

LES SET ANALYSIS

Sommaire

1	Utilité des SETS	3
2	L'identificateur	4
3	Les opérateurs	4
4	Les Modificateurs	5
4.1	Toutes les positions.....	5
4.2	Des positions connues et nommées.....	6
4.3	Recherche de caractères	7
4.4	Utilisation d'une borne pour une dimension entière	8
4.5	Utilisation d'une variable.....	8
4.5.1	Utilisation d'une variable stockant une ou plusieurs positions.....	8
4.5.2	Utilisation d'une variable stockant une recherche de caractères.....	9
4.5.3	Utilisation d'une variable stockant des entiers	9
4.5.4	Utilisation d'une variable stockant une dimension.....	9
4.5.5	Utilisation d'une variable stockant l'expression du set	10
4.6	Utilisation d'une fonction.....	10
4.6.1	Fonction retournant un numérique.....	10
4.6.2	Fonction retournant des positions.....	11
4.7	Utilisation des fonctions p() et e()	13
4.8	Utilisation de plusieurs champs	13
4.9	Les sets extrêmes	14
4.9.1	Le set de la totalité de la base	14
4.9.2	Le set vide.....	14

Avant-propos

J'espère que cette documentation éclairera ceux qui cherchent à comprendre et maîtriser cet épineux sujet mais central. Merci de me laisser vos commentaires et remonter les erreurs si vous en trouvez.

Bonne lecture.

Il s'agit de la version 1.0 de ce document. QlikView 11.20 SR2 a été utilisé.

Petite convention de nom :

Ma base ne respecte pas tout à fait les principes de QlikView. J'en suis désolé mais cela ne nuira pas à la lecture du document.

XXX_KEY : clef de la dimension

XXX_LDESC : longue description de la dimension

XXX_SDESC : courte description de la dimension

La base contient pour le temps :

- Year
- TIME_KEY, TIME_SNAME et TIME_NAME qui sont des périodes allant de 2010 à la période actuelle de 2013.

1 Utilité des SETS

Ils permettent de créer une sélection indépendante ou seulement légèrement différente de la sélection active utilisée dans le graphe ou le tableau. Le groupe créé via le set est différent de la sélection active ce qui permet de suivre l'agrégation (somme, comptage ...) de ce nouveau groupe et de la comparer à la sélection courante.

Ces totalisations différentes peuvent servir à afficher les données de l'année passée (et en calculer les écarts et les évolutions par rapport à l'année passée), un YTD même si l'utilisateur ne choisit qu'une période, des chiffres de la totalité de la dimension (donc calculer une part de marché, un pourcentage de la famille, catégorie, de la région ...), d'appliquer une sélection sauvegardée (bookmark) ...

Le set va modifier le contexte des données uniquement dans l'expression utilisée. Il est donc impératif de nommer correctement l'expression utilisée.

Les autres expressions, sauf utilisation d'un set analysis, vont utiliser la sélection standard ou du groupe associé au graphe (Alternate State). L'utilisateur peut donc sélectionner une année et le graphe en afficher deux (celle choisie, celle qui précède), choisir une période et le graphe afficher le YTD ou toutes les périodes qui précèdent la période choisie.

Un set analysis global est toujours écrit entre {}. Il peut être composé d'un ou de plusieurs sets imbriqués ce qui peut compliquer grandement l'écriture ou la lecture. Lorsque ces sets imbriqués sont avec modificateurs (afin d'en modifier la sélection de départ), ils sont écrits entre crochets angulaires (<>).

On peut modifier le contexte la sélection d'une ou plusieurs dimensions selon les besoins de calcul.

Un set analysis peut être composé de 3 morceaux distincts :

- L'identificateur
- L'opérateur
- Le modificateur

Chacun de ces 3 composants sont optionnels. Vous trouverez donc des sets sans identificateur, ou sans opérateur ou sans modificateur. Evidemment, sans les 3, vous n'aurez aucun set !! Ces 3 composants vont nous servir de trame à ce document.

2 L'identificateur

0 – un set vide

1 – le set de la totalité de la base

\$ - la selection courante (attention la selection courante n'est pas forcément celle utilisée par le tableau ou graphe si vous utilisez des Alternate States)

\$1 – la selection précédente (\$2: la précédente de la précédente, etc.)

