2015

C# Database Application

LEARN TO DESIGN & BUILD A WINDOWS C# DATABASE APPLICATION

STEPHEN O'CONNOR

Contact me:
stevo.joc@gmail
github.com/stevo5o
Stackoverflow Stevo5o

Database

What is a database?
A file structured for the repository of data. Organized for easy retrieval, sorting, grouping, relating to other data, used to analysis information in numerous ways.

Databinding

Utilizing Databinding in a C# Win forms App. Data sets working with the System.Data Namespace (aka ADO.NET) Working with the Visual Studio's IDE's tools, windows, etc. Microsoft Visual Studio 2013 makes it easily to create databases for beginners and experts.

Databinding the user interface controls, retrieve and display data from a data source without requiring the programmer to worry about all the programmatic details of this process. Each user interface control has different properties that can be bound to a data source.

ADO = ActiveX Data Objects
User interface controls must be data binding "aware", ADO.NET(System.Data) classes support data binding.

- ADO.NET creates a connection to a data source (database)
- ADO.NET manages the conversation (requests and responses) between your application and the database.
- ADO.NET manages the data that is retrieved forum the response to the database query.
- BindingSource manages the connection between the user interface controls and the underlying data set retrieved from the database. Provides an application interface to reduce learning curve for the end user. Restrict access to the database to maintain security. To control the presentation of the data. Maintain the integrity of the data.Practice

ADO.NET does a lot of the grunt work, it is not necessary to know all about ADO.NET.

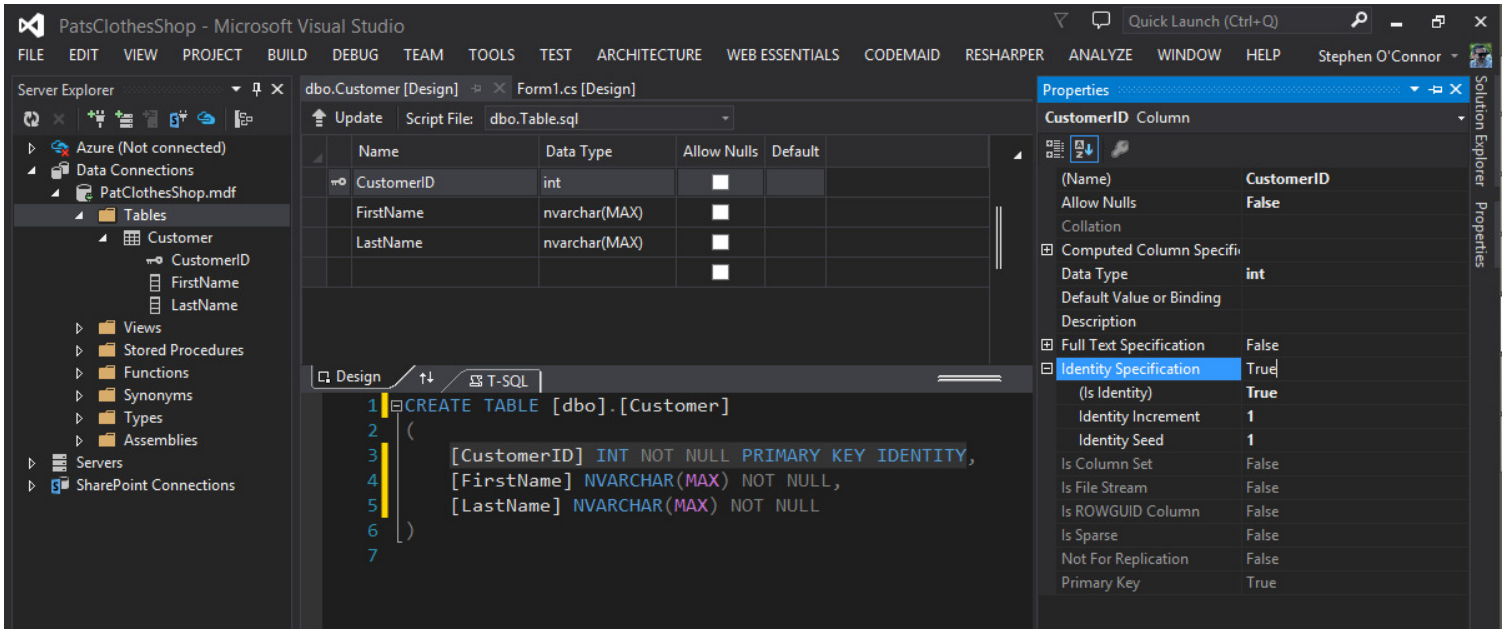
Write application interface

- Reduce the learning curve for the end user
- Restrict access to the database to maintain security
- To control the presentation of the data – website, content management system
- To maintain the integrity of the data

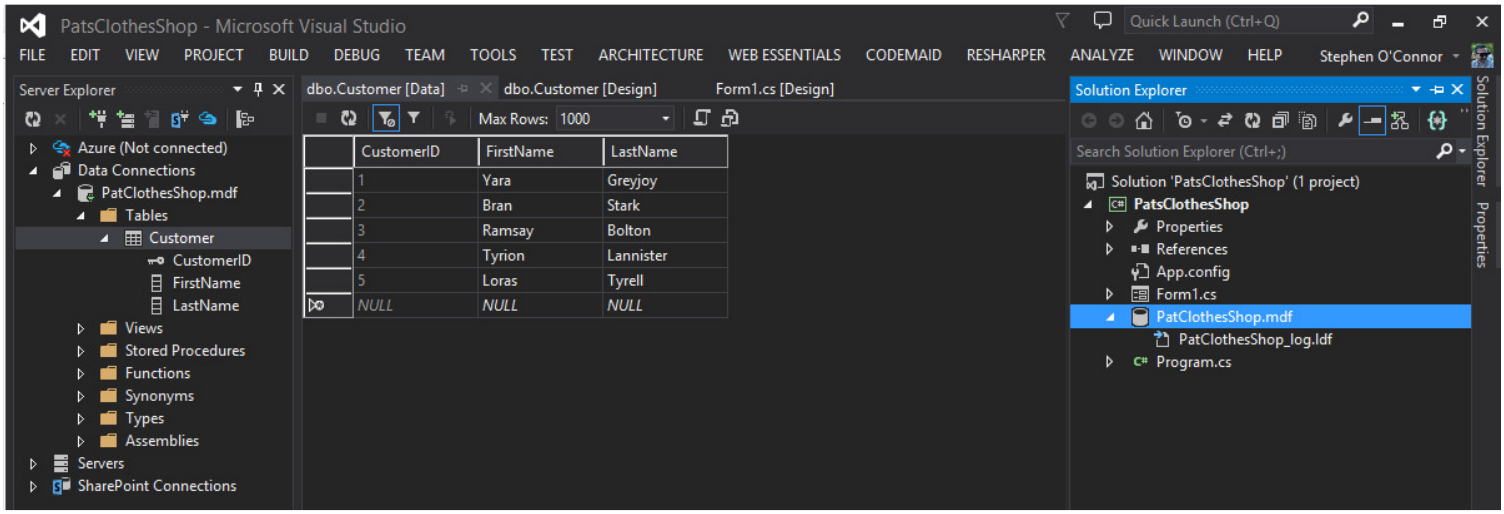
SQL Server
A high end relational database management system.
SQL server 2013 Express Edition, similar power, but intended for smaller projects.
In Visual Studio 2013 download the latest SQL Server Data Tools, if already not installed.

SQL Fundamentals

Learn by doing
Create a new project and a database called PatClothesShop
Create a project called Pats
Add a table called customer with the data below.

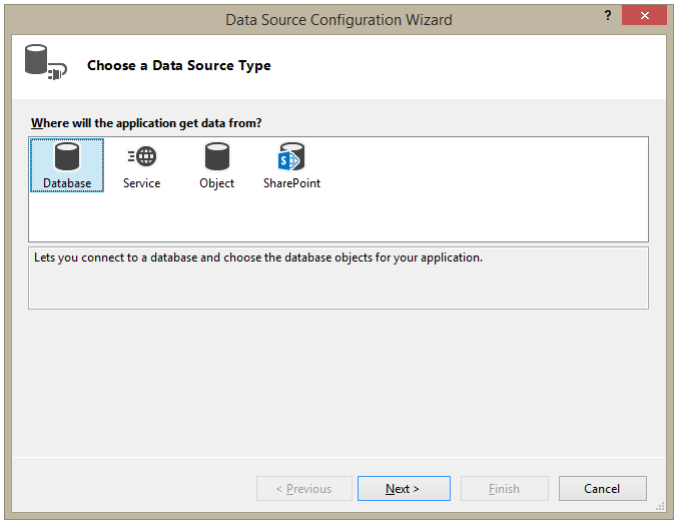


Go to show table data in the Server Explorer add five persons.

