(T10)比較 LinqToSql 的 AsEnumerable、AsQueryable
CourseGUID: 5ba9a6fe-7475-4b0c-8b99-bbcf7f5e2e1c
==
(T10)比較 LinqToSql 的 AsEnumerable、AsQueryable
==

==

0. Summary

1.
Deferred/Lazy Operators V.S. Immediate/Greedy Operators
Based on the behavior of query execution, Linq can be classified into 2 categories.
1.1. Deferred/Lazy Operators use deferred execution.
E.g. select, where, Take, Skip ...
1.2. Immediate/Greedy Operators use immediate execution.
E.g. count, average, min, max, ToList ...
1.3.
ToList, ToArray, ToDictionary, ToLookup, Cast, OfType, AsEnumerable, AsQueryable
are Linq Conversion Operators.

2.
2.1.
Queryable.AsQueryable<TElement>
(IEnumerable<TElement>)
Reference:
https://msdn.microsoft.com/en-us/library/bb507003(v=vs.110).aspx
https://stackoverflow.com/questions/17366907/what-is-the-purpose-of-asqueryable
Converts a generic IEnumerable<T> to a generic IQueryable<T>.
The main use of AsQueryable operator is unit testing to mock a queryable in-memory data source

3.
3.1.
Enumerable.AsEnumerable<TSource>
(this IEnumerable<TSource> source)
Reference:
https://msdn.microsoft.com/en-us/library/bb335435(v=vs.110).aspx
Returns the input typed as IEnumerable<T>.
3.2.
AsEnumerable operator split the Linq query into 2 parts.

In another words, AsEnumerable() move query processing to the client side.
3.2.1.
Linq to SQL part
The Linq query before AsEnumerable() is Linq to SQL part which reads data from SQL Server database to application.
3.2.2.
Linq to Objects part
The Linq query after AsEnumerable() is Linq to Objects part which process to the local client side machine.

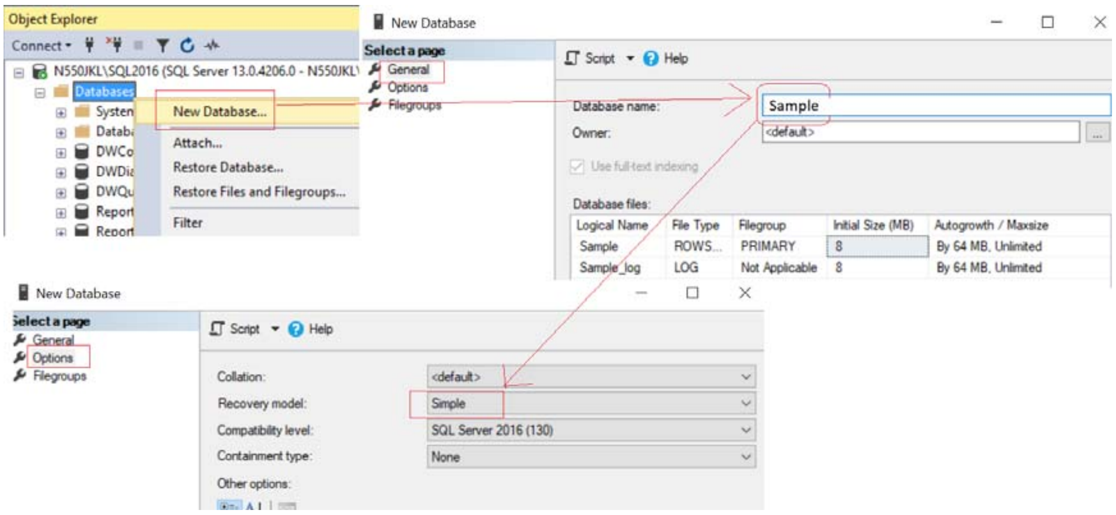
4.
IQueryable<T> V.S. IEnumerable<T> in Entity Framework
In the future, you might learn Entity Framework,
then you might see the following.
4.1.
//var xxxDbContext = new XxxDbContext();
// **IQueryable<Gamer>** gamers =xxxDbContext.Gamers;
//var gamer = Gamers.Where(g=>g.Level);
IQueryable<T> is **Deferred/Lazy** Operation type which use deferred execution.
It means it will generate the following TSQL.
Select ... From .... Where...
4.2.
//var xxxDbContext = new XxxDbContext();
//**IEnumerable<Gamer>** gamers = IQueryable.Gamers;
//var gamer = Gamers.Where(g=>g.Level);
IEnumerable<T> is **Immediate/Greedy** Operation type which use immediate execution.
It means it will generate the following TSQL.
Select ... From ....

==

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>
Use full-text indexing

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

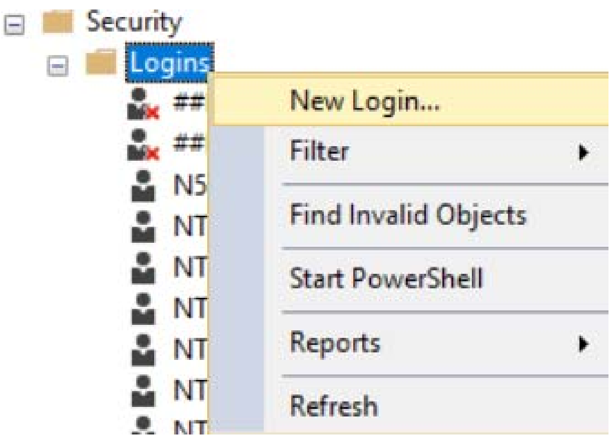
Collation: <default>
Recovery model: Simple
Compatibility level: SQL Server 2016 (130)
Containment type: None
Other options:

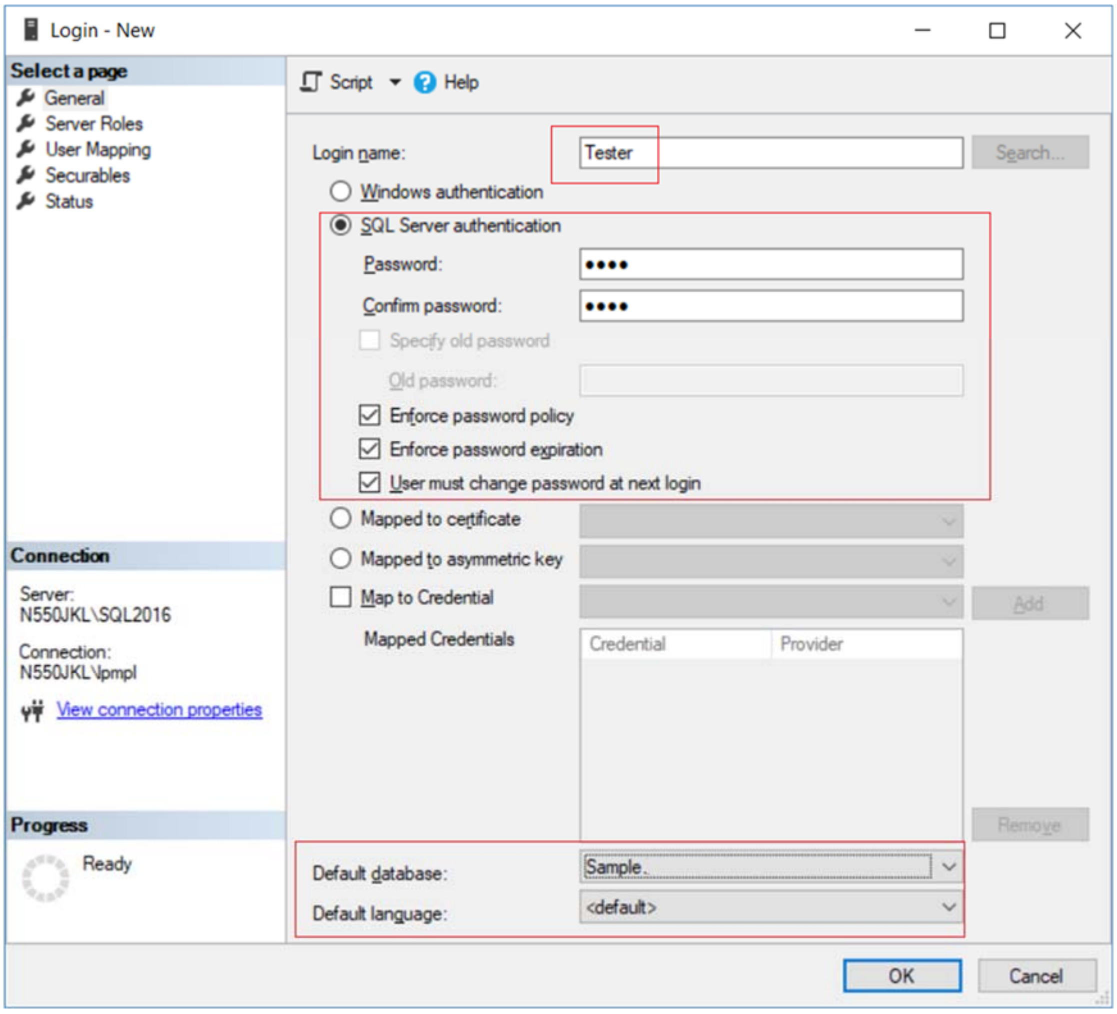
--Create an Sample DataBase and Run the following TSQL

```
Create Table Gamer
(
     Id int primary key IDENTITY(1,1),
    Name nvarchar(100),
     Gender nvarchar(50),
        Score int
)
GO
Insert into Gamer values ('Name1', 'Male', 3500)
Insert into Gamer values ('Name2', 'Female', 4000)
Insert into Gamer values ('Name3', 'Male', 5000)
Insert into Gamer values ('Name4', 'Female', 7000)
Insert into Gamer values ('Name5', 'Female', 3000)
Insert into Gamer values ('Name6', 'Male', 4500)
Insert into Gamer values ('Name7', 'Male', 4000)
Insert into Gamer values ('Name8', 'Male', 5500)
Insert into Gamer values ('Name9', 'Female', 6500)
Insert into Gamer values ('Name10', 'Female', 3500)
GO
```

