

Semestre
II, 2021

SUPPORT DE COURS

MDX

SSAS 2014

Ce document présente le support de cours pour les requêtes MDX. Ce support est utilisé dans le cadre du module OLAP/MDX enseigné en première année du Mastère Business Intelligence enseigné à ISET Radès.



I. Introduction

MDX (Multi Dimensional eXpression) est un langage de requêtes pour les bases de données multidimensionnelles, de la même manière que SQL est utilisé pour les requêtes sur les bases de données relationnelles. Dans son approche, MDX est proche du SQL sur son aspect select et where même si la similarité ne va pas plus loin. Le but des expressions multidimensionnelles MDX est de rendre aisé et intuitif l'accès aux données de différentes dimensions.

MDX a été inventé par Mosha Pasumansky au sein de Microsoft. Il fut présenté pour la première fois en 1997 comme un volet de la spécification OLE DB for OLAP (ODBO). La spécification fut rapidement suivie en 1998 par une version commerciale de Microsoft OLAP Services 7.0 puis par Microsoft Analysis Services. La dernière version de la spécification OLE DB for OLAP (ODBO) a été émise par Microsoft en 1992.

a. Références

Ce cours se base sur les cubes du projet AdventureWorksDW2014Multidimensional-EE¹ issus de l'entrepôt AdventureWorksDW2014² disponibles en téléchargement sur ce lien³. Nous avons testé les requêtes sur le cube Adventure Works et avons utilisé SQL Server Developer Edition 2014. Pour SQL Server 2017 et plus, d'autres entrepôts test sont disponibles sur ce lien⁴.

Une partie de ce cours est inspirée du cours « L'informatique des entrepôts de données » disponible à cette adresse⁵ ainsi que de la documentation officielle Microsoft. Une autre partie est inspirée du document « Reading OLAP Cube using MDX: Part 1 & Part 2 » disponible à l'adresse suivante⁶. Une autre partie est inspirée des exemples du document « MDX Guide for SQL Folks: Part I⁷, II⁸ & III⁹ ». Enfin, une partie est inspirée des exemples du livre MDX with Microsoft SQL Server 2016 Analysis Services Cookbook¹⁰.

b. De l'entrepôt de données au cube

Le cube est construit à partir de AdventureWorksDW2014¹¹ dont le schéma relationnel est le suivant.

¹ <https://gist.github.com/yancyn/0721da513bb2c3975a3352af4ef3537c>

² Disponible sur <https://docs.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver15>

³ <https://github.com/Microsoft/sql-server-samples/releases/tag/adventureworks>

⁴ <https://github.com/Microsoft/sql-server-samples/releases/tag/wide-world-importers-v1.0>

⁵ <http://benhur.telug.quebec.ca/SPIP/inf9004/>

⁶ <https://www.codeproject.com/Articles/801632/Reading-OLAP-Cube-using-MDX-Part>

⁷ <https://www.sqlservercentral.com/articles/mdx-guide-for-sql-folks-part-i-navigating-the-cube>

⁸ <https://www.sqlservercentral.com/articles/mdx-guide-for-sql-folks-part-ii-hierarchies-and-functions>

⁹ <https://www.sqlservercentral.com/articles/mdx-guide-for-sql-folks-part-iii-advanced-analysis-2>

¹⁰ Le code des exemples est disponible sur <https://github.com/PacktPublishing/MDX-with-Microsoft-SQL-Server-2016-Analysis-Services-Cookbook>

¹¹ Disponible sur <https://docs.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver15>