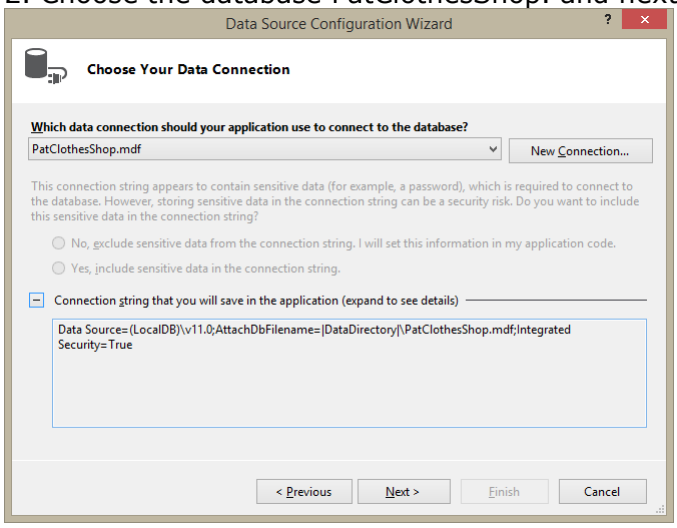


On the menu bar, choose View, Other Windows, Data Sources (or choose the Shift+Alt+D keys). Follow the steps.

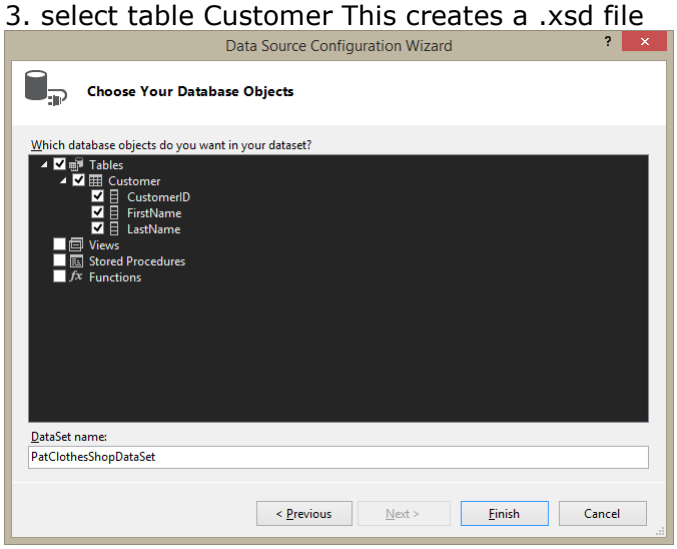
1. Choose Database and next



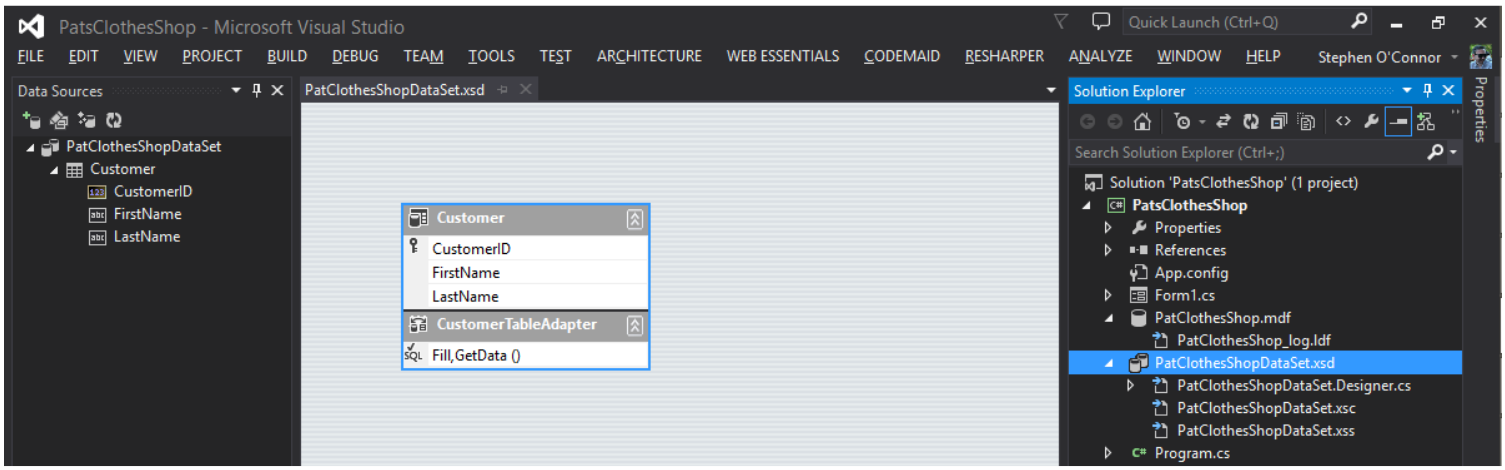
2. Choose the database PatClothesShop. and next



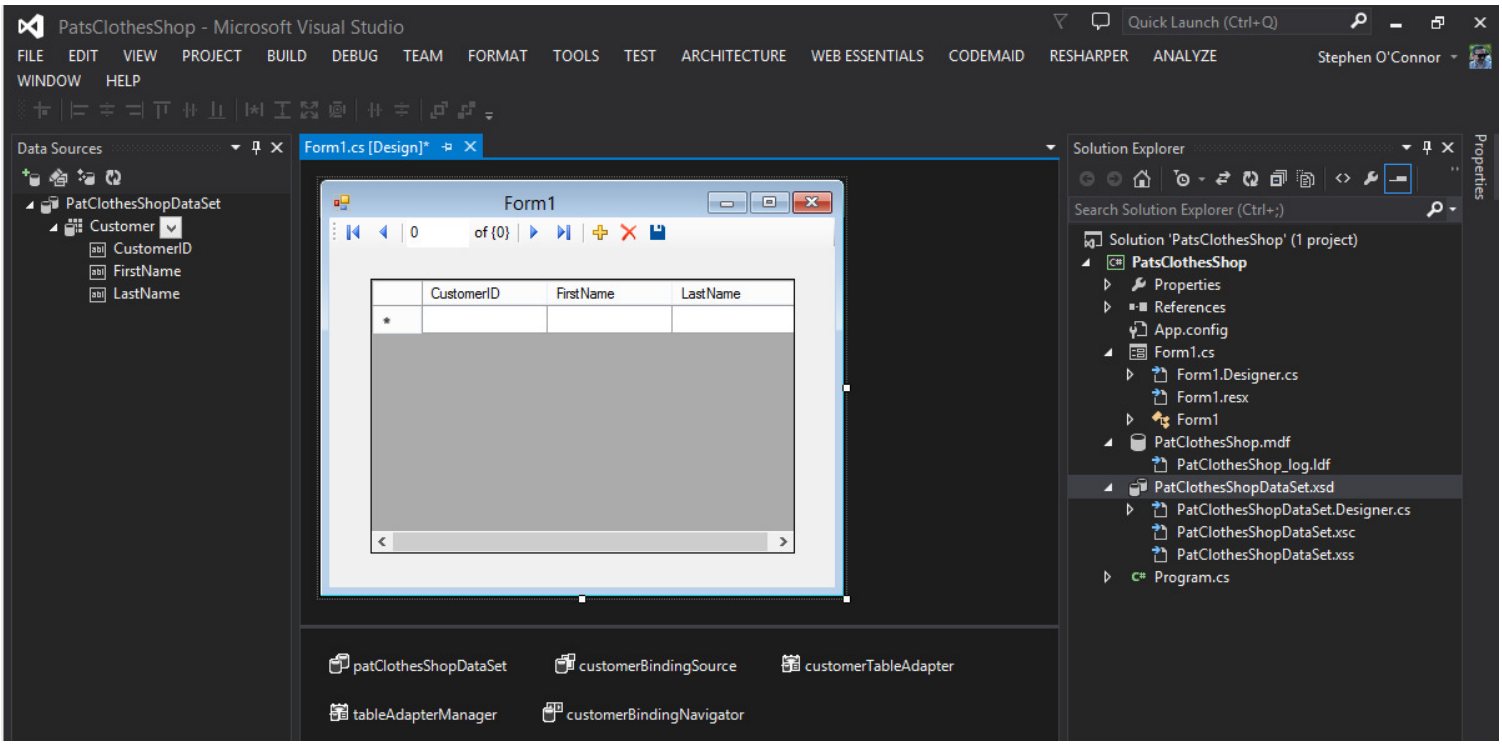
3. select table Customer This creates a .xsd file



