

## The New HCQU reporter Short and Sweet

A simple walkthrough using the HCQU reporter to create a query on a HCQU database

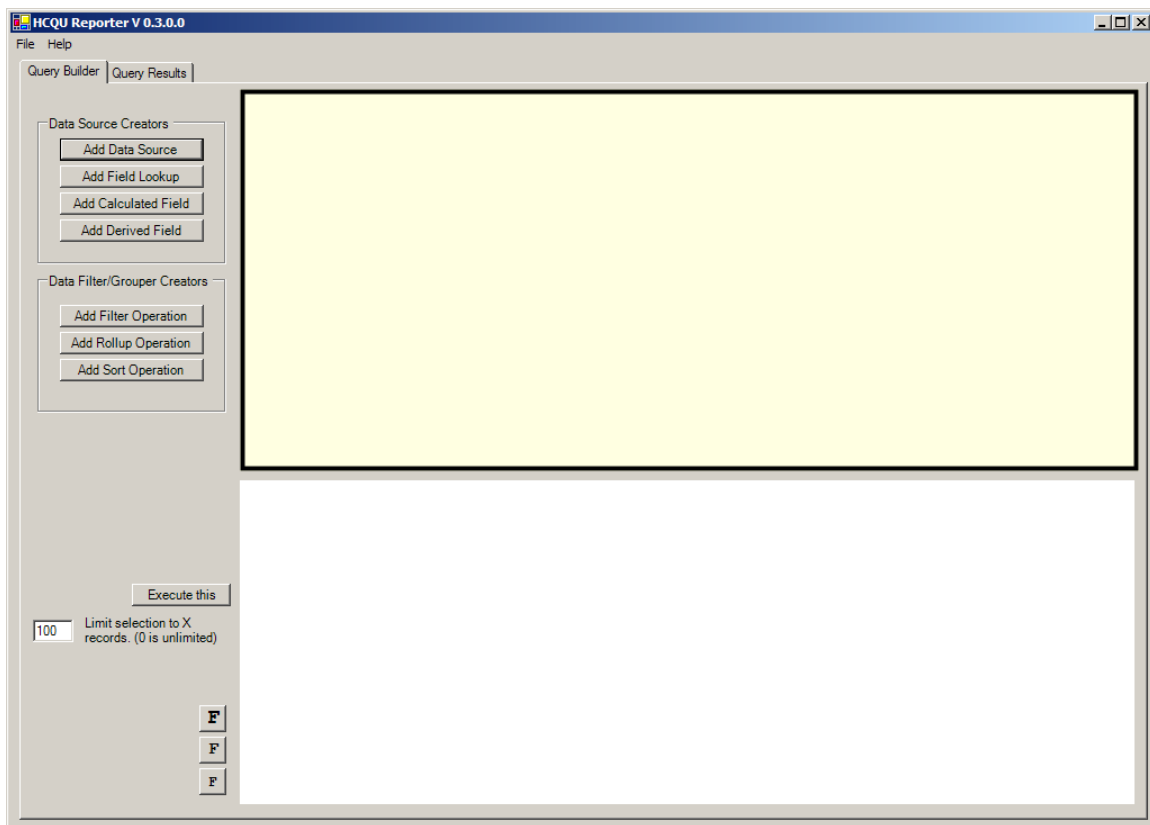
First edit the HCQUReporter.ini file that came along with the archive and change the following entries to point to your database environment

```
[HCQU Reporter initialization file]
DBASE=Driver={SQL Server};Server=(local);Database=SCRANTONHCQU;Trusted_Connection=Yes;
DRIVER={SQL Server}
SERVER=(local)
DATABASE=SCRANTONHCQU
PROVIDER=SQLOLEDB
```

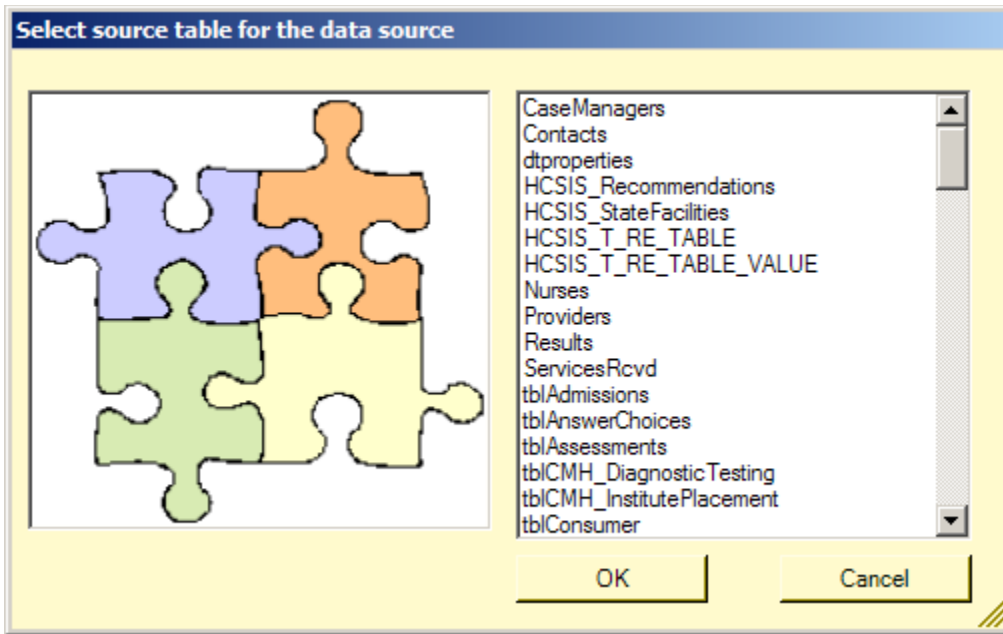
In the above case the (local) and the SCRANTONHCQU text entries need to be adjusted to meet your environment. (local) being replaced with the server name line TNSERVER1 for example, and SCRANTONHCQU reflecting that name of the database that contains the data you want to query.

Future versions will have a human interface of these settings to allow selecting any database and server combination visible to the end user.

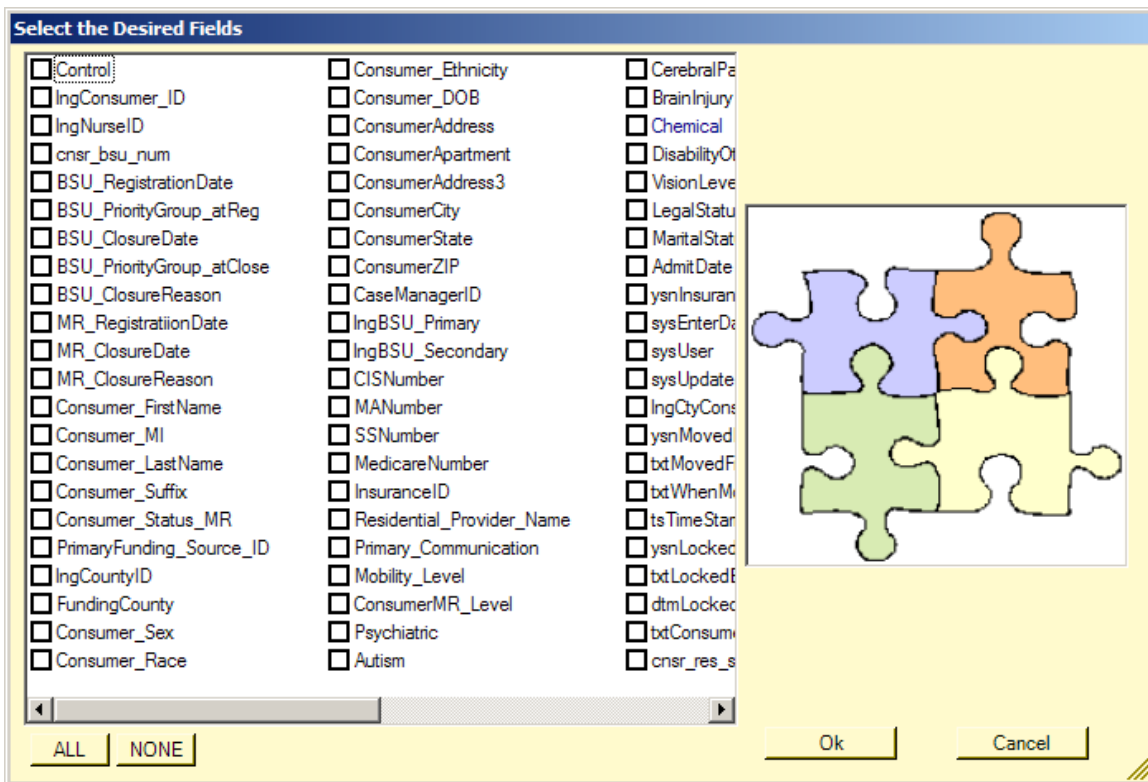
Now run the program and get the main screen.



