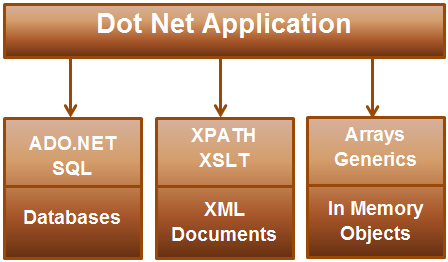
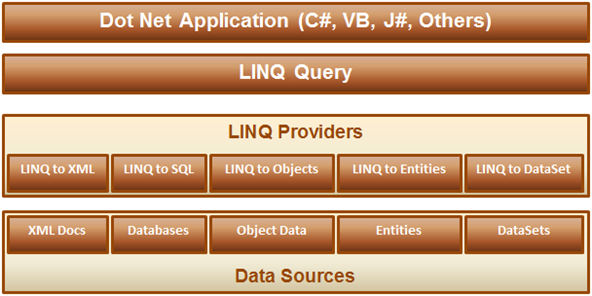
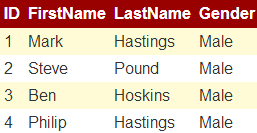
**In this video, we will discuss**  
**1.** What is LINQ  
**2.** Why should we use LINQ and what are the benefits of using LINQ  
**3.** LINQ Architecture  
**4.** LINQ Providers  
**5.** LINQ to SQL Example and LINQ to Objects example   
  
   
  
**What is LINQ**  
**LINQ**stands for **Language Integrated Query**. LINQ enables us to query any type of data store (SQL Server, XML documents, Objects in memory etc).   
  
   
  
**Why should we use LINQ and what are the benefits of using LINQ**   
   
  
**If the .NET application that is being developed**  
**a) Requires data from SQL Server -**Then the developer has to understand ADO.NET code and SQL specific to SQL Server Database  
**b) Requires data from an XML document -**Then the developer has to understand XSLT & XPATH queries  
  
**c) Need to query objects in memory (List<Customer>, List<Order> etc) -**Then the developer has to understand how to work with objects in memory  
  
LINQ enables us to work with these different data sources using a similar coding style without having the need to know the syntax specific to the data source. In our upcoming videos we will discuss querying different data sources using LINQ.  
  
Another benefit of using LINQ is that it provides intellisense and compile time error checking.   
  
**LINQ Architecture & LINQ Providers**   
  
  
**1.** LINQ query can be written using any .NET supported programming language  
**2.** LINQ provider is a component between the LINQ query and the actual data source, which converts the LINQ query into a format that the underlying data source can understand. For example LINQ to SQL provider converts a LINQ query to T-SQL that SQL Server database can understand.  
  
**For example**, the application that we are developing should display male students in a GridView control as shown below.    
   
  
**To achieve this**  
**Step 1:**We first create the required table

Create Table Students

(

     ID int primary key identity,

     FirstName nvarchar(50),

     LastName nvarchar(50),

     Gender nvarchar(50)

)

GO

Insert into Students values ('Mark', 'Hastings', 'Male')

Insert into Students values ('Steve', 'Pound', 'Male')

Insert into Students values ('Ben', 'Hoskins', 'Male')

Insert into Students values ('Philip', 'Hastings', 'Male')

Insert into Students values ('Mary', 'Lambeth', 'Female')

GO

**Step 2:** Write the required ADO.NET code to retrieve data from SQL Server database as shown below.

using System;

using System.Collections.Generic;

using System.Configuration;

using System.Data.SqlClient;

namespace Demo

{

    public partial class WebForm1 : System.Web.UI.Page

    {

        protected void Page\_Load(object sender, EventArgs e)

        {

            string cs = ConfigurationManager.ConnectionStrings["DBCS"].ConnectionString;

            SqlConnection con = new SqlConnection(cs);

            SqlCommand cmd = new SqlCommand

                ("Select ID, FirstName, LastName, Gender from Students where Gender='Male'", con);

