

TIME ANALYSIS

Sommaire

1	Time Complexity.....	2
1.1	Objective	2
1.2	Sum of periods	4
1.3	Current solutions	4
2	Data structure.....	5
2.1	Tables	5
2.2	The script	7
2.2.1	Current Period, Previous Period, Year Ago	8
2.2.2	Moving Total.....	9
2.2.3	YTD.....	10
3	Use of this structure.....	11
3.1	A dimension of the chart.....	11
3.2	The sets analysis.....	12
3.3	Axis must always be selected to 1 value.....	13

Foreword

That is my first blog and my first document.

I got really some difficulties to create a moving total, a YTD and a comparison versus the Year Ago compared to other BI tools. When I managed to create the chart I wanted to create, the solution is so simple that I decided to document it.

Have a nice reading

Thanks a lot

I did not find at first glance what I wanted to do. I would like to thanks people that have documented a way that made me succeed:

- G Wassenaar and this particular post : <http://community.qlikview.com/docs/DOC-4252>
- Henric Cronström

The version of this document is 1.0, the version of QlikView is 11.20 SR2..

1 Time Complexity

1.1 Objective

As usual, when the users have a BI tool, they want to aggregate, subtract, and compare the time values

- Raw data of the current period, Year ago, Quarter ago, 1 Period ago
- YTD, Fiscal YTD, YTD Year Ago, Fiscal YTD Year Ago
- A moving totalizing 3 months (for a moving quarter) or 12 months (for a moving year)

Raw data is not enough: users want also to compare the data of this period versus year ago or 1 month ago. We will see later how to build easily these measures.

I wanted to create such a chart (or table):

sum(VALUE)									
PERIOD	TIME ANALYSIS	Current Period	Period, Year Ago	Period, Previous Month	YTD	YTD, Year Ago	Rolling 3 periods	Rolling 12 periods	
January 2013		20	5	15	20	5	45	100	
February 2013		20	5	20	40	10	55	115	
March 2013		20	5	20	60	15	60	130	
April 2013		20	5	20	80	20	60	145	
May 2013		20	5	20	100	25	60	160	
June 2013		40	5	20	140	30	80	195	
July 2013		20	5	40	160	35	80	210	
August 2013		20	5	20	180	40	80	225	
September 2013		20	10	20	200	50	60	235	
October 2013		20	10	20	220	60	60	245	
November 2013		20	10	20	240	70	60	255	
December 2013		20	15	20	260	85	60	260	
January 2014		30	20	20	30	20	70	270	
February 2014		30	20	30	60	40	80	280	
March 2014		30	20	30	90	60	90	290	
April 2014		30	20	30	120	80	90	300	
May 2014		30	20	30	150	100	90	310	
June 2014		30	40	30	180	140	90	300	

The dimension TIME ANALYSIS contains different analyses the users want to perform with the application:

- **PERIOD** : data of the current period (data of the database)
- **PERIOD, Year Ago** : data of the same period but for the year ago. For example, data for June 2014 is 30, for the year ago it is 40 (we find this number in the June 2013 line, Current period Column)
- **PERIOD, Previous Period** : data of the previous period
- **YTD** : computes the data starting of the first month of the year, ie January. In June 2013, the YTD column contains the sum of the data from January 2013 to June 2013.
- **YTD, Year Ago**: computes the data starting of the first month of the year, ie January. In June 2013, the YTD column contains the sum of the data from January 2012 to June 2012
- **Rolling 3 periods** : computes the data for the previous 2 periods and the current period
- **Rolling 3 periods, Year Ago**: same but for the year ago

Some remarks:

- The dimension PERIOD contains all the months from January 2012 to the last month (here June 2014)
- The numeric expression is very simple in the chart : sum(VALUE)

Because TIME ANALYSIS is a dimension, it is possible to offer such a choice through a List Box:

The screenshot shows a BI tool interface. On the left, there are two list boxes: 'PROD_DESC' and 'TIME ANALYSIS'. The 'TIME ANALYSIS' list box is expanded, showing options like 'Current Period', 'YTD', and 'Rolling 3 periods'. The main area displays a data table for the expression 'sum(VALUE)'. The table has columns for 'PERIOD', 'TIME ANALYSIS', and four calculated metrics: 'Current Period', 'YTD', 'YTD, Year Ago', and 'Rolling 3 periods'. The data rows list months from January 2013 to June 2014.