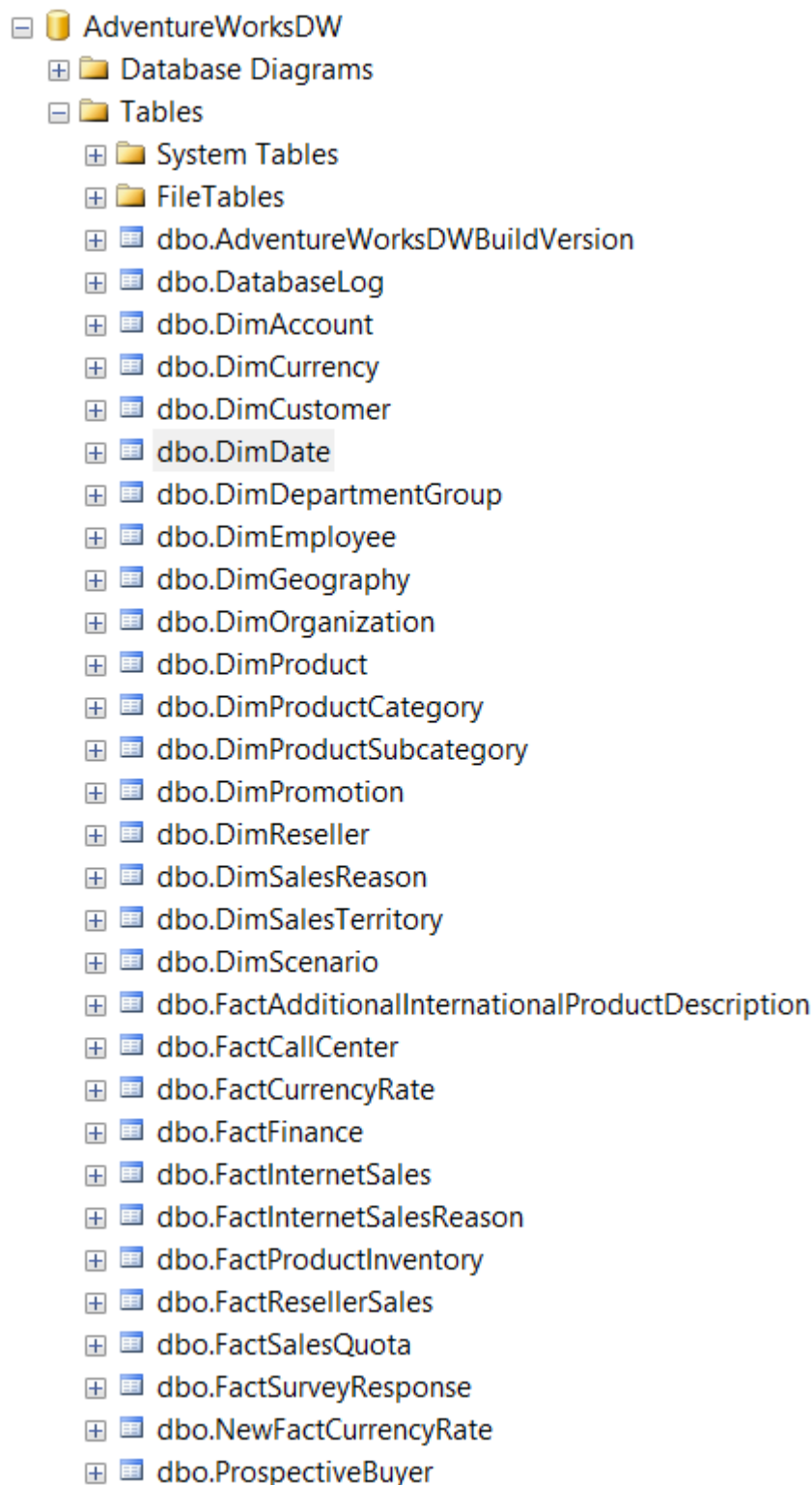


Figure 1. L'entrepôt de données AdventureWorksDW

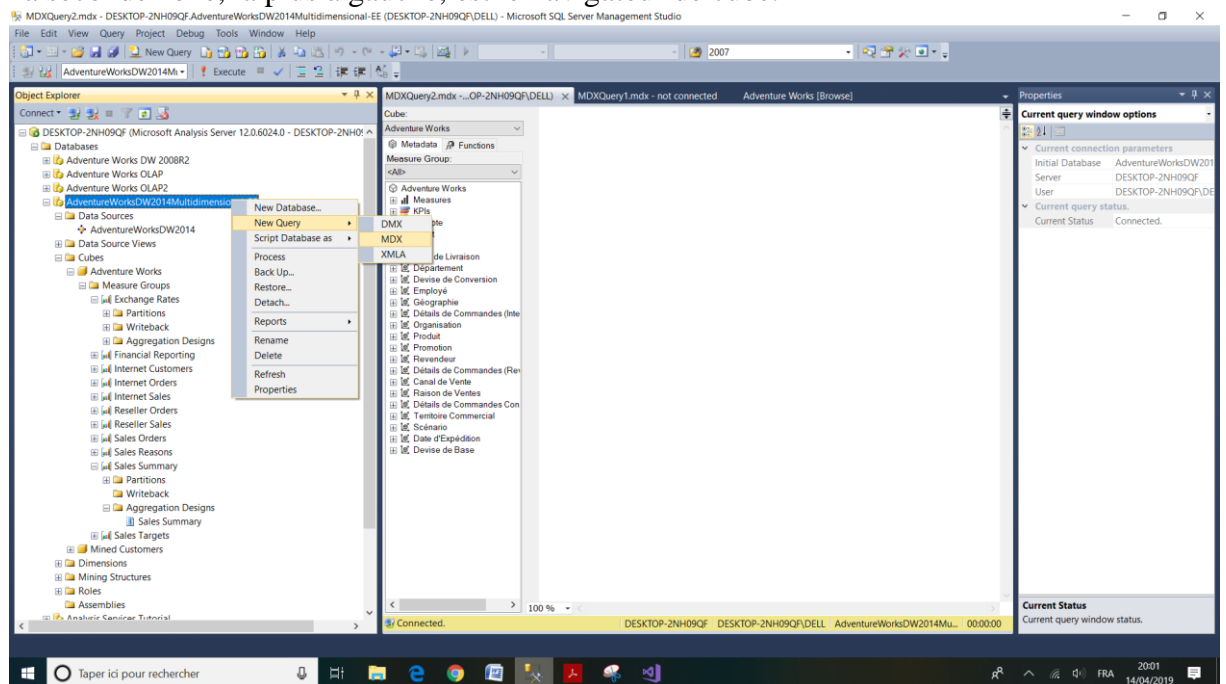
Pour interroger l'entrepôt, on exécute les requêtes SQL classiques.

c. Cube à manipuler

Dans la suite, nous étudions les requêtes MDX les plus communément utilisées. Pour écrire et exécuter les requêtes MDX, allez à SSMS, cliquez sur le bouton Requête MDX Analysis Services :