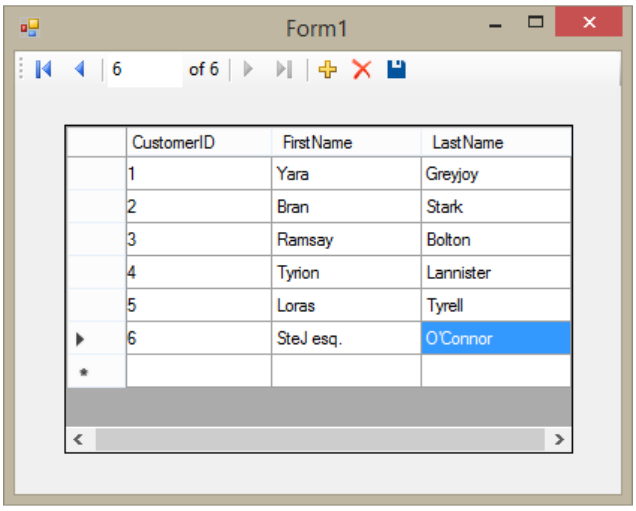
A PatClothesDataSet.xsd file right click on the xsd file and view designer mode. Xsd file is the xml schema document. The xsd file a local copy of database, this file defines the database, temporary stores the data.



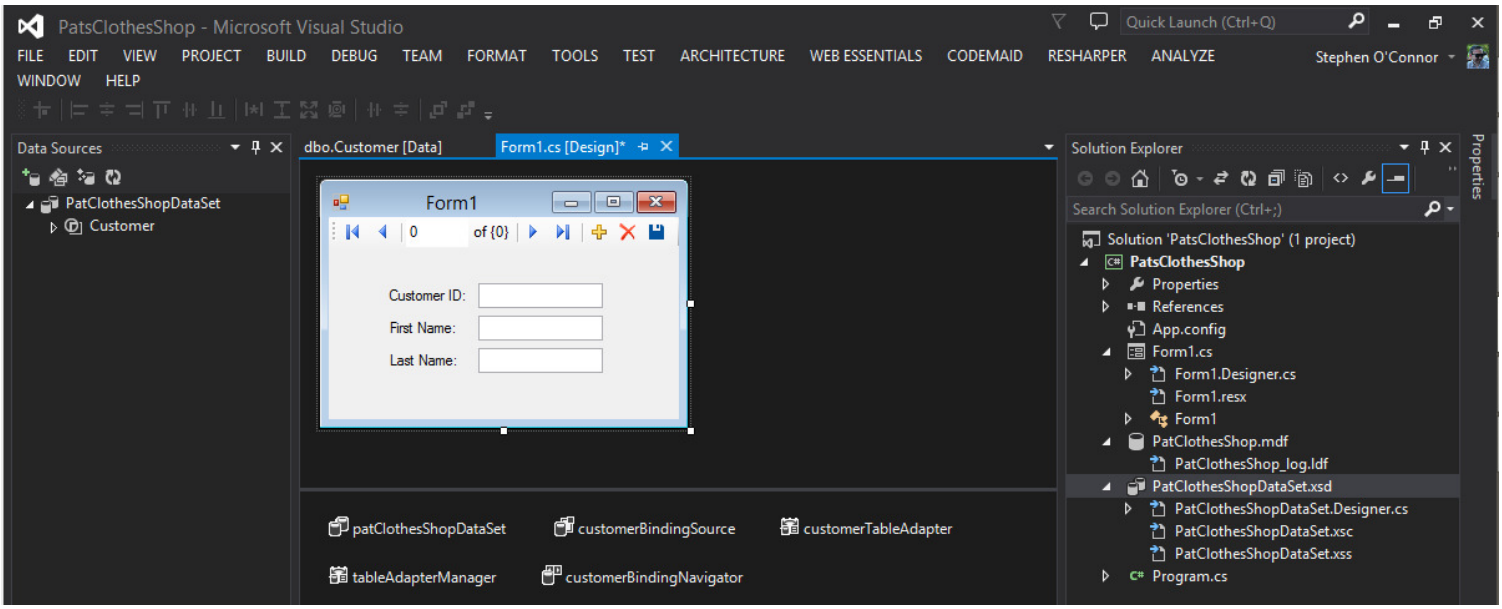
Drag and drop customer table from the Data Sources toolbar.



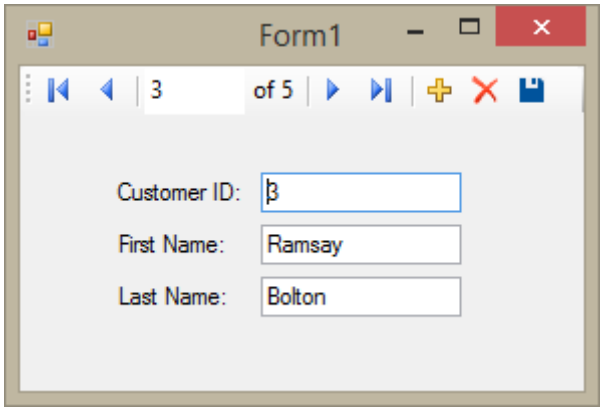
Run the project. A grid of the database navigate through the grid and add an extra row.



To create a Form view; select details from the drop down onto Form1



Run the application and navigate through the details view.



Designer tray.
patClothesShopDataSet
The local container for the data within the application. Once the form is opened the dataset gets populated from the data from the database. Temporary storage container.

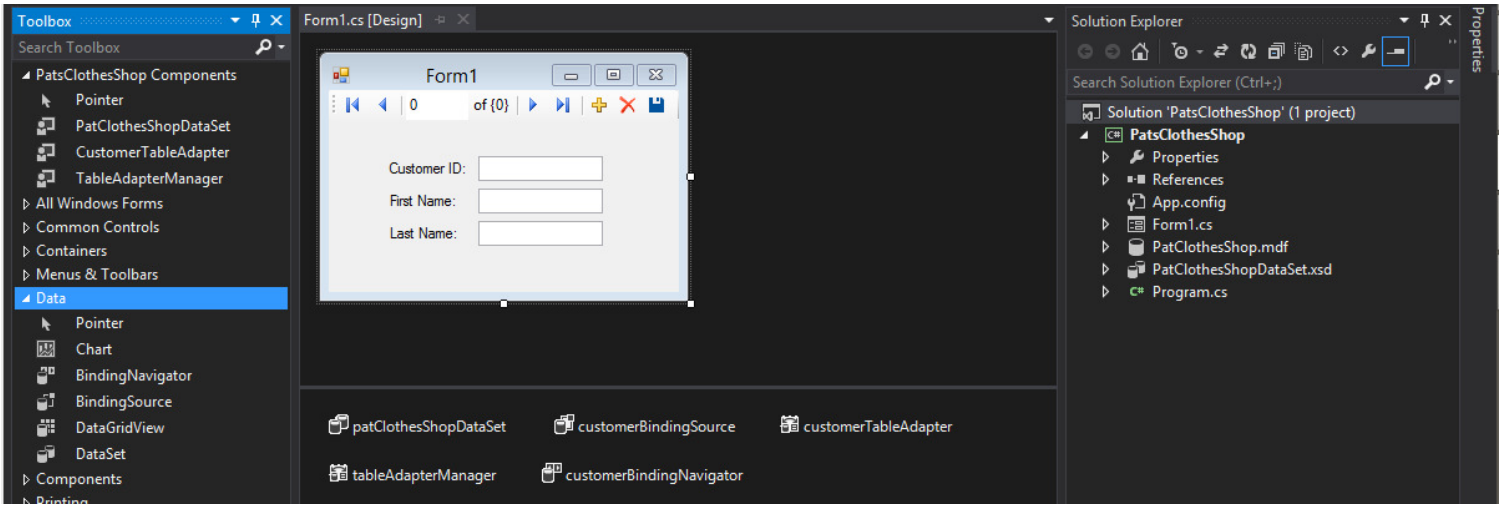
patClothesShopBindingSource
Object /bridge between the information in the dataset and the current row that's being displayed on the form. Keep all of the controls on the form bound to row of data in the DataSet. Co-ordinates what row of data should be currently displayed. The user indicated wanted to go to the first row or the next row or the last row

patClothesShopTableAdapter
Retrives data from the database, it contains a connection to the database. Contains an object that connects to the PatClothesShop.mdf to retrieve and resolve the information back into the database. Delete, update, add.

