Pastoral Dashboards

Jake Wills - jake.wills@kc.school.nz

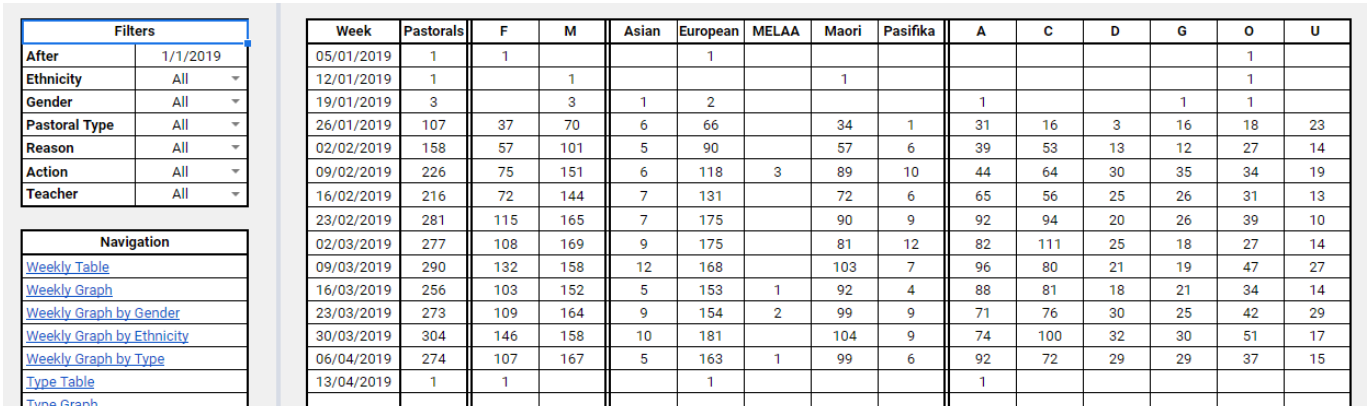
This guide steps through how to create pastoral dashboards using on Google Sheets using data from KAMAR that is automatically updated on a daily basis. The principles in this can be applied to many other types of data that is stored inside KAMAR.

KAMAR Makes the following disclaimers about running scripts:

- KAMAR Limited cannot provide support for this feature. The Helpdesk can assist you to follow the steps in this guide, but what you put in your script, and any troubleshooting thereof, is solely your responsibility.
- KAMAR Limited is not liable for any consequences as a result of calling the Action_ServerScript - this functionality is available as a 'use at your own risk' feature.

This is what we are using as a school… it's provided in an as-is where-is basis for anyone that wants to use it. If you have questions about this please email me.

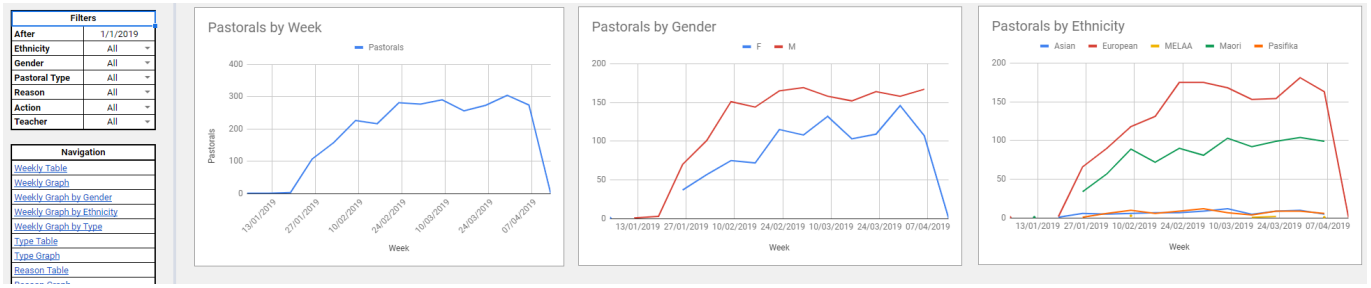
You can see the end outcome here by clicking here, and there are some selected screenshots below.

