(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)  
=======================================================================

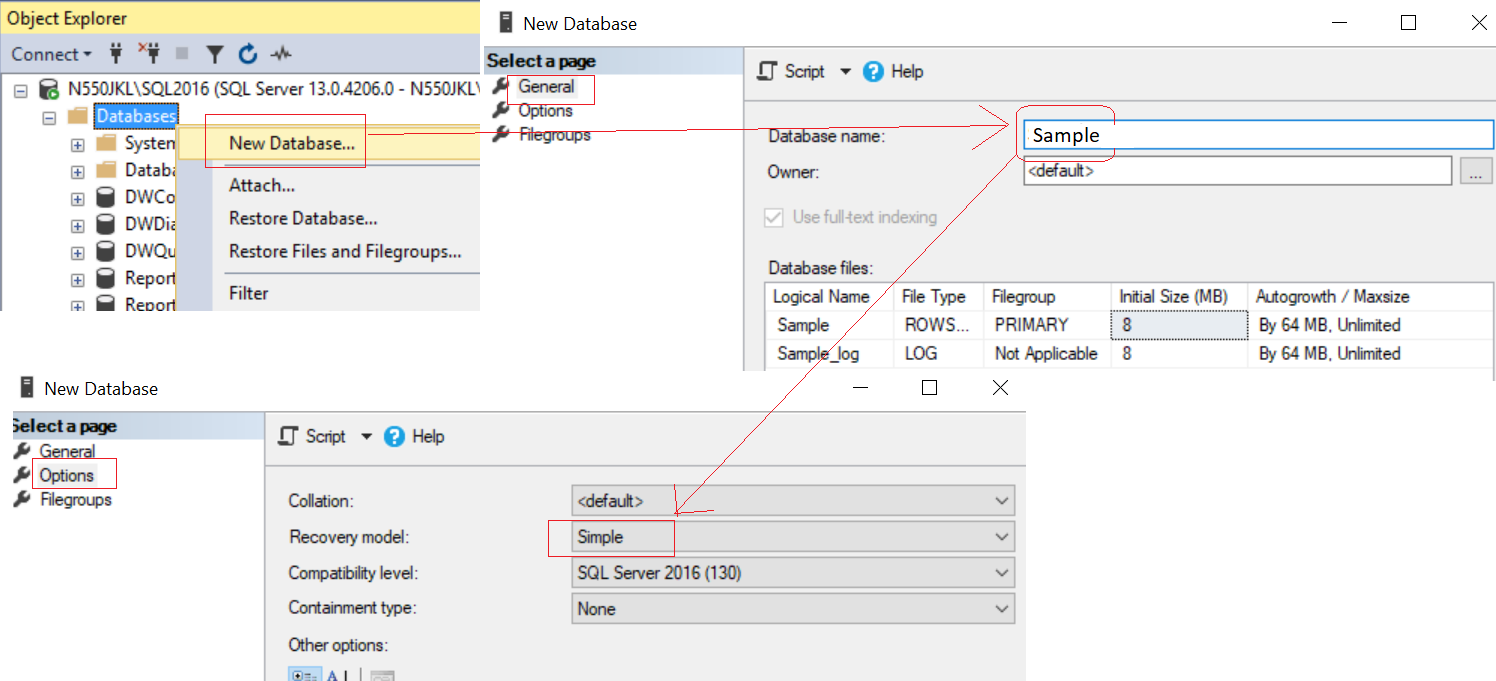
1. Web Form Application - Linq Query

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 PQRSTUVW', '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.

Graphical user interface, application

Description automatically generated with medium confidence



Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

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](http://asp.net/)**Web Application (.Net Framework)**

-->

Name:

**Sample**

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

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

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>

Graphical user interface, text, application

Description automatically generated

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 ....