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 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

Default database: Sample

Default language: <default>

Connection

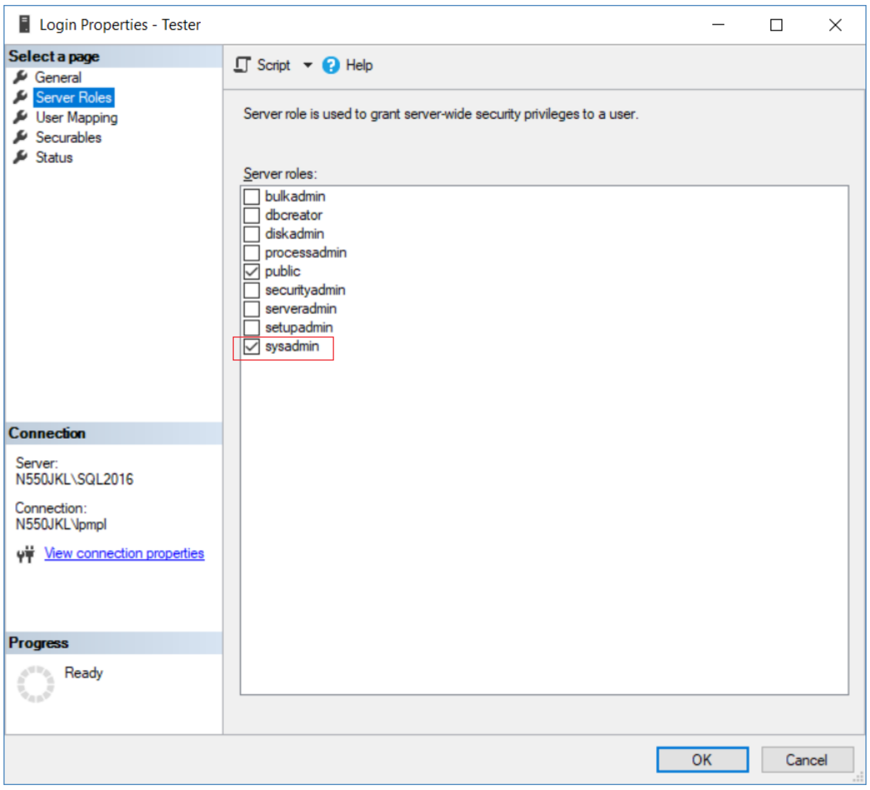
Server:
N550JKL\SQL2016

Connection:
N550JKL\pmpl

View connection properties

Progress

Ready


Login Properties - Tester

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

Script | Help

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

Server roles:
- [ ] bulkadmin
- [ ] dbcreator
- [ ] diskadmin
- [ ] processadmin
- [✓] public
- [ ] securityadmin
- [ ] serveradmin
- [ ] setupadmin
- [✓] sysadmin

