ASP.Net Database

- Objective:
  - What is a Dataset and Binding
  - What is a Dataset Control
  - Connection to DB

Display database records

During application development there will be many occasions to display database records on a web page.

Records are displayed after retrieving them from a database.
To do so, we need to use one DataSource control and one DataBound control.

We use the DataSource control -SqlDataSource control to access the data from the Sql server database.

Or Accesdatasource to access data from Access database.

The DataBound control we use to display the records is the GridView control.

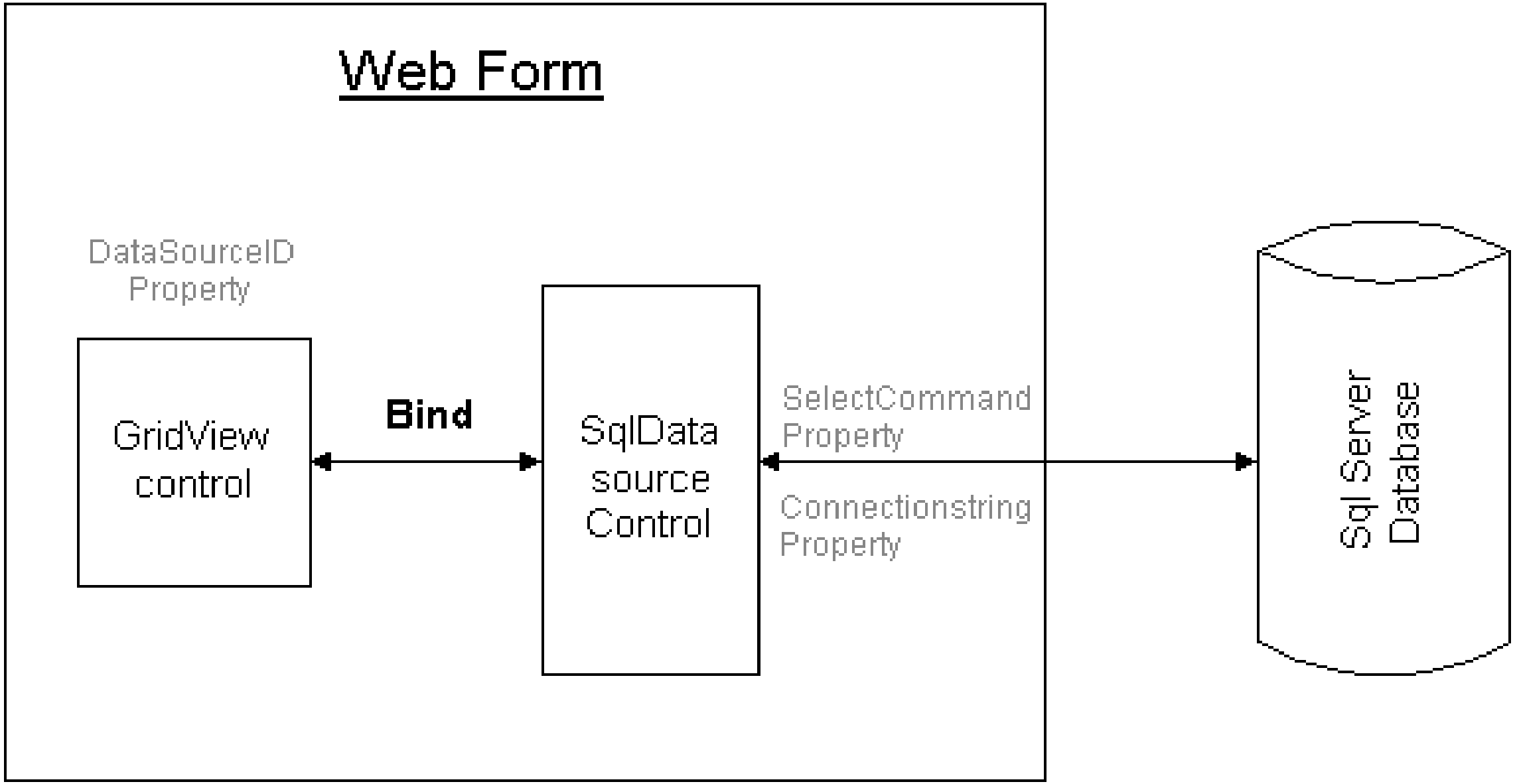
Data source controls serve as a bridge between the ASP.net page and the database.

Data source control only retrieves database data and does not have any capabilities for displaying the retrieved data on the page.

To Display data we need to use an additional web control such as Dropdownlist, Gridview, Detailview, and so on.

To be able to work with the database's data, the sqldatasource control needs to know : how to connect to the database and what query to issue to the database.

How to do this:


Web Form
DataSourceID
Property
GridView
control
Bind
SqlData
source
Control
SelectCommand
Property
Connectionstring
Property
Sql Server
Database

A **DataSet** is a container for one or more DataTable objects that contain the data you retrieve from the database.

We can set up Data Relations between these tables within the **DataSet**. The DataAdapter Object allows us to populate DataTables in a **DataSet**

Include DB in a Web form

In your Web Form applications,Switch to Design view, And follow the steps:

- From the Data group on the Toolbox, drag a SqlDataSource control or Accessdatasource control onto the page.
- If the smart tag panel is not displayed, right-click the control and then click Show Smart Tag.
- In the SqlDataSource Tasks list, click Configure Data Source.

