

LINQ and Entity Framework Lab Book

Table of Contents

Getting Started.....	3
Overview	3
Setup Checklist for LINQ and EF.....	3
Instructions.....	3
Learning More (Bibliography if applicable).....	4
Problem Statement/ Case Study (If applicable)	4
Lab 1.LINQ Baics.....	5
Lab 2.Creating Entity Data Model	9
Lab 3.Creating Entity Data Model using Model First Approach	11
Lab 4. Basic Query Operation using LINQ to Entities.....	40

Getting Started

Overview

This lab book is a guided tour for learning HTML version x.x. It comprises solved examples and 'To Do' assignments. Follow the steps provided in the solved examples and work out the 'To Do' assignments given.

Setup Checklist for HTML

Here is what is expected on your machine in order for the lab to work.

Minimum System Requirements

- Intel Core i3 or higher
- Microsoft Windows 7.
- Memory: 1GB of RAM (2GB or more recommended)
- Internet Explorer 10.0 or higher
- Google Chrome
- Connectivity to Sql Server
- LocalDB

Please ensure that the following is done:

- Visual Studio 2012 or above.
- .Net Framework 4.5 or above.

Instructions

- For all coding standards refer Appendix A. All lab assignments should refer coding standards.
- Create a directory by your name in drive <drive>. In this directory, create a subdirectory html_assgn. For each lab exercise create a directory as lab <lab number>.
- Download all files required to complete assignments from: http://pace.patni.com/TechRS/download.asp?course=Internet_HTML
- You may also look up the on-line help provided in the MSDN library.

Learning More (Bibliography if applicable)

- <http://msdn.microsoft.com>
- <http://www.asp.net/entity-framework>
- <https://msdn.microsoft.com/en-in/data/ef.aspx>
- Entity Framework 6 Recipes by Apress publication

Problem Statement/ Case Study (If applicable)

Give the case study used for this lab book here. If applicable.

Lab 1. LINQ Basics

Goals	Understand the process of Implementing LINQ to a Collection Learn to use LINQ Learn to use LINQ Operators
Time	60 minutes

- 1) Create a console application and add class named Employee with following field.

Employee Class

EmployeeID (Integer)
 FirstName (String)
 LastName (String)
 Title (String)
 DOB (Date)
 DOJ (Date)
 City (String)

- 2) Create a Generic List Collection empList and populate it with the following records.

EmployeeID	FirstName	LastName	Title	DOB	DOJ	City
1001	Malcolm	Daruwalla	Manager	16/11/1984	8/6/2011	Mumbai
1002	Asdin	Dhalla	AsstManager	20/08/1984	7/7/2012	Mumbai
1003	Madhavi	Oza	Consultant	14/11/1987	12/4/2015	Pune
1004	Saba	Shaikh	SE	3/6/1990	2/2/2016	Pune
1005	Nazia	Shaikh	SE	8/3/1991	2/2/2016	Mumbai
1006	Amit	Pathak	Consultant	7/11/1989	8/8/2014	Chennai
1007	Vijay	Natrajan	Consultant	2/12/1989	1/6/2015	Mumbai
1008	Rahul	Dubey	Associate	11/11/1993	6/11/2014	Chennai
1009	Suresh	Mistry	Associate	12/8/1992	3/12/2014	Chennai
1010	Sumit	Shah	Manager	12/4/1991	2/1/2016	Pune

- 3) Now once the collection created write down and execute the LINQ queries for collection as follows
 - i) Display detail of all the employee
 - ii) Display details of all the employee whose location is not Mumbai
 - iii) Display details of all the employee whose title is AsstManager
 - iv) Display details of all the employee whose Last Name start with S
 - v) Display a list of all the employee who have joined before 1/1/2015

- vi) Display a list of all the employee whose date of birth is after 1/1/1990
- vii) Display a list of all the employee whose designation is Consultant and Associate
- viii) Display total number of employees
- ix) Display total number of employees belonging to "Chennai"
- x) Display highest employee id from the list
- xi) Display total number of employee who have joined after 1/1/2015
- xii) Display total number of employee whose designation is not "Associate"
- xiii) Display total number of employee based on City
- xiv) Display total number of employee based on city and title
- xv) Display total number of employee who is youngest in the list

Lab 2. Creating Entity Data Model

Goals	At the end of this lab session you will be able to: <ul style="list-style-type: none">• Understand the need for frames in web pages.• Create and work with frames.• Manage large content with frame
Time	30 Minutes

Part 1:-

Using Code-First approach

Solution:-

Create a Console application and name the application as **CodeFirstConsoleApp**

After the project is created Now we have to add the Entity framework Library to Project .For that we will Nuget Package Manager Dialog

To Open Nuget Package Manager Dialog we need to follow the following step

Tools →Nuget Package Manager →Manage Nuget Package for the Solution

In the dialog box for Nuget Package Manager select entity framework and click on install this will install EntityFramework to the project

After installing the EntityFramework to the project we can see that EntityFrame dll file has been added to the References folder in Solution Explorer

Now as we have added the EntityFramework dll to the project now we have to add the Entity and Context to the project.

To add a entity class In the Solution Explorer right click on the project name than Add → Class and name the class as Product.cs

Once the class has been created add the following code to the class

After adding the class now we have to add a Context Class to the project .Context class will allow to perform database operation like add ,delete etc.

Context Class will always inherit from DbContext class available in System.Data.Entity namespace

After adding the context class add the following code to the class

In the Program.cs class file add the following code

When the above code is executed it will create the database and table based on the Entity and the above record into the table

After executing the application we will get the following output

Now we will check database which is created for that open Sql Server Management Studio and connect to the default instance. In the object explorer you can see the database and table being created

Part 2:-

Using Database First Approach

Solution:-

Open Sql Server Management Studio and Create a database name MusicStore and add aTable named Album with the following fields

Album
AlbumID
Name
Genre
Year
Price

Add some dummy record into the table.

Now create a console application name DatabaseFirstConsoleApp and add the entityframework as done in the previous example

Once the project is created and entity framework library is added. Now we have an Entity data Model to the project.

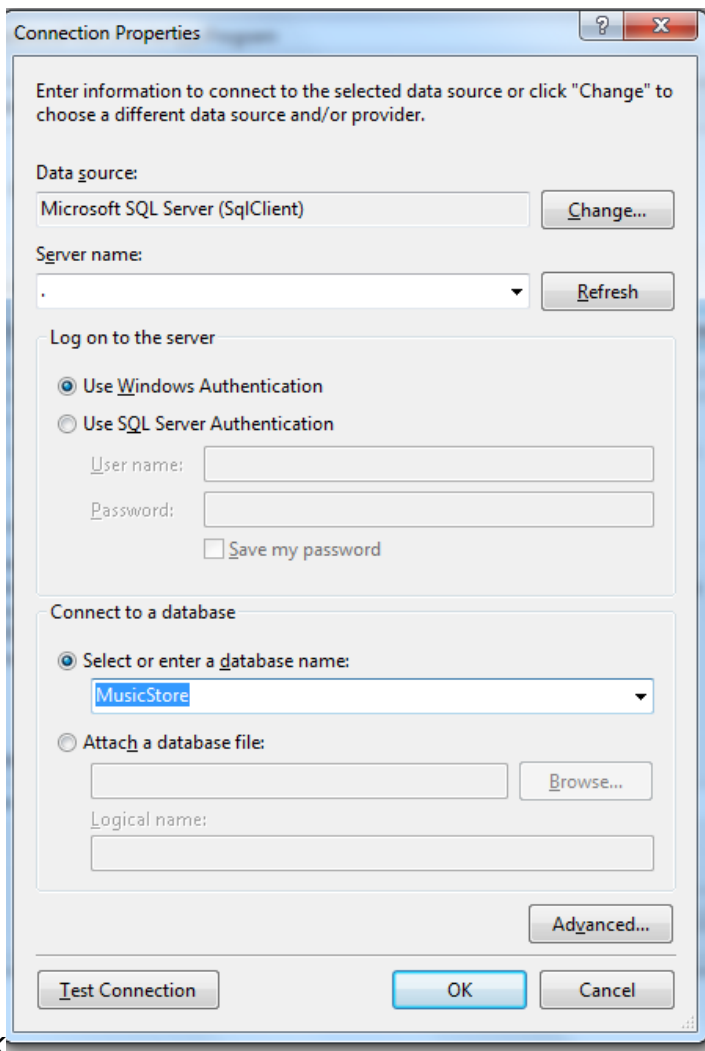
To add an Entity data Model to the Project in the solution explorer, right-click in the project → Add → New Item. Under the New Item dialog box, select ADO.NET Entity Data Model and give the name as MusicStoreModel and click on Add.

As we add the Entity Data model to the project, the Entity Data model wizard popup in which we have different options for initializing the entity data model.

In that dialog box, **select EF Designer from database** and click on Next.

Now on the next window, we have to select a database for Model creation, so now click on New Connection button in Choose Your Data Connection.

In the Connection properties dialog box, provide the Database server name and select the database you want to use. Click on Test Connection to test the connection and then click on



OK

Once the connection test is passed then click on OK

Now we can see the new connection string which we have created and a option to save the connection string in App.config or web.config file

Now Click on Next and Choose your Database Object and Settings option will be prompted .

In that we have select all the database object which we need to add to our Model.

We have select the Album table as we have only one table in the database

Now click on Finish this will add the EDM to the project and create all the required code.

In the above image we can see the model name MusicStoreModel.edmx containing Album Entity and the highlighted region show files generated for MusicStoreModel.edmx

Now as the model is created we can write code to interact with database and perform read/write operations.

To display details of the Albums write down the following code in Program.cs File

Output:-

To add an Album into the database table write the following code in Program.cs File

Output:-

Lab 3. Creating Entity Data Model Using Model First Approach.

Goals	Understand the process of Creating Entity Data Model using Model First Approach and Creating Complex Type Learn to use of Model First Approach to create database tables
Time	60 Minutes

Model First Approach :-

Model First allows you to create a new model using the Entity Framework Designer and then generate a database schema from the model. The model is stored in an EDMX file and can be viewed and edited in the Entity Framework Designer.

Solution:-

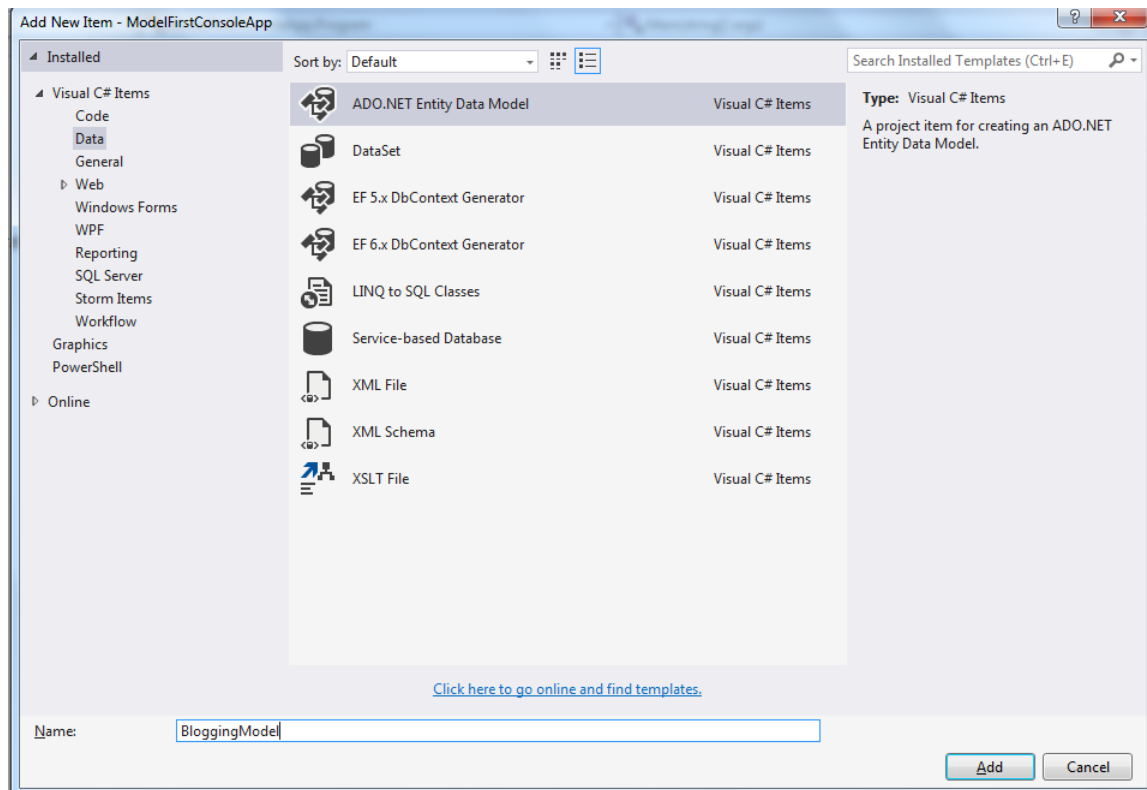
Open Visual Studio and create a console application named ModelFirstConsoleApp. After creating the project add Entityframework to the project using Nuget Package Manager

After creating the project and adding Entityframework to the project .Now we have a add a model to the project..

