

# SQL Fundamentals

LEARN TO DESIGN & BUILD A WINDOWS SQL  
DATABASE

STEPHEN O'CONNOR

## 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.

Example: Customer database

To store Customer Data, use a simple text file, not easily manageable. Why not use an Excel spreadsheet file?  
 No easy way to relate sales data to customer information

Relational Database Management System RDBMS

Relational Database Theory – Organizing data into tables that can be related together, this reducing redundancy and increasing the integrity of the data.

Normalization – the process of determining what information belongs in which tables to minimize redundancy and increase integrity.

Database Objects

- Tables contain
  - Columns
  - Rows or records the value of a single column and a single row is a called a “field”

## Data Integrity

Keeping data valid, of the correct data type, etc., so that it is usable for its intended purpose.

Customers can be moved separate tables

What Needs to be Defined

- The data type for each column in a table
- The maximum size of data that will be stores in the column
- The nullability of a column

In the customer table the credit limit field is a real data type, this is so that the user has to enter a numerical value, so that calculations can be made using the numerical data. The real data type is approximate numerical. The customer since is a date time data type, only dates can be enter into this field. Null fields allow records to be saved to the database, to maintain data integrity.

Primary Key

A field or combination of fields that make a given row unique in the database. A way of differentiating each row in a table when all other fields might be duplicated in other rows in the same table.

Identity column an attribute that will be automatically increment a field of data in each successive row added to the database. Typically used on the primary key fields to make them unique.

Foreign key

Relate one or more rows in one table to a record in another table that shares the same value in its primary key. To check who made the order use customerID as a foreign key to relate the tables in the database. Data is linked, order is linked to the customer who created the order, i.e. query the database; how much a customer has spent. Adding a FK constraint prevents deletions in the customer table to create orphaned rows in the order table. FK constraints enforce “Referential Integrity”.

Customer			
KEY	customerID	int (11)	
	firstName	varchar(50)	
	lastName	varchar(50)	
	address	varchar(200)	Null
	city	varchar(200)	Null
	county	char(10)	Null
	zip	char(10)	Null
	creditLimit	real	
	customerSince	datetime	

Order	
KEY	orderID orderDate orderAmount paymentType
FK	customerID

Download and PowerShell, run SQLite3.exe create new database.

```
sqlite>../sqlite3.exe PatsClothesShop.db
```

```
Customer      Order
```

```
sqlite> CREATE TABLE Customer(  
    costumerID INT PRIMARY KEY    NOT NULL,  
    firstName      CHAR(50) NOT NULL,  
    lastName       CHAR(50) NOT NULL,  
    address        VARCHAR(200),  
    city           CHAR(10),  
    county         CHAR(10),  
    creditLimit..... REAL,  
    costomerSince  DATETIME  
);
```

```
sqlite> CREATE TABLE Order(  
    orderID INT PRIMARY KEY    NOT NULL,  
    orderDate    DATETIME NOT NULL,  
    orderAmount  REAL,  
    paymentType  INT,  
    customerID   INT      NOT NULL  
);
```

```
sqlite>.tables
```

```
Customer      Order
```

```
sqlite>.header on
```

```
sqlite>.mode column
```

```
sqlite>.timer on
```

Insert 5 customers like below

```
INSERT INTO Customers (firstName, lastName, address, city, county, creditLimit,  
costomerSince)  
VALUES (1, 'Paul', 'Murphy' 32, 'Apt 1', 'Dublin', 'Dublin', 15000.00, '2007-01-01  
10:00:00' );
```

## 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 easy 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 from 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.

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.

The screenshot displays the Visual Studio 2013 interface for a project named 'PatClothesShop'. The 'Server Explorer' on the left shows the database structure, including a 'Customer' table with columns 'CustomerID', 'FirstName', and 'LastName'. The 'T-SQL' editor in the center shows the SQL script to create the 'Customer' table. The 'Properties' window on the right shows the properties for the 'CustomerID' column, including 'Is Identity' set to True.

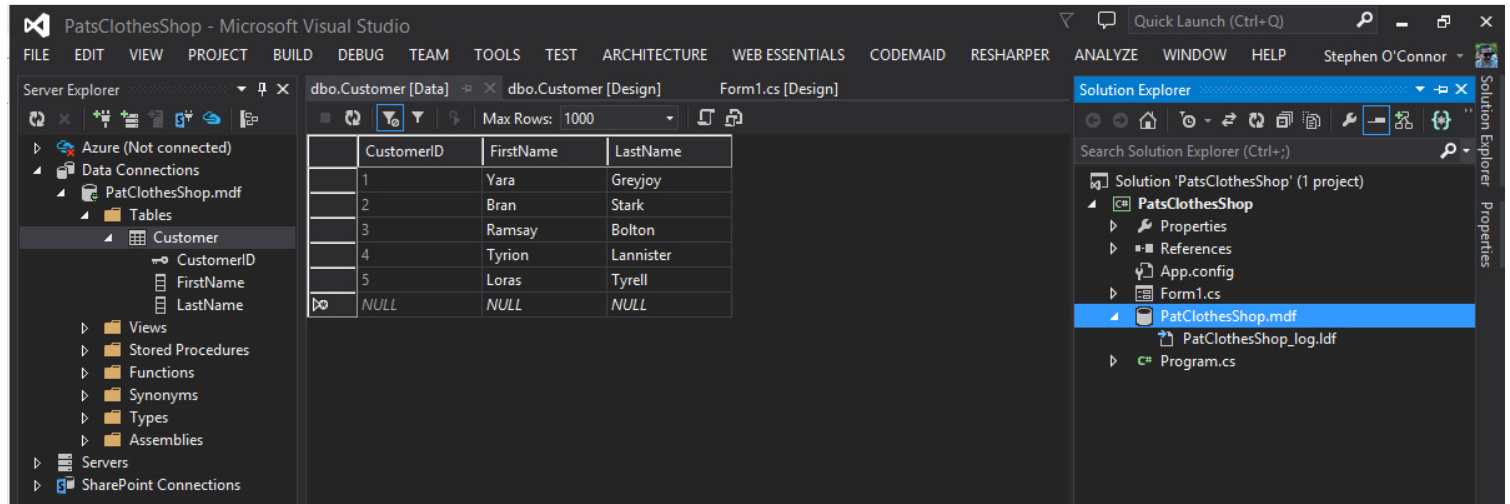
**SQL Script:**

```
1 CREATE TABLE [dbo].[Customer]
2 (
3     [CustomerID] INT NOT NULL PRIMARY KEY IDENTITY,
4     [FirstName] NVARCHAR(MAX) NOT NULL,
5     [LastName] NVARCHAR(MAX) NOT NULL
6 )
7
```