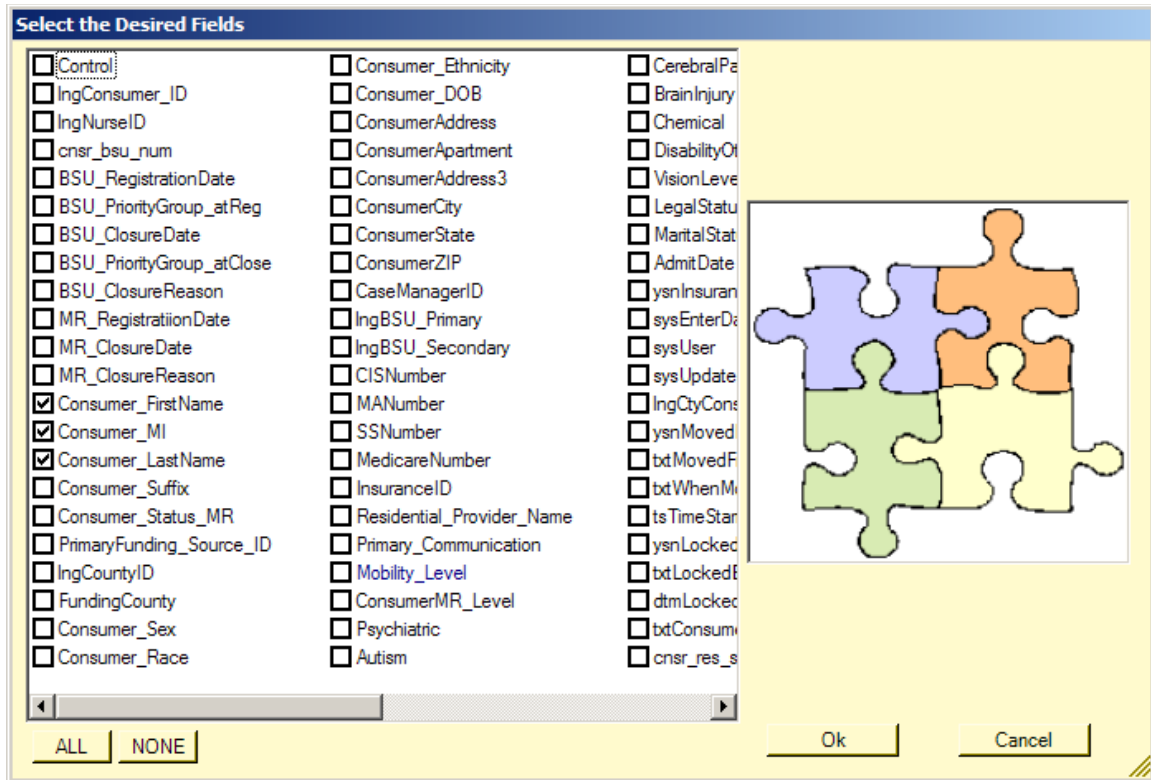
Click the Add Data Source button to bring up the table list dialog.



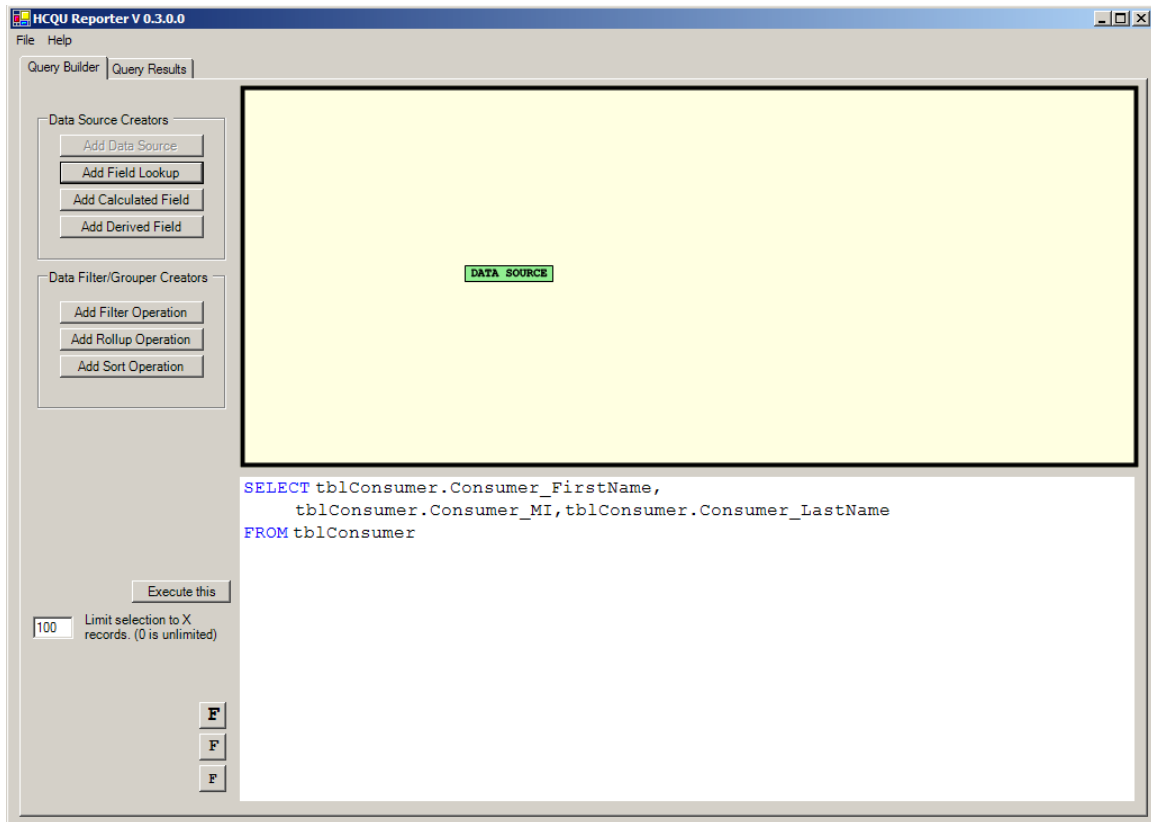
This dialog is the first in a short series that allows you to create your base query. Here you will see a list of all the tables that are in the database that we pointed at in the above steps. You can screen up and down to see the whole list of entries. For our example walkthrough we will select the tblConsumer table. Double click on this entry or select this entry and click OK. This brings up the field list dialog



The field list dialog is used to select the basic fields that you want to have in your resulting query brought verbatim from the table into your results. In our example we want to select the name of the consumer so we are going to select the Consumer\_firstname, Consumer\_MI, and the Consumer\_Lastname fields by clicking the checkboxes next to these field names in this dialog.



Now we will select the OK button to go back to our main screen with the basic query created.



Here we now see the simple DATA SOURCE object in our canvas and the simple query

```
SELECT tblConsumer.Consumer_FirstName,  
       tblConsumer.Consumer_MI, tblConsumer.Consumer_LastName  
FROM tblConsumer
```

Listed in the query code viewer, this code viewer also colorizes the syntactical elements of the resulting query code to assist the user in seeing what the tool is actually building for you.

Now we want to add some more complexity to this query. First we will examine the ability of the HCQU reporter to associate fields in the base table with values contained in other fields in other tables of the database. ( JOINS in SQL parlance )

To create the join we need four pieces of information.

The Field in the base table that represents what we are joining from ( or looking up from )

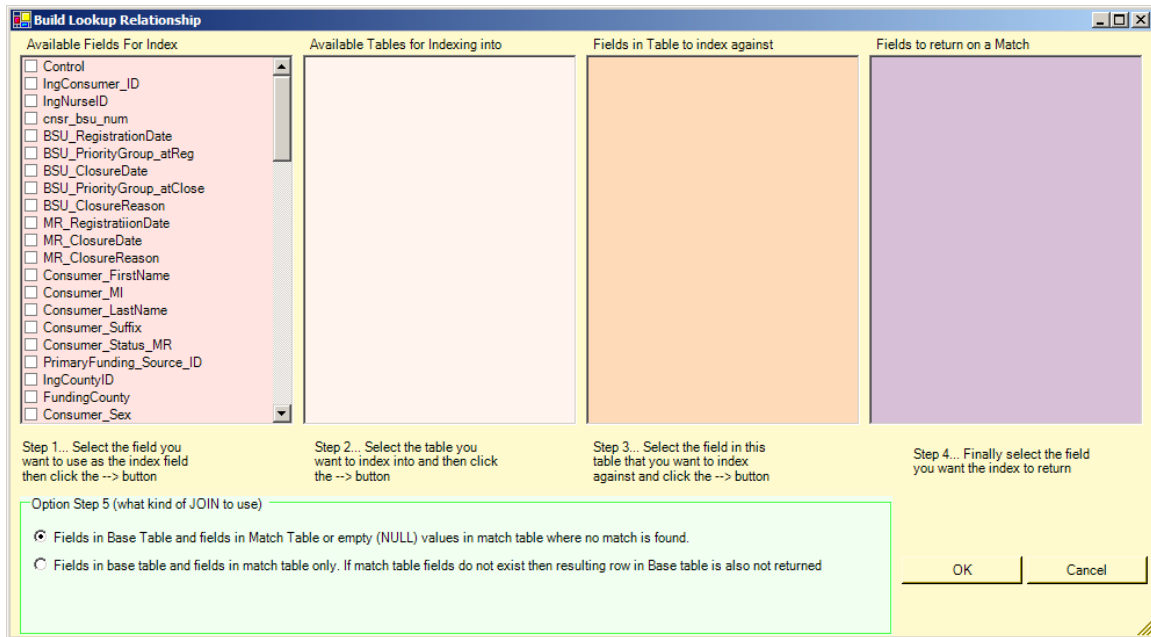
The Table that contains what we are looking into

