

HIERARCHY IN A LISTBOX

Summary

1	Objective	2
1.1	The geographical hierarchy	2
1.2	The source file.....	3
2	Script.....	4
2.1	The HIERARCHY prefix	4
2.2	The ListBox	7
2.2.1	Creation	7
2.2.2	The result	8
2.3	The problem	8
2.4	What we need to solve that problem.....	9
2.5	The HierarchyBelongsTo prefix	10
3	The chart object.....	12
4	The complete Script.....	14

Foreword

I spent many years in other BI hierarchical tools. Naturally, I wanted to replicate some of the structures I met in the past. QlikView offers a way to display some hierarchical data but it is clearly NOT a hierarchical tool: more an associative one based on tables.

I explain in this document how to create the hierarchy and what you can do with it.

Have a nice reading

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

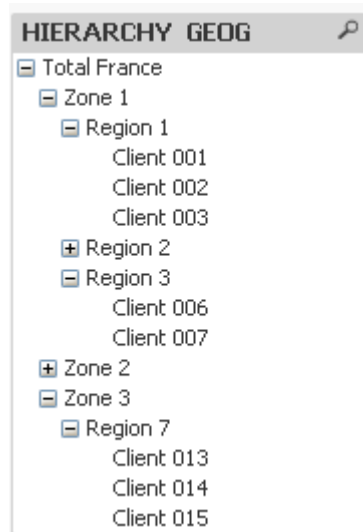
1 Objective

1.1 The geographical hierarchy

The company is structured such a way:



And the user wants to see the data such a way:



This hierarchical output can only be activated in the ListBox object.

1.2 The source file

For explanation purpose, I created a simple XLS file (that you can also download with this file):

	A	B	C
1	GEOG_HKEY	PARENT_KEY	GEOG_DESC
2	TOTALF		Total France
3	Z1	TOTALF	Zone 1
4	Z2	TOTALF	Zone 2
5	Z3	TOTALF	Zone 3
6	R1	Z1	Region 1
7	R2	Z1	Region 2
8	R3	Z1	Region 3
9	R4	Z2	Region 4
10	R5	Z2	Region 5
11	R6	Z2	Region 6
12	R7	Z3	Region 7
13	Client001	R1	Client 001
14	Client002	R1	Client 002
15	Client003	R1	Client 003
16	Client004	R2	Client 004
17	Client005	R2	Client 005
18	Client006	R3	Client 006

The source file is also hierarchical:

- The GEOG key
- The parent key
- The description of the GEOG
- Optionally, you can add the Parent description

2 Script

2.1 The HIERARCHY prefix

Before the LOAD statement, you need to describe how the table storing the hierarchy will be built :

```
Hierarchy(NodeKey, ParentKey, NodeDescription[, other optional arguments])
```

Where:

NodeKey: the field containing the key of GEOG (the child not the parent)

Parentkey: the field containing the key of the parent of the GEOG

NodeDescription: the field containing the description of GEOG (the child not the parent). This name will be used to create all the fields containing the description of the ancestors.

The 5 other arguments are also useful and according to me should be used. In this order:

ParentName: the name of the field that will contain the description of the parent

PathSource: The path in QlikView is a string containing one field per ancestor down to the node. A region will have this path: Total; Zone x; Region Y. A client will have this path: Total; Zone Y; Region Z; Client W. You can either build the path with the keys or with the descriptions of the ancestors.

PathName: the name of the field that will contain the path. You will have to use this field in the ListBox to get a hierarchical representation.

PathDelimiter: the letter to separate the different fields. Prefer a letter that will NOT be inside the key or description of the node. I use the pipe: |

HierarchyDepth: the name of the field that will contain the depth of the node. If a GEOG (like a region) has 2 ancestors (the zone and the total), it's depth will be 3. The depth is equal to the number of ancestors+1.

