Queryable Google Form Data

INTRODUCTION

When it comes to Google forms and the typical survey data that is collected via the platform. The analysis is hard to review with sporadic information received based on a simple queue. The main issue is dealing with a massive number of records that needs to be filtered. In order to find a solution to easily navigate through the database, we can incorporate formulas and tools to reduce the navigational strain. The aim of this post is to introduce varying types of automated assets and simplistic scripts.

A preliminary insight to what we're dealing with, the "Form Response" tab is the typical default name when Google form automatically takes it into Google sheets. All the responses from the Google form will be on this page.
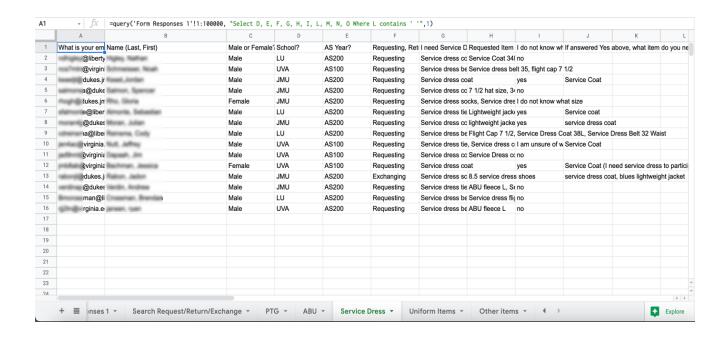
STATIC QUERY

In a new tab, you can filter the responses of a selected record with a distinguishing key. A typical formula may be:
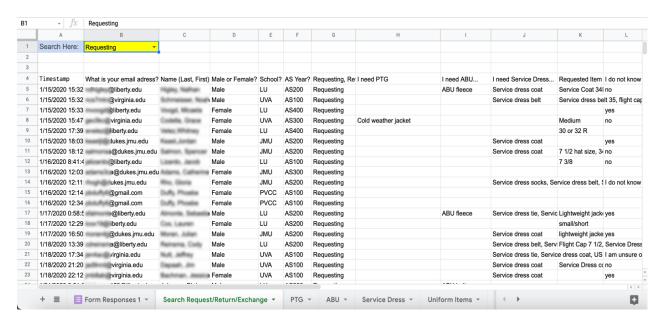
=query('Form Responses'!1:10000, "Select *columnOne*, **_columnTwo_**, Where *column* contains ' '", 1)

Selecting which columns are necessary and for additional removals you can filter out cells with formatting rules. If the Google form has expected responses, it will make it easier to create the formula.
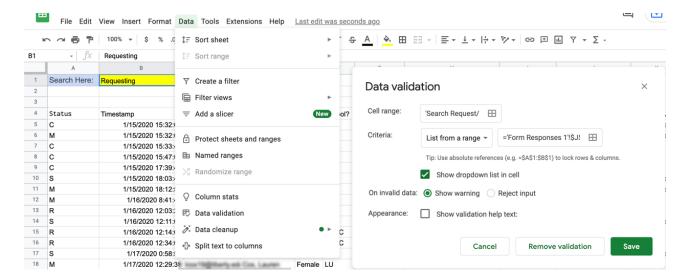
DYANAMIC QUERY

In exchange of using multiple sheets, a singular tab can be used to easily query the specific type of data you are searching for. With a drop-down selection, you can return the row with the following keyword in a specific column.

In order to create a dynamic drop-down selection to narrow the search, the first step is creating a cell that allows you to select from a range. The process is done through Google sheets, built in **Data Validation**.

Select the cell that you would like the rest of the data to follow and input a similar formula of

=QUERY('Form Responses'!C:AA, "SELECT * WHERE _COLUMN_ = """&_THE CELL_ _YOU CREATED THE DROP DOWN_&""", 1)

A4		fx	=QUERY('Form Responses 1'!C:AA, "SELECT * WHERE J = """&B1&"""", 1)			
	A	B	C	D	E	F
1	Search Here:	Requesting ▾				
2						
3						
4	Status	Timestamp	What is your email	Name (Last, First	Male or	School?

GOOGLE SCRIPTS

      We can clear redundant records by using the Google apps scripts that will allow for a simple integration into the Google sheet. The ability to customize the code to your needs. To start you can navigate to toolbar via "**Extensions**" and select "**Apps Script**". You should be brought into the editor. The script in this demonstration removes completed records based on the columns highlight color.
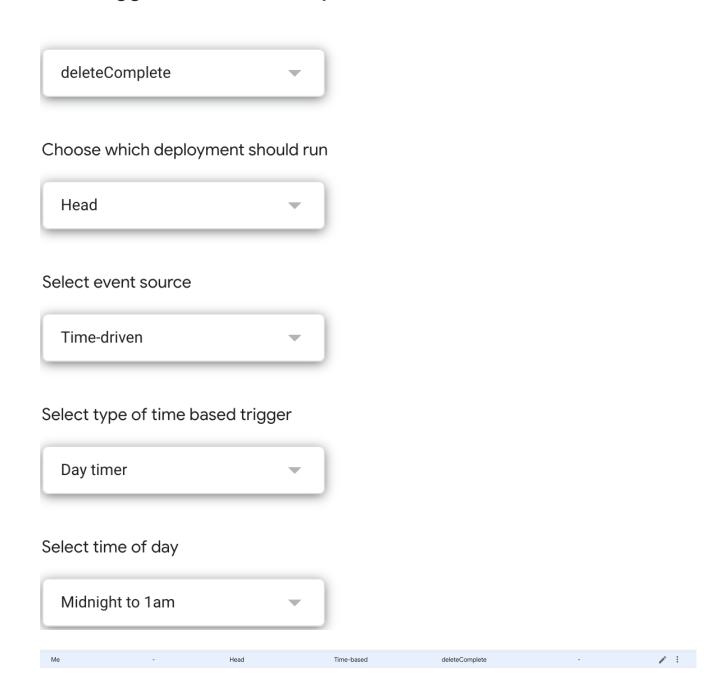
```
 1   function deleteComplete(column,color) {
 2
 3     var column = (typeof(column)!='undefined')?column:5;
 4     var color = (typeof(color)!='undefined')?color:'#00ff00';
 5     var ss = SpreadsheetApp.getActiveSpreadsheet();
 6     var sheet = ss.getActiveSheet();
 7     var range = ss.getDataRange();
 8     var rangeA = range.getBackgrounds()
 9
10     for(var i=rangeA.length-1; i>-1 ; i--) {
11       if(rangeA[i][column-1]==color) {
12         sheet.deleteRow(i+1);
13       }
14     }
15   }
```

Execution log

```
9:19:59 PM   Notice    Execution started
9:20:04 PM   Notice    Execution completed
```

The script to remove the completed row is set to run every time at midnight. To configure when the deployment of this automated task would be located under the same page of the script configuration trigger tab.

Add Trigger for DeleteComplete

deleteComplete ▼

Choose which deployment should run

Head ▼

Select event source

Time-driven ▼

Select type of time based trigger

Day timer ▼

Select time of day

Midnight to 1am ▼

Me	-	Head	Time-based	deleteComplete	-	✏ ⋮

A basic sheet principle is conditional formatting. The script that was introduce earlier is built off the premise that a status of a cell would be automatically highlighted to a certain color if the end user inputs a status acronym. The conditional formatting tool is under the highlight cell tool.