PERIOD	TIME ANALYSIS	Current Period	YTD	YTD, Year Ago	Rolling 3 periods
January 2013		20	20	5	45
February 2013		20	40	10	55
March 2013		20	60	15	60
April 2013		20	80	20	60
May 2013		20	100	25	60
June 2013		40	140	30	80
July 2013		20	160	35	80
August 2013		20	180	40	80
September 2013		20	200	50	60
October 2013		20	220	60	60
November 2013		20	240	70	60
December 2013		20	260	85	60
January 2014		30	30	20	70
February 2014		30	60	40	80
March 2014		30	90	60	90
April 2014		30	120	80	90
May 2014		30	150	100	90
June 2014		30	180	140	90

1.2 Sum of periods

We can guess the links between the periods. If we concentrate on the February 2013 month :

- For current period TIME ANALYSIS : the TIME will be simply February 2013
- For Year Ago TIME ANALYSIS : the TIME will be simply February 2012
- For Previous Period TIME ANALYSIS : the TIME will be simply January 2013
- For YTD TIME ANALYSIS : the TIME will be January 2013 and February 2013
- For YTD Year Ago TIME ANALYSIS : the TIME will be January 2012 and February 2012
- For Rolling 3 Periods TIME ANALYSIS : the TIME will be December 2011 to February 2012
- For Rolling 3 Periods, Year Ago TIME ANALYSIS : the TIME will be December 2010 to February 2011, but the database does contain such values : it is possible not to link anything

1.3 Current solutions

To create such charts or tables, we can use functions (aggr, sum, above ...) and set analysis. However, the use of functions and sets analysis may be terrific:

Expression for a rolling period : `sum(aggr(rangesum(above(sum(VALUE),0,3)),TIME_KEY))`

We just need 5 functions!

“Rolling 3” column contains correct data. Sum(aggr(...)) contains cumulative data starting first line of the report. But there is some data before in the database in 2012!

Value / Units				
PERIOD	Value	sum(aggr (rangesum(abo...	Rolling 3	
January 2013	80			215
February 2013	80		160	235
March 2013	80		240	240
April 2013	60		220	220
May 2013	60		200	200
June 2013	120		240	240
July 2013	60		240	240
August 2013	60		240	240
September 2013	80		200	200

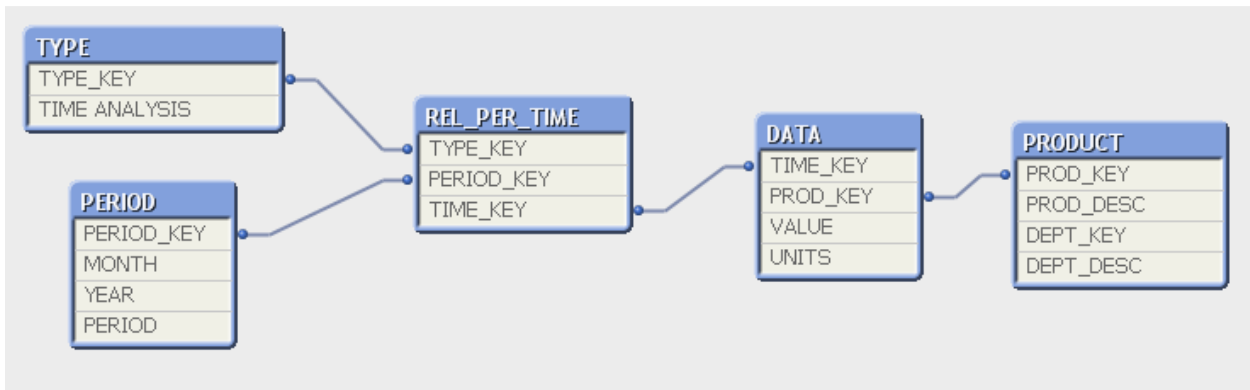
We need to add some Set Analysis and the expression becomes a little tricky:

`sum(aggr(rangesum(above(sum({<PERIOD=, MONTH=, YEAR=>} VALUE),0,3)),TIME_KEY))`

Value / Units				
PERIOD	Value	sum(aggr (rangesum(abo...	Rolling 3	
January 2013	80			215
February 2013	80		235	235
March 2013	80		240	240
April 2013	60		220	220
May 2013	60		200	200
June 2013	120		240	240

2 Data structure

2.1 Tables



The solution resides in **duplicating** the TIME_KEY (in data table) and the PERIOD_KEY shown to the user and included into charts and List Boxes.