The Field in this table that represents what we are associating with the field in the base table that we picked in step 1 above

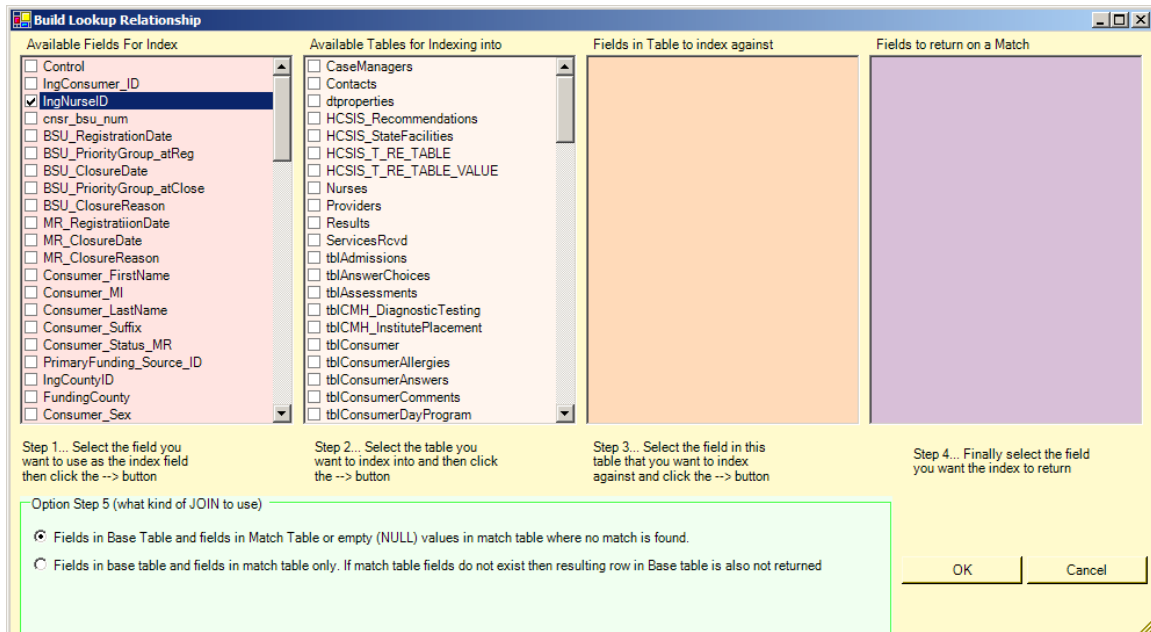
Finally the piece or pieces of data in this table we are looking into that we want to pull into our query.

In our example will use the nurseID field to point into the Nurses table and connect to the NurseID field in the nurses table. We will then pull back the nurses Firstname and Lastname fields in our join.

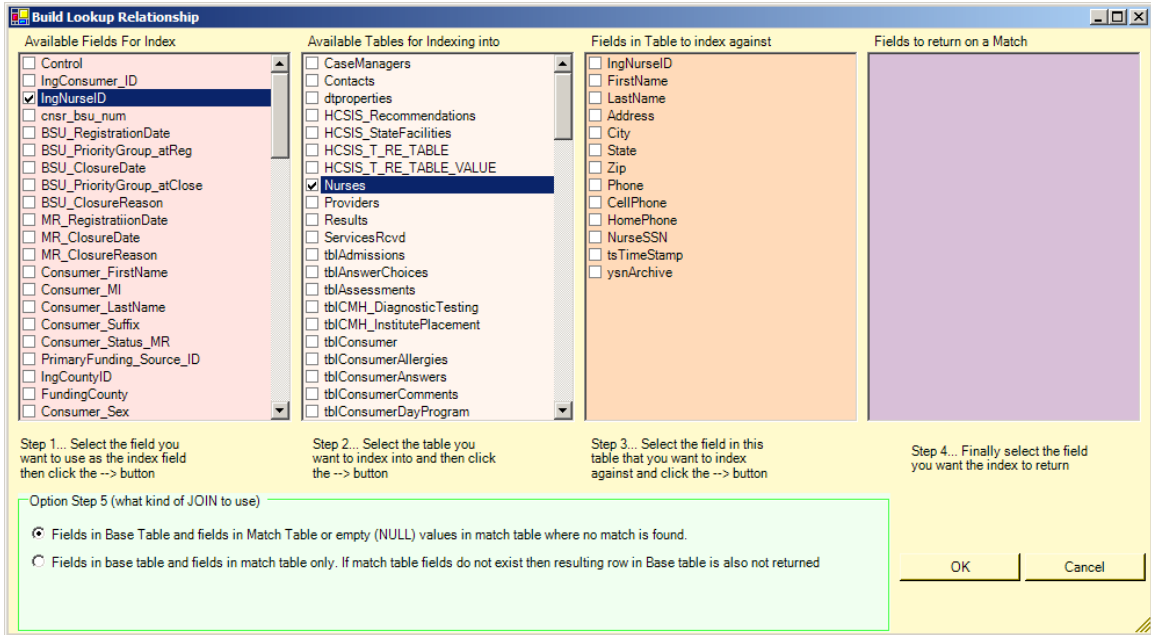
Click on the Add Field Lookup button to bring up the Add Field Lookup dialog



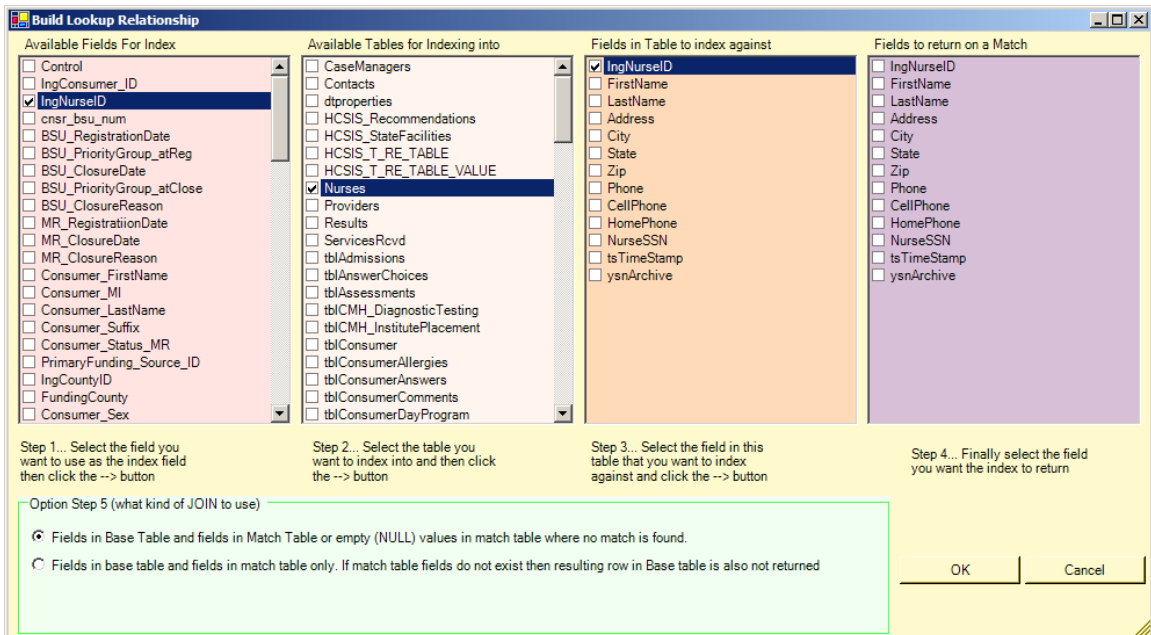
Select the lngNurseID field in the first list.



