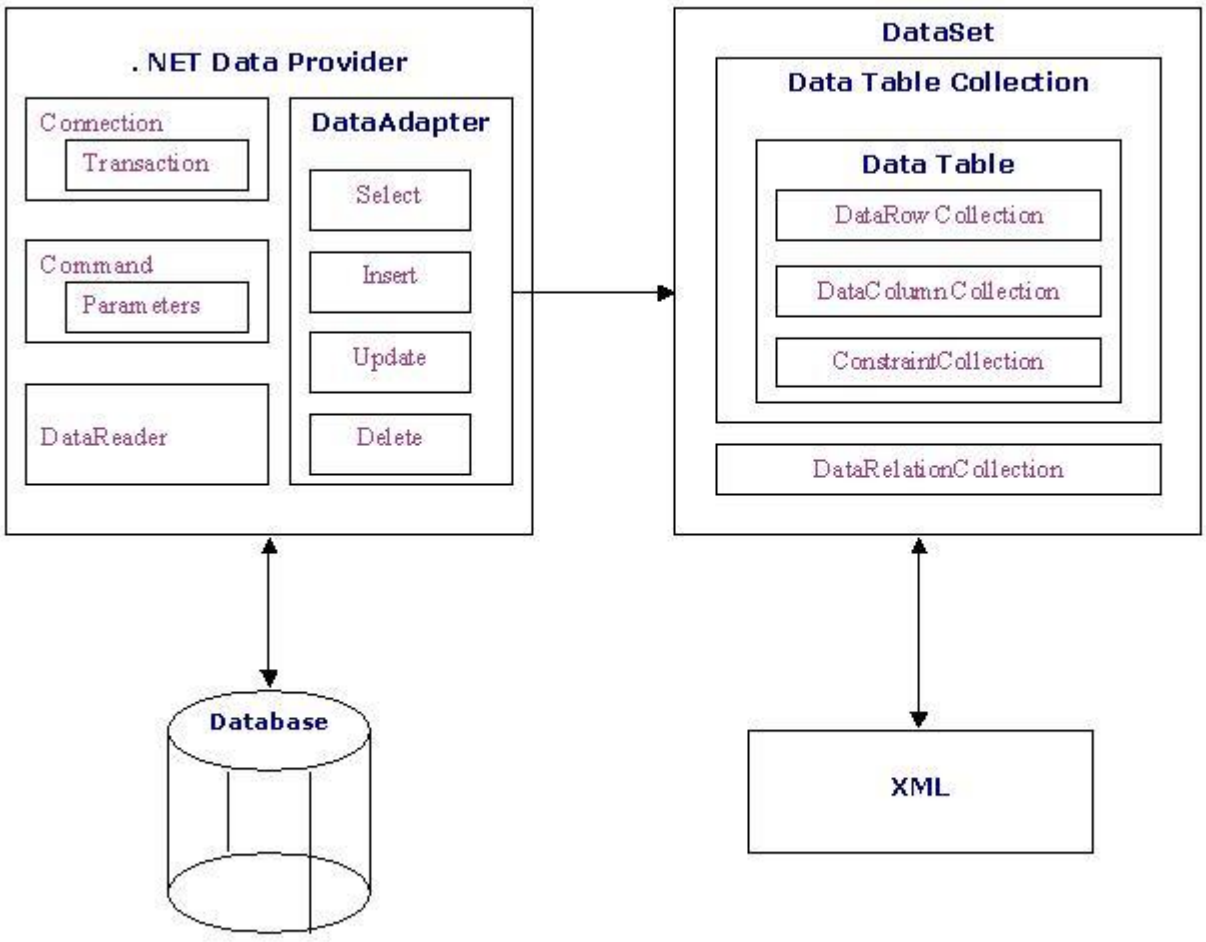
- A DataProvider is a set of related components that work together to provide data in an efficient and performance driven manner.
- The .NET Framework currently comes with two DataProviders: the SQL Data Provider which is designed only to work with Microsoft's SQL Server 7.0 or later and the OleDb DataProvider which allows us to connect to other types of databases like Access and Oracle. Each DataProvider consists of the following component classes:

The Connection object which provides a connection to the database
The Command object which is used to execute a command
The DataReader object which provides a forward-only, read only, connected recordset
The DataAdapter object which populates a disconnected DataSet with data and performs update

Data access with ADO.NET can be summarized as follows:

- A connection object establishes the connection for the application with the database.
- The command object provides direct execution of the command to the database. If the command returns more than a single value, the command object returns a DataReader to provide the data.

