

(T20)討論 LinqToSql 的 SingleTableInheritance，將 1Table 分成 3Class

CourseGUID: 5ba9a6fe-7475-4b0c-8b99-bbcf7f5e2e1c

(T20)討論 LinqToSql 的 SingleTableInheritance，將 1Table 分成 3Class

1. Web Form Application - Linq Query

1.1. TSQL

1.2. Set up SQL Authentication

2. Web Form App

2.1. Web.config

2.2. Linq to SQL

2.2.1. Add Connection

2.2.2. Sample.dbml

2.2.3. Sample.dbml - Single Table Inheritance

2.2.4. Sample.dbml - Single Table Inheritance - Set Discriminator 識別者

2.3. WebForm1.aspx

2.3.1. WebForm1.aspx

2.3.2. WebForm1.aspx.cs

2.4. WebForm1.aspx (Fix "btnGetAllGamers" issue)

2.4.1. dbml (Fix "btnGetAllGamers" issue)

2.4.2. WebForm1.aspx (Fix "btnGetAllGamers" issue)

2.4.3. WebForm1.aspx.cs (Fix "btnGetAllGamers" issue)

2.5. WebForm1.aspx (btnAddGamers)

2.5.1. WebForm1.aspx (btnAddGamers)

2.5.2. WebForm1.aspx.cs (btnAddGamers)

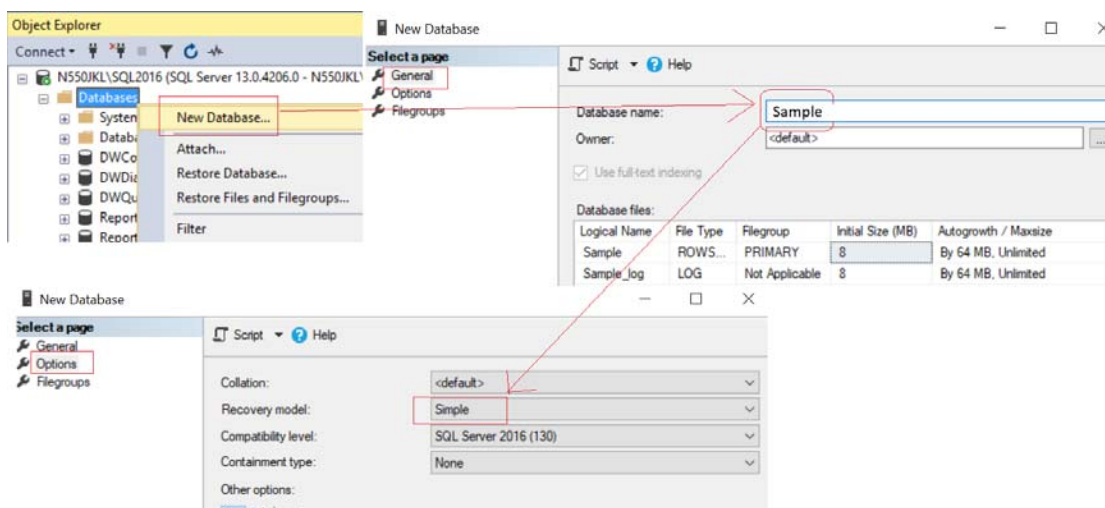
1. Web Form Application - Linq Query

1.1. TSQL

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

Database Name : Sample

Options --> Recovery Model : Simple



```

--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
CREATE TABLE Gamer
(
    Id INT PRIMARY KEY
        IDENTITY ,
    Name NVARCHAR(50) ,
    Gender NVARCHAR(50) ,
    Score INT ,
    Type NVARCHAR(50) ,
    CombatPower INT,
    MagicPower INT
);
GO -- Run the previous command and begins new batch
--2 -----
INSERT INTO Gamer
VALUES ( 'Name1 ABC', 'Male', 5000, 'Warrior', 500, NULL );
INSERT INTO Gamer
VALUES ( 'Name2 ABCDE', 'Female', 4500, 'Warrior', 350, NULL );
INSERT INTO Gamer
VALUES ( 'Name3 EFGH', 'Male', 6500, 'Magician', NULL, 600 );
INSERT INTO Gamer
VALUES ( 'Name4 HIJKLMN', 'Female', 45000, 'Magician', NULL, 650 );
INSERT INTO Gamer
VALUES ( 'Name5 NOP', 'Male', 3000, 'Magician', NULL, 700 );
INSERT INTO Gamer
VALUES ( 'Name6 PQRSTUWV', 'Male', 4000, 'Warrior', 450, NULL );
INSERT INTO Gamer
VALUES ( 'Name7 XYZ', 'Male', 4500, 'Warrior', 550, 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

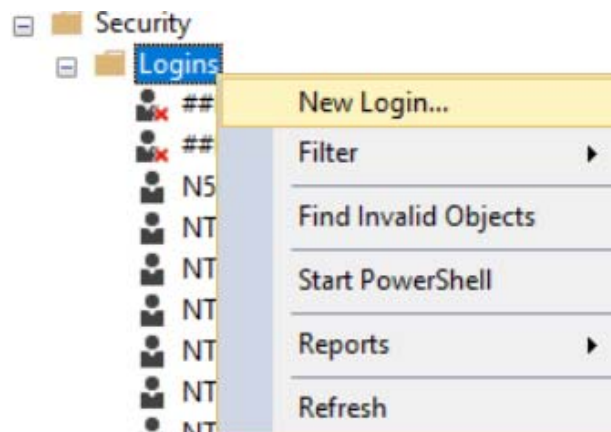
sysadmin

-->

User Mapping Tab

Select Sample

Select every Roles.



Login - New

Select a page

- General
- Server Roles
- User Mapping
- Securables
- Status

Script ? Help

Login name: 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

Mapped Credentials

Credential	Provider
------------	----------

Default database:

Default language:

OK Cancel

Connection

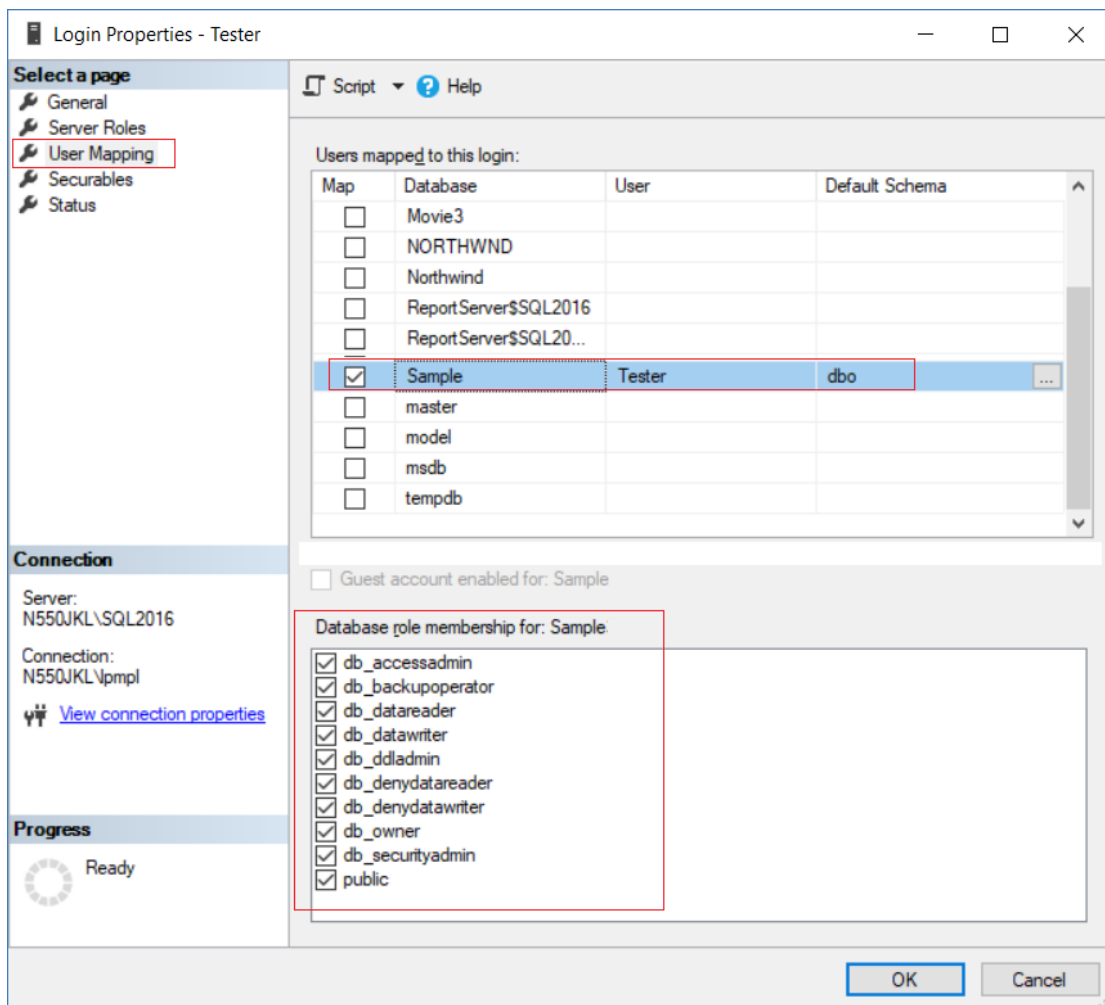
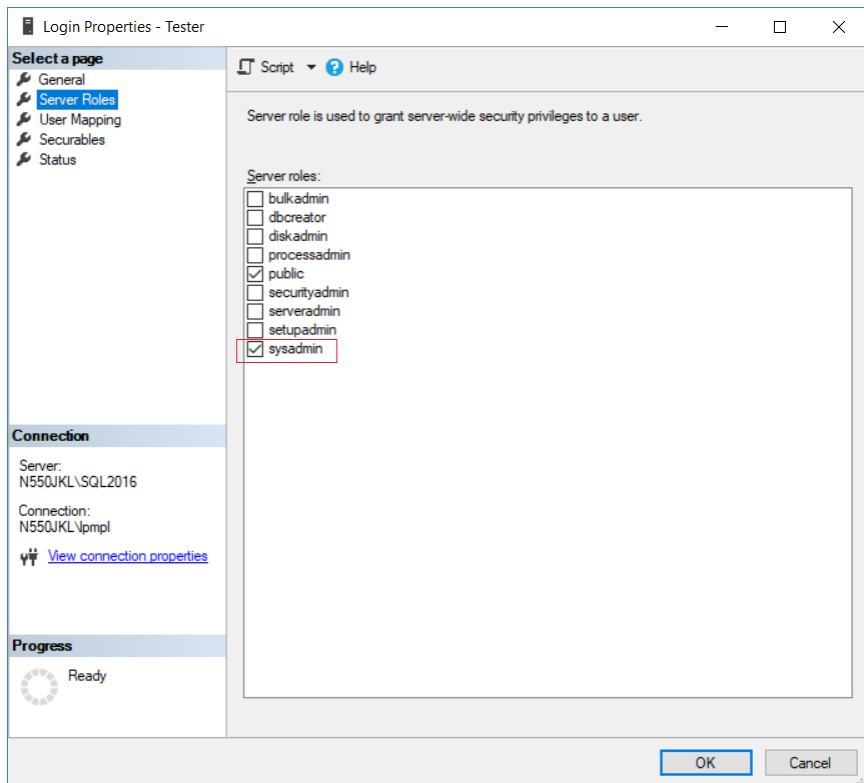
Server: N55QJKL\SQL2016

Connection: N55QJKL\pmpl

[View connection properties](#)

Progress

Ready



2. 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>

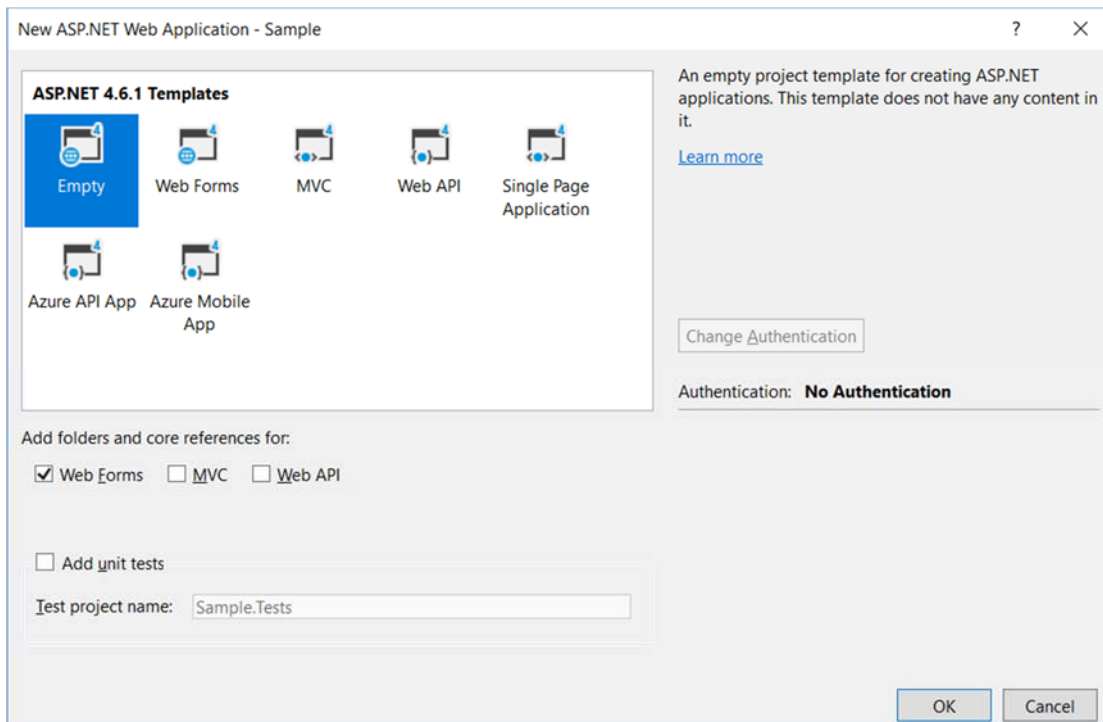
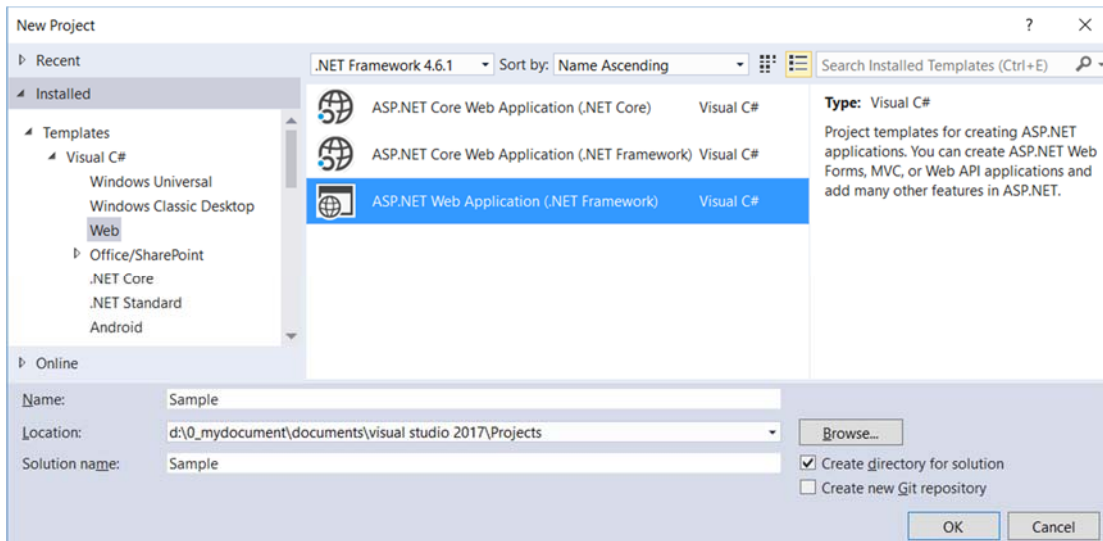
New Project --> Web --> [ASP.NET](#) Web Application (.Net Framework)

-->

Name:

Sample

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



2.1. 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>

