Creating the List - Data Layer

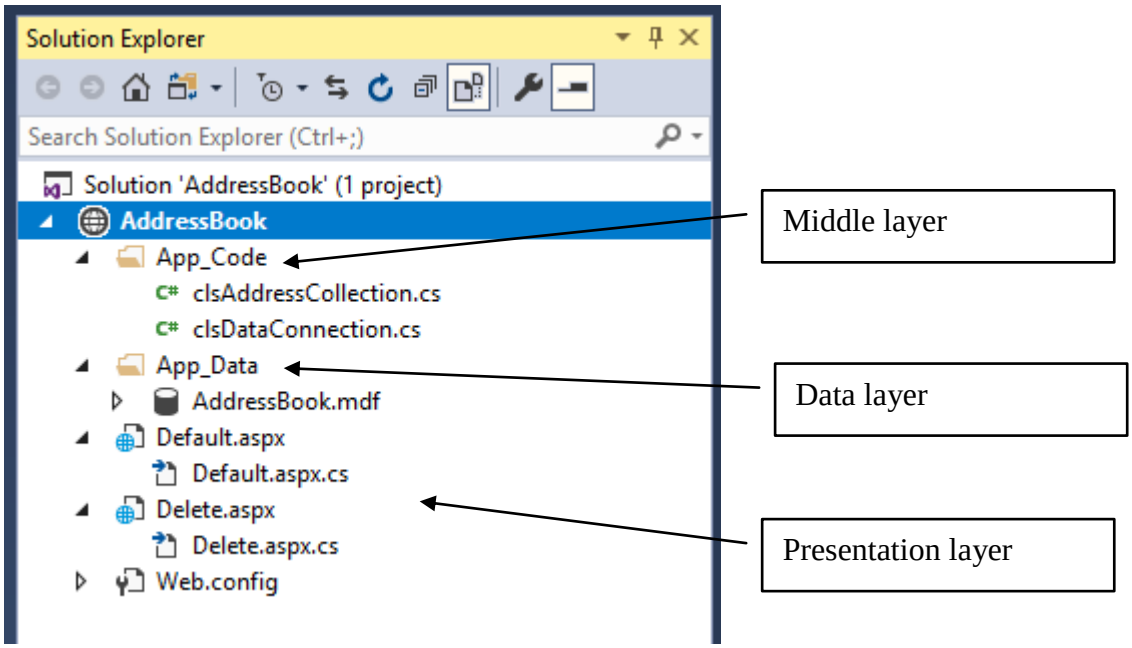
If you have not done so already you should spend a little time looking at the functionality provided by the finished address book sample site to give you an idea of what you need to do.

To complete this work, you will be extending the functionality developed so far.

Three Layer Design

The work completed so far should allow you to enter the primary key value for a record and delete that record (assuming it exists and the value is a valid numeric value!)

The system so far should have something like the following architecture...



So far, we have accepted a single parameter value from a text box, passed that value to the middle layer and used a stored procedure in the data layer to delete the specific record.

For this work we shall look at the procedure for processing a set of records from a table and displaying those records in a list box on the web form.

The following data in SQL server...

dbo.tblAddress [Data]

Max Rows: 1000

AddressNo	HouseNo	Street	Town	PostCode	CountyCode	DateAdded	Active
1	1		Some Street	LE1 1BE	34	07/08/2013	True
2	22		The Road	N19 6FF	48	07/08/2013	True
3	33	The Mews	High Street	LE1 6FG	34	07/08/2013	True
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Is displayed on the web form like so...

To achieve this, we are going to have to use a set of programming tools that allow us to model, manipulate and display the data in the table. We will use different tools depending on what layer we are working on.

- To model the data in the middle layer we shall use an object called a data table
- To model the data in the presentation layer we shall use an array list
- To display the data to the user we shall use an object called a List Box

We will look at each of the above over the next few weeks.

Modifying the Data layer

For any of this functionality to work we need to create a stored procedure in the data layer that will drive the process.

Create a new query using the following SQL...

