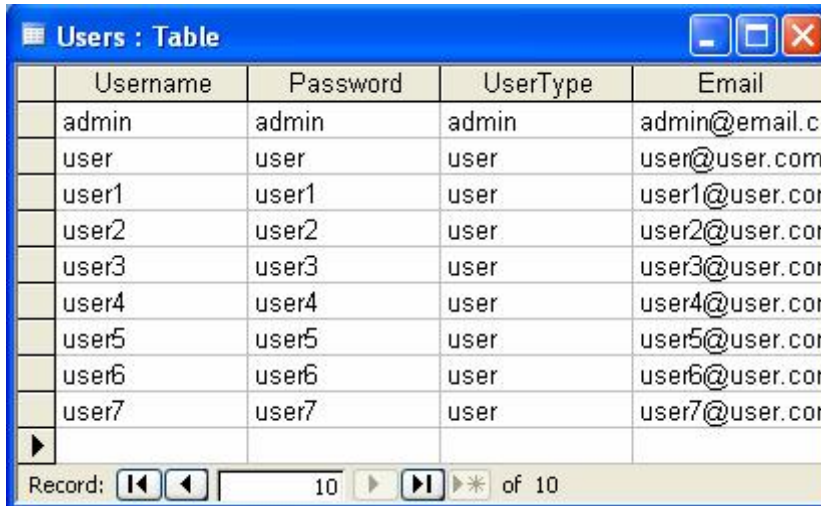


Tutorial 10: Form Authentication

Authentication

- § Authentication is the process of obtaining identification credentials such as name and password from a user and validating those credentials against some authority.
- § If the credentials are valid, the entity that submitted the credentials is considered an authenticated identity.
- § Once an identity has been authenticated, the authorization process determines whether that identity has access to a given resource.
- § These tutorial consisting 3 parts:
 1. Creating a Login Page
 2. Creating a Members Only Page
 3. Creating a Admin Only Page
- § Here is the table used to perform the user authentication.



	Username	Password	UserType	Email
	admin	admin	admin	admin@email.c
	user	user	user	user@user.com
	user1	user1	user	user1@user.cor
	user2	user2	user	user2@user.cor
	user3	user3	user	user3@user.cor
	user4	user4	user	user4@user.cor
	user5	user5	user	user5@user.cor
	user6	user6	user	user6@user.cor
	user7	user7	user	user7@user.cor

Record: 10 of 10

Creating a Login Page

- § In this example, we are going to use the Users table from the Auth database that attached with this PDF document.

Follow these steps:

1. Create a new Web Form and call the Login.aspx.
2. Switch to Design view, then open the Standard section of the toolbox and drag a 2 TextBox, 1 Label and 1 Button controls over to the page.

Login Page

Username:

Password:

3. Double click the button and enter the following statements.

```
protected void Button1_Click(object sender, EventArgs e)
{
    // Data source control that works with Microsoft Access databases
    AccessDataSource ads = new AccessDataSource();

    // Location of Microsoft Access .mdb file
    ads.DataFile = "~/App_Data/Auth.mdb";

    // Create parameters with specified names and values
    ads.SelectParameters.Add("Username", this.TextBox1.Text);
    ads.SelectParameters.Add("Password", this.TextBox2.Text);

    // Set the SQL string to retrieve data from the underlying database
    ads.SelectCommand = "SELECT * FROM [Users] WHERE [Username] = @Username
AND [Password] = @Password";

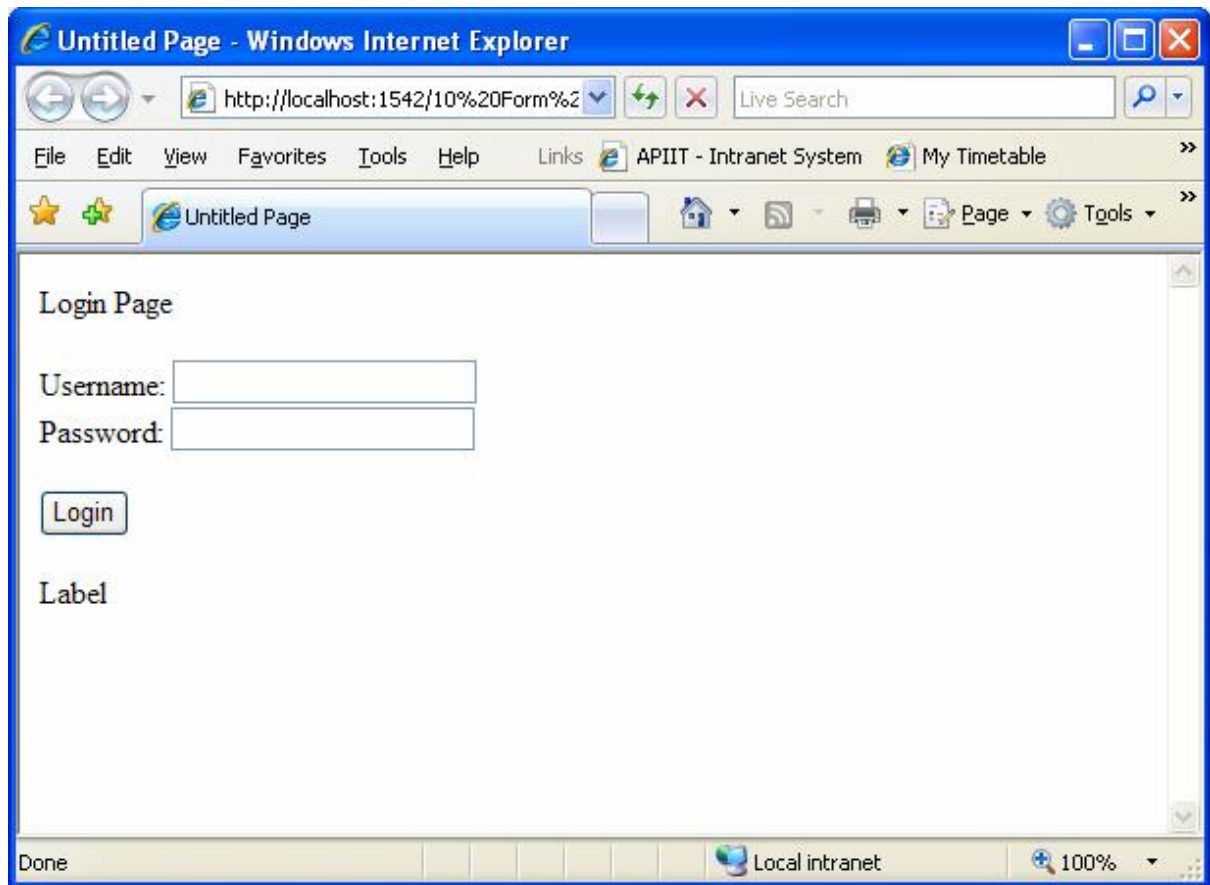
    // Retrieve data
    DataView dv = (DataView)ads.Select(DataSourceSelectArguments.Empty);

    // Display error message and return if the number of record is zero
    if (dv.Count == 0)
    {
        this.Label1.ForeColor = System.Drawing.Color.Red;
        this.Label1.Text = "Login Failed!";
        return;
    }

    // Create session variables
    this.Session["Username"] = dv[0].Row["Username"].ToString();
    this.Session["UserType"] = dv[0].Row["UserType"].ToString();

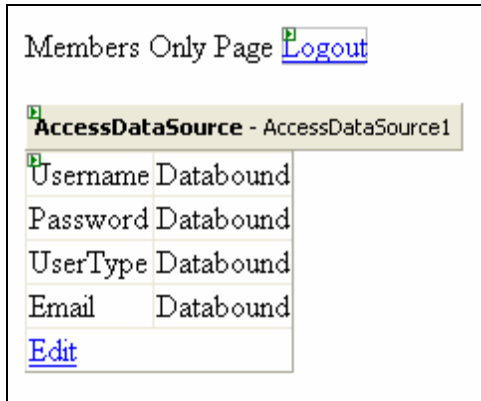
    // Redirect to respective page based on user type
    if (this.Session["UserType"].ToString().Equals("user"))
        Response.Redirect("MembersOnlyPage1.aspx");
    else if (this.Session["UserType"].ToString().Equals("admin"))
        Response.Redirect("AdminOnlyPage1.aspx");
}
```

