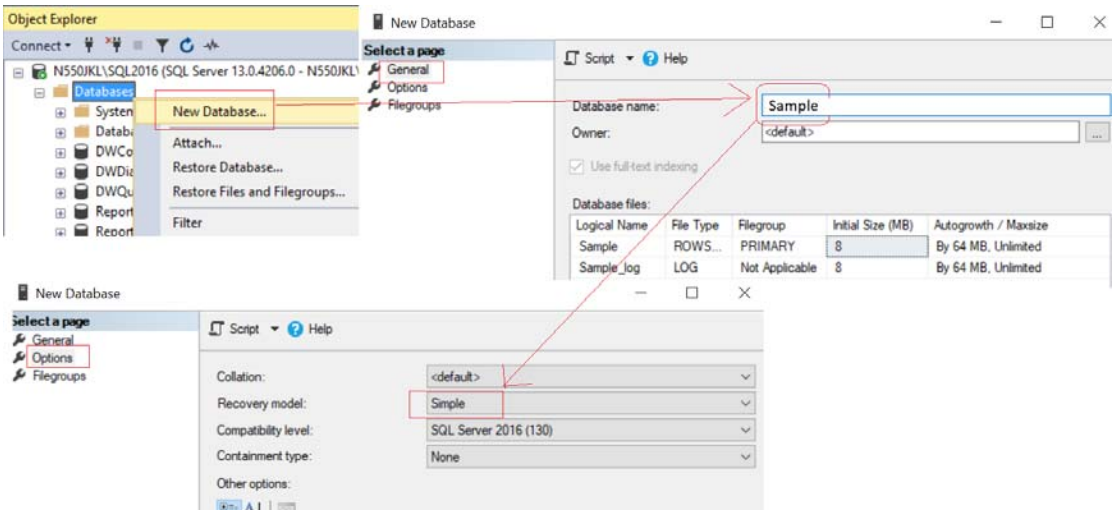
(T4)討論 EntityFrameworkDbFirst 連接資料庫。討論 LinqToObject 的 Where
CourseGUID: 5ba9a6fe-7475-4b0c-8b99-bbcf7f5e2e1c
==========
(T4)討論 EntityFrameworkDbFirst 連接資料庫。討論 LinqToObject 的 Where
==========
1. New Project
1.1. TSQL
1.2. Set up SQL Authentication
1.3. Create New Project : Sample
1.4. Entity Framework DbFirst
---
2. Sample : Program.cs
==========

1. New Project

1.1. TSQL

Database --> Right Click --> New Database -->
Database Name : **Sample**
Options --> Recovery Model : Simple



--Create an Sample DataBase and Run the following TSQL

```
--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
```

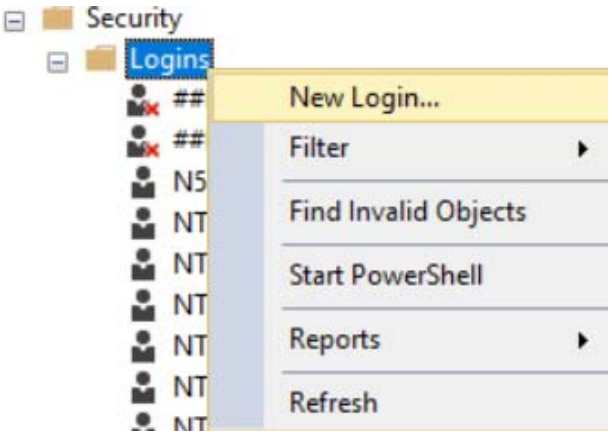
```
--IF OBJECT_ID('Team') IS NOT NULL
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.TABLES
              WHERE     TABLE_NAME = 'Team' ) )
    BEGIN
        TRUNCATE TABLE Team;
        DROP TABLE Team;
    END;
GO -- Run the previous command and begins new batch
CREATE TABLE Team
    (
      Id INT PRIMARY KEY
             IDENTITY(1, 1) ,
      Name NVARCHAR(50) ,
    );
GO -- Run the previous command and begins new batch
INSERT  INTO Team
VALUES  ( 'Team01');
INSERT  INTO Team
VALUES  ( 'Team02');
INSERT  INTO Team
VALUES  ( 'Team03');
GO -- Run the previous command and begins new batch
CREATE TABLE Gamer
    (
      Id INT PRIMARY KEY
             IDENTITY(1, 1) ,
      [Name] NVARCHAR(100) ,
      Gender NVARCHAR(50) ,
      GameScore INT ,
      TeamId INT FOREIGN KEY REFERENCES Team ( Id )
    );
GO -- Run the previous command and begins new batch
INSERT  INTO Gamer
VALUES  ( 'Name01', 'Male', 5000, 1 );
INSERT  INTO Gamer
VALUES  ( 'Name02', 'Female', 4500, 2 );
INSERT  INTO Gamer
VALUES  ( 'Name03', 'Male', 6000, 1 );
INSERT  INTO Gamer
VALUES  ( 'Name04', 'Male', 3500, 2 );
INSERT  INTO Gamer
VALUES  ( 'Name05', 'Male', 4700, 2 );
INSERT  INTO Gamer
VALUES  ( 'Name06', 'Male', 4800, 1 );
GO -- Run the previous command and begins new batch
```

