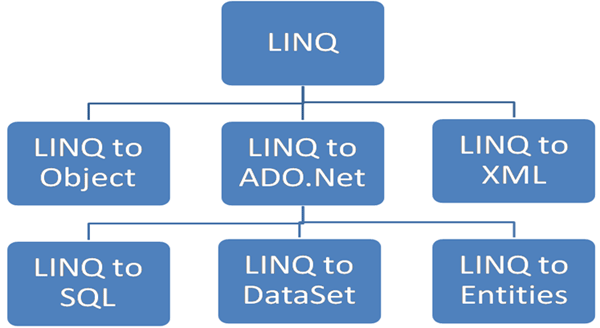
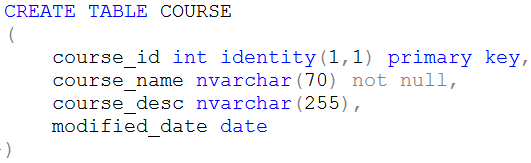
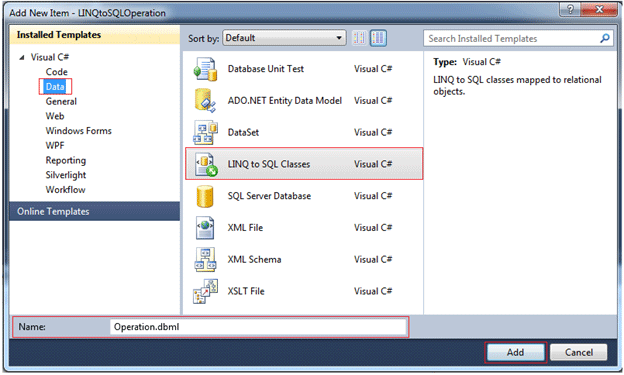
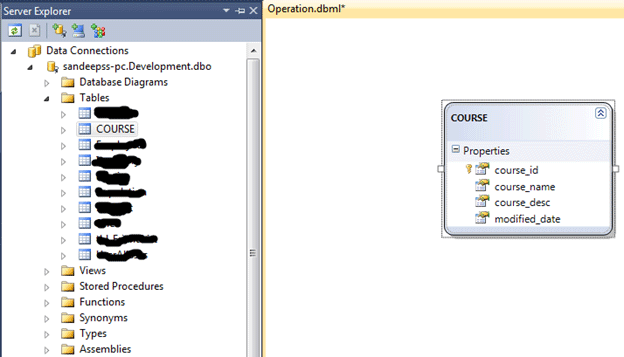
Language-INtegrated Query (LINQ) is a Microsoft .NET Framework component that adds native data querying capabilities to .NET languages. In other words LINQ has the power of querying on any source of data (Collection of objects, database tables or XML Files). We can easily retrieve data from any object that implements the IEnumerable<T> interface and any provider that implements the IQueryable<T> interface.  
  
Microsoft basically divides LINQ into the following three areas:

* LINQ to Object : Queries performed against in-memory data
* LINQ to ADO.Net
  + LINQ to SQL (formerly DLinq) : Queries performed against the relation database; only Microsoft SQL Server is supported.
  + LINQ to DataSet : Supports queries by using ADO.NET data sets and data tables.
  + LINQ to Entities : Microsoft ORM solution
* LINQ to XML (formerly XLinq) : Queries performed against the XML source.

**  
LINQ to SQL**  
LINQ to SQL translates our actions to SQL and submits the changes to the database. Here we will perform Select, Insert, Update and Delete operations on a COURSE table.  
  
**Step 1:** Create a COURSE Table in the database  
  
  
  
**Step 2:** Create a ContextData file using the Object Relational Designer:  
  
Create a new item, select the LINQ to SQL classes (as shown in the following figure) and name it Operation.dbml.  
  
  
  
After clicking the Add button the ContextData file is created. Now we should drag all the tables onto the left-hand side of the designer and save (as shown in the following figure). This will create all the mappings and settings for each table and their entities.  
  
  
  
For .dbml files the database connection string is defined in the web.config file as:  
  
<connectionStrings>  
<add name="DevelopmentConnectionString" connectionString="Data Source=sandeepss-PC;Initial Catalog=Development;User ID=sa;  
Password=\*\*\*\*\*\*\*" providerName="System.Data.SqlClient" />  
 </connectionStrings>  
  
We can use a connection string from the web.config file or we can pass a connection string as a parameter in the constructor of the DataContext class to create an object of the DataContext class.  
  
**The SELECT Operation**private void GetCourses()  
{  
      //create DataContext object  
      OperationDataContext OdContext = new OperationDataContext();  
      var courseTable = from course in OdContext.GetTable<COURSE>() select course;  
      //grdCourse is gridview id  
      grdCourse.DataSource = courseTable;  
      grdCourse.DataBind();  
}  
  
**The INSERT Operation**  
private void AddNewCourse()  
{  
      //Data maping object to our database  
      OperationDataContext OdContext = new OperationDataContext();  
      COURSE objCourse = new COURSE();  
      objCourse.course\_name = "B.Tech";  
      objCourse.course\_desc = "Bachelor Of Technology";  
      objCourse.modified\_date = DateTime.Now;  
      //Adds an entity in a pending insert state to this System.Data.Linq.Table<TEntity>and parameter is the entity which to be added  
      OdContext.COURSEs.InsertOnSubmit(objCourse);  
      // executes the appropriate commands to implement the changes to the database  
      OdContext.SubmitChanges();  
}  
  
**The Update Operation**  
  
private void UpdateCourse()  
{  
      OperationDataContext OdContext = new OperationDataContext();  
      //Get Single course which need to update  
      COURSE objCourse = OdContext.COURSEs.Single(course => course.course\_name == "B.Tech");  
      //Field which will be update  
      objCourse.course\_desc = "Bachelor of Technology";  
      // executes the appropriate commands to implement the changes to the database  
      OdContext.SubmitChanges();  
 }  
  
**The DELETE Operation**  
  
private void DeleteCourse()  
{  
      OperationDataContext OdContext = new OperationDataContext();  
      //Get Single course which need to Delete  
      COURSE objCourse = OdContext.COURSEs.Single(course => course.course\_name == "B.Tech");  
      //Puts an entity from this table into a pending delete state and parameter is the entity which to be deleted.  
      OdContext.COURSEs.DeleteOnSubmit(objCourse);  
      // executes the appropriate commands to implement the changes to the database  
      OdContext.SubmitChanges();  
}  
  
**Conculsion**