Example 1:

```
GEOGH:
HIERARCHY(GEOG_KEY, PARENT_KEY, GEOG_DESC)
LOAD GEOG_HKEY as GEOG_KEY,
      PARENT_KEY,
      GEOG_DESC
FROM
DataHierarchy.xlsx
(ooxml, embedded labels, table is HIERARCHY);
```

That creates a GEOGH table:

GEOGH
GEOG_KEY
PARENT_KEY
GEOG_DESC
GEOG_DESC1
GEOG_DESC2
GEOG_DESC3
GEOG_DESC4

Populated like that:

GEOG_DESC1	GEOG_DESC2	GEOG_DESC3	GEOG_DESC4	GEOG_KEY	PARENT_KEY	GEOG_DESC
Total France	-	-	-	TOTALF	-	Total France
Total France	Zone 1	-	-	Z1	TOTALF	Zone 1
Total France	Zone 2	-	-	Z2	TOTALF	Zone 2
Total France	Zone 3	-	-	Z3	TOTALF	Zone 3
Total France	Zone 1	Region 1	-	R1	Z1	Region 1
Total France	Zone 1	Region 2	-	R2	Z1	Region 2
Total France	Zone 1	Region 3	-	R3	Z1	Region 3
Total France	Zone 2	Region 4	-	R4	Z2	Region 4
Total France	Zone 2	Region 5	-	R5	Z2	Region 5
Total France	Zone 2	Region 6	-	R6	Z2	Region 6
Total France	Zone 3	Region 7	-	R7	Z3	Region 7
Total France	Zone 1	Region 1	Client 001	Client001	R1	Client 001
Total France	Zone 1	Region 1	Client 002	Client002	R1	Client 002
Total France	Zone 1	Region 1	Client 003	Client003	R1	Client 003
Total France	Zone 1	Region 2	Client 004	Client004	R2	Client 004
Total France	Zone 1	Region 2	Client 005	Client005	R2	Client 005
Total France	Zone 1	Region 3	Client 006	Client006	R3	Client 006

You can see that the GEOG_DESCx fields contain the description of the ancestors at the Xth level.

Example 2:

```
GEOGH:  
HIERARCHY(GEOG_KEY, PARENT_KEY, GEOG_DESC, [GEOG PARENT NAME], GEOG_DESC, [HIERARCHY  
GEOG], '|', 'HIERARCHY DEPTH')  
LOAD GEOG_HKEY as GEOG_KEY,  
PARENT_KEY,  
GEOG_DESC  
FROM  
DataHierarchy.xlsx  
(ooxml, embedded labels, table is HIERARCHY);
```

That creates such a table:

GEOGH
GEOG_KEY
PARENT_KEY
GEOG_DESC
GEOG_DESC1
GEOG_DESC2
GEOG_DESC3
GEOG_DESC4
GEOG PARENT NAME
HIERARCHY GEOG
HIERARCHY DEPTH

Populated like that:

HIERARCHY GEOG	GEOG_KEY	PARENT_KEY	GEOG_DESC	HIERARCHY DE...
Total France	TOTALF	-	Total France	1
Total France Zone 1	Z1	TOTALF	Zone 1	2
Total France Zone 2	Z2	TOTALF	Zone 2	2
Total France Zone 3	Z3	TOTALF	Zone 3	2
Total France Zone 1 Region 1	R1	Z1	Region 1	3
Total France Zone 1 Region 2	R2	Z1	Region 2	3
Total France Zone 1 Region 3	R3	Z1	Region 3	3
Total France Zone 2 Region 4	R4	Z2	Region 4	3
Total France Zone 2 Region 5	R5	Z2	Region 5	3
Total France Zone 2 Region 6	R6	Z2	Region 6	3
Total France Zone 3 Region 7	R7	Z3	Region 7	3
Total France Zone 1 Region 1 Client 001	Client001	R1	Client 001	4
Total France Zone 1 Region 1 Client 002	Client002	R1	Client 002	4
Total France Zone 1 Region 1 Client 003	Client003	R1	Client 003	4
Total France Zone 1 Region 2 Client 004	Client004	R2	Client 004	4
Total France Zone 1 Region 2 Client 005	Client005	R2	Client 005	4
Total France Zone 1 Region 3 Client 006	Client006	R3	Client 006	4

You can see the path built in the **HIERARCHY GEOG** field:

- The path is the concatenation of the fields from the top of the hierarchy to the node itself
- Use of description to populate the path : GEOG_DESC argument
- Fields separated with a pipe : '|' argument

N.B:

