When should the Aggr() function NOT be used?

Posted by Henric Cronström in QlikView Design Blog on Mar 11, 2013 11:12:49 PM

Last week Adam wrote a post about the Aggr() function and a Technical Brief about how to use it (Explaining the Aggr function). If you haven't read it, I strongly recommend that you do. It is an excellent paper describing a function that can be used for nested aggregations - complex calculations where you need to make a second aggregation based on the result of a first aggregation.

However, I have noticed in the discussion forum that Aggr() often is used when it isn't necessary. And since you get a performance penalty when you use the function, today's post is about ***discouraging*** you from using it when you don't need to.

So, when should you ***not*** use it?

1. Standard Aggregations
Standard, non-nested aggregations, e.g. "=Sum(Sales)" or "=Count(distinct OrderID)" can be used almost anywhere in QlikView, in charts, in text boxes and as labels in any object. They will work directly as they are. *Here you do not need any Aggr() function.*

If you want to calculate the aggregation several times, e.g. once per customer, just use the aggregation function in a chart and use Customer as dimension. Nothing else.

This seems obvious, but I have on multiple occasions seen developers use the Aggr() function in such situations. Totally unnecessary - and it will just increase response times.

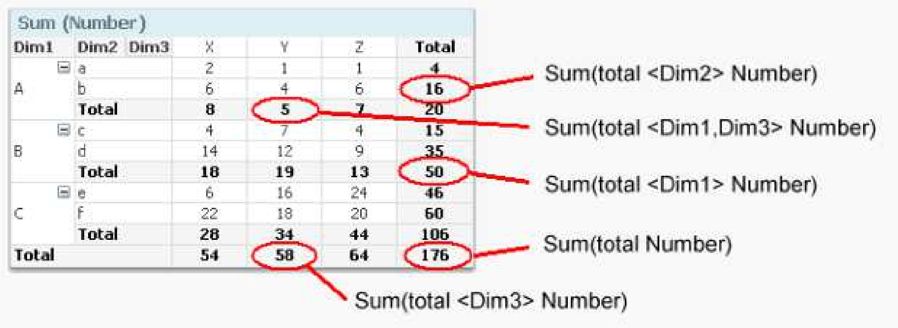
Bottom line: You should not use Aggr() for standard non-nested aggregations.

2. Calculation of a subtotal within the same chart
In some cases you need to use a subtotal from the same chart in your calculation. Then you should use the *total* qualifier inside your Sum() function (or other aggregation function). It will perform the given calculation *disregarding* one or several dimensions of the chart. It will however respect the selection.

For instance, the sales in percent can be calculated using

```
Sum( Sales ) / Sum( total Sales )
```

This expression will return the sum of sales for each possible dimensional value, as a fraction of all possible sales. In other words; "Sum(total Sales)" disregards *all* dimensions of the chart.



Sum (Number)

Dim1	Dim2	Dim3 X	Y	Z	Total
	⊟ a	2	1	1	4
A	b	6	4	6	16
	Total	8	5	7	20
	⊟ c	4	7	4	15
B	d	14	12	9	35
	Total	18	19	13	50
	⊟ e	6	16	24	46
C	f	22	18	20	60
	Total	28	34	44	106
Total		54	58	64	176

Bottom line: You should not use Aggr() for calculating subtotals that you can calculate with the total qualifier.

3. Calculation of an Average
If you can avoid the Aggr() function by writing your expression smarter, you should. One specific case is a standard average.

For example, say that you want to calculate the average order value. This is a nested aggregation: First you need to sum the sales value per order (an order can have several order lines), and then you need to average over the resulting set of sales values per order. In other words: You can calculate it using:

```
Avg( Aggr( Sum( Sales ), OrderID ) )
```

But since it is a linear calculation, you can also calculate the same number just by dividing with the number of orders:

```
Sum( Sales ) / Count( distinct OrderID )
```

… which in most cases is evaluated faster. Also, the latter expression is easier to understand for the person responsible for the application maintenance.

Bottom line: You should usually not use Aggr() for calculating an average that is a ratio of two separate aggregations.

4. Static aggregations
Sometimes you want to classify a dimensional value (customers, products, suppliers, etc.) using static aggregations, e.g. "Customers that only placed one order" or "Customers that bought goods at a total value higher than X". "Static" meaning that you do *not* want the classification to change as you make a selection.

In such a case, the aggregation should not be made in the UI, but rather when creating the data model and stored as a separate field.

Bottom line: Aggregations for classification of dimensional values should often be made by creating an attribute field in the script using a "Group By".

HIC

6813 Views Tags: aggr , subtotals , aggr_function , averages , aggregations , nested_aggregations , total_qualifier

15 Comments

paul johansson Mar 12, 2013 6:21 AM

Thanks for the clearification,

One HIC blog post a day keeps the doctor away

Like (2)

devududevudu Mar 12, 2013 2:45 PM

Hi All,

I have the data in design level like 2001-02,2002-03......2012-13,2013-14.

The requirement is like i have put **next and previou** buttons before and after the list box.

and i have to **disable the next** option for last year is like 2013-14 and **disable the previous button** like for 2001-02.

Thanks in advance.

Like (0)

Henric Cronström Mar 12, 2013 3:02 PM (in response to devududevudu)

You should look at "Semantic Links". See also the example "Presidents" that you get in your installation.

HIC

Like (0)

Ricardo Reaño Mar 12, 2013 5:35 PM

Great!

Now I'm going to to apply this recommendation on my application.

Thanks,
Ricardo

Like (0)

Kalyan Sundaram Mar 13, 2013 1:09 AM

hi henric,

thanks for your post...its clear now for the beginning developers

Like (0)

devududevudu Mar 15, 2013 5:41 AM

Could you plz expalin in detail am unable to catch my requirement.