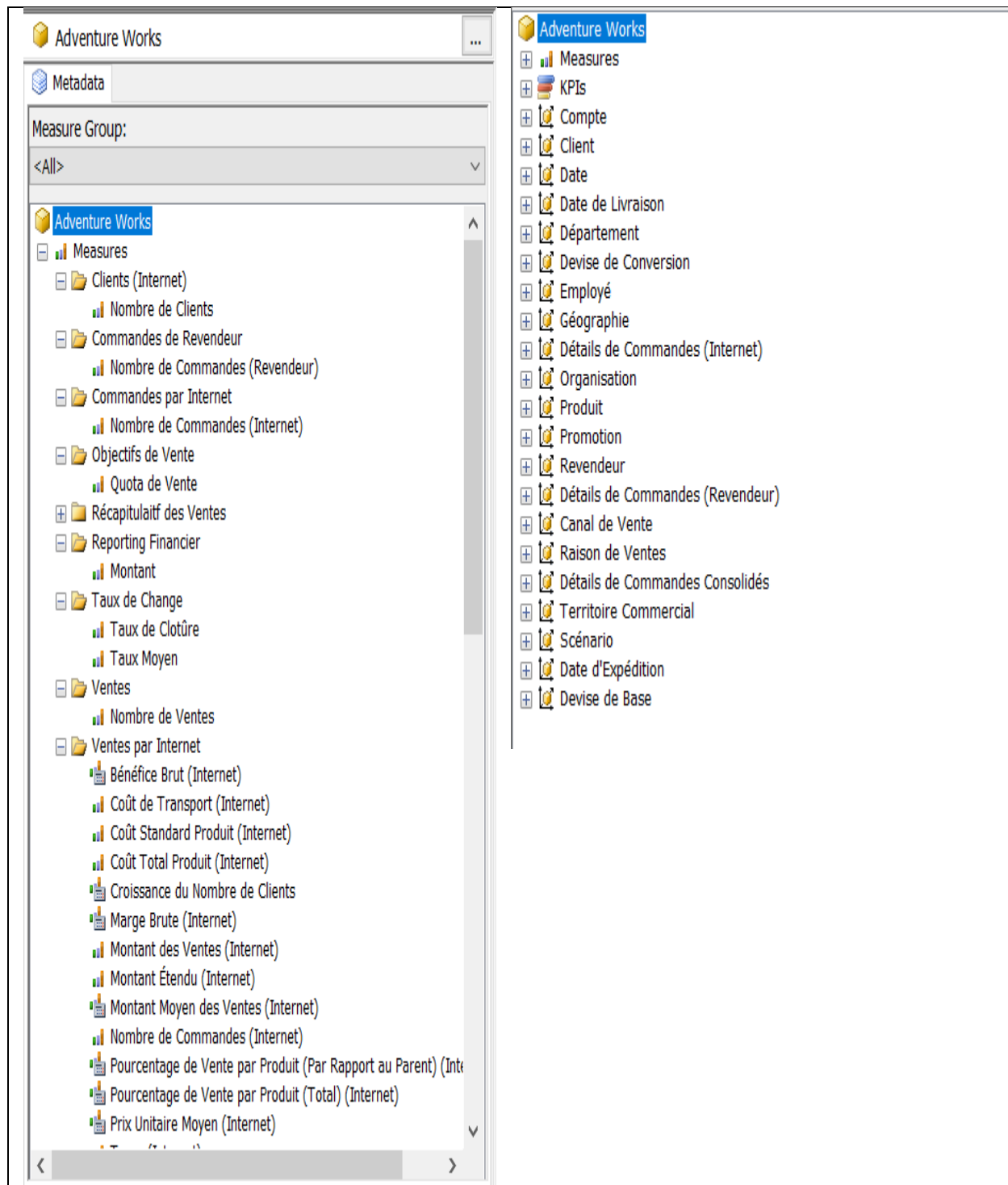
- Le requêteur MDX de SSMS est composé de deux zones :

- La première zone, la plus à droite, est la zone de travail.
- La seconde zone, la plus à gauche, est le navigateur de cube.



Les éléments de mesures, de dimension, d'attributs, de hiérarchie et même de membres se glissent par glisser/déposer directement dans la zone de travail. Faites quelques essais pour comprendre comment s'utilise l'interface.

Les données à manipuler sont celles du cube Adventure Works. Ci-dessous, les mesures, les dimensions et leurs hiérarchies.