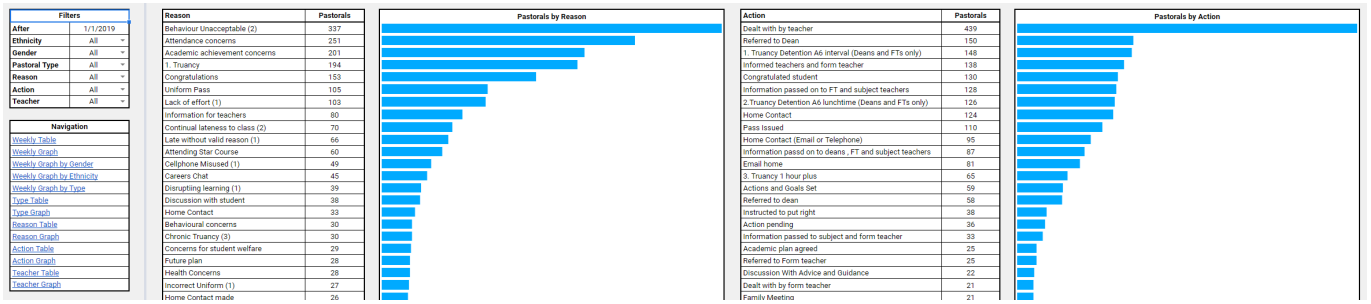


Filters	
After	1/1/2019
Ethnicity	All
Gender	All
Pastoral Type	All
Reason	All
Action	All
Teacher	All

Navigation
Weekly Table
Weekly Graph
Weekly Graph by Gender
Weekly Graph by Ethnicity
Weekly Graph by Type
Type Table
Type Graph

Week	Pastorals	F	M	Asian	European	MELAA	Maori	Pasifika	A	C	D	G	O	U
05/01/2019	1	1			1								1	
12/01/2019	1		1				1						1	
19/01/2019	3		3	1	2				1			1	1	
26/01/2019	107	37	70	6	66		34	1	31	16	3	16	18	23
02/02/2019	158	57	101	5	90		57	6	39	53	13	12	27	14
09/02/2019	226	75	151	6	118	3	89	10	44	64	30	35	34	19
16/02/2019	216	72	144	7	131		72	6	65	56	25	26	31	13
23/02/2019	281	115	165	7	175		90	9	92	94	20	26	39	10
02/03/2019	277	108	169	9	175		81	12	82	111	25	18	27	14
09/03/2019	290	132	158	12	168		103	7	96	80	21	19	47	27
16/03/2019	256	103	152	5	153	1	92	4	88	81	18	21	34	14
23/03/2019	273	109	164	9	154	2	99	9	71	76	30	25	42	29
30/03/2019	304	146	158	10	181		104	9	74	100	32	30	51	17
06/04/2019	274	107	167	5	163	1	99	6	92	72	29	29	37	15
13/04/2019	1	1			1				1					

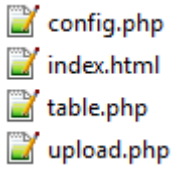




Step 1: Creating a Location Where we can upload the exports so Google Sheets can access them
KAMAR can export to a web service, but exporting directly to Google Sheets is quite tricky… to make it a bit easier we can upload the file to a web server that acts as an intermediary. We have set up ours on the same server that runs our portal.

You can download the files needed from https://portal.kc.school.nz/upload/upload.zip

There are 4 files in this folder:



You need to extract them into a folder called “upload” inside the root of your portal… most probably called “wwwroot”.

1. config.php
You need to edit the first two parts of this file… the storage location (if your portal setup is standard you probably don’t need to change this line) and the security key… just make this a random string that no-one else knows.

```php
<?php
// location of your kamarwritable folder... must end in a /
$storage = '../../kamarwritable/';

// the security key you want to use
$security = 'LtuoK3Ha9IHk';

// these lines prevent access to the files without the correct storage key
if(!array_key_exists('security',$_GET)){die("No Access");}
if($_GET['security']!=$security){die("No Access");}

// This code makes the directory if it doesn't already exist.
if (!file_exists($storage.'uploadstorage')) {
    mkdir($storage.'uploadstorage', 0777, true);
}
```

2. index.php
This file stops people seeing what is in this folder

3. table.php
This file presents the data so Google Sheets can access it

4. upload.php
This file allows you to upload the data.

Step 2: Writing the Script to get Data out of KAMAR

KAMAR has a neat custom script function that lets you build your own scripts.

