ASP.net Databases

While working with databases, the following concepts which are common across all databases.

- **Connection** – To work with the data in a database, the first obvious step is the connection. The connection to a database normally consists of the below-mentioned parameters.**Database name or Data Source** – The first important parameter is the database name. Each connection can only work with one database at a time.
- **Credentials** – The next important aspect is the 'username' and 'password'. This is used to establish a connection to the database.
- **Optional parameters** - You can specify optional parameters on how .net should handle the connection to the database. For example, one can specify a parameter for how long the connection should stay active.

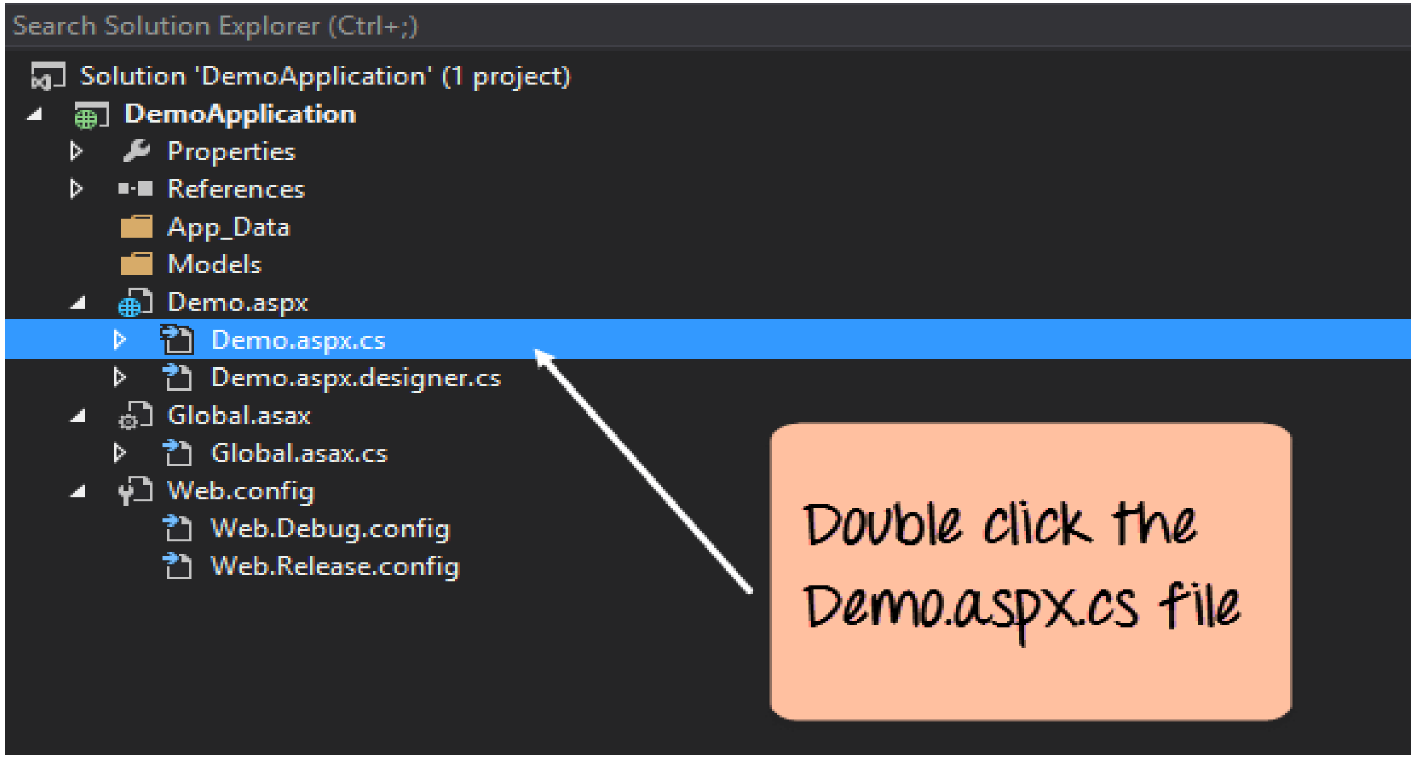
- **Selecting data from the database** – Once the connection is established, data is fetched from the database. ASP.Net has the ability to execute 'sql' select command against the database. The 'sql' statement can be used to fetch data from a specific table in the database.
- **Inserting data into the database** – ASP.Net is used to insert records into the database. Values for each row that needs to be inserted in the database are specified in ASP.Net.
- **Updating data into the database** – ASP.Net can also be used to update existing records into the database. New values can be specified in ASP.Net for each row that needs to be updated into the database.
- **Deleting data from a database** – ASP.Net can also be used to delete records from the database. The code is written to delete a particular row from the database.

