(T19)比較 LinqToSql 的 LazyLoading、EagerLoading
CourseGUID: 5ba9a6fe-7475-4b0c-8b99-bbcf7f5e2e1c
===
==========
(T19)比較 LinqToSql 的 LazyLoading、EagerLoading
===
==========

===
==========

0. Summary

1.

LazyLoading V.S. EagerLoading

Reference:

https://msdn.microsoft.com/en-us/library/jj574232(v=vs.113).aspx

https://stackoverflow.com/questions/97197/what-is-n1-select-query-issue

1.1.

LazyLoading

1.1.1.

We retrieve just the amount of data that we need in a single query.

When we need more data, then it issues more queries to the database.

That means we might have to request the data from database many times,

and this might cost the performance.

1.1.2.

LazyLoading might cause N+1 select problem.

E.g.

One Team can have many Gamers.

One Gamer can have one Team.

This isOne-to-Many relationship.

When we have N teams, and when we select for the Teams,

and then additional selects to retrieve the Gamers **belonging to each Team.**

That means we have to request the data from database additional N times.

This is N+1 select problem.

1.2.

EagerLoading

We retrieve all data that we need in a single query,

and then be cached to improve the application performance.

That means we just have to request the data from database once,

but this cost memory consumption.

1.3.

Conclusion

1.3.1.

If you need only Team data,

then "lazy loading" works best.

If you choose to use "Eager loading" in this case,

it will cost memory consumption.

1.3.2.

However, if you need Team data and its Gamers data,

then "Eager loading" works best.

If you choose to use "lazy loading" in this case,

it will request the data from database too many times,

this cost application performance.

==

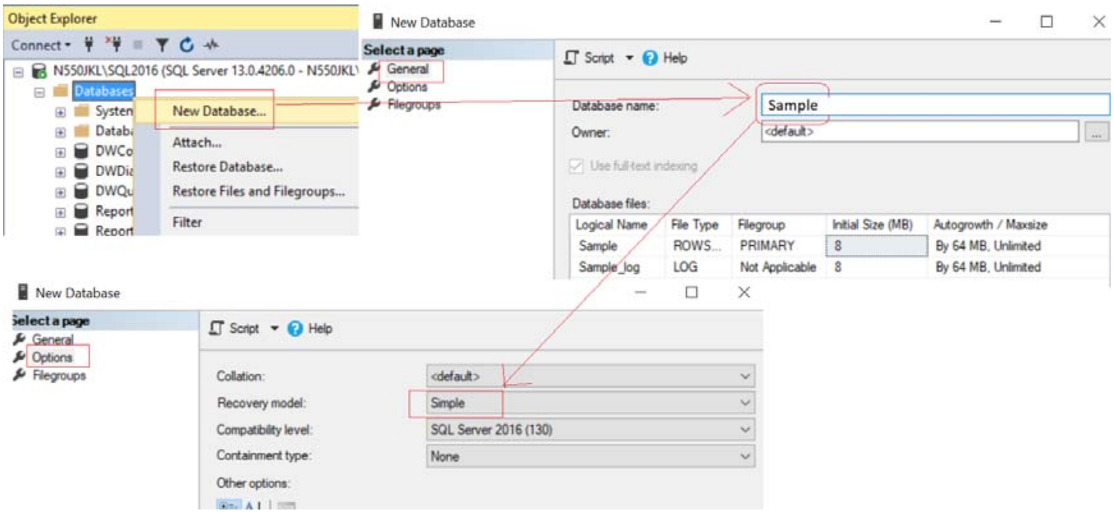
1. Web Form Application - Linq Query

1.1. TSQL

Database --> Right Click --> New Database -->

Database Name : Sample

Options --> Recovery Model : Simple



Database name: Sample

Owner: <default>

Database files:

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize
Sample	ROWS...	PRIMARY	8	By 64 MB, Unlimited
Sample_log	LOG	Not Applicable	8	By 64 MB, Unlimited

Collation: <default>

Recovery model: Simple

Compatibility level: SQL Server 2016 (130)

Containment type: None

--Create a Sample DataBase and Run the following TSQL

```
/*
```

```
1.
One Team can have many Gamers
One Gamer can have One Team.
This is One to Many Relationship.
2.
Team Id==4 has no Gamer.
Gamer Id==7 has no Team.
*/
--1 ----------------------------------------------------------
--Drop Table if it exists.
--IF OBJECT_ID('Gamer') IS NOT NULL
IF ( EXISTS ( SELECT    *
             FROM      INFORMATION_SCHEMA.TABLES
             WHERE     TABLE_NAME = 'Gamer' ) )
    BEGIN
        TRUNCATE TABLE Gamer;
        DROP TABLE Gamer;
    END;
GO -- Run the previous command and begins new batch
--Drop Table if it exists.
--IF OBJECT_ID('Team') IS NOT NULL
IF ( EXISTS ( SELECT    *
             FROM      INFORMATION_SCHEMA.TABLES
             WHERE     TABLE_NAME = 'Team' ) )
    BEGIN
        TRUNCATE TABLE Team;
        DROP TABLE Team;
    END;
GO -- Run the previous command and begins new batch
--Create Tables
CREATE TABLE Team
    (
      Id INT PRIMARY KEY
```

```
             IDENTITY ,
      Name NVARCHAR(100) ,
      Type NVARCHAR(100)
    );
GO -- Run the previous command and begins new batch
CREATE TABLE Gamer
    (
       Id INT PRIMARY KEY
             IDENTITY ,
      Name NVARCHAR(50) ,
       Gender NVARCHAR(50) ,
       Score INT ,
      Type NVARCHAR(50) ,
       TeamId INT FOREIGN KEY REFERENCES Team ( Id )
    );
GO -- Run the previous command and begins new batch
--2 ---------------------------------------------------------
--Insert Data
INSERT  INTO Team
VALUES  ( 'Team1_Guardian', 'Guardian' );
INSERT  INTO Team
VALUES  ( 'Team2_Assassinator', 'Assassinator' );
INSERT  INTO Team
VALUES  ( 'Team3_Soldier', 'Soldier' );
INSERT  INTO Team
VALUES  ( 'Team4_Civilian', 'Civilian' );
GO -- Run the previous command and begins new batch
INSERT  INTO Gamer
VALUES  ( 'Name1 ABC', 'Male', 5000, 'Water', 1 );
INSERT  INTO Gamer
VALUES  ( 'Name2 ABCDE', 'Female', 4500, 'Fire', 3 );
```

```
INSERT  INTO Gamer
VALUES  ( 'Name3 EFGH', 'Male', 6500, 'Fire', 2 );
INSERT  INTO Gamer
VALUES  ( 'Name4 HIJKLMN', 'Female', 45000, 'Water', 2 );
INSERT  INTO Gamer
VALUES  ( 'Name5 NOP', 'Male', 3000, 'Wood', 1 );
INSERT  INTO Gamer
VALUES  ( 'Name6 PQRSTUVW', 'Male', 4000, 'Earth', 3 );
INSERT  INTO Gamer
VALUES  ( 'Name7 XYZ', 'Male', 4500, 'Metal', NULL );
GO -- Run the previous command and begins new batch
```

1.2. Set up SQL Authentication

In SQL server

Object Explorer --> Security --> Logins --> New Logins

-->

General Tab

Login Name :

Tester

Password:

1234

Default Database:

Sample

-->

Server Roles Tab

Select

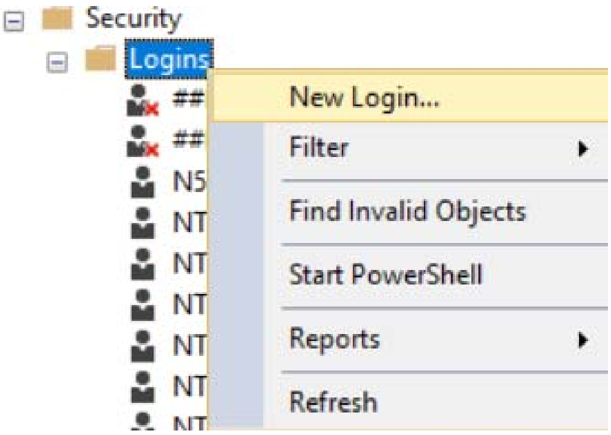
sysadmin

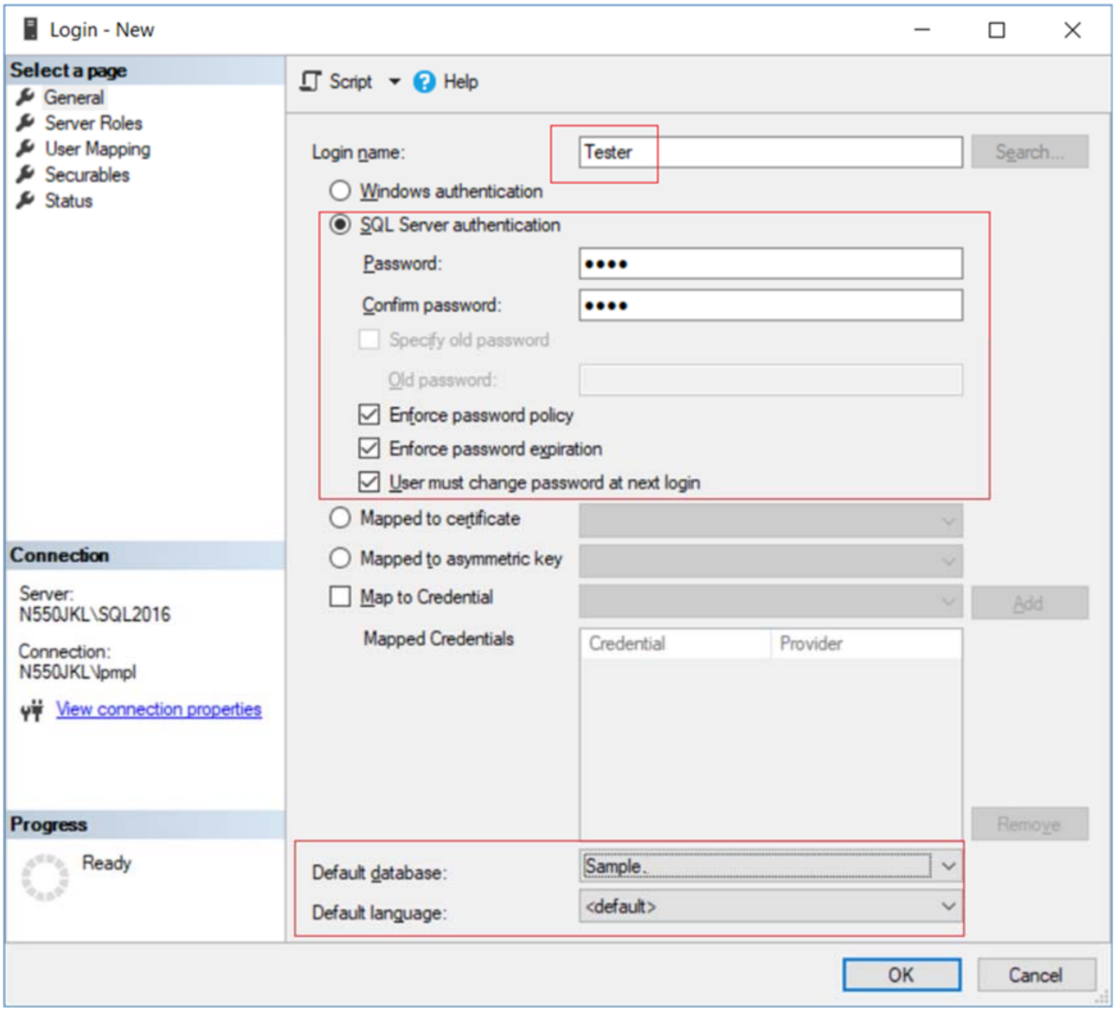
-->

User Mapping Tab

Select **Sample**

Select every Roles.





Login - New

Select a page
- General
- Server Roles
- User Mapping
- Securables
- Status

Script ▾ Help

Login name: Tester

○ Windows authentication

● SQL Server authentication

Password: ••••

Confirm password: ••••

□ Specify old password

Old password:

☑ Enforce password policy

☑ Enforce password expiration

☑ User must change password at next login

○ Mapped to certificate

○ Mapped to asymmetric key

□ Map to Credential

Mapped Credentials

Credential	Provider

Connection

Server:
N550JKL\SQL2016

Connection:
N550JKL\lpmpl

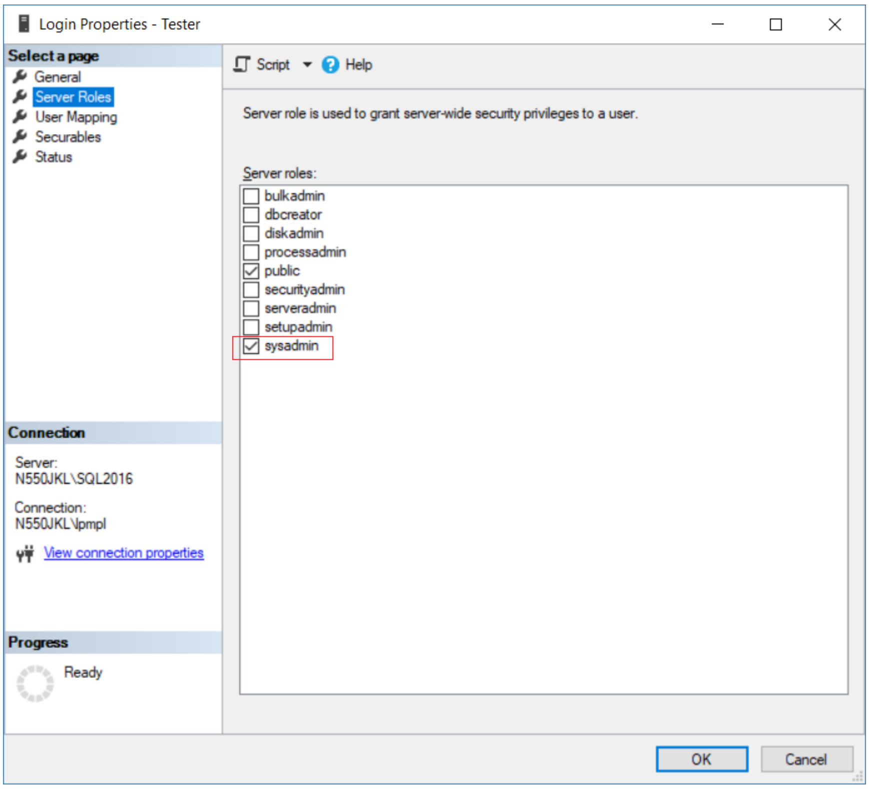
View connection properties

Progress

Ready

Default database: Sample

Default language: <default>



Server role is used to grant server-wide security privileges to a user.