We are going to build this script here:

```
# This script will export Student Results to the FileMaker 'Documents' folder on the server
# Caution should be used whenever running any scripts on server.
# Ensure you only use script steps which are compatible with running on Server.
#

# Set the variables needed
Set Variable [$school; Value: "demo"]
Set Variable [$sec; Value: "LtuoK3Ha9IHk"]
Set Variable [$fileName; Value: "pastoral.csv"]

# Select the layout to use as the base table for this export
Go to Layout [“Pastoral” (Pastoral_); Animation:None]

# Find All students in the current year.
Enter Find Mode [Pause:Off]
Set Field [Pastoral_::Date_of_Incident; ">= 1/1/" & Year( Get( CurrentDate ) )]
Perform Find []

# When running on server, we can only export to the 'Documents' or 'Temporary' folders.
# eg.  filewin:/C:/Program Files/FileMaker/FileMaker Server/Data/Documents/StudentExport.csv
#      filemac:/Macintosh HD/Library/FileMaker Server/Data/Documents/StudentExport.csv
Set Variable [$exportPath; Value: "filewin:" & Get( DocumentsPath ) & $fileName]

# This uses the standard FileMaker export function.  Useful for basic export fields.
# The 'File Type' is set to 'Merge' -  this is the same as 'csv' but includes the column headings.
Export Records [With dialog:Off; Create directories:Off; “$exportPath”; Windows (ANSI)]
Set Variable [$exportResult; Value: Get( LastError )]

Set Variable [$result; Value: If( $exportResult = 0 ; "Exported to: " & $exportPath ;  "Export Failed: " & idma_FMErrorCodeInFull( $exportResult ) & "¶Export Path" & $exportPath  )]

Set Variable [$upload; Value: idma_HTTP_PostFile( "https://portal.kc.school.nz/upload/upload.php?file=" & $school & "-" & $fileName & "&security=" & $sec; 443; $exportPath; "filewin:" & Get ( DocumentsPath ) & "uploadlog.txt")]

# KAMAR logs the Script Result to the  .../Databases/KAMAR_Logs/Custom/
Exit Script [Text Result:$result]
```

There are two ways you can do this…

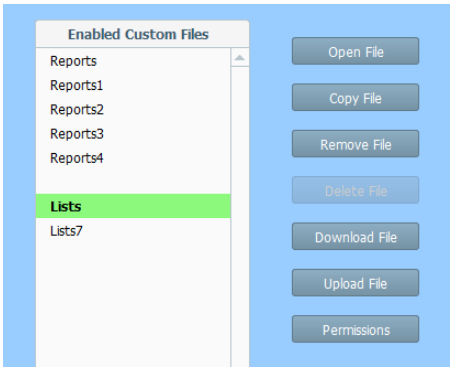
1: Upload this file here (easier)

On KAMAR go to Printing -> Progress Reports -> Global -> Custom Files. Click upload file and upload this file: https://portal.kc.school.nz/upload/k_Lists1.fmp12

If you do this you can skip to step 3.

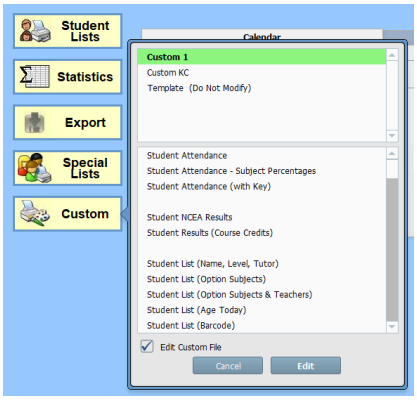
Or 2: Follow all these instructions (harder, but if you want to know what’s happening this might be helpful)

The first thing you should do if you do not already have a custom list file you are using for other things is copy the default custom list file. On KAMAR go to Printing -> Progress Reports -> Global -> Custom Files. Click on the “Lists” file and press Copy File.

