Google Sheets Query Function

About this document

This is a brief introduction to using the Google Query function in Google Sheets. It's part of a group of resources for learning about spreadsheets. You'll find some more about this, with some links to examples, at the end of this document.

Once you've got the hang of the basics, you should be able to figure out the rest from the Google Support material on the Query language.

Please feel free to make copies of this document and the example files for your own personal use.

TIP: Incidentally, I've not been too careful with page breaks, so it's best to view this with *Print Layout* turned off (*View* menu). Enable the **Outline View** for navigation (Tools > Document outline).

Why the Query function?

The query function enables you to retrieve rows from tabulated data using a query expression that is very similar to that used in Structured Query Language (SQL). Some advantages over using simple filtering or filter views are:

- The view of the source data does not change for other collaborators
- You can choose to view only data from selected columns, without resorting to 'hide'
- More complex criteria can be used, including aggregation (sum, average etc)

The basic function syntax is:

=query(*source_data,"query expression"*)

Although you can use it on the sheet that contains the data, you are more likely to use this function on another worksheet in the file. The function is entered in just one cell, which becomes the top left cell of the retrieved data.

source_data	This is the range of cells containing the data. Since you will most likely be referring to another worksheet, it needs to include the sheet name if you use row-column notation: **data_sheet!A1:G50** You can also refer to entire columns if you don't know how many rows there will be: **data_sheet!A:G** But defining this as a **named range** makes the syntax easier, and allows easy extension of the number of columns used. **Note:** If you define whole columns in a named range, the query may include blank rows - you can stop them appearing by using an appropriate 'where' clause.
"query criteria"	Enclosed in quotes, this is the query expression which defines the columns to be used, criteria to be applied, sort orders and grouping. The expression can refer to values in other cells in the spreadsheet file.

Query expression syntax

The expression ***must always be enclosed in quotes***, though when referencing cells for criteria you may need to break these into two or more sections, joined using ampersand (&), as cell references will not be recognised within the quotes.

In these examples the data is in a named range called **dataList**.

Select and sort

=query(dataList, "select A,B,F")	simple select - column order can be changed
=query(dataList, "select A,F,B order by B desc")	select and sort by one column
=query(dataList, "select A,F,B order by B,C asc")	select and sort by 2 columns (note sort columns do not need to be selected)

Adding a 'where' clause for criteria

Numbers and Text require slightly different syntax. Text values are enclosed in single quotes, whereas numbers are not. Dates require a special syntax that needs a bit more work.

=query(dataList,"select A,B,F where F=3")	select from these columns rows where the data in F is the numerical value 3

=query(dataList,"select A,B,F where D="&F2)	select from these columns rows where the data in D is the numerical value found in cell F2
=query(dataList,"select A,B,F where D>="&F2)	select from these columns rows where the data in D is greater than or equal to the numerical value found in cell F2
=query(dataList,"select A,B where F='Sold'")	select from these columns rows where the data in F is the text 'Sold'
=query(dataList,"select A,B where F=' "&E2&" ' ")	select from these columns rows where the data in F is the text found in cell E2

Making sense of the syntax

For numerical values, the number can be used directly in the expression. When a cell reference is used it must be outside the double quotes or it will not be regarded as a cell reference. The concatenation operator (ampersand &) is used to join it to the rest of the expression:

"select A,B,F where D = " & F2

When using text values, you must enclose text in single quotes:

"select A,B where F = 'Sold' "

These single quotes are a part of the expression, so must remain within the double quotes. However, the cell reference cannot simply be enclosed in single quotes as it would be treated as text, just like the word Sold.

So we break the expression apart, closing each portion with double quotes, and replacing the text with the cell reference. This requires the expression to be joined using two concatenation operators (ampersand &):

"select A,B where F = ' " & E2 & " ' "

Partial text: If you want to use only part of a text value in the *where* clause, use contains instead of equals:

"select A,B where F contains 'br' "

This would filter for all instances of text containing **br**, eg broken, library, cobra.

Using dates in the 'where' clause is a little more complicated, as the date value needs a precise syntax that doesn't match the date format used in date/time columns:

"select C,B,E,H where H < date '1995-12-31' "

Notice that:

- The word **date** must be used as an identifier in the expression
- The date is a text string (enclosed in single quotes)

- The date ***must*** use the format '**yyyy-mm-dd**' with a hyphen separator

This creates an interesting challenge if you want to reference a cell value in the expression: if you enter the desired date in the 'yyyy-mm-dd' format, Google sheets is likely to recognise it as a date and convert it to a 'proper' date/time value - which won't work in the query expression.

To be sure of getting a correct 'string' value for the date, enter the date in your usual format and then use functions to generate the string. Suppose my date is in cell A1:

Option 1 - the long-winded approach

=year(A1) gets just the year from a date

=text(year(A1), "0000") gets the 4-digit year and converts it into text

Similarly, you can obtain a text value for month and day:

=text(month(A1), "00")

=text(day(A1), "00")

These must be joined (concatenated) with a hyphen "-" between them. As a single expression this then becomes:

=text(year(A1),"0000") & "-" & text(month(A1),"00") & "-" & text(day(A1),"00")

If you do this conversion in a nearby cell (eg cell B1), then you can reference this in the 'where' clause of the query function:

"select C,B,E,H where H < date ' " & B1 & " ' "

Option 2 - the more compact form

The text function lets you define a range of formats, so you can get the date string from:

=text(A1,"yyyy-MM-dd")

As before, do this conversion in a nearby cell (eg cell B1) and reference it in the where clause.

The file, More Query function examples, includes an example of this on the where (dates) tab.

Using a 'where' clause to eliminate blank rows

If a named range is defined using entire column (ie including blank rows) you may find these blanks appear in the query result (which, depending on the sort order, could be at the top!). To stop these appearing include a **where** clause using this syntax (assuming column A):

"...where A <> ' ' " (for text fields)
"...where A <>0" (for numeric fields)

This means 'where values in column a are not zero-length text.

Combining where clauses

Multiple criteria can be included, but these must be joined logically by **And** or **Or**. When **And** is used, only rows in which both conditions are true are included, whereas **Or** results in all rows when either (or both) conditions are met.

Without cell references:

"select C,B,E,D where E = 'Derwith' And D=2"

With cell references:

"select C,B,E,D where E = ' " & G2 & " ' And D=" & H2

Grouping

When grouping of records is appropriate, aggregate functions can be included in order to calculate averages, totals etc. In this case you must define the columns to be presented, the function to be used and on which column to group records:

"select E, avg(F) group by E"

This means, "Display values from column E and the average of grouped values in column F, grouping records together that contain the same value in column E."

Examples of Query Function

These two Google Sheets files include some examples of using the query function:

Query function examples (opens Google Sheets document in new tab/window)

More Query function examples (opens Google Sheets document in new tab/window)

In both these examples the **dataList** worksheet includes module results for a number of (fictitious) students. As most students have taken more than one module, they appear several times. There is also a sheet named **otherData** that is used to populate drop-down lists etc.

In each of the example sheets, cells shaded in green (usually on the third row) contain the query function. Some comments are included to help you understand the way the functions are used.

Feel free to make copies of these to experiment with.

These example files are also included in our Essential Spreadsheets practical guide.

Last update: June 2017