Select the Nurses table in the now filled in second list

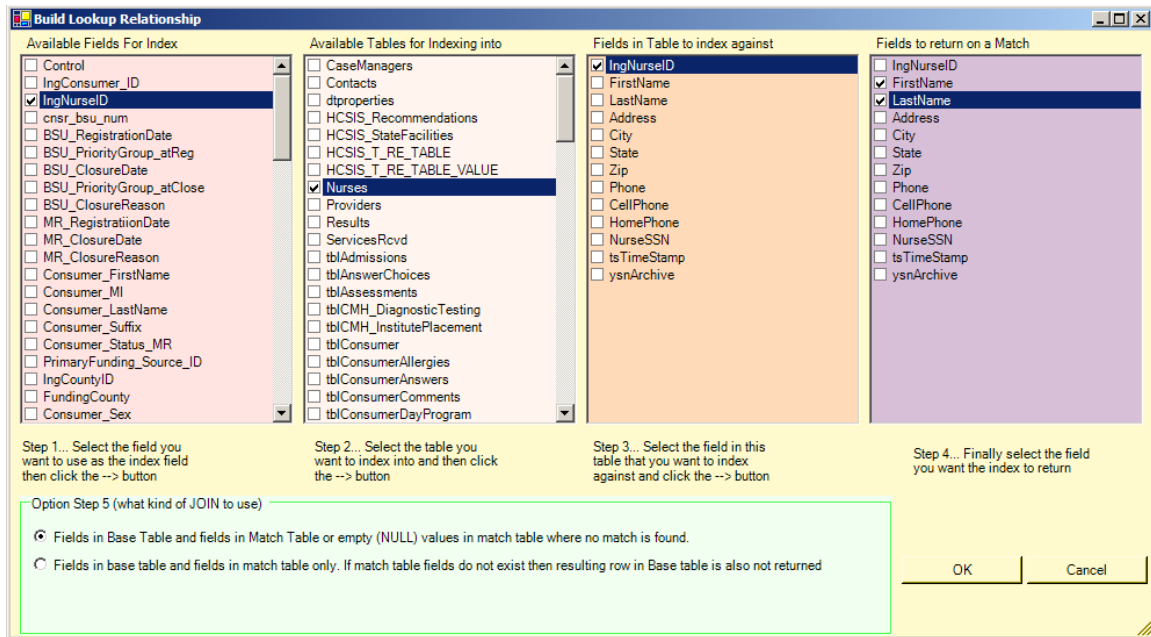


Select the IngNurseid field in the now filled in third list

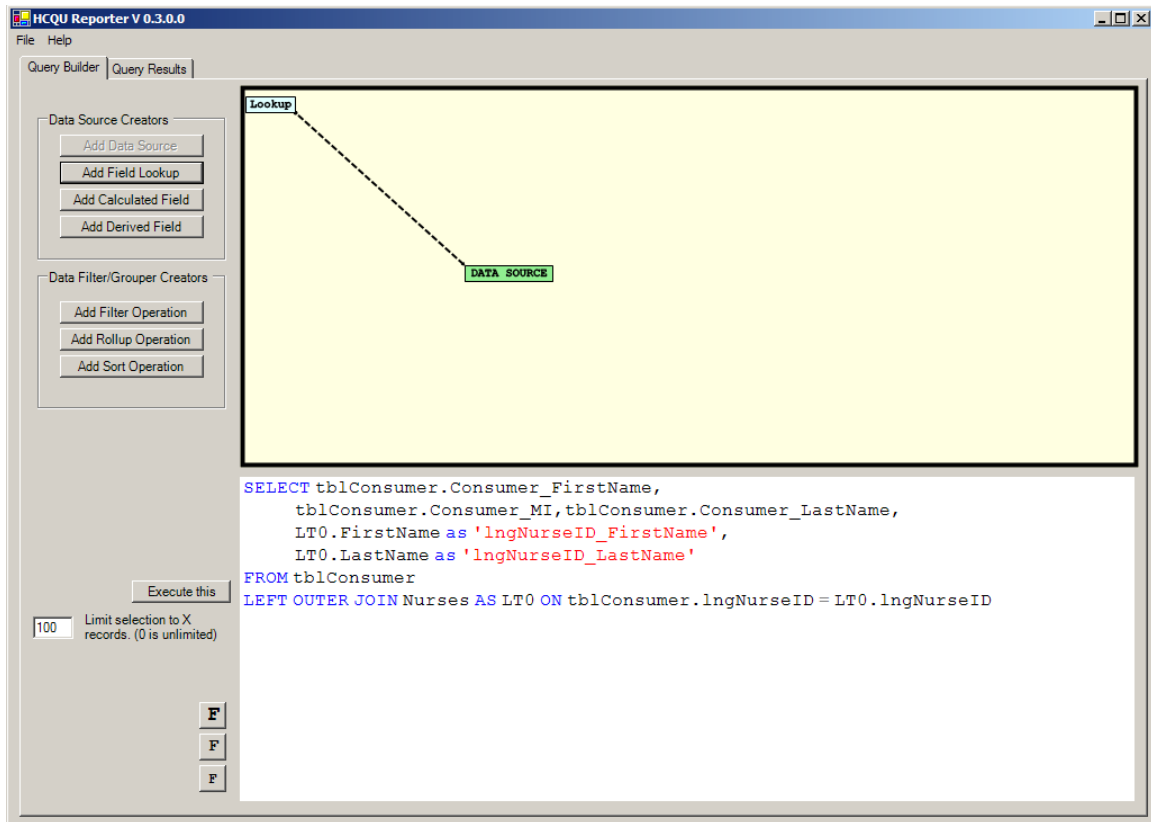


Finally select the fields you want to pull into our query in the final list that is now filled in.

Firstname and Lastname in our example

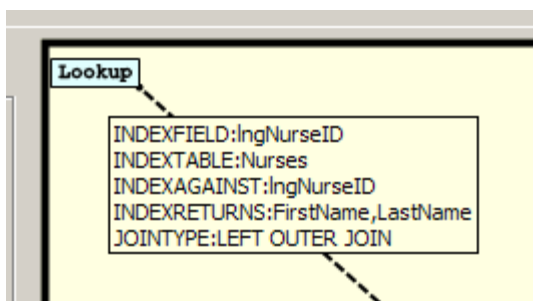


you also have the option to alter the kind of join being created. In most cases the want to create a scenario where values that are being used as lookups pull empty values if no match is made. In SQL parlance this is referred to as a LEFT OUTER JOIN, and this is the default kind of join that the software creates. If you wanted to create a more restrictive join you would select the other join type in the optional step 5 at the bottom of this dialog. In our case we want the less restrictive join so simple click the OK button to add our new join to the resulting query that is being built.



You can now see a Lookup has been added and it has been connected to our data source by a dashed line. The dashed line indicates a loose join exists between the lookup and min data source.

If you hove over the lookup icon in the canvas you will see some popup dialogs that show the particulars of the lookup that we created in the above dialogs



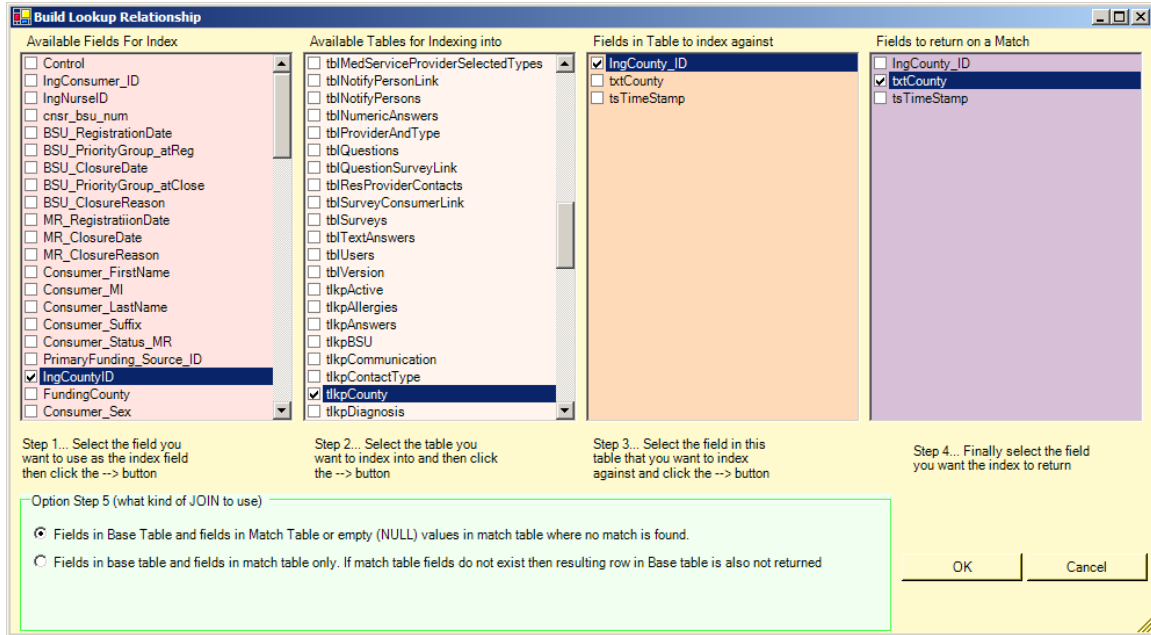
Here we can see the fields used on both side of the join, what table we are joining with the name table. What fields we are pulling into our query through the join, and what kind of join we are using.

You can also see the results of this new join on the SQL code window.



Now lets quickly grab one more field in lookup. Lets use the lookup function to convert the COUNTY ID value in the consumer table to a more readable County name by building a lookup on the county table nd pull the human readable county description from that table.

