

Practical 10

Aim:- Design a webpage to demonstrate the working of a simple stored procedure.

SQL Server stored procedure is a batch of statements grouped as a logical unit and stored in the database. The stored procedure accepts the parameters and executes the T-SQL statements in the procedure, returns the result set if any.

Benefits of using a stored procedure

It can be easily modified:

Reduced network traffic

Reusable:

Security:

Performance:

Stored procedures (SPs) in SQL Server are just like procedures/routines in other DBMSs or programming languages. Each procedure has one or more statements. In our case, these are SQL statements. So, you can write a procedure that will – insert new data, update or delete existing, retrieve data using the SELECT statement. And even better, you can combine more (different statements) in the stored procedures. Also, inside the procedure, you can call another SP, function, use the IF statement, etc. Therefore, it's pretty obvious SP can do much more than a single select query.

WebForm1.aspx:

dbo.insertData.sql WebForm1.aspx.cs WebForm1.aspx Web.config

DataBase Access Using Connected Archieture Using Stored Procedure

Enter Loan Id:

Enter Loan User Name:

Enter User Address:

| Column0 | Column1 | Column2 |
|---------|---------|---------|
| abc | abc | abc |
| abc | abc | abc |
| abc | abc | abc |
| abc | abc | abc |
| abc | abc | abc |

Add Using Stored Procedure

Source code:

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

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
</div>
<asp:Label ID="Label1" runat="server" Text="DataBase Access Using Connected
Archieture Using Stored Procedure"></asp:Label>
<p>
<asp:Label ID="Label2" runat="server" Text="Enter Loan Id:"></asp:Label>
<asp:TextBox ID="txtRollNo" runat="server"></asp:TextBox>
</p>
<asp:Label ID="Label3" runat="server" Text="Enter Loan User
Name:"></asp:Label>
<asp:TextBox ID="txtName" runat="server"></asp:TextBox>
<p>
<asp:Label ID="Label4" runat="server" Text="Enter User
Address:"></asp:Label>
```

```

        <asp:TextBox ID="txtAddress" runat="server"></asp:TextBox>
    </p>
    <asp:GridView ID="gvStudDetails" runat="server">
</asp:GridView>
    <p>
        <asp:Button ID="Button5" runat="server" OnClick="Button5_Click" Text="Add
Using Stored Procedure" />
    </p>
</form>
</body>
</html>

```

WebForm1.aspx.cs:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

using System.Configuration; //add this for connectionString from web.config file
using System.Data;
using System.Data.SqlClient; //name spaces for MS SQL server database classes

namespace ConnectedArchiectureStoredProcedure
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        static string conStr =
ConfigurationManager.ConnectionStrings["LoanDB"].ToString();
        SqlConnection conn = null;
        SqlCommand cmd = null;
        SqlDataReader dr = null;
        DataTable dt = null;

        public void clearTextBox()
        {
            txtRollNo.Text = "";
            txtName.Text = "";
            txtAddress.Text = "";
        }

        public void showData()
        {
            try
            {
                cmd = new SqlCommand("SELECT * FROM loan", conn);
                //cmd=new SqlCommand();
                //cmd.Connection=conn;
                //cmd.CommandText="SELECT * FROM stud";
                //cmd.CommandType=CommandType.Text;
                if (conn.State == ConnectionState.Closed)
                {
                    conn.Open();
                }
                dt = new DataTable();
                dr = cmd.ExecuteReader();
                dt.Load(dr);
                gvStudDetails.DataSource = dt;
            }
            catch { }
        }
    }
}