c. De l'entrepôt relationnel au cube

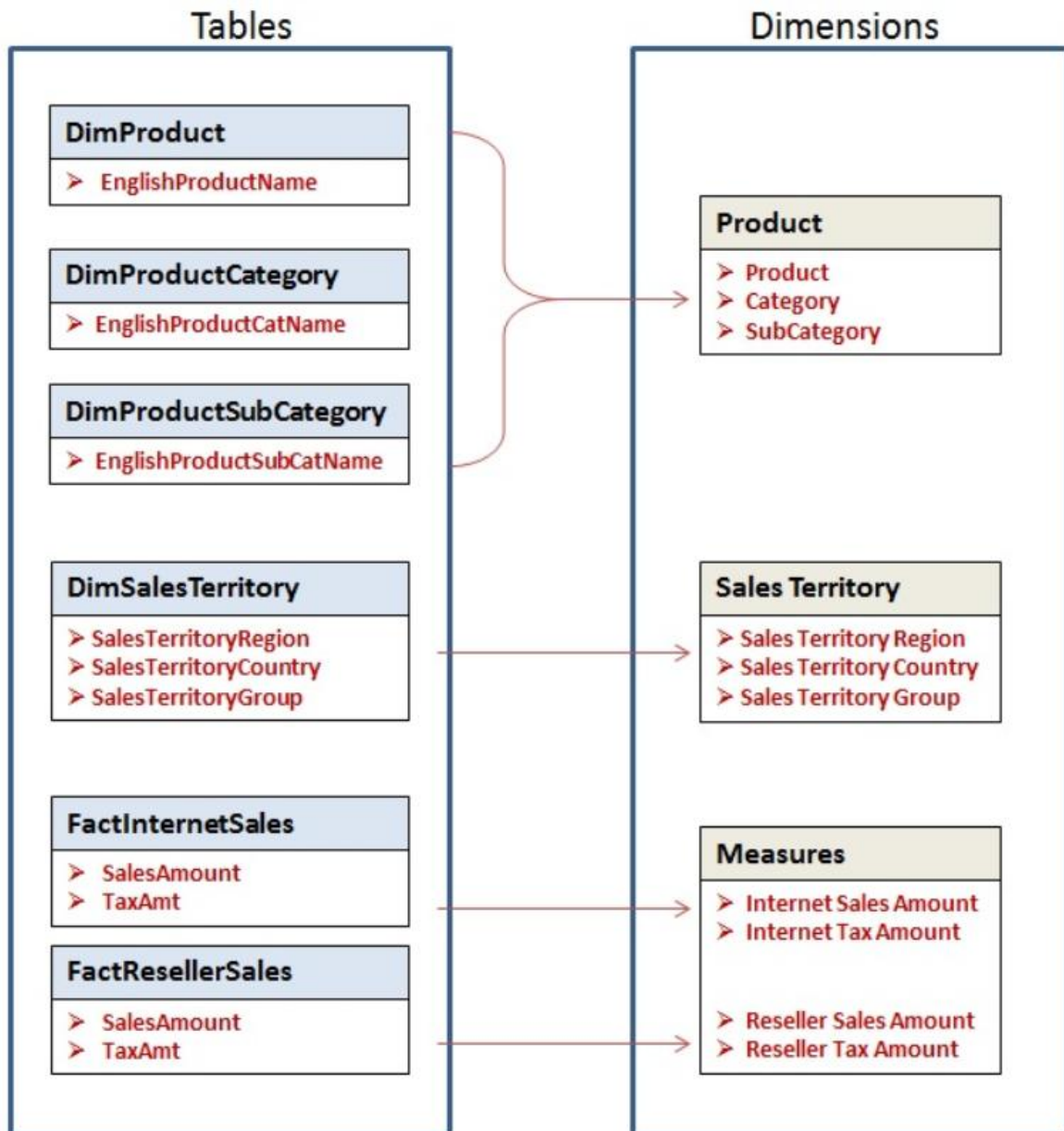


Figure 2. Des tables de l'entrepôt aux dimensions du cube¹²

III. MDX, syntaxe de base

Dans son approche MDX est proche du SQL sur son aspect select et where, même si la similarité ne va pas plus loin. Un prototype d'une requête MDX est donné par la syntaxe suivante.

```
select [<axis_specification> [, <spécification_des_axes>...]]
from [<spécification_d_un_cube>]
[where [<spécification_de_filtres>]]
```

Dans MDX, l'instruction SELECT permet de spécifier un jeu de résultats doté d'un sous-ensemble de données multidimensionnelles qui ont été retournées à partir d'un cube. Pour spécifier un jeu de résultats, une requête MDX doit renfermer les informations suivantes :

- Nombre d'axes que le jeu de résultats doit contenir. Vous pouvez spécifier jusqu'à 128 axes dans une requête MDX.

¹² <https://www.sqlservercentral.com/articles/mdx-guide-for-sql-folks-part-i-navigating-the-cube>

- Jeu de membres ou de tuples à inclure sur chaque axe de la requête MDX.
- Le nom du cube qui définit le contexte de la requête MDX.
- Jeu de membres ou de tuples à inclure sur l'axe de secteur.

