



Chapter No.9

LINQ to SQL

In

ASP.Net

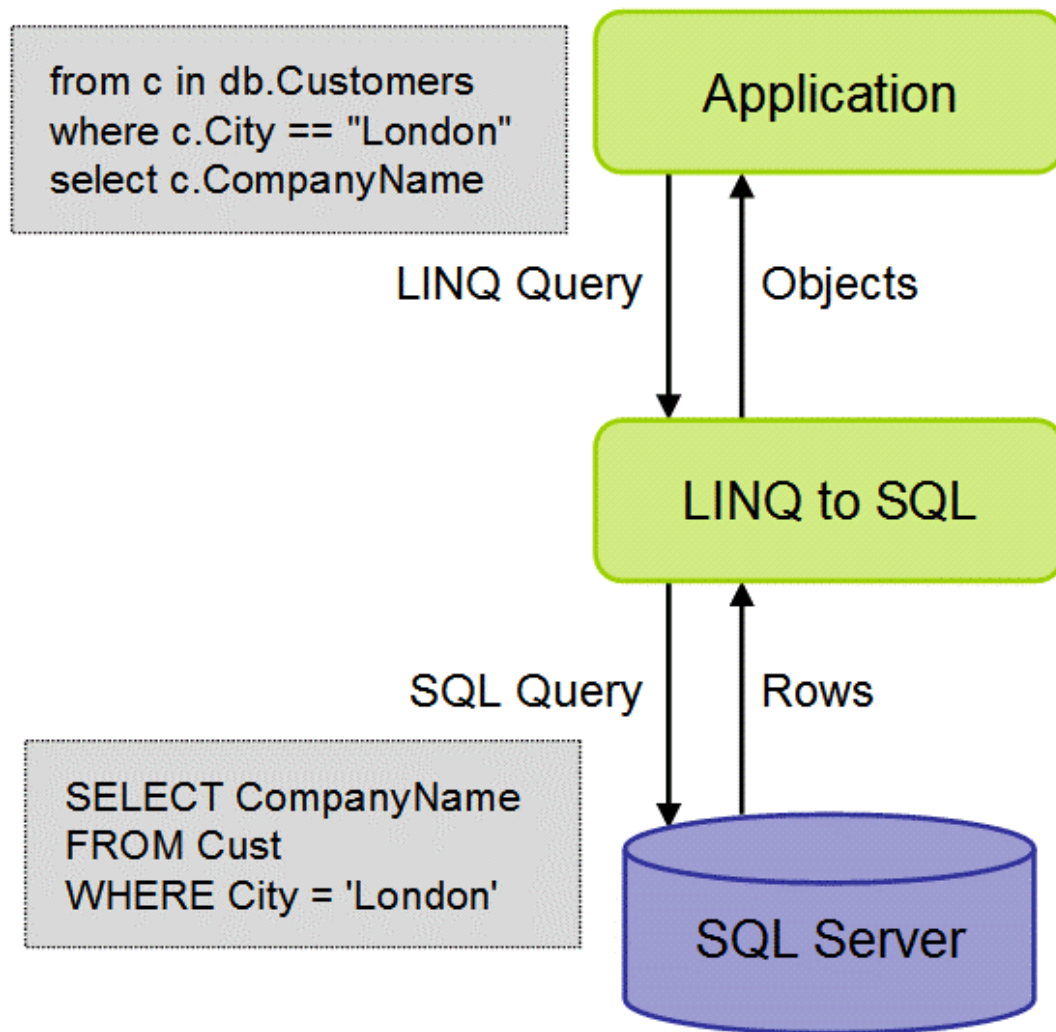
Today we will learn,

- ✓ What is LINQ
- ✓ LINQ to SQL
- ✓ Filling GridView Using LINQ To SQL
- ✓ Adding Data Using LINQ To SQL
- ✓ Updating Data Using LINQ To SQL
- ✓ Deleting Record using LINQ To SQL

9.1 What is LINQ To SQL

LINQ stands for 'Language Integrated Query', which replaces the traditional SQL query execution process. Moreover, it doesn't only applicable to manipulate database results, but it can also be used to manipulates array/list collections. LINQ was released as part of the .NET framework 3.0 and can be used from languages supported by .NET framework like C#, VB etc. The term 'LINQ To SQL refers to the technology by which we can use LINQ for access SQL Databases. In this chapter we will see LINQ to SQL step by step ways to get started with LINQ To SQL programming with C#.

LINQ to SQL, a component of Visual Studio Code Name "Orcas", provides a run-time infrastructure for managing relational data as objects without losing the ability to query. It does this by translating language-integrated queries into SQL for execution by the database, and then translating the tabular results back into objects defined. The application is then free to manipulate the objects while LINQ to SQL stays in the background tracking the changes automatically. It also allows to update, delete and insert data, but the only drawback from which it suffers is its limitation to the SQL server database. However, there are many benefits of LINQ to SQL over ADO.NET like reduced complexity, few lines of coding and many more. Below is a diagram showing the execution architecture of LINQ to SQL.

