GURU99

C# Database Connection: How to connect SQL Server (Example)

By : Barbara Thompson Updated August 17, 2023

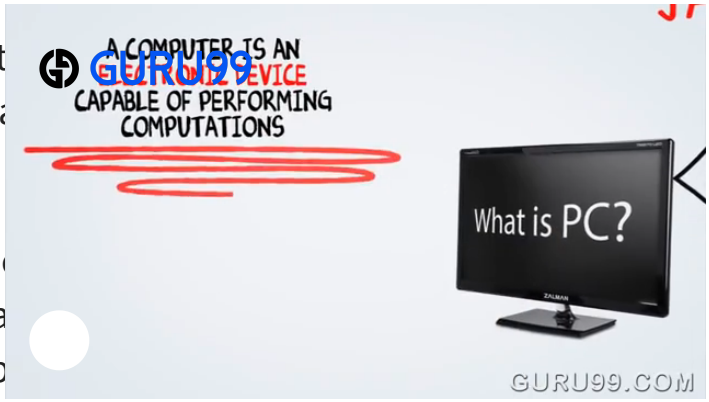
Accessing Data from a database is one of the important aspects of any programming language. It is an absolute necessity for any programming language to have the ability to work with databases. C# is no different.

It can work with different types of databases. It can work with the most common databases such as Oracle and Microsoft SQL Server.

It also can work with new forms of databases such as MongoDB and MySQL.

Table of Content:

Fundamentals of Database connectivity



C# and .Net can work with a majority of dat
and Microsoft SQL Server. But with every da
all of them is mostly the same.

In our examples, we will look at working th
For learning purposes, one can download a
Edition, which is a free database software p

In working with databases, the following are the concepts which are common to all databases.

1. **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.

    1. **Database name or Data Source** – The first important parameter is the database name to which the connection needs to be established. Each connection can only work with one database at a time.
    2. **Credentials** – The next important aspect is the username and password which needs to be used to establish a connection to the database. It ensures that the username and password have the necessary privileges to connect to the database.
    3. **Optional parameters** – For each database type, you can specify optional parameters to provide more information on how .net should handle the connection to the database. For example, one can specify a parameter for how long the
    operation is performed for a
    parameter would determine i

2. **Selecting data from the database** – On
the next important aspect is to fetch th
execute 'SQL' select command against
be used to fetch data from a specific table in the database.
3. **Inserting data into the database** – C# can also be used to insert records into the database. Values can be specified in C# for each row that needs to be inserted into the database.

4. **Updating data into the database** – C# can also be used to update existing records into the database. New values can be specified in C# for each row that needs to be updated into the database.
5. **Deleting data from a database** – C# can also be used to delete records into the database. Select commands to specify which rows need to be deleted can be specified in C#.

Ok, now that we have seen the theory of each operation, let's jump into the further sections to look at how we can perform database operations in C#.

SQL Command in c#

SqlCommand in C# allow the user to query and send the commands to the database. SQL command is specified by the SQL connection object. Two methods are used, ExecuteReader method for results of query and ExecuteNonQuery for insert, Update, and delete commands. It is the method that is best for the different commands.

How to connect C# to Database

Let's now look at the code, which needs to be kept in place to create a connection to a database. In our example, we will connect to a database which has the name of Demodb. The credentials used to connect to the database are given below

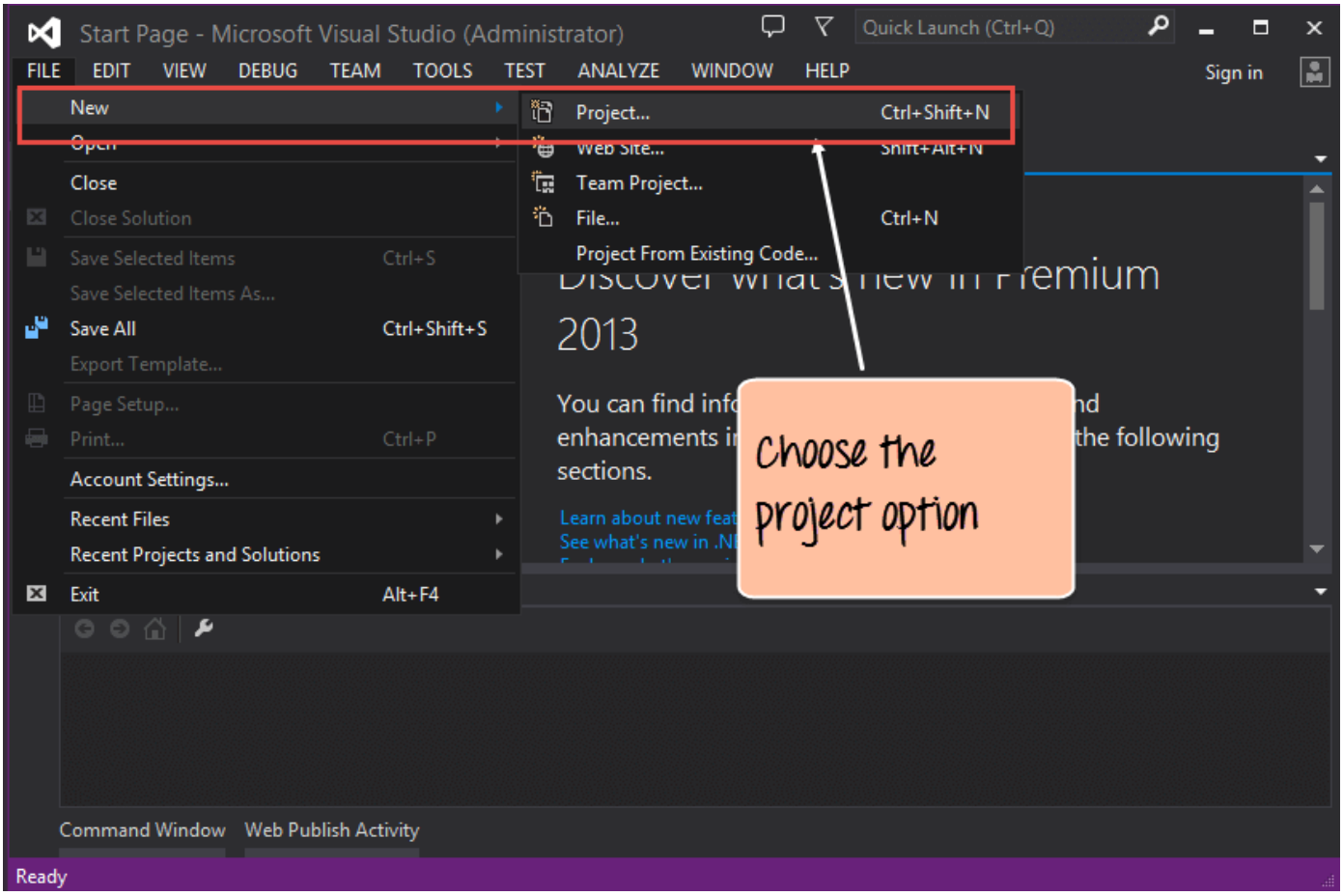
- **Username – sa**
- **Password – demo123**

GURU99

We will see a simple Windows forms applic
have a simple button called "Connect" whi
database.

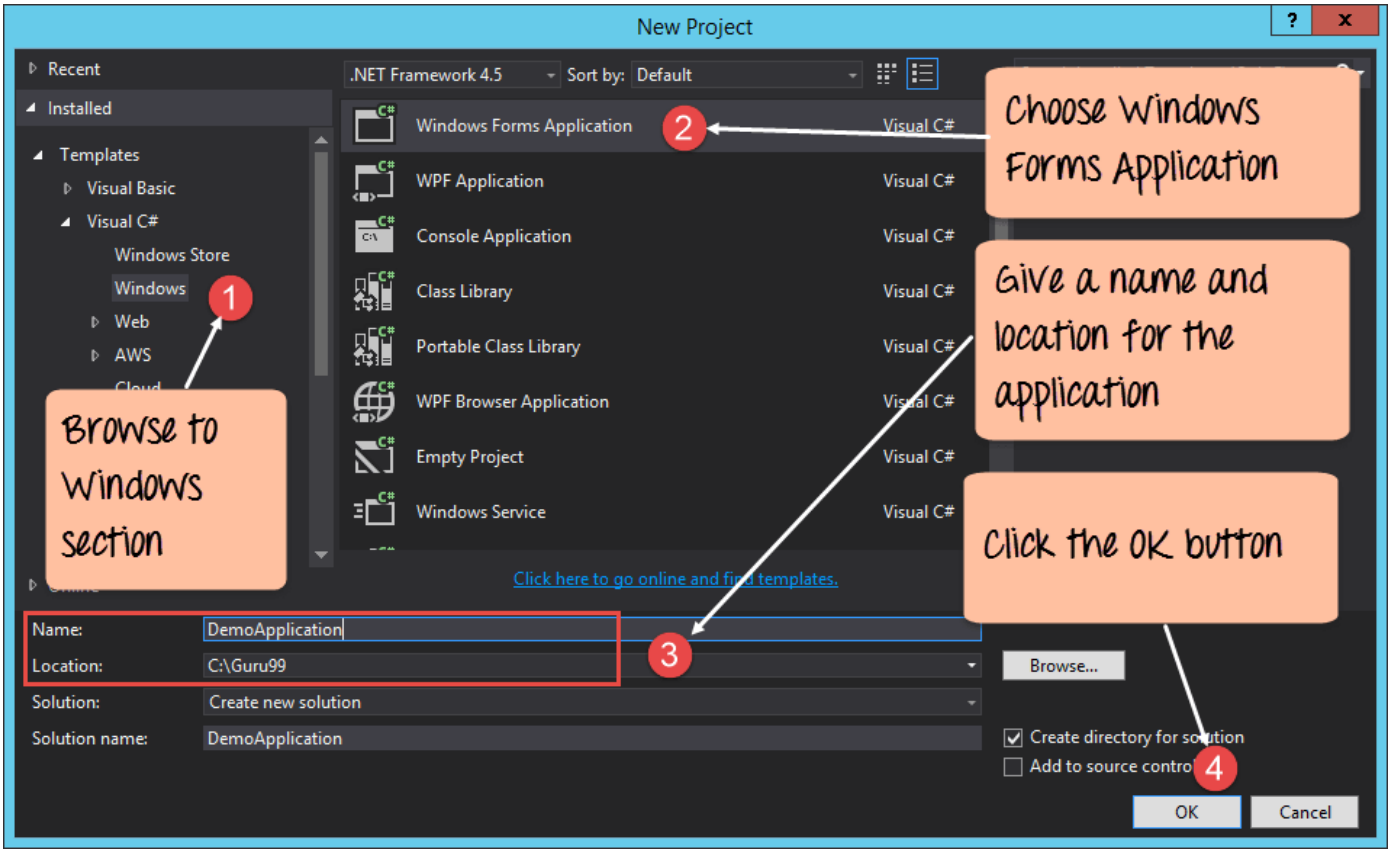
So let's follow the below steps to achieve th

Step 1) The first step involves the creation of a new project in Visual Studio. After launching Visual Studio, you need to choose the menu option New->Project.



Step 2) The next step is to choose the project type as a Windows Forms application. Here, we also need to mention the name and location of our project.