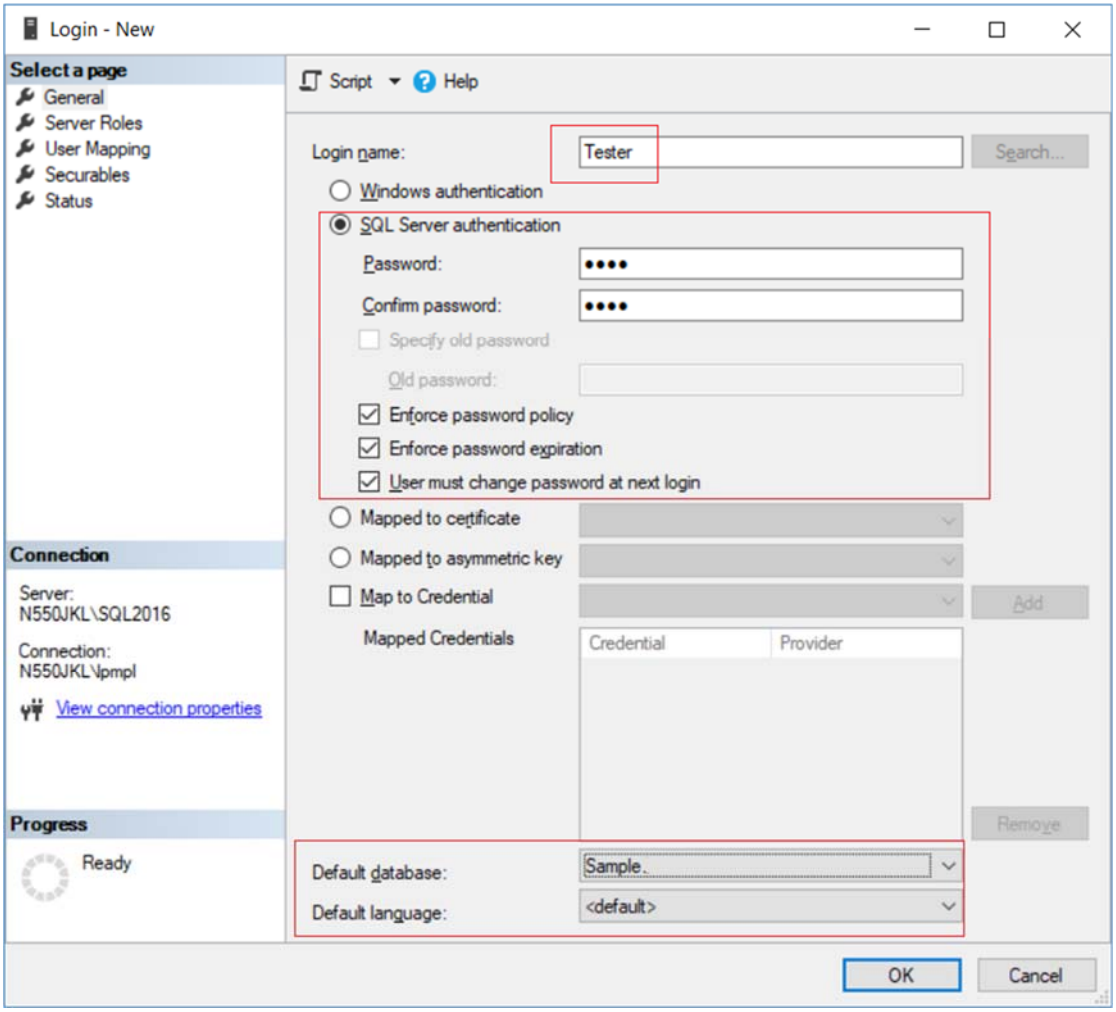
1.2. Set up SQL Authentication

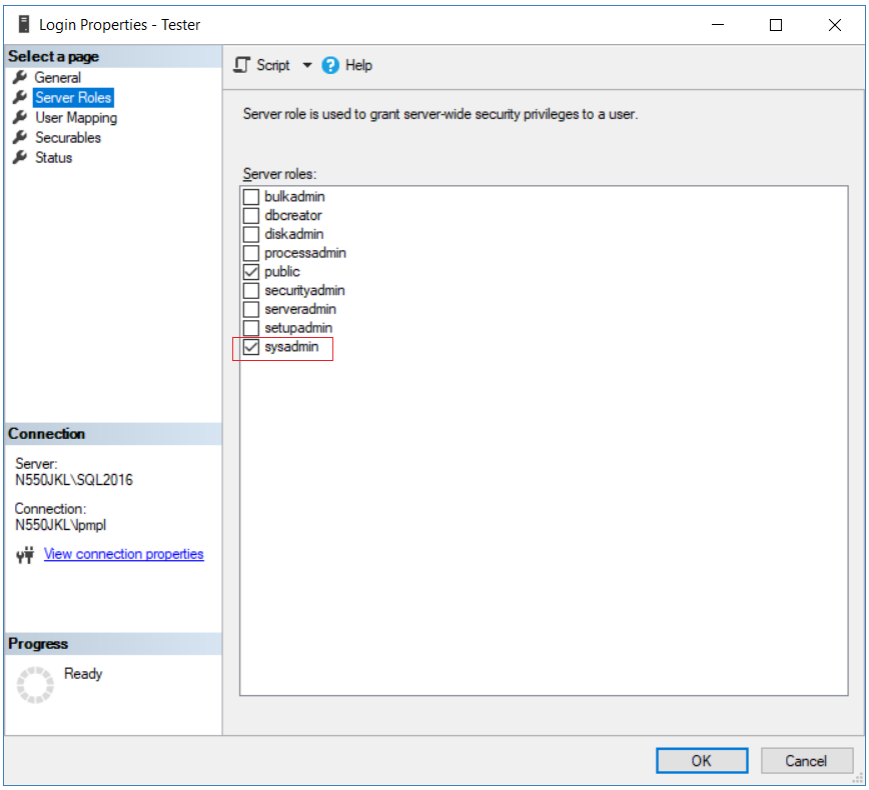
In SQL server

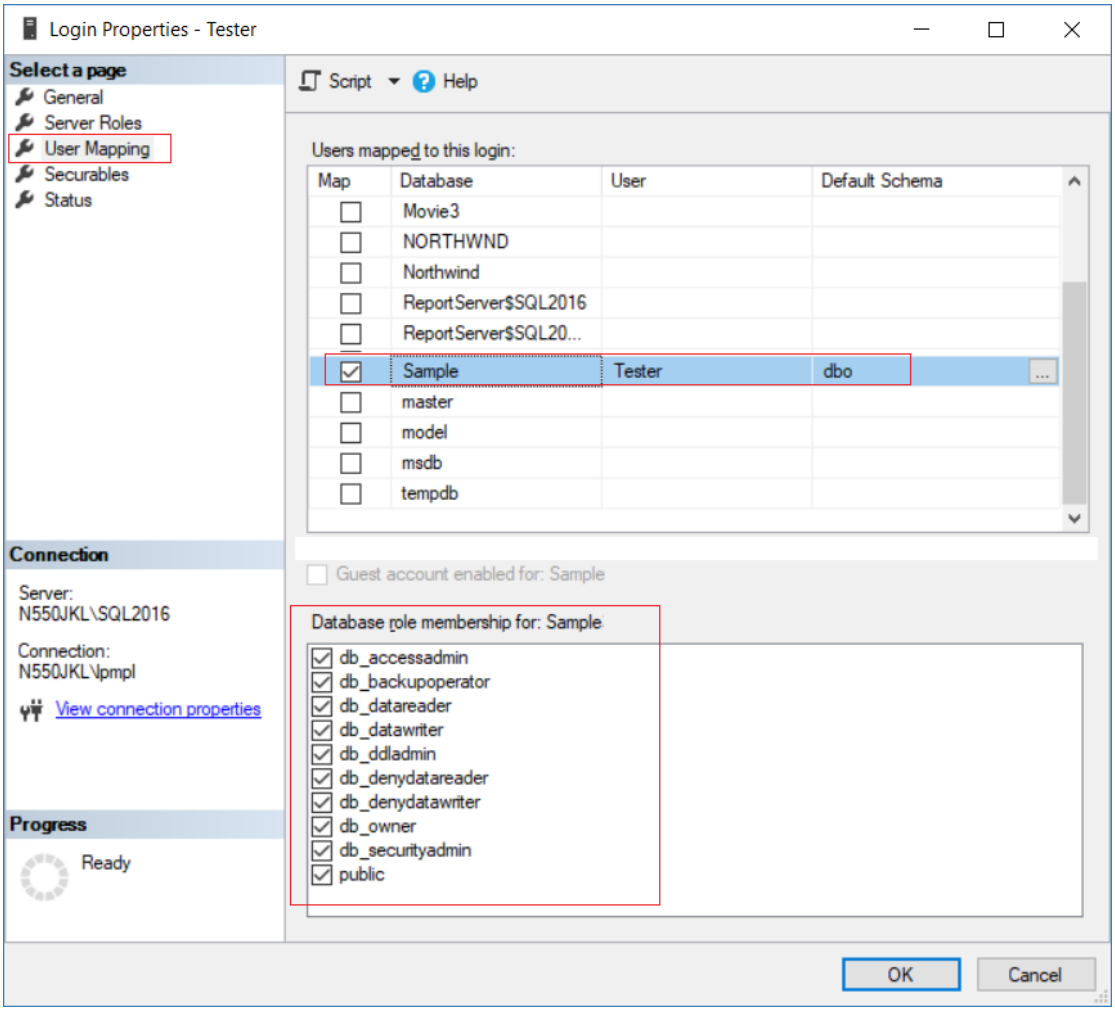
Object Explorer --> Security --> Logins --> New Logins

-->
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
Search...
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
Add
Mapped Credentials
Credential
Provider
Remove
Default database:
Sample
Default language:
<default>
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
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

