

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

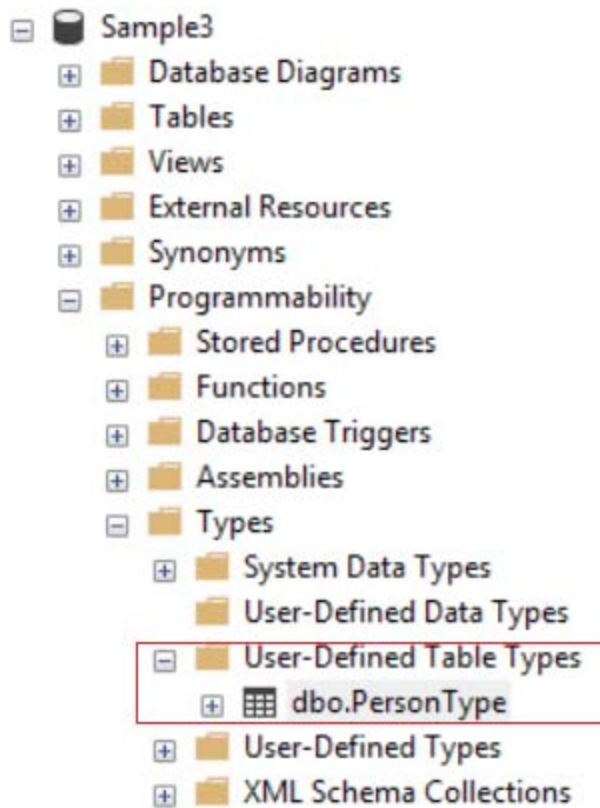
```
--=====
--T031_01_Stored Procedure with TableValueType Parameter
--=====
--T031_01_01
--Sample code to create a Table
--If Table exists then DROP it
IF ( EXISTS ( SELECT      *
                FROM        INFORMATION_SCHEMA.TABLES
                WHERE       TABLE_NAME = 'PersonA' ) )
BEGIN
    TRUNCATE 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...
--);
*/
```



```
-----
--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
```

	Id	Name	Gender
1	1	Name01	Male
2	2	Name02	Female
3	3	Name03	Female
4	4	Name04	Male
5	5	Name05	Male

```

=====
--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 ASP.NET

```

=====
--T031_02_Stored Procedure with TableValueType Parameter in ASP.NET
=====
--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;

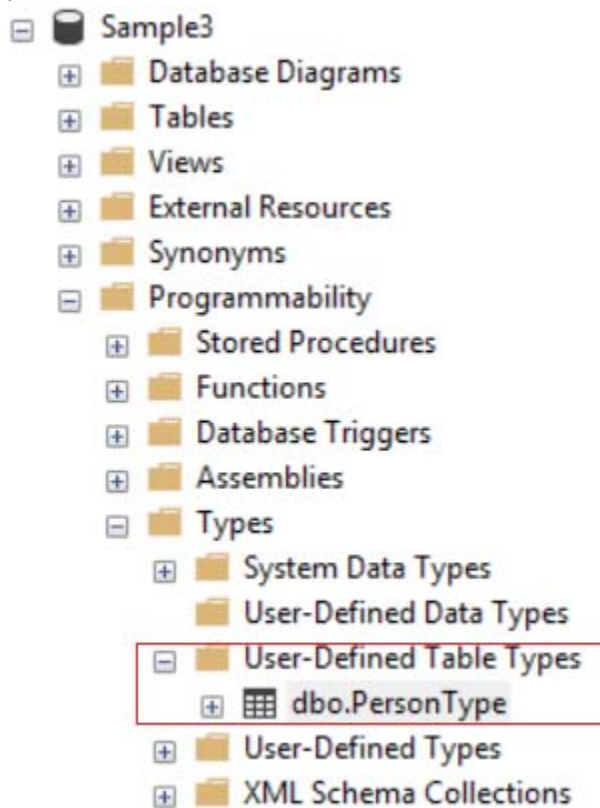
GO -- Run the previous command and begins new batch
--Create Table Type
CREATE TYPE PersonType AS TABLE
(
    [Name] NVARCHAR(100),
    Gender NVARCHAR(10)
);

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...
--);
*/
```



```
-----
--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
```

	Id	Name	Gender
1	1	Name01	Male
2	2	Name02	Female
3	3	Name03	Female
4	4	Name04	Male
5	5	Name05	Male