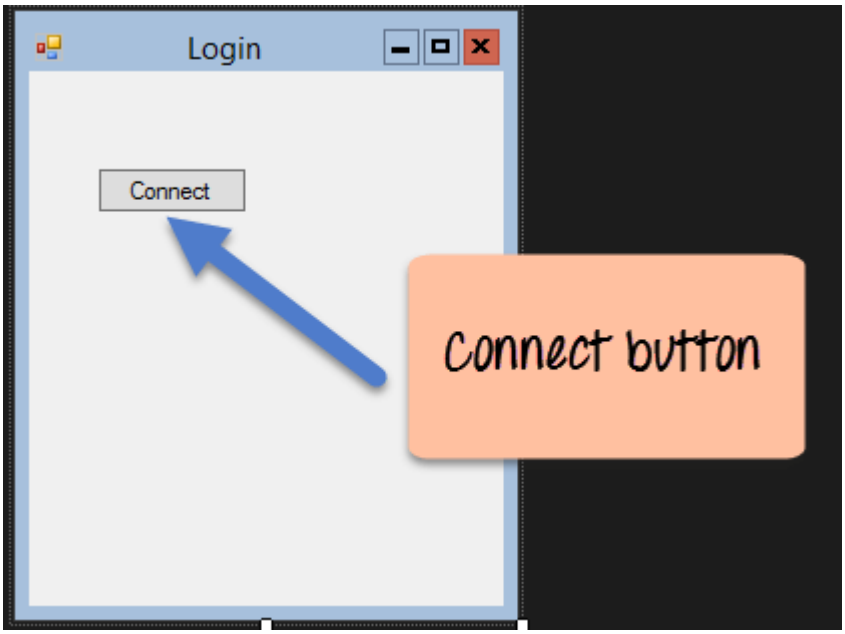
GURU99



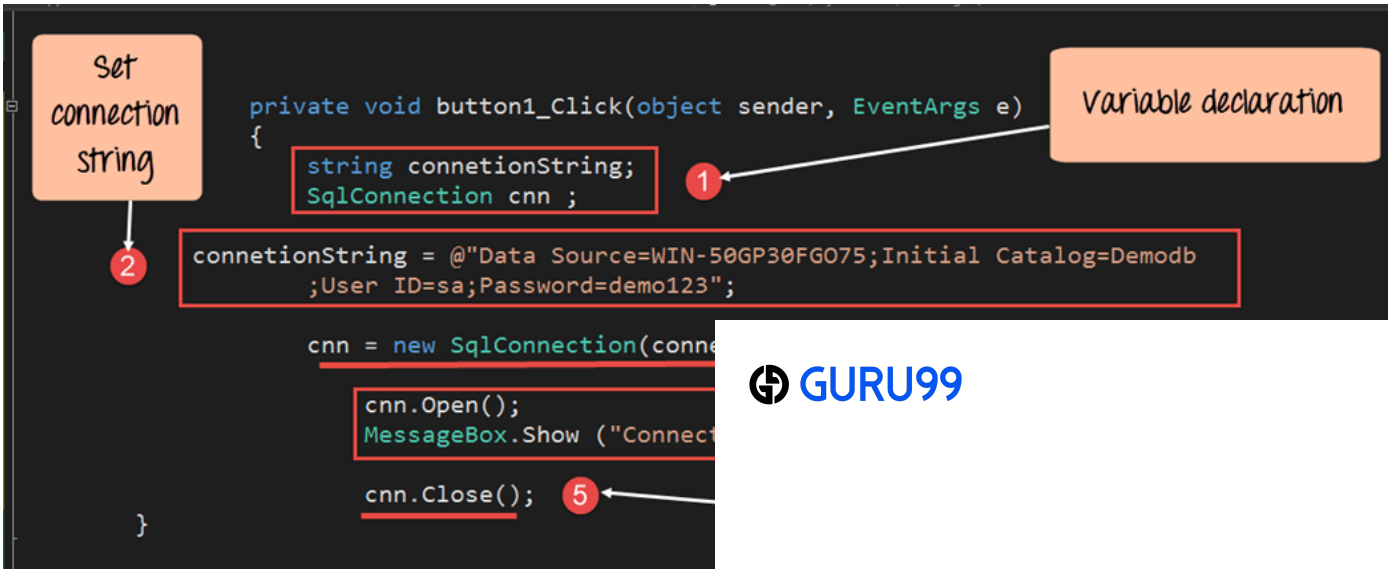
1. In the project dialog box, we can see various options for creating different types of projects in Visual Studio. Click the Windows option on the left-hand side.
2. When we click the Windows options in the previous step, we will be able to see an option for Windows Forms Application. Click this option.
3. We then give a name for the application which in our case is "DemoApplication". We also need to provide a location to store our application.
4. Finally, we click the 'OK' button to let \

GURU99

Step 3) Now add a button from the toolbox to the Windows form. Put the text property of the Button as Connect. This is how it will look like



Step 4) Now double click the form so that an event handler is added to the code for the button click event. In the event handler, add the below code.




```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
```

```
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DemoApplication1
{
 public partial class Form1 : Form
 {
  public Form1()
  {
   InitializeComponent();
  }

  private void button1_Click(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();
   MessageBox.Show("Connection Open  !");
   cnn.Close();
  }
 }
}
```

GURU99

Code Explanation:-

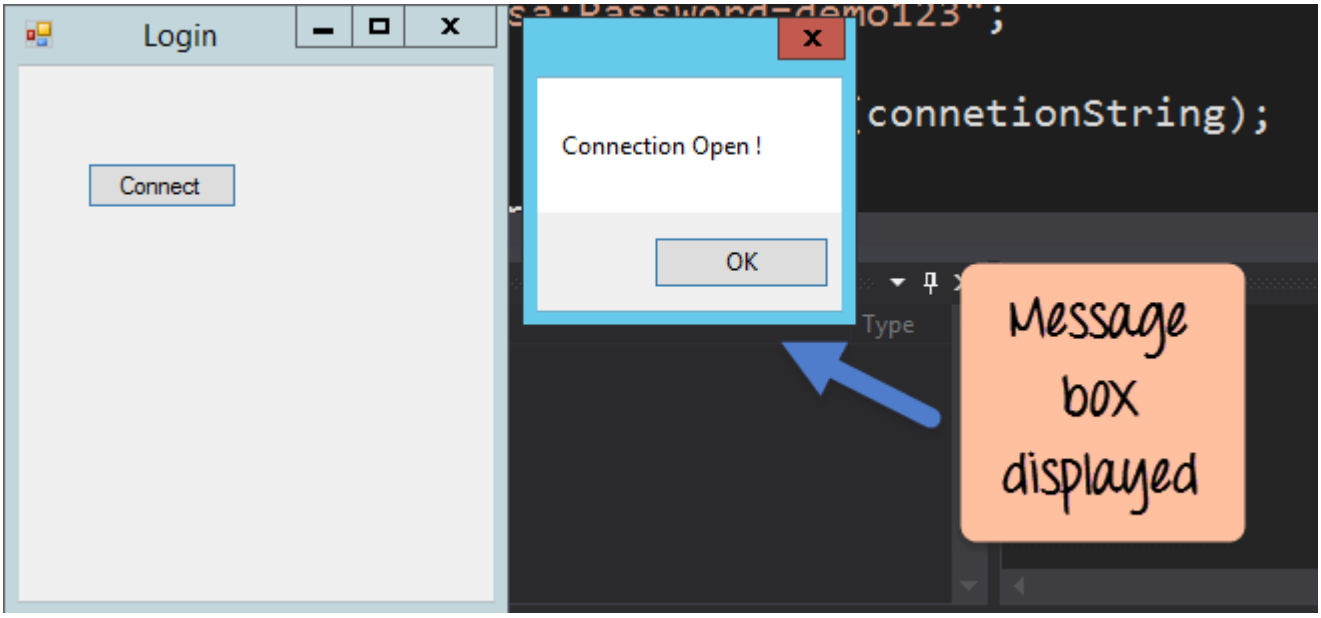
1. The first step is to create variables, wh
   string and the connection to the SQL S
2. The next step is to create the connecti
   be specified correctly for C# to understand the connection string. The
   connection string consists of the following parts

1. 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.
2. The Initial Catalog is used to specify the name of the database
3. The UserID and Password are the credentials required to connect to the database.

3. Next, we assign the connecting string to the variable cnn. The variable cnn, which is of type SqlConnection is used to establish the connection to the database.
4. Next, we use the Open method of the cnn variable to open a connection to the database. We then just display a message to the user that the connection is established.
5. Once the operation is completed successfully, we then close the connection to the database. It is always a good practice to close the connection to the database if nothing else is required to be done on the database.

When the above code is set, and the project is executed using Visual Studio, you will get the below output. Once the form is displayed, click the Connect button.

Output:-



When you click on “connect” button, from the output, you can see that the database connection was established. Hence, the message box was displayed.

Access data with the SqlDataReader

To showcase how data can be accessed using C#, let us assume that we have the following artifacts in our database.

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

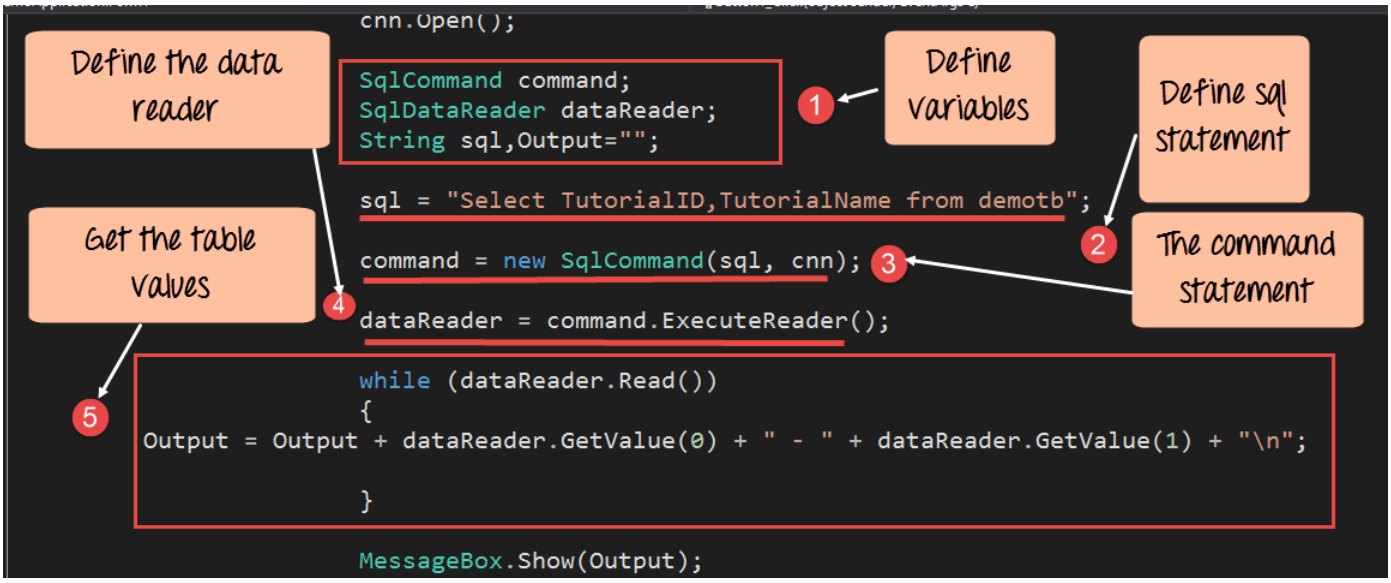
GURU99

TutorialID	Tutori
1	C#
2	ASP.N

Let’s change the code in our form, so that we can query for this data and display the information via a Messagebox. Note that all the code entered below is a continuation of the code written for the data connection in the previous section.

Step 1) Let's split the code into 2 parts so that it will be easy to understand for the user.

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



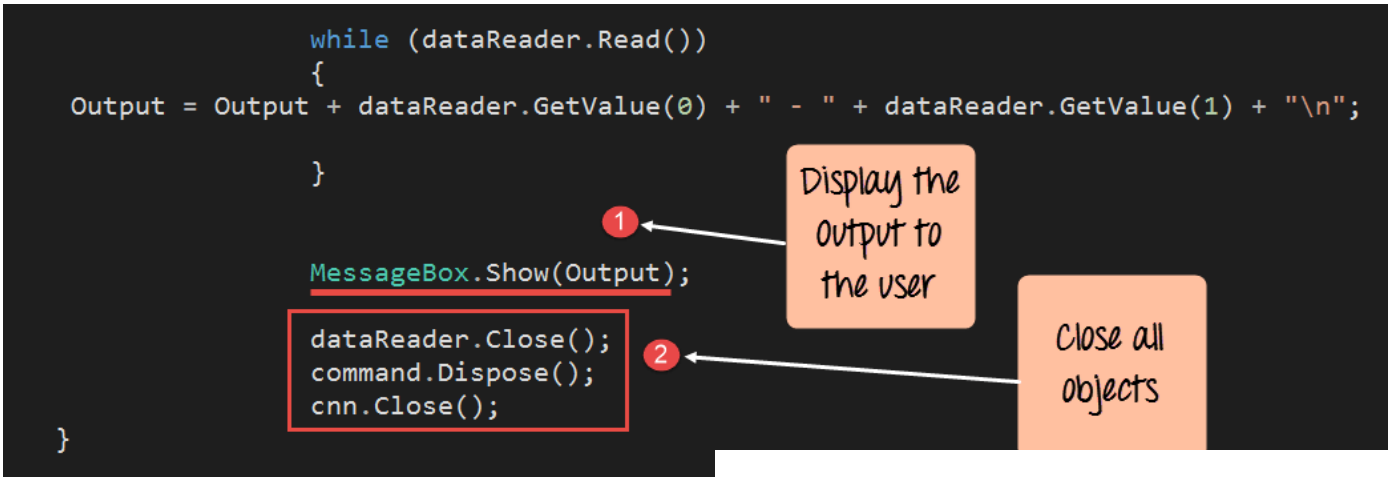
Code Explanation:-

1. The first step is to create the following variables

    1. 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 firs
    2. The DataReader object is use SQL query. We can then read data reader.
    3. We then define 2 string varia command string. The next is the "Output" which will contain all the table values.

2. The next step is to define the SQL statement, which 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.
3. 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.
4. Next, we will execute the data reader command, which will fetch all the rows from the demotb table.
5. 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 and close all the objects related to the database operation.

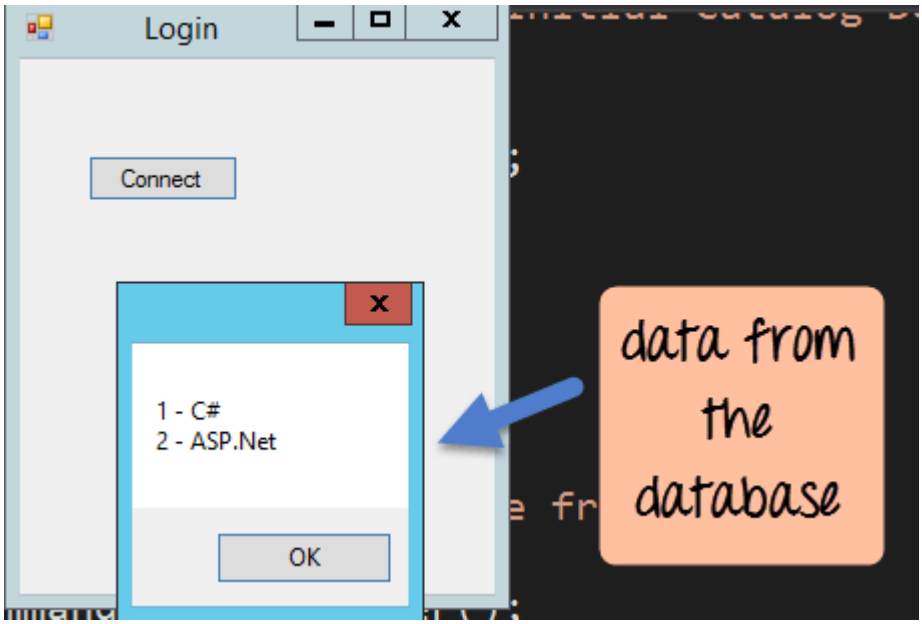


Code Explanation:-

1. We will continue our code by displayin the MessageBox. The Output variable demotb table.
2. We finally close all the objects related this is always a good practice.