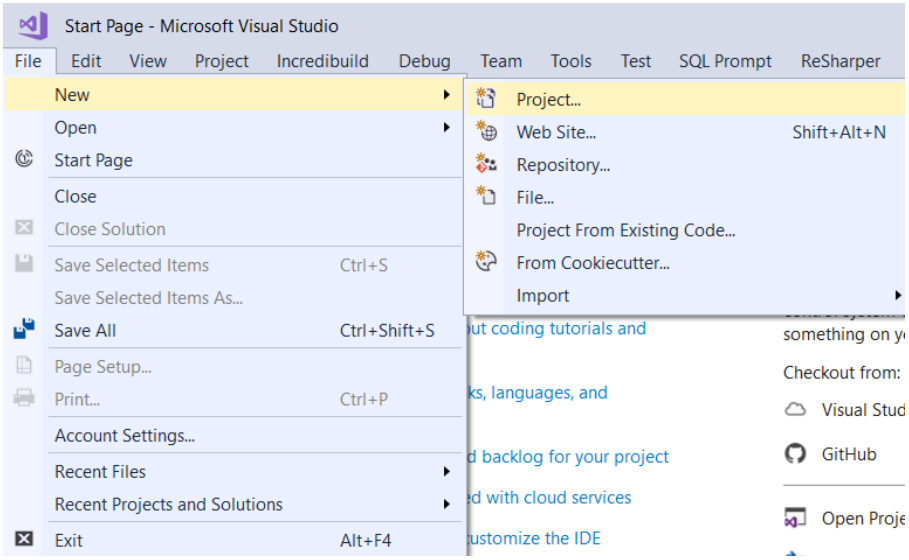


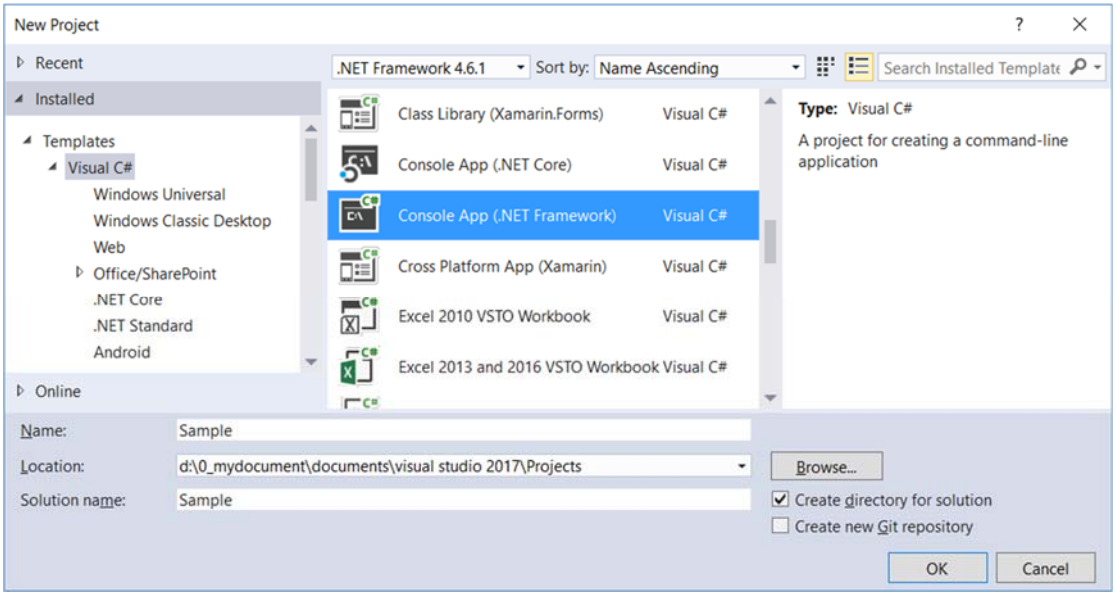
1.3. Create New Project : Sample

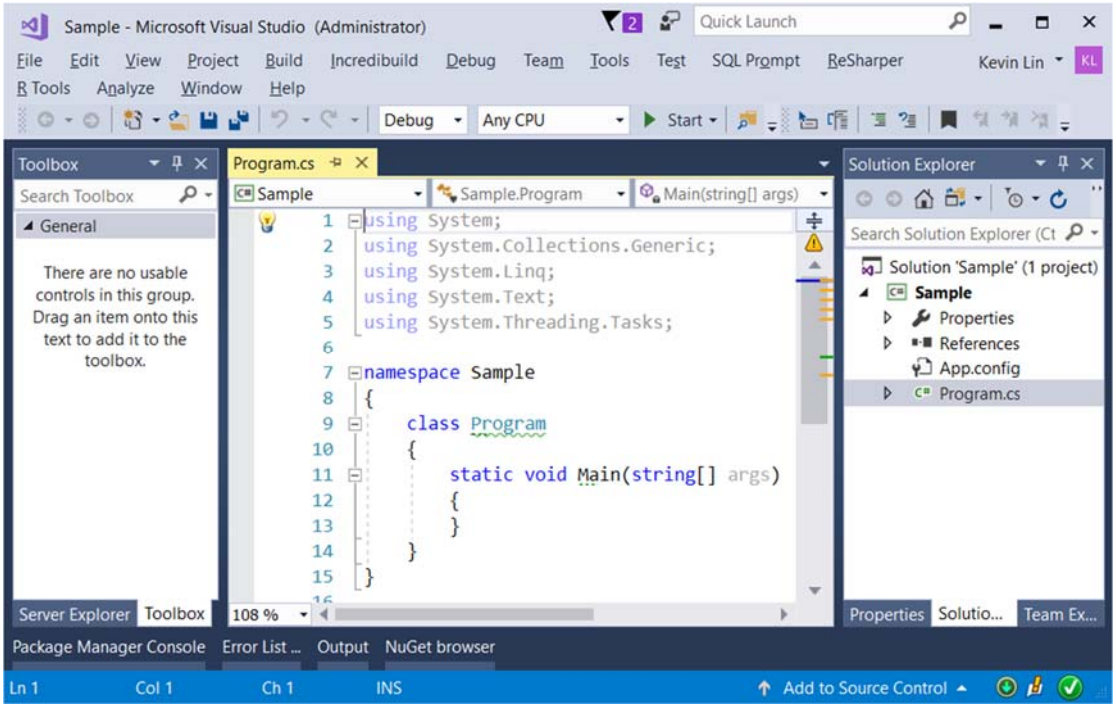
File --> New --> Project... -->

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