Include DB in a Web form

- The Configure Data Source wizard is displayed
- In the Configure Data Source dialog box, click New Connection.
- If the Add Connection dialog box is displayed, click Change to display the Change Data Source dialog box.
- In the Change Data Source dialog box, click Microsoft ODBC Data Source in the Data source list, and then click OK.
- The Add Connection dialog box is displayed.

ODBC data source

- If you have an existing ODBC data source, click Use user or system data source name and select an existing ODBC data source from the list.

- If you do not have an existing ODBC data source, click Use connection string and then type the connection string or click Build to display the Select Data Source dialog box where you can build an ODBC data source name (DSN).

- If necessary, enter the username and password required to connect to the database.

Click Test connection to verify the connection to the ODBC data source, and then close the Add Connection dialog box to return to the Configure Data Source wizard.

Click Next, and then click Next again to have the connection string information saved as part of the Web.config file under the name you provide.

- In the Configure the Select Statement pane, if you want to use the wizard to create a SQL query, click Specify columns from a table or view and use the options in the pane to configure your query.

If you want to use the query builder or write your Select query, click Specify a custom SQL statement or stored procedure,

- Click Next, and then write your SQL statement.
- Click the UPDATE, INSERT, or DELETE tabs to create update statements.
- Click Next.
- In the Test Query pane, click Test Query to determine whether the query returns the results you want,
- and then Click Finish.

Attaching the Sample Database

- Attaching a database is a great way to add an already existing database to your development project
  - But it's a little tricky in Visual Studio
- In the Work Files folder, you will find a folder named "Sample Database"
  - Inside that folder are two files
    - XYZCompany.mdf
    - XYZCompany_log.ldf

Attaching the Sample Database

- Copy both files to this location:
  - C:\Users\username\AppData\Local\Microsoft\VisualStudio\SSDT
- With both files in this location, we can attach the database
- Let's take a look at how to attach a database in Visual Studio...

A select statement must specify what database table and columns to return

Select Column1, column2, …,columnn
from tablename

You may select * instead of entering each column. To get all columns.

To test the query, select server explorer, data connection and right click on the table

Select new query

Select * from Table where field = ‘value’

Operator :

< > <= >= <> =

Statement Syntax

Insert into tablename (col1,col2,….)
values(col1value,col2value…..)

Delete from tablename where col = NN

Update tablename set col1= value,col2,…. Where col = NN

Data Web Controls

ASP.net contains a number of web controls whose sole purpose is to display data from a data source. Control (find in toolbox under Data region. (**Gridview, Detailsview**,……)

Gridview show all the records at once,

Detailsview shows one record at a time. Enable paging to allows moving between records. You can customize paging in page setting properties.

Select the data web control and drop it on to the ASP.net page.
Then specify what data source control to use.

Using INSERT INTO on Button click

```
Protected Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim Badgeid, firstname, Lastname, dept, posit, state As String
    Badgeid = TextBox1.Text
    firstname = TextBox2.Text
    Lastname = TextBox3.Text
    dept = TextBox4.Text
    posit = TextBox5.Text
    state = TextBox6.Text

    ' Dim AccessConn1 As New System.Data.OleDb.OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=~\Batinservice.mdb")
    ' AccessConn1.Open()
    Dim constring As String = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=~\Webdata\App_Data\hospital.mdb"

  Using myconnection As New OleDbConnection(constring)
        myconnection.Open()

        Dim sqlQry As String = "INSERT INTO [employees] ([badgeid], [firstname],[lastname], [dept], [posit], [state]) VALUES (@badgeid,
@first,@lname, @dept, @posit,@state)"
        Using cmd As New OleDbCommand(sqlQry, myconnection)
            cmd.Parameters.AddWithValue("@badgeid", Badgeid)
            cmd.Parameters.AddWithValue("@first", firstname)
            cmd.Parameters.AddWithValue("@lname", Lastname)
            cmd.Parameters.AddWithValue("@dept", dept)
            cmd.Parameters.AddWithValue("@posit", posit)
            cmd.Parameters.AddWithValue("@state", state)
            cmd.ExecuteNonQuery()
        End Using
    End Using
```

ASP.NET provides robust login (authentication) functionality for ASP.NET **Web applications without requiring programming.**

The default Visual Studio project templates for Web applications and for Web sites include prebuilt pages that let users register a new account, log in, and change their passwords

Restrict Access to page

You can also create your own pages that you can add ASP.NET login controls to in order to add login functionality. To use the login controls, you create a Web pages and then add the login controls to them from the Toolbox.

Typically, you restrict access to ASP.NET pages by putting them into a protected folder. You then configure the folder to deny access to anonymous users (users who are not logged in) and to grant access to authenticated (logged-in) users.

Attaching the Sample Database

- Attaching a database is a great way to add an already existing database to your development project
  - But it's a little tricky in Visual Studio
- In the Work Files folder, you will find a folder named "Sample Database"
  - Inside that folder are two files
    - XYZCompany.mdf
    - XYZCompany_log.ldf

Attaching the Sample Database

- Copy both files to this location:
  - C:\Users\username\AppData\Local\Microsoft\VisualStudio\SSDT
- With both files in this location, we can attach the database
- Let's take a look at how to attach a database in Visual Studio...