

VISUAL PROGRAMMING

Database Connection

Zuhaib A. Shaikh,
Asst. Prof., CSE Deptt., QUEST
Web: zuhaib-shaikh.neocities.org

Introduction

- MS Visual Studio provides several database connection drivers
- Connections can be made and queries can be executed
- To connect MS Access DB, OleDb (Object Link and Embedding DB) driver is used, by using System.Data.OleDb
- Operations to be performed on database via SQL language
- Database should be connected and disconnected via provided driver
 - Adding database as data source
 - Database connection
 - Initiating connection: To work with database
`OleDbConnection myCon = new OleDbConnection(Provider_driver;path;username;password);`
 - Opening connection: To open connection for executing queries
`myCon.Open();`
 - Closing connection: To close connection with database
`myCon.Close();`

Fetching record(s)

- DML Operations:

- Reading record(s) from table

- Writing SQL Query

- ```
var myQry = "ANY_SQL_SELECT_QUERY;";
```

- Setting SQL command

- ```
var myCmd = OleDbCommand(myQry, myCon);
```

- Executing reader command

- ```
var myRdr = command.ExecuteReader();
```

- Reading Data

- myRdr.read() – reads a single line form returned data. While loop can be used to perform read operations on all of the data.

- myRdr["*Column\_name*"].ToString() – returns the data from specified column

E.g:

# Deleting record(s)

- DML Operations:
  - Deleting record(s) from table
    - Setting SQL Command  
`var myCmd = OleDbCommand();`
    - Setting SQL command connection  
`myCmd.Connection = myCon;`
    - Writing SQL Query  
`myCmd.CommandText = "ANY_SQL_DELETE_QUERY;"`;
    - Setting SQL command parameters  
`myCmd.Parameters.Add(@id, OleDbType.VarChar).Value = String;`
    - Executing command, returns no. of affected records  
`var n = myCmd.ExecuteNonQuery();`

E.g.

```
var myCmd = new OleDbCommand();
myCmd.Connection = myCon;
myCmd.CommandText = "DELETE FROM Login WHERE [UserID] = @uid;";
myCmd.Parameters.Add("@uid", OleDbType.VarChar).Value = tbox.Text;
var n = myCmd.ExecuteNonQuery();
```

# Updating record(s)

- DML Operations:
  - Updating record(s) in the table
    - Setting SQL Command  
`var myCmd = OleDbCommand();`
    - Setting SQL command connection  
`myCmd.Connection = myCon;`
    - Writing SQL Query  
`myCmd.CommandText = "ANY_SQL_UPDATE_QUERY";`
    - Setting SQL command parameters  
`myCmd.Parameters.Add(@id, OleDbType.VarChar).Value = String;`
    - Executing command, returns no. of affected records  
`var n = myCmd.ExecuteNonQuery();`

E.g.

```
var myCmd = new OleDbCommand();
myCmd.Connection = myCon;
myCmd.CommandText = "UPDATE Login SET [Password] = @pass WHERE UserID =
 @uid";
myCmd.Parameters.Add("@pass", OleDbType.VarChar).Value = pbox.Password;
myCmd.Parameters.Add("@uid", OleDbType.VarChar).Value = tbox.Text;
var n = myCmd.ExecuteNonQuery();
```

# Inserting a record

- DML Operations:
  - Inserting a record in the table
    - Setting SQL Command  
`var myCmd = OleDbCommand();`
    - Setting SQL command connection  
`myCmd.Connection = myCon;`
    - Writing SQL Query  
`myCmd.CommandText = "ANY_SQL_INSERT_QUERY;";`
    - Setting SQL command parameters  
`myCmd.Parameters.Add(@id, OleDbType.VarChar).Value = String;`
    - Executing command, returns no. of affected records  
`var n = myCmd.ExecuteNonQuery();`

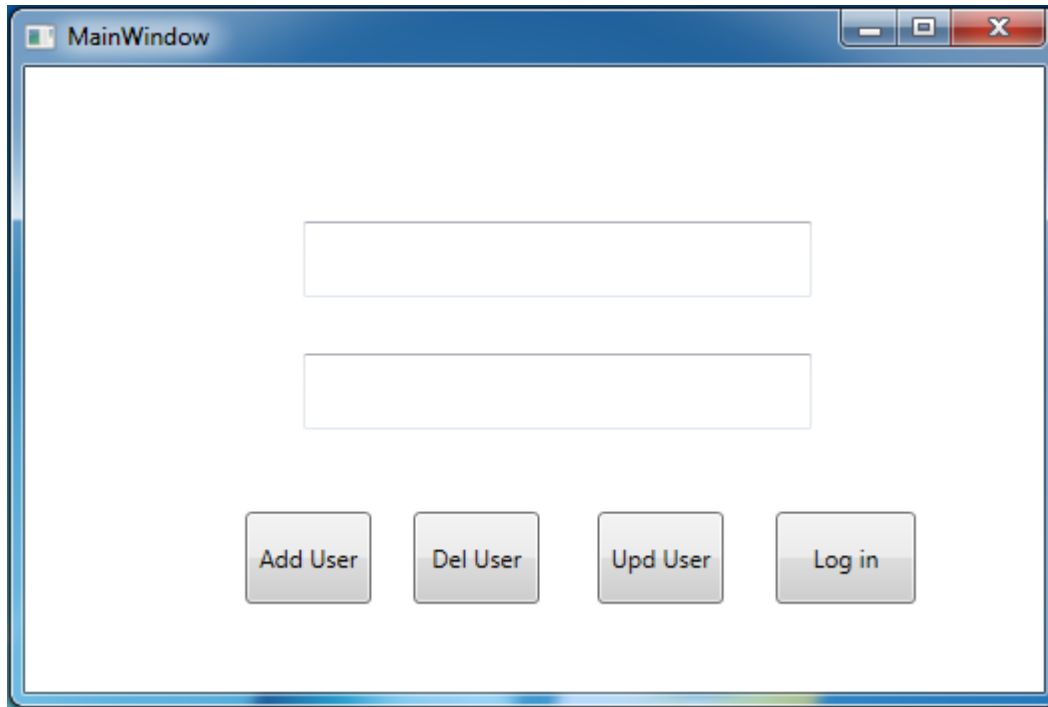
E.g.

```
var myCmd = new OleDbCommand();
myCmd.Connection = myCon;
myCmd.CommandText = "INSERT INTO Login ([UserID], [Password]) values (@uid,
 @pass);";

myCmd.Parameters.Add("@uid", OleDbType.VarChar).Value = textbox.Text;
myCmd.Parameters.Add("@pass", OleDbType.VarChar).Value = passwordbox.Text;
var n = myCmd.ExecuteNonQuery();
```



