**Code-first approach allows us to create our custom classes first and based on those custom classes entity framework can generate database automatically for us.** Let's understand this with an example. We will be modifying the example we worked with in  
**Step 1:** Delete **EmployeeModel.edmx** && **EmployeeModel.edmx.sql** files from the solution explorer.  
  
**Step 2:** Add a class file to the project. Name it **Employee.cs.** Copy and paste the following code.

publicclassEmployee

**{**

// Scalar Properties

publicintId **{** get**;** set**; }**

publicstringFirstName **{** get**;** set**; }**

publicstringLastName **{** get**;** set**; }**

publicstringGender **{** get**;** set**; }**

publicintSalary **{** get**;** set**; }**

// Navigation Property

publicDepartmentDepartment **{** get**;** set**; }**

**}**

**Step 3:** Add a class file to the project. Name it **Department.cs.** Copy and paste the following code.

publicclassDepartment

**{**

// Scalar Properties

publicintId **{** get**;** set**; }**

publicstringName **{** get**;** set**; }**

publicstringLocation **{** get**;** set**; }**

// Navigation Property

publicList<Employee>Employees **{** get**;** set**; }**

**}**

**Step 4:** Add a class file to the project. Name it **EmployeeDBContext.cs**. Copy and paste the following code.

// EmployeeDBContext class must inherit from DbContext

// present in System.Data.Entity namespace

publicclassEmployeeDBContext **:** DbContext

**{**

publicDbSet<Department>Departments **{** get**;** set**; }**

publicDbSet<Employee>Employees **{** get**;** set**; }**

**}**

**Step 5:** Add a class file to the project. Name it **EmployeeRepository.cs**. Copy and paste the following code.

publicclassEmployeeRepository

**{**

publicList<Department>GetDepartments**()**

**{**

EmployeeDBContextemployeeDBContext=newEmployeeDBContext**();**

returnemployeeDBContext.Departments.Include**(**"Employees"**)**.ToList**();**

**}**

**}**

**Step 6:** Add the database connection string in **web.config** file.

<connectionStrings>

  <add name="EmployeeDBContext"

       connectionString="server=.; database=Sample; integrated security=true;"

       providerName="System.Data.SqlClient"/>

</connectionStrings>

**Please Note:** If **ProviderName** is not specified the following runtime error will be thrown.  
The connection string 'EmployeeDBContext' in the application's configuration file does not contain the required providerName attribute."  
  
**Step 7:** Configure **Object Data Source** control  
**a)** Delete **EntityDataSource** control, that is already there in **WebForm1.aspx**.   
**b)** Drag and Drop **ObjectDataSource** control.  
**c)** Right click on **ObjectDataSource** control and select **"Show Smart Tag"** option from the context menu  
**d)** Click on **"Configure Data Source..."** link  
**e)** On **"Choose a Business Object"** screen, select **"EmployeeRepository"** and click **"Next"**  
**f)** On **"Define Data Methods"** screen, select **GetDepartments()** method and click **"Finish"**  
  
**Step 8:** Configure GridView control  
**a)** Right click on **GridView** control and select **"Show Smart Tag"** option from the context menu  
**b)** Select **"ObjectDataSource1"** from **"Choose Data Source"** dropdownlist  
**c)** Click **"No"** to **"Refresh Fields and Keys for GridView1"** when prompted  
  
**Step 9:** Rebuild the solution.  
  
**Step 10:** Delete the already existing database from SQL Server Management Studio.  
  
**Step 11:** Run the application by pressing CTRL + F5. Notice that we don't have any data displayed on WebForm1. This is because we don't have any data in the Departments and Employees tables. At this point we have the following created automatically.  
**a)** Sample database  
**b)** Departments table  
**c)** Employees table   
  
**Step 12:** Use the SQL script to populate the tables 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 13:** Refresh the Web Form and we should see the data we expect.