When the above code is set, and the project is run using Visual Studio, you will get the below output. Once the form is displayed, click the Connect button.

Output:-



From the output, you can clearly see that the program was able to get the values from the database. The data is then displayed in the message box.

C# Insert Into Database

Just like Accessing data, C# has the ability to insert records into the database as well. To showcase how to insert records into our database, let's take the same table structure which was used above.

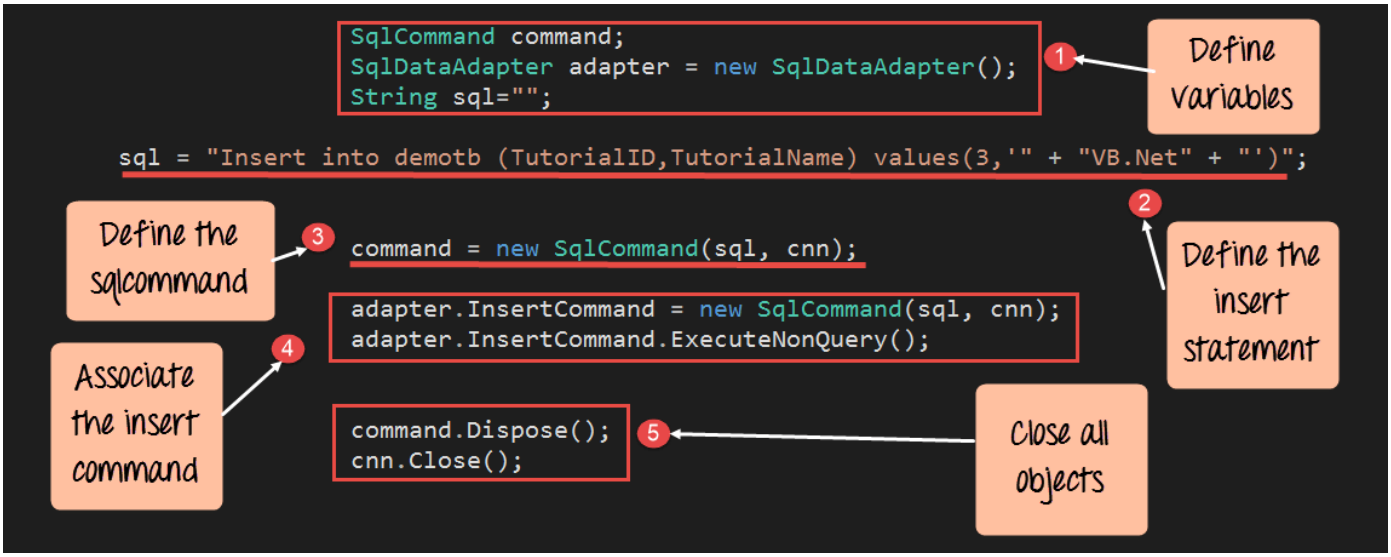
TutorialID	TutorialName
1	C#
2	ASP.N

GURU99

Let's change the code in our form, so that w
table

TutorialID	Tutorialname
3	VB.Net

So let's add the following code to our program. The below code snippet will be used to insert an existing record in our database.



Code Explanation:-

1. The first step is to create the following variables

    1. SQLCommand – This data type is used to define objects which are used to perform SQL operations against a database. This object will hold the SQL command which will run against our SQL Server database.
    2. The DataAdapter object is used to perform specific SQL operations such as insert, delete and update commands.
    3. We then define a string variable, which is "SQL" to hold our SQL command string.

GURU99

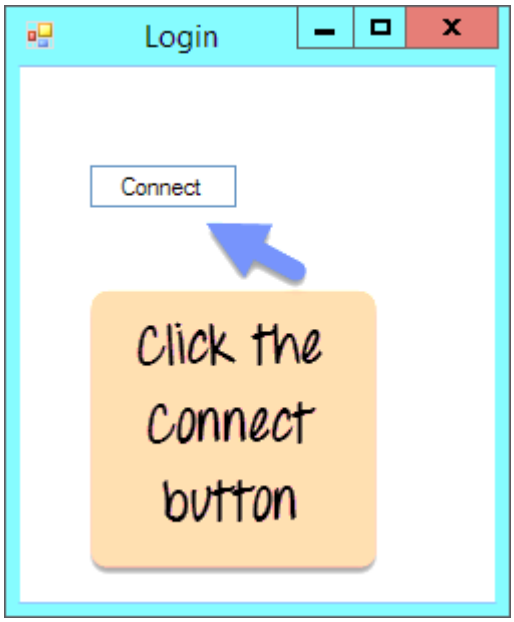
2. The next step is to actually define the S against our database. In our case, we a will insert the record of TutorialID=1 ar
3. Next, we create the command object w statement against the database. In the connection object and the SQL string
4. In our data adapter command, we now associate the insert SQL command to our adapter. We also then issue the ExecuteNonQuery method which is used to execute the Insert statement against our database. The 'ExecuteNonQuery'

method is used in C# to issue any DML statements against the database. By DML statements, we mean the insert, delete, and update operation. In C# , if you want to issue any of these statements against a table, you need to use the ExecuteNonQuery method.

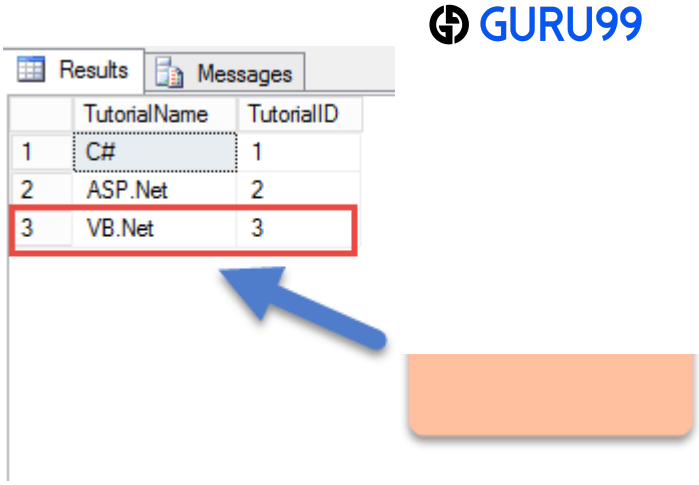
5. We finally close all the objects related to our database operation. Remember this is always a good practice.

When the above code is set, and the project is executed using Visual Studio, you will get the below output. Once the form is displayed, click the Connect button.

Output:-



If you go to SQL Server Express and see the rows in the demotb table, you will see the row inserted as shown below



C# Update Database

Just like Accessing data, C# has the ability to update existing records from the database as well. To showcase how to update records into our database, let's take the same table structure which was used above.

TutorialID	TutorialName
1	C#
2	ASP.Net
3	VB.Net

Let's change the code in our form, so that we can update the following row. The old row value is TutorialID as "3" and Tutorial Name as "VB.Net". Which we will update it to "VB.Net complete" while the row value for Tutorial ID will remain same.

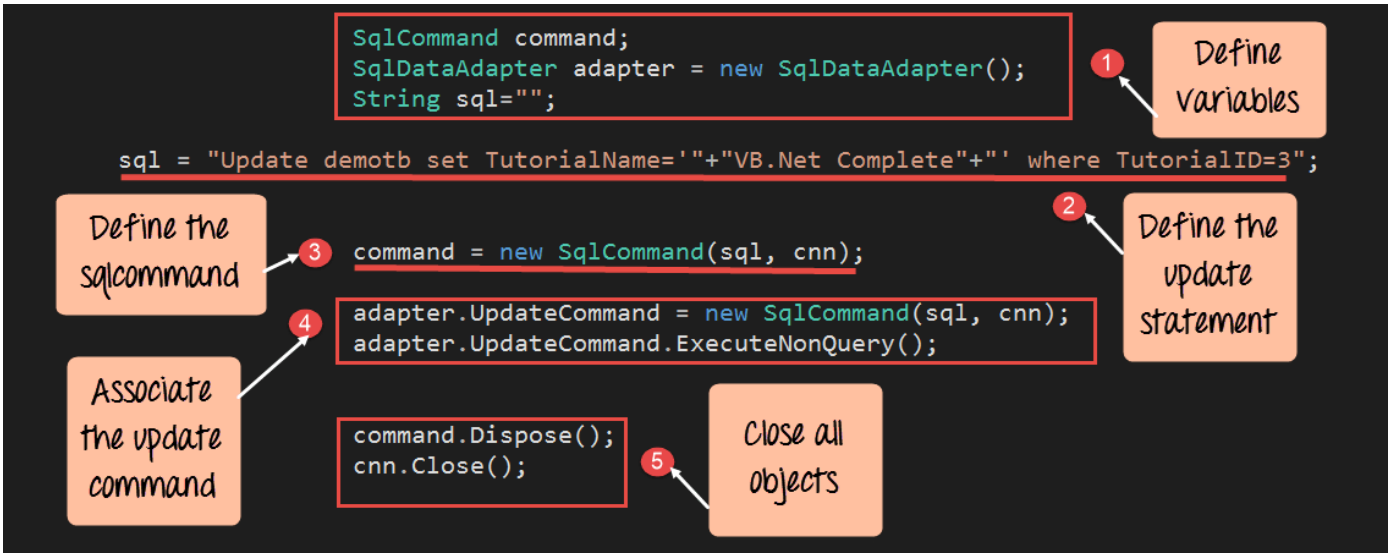
Old row

TutorialID	TutorialName
3	VB.Net

New row

TutorialID	TutorialN
3	VB.Net co

So let's add the following code to our progr
used to update an existing record in our da



C# SqlCommand Example With Code Explanation:-

1. The first step is to create the following variables

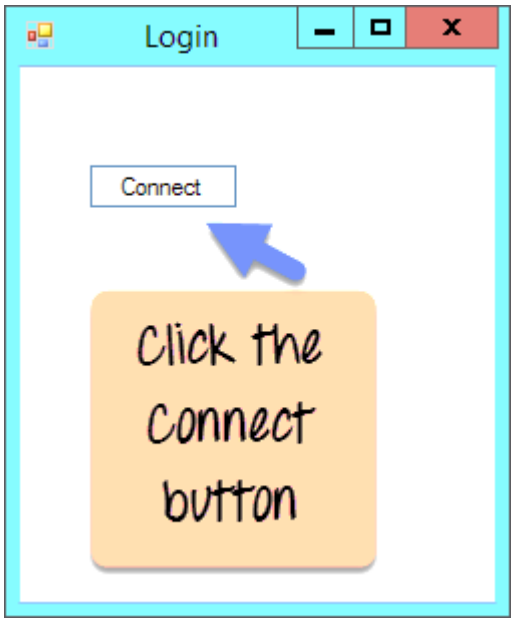
    1. SQLCommand – This data type is used to define objects which are used to perform SQL operations against a database. This object will hold the SQL command which will run against our SQL Server database.
    2. The dataadapter object is used to perform specific SQL operations such as insert, delete and update commands.
    3. We then define a string variable, which is SQL to hold our SQL command string.

2. The next step is to define the SQL statement which will be used against our database. In our case we are issuing ar Tutorial name to “VB.Net Complete” w kept as 3.
3. Next, we will create the command obje statement against the database. In the connection object and the SQL string.
4. In our data adapter command, we nov our adapter. We also then issue the ExecuteNonQuery method which is used to execute the Update statement against our database.
5. We finally close all the objects related to our database operation. Remember this is always a good practice.

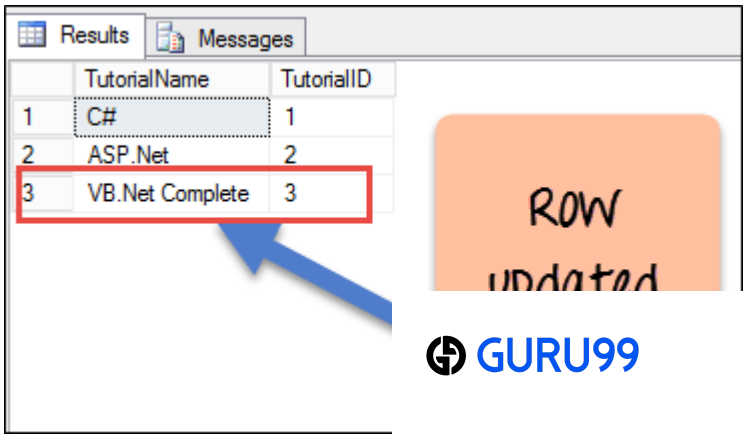
GURU99

When the above code is set, and the project is executed using Visual Studio, you will get the below output. Once the form is displayed, click the Connect button.

Output:-



If you actually go to SQL Server Express and see the rows in the demotb table, you will see the row was successfully updated as shown below.



