

Tri alternatifs dans Spectre

Description

Nous avions déjà évoqué les tris dans les astuces précédentes :

https://blog.difrance.com/tri-naturel-versus-tri-alphanumerique/ https://blog.difrance.com/changer-lordre-de-tri-des-mois/ https://blog.difrance.com/dimension-triee-en-fonction-d-une-autre-colonne/

L'astuce actuelle se focalise sur la mise en place de tris via Spectre.

Nous souhaitons répondre aux questions suivantes :

Comment créer une Dimension avec les noms de mois débutant par avril ?

Comment définir les valeurs de Dimension dans un ordre chronologique, c'est à dire commençant par janvier?

La réponse est l'utilisation d'un tri alternatif.

Le tri naturel est le tri par défaut dans Spectre. Les chaines de caractères comme les noms de mois, par exemple "Janvier", sont triés par ordre alphabétique (cf. https://blog.difrance.com/tri-naturel-versus-tri-alphanumerique/).

_		Month_Nar	ne (💿 🔍
m	onth_names.txt $ imes$	Month_Name	
1	Month Name	Totals	
2	Jan	Apr	
3	Feb	Aug	
4	Mar	Dec	
5	Ann	Feb	
6	May	Jan	
7	Tur	Jul	
6	Tal	Jun	
0	342	Mar	
10	Aug	May	
10	Sep	Nov	
11	Oct	Oct	
12	Nov	Sep	
13	Dec		

Fichier texte

cBase

Un tri alternatif peut être appliqué pour modifier l'ordre d'affichage initial des valeurs de Dimension.

Un tri alternatif requiert l'utilisation de l'attribut **sort-by** et une colonne de tri dans le flux de données (ou une expression dans un cPlan contenant l'ordre associé.

Dans un script .build, l'option sort-by se trouve ici, au niveau de l'objet d'entrée texte :

onth_names.bt				
'ath (1)			- Comme	ents
Input Files			Enter you	ar comments for this object here
/data/month_names_dates	abit			
Columns (2)			Nopert	ies
Column	Туре Кеер	Remove	Alias 🚼 🏄	Quick Search
Month_Name	unknown -		Label	
Order Date	date - B	2 0	Nulls	
			Requir	ed Dimension
	\backslash		Sort B	/

Voici l'extrait du même script faisant référence à l'option sort-by vu depuis l'éditeur de texte de Workbench :

 text-input "/data/month_names_alt_sort.txt" { column "month_name" sort-by="alt_sort_month_names"
 }

Ci-dessous le fichier texte utilisé pour générer le cBase :

month_names_alt_sort.txt $ imes$						
1	Month N	lame	alt sort month names			
2	Jan 1					
3	Feb 2					
4	Mar 3					
5	Apr 4					
6	May 5					
7	Jun 6					
8	Jul 7					
9	Aug 8					
10	Sep 9					
11	Oct 10					
12	Nov 11					
13	Dec 12					

Ci-dessous le script complet utilisé pour générer le cBase :



Voici le résultat obtenu dans ProDiver :

month_nar	me [month_names_sorted.cbase-Dive A]	- • ×
month_name		
Totals		
Jan		
Feb		
Mar		
Apr		
May		
Jun		
Jul		
Aug		
Sep		
Oct		
Nov		
Dec		

Le tri alternatif peut être appliqué sur toute colonne de chaîne de caractères.

La colonne de tri utilisée doit être une chaîne de caractère ou un entier (Integer).

Voici un autre exemple de tri :

Catégorie âge patient :

- nouveau né
- enfant
- adolescent
- adulte
- personne âgée

Si l'on combine des cBases dans un cPlan, le tri doit être identique pour une même Dimension.





Vous pouvez :

- utiliser un fichier Lookup contenant la colonne de tri
- créer une colonne d'ordre de tri au sein du script

L'exemple ci-dessous montre la colonne de tri contenue dans le fichier d'entrée.

```
month names alt sort.build 	imes
   build {
 1 🗆
 2
      text-input "/data/month names dates.txt"
                                                     {
 3
        column "month name"
 4
        column "Order Date" type="date"
 5
        3
 6
 7
      add "MO sort order" `month(value("Order Date")
 8
      replace "Month name" `value("month name")`
                                                      301
 9
      output "/cbases/month names date sort.cbase"
10
11
12
```

La colonne de tri alternative doit tenir compte de toutes les valeurs de la colonne à trier.

Un tri alternatif peut-être ajouté dans un cPlan en utilisant une expression :



Tags

- 1. cBase
- 2. cplan
- 3. script
- 4. Spectre