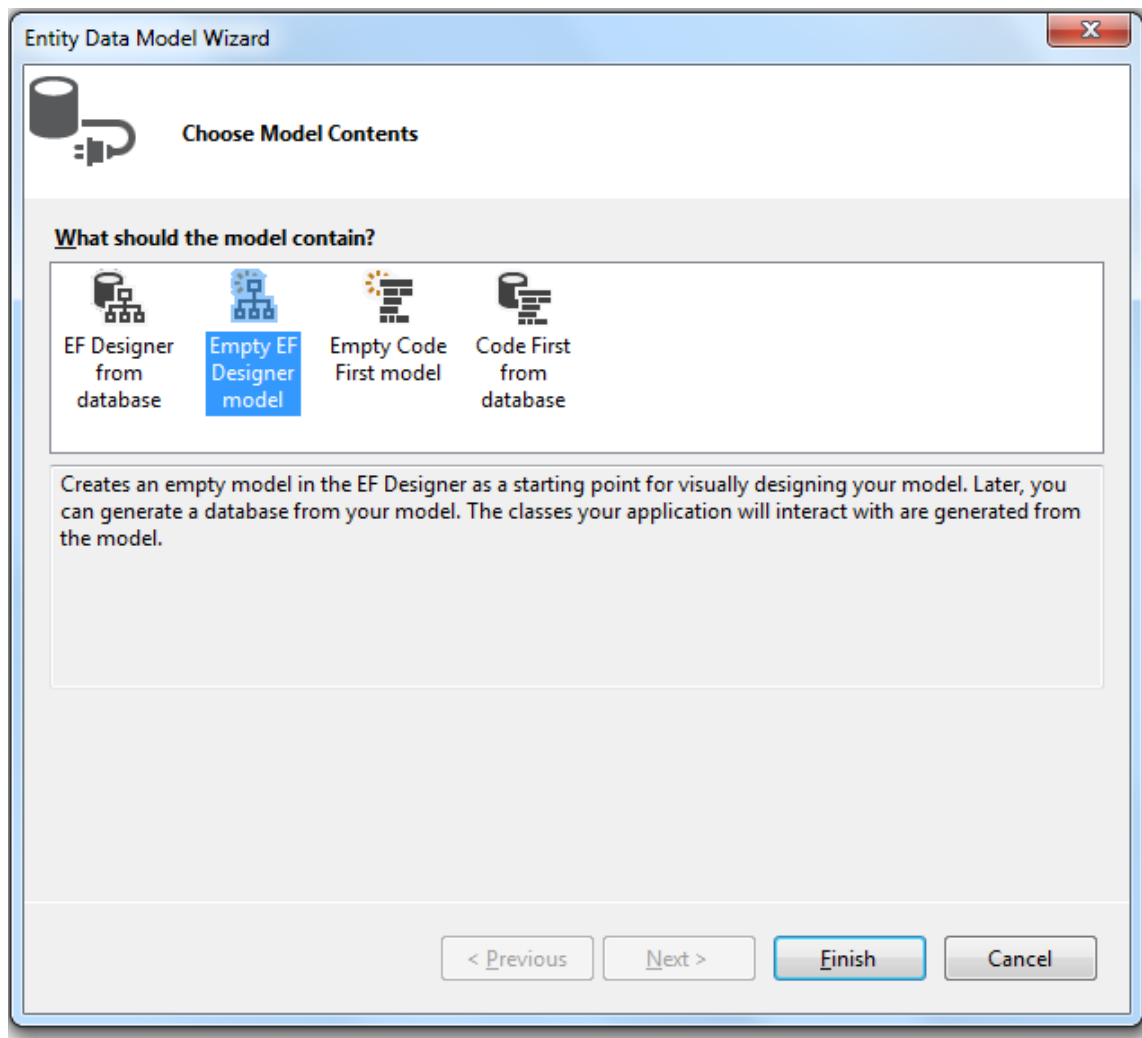
For adding a model to the project Right Click on the Project in Solution Explorer

→Add→New Item

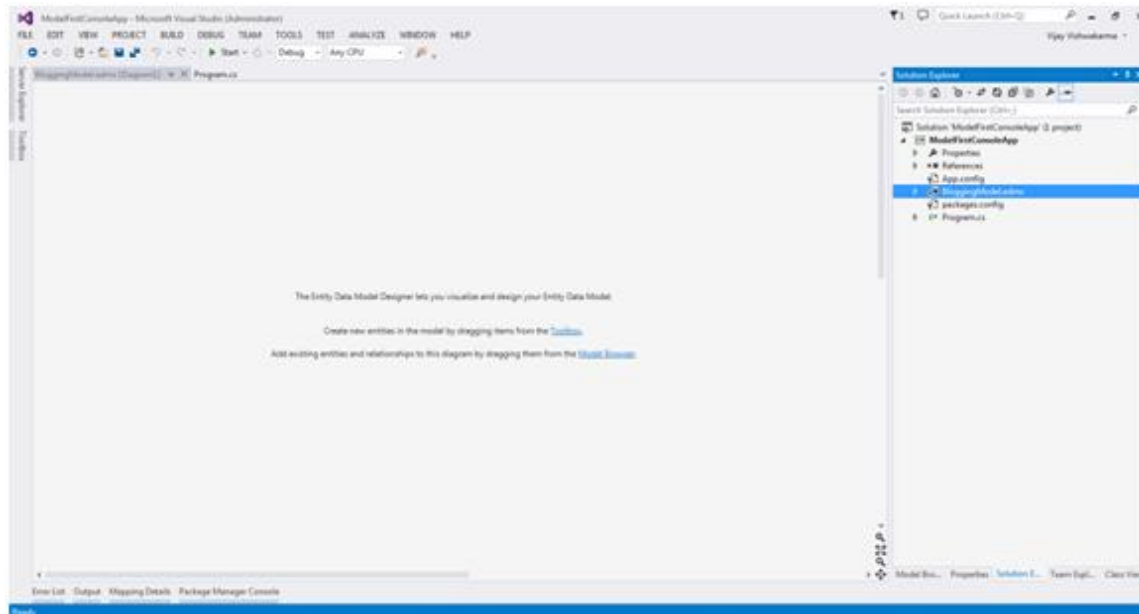
Now in the add new item dialog box select ADO.Net Entity Data Model and named it as BloggingModel and click on ADD



In the Entity Data Model Wizard select Empty EF Designer Model and click on Finish .

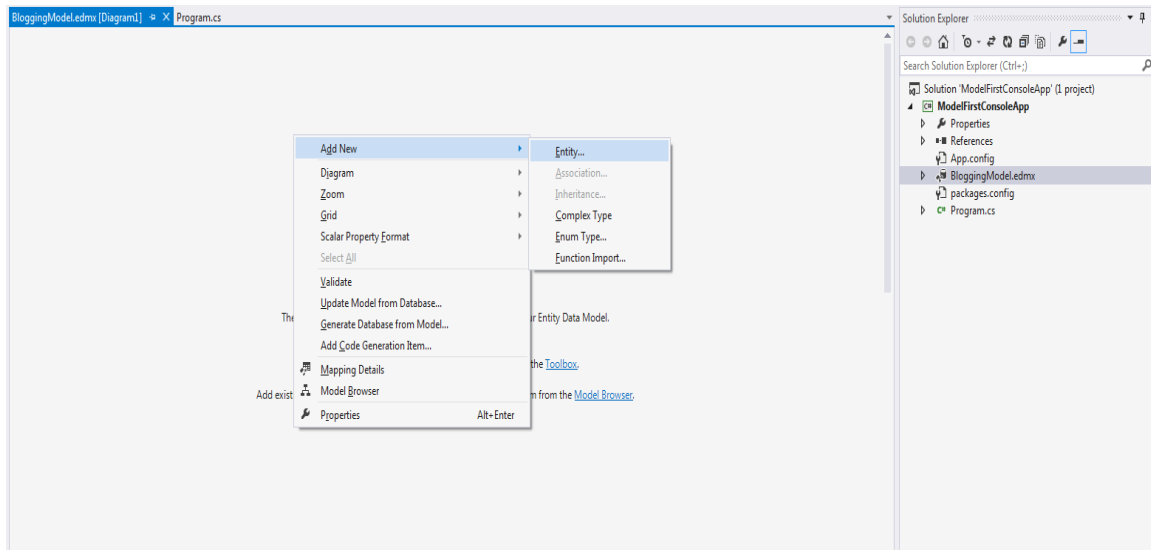


This will open up an empty EDMX on which we create Model from Scratch



Now we have to add entity in this model and create a database form it.

To add a Entity , Right click on the open edmx file and select Add New → Entity



After you select you will be prompted with the following Dialog for creating a new entity

The Dialog Display information like

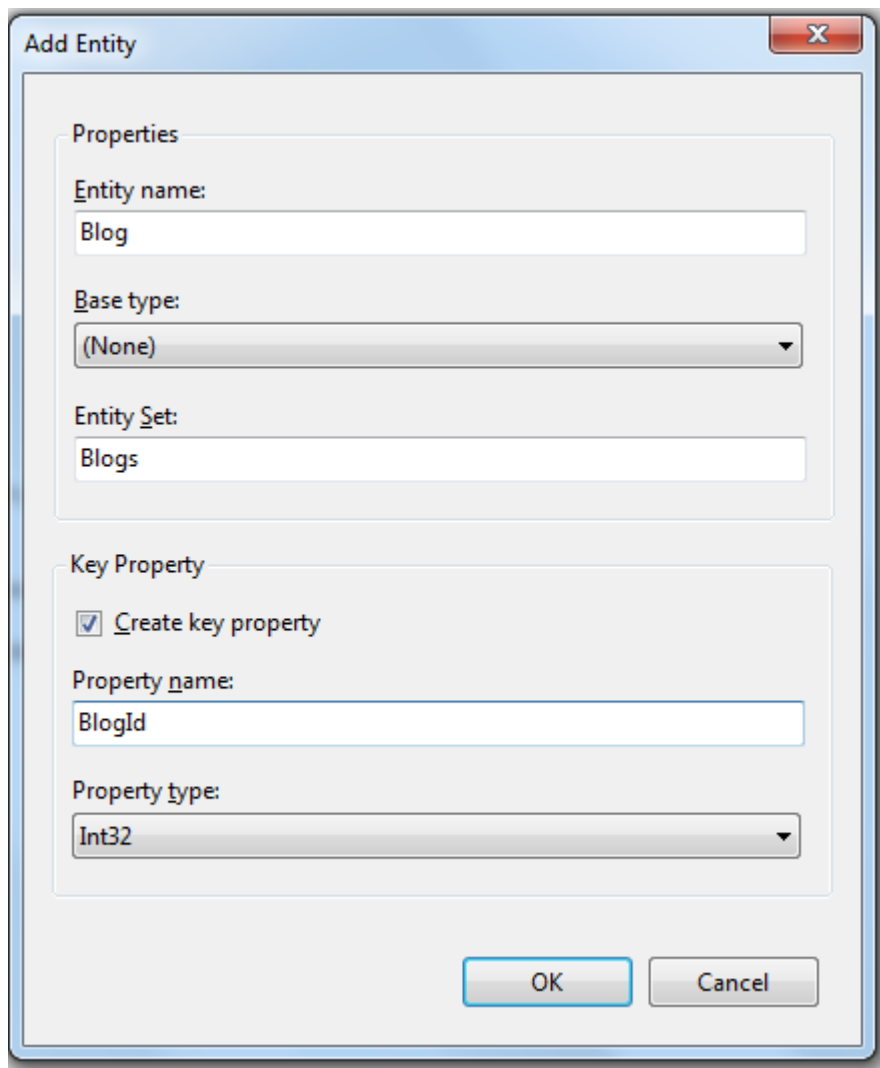
Entity Name :- This will be the name of Entity