- we will use this field HIERARCHY GEOG to build the hierarchy in the List Box
- all the fields will appear in the different fields to build the objects : you may rename them in order to have more explicit fields

```
RENAME FIELD GEOG_DESC1 to TOTAL;  
RENAME FIELD GEOG_DESC2 to ZONE;  
RENAME FIELD GEOG_DESC3 to REGION;  
RENAME FIELD GEOG_DESC4 to CLIENT;
```

TOTAL	ZONE	REGION	CLIENT	GEOG PARENT ...	HIERARCHY GE...	GEOG_KEY
Total France	-	-	-	-	Total France	TOTALF
Total France	Zone 1	-	-	Total France	Total France Zone	Z 1
Total France	Zone 2	-	-	Total France	Total France Zone	Z 2
Total France	Zone 3	-	-	Total France	Total France Zone	Z 3
Total France	Zone 1	Region 1	-	Zone 1	Total France Zone	R 1
Total France	Zone 1	Region 2	-	Zone 1	Total France Zone	R 2
Total France	Zone 1	Region 3	-	Zone 1	Total France Zone	R 3
Total France	Zone 2	Region 4	-	Zone 2	Total France Zone	R 4
Total France	Zone 2	Region 5	-	Zone 2	Total France Zone	R 5

2.2 The ListBox

2.2.1 Creation

The screenshot shows the 'List Box Properties [HIERARCHY GEOG]' dialog box. The 'Field' dropdown is set to 'HIERARCHY GEOG'. The 'Show as TreeView' checkbox is checked and highlighted with a blue box. The 'With Separator' checkbox is also checked. The 'Object ID' is 'LB01'. The 'Search' section has 'Include Excluded Values in Search' set to '<use default>' and 'Default Search Mode' set to '<use default>'. The 'Print' section has 'Layout Dependent Print' checked. The 'General' tab is selected.

To create the ListBox, you just need to:

- Use the field containing the path from top to the node

- Mark the "Show as TreeView"
- Enter the separator if you use a different letter than the default one

2.2.2 The result

sum(VALUE)		January	February	March	April	May	June
PRODUCT	MONTH						
Product 1			26	27	28	29	30
Product 2			31	32	33	34	35
Product 3			31	32	33	34	35

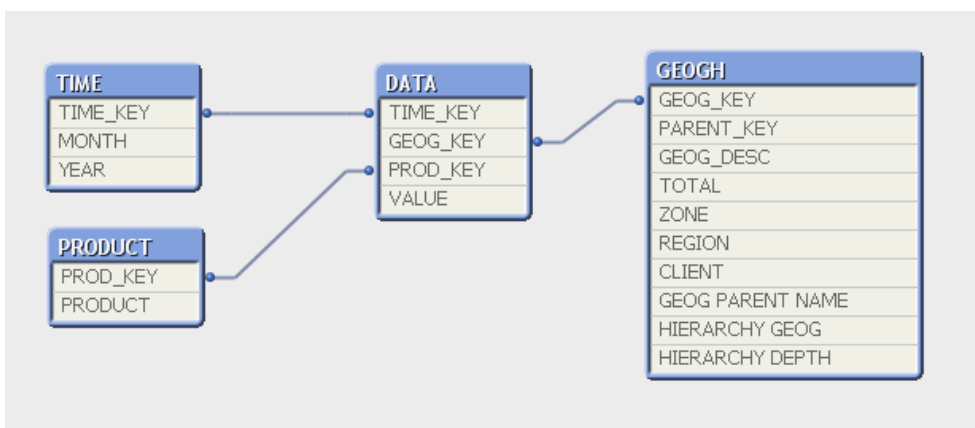
We can drill down the GEOG hierarchy and click on a client: we see the result.

2.3 The problem

But, when we click on an aggregated value (like a zone or a region, we have nothing):

sum(VALUE)	
PRODUCT	MONTH

In fact, we have loaded into the data tables only detailed data (at client level). So, the link between the Region 1 and the data table is just Nothing.

