IPI International

Working with PowerShell

Name : IPI Excel Connection
Type : PowerShell Script
Designer : Paul I Ighofose
Date Created : 28/12/2019
Date Updated : 30/12/2019
Requirements : Microsoft Windows 7 upwards
Function : PowerShell Query Tool for Excel using either JET or ACE OLEDB and –ComObject connections.

Introduction

When the IT department kept updating our CEO’s Office package and losing the VBA User forms I had developed for him and his PA, I thought it time to have another bash at PowerShell Win-forms. Having achieved more experience from the HTA Application **IPI Database Copy and Search.hta**, I applied similar approaches to dealing with filtering and Query Syntax building.

This package comes close to the Query Environment offered by Ms Access and a lot had to be done to manage the Excel Application when forced to use the JET driver on xlsx and xlsm files.

Package Structure

A single script file enables current function.

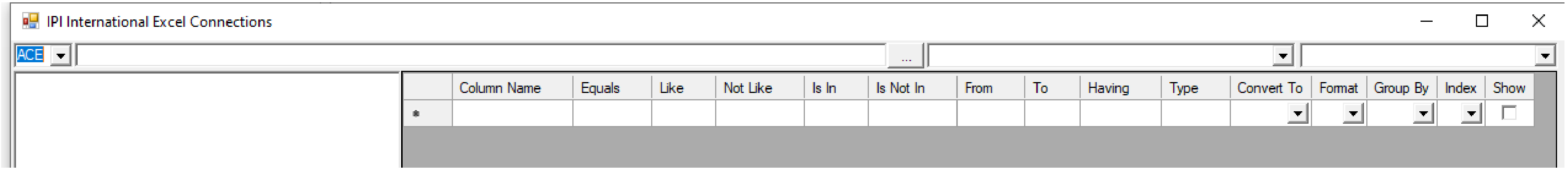
Operational Guide

Create a Shortcut Link File

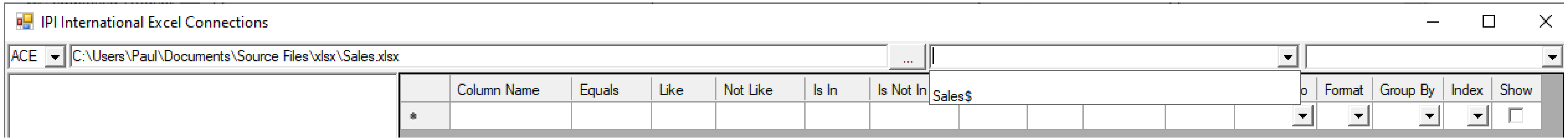
Create Shortcut

- Paste the following text in to the Shortcut Properties, changing the file path as needed
  powershell.exe -WindowStyle Hidden -ExecutionPolicy Unrestricted -File "%userprofile%\Documents\IPI International\IPI Powershell\IPI Excel Connection.ps1"