Connection

Server:
N550JKL\SQL2016

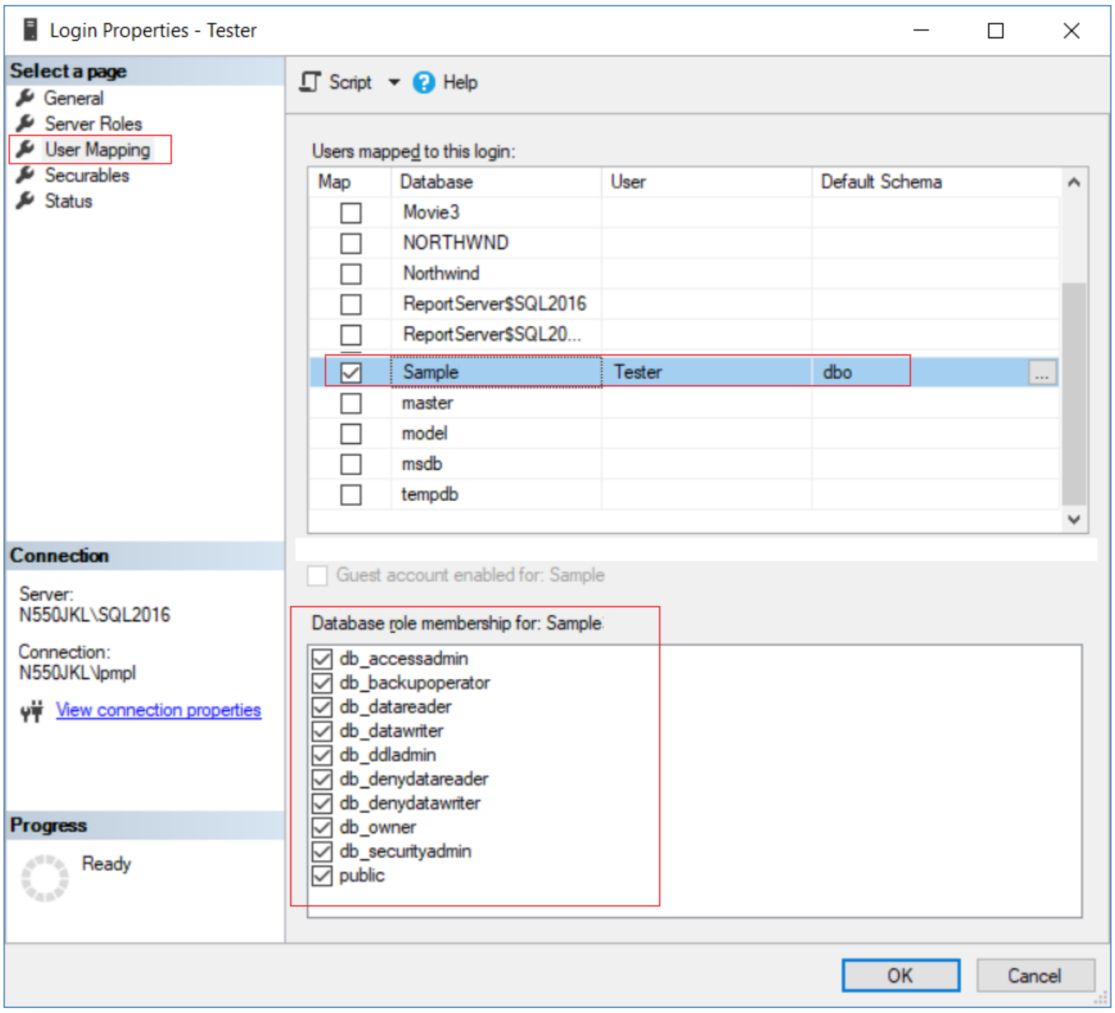
Connection:
N550JKL\lpmpl

View connection properties

Progress

Ready

OK | Cancel


Login Properties - Tester

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

Script | Help

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

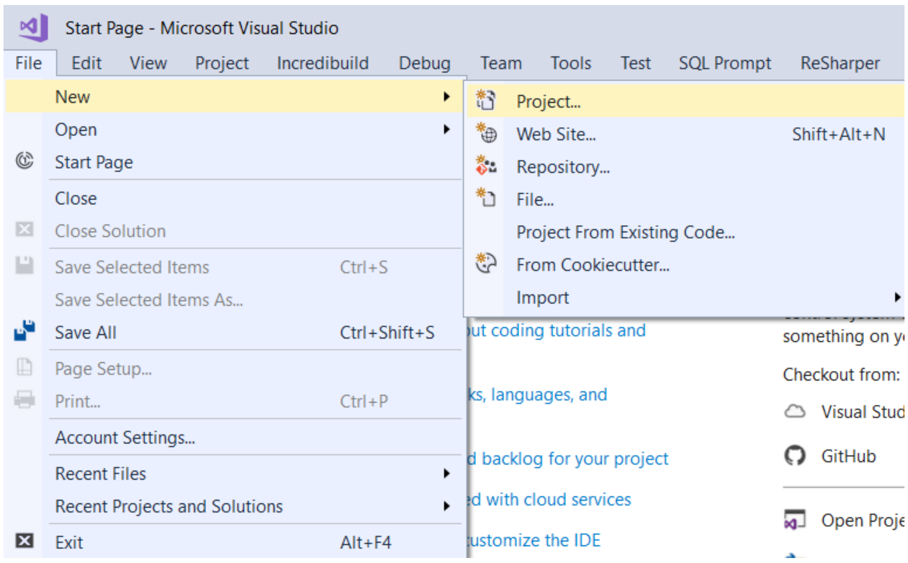
OK | Cancel

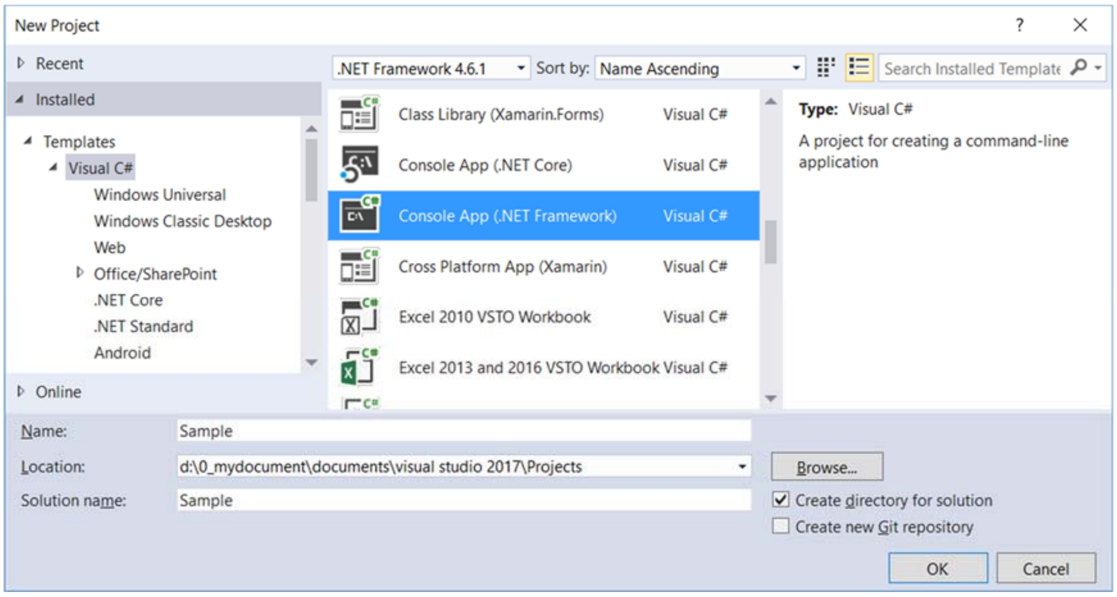
1.3. Create New Project : Sample

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
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#
Search Installed Template
Browse...
OK
Cancel

.NET Framework 4.6.1 Sort by: Name Ascending

Type: Visual C#

A project for creating a command-line application

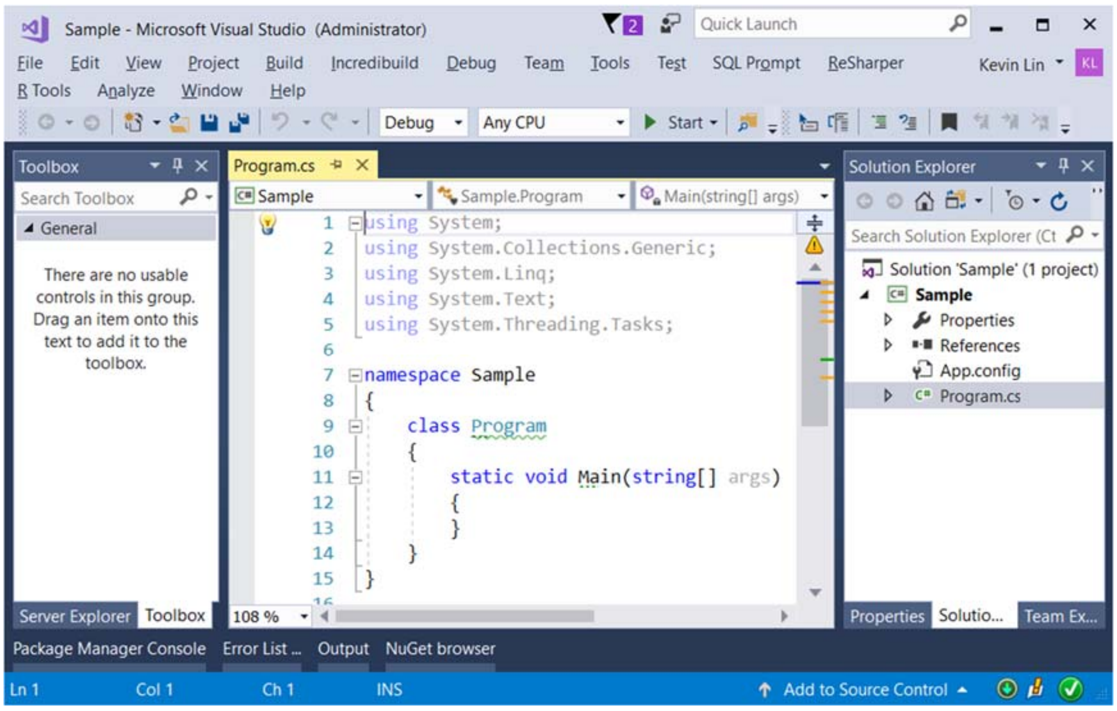
Name: Sample

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

Solution name: Sample

[✓] Create directory for solution

[ ] Create new Git repository

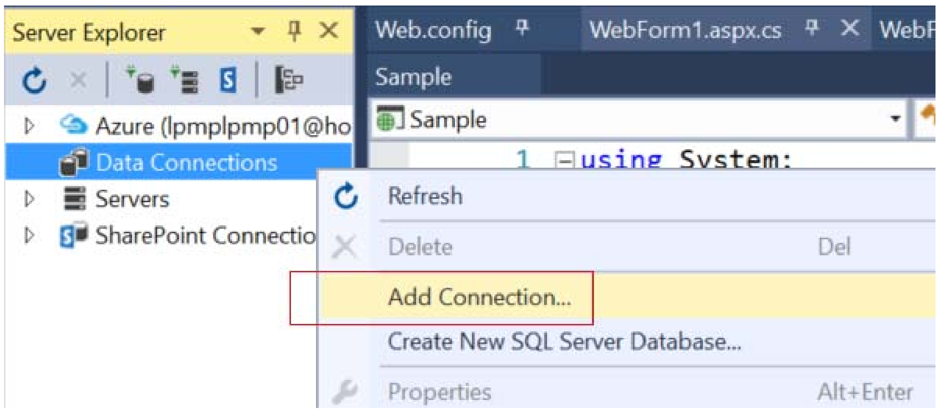


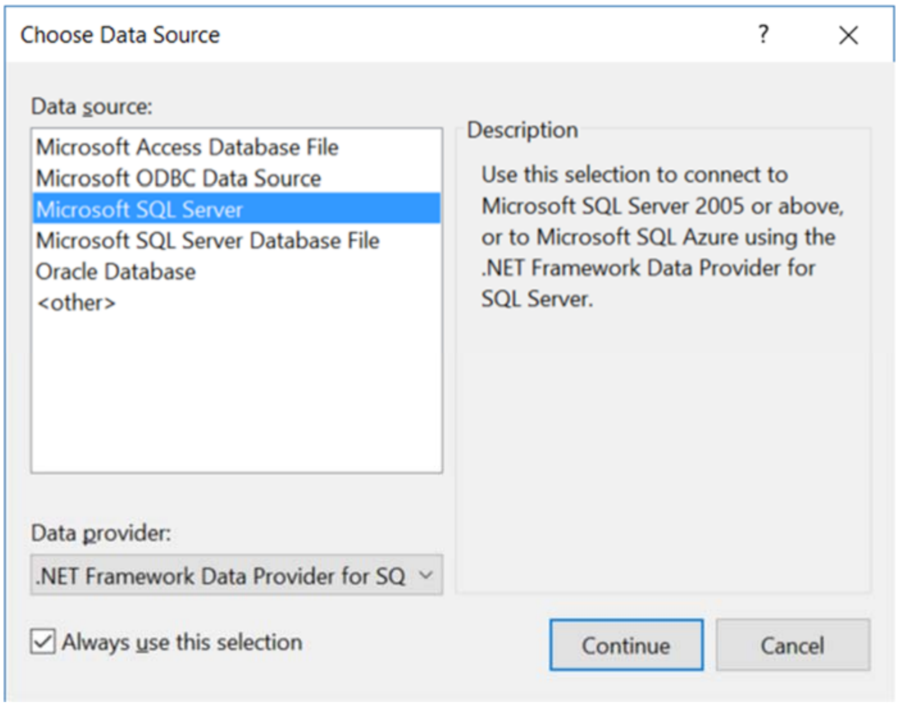
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