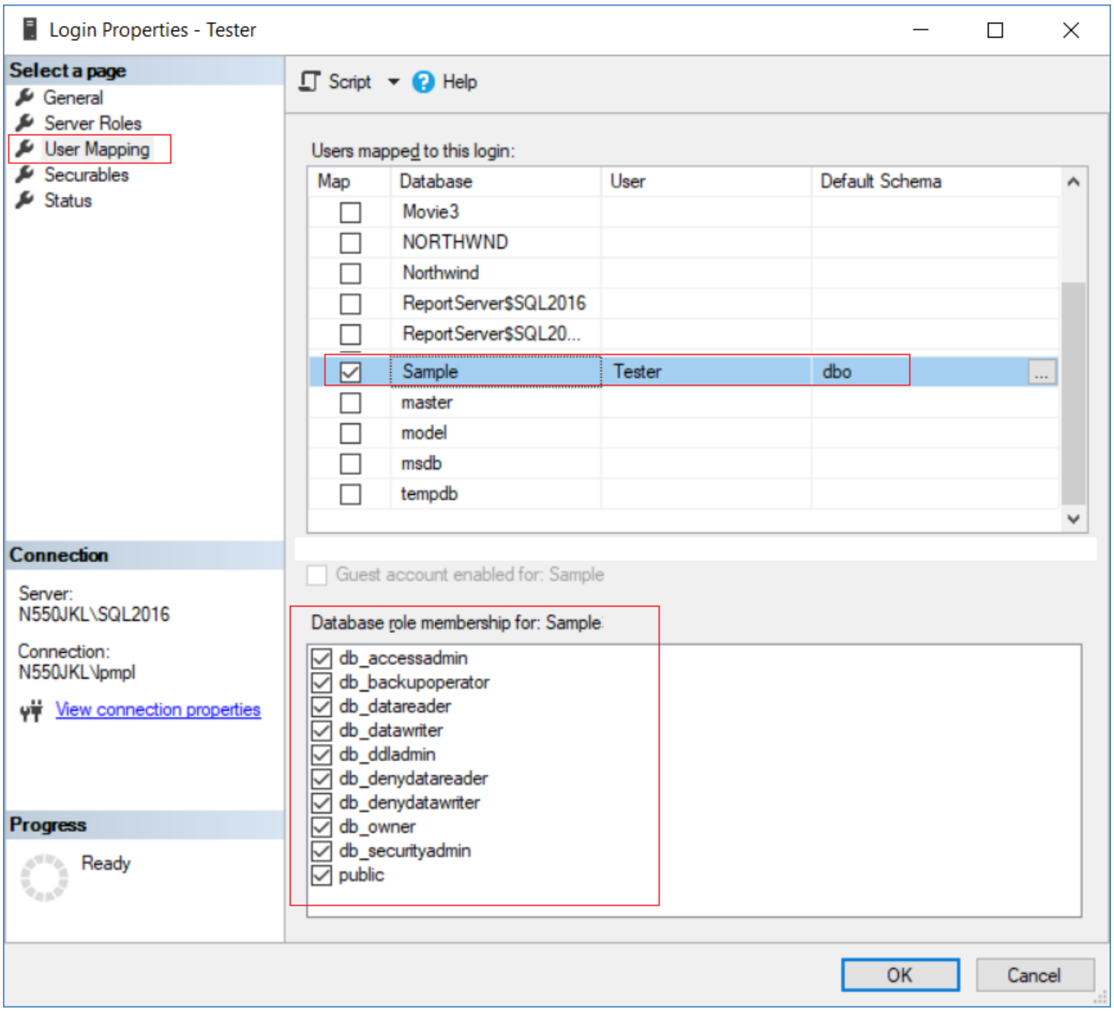
Server roles:

□ bulkadmin
□ dbcreator
□ diskadmin
□ processadmin
☑ public
□ securityadmin
□ serveradmin
□ setupadmin
☑ sysadmin

Server:
N550JKL\SQL2016

Connection:
N550JKL\pmpl

Ready



Users mapped to this login:

Map	Database	User	Default Schema
□	Movie3		
□	NORTHWND		
□	Northwind		
□	ReportServer$SQL2016		
□	ReportServer$SQL20...		
☑	Sample	Tester	dbo
□	master		
□	model		
□	msdb		
□	tempdb		

□ Guest account enabled for: Sample

Database role membership for: Sample

- ☑ db_accessadmin
- ☑ db_backupoperator
- ☑ db_datareader
- ☑ db_datawriter
- ☑ db_ddladmin
- ☑ db_denydatareader
- ☑ db_denydatawriter
- ☑ db_owner
- ☑ db_securityadmin
- ☑ public

Connection

Server:
N550JKL\SQL2016

Connection:
N550JKL\lpmpl

View connection properties

Progress

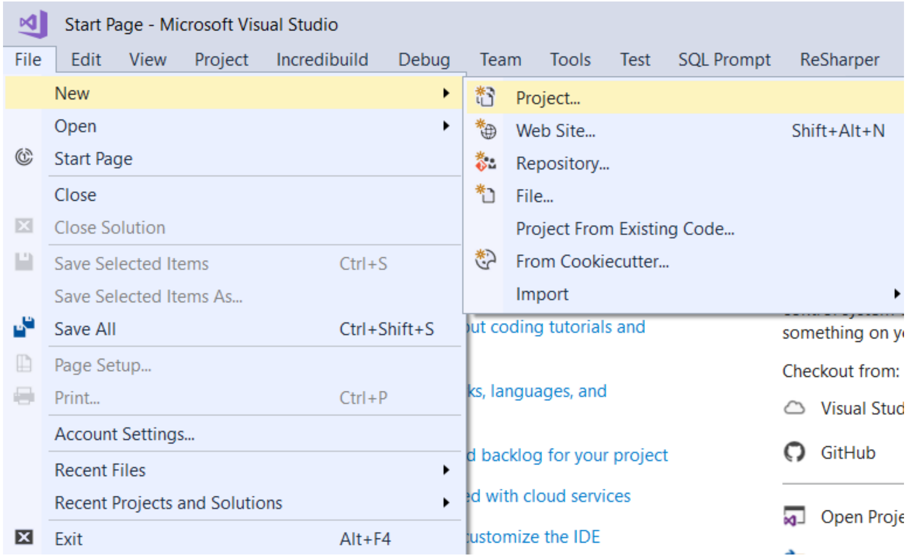
Ready

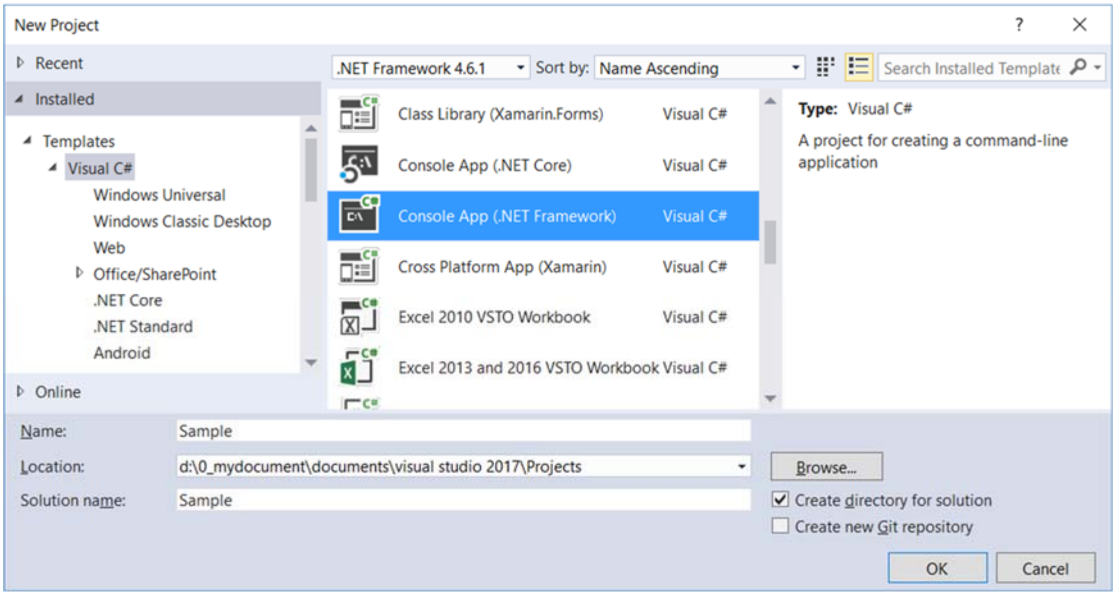
2. Console App

File --> New --> Project... -->

Visual C# --> **Console App (.Net Framework)** -->

Name: **Sample**




New Project
?
Recent
Installed
Templates
Visual C#
Windows Universal
Windows Classic Desktop
Web
Office/SharePoint
.NET Core
.NET Standard
Android
Online
.NET Framework 4.6.1
Sort by:
Name Ascending
Search Installed Template
Class Library (Xamarin.Forms) Visual C#
Console App (.NET Core) Visual C#
Console App (.NET Framework) Visual C#
Cross Platform App (Xamarin) Visual C#
Excel 2010 VSTO Workbook Visual C#
Excel 2013 and 2016 VSTO Workbook Visual C#
Browse...
OK
Cancel

Type: Visual C#

A project for creating a command-line application

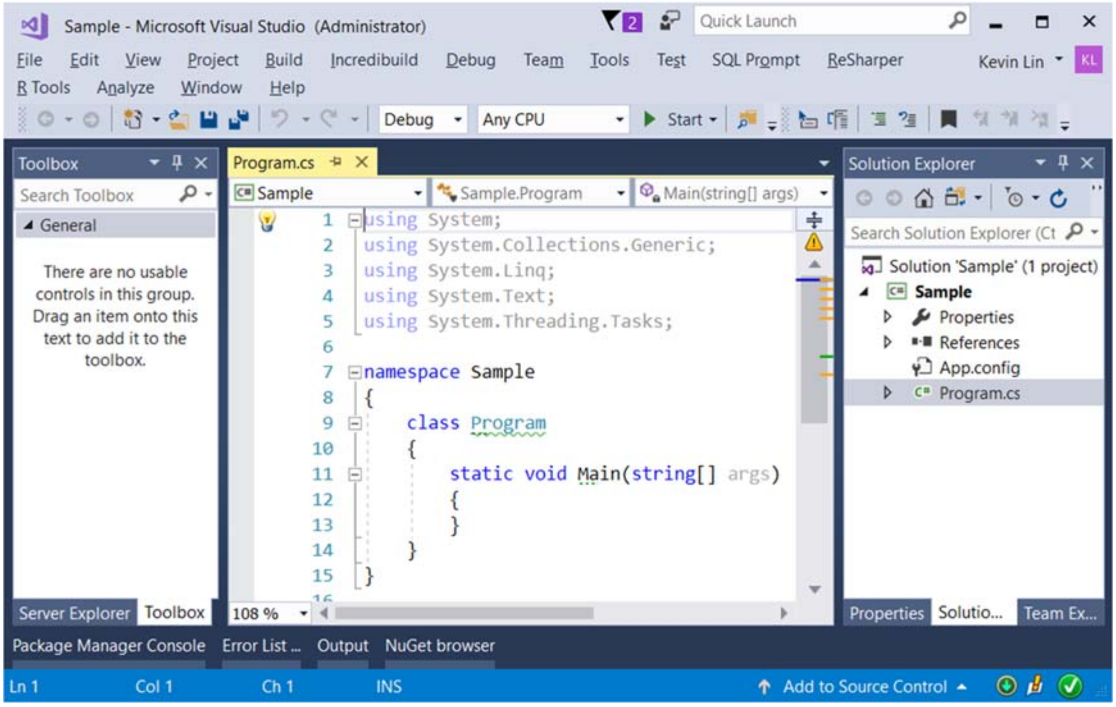
Name: Sample

Location: d:\0_mydocument\documents\visual studio 2017\Projects

Solution name: Sample

☑ Create directory for solution

□ Create new Git repository


Sample - Microsoft Visual Studio (Administrator)
Quick Launch
File Edit View Project Build Incredibuild Debug Team Tools Test SQL Prompt ReSharper
Kevin Lin KL
R Tools Analyze Window Help
Debug Any CPU Start
Toolbox
Search Toolbox
General
There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.
Server Explorer Toolbox
Program.cs
Sample Sample.Program Main(string[] args)
108 %
Solution Explorer
Search Solution Explorer (Ct
Solution 'Sample' (1 project)
Sample
Properties
References
App.config
Program.cs
Properties Solutio... Team Ex...
Package Manager Console Error List ... Output NuGet browser
Ln 1 Col 1 Ch 1 INS
Add to Source Control

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Sample
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

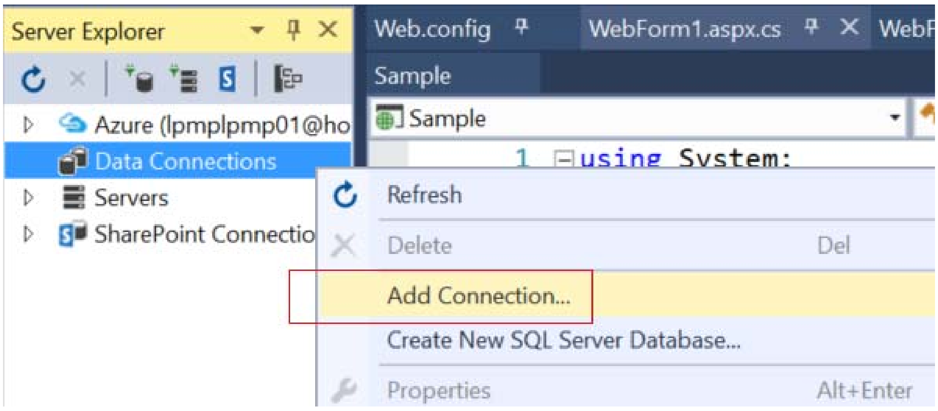
2.1. Linq to SQL

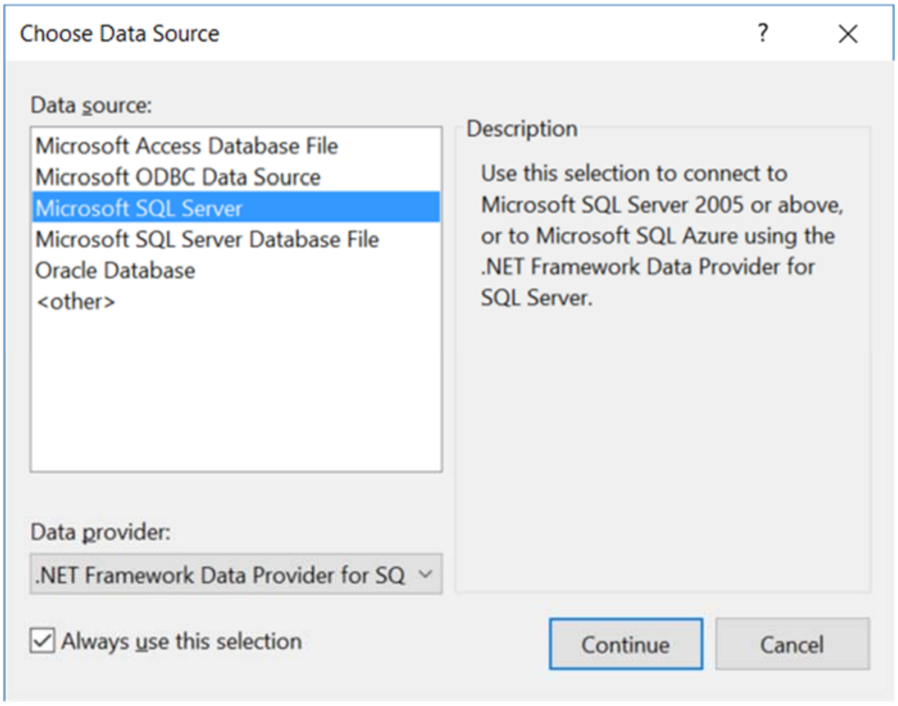
2.1.1. Add Connection

Server Explorer --> Data Connections --> Right click --> Add Connection...