```

=====
--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'

```

	Id	Name	Gender
1	1	Name01	Male
2	2	Name02	Female
3	3	Name03	Female
4	4	Name04	Male
5	5	Name05	Male

	Id	Name	Gender
1	1	Name01	Male
2	4	Name04	Male
3	5	Name05	Male

	Id	Name	Gender
1	4	Name04	Male

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

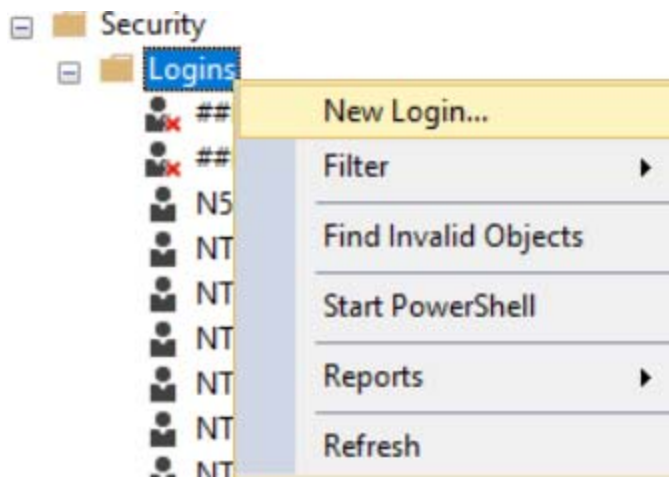
sysadmin

-->

User Mapping Tab

Select **Sample**

Select every Roles.



Login - New

Select a page

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

Connection

Server: N55QJKL\SQL2016

Connection: N55QJKL\pmpl

[View connection properties](#)

Progress

Ready

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 Add

Mapped Credentials

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

Remove

Default database:

Default language:

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\pmp1

View connection properties

Progress

Ready

OK

Cancel

Login Properties - Tester

Select a page

General

Server Roles

User Mapping

Securables

Status

Script

Help

Users mapped to this login:

Map	Database	User	Default Schema
<input type="checkbox"/>	Movie3		
<input type="checkbox"/>	NORTHWND		
<input type="checkbox"/>	Northwind		
<input type="checkbox"/>	ReportServer\$SQL2016		
<input type="checkbox"/>	ReportServer\$SQL20...		
<input checked="" type="checkbox"/>	Sample	Tester	dbo
<input type="checkbox"/>	master		
<input type="checkbox"/>	model		
<input type="checkbox"/>	msdb		
<input type="checkbox"/>	tempdb		

☐ Guest account enabled for: Sample

Database role membership for: Sample

☒ db_accessadmin

☒ db_backupoperator

☒ db_datareader

☒ db_datawriter

☒ db_ddladmin

☒ db_denydatareader

☒ db_denydatawriter

☒ db_owner

☒ db_securityadmin

☒ public

Connection

Server:
N550JKL\SQL2016

Connection:
N550JKL\pmp1

View connection properties

Progress

Ready

OK

Cancel

=====

4. Create Web Application

Do **not** Execute

--Clean up
in previous section

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

-->

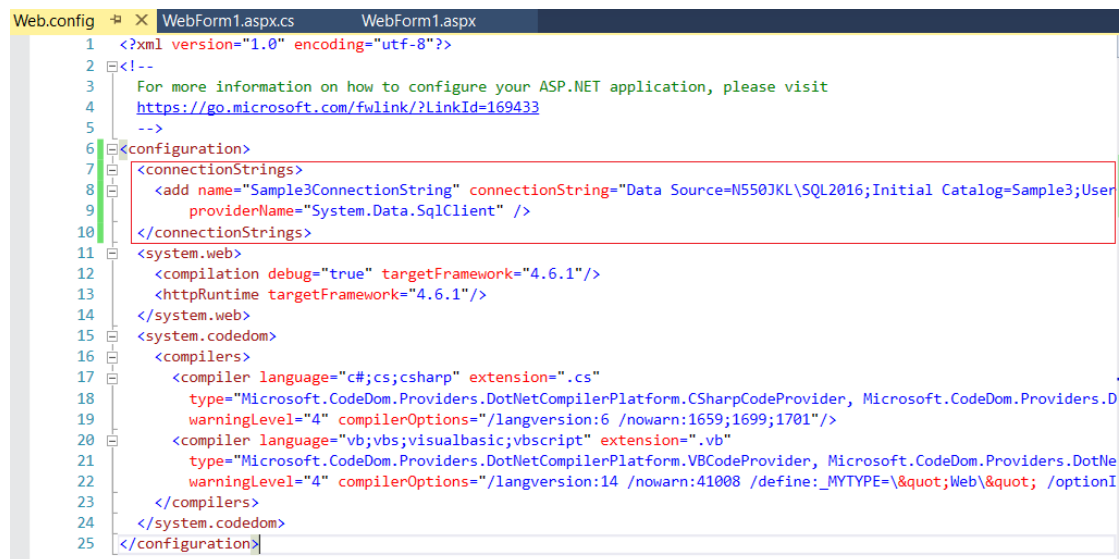
Name:

Sample

--> Web Forms --> OK

4.1. Web.config

Add connection String



```
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="Sample3ConnectionString" connectionString="Data Source=N550JKL\SQL2016;Initial Catalog=Sample3;User
9       providerName="System.Data.SqlClient" />
10   </connectionStrings>
11   <system.web>
12     <compilation debug="true" targetFramework="4.6.1"/>
13     <httpRuntime targetFramework="4.6.1"/>
14   </system.web>
15   <system.codedom>
16     <compilers>
17       <compiler language="c#;cs;csharp" extension=".cs"
18         type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.CSharpCodeProvider, Microsoft.CodeDom.Providers.D
19         warningLevel="4" compilerOptions="/langversion:6 /nowarn:1659;1699;1701"/>
20       <compiler language="vb;vbs;visualbasic;vbscript" extension=".vb"
21         type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.VBCodeProvider, Microsoft.CodeDom.Providers.DotNe
22         warningLevel="4" compilerOptions="/langversion:14 /nowarn:41008 /define:_MYTYPE=\"Web\" /optionI
23     </compilers>
24   </system.codedom>
25 </configuration>
```

```
<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">
<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">

```



```

        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>

Fill Dummy Data

Name :	<input type="text" value="Name01"/>	Gender :	<input type="text" value="Female"/>
Name :	<input type="text" value="Name02"/>	Gender :	<input type="text" value="Female"/>
Name :	<input type="text" value="Name03"/>	Gender :	<input type="text" value="Male"/>
Name :	<input type="text" value="Name04"/>	Gender :	<input type="text" value="Female"/>
Name :	<input type="text" value="Name05"/>	Gender :	<input type="text" value="Male"/>

Insert

Search Person

Name Gender

Search

Id	Name	Gender
1	Name01	Male
2	Name02	Female
3	Name03	Female
4	Name04	Male
5	Name05	Male
6	Name01	Female
7	Name02	Female

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

```