\$_1 – la selection suivante

Bookmark01 – le nom ou l'ID de la sélection sauvegardée (bookmark)

Group – le nom du groupe (Alternate State)

Sum({Book1} [Sales]) : les ventes de la sélection Book1

Sum({Group1} [Sales]) : les ventes de l' Alternate State Group1

Sum({1} [Sales]) : les sommes totales de la base (toutes les dimensions sont à All)

Sum({\$} [Sales]) : les ventes de la sélection courante (= sum([Sales])

3 Les opérateurs

+ : union des sets

*: intersection des sets

- : Exclusion des deux sets

/ : Set dont les membres n'appartiennent qu'à un seul des deux sets

Ces opérateurs ne sont pas présents dans tous les sets mais nous les verrons dans le chapitre suivant :

<Dimension += autre set> : ajouter l'autre set de sélection à la sélection actuelle

<Dimension -= autre set> : retirer l'autre set de sélection de la sélection actuelle

Sum({1-\$} [Sales]) : somme des ventes pour tout ce qui n'est pas dans la sélection (1=tout, \$ signifie ici la sélection courante par défaut

Sum({GROUP1 * Book1} [Sales]) : somme des ventes pour tout ce qui est commun entre la sélection courante de GROUP1 et le Bookmark Book1

Les sets que l'on va étudier dans la section des modificateurs sont insérés entre crochets angulaires : <>. On peut créer des sets et les assembler via les opérateurs ci-dessus.

<set 1> -<set 2> ou **{set 1}-{set 2}**: positions dans le set 1 qui ne sont pas dans le set 2

4 Les Modificateurs

Ils permettent de réduire ou d'élargir la sélection sur une ou plusieurs dimensions. C'est la partie compliquée des Sets Analysis.

Le set ne reproduit pas les clics Utilisateur dans les ListBox. Si vous décidez de fixer dans le set un produit qui n'est pas vendu chez un client, malheureusement sélection de l'utilisateur, votre set sera vide. Une telle sélection produit dans les ListBox aurait eu pour effet de resélectionner les clients vendant ce produit, mais gardez en tête que ce n'est pas le cas des sets.

Attention également aux périodes : vouloir sélectionner l'année précédente (Year-1) ou un YTD va nous obliger parfois à remettre à plat la sélection sur l'ensemble des dimensions temporelles.

4.1 Toutes les positions

```
{Sélection de départ <Dimension1 = {*} >}
```

Ou

```
{Sélection de départ <Dimension1 = [, Dimension2 = {position 1, position 2, ...}] >}
```

Pour récupérer toutes les positions de la dimension, on peut donc mettre après le signe = :

- {*} ou {"*"} s'il s'agit de texte
- rien

Comme on le voit dans les exemples ci-dessous, le {*} peut être omis.

```
Ex : Tous les fabricants mais sur la catégorie ACC  
{<FABRICANT_LDESC = {"*"}, CATEGORY_LDESC={"ACC"} >}
```

```
{<FABRICANT_LDESC =, CATEGORY_LDESC={"ACC"} >}
```



Attention, dans Qlikview nous créons souvent plusieurs ListBox agissant peu ou prou sur la même dimension. C'est le cas de l'axe temporel où l'on dispose une ListBox sur les mois, l'autre sur l'année. Et évidemment, l'utilisateur peut se servir des deux ListBox.

Regardez les différences entre les 2 sets ci-dessous :

{<YEAR = {2012}>} revient à prendre les mois actuellement sélectionnés mais pour l'année 2012, peut-être tous, peut-être pas

{<MONTH=, YEAR = {2012}>} revient à prendre tous les mois de l'année 2012

Les deux dimensions sont en quelque sorte indépendantes. Et **il convient de remettre la sélection à ALL sur tous les champs qui sont sélectionnables via ListBox dans un onglet.**

4.2 Des positions connues et nommées

Syntaxe de base :

{Sélection de départ <Dimension1 = {position 1, position 2, ...} [, Dimension2 = {position 1, position 2, ...}] >}