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

2. Linq to SQL

2.1. 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>

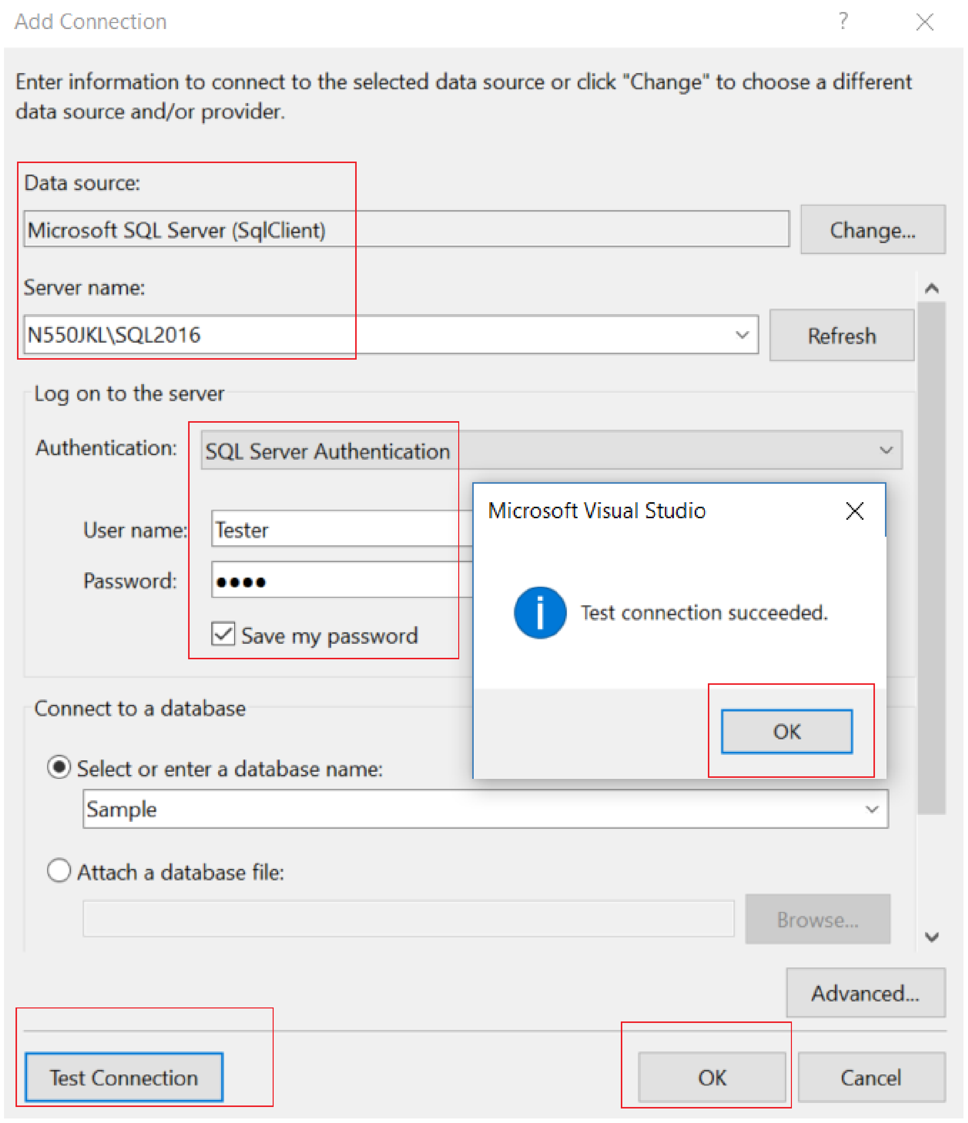
Description

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

Continue | Cancel


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) — Change...

Server name: N550JKL\SQL2016 — Refresh

Log on to the server

Authentication: SQL Server Authentication

User name: Tester

Password: ●●●●

☑ Save my password

Microsoft Visual Studio

Test connection succeeded.

OK

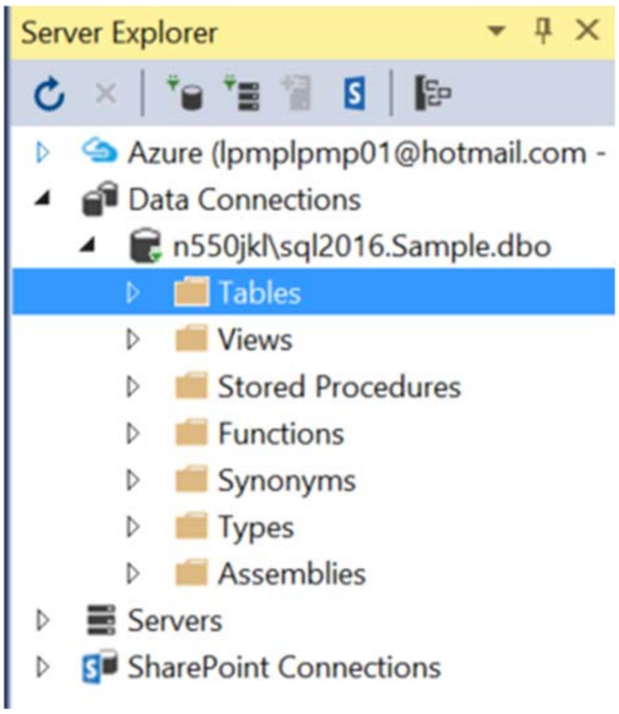
Connect to a database

● Select or enter a database name: Sample

○ Attach a database file:

Browse... | Advanced...