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

2.2.2. Sample.dbml

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

--> Linq to SQL classes -->

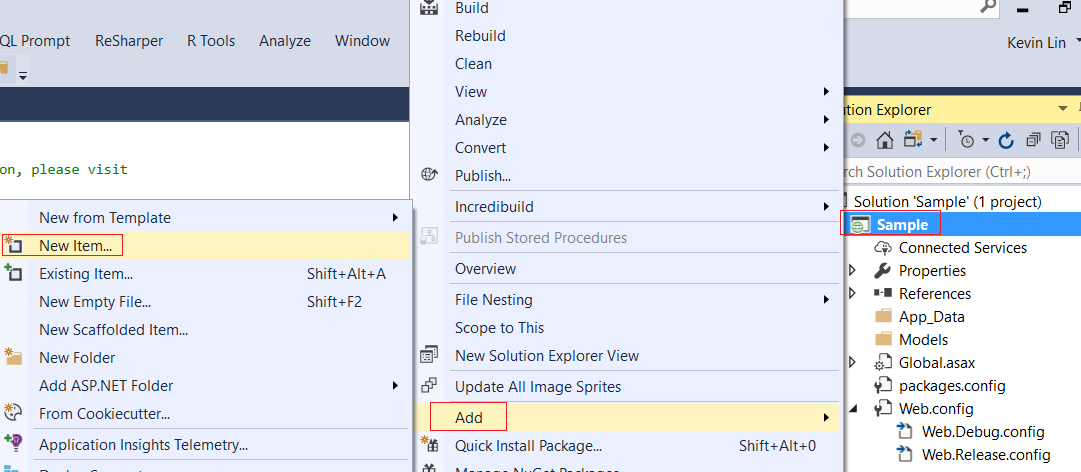
Name : Sample.dbml

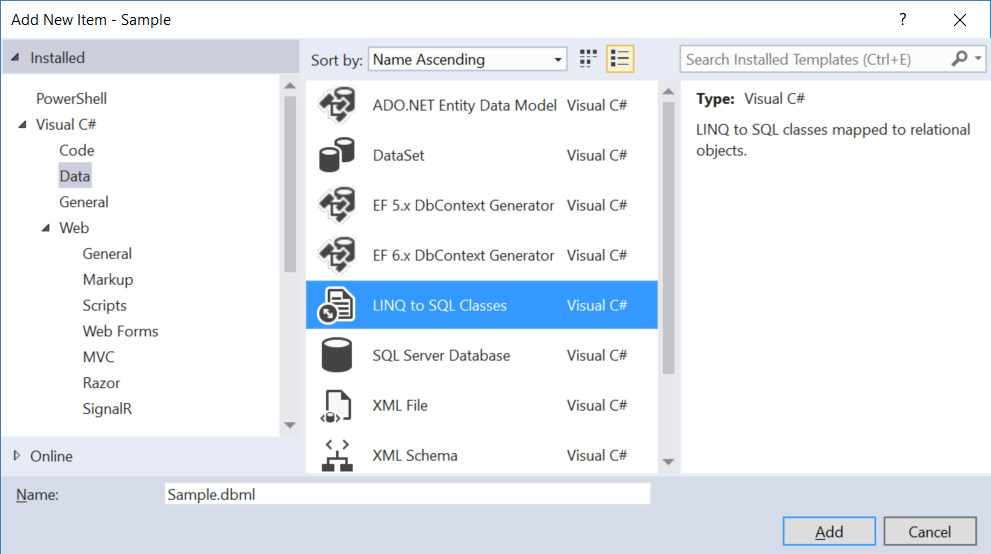
I name it as "Sample.dbml",

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

-->

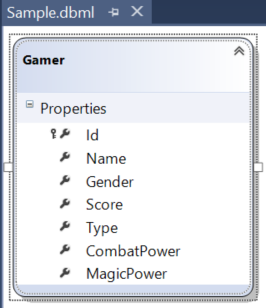
Drag Table from Server Explorer into DBML





Graphical user interface, application

Description automatically generated



Save the dbml, it will generate the following files.

The DataContext context is the entry point to database.

Graphical user interface, text, application

Description automatically generated

2.2.3. Sample.dbml - Single Table Inheritance

In Dbml

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

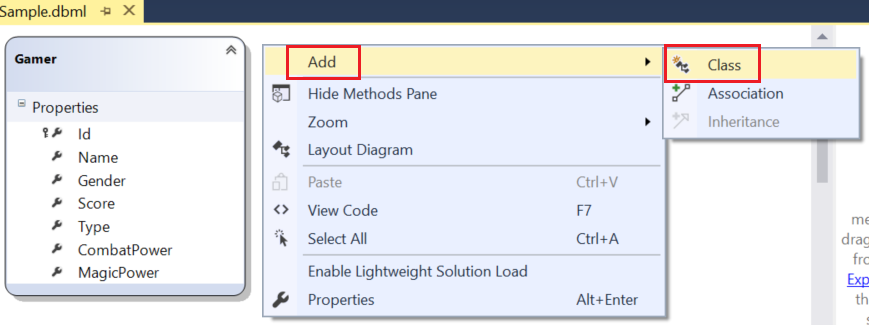
Change class name to

"Warrior"