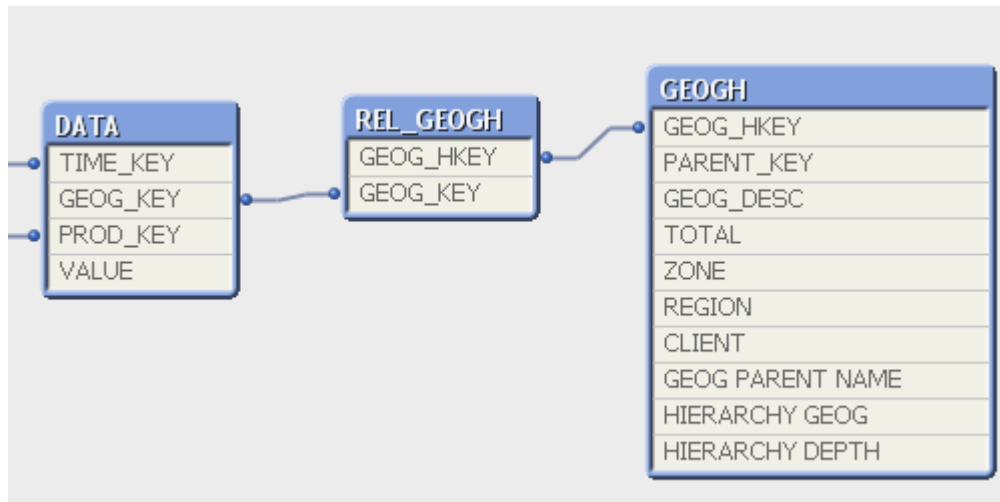


2.4 What we need to solve that problem

We need to:

- unlink the GEOG shown to the user and the GEOG storing the data: we rename a field
- create a table that will sum the different GEOG of the data table

The data model should be like that:



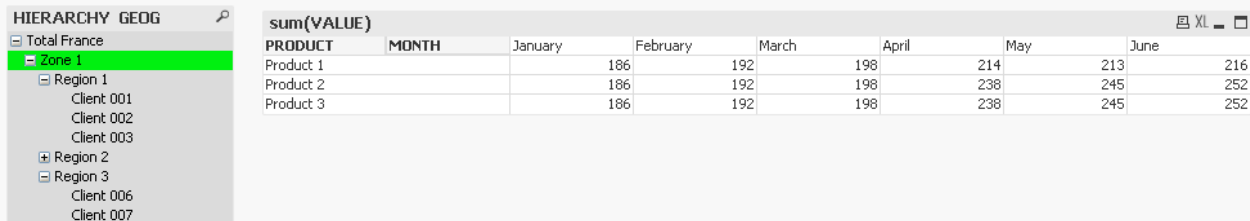
We need for each GEOG a link to all its descendants keys that contain data. Also the bottom level (the clients) should contain a link. REL_GEOGH is the table doing the link between aggregated value (like a zone) and the descendants that have data (the clients):

A	B
GEOG_HKEY	GEOG_KEY
R1	Client001
R1	Client002
R1	Client003
R2	Client004
R2	Client005
R3	Client006
R3	Client007
Z1	Client001
Z1	Client002
Z1	Client003
Z1	Client004
Z1	Client005
Z1	Client006
Z1	Client007
Client001	Client001
Client002	Client002
Client003	Client003
Client004	Client004
Client005	Client005
Client006	Client006
Client007	Client007

For example, the Region R1 is related to the clients CLIENT001, CLIENT002, CLIENT003. The Zone Z1 is related to more clients : from 001 to 007.

Note that the clients are also related to themselves.

And the result is what we want to get (aggregated data and link with the ListBox):



HIERARCHY GEOG		sum(VALUE)							
Total France		PRODUCT	MONTH	January	February	March	April	May	June
- Zone 1		Product 1		186	192	198	214	213	216
Region 1		Product 2		186	192	198	238	245	252
Client 001		Product 3		186	192	198	238	245	252
Client 002									
Client 003									
Region 2									
Region 3									
Client 006									
Client 007									

2.5 The HierarchyBelongsTo prefix

If we do not want to compute these descendants, we can use this prefix before the LOAD statement. You need to describe how the table storing the hierarchy will be built :

HierarchyBelongsTo(NodeKey, ParentKey, NodeDescription, AncestorKey, Ancestor Description, [Depth])

Where:

NodeKey: the field containing the key of GEOG (the child not the parent)

Parentkey: the field containing the key of the parent of the GEOG

NodeDescription: the field containing the description of GEOG (the child not the parent). This name will be used to create all the fields containing the description of the ancestors.