```

```

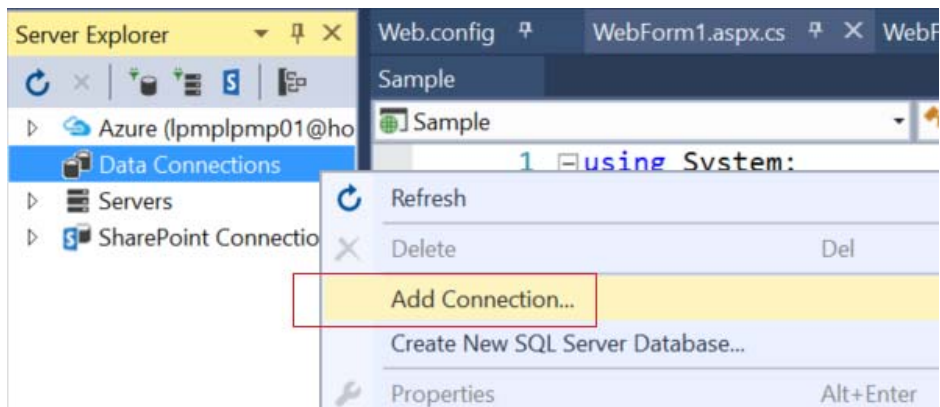
Web.config  Sample
1  <?xml version="1.0" encoding="utf-8"?>
2  <!--
3  For more information on how to configure your ASP.NET application, please visit
4  https://go.microsoft.com/fwlink/?linkid=169433
5  -->
6  <configuration>
7  <connectionStrings>
8    <add name="SampleConnectionString" connectionString="Data Source=N550JKL\SQL2016;Initial Catalog=Sample;User ID=Tester;Password=1234"
9    providerName="System.Data.SqlClient" />
10 </connectionStrings>
11 </configuration>
12 <system.web>
13 <compilation debug="true" targetFramework="4.6.1"/>
14 <httpRuntime targetFramework="4.6.1"/>

```

2.2. Linq to SQL

2.2.1. Add Connection

Server Explorer --> Data Connections --> Right click --> Add Connection...
 --> Microsoft SQL server -->
 Enter your server and database details



Choose Data Source ? X

Data source:

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

Data provider:
.NET Framework Data Provider for SQ v

☒ Always use this selection

Continue Cancel

Add Connection ? X

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 v

User name: Tester

Password: ●●●●

☒ Save my password

Connect to a database

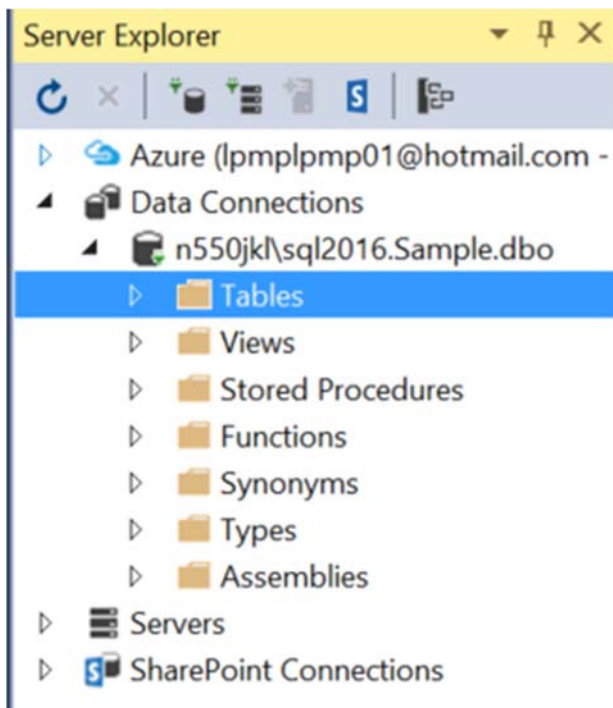
☒ Select or enter a database name:
Sample v

☐ Attach a database file:
Browse...