**Properties Window (CustomerID Column):**

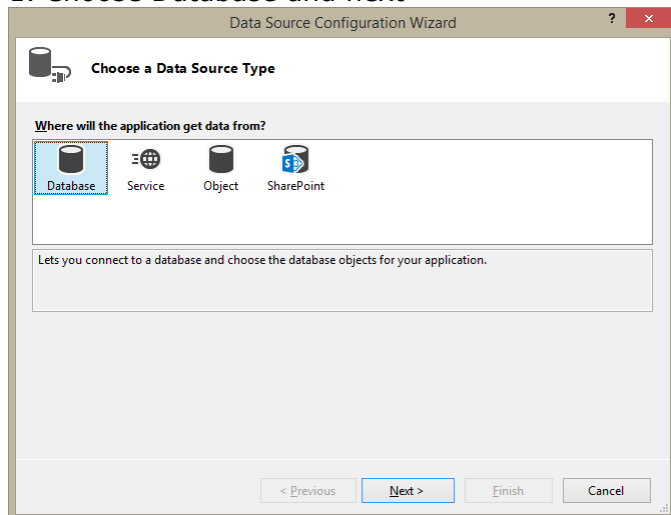
(Name)	CustomerID
Allow Nulls	False
Collation	
Computed Column Specification	
Data Type	int
Default Value or Binding	
Description	
Full Text Specification	False
Identity Specification	True
(Is Identity)	True
Identity Increment	1
Identity Seed	1
Is Column Set	False
Is File Stream	False
Is ROWGUID Column	False
Is Sparse	False
Not For Replication	False
Primary Key	True

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

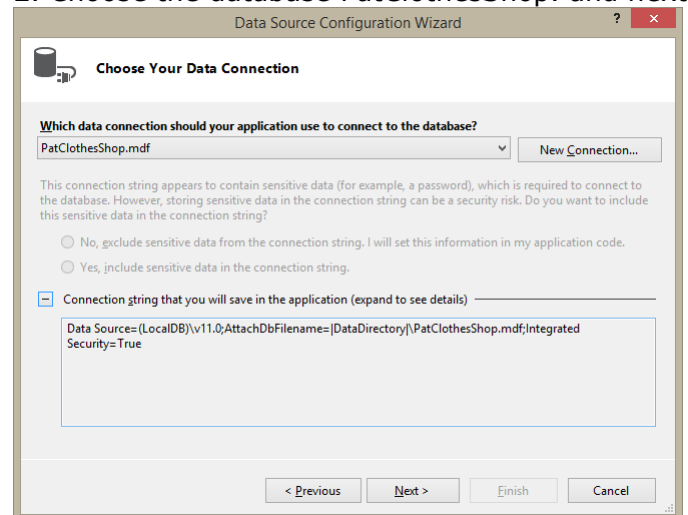


On the menu bar, choose View, Other Windows, Data Sources (or choose the Shift+Alt+D keys). Follow the steps.

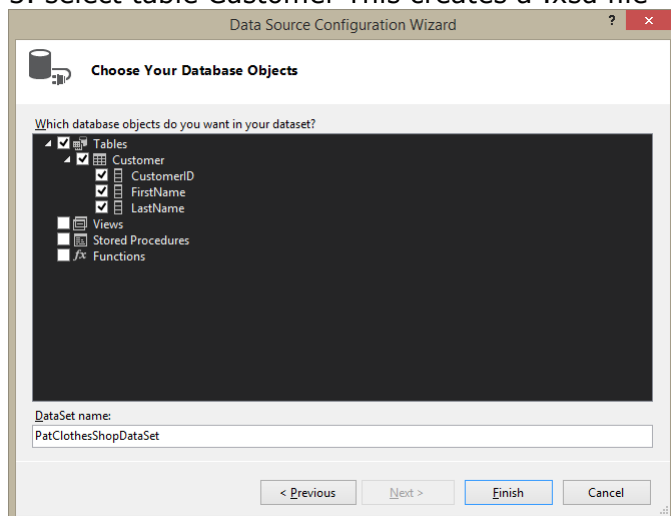
1. Choose Database and next



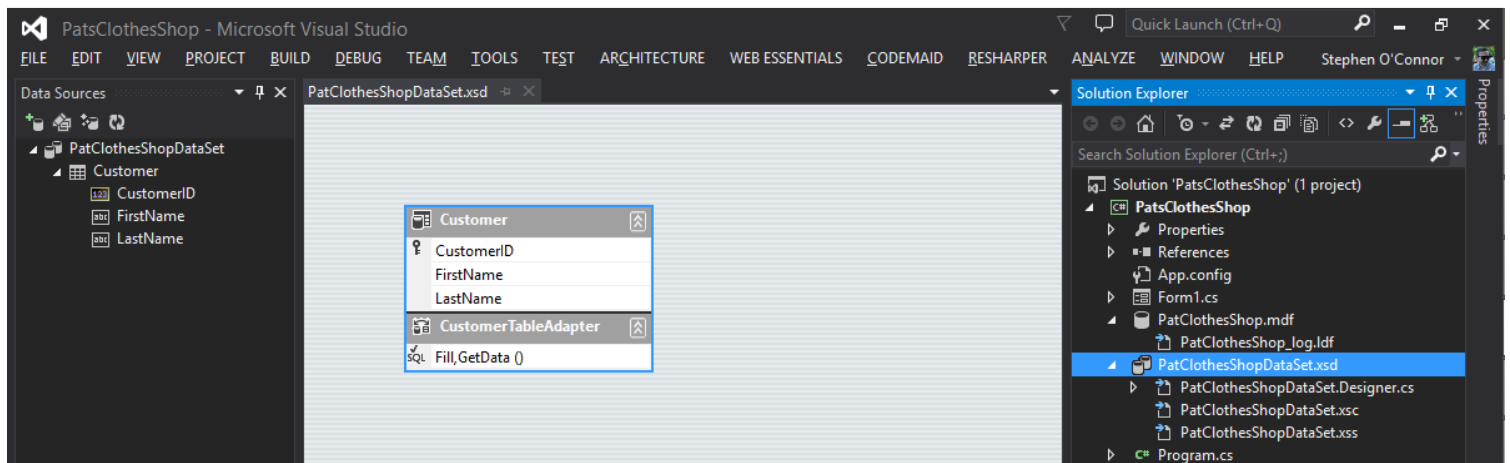
2. Choose the database PatClothesShop. and next