Ex : {<FABRICANT_LDESC = {AMBOISE, BELLE, AUTREMENT}>}

Ex : {<FABRICANT_LDESC = {AMBOISE, BELLE, AUTREMENT}, CATEGORY_LDESC={ACC}>}

Ex dans la syntaxe globale : `sum({<FABRICANT_LDESC = {AMBOISE, BELLE, AUTREMENT}>} [Ventes Valeur en Euro])`

Noter que :

- La sélection de départ peut être la sélection courante (\$) qui est le défaut, la totalité de la base (1), une sélection sauvegardée (un bookmark), un groupe (Alternate State). Ci-dessus, l'application somme la sélection courante sauf sur la dimension Fabricant qui ne contiendra que les 3 positions nommées.
- Il n'y a pas de virgule entre le set et la mesure à sommer
- La dimension doit être insérée entre crochets droits [] si elle contient un caractère spécial comme un espace, un trait d'union
- Les positions sont toujours insérées également dans des {} qu'elles soient nommées ou trouvées via un test
- Les positions sont séparées d'une virgule (,)
- Les positions numériques ne sont pas insérées entre apostrophes ou guillemets
- Les positions texte ne sont pas nécessairement insérées entre guillemets ou apostrophes : ceux-ci sont indispensables si les positions ont des caractères spéciaux tels les espaces. Néanmoins, en mettre ne nuit pas pour les caractères.

Ex : {<FABRICANT_LDESC = {"AMBOISE", "BELLE"}, CATEGORY_LDESC={"ACC", "PLATS CHAUDS"}>}

Ex : {<FABRICANT_KEY = {253, 2789, 1200}, CATEGORY_LDESC={"ACC", "PLATS CHAUDS"}>}

Si l'on omet des positions sur l'une des dimensions, le set est vide :

Ex : {<FABRICANT_LDESC = {}, CATEGORY_LDESC={ACC} >}

Apostrophes, guillemets ou crochets

QlikView accepte les trois syntaxes pour les positions texte. On peut donc écrire :

{<FABRICANT_LDESC = {"AMBOISE"}>}

{<FABRICANT_LDESC = {'AMBOISE'}>}

{<FABRICANT_LDESC = [AMBOISE] >}

Et comme il n'y a pas d'espace dans le champ, on peut l'écrire d'une quatrième manière :

{<FABRICANT_LDESC = {AMBOISE}>}

Nous verrons plus tard que l'on peut inclure une fonction d'agrégation qui va rechercher les positions en fonction d'un critère. Il est alors nécessaire d'utiliser les guillemets pour le général, et les apostrophes ou crochets pour le texte intérieur si cela est nécessaire.

4.3 Recherche de caractères

On peut aussi vouloir tous les fabricants qui commencent par certaines lettres ou qui ont certains caractères dans leur nom :

Syntaxe : {<Dimension = {"*recherche*"}>}

Si l'on souhaite faire plusieurs recherches en plus d'un texte fixe :

Syntaxe : {<Dimension = {"*recherche1*", "texte fixe", "?echerche2*"}>

Ex : {<FABRICANT_LDESC = {"ENT*", "?LIS*"}, CATEGORY_LDESC={"A?? " } >}

Et bien sûr si l'on veut toutes les positions d'une dimension : <Dimension = {"*"}> qu'on peut aussi écrire simplement <Dimension =>

Comme on le voit dans les exemples ci-dessous, le {*} peut être omis.

Ex : Tous les fabricants mais sur la catégorie ACC
{<FABRICANT_LDESC = {"*"}, CATEGORY_LDESC={"ACC"} >}

{<FABRICANT_LDESC =, CATEGORY_LDESC={"ACC"} >}

Ex : Tous les fabricants sauf ceux qui commencent par ENT
{<FABRICANT_LDESC = {"*"} - {"ENT*"}>}

{<FABRICANT_LDESC = - {"ENT*"}>}

Ex : Tous les fabricants de la sélection + ceux qui commencent par ENT
{<FABRICANT_LDESC += {"ENT*"}>}

Ex : Tous les fabricants de la sélection sauf ceux qui commencent par ENT
{<FABRICANT_LDESC -= {"ENT*"}>}



Attention : {<FABRICANT_LDESC -= {"ENT*"}>} et {<FABRICANT_LDESC = -{"ENT*"}>}

sont très différents. Le premier retire de la sélection actuelle tous les fabricants commençant par ENT, tandis que le second prend tous les fabricants sauf ceux commençant par ENT : le {"*"} est par défaut.