Thanks inadvance.

Like (0)

John Cavoulas Mar 17, 2013 11:56 AM

One item missing from the AGGR white paper is the source data used. If the data were made known/availabe it would go further in explaining the AGGR function. Without knowing how the data looks, how does one apply any function appropriately?

Like (0)

Adam Bellerby Mar 18, 2013 8:47 AM (in response to John Cavoulas)

Hi,

The data used in the AGGR blog / paper is a really simple sales table, loaded from an INLINE load. A snapshot is below...

```
Data:
LOAD * INLINE [
    Country, Salesperson, Sales, Order
    UK, Bob, 99, 1
    UK, Bill, 87, 2
    UK, Fred, 86, 3
    UK, Sue, 58, 4
    UK, Barbara, 103, 5
    UK, Maureen, 90, 6
    USA, Bob, 199, 7
    USA, Bill, 187, 8
```

USA, Fred, 186, 9
USA, Sue, 158, 10
USA, Barbara, 203, 11
USA, Maureen, 190, 12
France, Bob, 98, 13
France, Bill, 67, 14
France, Fred, 59, 15
France, Sue, 38, 16
France, Barbara, 76, 17

Thanks.

Like
(0)

John Cavoulas Mar 18, 2013 9:22 AM (in response to Adam Bellerby)

Thanks Adam! Now perhaps I'll understand this function.

Like
(0)

qlik_techie May 20, 2013 7:08 AM

Hi HIC,

I've two tables:

group	prod
X	A
Y	A
Y	B
Y	C

prod	QTY
A	5
B	7
C	10

How can i get the following table using total<> method u've suggested ?

group	prod	QTY
Total		27
X	Total	5
	A	5
Y	Total	22
	A	5
	B	7
	C	10

Right now it is obtained using aggr

Regards

Like
(0)

Henric Cronström May 20, 2013 7:20 AM (in response to qlik_techie)

You can't.

The data is not consistent with the numbers you want to achieve. The Sum(QTY) clearly sums to 22 but you want to show 27 instead. You want to count the A=5 twice since A belongs to two groups. But there is no way that QlikView can "know" that.

However, you can use Aggr() to define this type of logic. Bottom line: You ***must*** use Aggr() to do this.

HIC

Like
(0)

qlik_techie May 20, 2013 7:33 AM (in response to Henric Cronström)

thanks for the prompt answer.

can u also throw some light on
1) In the screenshot below, do u use 'if' with dimensionality() to decide which expression should apply ?

unable to upload the picture (refer to the screenshot example in ur explanation with expressions and arrows at different levels)

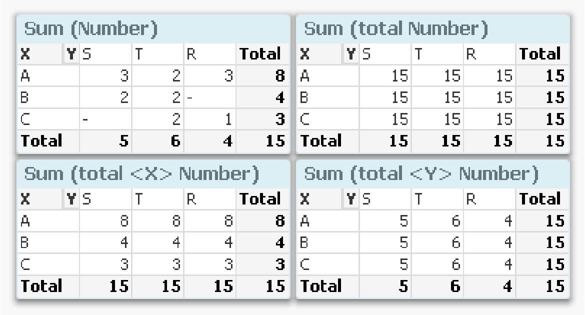
2) Can we list out the difference between aggr() and total<fld> method ?

Regards

Like
(0)

Henric Cronström May 20, 2013 8:41 AM (in response to qlik_techie)

You could probably write some complicated formula using if() and dimensionality(), but that would be to complicate it. Look at the charts below.



Sum (Number)

X	Y S	T	R	Total
A	3	2	3	8
B	2	2	-	4
C	-	2	1	3
Total	5	6	4	15

Sum (total Number)

X	Y S	T	R	Total
A	15	15	15	15
B	15	15	15	15
C	15	15	15	15
Total	15	15	15	15

Sum (total <X> Number)

X	Y S	T	R	Total
A	8	8	8	8
B	4	4	4	4
C	3	3	3	3
Total	15	15	15	15

Sum (total <Y> Number)

X	Y S	T	R	Total
A	5	6	4	15
B	5	6	4	15
C	5	6	4	15
Total	5	6	4	15

The top left chart contains the "real" numbers. The captions are the expressions with the total qualifier - exactly as I have written them. So by using the correct parameter within the brackets after the total you can specify which number you want.

The difference between aggr() and total:

- "Total" always uses the grouping of the chart dimension(s). So it always returns a number that is a subtotal - a number that already is calculated in the chart.
- The Aggr() is a way to define a different grouping than the one of the chart dimension: An additional grouping that is calculated before the calculations in the chart grouping are made.

HIC

Like
(0)

qlik_techie May 20, 2013 9:01 AM (in response to Henric Cronström)

Hi,

thanks again for ur time.
i understand that we can have 4 different charts with these expressions, but i wanted to ask how to use them together in one chart as you've shown in ur main post denoting each by arrow. Have u made it possible without 'if' and 'dimenisonality()'
or you just suggested that such expressions could replace the aggr functions. ?

Aggr vs total: i undertood that total <fieldname> where field name shud be a part of the chart dimension already i.e. other dimensions can't be used in the chart.

But Aggr can use other dimensions also.

I hope i understood this right

Like
(0)

qlik_techie May 20, 2013 9:05 AM (in response to qlik_techie)

i understood now what u meant by the arrows.

Like
(0)

Sections

Discussion Forums
Resources
Ideas

Groups
Blogs
New to QlikView

Blogs

Business Discovery
QlikView Design
Community Manager Blog

QlikView Support Updates
Technical Bulletin
All Blogs

QlikView Sites

QlikView.com
Qonnect Partner Portal
Customer Portal
QlikMarket
Demos

Trademarks Privacy Terms of Use