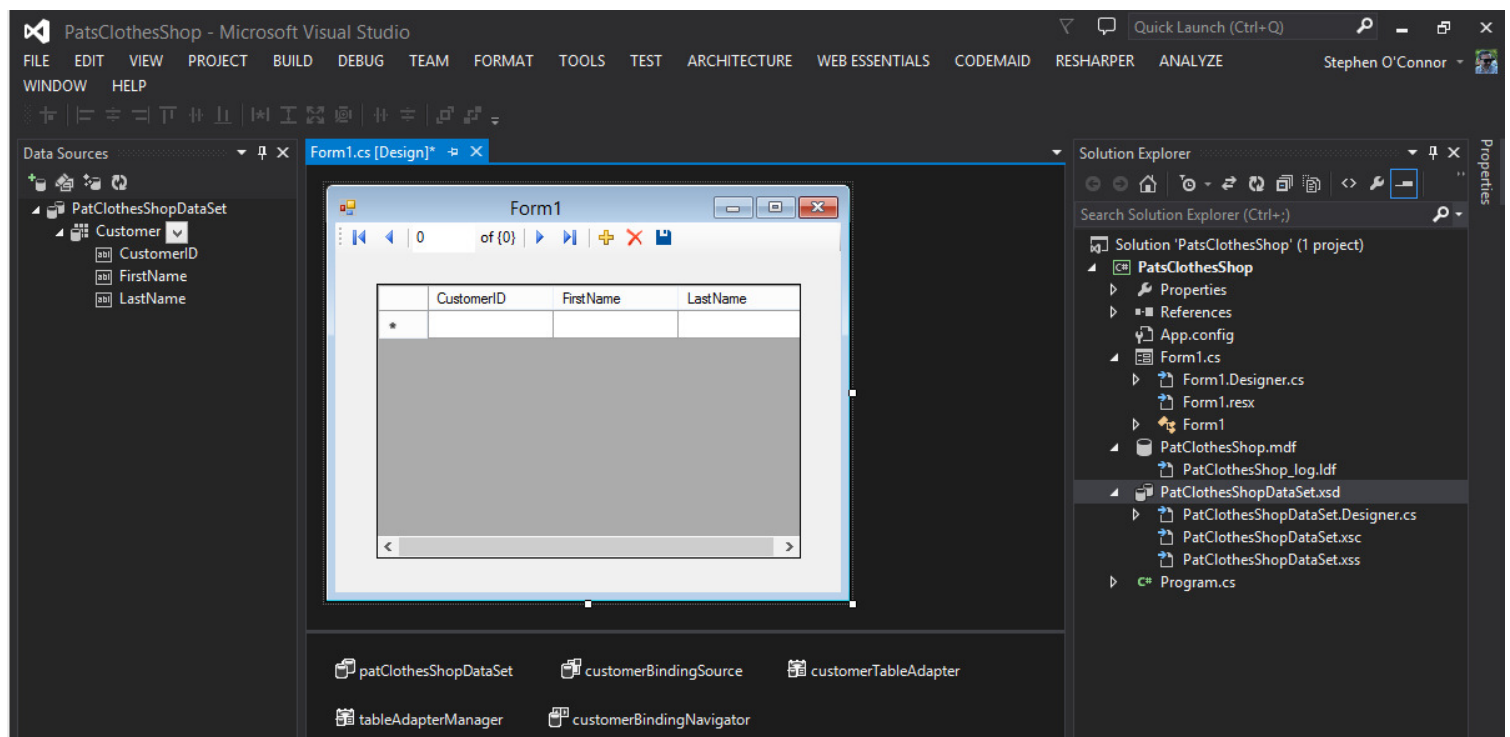
3. select table Customer This creates a .xsd file



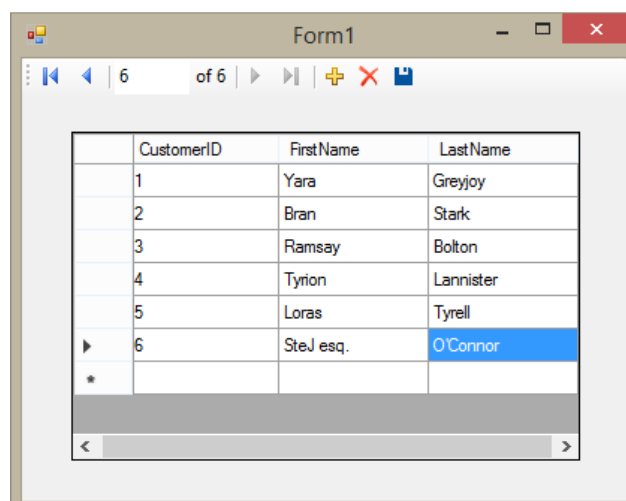
A PatClothesDataSet.xsd file right click on the xsd file and view designer mode. Xsd file is the xml schema document. The xsd file a local copy of database, this file defines the database, temporary stores the data.