--> Microsoft SQL server -->

Enter your server and database details ....





Choose Data Source

Data source:

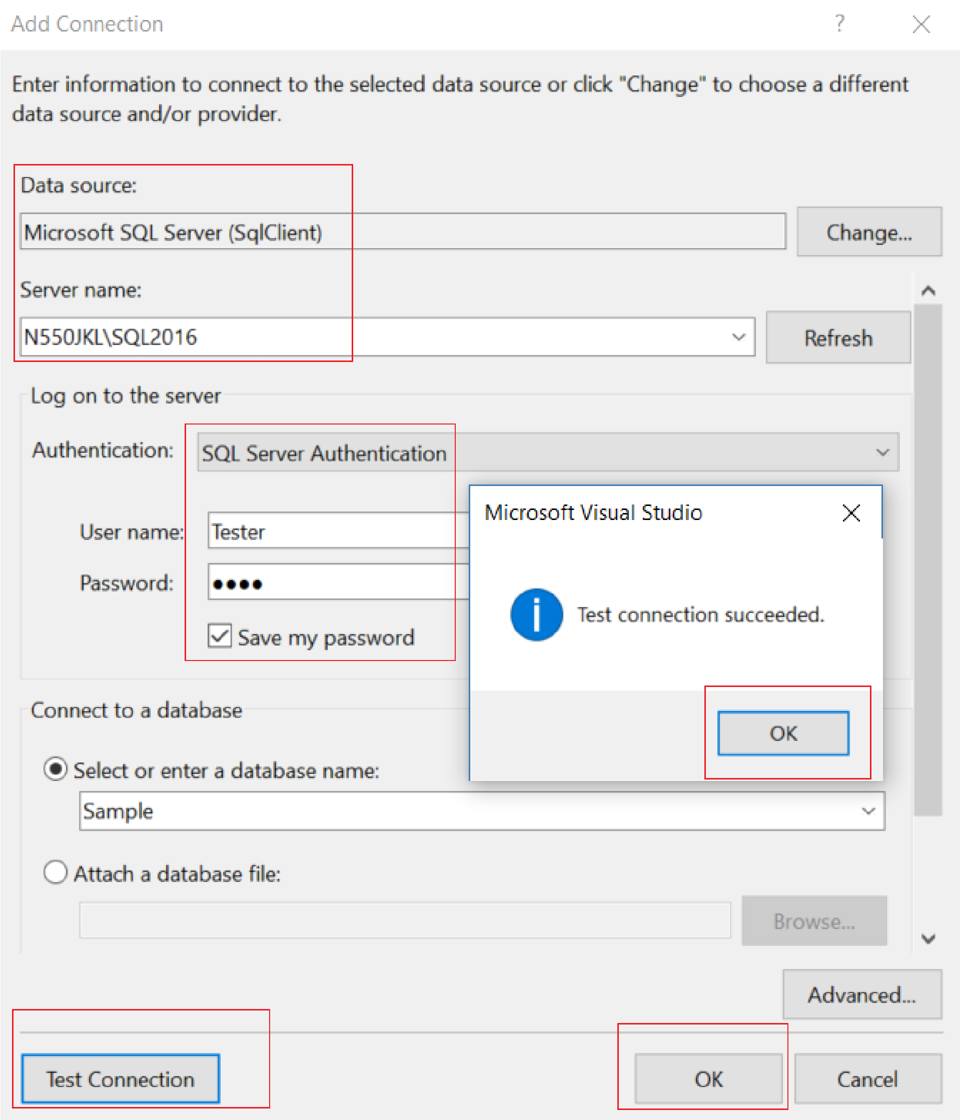
- Microsoft Access Database File
- Microsoft ODBC Data Source
- Microsoft SQL Server
- Microsoft SQL Server Database File
- Oracle Database
- <other>

Description

Use this selection to connect to Microsoft SQL Server 2005 or above, or to Microsoft SQL Azure using the .NET Framework Data Provider for SQL Server.

Data provider: .NET Framework Data Provider for SQ

☑ Always use this selection



Add Connection

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)

Server name: N550JKL\SQL2016

Log on to the server

Authentication: SQL Server Authentication

User name: Tester

Password: ●●●●

☑ Save my password

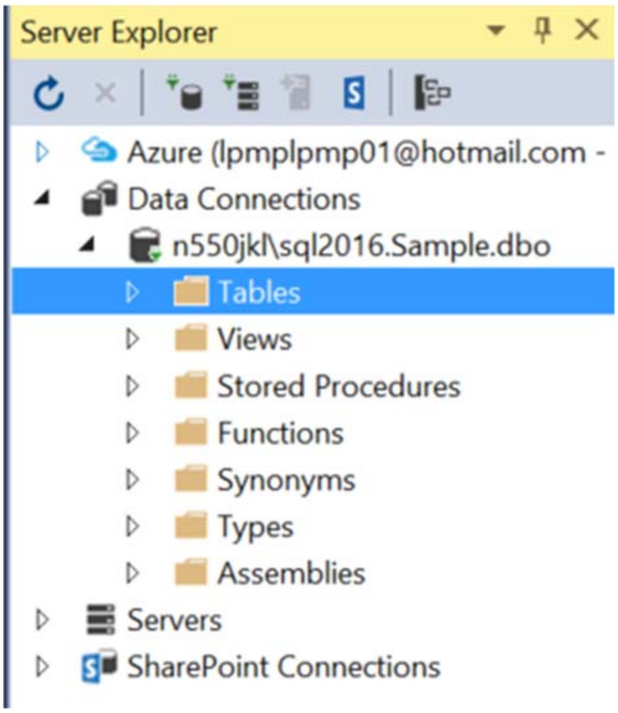
Connect to a database

● Select or enter a database name: Sample

○ Attach a database file:

Microsoft Visual Studio

Test connection succeeded.



2.1.2. Sample.dbml

ProjectName --> Right Click --> Add --> New Item...

--> Linq to SQL classes -->

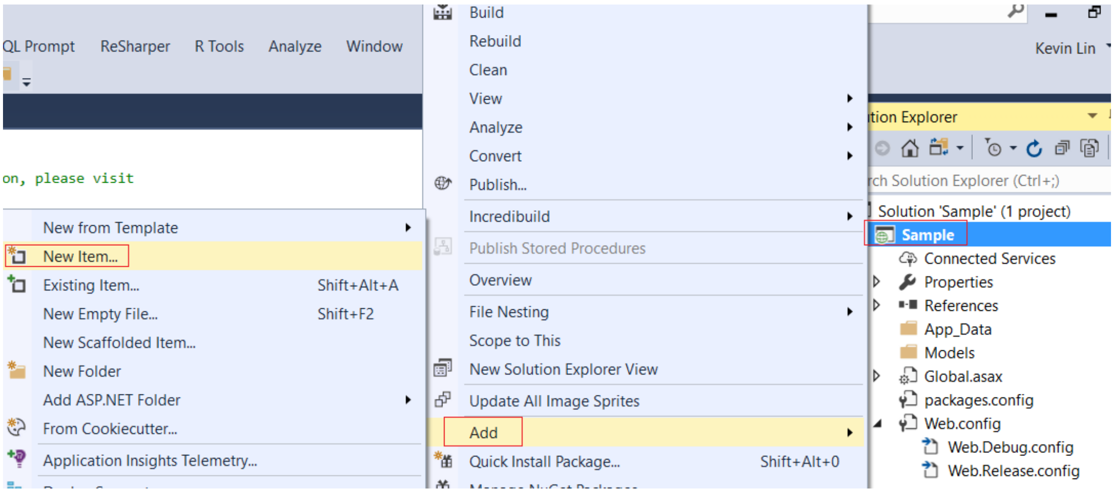
Name : **Sample.dbml**

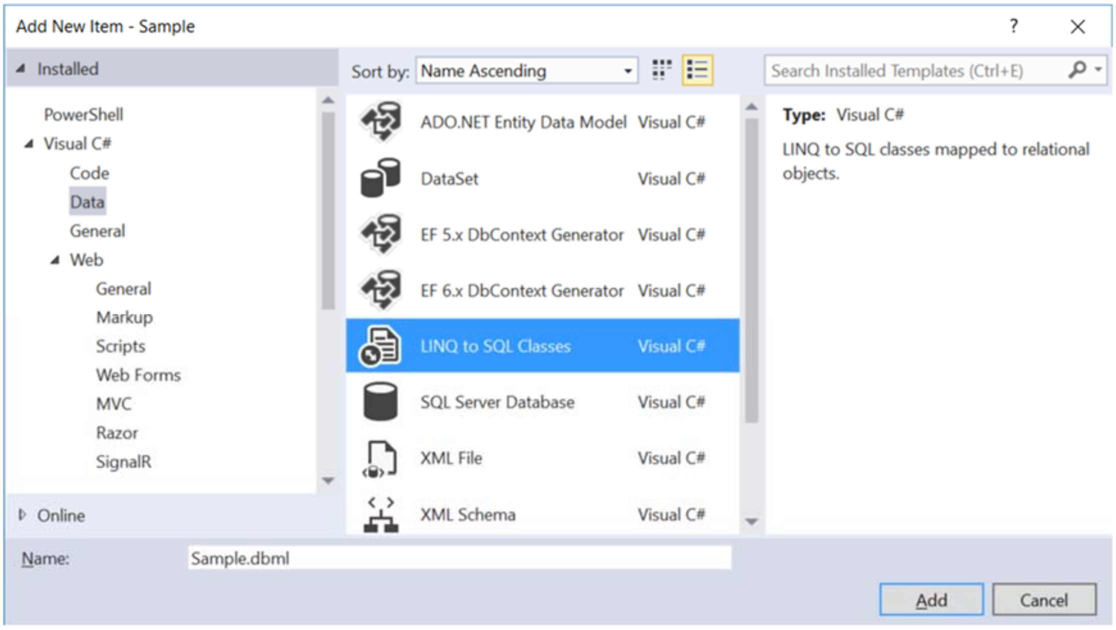
I name it as "Sample.dbml",

because I know this is for connection to "Sample" Database.

-->

Drag Table from Server Explorer into DBML



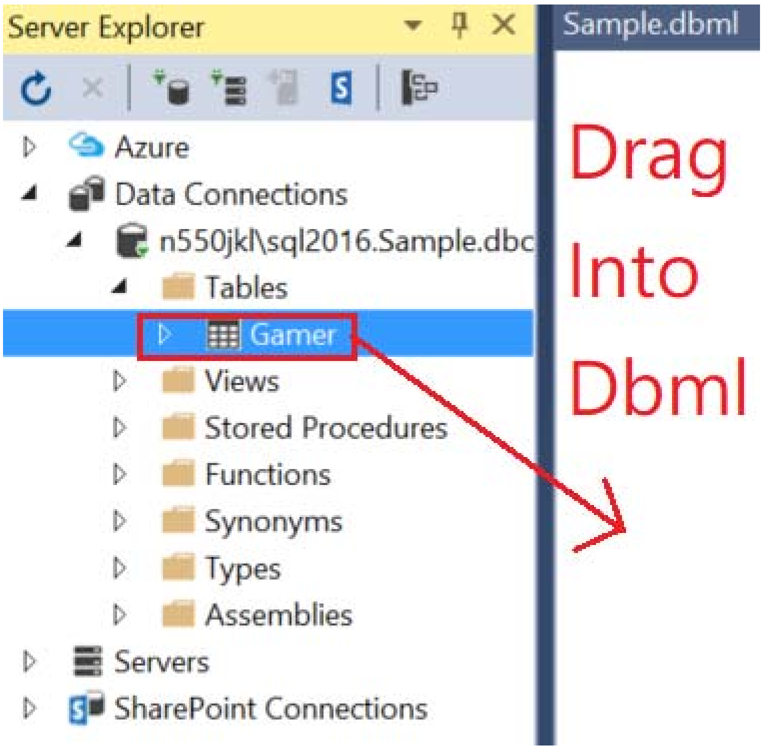

Add New Item - Sample
?
Installed
PowerShell
Visual C#
Code
Data
General
Web
General
Markup
Scripts
Web Forms
MVC
Razor
SignalR
Online
ADO.NET Entity Data Model Visual C#
DataSet Visual C#
EF 5.x DbContext Generator Visual C#
EF 6.x DbContext Generator Visual C#
LINQ to SQL Classes Visual C#
SQL Server Database Visual C#
XML File Visual C#
XML Schema Visual C#
Search Installed Templates (Ctrl+E)
Add
Cancel