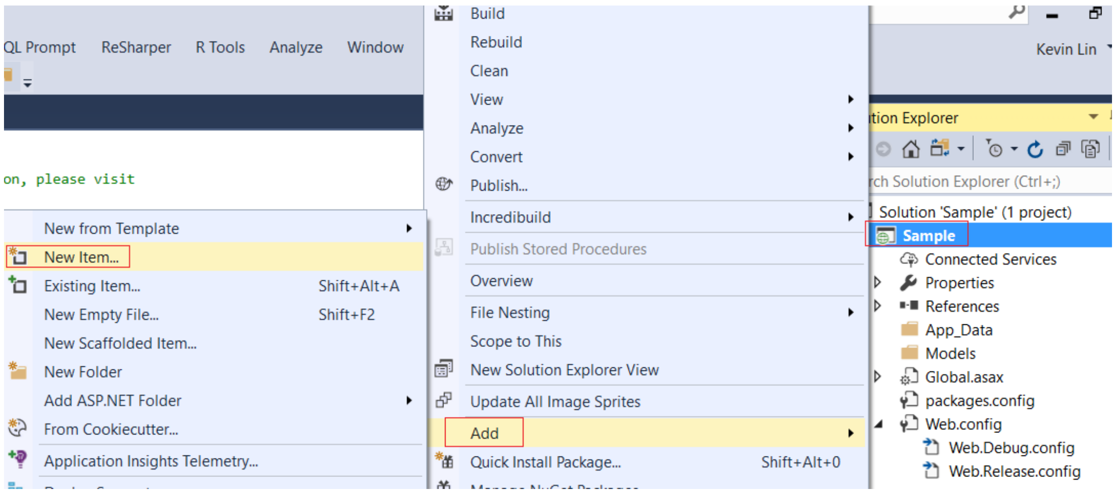
Test Connection | OK | Cancel

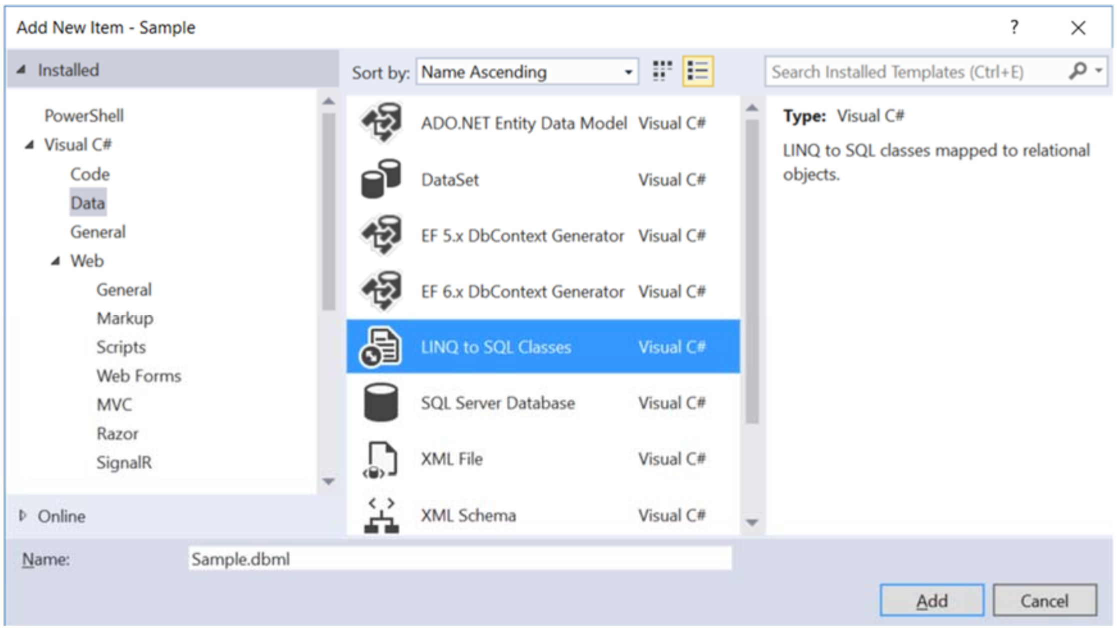


2.2. DataClasses1.dbml

ProjectName --> Right Click --> Add --> New Item...
--> Linq to SQL classes -->
Name : **Sample.dbml**

-->
Drag Table from Server Explorer into DBML



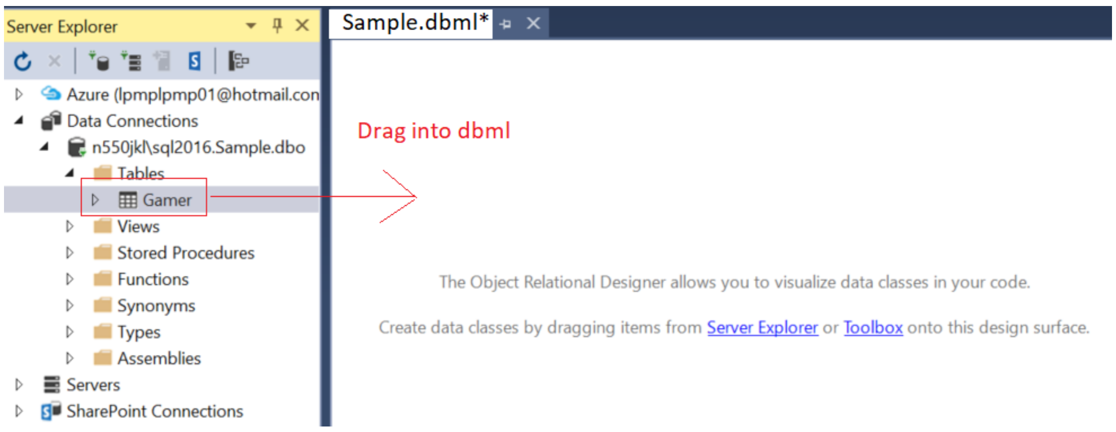

Add New Item - Sample
Installed
PowerShell
Visual C#
Code
Data
General
Web
General
Markup
Scripts
Web Forms
MVC
Razor
SignalR
Online
ADO.NET Entity Data Model Visual C#
DataSet Visual C#
EF 5.x DbContext Generator Visual C#
EF 6.x DbContext Generator Visual C#
LINQ to SQL Classes Visual C#
SQL Server 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#

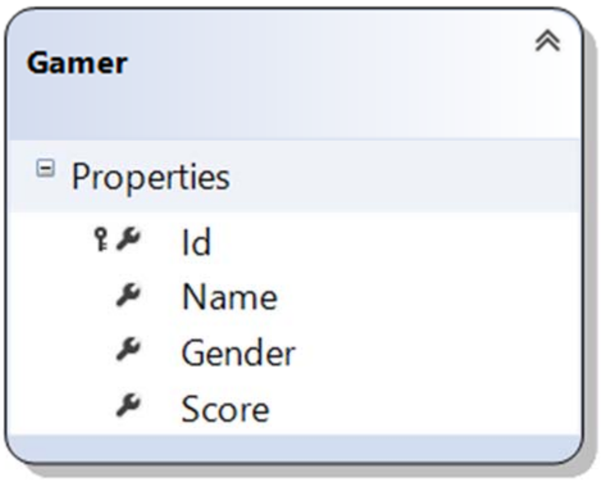
LINQ to SQL classes mapped to relational objects.

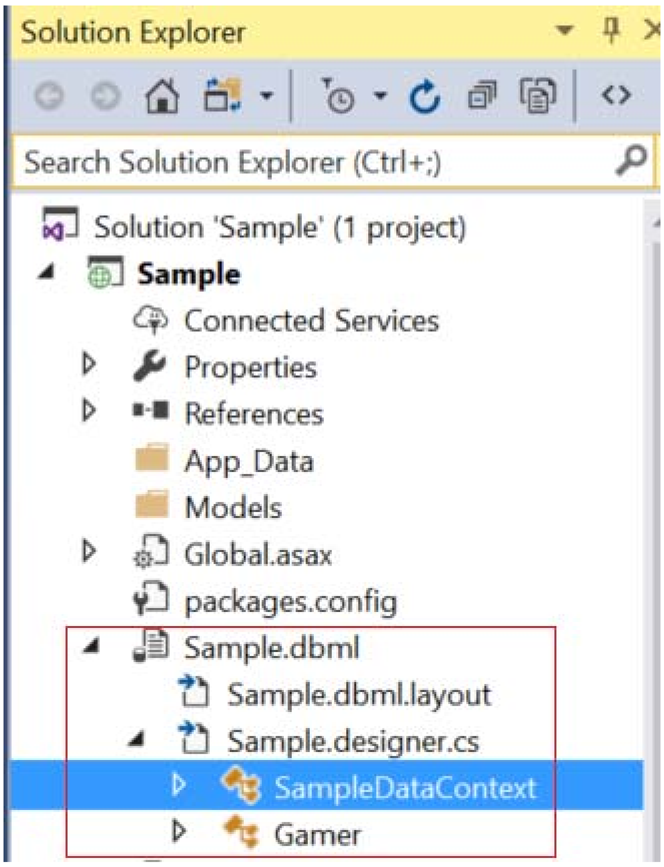
Name: Sample.dbml



The Object Relational Designer allows you to visualize data classes in your code.

Create data classes by dragging items from Server Explorer or Toolbox onto this design surface.

