(T31)討論StoredProcedure搭配TableValueTypeParameter。實作在Asp.NetWebForm  
CourseGUID: e48417fc-9db5-4e99-822c-706c5ccef6cc  
=======================================================================  
(T31)討論StoredProcedure搭配TableValueTypeParameter。實作在Asp.NetWebForm  
=======================================================================  
0. Summary

1. Stored Procedure with TableValueType Parameter

2. Stored Procedure with TableValueType Parameter in AspNet

3. Set up SQL Authentication

-----------

4. Create Web Application

4.1. Web.config

4.2. WebForm1.aspx

4.3. WebForm1.aspx.cs

-----------

5. Clean up  
=======================================================================

0. Summary

1.

Create Table Value Type

1.1.

Reference:

<https://sqltimes.wordpress.com/2014/05/10/sql-server-how-to-check-if-a-user-define-table-type-exists-using-query-tsql/>

Check if table value type exists

1.2.

Syntax

--CREATE TYPE TableName AS TABLE

--(

--     Columns...

--);

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

1. Stored Procedure with TableValueType Parameter

--===============================================================

--T031\_01\_Stored Procedure with TableValueType Parameter

--===============================================================

--===============================================================

--T031\_01\_01

--Sample code to create a Table

--If Table exists then DROP it

IF ( EXISTS ( SELECT    \*

              FROM      INFORMATION\_SCHEMA.TABLES

              WHERE     TABLE\_NAME = 'PersonA' ) )

    BEGIN

        TRUNCATE TABLE dbo.PersonA;

        DROP TABLE PersonA;

    END;

GO -- Run the previous command and begins new batch

CREATE TABLE PersonA

    (

      Id INT PRIMARY KEY ,

      [Name] NVARCHAR(100) ,

      Gender NVARCHAR(10)

    );

GO -- Run the previous command and begins new batch

--===============================================================

--T031\_01\_02

--Table Value Parameter

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

--T031\_01\_02\_01

--Create Table Type and stored procedure

--If store procedure exist

IF ( EXISTS ( SELECT    \*

              FROM      INFORMATION\_SCHEMA.ROUTINES

              WHERE     ROUTINE\_TYPE = 'PROCEDURE'

                        AND LEFT(ROUTINE\_NAME, 3) NOT IN ( 'sp\_', 'xp\_', 'ms\_' )

                        AND SPECIFIC\_NAME = 'spInsertPersonA' ) )

    BEGIN

        DROP PROCEDURE spInsertPersonA;

    END;

GO -- Run the previous command and begins new batch

--Drop Table Type if exists

IF EXISTS ( SELECT  \*

            FROM    sys.types

            WHERE   is\_table\_type = 1

                    AND name = 'PersonType' )

    BEGIN

        DROP TYPE PersonType;

    END;

GO -- Run the previous command and begins new batch

CREATE TYPE PersonType AS TABLE

(

Id INT PRIMARY KEY,

[Name] NVARCHAR(100),

Gender NVARCHAR(10)

);

GO -- Run the previous command and begins new batch

--Pass the Table Type as parameter to Store procedure

CREATE PROCEDURE spInsertPersonA

    @PersonTableType PersonType READONLY

AS

    BEGIN

        INSERT  INTO PersonA

                SELECT  \*

                FROM    @PersonTableType;

    END;

GO -- Run the previous command and begins new batch

/\*

1.

Create Table Value Type

1.1.

Reference:

<https://sqltimes.wordpress.com/2014/05/10/sql-server-how-to-check-if-a-user-define-table-type-exists-using-query-tsql/>

Check if table value type exists

1.2.

Syntax

--CREATE TYPE TableName AS TABLE

--(

--     Columns...