Base Type :- Shows the base type of the entity

EntitySet :- Show the name of the Entity Set

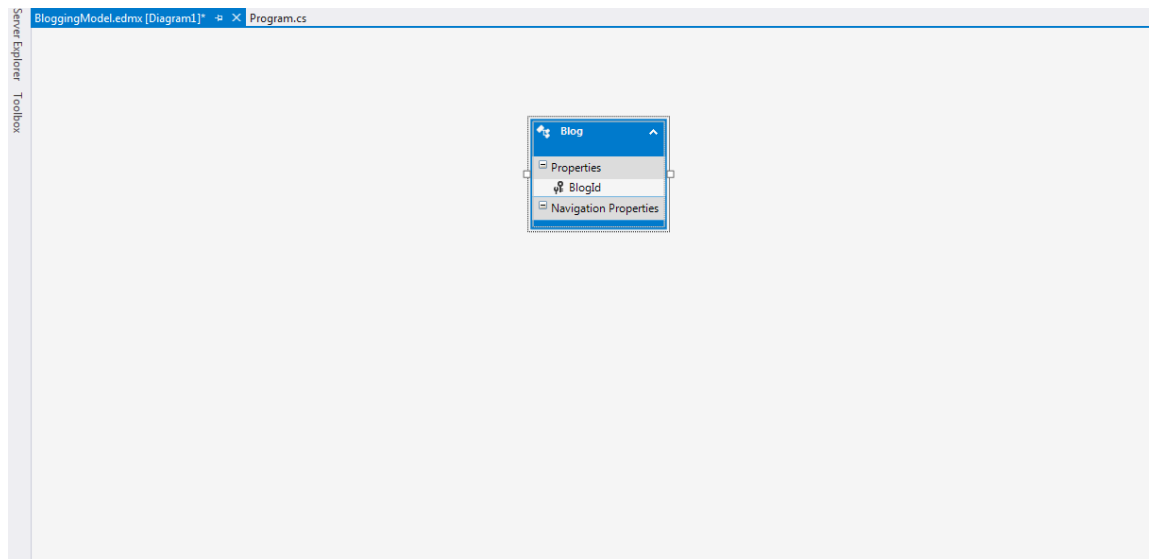
It also shows the option for creating Key Property where we have to specify Key Property Name and its Data Type

For our example we will name the entity as Blog and Entity Set as Blogs. Entity Set is automatically pluralized as we add the name of the Entity. and change Key Property from Id to BlogId



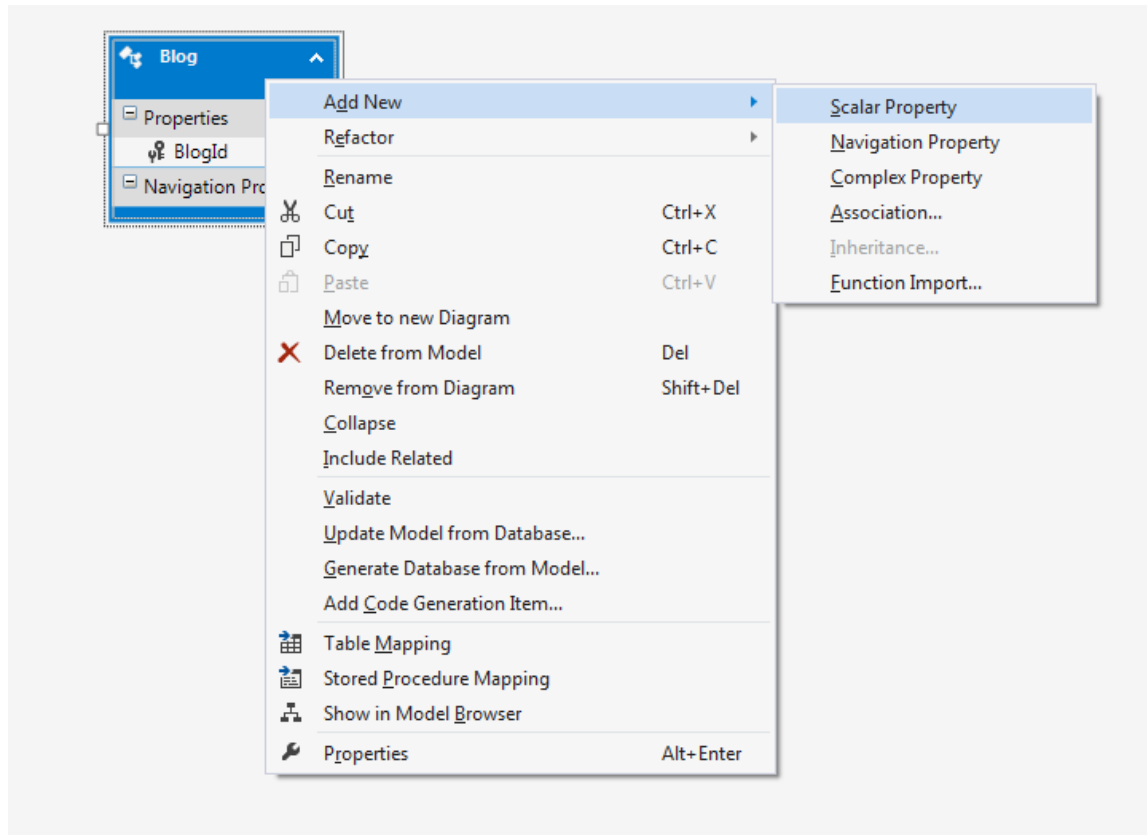
The image shows a Windows-style dialog box titled "Add Entity". It has a standard title bar with a close button (X) in the top right corner. The dialog is divided into two main sections. The first section, titled "Properties", contains three fields: "Entity name:" with the text "Blog", "Base type:" with a dropdown menu showing "(None)", and "Entity Set:" with the text "Blogs". The second section, titled "Key Property", contains a checked checkbox labeled "Create key property", a "Property name:" field with the text "BlogId", and a "Property type:" dropdown menu showing "Int32". At the bottom of the dialog are two buttons: "OK" and "Cancel".

After changing all the details click on OK .This will add the entity to EDMX designer

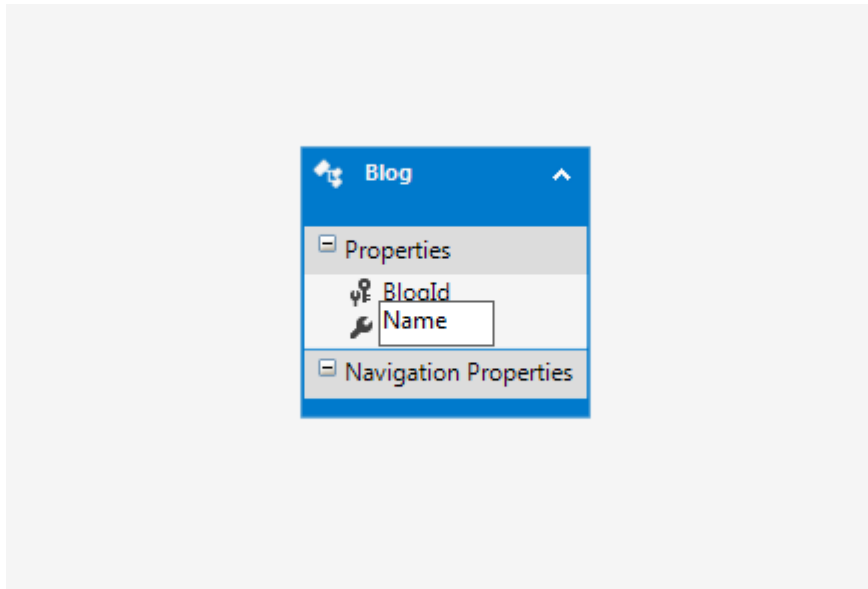


Now we have add Field to this entity for storing other information of Blog.

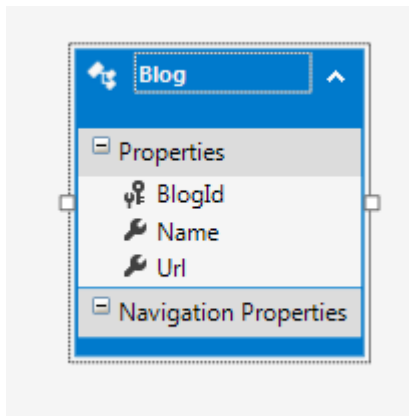
To add a Field or property to the Entity right on the Entity and select Add New → Scalar Property



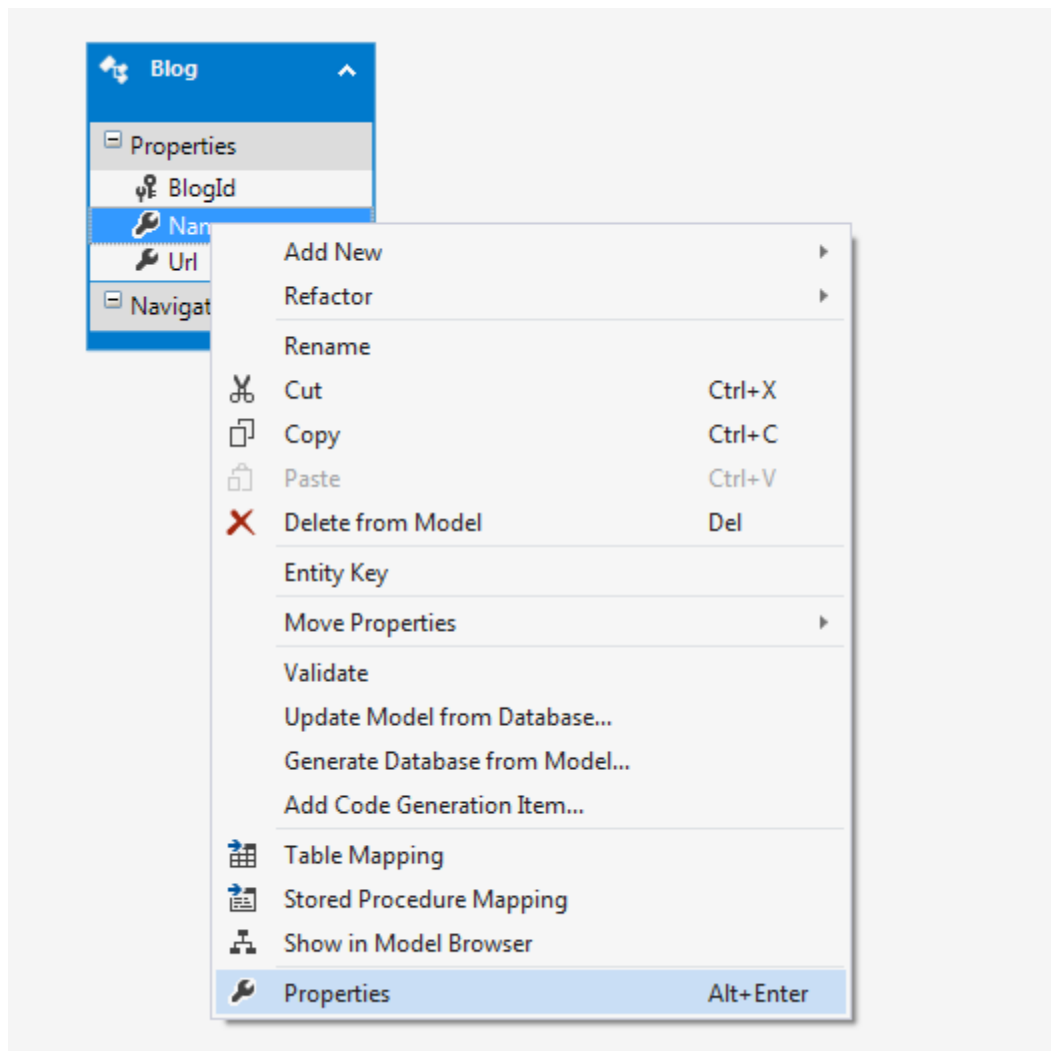
After selecting the Scalar Property , name it as Name