Name: **Sample**





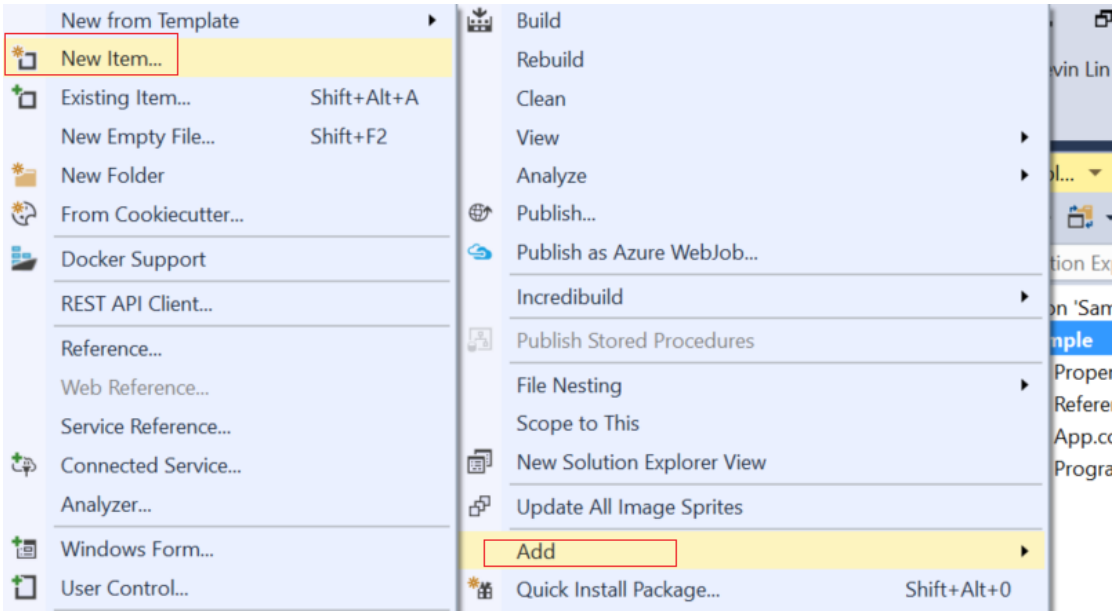


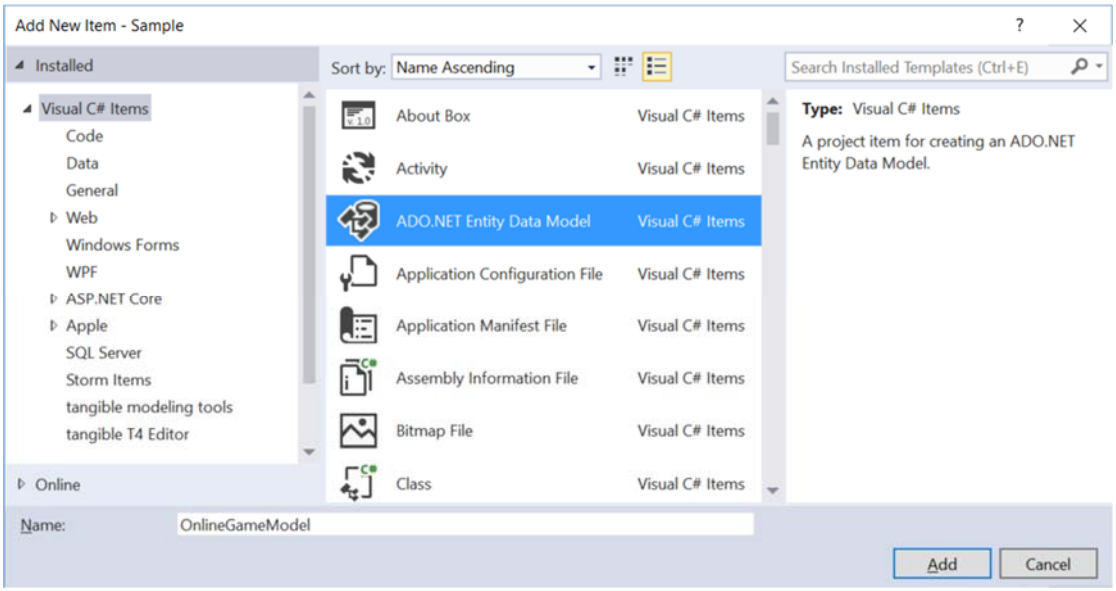
1.4. Entity Framework DbFirst

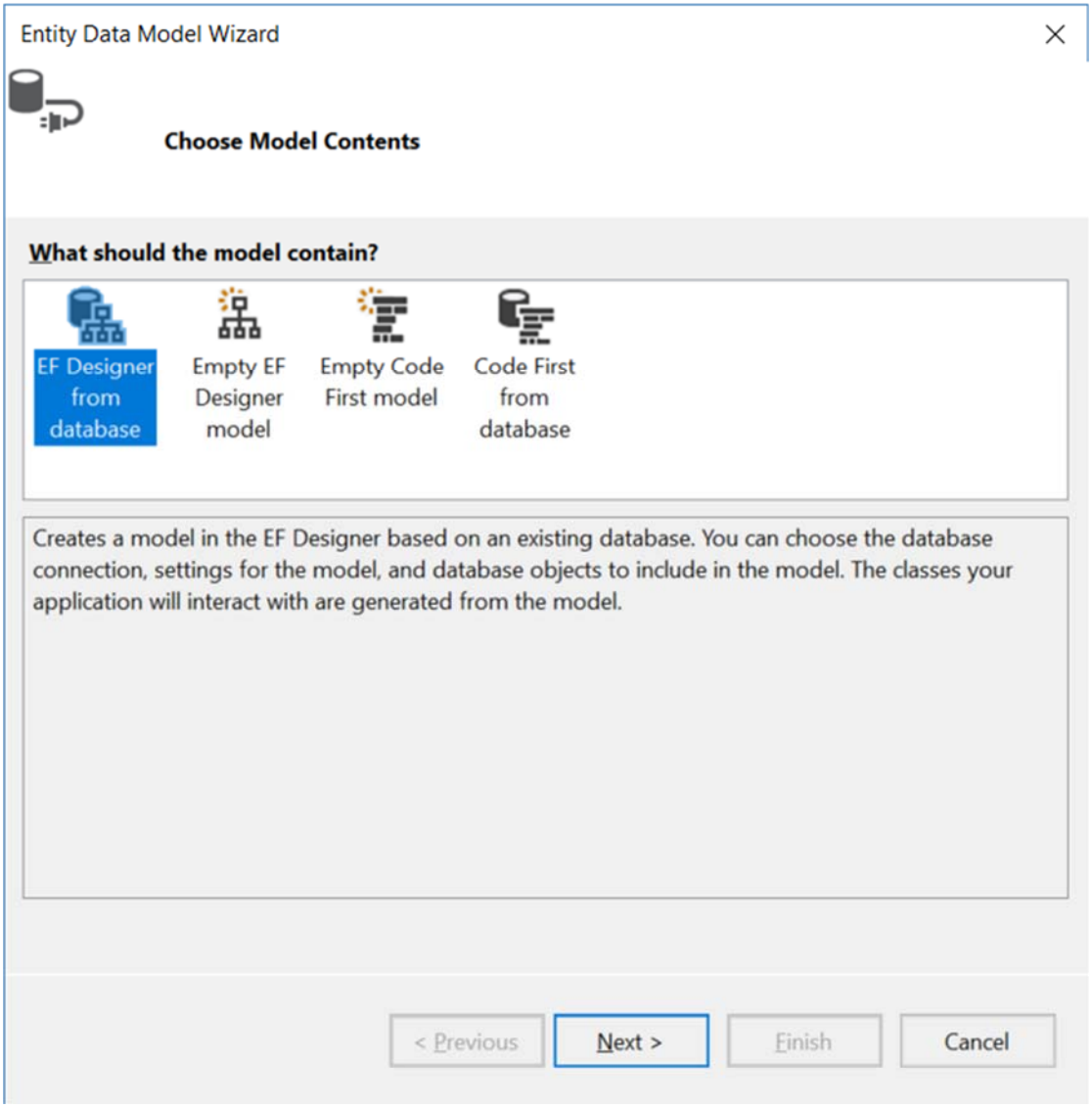
Project Name --> Right Click --> Add --> New Item ...