Advanced...

Test Connection OK Cancel

Microsoft Visual Studio
Test connection succeeded.
OK



2.2.2. Sample.dbml

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

--> Linq to SQL classes -->

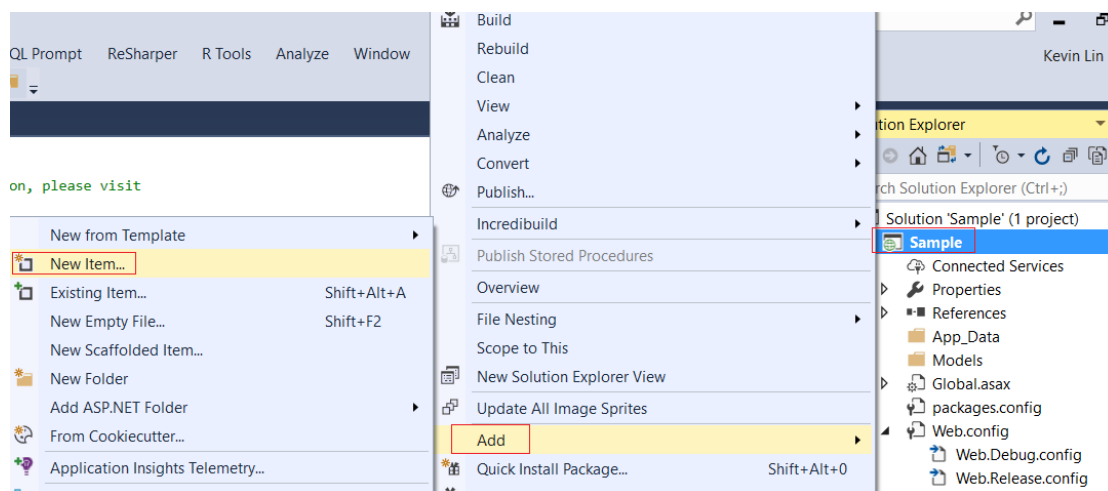
Name : Sample.dbml

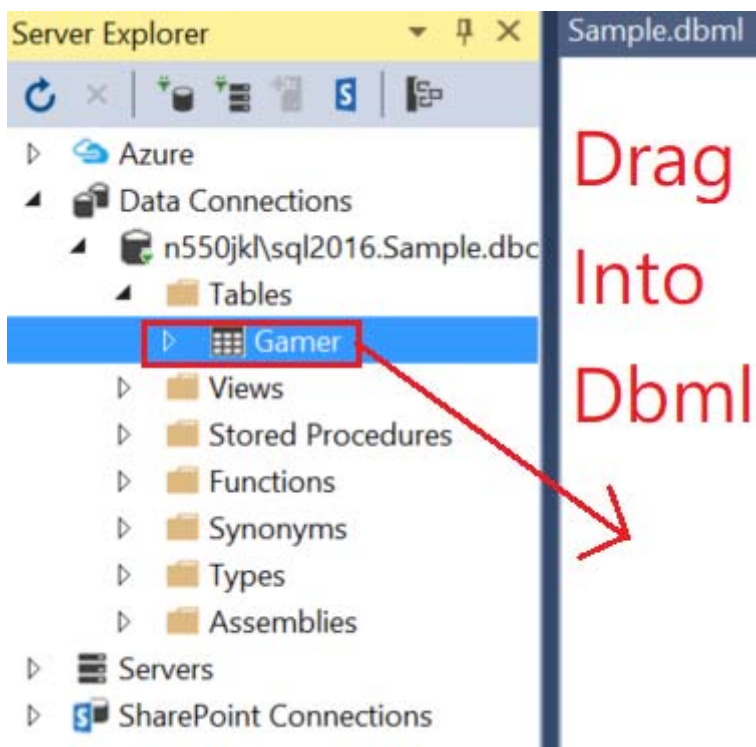
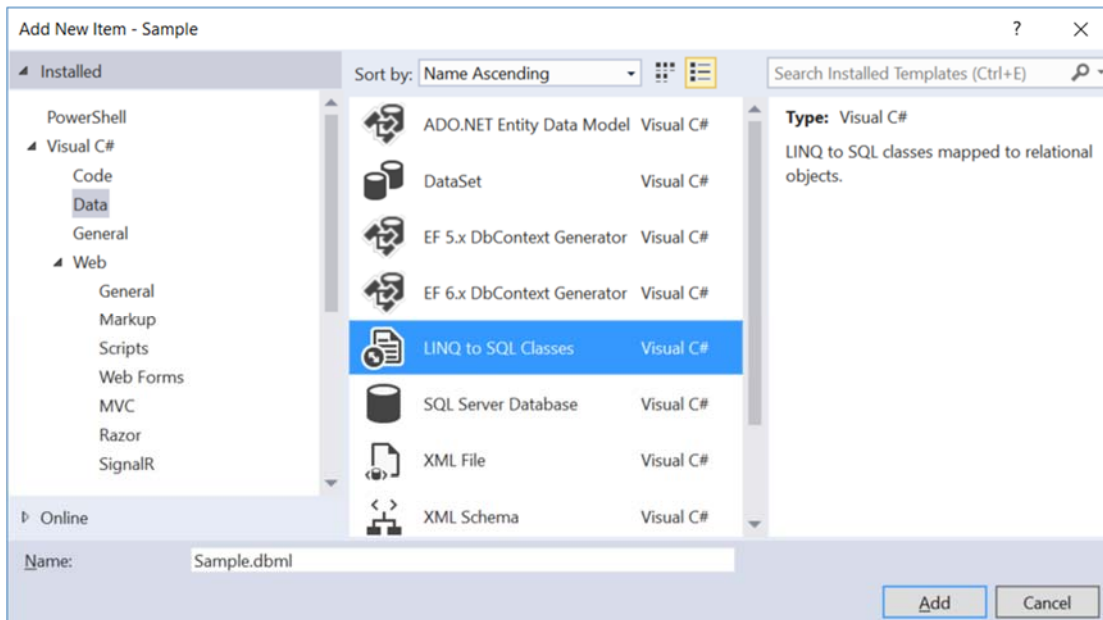
I name it as "Sample.dbml",

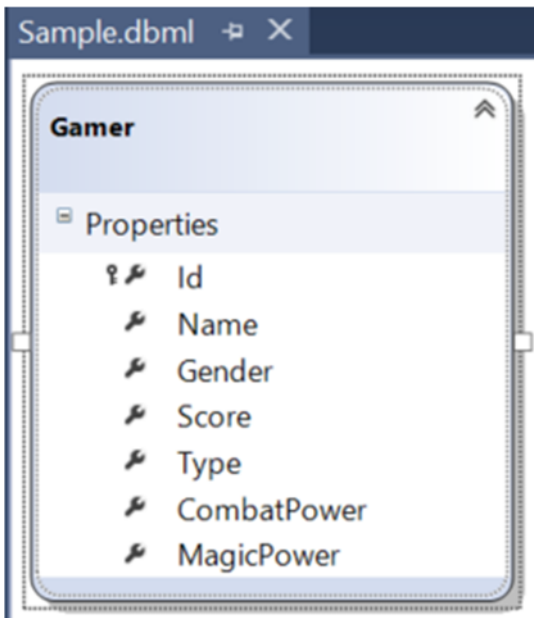
because I know this is for connection to "Sample" Database.

-->

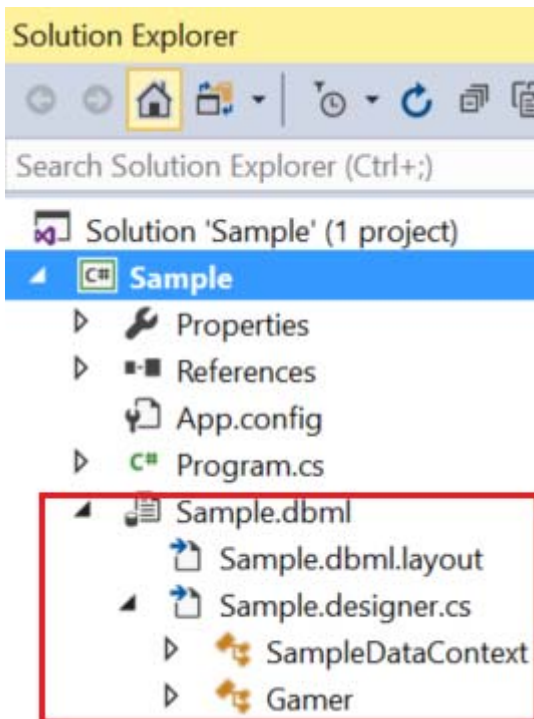
Drag Table from Server Explorer into DBML







Save the dbml, it will generate the following files.
The DataContext context is the entry point to database.