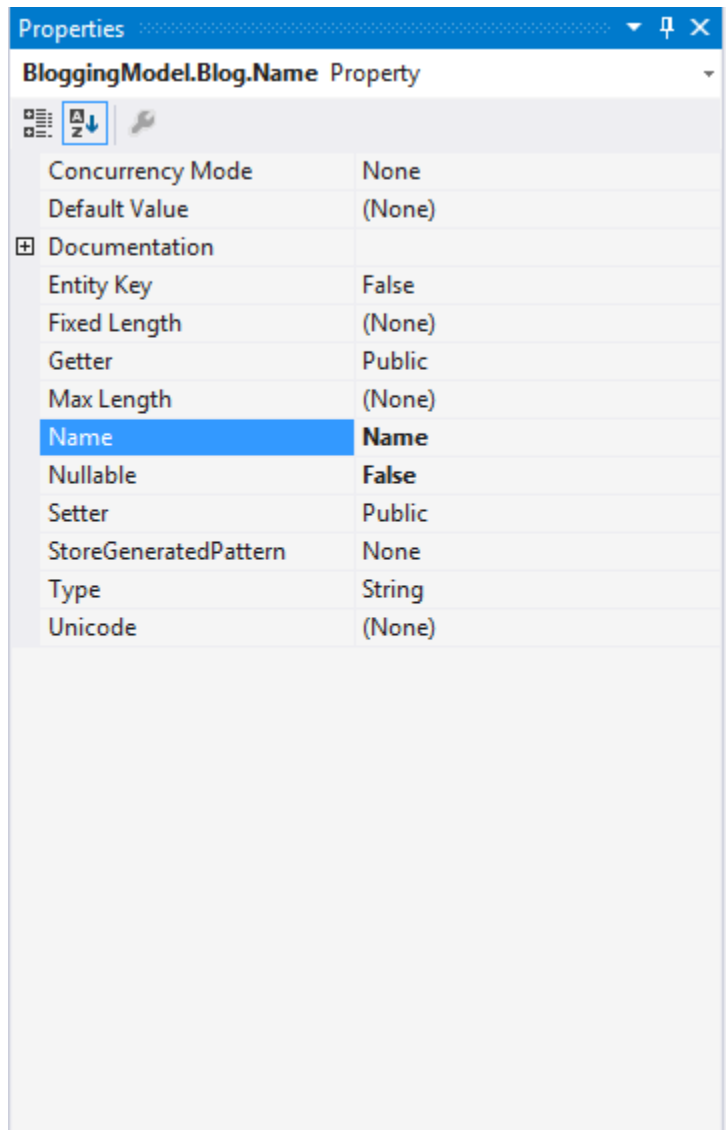


Similarly add another property named URL to the Entity. After adding the URL property the Entity Will look as follows



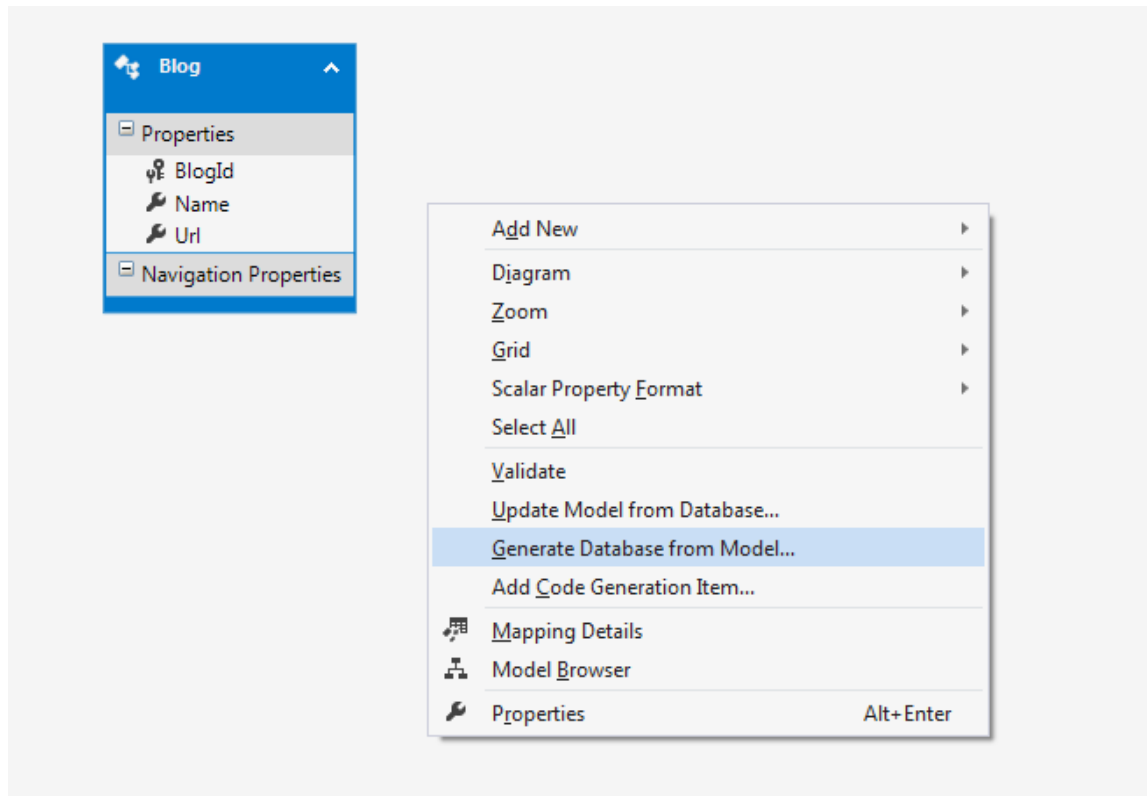
We can view the properties of each Property add the Entity by just right clicking and selecting the properties option





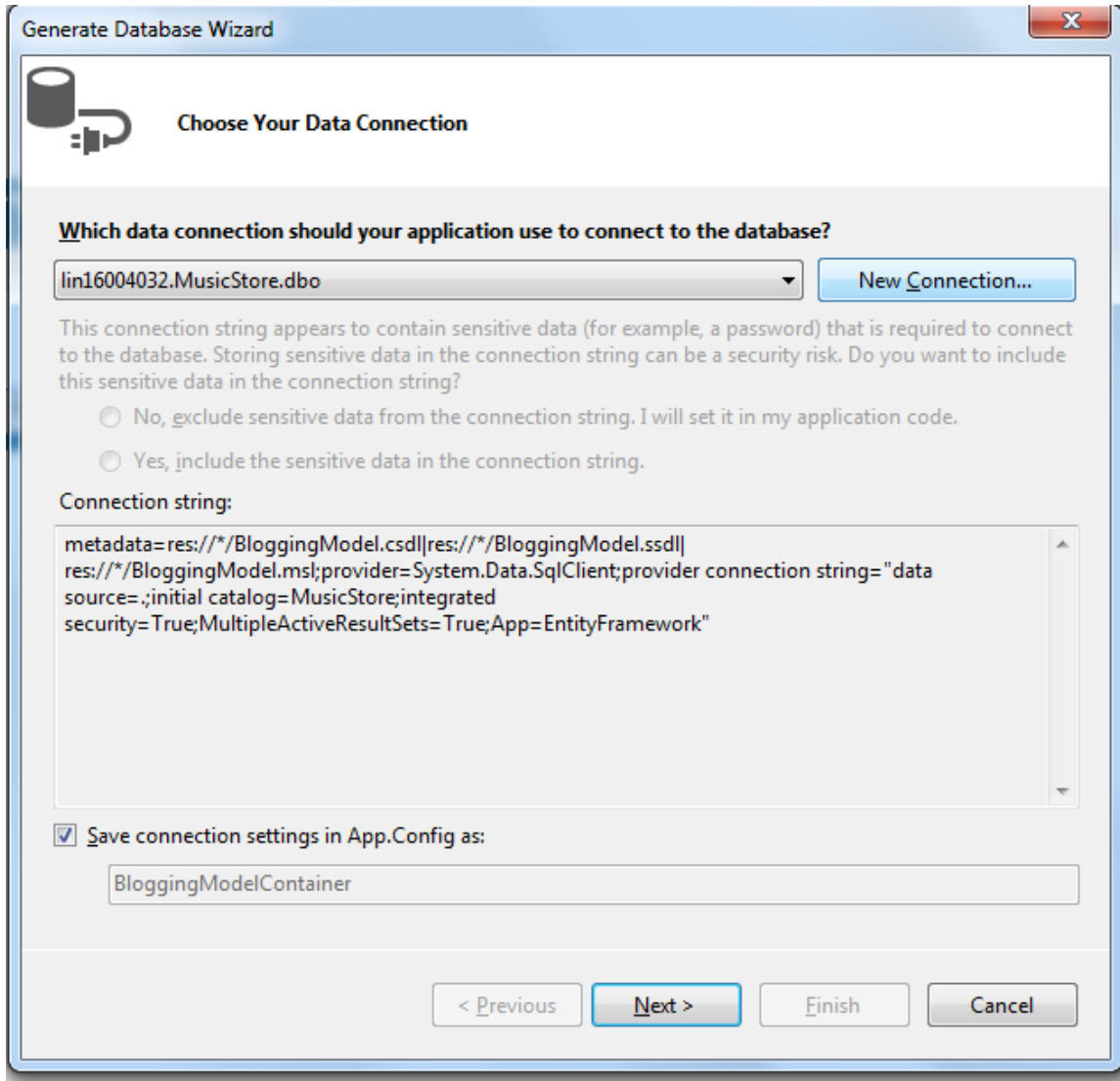
We can see above the properties show a lot of detail about the Name scalar property add to the entity .

Once the Entity has been created , now we have to create Database from the Model . To create the database from the model right click on the EDMX designer and select Generate Database from Model



After you click on the Generate Database from Model , Generate Database Wizard will be prompted .

In the Generate Database Wizard click on the New Connection to create a new connection .