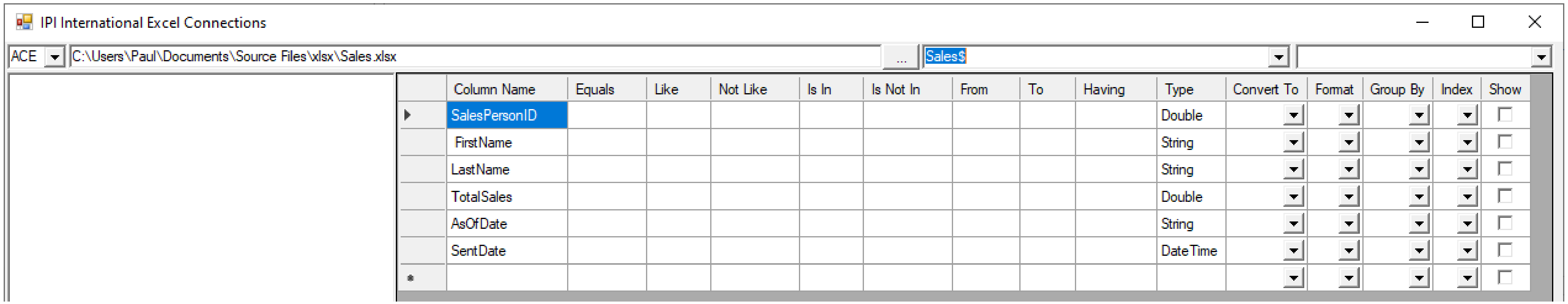
Initial Display on Load



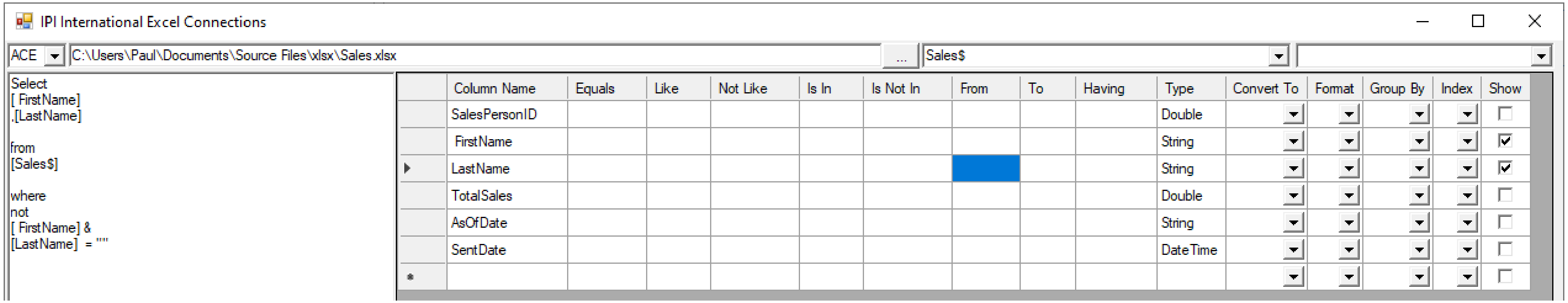
- A driver selection Combo box is located at the top-left and currently offers either ACE or JET OLEDB connection to Excel for running SQL Queries.
- The … command button will display a File Open Dialog and the result will be entered in the textbox to the right of it.
- This function also runs a –ComObject Excel.Application connection to retrieve all Named Ranges and Worksheet names used in populating the combo box to the right of it.



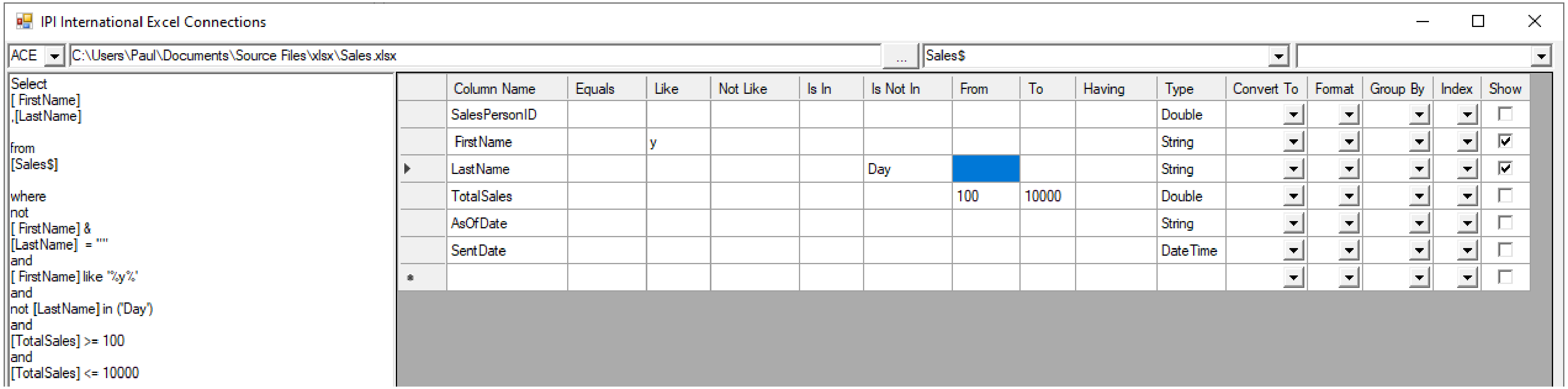
- When using JET and selecting a Named ranged, there is a rather tedious delay, not so when using ACE.
- Selecting a name from the combo box will initiate an OLEDB connection to the file path displayed and populate the DataGridView below with the Column Names and the Data Types (Type column) of each column in the worksheet.



- Checking the **Show** column check boxes will populate the **Select** portion of the textbox to the left of the DataGridView with the SQL syntax and overwrites entries every time a change is made in the DataGridView.