The table **REL_PER_TIME** will contain the links mentioned above in section 1.2 (Sum of periods). The table **PERIOD** contains all information of period that we will show to the users. The **TYPE** table contains the the TIME ANALYSIS dimension (key and description).

Table REL_PER_TIME :

Remember the example of February 2013 (to make it simple: key 14 because it is the 14th month of the database):

A	B	C
PERIOD_KEY	TIME_KEY	TYPE_KEY
14	14	CURRENT
14	2	Y-1
14	13	P-1
14	13	YTD
14	14	YTD
14	1	YTD-1
14	2	YTD-1
14	12	MVTOT3
14	13	MVTOT3
14	14	MVTOT3

This screenshot has been done in Excel for me to show easily how the PERIOD_KEY 14 is related to the TIME_KEY. We will see later how to create by script a way to populate such a table.

Remarks:

Because the TIME ANALYSIS is a part of the model, we can say that it is a fixed dimension: it is “difficult” to modify it. But the true situation is quite the same for the users:

- The YTD starts always at the same date : it will not change soon
- The rolling periods used in reports are also very stable: the users do not want every day to change the rolling periods they use.

Table PERIODE_KEY :

KEY	MONTH	YEAR
1	January	2012
2	February	2012
3	March	2012
4	April	2012
5	May	2012
6	June	2012
7	July	2012
8	August	2012
9	September	2012
10	October	2012
11	November	2012
12	December	2012
13	January	2013
14	February	2013
15	March	2013
16	April	2013
17	May	2013
18	June	2013

The dimension PERIOD_KEY has exactly the same numbers as TIME_KEY (the one in the DATA table). The PERIOD dimension is a concatenation between MONTH and YEAR.

Table TYPE :

A	B	
TYPE_KEY	TIME ANALYSIS	
CURRENT	Current Period	
Y-1	Period, Year Ago	
P-1	Period, Previous Month	
YTD	YTD	
YTD-1	YTD, Year Ago	
MVTOT3	Rolling 3 periods	
MVTOT3 Y-1	Rolling 3 periods, Year Ago	
MVTOT12	Rolling 12 periods	
MVTOT12 Y-1	Rolling 12 periods, Year Ago	

This table contains the key and description for all the TIME ANALYSIS used in the application. Just an Excel file is enough for it.

2.2 The script

We can use a file, a SQL view but we can also add a script to populate the **REL_PER_TIME** table.

Just a simple recall with the 14th period (February 2013) :

A	B	C	D
PERIODE_KEY	TIME_KEY	TYPE_KEY	
14	14	CURRENT	
14	2	Y-1	
14	13	P-1	
14	13	YTD	
14	14	YTD	
14	1	YTD-1	
14	2	YTD-1	
14	12	ROLLING3	
14	13	ROLLING3	
14	14	ROLLING3	

I have created 3 SUB routines: 1 to populate the prior periods, 1 to populate the YTD, and 1 to populate the rolling periods.

2.2.1 Current Period, Previous Period, Year Ago

```
SUB LOAD_nAGO(vTypeTemporel, vOffset)
// F. AUNEZ, sep 2013
// Population of the relation PERIOD - TIME
// Creation of the AGO part (1 period ago, 1 year ago) and the CURRENT part
//
*****
*****

// the TIME values are in DATA table are numbered as of 1
// the first key is 1 for January 2012
// The PERIOD dimension has exactly the same numbers as TIME (mirroring dimension)
// the PERIOD 14 sur get a link on the TIME 13 (14-) for the previous period, et 2 for
the Year Ago (14-12)
REL_PER_TIME:
LOAD PERIOD_KEY,
    PERIOD_KEY-$(vOffset) as TIME_KEY,
    '$(vTypeTemporel)' as TYPE_KEY
RESIDENT PERIOD
where PERIOD_KEY > $(vOffset);

End sub
```

To call this sub routine in the main program:

```
call LOAD_nAGO('CURRENT', 0);
call LOAD_nAGO('P-1', 1);
call LOAD_nAGO('Y-1', 12);
```

Notice that the first argument is the key of the TIME ANALYSIS, you just need to be coherent with your Excel file. See page 7.