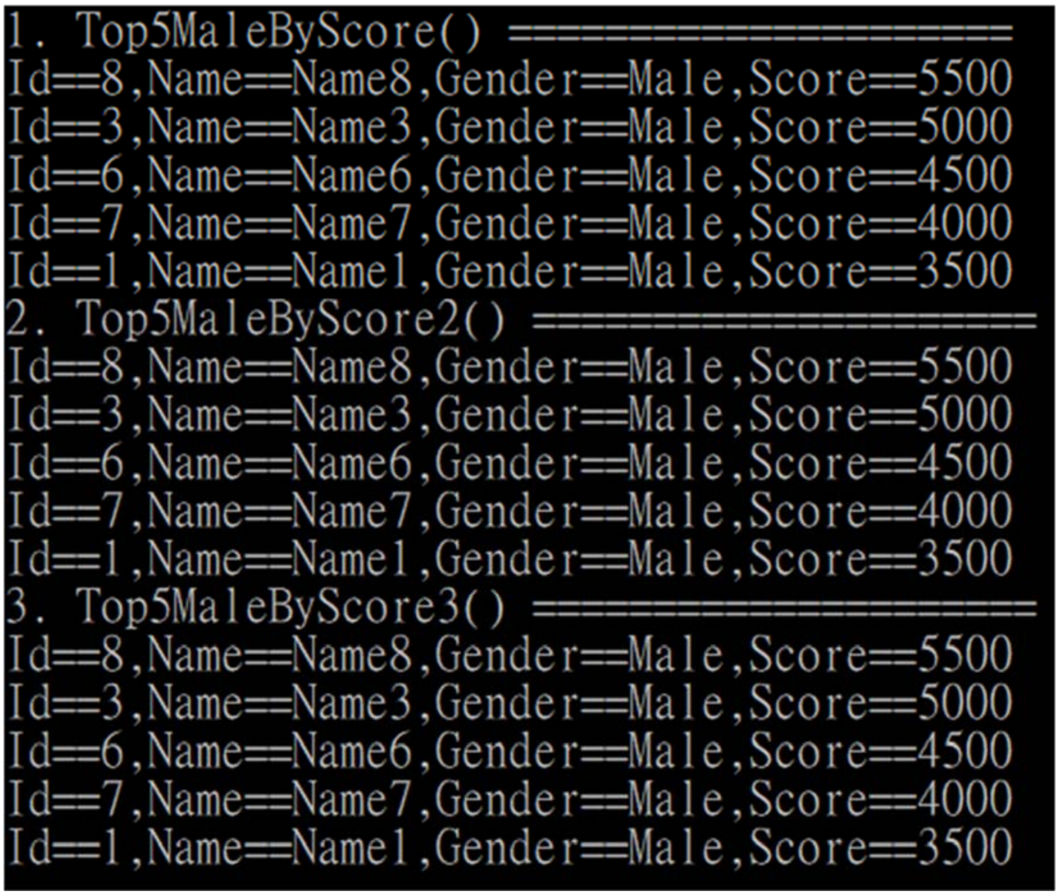




2.3. Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
namespace Sample
{
    class Program
    {
        static void Main(string[] args)
        {
            // 1. ==========================================
            //Top5MaleByScore()
            Console.WriteLine("1. Top5MaleByScore() ===================== ");
            Top5MaleByScore();
            // 2. ==========================================
            //Top5MaleByScore2()
            Console.WriteLine("2. Top5MaleByScore2() ===================== ");
            Top5MaleByScore2();
            // 3. ==========================================
            //Top5MaleByScore3()
            Console.WriteLine("3. Top5MaleByScore3() ===================== ");
            Top5MaleByScore3();
            Console.ReadLine();
        }


        // 1. ==========================================
//Top5MaleByScore()
private static void Top5MaleByScore()
{
    SampleDataContext dataContext = new SampleDataContext();
    //Get Top 5 Male by Score
    IQueryable<Gamer> top5MaleByScore =
         dataContext.Gamers
         .Where(g => g.Gender == "Male")
         .OrderByDescending(g => g.Score)
         .Take(5);
    foreach (Gamer gamer in top5MaleByScore)
    {
        Console.WriteLine(gamer);
    }
}
//1.1.
//Id==8,Name==Name8,Gender==Male,Score==5500
//Id==3,Name==Name3,Gender==Male,Score==5000
//Id==6,Name==Name6,Gender==Male,Score==4500
//Id==7,Name==Name7,Gender==Male,Score==4000
//Id==1,Name==Name1,Gender==Male,Score==3500
//1.2.
//Notice that the following SQL Query is executed against the database.
//exec sp_executesql N'SELECT TOP (5) [t0].[Id], [t0].[Name], [t0].[Gender], [t0].[Score]
//FROM[dbo].[Gamer] AS[t0]
//WHERE[t0].[Gender] = @p0
//ORDER BY[t0].[Score] DESC',N'@p0 nvarchar(4000)',@p0=N'Male'
// 2. ==========================================
//Top5MaleByScore2()
private static void Top5MaleByScore2()
{
    SampleDataContext dataContext = new SampleDataContext();
    //Get Top 5 Male by Score
    IEnumerable<Gamer> top5MaleByScore =
         dataContext.Gamers.AsEnumerable()
         .Where(g => g.Gender == "Male")
         .OrderByDescending(g => g.Score)
         .Take(5);
    foreach (Gamer gamer in top5MaleByScore)
    {
        Console.WriteLine(gamer);
    }
}
//1.1.
//Id==8,Name==Name8,Gender==Male,Score==5500
//Id==3,Name==Name3,Gender==Male,Score==5000
//Id==6,Name==Name6,Gender==Male,Score==4500
//Id==7,Name==Name7,Gender==Male,Score==4000
//Id==1,Name==Name1,Gender==Male,Score==3500
//1.2.
//Notice that the following SQL Query is executed against the database.
//SELECT [t0].[Id], [t0].[Name], [t0].[Gender], [t0].[Score]
//FROM[dbo].[Gamer] AS[t0]


// 3. ==========================================
        //Top5MaleByScore3()
        private static void Top5MaleByScore3()
        {
            SampleDataContext dataContext = new SampleDataContext();
            //Get Top 5 Male by Score
            IEnumerable<Gamer> top5MaleByScore =
                dataContext.Gamers
                .Where(g => g.Gender == "Male")
                .AsEnumerable()
                .OrderByDescending(g => g.Score)
                .Take(5);
            foreach (Gamer gamer in top5MaleByScore)
            {
                Console.WriteLine(gamer);
            }
        }
        //3.1.
        //Id==8,Name==Name8,Gender==Male,Score==5500
        //Id==3,Name==Name3,Gender==Male,Score==5000
        //Id==6,Name==Name6,Gender==Male,Score==4500
        //Id==7,Name==Name7,Gender==Male,Score==4000
        //Id==1,Name==Name1,Gender==Male,Score==3500
        //3.2.
        //Notice that the following SQL Query is executed against the database.
        //exec sp_executesql N'SELECT [t0].[Id], [t0].[Name], [t0].[Gender], [t0].[Score]
        //FROM[dbo].[Gamer]
        //AS[t0]
        //WHERE[t0].[Gender] = @p0',N'@p0 nvarchar(4000)',@p0=N'Male'
    }
    public partial class Gamer
    {
        public override string ToString()
        {
            return $"Id=={Id},Name=={Name},Gender=={Gender},Score=={Score}";
        }
    }
}