Deleting Records

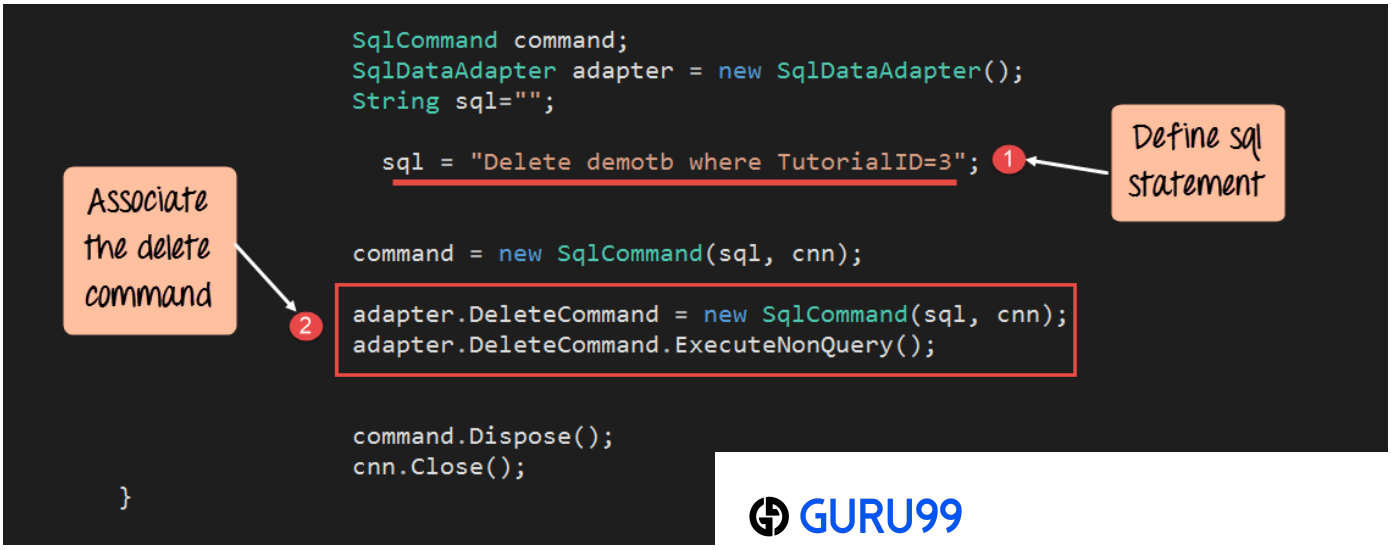
Just like Accessing data, C# has the ability t
database as well. To showcase how to delet
the same table structure which was used above.

TutorialID	TutorialName
1	C#
2	ASP.Net
3	VB.Net complete

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

TutorialID	TutorialName
3	VB.Net complete

So let's add the following code to our program. The below code snippet will be used to delete an existing record in our database.

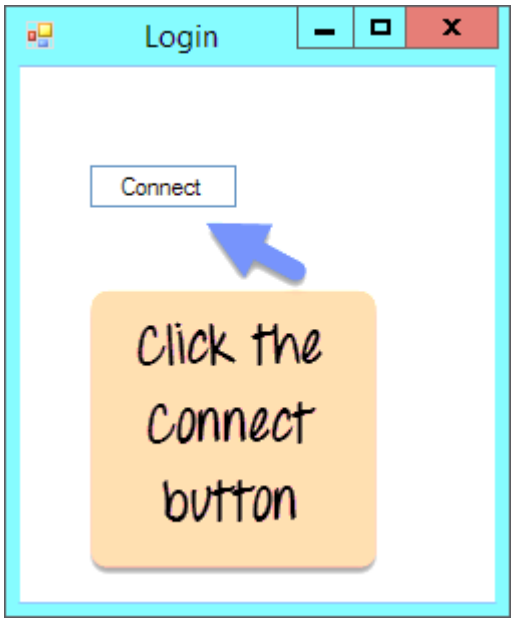


Code Explanation:-

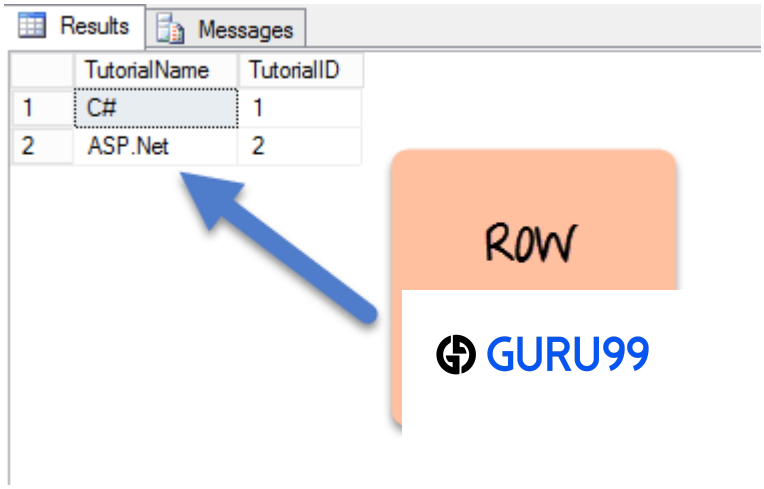
1. The Key difference in this code is that \
   statement. The delete statement is use
   in which the TutorialID has a value of 3.
2. In our data adapter command, we now associate the insert SQL command to our adapter. We also then issue the ExecuteNonQuery method which is used to execute the Delete statement against our database.

When the above code is set, and the project is executed using Visual Studio, you will get the below output. Once the form is displayed, click the Connect button.

Output:-



If you actually go to SQL Server Express and see the rows in the demotb table, you will see the row was successfully deleted as shown below.



Connecting Controls to Data

In the earlier sections, we have seen how tc
SQLCommand and SQLReader to fetch data from a database. We also saw how we read each row of the table and use a messagebox to display the contents of a table to the user.

But obviously, users don't want to see data sent via message boxes and would want better controls to display the data. Let's take the below data structure in a table

TutorialID	TutorialName
1	C#
2	ASP.Net
3	VB.Net complete

From the above data structure, the user would ideally want to see the TutorialID and Tutorial Name displayed in a textbox. Secondly, they might want to have some sort of button control which could allow them to go to the next record or to the previous record in the table. This would require a bit of extra coding from the developer's end.

The good news is that C# can reduce the additional coding effort by allowing binding of controls to data. What this means is that C# can automatically populate the value of the textbox as per a particular field of the table.

So, you can have 2 textboxes in a windows form. You can then link one text box to the TutorialID field and another textbox to the TutorialName field. This linking is done in the Visual Studio designer itself, and you don't need to write extra code for this.

Visual Studio will ensure that it writes the c
Then when you run your application, the te
connect to the database, fetch the data anc
coding is required from the developer's enc

GURU99

Let's look at a code example of how we can

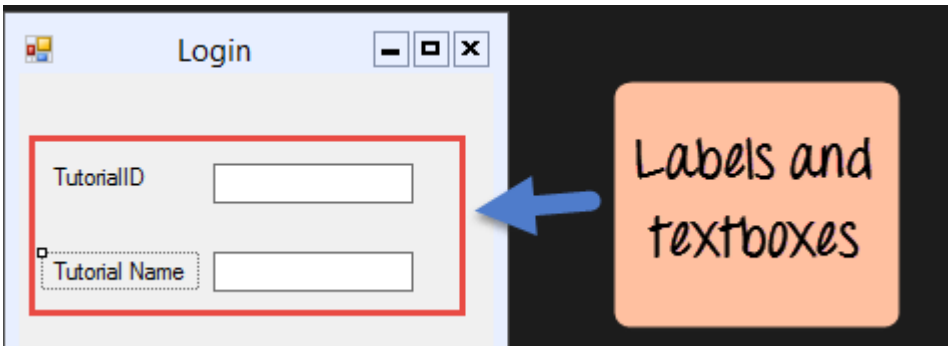
In our example, we are going to create 2 tex
going to represent the Tutorial ID and Tutorial Name respectively. They will be bound to the Tutorial ID and TutorialName fields of the database accordingly.

Let's follow the below-mentioned steps to achieve this.

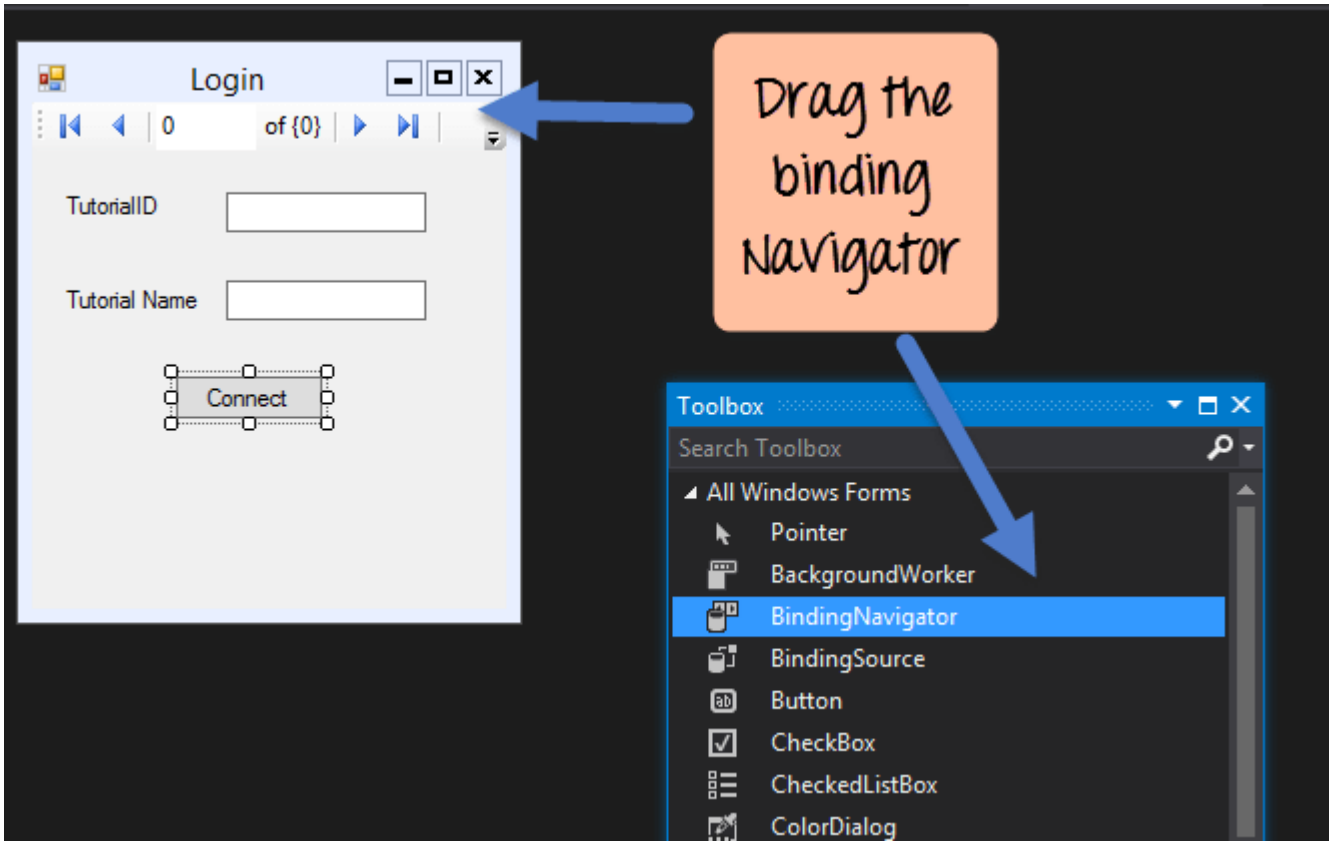
Step 1) Construct the basic form. In the form drag and drop 2 components- labels and textboxes. Then carry out the following substeps

1. Put the text value of the first label as TutorialID
2. Put the text value of the second label as TutorialName
3. Put the name property of the first textbox as txtID
4. Put the name property of the second textbox as txtName

Below is the how the form would look like once the above-mentioned steps are performed.



Step 2) The next step is to add a binding Navigator to the form. The binding Navigator control can automatically navigate through each row of the table. To add the binding navigator, just go to the toolbox and drag it to the form.

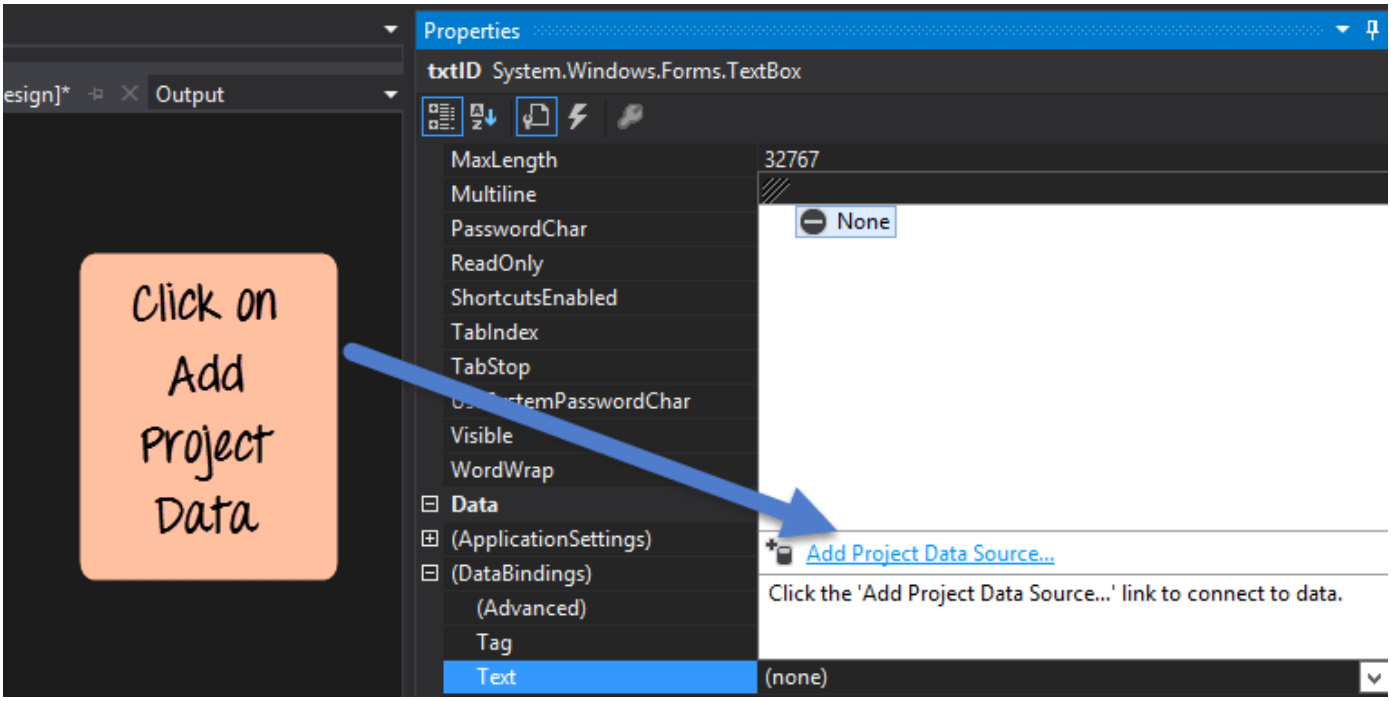


Step 3) The next step is to add a binding to our database. This can be done by going to any of the Textbox control and clicking on the DataBindings->Text property. The Binding Navigator is used to establish a link from your application to a database.