4. Open this page in the Web browser; it will look like figure below.



Creating a Members Only Page

1. Create a new Web Form and call the MembersOnlyPage1.aspx.



2. Switch to Design view, then open the Data section of the toolbox and drag a AccessDataSource control over to the page.
3. Click Configure Data Source in the Smart Tag menu. This summons the familiar Configure Data Source Wizard dialog box.
4. In the Configure Data Source Wizard, click the browse...button. This opens the Select Microsoft Access Database dialog box.
5. In the Select Microsoft Access Database dialog box, select the Auth database and click OK button. This opens the Configure Data Source dialog box.

- Click **Next** in the Configure Data Source Wizard. The page that opens allows you to choose a table.

The screenshot shows the 'Configure Data Source - AccessDataSource1' dialog box. The title bar includes a help icon and a close button. The main area is titled 'Configure the Select Statement' and features a yellow cylinder icon. Below the title, the question 'How would you like to retrieve data from your database?' is followed by two radio buttons: 'Specify a custom SQL statement or stored procedure' (unselected) and 'Specify columns from a table or view' (selected). Under the selected option, there is a 'Name:' dropdown menu with 'Users' selected. Below that is a 'Columns:' list box containing a checked '*' and unchecked 'Username', 'Password', 'UserType', and 'Email'. To the right of the list box is a checkbox for 'Return only unique rows' (unchecked) and three buttons: 'WHERE...', 'ORDER BY...', and 'Advanced...'. At the bottom of the main area is a 'SELECT statement:' text box containing 'SELECT * FROM [Users]'. The bottom of the dialog has four buttons: '< Previous', 'Next >', 'Finish', and 'Cancel'.

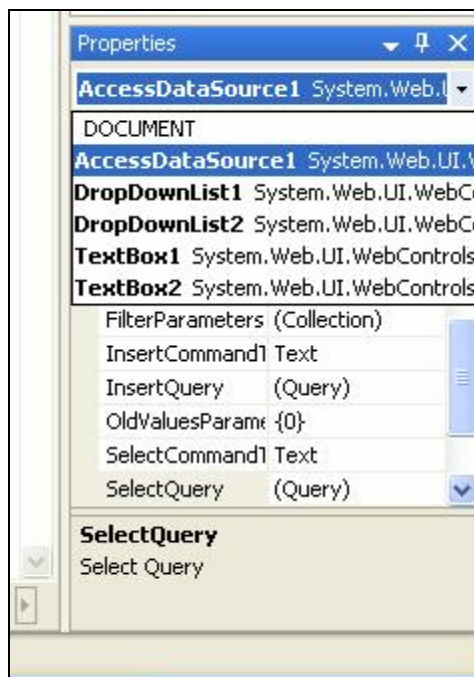
- Select the Users table and click the * in the Columns.
- Click the **Advanced** button. The Advanced SQL Generation Options dialog box opens.

The screenshot shows the 'Advanced SQL Generation Options' dialog box. The title bar includes a help icon and a close button. The main area contains the text: 'Additional INSERT, UPDATE, and DELETE statements can be generated to update the data source.' Below this are two checkboxes: 'Generate INSERT, UPDATE, and DELETE statements' (unchecked) and 'Use optimistic concurrency' (unchecked). The first checkbox has a description: 'Generates INSERT, UPDATE, and DELETE statements based on your SELECT statement. You must have all primary key fields selected for this option to be enabled.' The second checkbox has a description: 'Modifies UPDATE and DELETE statements to detect whether the database has changed since the record was loaded into the DataSet. This helps prevent concurrency conflicts.' At the bottom are two buttons: 'OK' and 'Cancel'.