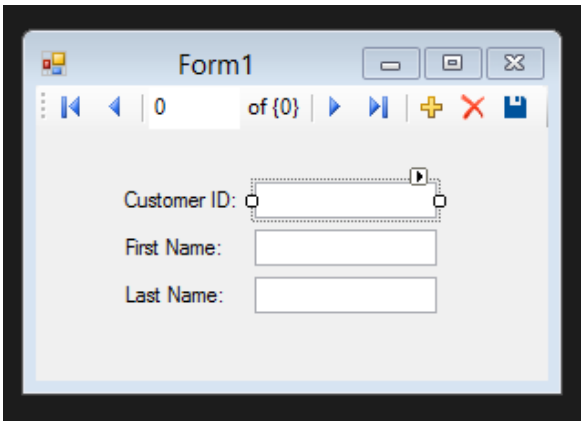
patClothesShopAdapterManager
Service interface

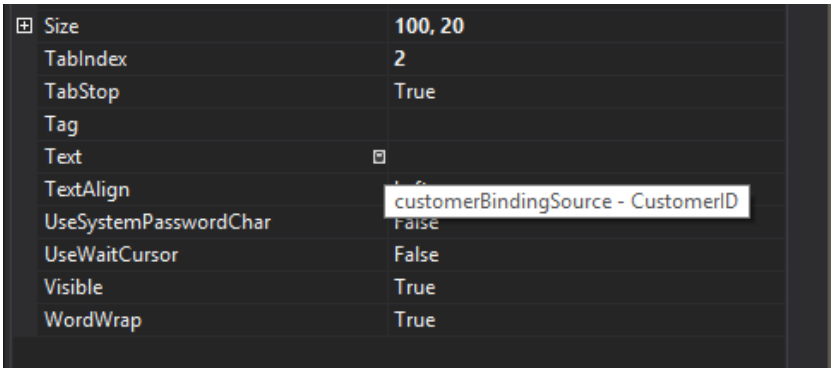
patClothesShopBindingNavigator
Toolbar at the top of the form.

Go to toolbox and select data to show data tools that can be added manually to the form.

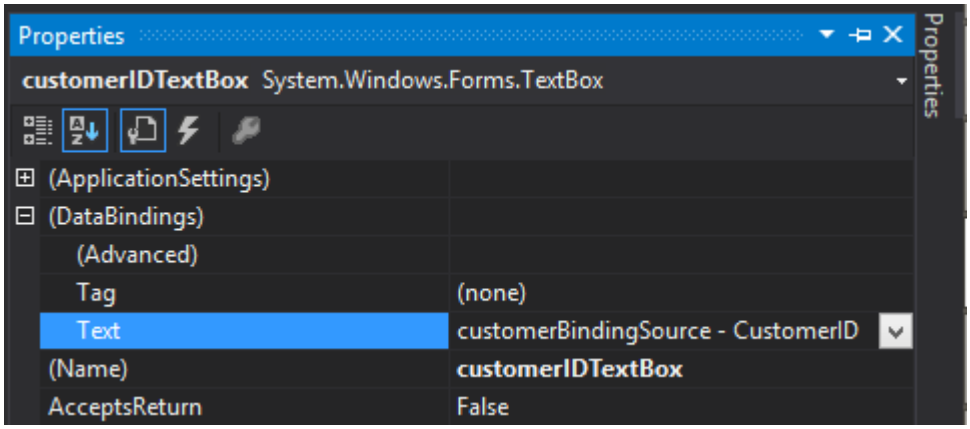


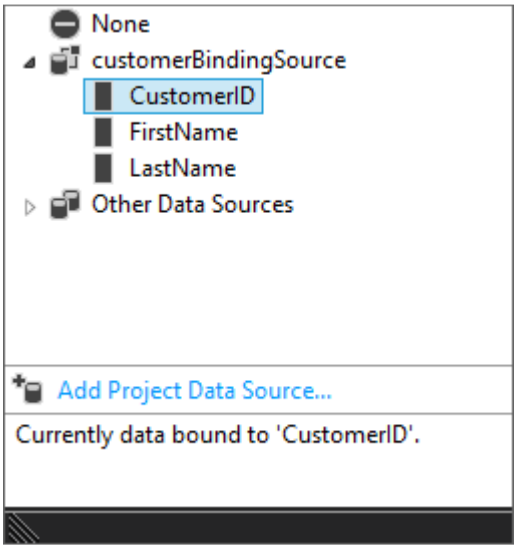
Properties of the first textbox. In the Text a Database icon is displayed.



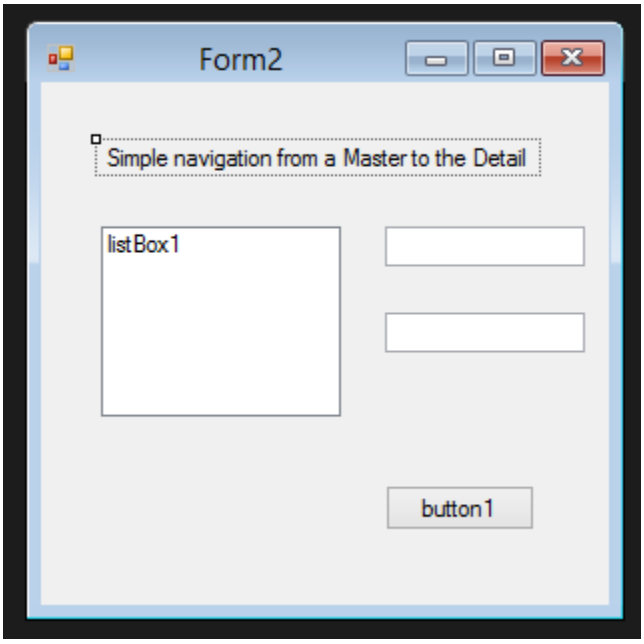


Select and right click to display the propertites of the textbox. CustomerID is binded to the first textbox. Binding source schema document. By using the data sources toolbar the database can be dragged and dropped onto the form, sets the textboxs automatically.