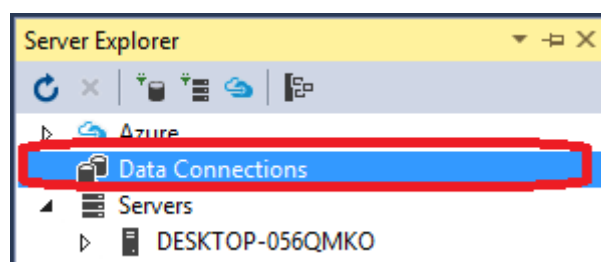


9.2 Filling GridView Using LINQ To SQL

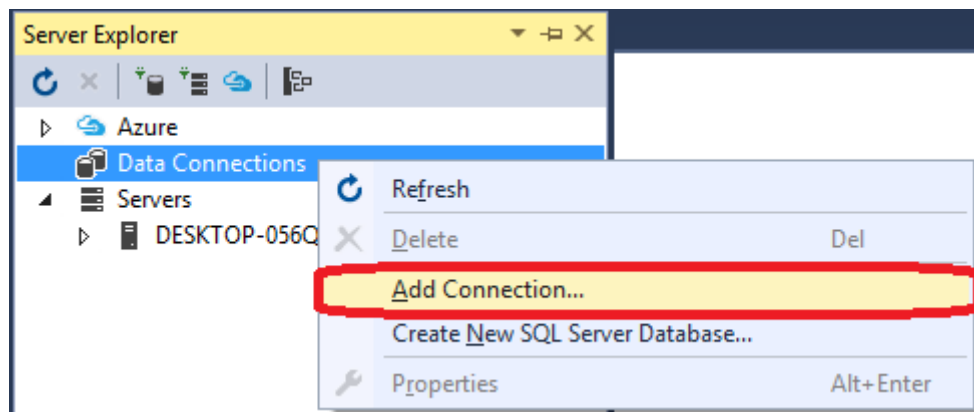
To use LINQ to SQL in ASP.NET, following steps are used.

- **Create a Data Connection in Server Explorer**

Go to View ➔ Server Explorer, this will open Server Explorer as shown



Right Click on the Data Connection ➔ Add Connection



Here, Enter Server Name and User Id and Password in case you have installed the SQL Server with SQL Server Authentication or Windows Authentication. Choose your Database. and press the OK button

Add Connection

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:
Microsoft SQL Server (SqlClient) Change...

Server name:
DESKTOP-056QMKO Refresh

Log on to the server

Authentication: SQL Server Authentication

User name: sa

Password: ••••••

☐ Save my password

Connect to a database

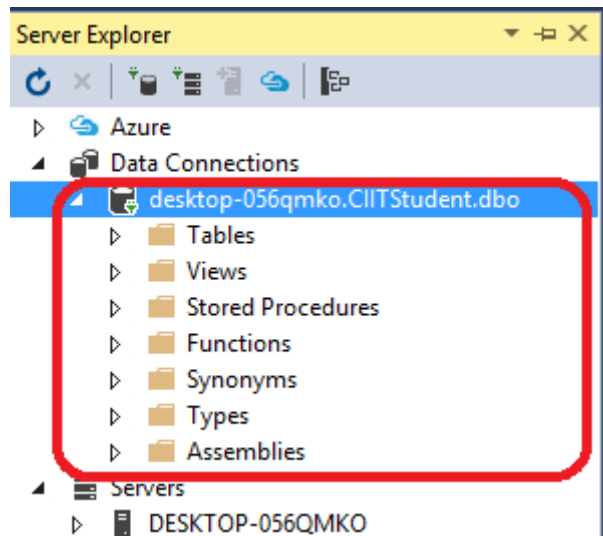
☒ Select or enter a database name:

☐ Advanced...

CIITStudent
master
model
msdb
MyDB
QuizSystem
tempdb

Test Connection OK Cancel

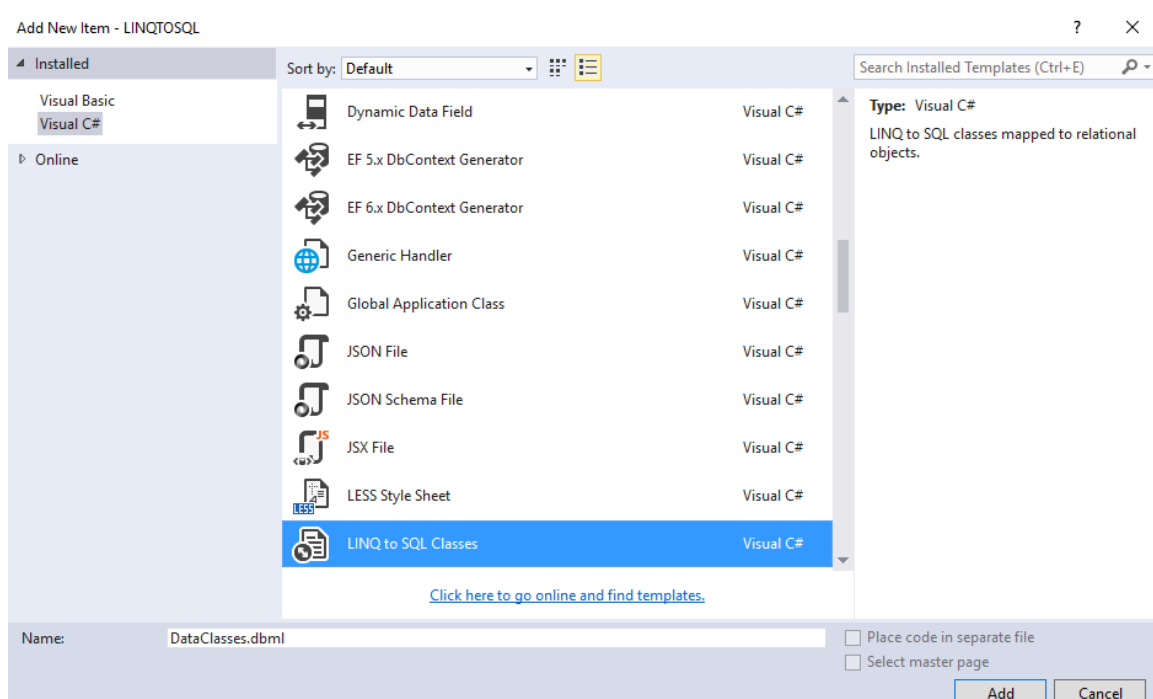
You will see that SQL Server connection has been added to your Server Explorer as shown