When you perform this step, Visual Studio will automatically add the required code to the application to make sure the application is linked to the database. Normally the database in Visual Studio is referred to as a Project Data Source. So to ensure the connection is established between the application and the database, the first step is to create a project data source.

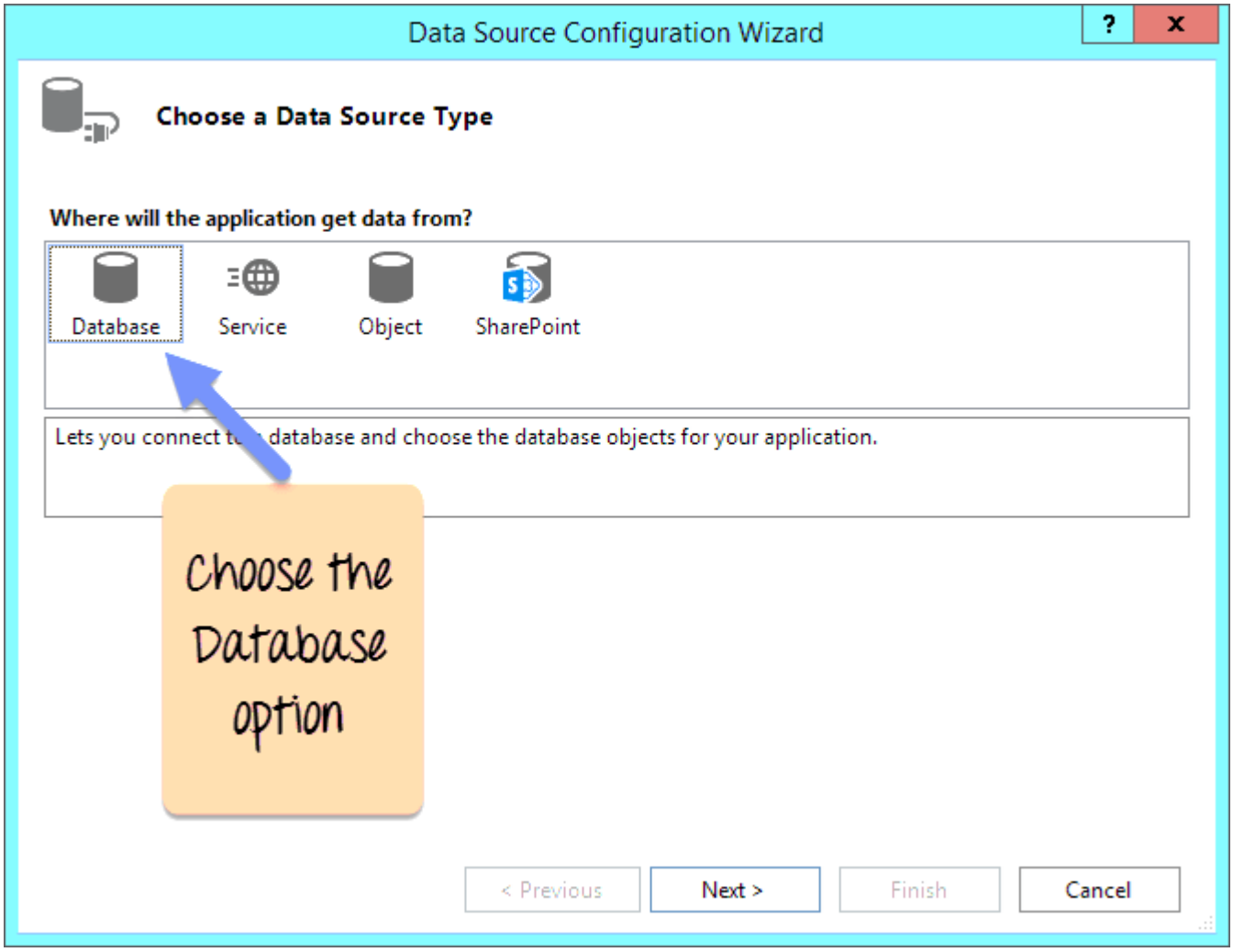
The following screen will show up. Click on
When you click on the project data source,
will allow you to define the database conne

GURU99



Step 4) Once you click on the Add Project Data Source link, you will be presented with a wizard which will be used to create a connection to the demotb database. The following steps show in detail what needs to be configured during each step of the wizard.

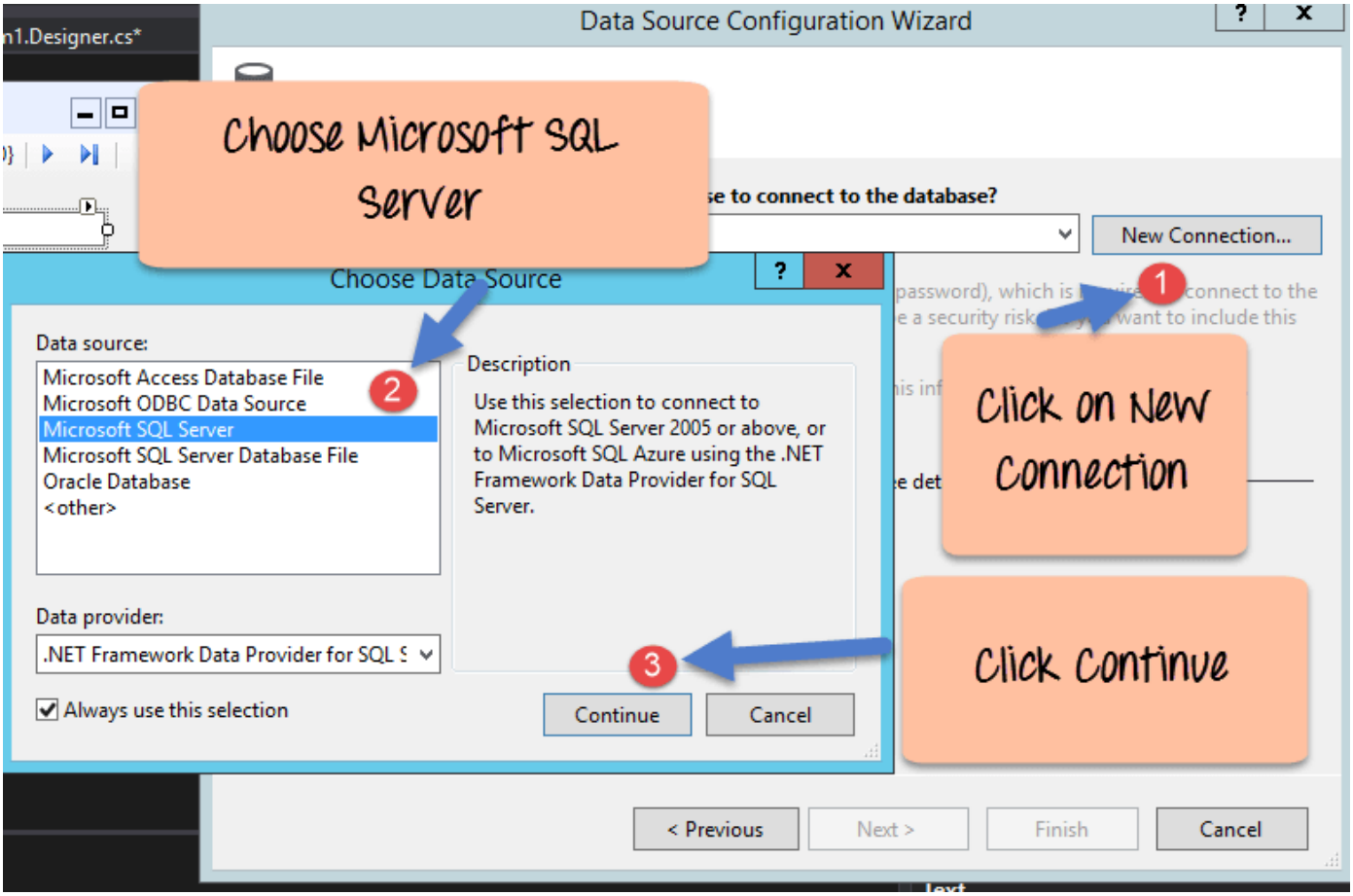
1. In the screen which pops up , choose the Data Source type as Database and then click on next button.



2. In the next screen, you need to start the creation of the connection string to the database. The connection string is required for the application to establish a connection to the database. It contains the parameters such as server name, database name, and the name of the driver.

    1. Click on the New connection butt
    2. Choose the Data Source as Micros
    3. Click the Continue button.

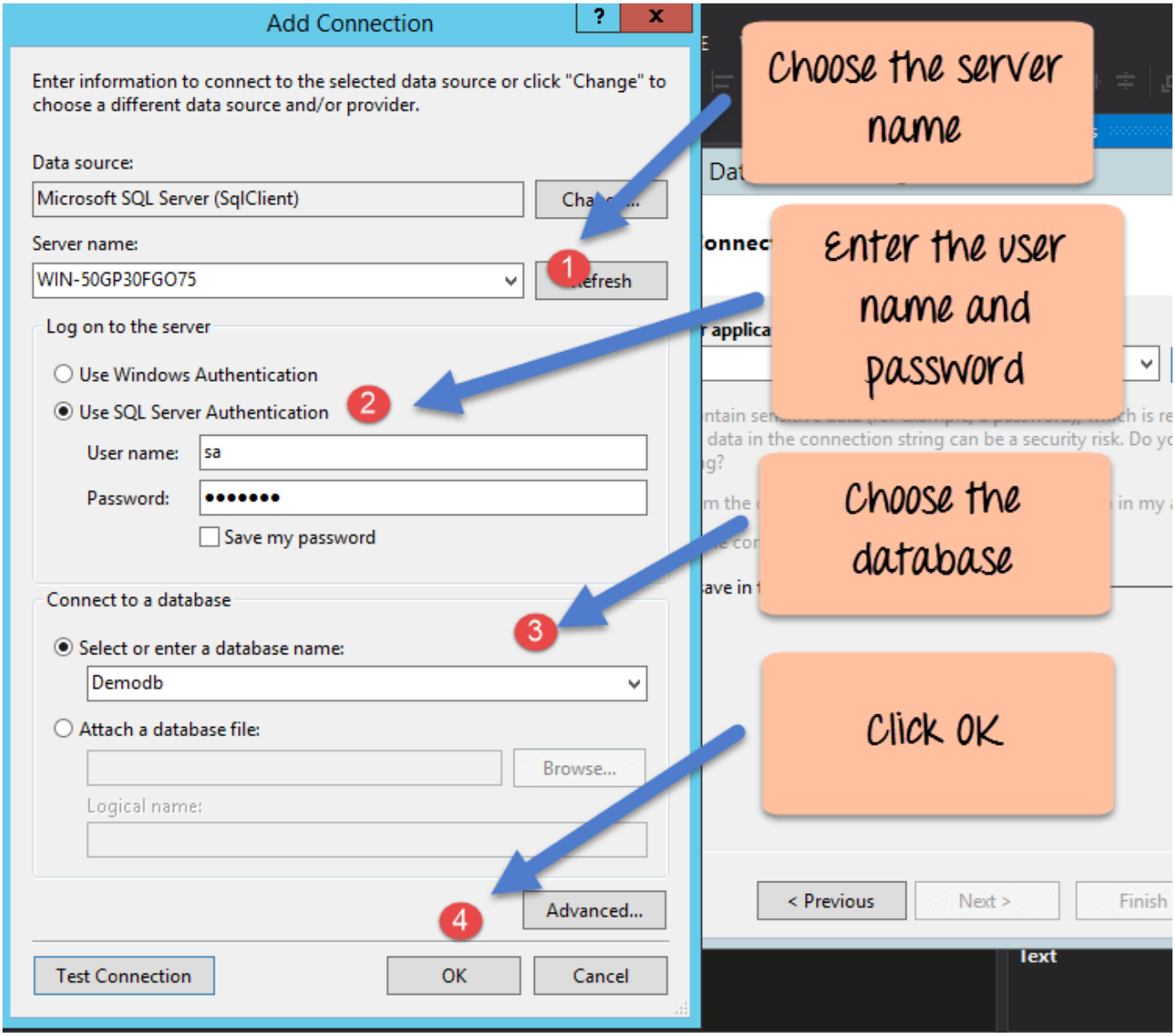
GURU99



3. Next, you need to add the credentials to connect to the database

    1. Choose the server name on which the SQL Server resides
    2. Enter the user id and password to connect to the database
    3. Choose the database as demotb
    4. Click the ‘ok’ button.