/*
1.
Deferred/Lazy Operators  V.S.  Immediate/Greedy Operators
Based on the behavior of query execution, Linq can be classified into 2 categories.
1.1. Deferred/Lazy Operators use deferred execution.
E.g.  select, where, Take, Skip ...
1.2. Immediate/Greedy Operators use immediate execution.
E.g.  count, average, min, max, ToList ...
1.3.
ToList, ToArray, ToDictionary, ToLookup, Cast, OfType, AsEnumerable, AsQueryable
are Linq Conversion Operators.
2.
Queryable.AsQueryable<TElement>
(IEnumerable<TElement>)
Reference:
https://msdn.microsoft.com/en-us/library/bb507003(v=vs.110).aspx
https://stackoverflow.com/questions/17366907/what-is-the-purpose-of-asqueryable
Converts a generic IEnumerable<T> to a generic IQueryable<T>.
The main use of AsQueryable operator is unit testing to mock a queryable in-memory data source
3.
3.1.
Enumerable.AsEnumerable<TSource>
(this IEnumerable<TSource> source)
```

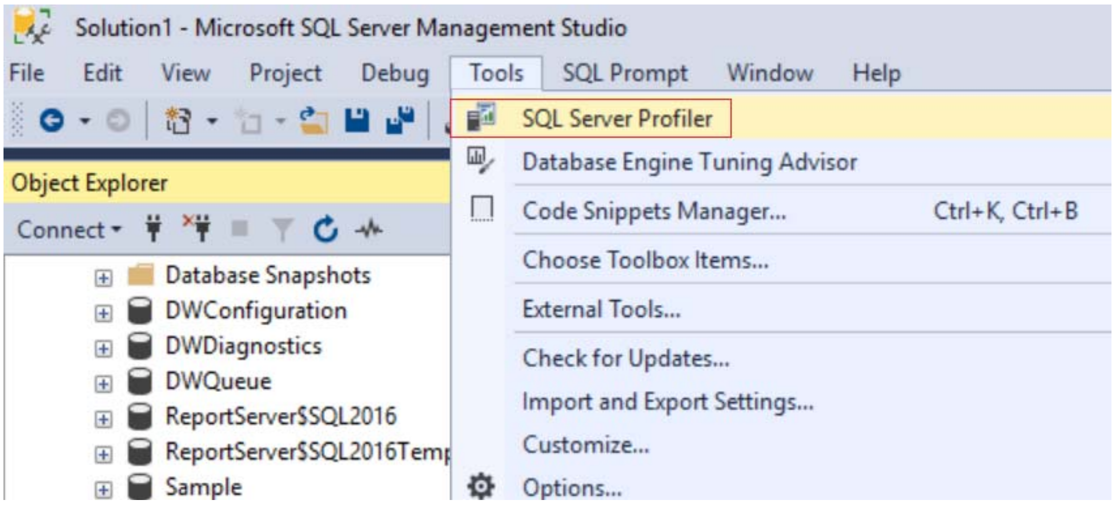
```
Reference:
https://msdn.microsoft.com/en-us/library/bb335435(v=vs.110).aspx
Returns the input typed as IEnumerable<T>.
3.2.
AsEnumerable operator split the Linq query into 2 parts.
In another words, AsEnumerable() move query processing to the client side.
3.2.1.
Linq to SQL part
The Linq query before AsEnumerable() is Linq to SQL part which reads data from SQL Server database to
application.
3.2.2.
Linq to Objects part
The Linq query after AsEnumerable() is Linq to Objects part which process to the local client side
machine.
*/
```

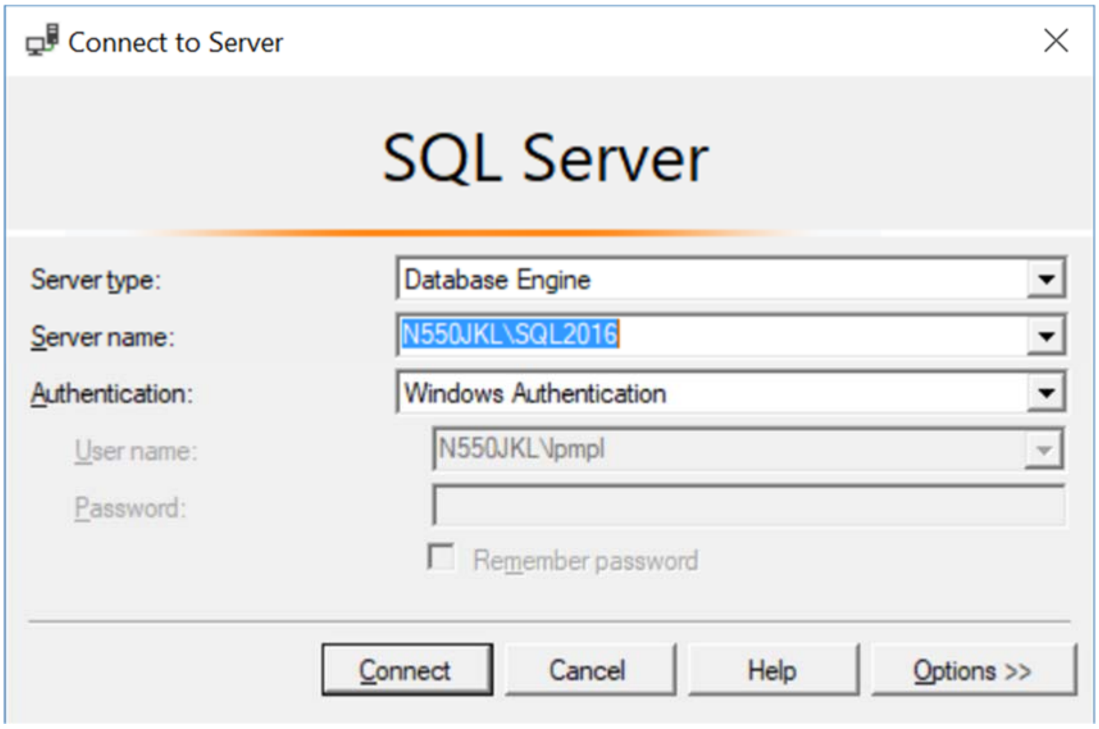


```
1. Top5MaleByScore() ====================
Id==8,Name==Name8,Gender==Male,Score==5500
Id==3,Name==Name3,Gender==Male,Score==5000
Id==6,Name==Name6,Gender==Male,Score==4500
Id==7,Name==Name7,Gender==Male,Score==4000
Id==1,Name==Name1,Gender==Male,Score==3500
2. Top5MaleByScore2() ====================
Id==8,Name==Name8,Gender==Male,Score==5500
Id==3,Name==Name3,Gender==Male,Score==5000
Id==6,Name==Name6,Gender==Male,Score==4500
Id==7,Name==Name7,Gender==Male,Score==4000
Id==1,Name==Name1,Gender==Male,Score==3500
3. Top5MaleByScore3() ====================
Id==8,Name==Name8,Gender==Male,Score==5500
Id==3,Name==Name3,Gender==Male,Score==5000
Id==6,Name==Name6,Gender==Male,Score==4500
Id==7,Name==Name7,Gender==Male,Score==4000
Id==1,Name==Name1,Gender==Male,Score==3500
```

2.4. SQL Profiler

Tools --> SQL Server Profiler




Connect to Server

SQL Server

Server type: Database Engine

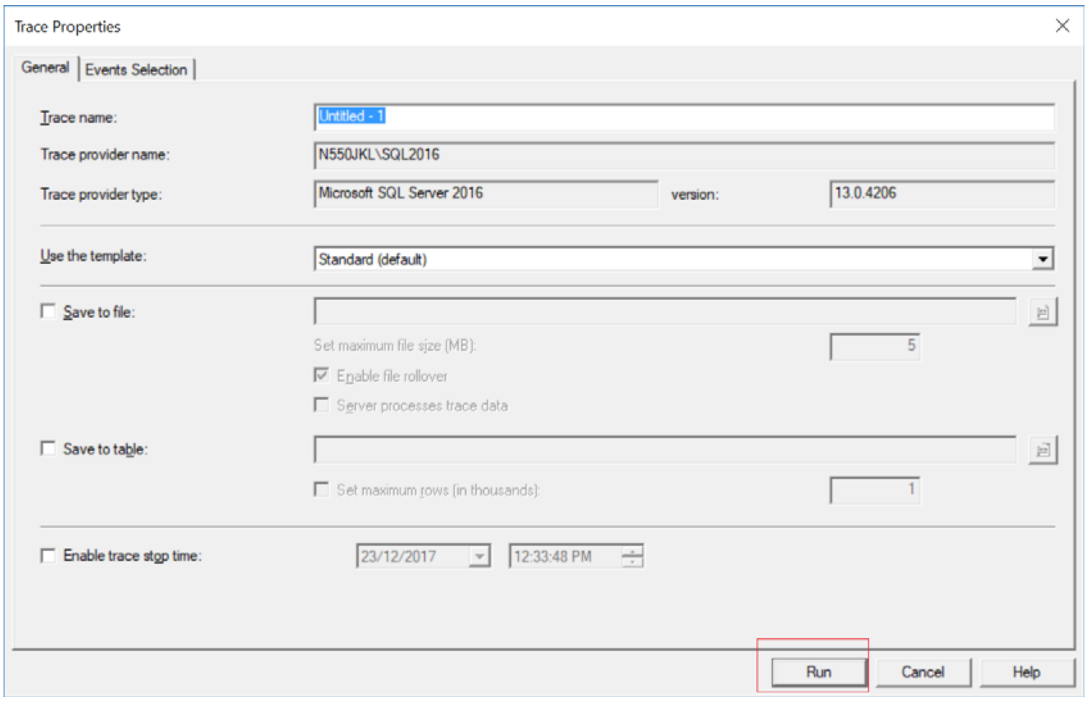
Server name: N550JKL\SQL2016

Authentication: Windows Authentication

User name: N550JKL\lpmpl

Password:

☐ Remember password


Trace Properties

Trace name: Untitled - 1

Trace provider name: N550JKL\SQL2016

Trace provider type: Microsoft SQL Server 2016 version: 13.0.4206

Use the template: Standard (default)

☐ Save to file:

Set maximum file size (MB): 5

☑ Enable file rollover

☐ Server processes trace data

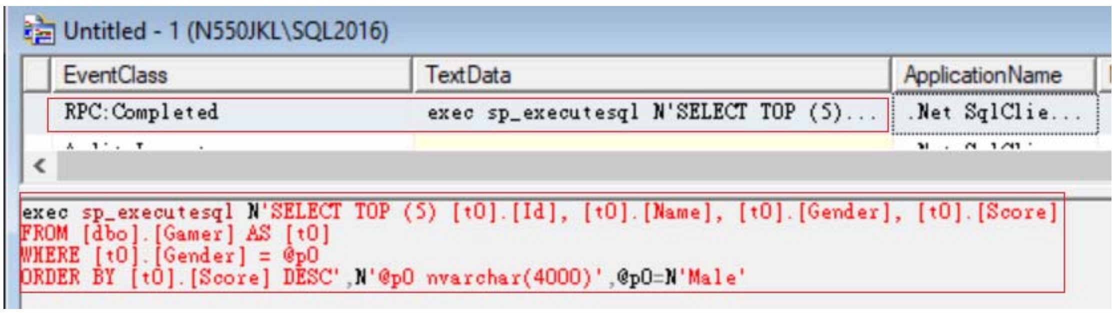
☐ Save to table:

☐ Set maximum rows (in thousands): 1

☐ Enable trace stop time: 23/12/2017 12:33:48 PM

Now, go back to VS2017, and run again
You will see Linq to SQL provider convert Linq to TSQL.