- The credentials used to connect to the database are given below

  Username – sa

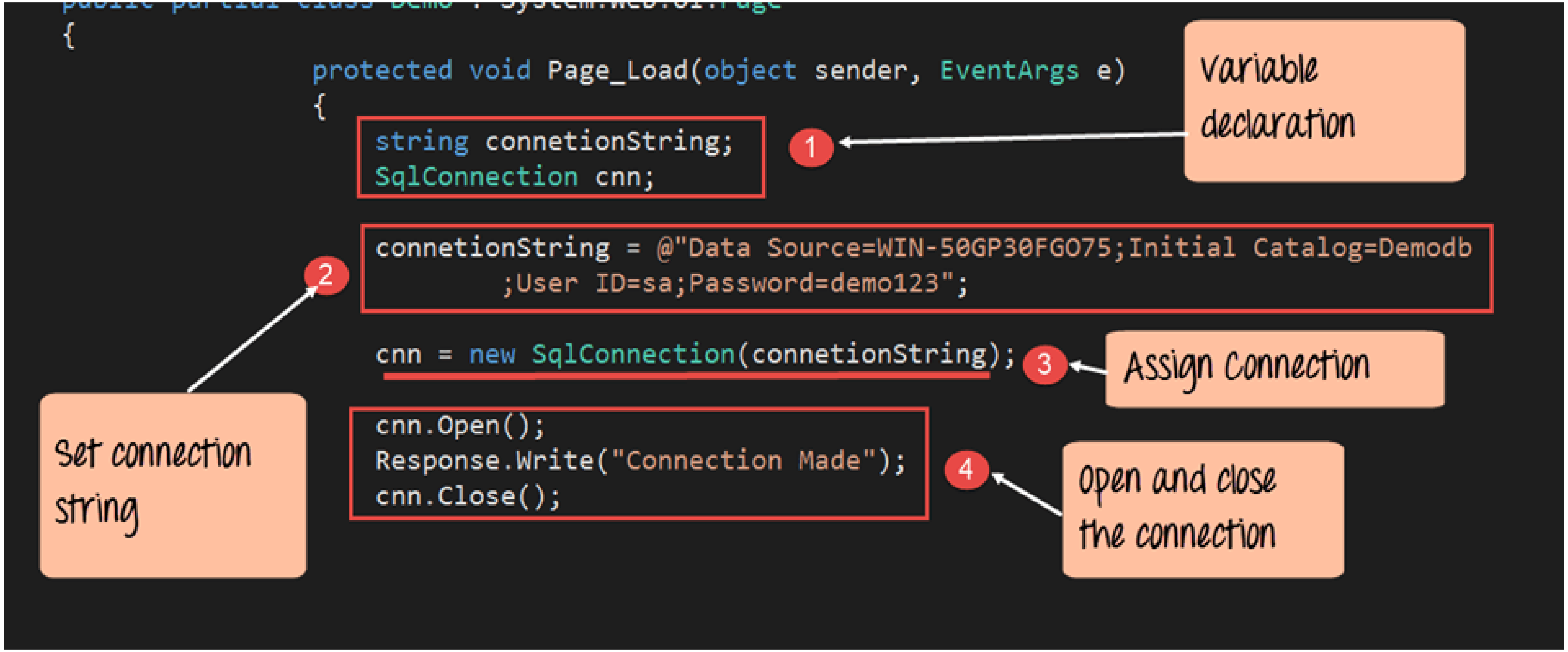
  Password – demo123

- **Step 1)** Let's first ensure that you have your web application (DemoApplication) opened in Visual Studio. Double click the 'demo.aspx.cs' file to enter the code for the database connection.