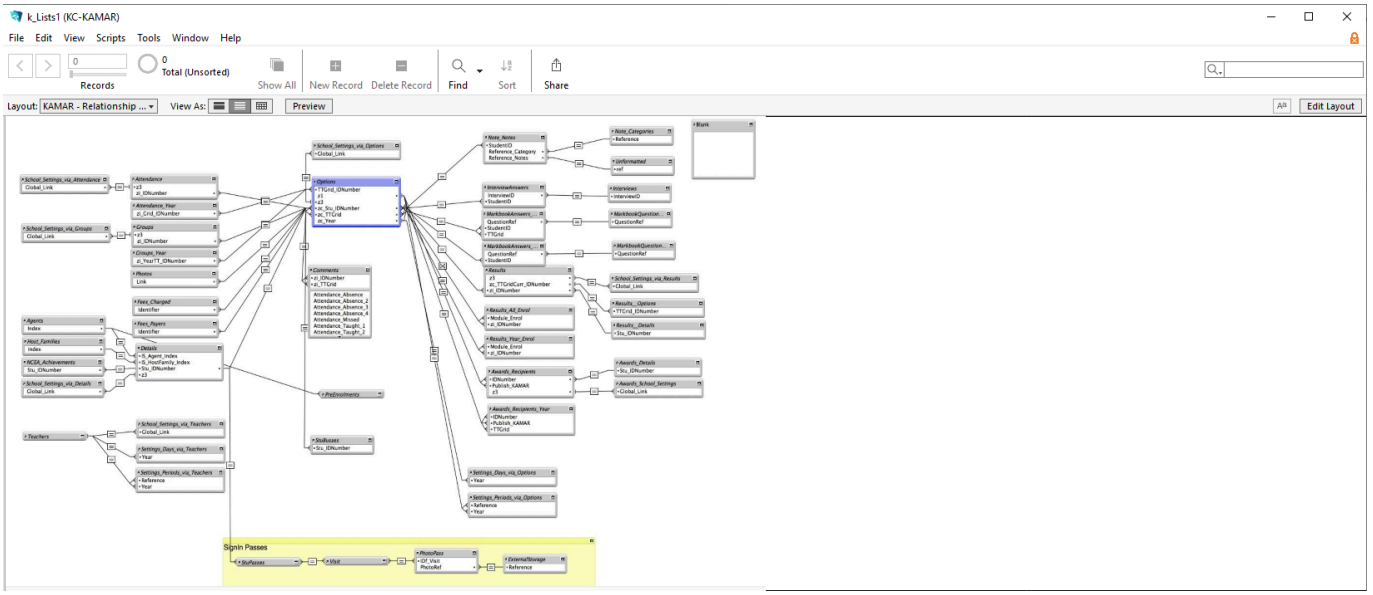


Copy it to whatever list you would like it to be in… for this tutorial I will use Lists1.

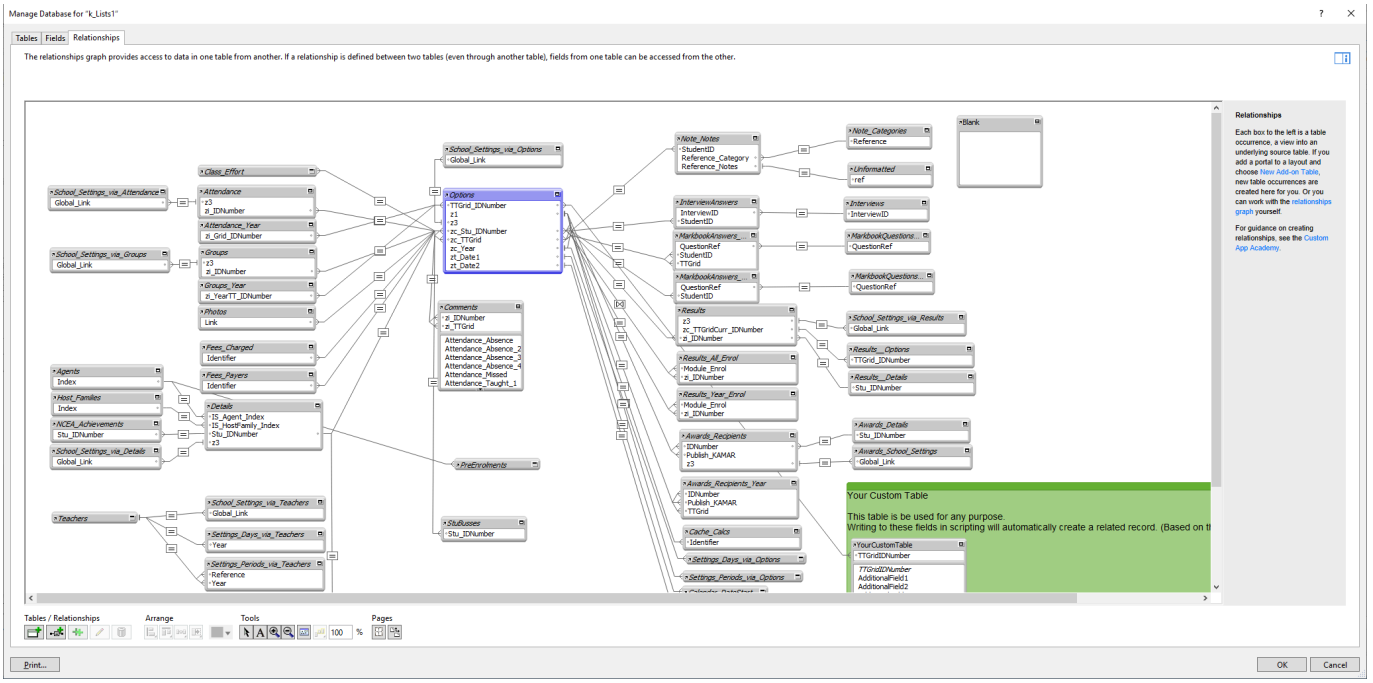
Then go to Printing -> Custom 1 and tick the “Edit Custom File” at the bottom of the popup