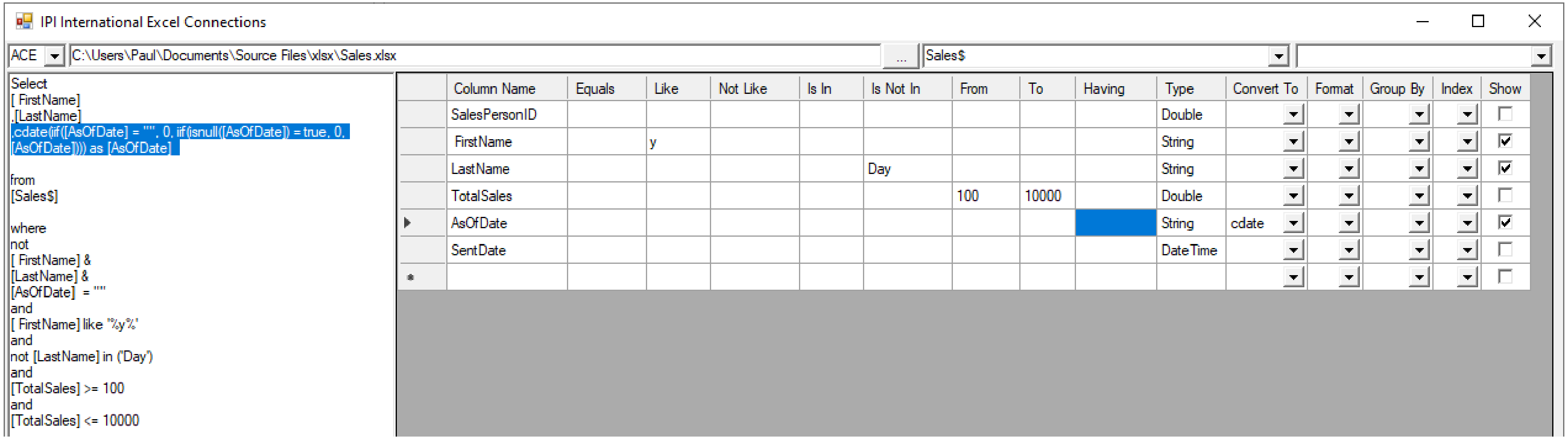


- Making entries in the filter columns will populate the **Where** portion of the textbox.

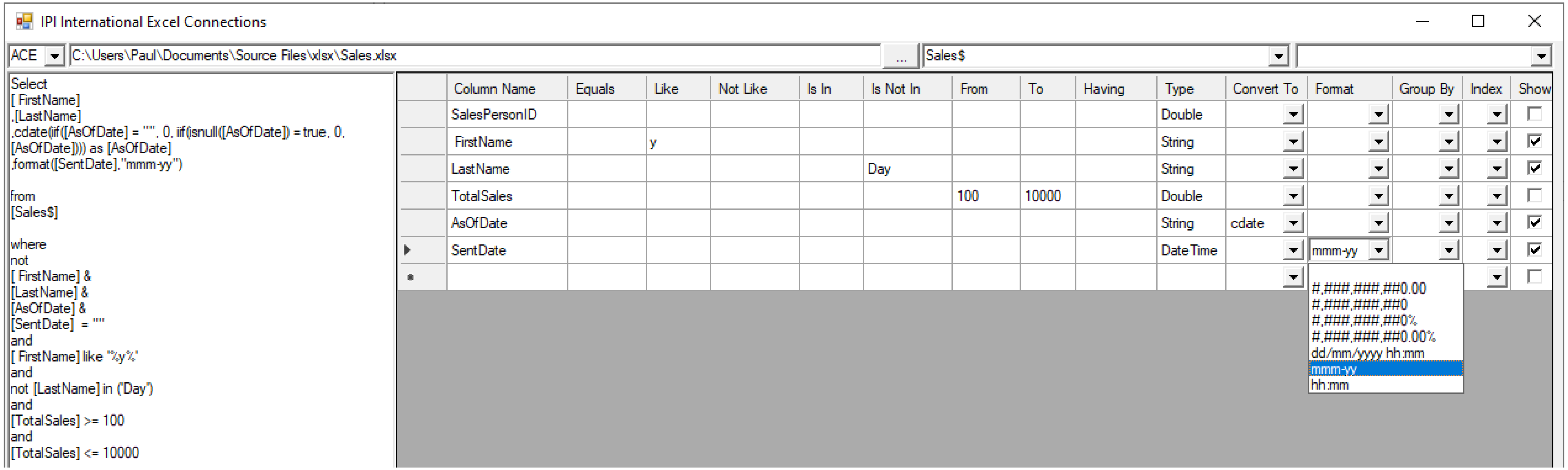


- Sizing of the form will adjust the sizes of the Editor Textbox and Filter Grid View if they are currently displayed, or, will adjust the sizing of the Results Grid View
- A lot has been done to stop Excel from opening multiple instances in the background when using the JET OLEDB driver and the –ComObject without affecting any other Excel Application currently running open workbooks. Occasionally there may still be an open excel document when the call to close it has not succeeded, but is a rare occurrence and more prone to be a result of testing.