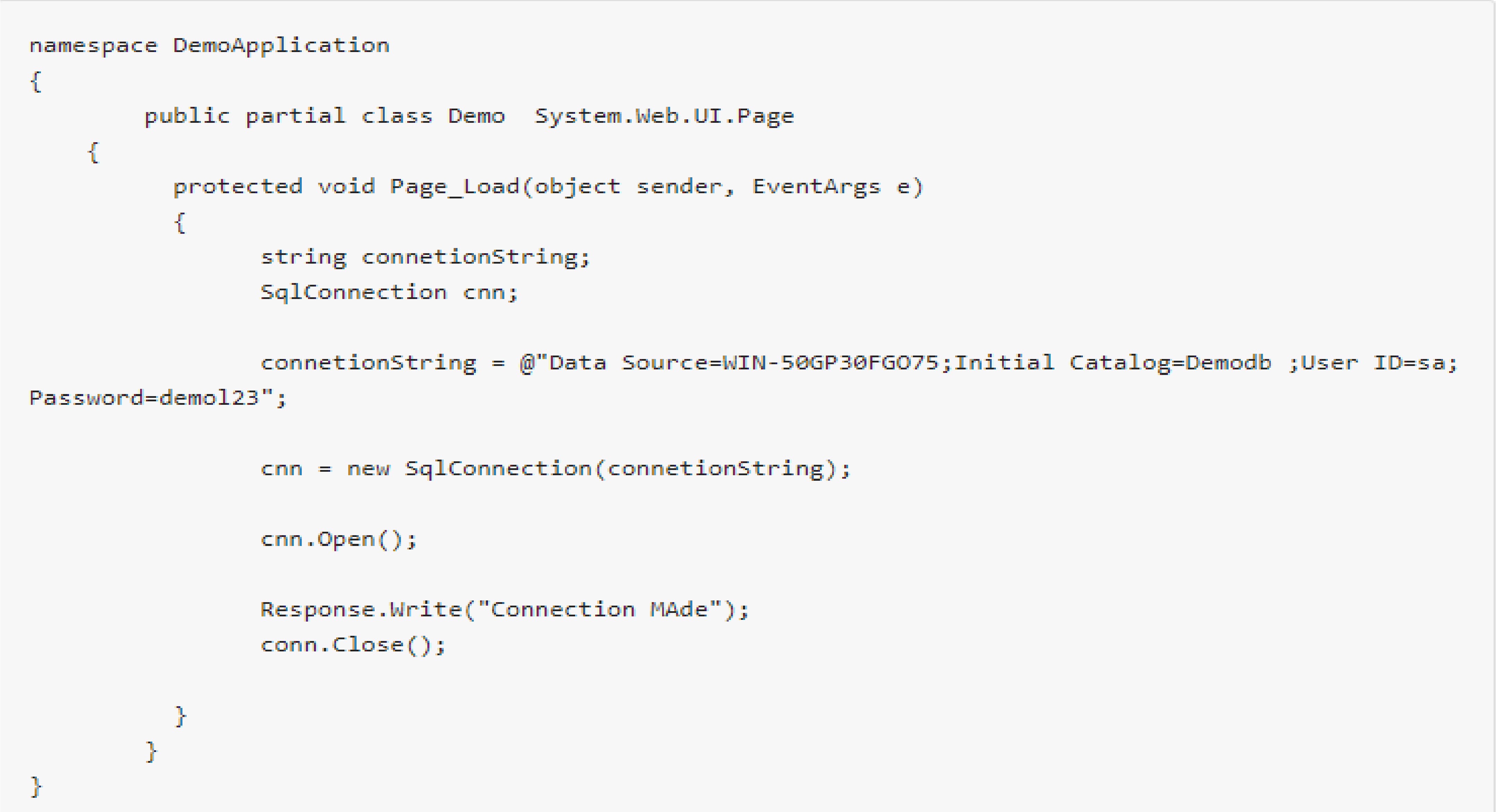


Double click the Demo.aspx.cs file

Step 2) Add the below code which will be used to establish a connection to the database.



```
public partial class Demo : System.Web.UI.Page
{
                protected void Page_Load(object sender, EventArgs e)
                {
                    string connetionString;
                    SqlConnection cnn;

                    connetionString = @"Data Source=WIN-50GP30FGO75;Initial Catalog=Demodb
                            ;User ID=sa;Password=demo123";

                    cnn = new SqlConnection(connetionString);

                    cnn.Open();
                    Response.Write("Connection Made");
                    cnn.Close();
```

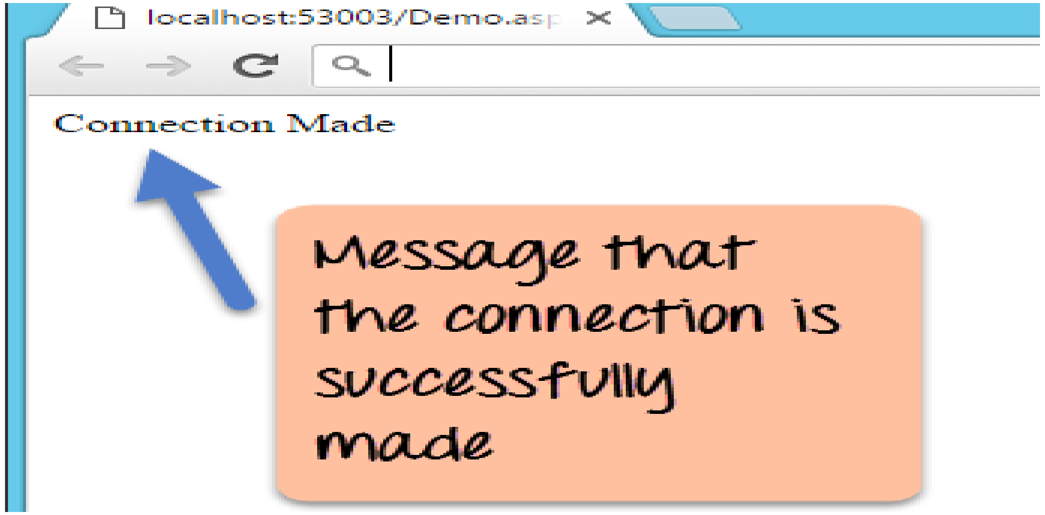


```
namespace DemoApplication
{
        public partial class Demo  System.Web.UI.Page
    {
          protected void Page_Load(object sender, EventArgs e)
          {
                string connetionString;
                SqlConnection cnn;

                connetionString = @"Data Source=WIN-50GP30FGO75;Initial Catalog=Demodb ;User ID=sa;
Password=demol23";

                cnn = new SqlConnection(connetionString);

                cnn.Open();

                Response.Write("Connection MAde");
                conn.Close();

          }
        }
}
```

Code explanation

- The first step is to create variables. It will be used to create the connection string and the connection to the SQL Server database.
- The next step is to actually create the connection string. The connection string consists of the following parts
- Data Source – This is the name of the server on which the database resides. In our case, it resides on a machine called WIN-50GP30FGO75.
- The Initial Catalog is used to specify the name of the database
- The UserID and Password are the credentials required to connect to the database.

- Next, we assign the connecting string to the variable 'cnn'.
- The variable cnn is of type SqlConnection. This is used to establish a connection to the database.
- SqlConnection is a class in ASP.Net, which is used to create a connection to a database.
- To use this class, you have to first create an object of this class. Hence, here we create a variable called 'cnn' which is of the type SqlConnection.
- Next, we use the open method of the cnn variable to open a connection to the database. We display a message to the user that the connection is established. This is done via the 'response.write' method. We then close the connection to the database.



```
Connection Made
```

Message that the connection is successfully made

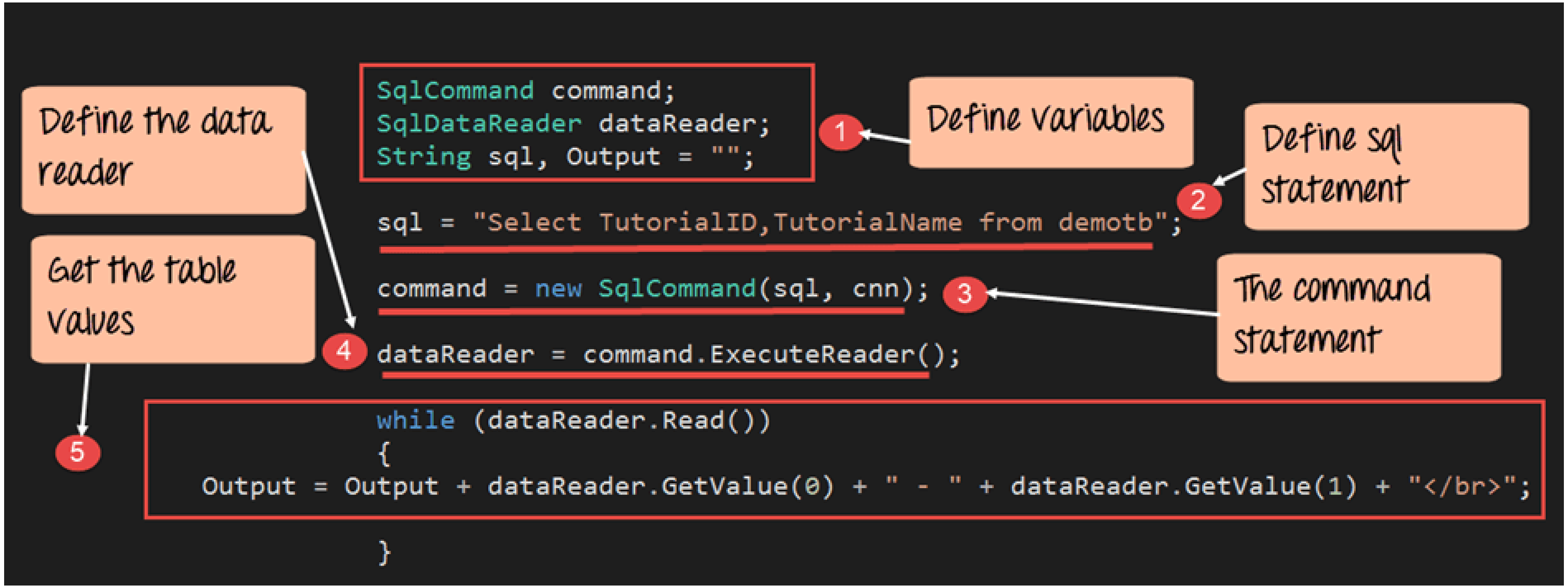
ASP.NET Read Database using SqlDataReader