--);

\*/

Graphical user interface

Description automatically generated with medium confidence

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

--T031\_01\_02\_02

--Declare a variable as Table Type

--Insert some data to Table Type

--Pass Table type as parameter to store procedure

DECLARE @PersonTableType2 PersonType;

INSERT  INTO @PersonTableType2

VALUES  ( 1, 'Name01', 'Male' );

INSERT  INTO @PersonTableType2

VALUES  ( 2, 'Name02', 'Female' );

INSERT  INTO @PersonTableType2

VALUES  ( 3, 'Name03', 'Female' );

INSERT  INTO @PersonTableType2

VALUES  ( 4, 'Name04', 'Male' );

INSERT  INTO @PersonTableType2

VALUES  ( 5, 'Name05', 'Male' );

EXECUTE spInsertPersonA @PersonTableType2;

GO -- Run the previous command and begins new batch

SELECT  \*

FROM    PersonA;

GO -- Run the previous command and begins new batch

Table

Description automatically generated

--===============================================================

--T031\_01\_03

--Clean up

--Drop Table if it exists

IF ( EXISTS ( SELECT    \*

              FROM      INFORMATION\_SCHEMA.TABLES

              WHERE     TABLE\_NAME = 'PersonA' ) )

    BEGIN

        TRUNCATE TABLE dbo.PersonA;

        DROP TABLE PersonA;

    END;

GO -- Run the previous command and begins new batch

--Drop Stored Procedure if it exists

IF ( EXISTS ( SELECT    \*

              FROM      INFORMATION\_SCHEMA.ROUTINES

              WHERE     ROUTINE\_TYPE = 'PROCEDURE'

                        AND LEFT(ROUTINE\_NAME, 3) NOT IN ( 'sp\_', 'xp\_', 'ms\_' )

                        AND SPECIFIC\_NAME = 'spInsertPersonA' ) )

    BEGIN

        DROP PROCEDURE spInsertPersonA;

    END;

GO -- Run the previous command and begins new batch

--Drop Table Type if exists

IF EXISTS ( SELECT  \*

            FROM    sys.types

            WHERE   is\_table\_type = 1

                    AND name = 'PersonType' )

    BEGIN

        DROP TYPE PersonType;

    END;

GO -- Run the previous command and begins new batch

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

2. Stored Procedure with TableValueType Parameter in AspNet

--===============================================================

--T031\_02\_Stored Procedure with TableValueType Parameter in AspNet

--===============================================================

--===============================================================

--T031\_02\_01

--Recreate a Table

IF ( EXISTS ( SELECT    \*

              FROM      INFORMATION\_SCHEMA.TABLES

              WHERE     TABLE\_NAME = 'PersonA' ) )

    BEGIN

        TRUNCATE TABLE dbo.PersonA;

        DROP TABLE PersonA;

    END;

GO -- Run the previous command and begins new batch

CREATE TABLE PersonA

    (

      Id INT IDENTITY(1, 1)

             PRIMARY KEY ,

      [Name] NVARCHAR(100) ,

      Gender NVARCHAR(10)

    );

GO -- Run the previous command and begins new batch

--===============================================================

--T031\_02\_02

--Stored Procedure with Table Type Parameter

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

--T031\_02\_02\_01

--ReCreate Table Type and Stored Procedure with Table Type Parameter

--Drop Stored Procedure if it exists

IF ( EXISTS ( SELECT    \*

              FROM      INFORMATION\_SCHEMA.ROUTINES

              WHERE     ROUTINE\_TYPE = 'PROCEDURE'

                        AND LEFT(ROUTINE\_NAME, 3) NOT IN ( 'sp\_', 'xp\_', 'ms\_' )

                        AND SPECIFIC\_NAME = 'spInsertPersonA' ) )

    BEGIN

        DROP PROCEDURE spInsertPersonA;

    END;

GO -- Run the previous command and begins new batch

--Drop Table Type Procedure if it exists

IF EXISTS ( SELECT  \*

            FROM    sys.types

            WHERE   is\_table\_type = 1

                    AND name = 'PersonType' )

    BEGIN

        DROP TYPE PersonType;

    END;

GO -- Run the previous command and begins new batch

--Create Table Type

CREATE TYPE PersonType AS TABLE

(

[Name] NVARCHAR(100),

Gender NVARCHAR(10)

);

GO -- Run the previous command and begins new batch

--Pass the Table Type as parameter to Store procedure

CREATE PROCEDURE spInsertPersonA

    @PersonTableType PersonType READONLY

AS

    BEGIN

        INSERT  INTO PersonA

                SELECT  \*

                FROM    @PersonTableType;

    END;

GO -- Run the previous command and begins new batch

/\*

1.