The image shows a screenshot of the 'Generate Database Wizard' dialog box. The title bar reads 'Generate Database Wizard'. Inside the dialog, there is a section titled 'Choose Your Data Connection' with a database icon. Below this, a question asks: 'Which data connection should your application use to connect to the database?'. A dropdown menu shows 'lin16004032.MusicStore.dbo', and a 'New Connection...' button is to its right. A warning message states: 'This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?'. There are two radio buttons: 'No, exclude sensitive data from the connection string. I will set it in my application code.' and 'Yes, include the sensitive data in the connection string.'. Below this, the 'Connection string:' is displayed in a text area with the following content: `metadata=res://*/BloggingModel.csdl|res://*/BloggingModel.ssdl|res://*/BloggingModel.msl;provider=System.Data.SqlClient;provider connection string="data source=.;initial catalog=MusicStore;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework"`. A checkbox 'Save connection settings in App.Config as:' is checked, and a text box below it contains 'BloggingModelContainer'. At the bottom, there are four buttons: '< Previous', 'Next >', 'Finish', and 'Cancel'.

Generate Database Wizard

Choose Your Data Connection

Which data connection should your application use to connect to the database?

lin16004032.MusicStore.dbo New Connection...

This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

☐ No, exclude sensitive data from the connection string. I will set it in my application code.

☐ Yes, include the sensitive data in the connection string.

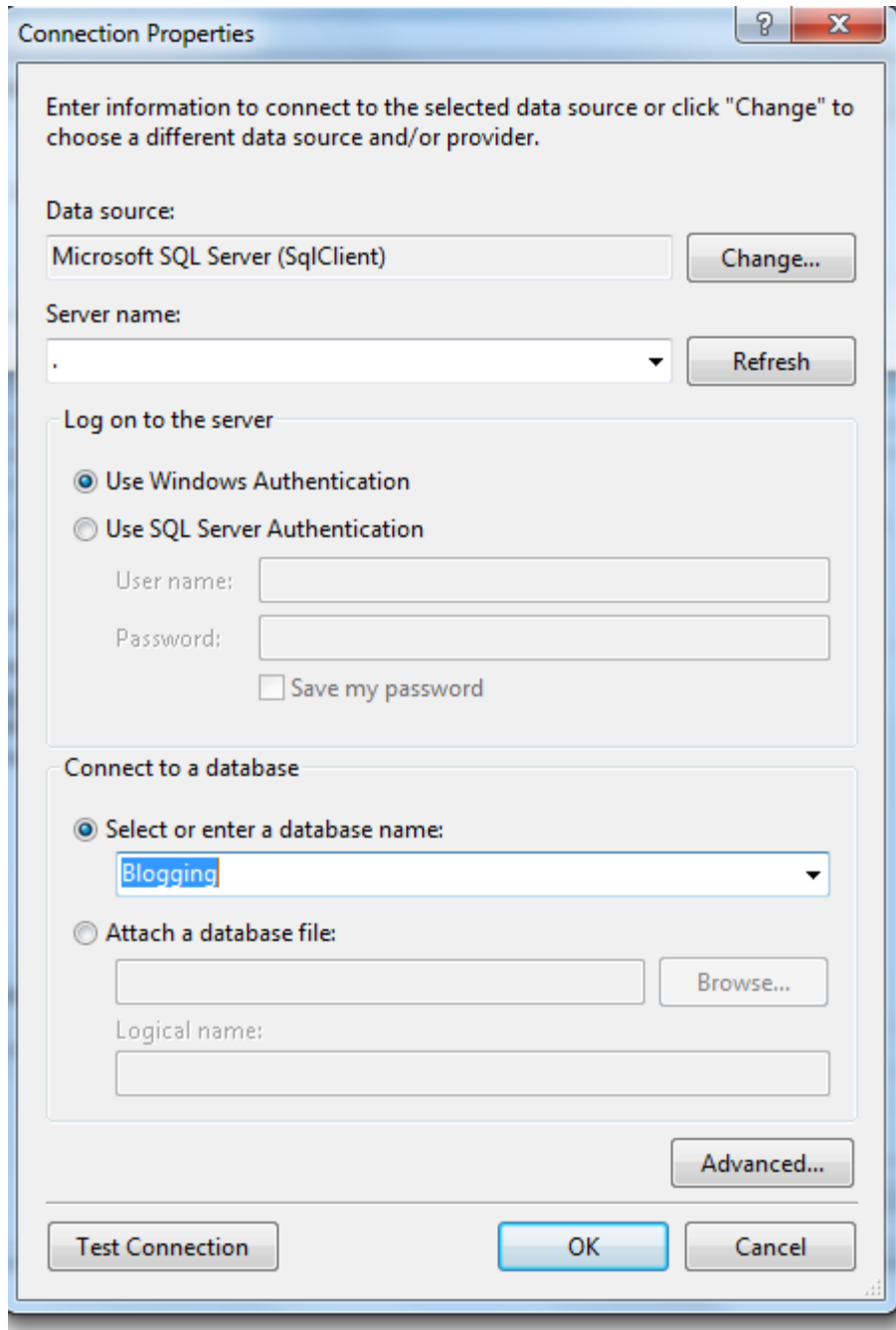
Connection string:

```
metadata=res://*/BloggingModel.csdl|res://*/BloggingModel.ssdl|
res://*/BloggingModel.msl;provider=System.Data.SqlClient;provider connection string="data
source=.;initial catalog=MusicStore;integrated
security=True;MultipleActiveResultSets=True;App=EntityFramework"
```

☒ Save connection settings in App.Config as:

BloggingModelContainer

< Previous Next > Finish Cancel



Connection Properties

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:
. Refresh

Log on to the server

☒ Use Windows Authentication
☐ Use SQL Server Authentication

User name:
Password:
☐ Save my password

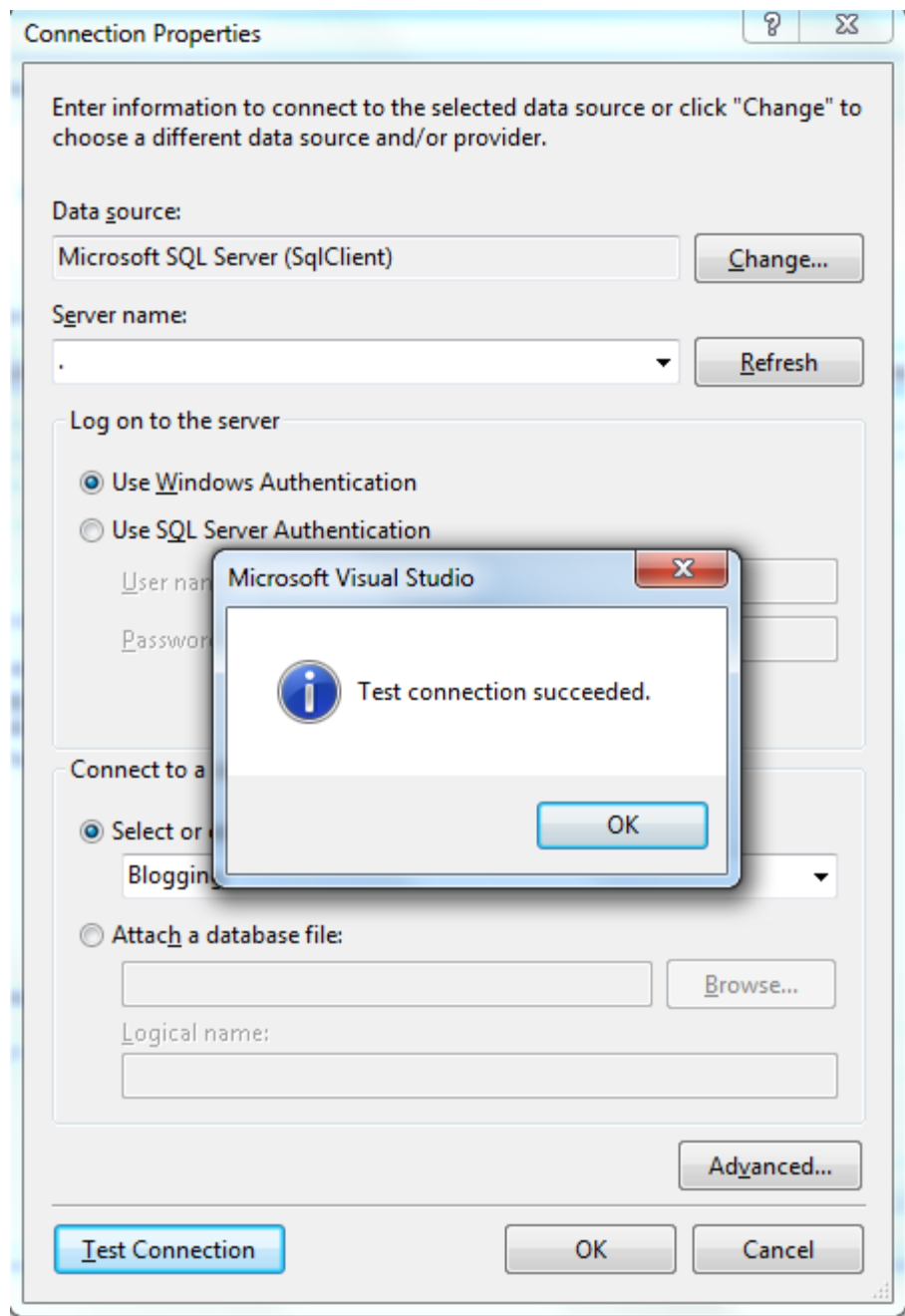
Connect to a database

☒ Select or enter a database name:
Blogging ▼

☐ Attach a database file:
 Browse...
Logical name:

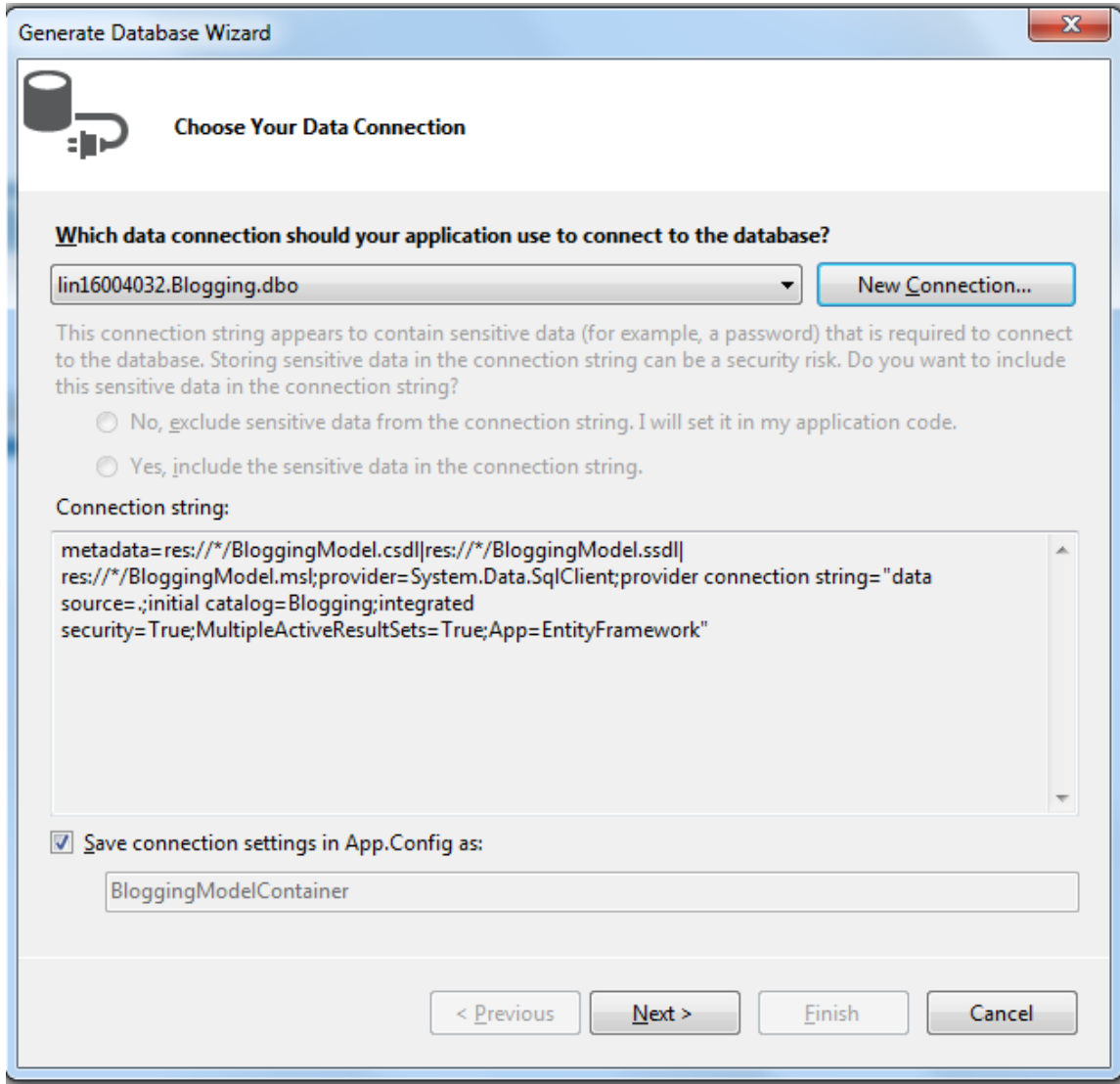
Advanced...