2.2.2 Moving Total

```
SUB LOAD_MVTOT(vTypeTemporel, vNbPeriods, vOffset)
// F. AUNEZ, sep 2013
// Population of the relation PERIODE - TIME
// Creation of the moving totals (and averages ...)
//
*****
*****

// each PERIOD key will get N links to TIME : the period itself + the N-1 TIME values
// for example:
// Feb 2013 (clef 14) for the moving total 3 months will have 3 links: 12 (Dec 2012),
// 13 (Jan 2013) et 14 (Fev 2013)
// feb 2013 (clef 14) for the moving total 12 months will have 12 links: 3 (March 2012)
// to 14 (Feb 2013)

//iter() is a way to loop
//for each PERIOD_KEY, we will create vNbPeriods lines to TIME (while iter() <=
vNbLines)
// we do not create an incomplete moving total (last condition)
Concatenate (REL_PER_TIME)
LOAD
    PERIOD_KEY,
    PERIOD_KEY + 1 - IterNo()-$(vOffset) as TIME_KEY,
    '$(vTypeTemporel)' as TYPE_KEY
Resident PERIOD
while IterNo() <= $(vNbPeriods) AND PERIOD_KEY >= $(vNbPeriods)+$(vOffset)
;

END SUB
```

And in the main program, 2 calls to this routine:

```
call LOAD_MVTOT ('MVTOT3', 3, 0)
call LOAD_MVTOT ('MVTOT12', 12, 0)
and for Year Ago:
call LOAD_MVTOT ('MVTOT3 Y-1', 3, 12)
call LOAD_MVTOT ('MVTOT12 Y-1', 12, 12)
```

2.2.3 YTD

A YTD starts in January and ends in the current period.

The keys for January are 1, 13, 25, ... $1+n*12$ (that is what the last div(..) part does

```
SUB LOAD_YTD(vTypeTemporel, vOffset)
// F. AUNEZ, sep 2013
// Population of the relation PERIODE - TIME
// Creation of the YTD
//
*****
*****

// Féb 2013 (key 14) will totalize the TIME Jan and Feb 2013
// June 2013 (key 18) will totalize from January (Key 13) to June (Key 18)
// For the YTD Year Ago, same totalization but for the Year Ago (12 keys back)

// we will use Iter() in order to loop back until January (Key 1, 13, 25 ...)
// that what the div( ....) part does
Concatenate (REL_PER_TIME)
LOAD
    PERIOD_KEY,
    PERIOD_KEY + 1 - IterNo() - $(vOffset) as TIME_KEY,
    '$(vTypeTemporel)' as TYPE_KEY
Resident PERIOD
while PERIOD_KEY - $(vOffset)-IterNo()+1 >= div( if( mod(PERIOD_KEY- $(vOffset),
12)=0, PERIOD_KEY- $(vOffset)-1, PERIOD_KEY- $(vOffset)), 12)*12 +1
;

END SUB
```

And in the main program:

```
call LOAD_YTD('YTD', 0)
call LOAD_YTD('YTD-1', 12)
```

Remark: some (many) companies do have a Fiscal Year to Date starting another month. By including another offset, you should be able to adapt this YTD process to your needs.

