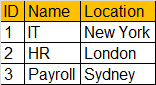
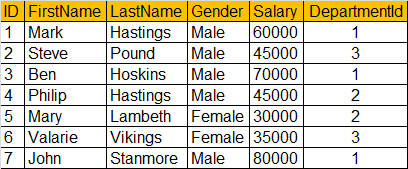
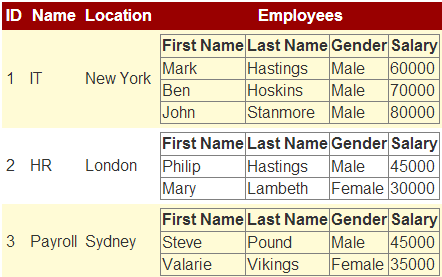
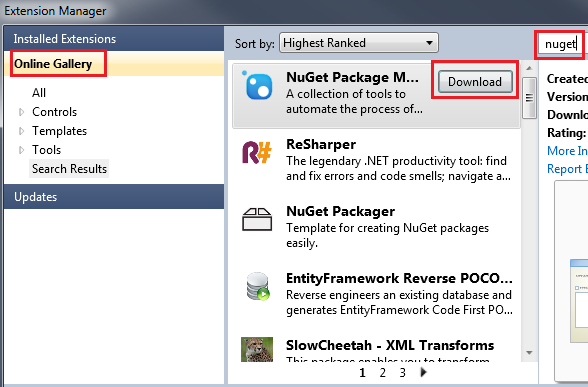
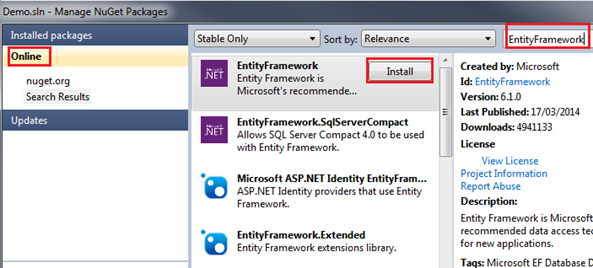
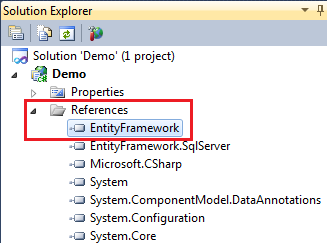
**What is Entity Framework**  
Entity Framework is an ORM framework. ORM stands for Object Relational Mapping.  
  
**What is Object Relational Mapping framework**  
Object Relational Mapping framework automatically creates classes based on database tables, and the vice versa is also true, that is, it can also automatically generate necessary SQL to create database tables based on classes.   
  
  
  
**Let's understand what entity framework can provide with an example**. Assume we have the following 2 tables   
  
  
  
  
We want to display the above data from both the tables in a webform as shown below.   
  
  
**To achieve this**  
1. We need to create Department and Employee classes  
2. Write ADO.NET code to retrieve data from the database  
3. Once the data is retrieved we need to create Department and Employee objects and populate them with data.  
  
Entity Framework can do all of the above automatically, if we provide it with the database schema.  
  
**Installing NuGet Package Manager**  
1. From Visual Studio 2010 **Tools** menu, select **Extension Manager**  
2. Click on **Online Gallery** in the **Extension Manager** window  
3. Search for **NuGet**  
4. Finally **Download** NuGet Package Manager and install   
  
  
**Please note:** You must restart visual studio for the changes to take effect.  
  
**Step 1:** Create a new **"Empty ASP.NET Web Application"** with name=**Demo.**  
  
**Step 2:** Installing Entity Framework  
**a)** Click on **Tools - NuGet Package Manager - Manage NuGet Packages for solution**  
**b)** Click on **"Online"** tab in **"Manage NuGet Packages"** window  
**c)** Type **"EntityFramework"** in the **search textbox** on the top right hand corner  
**d)** Finally click on the **"Install"** button.   
  
  
At this point **Entity Framework version 6.1** is installed and a reference to EntityFramework assembly is also automatically added.   
  
  
**Step 3:** Create **"Departments"** and **"Employees"** tables.

Create table Departments

(

     ID int primary key identity,

     Name nvarchar(50),

     Location nvarchar(50)

)

Create table Employees

(

     ID int primary key identity,

     FirstName nvarchar(50),

     LastName nvarchar(50),

     Gender nvarchar(50),

     Salary int,

     DepartmentId int foreign key references Departments(Id)

)

**Step 4:** Populate the tables created in Step 3, with data

Insert into Departments values ('IT', 'New York')

Insert into Departments values ('HR', 'London')

Insert into Departments values ('Payroll', 'Sydney')

Insert into Employees values ('Mark', 'Hastings', 'Male', 60000, 1)

Insert into Employees values ('Steve', 'Pound', 'Male', 45000, 3)