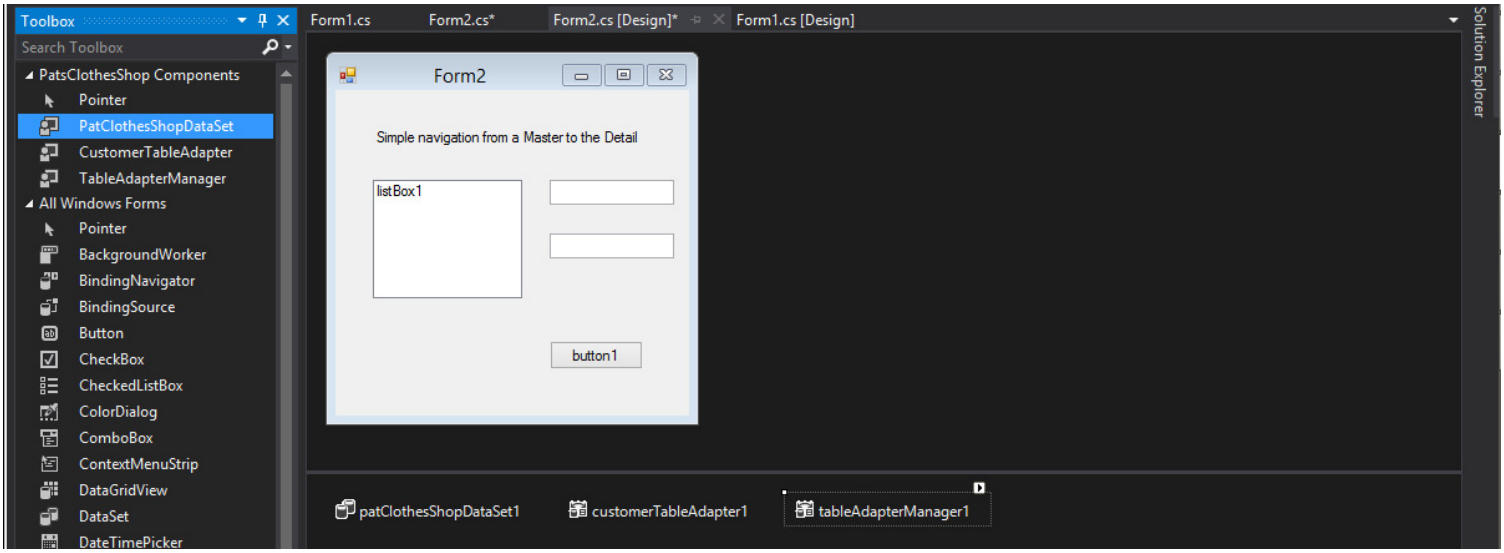


Add the data items manually.
Strongly typed components or preconfigured.
Create a new form form2 add the items displayed in the below image. Label, Listbox, 2 textboxs and a button.



Add a button to form1 to open form2. Type the code below to create an instance of the class Form2

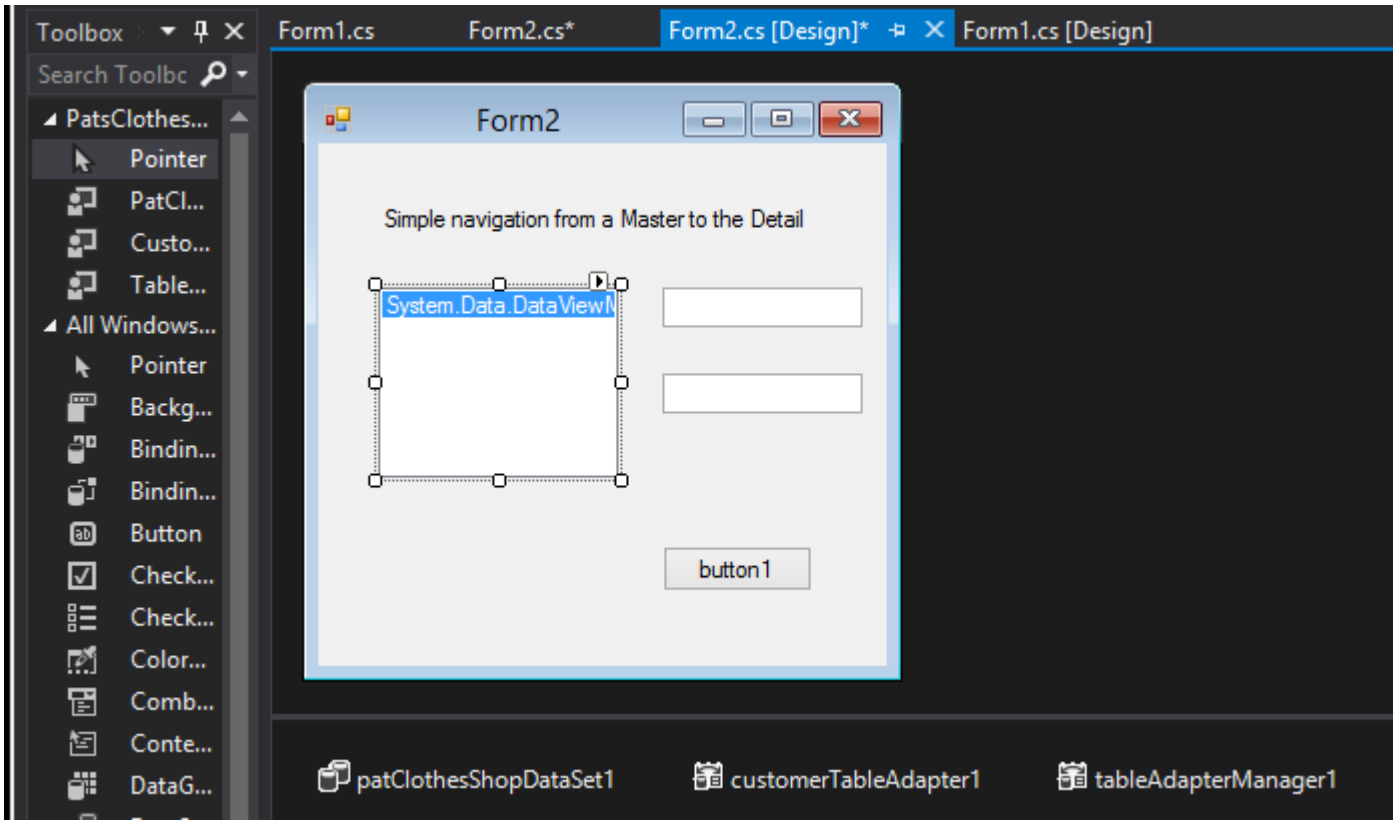
```
Form2 myForm = new Form2();
            myForm.Show();
```

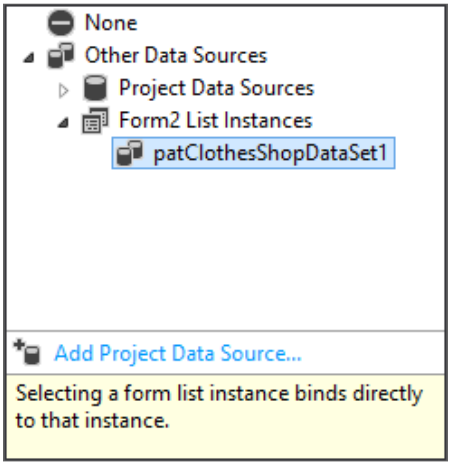


Double click form2 type the code below. Fill method passes in `patClothesShopDataSet1.Customer` Fill method takes action to grab the data from the database and populate the customer table of the database with the data it retrieves.

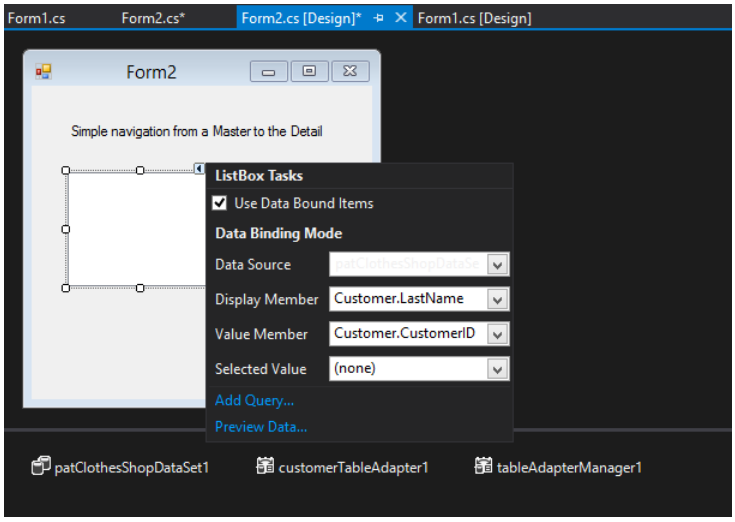
```
private void Form2_Load(object sender, EventArgs e)
        {
            customerTableAdapter1.Fill(patClothesShopDataSet1.Customer);
        }
```

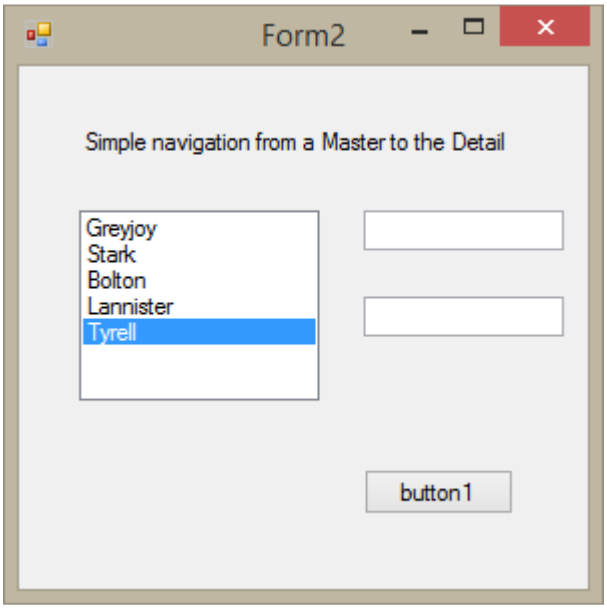
Select the listbox click the arrow and select patClothesShopDataSet1, from the pop-up box.