Click on the Add Field Lookup again



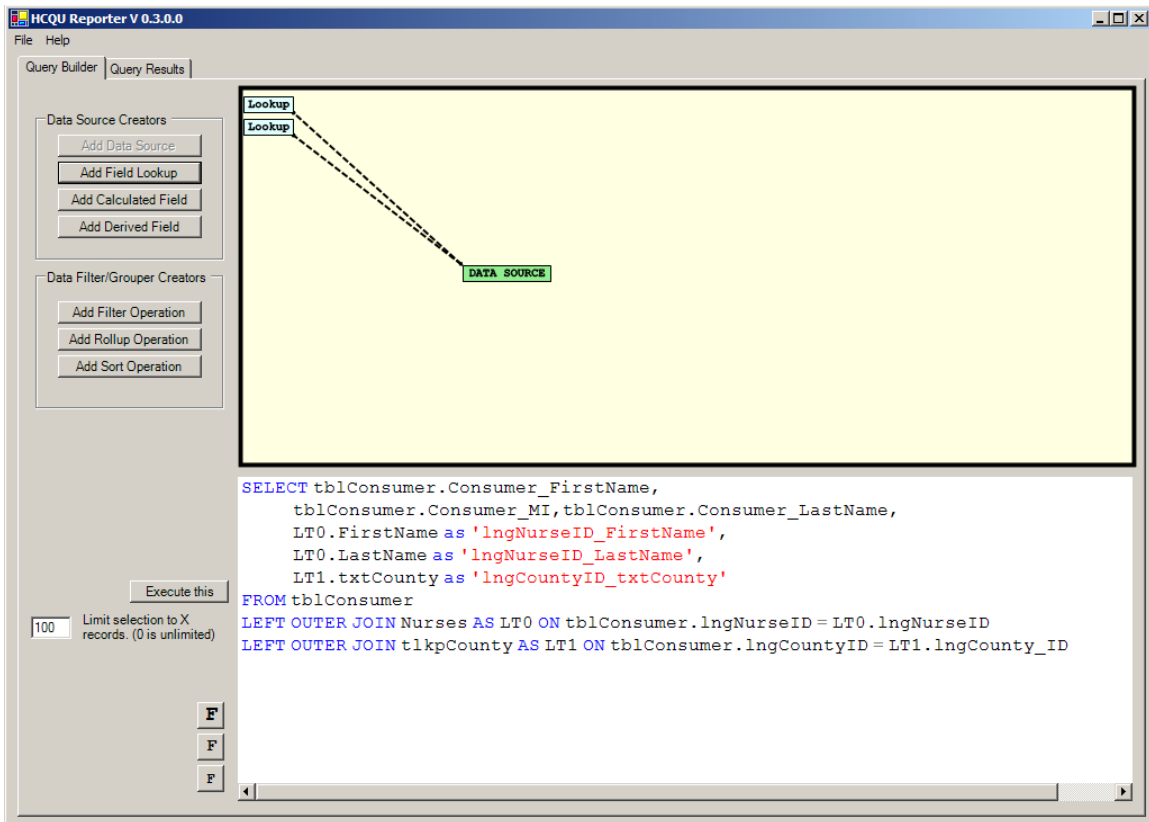
Here I selected the lngCountyID field to start my lookup.

Selected the tlkpCounty table as my lookup table

Selected lngCountyID as my field to lookup against

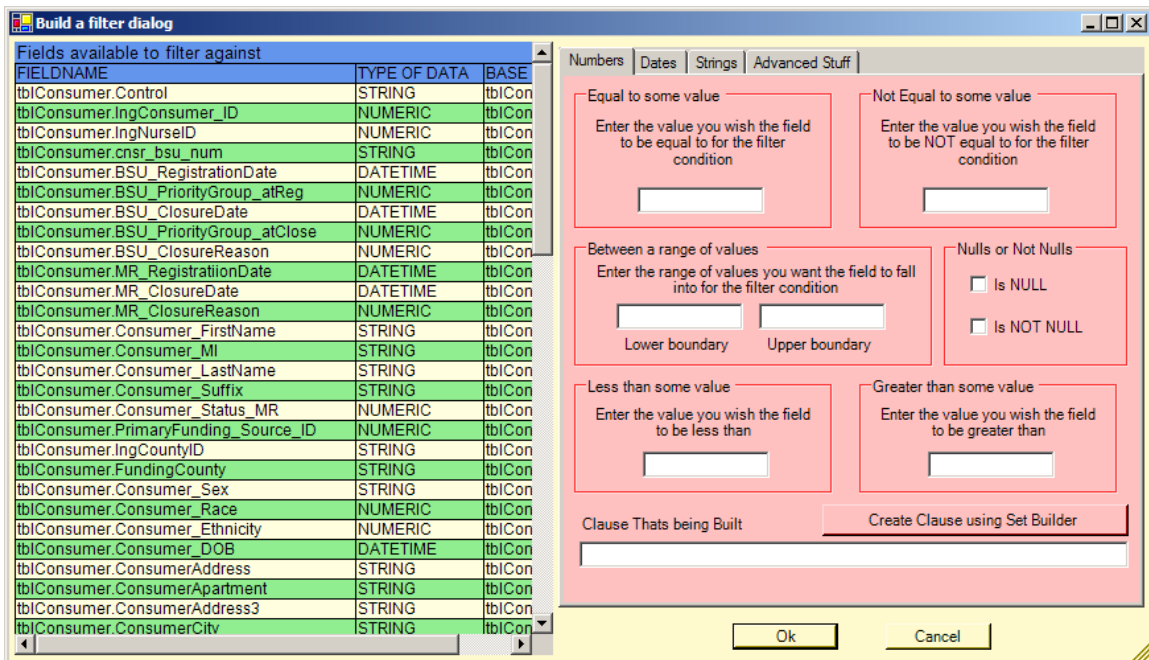
Finally selected the txtCounty field as what I am going to pull back across the join.

Click OK



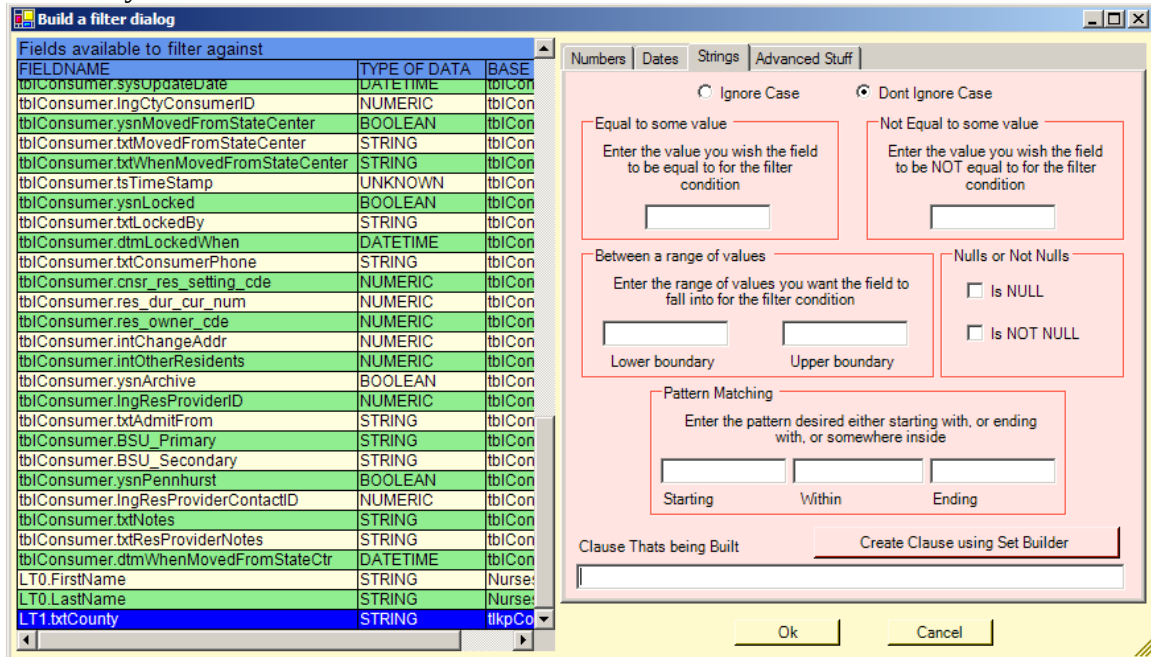
I now have my second lookup added to my query.