Ex : Tous les fabricants, toutes catégories quelles que soient les sélections faites sur ces deux dimensions

{<FABRICANT_LDESC = {"*"}, CATEGORY_LDESC={"*"} >}

Ex : Toutes les années (c'est du numérique, pas de guillemets), toutes les catégories
{<Year = {*}, CATEGORY_LDESC={"*"} >}

Part des catégories sélectionnées pour chaque fabricant, sachant que les fabricants dimensionnent le tableau (mais pas les catégories) :

```
%Categorie : sum([Ventes Volume])/sum({$ <CATEGORY_LDESC={"*"}>} [Ventes Volume])
%Fabricant : sum([Ventes Volume])/sum(TOTAL [Ventes Volume])
```

FABRICANT_LName	Somme de la sé...	Total catégories	% Catégorie	% Fabricant
AUTRES FABRICANTS	927 121	1 977 973	46,87%	53,46%
BO	10 920	71 546	15,26%	0,63%
EN	381 271	1 406 685	27,10%	21,99%
ER	409 735	426 648	96,04%	23,63%
EUR	141	141	100,00%	0,01%
FR	663	663	100,00%	0,04%
FR	451	2 758	16,35%	0,03%
GI	1 360	1 360	100,00%	0,08%
ISI	2 447	2 447	100,00%	0,14%
Totals	1 734 109	3 890 221	44,58%	100,00%

(chiffres fictifs, acteurs réels)

Rappel : QlikView accepte le guillemet, l'apostrophe ou le crochet droit. C'est le cas aussi pour la recherche de caractère. On peut donc écrire :

```
Ex :
{<FABRICANT_LDESC = {"E*"}, CATEGORY_LDESC={"A??"} >}
{<FABRICANT_LDESC = \'E*\', CATEGORY_LDESC=\'A??\' >}
{<FABRICANT_LDESC = [E*], CATEGORY_LDESC=[A??] >}
```

4.4 Utilisation d'une borne pour une dimension entière

Il arrive fréquemment que les codes des années soient des entiers (2000, 2001, ...), que certains codes de dimension le soit aussi.

Il faut avouer que la syntaxe n'est pas évidente puisqu'il faut insérer le test booléen entre guillemets :

```
{<TIME_KEY = {"<12"}>}: toutes les périodes dont TIME_KEY est strictement inférieure à 12
{<TIME_KEY = {">=12 <=18"}>}: entre 12 et 18, les 2 bornes comprises
```

Ex: Les périodes 4,5, 6 et celles entre 12 comprise et 20 exclue

```
{<TIME_KEY = {4, 5, 6} + {">=12 <20"}>}
```

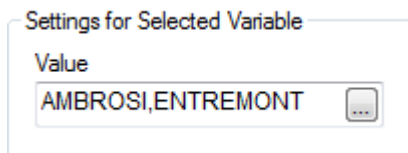
4.5 Utilisation d'une variable

4.5.1 Utilisation d'une variable stockant une ou plusieurs positions

```
Syntaxe : <Dimension = {$(NomVariable)} >
Ou       <Dimension = {$(=NomVariable)} >
```

(avec ou sans le signe = devant la variable)

A noter que $\$(=NomVariable)$ va interpréter la variable et « réécrire » la syntaxe avant exécution. Le contenu de cette variable peut donc contenir plusieurs positions de la dimension qui doivent être séparées comme on l'écrirait dans la syntaxe, c'est-à-dire d'une simple virgule. Les guillemets sont absents de cette syntaxe même si on peut l'écrire avec dans certains cas.



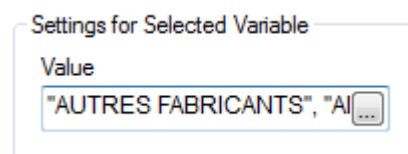
Ex :

```
<FABRICANT_LDESC = {$(vChoixFabricant)}, CATEGORY_LDESC={$(=ChoixCategorie)}>
```

Attention : on trouve aussi le \$ dans les set analysis signifiant sélection courante par défaut.

\$(NomVariable) : peut être inséré ou non entre guillemets si la variable elle-même n'en contient pas.