9. Check the Generate INSERT, UPDATE, and DELETE Statements check box (shown in below); and then click OK to close this dialog box.

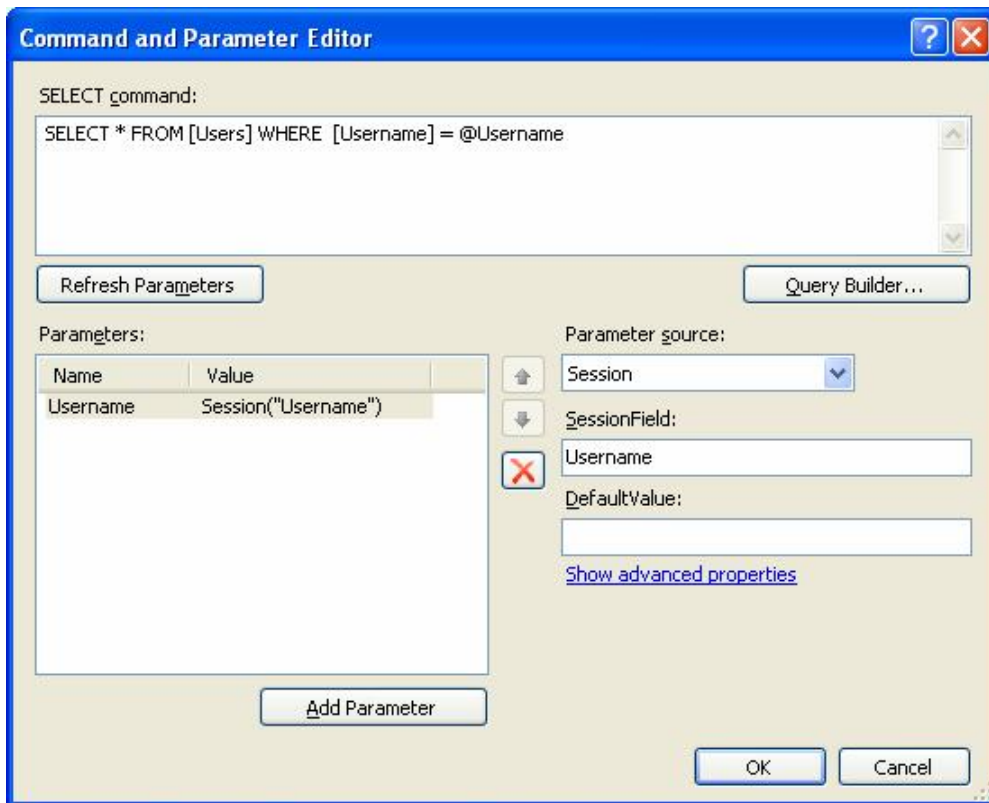


10. Click Next to test out the query. You will see a table that shows the rows from the table.
11. Click Finish. Now the data source has been configured.
12. Select AccessDataSource1 from the Properties Panel

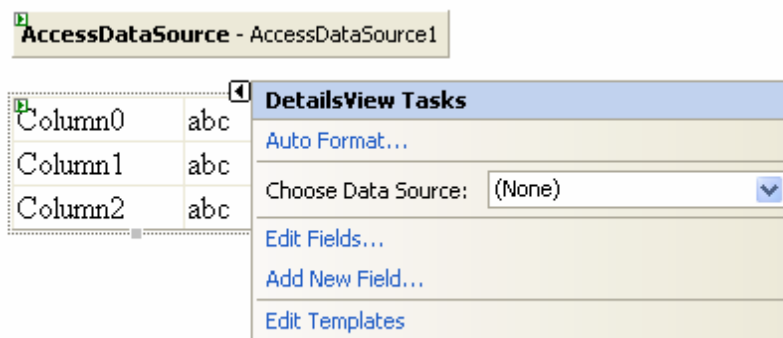


13. Select the SelectQuery property and click the icon next to it. This opens the Command and Parameter Editor.

14. Change the Select command and configure the username parameter as follow.



15. Drop a DetailView control on your form; expand its Smart Tag by clicking the little arrow in the up-right corner, as shown in figure below.