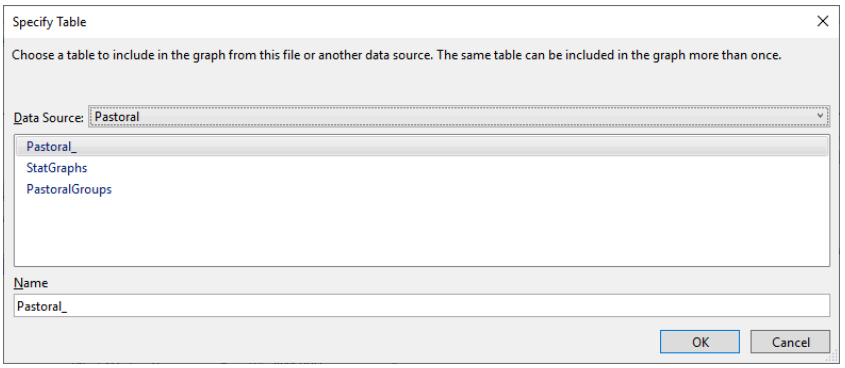
This will bring up the script workspace. Move this aside to find the layouts window that looks like this:



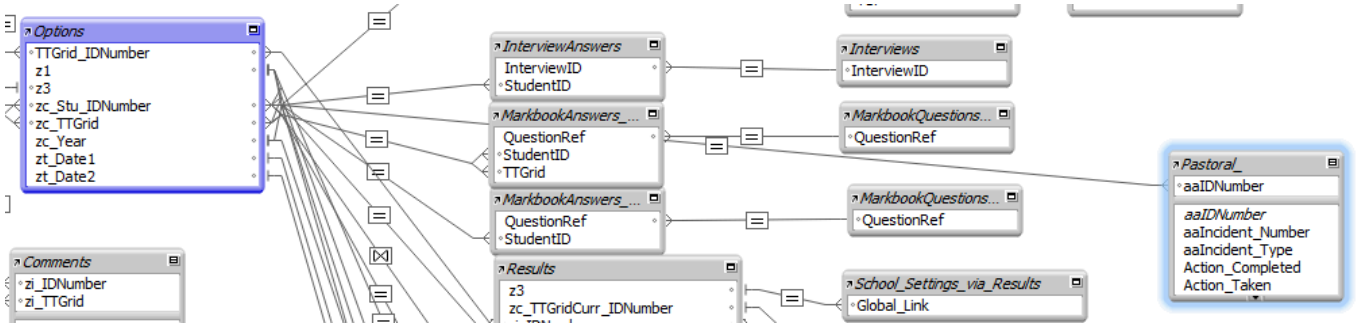
We need to add the pastoral table into our file. To do this go to File -> Manage -> Database. Click on the Relationships Tab at the top if this isn't already showing:



Press the bottom left button which allows you to add a table in, and choose the "Pastoral" data source, and the "Pastoral_" table, and press OK.