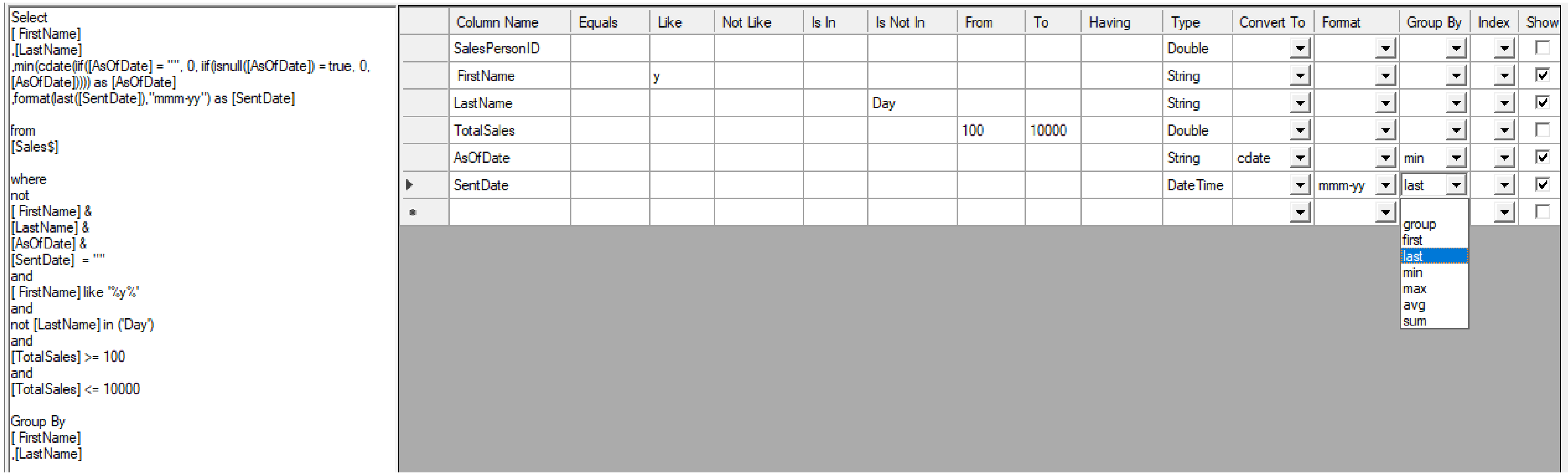
- Convert To currently only offers CDate and CDbl conversion and is applied to all portions save **Group By**.



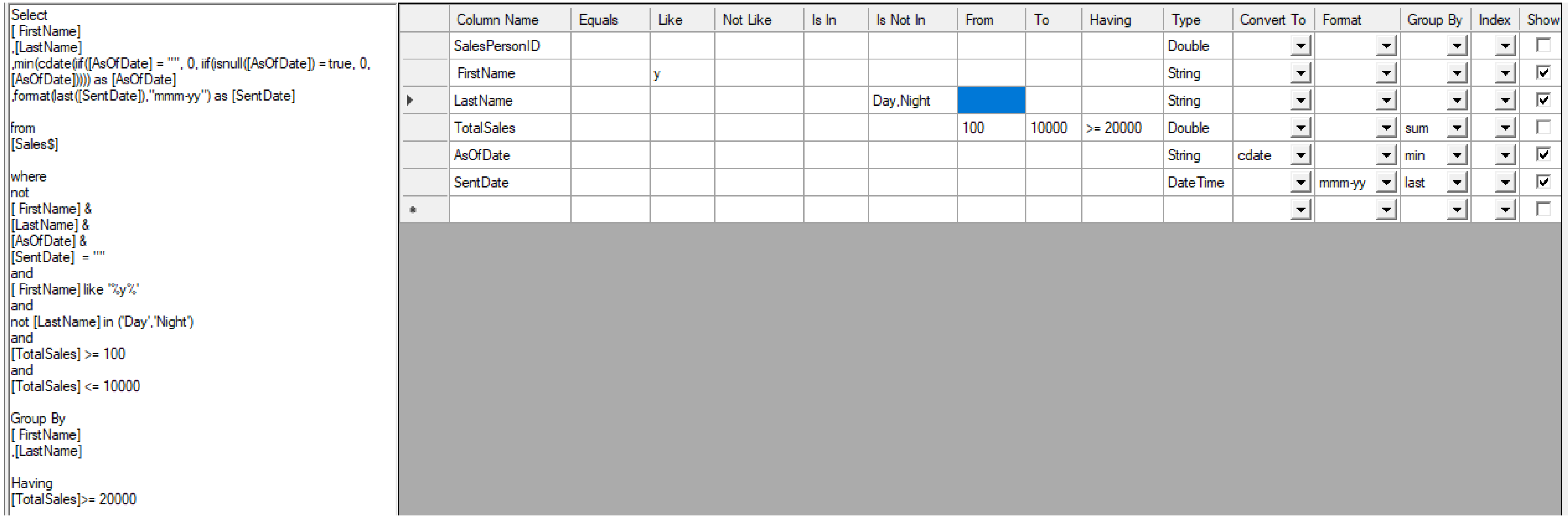
- Format offers decimal, percent, date-time, month-year and time formatting for the Select portion.