Test Connection OK Cancel




After testing the connection click on OK .

Now we have a new connection in the Generate Database Wizard



The screenshot shows the 'Generate Database Wizard' window, specifically the 'Choose Your Data Connection' step. The window has a title bar with the text 'Generate Database Wizard' and a close button. Below the title bar is a header area with a database icon and the text 'Choose Your Data Connection'. The main content area contains the question 'Which data connection should your application use to connect to the database?'. Below this question is a dropdown menu showing 'lin16004032.Blogging.dbo' and a 'New Connection...' button. A warning message states: 'This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?'. There are two radio buttons: 'No, exclude sensitive data from the connection string. I will set it in my application code.' and 'Yes, include the sensitive data in the connection string.'. Below the radio buttons is a text box labeled 'Connection string:' containing the following text: 'metadata=res:/*/*BloggngModel.csdl|res:/*/*BloggngModel.ssdl|res:/*/*BloggngModel.msl;provider=System.Data.SqlClient;provider connection string="data source=.;initial catalog=Blogging;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework"'. At the bottom of the window, there is a checkbox labeled 'Save connection settings in App.Config as:' which is checked. Below this checkbox is a text box containing 'BloggingModelContainer'. At the very bottom of the window are four buttons: '< Previous', 'Next >', 'Finish', and 'Cancel'.

Generate Database Wizard

 Choose Your Data Connection

Which data connection should your application use to connect to the database?

lin16004032.Blogging.dbo New Connection...

This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

☐ No, exclude sensitive data from the connection string. I will set it in my application code.

☐ Yes, include the sensitive data in the connection string.

Connection string:

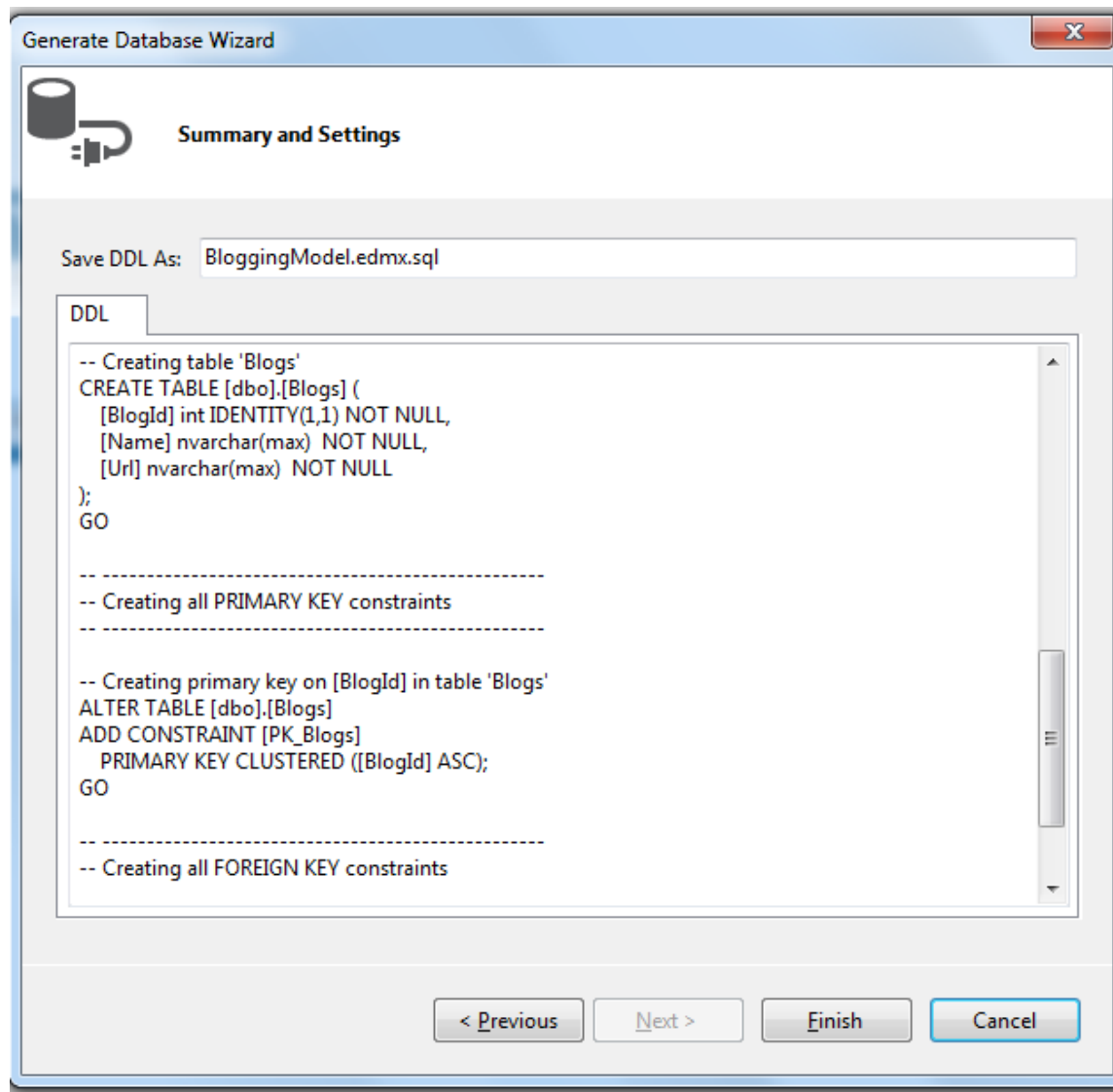
metadata=res:/*/*BloggngModel.csdl|res:/*/*BloggngModel.ssdl|res:/*/*BloggngModel.msl;provider=System.Data.SqlClient;provider connection string="data source=.;initial catalog=Blogging;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework"

☒ Save connection settings in App.Config as:

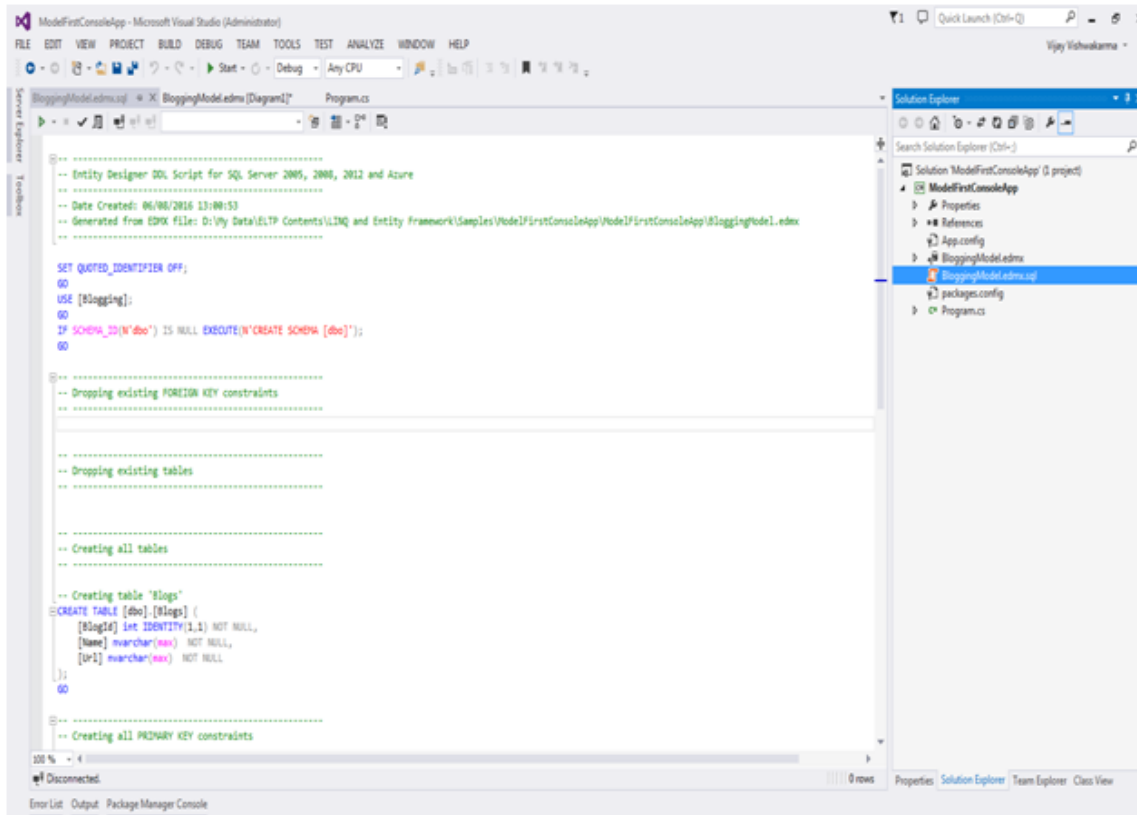
BloggingModelContainer

< Previous Next > Finish Cancel

Now click on Next , DDL query for generating the Database and table will be auto generated.



Now click on finish , a file named BloggingModel.edmx.sql will be created which will contain sql shown in the dialog for creating the database.