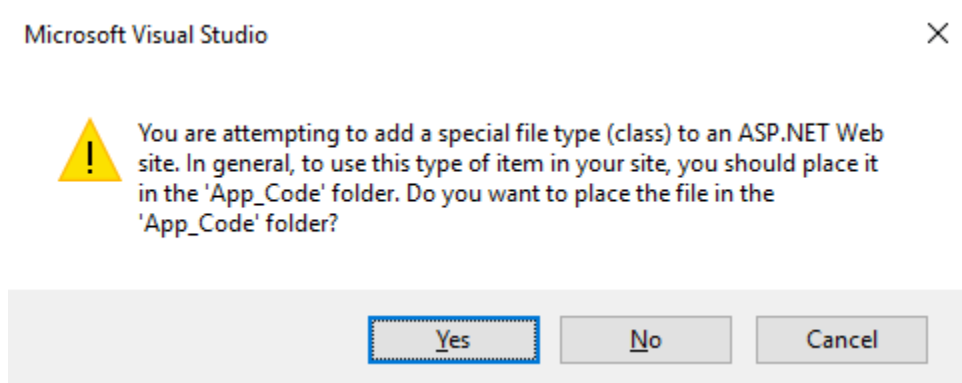
Next Step is to mapping SQL Server Database table to LINQ to SQL.

- **Mapping LINQ To SQL Class From SQL Server Database**

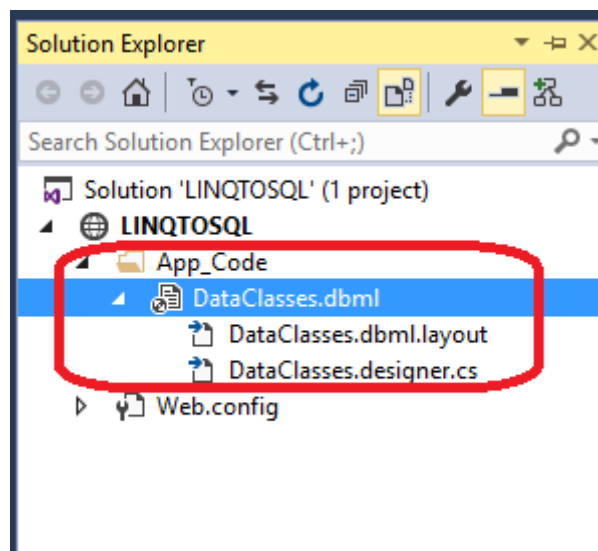
Create a New ASP.Net Web site. Right Click on the web site name and Choose Add ➔ Add New Item ➔ LINQ To SQL Class and press the **Add button.** this will add a .dbml file created along with designer interface, as shown



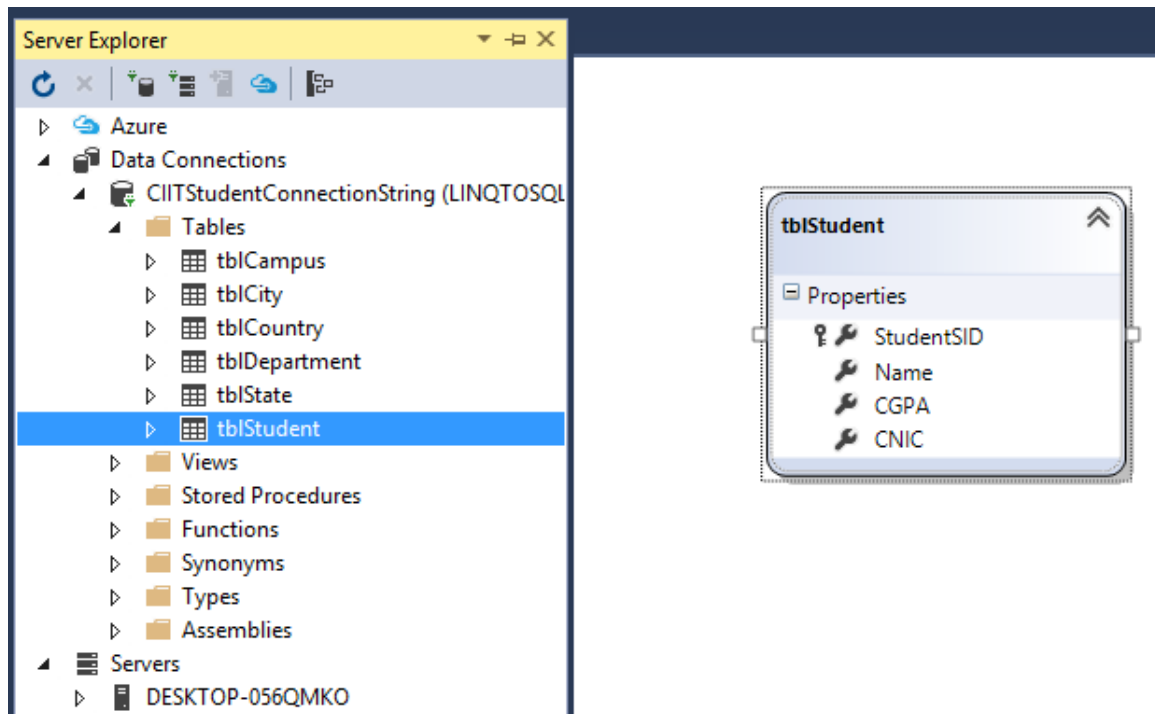
When you press the Add button, you will be asked to Would you like to add LINQ To SQL class to app_code folder, here press the **Yes button**