Toolbox
Search Toolbc
PatsClothes...
Pointer
PatCl...
Custo...
Table...
All Windows...
Pointer
Backg...
Bindin...
Bindin...
Button
Check...
Check...
Color...
Comb...
Conte...
DataG...
Form1.cs
Form2.cs*
Form2.cs [Design]*
Form1.cs [Design]
Form2
Simple navigation from a Master to the Detail
System.Data.DataViewM
button1
patClothesShopDataSet1
customerTableAdapter1
tableAdapterManager1

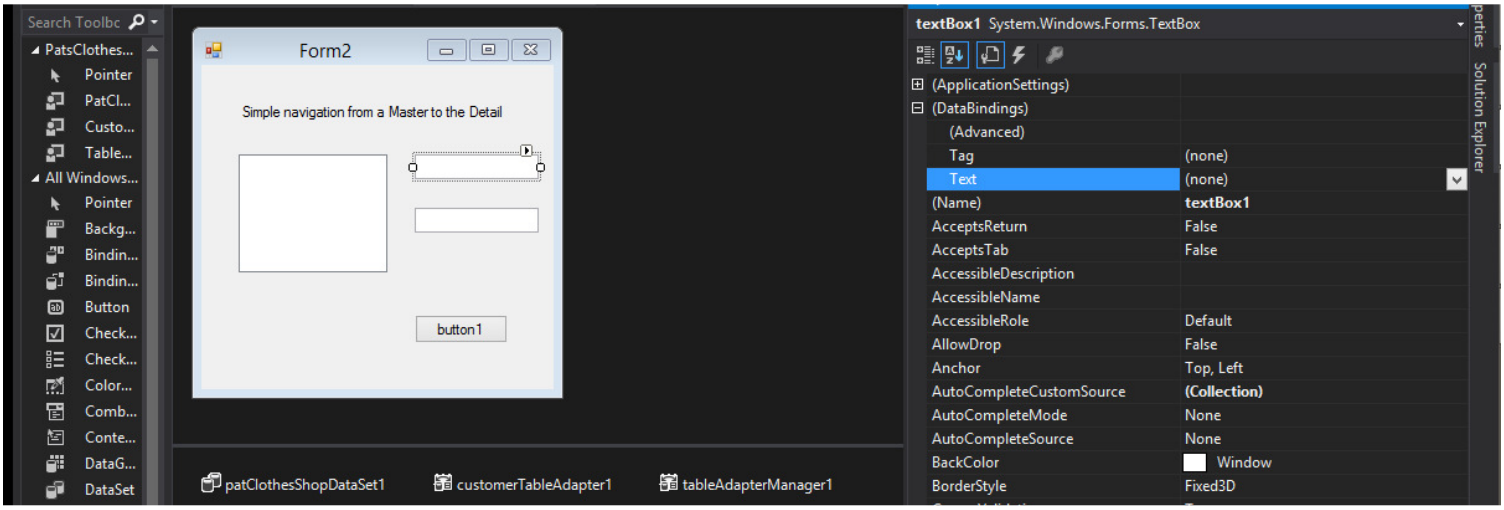

None
Other Data Sources
Project Data Sources
Form2 List Instances
patClothesShopDataSet1
Add Project Data Source...
Selecting a form list instance binds directly to that instance.

Select the list box and click the arrow button to data bound the listbox to the dataabse. Check the checkbox Use data bound items. From the pop-up select the data source patclothesshop1. In Display member select the last name and Customer last name this will displayed. In value member select the CustomerID, which is always unique.

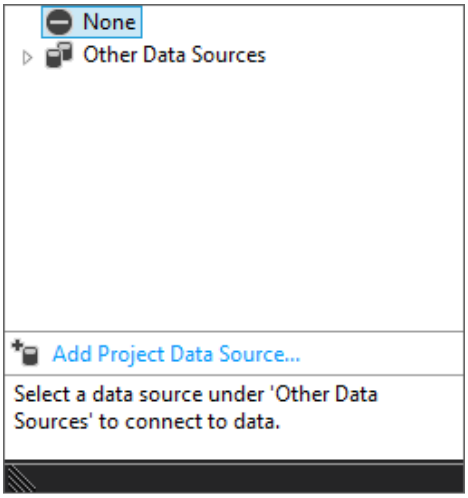


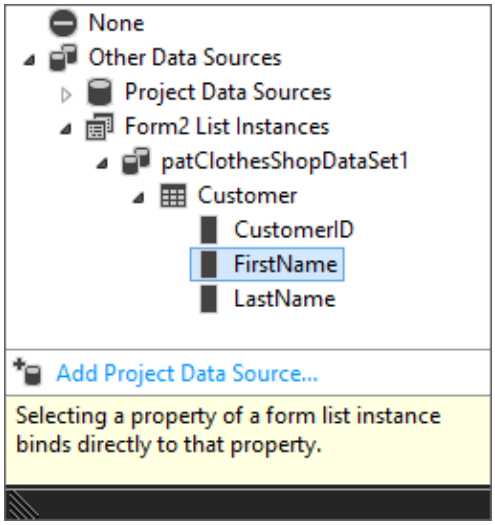


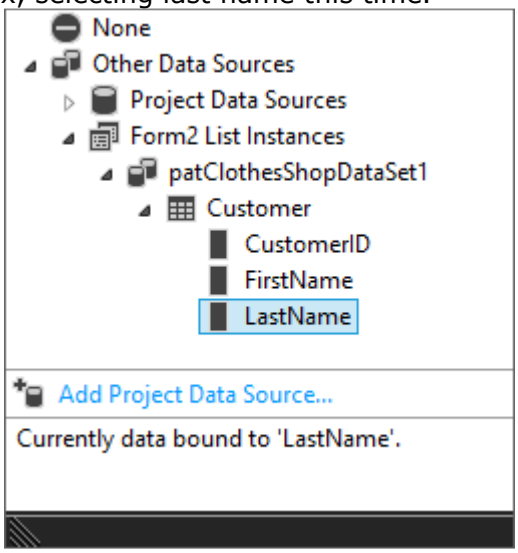
Bind the textboxs on the right with the listbox on the left. Select the first textbox and right click properties go to databindings and select text click the drop down arrow.



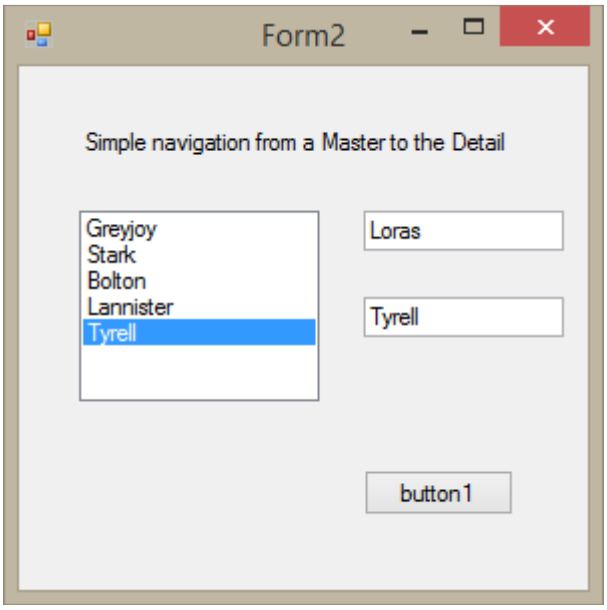
From the pop-up select first name for the first textbox. Go back to the form and select and right click the second textbox go to properties and repeat steps for the second textbox, selecting last name this time.