16. Click the Choose Data Source drop-down list, and select AccessDataSource1, follow by checking the Enable Paging, Enable Editing option only.

DetailsView Tasks	
Auto Format...	
Choose Data Source:	AccessDataSource1 ▼
Configure Data Source...	
Refresh Schema	
Edit Fields...	
Add New Field...	
<input type="checkbox"/>	Enable Paging
<input type="checkbox"/>	Enable Inserting
<input checked="" type="checkbox"/>	Enable Editing
<input type="checkbox"/>	Enable Deleting
Edit Templates	

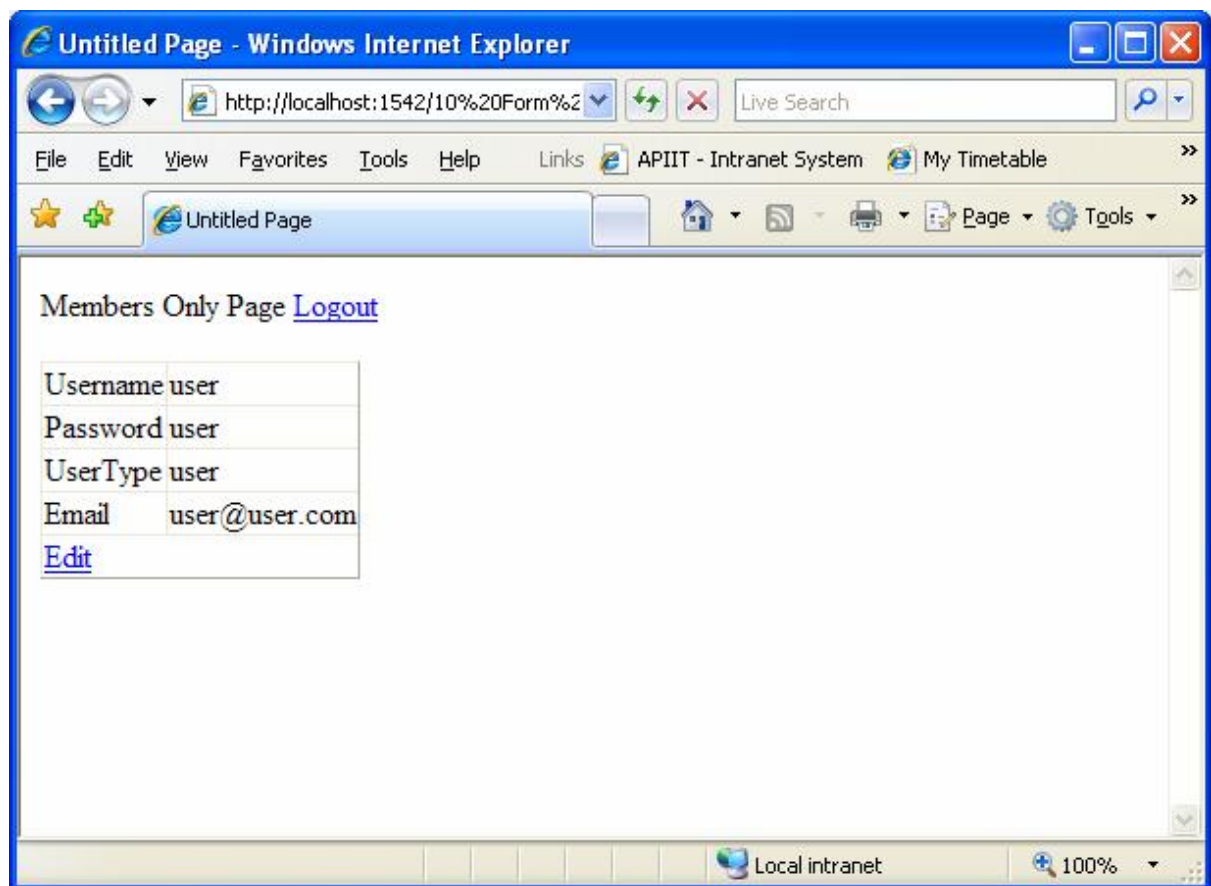
17. Double click the any blank space on the form button and enter the following statements for the page load event.

```
protected void Page_Load(object sender, EventArgs e)
{
    // Return to Login page if the session variable not exist
    if (Session["UserType"] == null ||
        !Session["UserType"].Equals("user"))
        Response.Redirect("Login.aspx");
}
```