You will finally see that a class LINQ To SQL is added to your web site under the App_Code folder, as shown here



Next, drag the table from Server explorer to LINQ To SQL class designer interface. This is shown in the next screen shot



Save all the work done so far.

You can see the code generated by dragging and dropping the **tblStudent** from Server Explorer to DataClasses.dbml.layout in the file DataClasses.designer.cs file , as shown.

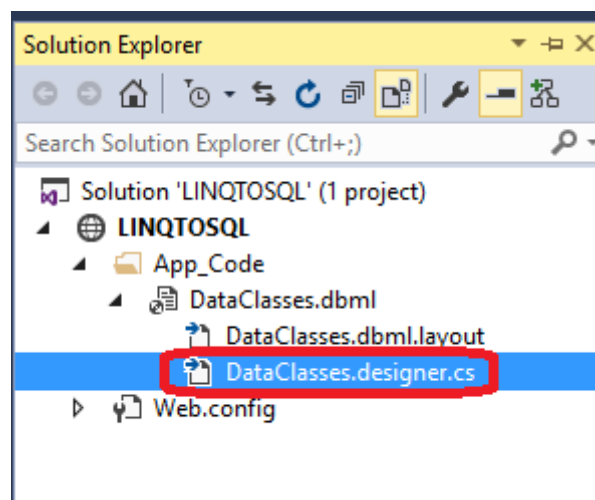


Table structure is shown here

DESKTOP-056QMKO.C...- dbo.tblStudent X			
	Column Name	Data Type	Allow Nulls
▶	StudentSID	int	<input type="checkbox"/>
	Name	varchar(50)	<input checked="" type="checkbox"/>
	CGPA	varchar(50)	<input checked="" type="checkbox"/>
	CNIC	varchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

The generated code snapshot from drag and dropping the tblStudent to LINQ To SQL designer are is

```

8  // the code is regenerated.
9  // </auto-generated>
10 //-----
11
12 *using ...
13
14
15 [global::System.Data.Linq.Mapping.DatabaseAttribute(Name="CIITStudent")]
16 public partial class DataClassesDataContext : System.Data.Linq.DataContext
17 {
18     private static System.Data.Linq.Mapping.MappingSource mappingSource = new AttributeMappingSource();
19
20     Extensibility Method Definitions
21 }

```

You can see all the other code by scrolling down the page. Upto this point we have done successfully. Next step is to add an ASP.NET web form and calling the method from the LINQ To SQL class and filling the GridView.

- **Using LINQ To SQL Class in an ASP.NET Web form**

- Add a new ASP.NET web form, rename it as LINQToSQLDemo.
- Drag and drop the GridView to this web form. Rename the GridView1 to gvStudents.
- Open the Page_Load() event of the LINQToSQLDemo web form as shown

```

8  public partial class LINQToSQLDemo : System.Web.UI.Page
9  {
10     protected void Page_Load(object sender, EventArgs e)
11     {
12
13     }
14 }

```

- Type the Code in the Red encircled Page_Load() event as shown here

```

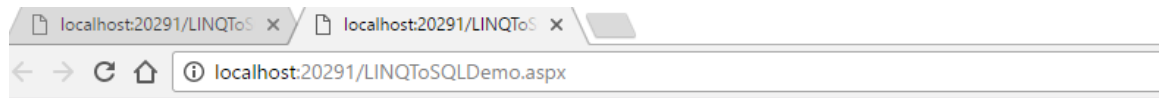
10 protected void Page_Load(object sender, EventArgs e)
11 {
12     DataClassesDataContext obj = new DataClassesDataContext();
13
14     var results = from std in obj.tblStudents
15                   select std;
16
17     gvStudents.DataSource = results;
18     gvStudents.DataBind();
19 }

```

The **DataContext** is the main conduit by which we retrieve objects from the database and resubmit changes. We use it in the same way that you would use an ADO.NET Connection. In fact, the **DataContext** is initialized with a connection or connection string you supply. The purpose of the **DataContext** is to translate your requests for objects into SQL queries made against the database and then assemble objects out of the results. The **DataContext** enables language-integrated query by implementing the same operator pattern as the standard query operators such as Where and Select.