Create Table Value Type

1.1.

Reference:

<https://sqltimes.wordpress.com/2014/05/10/sql-server-how-to-check-if-a-user-define-table-type-exists-using-query-tsql/>

Check if table value type exists

1.2.

Syntax

--CREATE TYPE TableName AS TABLE

--(

--     Columns...

--);

\*/

Graphical user interface

Description automatically generated with medium confidence

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

--T031\_02\_02\_02

--Use Stored Procedure with Table Type Parameter

--Declare a variable as Table Type

--Insert some data to Table Type

--Pass Table type as parameter to store procedure

DECLARE @PersonTableType2 PersonType;

INSERT  INTO @PersonTableType2

VALUES  ( 'Name01', 'Male' );

INSERT  INTO @PersonTableType2

VALUES  ( 'Name02', 'Female' );

INSERT  INTO @PersonTableType2

VALUES  ( 'Name03', 'Female' );

INSERT  INTO @PersonTableType2

VALUES  ( 'Name04', 'Male' );

INSERT  INTO @PersonTableType2

VALUES  ( 'Name05', 'Male' );

EXECUTE spInsertPersonA @PersonTableType2;

GO -- Run the previous command and begins new batch

SELECT  \*

FROM    PersonA;

GO -- Run the previous command and begins new batch

Table

Description automatically generated

--===============================================================

--T031\_02\_03

--spSearchPersonA

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

--T031\_02\_03\_01

--Create spSearchPersonA

IF ( EXISTS ( SELECT    \*

              FROM      INFORMATION\_SCHEMA.ROUTINES

              WHERE     ROUTINE\_TYPE = 'PROCEDURE'

                        AND Left(ROUTINE\_NAME, 3) NOT IN ( 'sp\_', 'xp\_', 'ms\_' )

                        AND SPECIFIC\_NAME = 'spSearchPersonA' ) )

    BEGIN

        DROP PROCEDURE spSearchPersonA;

    END;

GO -- Run the previous command and begins new batch

CREATE PROC spSearchPersonA

    (

      @NameLike NVARCHAR(100) = NULL ,

      @Gender NVARCHAR(10) = NULL

       )

AS

    BEGIN

        SELECT  \*

        FROM    PersonA p

        WHERE   ( p.[Name] LIKE ('%' +  @NameLike + '%')

                  OR @NameLike IS NULL

                )

                AND

                           ( p.Gender = @Gender

                  OR @Gender IS NULL

                )

    END;

GO -- Run the previous command and begins new batch

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

--T031\_02\_03\_02

--Test spSearchPersonA

EXECUTE spSearchPersonA

EXECUTE spSearchPersonA @Gender='Male'

EXECUTE spSearchPersonA @NameLike='04', @Gender='Male'

Table

Description automatically generated with medium confidence

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

3. Set up SQL Authentication

Do **not** Execute

--Clean up

in previous section

In SQL server

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

-->

General Tab

Login Name :

Tester

Password:

1234

Default Database:

**Sample**

-->

Server Roles Tab

Select

**sysadmin**

-->

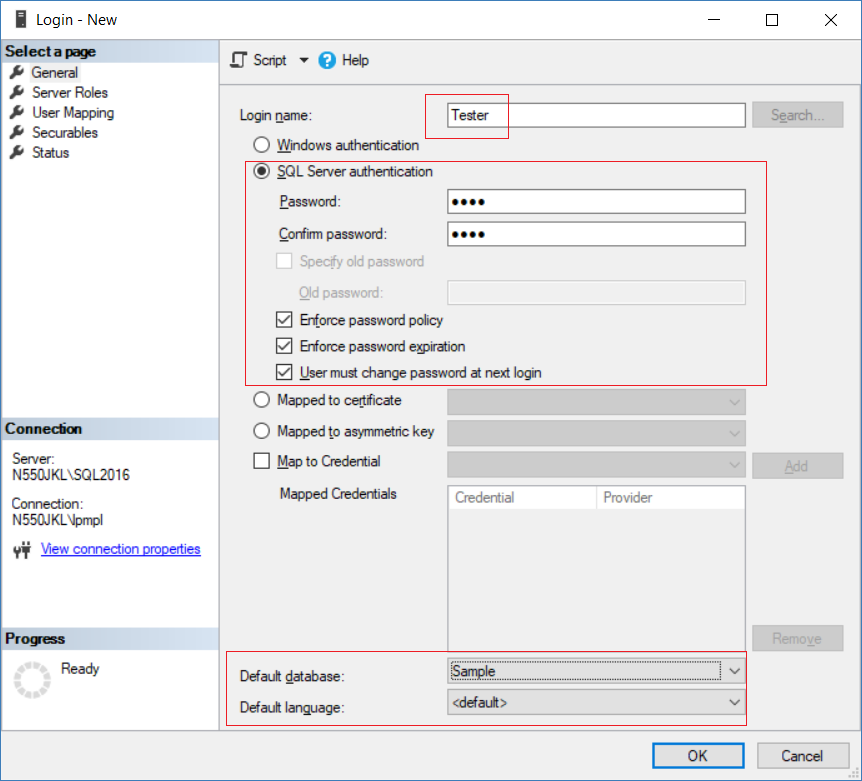
User Mapping Tab

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

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

4. Create Web Application

Do **not** Execute

--Clean up

in previous section

New Project --> Web --> [ASP.NET](http://asp.net/) Web Application (.Net Framework)

-->

Name:

**Sample**

--> Web Forms --> OK

4.1. Web.config

Add connection String

Graphical user interface, text, application, email

Description automatically generated

<configuration>

  <connectionStrings>

    <add name="SampleConnectionString" connectionString="Data Source=N550JKL\SQL2016;Initial Catalog=Sample;User ID=Tester;Password=1234"

        providerName="System.Data.SqlClient" />

  </connectionStrings>

.....

4.2. WebForm1.aspx

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

<!DOCTYPE html>

<html xmlns="[http://www.w3.org/1999/xhtml">](http://www.w3.org/1999/xhtml%22%3E);

<head runat="server">

    <title></title>

</head>

<body>

    <form id="form1" runat="server">

        <div>

            <asp:Button ID="btnDummy" runat="server" Text="Fill Dummy Data"

                OnClick="btnDummy\_Click" />

            <br />

            <br />

            <table>

                <tr>

                    <td>Name :

                        <asp:TextBox ID="tbName1" runat="server"></asp:TextBox>

                    </td>

                    <td>Gender :

                        <asp:TextBox ID="tbGender1" runat="server"></asp:TextBox>

                    </td>

                </tr>

                <tr>

                    <td>Name :

                        <asp:TextBox ID="tbName2" runat="server"></asp:TextBox>

                    </td>

                    <td>Gender :

                        <asp:TextBox ID="tbGender2" runat="server"></asp:TextBox>

                    </td>

                </tr>

                <tr>

                    <td>Name :

                        <asp:TextBox ID="tbName3" runat="server"></asp:TextBox>

                    </td>

                    <td>Gender :

                        <asp:TextBox ID="tbGender3" runat="server"></asp:TextBox>

                    </td>

                </tr>

                <tr>

                    <td>Name :

                        <asp:TextBox ID="tbName4" runat="server"></asp:TextBox>

                    </td>

                    <td>Gender :