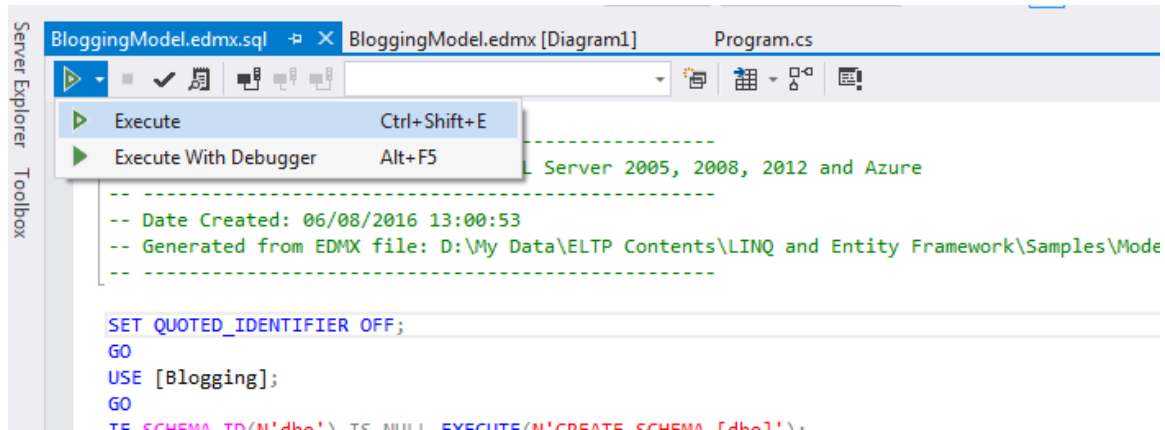


```
-- Entity Designer SQL Script for SQL Server 2005, 2008, 2012 and Azure
-- Date Created: 06/06/2016 13:00:53
-- Generated from EDMX File: D:\Vij Data\ELTP Contents\LINQ and Entity Framework\Samples\ModelFirstConsoleApp\ModelFirstConsoleApp\BloggingModel.edmx
--
SET QUOTED_IDENTIFIER OFF;
GO
USE [Blogging];
GO
IF SCHEMA_ID(N'dbo') IS NULL EXECUTE(N'CREATE SCHEMA [dbo]');
GO

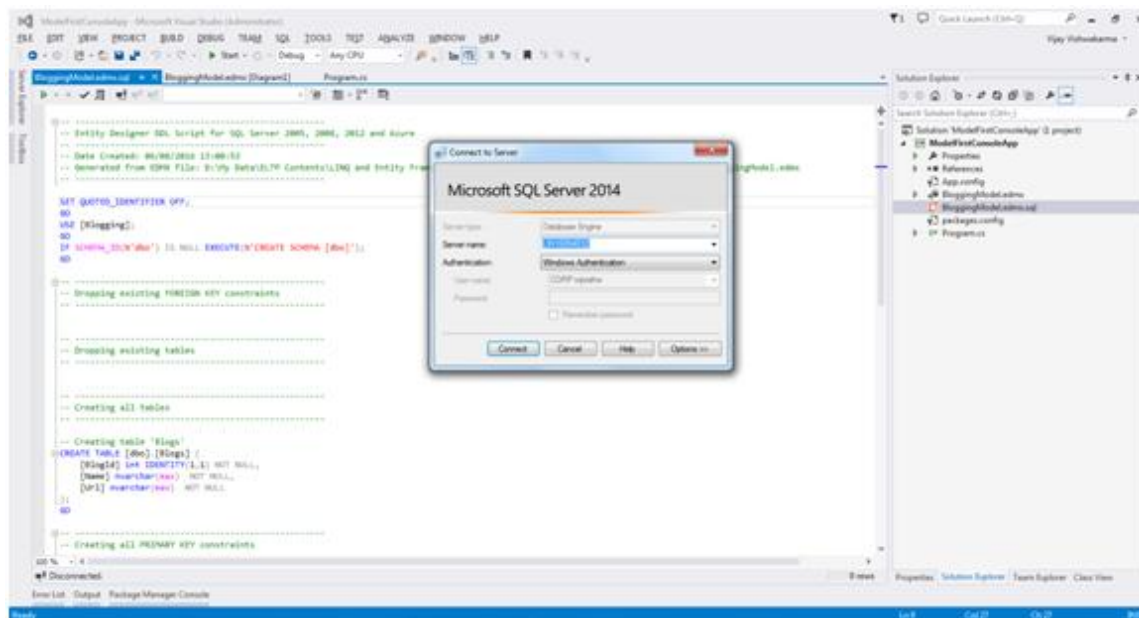
-- Dropping existing FOREIGN KEY constraints
--
-- Dropping existing tables
--
-- Creating all tables
--
-- Creating table 'Blogs'
CREATE TABLE [dbo].[Blogs] (
    [BlogId] INT IDENTITY(1,1) NOT NULL,
    [Name] nvarchar(max) NOT NULL,
    [Url] nvarchar(max) NOT NULL
);
GO

-- Creating all PRIMARY KEY constraints
```

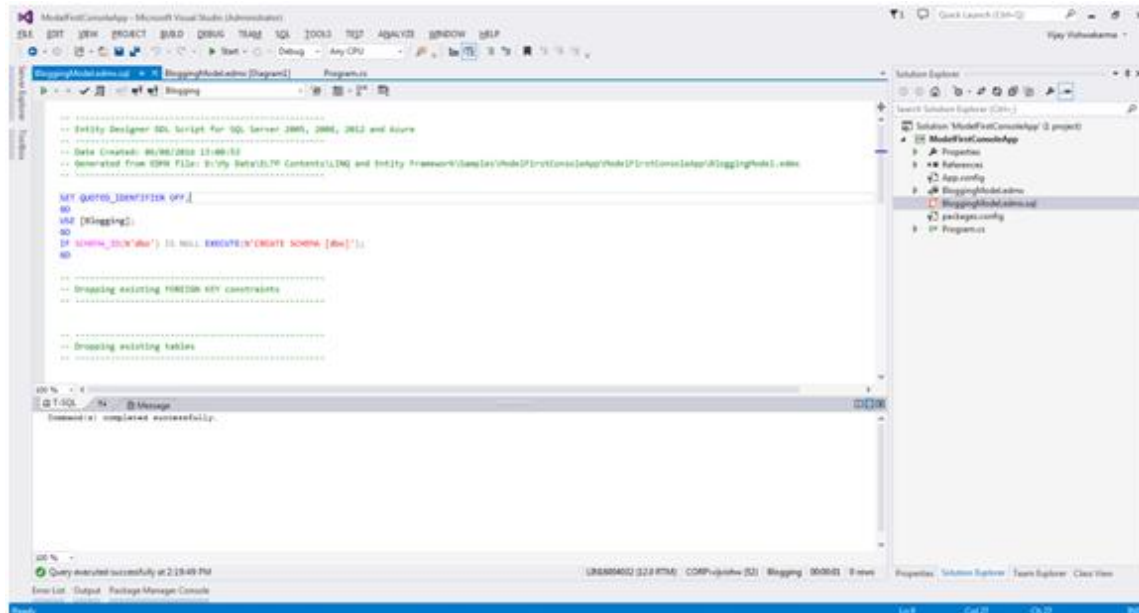
Now In the BloggingModel.edmx.sql file select Execute and click on it



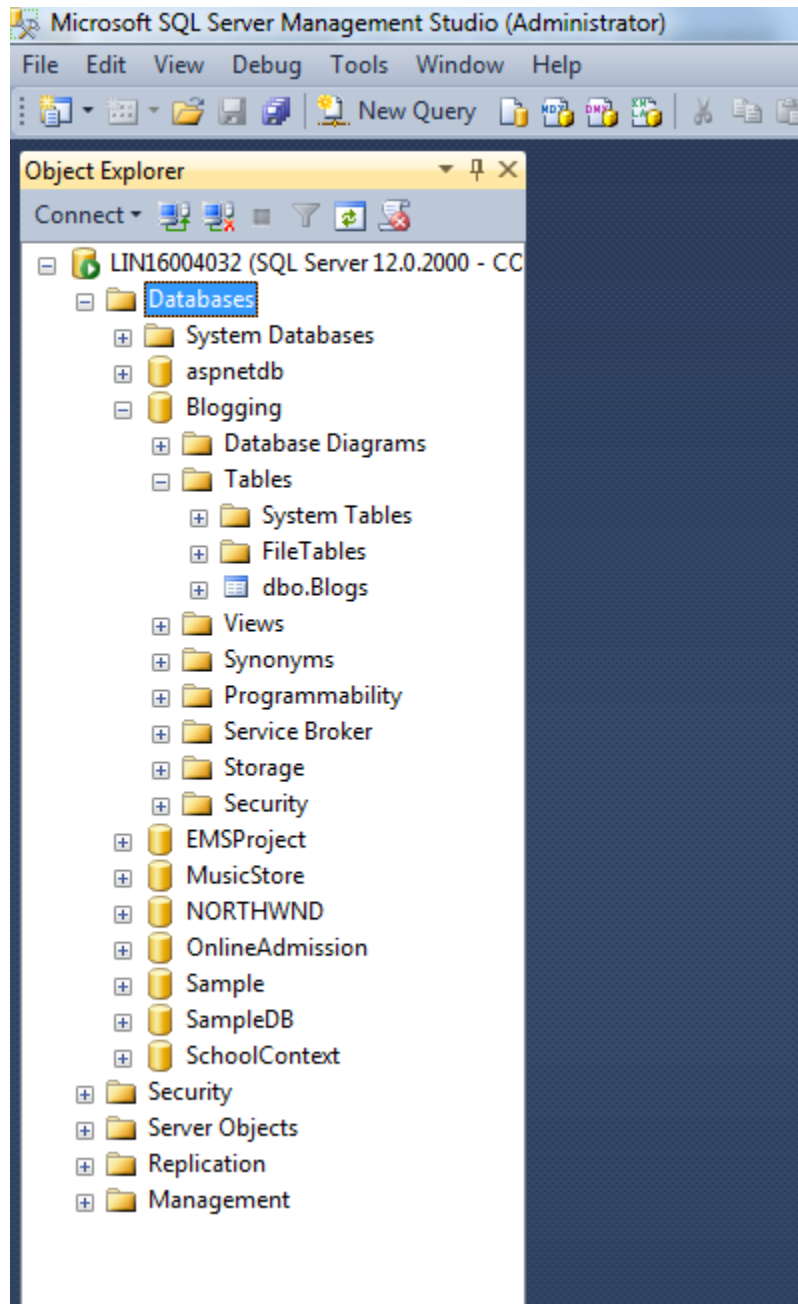
This will execute the query against the connected database instance if not connected you will be prompted to connect to sql server instance.



Click on connect and query will executed



Now in the Sql Server Management Studio we can the table which has been created



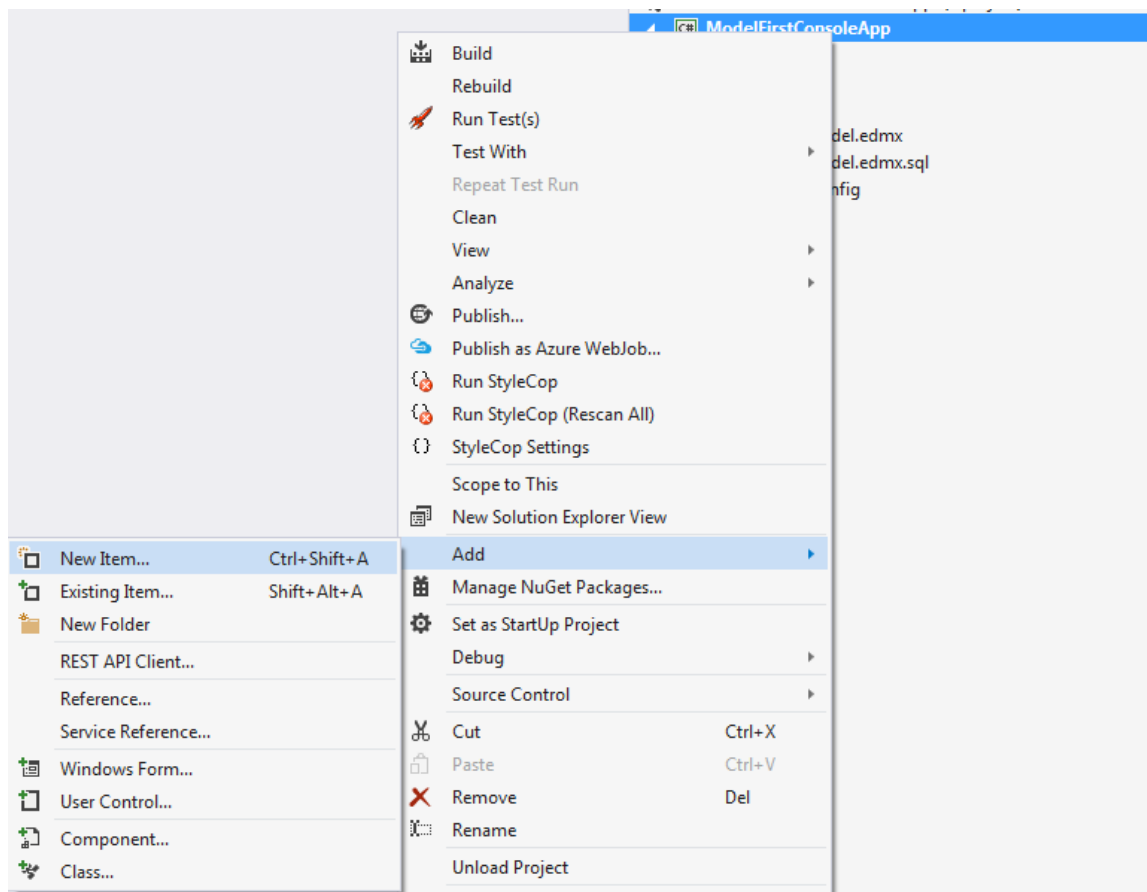
Complex Type:-

Complex types are non-scalar properties of entity types that enable scalar properties to be organized within entities. Like entities, complex types consist of scalar properties or other complex type properties.