AncestorKey: the name of the field containing the key of the ancestor

AncestorDescription: the name of the field containing the name of the ancestor

Depth: the optional name of the field containing the depth of the NodeKey

Example 1:

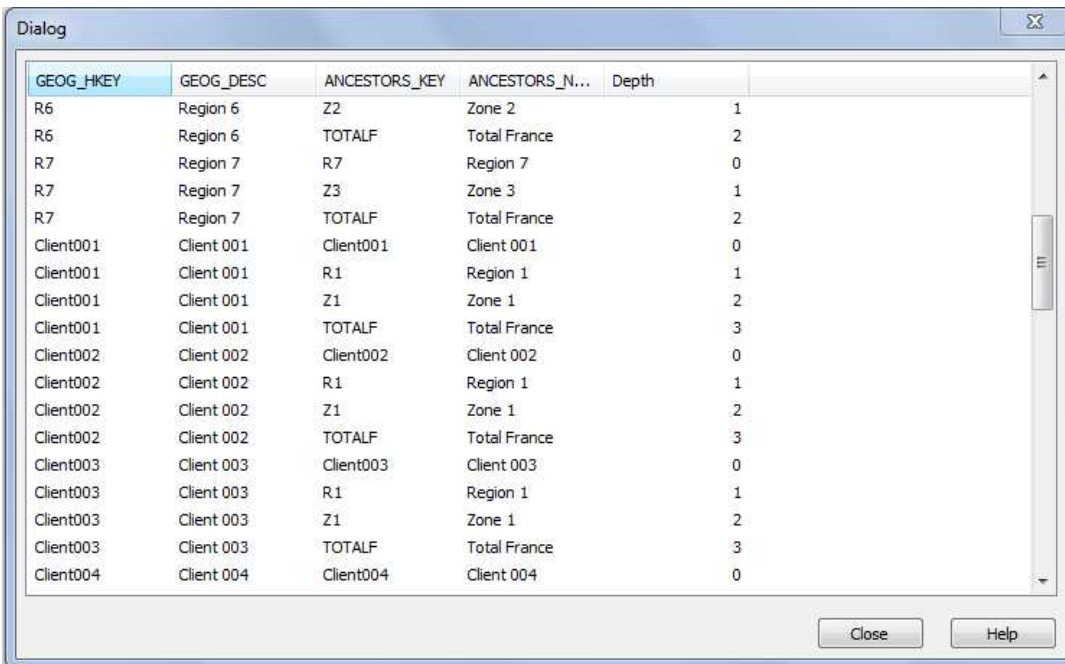
GEOGH2 :

```
HierarchyBelongsTo(GEOG_HKEY, PARENT_KEY, GEOG_DESC, 'ANCESTORS_KEY',
'ANCESTORS_NAME', 'Depth')
LOAD GEOG_HKEY,
      PARENT_KEY,
      GEOG_DESC
FROM
DataHierarchy.xlsx
(ooxml, embedded labels, table is HIERARCHY);
```

That creates such a table:

GEOGH2
GEOG_HKEY
GEOG_DESC
ANCESTORS_KEY
ANCESTORS_NAME
Depth

Populated like that:



GEOG_HKEY	GEOG_DESC	ANCESTORS_KEY	ANCESTORS_N...	Depth
R6	Region 6	Z2	Zone 2	1
R6	Region 6	TOTALF	Total France	2
R7	Region 7	R7	Region 7	0
R7	Region 7	Z3	Zone 3	1
R7	Region 7	TOTALF	Total France	2
Client001	Client 001	Client001	Client 001	0
Client001	Client 001	R1	Region 1	1
Client001	Client 001	Z1	Zone 1	2
Client001	Client 001	TOTALF	Total France	3
Client002	Client 002	Client002	Client 002	0
Client002	Client 002	R1	Region 1	1
Client002	Client 002	Z1	Zone 1	2
Client002	Client 002	TOTALF	Total France	3
Client003	Client 003	Client003	Client 003	0
Client003	Client 003	R1	Region 1	1
Client003	Client 003	Z1	Zone 1	2
Client003	Client 003	TOTALF	Total France	3
Client004	Client 004	Client004	Client 004	0

So, this table is the one that contains the link between the GEOG shown to the user (ANCESTOR_KEY or NAME) and the GEOG containing the data



But it does not contain the path used by the ListBox.