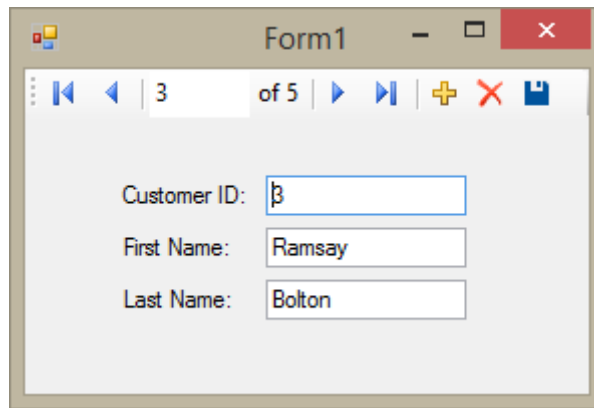
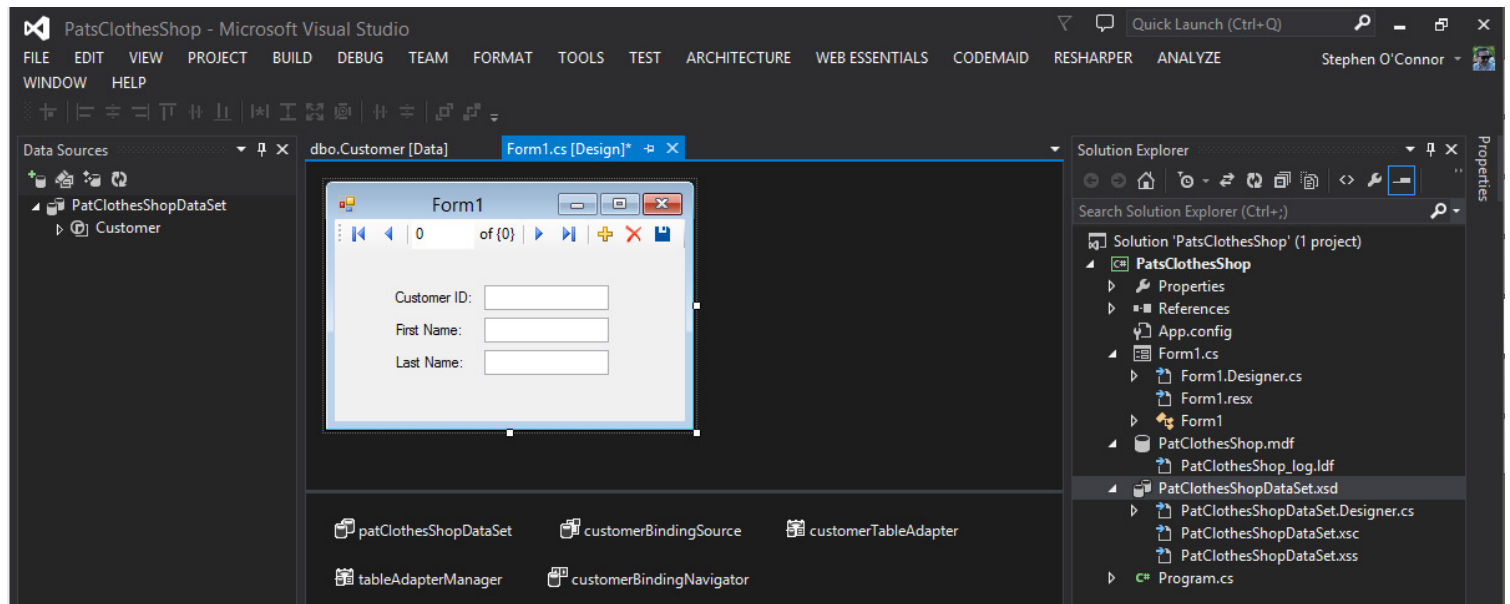
Drag and drop customer table from the Data Sources toolbar.



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



To designer tray.

patClothesShopDataSet

the local container for the data. Once the form is open the dataset gets populated The dataset gets populated Temporary storage container. Updated deleted

patClothesShopBindingSource

Object /bridge keep all of the controls on the form go to next row information in the dataset and the row of data in the DataSet. Keep all of the controls of the form binded to the rows within the dataset. Coordinates wha row of data should be currentlt displayed. The user wants to go to the next row

patClothesShopTableAdapter

Retrives data from the database, it contains a connection to the database.

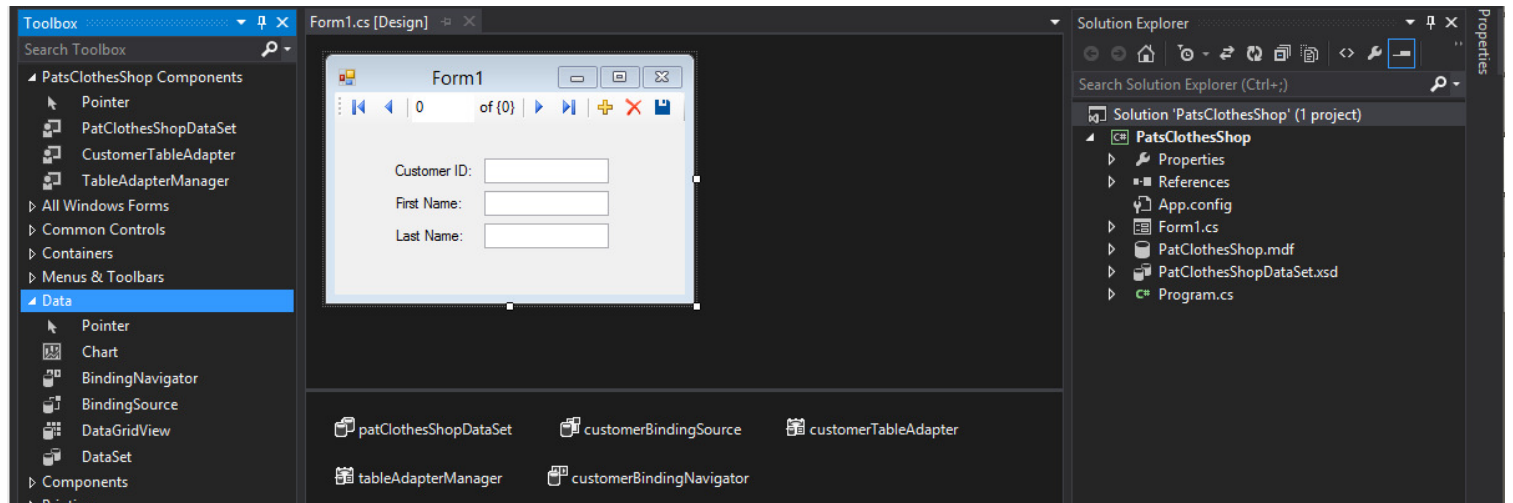
patClothesShopAdapterManager

Service interface

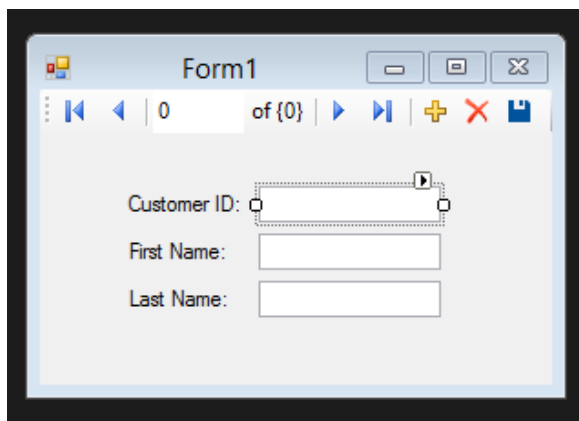
patClothesShopBindingNavigator

Toolbar at the top of the form.