Sort by: Name Ascending

Type: Visual C#

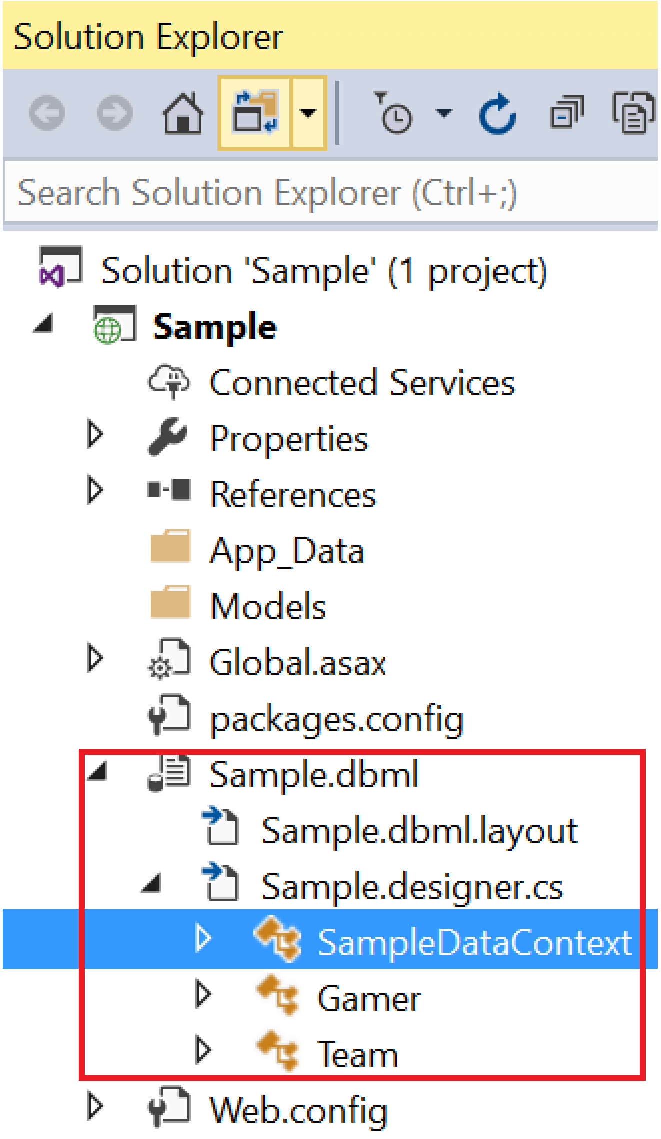
LINQ to SQL classes mapped to relational objects.

Name: Sample.dbml



Save the dbml, it will generate the following files.

The DataContext context is the entry point to database.

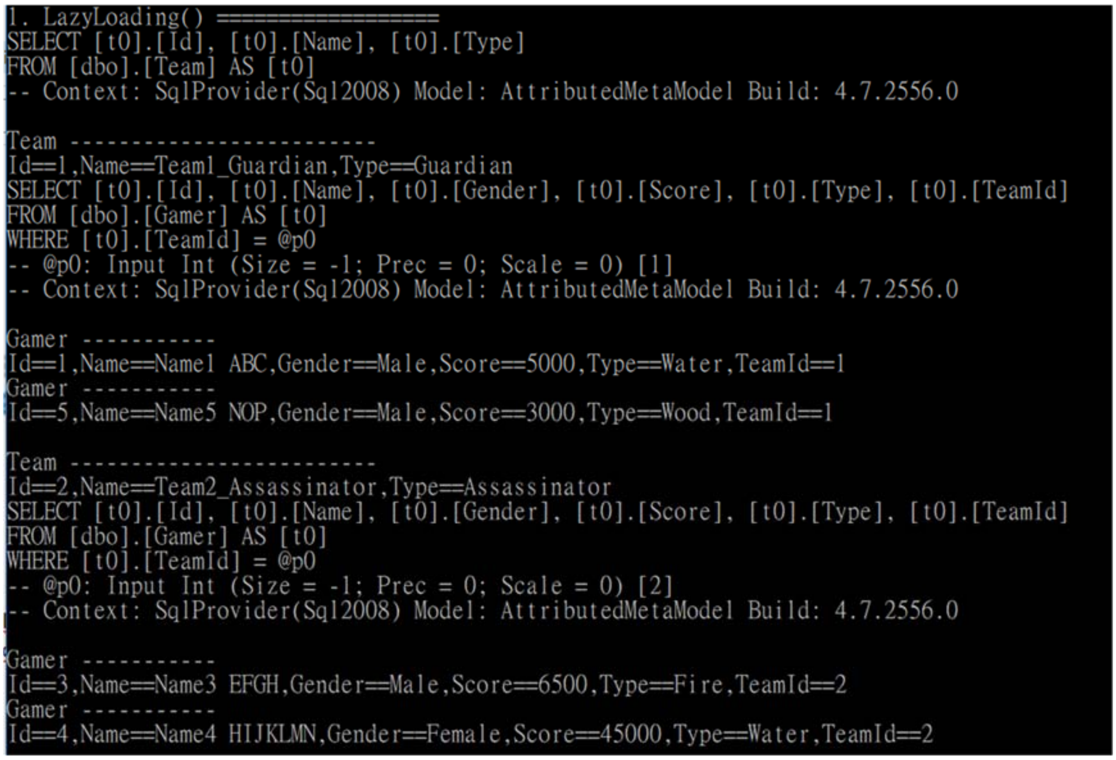


2.2. Program.cs

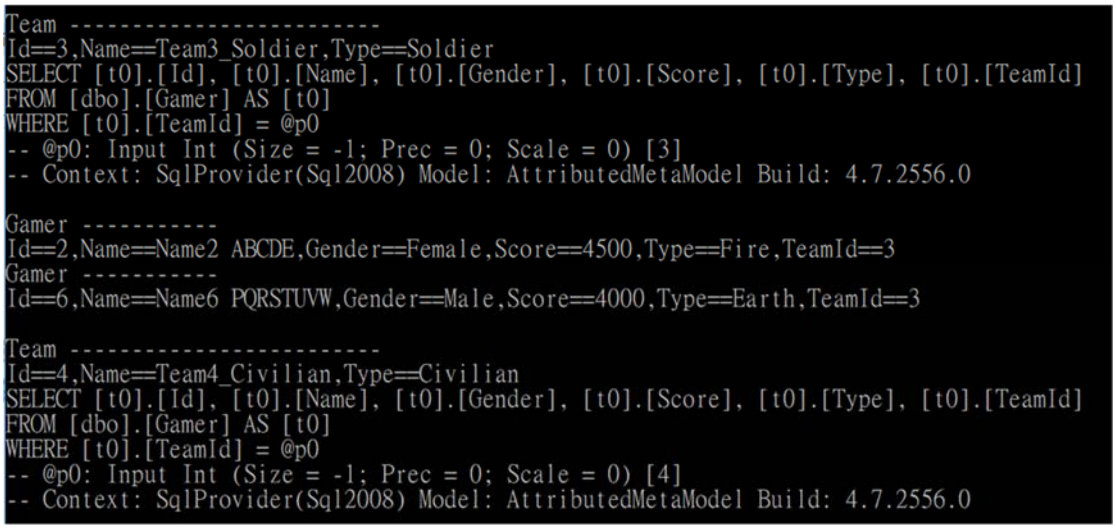
```
using System;
using System.Collections.Generic;
using System.Data.Linq;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Sample
{
```

```
class Program
 {
    static void Main(string[] args)
     {
        // 1. ===================================
        //LazyLoading()
        Console.WriteLine("1. LazyLoading() ================== ");
         LazyLoading();
        // 2. ===================================
        //EagerLoading()
        Console.WriteLine("2. EagerLoading() ================== ");
         EagerLoading();
        // 3. ===================================
        //EagerLoading2()
        Console.WriteLine("3. EagerLoading2() ================== ");
         EagerLoading2();
        Console.ReadLine();
     }
    // 1. ===================================
    //LazyLoading()
    static void LazyLoading()
     {
        using (SampleDataContext dbContext = new SampleDataContext())
         {
            //Write the generated sql query to the Console window
             dbContext.Log = Console.Out;
            foreach (Team team in dbContext.Teams)
             {
                Console.WriteLine("Team -------------------------");
                Console.WriteLine(team);
                foreach (Gamer gamer in team.Gamers)
                 {
```

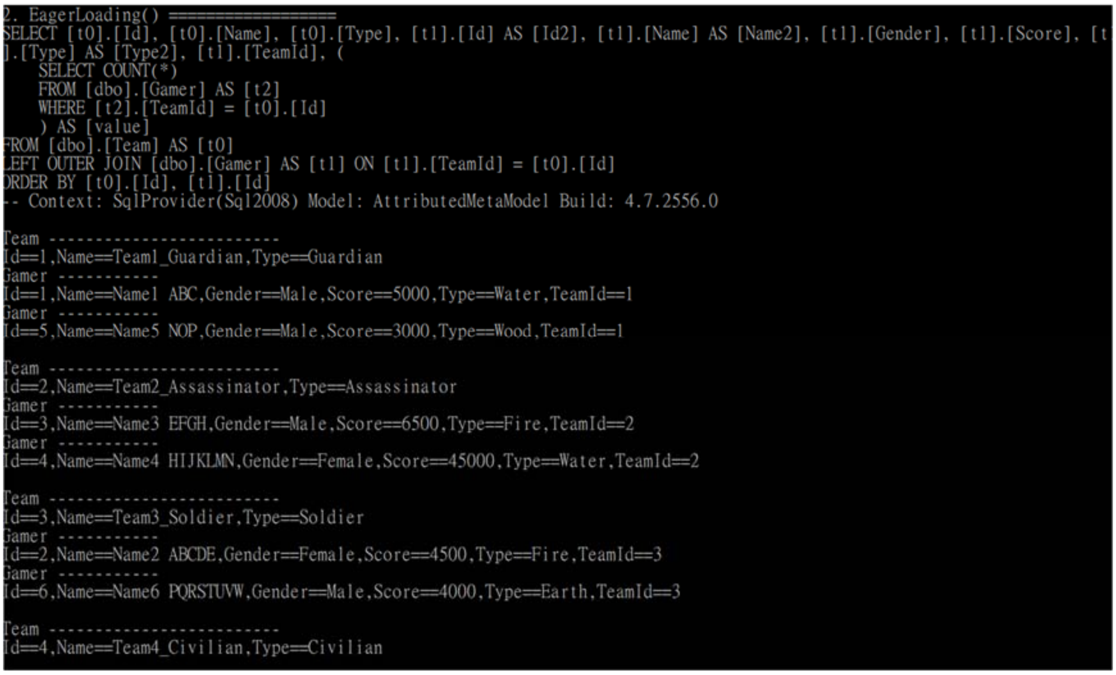
```
                Console.WriteLine("Gamer -----------");
                Console.WriteLine(gamer);
            }
            Console.WriteLine();
        }
    }
}
// 2. ===================================
//EagerLoading()
static void EagerLoading()
{
    using (SampleDataContext dbContext = new SampleDataContext())
    {
        //Write the generated sql query to the Console window
        dbContext.Log = Console.Out;
        // Load related Employee entities along with the Department entity
        DataLoadOptions loadOptions = new DataLoadOptions();
        loadOptions.LoadWith<Team>(t => t.Gamers);
        dbContext.LoadOptions = loadOptions;
        foreach (Team team in dbContext.Teams)
        {
            Console.WriteLine("Team -------------------------");
            Console.WriteLine(team);
            foreach (Gamer gamer in team.Gamers)
            {
                Console.WriteLine("Gamer -----------");
                Console.WriteLine(gamer);
            }
            Console.WriteLine();
        }
    }
}
// 3. ===================================
        //EagerLoading2()
        static void EagerLoading2()
        {
            using (SampleDataContext dbContext = new SampleDataContext())
            {
                //Write the generated sql query to the Console window
                dbContext.Log = Console.Out;
                var linqQuery = from team in dbContext.Teams
                                select new { Team = team, Gamers = team.Gamers };
                foreach (var linqQueryItem in linqQuery)
                {
                    Console.WriteLine("Team -------------------------");
                    Console.WriteLine(linqQueryItem.Team);
                    foreach (Gamer gamer in linqQueryItem.Gamers)
                    {
                        Console.WriteLine("Gamer -----------");
                        Console.WriteLine(gamer);
                    }
                    Console.WriteLine();
                }
            }
        }
    }
    public partial class Gamer
    {
        public override string ToString()
        {
            return $"Id=={Id},Name=={Name},Gender=={Gender},Score=={Score},Type=={Type},TeamId=={TeamId}";
        }
    }
    public partial class Team
    {
        public override string ToString()
        {
            return $"Id=={Id},Name=={Name},Type=={Type}";
        }
    }
}
```