Now lets use the Add Filter operation to select only a certain subset of the counties for our query. Click the Add filter button

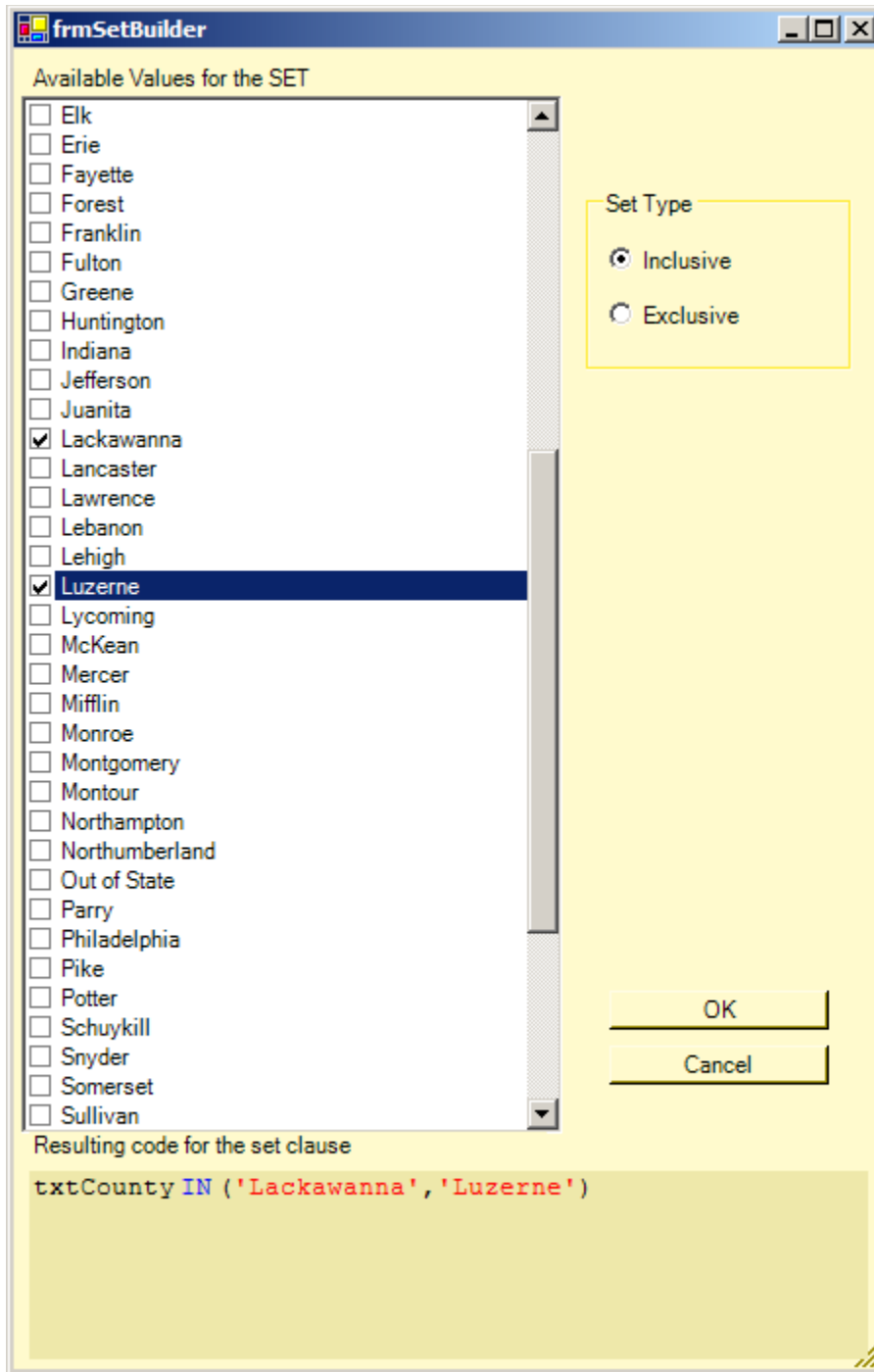


In the resulting dialog you can enter all manner of different kinds of filtering options to reduce the amount of data you get from your queries so you get only the data you want. In our case here we want to only look for counties that are in a small set of the actual counties that are present in the database. The reporter helps you narrow this scope by allowing you to observe that data that is in the complete set and then selectively hone that set down to get only what you want.

For this example we will scroll the Fields Available to Filter Against” grid to the bottom and select the LT1.TXTcounty field that we added to our query with the lookup against the county table

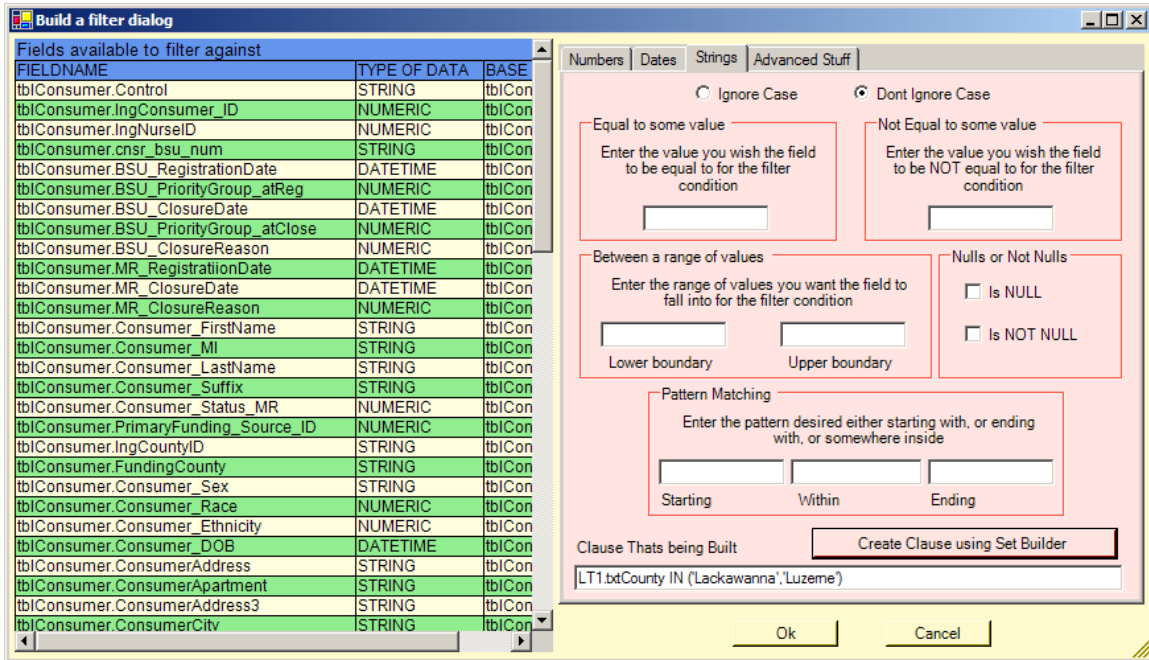


Now we don't know what data is in this field yet but we can examine this data by clicking on the Create Clause using Set Builder button

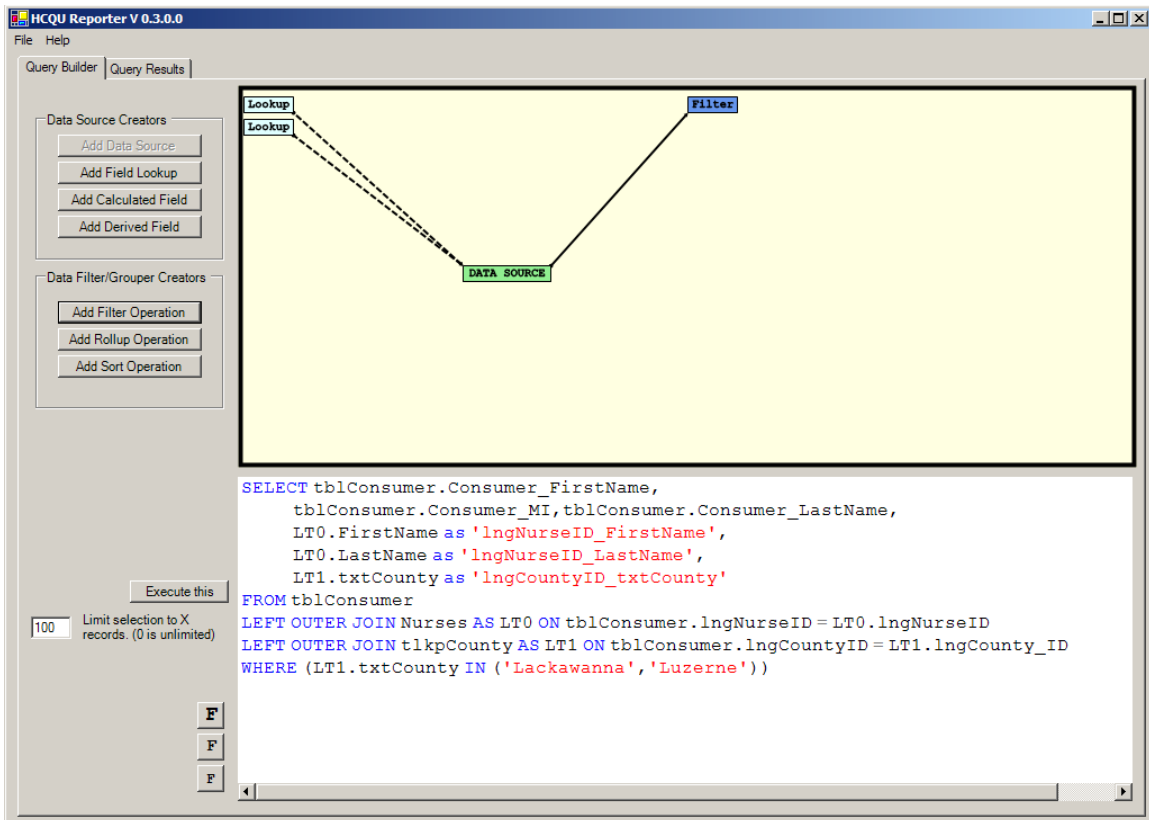


The resulting dialog shows all the distinct values for the field that we selected and now I can select only those values that we want to select on, Lackawanna and Luzerne in our example. Notice the SQL specific IN clause that the tool has built for us in the bottom of this dialog. We also have the option to have it build an Inclusive set ( the default ) meaning I want matches that are IN this set of values, or exclusive sets meaning matches that are NOT IN this set of values.