We just need to:

- Populate the table REL_GEOGH with the table created by HierarchyBelongsTo
- Drop this table

REL_GEOGH:

```
LOAD ANCESTORS_KEY as GEOG_HKEY,  
     GEOG_HKEY as GEOG_KEY
```

```
RESIDENT GEOGH2;
```

```
DROP table GEOGH2;
```

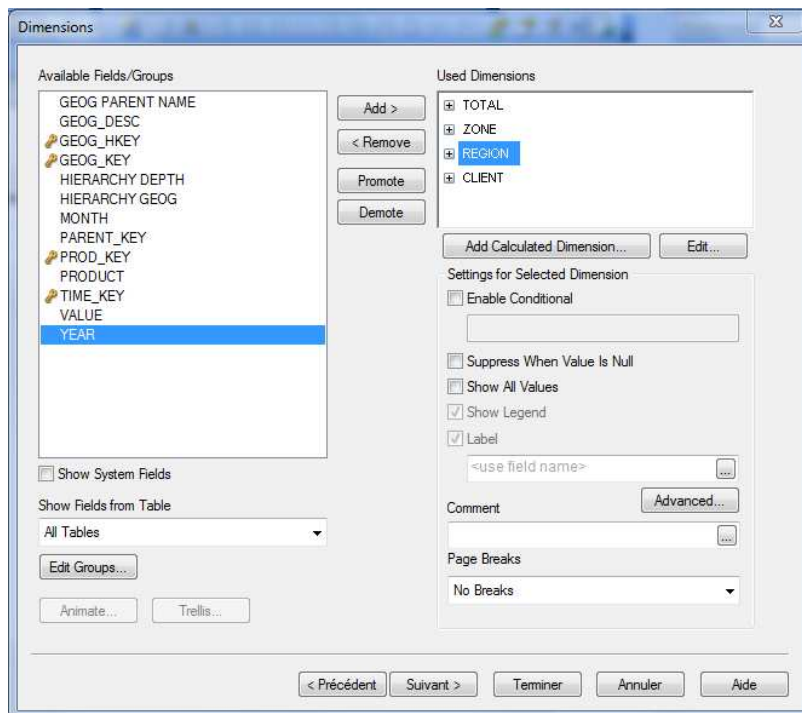
3 The chart object

To build a chart, you cannot use the HIERARCHY GEOG dimension. There is no way to get a hierarchy as in the ListBox on the left:

HIERARCHY GEOG	sum(VALUE)
Total France	4129
Total France Zone 1	3841
Total France Zone 1 Region 1	1599
Total France Zone 1 Region 1 Client 001	483
Total France Zone 1 Region 1 Client 002	543
Total France Zone 1 Region 1 Client 003	573
Total France Zone 1 Region 2	996
Total France Zone 1 Region 2 Client 004	543
Total France Zone 1 Region 2 Client 005	453
Total France Zone 1 Region 3	1246
Total France Zone 1 Region 3 Client 006	1023
Total France Zone 1 Region 3 Client 007	223
Total France Zone 2	288
Total France Zone 2 Region 4	288
Total France Zone 2 Region 4 Client 008	288



You need to use the different levels of the hierarchy (it was a good idea to rename the fields):



HIERARCHY GEOG				
[-] Total France				
[-] Zone 1				
[-] Region 1				
Client 001				
Client 002				
Client 003				
[+] Region 2				
[-] Region 3				
Client 006				
Client 007				
[+] Zone 2				
[-] Zone 3				
[-] Region 7				
Client 013				
Client 014				
Client 015				

sum(VALUE)				
TOTAL	ZONE	REGION	CLIENT	sum(VALUE)
Total France	Zone 1	Region 1	Client 001	483
			Client 002	543
			Client 003	573
	-	-	1599	
	Zone 1	Region 2	[+]	996
		Region 3	[+]	1023
	-	-	223	
	-	-	1246	
	[+]	[+]	3841	
	Zone 2	Region 4	[+]	288
	-	-	[+]	288
-	-	[+]	4129	

But if you select a GEOG, it naturally impacts the table:

HIERARCHY GEOG				
[-] Total France				
[-] Zone 1				
[-] Region 1				
Client 001				
Client 002				
Client 003				
[+] Region 2				
[-] Region 3				
Client 006				
Client 007				
[+] Zone 2				
[-] Zone 3				
[-] Region 7				
Client 013				
Client 014				
Client 015				

sum(VALUE)				
TOTAL	ZONE	REGION	CLIENT	sum(VALUE)
Total France	Zone 1	Region 3	Client 006	1023

It is why I prefer the hierarchy to get used in page as a filter.