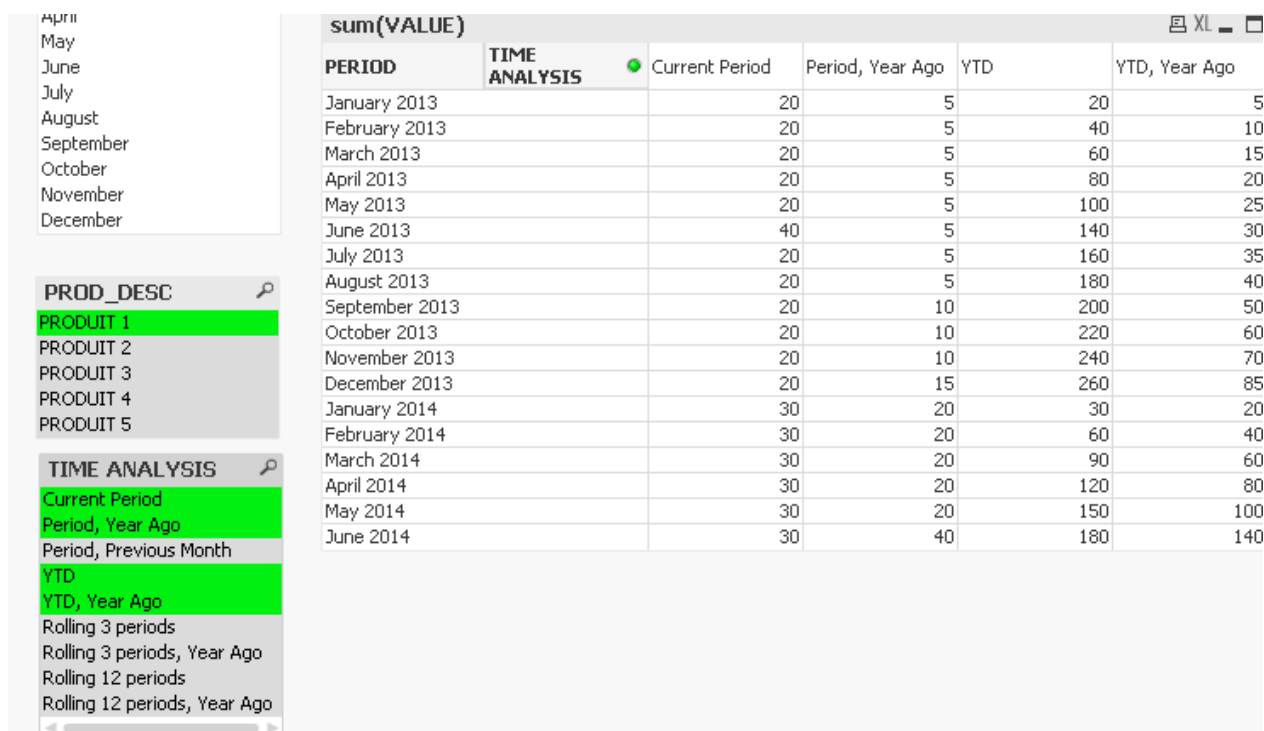
3 Use of this structure

The main advantage of such a model is the simplicity:

- To let the user choose between a YTD and current periods
- To do comparisons (and computations) between current and year ago periods
- To simplify all the needed expressions to do such tasks

3.1 A dimension of the chart

As shown in the beginning of this document, we can add a List Box to let the user choose which TIME ANALYSIS he wants to see:



The screenshot shows a user interface for data analysis. On the left, there are two list boxes: 'PROD_DESC' and 'TIME ANALYSIS'. 'PROD_DESC' lists 'PRODUIT 1' through 'PRODUIT 5', with 'PRODUIT 1' selected. 'TIME ANALYSIS' lists 'Current Period', 'Period, Year Ago', 'YTD', 'YTD, Year Ago', 'Rolling 3 periods', 'Rolling 3 periods, Year Ago', 'Rolling 12 periods', and 'Rolling 12 periods, Year Ago', with 'Current Period' selected. The main area displays a table titled 'sum(VALUE)' with columns for 'PERIOD', 'TIME ANALYSIS', 'Current Period', 'Period, Year Ago', 'YTD', and 'YTD, Year Ago'. The table contains data for months from January 2013 to June 2014.