18. Drag a LinkButton control on the page. Double click the LinkButton and enter the following statements.

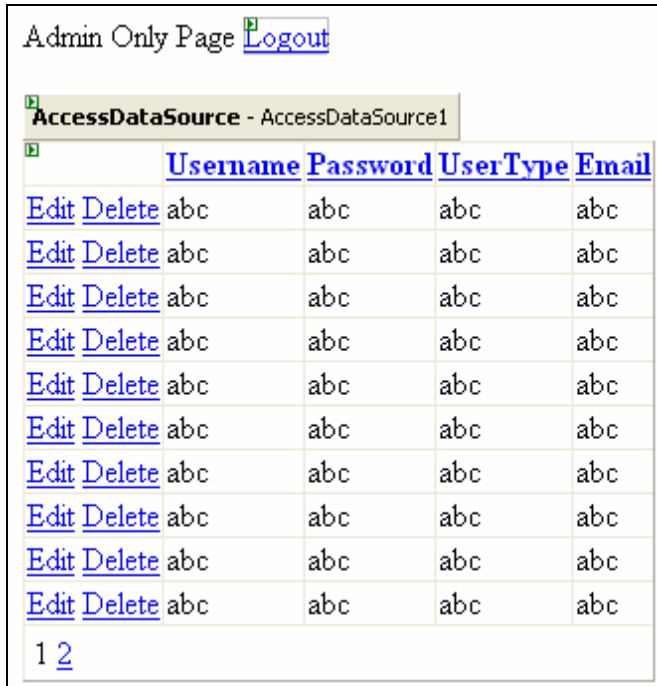
```
protected void LinkButton1_Click(object sender, EventArgs e)
{
    // Cancel the current session
    Session.Abandon();
    Response.Redirect("MembersOnlyPage1.aspx");
}
```


19. Open this page in the Web browser it will redirect to Login Page. Only successful login can view the page.



Creating an Admin Only Page

1. Create a new Web Form and call the AdminOnlyPage1.aspx.



Admin Only Page [Logout](#)