Aucun guillemet dans la syntaxe comme ci-dessus doit être préféré car cela permet d'avoir une variable qui contient des champs avec des espaces comme « AUTRES FABRICANTS » et qu'il faut insérer, eux, entre guillemets :



4.5.2 Utilisation d'une variable stockant une recherche de caractères

Exemple de recherche de caractères : `<FABRICANT_LDESC = {"E*"}, CATEGORY_LDESC={"A??"} >`

Noter que si la variable vChoixProduit contient une recherche de caractères comme "E*" (les guillemets sont indispensables et font partie de la valeur de la variable), alors on peut écrire la syntaxe ci-dessus de cette façon :

```
<FABRICANT_LDESC = {$(vChoixProduit)}, CATEGORY_LDESC={["A??"]} >
```

4.5.3 Utilisation d'une variable stockant des entiers

Avec les entiers, il est nécessaire de mettre les guillemets :

```
<TIME_KEY = {"<$(vPremierePeriode)"}>
```

```
Ou <TIME_KEY = {"<$(=vPremierePeriode)"}>
```

Entre deux périodes stockées dans des variables :

```
<TIME_KEY = {">$(vPremierePeriode) <$(=vDernierePeriode)"} >
```

4.5.4 Utilisation d'une variable stockant une dimension

Nous avons vu que QlikView permettait l'utilisation d'une variable dans le set via son interprétation avec le signe dollar (\$).

On peut également créer une variable qui stockera une dimension et l'utiliser dans le set :

```
Ex : sum( {$ <$(vDim)={"*"} >} [Ventes Volume])
```

On souhaite avoir ci-dessus l'ensemble des positions de la dimension contenue dans la variable vDim.

4.5.5 Utilisation d'une variable stockant l'expression du set

On peut aller encore plus loin en stockant la totalité de la commande du set, ou seulement une partie si on souhaite amender celle-ci afin de l'inclure dans plus de commandes.

Ex : `sum({$ <$(vSet) >} [Ventes Volume])`

On souhaite avoir ci-dessus la somme des Ventes de ce qui est contenu dans la variable vSet.

Ex : `sum({$ <$(vSet), CATEGORY_LDESC= "ACC" >} [Ventes Volume])`

On souhaite avoir ci-dessus la somme des Ventes de ce qui est contenu dans la variable vSet mais uniquement pour la catégorie ACC

La variable **vSet** doit contenir un set valide mis à part les <> :

- `FABRICANT_LDESC = {"*"}, TIME_SDESC={"P 01/13"}`
- `Year={$(=max({1} Year))}, TIME_LDESC=`
- ...

4.6 Utilisation d'une fonction

Avec une telle fonction on peut trouver la dernière ou l'avant-dernière période, les catégories, fabricants qui réalisent au moins tel CA, sur telle période, telle famille de produits ou sur l'ensemble de la base. On peut également imbriquer plusieurs fonctions entre elles afin de former un groupe des 5 meilleurs produits ou fabricants la dernière année, et de suivre ce Top 5 sur un historique choisi.

4.6.1 Fonction retournant un numérique

Syntaxe : `<Dimension = {" $(=f(args)) "}`



le \$ de la syntaxe est ici le signe d'expansion, c'est-à-dire l'interprétation de la fonction.

Trouver l'avant dernière période peut se réaliser avec la fonction `max([set] Champ)`

`<Year = {"$(=max({1} Year)-1)">`

A noter que la fonction `max()` tient compte de la sélection de l'utilisateur. Elle renvoie donc la dernière période choisie par l'utilisateur. D'où le **{1}** qui signifie la totalité de la base (voir section sur les opérateurs) Si l'on recherchait la dernière année sélectionnée par l'utilisateur, on pourrait écrire :

- `<Year = {"$(=max({$} Year)-1)">`
- `<Year = {"$(=max(Year)-1)">` puisque le second \$ signifie « la sélection courante » qui est le défaut

Lorsqu'on cherche à sélectionner les périodes précédentes (A-1, M-1), les choses se compliquent dangereusement. Il est souvent préférable de modifier la structure de la base par exemple pour stocker des informations de type A-1, YTD ... Ci-dessous un exemple où l'on stocke l'année précédente (champ A-1), la période précédente (champ P-1), si oui ou non la période doit être calculée dans le YTD (YTD) :