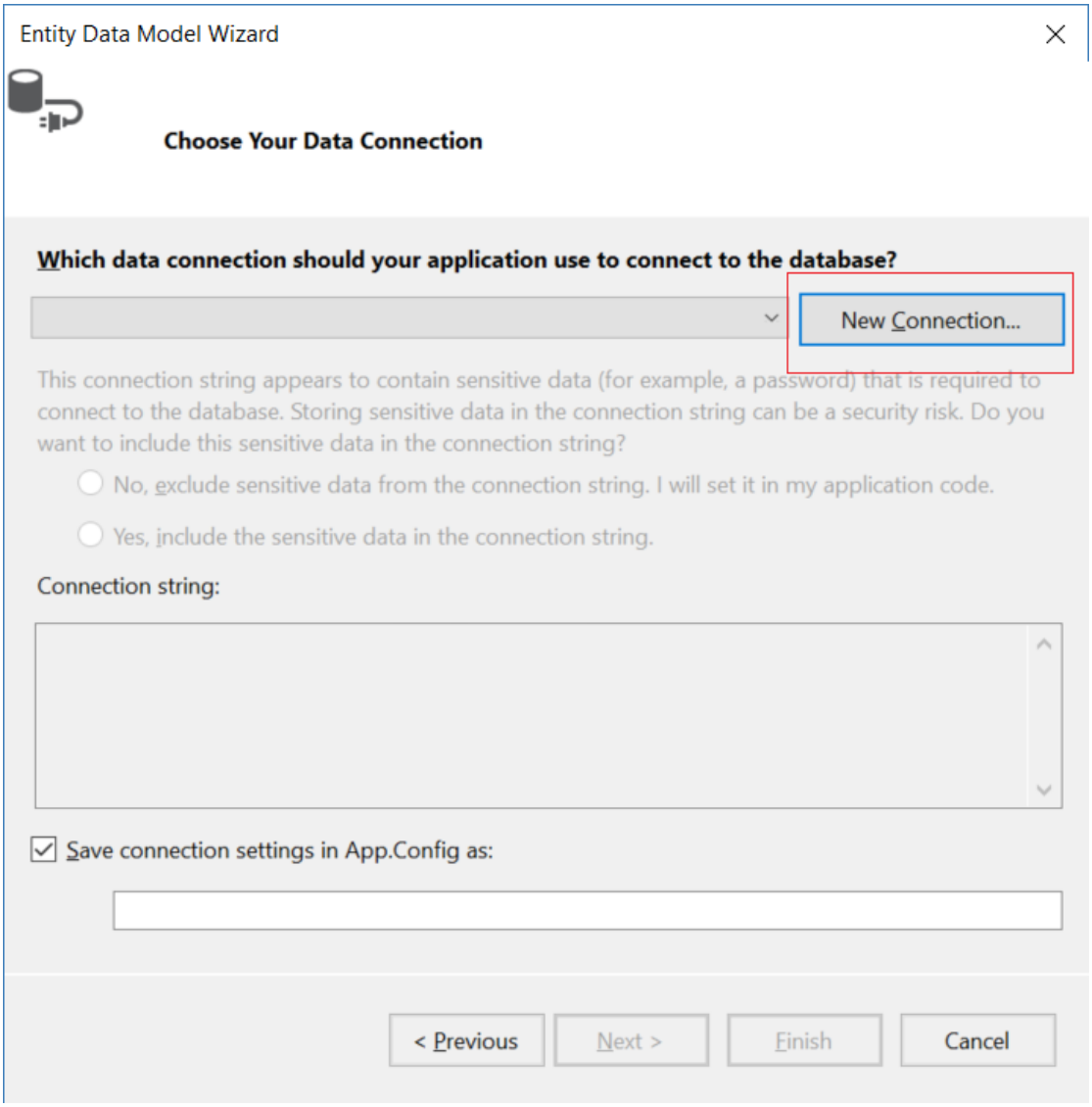
Visual C# --> **ADO.NET Entity Data Model** -->