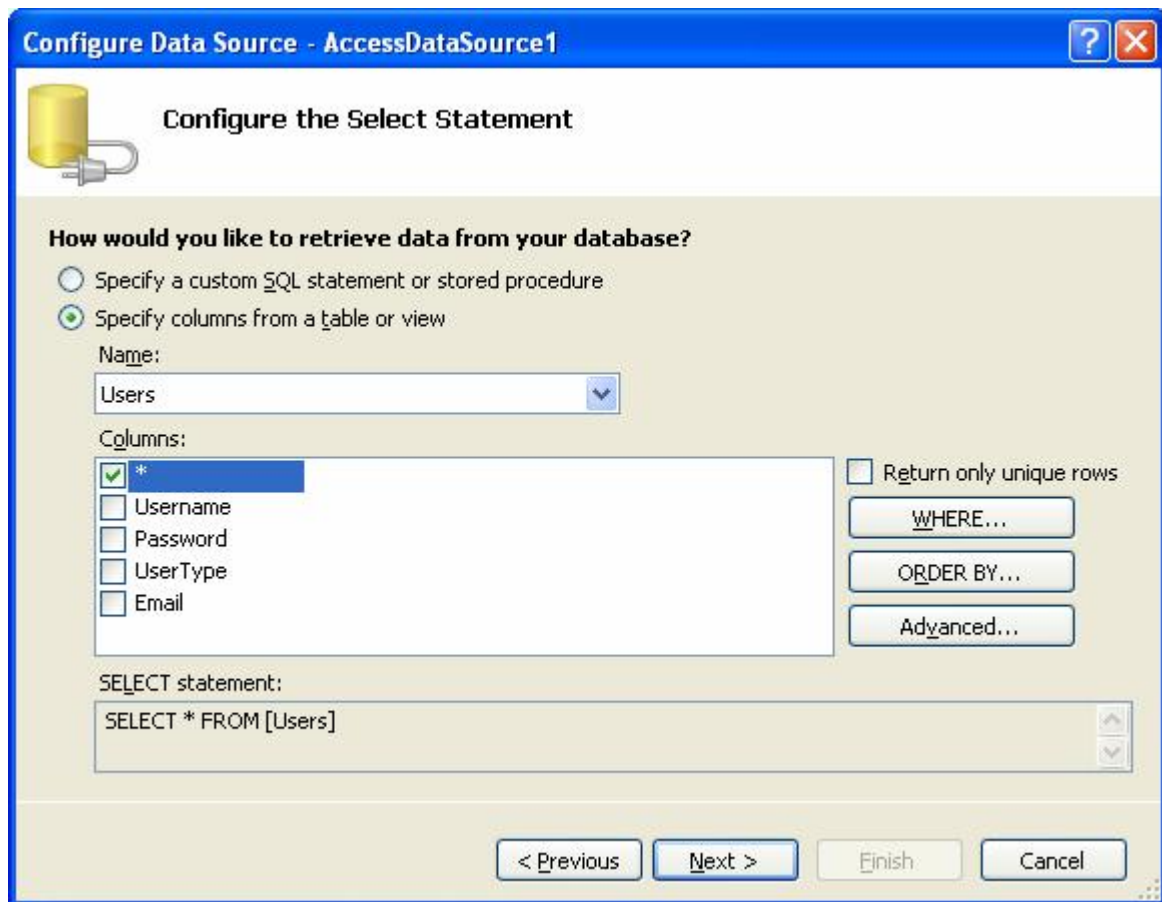
AccessDataSource - AccessDataSource1

	Username	Password	UserType	Email
Edit Delete	abc	abc	abc	abc
Edit Delete	abc	abc	abc	abc
Edit Delete	abc	abc	abc	abc
Edit Delete	abc	abc	abc	abc
Edit Delete	abc	abc	abc	abc
Edit Delete	abc	abc	abc	abc
Edit Delete	abc	abc	abc	abc
Edit Delete	abc	abc	abc	abc
Edit Delete	abc	abc	abc	abc

1 [2](#)

2. Switch to Design view, then open the Data section of the toolbox and drag a AccessDataSource control over to the page.
3. Click Configure Data Source in the Smart Tag menu. This summons the familiar Configure Data Source Wizard dialog box.
4. In the Configure Data Source Wizard, click the browse...button. This opens the Select Microsoft Access Database dialog box.
5. In the Select Microsoft Access Database dialog box, select the Auth database and click OK button. This opens the Configure Data Source dialog box.
6. Click **Next** in the Configure Data Source Wizard. The page that opens allows you to choose a table.

7. Select the Members table and click the * in the Columns. Finally, click Next to test out the query. You will see a table that shows the rows from the table.



The image shows the 'Configure Data Source - AccessDataSource1' dialog box. The title bar is blue with a question mark and a close button. The main area has a yellow background with a yellow cylinder icon and the text 'Configure the Select Statement'. Below this, the question 'How would you like to retrieve data from your database?' is followed by two radio buttons: 'Specify a custom SQL statement or stored procedure' (unselected) and 'Specify columns from a table or view' (selected). Under the selected option, there is a 'Name:' dropdown menu with 'Users' selected. Below that is a 'Columns:' list box containing a checked checkbox for '*' and unchecked checkboxes for 'Username', 'Password', 'UserType', and 'Email'. To the right of the list box is a checkbox for 'Return only unique rows' (unchecked) and three buttons: 'WHERE...', 'ORDER BY...', and 'Advanced...'. At the bottom, there is a 'SELECT statement:' text box containing 'SELECT * FROM [Users]'. The bottom of the dialog has four buttons: '< Previous', 'Next >', 'Finish', and 'Cancel'.