1. A table called demotb. This table will be used to store the ID and names of various Tutorials.
2. The table will have two columns, one called "TutorialID" and the other called "TutorialName."
3. For the moment, the table will have two rows as shown below.

TutorialID	TutorialName
1	C#
2	ASP.Net

Step 1) Let's split the code into two parts,

- The first part will be to construct our "select" statement. It will be used to read the data from the database.
- We will then execute the "select" statement against the database. This will fetch all the table rows accordingly.



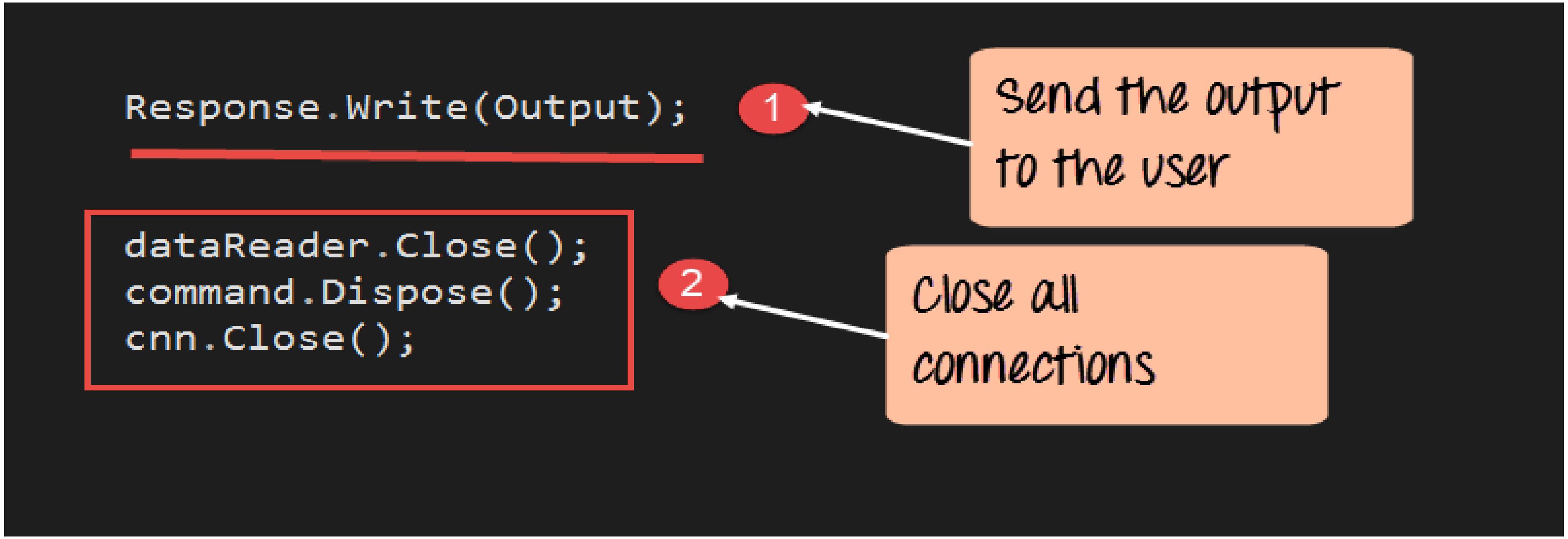
```
SqlCommand command;
SqlDataReader dataReader;
String sql, Output = "";

sql = "Select TutorialID,TutorialName from demotb";

command = new SqlCommand(sql, cnn);

dataReader = command.ExecuteReader();

while (dataReader.Read())
{
Output = Output + dataReader.GetValue(0) + " - " + dataReader.GetValue(1) + "</br>";

}
```

Code Explanation:-

- The first step is to create the following variables
- SQLCommand – The 'SQLCommand' is a class defined within C#. This class is used to perform operations of reading and writing into the database. Hence, the first step is to make sure that we create a variable type of this class. This variable will then be used in subsequent steps of reading data from our database.
- The DataReader object is used to get all the data specified by the SQL query. We can then read all the table rows one by one using the data reader.
- We then define two string variables. One is "SQL" to hold our SQL command string. The next is the "Output" which will contain all the table values.

- The next step is to actually define the SQL statement. This will be used against our database. In our case, it is "Select TutorialID, TutorialName from demotb". This will fetch all the rows from the table demotb.
- Next, we create the command object which is used to execute the SQL statement against the database. In the SQL command, you have to pass the connection object and the SQL string.
- Next, we will execute the data reader command, which will fetch all the rows from the demotb table.
- Now that we have all the rows of the table with us, we need a mechanism to access the row one by one.
- For this, we will use the 'while' statement.
- The 'while' statement will be used to access the rows from the data reader one at a time.
- We then use the 'GetValue' method to get the value of TutorialID and TutorialName.