2.2.3. Sample.dbml - Single Table Inheritance

In Dbml

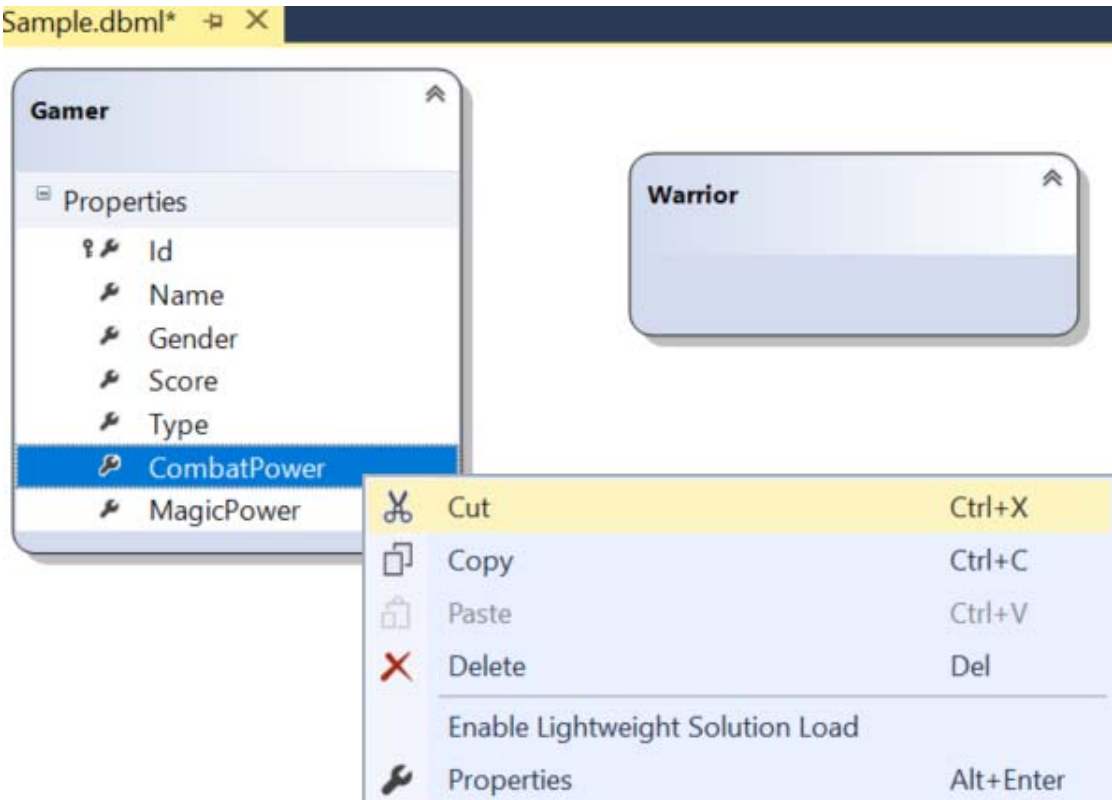
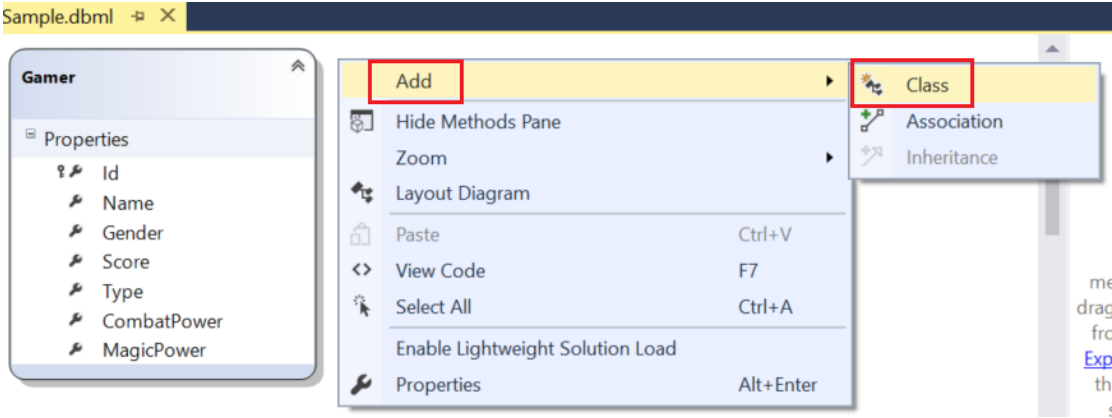
Right click --> Add --> Class -->

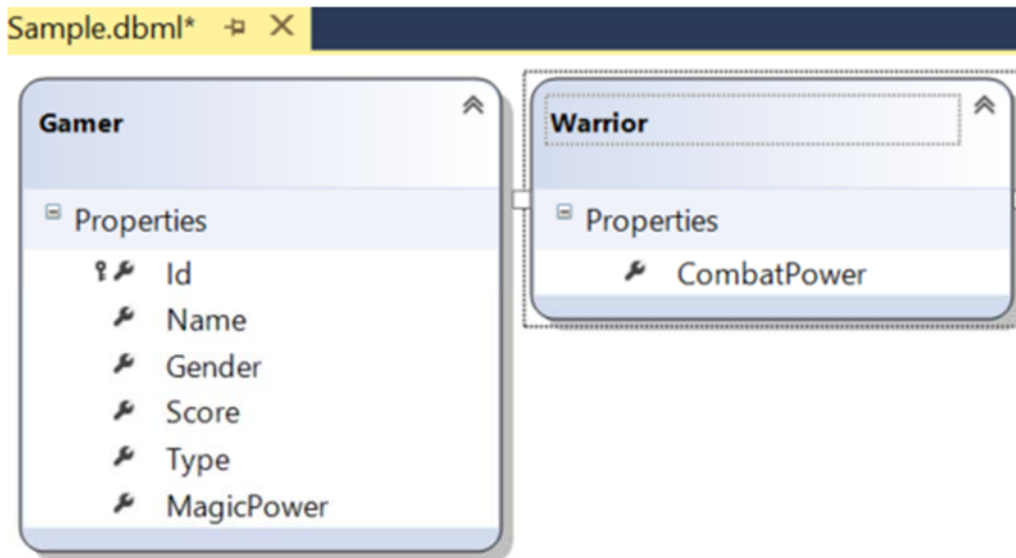
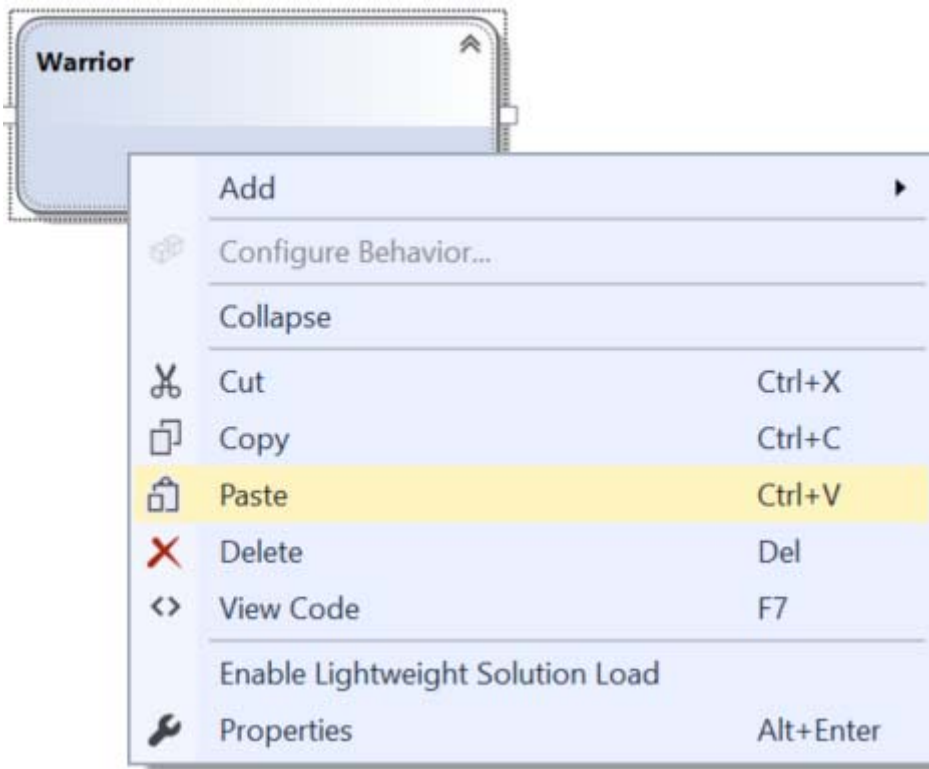
Change class name to

"Warrior"

-->

Cut "CombatPower" property from Gamer class
and paste it in "Warrior" class.





In Dbml

Right click --> Add --> Class -->

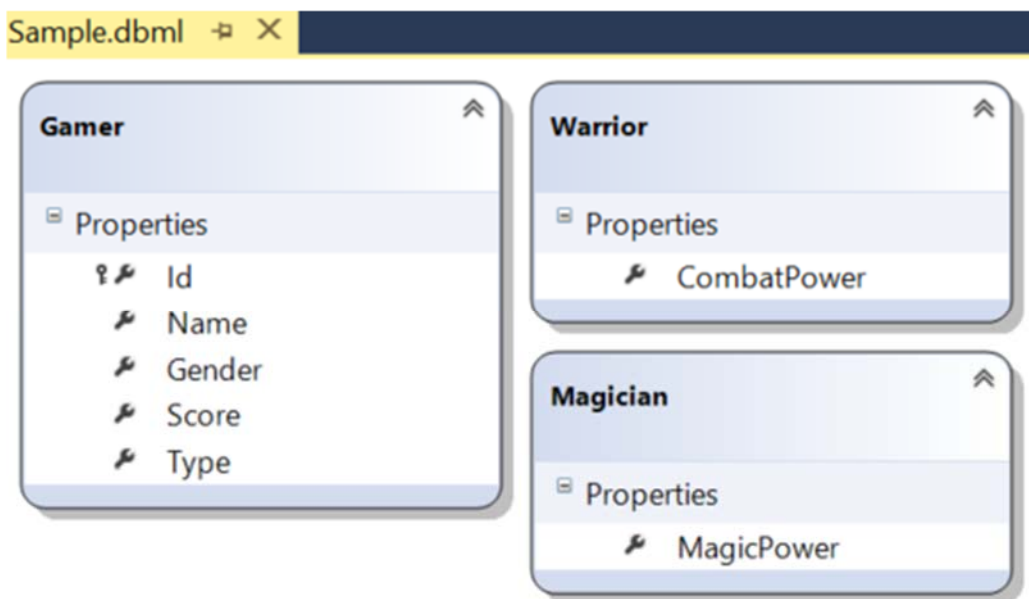
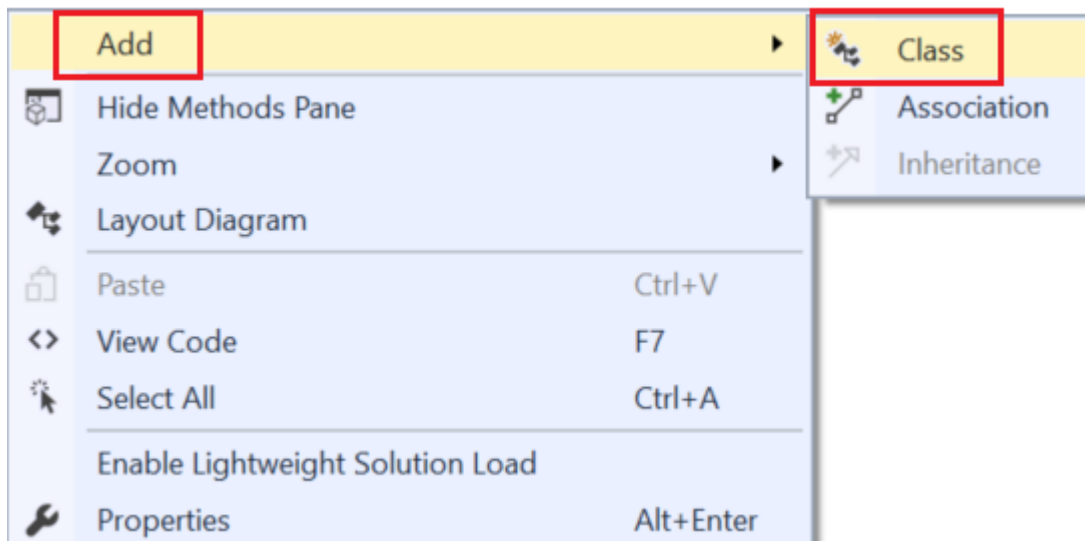
Change class name to