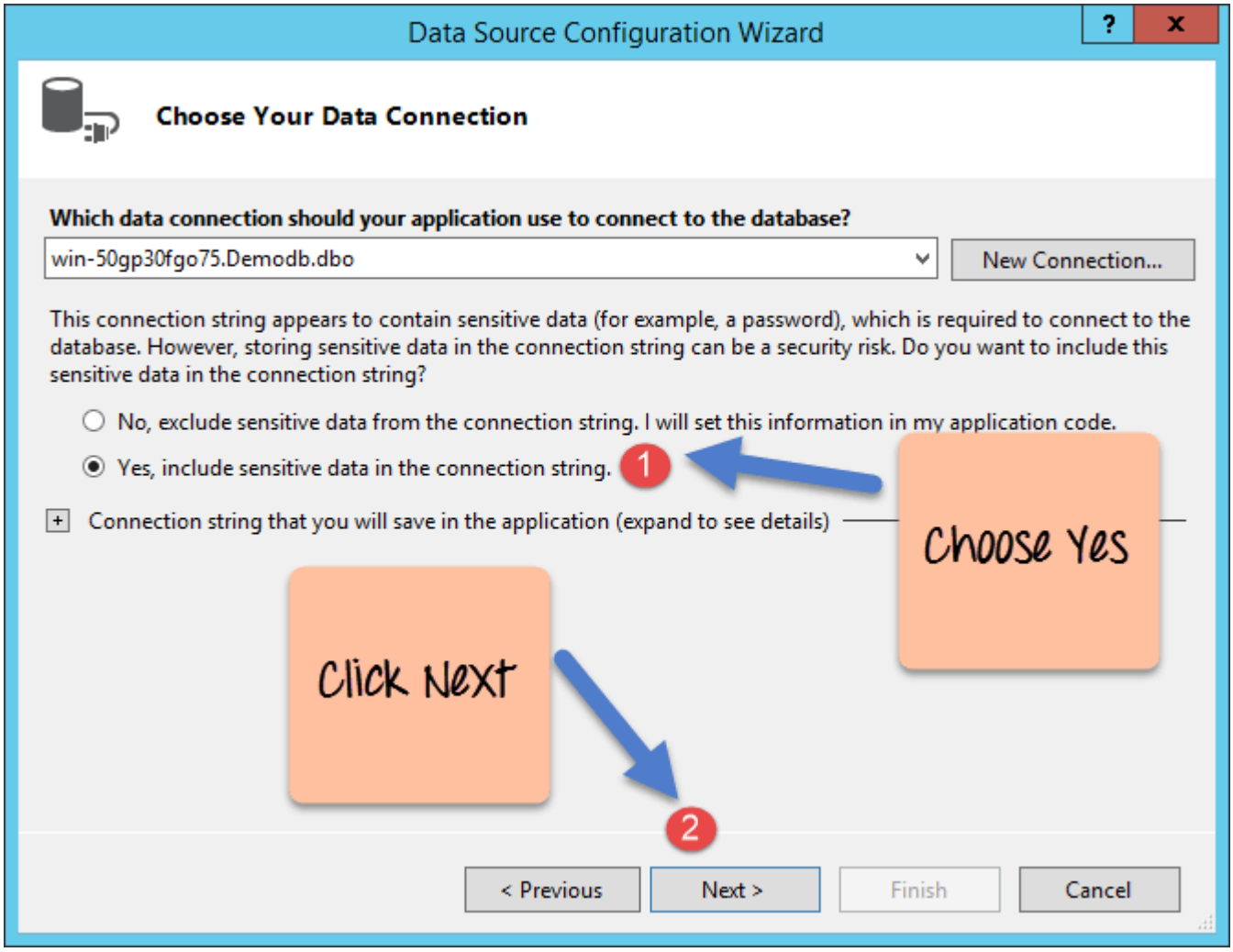
GURU99



4. In this screen, we will confirm all the settings which were carried on the previous screens.

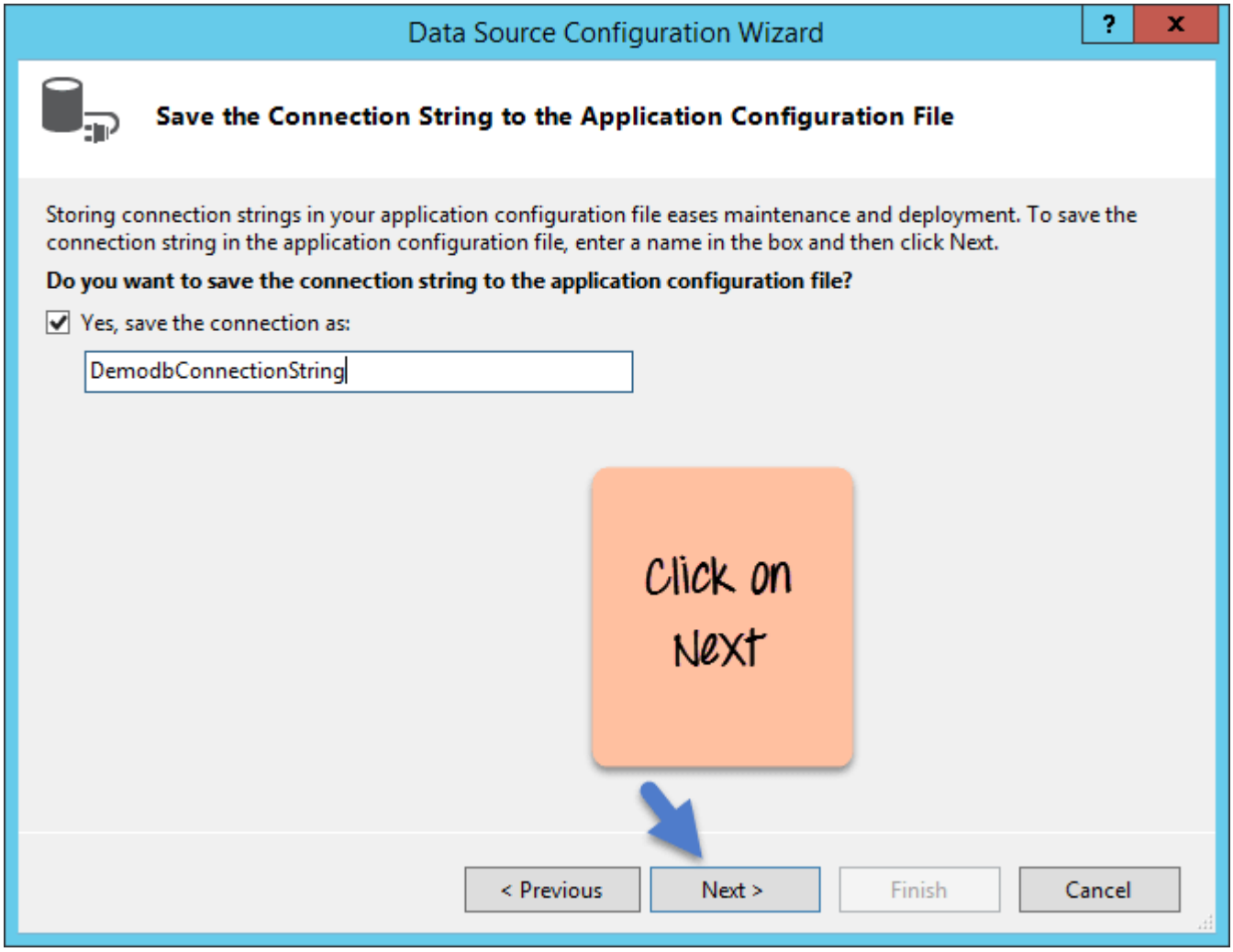
GURU99

1. Choose the option “Yes” to includ
2. Click on the “Next” button.



5. In the next screen, click on the “Next” button to confirm the creation of the connection string

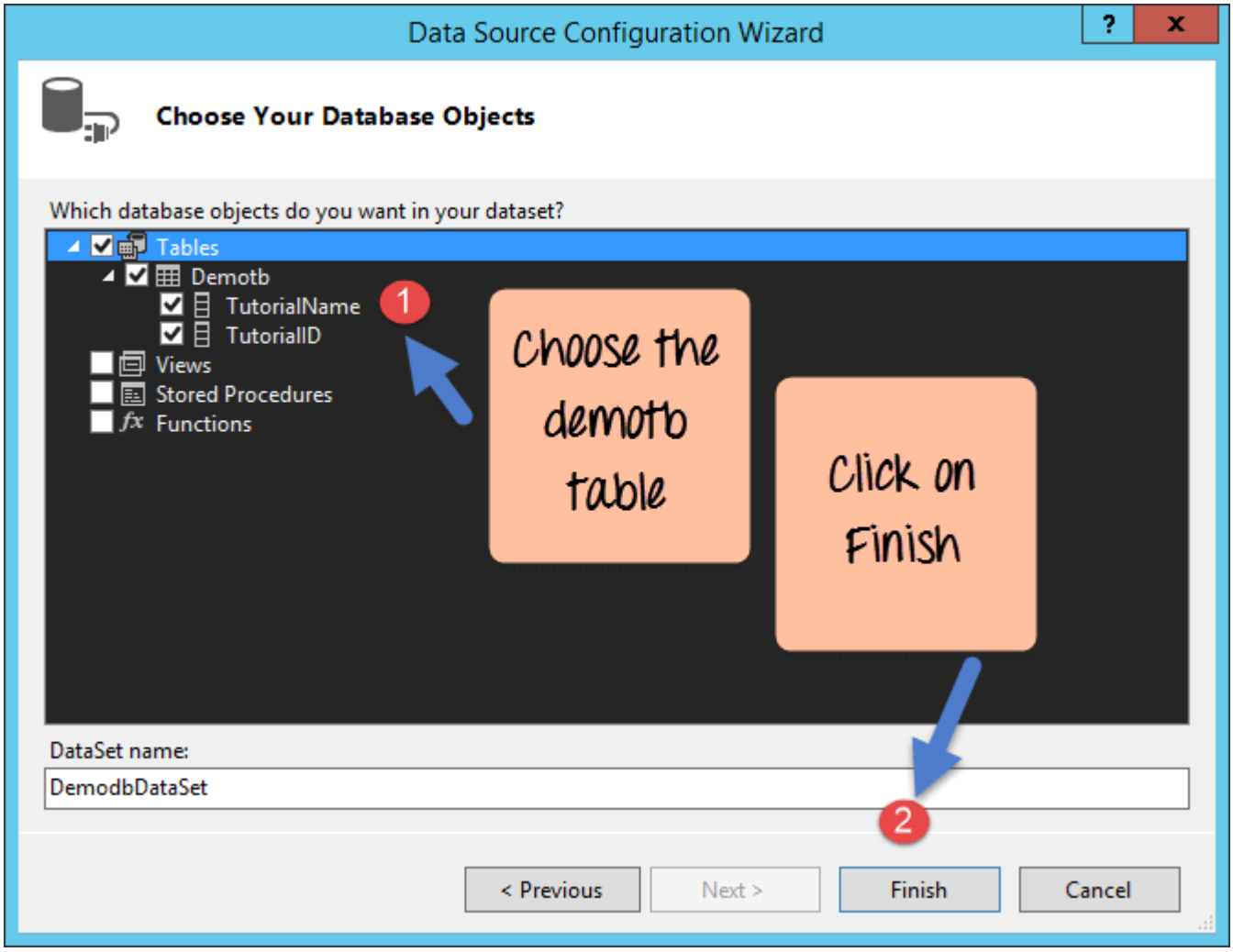
GURU99



6. In this step,

1. Choose the tables of Demotb, which will be shown in the next screen.
2. This table will now become an available data source in the C# project

GURU99



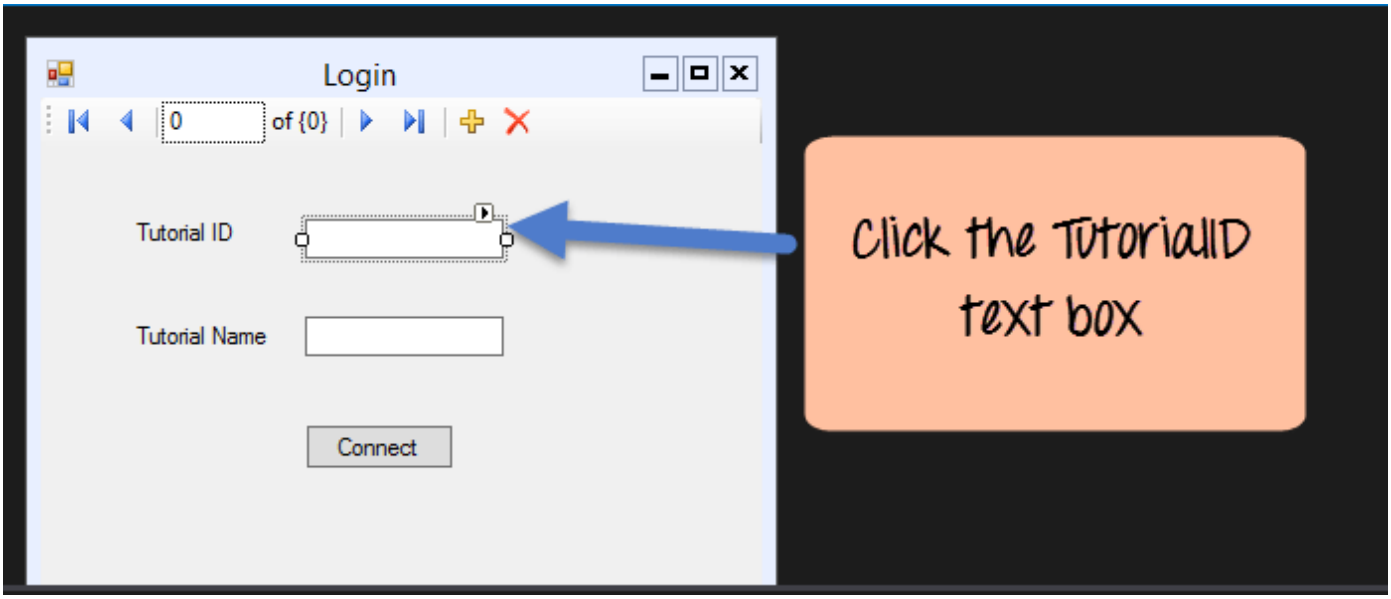
When you click the Finish button, Visual Studio will now ensure that the application can query all the rows in the table Demotb.