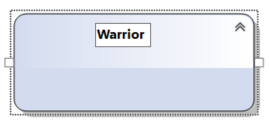
-->

Cut "CombatPower" property from Gamer class

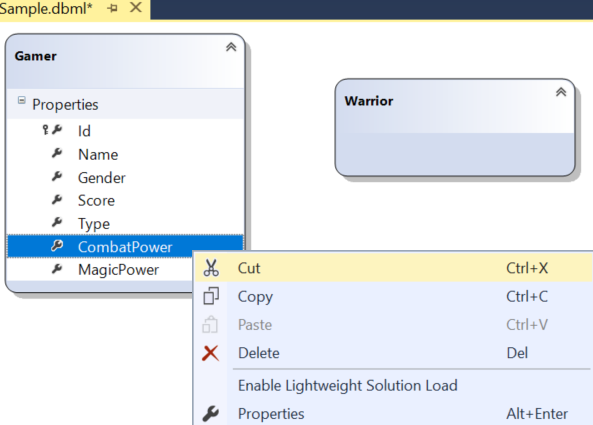
and paste it in "Warrior" class.



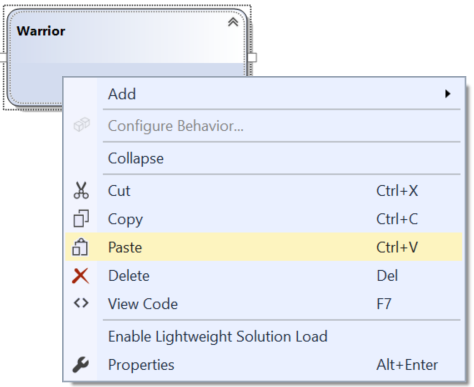
---------------------



---------------------



---------------------



---------------------

Graphical user interface, text, application, chat or text message

Description automatically generated

---------------------

In Dbml

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

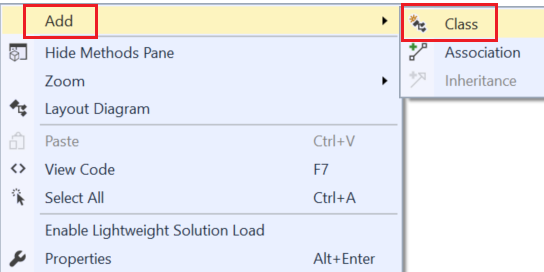
Change class name to

"Magician"

-->

Cut "MagicPower" property from Gamer class

and paste it in "Magician" class.



---------------------

Graphical user interface, application

Description automatically generated

---------------------

Graphical user interface, application, chat or text message

Description automatically generated

---------------------

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

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

---------------------

In Dbml

Right Click --> Layout Diagram

it will sort the diagram layout

Graphical user interface, application, chat or text message

Description automatically generated

---------------------

Graphical user interface, application

Description automatically generated

---------------------

Diagram

Description automatically generated

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)

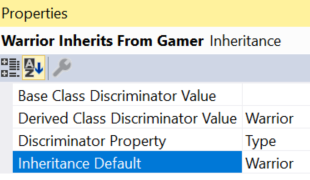
and select "Properties".

Set the properties of the inheritance relationship as shown below.

Diagram

Description automatically generated

---------------------------



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.

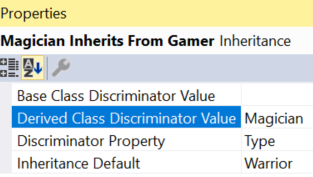
---------------------------

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.



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**

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

<%@ 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)

        {

        }

    }

}

Table

Description automatically generated

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.

Table

Description automatically generated

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.

Table

Description automatically generated

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

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

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](http://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)

        {

        }

    }

}

Table

Description automatically generated

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](http://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();

            }

        }

    }

}

Table

Description automatically generated