4 The complete Script

```
//Fabrice AUNEZ : sep 2013

// Hierarchy in QlikView

SET ThousandSep=' ';
SET DecimalSep=',';
SET MoneyThousandSep=' ';
SET MoneyDecimalSep=',';
SET MoneyFormat='# ##0,00 €;-# ##0,00 €';
SET TimeFormat='hh:mm:ss';
SET DateFormat='DD/MM/YYYY';
SET TimestampFormat='DD/MM/YYYY hh:mm:ss[.fff]';
SET MonthNames='janv.;févr.;mars;avr.;mai;juin;juil.;août;sept.;oct.;nov.;déc.';
SET DayNames='lun.;mar.;mer.;jeu.;ven.;sam.;dim.';

PRODUCT:
LOAD KEY as PROD_KEY,
     DESC as PRODUCT
FROM
DataHierarchy.xlsx
(ooxml, embedded labels, table is PROD);

TIME:
LOAD KEY as TIME_KEY,
     MONTH,
     YEAR
FROM
DataHierarchy.xlsx
(ooxml, embedded labels, table is TIME);

DATA:
LOAD PROD as PROD_KEY,
     GEOG as GEOG_KEY,
     TIME as TIME_KEY,
     VALUE
FROM
DataHierarchy.xlsx
(ooxml, embedded labels, table is DATA);

//8 arguments:
/* arg1: NodeID (here the Geog key) : it will be the key of the table
   arg2: ParentID (here the parent key)
   arg3: Node description : QV will create several fields GEOG_DESC1 to GEOG_DESCn
   where n is the number of levels
   The others arguments are optional. They let you create and name the necessary fields:
   arg4: ParentName : the field will contain the description of the parent, here GEOG
   PARENT NAME
   arg5: PathSource : for the ListBox, we need a path from the total to the current
   Node, sth like
       Grandparent 1; Parent 1; Node
   The way to create this string with all the ancestors will be either the key or
   the desc
   arg6: PathName, the name of the field that will contain this string
   arg7: PathDelimiter : the letter to separate the fields of the path, here the pipe
   AltGR6 because I am sure it will never appear in the desc
   arg8: HierarchyDepth: name of the field that will contain the depth of the node, in
   other words the number of ancestors until the top +1 (the top has the depth 1)

*/
```

GEOGH:

```
HIERARCHY(GEOG_HKEY, PARENT_KEY, GEOG_DESC, [GEOG PARENT NAME], GEOG_DESC, [HIERARCHY  
GEOG], '|', 'HIERARCHY DEPTH')  
LOAD GEOG_HKEY,  
      PARENT_KEY,  
      GEOG_DESC  
FROM  
DataHierarchy.xlsx  
(ooxml, embedded labels, table is HIERARCHY);
```

GEOGH2:

```
HierarchyBelongsTo(GEOG_HKEY, PARENT_KEY, GEOG_DESC, 'ANCESTORS_KEY',  
'ANCESTORS_NAME', 'Depth')  
LOAD GEOG_HKEY,  
      PARENT_KEY,  
      GEOG_DESC  
FROM  
DataHierarchy.xlsx  
(ooxml, embedded labels, table is HIERARCHY);
```

REL_GEOGH:

```
LOAD ANCESTORS_KEY as GEOG_HKEY,  
      GEOG_HKEY as GEOG_KEY  
RESIDENT GEOGH2;
```

```
DROP table GEOGH2;
```

```
RENAME FIELD GEOG_DESC1 to TOTAL;  
RENAME FIELD GEOG_DESC2 to ZONE;  
RENAME FIELD GEOG_DESC3 to REGION;  
RENAME FIELD GEOG_DESC4 to CLIENT;
```

```
//why keep them in this application ?  
//DROP FIELDS [GEOG PARENT NAME], [HIERARCHY DEPTH], [PARENT_KEY]
```