# Example: Testing DML operations (Layout)



# Example: Testing DML operations (XAML)

```
<Window x:Class="WPFDB.MainWindow"
 xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
 xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
 Title="MainWindow" Height="350" Width="525">

 <Grid>

 <Button x:Name="login" Content="Log in" HorizontalAlignment="Left" Height="46" Margin="375,222,0,0"
 VerticalAlignment="Top" Width="70" Click="Button_Click_1"/>

 <TextBox x:Name="tbox" HorizontalAlignment="Left" Height="38" Margin="139,77,0,0" TextWrapping="Wrap"
 VerticalAlignment="Top" Width="254"/>

 <PasswordBox x:Name="pbox" HorizontalAlignment="Left" Height="38" Margin="139,143,0,0"
 VerticalAlignment="Top" Width="254"/>

 <Button x:Name="auser" Content="Add User" HorizontalAlignment="Left" Height="46" Margin="110,222,0,0"
 VerticalAlignment="Top" Width="63" Click="Button_Click_1"/>

 <Button x:Name="Duser" Content="Del User" HorizontalAlignment="Left" Height="46" Margin="194,222,0,0"
 VerticalAlignment="Top" Width="63" Click="Button_Click_1"/>

 <Button x:Name="Uuser" Content="Upd User" HorizontalAlignment="Left" Height="46" Margin="286,222,0,0"
 VerticalAlignment="Top" Width="63" Click="Button_Click_1"/>

 </Grid>
</Window>
```



# Example: Testing DML operations (C#)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using System.Data.OleDb;

namespace WPFDB
{
 public partial class MainWindow : Window
 {
 public MainWindow()
 {
 InitializeComponent();
 }
 private void Button_Click_1(object sender, RoutedEventArgs e)
 {
 OleDbConnection myCon = new OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data
 Source=|DataDirectory|\\APP1.mdb")
```

# Example: Testing DML operations (C#) Cont.

```
if (sender == login)
{
 myCon.Open();
 var myCmd = new OleDbCommand();
 myCmd.Connection = myCon;
 myCmd.CommandText = "SELECT * FROM Login WHERE [UserID] = @uid;";
 myCmd.Parameters.Add("@uid", OleDbType.VarChar).Value = tbox.Text;
 var myRr = myCmd.ExecuteReader();
 if (!myRr.HasRows)
 MessageBox.Show("Invalid Username/Password");
 else
 {
 myRr.Read();
 if (pbox.Password == myRr[1].ToString())
 {
 MessageBox.Show("Welcome");
 }
 else
 MessageBox.Show("Invalid Username/Password");
 }
 myCon.Close();
}
```

# Example: Testing DML operations (C#) Cont.

```
else if (sender == auser)
{
 try
 {
 myCon.Open();
 var myQry = "INSERT INTO Login([UserID], [Password]) VALUES(@uid, @pass)";
 var myCmd = new OleDbCommand(myQry, myCon);
 myCmd.Parameters.Add("@uid", OleDbType.VarChar).Value = tbox.Text;
 myCmd.Parameters.Add("@pass", OleDbType.VarChar).Value = pbox.Password;
 var n = myCmd.ExecuteNonQuery();
 myCon.Close();
 if (n > 0)
 MessageBox.Show("User added sucessfully");
 else
 MessageBox.Show("Error occured, returned value " + n);
 }
 catch (Exception er)
 {
 MessageBox.Show("Error: " + er);
 }
}
```

# Example: Testing DML operations (C#) Cont.

```
else if (sender == Duser)
{
 myCon.Open();
 var myCmd = new OleDbCommand();
 myCmd.Connection = myCon;
 myCmd.CommandText = "DELETE FROM Login WHERE [UserID] = @uid;";
 myCmd.Parameters.Add("@uid", OleDbType.VarChar).Value = tbox.Text;
 var n = myCmd.ExecuteNonQuery();
 myCon.Close();
 if (n > 0)
 MessageBox.Show("User deleted sucessfully");
 else
 MessageBox.Show("Error occured, returned value "+n);
}
else
{
 myCon.Open();
 var myCmd = new OleDbCommand();
 myCmd.Connection = myCon;
 myCmd.CommandText = "UPDATE Login SET [Password] = @pass WHERE UserID = @uid;";
 myCmd.Parameters.Add("@pass", OleDbType.VarChar).Value = pbox.Password;
 myCmd.Parameters.Add("@uid", OleDbType.VarChar).Value = tbox.Text;
 var n = myCmd.ExecuteNonQuery();
 myCon.Close();
 if (n > 0)
 MessageBox.Show("User updated sucessfully");
 else
 MessageBox.Show("Error occured, returned value "+n);
}
}
}}
```



# Questions