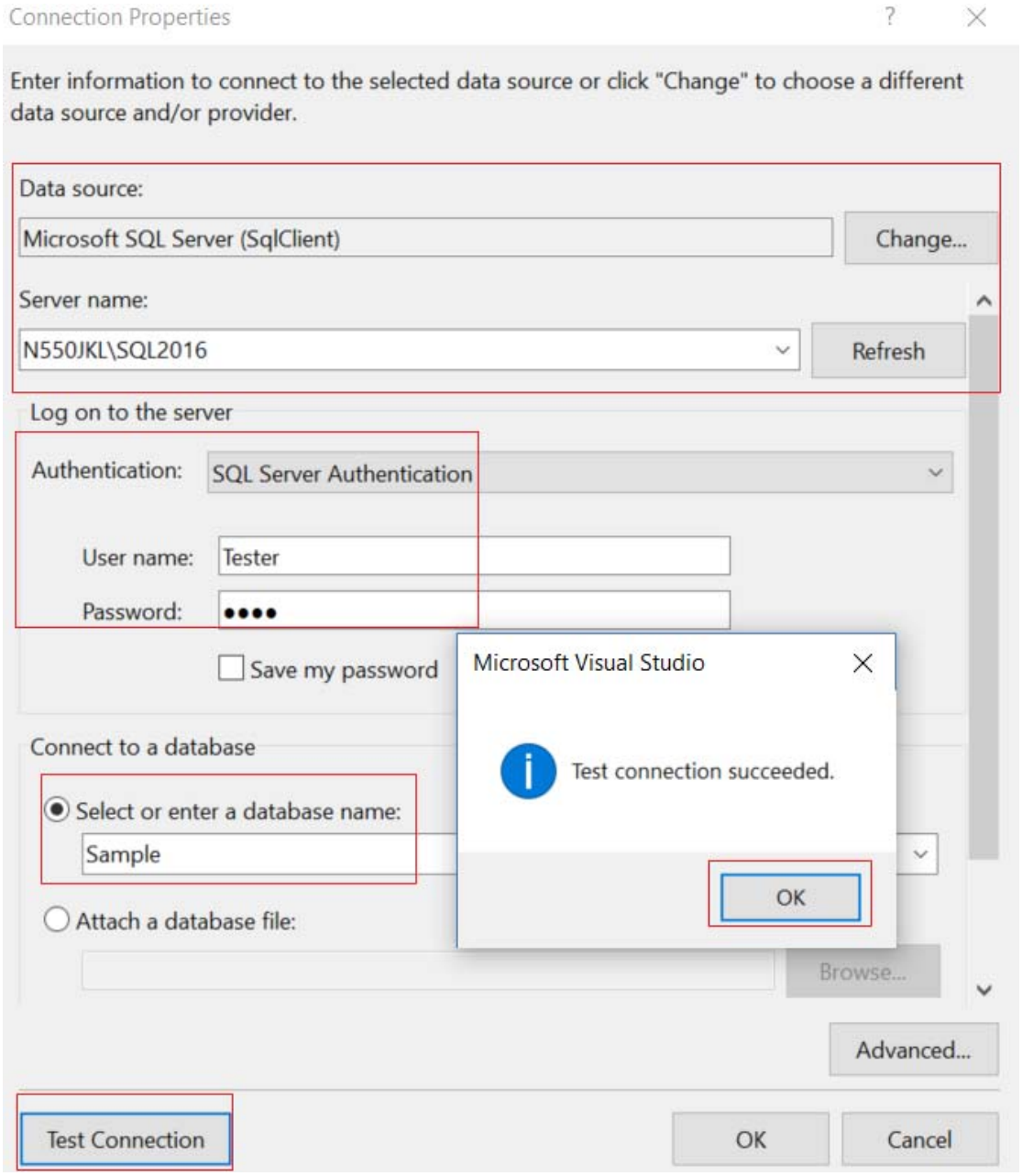
Name: **OnlineGameModel**

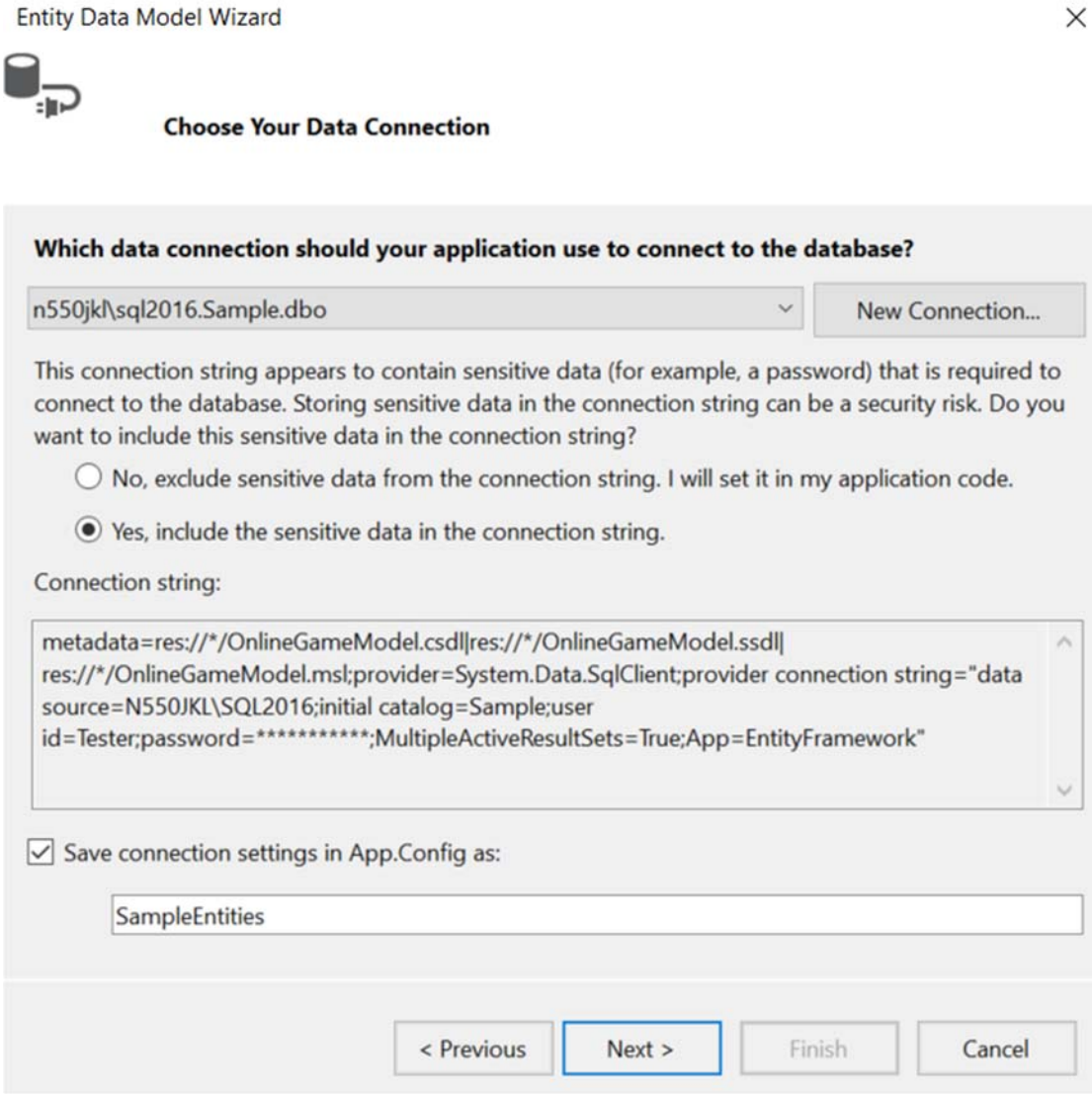

New from Template
New Item...
Existing Item... Shift+Alt+A
New Empty File... Shift+F2
New Folder
From Cookiecutter...
Docker Support
REST API Client...
Reference...
Web Reference...
Service Reference...
Connected Service...
Analyzer...
Windows Form...
User Control...
Build
Rebuild
Clean
View
Analyze
Publish...
Publish as Azure WebJob...
Incredibuild
Publish Stored Procedures
File Nesting
Scope to This
New Solution Explorer View
Update All Image Sprites
Add
Quick Install Package... Shift+Alt+0

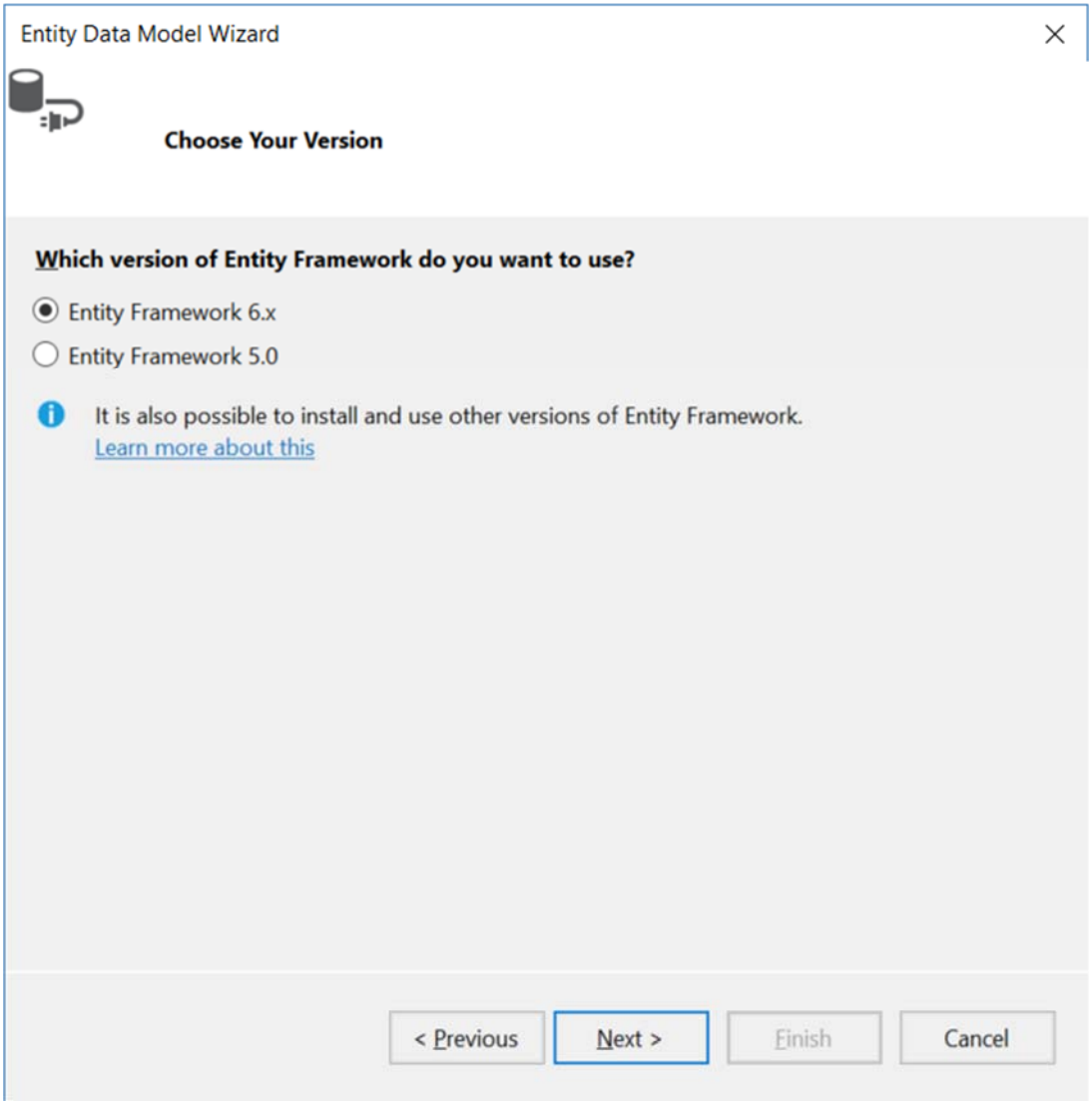

Add New Item - Sample
Installed
Visual C# Items
Code
Data
General
Web
Windows Forms
WPF
ASP.NET Core
Apple
SQL Server
Storm Items
tangible modeling tools
tangible T4 Editor
Online
Sort by: Name Ascending
Search Installed Templates (Ctrl+E)
About Box Visual C# Items
Activity Visual C# Items
ADO.NET Entity Data Model Visual C# Items
Application Configuration File Visual C# Items
Application Manifest File Visual C# Items
Assembly Information File Visual C# Items
Bitmap File Visual C# Items
Class Visual C# Items
Type: Visual C# Items
A project item for creating an ADO.NET Entity Data Model.
Name: OnlineGameModel
Add
Cancel