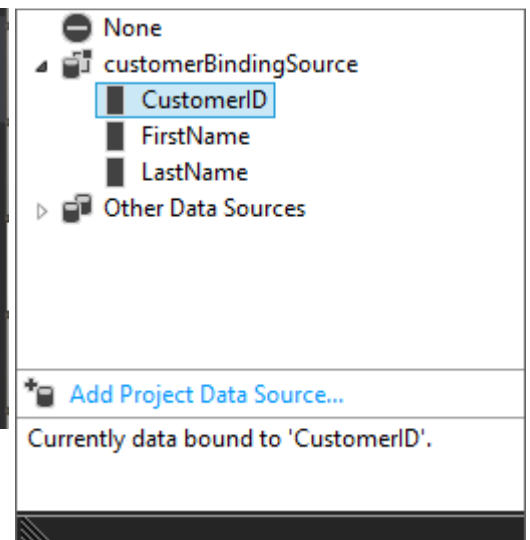
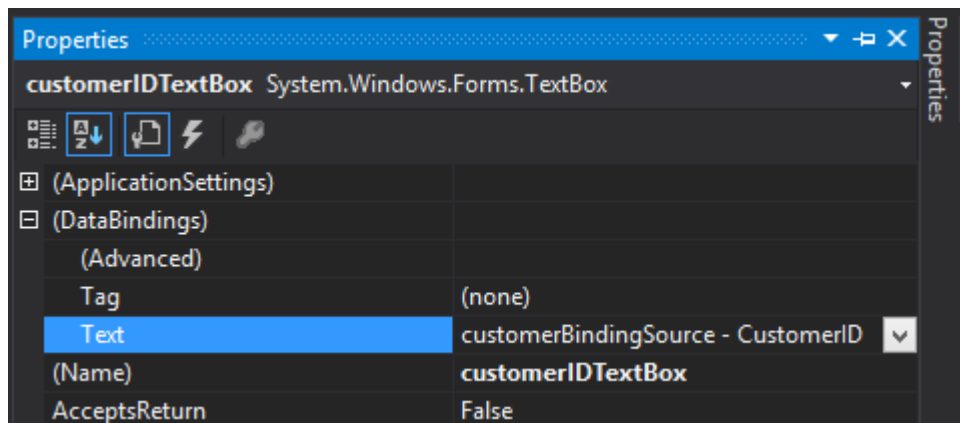




Properties of the first textbox. Binding source schema document. Dragging and dropping sets automatically.



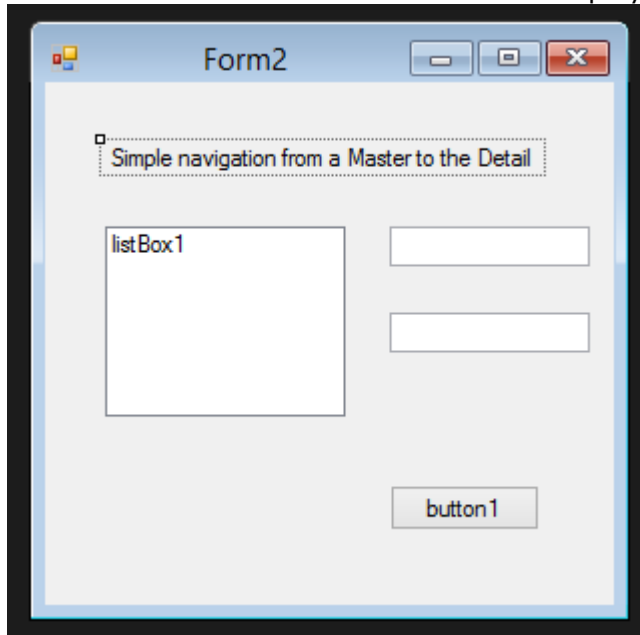
Size	100, 20
TabIndex	2
TabStop	True
Tag	
Text	
TextAlign	
UseSystemPasswordChar	False
UseWaitCursor	False
Visible	True
WordWrap	True



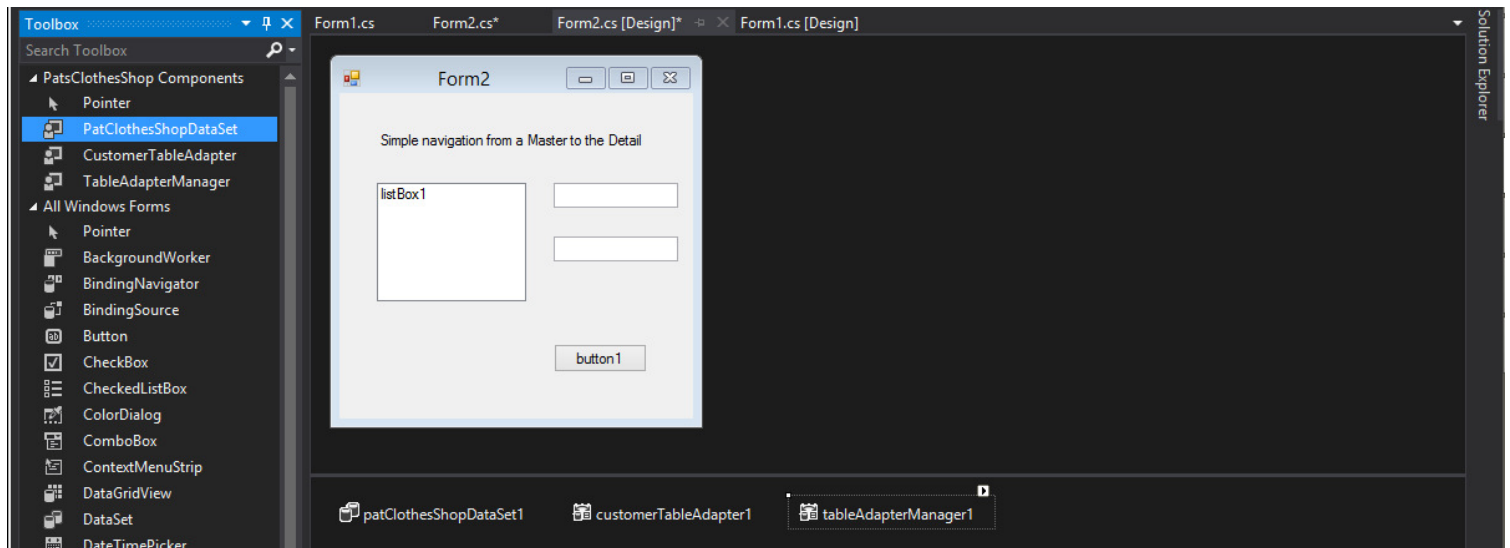


Add the data items manually.  
Strongly typed components or preconfigured.