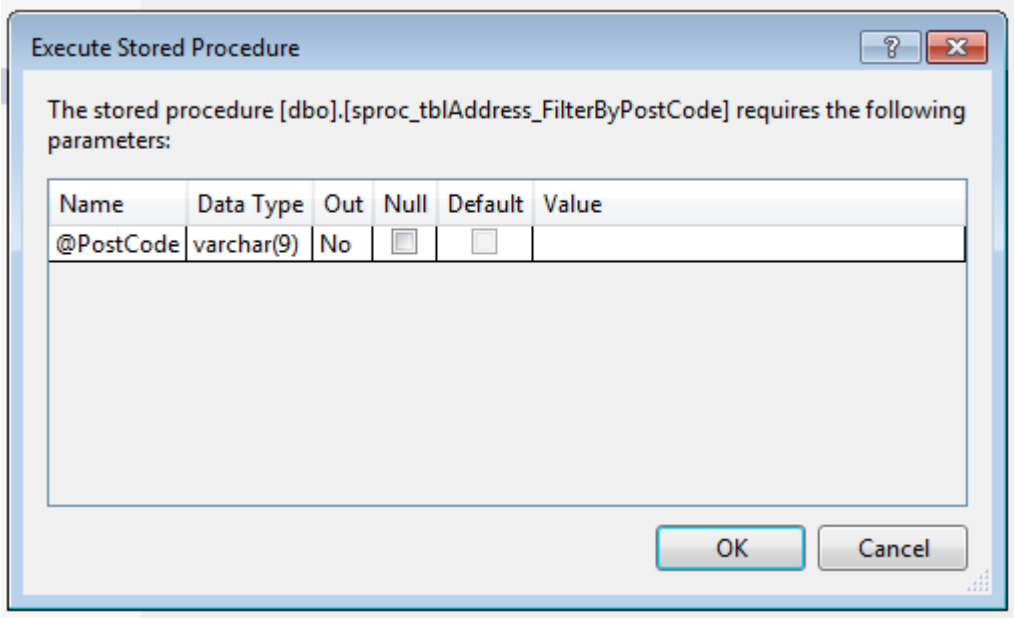
```
CREATE PROCEDURE sproc_tblAddress_FilterByPostCode
    --parameter for post code
    @PostCode varchar(9)

AS
    --get all records where PostCode is like the value passed as a parameter
    SELECT *
    FROM tblAddress
    WHERE PostCode Like @PostCode + '%';

RETURN 0
```

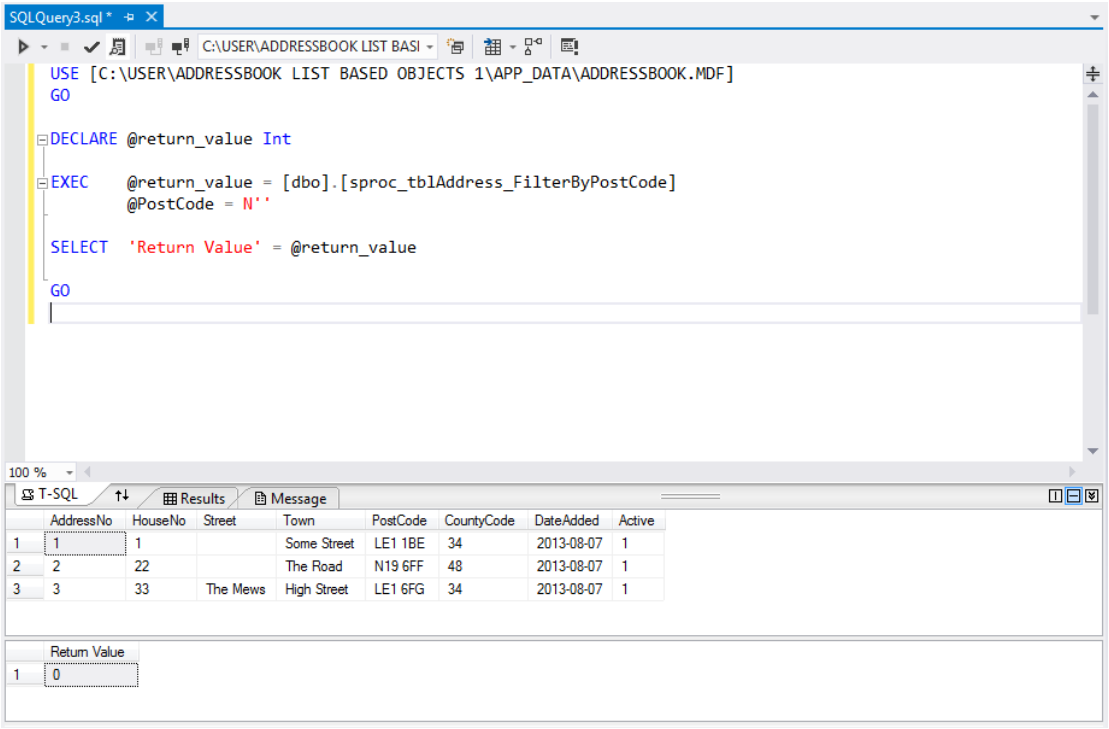
Save the stored procedure as sproc_tblAddress_FilterByPostCode and run it to see what happens.

Upon running the query, you will be asked for a value to the parameter @PostCode...