```

```

        gvStudDetails.DataBind();
    }
    catch (Exception ex)
    {
        Response.Write("<script
type=\"text/javascript\">alert('Exception!' + ex.Message);</script>");
    }
    finally
    {
        conn.Close();
    }
}
protected void Page_Load(object sender, EventArgs e)
{
    conn = new SqlConnection(conStr);
    showData();
}

protected void Button5_Click(object sender, EventArgs e)
{
    try
    {
        if (txtRollNo.Text != "" && txtName.Text != "" && txtAddress.Text
!= "")
        {
            cmd = new SqlCommand();
            if (conn.State == ConnectionState.Closed)
            {
                conn.Open();
            }
            cmd.Connection = conn;
            //set command type as stored procedure
            cmd.CommandType = CommandType.StoredProcedure;
            //pass the stored procedure name
            cmd.CommandText = "insertData";

            cmd.Parameters.Add(new SqlParameter("@LoanId",
SqlDbType.SmallInt)).Value = Convert.ToInt16(txtRollNo.Text);
            cmd.Parameters.Add(new SqlParameter("@Name",
SqlDbType.VarChar)).Value = txtName.Text;
            cmd.Parameters.Add(new SqlParameter("@add",
SqlDbType.VarChar)).Value = txtAddress.Text;

            int r = cmd.ExecuteNonQuery();
            if (r != 0)
            {
                Response.Write("<script
type=\"text/javascript\">alert('Record inserted successfully!');</script>");
            }
            else
            {
                Response.Write("<script
type=\"text/javascript\">alert('Record not inserted successfully');</script>");
            }
        }
    }
    else
    {

```

```

        Response.Write("<script type=\"text/javascript\">alert('Please
enter all details properly');</script>");
    }
}
catch (Exception ex)
{
    Response.Write("<script
type=\"text/javascript\">alert('Exception!' + ex.Message);</script>");
}
finally
{
    conn.Close();
    clearTextBox();
    showData();
}
}
}
}

```

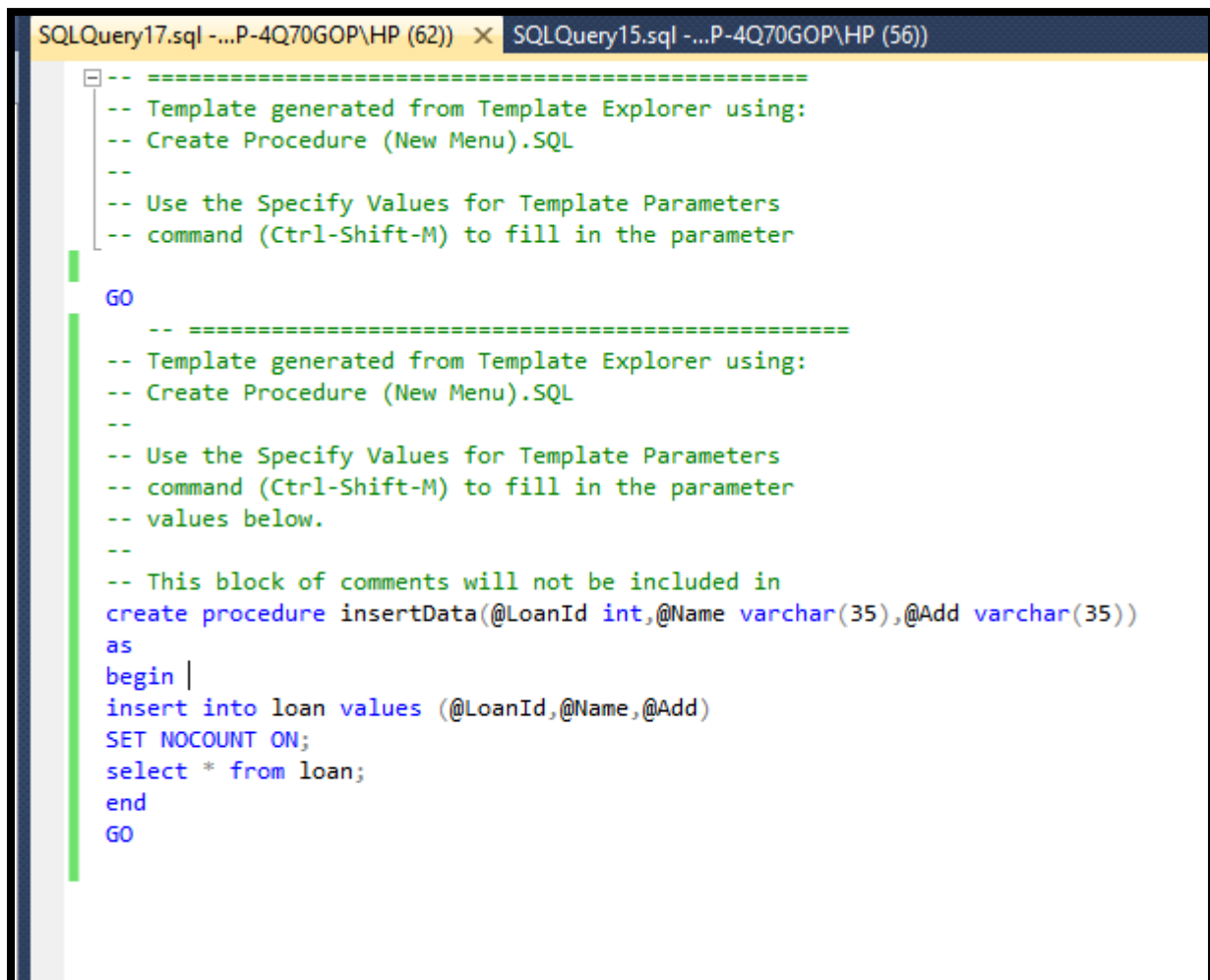
WebConfig:

```

<?xml version="1.0" encoding="utf-8"?>
<!--
    For more information on how to configure your ASP.NET application, please visit
    https://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
  <system.web>
    <compilation debug="true" targetFramework="4.7.2" />
    <httpRuntime targetFramework="4.7.2" />
  </system.web>
  <system.codedom>
    <compilers>
      <compiler language="c#;cs;csharp" extension=".cs"
type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.CSharpCodeProvider,
Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=2.0.1.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35" warningLevel="4"
compilerOptions="/langversion:default /nowarn:1659;1699;1701" />
      <compiler language="vb;vbs;visualbasic;vbscript" extension=".vb"
type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.VBCodeProvider,
Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=2.0.1.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35" warningLevel="4"
compilerOptions="/langversion:default /nowarn:41008 /define:_MYTYPE=\"Web\";
/optionInfer+" />
    </compilers>
  </system.codedom>
  <connectionStrings>
    <add name="LoanDB"
        connectionString="Data Source=DESKTOP-4Q70G0P\\SQLEXPRESS;Initial
Catalog=LoanDB;Integrated Security=True"
        providerName="System.Data.SqlClient"/>
  </connectionStrings>
</configuration>