"Magician"

-->

Cut "MagicPower" property from Gamer class

and paste it in "Magician" class.



Gamer in Dbml

Right click --> Add --> Inheritance

-->

Select a base class:

Gamer

Select a derived class:

Warrior

Gamer in Dbml

Right click --> Add --> Inheritance

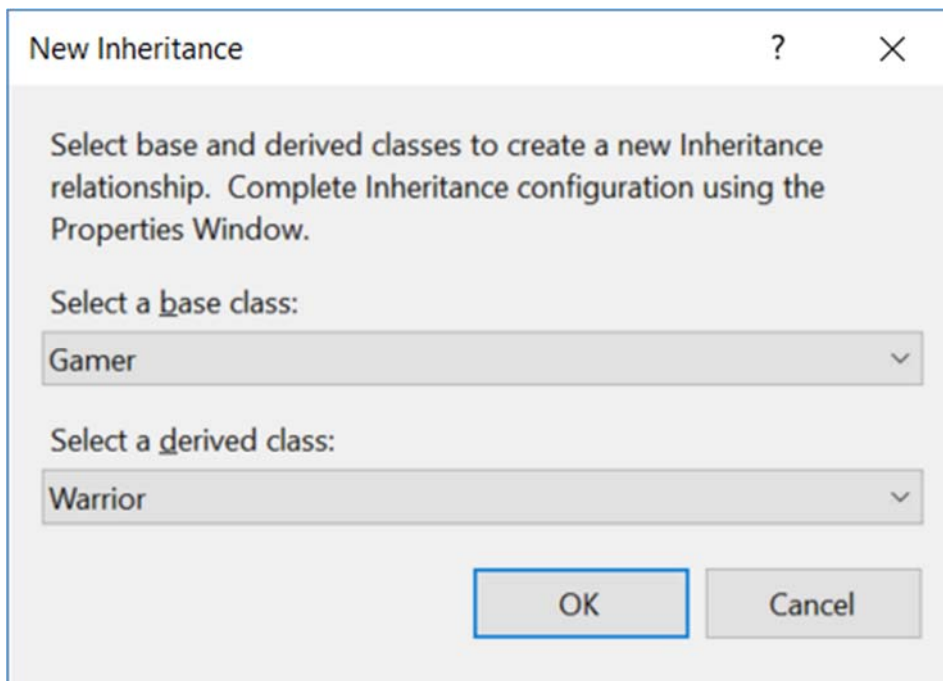
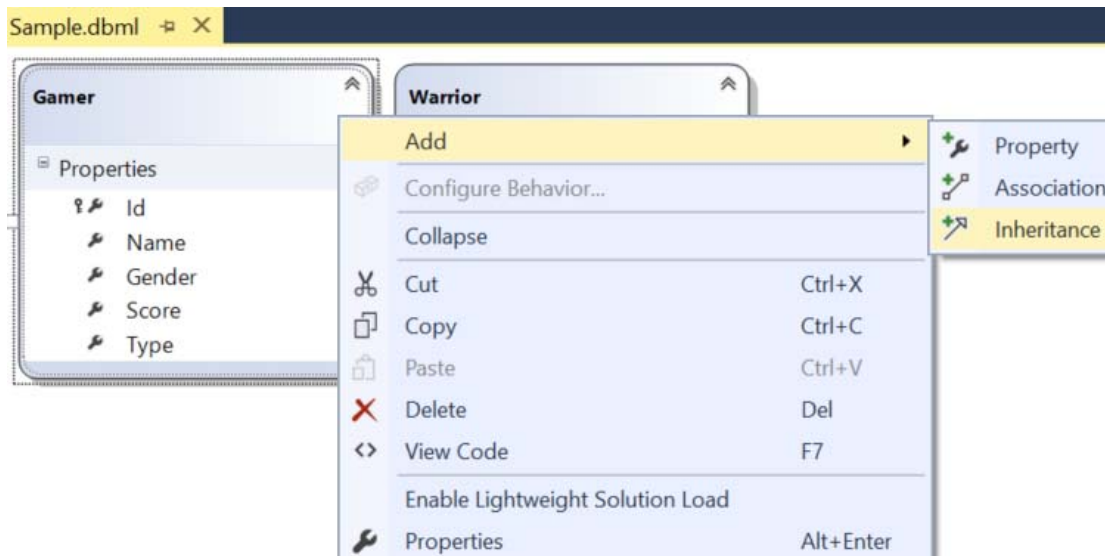
-->

Select a base class:

Gamer

Select a derived class:

Magician



New Inheritance ? X

Select base and derived classes to create a new Inheritance relationship. Complete Inheritance configuration using the Properties Window.

Select a base class:

Gamer v

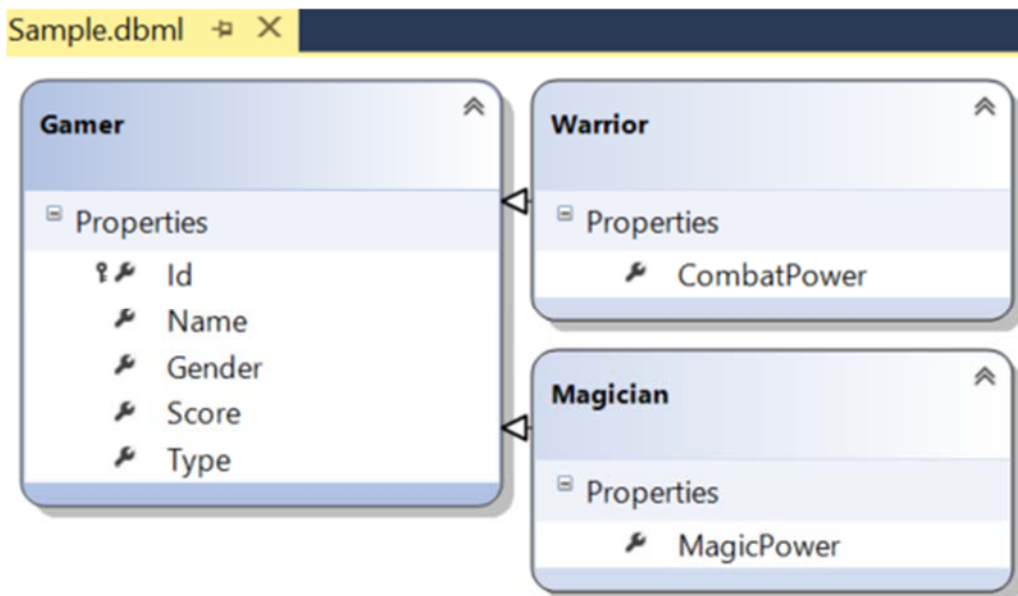
Select a derived class:

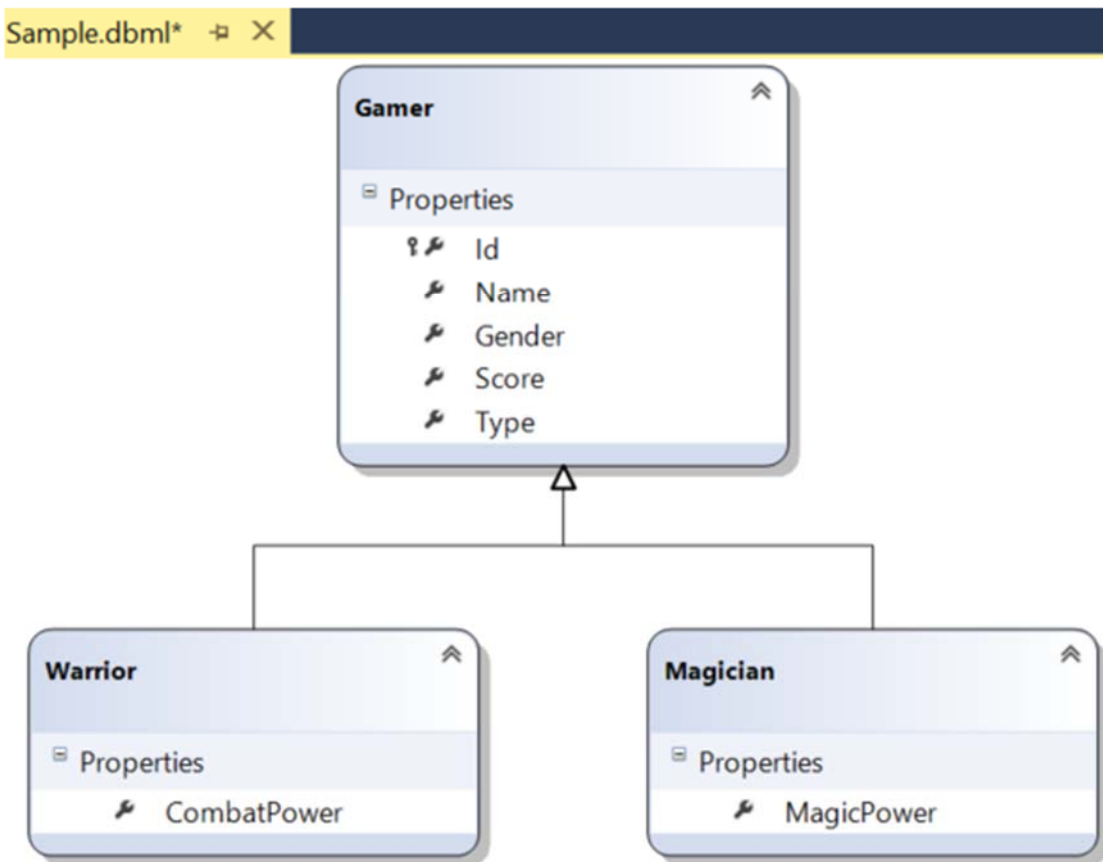
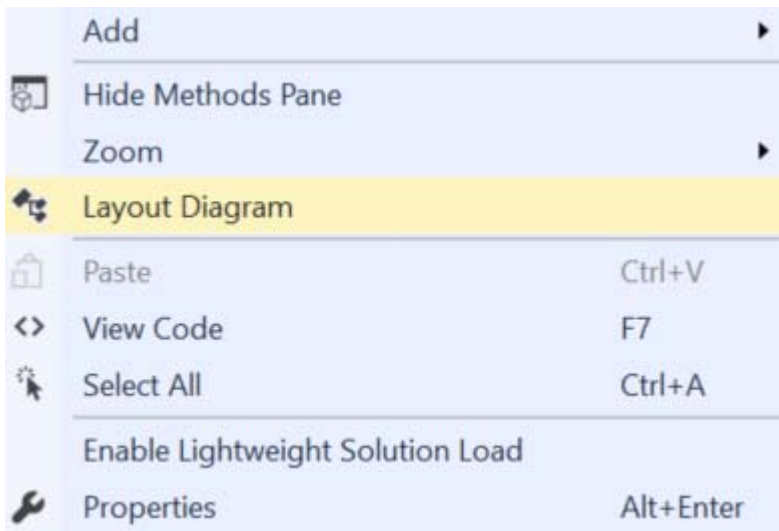
Magician v

OK Cancel

In Dbml

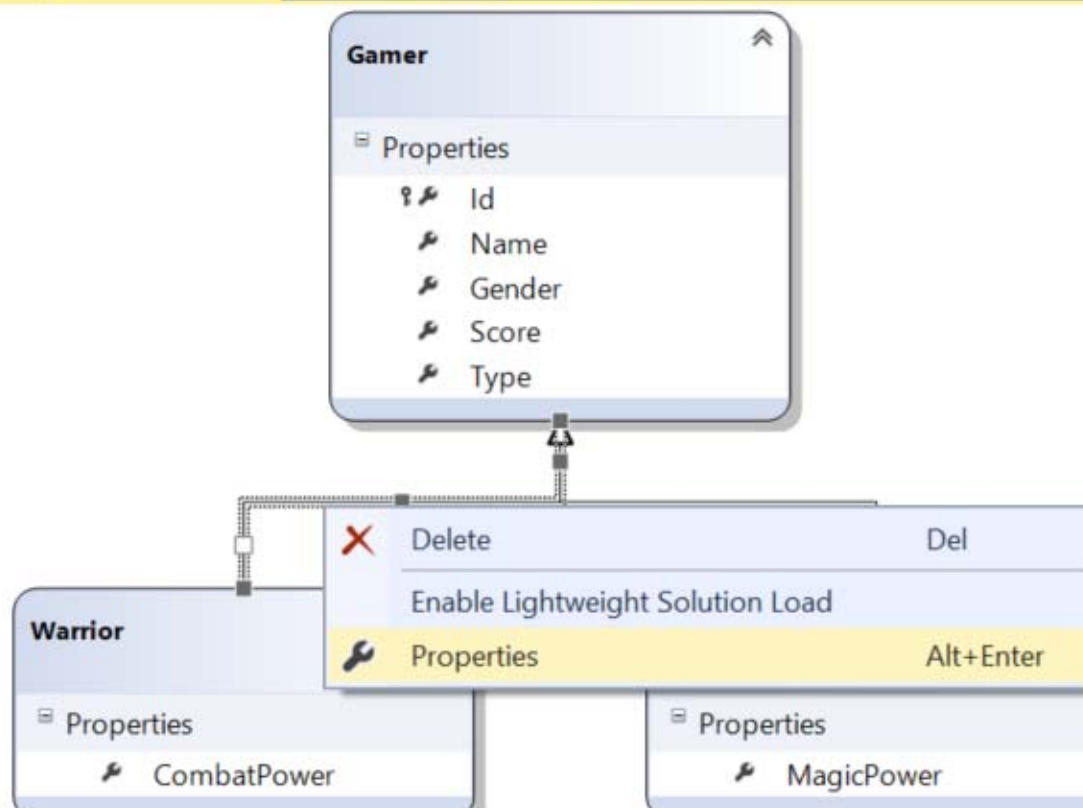
Right Click --> Layout Diagram
it will sort the diagram layout





2.2.4. Sample.dbml - Single Table Inheritance - Set Discriminator 識別者

Right click on the inheritance relationship arrow
(that connects Gamer and Warrior classes)
and select "Properties".
Set the properties of the inheritance relationship as shown below.



Properties

Warrior Inherits From Gamer Inheritance

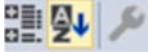
Base Class Discriminator Value	
Derived Class Discriminator Value	Warrior
Discriminator Property	Type
Inheritance Default	Warrior

This means use "Type" property in Gamer class as Discriminator Property.
 If [Type]=="Warrior", then the data row will generate an object of "Warrior" class.
 In addition, "Inheritance Default" means
 we set it will generate an object of "Warrior" class by default.

Right click on the inheritance relationship arrow
 (that connects Gamer and Magician classes)
 and select "Properties".
 Set the properties of the inheritance relationship as shown below.

Properties

Magician Inherits From Gamer Inheritance



Base Class Discriminator Value	
Derived Class Discriminator Value	Magician
Discriminator Property	Type
Inheritance Default	Warrior

This means use "Type" property in Gamer class as Discriminator Property.

If [Type]=="Magician", then the data row will generate an object of "Magician" class.

In addition, "Inheritance Default" means

we set it will generate an object of "Warrior" class by default.

If you don't know anything about **attributes**,
please read my C# tutorial before you continue.

Look at the auto generated code in **Sample.designer.cs**

```
global::System.Data.Linq.Mapping.TableAttribute(Name="dbo.Gamer")]]
[global::System.Data.Linq.Mapping.InheritanceMappingAttribute(Code="Warrior", Type=typeof(Warrior),
IsDefault=true)]
[global::System.Data.Linq.Mapping.InheritanceMappingAttribute(Code="Magician", Type=typeof(Magician))]
public partial class Gamer : INotifyPropertyChanging, INotifyPropertyChanged
....
```

```
public partial class Warrior : Gamer
```

```
...
```

```
public partial class Magician : Gamer
```

```
...
```

```
global::System.Data.Linq.Mapping.TableAttribute(Name="dbo.Gamer")]]
```

it means the properties of Gamer class and its sub-Class, Warrior class and Magician class,
are mapped to the columns of "dbo.Gamer" Table in "Sample" Database

```
[global::System.Data.Linq.Mapping.InheritanceMappingAttribute(Code="Magician", Type=typeof(Magician))]
```

In our case, we use "Type" property in Gamer class as Discriminator Property.

Code="Magician" means when Discriminator Property [Type]=="Magician",

then **Type=typeof(Magician))** means the data row will generate an object of "Magician" class.

```
[global::System.Data.Linq.Mapping.InheritanceMappingAttribute(Code="Warrior", Type=typeof(Warrior),
IsDefault=true)]
```

In our case, we use "Type" property in Gamer class as Discriminator Property.

Code="Warrior" means when Discriminator Property [Type]=="Warrior",

then **Type=typeof(Warrior))** means the data row will generate an object of "Warrior" class.

IsDefault=true means we set it will generate an object of "Warrior" class by default.