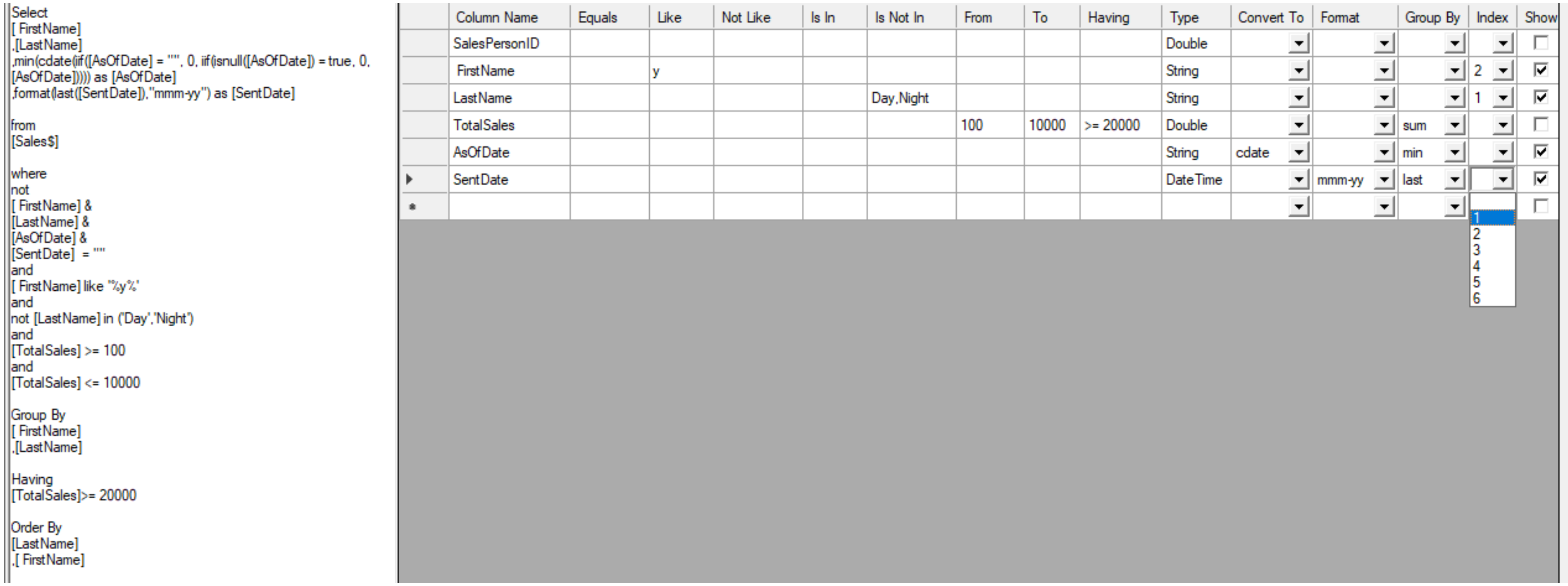
- If an Item has Show ticked but no Group By option selected and any other Column has a Group By selection, then that item will also appear in the Group By portion.