Step 5) Now that the data source is defined, we now need to connect the TutorialID and TutorialName textbox to the demotb table. When you click on the Text property of either the TutorialID or TutorialName tex
source to Demotb is available.

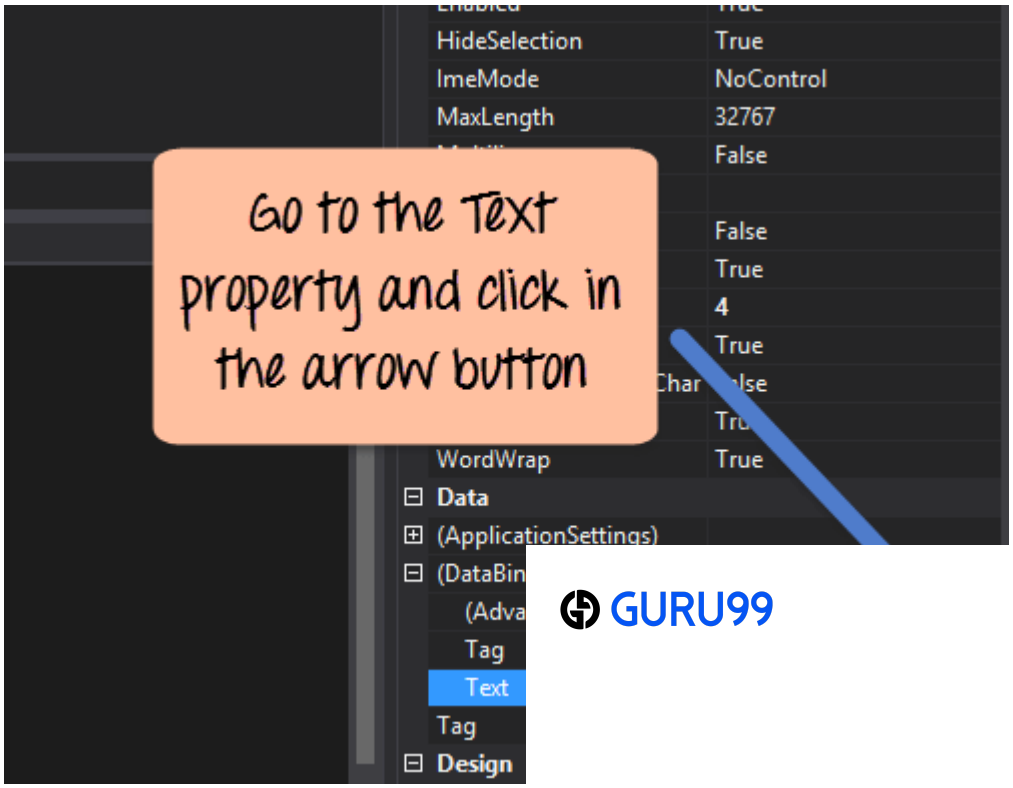
GURU99

For the first text box choose the Tutorial ID.
and choose the field as TutorialName. The
to each control and change the binding acc

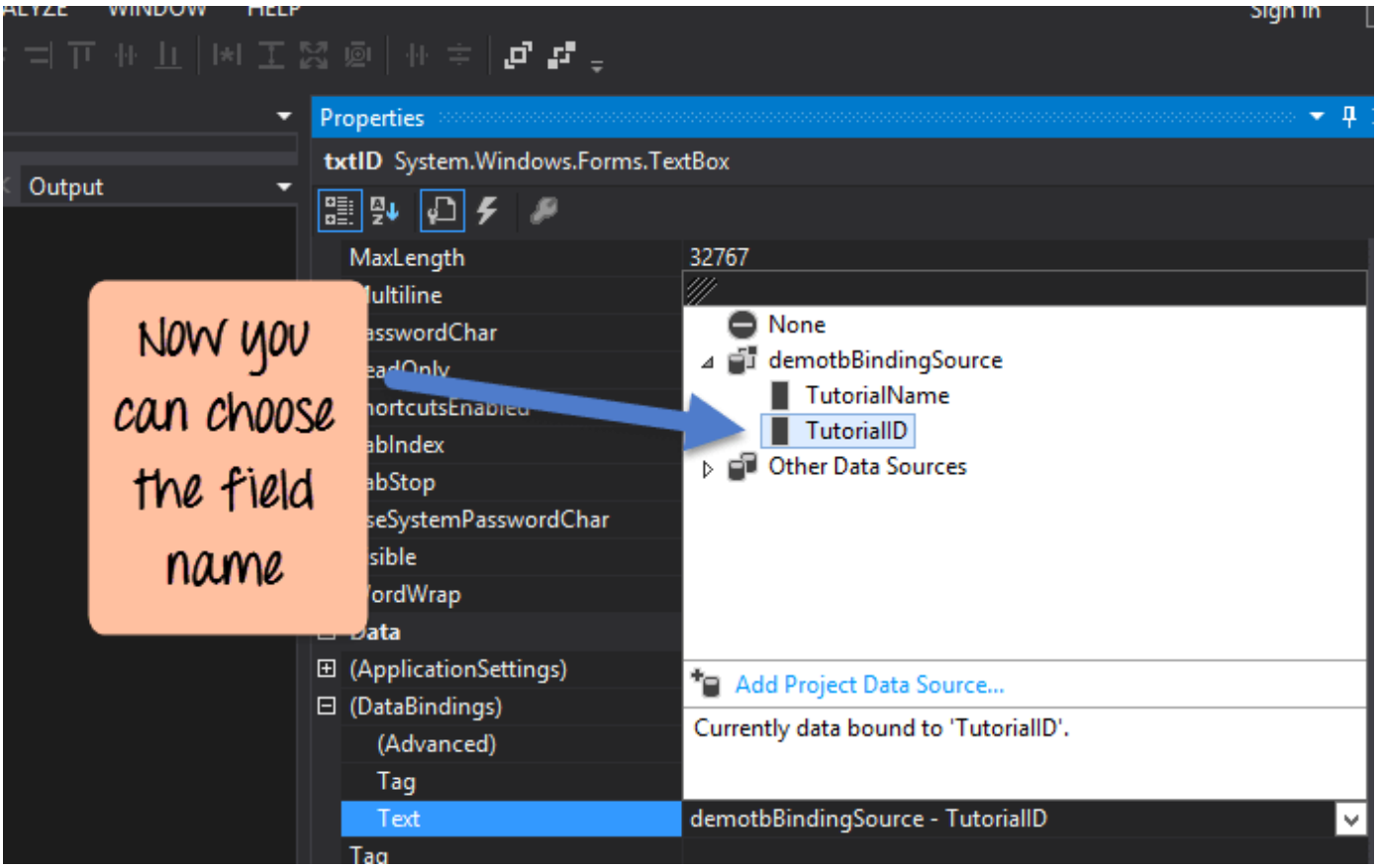
1. Click on the Tutorial ID control.



2. In the Properties window, you will see the properties of the TutorialID textbox. Go to the text property and click on the down arrow button.



3. When you click the down arrow buttor
Source option. And under this, you wil
TutorialID. Choose the Tutorial ID one.



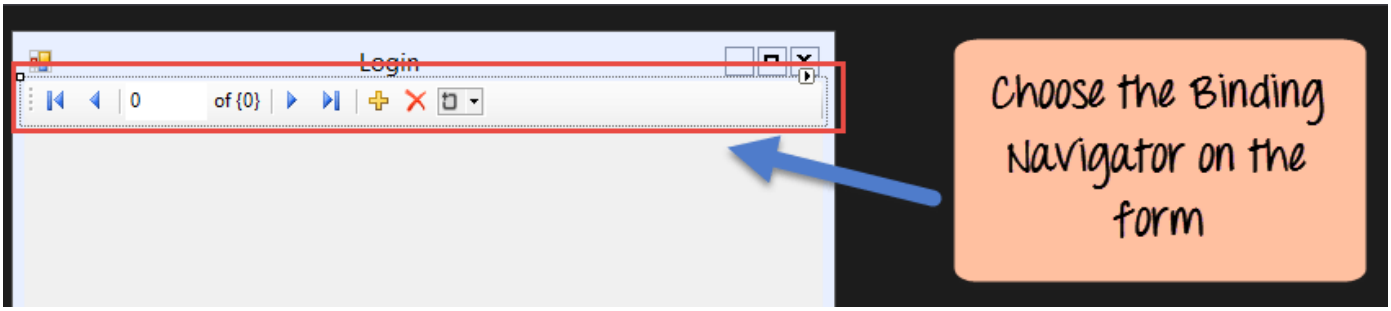
Repeat the above 3 steps for the Tutorial Name text box.

1. So click on the Tutorial Name text box
2. Go to the properties window
3. Choose the Text property
4. Choose the TutorialName option under demotbBindingSource

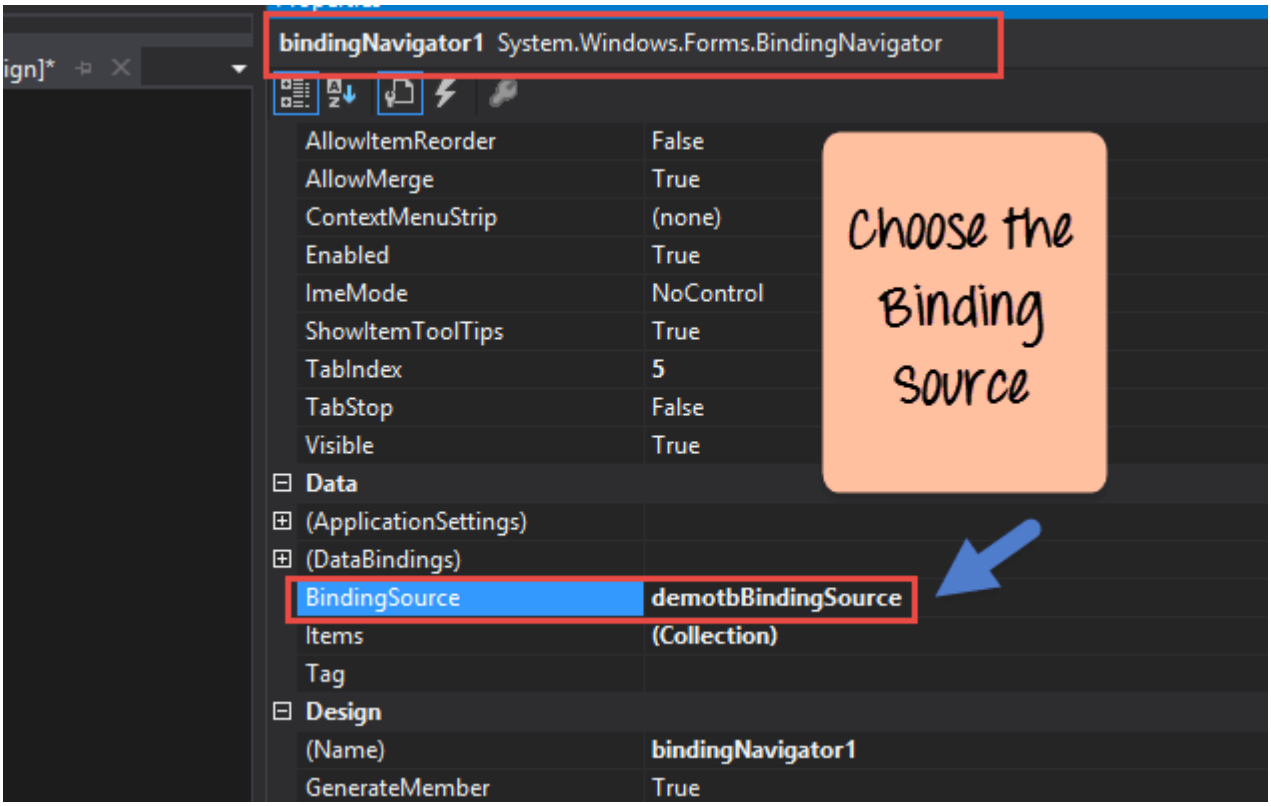
Step 6) Next we need to change the Binding Source property of the BindingNavigator to point to our Demotb d
the Binding Navigator also needs to know v

GURU99

The Binding Navigator is used to select the
even though the data source is added to th
control, we still need to ensure the Binding
source. In order to do this, we need to click
Binding Source property and choose the or

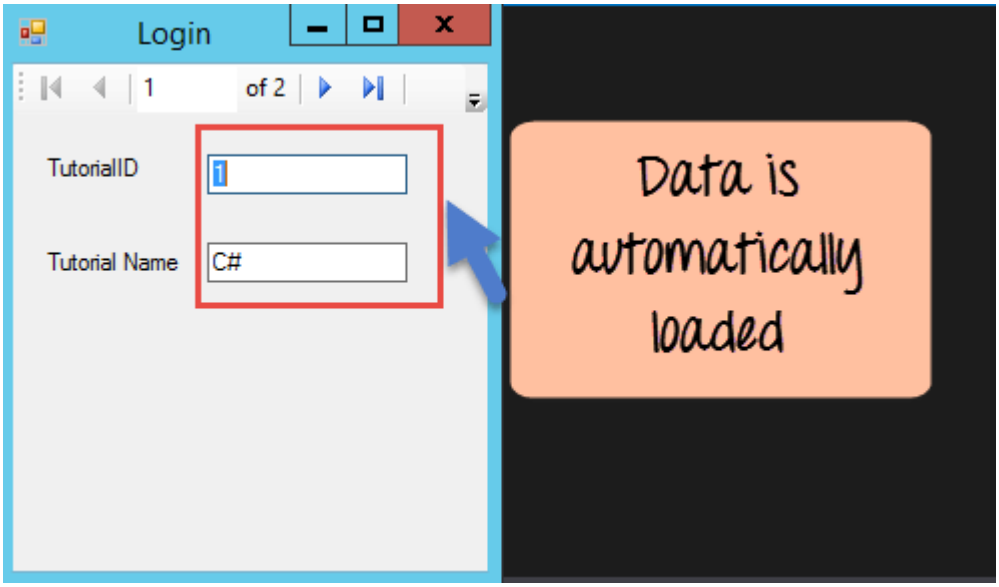


Next, we need to go to the Properties window so that we can make the change to Binding Source property.