Pour identifier les axes de requête, le cube qui fera l'objet de la requête et l'axe de secteur, l'instruction SELECT MDX utilise les clauses suivantes :

- Une clause SELECT qui détermine les axes de requête d'une instruction SELECT MDX.
- Clause FROM qui détermine quel cube fera l'objet d'une requête.
- Clause WHERE facultative qui détermine les membres ou tuples à utiliser sur l'axe de secteur pour restreindre les données retournées.

Un cube contient une mesure par défaut. Si aucune mesure n'est spécifiée, c'est cette mesure qui sera utilisée.

On peut le vérifier par les requêtes suivantes

<code>Select</code> <code>from</code> [Adventure Works]	\$80,450,596.98
--	-----------------

L'exécution de la requête pour afficher la mesure Reseller Sales Amount

<code>Select</code> [Measures].[Reseller Sales Amount] <code>on</code> 0 <code>from</code> [Adventure Works]	Montant des Ventes (Revendeur) \$80,450,596.98
---	---

Par la suite, si aucune mesure n'est spécifiée, SSAS va utiliser cette mesure pour afficher les résultats.

a. Spécification d'axes

La spécification d'un axe doit être un set suivi du mot clef `on` qui est suivi à son tour d'un nom d'axe.

La notion d'axe peut faire référence à un numéro d'ordre s'il y a plus de deux axes de restitution, ou tout simplement aux noms d'axes explicites "columns" tout d'abord et "rows".

Sur un tableur, on peut afficher les résultats par Ligne et par Colonne comme suit.

	Internet Sales Amount
Australia	\$9,061,000.58
Canada	\$1,977,844.86
France	\$2,644,017.71
Germany	\$2,894,312.34
NA	(null)
United Kingdom	\$3,391,712.21
United States	\$9,389,789.51

Pour obtenir le même résultat avec MDX, il faut préciser la mesure à afficher sur la colonne Internet Sales Amount et les valeurs à afficher ligne par ligne avec le mot clé `On Rows`.

On obtient :

<code>select</code> [Measures].[Internet Sales Amount] <code>on</code> columns, [Sales Territory].[Sales Territory Country].[Sales Territory Country] <code>on</code> rows <code>from</code> [Adventure Works]
--

	Montant des Ventes (Internet)
Australia	\$9,061,000.58
Canada	\$1,977,844.86
France	\$2,644,017.71
Germany	\$2,894,312.34
NA	(null)
United Kingdom	\$3,391,712.21
United States	\$9,389,789.51

On obtient le même résultat avec Axis(0) et Axis (1)

<pre>SELECT [Measures].[Internet Sales Amount] ON axis(0) , [Product].[Product] ON Axis(1) FROM [Adventure Works]</pre>	<table> <tr> <th></th><th>Montant des Ventes (Internet)</th></tr> <tr> <td>Tous les Produits</td><td>\$29,358,677.22</td></tr> </table>		Montant des Ventes (Internet)	Tous les Produits	\$29,358,677.22												
	Montant des Ventes (Internet)																
Tous les Produits	\$29,358,677.22																
<pre>select [Measures].[Internet Sales Amount] ON axis(0), [Sales Territory].[Sales Territory Country].[Sales Territory Country] ON Axis(1) from [Adventure Works]</pre>	<table> <tr> <th></th><th>Montant des Ventes (Internet)</th></tr> <tr> <td>Australia</td><td>\$9,061,000.58</td></tr> <tr> <td>Canada</td><td>\$1,977,844.86</td></tr> <tr> <td>France</td><td>\$2,644,017.71</td></tr> <tr> <td>Germany</td><td>\$2,894,312.34</td></tr> <tr> <td>NA</td><td>(null)</td></tr> <tr> <td>United Kingdom</td><td>\$3,391,712.21</td></tr> <tr> <td>United States</td><td>\$9,389,789.51</td></tr> </table>		Montant des Ventes (Internet)	Australia	\$9,061,000.58	Canada	\$1,977,844.86	France	\$2,644,017.71	Germany	\$2,894,312.34	NA	(null)	United Kingdom	\$3,391,712.21	United States	\$9,389,789.51
	Montant des Ventes (Internet)																
Australia	\$9,061,000.58																
Canada	\$1,977,844.86																
France	\$2,644,017.71																
Germany	\$2,894,312.34																
NA	(null)																
United Kingdom	\$3,391,712.21																
United States	\$9,389,789.51																