The result is assigned to a variable of type '**var**'. This data type is also new from .NET framework 3.0 and used to represent data with **dynamic types**. That means, here any kind of data returned from the LINQ query will be assigned to that variable and you will have to just cast that to the proper data type.

- v. Compile, Save and Run the web site. You will see that GridView is filled with the data but this time the GridView is filled using LINQ to SQL class instead of Dataset or any other source.



:: LINQ Demo (Filling the Gridview) ::

StudentSID	Name	CGPA	CNIC
1	Asif	2.5	12101-5732440-9
2	Tariq	3	23456-5656447-0
3	Inam	4	34234-9089009-9
4	Sajida	3	23234-9087789-3
5	Saif Ur Rehman	3.8	12101-5732009-0
7	Muhammad Hamza	3.4	12345-90989-9

9.3 Adding Data Using LINQ To SQL

Next step is to add a new Record to Database using LINQ To SQL class. For this follow these steps

- Suppose we have the table structure as follow

DESKTOP-056QMKO.C...- dbo.tblStudent X			
	Column Name	Data Type	Allow Nulls
PK	StudentSID	int	<input type="checkbox"/>
	Name	varchar(50)	<input checked="" type="checkbox"/>
	CGPA	varchar(50)	<input checked="" type="checkbox"/>
	CNIC	varchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Where, **StudentSID** is the Primary Key and is auto generated and auto incremented.

- Design the form as shown here

:: LINQ Demo (Adding New Record) ::

Student Name	<input type="text"/>
CGPA	<input type="text"/>
CNIC #	<input type="text"/>
<input type="button" value="Save Record"/>	

- Rename TextBoxes to txtSName, txtCGPA and txtCNIC respectively.
- Double Click on the Save Record button to create the button Click event
- Add the following code in the Click event of the Save Record Button as shown

```

21     protected void btnSaveStudent_Click(object sender, EventArgs e)
22     {
23         DataClassesDataContext objDcon = new DataClassesDataContext();
24         tblStudent objStudent = new tblStudent();
25         objStudent.Name = txtSName.Text;
26         objStudent.CGPA = txtCGPA.Text;
27         objStudent.CNIC = txtCNIC.Text;
28
29         objDcon.tblStudents.InsertOnSubmit(objStudent);
30         objDcon.SubmitChanges();
31     }
32

```

Code Explanation

Line Number 23: Instantiate the LINQ To SQL class

Line Number 24: Instantiate the **tblStudent**, that we drag and drop on the designer of the LINQ To SQL surface

Line Number 25-27: Code is reading the values of the tblStudent table attributes

Line Number 29: Here, the object of the tblStudent database table is sent for saving to database by calling the method **InsertOnSubmit()** , this method takes entity type (i.e. table object as parameter). This method is called using the LINQ To SQL object and then the name of the database table.

Line Number 30: This line of code is used to commit the changes to database by calling the method **SubmitChanges()**. This method is important/mandatory to call to commit the changes to database.

- Compile, Save and browse the Web Site
- Enter the sample Student Record

:: LINQ Demo (Adding New Record) ::

Student Name	<input type="text" value="Tobba Rehman"/>
CGPA	<input type="text" value="4.0"/>
CNIC #	<input type="text" value="12101-5489009-0"/>
	<input type="button" value="Save Record"/>

Press the Save Button, Record will be Saved to Database successfully...

9.4 Deleting Record using LINQ To SQL

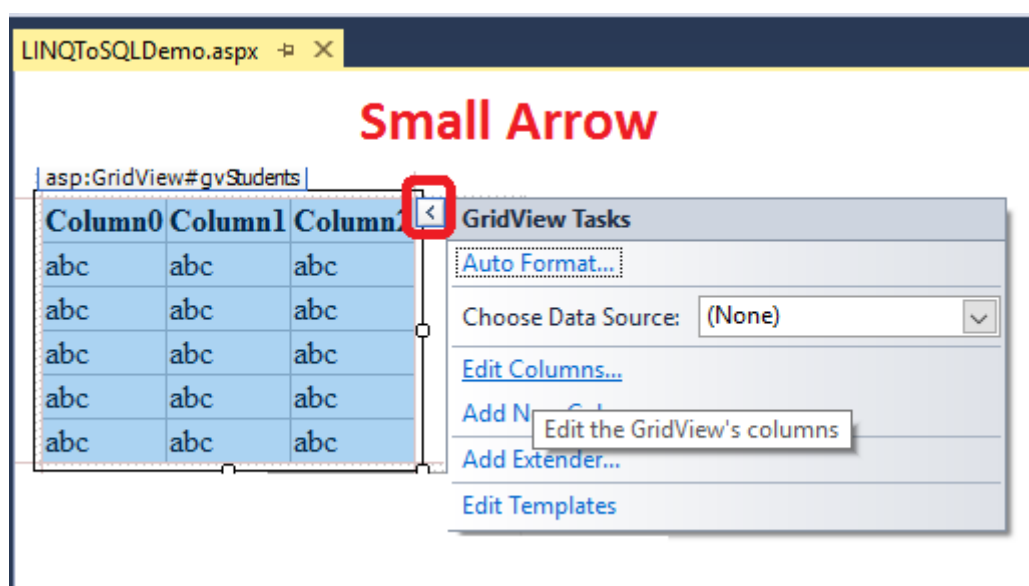
After adding a database record to table using LINQ To SQL, next step is to learn how to Delete an Existing Record from Database using LINQ To SQL class. For this follow these steps