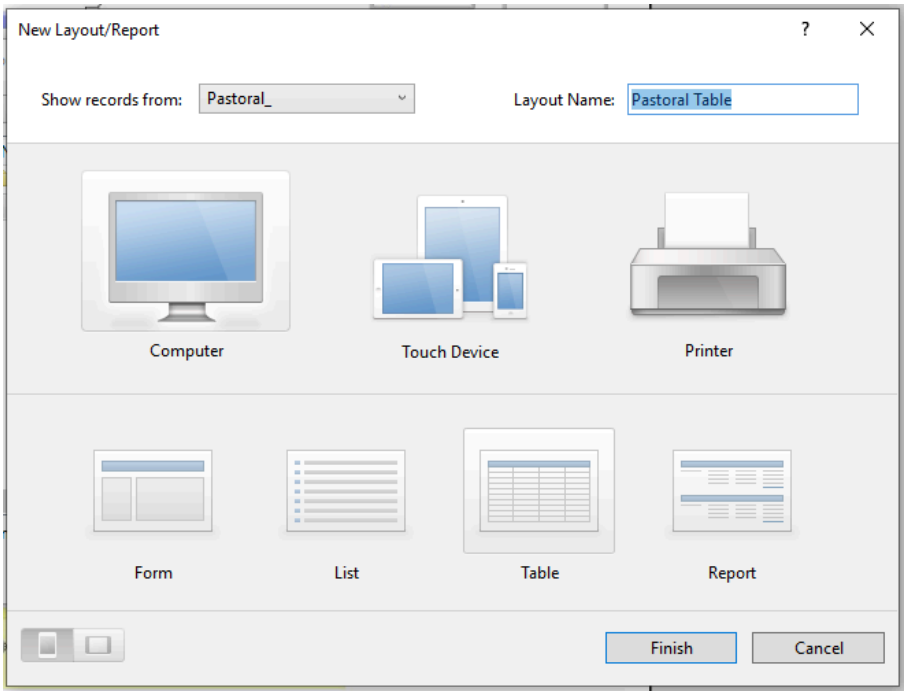


We then need to create the relationship between this and the Options table so that we can link the Ethnicity and Gender later on in our exports. To do this click on “aaIDNumber” and drag it on top of “zc_Stu_IDNumber”. It should now look like this:



We can now close the relationships window by pressing OK at the bottom and are ready to create our layout.

Go to File -> Manage -> Layouts and press “New”. Show records from “Pastoral_” and choose computer and table. Give it a name in the top right… I’ve called mine “Pastoral Table” and press Finished.



It may ask you to save the layout or add fields… you can just press okay or close those popups.

You then want to go back to the scripts window that we moved aside before. If you can’t find it just go to File -> Manage -> Scripts

On the left scroll down until you find the “Server Scripts” folder and duplicate “Sample_Server_Script_Export” by right clicking on it and pressing Duplicate.

Rename this “Pastoral_Export” by right clicking and pressing Rename.

Go to “Action__ServerScript” close to the top and double click to open it.

Add in lines 24-26… you can copy lines 25 and 26 from lines 21 and 22, just changing the “Sample_Server_Script_Export” to “Pastoral_Export”, but will need to type in line 24.

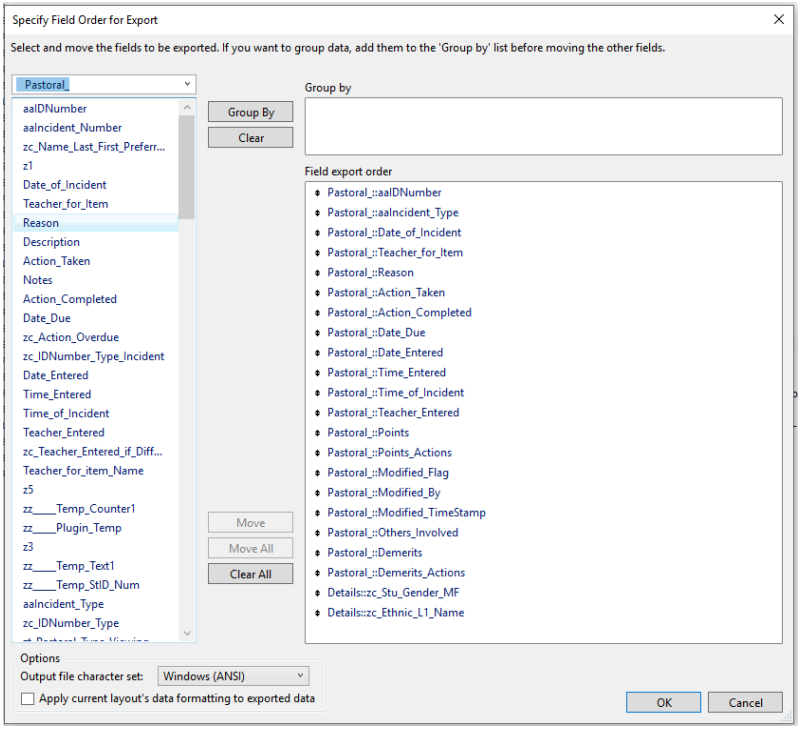
```
If [Trim( $sp ) = "StudentExport"]
    Perform Script [Specified:From list; “Sample_Server_Script_Export”; Parameter: ]
    Set Variable [$scriptResult; Value: "Script Export Result: " & Get( ScriptResult )]

Else If [Trim( $sp ) = "PastoralExport"]
    Perform Script [Specified:From list; “Pastoral_Export”; Parameter: ]
    Set Variable [$scriptResult; Value: "Script Export Result: " & Get( ScriptResult )]

Else If [not(IsEmpty( $sp ))]
    Set Variable [$scriptResult; Value: "Custom server script had parameter: " & $sp]
```

