The correct way to create category combos

Edit New Page

Jason Pickering edited this page on Dec 16, 2016 · 2 revisions

1. Create category options.

Category options should be individual groups for a given category. If you are creating a category called "Age <1,1-5,5+", you should have three options.

- <1
- 1-5
- 5+

You should not create category options which mix different categories. A category option such as "Female (0-14 years)" really mixes two separate categories: Gender and Age.

You should be very careful not to create duplicative category options. For instance

- <1
- Under 1

really refer to the same group. Ensure that you scan all of the category options very carefully before creating a new one.

2) Create the category.

Categories should consist of a group of related category options. Categories should represent all possible groups for a given data element.

As with category options, ensure that an existing category does not already exist.

Do not mix different categories together. For instance, a category of "Age + Gender" would not be appropriate. This is better represented through the use of a category combination, which can be used to combine two seperate categories together.

For complex categories, it may be useful to describe what the category contains.

Use generic descriptors for the categories which are specific enough to identify what the category is, but which are not overly specific to any one use.

Here are some bad examples of categories.

- Age + Gender : This is really a combination of two categories, not one.
- Age : This is too generic, and does not provide enough information to users as to what is contained in the category.
- Age (HIV Testing): The use of this name is too specific. Likely this category could be reused for other purposes, but it would appear from the name, that its sole purpose is for as specific indicator.
- Age (<1) : This would appear to contain a single option, and is not inclusive of all possible ages.
- Age (<1,>1,<5,5+): This category would appear to contain overlapping options, such as <1 and <5. Category options within a category should consist of mutually exclusive groups. There can be exceptions to this rule, but in general, its a good idea to ensure that all of the category options within a category are mutually exclusive.

Here is a good example of a category.

Pages 4

Home

HIA1 Top ten

Server notes

The correct way to create category combos

+ Add a custom sidebar

Clone this wiki locally

https://github.com/jason-

- Age (<1/1-5/<5/5+) : All possible ages are represented. From the name of the category, it is clear what different category options are available.

DHIS2 provides an option for "Disaggregation" or "Attributes" for categories. Typically, you should choose "Disaggregation", unless you plan on using the category for a data value attribute (such as funding partner).

DHIS2 provides an option for whether the category should be considered to be a "Data dimension". You should be very careful with this option, and not use this for all categories. Very important and widely used categories (such as Gender) are good candidates for enabling as a data dimension. In the analytics modules, like the Pivot Tables, enabling a category as a dimension will allow you to perform group set analysis on this category as a dimension. The reason you do not want to enable too many of these is because it will increase the analytics work load and can result in drastically increased disk usage.

3) Create a category combination.

Once you have created all of the necessary category options for your categories, and created all of the categories, you are ready to create a category combination.

This is a straight forward process, except, you should be **EXTREMELY** careful not to modify the category combination after you have created it. The reason for this is that each time you modify a category combination, new category option combinations may be created.

For instance, lets say you create a category combo called "Age + Gender". You then proceed to modify the "Age" category. In order for the new category option combos to be created you would need to perform a maintenance task to "Update category option combinations". This maintenance operation will remove any obsolete category option combinations as well as create any new ones which are required. However, this may have serious affect on existing data, validation rules and indicators (since they may be linked to specific category option combinations).

Be sure to check the category option combos of the category combo before you start to use it. Using an API call like

https://dev.zambiahmis.org/api/25/categoryCombos/LiKMwQP76K5.csv?fields=categoryOptionCombos[id,name]

where "LiKMwQP76K5" is the UID of the category option combo, you can get a CSV file of the category combo with all of the category option combinations which looks like this.

name	id
Male, 25-49 Yrs	NWvZjeSlJZJ
Male, 50+ Yrs	KMMieRs6MDT
Female, 25-49 Yrs	MmYqfXO2V8Q
Female, 1-4 Yrs	q5TEkQYPUOc
Female, <1 Yr	gbthkZTyA9c
Male, 15-19 Yrs	MpY1Botik4e
Male, <1 Yr	XwI1RomslEf
Female, 20-24 Yrs	s1mFUr4AThu
Male, 10-14 Yrs	GTIlVVypHFF
Female, 10-14 Yrs	Y3HvkleI2Q0
Female, 50+ Yrs	zgncZl79PlR
Male, 5-9 Yrs	MoxLdp9LdRN
Female, 5-9 Yrs	HNXMcXXiPIh
Male, 1-4 Yrs	dQiesY2GUie

name	id
Male, 20-24 Yrs	xtW3L0NldrN
Female, 15-19 Yrs	eyt8zcyKlOS

This category combo consists of two categories "Age (<1-50+,8)" which consists of 8 category options. Gender consists of two options (Male and Female). We would thus expect there to be 16 category option combos (2 * 8).

Before creating any data elements, indicators of validation rules with the new category combo, be sure that you confirm that all category option combos are correct before proceeding, as it an become difficult to rectify problems with these after they have been implemented and used.

+ Add a custom footer