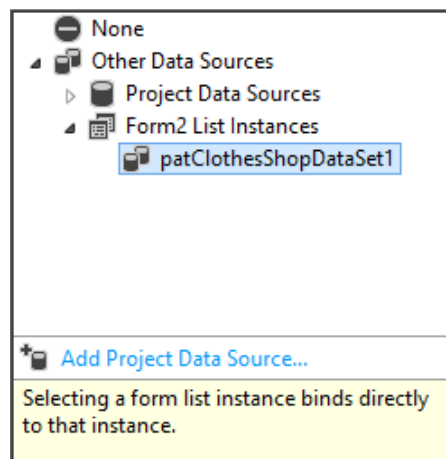
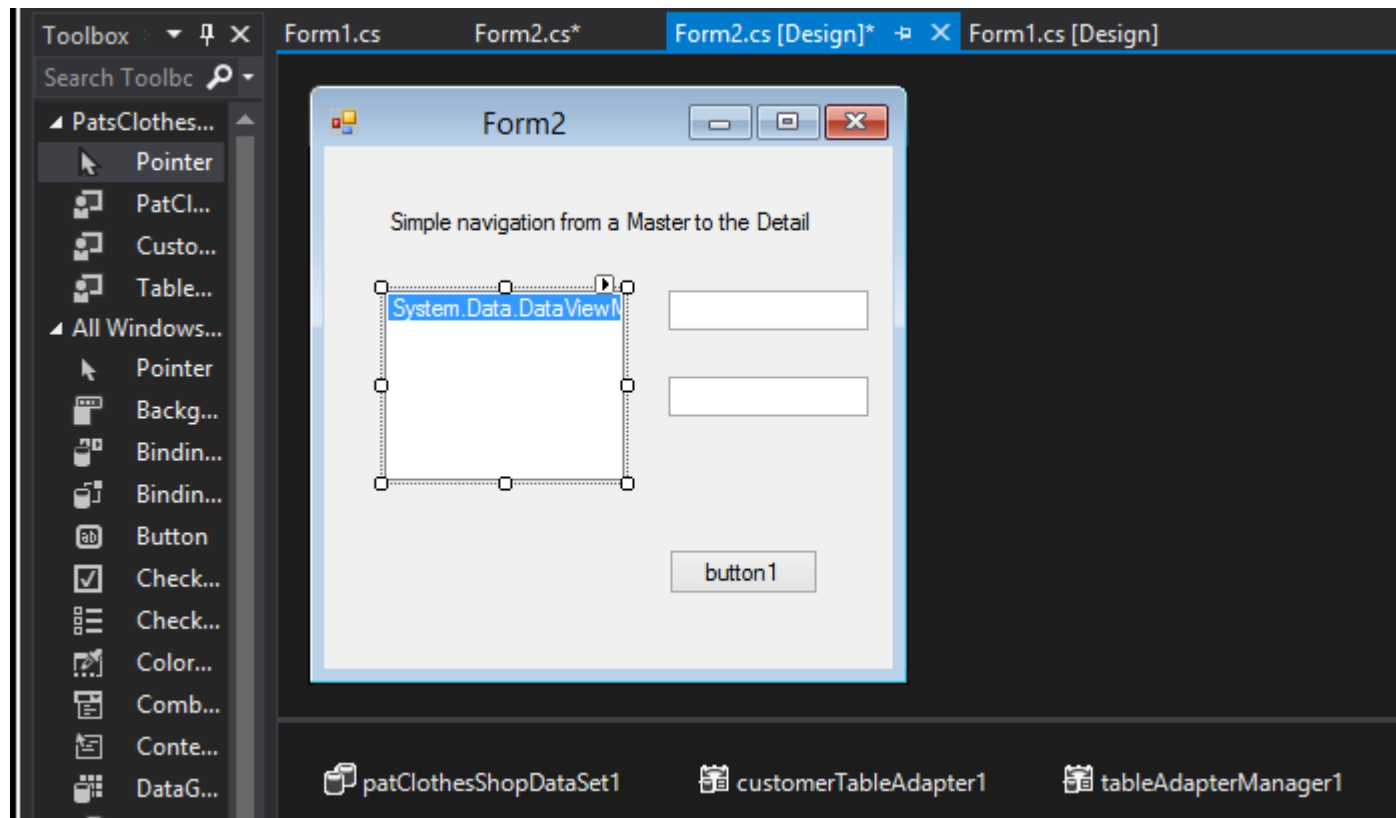
Create a new form form2 add the items displayed in the below image.



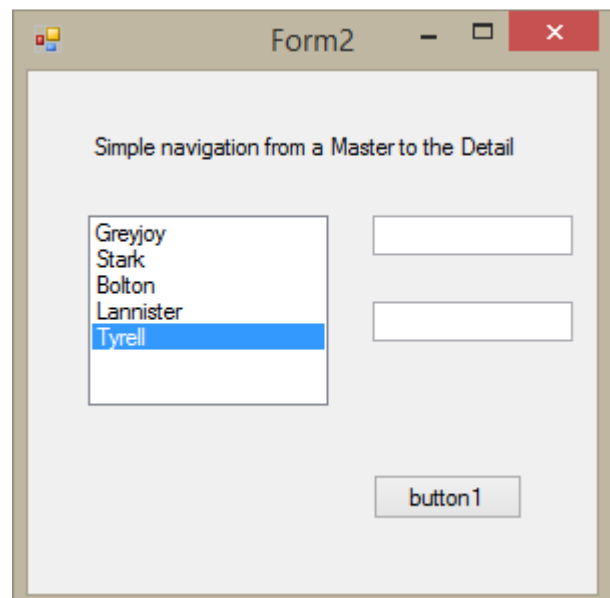
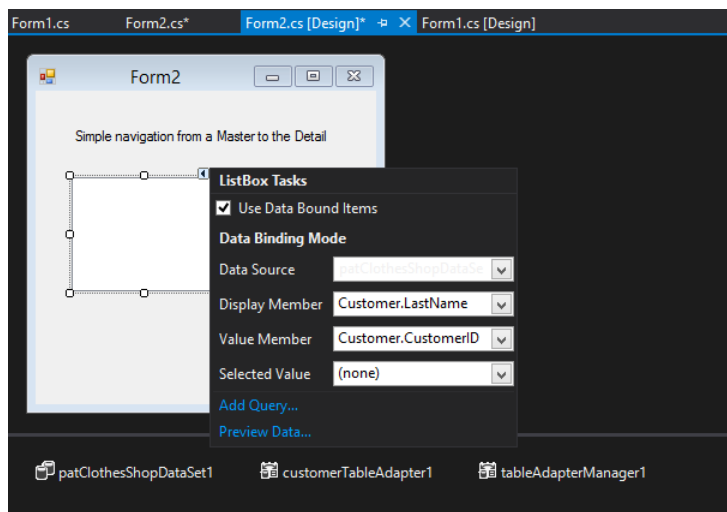
Add a button to form1 to open form2.