- **Step 2)** In the final step, we will just display the output to the user. Then we will close all the objects related to the database operation.

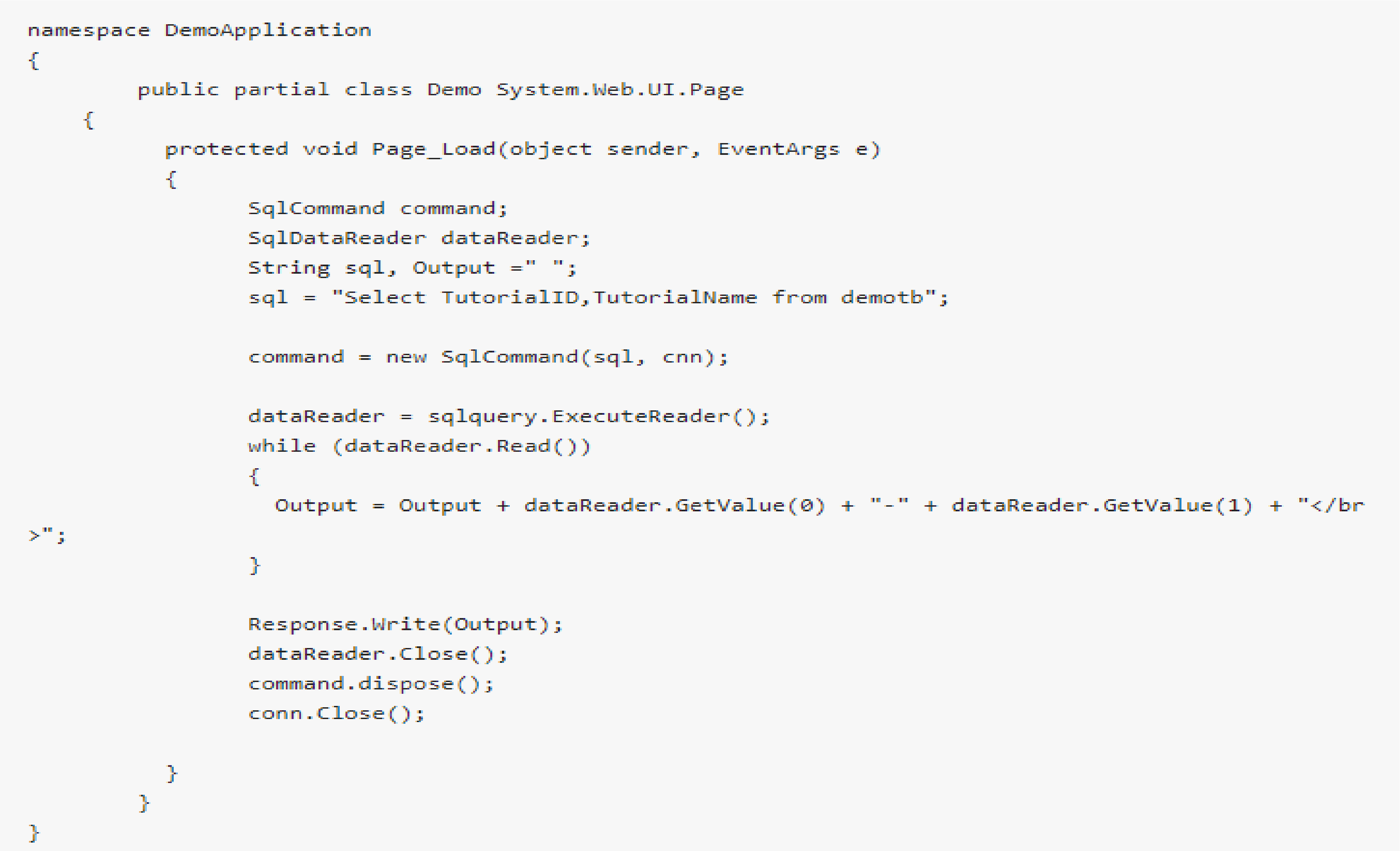


```
Response.Write(Output);
```

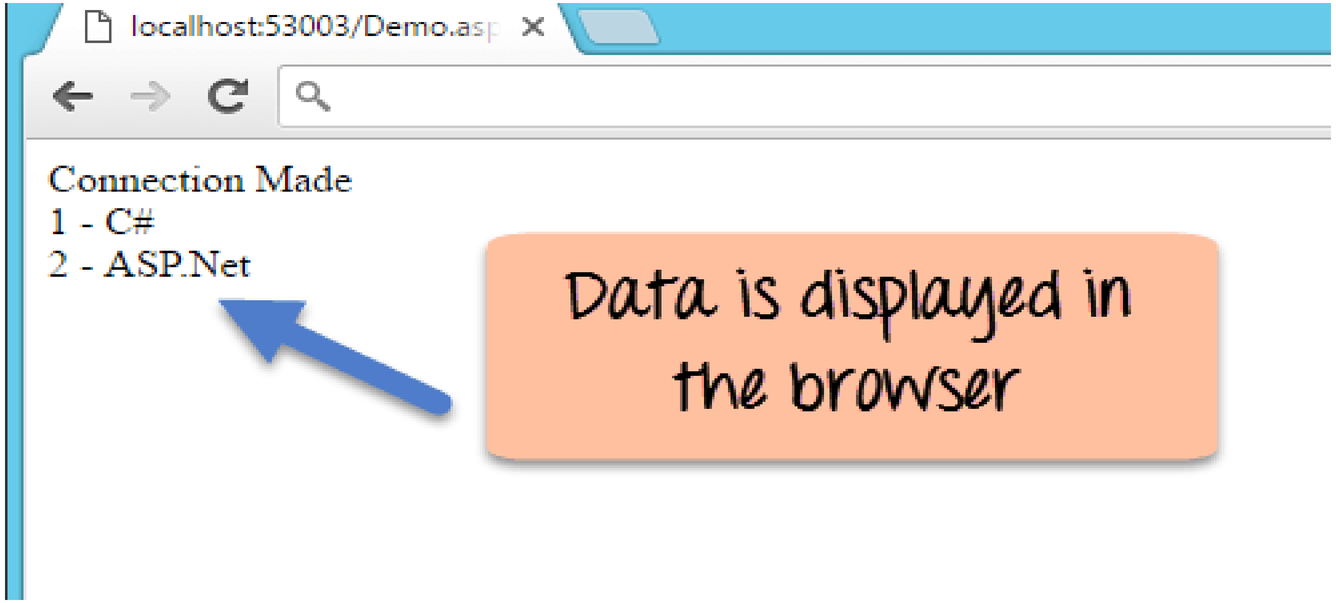
```
dataReader.Close();
command.Dispose();
cnn.Close();
```