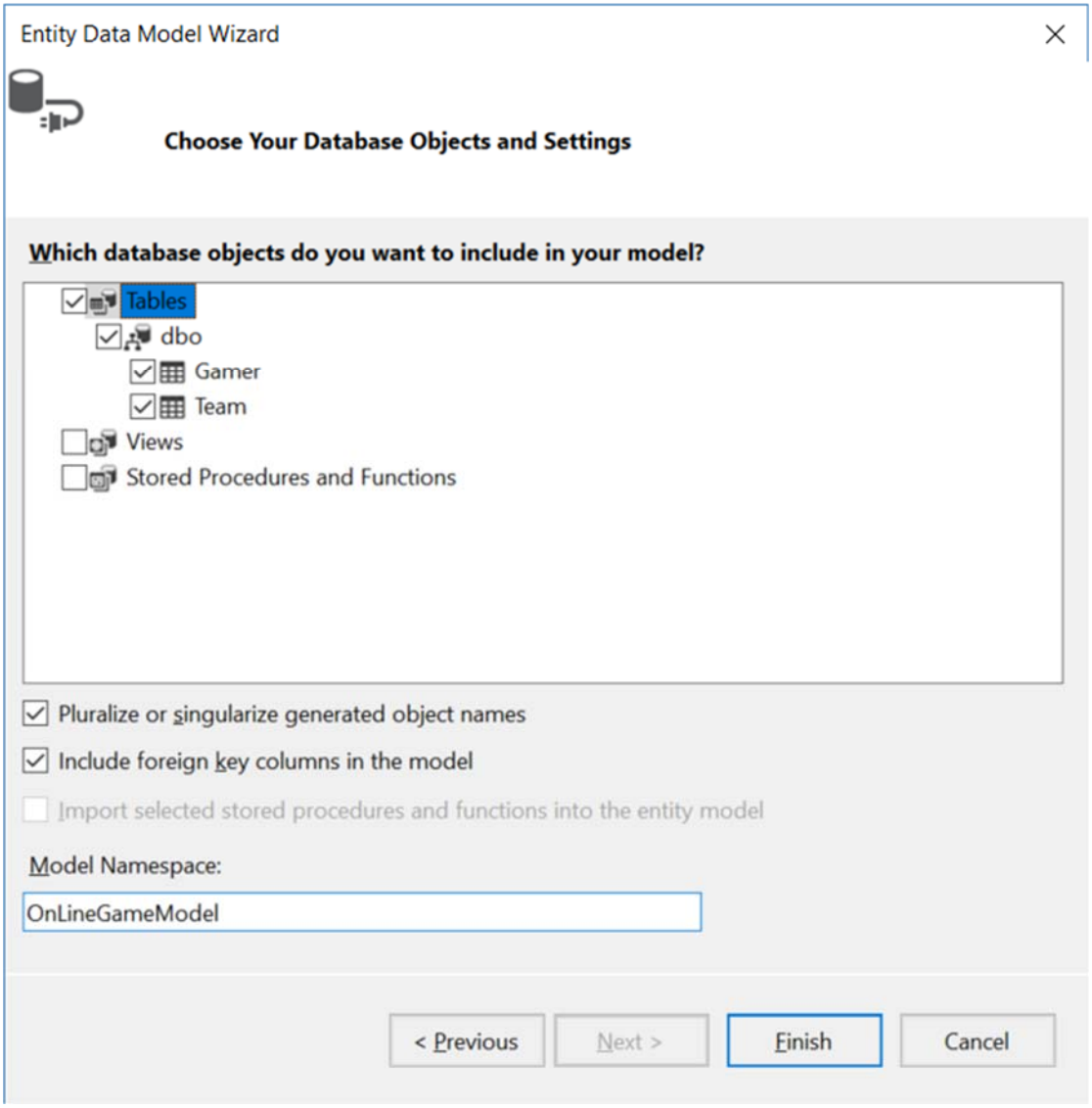

Entity Data Model Wizard
Choose Model Contents
What should the model contain?
EF Designer from database
Empty EF Designer model
Empty Code First model
Code First from database
Creates a model in the EF Designer based on an existing database. You can choose the database connection, settings for the model, and database objects to include in the model. The classes your application will interact with are generated from the model.
< Previous
Next >
Finish
Cancel

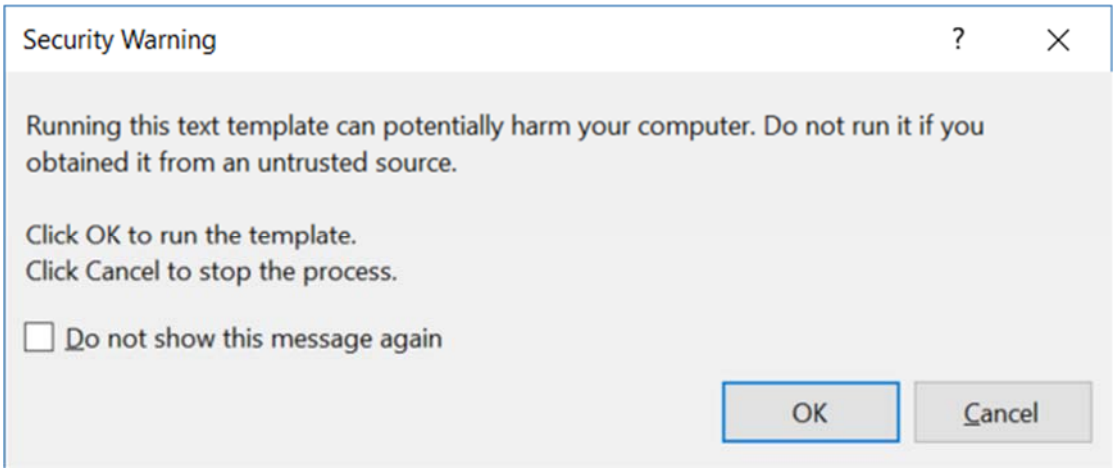

Entity Data Model Wizard
Choose Your Data Connection
Which data connection should your application use to connect to the database?
New Connection...
This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in 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:
Save connection settings in App.Config as:
< Previous
Next >
Finish
Cancel

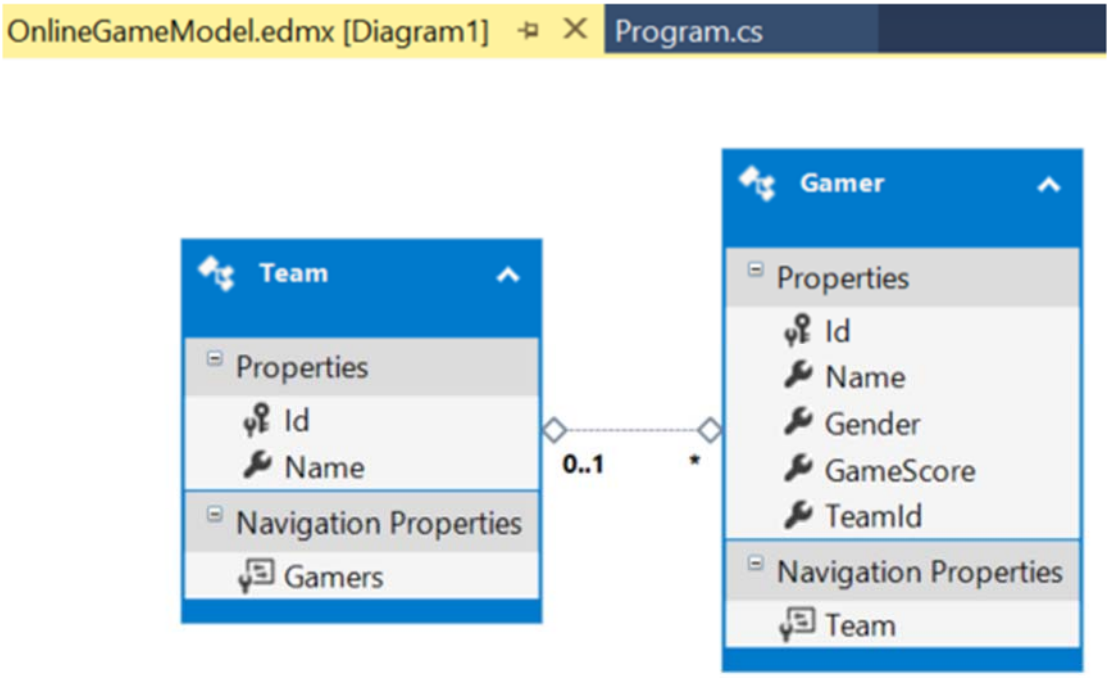

Connection Properties
Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.
Data source:
Microsoft SQL Server (SqlClient)
Change...
Server name:
N550JKL\SQL2016
Refresh
Log on to the server
Authentication:
SQL Server Authentication
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