PERIOD	TIME ANALYSIS	Current Period	Period, Year Ago	YTD	YTD, Year Ago
January 2013		20	5	20	5
February 2013		20	5	40	10
March 2013		20	5	60	15
April 2013		20	5	80	20
May 2013		20	5	100	25
June 2013		40	5	140	30
July 2013		20	5	160	35
August 2013		20	5	180	40
September 2013		20	10	200	50
October 2013		20	10	220	60
November 2013		20	10	240	70
December 2013		20	15	260	85
January 2014		30	20	30	20
February 2014		30	20	60	40
March 2014		30	20	90	60
April 2014		30	20	120	80
May 2014		30	20	150	100
June 2014		30	40	180	140

As we can see, the expression is very simple: sum(VALUE). If we wanted to compute:

- a price, we would have done: $\text{sum(VALUE)/sum(UNITS)}$
- the percentage a product represents in the total sales: $\text{sum(VALUE)/sum(\{<PROD_DESC=>\} VALUE)}$

3.2 The sets analysis

If we do not want a TIME ANALYSIS dimension in the chart because we want to focus on specific measures, like in this table:

Value / Units								
PERIOD	Value	Units	Value YTD	Difference Value YTD vs YA	Difference Value YTD vs YA (%)	Difference Units YTD vs YA (%)	Price YTD	
January 2012	5	20	5	5	-	-	0,25	
February 2012	5	20	10	10	-	-	0,25	
March 2012	10	40	20	20	-	-	0,25	
April 2012	15	40	35	35	-	-	0,29	
May 2012	15	40	50	50	-	-	0,31	
June 2012	15	40	65	65	-	-	0,33	
July 2012	15	40	80	80	-	-	0,33	
August 2012	15	40	95	95	-	-	0,34	
September 2012	20	40	115	115	-	-	0,36	
October 2012	20	40	135	135	-	-	0,38	
November 2012	20	40	155	155	-	-	0,39	
December 2012	25	40	180	180	-	-	0,41	
January 2013	40	80	40	35	700,00%	300,00%	0,50	
February 2013	40	80	80	70	700,00%	300,00%	0,50	
March 2013	40	80	120	100	500,00%	200,00%	0,50	
April 2013	40	80	160	125	357,14%	166,67%	0,50	
May 2013	40	80	200	150	300,00%	150,00%	0,50	
June 2013	80	160	280	215	330,77%	180,00%	0,50	
July 2013	40	80	320	240	300,00%	166,67%	0,50	
August 2013	40	80	360	265	278,95%	157,14%	0,50	
September 2013	40	80	400	285	247,83%	150,00%	0,50	
October 2013	40	80	440	305	225,93%	144,44%	0,50	
November 2013	40	80	480	325	209,68%	140,00%	0,50	
December 2013	40	80	520	340	188,89%	136,36%	0,50	
January 2014	60	120	60	20	50,00%	50,00%	0,50	
February 2014	60	120	120	40	50,00%	50,00%	0,50	

In this case, we can write a very simple set analysis. To do it, we can use either the TYPE_KEY dimension or it's description [TIME ANALYSIS]. For example, to write a YTD Year Ago, I can do:

Sum({<TYPE_KEY={ 'YTD-1' }>} VALUE)

ou

Sum({<[TIME ANALYSIS] ={'Year To Date, Year Ago'}>} VALUE)

For the price over the 3 months rolling :

Sum({<TYPE_KEY={ 'MVTOT3' }>} VALUE) / Sum({<TYPE_KEY={ 'MVTOT3' }>} UNITS)

To compute a Rolling 3 difference versus Year Ago, the syntax is also quite simple:

Sum({<TYPE_KEY={ 'MVTOT3' }>} VALUE) - Sum({<TYPE_KEY={ 'MVTOT3-1' }>} VALUE)

The difference in percent is quite the same. We can also add a test to return a computation only when it is correct:

If(TIME_KEY < 13, null(),

(Sum({<TYPE_KEY={ 'YTD' }>} VALUE) / Sum({<TYPE_KEY={ 'YTD-1' }>} VALUE)) - 1)

3.3 Axis must always be selected to 1 value

Take care to be sure that the computation made by QlikView concerns only 1 TIME ANALYSIS :

- **By introducing the TIME ANALYSIS dimension in the chart**
- **By using a Set Analysis in the expression**

If not, you may add a YTD with a Rolling Period Year Ago!!