Send the output to the user

Close all connections



```
namespace DemoApplication
{
        public partial class Demo System.Web.UI.Page
    {
          protected void Page_Load(object sender, EventArgs e)
          {
                SqlCommand command;
                SqlDataReader dataReader;
                String sql, Output =" ";
                sql = "Select TutorialID,TutorialName from demotb";

                command = new SqlCommand(sql, cnn);

                dataReader = sqlquery.ExecuteReader();
                while (dataReader.Read())
                {
                  Output = Output + dataReader.GetValue(0) + "-" + dataReader.GetValue(1) + "</br
>";
                }

                Response.Write(Output);
                dataReader.Close();
                command.dispose();
                conn.Close();

          }
        }
}
```



```
Connection Made
1 - C#
2 - ASP.Net
```

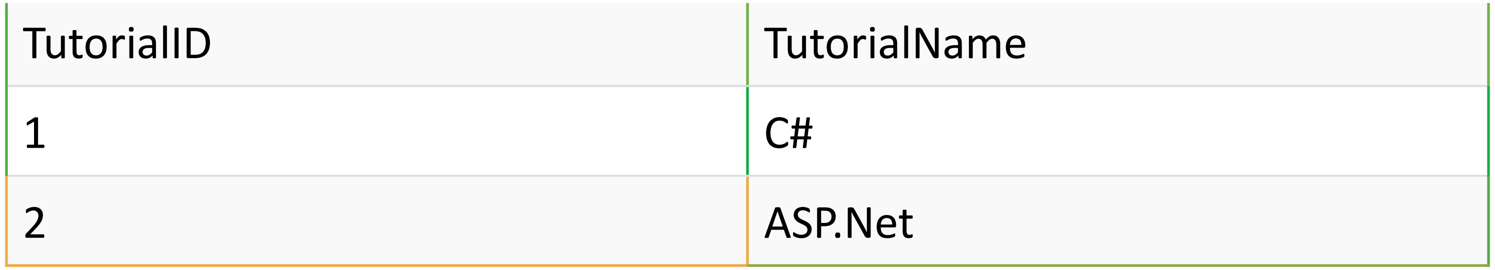
Data is displayed in the browser

Insert Database Record using InsertCommand

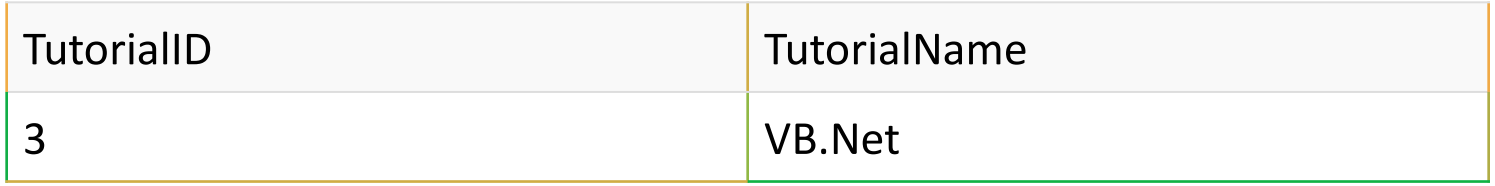
Just like Accessing data, ASP.Net has the ability to insert records into the database as well.

Let's take the same table structure used for inserting records.

Let's change the code in our form, so that we can insert the following row into the table