- Suppose we have the table structure as follow

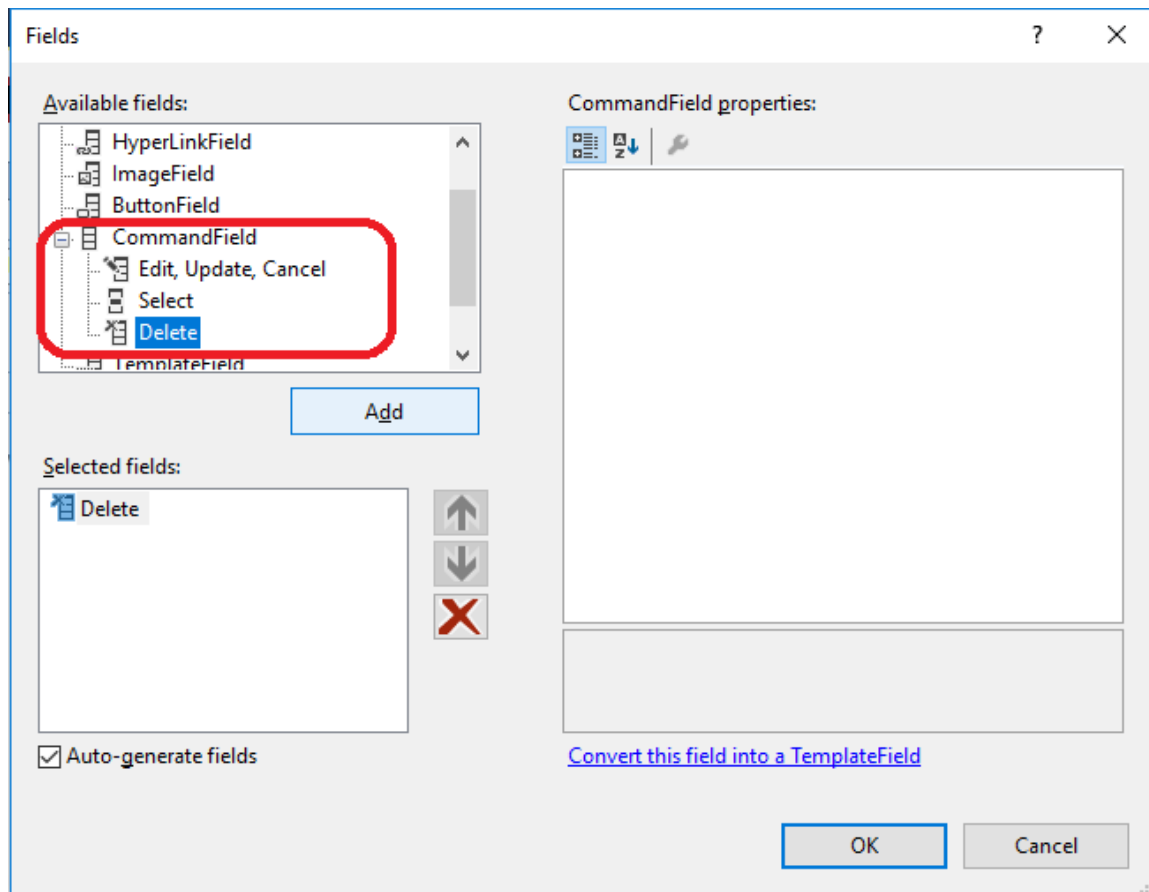
DESKTOP-056QMKO.C...- dbo.tblStudent			
	Column Name	Data Type	Allow Nulls
PK	StudentSID	int	<input type="checkbox"/>
	Name	varchar(50)	<input checked="" type="checkbox"/>
	CGPA	varchar(50)	<input checked="" type="checkbox"/>
	CNIC	varchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Where, **StudentSID** is the Primary Key and is auto generated and auto incremented.

- Add a new column to the GridView Control by Right Click on the small arrow on the Right Top Corner of the GridView as shown



Choose the Edit Column option. From the open Window, expand the CommandField from the **Available Fields**, and Select the Delete Option and press the **Add button**. This will add a Delete link to the **Selected Fields** as shown below. Press the OK button, you will see that a Delete Link is added in the GridView per Record. Save the File and Browse the Web Site.