            List<Student> listStudents = new List<Student>();

            con.Open();

            SqlDataReader rdr = cmd.ExecuteReader();

            while (rdr.Read())

            {

                Student student = new Student();

                student.ID = Convert.ToInt32(rdr["ID"]);

                student.FirstName = rdr["FirstName"].ToString();

                student.LastName = rdr["LastName"].ToString();

                student.Gender = rdr["Gender"].ToString();

                listStudents.Add(student);

            }

            con.Close();

            GridView1.DataSource = listStudents;

            GridView1.DataBind();

        }

    }

    public class Student

    {

        public int ID { get; set; }

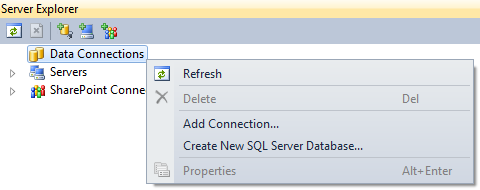
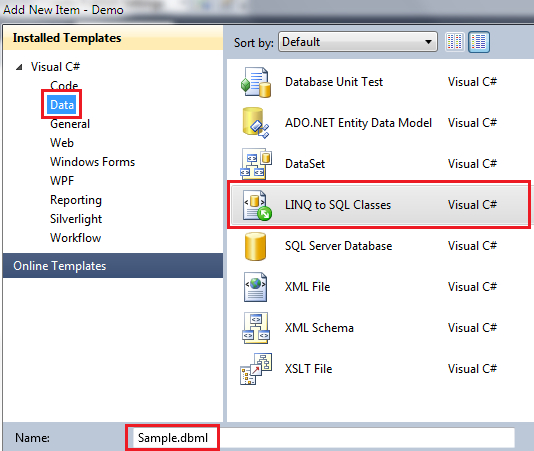
        public string FirstName { get; set; }

        public string LastName { get; set; }

        public string Gender { get; set; }

    }

}

If we misspell table or column names in the SQL Query, we will not know about it at compile time. At run time the page crashes and that's when we will know about this error. Also notice that there is no **intellisense**when typing table and column names. Misspelled column names when reading from the reader will also cause the same problem. With LINQ we will have intellisense and compile time error checking.  
  
**Now let's achieve the same thing using LINQ to SQL.**  
**Step 1:** Create a new empty asp.net web application and name it **Demo**  
  
**Step 2:**Click on **"View"**menu item and select **"Server Explorer"**  
  
**Step 3:**In **"Server Explorer"**window, right click on **"Data Connections"**and select **"Add Connection"**option   
   
  
**Step 4:** Specify your SQL Server name and the credentials to connect to SQL Server. At this point we should be connected to SQL Server from Visual Studio.  
  
**Step 5:** Adding **LINQ to SQL Classes**  
**a)** Right click on the **"Demo"**project in solution explorer and select **"Add New Item"**option  
**b)** In the **"Add New Item"**dialog box, select **"Data"**under **"Installed Templates"**  
**c)** Select **"LINQ to SQL Classes"**  
**d)** Set **Name = Sample.dbml**  
**e)** Finally click **"Add"**button   
   
  
**Step 6:** From **"Server Explorer"**window drag and drop **"Students"**table onto **"Sample.dbml"**designer file.  
  
**Step 7:**Add a webform. Drag and drop a gridview control.  
  
**Step 8:**Copy and paste the following code in the code-behind file

using System;

using System.Linq;

namespace Demo

{

    public partial class WebForm1 : System.Web.UI.Page

    {

        protected void Page\_Load(object sender, EventArgs e)

        {

            SampleDataContext dataContext = new SampleDataContext();

            GridView1.DataSource = from student in dataContext.Students

                                   where student.Gender == "Male"

                                   select student;

            GridView1.DataBind();

        }

    }

}

Notice that, with LINQ we are getting intellisense. If we misspell the table or column names we will get to know about them at compile time. Open SQL Profiler. Run the application, and notice the SQL Query that is generated.