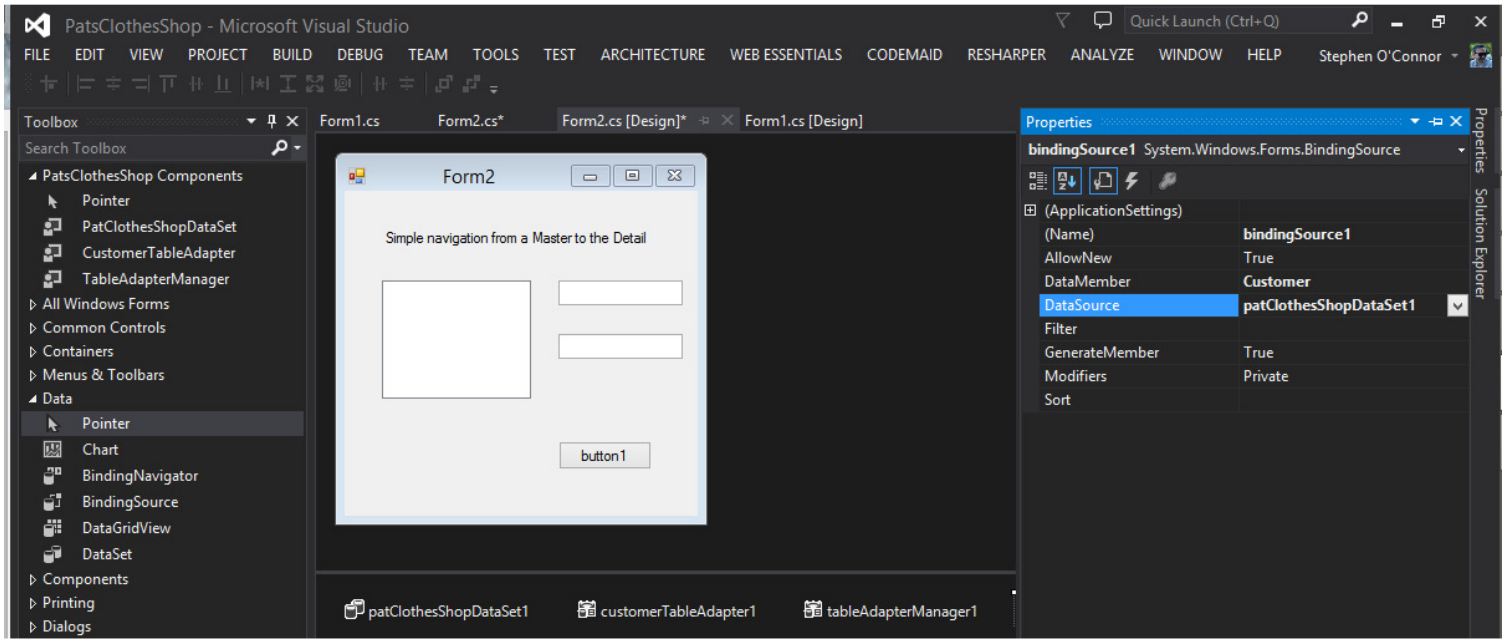




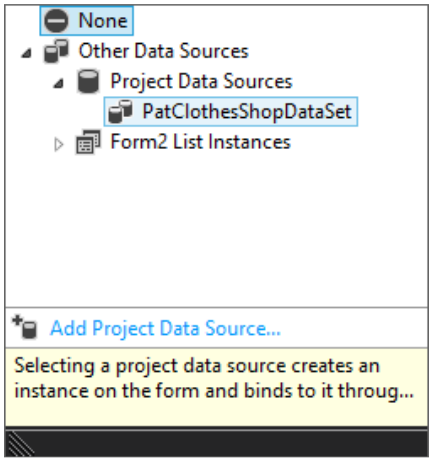
Run the application, open second form. Select from the listbox to display the first name and last name of the person in the textboxs.



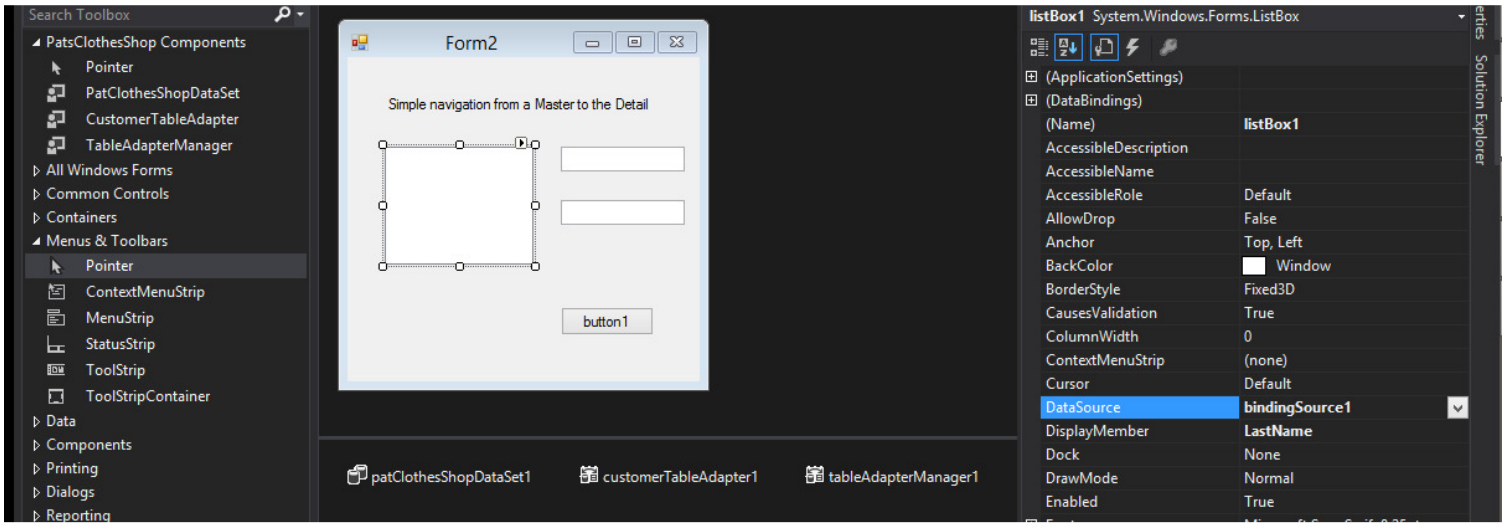
Updating capabilities. Drag and drop the BindingSource onto form2. To update the data. Hover over and read the description of the BindingSource tool. The bindingSource1 has been added to the designer tray.

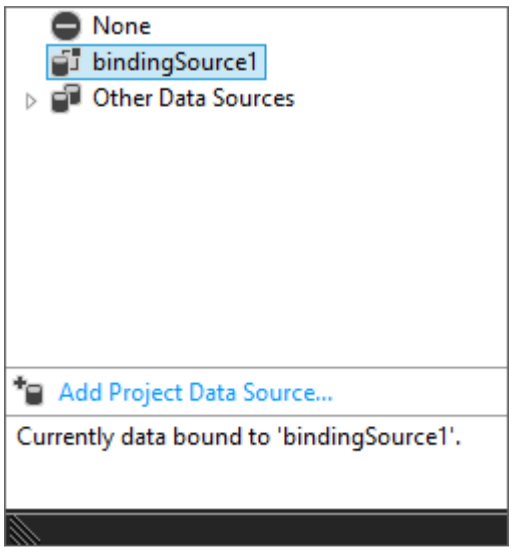


Click bindingSource1 and select PatClothesShopDataSet from the properties DataSource pop-up. Select Customer as the DataMember.