8. Click the Advanced button. The Advanced SQL Generation Options dialog box opens.

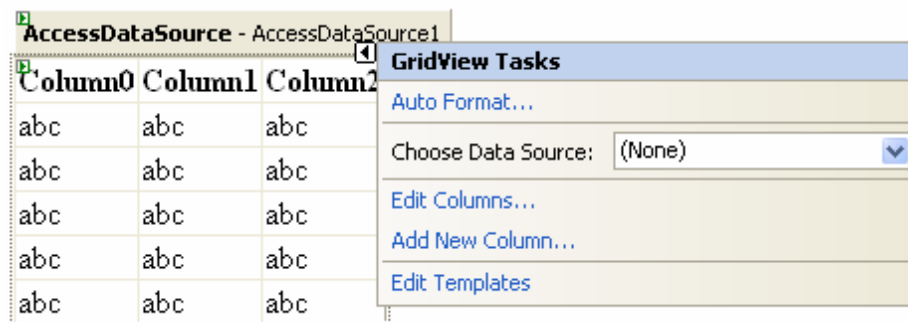


The image shows the 'Advanced SQL Generation Options' dialog box. The title bar is blue with a question mark and a close button. The main area has a yellow background. At the top, it says 'Additional INSERT, UPDATE, and DELETE statements can be generated to update the data source.' Below this are two checkboxes: 'Generate INSERT, UPDATE, and DELETE statements' (unchecked) and 'Use optimistic concurrency' (unchecked). The first checkbox has a description: 'Generates INSERT, UPDATE, and DELETE statements based on your SELECT statement. You must have all primary key fields selected for this option to be enabled.' The second checkbox has a description: 'Modifies UPDATE and DELETE statements to detect whether the database has changed since the record was loaded into the DataSet. This helps prevent concurrency conflicts.' At the bottom, there are two buttons: 'OK' and 'Cancel'.

9. Check the Generate INSERT, UPDATE, and DELETE Statements check box (shown in below); and then click OK to close this dialog box.



10. Back in the Configure Data Source Wizard, click Next to move to the Test Query page of the wizard.
11. In the Test Query page, you can test the page if you would like by clicking the Test Query button.
12. Click Finish. Now the data source has been configured.
13. Drop a GridView control on your form; expand its Smart Tag by clicking the little arrow in the upper-right corner, as shown in figure below.



14. Click the Choose Data Source drop-down list, and select AccessDataSource1, follow by checking the Enable Paging, Enable Paging, Sorting, Editing and Deleting options.

The screenshot shows the 'GridView Tasks' pane. At the top, there's a link 'Auto Format...'. Below it, the 'Choose Data Source:' dropdown is set to 'AccessDataSource1'. Underneath, there are links for 'Configure Data Source...', 'Refresh Schema', 'Edit Columns...', and 'Add New Column...'. A section contains five checkboxes: 'Enable Paging' (checked), 'Enable Sorting' (checked), 'Enable Editing' (checked), 'Enable Deleting' (checked), and 'Enable Selection' (unchecked). At the bottom, there's a link 'Edit Templates'.

15. Double click the any blank space on the form button and enter the following statements for the page load event.

```
protected void Page_Load(object sender, EventArgs e)
{
    // Return to Login page if the session variable not exist
    if (Session["UserType"] == null ||
        !Session["UserType"].Equals("admin"))
        Response.Redirect("Login.aspx");
}
```

16. Drag a LinkButton control on the page. Double click the LinkButton and enter the following statements.

```
protected void LinkButton1_Click(object sender, EventArgs e)
{
    // Cancel the current session
    Session.Abandon();
    Response.Redirect("AdminOnlyPage1.aspx");
}
```

17. Open this page in the Web browser it will redirect to Login Page. Only successful login can view the page.