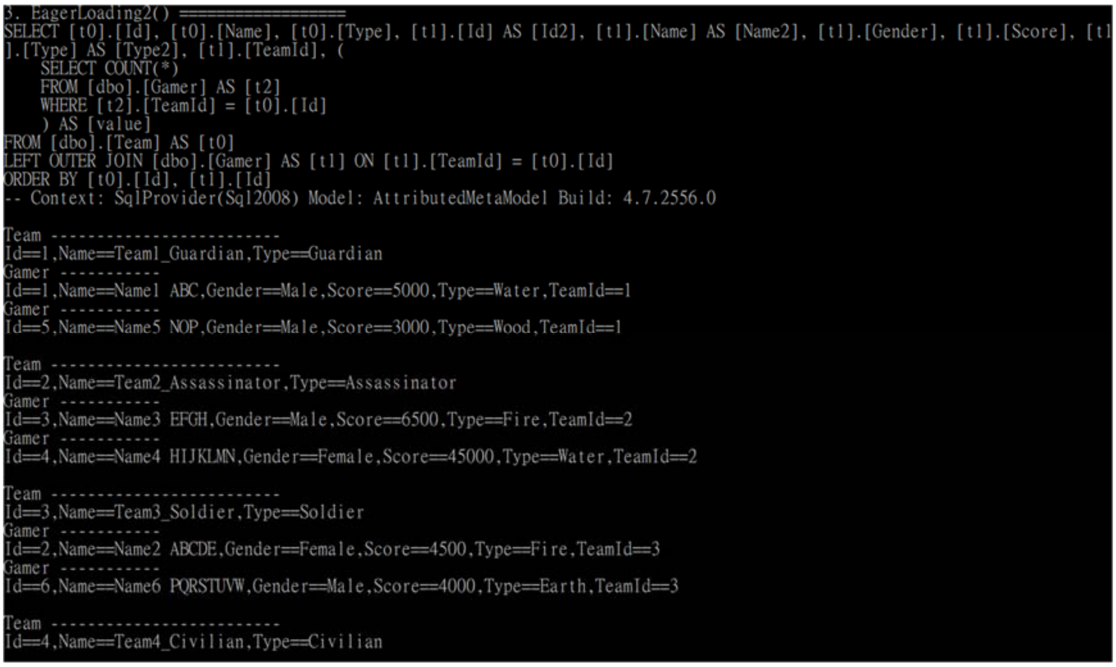
```
1. LazyLoading() ==================
SELECT [t0].[Id], [t0].[Name], [t0].[Type]
FROM [dbo].[Team] AS [t0]
-- Context: SqlProvider(Sql2008) Model: AttributedMetaModel Build: 4.7.2556.0

Team -------------------------
Id==1,Name==Team1_Guardian,Type==Guardian
SELECT [t0].[Id], [t0].[Name], [t0].[Gender], [t0].[Score], [t0].[Type], [t0].[TeamId]
FROM [dbo].[Gamer] AS [t0]
WHERE [t0].[TeamId] = @p0
-- @p0: Input Int (Size = -1; Prec = 0; Scale = 0) [1]
-- Context: SqlProvider(Sql2008) Model: AttributedMetaModel Build: 4.7.2556.0

Gamer -----------
Id==1,Name==Name1 ABC,Gender==Male,Score==5000,Type==Water,TeamId==1
Gamer -----------
Id==5,Name==Name5 NOP,Gender==Male,Score==3000,Type==Wood,TeamId==1

Team -------------------------
Id==2,Name==Team2_Assassinator,Type==Assassinator
SELECT [t0].[Id], [t0].[Name], [t0].[Gender], [t0].[Score], [t0].[Type], [t0].[TeamId]
FROM [dbo].[Gamer] AS [t0]
WHERE [t0].[TeamId] = @p0
-- @p0: Input Int (Size = -1; Prec = 0; Scale = 0) [2]
-- Context: SqlProvider(Sql2008) Model: AttributedMetaModel Build: 4.7.2556.0

Gamer -----------
Id==3,Name==Name3 EFGH,Gender==Male,Score==6500,Type==Fire,TeamId==2
Gamer -----------
Id==4,Name==Name4 HIJKLMN,Gender==Female,Score==45000,Type==Water,TeamId==2
```



```
Team -------------------------
Id==3,Name==Team3_Soldier,Type==Soldier
SELECT [t0].[Id], [t0].[Name], [t0].[Gender], [t0].[Score], [t0].[Type], [t0].[TeamId]
FROM [dbo].[Gamer] AS [t0]
WHERE [t0].[TeamId] = @p0
-- @p0: Input Int (Size = -1; Prec = 0; Scale = 0) [3]
-- Context: SqlProvider(Sql2008) Model: AttributedMetaModel Build: 4.7.2556.0

Gamer -----------
Id==2,Name==Name2 ABCDE,Gender==Female,Score==4500,Type==Fire,TeamId==3
Gamer -----------
Id==6,Name==Name6 PQRSTUVW,Gender==Male,Score==4000,Type==Earth,TeamId==3

Team -------------------------
Id==4,Name==Team4_Civilian,Type==Civilian
SELECT [t0].[Id], [t0].[Name], [t0].[Gender], [t0].[Score], [t0].[Type], [t0].[TeamId]
FROM [dbo].[Gamer] AS [t0]
WHERE [t0].[TeamId] = @p0
-- @p0: Input Int (Size = -1; Prec = 0; Scale = 0) [4]
-- Context: SqlProvider(Sql2008) Model: AttributedMetaModel Build: 4.7.2556.0
```



```
2. EagerLoading() ================
SELECT [t0].[Id], [t0].[Name], [t0].[Type], [t1].[Id] AS [Id2], [t1].[Name] AS [Name2], [t1].[Gender], [t1].[Score], [t
].[Type] AS [Type2], [t1].[TeamId], (
    SELECT COUNT(*)
    FROM [dbo].[Gamer] AS [t2]
    WHERE [t2].[TeamId] = [t0].[Id]
    ) AS [value]
FROM [dbo].[Team] AS [t0]
LEFT OUTER JOIN [dbo].[Gamer] AS [t1] ON [t1].[TeamId] = [t0].[Id]
ORDER BY [t0].[Id], [t1].[Id]
-- Context: SqlProvider(Sql2008) Model: AttributedMetaModel Build: 4.7.2556.0

Team ------------------------
Id==1,Name==Team1_Guardian,Type==Guardian
Gamer -----------
Id==1,Name==Name1 ABC,Gender==Male,Score==5000,Type==Water,TeamId==1
Gamer -----------
Id==5,Name==Name5 NOP,Gender==Male,Score==3000,Type==Wood,TeamId==1

Team ------------------------
Id==2,Name==Team2_Assassinator,Type==Assassinator
Gamer -----------
Id==3,Name==Name3 EFGH,Gender==Male,Score==6500,Type==Fire,TeamId==2
Gamer -----------
Id==4,Name==Name4 HIJKLMN,Gender==Female,Score==45000,Type==Water,TeamId==2

Team ------------------------
Id==3,Name==Team3_Soldier,Type==Soldier
Gamer -----------
Id==2,Name==Name2 ABCDE,Gender==Female,Score==4500,Type==Fire,TeamId==3
Gamer -----------
Id==6,Name==Name6 PQRSTUVW,Gender==Male,Score==4000,Type==Earth,TeamId==3

Team ------------------------
Id==4,Name==Team4_Civilian,Type==Civilian
```



```
3. EagerLoading2() ================
SELECT [t0].[Id], [t0].[Name], [t0].[Type], [t1].[Id] AS [Id2], [t1].[Name] AS [Name2], [t1].[Gender], [t1].[Score], [t1
].[Type] AS [Type2], [t1].[TeamId], (
    SELECT COUNT(*)
    FROM [dbo].[Gamer] AS [t2]
    WHERE [t2].[TeamId] = [t0].[Id]
    ) AS [value]
FROM [dbo].[Team] AS [t0]
LEFT OUTER JOIN [dbo].[Gamer] AS [t1] ON [t1].[TeamId] = [t0].[Id]
ORDER BY [t0].[Id], [t1].[Id]
-- Context: SqlProvider(Sql2008) Model: AttributedMetaModel Build: 4.7.2556.0

Team ------------------------
Id==1,Name==Team1_Guardian,Type==Guardian
Gamer -----------
Id==1,Name==Name1 ABC,Gender==Male,Score==5000,Type==Water,TeamId==1
Gamer -----------
Id==5,Name==Name5 NOP,Gender==Male,Score==3000,Type==Wood,TeamId==1

Team ------------------------
Id==2,Name==Team2_Assassinator,Type==Assassinator
Gamer -----------
Id==3,Name==Name3 EFGH,Gender==Male,Score==6500,Type==Fire,TeamId==2
Gamer -----------
Id==4,Name==Name4 HIJKLMN,Gender==Female,Score==45000,Type==Water,TeamId==2

Team ------------------------
Id==3,Name==Team3_Soldier,Type==Soldier
Gamer -----------
Id==2,Name==Name2 ABCDE,Gender==Female,Score==4500,Type==Fire,TeamId==3
Gamer -----------
Id==6,Name==Name6 PQRSTUVW,Gender==Male,Score==4000,Type==Earth,TeamId==3

Team ------------------------
Id==4,Name==Team4_Civilian,Type==Civilian
```

3. Web Form App

Open Visual Studio, I am currently using VS2017

If you don't have it, you may following the instruction here to download.

http://ithandyguytutorial.blogspot.com/2017/10/ch00install-visual-studio-2017-offline.html

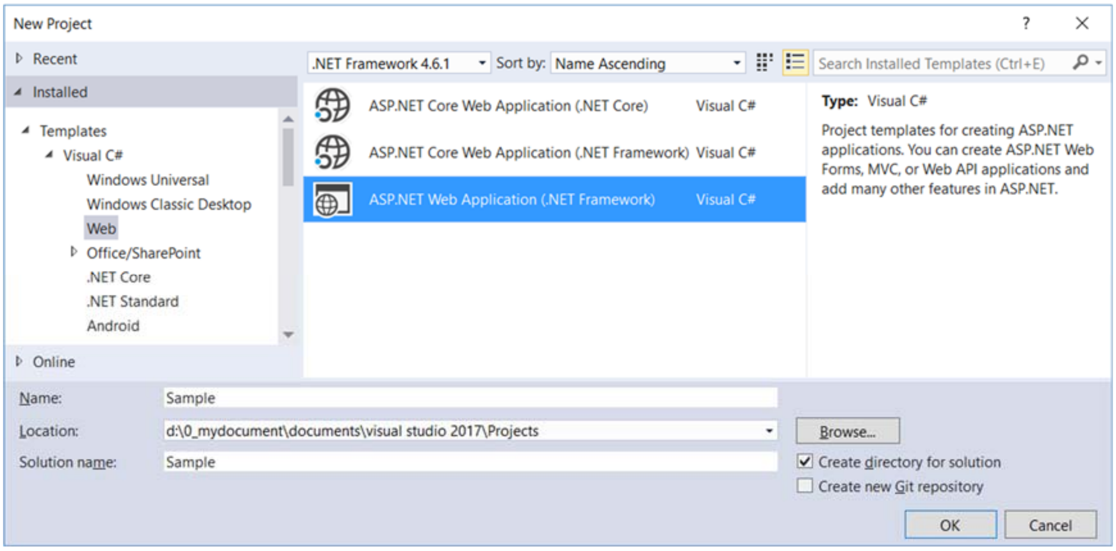
New Project --> Web -->ASP.NET**Web Application (.Net Framework)**

-->

Name:

Sample

--> **Empty** --> Select "**Web Forms**" --> OK



New Project

- Recent
- Installed
  - Templates
    - Visual C#
      - Windows Universal
      - Windows Classic Desktop
      - Web
      - Office/SharePoint
      - .NET Core
      - .NET Standard
      - Android
- Online

.NET Framework 4.6.1 | Sort by: Name Ascending | Search Installed Templates (Ctrl+E)

ASP.NET Core Web Application (.NET Core)	Visual C#
ASP.NET Core Web Application (.NET Framework)	Visual C#
ASP.NET Web Application (.NET Framework)	Visual C#

Type: Visual C#

Project templates for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.

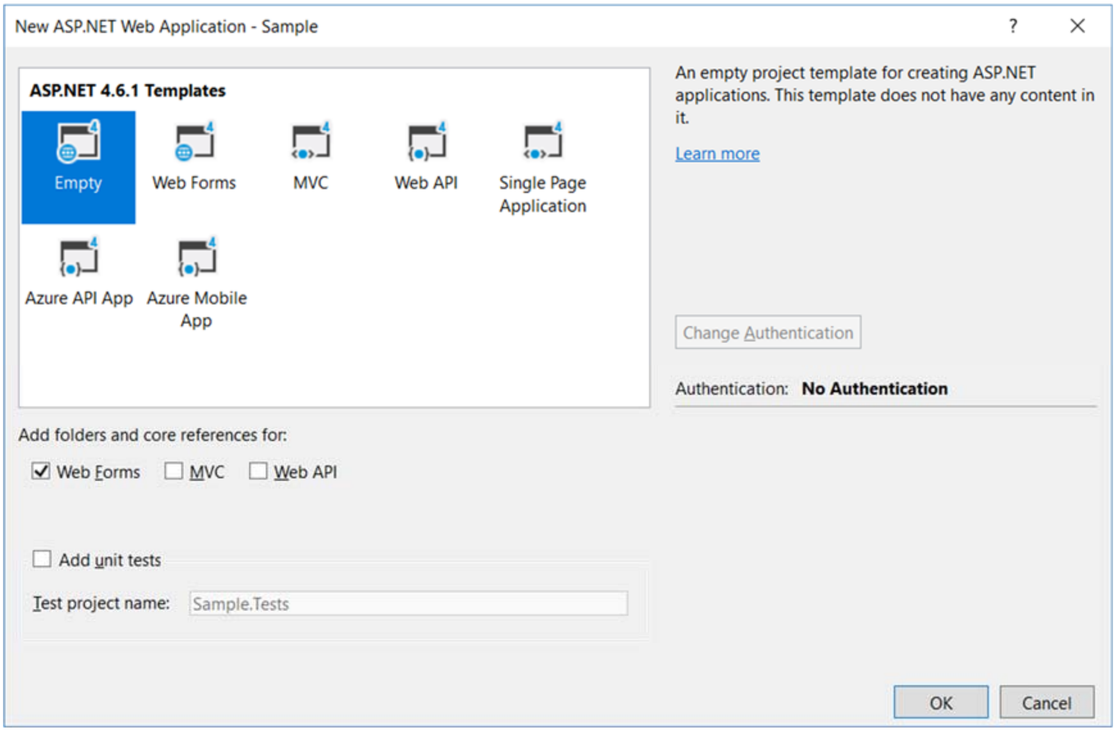
Name: Sample

Location: d:\0_mydocument\documents\visual studio 2017\Projects

Solution name: Sample

☑ Create directory for solution

□ Create new Git repository



New ASP.NET Web Application - Sample

ASP.NET 4.6.1 Templates

An empty project template for creating ASP.NET applications. This template does not have any content in it.