                        <asp:TextBox ID="tbGender4" runat="server"></asp:TextBox>

                    </td>

                </tr>

                <tr>

                    <td>Name :

                        <asp:TextBox ID="tbName5" runat="server"></asp:TextBox>

                    </td>

                    <td>Gender :

                        <asp:TextBox ID="tbGender5" runat="server"></asp:TextBox>

                    </td>

                </tr>

            </table>

            <br />

            <asp:Button ID="btnInsert" runat="server" Text="Insert"

                OnClick="btnInsert\_Click" />

        </div>

        <br />

        <br/>

        <div>

            <table>

                <tr>

                    <td colspan="4">

                        <b>Search Person</b>

                    </td>

                </tr>

                <tr>

                    <td>

                        <b>Name</b>

                    </td>

                    <td>

                        <asp:TextBox ID="tbNameLike" runat="server"></asp:TextBox>

                    </td>

                    <td>

                        <b>Gender</b>

                    </td>

                    <td>

                        <asp:TextBox ID="tbGender" runat="server"></asp:TextBox>

                    </td>

                </tr>

                <tr>

                    <td colspan="4">

                        <asp:Button ID="btnSerach" runat="server" Text="Search"

                            OnClick="btnSerach\_Click" />

                    </td>

                </tr>

                <tr>

                    <td colspan="4">

                        <asp:GridView ID="gvGetData" runat="server">

                        </asp:GridView>

                    </td>

                </tr>

            </table>

        </div>

    </form>

</body>

</html>

4.3. WebForm1.aspx.cs

using System;

using System.Collections.Generic;

using System.Configuration;

using System.Data;

using System.Data.SqlClient;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace Sample

{

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

    {

        protected void Page\_Load(object sender, EventArgs e)

        {

            if (!IsPostBack)

                GetData();

        }

        private DataTable GetInsetData()

        {

            DataTable dt = new DataTable();

            dt.Columns.Add("Name");

            dt.Columns.Add("Gender");

            dt.Rows.Add(tbName1.Text, tbGender1.Text);

            dt.Rows.Add(tbName2.Text, tbGender2.Text);

            dt.Rows.Add(tbName3.Text, tbGender3.Text);

            dt.Rows.Add(tbName4.Text, tbGender4.Text);

            dt.Rows.Add(tbName5.Text, tbGender5.Text);

            return dt;

        }

        protected void btnInsert\_Click(object sender, EventArgs e)

        {

            string cs = ConfigurationManager.ConnectionStrings["SampleConnectionString"].ConnectionString;

            using (SqlConnection con = new SqlConnection(cs))

            {

                SqlCommand cmd = new SqlCommand("spInsertPersonA", con);

                cmd.CommandType = CommandType.StoredProcedure;

                SqlParameter tableValueParameter = new SqlParameter()

                {

                    ParameterName = "@PersonTableType",

                    Value = GetInsetData()

                };

                cmd.Parameters.Add(tableValueParameter);

                con.Open();

                cmd.ExecuteNonQuery();

                con.Close();

            }

        }

        protected void btnDummy\_Click(object sender, EventArgs e)

        {

            tbName1.Text = "Name01";

            tbName2.Text = "Name02";

            tbName3.Text = "Name03";

            tbName4.Text = "Name04";

            tbName5.Text = "Name05";

            tbGender1.Text = "Female";

            tbGender2.Text = "Female";

            tbGender3.Text = "Male";

            tbGender4.Text = "Female";

            tbGender5.Text = "Male";

        }

        //Search Table-----------------------------------------------------------------------

        protected void btnSerach\_Click(object sender, EventArgs e)

        {

            GetData();

        }

        private void AttachParameter(SqlCommand command, string parameterName, Control control)

        {

            if (control is TextBox && ((TextBox)control).Text != string.Empty)

            {

                var parameter = new SqlParameter(parameterName, ((TextBox)control).Text);

                command.Parameters.Add(parameter);

            }

            else if (control is DropDownList && ((DropDownList)control).SelectedValue != "-1")