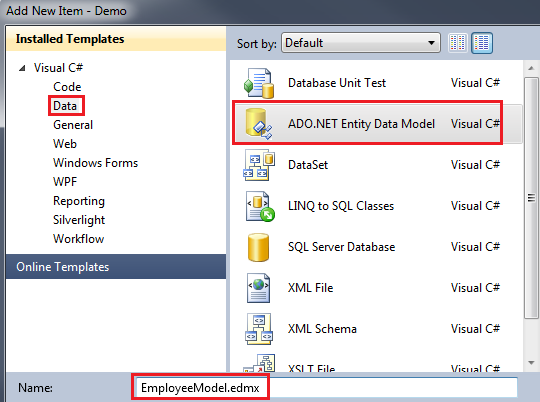
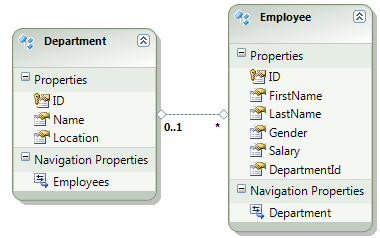
Insert into Employees values ('Ben', 'Hoskins', 'Male', 70000, 1)

Insert into Employees values ('Philip', 'Hastings', 'Male', 45000, 2)

Insert into Employees values ('Mary', 'Lambeth', 'Female', 30000, 2)

Insert into Employees values ('Valarie', 'Vikings', 'Female', 35000, 3)

Insert into Employees values ('John', 'Stanmore', 'Male', 80000, 1)

**Step 5:** Right click on the project in solution explorer and add **ADO.NET Entity Data Model**. Change the name from **Model1.edmx** to **EmployeeModel.edmx**   
  
  
**Step 6:** Select **"Generate from database"** and click **"Next"**  
  
**Step 7:** Choose Your Data Connection  
**a)** Click on **"New Connection"** button  
**b)** Select **"Microsoft SQL Server"** as Data source, and "**.Net Framework Data Provider for SQL Server"** option from **"Data provider"** dropdownlist. Click **Continue.**  
**c)** On **"Connection Properties"** screen, specify SQL Server Name. If you are using local installation of SQL Server then use (local) or . in the **"server name"** dropdownlist.  
**d)** Specify the **Authentication** you want to use.  
**e)** Select the database from **"Select or enter database name"** dropdownlist.  
**f)** Finally **"Test connection"** and click **"OK"**  
**g)** At this point we should be back on **"Choose Your Data Connection"** window. Make sure **"Save entity connection settings in Web.Config as"** checkbox is selected and change the name of the connection string to **"EmployeeDBContext"** and then Click **"Next"**  
  
**Step 8:** On **"Choose Your Database Objects"** screen, select **"Departments"** and **"Employees"** tables. Change the Model Namespace to **"EmployeeModel"** and click **"Finish".** At this point you should have **EmployeeModel.edmx** created.   
  
  
**EmployeeModel.Designer.cs** file is also generated. This file contains Employee and Department classes. **Tables** are mapped to **classes** and **columns** are mapped to **class properties.**  
  
**Step 9:** Add a webform. Drag and drop a **GridView** and an **EntityDataSource** control on the webform.  
  
**Step 10:** Build the solution. Flip the **WebForm1.aspx** to design mode.   
**a)** Right click on **EntityDataSource** control and select **"Show smart tag"** option from the context menu.  
**b)** Click on **"Configure Data Source"** link  
**c)** Select **"Named Connection"** radiobutton and select **"EmployeeDBContext"** from the dropdownlist.  
**d)** Select **"EmployeeDBContext"** option from **"DefaultContainerName"** dropdownlist and click **"Next"**  
**e)** On **"Configure Data Selection"** screen, select **"Departments"** from **"EntitySetName"** dropdownlist and click **"Finish"**  
**f)** Right click on **"GridView1"** control and select **"Show smart tag"** option.  
**g)** Click on **"Auto Format"** link and select **"Colorful"** option from **"AutoFormat"** window and click **"OK"**.  
**h)** Select **"EntityDataSource1"** from **"Choose Data Source"** dropdownlist.  
**I)** Click on **"Eidt Columns"** link and add a **"Template Field".** Set **HeaderText=Employees** and click **OK**.  
**j)** Now click **"Edit Templates"** link.  
**k)** Drag and drop a GridView control  
**l)** Select **"Edit DataBindings"** link   
**m)** Select **"Custom binding"** radiobutton and type **Eval("Employees")** in **"Code expression"** textbox and click **OK**.  
**n)** Select **"End Template Editing"** option from **"GridView1"** smart tasks pane.  
  
**Step 11:** Right click on **"EntityDataSource1"** control and select **"Properties".** In properties window set **Include=Employees**  
  
Run the web application and notice that **Departments and Employees** are displayed as expected. We have achieved all this without writing a single line of code.  
  
In this demo, we have used **schema first approach** of entity framework. We can also use **Model First** or **Code First approaches.** We will discuss 