Learn more

Change Authentication

Authentication: **No Authentication**

Add folders and core references for:

☑ Web Forms □ MVC □ Web API

□ Add unit tests

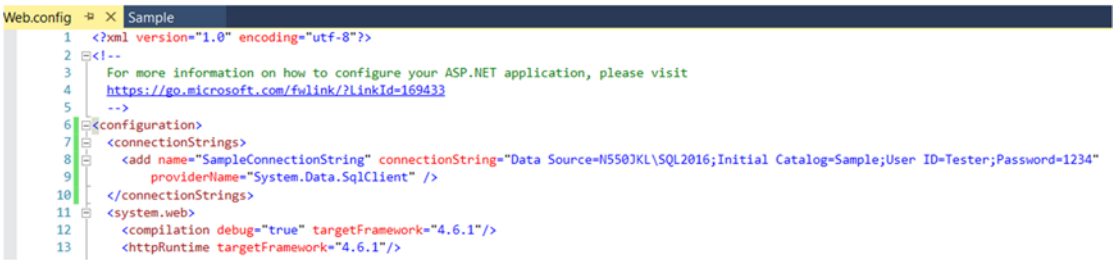
Test project name: Sample.Tests

3.1. Web.config

Add connection String

If you use Linq to Sql, you don't have to set this connection string.

```
<configuration>
  <connectionStrings>
    <add name="SampleConnectionString" connectionString="Data Source=N550JKL\SQL2016;Initial Catalog=Sample;User ID=Tester;Password=1234"
        providerName="System.Data.SqlClient" />
  </connectionStrings>
```



```
<?xml version="1.0" encoding="utf-8"?>
<!--
  For more information on how to configure your ASP.NET application, please visit
  https://go.microsoft.com/fwlink/?LinkId=169433
  -->
<configuration>
  <connectionStrings>
    <add name="SampleConnectionString" connectionString="Data Source=N550JKL\SQL2016;Initial Catalog=Sample;User ID=Tester;Password=1234"
        providerName="System.Data.SqlClient" />
  </connectionStrings>
  <system.web>
    <compilation debug="true" targetFramework="4.6.1"/>
    <httpRuntime targetFramework="4.6.1"/>
```

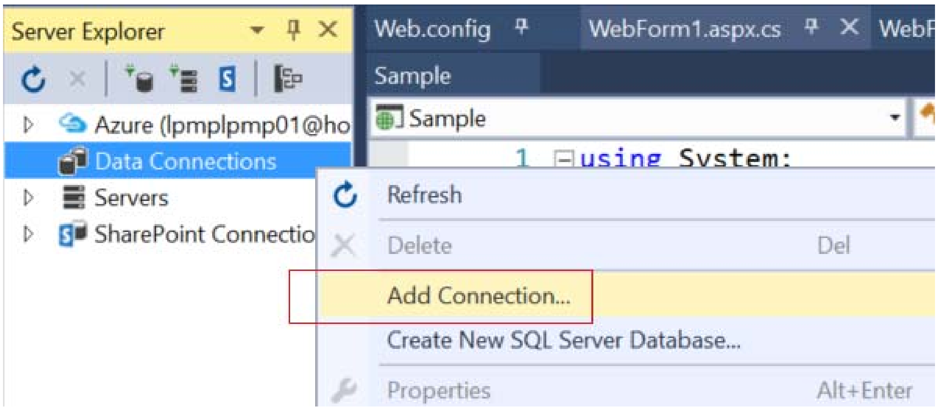
3.2. Linq to SQL

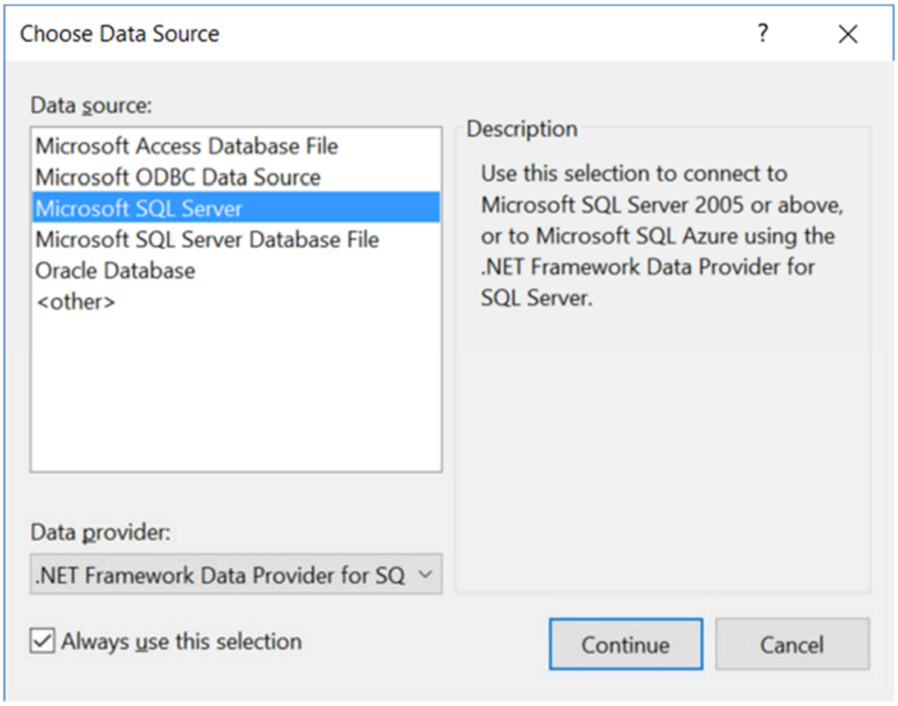
3.2.1. Add Connection

Server Explorer --> Data Connections --> Right click --> Add Connection...

--> Microsoft SQL server -->

Enter your server and database details ....



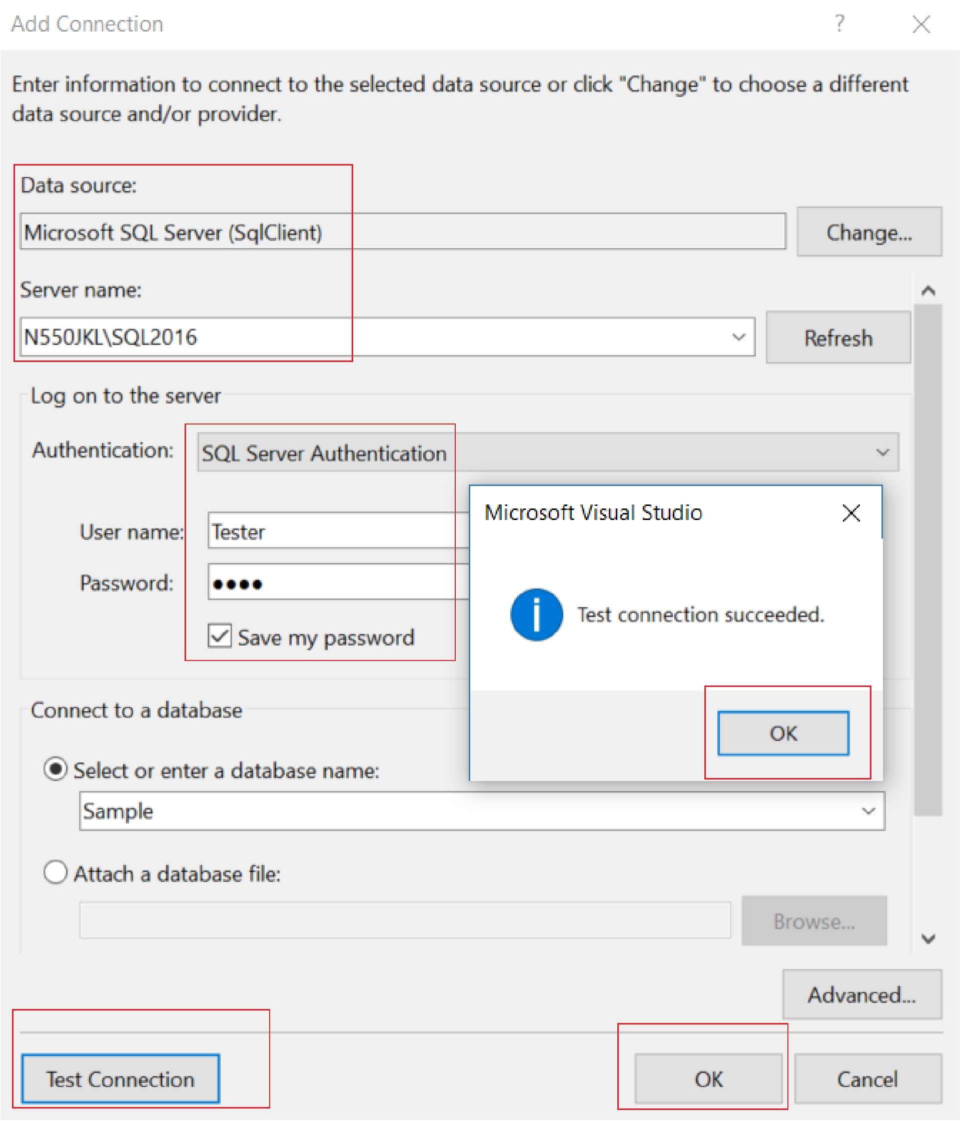


Use this selection to connect to Microsoft SQL Server 2005 or above, or to Microsoft SQL Azure using the .NET Framework Data Provider for SQL Server.

Data source: Microsoft SQL Server

Data provider: .NET Framework Data Provider for SQ

☑ Always use this selection



Add Connection

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)

Server name: N550JKL\SQL2016

Log on to the server

Authentication: SQL Server Authentication

User name: Tester

Password: ●●●●

☑ Save my password

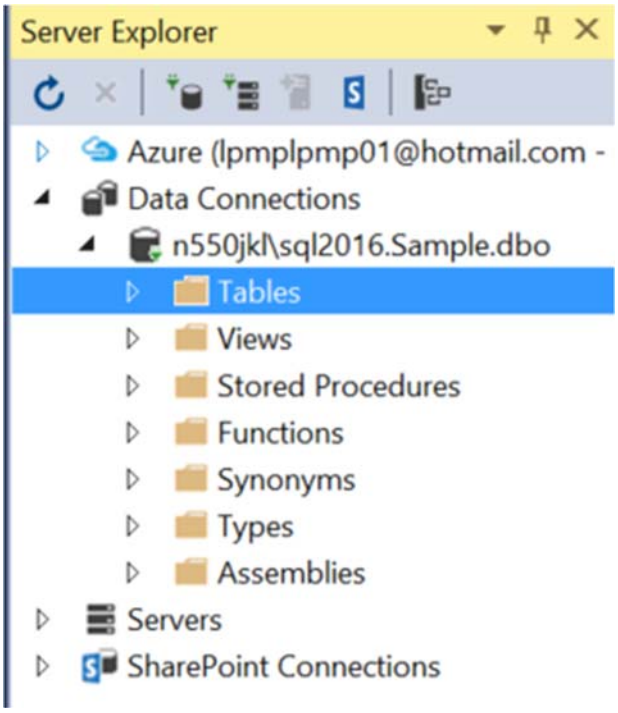
Connect to a database

● Select or enter a database name: Sample

○ Attach a database file:

Microsoft Visual Studio

Test connection succeeded.



3.2.2. Sample.dbml

ProjectName --> Right Click --> Add --> New Item...

--> Linq to SQL classes -->

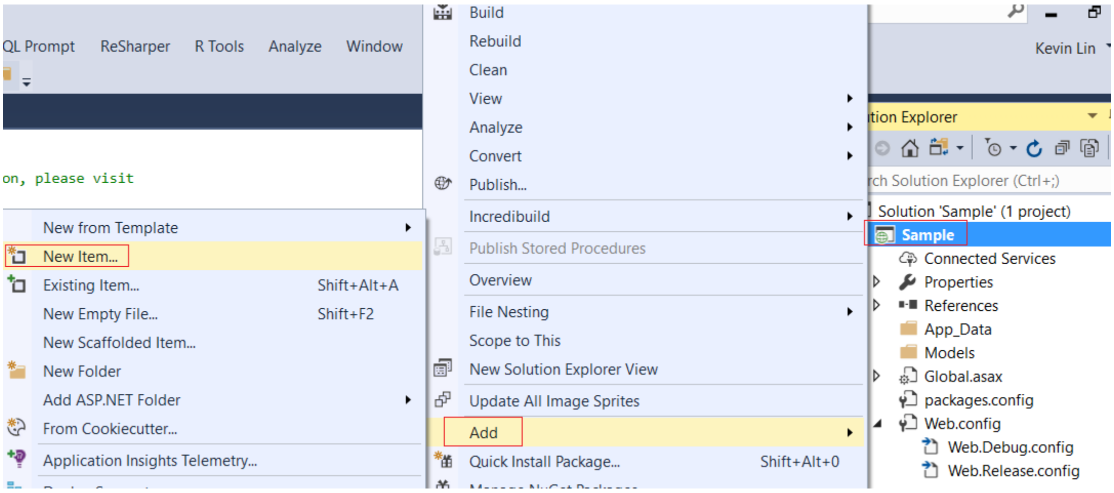
Name : **Sample.dbml**

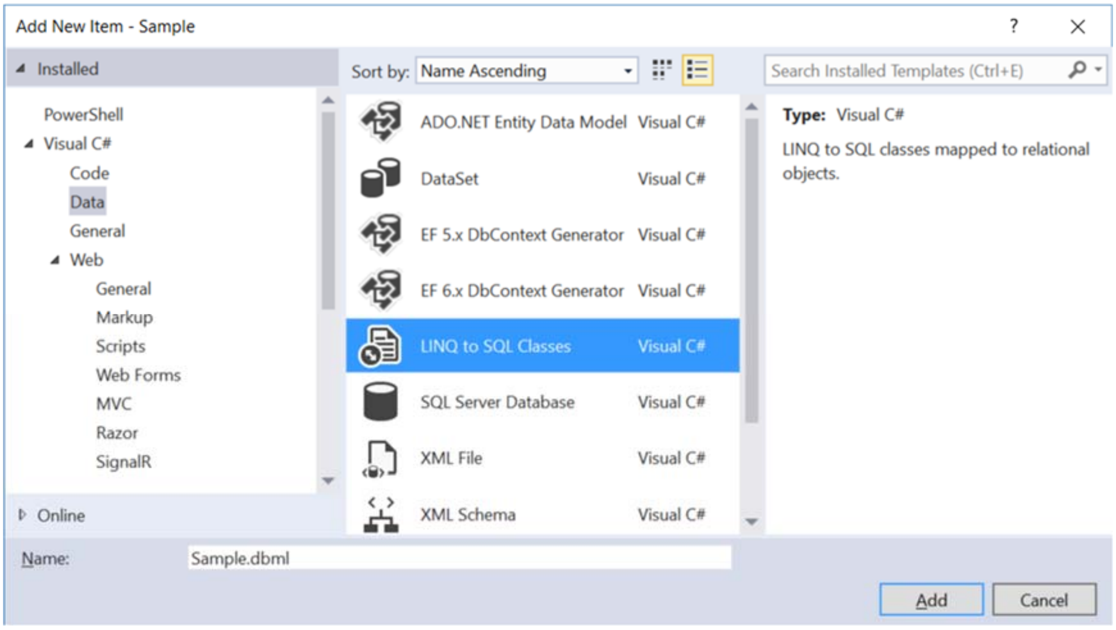
I name it as "Sample.dbml",

because I know this is for connection to "Sample" Database.