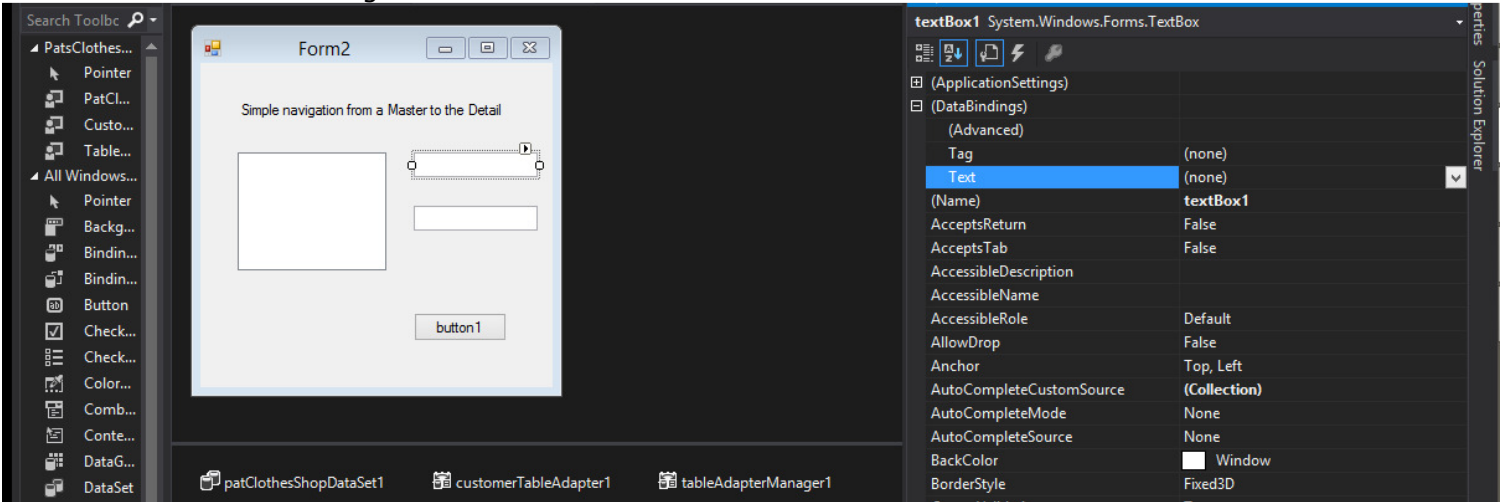
```
private void Form2_Load(object sender, EventArgs e)
{
    customerTableAdapter1.Fill(patClothesShopDataSet1.Customer);
}
```