Entity Data Model Wizard
Choose Your Data Connection
Which data connection should your application use to connect to the database?
n550jkl\sql2016.Sample.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://*/OnlineGameModel.csdl|res://*/OnlineGameModel.ssdl|
res://*/OnlineGameModel.msl;provider=System.Data.SqlClient;provider connection string="data
source=N550JKL\SQL2016;initial catalog=Sample;user
id=Tester;password=**********;MultipleActiveResultSets=True;App=EntityFramework"
Save connection settings in App.Config as:
SampleEntities
< Previous
Next >
Finish
Cancel


Entity Data Model Wizard
Choose Your Version
Which version of Entity Framework do you want to use?
Entity Framework 6.x
Entity Framework 5.0
It is also possible to install and use other versions of Entity Framework.
Learn more about this
< Previous
Next >
Finish
Cancel


Entity Data Model Wizard
Choose Your Database Objects and Settings
Which database objects do you want to include in your model?
Tables
dbo
Gamer
Team
Views
Stored Procedures and Functions
Pluralize or singularize generated object names
Include foreign key columns in the model
Import selected stored procedures and functions into the entity model
Model Namespace:
OnLineGameModel
< Previous
Next >
Finish
Cancel


Security Warning
Running this text template can potentially harm your computer. Do not run it if you obtained it from an untrusted source.
Click OK to run the template.
Click Cancel to stop the process.
Do not show this message again
OK
Cancel



==========

2. Sample : Program.cs

```
using System;
using System.Data.Entity;
using System.Linq;
namespace Sample
{
    class Program
    {
        static void Main(string[] args)
        {
            // 1. ============================================
            Console.WriteLine("1. ADO.NET with Entity Framework 6 Sample");
             AdoNetWithEntityFramework6Sample();
            Console.ReadLine();
        }
        private static void AdoNetWithEntityFramework6Sample()
        {
            SampleEntities context = new SampleEntities();
            //1.1. teamDbSet -------------------------------
            Console.WriteLine("1.1. teamDbSet ------------");
            DbSet<Team> teamDbSet = context.Teams;
            foreach (Team teamDbSetItem in teamDbSet)
            {
                Console.WriteLine($"teamDbSetItem.Id=={teamDbSetItem.Id},
teamDbSetItem.Name=={teamDbSetItem.Name}");
            }
            //1.2. gamersDbSet -----------------------------
            Console.WriteLine("1.2. gamersDbSet ------------");
            DbSet<Gamer> gamersDbSet = context.Gamers;
```

```
            foreach (Gamer gamersDbSetItem in gamersDbSet)
            {
                Console.WriteLine($"gamersDbSetItem.Id=={gamersDbSetItem.Id},
gamersDbSetItem.Name=={gamersDbSetItem.Name}, gamersDbSetItem.Gender=={gamersDbSetItem.Gender},
gamersDbSetItem.GameScore=={gamersDbSetItem.GameScore}, gamersDbSetItem.TeamId=={gamersDbSetItem.TeamId},
gamersDbSetItem.Team.Name=={gamersDbSetItem.Team.Name}");
            }
            //1.3. team01AndTeam02Queryable ------------------------------
            Console.WriteLine("1.3. team01AndTeam02Queryable ------------");
            IQueryable<Team> team01AndTeam02Queryable = context.Teams.Where(t => t.Name.Equals("Team01")
|| t.Name.Equals("Team02"));
            foreach (Team team01AndTeam02QueryableItem in team01AndTeam02Queryable)
            {
                Console.WriteLine($"team01AndTeam02QueryableItem.Id=={team01AndTeam02QueryableItem.Id},
team01AndTeam02QueryableItem.Name=={team01AndTeam02QueryableItem.Name}");
            }
            //1.4. maleGamersInTeam01Item ------------------------------
            Console.WriteLine("1.4. maleGamersInTeam01Item ------------");
            IQueryable<Gamer> maleGamersInTeam01 = context.Gamers.Where(g => g.Team.Name.Equals("Team01")
&& g.Gender.Equals("Male"));
            foreach (Gamer maleGamersInTeam01Item in maleGamersInTeam01)
            {
                Console.WriteLine($"maleGamersInTeam01Item.Id=={maleGamersInTeam01Item.Id},
maleGamersInTeam01Item.Name=={maleGamersInTeam01Item.Name},
maleGamersInTeam01Item.Gender=={maleGamersInTeam01Item.Gender},
maleGamersInTeam01Item.Team.Name=={maleGamersInTeam01Item.Team.Name}");
            }
        }
    }
}
```

```
1. ADO.NET with Entity Framework 6 Sample
1.1. teamDbSet -----------
teamDbSetItem.Id==1, teamDbSetItem.Name==Team01
teamDbSetItem.Id==2, teamDbSetItem.Name==Team02
teamDbSetItem.Id==3, teamDbSetItem.Name==Team03
1.2. gamersDbSet -----------
gamersDbSetItem.Id==1, gamersDbSetItem.Name==Name01, gamersDbSetItem.Gender==Male, gamersDbSetItem.GameScore==5000, gamersDbSetItem.TeamId==1, ga
mersDbSetItem.Team.Name==Team01
gamersDbSetItem.Id==2, gamersDbSetItem.Name==Name02, gamersDbSetItem.Gender==Female, gamersDbSetItem.GameScore==4500, gamersDbSetItem.TeamId==2,
gamersDbSetItem.Team.Name==Team02
gamersDbSetItem.Id==3, gamersDbSetItem.Name==Name03, gamersDbSetItem.Gender==Male, gamersDbSetItem.GameScore==6000, gamersDbSetItem.TeamId==1, ga
mersDbSetItem.Team.Name==Team01
gamersDbSetItem.Id==4, gamersDbSetItem.Name==Name04, gamersDbSetItem.Gender==Male, gamersDbSetItem.GameScore==3500, gamersDbSetItem.TeamId==2, ga
mersDbSetItem.Team.Name==Team02
gamersDbSetItem.Id==5, gamersDbSetItem.Name==Name05, gamersDbSetItem.Gender==Male, gamersDbSetItem.GameScore==4700, gamersDbSetItem.TeamId==2, ga
mersDbSetItem.Team.Name==Team02
gamersDbSetItem.Id==6, gamersDbSetItem.Name==Name06, gamersDbSetItem.Gender==Male, gamersDbSetItem.GameScore==4800, gamersDbSetItem.TeamId==1, ga
mersDbSetItem.Team.Name==Team01
1.3. team01AndTeam02Queryable ------------
team01AndTeam02QueryableItem.Id==1, team01AndTeam02QueryableItem.Name==Team01
team01AndTeam02QueryableItem.Id==2, team01AndTeam02QueryableItem.Name==Team02
1.4. maleGamersInTeam01Item ------------
maleGamersInTeam01Item.Id==1, maleGamersInTeam01Item.Name==Name01, maleGamersInTeam01Item.Gender==Male, maleGamersInTeam01Item.Team.Name==Team01
maleGamersInTeam01Item.Id==3, maleGamersInTeam01Item.Name==Name03, maleGamersInTeam01Item.Gender==Male, maleGamersInTeam01Item.Team.Name==Team01
maleGamersInTeam01Item.Id==6, maleGamersInTeam01Item.Name==Name06, maleGamersInTeam01Item.Gender==Male, maleGamersInTeam01Item.Team.Name==Team01
```