-->

Drag Table from Server Explorer into DBML

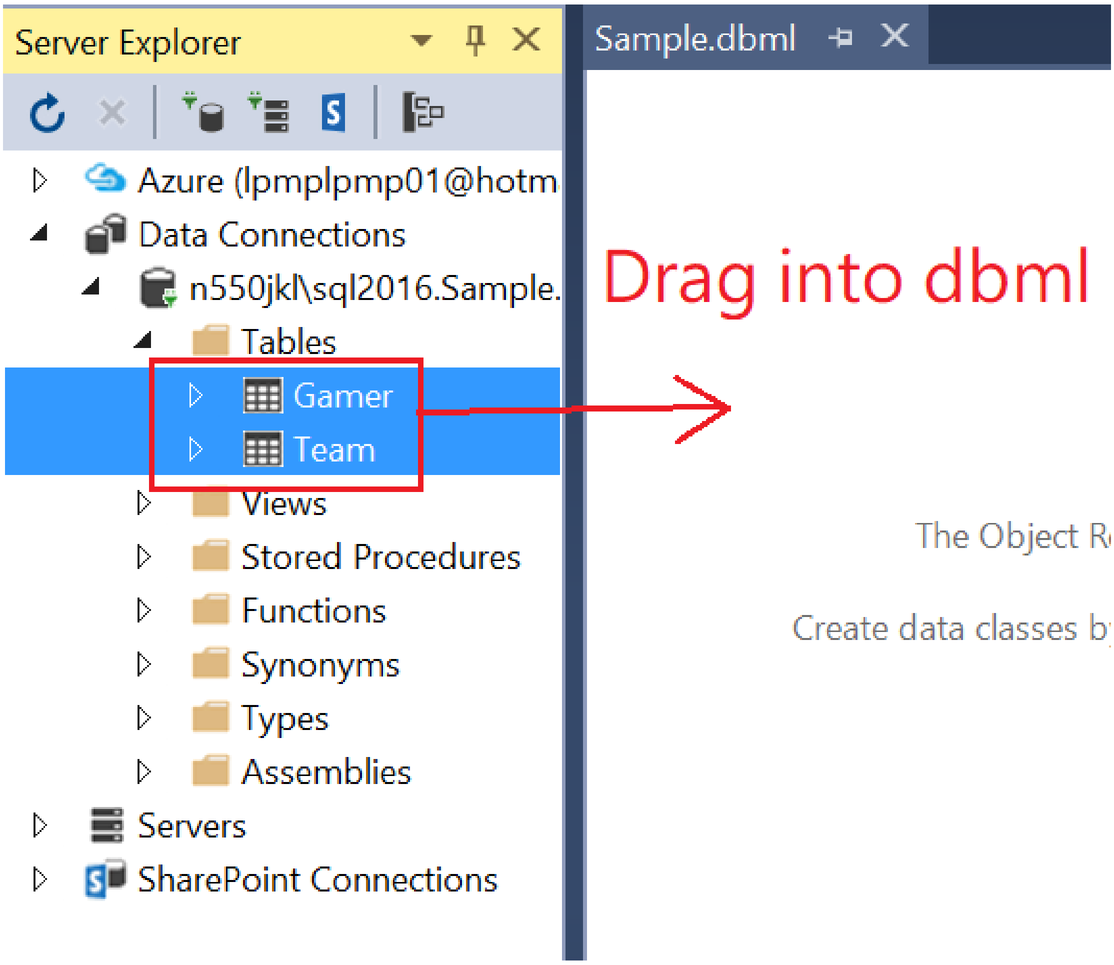


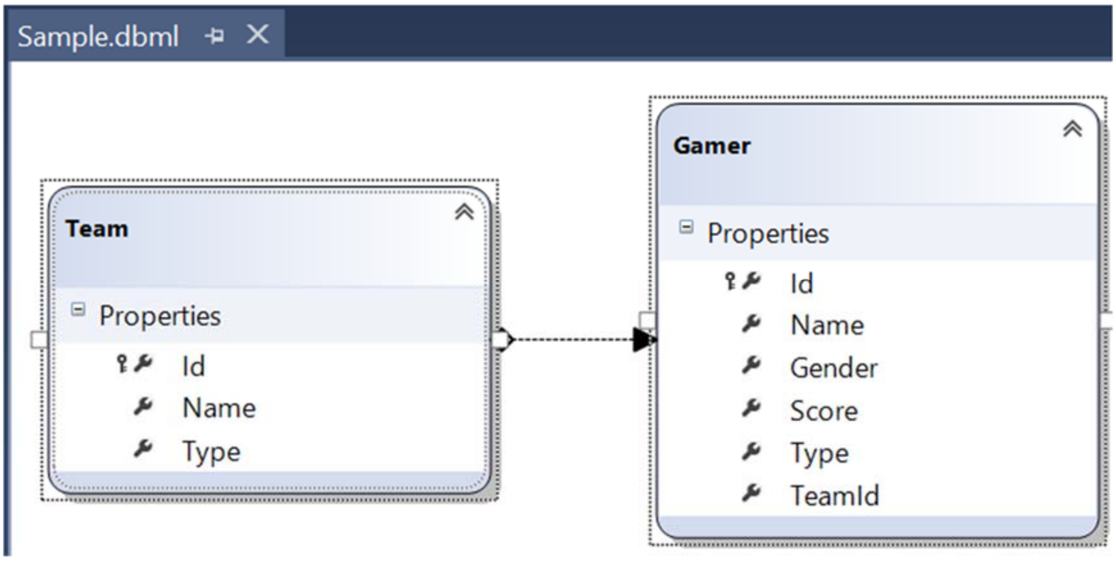


Type: Visual C#

LINQ to SQL classes mapped to relational objects.

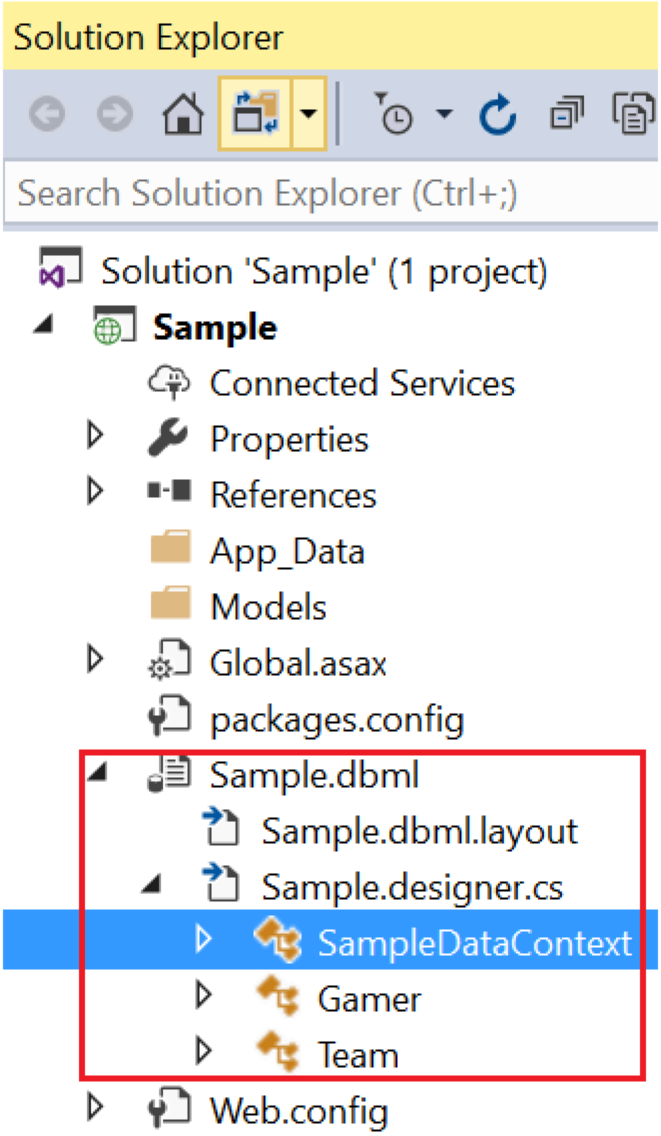
Name: Sample.dbml





Save the dbml, it will generate the following files.

The DataContext context is the entry point to database.



Solution Explorer

Search Solution Explorer (Ctrl+;)

```
Solution 'Sample' (1 project)
  Sample
    Connected Services
    Properties
    References
    App_Data
    Models
    Global.asax
    packages.config
    Sample.dbml
      Sample.dbml.layout
      Sample.designer.cs
        SampleDataContext
        Gamer
        Team
    Web.config
```

3.3. WebForm1.aspx

3.3.1. WebForm1.aspx

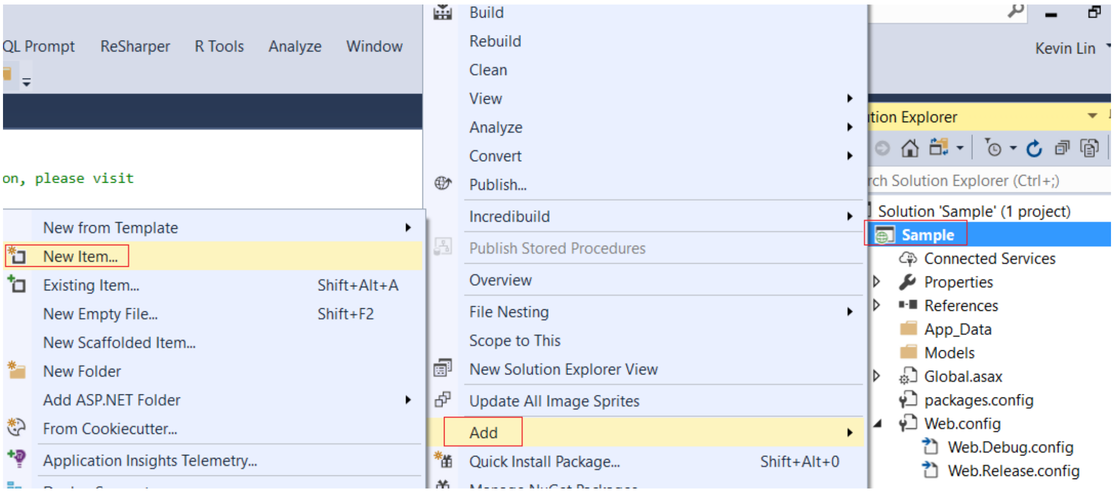
ProjectName --> Right Click --> Add --> New Item...

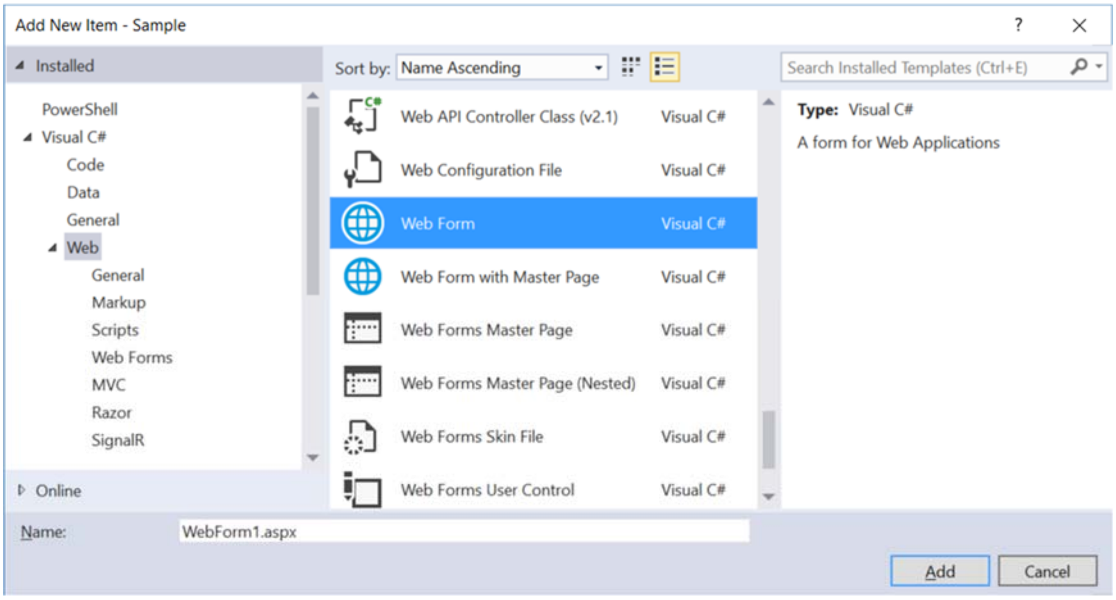
-->

WebForm

Name :

WebForm1.aspx




Add New Item - Sample
?
Installed
PowerShell
Visual C#
Code
Data
General
Web
General
Markup
Scripts
Web Forms
MVC
Razor
SignalR
Online
Sort by: Name Ascending
Web API Controller Class (v2.1) Visual C#
Web Configuration File Visual C#
Web Form Visual C#
Web Form with Master Page Visual C#
Web Forms Master Page Visual C#
Web Forms Master Page (Nested) Visual C#
Web Forms Skin File Visual C#
Web Forms User Control Visual C#
Search Installed Templates (Ctrl+E)
Add
Cancel

Type: Visual C#

A form for Web Applications

Name: WebForm1.aspx

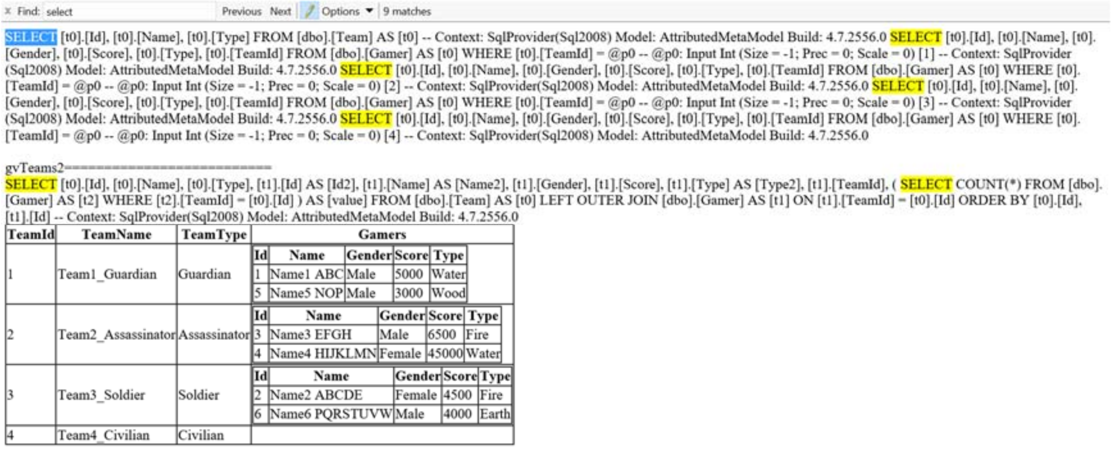
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="
WebApplication1.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