Click OK to accept this Inclusive set.



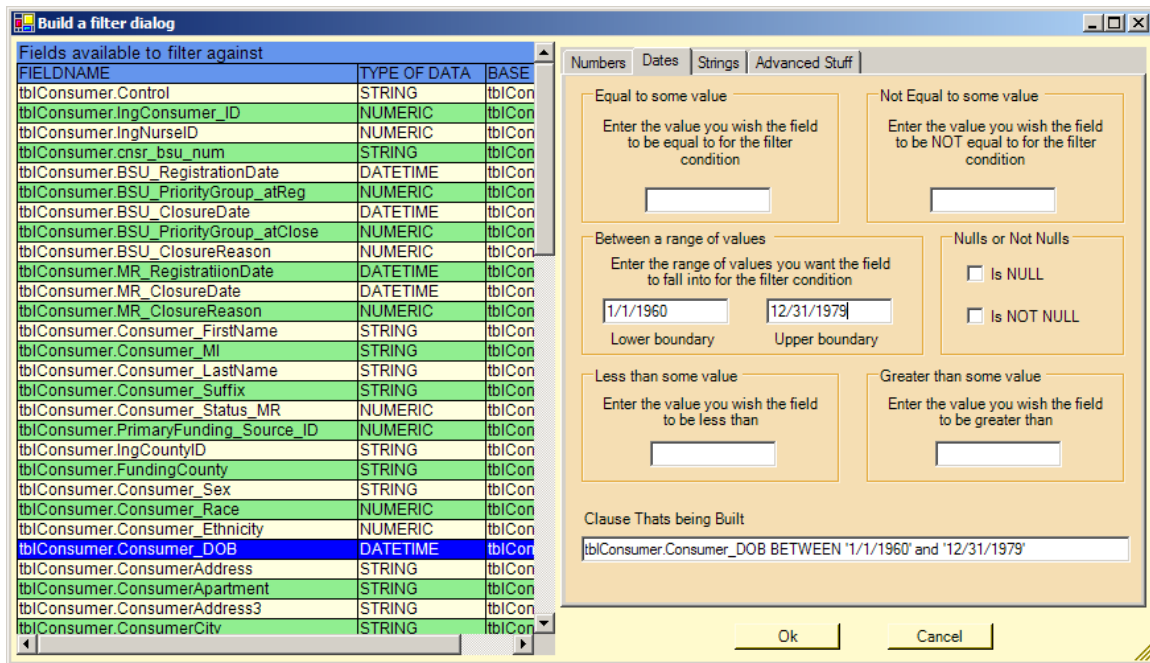
Now click OK again to add this clause to our filters.



Now you can see the added filter to our query.

You are not limited to selecting a filter on a field that you actually have pulled into the query however. Now lets add a filter on the consumers birth date, looking for any consumer that was born in the 60's and 70's.

Click Add filter again

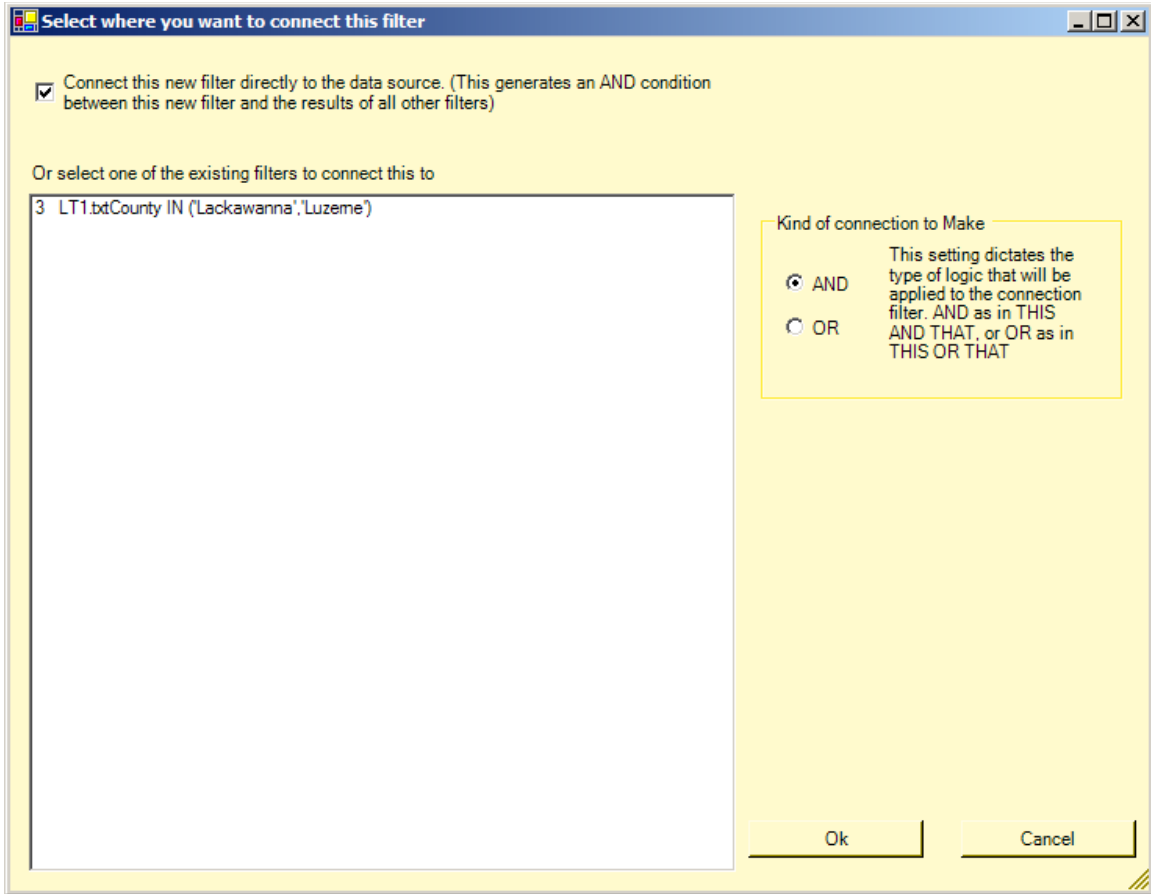


Here I selected the Consumer\_DOB field and then keyed 1/1/1960 and 12/31/1979 into the date range between text boxes.

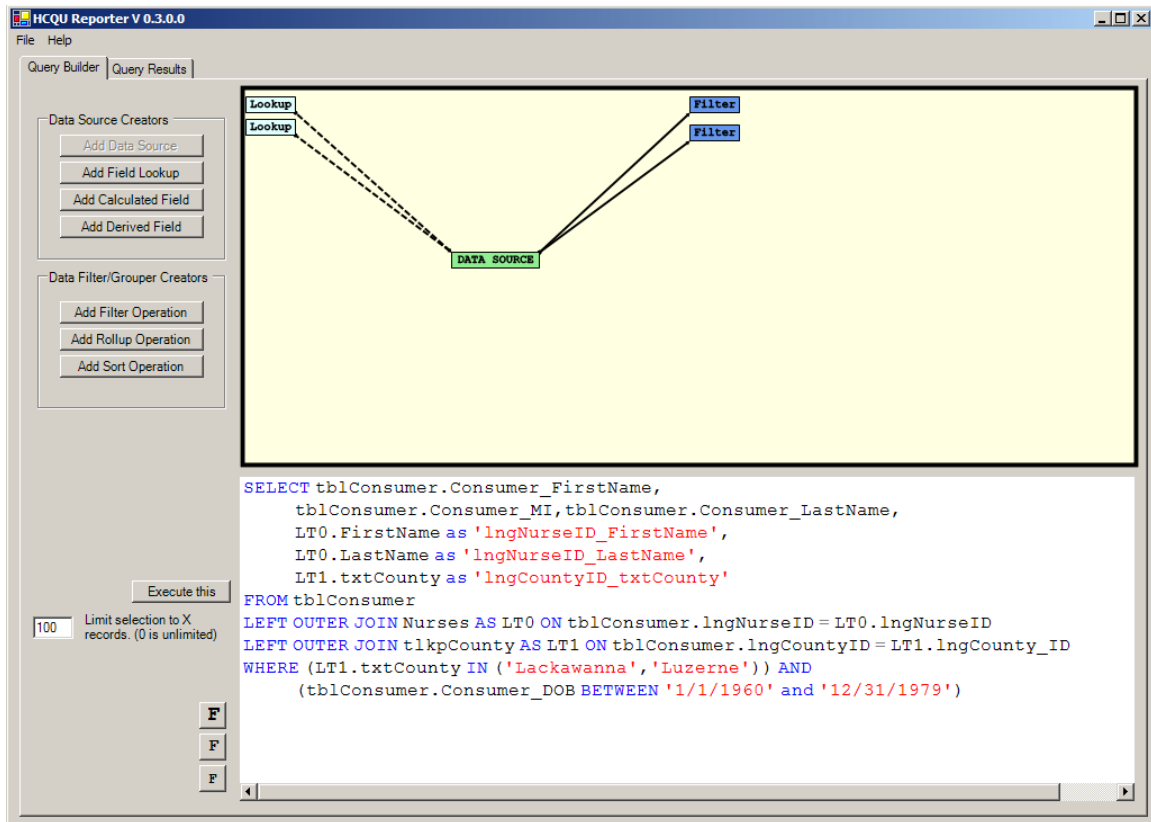
Click OK again

But here we get another dialog because we have already placed a filter into the query the HCQU reporter needs to know what our intention is for this new filter.