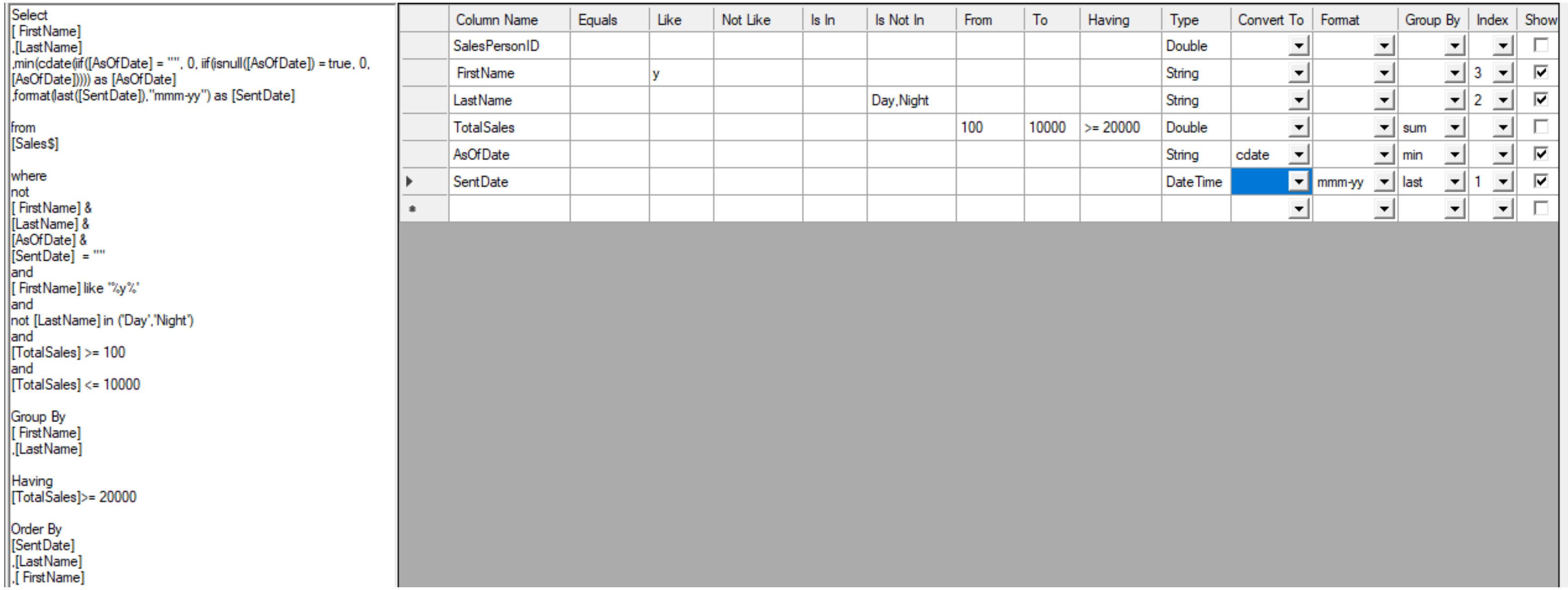


- Making an entry without using operator symbols in the Having column will populate the **Having** portion of the textbox with a filter that looks for equality.