Then open the “Pastoral_Export” script by double clicking on it… you want to edit it so it looks like this:

```
# This script will export Student Results to the FileMaker 'Documents' folder on the server
# Caution should be used whenever running any scripts on server.
# Ensure you only use script steps which are compatible with running on Server.
#

# Set the variables needed
Set Variable [$school; Value: "demo"]
Set Variable [$sec; Value: "LtuoK3Ha9IHk"]
Set Variable [$fileName; Value: "pastoral.csv"]

# Select the layout to use as the base table for this export
Go to Layout [“Pastoral” (Pastoral_); Animation:None]

# Find All students in the current year.
Enter Find Mode [Pause:Off]
Set Field [Pastoral_::Date_of_Incident; ">= 1/1/" & Year( Get( CurrentDate ) )]
Perform Find []

# When running on server, we can only export to the 'Documents' or 'Temporary' folders.
# eg.  filewin:/C:/Program Files/FileMaker/FileMaker Server/Data/Documents/StudentExport.csv
#      filemac:/Macintosh HD/Library/FileMaker Server/Data/Documents/StudentExport.csv
Set Variable [$exportPath; Value: "filewin:" & Get( DocumentsPath ) & $fileName]

# This uses the standard FileMaker export function.  Useful for basic export fields.
# The 'File Type' is set to 'Merge' -  this is the same as 'csv' but includes the column headings.
Export Records [With dialog:Off; Create directories:Off; “$exportPath”; Windows (ANSI)]
Set Variable [$exportResult; Value: Get( LastError )]

Set Variable [$result; Value: If( $exportResult = 0 ; "Exported to: " & $exportPath ;  "Export Failed: " & idma_FMErrorCodeInFull( $exportResult ) & "¶Export Path" & $exportPath  )]

Set Variable [$upload; Value: idma_HTTP_PostFile( "https://portal.kc.school.nz/upload/upload.php?file=" & $school & "-" & $fileName & "&security=" & $sec; 443; $exportPath; "filewin:" & Get ( DocumentsPath ) & "uploadlog.txt")]

# KAMAR logs the Script Result to the  .../Databases/KAMAR_Logs/Custom/
Exit Script [Text Result:$result]
```

When you choose the export records:

You’ll need to specify this export order:



The table name that you need is given before the :: the field is after.

Step 3: Editing the Script
You need to change lines 7 and 8 to be your school name and the security you had from before.
You also need to edit line 31 so that it points to your portal.
You can then run the script by pressing the “Run” button at the top of the screen and it should

Step 4: Automating This Script
KAMAR has provided instructions on how to run these custom scripts here:
https://www.kamar.nz/setup/server/maintenance-scripts/custom-server-script
We have ours set to run on a nightly basis at 11:30pm.

Step 5: Putting into Google Sheets
You can make a copy of this Google Sheet:
https://docs.google.com/spreadsheets/d/13o53wd8F955Th8Gzu1imD7wJ4lWw2TlhGAmvQWgOHK0/edit#gid=0
Then the only change you need to make is on the “From KAMAR” tab, in the top left cell, just change the link so it points to your own portal.

```
=IMPORTHTML("https://portal.kc.school.nz/upload/table.php?file=demo-pastoral.csv&security=LtuoK3Ha9IHk&start=0","table",1)
```

A note about IMPORTHTML and the tabe.php script
Google Sheets doesn’t like lots of data coming in at once… so the table.php file limits it to 10,000 rows at once… if you need more rows than this then in row 10,001 you can just add in the formula again, but with start=10000 at the end instead of start=0.