- Alternatively, the DataAdapter can be used to fill the Datasetobject. The database can be updated using the command object or the DataAdapter.



ADO .NET Data Architecture

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)

- **Ado.NET Objects**

  ADO.NET consists of many objects that are used to work with data.

  a) Connection Object
  b) Command Object
  c) DataAdapter Object
  d) DataReader Object

a) Connection Object

- To establish connection with a database,you must have a connection object.
- The connection object helps to identify the datadase sever,the database name,user name,password,and other parameters that are required for connecting to the database.
- A connection object is used by command objects so that it will know on which database the command is executed.

Connection String – A string that specifies information about a data source and the means of connection to it is called Connection String.

```
Dim con As New SqlConnection

    con.ConnectionString = "Data
    Source=.\SQLEXPRESS;AttachDbFilename=D:\jigisha\vb.netDemo_s
    y6\sy6\sy6\Database1.mdf;Integrated Security=True;User
    Instance=True"
```

Properties

Properties	Description
conectionString	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 return the state of thae connectionEX.IsClose or IsOpen

Connection TimeOut	Gets the time to wait while trying to establish a connection before terminating the attempt and generating an error.

Methods

Methods	Description
Open	It opens the connection
Close	It closes the connection
BeginTransaction	It creates the Transaction Object.
CreateCommand	It creates and returns a SqlCommand object associated with the SqlConnection.
ChangeDatabase	It changes the current database for an open SqlConnection.

(b) Command Object

- It is used to retrieve a subset of data.Also invoking SQL statements insert,Update and Delete are directly require to set certain parameters on the command before executing the statement.common use of the command object
- Is to execute stored procedure and pass the appropriate parameters to the stored procedure.

Properties

Properties	Description
Connection	To set a connection object.
CommandText	It specifies the SQL string or stored procedure to be executed.
CommandType	It is used to determine how to interpret command text. Ex. CommandText is storedprocedure or Text or DirectTable.
CommandTimeOut	Gets the time to wait while trying to execute the command before terminating the attempt and generating an error.

Methods

Methods	Description
ExecuteNonQuery	It will execute the SQL statement and returns the number of rows affected by the query.
ExecuteScalar	It will execute the SQL statement which return the

	singleton value.
ExecuteReader	It will execute the SQL statement and returns the records in the form of DataReader.Ex.it is used to create the object of DataReader.
CreateParameter	It creates and returns a Sqlparameter object associated with the SqlCommand.
Cancel	It is used to cancel the command given for for execution.
ResetCommandTimeOut	It is used to reset Command time out property to its default value.

(C) DataReader Object

- A SqlDatReader is used to read data in the most efficient manner.You cannot use it for writing data.
- You can read forward-only and in sequential manner from SqldataReader.

Properties

Properties	Description
FieldCount	It stroes number of fields in a row.
HasRows	It specifies that the rows are selected or not for reading.
IsClosed	It specifies that DataReader is closed or not.
RecordsAffected	It returns -1 as DataReader is created on server.
Item	It gets the value of the specified column name.

Methods

Methods	Description
Read	It reads the Next Record of DataReader.
Close	It is used to Close the DataReader Connection with the database.
IsDBNull	It checks that the value of the column is Null or net.
GetSchemaTable	It returns the object of the DataTable for which the DataReader is created.
GetValues	It returns the array of the values for the row.
NextResult	It is used to nagate from one record set to another when

	more than one record sets are used in the command.

(e) DataAdapter

- It acts as a bridge between data source and in-memory data objects such as the Dataset.