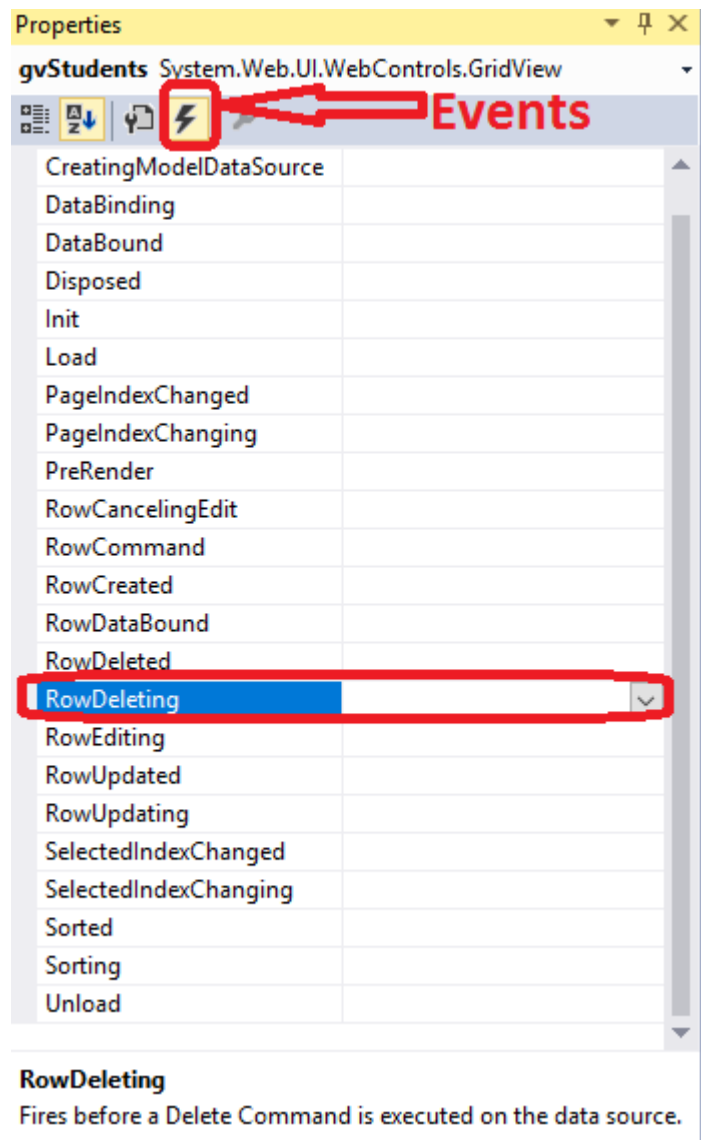


- When you browse the web site, you will see

:: LINQ Demo (Deleting Record From Database) ::

	StudentSID	Name	CGPA	CNIC
Delete	1	Asif	2.5	12101-5732440-9
Delete	2	Tariq	3	23456-5656447-0
Delete	3	Inam	4	34234-9089009-9
Delete	4	Sajida	3	23234-9087789-3
Delete	5	Saif Ur Rehman	3.8	12101-5732009-0
Delete	7	Muhammad Hamza	3.4	12345-90989-9
Delete	8	Tobba Rehman	4.0	12101-5489009-0

9.5 Now, we will add the Delete Link button code. Open the property windows of the GridView Control and Look for RowDeleting event as shown here,



Double click in the **RowDeleting** Event to generate the **gvStudent_RowDeleting()** event as shown here.

```
33 protected void gvStudent_RowDeleting(object sender, EventArgs e)
34 {
35
36 }
```

- Add the following code in this event

```

38     protected void gvStudents_RowDeleting(object sender,
39                                           GridViewDeleteEventArgs e)
40     {
41         DataClassesDataContext objcon = new DataClassesDataContext();
42         int id = Convert.ToInt16(gvStudents.Rows[e.RowIndex].Cells[1].Text);
43         var LinqDelete = from students in objcon.tblStudents
44                           where students.StudentSID == id
45                           select students;
46         foreach (var Student in LinqDelete)
47         {
48             objcon.tblStudents.DeleteOnSubmit(Student);
49         }
50         objcon.SubmitChanges();
51     }

```

Code Explanation

Line Number 41: Instantiate the LINQ To SQL class

Line Number 42: Retrieves the StudentSID value from the GridView Row which is being clicked by the user. When you browse the web site, you can see that StudentSID is displayed in the Second Cell of the GridView, which has index 1

Line Number 43-45: LINQ To SQL query with Where Clause to select the Record to be deleted.

Line Number 46-49: Here, Record of each student is deleted from tblStudent database by calling the method **DeleteOnSubmit()**, this method takes entity type (i.e. table object as parameter). This method is called using the LINQ To SQL object and then the name of the database table.

Line Number 50: This line of code is used to commit the changes to database by calling the method **SubmitChanges()**. This method is important/mandatory to call to commit the changes to database.

- Compile, Save and browse the Web Site

- Click on any Delete Link in the any GridView record. As User Clicks on any Link then the record will be deleted.

9.6 Updating Data Using LINQ To SQL

Next Task is to update a database record using LINQ To SQL. For this follow these steps

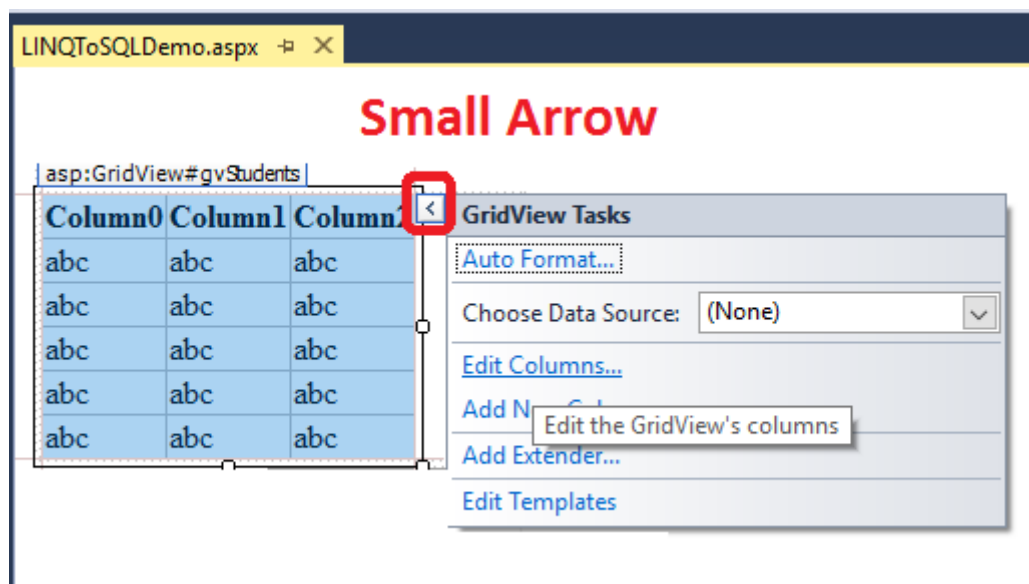
- Suppose we have the table structure as follow