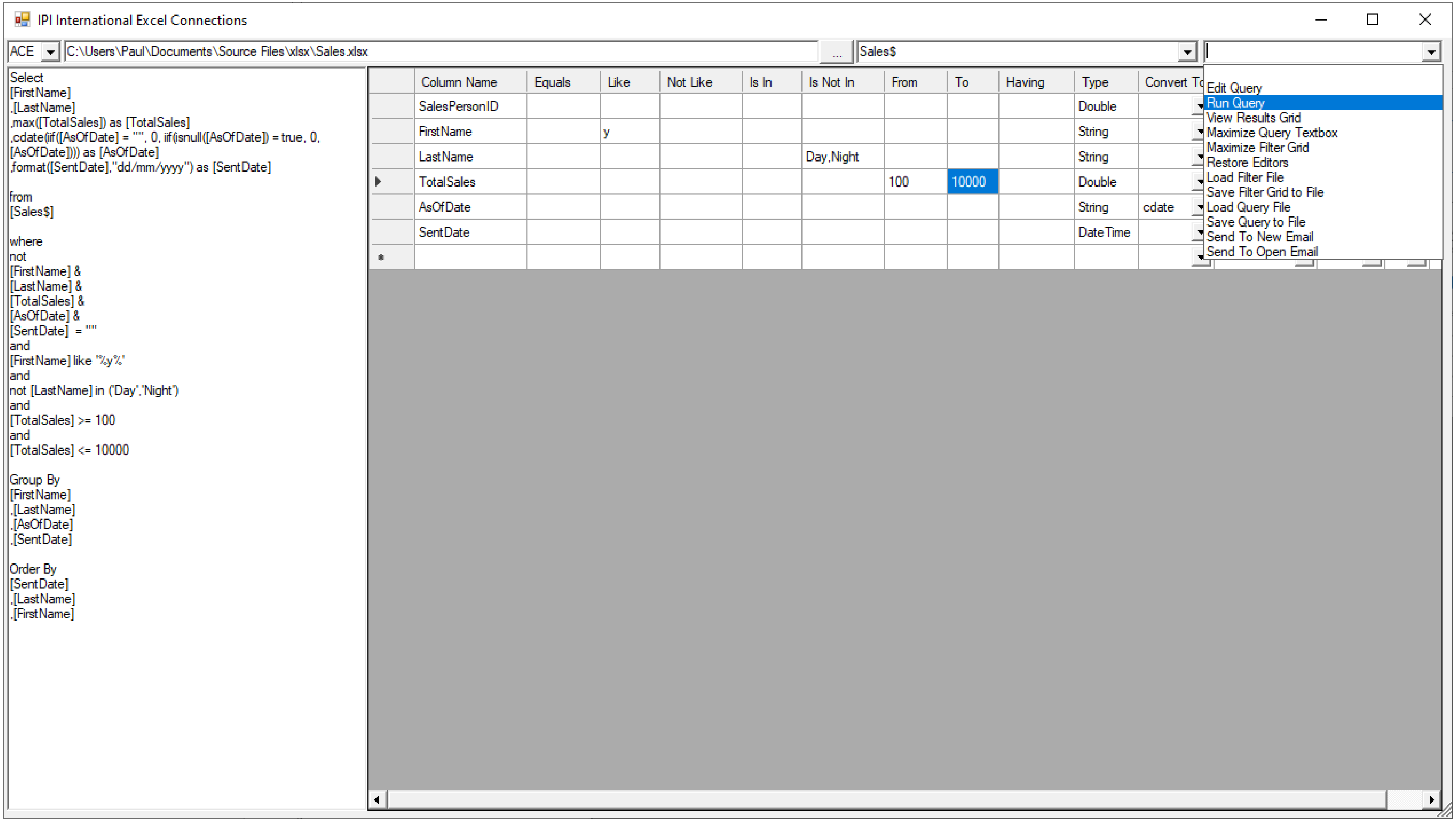


- Index will change automatically if the same index number is selected later for another column, iterating through all and adjusting to suit. This dictates the order Column Names will appear in the **Order By** portion.



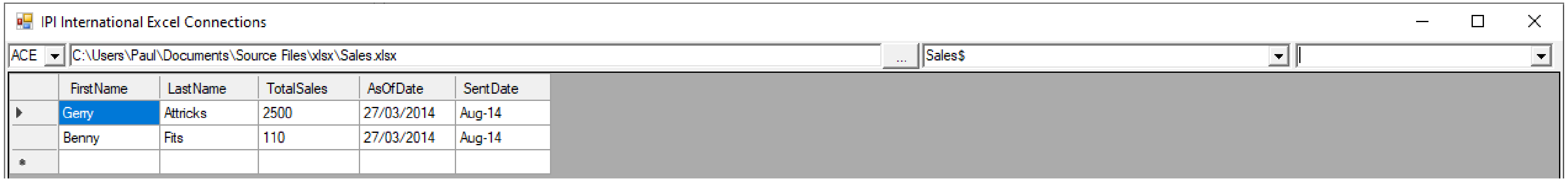


SQL Results

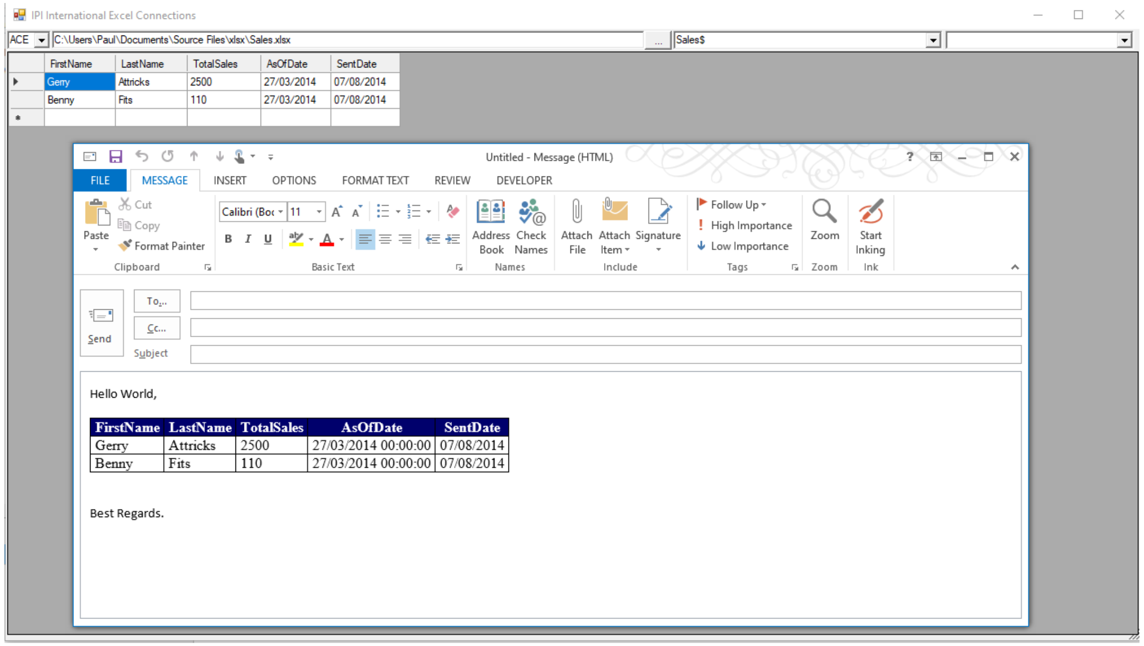


- After selecting the Run Query from the Functions combo box all results are displayed in a Grid View on the Filter Grid View and Edit Textbox are hidden

- Clicking the headings will sort either Ascending or Descending by that heading (although you may have already specified sort orders in you statement)



- Selecting the **Send To Open Email** in the functions combo box will place the results at the current cursor position of an open email, or two lines down using the **Send To New Emai**l function.