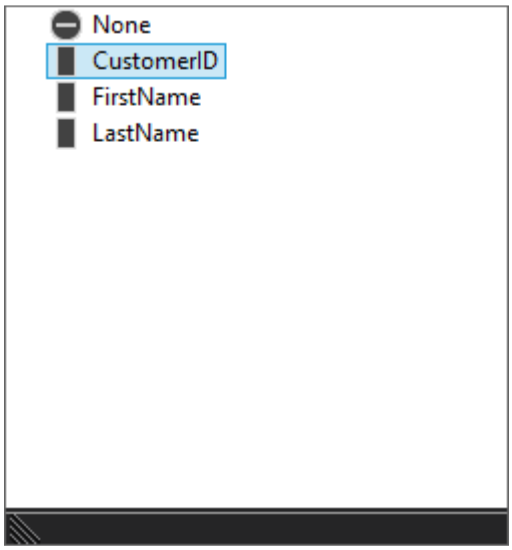


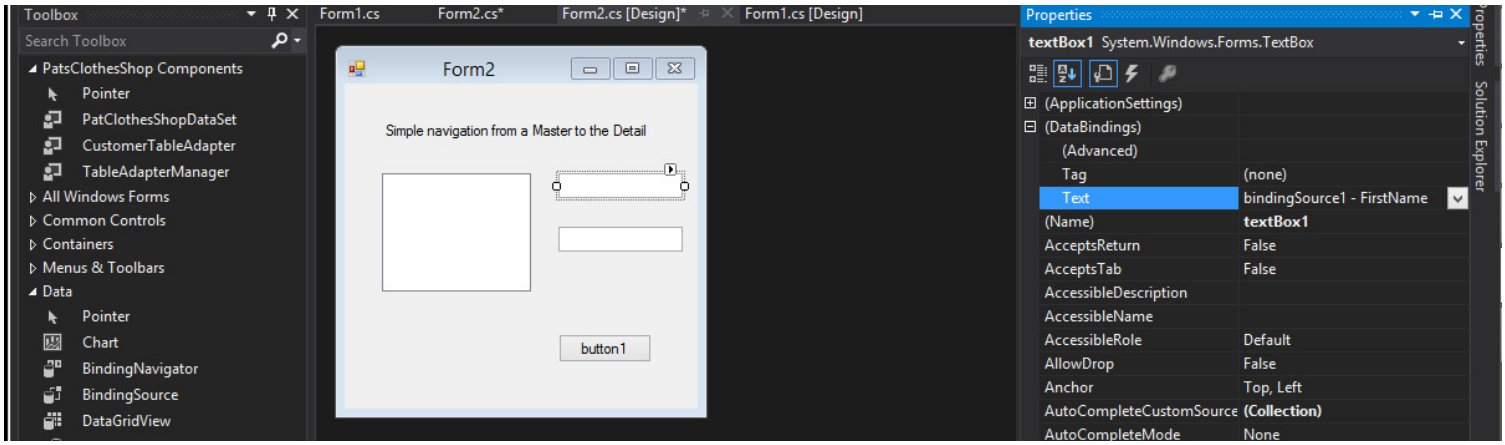
Select listbox go to data source select bindingSource1. Change the DisplayMember to LastName.



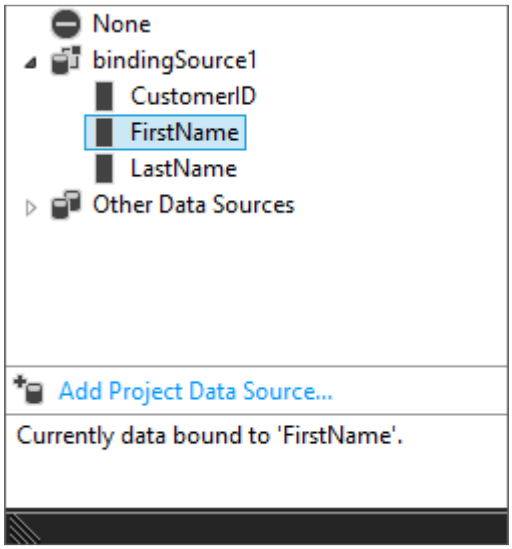


Select Value member in the listbox from properties and change to CustomerID.

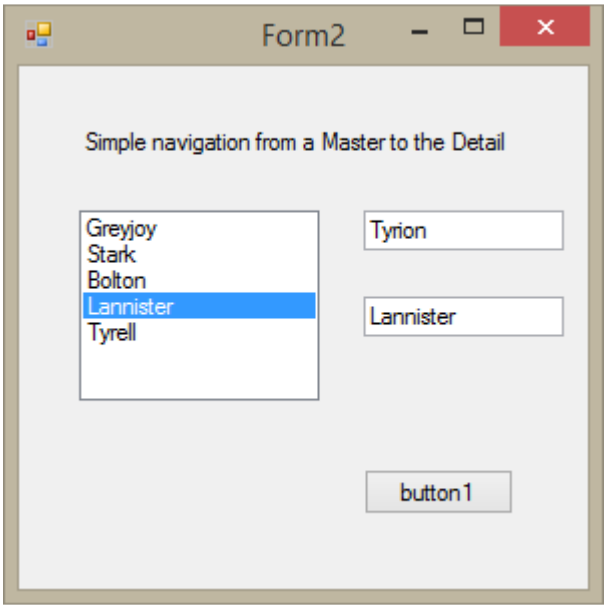




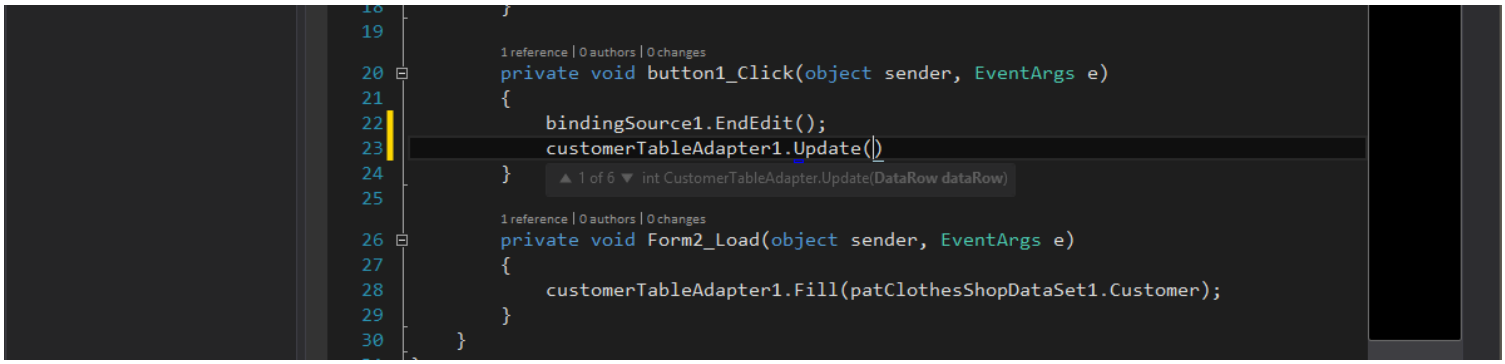
Select the properties of the first and second textboxs and select first and last name respectively.



Run application to check if it still works.



Double click the button on form2




```
private void button1_Click(object sender, EventArgs e)
        {
            bindingSource1.EndEdit();
            customerTableAdapter1.Update(patClothesShopDataSet1.Customer);
        }
```

```
            // save changes to the dataset
            bindingSource1.EndEdit();

            // select TableAdapter and return number of items updated
            customerTableAdapter1.Update(patClothesShopDataSet1.Customer);
```

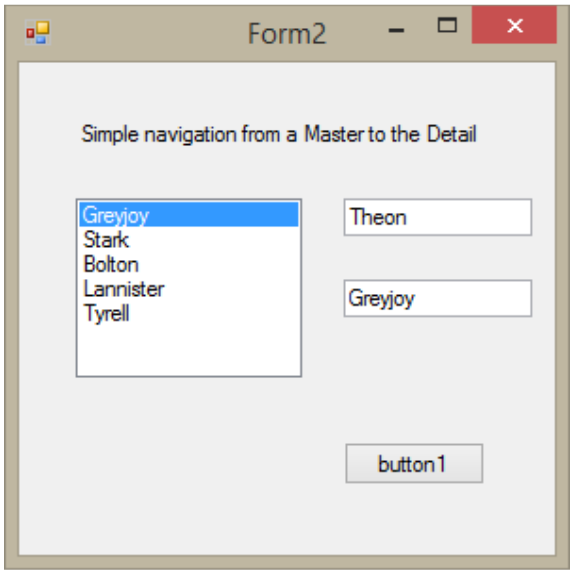
```
            // save changes to the dataset
            bindingSource1.EndEdit();

            int result = 0;

            // return number of items updated
            result = customerTableAdapter1.Update(patClothesShopDataSet1.Customer);

            // display the row has been updated
            MessageBox.Show(result.ToString());
```

Open the .exe file in the debug /bin folder. Change Yara (first name to Theon) close and reopen .exe file and see the change made to the database.



stackoverflow
Actually this is not a issue with update command. Visual Studio keeps two databases. One in project folder and one in bin/debug folder. Database in bin/debug folder always update with Database in project folder. If you view Database through Visual Studio, it always shows the Database inside the project folder not other one inside bin/debug folder.

My Set-up