We can add it to the existing filter with either an AND or and OR clause. Or we can add this filter to the body of the main query also with an AND or and OR clause. Normally folks thing is terms of ANDS so that is the default. But the software does allow for some very complex Boolean logic trees to be built using careful attachments and the AND/OR clauses.



Just click the OK button to accept the default conditions and connections



We now have our new query with two lookups and two filters.

Now if you want to run the query and examine the results you can select the Execute This button. Then select the Query Results tab at the top of the screen to display the results Panel.



HCQU Reporter V 0.3.0.0

File Help

Query Builder Query Results

To Excel F F F

### Query Results

Consumer_FirstName	Consumer_Mi	Consumer_LastName	IngNurseID_FirstName	IngNurseID_LastName	IngCountyID_txtCounty
NADINE	{NULL}	FLOCK	{NULL}	{NULL}	Lackawanna
TAMI LYNN	{NULL}	PALUCH	Sharon	Fairburn	Luzerne
Michael	{NULL}	Boruta	Lisa	Sinclair	Lackawanna
JENNIFER	{NULL}	KELLMER	Lisa	Sinclair	Luzerne
MARYANN	{NULL}	ABRAMCHECK	Lisa	Timek	Luzerne
JOHN	{NULL}	POPPE	Lisa	Sinclair	Luzerne
John	{NULL}	Sloss	{NULL}	{NULL}	Lackawanna
BARBARA	{NULL}	KNEPP	Sharon	Fairburn	Luzerne
DIANE	{NULL}	SAMUELS	Lisa	Timek	Luzerne
HELEN	{NULL}	PETROSKY	Sharon	Fairburn	Luzerne
KEVIN	{NULL}	DOMIN	Lisa	Timek	Luzerne
MARY	{NULL}	CASO	Lisa	Timek	Luzerne
BRENDA	{NULL}	MCMANUS	Sharon	Fairburn	Luzerne
PHILLIP	{NULL}	GALLAGHER	Lisa	Sinclair	Luzerne
MICHAEL	{NULL}	SABOL	Lisa	Timek	Luzerne
Kelly	{NULL}	Green	Sharon	Fairburn	Lackawanna
JACQUELINE	{NULL}	CIANNELLI	Tara	Morgan	Luzerne
KRISTINA	{NULL}	FETCHEN	Tara	Morgan	Luzerne
Steven	{NULL}	Perko	Tara	Morgan	Lackawanna
RICHARD	{NULL}	MOSHER	Lisa	Sinclair	Luzerne
AMY	{NULL}	REIDLINGER	Lisa	Timek	Luzerne
SEAN	{NULL}	FLYNN	Tara	Morgan	Luzerne
LEANN	{NULL}	HUMKO	Lisa	Timek	Luzerne
PATRICK	{NULL}	DINOFRIO	Sharon	Fairburn	Luzerne
JARRET M.	{NULL}	PEARSON	Lisa	Sinclair	Luzerne
DAVID	{NULL}	SIROTA	Tara	Morgan	Luzerne
DANIEL	{NULL}	EICHHORN	Tara	Morgan	Luzerne
DONNA	{NULL}	ANDERSON	{NULL}	{NULL}	Luzerne
CHARLES	{NULL}	KIRKENDAL	Tara	Morgan	Luzerne
ARLENE	{NULL}	SMITH	Sharon	Fairburn	Luzerne
MARGARET	{NULL}	NAMEY	Tara	Morgan	Luzerne
HENRY	{NULL}	RICCO	Lisa	Sinclair	Luzerne
THOMAS	{NULL}	KRUZEL	Lisa	Sinclair	Luzerne
DAVID	{NULL}	NAULTY	Sharon	Fairburn	Luzerne
JOHN	{NULL}	MCMANUS	Lisa	Timek	Luzerne
RICHARD	{NULL}	MORGAN	Tara	Morgan	Luzerne
RANDY	{NULL}	SEVERCOOL	Jane	Murphy	Luzerne
RICHARD	{NULL}	CAREY	Lisa	Sinclair	Luzerne
EDWARD	{NULL}	ROMAN	Sharon	Fairburn	Luzerne
JOLENE	{NULL}	CHIUMENTO	Lisa	Timek	Luzerne

This panel shows the results of the query that was built using the reporter and the resulting data as it was brought in from the database.

Clicking the To Excel button will send the grid to Excel ( If excel is installed ) to allow further manipulation.

	A	B	C	D	E	F	G	H
1	Consumer_FirstName	Consumer_MI	Consumer_LastName	IngNurseID_FirstName	IngNurseID_LastName	IngCountyID	txtCounty	
2	NADINE	{NULL}	FLOCK	{NULL}	{NULL}	Lackawanna		
3	TAMI LYNN	{NULL}	PALUCH	Sharon	Fairburn	Luzerne		
4	Michael	{NULL}	Boruta	Lisa	Sinclair	Lackawanna		
5	JENNIFER	{NULL}	KELLMER	Lisa	Sinclair	Luzerne		
6	MARYANN	{NULL}	ABRAMCHECK	Lisa	Timek	Luzerne		
7	JOHN	{NULL}	POPPLE	Lisa	Sinclair	Luzerne		
8	John	{NULL}	Sloss	{NULL}	{NULL}	Lackawanna		
9	BARBARA	{NULL}	KNEPP	Sharon	Fairburn	Luzerne		
10	DIANE	{NULL}	SAMUELS	Lisa	Timek	Luzerne		
11	HELEN	{NULL}	PETROSKY	Sharon	Fairburn	Luzerne		
12	KEVIN	{NULL}	DOMIN	Lisa	Timek	Luzerne		
13	MARY	{NULL}	CASO	Lisa	Timek	Luzerne		
14	BRENDA	{NULL}	MCMANUS	Sharon	Fairburn	Luzerne		
15	PHILLIP	{NULL}	GALLAGHER	Lisa	Sinclair	Luzerne		
16	MICHAEL	{NULL}	SABOL	Lisa	Timek	Luzerne		
17	Kelly	{NULL}	Green	Sharon	Fairburn	Lackawanna		
18	JACQUELINE	{NULL}	CIANNELLI	Tara	Morgan	Luzerne		
19	KRISTINA	{NULL}	FETCHEN	Tara	Morgan	Luzerne		
20	Steven	{NULL}	Perko	Tara	Morgan	Lackawanna		
21	RICHARD	{NULL}	MOSHER	Lisa	Sinclair	Luzerne		
22	AMY	{NULL}	REIDLINGER	Lisa	Timek	Luzerne		
23	SEAN	{NULL}	FLYNN	Tara	Morgan	Luzerne		
24	LEANN	{NULL}	HUMKO	Lisa	Timek	Luzerne		
25	PATRICK	{NULL}	DINOFRIO	Sharon	Fairburn	Luzerne		
26	JARRET M.	{NULL}	PEARSON	Lisa	Sinclair	Luzerne		
27	DAVID	{NULL}	SIROTA	Tara	Morgan	Luzerne		
28	DANIEL	{NULL}	EICHORN	Tara	Morgan	Luzerne		
29	DONNA	{NULL}	ANDERSON	{NULL}	{NULL}	Luzerne		
30	CHARLES	{NULL}	KIRKENDAL	Tara	Morgan	Luzerne		
31	ARLENE	{NULL}	SMITH	Sharon	Fairburn	Luzerne		
32	MARGARET	{NULL}	NAMEY	Tara	Morgan	Luzerne		

You can also save built queries and open saved queries by using the File pull down menu at the top of the main window.

TODO

This is ALPHA release software and as such there are a number of areas that need work.

The software is way to sensitive still and is prone for crashes so these areas need work.

Its needs more Joins to be coded,

It needs to be able to edit clauses after they are built, now to change a filter you have to remove it on its context menu ( Right mouse button over it in the canvas) and select the Remove Object item.

It needs just about everything..

But it is approaching usability.

We are releasing this as a preview of things to come from