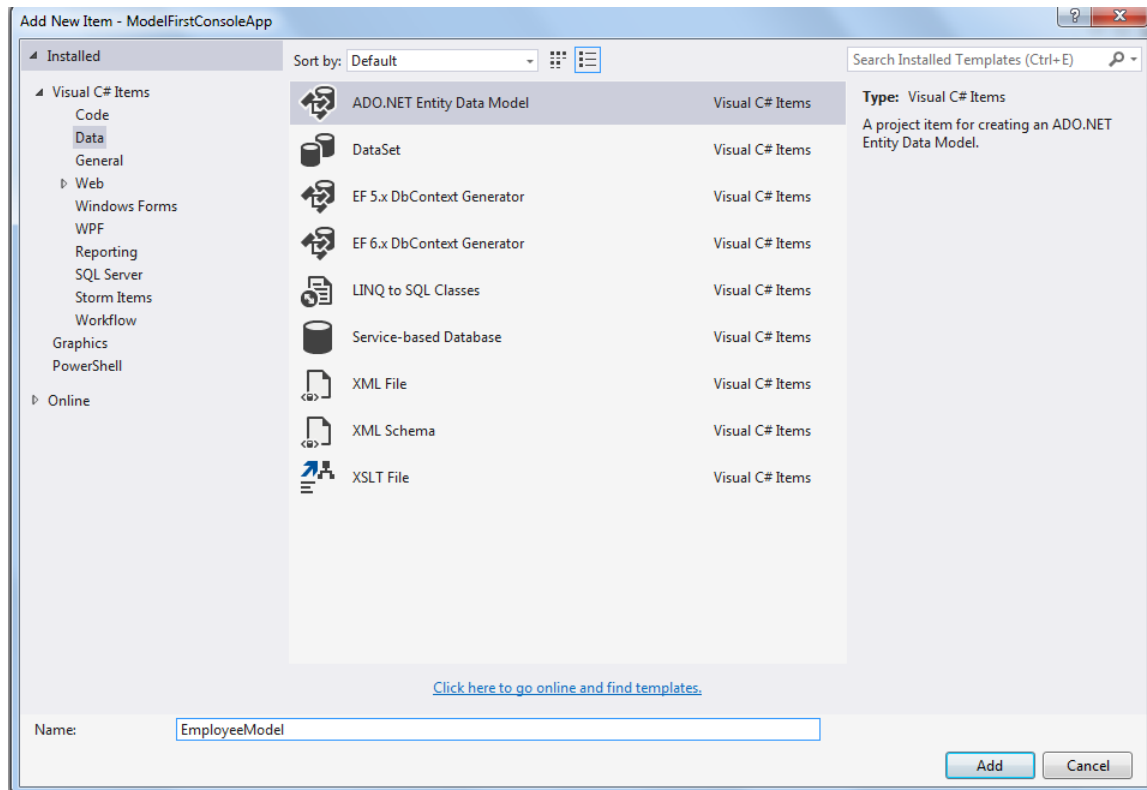
Solution:-

Creating a Complex Type using EF Designer.

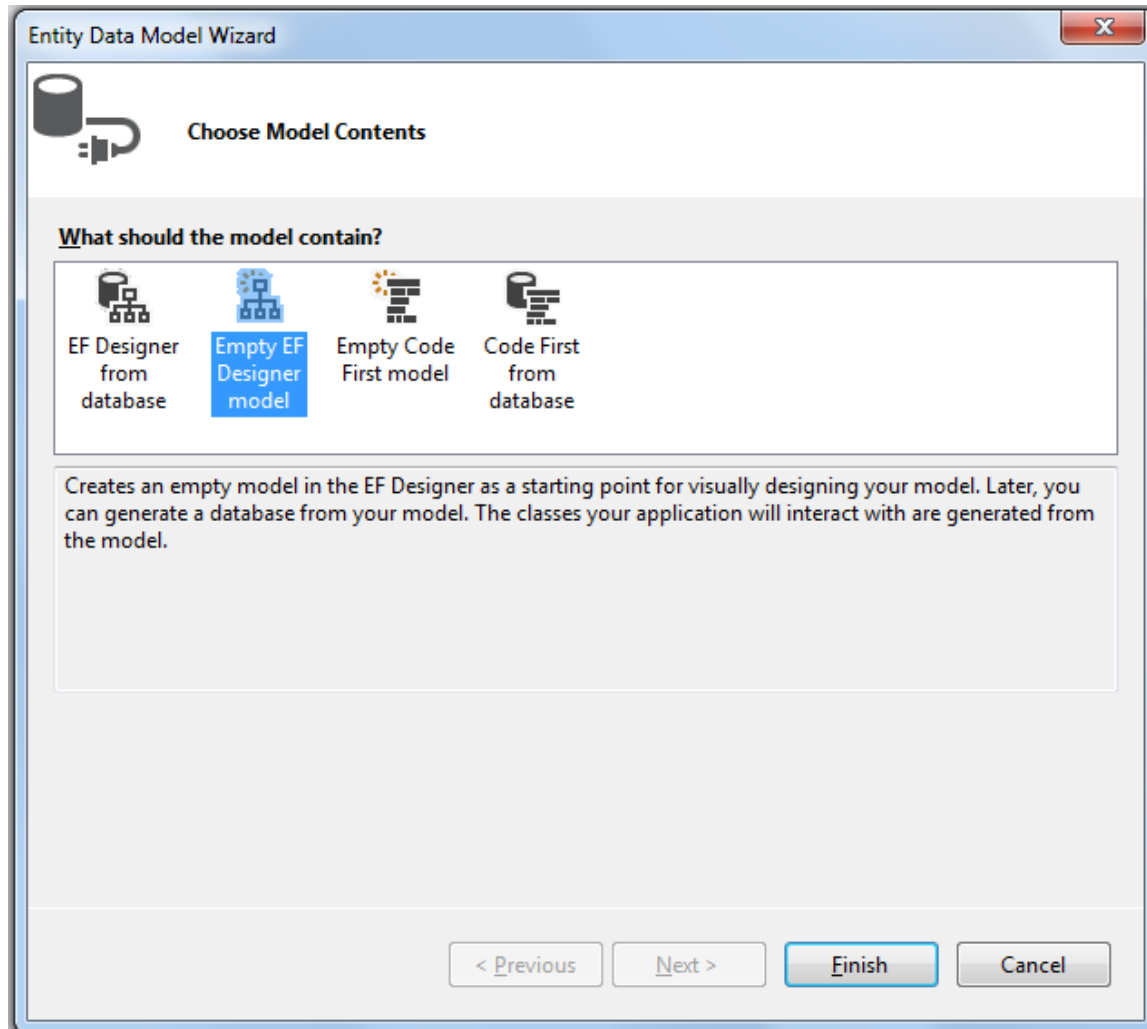
In the ModelFirstConsoleApp Example, Right click on the project in solution explorer then click on Add→New Item



In the Add New Item Dialog select ADO.Net Entity Data Model and Name it as EmployeeModel and click on Add



In the Entity Data Model Wizard Select Empty EF Designer Model and click on Finish.



Now after adding the EmployeeModel.edmx file , In the Model Browser windows right click on the Complex Types folder of EmployeeModel and Click on Add new Complex Type and Name it as AddressType

Now after creating the type Now we have to add Scalar property to it .

For Adding scalar property to the complex type right click on the complex type then Add→Scalar Property →Select data type of the property and then named the property as StreetName.

Now Similarly Three More property to the Complex Type as follows

- 1) Locality
- 2) City
- 3) Pincode

Now we will add a Entity to the Designer. Right click on the designer click on Add New → Entity and name the entity as Employee

Now in the Entity add a Scalar property named as Name and DOB

Now to Add the Complex type to the Entity right click on the entity and click on Add New →Complex Property and name the property as Address

As we have only one Complex Type in the Model so the newly added comple property in the Entity is Model to that complex type

Now we have to generate database from the model for that Right click on the designer and click on the Generate Database from Model.

In this we will use the connection that we have created for Blogging database.

Click on Finish this will create a Sql file named as EmployeeModel.edmx.sql which we will execute against database for creating database and tables .

Now click on the Execute button to execute the sql query against the database.

Connect to database

Now to view the table open Sql Server Management Studio and in the Object Explorer expand blogging database and then expand table tab to view the newly created table.

Lab 4. Basic Query Operations using LINQ to Entities

Goals	Understand the process of performing LINQ queries on Entity Models Learn to use of LINQ to Entities Learn to Manipulate Data
Time	60 Minutes

Solution:-

Open Visual studio and create a new console application named as LINQEFCConsoleApp and the Entity Framework Library using the Nuget Package Manager to the application.

Now we have to create a Entity Data Model from an existing database.

Right click on the project in the solution explorer and select Add→ New Item

In the new Item Dialog Box select ADO.Net Entity Data Model and name it as Training Model

In the Entity Data Model Wizard select EF Designer from database and click on Next and configure the datasource.

Now Click on New Connection in the Choose Your Data source option

In the connection properties windows Provide the sql server name , authentication type and database name click on Test Connection to test the connection .

Click on OK

Now click on next to select the database objects which will be part of Entity Data Model.

Tick the Staff_Master and Student_Master Table and click on Finish this will add the entities to the model.

Now we have to write LINQ query against the model for reading the data.

Write Linq queries for the following.

- 1) To display staff details write the following query
- 2) To display list of employee whose salary is more than 30000
Perform the following query by yourself
- 3) Display the list of student where city is not null
- 4) Display the list of student which includes Student name, department and date of birth
- 5) Display count of total student belonging to Bangalore
- 6) Display list of employees whose salary is more than the average salary of the employee.

Data Manipulation:-

Now we will CRUD operation on the model that we have create . we will use Student_Master Entity.

To ADD a record to the student master entity write the following code

Output :-

To Update a Record Add the following code

Output :-

To Delete a Record Add the following code

Output:-

To-Do Assignments

- 1) Create a Console application to perform CRUD operation . You have to perform following step.
 - a. Create a table named Employee which will have the following fields
 - i. ID
 - ii. Name
 - iii. DOB
 - iv. DOJ
 - v. Designation
 - vi. Salary
 - b. Add a Entity Data Model to project which will include above mentioned Entity.
 - c. Using LINQ to Entities write the following functionality and execute them
 - i. Add a Employee details
 - ii. Updating a Employee details
 - iii. Searching for an Employee based on It ID
 - iv. Deleteing an employee.