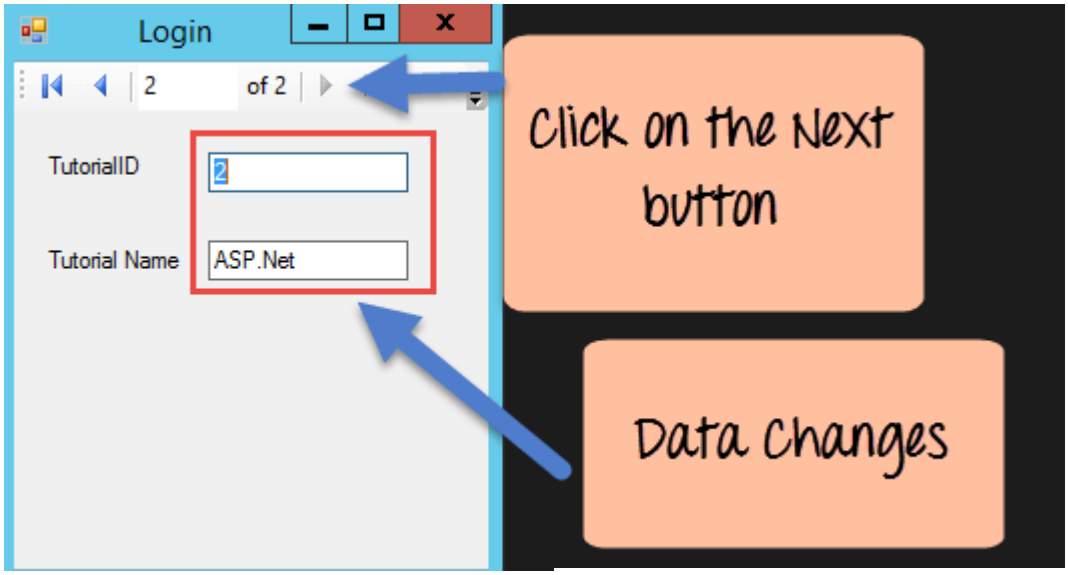


When all of the above steps are executed su GURU99 mentioned output.

Output:-



Now when the project is launched, you can see that the textboxes automatically get the values from the table.



When you click the Next button on the Navi record in the table. And the values of the ne boxes

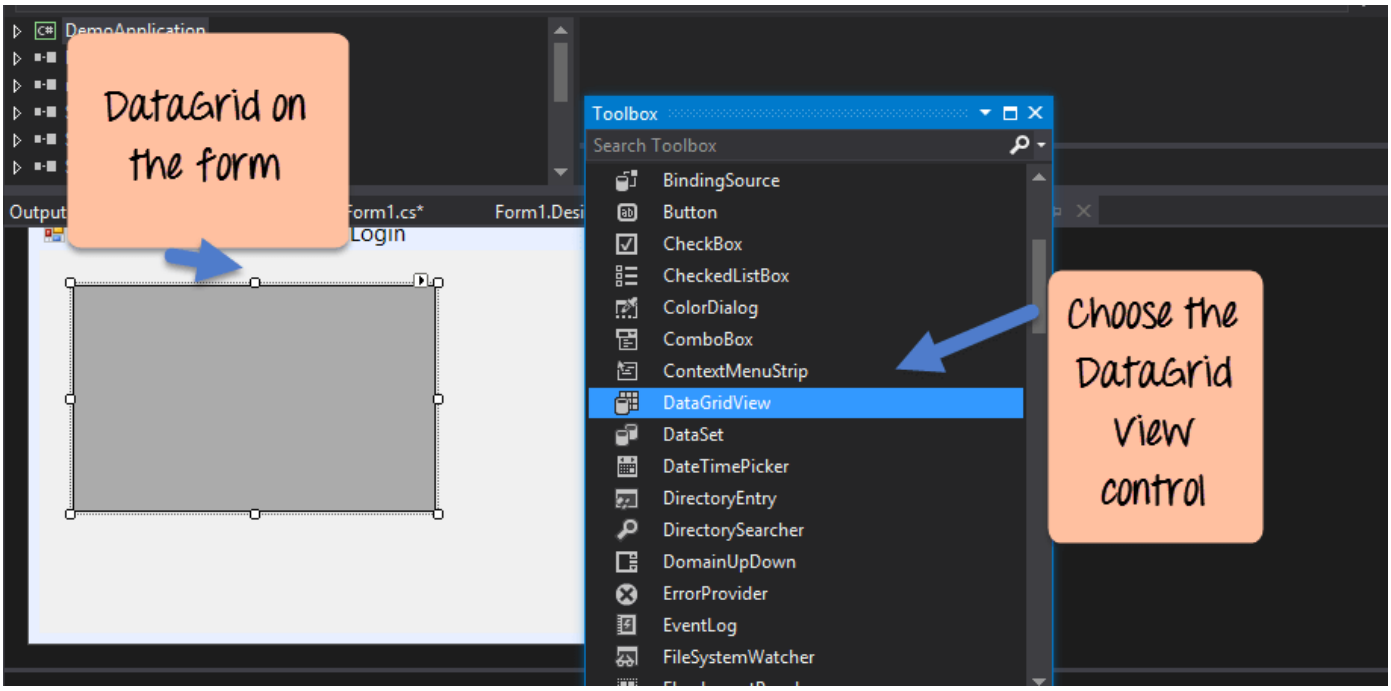
GURU99

C# DataGridView

Data Grids are used to display data from a table in a grid-like format. When a user sees's table data, they normally prefer seeing all the table rows in one shot. This can be achieved if we can display the data in a grid on the form.

C# and Visual Studio have inbuilt data grids, this can be used to display data. Let's take a look at an example of this. In our example, we will have a data grid, which will be used to display the Tutorial ID and Tutorial Name values from the demotb table.

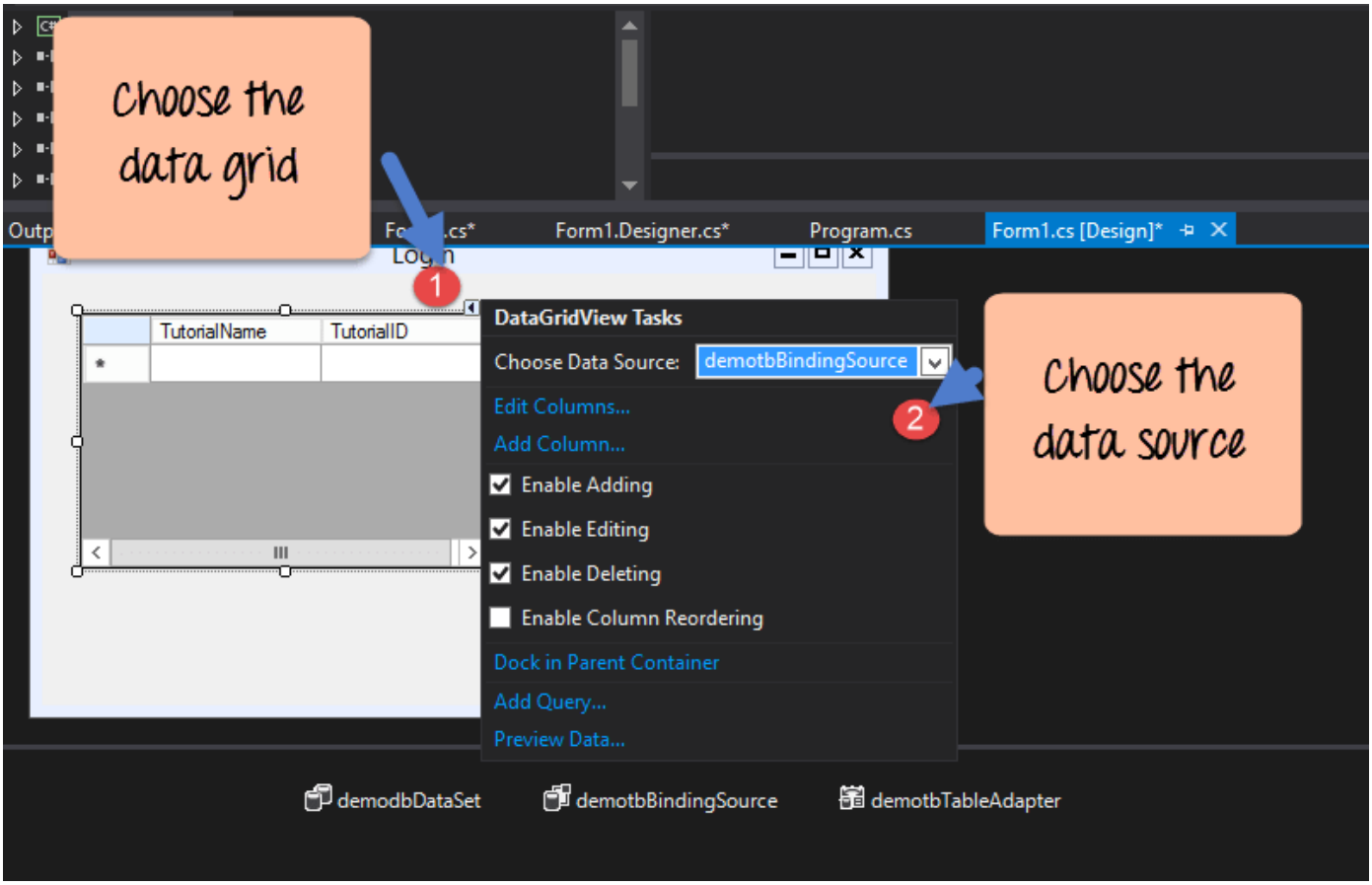
Step 1) Drag the DataGridView control from the toolbox to the Form in Visual Studio. The DataGridView control is used in Visual Studio to display the rows of a table in a grid-like format.



Step 2) In the next step, we need to connect our data grid to the database. In the last section, we had created a project data source. Let's use the same data source in our example.

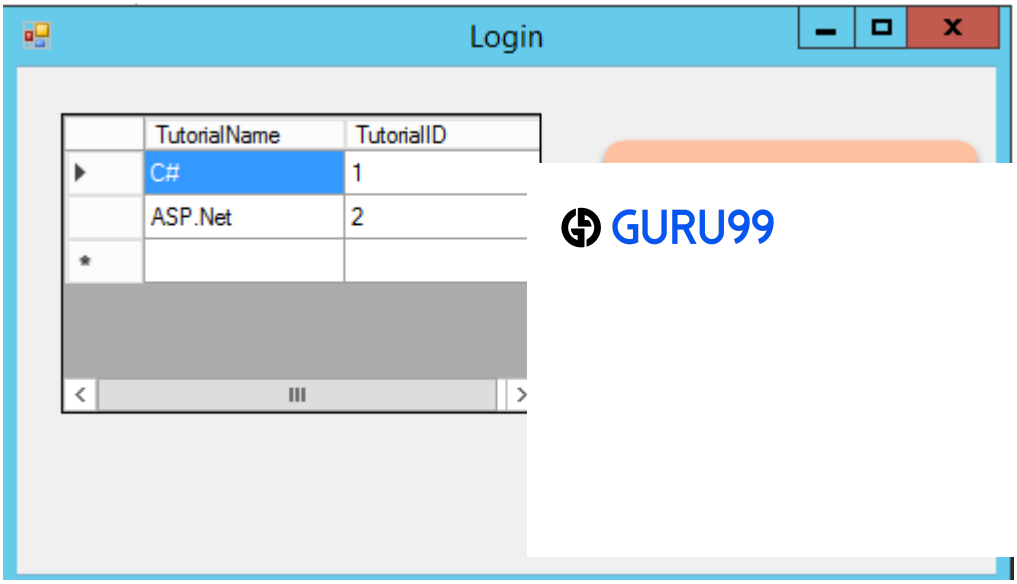
GURU99

1. First, you need to choose the grid and bring up the grid configuration option
2. In the configuration options, just choo demotbBindingSource which was the section.



If all the above steps are executed as shown, you will get the below-mentioned output.

Output:-



From the output, you can see that the grid was populated by the values from the database.

Summary

- C# SQL can work with databases such as Oracle and Microsoft SQL Server.
- This C# database tutorial has all the commands which are required to work with databases. This involves establishing a connection to the database. You can perform operations such as select, update, insert and delete using the commands in C#.
- The DataReader object in C# is used to hold all the data returned by the database. The While loop in C# can be used to read the data rows one at a time.
- The data adapter object is used to perform SQL operations such as insert, delete, and update against the database.
- C# can bind controls to the various fields in a table. They are bound by defining a data source in C#. The data source is used to pull the data from the database and populate them in the controls.
- The binding navigator is used to automatically navigate through the rows in a table.
- The data grid in C# can connect to the database and display all the values from the table in a grid-like format.

You Might Like:

- C# Array Tutorial: Create, Declare, Initialize
- Top 50 C# Interview Questions and Answers (2023)
- C# Tutorial PDF (Download Now)
- 13 BEST C# Books (2023 Update)
- BEST C# IDE for Windows, Linux & Mac

GURU99

Prev Report a