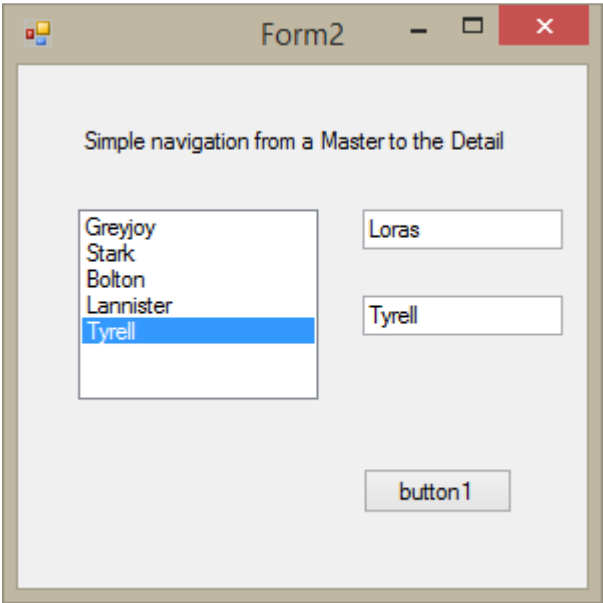
Filled the dataset with data



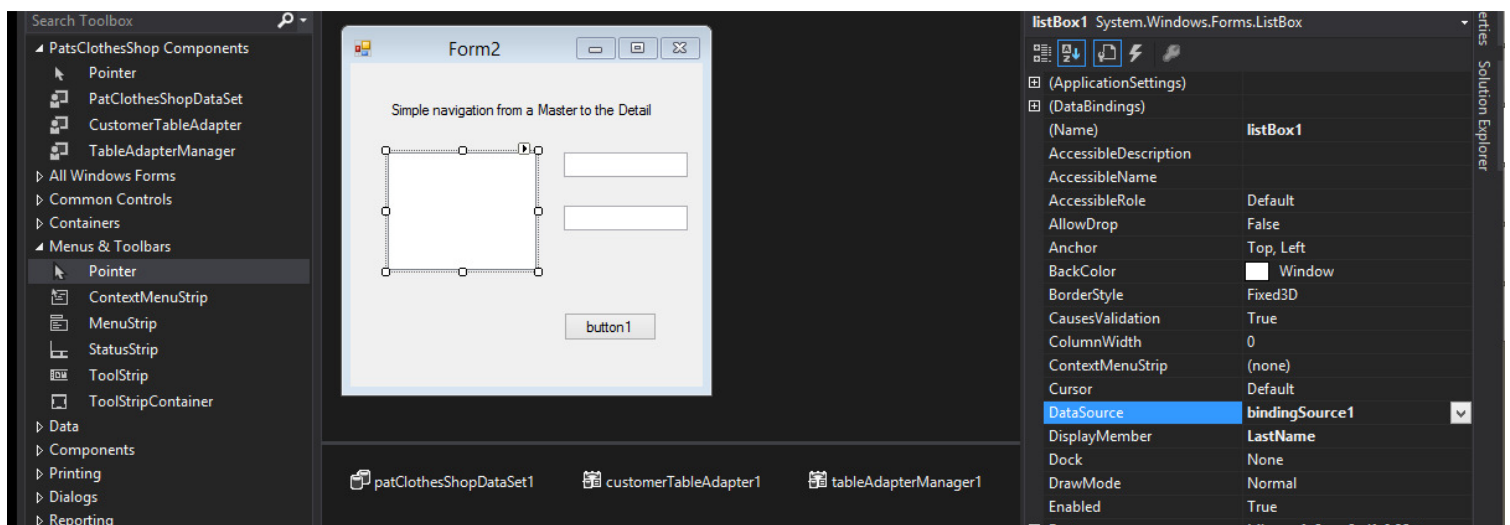
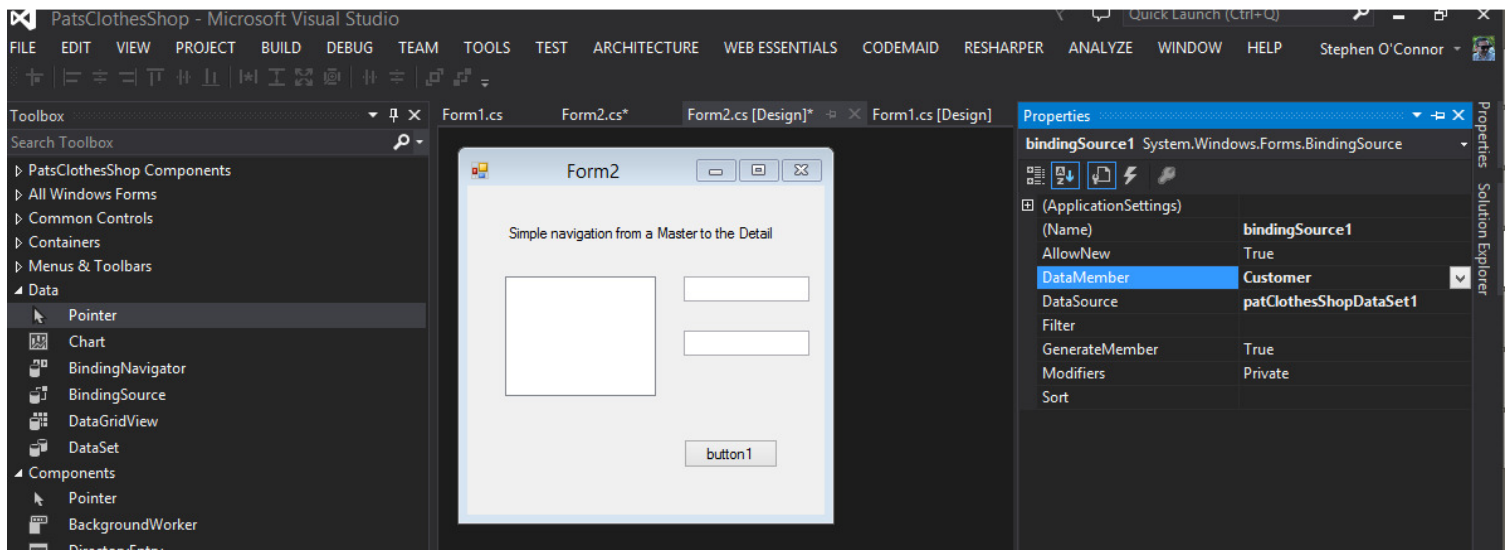
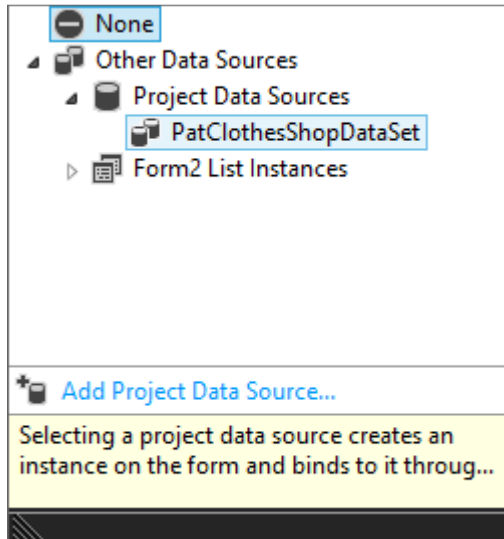
Bind the textboxes on the right with the listbox on the left.



<p>None</p> <p>Other Data Sources</p>	<p>None</p> <p>Other Data Sources</p> <p>Project Data Sources</p> <p>Form2 List Instances</p> <p>patClothesShopDataSet1</p> <p>Customer</p> <p>CustomerID</p> <p>FirstName</p> <p>LastName</p>	<p>None</p> <p>Other Data Sources</p> <p>Project Data Sources</p> <p>Form2 List Instances</p> <p>patClothesShopDataSet1</p> <p>Customer</p> <p>CustomerID</p> <p>FirstName</p> <p>LastName</p>
<p>Add Project Data Source...</p> <p>Select a data source under 'Other Data Sources' to connect to data.</p>	<p>Add Project Data Source...</p> <p>Selecting a property of a form list instance binds directly to that property.</p>	<p>Add Project Data Source...</p> <p>Currently data bound to 'LastName'.</p>



## Updating capabilities.



None

bindingSource1

Other Data Sources

Add Project Data Source...

Currently data bound to 'bindingSource1'.

Value member

None

CustomerID

FirstName

LastName

## My Set-up

The screenshot is divided into two main sections. The left section displays the GitHub profile of Stephen J O'Connor (Stevo5o). The profile includes a profile picture of a man with sunglasses, his name, and his GitHub handle. Below this, it lists his location (Dublin), email address (stev5o.jc@gmail.com), and the date he joined GitHub (January 3, 2013). It also shows his statistics: 21 followers, 25 repositories starred, and 13 repositories he is following. The main part of the profile is divided into three tabs: Contributions, Repositories, and Public activity. The Contributions tab is active, showing a grid of green squares representing contributions over time. Below the grid, it summarizes the number of pull requests, issues opened, and commits. The right section of the screenshot shows a terminal window with a PowerShell prompt. The user has entered the command 'git add .' to stage all changes, followed by 'git commit -am "Lesson Plan clean up"' to commit the changes. The terminal output shows the commit was successful, with 1 file changed, 0 insertions, and 0 deletions. The user then enters 'git push origin master' to push the changes to the master branch. The terminal output shows the push was successful, with 1 file changed, 0 insertions, and 0 deletions. The user then enters 'git status' to check the status of the repository, and the output shows that the repository is clean and up-to-date.