2.3. WebForm1.aspx

2.3.1. WebForm1.aspx

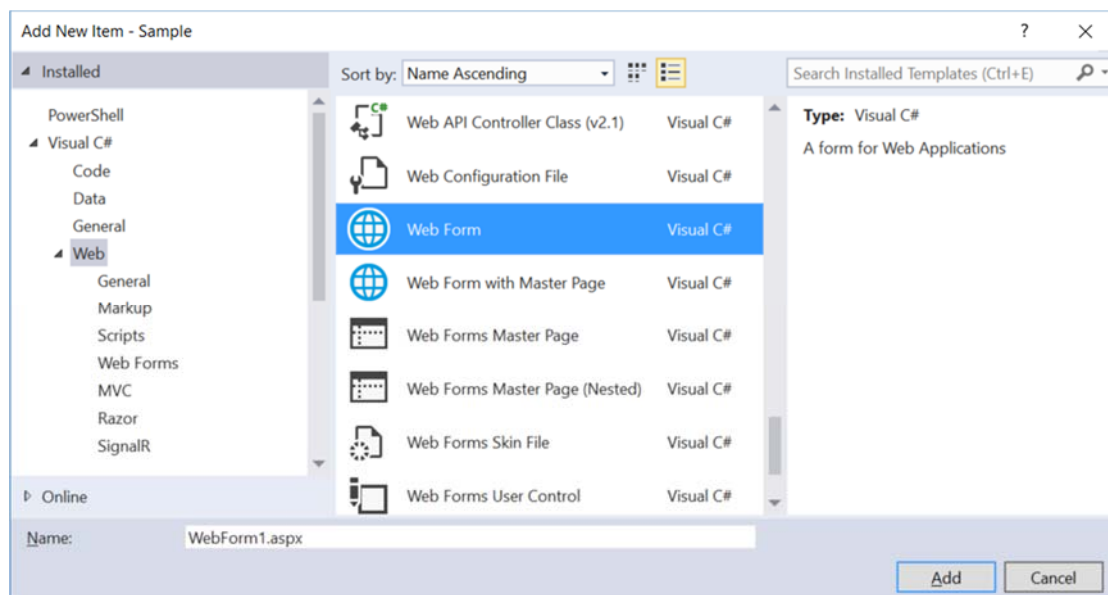
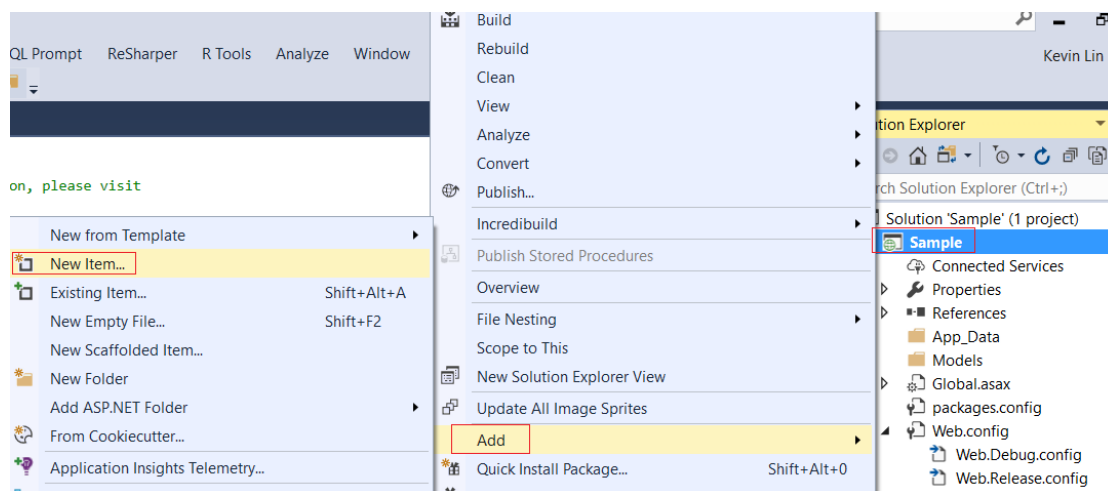
ProjectName --> Right Click --> Add --> New Item...

-->

WebForm

Name :

WebForm1.aspx



```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="Sample.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="GridView1" runat="server"></asp:GridView>
        <asp:Button ID="btnGetAllGamers" runat="server" Text="Get All Gamers"
        OnClick="btnGetAllGamers_Click" />
        <asp:Button ID="btnGetAllWarriors" runat="server" Text="Get All Warriors"
        OnClick="btnGetAllWarriors_Click" />
        <asp:Button ID="btnGetAllMagicians" runat="server" Text="Get All Magicians"
        OnClick="btnGetAllMagicians_Click" />
        <asp:Button ID="btnAddGamers" runat="server" Text="Add Gamers"
        OnClick="btnAddGamers_Click" />
    </div>
</form>
</body>
</html>

```

2.3.2. WebForm1.aspx.cs

```

using System;
using System.Linq;
namespace Sample
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
        }
        protected void btnGetAllGamers_Click(object sender, EventArgs e)
        {
            using (SampleDataContext dbContext = new SampleDataContext())
            {
                // Write the generated sql query to the webform
                dbContext.Log = Response.Output;

                Table<Gamer> gamers = dbContext.Gamers;
                GridView1.DataSource =
                    dbContext.Gamers.ToList();
                GridView1.DataBind();
            }
        }
        protected void btnGetAllWarriors_Click(object sender, EventArgs e)
        {
            using (SampleDataContext dbContext = new SampleDataContext())
            {
                // Write the generated sql query to the webform
                dbContext.Log = Response.Output;
                IQueryable<Warrior> warriors = dbContext.Gamers.OfType<Warrior>();
                GridView1.DataSource =
                    dbContext.Gamers.OfType<Warrior>().ToList();
                GridView1.DataBind();
            }
        }
        protected void btnGetAllMagicians_Click(object sender, EventArgs e)
        {
            using (SampleDataContext dbContext = new SampleDataContext())
            {
                // Write the generated sql query to the webform

```

```

        dbContext.Log = Response.Output;
        IQueryable<Magician> magicians = dbContext.Gamers.OfType<Magician>();
        GridView1.DataSource =
            dbContext.Gamers.OfType<Magician>().ToList();
        GridView1.DataBind();
    }
}
protected void btnAddGamers_Click(object sender, EventArgs e)
{
}
}

```

SELECT [t0].[Type], [t0].[CombatPower], [t0].[Id], [t0].[Name], [t0].[Gender], [t0].[Score] FROM [dbo].[Gamer] AS [t0] WHERE ([t0].[Type] <> @p0) OR ([t0].[Type] IS NULL) -- @p0: Input NVarChar (Size = 4000; Prec = 0; Scale = 0) [Magician] -- Context: SqlProvider (Sql2008) Model: AttributedMetaModel Build: 4.7.2556.0

CombatPower	Id	Name	Gender	Score	Type
500	1	Name1 ABC	Male	5000	Warrior
350	2	Name2 ABCDE	Female	4500	Warrior
450	6	Name6 PQRSTU VW	Male	4000	Warrior
550	7	Name7 XYZ	Male	4500	Warrior

```

IQueryable<Warrior> warriors = dbContext.Gamers.OfType<Warrior>();
GridView1.DataSource = dbContext.Gamers.OfType<Warrior>().ToList();
GridView1.DataBind();

```

"Warrior" class one property, CombatPower, so the first column is "CombatPower".

In addition, "Warrior" class inherit all properties from its base class, "Gamer". Thus, the following columns is from the properties of "Gamer" class.

SELECT [t0].[Type], [t0].[MagicPower], [t0].[Id], [t0].[Name], [t0].[Gender], [t0].[Score] FROM [dbo].[Gamer] AS [t0] WHERE ([t0].[Type] = @p0) AND ([t0].[Type] IS NOT NULL) -- @p0: Input NVarChar (Size = 4000; Prec = 0; Scale = 0) [Magician] -- Context: SqlProvider(Sql2008) Model: AttributedMetaModel Build: 4.7.2556.0

MagicPower	Id	Name	Gender	Score	Type
600	3	Name3 EFGH	Male	6500	Magician
650	4	Name4 HIJKLMN	Female	45000	Magician
700	5	Name5 NOP	Male	3000	Magician

```

IQueryable<Magician> magicians = dbContext.Gamers.OfType<Magician>();
GridView1.DataSource = dbContext.Gamers.OfType<Magician>().ToList();
GridView1.DataBind();

```

"Magician" class one property, MagicPower,

so the first column is "MagicPower".

In addition, "Magician" class inherit all properties from its base class, "Gamer".

Thus, the following columns is from the properties of "Gamer" class.

```
SELECT [t0].[Type], [t0].[CombatPower], [t0].[Id], [t0].[Name], [t0].[Gender], [t0].[Score], [t0].[MagicPower] FROM [dbo].[Gamer] AS [t0]
-- Context: SqlProvider(Sql2008) Model: AttributedMetaModel Build: 4.7.2556.0
```

Id	Name	Gender	Score	Type
1	Name1 ABC	Male	5000	Warrior
2	Name2 ABCDE	Female	4500	Warrior
3	Name3 EFGH	Male	6500	Magician
4	Name4 HIJKLMN	Female	45000	Magician
5	Name5 NOP	Male	3000	Magician
6	Name6 PQRSTUVW	Male	4000	Warrior
7	Name7 XYZ	Male	4500	Warrior

Get All Gamers

Get All Warriors

Get All Magicians

Add Gamers

Gamer class does not have "MagicPower" and "CombatPower" properties, so it will not show these two properties of Magician and Warrior.

This is wrong, the GridView should display "MagicPower" and "CombatPower" columns.

So we need to fix it.

2.4. WebForm1.aspx (Fix "btnGetAllGamers" issue)

2.4.1. dbml (Fix "btnGetAllGamers" issue)

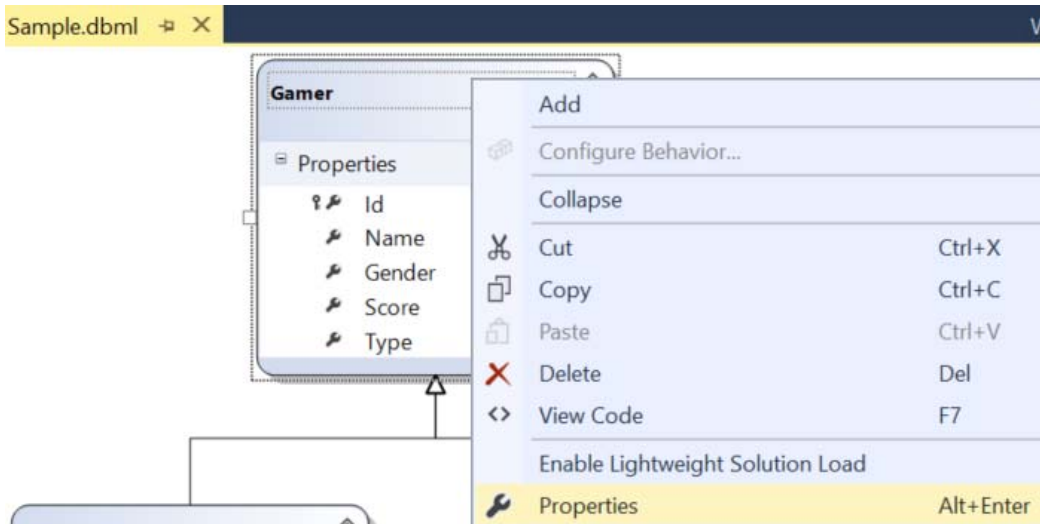
We have to set Gamer as abstract class,
so no one can create an instance of Gamer.