TIME_KEY	MONTH	YEAR	A-1	P-1	YTD	
13	Janvier	2013	1	12	1	
14	Février	2013	2	13	1	
15	Mars	2013	3	14	1	
16	Avril	2013	4	15	1	
17	Mai	2013	5	16	1	
18	Juin	2013	6	17	1	
19	Juillet	2013	7	18	0	
20	Août	2013	8	19	0	
21	Septembre	2013	9	20	0	
22	Octobre	2013	10	21	0	
23	Novembre	2013	11	22	0	
24	Décembre	2013	12	23	0	
25	Janvier	2014	13	24	1	
26	Février	2014	14	25	1	
27	Mars	2014	15	26	1	

Trouver les ventes de A-1, un champ [A-1] contient pour chaque TIME_KEY la position de A-1 en tant que clef étrangère (un numérique donc) :

```
sum({<MONTH=, YEAR=, TIME_KEY = {$(=concat([A-1], ',')) } >} VALEUR)
```

YTD :

```
sum({<MONTH=, YTD = {1}, YEAR = >} VALEUR)
```

YTD de A-1 :

```
sum({<MONTH=, YTD = {1}, YEAR = {$(=YEAR-1)} >} VALEUR)
```

MOVING TOTAL 12 périodes :

```
sum({<MONTH=, YEAR=, TIME_KEY={ ">$(=Max(TIME_KEY)-12) <=$(=Max(TIME_KEY)) " } >} VALEUR)
```

MOVING TOTAL 12 périodes A-1:

```
sum({<MONTH=, YEAR=, TIME_KEY={ ">$(=Max(TIME_KEY)-24) <=$(=Max(TIME_KEY)-12)" } >} VALEUR)
```

Comme les sets Analysis peuvent vite devenir compliqués, on peut gérer les YTD, Moving total, les YearAgo en modifiant le modèle (voir le document que j'ai écrit à ce sujet :

<http://community.qlikview.com/docs/DOC-4821>)

A noter également que certains créent des flags (un peu comme ci-dessus avec des 1 et 0) pour ensuite :
Sum(VALEUR*Flag)

Cette façon a l'avantage d'être très simple. Mais elle est plus coûteuse que sum({<Flag = {1}> VALEUR) car QlikView dans le second cas réduit **avant** le calcul les lignes rentrant dans le périmètre. Avec sum(VALEUR*Flag), QlikView ne réduit pas le périmètre de calcul. Voir l'excellente réponse de John Witherspoon lors de la discussion : <http://community.qlikview.com/message/1434>

4.6.2 Fonction retournant des positions

La fonction texte doit être insérée entre guillemets et précédé du signe égal (=).

Syntaxe : <Dimension = { "=f(arguments de f) " } >

Sommer le volume des fabricants qui réalisent plus de 100K€ de ventes sur la sélection courante (ici le \$ est nommé)

```
sum( {$ <FABRICANT_LDESC={"=sum([Ventes Valeur en Euro])>100000"} >} [Ventes Volume])
```

Sommer le volume des fabricants qui réalisent plus de 100K de Ventes Volume sur les catégories ACC et GATEAUX FROMAGERS sur la première période 2013 : il faut donc insérer un set dans la fonction sum() qui calcule le set principal.

```
sum( {$ <FABRICANT_LDESC={"=sum({1<TIME_SDESC={'P 01/13'},  
CATEGORY_LDESC={'ACC', 'GATEAUX FROMAGERS'}>} [Ventes Volume])>100000"} >} [Ventes  
Volume])
```

Comme nous l'avons vu plus haut, QlikView accepte les guillemets ou les apostrophes. On en voit ici l'intérêt (les positions pour le set intérieur sont entre apostrophes ou crochets droits) pour ne pas fermer précipitamment la fonction.

Sommer les fabricants qui réalisent plus de 100K de Ventes Volume sur les catégories ACC et GATEAUX FROMAGERS sur la première période 2013. Retirer de ce total les fabricants qui réalisaient moins de 50K sur toutes les catégories sur la première période 2012.