```
// 1. ==========================================
//Top5MaleByScore()
private static void Top5MaleByScore()
{
    SampleDataContext dataContext = new SampleDataContext();
    //Get Top 5 Male by Score
    IQueryable<Gamer> top5MaleByScore =
         dataContext.Gamers
         .Where(g => g.Gender == "Male")
         .OrderByDescending(g => g.Score)
         .Take(5);
    foreach (Gamer gamer in top5MaleByScore)
     {
        Console.WriteLine(gamer);
     }
}
//1.1.
//Id==8,Name==Name8,Gender==Male,Score==5500
//Id==3,Name==Name3,Gender==Male,Score==5000
//Id==6,Name==Name6,Gender==Male,Score==4500
//Id==7,Name==Name7,Gender==Male,Score==4000
//Id==1,Name==Name1,Gender==Male,Score==3500
//1.2.
//Notice that the following SQL Query is executed against the database.
//exec sp_executesql N'SELECT TOP (5) [t0].[Id], [t0].[Name], [t0].[Gender], [t0].[Score]
//FROM[dbo].[Gamer] AS[t0]
//WHERE[t0].[Gender] = @p0
//ORDER BY[t0].[Score] DESC',N'@p0 nvarchar(4000)',@p0=N'Male'
```

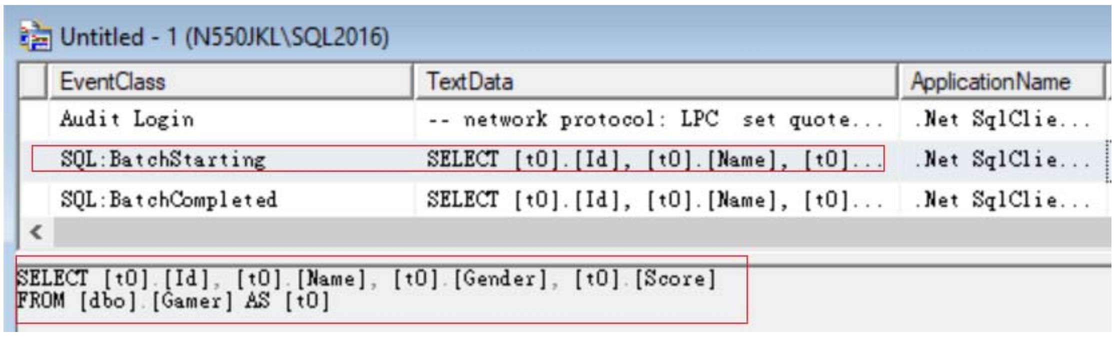

Untitled - 1 (N550JKL\SQL2016)

EventClass	TextData	ApplicationName
RPC:Completed	exec sp_executesql N'SELECT TOP (5)...	.Net SqlClie...

```
exec sp_executesql N'SELECT TOP (5) [t0].[Id], [t0].[Name], [t0].[Gender], [t0].[Score]
FROM [dbo].[Gamer] AS [t0]
WHERE [t0].[Gender] = @p0
ORDER BY [t0].[Score] DESC',N'@p0 nvarchar(4000)',@p0=N'Male'
```

```
// 2. ==========================================
//Top5MaleByScore2()
private static void Top5MaleByScore2()
{
    SampleDataContext dataContext = new SampleDataContext();
    //Get Top 5 Male by Score
    IEnumerable<Gamer> top5MaleByScore =
         dataContext.Gamers.AsEnumerable()
         .Where(g => g.Gender == "Male")
         .OrderByDescending(g => g.Score)
         .Take(5);
    foreach (Gamer gamer in top5MaleByScore)
     {
        Console.WriteLine(gamer);
     }
}
```

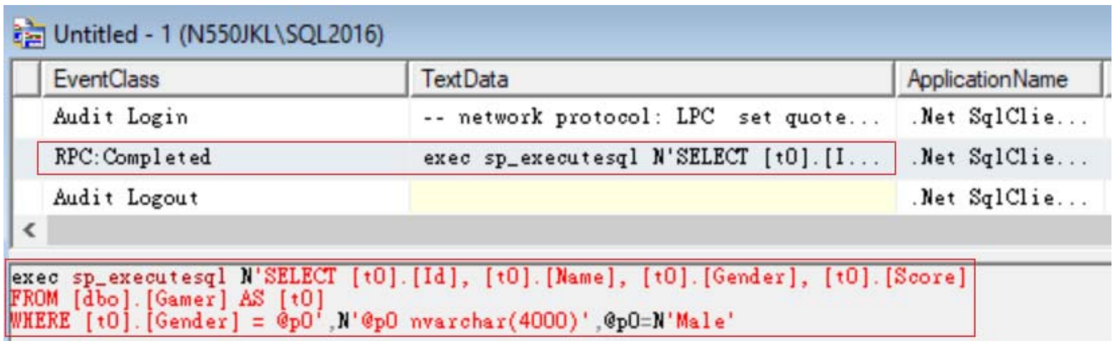
```
//1.1.
//Id==8,Name==Name8,Gender==Male,Score==5500
//Id==3,Name==Name3,Gender==Male,Score==5000
//Id==6,Name==Name6,Gender==Male,Score==4500
//Id==7,Name==Name7,Gender==Male,Score==4000
//Id==1,Name==Name1,Gender==Male,Score==3500
//1.2.
//Notice that the following SQL Query is executed against the database.
//SELECT [t0].[Id], [t0].[Name], [t0].[Gender], [t0].[Score]
//FROM[dbo].[Gamer] AS[t0]
```


Untitled - 1 (N550JKL\SQL2016)

EventClass	TextData	ApplicationName
Audit Login	-- network protocol: LPC set quote...	.Net SqlClie...
SQL:BatchStarting	SELECT [t0].[Id], [t0].[Name], [t0]...	.Net SqlClie...
SQL:BatchCompleted	SELECT [t0].[Id], [t0].[Name], [t0]...	.Net SqlClie...

```
SELECT [t0].[Id], [t0].[Name], [t0].[Gender], [t0].[Score]
FROM [dbo].[Gamer] AS [t0]
```

```
// 3. ==========================================
//Top5MaleByScore3()
private static void Top5MaleByScore3()
{
    SampleDataContext dataContext = new SampleDataContext();
    //Get Top 5 Male by Score
    IEnumerable<Gamer> top5MaleByScore =
        dataContext.Gamers
        .Where(g => g.Gender == "Male")
        .AsEnumerable()
        .OrderByDescending(g => g.Score)
        .Take(5);
    foreach (Gamer gamer in top5MaleByScore)
     {
        Console.WriteLine(gamer);
     }
}
//3.1.
//Id==8,Name==Name8,Gender==Male,Score==5500
//Id==3,Name==Name3,Gender==Male,Score==5000
//Id==6,Name==Name6,Gender==Male,Score==4500
//Id==7,Name==Name7,Gender==Male,Score==4000
//Id==1,Name==Name1,Gender==Male,Score==3500
//3.2.
//Notice that the following SQL Query is executed against the database.
//exec sp_executesql N'SELECT [t0].[Id], [t0].[Name], [t0].[Gender], [t0].[Score]
//FROM[dbo].[Gamer]
//AS[t0]
//WHERE[t0].[Gender] = @p0',N'@p0 nvarchar(4000)',@p0=N'Male'
```


Untitled - 1 (N550JKL\SQL2016)

EventClass	TextData	ApplicationName
Audit Login	-- network protocol: LPC set quote...	.Net SqlClie...
RPC:Completed	exec sp_executesql N'SELECT [t0].[I...	.Net SqlClie...
Audit Logout		.Net SqlClie...

```
exec sp_executesql N'SELECT [t0].[Id], [t0].[Name], [t0].[Gender], [t0].[Score]
FROM [dbo].[Gamer] AS [t0]
WHERE [t0].[Gender] = @p0',N'@p0 nvarchar(4000)',@p0=N'Male'
```