In the dbml

Gamer --> Right click --> Properties

-->

Inheritance Modifier : abstract



Properties	
Gamer Data Class	
Access	Public
Delete	Use Runtime
Inheritance Modifier	abstract
Insert	Use Runtime
Name	Gamer
Source	dbo.Gamer
Update	Use Runtime

2.4.2. WebForm1.aspx (Fix "btnGetAllGamers" issue)

Nothing change here.

2.4.3. WebForm1.aspx.cs (Fix "btnGetAllGamers" issue)

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Linq;
using System.Linq;
namespace Sample
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
        }
        private DataTable ConvertGamersForDisplay(List<Gamer> gamers)
        {
            DataTable dt = new DataTable();
        }
    }
}
```

```

dt.Columns.Add("Id");
dt.Columns.Add("Name");
dt.Columns.Add("Gender");
dt.Columns.Add("Score");
dt.Columns.Add("Type");
dt.Columns.Add("CombatPower");
dt.Columns.Add("MagicPower");
foreach (Gamer gamer in gamers)
{
    DataRow dr = dt.NewRow();
    dr["Id"] = gamer.Id;
    dr["Name"] = gamer.Name;
    dr["Gender"] = gamer.Gender;
    dr["Score"] = gamer.Score;
    //Because Gamer is an abstract class,
    //No one can create Gamer instance.
    //So gamer is actually an object of "Warrior" or "Magician"
    //If gamer is Warrior,
    //then add [CombatPower] column to DataRow
    //and set dr["Type"] = "Warrior".
    //If gamer is Magician,
    //then add [MagicPower] column to DataRow
    //and set dr["Type"] = "Magician".
    Warrior warrior = gamer as Warrior;
    if (warrior != null)
    {
        dr["CombatPower"] = warrior.CombatPower;
        dr["Type"] = "Warrior";
    }
    Magician magician = gamer as Magician;
    if (magician != null)
    {
        dr["MagicPower"] = magician.MagicPower;
        dr["Type"] = "Magician";
    }
    dt.Rows.Add(dr);
}
return dt;
}
protected void btnGetAllGamers_Click(object sender, EventArgs e)
{
    using (SampleDataContext dbContext = new SampleDataContext())
    {
        // Write the generated sql query to the webform
        dbContext.Log = Response.Output;
        Table<Gamer> gamers = dbContext.Gamers;
        //GridView1.DataSource =
        //    dbContext.Gamers.ToList();
        GridView1.DataSource =
            ConvertGamersForDisplay(dbContext.Gamers.ToList());
        GridView1.DataBind();
    }
}
protected void btnGetAllWarriors_Click(object sender, EventArgs e)
{
    using (SampleDataContext dbContext = new SampleDataContext())
    {
        // Write the generated sql query to the webform
        dbContext.Log = Response.Output;

```



```

        IQueryable<Warrior> warriors = dbContext.Gamers.OfType<Warrior>();
        GridView1.DataSource =
            dbContext.Gamers.OfType<Warrior>().ToList();
        GridView1.DataBind();
    }
}
protected void btnGetAllMagicians_Click(object sender, EventArgs e)
{
    using (SampleDataContext dbContext = new SampleDataContext())
    {
        // Write the generated sql query to the webform
        dbContext.Log = Response.Output;
        IQueryable<Magician> magicians = dbContext.Gamers.OfType<Magician>();
        GridView1.DataSource =
            dbContext.Gamers.OfType<Magician>().ToList();
        GridView1.DataBind();
    }
}
protected void btnAddGamers_Click(object sender, EventArgs e)
{
}
}
}

```

SELECT [t0].[Type], [t0].[CombatPower], [t0].[Id], [t0].[Name], [t0].[Gender],
[t0].[Score], [t0].[MagicPower] FROM [dbo].[Gamer] AS [t0] -- Context:
SqlProvider(Sql2008) Model: AttributedMetaModel Build: 4.7.2556.0

Id	Name	Gender	Score	Type	CombatPower	MagicPower
1	Name1 ABC	Male	5000	Warrior	500	
2	Name2 ABCDE	Female	4500	Warrior	350	
3	Name3 EFGH	Male	6500	Magician		600
4	Name4 HIJKLMN	Female	45000	Magician		650
5	Name5 NOP	Male	3000	Magician		700
6	Name6 PQRSTU VW	Male	4000	Warrior	450	
7	Name7 XYZ	Male	4500	Warrior	550	

2.5. WebForm1.aspx (btnAddGamers)

2.5.1. WebForm1.aspx (btnAddGamers)

Nothing change here.

2.5.2. WebForm1.aspx.cs (btnAddGamers)

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Linq;
using System.Linq;
namespace Sample
{
    public partial class WebForm1 : System.Web.UI.Page

```

```

{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    private DataTable ConvertGamersForDisplay(List<Gamer> gamers)
    {
        DataTable dt = new DataTable();
        dt.Columns.Add("Id");
        dt.Columns.Add("Name");
        dt.Columns.Add("Gender");
        dt.Columns.Add("Score");
        dt.Columns.Add("Type");
        dt.Columns.Add("CombatPower");
        dt.Columns.Add("MagicPower");
        foreach (Gamer gamer in gamers)
        {
            DataRow dr = dt.NewRow();
            dr["Id"] = gamer.Id;
            dr["Name"] = gamer.Name;
            dr["Gender"] = gamer.Gender;
            dr["Score"] = gamer.Score;
            //Because Gamer is an abstract class,
            //No one can create Gamer instance.
            //So gamer is actually an object of "Warrior" or "Magician"
            //If gamer is Warrior,
            //then add [CombatPower] column to DataRow
            //and set dr["Type"] = "Warrior".
            //If gamer is Magician,
            //then add [MagicPower] column to DataRow
            //and set dr["Type"] = "Magician".
            Warrior warrior = gamer as Warrior;
            if (warrior != null)
            {
                dr["CombatPower"] = warrior.CombatPower;
                dr["Type"] = "Warrior";
            }
            Magician magician = gamer as Magician;
            if (magician != null)
            {
                dr["MagicPower"] = magician.MagicPower;
                dr["Type"] = "Magician";
            }
            dt.Rows.Add(dr);
        }
        return dt;
    }
    private void GetAllGamers()
    {
        using (SampleDataContext dbContext = new SampleDataContext())
        {
            // Write the generated sql query to the webform
            dbContext.Log = Response.Output;
            Table<Gamer> gamers = dbContext.Gamers;
            //GridView1.DataSource =
            //    dbContext.Gamers.ToList();
            GridView1.DataSource =
                ConvertGamersForDisplay(dbContext.Gamers.ToList());
            GridView1.DataBind();
        }
    }
}

```

```

protected void btnGetAllGamers_Click(object sender, EventArgs e)
{
    GetAllGamers();
}
protected void btnGetAllWarriors_Click(object sender, EventArgs e)
{
    using (SampleDataContext dbContext = new SampleDataContext())
    {
        // Write the generated sql query to the webform
        dbContext.Log = Response.Output;
        IQueryable<Warrior> warriors = dbContext.Gamers.OfType<Warrior>();
        GridView1.DataSource =
            dbContext.Gamers.OfType<Warrior>().ToList();
        GridView1.DataBind();
    }
}
protected void btnGetAllMagicians_Click(object sender, EventArgs e)
{
    using (SampleDataContext dbContext = new SampleDataContext())
    {
        // Write the generated sql query to the webform
        dbContext.Log = Response.Output;
        IQueryable<Magician> magicians = dbContext.Gamers.OfType<Magician>();
        GridView1.DataSource =
            dbContext.Gamers.OfType<Magician>().ToList();
        GridView1.DataBind();
    }
}
protected void btnAddGamers_Click(object sender, EventArgs e)
{
    using (SampleDataContext dbContext = new SampleDataContext())
    {
        Warrior warrior = new Warrior
        {
            Name = "warriorName",
            Gender = "Female",
            Score = 3000,
            CombatPower = 100
        };
        Magician magician = new Magician
        {
            Name = "magicianName",
            Gender = "Female",
            Score = 3000,
            MagicPower = 101
        };
        dbContext.Gamers.InsertOnSubmit(warrior);
        dbContext.Gamers.InsertOnSubmit(magician);
        dbContext.SubmitChanges();
        GetAllGamers();
    }
}
}

```

SELECT [t0].[Type], [t0].[CombatPower], [t0].[Id], [t0].[Name], [t0].[Gender],
[t0].[Score], [t0].[MagicPower] FROM [dbo].[Gamer] AS [t0] -- Context:
SqlProvider(Sql2008) Model: AttributedMetaModel Build: 4.7.2556.0

Id	Name	Gender	Score	Type	CombatPower	MagicPower
1	Name1 ABC	Male	5000	Warrior	500	
2	Name2 ABCDE	Female	4500	Warrior	350	
3	Name3 EFGH	Male	6500	Magician		600
4	Name4 HIJKLMN	Female	45000	Magician		650
5	Name5 NOP	Male	3000	Magician		700
6	Name6 PQRSTU VW	Male	4000	Warrior	450	
7	Name7 XYZ	Male	4500	Warrior	550	
8	warriorName	Female	3000	Warrior	100	
9	magicianName	Female	3000	Magician		101

Get All Gamers

Get All Warriors

Get All Magicians

Add Gamers