            {

                var parameter = new SqlParameter(parameterName, ((DropDownList)control).SelectedValue);

                command.Parameters.Add(parameter);

            }

        }

        private void GetData()

        {

            //string cs = ConfigurationManager.ConnectionStrings["DBCS"].ConnectionString;

            string cs = ConfigurationManager.ConnectionStrings["SampleConnectionString"].ConnectionString;

            using (var con = new SqlConnection(cs))

            {

                var cmd = new SqlCommand("spSearchPersonA", con);

                cmd.CommandType = CommandType.StoredProcedure;

                AttachParameter(cmd, "@NameLike", tbNameLike);

                AttachParameter(cmd, "@Gender", tbGender);

                con.Open();

                gvGetData.DataSource = cmd.ExecuteReader();

                gvGetData.DataBind();

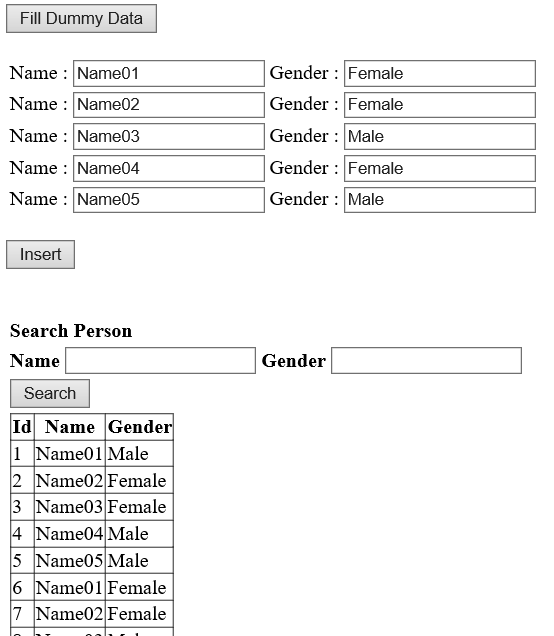
            }

        }

    }

}

<http://localhost:62110/WebForm1.aspx>



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

5. Clean up

--===============================================================

--T031\_03\_Clean up

--===============================================================

--Drop Table if it exists.

IF ( EXISTS ( SELECT    \*

              FROM      INFORMATION\_SCHEMA.TABLES

              WHERE     TABLE\_NAME = 'PersonA' ) )

    BEGIN

        TRUNCATE TABLE dbo.PersonA;

        DROP TABLE PersonA;

    END;

GO -- Run the previous command and begins new batch

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

--Drop Stored Procedure if it exists.

IF ( EXISTS ( SELECT    \*

              FROM      INFORMATION\_SCHEMA.ROUTINES

              WHERE     ROUTINE\_TYPE = 'PROCEDURE'

                        AND LEFT(ROUTINE\_NAME, 3) NOT IN ( 'sp\_', 'xp\_', 'ms\_' )

                        AND SPECIFIC\_NAME = 'spInsertPersonA' ) )

    BEGIN

        DROP PROCEDURE spInsertPersonA;

    END;

GO -- Run the previous command and begins new batch

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

--Drop Table Type if it exists.

IF EXISTS ( SELECT  \*

            FROM    sys.types

            WHERE   is\_table\_type = 1

                    AND name = 'PersonType' )

    BEGIN

        DROP TYPE PersonType;

    END;

GO -- Run the previous command and begins new batch

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

--Drop Stored Procedure if it exists.

IF ( EXISTS ( SELECT    \*

              FROM      INFORMATION\_SCHEMA.ROUTINES

              WHERE     ROUTINE\_TYPE = 'PROCEDURE'

                        AND Left(ROUTINE\_NAME, 3) NOT IN ( 'sp\_', 'xp\_', 'ms\_' )

                        AND SPECIFIC\_NAME = 'spSearchPersonA' ) )

    BEGIN

        DROP PROCEDURE spSearchPersonA;

    END;

GO -- Run the previous command and begins new batch