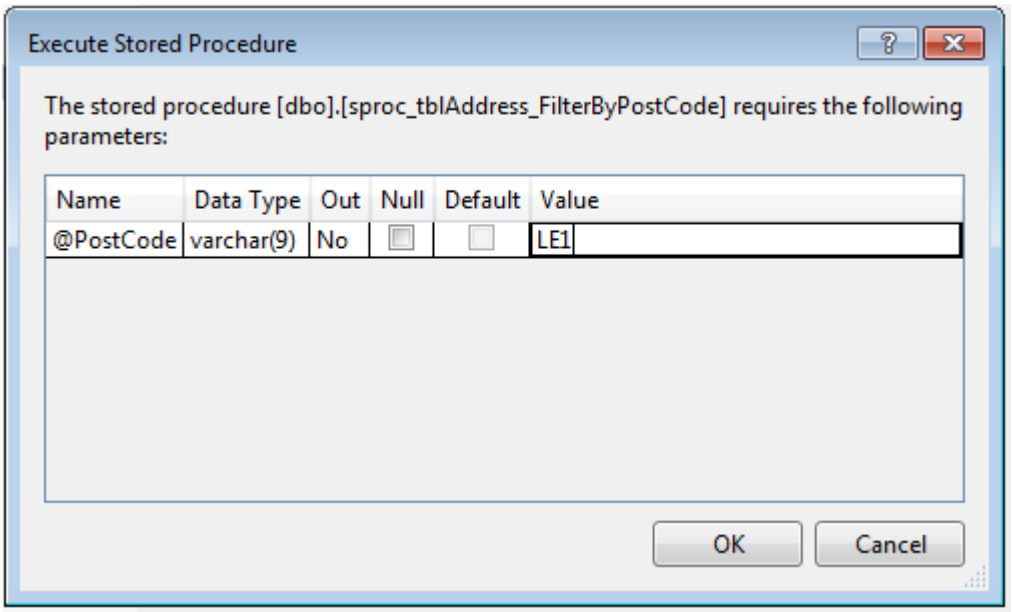


Leave it blank on the first run and press OK.

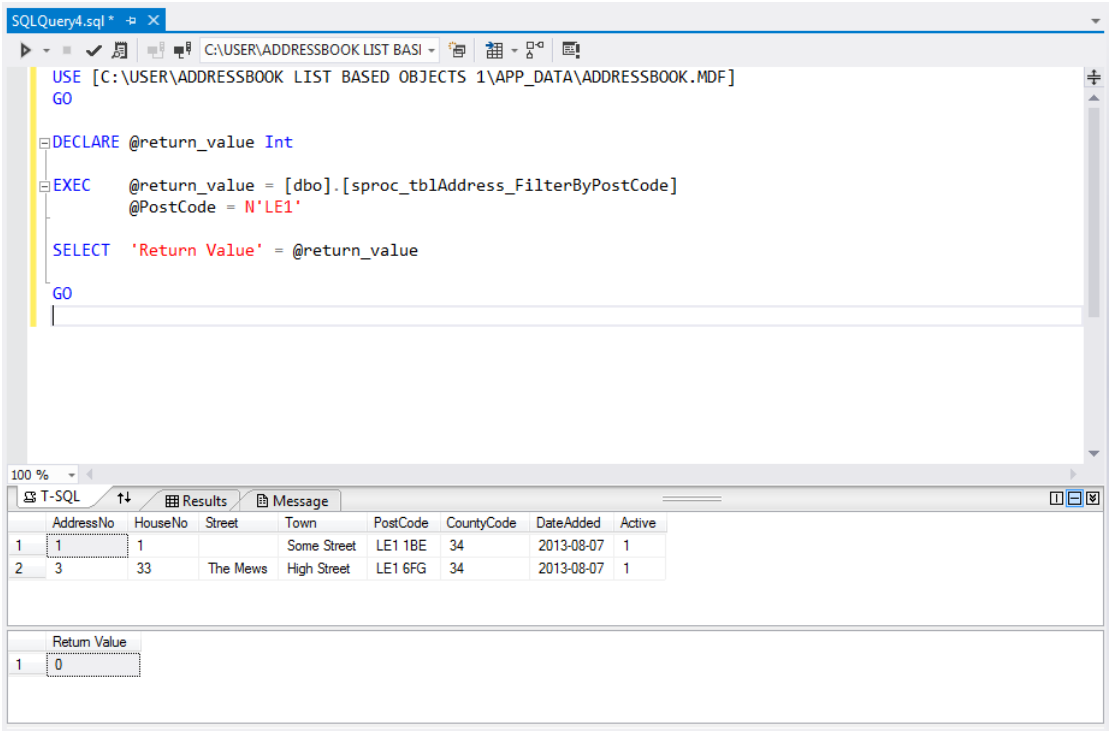
You should see the following output for the query...



Close the output and run it again, this time entering LE1 as the value of the parameter...



What you should see is only the data with a post code starting with LE1...



The Like SQL Function

Like is a useful SQL function that allows us to apply pattern matching.

For example, Like ("A%") would retrieve all values starting with 'A'.

Like("%E") would return all values ending in 'E'.

In the SQL used in this query we are concatenating the value of the parameter with the wild card character % using the + symbol.

```
WHERE PostCode Like @PostCode + '%';
```

So, when we enter 'LE1' as the value of the parameter the string concatenates to...

SELECT *
FROM tblAddress
WHERE PostCode Like ("LE1%");

The data layer is now complete. In the next session, we shall start to build the middle layer.