```

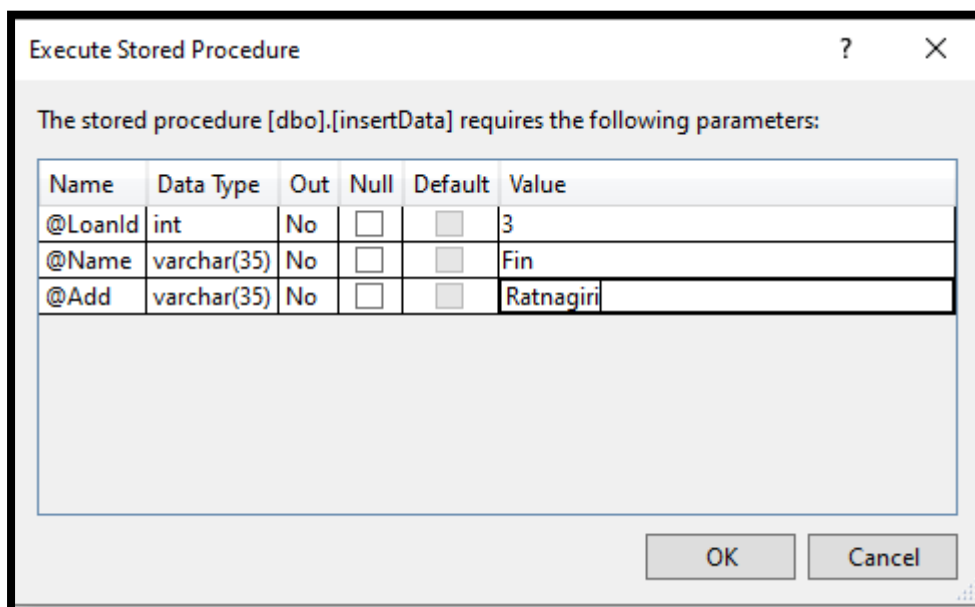
SqlStoredProcedure:



The screenshot shows a SQL Server Enterprise Manager interface with two query windows open. The active window, titled 'SQLQuery15.sql -...P-4Q70GOP\HP (56)', contains a SQL script. The script is a template for creating a stored procedure named 'insertData'. It includes comments explaining the template's origin and usage. The script defines three parameters: @LoanId (int), @Name (varchar(35)), and @Add (varchar(35)). The procedure body includes a 'begin' block with an 'insert into loan values' statement, 'SET NOCOUNT ON;', and a 'select * from loan;' statement. The script ends with 'end' and 'GO'.

```
-- =====
-- Template generated from Template Explorer using:
-- Create Procedure (New Menu).SQL
--
-- Use the Specify Values for Template Parameters
-- command (Ctrl-Shift-M) to fill in the parameter
GO

-- =====
-- Template generated from Template Explorer using:
-- Create Procedure (New Menu).SQL
--
-- Use the Specify Values for Template Parameters
-- command (Ctrl-Shift-M) to fill in the parameter
-- values below.
--
-- This block of comments will not be included in
create procedure insertData(@LoanId int,@Name varchar(35),@Add varchar(35))
as
begin |
insert into loan values (@LoanId,@Name,@Add)
SET NOCOUNT ON;
select * from loan;
end
GO
```



The screenshot shows the 'Execute Stored Procedure' dialog box. The title bar reads 'Execute Stored Procedure'. The main text states: 'The stored procedure [dbo].[insertData] requires the following parameters:'. Below this is a table with columns: Name, Data Type, Out, Null, Default, and Value. The table contains three rows of parameters: @LoanId (int), @Name (varchar(35)), and @Add (varchar(35)). The 'Value' column contains the values 3, Fin, and Ratnagiri respectively. At the bottom right are 'OK' and 'Cancel' buttons.

| Name | Data Type | Out | Null | Default | Value |
|---------|-------------|-----|--------------------------|--------------------------|-----------|
| @LoanId | int | No | <input type="checkbox"/> | <input type="checkbox"/> | 3 |
| @Name | varchar(35) | No | <input type="checkbox"/> | <input type="checkbox"/> | Fin |
| @Add | varchar(35) | No | <input type="checkbox"/> | <input type="checkbox"/> | Ratnagiri |

100 % No issues found

T-SQL Results Message

| | loanid | name | address |
|---|--------|---------|----------|
| 1 | 1 | Madhuri | Kudal |
| 2 | 2 | Alex | kank... |
| 3 | 3 | eee | ttt |
| 4 | 3 | Fin | Ratna... |

Output:

https://localhost:44372/WebForm x +

localhost:44372/WebForm1.aspx

DataBase Access Using Connected Archieture Using Stored Procedure

Enter Loan Id:

Enter Loan User Name:

Enter User Address:

| loanid | name | address |
|--------|---------|-----------|
| 1 | Madhuri | Kudal |
| 2 | Alex | kankavli |
| 3 | eee | ttt |
| 3 | Fin | Ratnagiri |

Add Using Stored Procedure

https://localhost:44372/WebForm x +

localhost:44372/WebForm1.aspx

localhost:44372 says
Record inserted successfully!

OK

https://localhost:44372/WebForm x +

localhost:44372/WebForm1.aspx

DataBase Access Using Connected Archieture Using Stored Procedure

Enter Loan Id:

Enter Loan User Name:

Enter User Address:

| loanid | name | address |
|--------|----------|-----------|
| 1 | Madhuri | Kudal |
| 2 | Alex | kankavli |
| 3 | eee | ttt |
| 3 | Fin | Ratnagiri |
| 4 | Majlekar | Londan |

Add Using Stored Procedure