Properties

Properties	Description
selectCommand	It is used to hold a Command that retrieves data from the data source.
UpdateCommand	It is used to hold a Command that updates data from the data source.
DeleteCommand	It is used to hold a Command that delete data from the data source
InsertCommand	It is used to hold a Command that insert data from the data source
Command and Type	It indicates CommandText property which contains SQL statement or stored procedure.If commandText property contains stroed procedure than user can set the value to CommandType.stored procedure.Default value is CommandType.Text for SQL statement.

Methods

Methods	Description
Fill	It is used to populate a dataset object with the data that the DataAdapter object retrieve from the data store using its SelectCommand.But before that we must initialize a Dataset object.
Update	It is used to update the database according to the changes that are made in the DataSet.

(2) DataSet

- The dataset is a disconnected, in-memory representation of data. It can be considered as a local copy of the relevant portions of the database.
- The DataSet is continue in memory and the data in 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 database for updating.
- The data in DataSet can be loaded from any valid data source like Microsoft SQL server database, an Oracle database or from a Microsoft Access database.

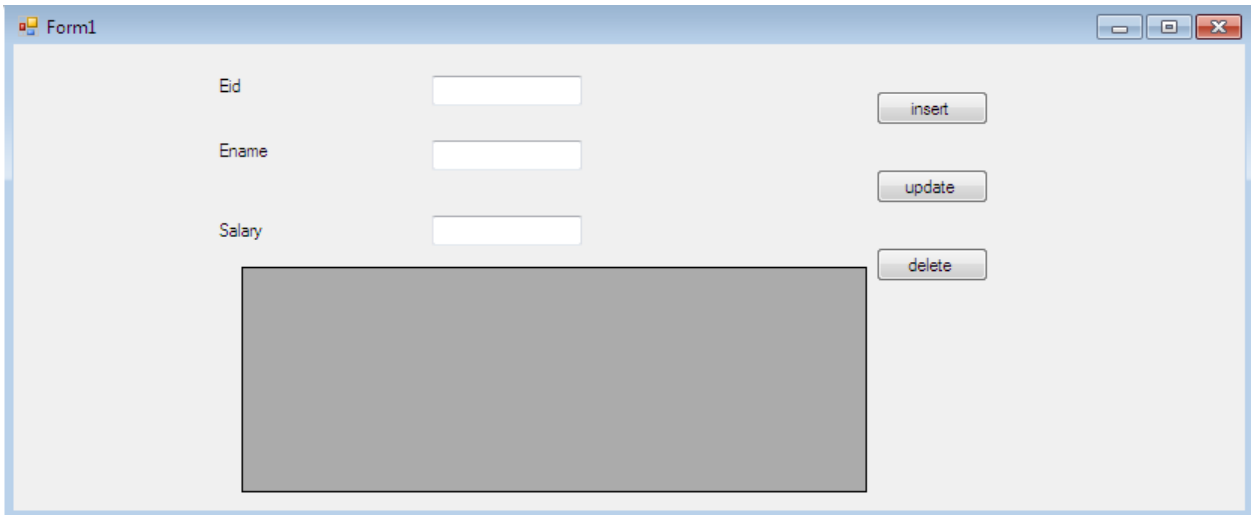
5.3. ADO.NET Programing

```
Imports System.Data.SqlClient
Public Class Form1
    Dim con As New SqlConnection
    Dim cmd As New SqlCommand
    Dim Eid As Integer


    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        Try

            con.ConnectionString = "Data
Source=.\SQLEXPRESS;AttachDbFilename=D:\jigisha\vb.net
Demo_Example\Demo_Example\Demo_Example\Database1.mdf;Integrated Security=True;User
Instance=True"
            cmd.Connection = con

        Catch ex As Exception
            MsgBox(ex.Message)

        End Try
        display()
    End Sub

    Sub display()
        Try
            cmd.CommandText = "select * from Employee"
            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 insert_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles insert.Click
        Try
```