    <title></title>

</head>

<body>

    <form id="form1" runat="server">
```

```
        <div>
            <asp:GridView ID="gvTeams" runat="server" AutoGenerateColumns="False">
                <Columns>
                    <asp:BoundField HeaderText="TeamId" DataField="Id" />
                    <asp:BoundField HeaderText="TeamName" DataField="Name" />
                    <asp:BoundField HeaderText="TeamType" DataField="Type" />
                    <asp:TemplateField HeaderText="Gamers">
                        <ItemTemplate>
                            <asp:GridView ID="gvGamers" runat="server"
                                AutoGenerateColumns="false"
                                DataSource='<%# Eval("Gamers") %>'>
                                <Columns>
                                    <asp:BoundField DataField="Id" HeaderText="Id" />
                                    <asp:BoundField DataField="Name" HeaderText="Name
" />
                                    <asp:BoundField DataField="Gender" HeaderText="Ge
nder" />
                                    <asp:BoundField DataField="Score" HeaderText="Sco
re" />
                                    <asp:BoundField DataField="Type" HeaderText="Type
" />
                                </Columns>
                            </asp:GridView>
                        </ItemTemplate>
                    </asp:TemplateField>
                </Columns>
            </asp:GridView>
            <br />
            <br />
            <asp:GridView ID="gvTeams2" runat="server" AutoGenerateColumns="False">
                <Columns>
                    <asp:BoundField HeaderText="TeamId" DataField="Id" />
                    <asp:BoundField HeaderText="TeamName" DataField="Name" />

                    <asp:BoundField HeaderText="TeamType" DataField="Type" />

                    <asp:TemplateField HeaderText="Gamers">

                        <ItemTemplate>

                            <asp:GridView ID="gvGamers2" runat="server"

                                AutoGenerateColumns="false"

                                DataSource='<%# Eval("Gamers") %>'>

                                <Columns>

                                    <asp:BoundField DataField="Id" HeaderText="Id" />

                                    <asp:BoundField DataField="Name" HeaderText="Name
" />

                                    <asp:BoundField DataField="Gender" HeaderText="Ge
nder" />

                                    <asp:BoundField DataField="Score" HeaderText="Sco
re" />

                                    <asp:BoundField DataField="Type" HeaderText="Type
" />

                                </Columns>

                            </asp:GridView>

                        </ItemTemplate>

                    </asp:TemplateField>

                </Columns>

            </asp:GridView>

            <br />

            <br />

            <asp:Label ID="lbl3" runat="server" Text=""></asp:Label>

            <asp:GridView ID="gvTeams3" runat="server" AutoGenerateColumns="False">

                <Columns>

                    <asp:BoundField HeaderText="TeamId" DataField="Id" />

                    <asp:BoundField HeaderText="TeamName" DataField="Name" />

                    <asp:BoundField HeaderText="TeamType" DataField="Type" />

                    <asp:TemplateField HeaderText="Gamers">
```

```
                        <ItemTemplate>
                            <asp:GridView ID="gvGamers3" runat="server"
                                AutoGenerateColumns="false"
                                DataSource='<%# Eval("Gamers") %>'>
                                <Columns>
                                    <asp:BoundField DataField="Id" HeaderText="Id" />
                                    <asp:BoundField DataField="Name" HeaderText="Name
" />
                                    <asp:BoundField DataField="Gender" HeaderText="Ge
nder" />
                                    <asp:BoundField DataField="Score" HeaderText="Sco
re" />
                                    <asp:BoundField DataField="Type" HeaderText="Type
" />
                                </Columns>
                            </asp:GridView>
                        </ItemTemplate>
                    </asp:TemplateField>
                </Columns>
            </asp:GridView>
            <br />
            <br />
        </div>
    </form>
</body>
</html>
```

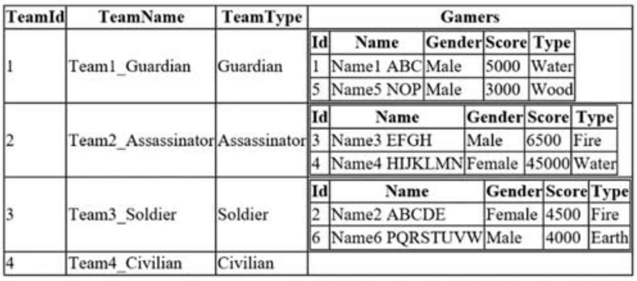
3.3.2. WebForm1.aspx.cs

```
using System;
using System.Data.Linq;
using System.Linq;

namespace WebApplication1

{

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

    {

        protected void Page_Load(object sender, EventArgs e)

        {

            // 1. ===================================

            LazyLoading();

            // 2. ===================================

            EagerLoading();

            // 3. ===================================

            EagerLoading2();

        }

        // 1. ===================================

        private void LazyLoading()

        {

            using (SampleDataContext dbContext = new SampleDataContext())

            {

                ////Write the generated sql query to the Console window

                //dbContext.Log = Console.Out;

                //Write the generated sql query to the webform

                dbContext.Log = Response.Output;

                //IQueryable<Team> linqQuery =

                //    from team in dbContext.Teams

                //    select team;

                //Response.Write($"<br/>dbContext.GetCommand(linqQuery).CommandText<
br/>{dbContext.GetCommand(linqQuery).CommandText}<br/><br/>");

                gvTeams.DataSource = dbContext.Teams;

                gvTeams.DataBind();

            }

        }

        // 2. ===================================
        private void EagerLoading()
        {
            using (SampleDataContext dbContext = new SampleDataContext())
            {
                Response.Write("<br/><br/>gvTeams2==========================<br/>")
;
                //Write the generated sql query to the webform
                dbContext.Log = Response.Output;
                DataLoadOptions loadOptions = new DataLoadOptions();
                loadOptions.LoadWith<Team>(t => t.Gamers);
                dbContext.LoadOptions = loadOptions;
                gvTeams2.DataSource = dbContext.Teams;
                gvTeams2.DataBind();
            }
        }
        // 3. ===================================
        private void EagerLoading2()
        {
            using (SampleDataContext dbContext = new SampleDataContext())
            {
                var linqQuery = from team in dbContext.Teams
                                select new { Id = team.Id, Name=team.Name,
Type=team.Type, Gamers = team.Gamers };
                lbl3.Text
= $"<br/>dbContext.GetCommand(linqQuery).CommandText<br/>{dbContext.GetCommand(linqQ
uery).CommandText}<br/><br/>";


                gvTeams3.DataSource = linqQuery;
                gvTeams3.DataBind();
            }
        }
    }
}
```

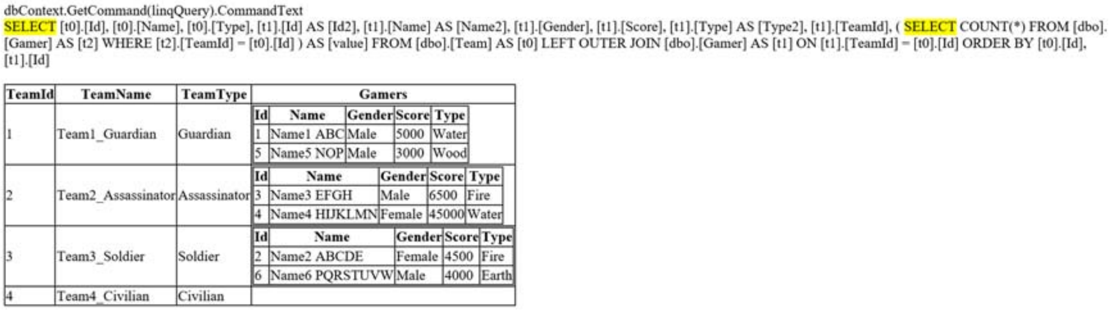


```
SELECT [t0].[Id], [t0].[Name], [t0].[Type] FROM [dbo].[Team] AS [t0] -- Context: SqlProvider(Sql2008) Model: AttributedMetaModel Build: 4.7.2556.0 SELECT [t0].[Id], [t0].[Name], [t0].
[Gender], [t0].[Score], [t0].[Type], [t0].[TeamId] FROM [dbo].[Gamer] AS [t0] WHERE [t0].[TeamId] = @p0 -- @p0: Input Int (Size = -1; Prec = 0; Scale = 0) [1] -- Context: SqlProvider
(Sql2008) Model: AttributedMetaModel Build: 4.7.2556.0 SELECT [t0].[Id], [t0].[Name], [t0].[Gender], [t0].[Score], [t0].[Type], [t0].[TeamId] FROM [dbo].[Gamer] AS [t0] WHERE [t0].
[TeamId] = @p0 -- @p0: Input Int (Size = -1; Prec = 0; Scale = 0) [2] -- Context: SqlProvider(Sql2008) Model: AttributedMetaModel Build: 4.7.2556.0 SELECT [t0].[Id], [t0].[Name], [t0].
[Gender], [t0].[Score], [t0].[Type], [t0].[TeamId] FROM [dbo].[Gamer] AS [t0] WHERE [t0].[TeamId] = @p0 -- @p0: Input Int (Size = -1; Prec = 0; Scale = 0) [3] -- Context: SqlProvider
(Sql2008) Model: AttributedMetaModel Build: 4.7.2556.0 SELECT [t0].[Id], [t0].[Name], [t0].[Gender], [t0].[Score], [t0].[Type], [t0].[TeamId] FROM [dbo].[Gamer] AS [t0] WHERE [t0].
[TeamId] = @p0 -- @p0: Input Int (Size = -1; Prec = 0; Scale = 0) [4] -- Context: SqlProvider(Sql2008) Model: AttributedMetaModel Build: 4.7.2556.0

gvTeams2==========================
SELECT [t0].[Id], [t0].[Name], [t0].[Type], [t1].[Id] AS [Id2], [t1].[Name] AS [Name2], [t1].[Gender], [t1].[Score], [t1].[Type] AS [Type2], [t1].[TeamId], ( SELECT COUNT(*) FROM [dbo].
[Gamer] AS [t2] WHERE [t2].[TeamId] = [t0].[Id] ) AS [value] FROM [dbo].[Team] AS [t0] LEFT OUTER JOIN [dbo].[Gamer] AS [t1] ON [t1].[TeamId] = [t0].[Id] ORDER BY [t0].[Id],
[t1].[Id] -- Context: SqlProvider(Sql2008) Model: AttributedMetaModel Build: 4.7.2556.0
```

TeamId	TeamName	TeamType	Gamers				
			Id	Name	Gender	Score	Type
1	Team1_Guardian	Guardian	1	Name1 ABC	Male	5000	Water
			5	Name5 NOP	Male	3000	Wood
			Id	Name	Gender	Score	Type
2	Team2_Assassinator	Assassinator	3	Name3 EFGH	Male	6500	Fire
			4	Name4 HIJKLMN	Female	45000	Water
			Id	Name	Gender	Score	Type
3	Team3_Soldier	Soldier	2	Name2 ABCDE	Female	4500	Fire
			6	Name6 PQRSTUVW	Male	4000	Earth
4	Team4_Civilian	Civilian					



TeamId	TeamName	TeamType	Gamers				
			Id	Name	Gender	Score	Type
1	Team1_Guardian	Guardian	1	Name1 ABC	Male	5000	Water
			5	Name5 NOP	Male	3000	Wood
			Id	Name	Gender	Score	Type
2	Team2_Assassinator	Assassinator	3	Name3 EFGH	Male	6500	Fire
			4	Name4 HIJKLMN	Female	45000	Water
			Id	Name	Gender	Score	Type
3	Team3_Soldier	Soldier	2	Name2 ABCDE	Female	4500	Fire
			6	Name6 PQRSTUVW	Male	4000	Earth
4	Team4_Civilian	Civilian					



```
dbContext.GetCommand(linqQuery).CommandText
SELECT [t0].[Id], [t0].[Name], [t0].[Type], [t1].[Id] AS [Id2], [t1].[Name] AS [Name2], [t1].[Gender], [t1].[Score], [t1].[Type] AS [Type2], [t1].[TeamId], ( SELECT COUNT(*) FROM [dbo].
[Gamer] AS [t2] WHERE [t2].[TeamId] = [t0].[Id] ) AS [value] FROM [dbo].[Team] AS [t0] LEFT OUTER JOIN [dbo].[Gamer] AS [t1] ON [t1].[TeamId] = [t0].[Id] ORDER BY [t0].[Id],
[t1].[Id]
```

TeamId	TeamName	TeamType	Gamers				
			Id	Name	Gender	Score	Type
1	Team1_Guardian	Guardian	1	Name1 ABC	Male	5000	Water
			5	Name5 NOP	Male	3000	Wood
			Id	Name	Gender	Score	Type
2	Team2_Assassinator	Assassinator	3	Name3 EFGH	Male	6500	Fire
			4	Name4 HIJKLMN	Female	45000	Water
			Id	Name	Gender	Score	Type
3	Team3_Soldier	Soldier	2	Name2 ABCDE	Female	4500	Fire
			6	Name6 PQRSTUVW	Male	4000	Earth
4	Team4_Civilian	Civilian					

4. LazyLoading V.S. EagerLoading

1.

LazyLoading V.S. EagerLoading

Reference:

https://msdn.microsoft.com/en-us/library/jj574232(v=vs.113).aspx

https://stackoverflow.com/questions/97197/what-is-n1-select-query-issue

1.1.

LazyLoading

1.1.1.

We retrieve just the amount of data that we need in a single query.

When we need more data, then it issues more queries to the database.

That means we might have to request the data from database many times,

and this might cost the performance.

1.1.2.

LazyLoading might cause N+1 select problem.

E.g.

One Team can have many Gamers.

One Gamer can have one Team.

This isOne-to-Many relationship.

When we have N teams, and when we select for the Teams,

and then additional selects to retrieve the Gamers belonging to each Team.

That means we have to request the data from database additional N times.

This is N+1 select problem.

1.2.

EagerLoading

We retrieve all data that we need in a single query,

and then be cached to improve the application performance.

That means we just have to request the data from database once,

but this cost memory consumption.

1.3.

Conclusion

1.3.1.

If you need only Team data,

then "lazy loading" works best.

If you choose to use "Eager loading" in this case,

it will cost memory consumption.

1.3.2.

However, if you need Team data and its Gamers data,

then "Eager loading" works best.

If you choose to use "lazy loading" in this case,

it will request the data from database too many times,

this cost application performance.