- Oddly enough, when I updated the PowerShell Script file I developed for our CEO, and tried running it through the PowerShell IDE, I kept getting the all frustrating Exception ACE OLEDB Microsoft Driver not properly installed. But, when I ran the same script using the shortcut link, the ACE driver selection worked without any Exceptions
- Also note that if you set the Shortcut to start this script with Administrator Access, the Send to Outlook functions will not work unless Outlook is also opened using Elevated Privileges
- The **Maximize Query Textbox** function will hide the Filter Grid View and expand the Editor Textbox, also setting it to automatically resize with Form resizing. And vice versa for the **Maximize Filter Grid** function. **Restore Editors**, restores sizing to initial sizing

ACE | C:\Users\Paul\Documents\Source Files\xlsx\Sales.xlsx | ... | Sales$

```
Select
[FirstName]
,[LastName]
,max([TotalSales]) as [TotalSales]
,cdate(iif([AsOfDate] = "", 0, iif(isnull([AsOfDate]) = true, 0, [AsOfDate]))) as [AsOfDate]
,format([SentDate],"dd/mm/yyyy") as [SentDate]

from
[Sales$]

where
not
[FirstName] &
[LastName] &
[TotalSales] &
[AsOfDate] &
[SentDate]  = ""
and
[FirstName] like '%y%'
and
not [LastName] in ('Day','Night')
and
[TotalSales] >= 100
and
[TotalSales] <= 10000

Group By
[FirstName]
,[LastName]
,[AsOfDate]
,[SentDate]

Order By
[SentDate]
,[LastName]
,[FirstName]
```

ACE | C:\Users\Paul\Documents\Source Files\xlsx\Sales.xlsx | ... | Sales$

Column Name	Equals	Like	Not Like	Is In	Is Not In	From	To	Having	Type	Convert To	Format	Group By	Index	Show
SalesPersonID									Double					☐
FirstName		y							String				3	☑
LastName					Day,Night				String				2	☑
TotalSales						100	10000		Double			max		☑
AsOfDate									String	cdate				☑
SentDate									DateTime		dd/mm/yyyy	group	1	☑
														☐

- **Save Filter Grid to File** will create a tab-delimited file (default extension on offer is .grid), and the **Load Filter File** will reload the settings from it. The Save function inserts the File Path displayed in the File Path textbox on the first line, then the Named Range/Worksheet name on the second line followed by the Filter Grid values. It loads the file; gets the Named Ranges and Worksheet names; populates the File Path and Names combo box; populates the grid with the current column names of the given file location and Named Range/Worksheet Name; then it will only update the Filter Grid Columns where the Column Name matches those contained in the grid file

Sales on Github.grid - Notepad

File Edit Format View Help

```
C:\Users\Paul\Documents\Source Files\xlsx\Sales.xlsx
Sales$
SalesPersonID                                                               Double
FirstName               y                                                   String                          3       True
LastName                                        Day,Night                           String                  2       True
TotalSales                                              100     10000       Double                  max             True
AsOfDate                                                                    String  cdate                   True
SentDate                                                                    DateTime                dd/mm/yyyy      group   1       True
```

- **Save Query To File** will create a text file (default extension on offer is .sql), and the **Load Query File** will reload the SQL statement from it. The Save function inserts a top line as # File Path = [The File Path displayed in the File Path textbox]

Sales on GitHub.sql - Notepad

File Edit Format View Help

```
# File Path = C:\Users\Paul\Documents\Source Files\xlsx\Sales.xlsx
Select
[FirstName]
,[LastName]
,max([TotalSales]) as [TotalSales]
,cdate(iif([AsOfDate] = "", 0, iif(isnull([AsOfDate]) = true, 0, [AsOfDate]))) as [AsOfDate]
,format([SentDate],"dd/mm/yyyy") as [SentDate]

from
[Sales$]

where
not
[FirstName] &
[LastName] &
[TotalSales] &
[AsOfDate] &
[SentDate]  = ""
and
[FirstName] like '%y%'
and
not [LastName] in ('Day','Night')
and
[TotalSales] >= 100
and
[TotalSales] <= 10000

Group By
[FirstName]
,[LastName]
,[AsOfDate]
,[SentDate]

Order By
[SentDate]
,[LastName]
,[FirstName]
```