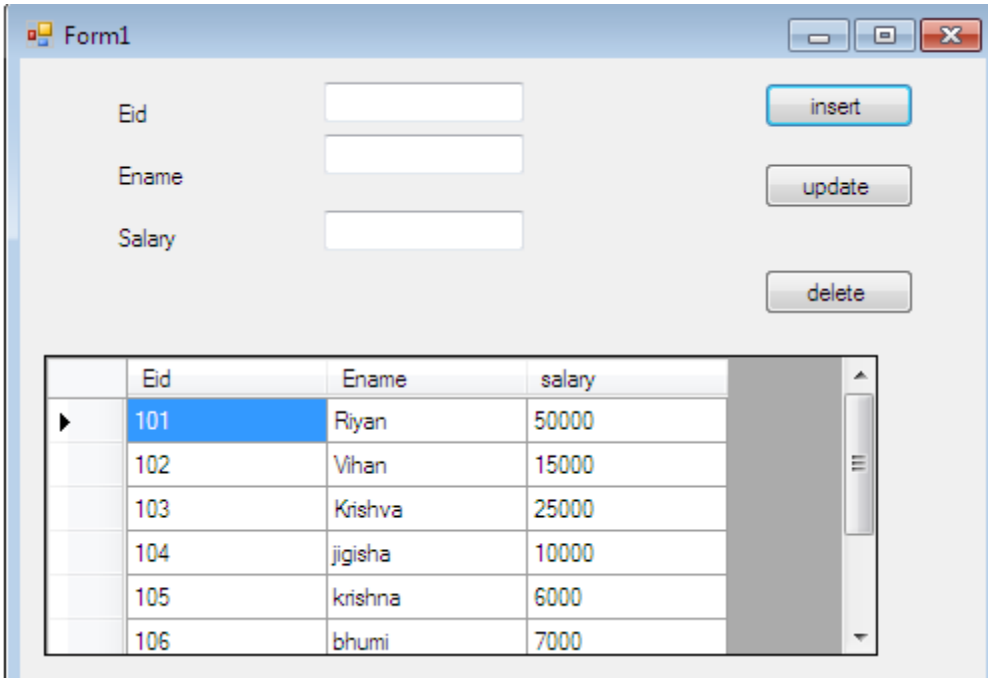
```
            cmd.CommandText = "insert into Employee values('" & txtid.Text & "','" &
txtname.Text & "'," & txtsalary.Text & " )"
            con.Open()
            cmd.ExecuteNonQuery()
            con.Close()
            MsgBox("Record is inserted")
        Catch ex As Exception
            MsgBox(ex.Message)

        End Try
        display()
        clearconrtol()
    End Sub

    Private Sub DataGridView1_CellContentClick(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.DataGridViewCellEventArgs) Handles DataGridView1.CellContentClick
        Try
            txtid = DataGridView1.Rows(e.RowIndex).Cells(0).Value
            txtname = DataGridView1.Rows(e.RowIndex).Cells(1).Value
            txtsalary = DataGridView1.Rows(e.RowIndex).Cells(2).Value

        Catch ex As Exception
            MsgBox(ex.Message)

        End Try
    End Sub

    Private Sub update_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles update.Click
        Try
            cmd.CommandText = "update Employee set Ename ='" & txtname.Text & "', salary
=" & txtsalary.Text & " where Eid = '" & txtid.Text & "' "
            con.Open()
            cmd.ExecuteNonQuery()
            con.Close()
            MsgBox("Record is updated")
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
        display()
        clearconrtol()

    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
        Try
            cmd.CommandText = "delete from  Employee where Eid=" & txtid.Text & " "
            con.Open()
            cmd.ExecuteNonQuery()
            con.Close()
            MsgBox("Record is deleted")
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
        display()
        clearconrtol()
```

```
    End Sub
    Sub clearconrtol()
        txtname.Text = ""
        txtid.Text = ""
        txtsalary.Text = ""
    End Sub
End Class
```

Output:



Output:


Form1
Eid
Ename
Salary
insert
update
delete
Eid
Ename
salary
101
Riyan
50000
102
Vihan
15000
103
Krishva
25000
104
jigisha
10000
105
krishna
6000
106
bhumi
7000