Il va falloir donc créer deux sets que l'on va soustraire : {<set 1> - <set 2>}, chacun des sets utilisant une fonction sum() incluant elle-même un modificateur :

```
Le set 1 = 1 <FABRICANT_LDESC={"=sum({1< TIME_SDESC={'P  
01/13'},CATEGORY_LDESC={'ACC', 'GATEAUX FROMAGERS'}>} [Ventes Volume])>50000"} >
```

```
Le set 2 = <FABRICANT_LDESC={"=sum({1< TIME_SDESC={'P 01/12'},CATEGORY_LDESC={'*'}>}  
[Ventes Volume])>500000"} >}
```

Pour une syntaxe un peu longue :

```
sum( {1 <FABRICANT_LDESC={"=sum({1< TIME_SDESC={'P  
01/13'},CATEGORY_LDESC={'ACC', 'GATEAUX FROMAGERS'}>} [Ventes Volume])>50000"} >  
- <FABRICANT_LDESC={"=sum({1< TIME_SDESC={'P 01/12'},CATEGORY_LDESC={'*'}>} [Ventes  
Volume])<50000"} >}  
[Ventes Volume])
```

qu'on pourrait légèrement réduire :

```
sum( {1 <FABRICANT_LDESC={"=sum({1< TIME_SDESC={'P  
01/13'},CATEGORY_LDESC={'ACC', 'GATEAUX FROMAGERS'}>} [Ventes Volume])>50000"} -  
{"=sum({1< TIME_SDESC={'P 01/12'},CATEGORY_LDESC={'*'}>} [Ventes Volume])<50000"} >}  
[Ventes Volume])
```

Attention à bien écrire les champs au sein de sum() : comme il s'agit de texte, vous n'avez aucune aide de QlikView qui est très sensible à la casse. Ecrire TIME_sDESC au lieu de TIME_SDESC vous garantit d'avoir 0 ou un résultat faux car votre qualification sur TIME_SDESC ne sera pas prise (retour à la sélection). Il est de bon usage de tester le sum() intérieur indépendamment dans un tableau afin d'en valider les résultats.

4.7 Utilisation des fonctions p() et e()

Ces fonctions permettent de retourner les valeurs d'une dimension en fonction d'un test tel les clients qui ont eu des ventes telle année (et suivre ainsi ce groupe de clients sur un historique donné), les nouveaux produits ou ceux abandonnés cette année Contrairement à la fonction sum() ci-dessus, QlikView opère juste par associativité : il y a des données ou il n'y en a pas.

Syntaxe : {<Dimension1 = P(< {< {Set Dimension à fixer} Dimension1 }>) [, Autres dimensions du Set]>}

Remarques :

- Le set de la dimension à fixer va suivre les règles vues plus haut ainsi que ceux concernant les autres dimensions
- Les fonctions **p()** et **e()** ont la même syntaxe : l'une renvoie les possibles, l'autre les valeurs exclues
- Ces fonctions vont être insérées au sein d'un set plus grand (voir ci-dessous)
- Ces fonctions ne permettent pas de rechercher les N meilleurs fabricants (ou ceux qui sont en augmentation de CA) afin de constituer un groupe (ou set) que l'on va suivre sur l'historique : il n'y a aucune mesure dans la syntaxe, seule l'associativité est utilisée (il y a des données ou il n'y en a pas)

Rappel : le nom de la dimension doit être inséré entre [] s'il contient des espaces.

Ex : Recherche des fabricants qui vendent sur le marché ACC lors de P01/13. Et afficher les ventes de ces fabricants pour toutes les catégories.

```
sum({$ <CATEGORY_LDESC={"*"}, FABRICANT_LDESC=p({1<CATEGORY_LDESC={"ACC"},  
TIME_SDESC= "P 01/13">} FABRICANT_LDESC) >} [Ventes Valeur en Euro])
```

Ex : Recherche des nouveaux fabricants entrés sur le marché ACC (présents sur la dernière période et non sur toute l'année précédente)

4.8 Utilisation de plusieurs champs

On peut vouloir garder toutes les positions d'une dimension lorsque celles-ci égalent un autre champ :

- Client facturé = client livré
- Date Commande = Date Livraison
- Marque = Compagnie (pour les compagnies éponymes)
- ...

La syntaxe se rapproche de celle vue en 4.6.2.