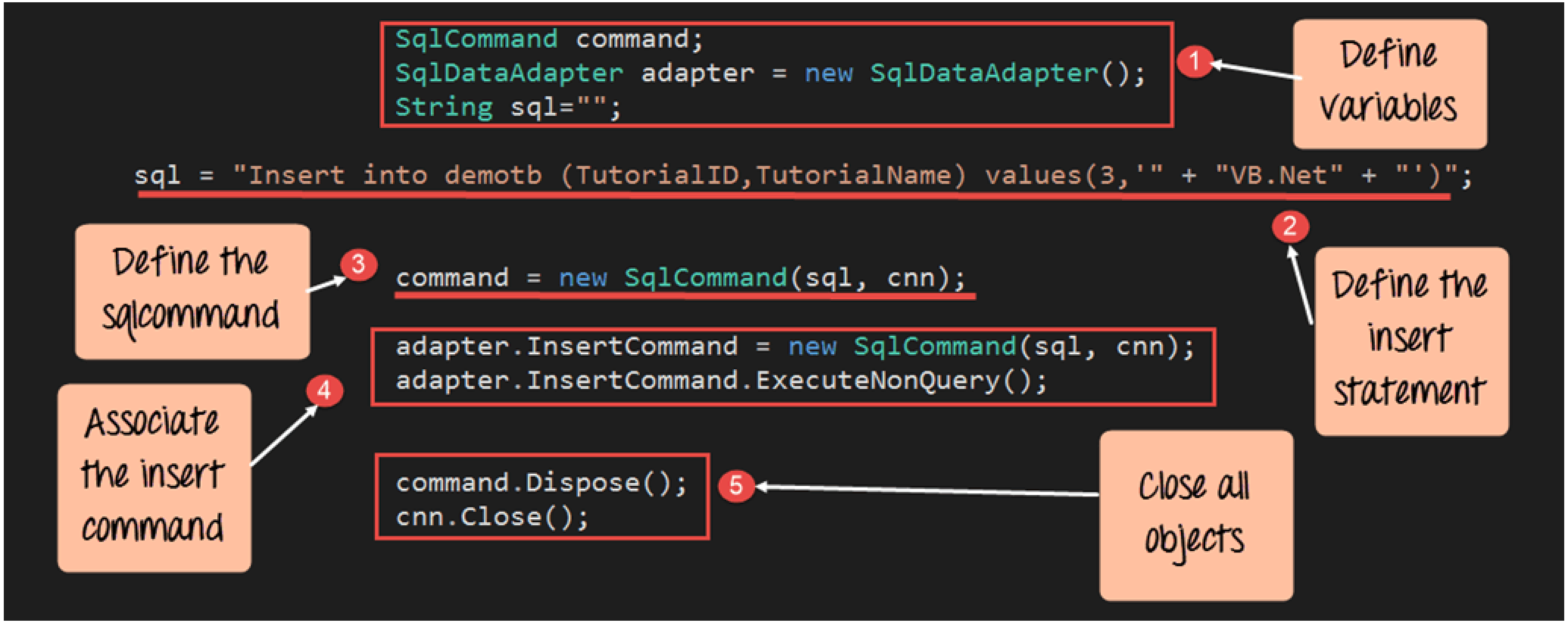


TutorialID	TutorialName
1	C#
2	ASP.Net

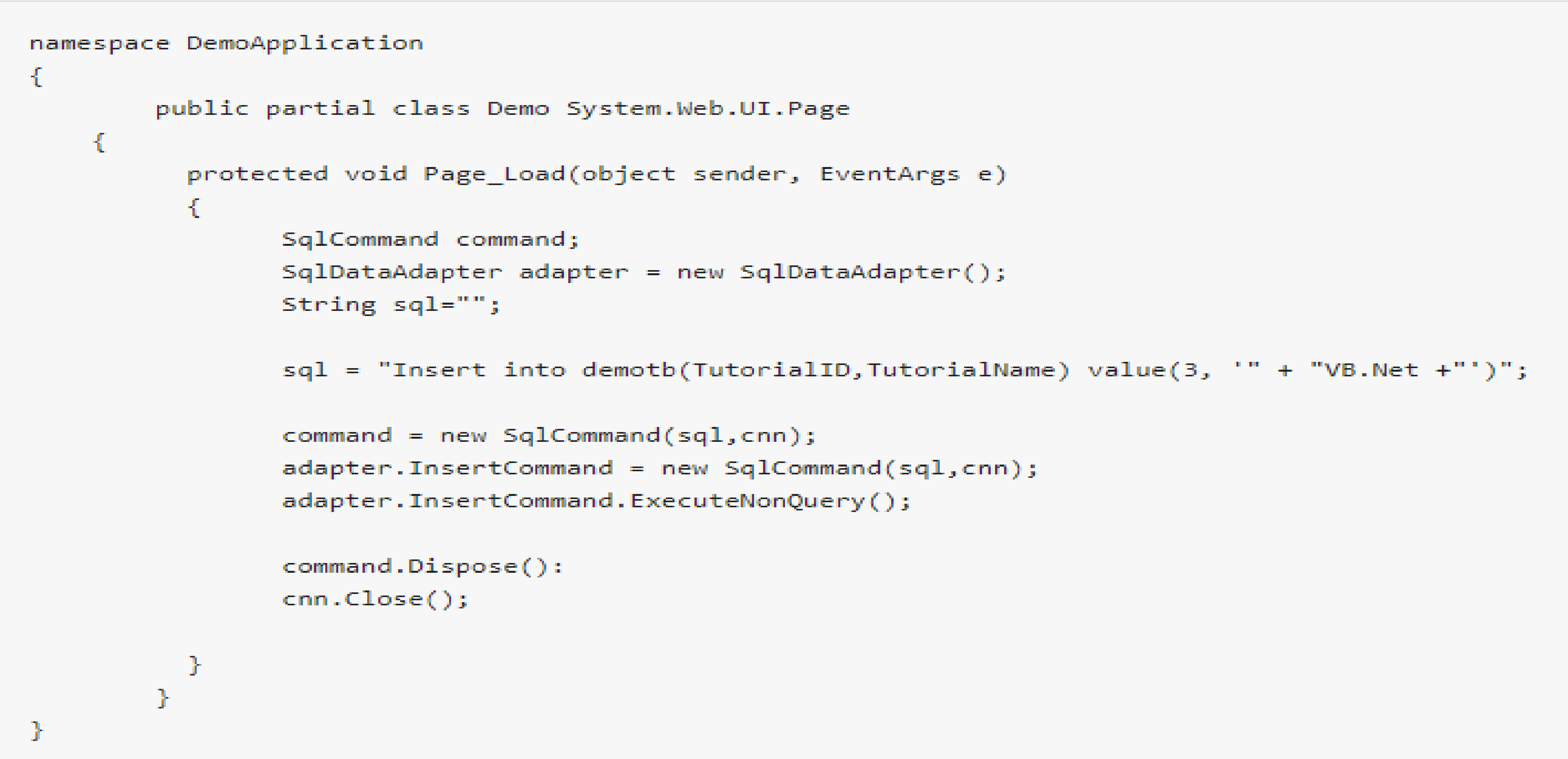


TutorialID	TutorialName
3	VB.Net

- **Step 1)** As the first step let's add the following code to our program. The below code snippet will be used to insert an existing record in our database.