DESKTOP-056QMKO.C...- dbo.tblStudent X			
	Column Name	Data Type	Allow Nulls
PK	StudentSID	int	<input type="checkbox"/>
	Name	varchar(50)	<input checked="" type="checkbox"/>
	CGPA	varchar(50)	<input checked="" type="checkbox"/>
	CNIC	varchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

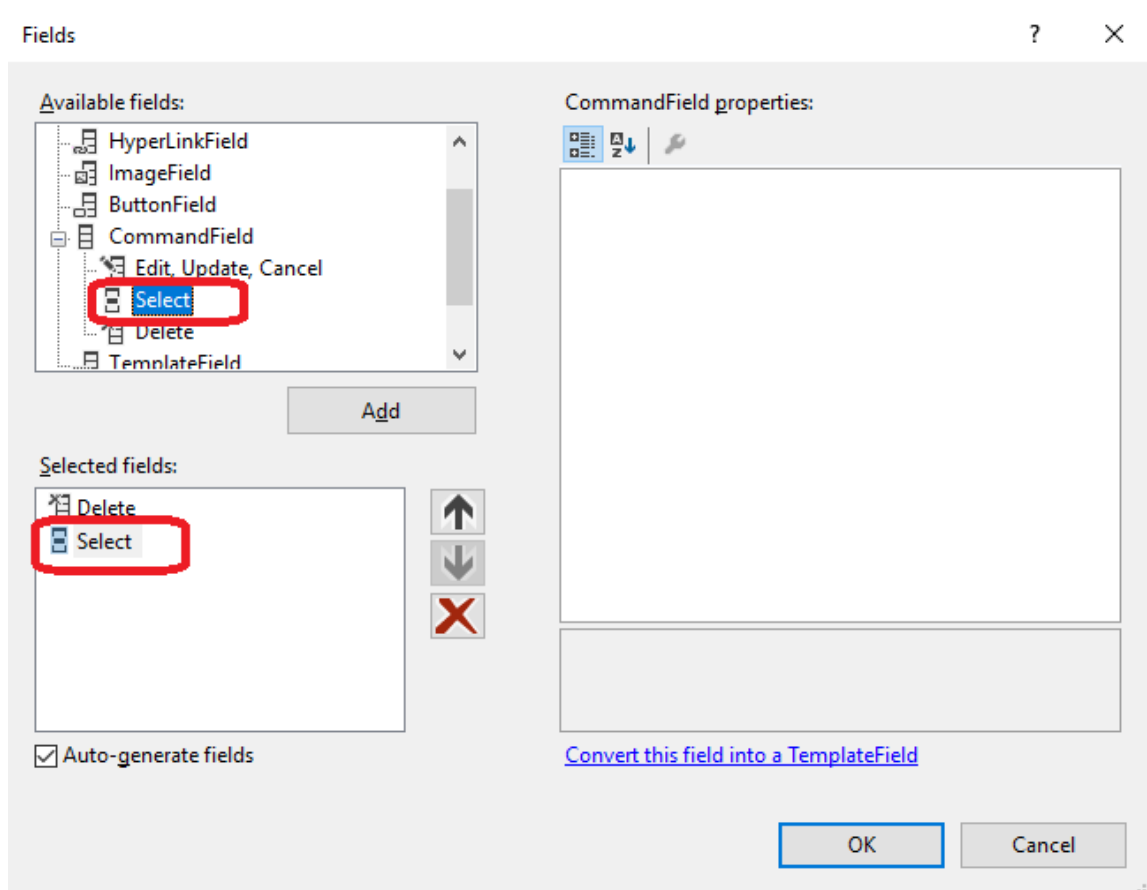
Where, **StudentSID** is the Primary Key and is auto generated and auto incremented.

- Add a new column to the GridView Control by Right Click on the small arrow on the

Right Top Corner of the GridView as shown



Choose the Edit Column option. From the open Window, expand the CommandField from the **Available Fields**, and Select the **Select Button** Option and press the **Add button**. This will add a **Select** link to the **Selected Fields** as shown below.



Press the OK button, you will see that a **Select Link** is added in the GridView per Record.
Save the File and Browse the Web Site.

Congratulations! You have successfully Bind the FormView

Control to Database Table

These ASP.Net Tutorials are prepared only for the purpose of learning ASP.Net with C#.

Anyone can copy, print or reuse it in any format

Saif Ur Rehman Saifi

Assistant Professor, UIIT

PMAS Arid Agriculture University, Rawalpindi

If you found this tutorial helpful or you want to give us any valuable suggestions contact at

www.ASPUIIT.blogspot.com