Syntaxe : {<Dimension = {"=condition bool"}>}



Attention : la dimension recherchée ne peut être de nouveau dans la condition. Au besoin, il est nécessaire via **Autonumber()** de numéroter les lignes afin de créer un champ de clef.

On veut par exemple calculer les ventes qui ont été livrées le jour même. Nous disposons de deux champs JourCommande et JourLivraison.

```
Sum(<{<KeyAutoNumber = {" =(JourCommande=JourLivraison)" } >} Ventes)
```

On en déduit facilement la recherche des ventes qui n'ont pas été effectuées dans les 7 jours (le délai acceptable peut être saisi et stocké dans une variable) :

```
Sum(<{<KeyAutoNumber = {" =(JourCommande<JourLivraison-7)" } >} Ventes)
```

```
Sum(<{<KeyAutoNumber = {" =(JourCommande<JourLivraison-$(vDelai))" } >} Ventes)
```

Il est néanmoins plus facile d'écrire un test dans la fonction **sum** même si cette syntaxe est plus longue à exécuter. En effet, QlikView va balayer la totalité de la sélection pour le calcul et non une petite partie liée au set :

```
Sum(if(JourCommande=JourLivraison, Ventes))
```

La fonction if() retourne Null() si le 3^e paramètre a été oublié (comme ci-dessus) et que la condition est fautive. Cette syntaxe ne nécessite pas un champ clef supplémentaire aux deux champs utilisés.

4.9 Les sets extrêmes

4.9.1 Le set de la totalité de la base

Il suffit d'utiliser le 1 inséré dans les {} : {1}

```
sum({1} [Ventes Valeur en Euro])
```

Dernière année (nous l'avons déjà vu) :

```
$(=Max({1} YEAR))
```

4.9.2 Le set vide

Il arrive que le résultat de notre magnifique set soit 0. Cela peut venir d'une volonté mais plus probablement d'une erreur. Reste à vérifier la syntaxe et regardé si on a :

- utilisé le {}, **sans rien**, comme déjà évoqué dans ce document
- mal nommé les composantes du set (attention, QlikView est sensible à la casse : Year est donc différent de YEAR)
- écrit un ensemble invalide (qui ne fait pas partie des sélections possibles) comme écrire un couple Produit-Client alors que ce dernier n'achète pas le produit en question
- utilisé dans un set imbriqué une fonction test (comme sum > x) qui ne renvoie aucune position

- oublié des champs qui rendent l'ensemble invalide avec la sélection actuelle : d'une façon générale, si sur une même dimension comme le temps, on dispose plusieurs ListBox, l'utilisateur a peut-être sélectionné ces différents champs rendant l'intersection avec le set analysis ... vide => il faut donc remettre à ALL la sélection d'un ou plusieurs champs.

Ex : on veut sommer la totalité de l'année 2012 et l'utilisateur a toujours le choix pour sélectionner les périodes dans un champ TIME_DESC

```
sum({$ <Year = {2012}, CATEGORY_LDESC= "ACC" >} [Ventes Volume])
```

La somme ci-dessus sera l'intersection de :

- Year 2012
- Les sélections sur TIME_DESC (donc en aucun cas la totalité de 2012)

Si l'utilisateur a choisi la période 'P 01/11' de notre base (TIME_SDESC contient ici les périodes de 'P01/10 à 'P07/13), le champ des possibles entre cette sélection et Year 2012 est vide. La somme sera donc égale à 0. Et lorsque l'utilisateur choisit 'P 01/12' la somme est celle de cette période : en effet, l'intersection entre Year 2012 et 'P 01/12' est ... 'P 01/12' !!

Il faut donc réécrire la commande pour sélectionner tous les TIME_SDESC:

```
sum({$ <Year = {2012}, TIME_SDESC = , CATEGORY_LDESC= "ACC" >} [Ventes Volume])
```

Donc pour le temps, on a tendance à nommer toutes les dimensions TEMPS dans le set afin d'éviter ce genre de désagrément. Dans une base qui aurait comme dimensions liées au temps YEAR, QUARTER, MONTH et DAY, on écrirait pour avoir la somme de l'année 2012 :

```
sum({$ <Year = {2012}, QUARTER =, MONTH=, DAY= >} [Ventes Volume])
```

Ainsi, quelle que soit la sélection de l'utilisateur, on aurait la totalité de l'année 2012.