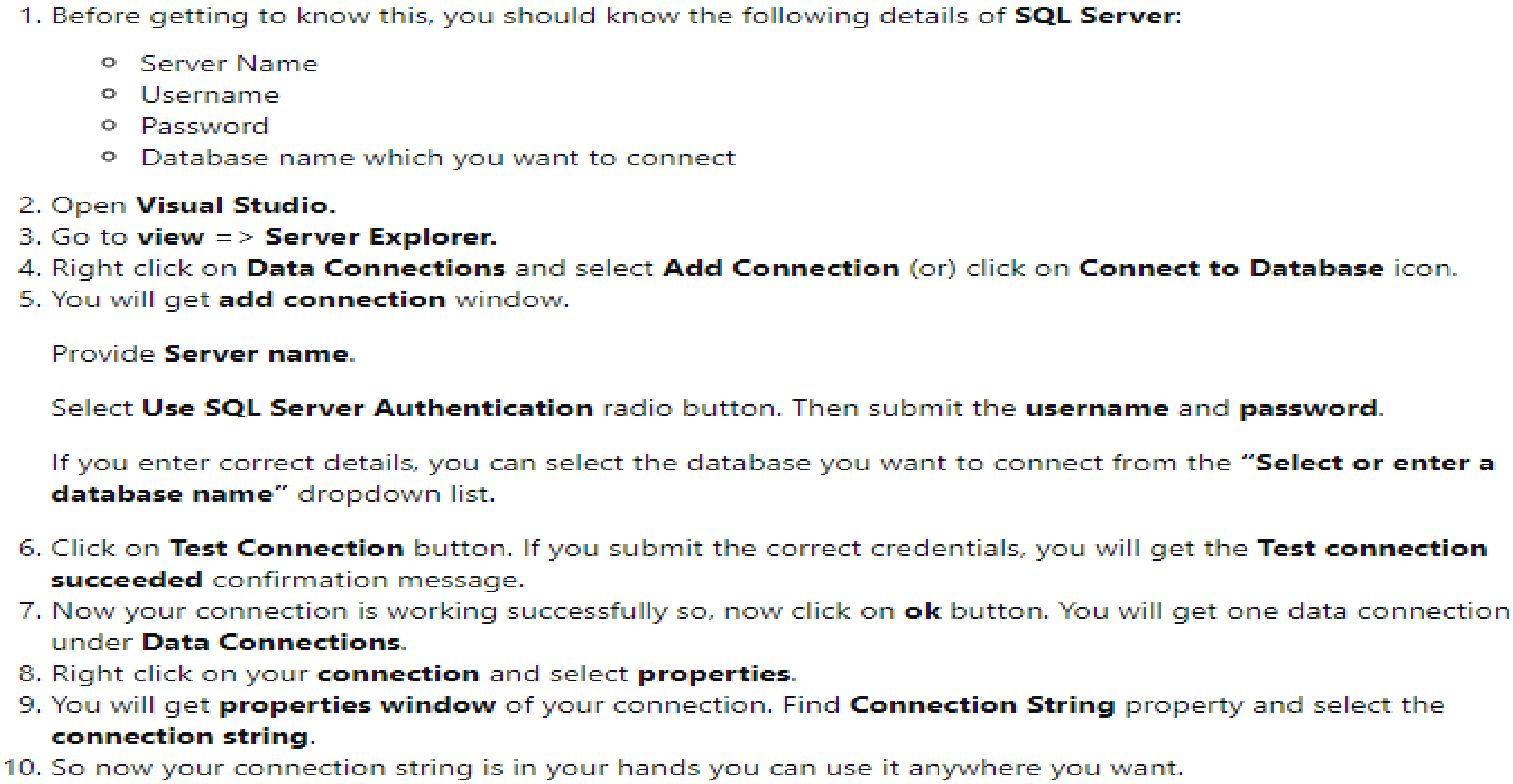


```
SqlCommand command;
SqlDataAdapter adapter = new SqlDataAdapter();
String sql="";

sql = "Insert into demotb (TutorialID,TutorialName) values(3,'" + "VB.Net" + "')";

command = new SqlCommand(sql, cnn);

adapter.InsertCommand = new SqlCommand(sql, cnn);
adapter.InsertCommand.ExecuteNonQuery();

command.Dispose();
cnn.Close();
```



```
namespace DemoApplication
{
        public partial class Demo System.Web.UI.Page
    {
          protected void Page_Load(object sender, EventArgs e)
          {
                SqlCommand command;
                SqlDataAdapter adapter = new SqlDataAdapter();
                String sql="";

                sql = "Insert into demotb(TutorialID,TutorialName) value(3, '" + "VB.Net +"')";

                command = new SqlCommand(sql,cnn);
                adapter.InsertCommand = new SqlCommand(sql,cnn);
                adapter.InsertCommand.ExecuteNonQuery();

                command.Dispose():
                cnn.Close();

          }
        }
}
```

How to get the connection string



1. Before getting to know this, you should know the following details of **SQL Server**:
   - Server Name
   - Username
   - Password
   - Database name which you want to connect
2. Open **Visual Studio.**
3. Go to **view** => **Server Explorer.**
4. Right click on **Data Connections** and select **Add Connection** (or) click on **Connect to Database** icon.
5. You will get **add connection** window.

   Provide **Server name**.

   Select **Use SQL Server Authentication** radio button. Then submit the **username** and **password**.

   If you enter correct details, you can select the database you want to connect from the **"Select or enter a database name"** dropdown list.
6. Click on **Test Connection** button. If you submit the correct credentials, you will get the **Test connection succeeded** confirmation message.
7. Now your connection is working successfully so, now click on **ok** button. You will get one data connection under **Data Connections**.
8. Right click on your **connection** and select **properties**.
9. You will get **properties window** of your connection. Find **Connection String** property and select the **connection string**.
10. So now your connection string is in your hands you can use it anywhere you want.