La requête suivante donne le coût total de produits par internet pour tous les produits, années, etc.

<pre>SELECT [Measures].[Internet Total Product Cost] on columns FROM [Adventure Works]</pre>	<table> <tr> <th>Internet Total Pr...</th><th></th></tr> <tr> <td>17277793,5757</td><td></td></tr> </table>	Internet Total Pr...		17277793,5757							
Internet Total Pr...											
17277793,5757											
<pre>Select { [measures].[internet sales amount] } on columns, { [product].[product categories].[category] } on rows from [adventure works]</pre>	<table> <tr> <th></th><th>Montant des Ventes (Inter</th></tr> <tr> <td>Accessoire</td><td>\$700,759.96</td></tr> <tr> <td>Vélo</td><td>\$28,318,144.65</td></tr> <tr> <td>Vêtements</td><td>\$339,772.61</td></tr> <tr> <td>Composant</td><td>(null)</td></tr> </table>		Montant des Ventes (Inter	Accessoire	\$700,759.96	Vélo	\$28,318,144.65	Vêtements	\$339,772.61	Composant	(null)
	Montant des Ventes (Inter										
Accessoire	\$700,759.96										
Vélo	\$28,318,144.65										
Vêtements	\$339,772.61										
Composant	(null)										

On peut remplacer les mots clés Columns et Rows par 0 et 1 respectivement.

<pre>Select { [measures].[internet sales amount] } on 0, {[product].[product categories].[category]} on 1 from [adventure works]</pre>	<table> <tr> <th></th><th>Montant des Ventes (Internet)</th></tr> <tr> <td>Accessoire</td><td>\$700,759.96</td></tr> <tr> <td>Vélo</td><td>\$28,318,144.65</td></tr> <tr> <td>Vêtements</td><td>\$339,772.61</td></tr> <tr> <td>Composant</td><td>(null)</td></tr> </table>		Montant des Ventes (Internet)	Accessoire	\$700,759.96	Vélo	\$28,318,144.65	Vêtements	\$339,772.61	Composant	(null)
	Montant des Ventes (Internet)										
Accessoire	\$700,759.96										
Vélo	\$28,318,144.65										
Vêtements	\$339,772.61										
Composant	(null)										

b. Spécification de filtres (Slicers)

Dans la syntaxe d'une requête MDX, nous disposons d'une clause where dans laquelle on indique un set pour filtrer les données.

La requête suivante donne les volumes des vendus mais uniquement pour l'année 2010.


```
SELECT [Measures].[Internet Total Product Cost] on columns
FROM [Adventure Works]
where [Date].[Calendar].[Calendar Year].&[2010]
```

Internet Total Product Cost

25572,064

MDX permet d'utiliser plusieurs mesures par cube. Il suffit d'utiliser la dimension "[Measure]".

Notons ici que le calendrier 2010 n'est pas affiché puisqu'il est utilisé dans la clause Where.

c. Insertion de commentaires

Les commandes MDX peuvent être commentées de trois façons différentes.

// Commentaire en fin de ligne

-- Commentaire en fin de ligne

/* Commentaire sur plusieurs lignes */

Les commentaires peuvent être imbriqués comme le montre l'exemple ci-dessous

```
SELECT [Measures].[Internet Total Product Cost] on columns
/*commentaire
sur plusieurs lignes encore un test
/*commentaire imbriqué *//
FROM [Adventure Works]
where [Date].[Calendar].[Calendar Year].&[2010]
```

Internet Total Product Cost

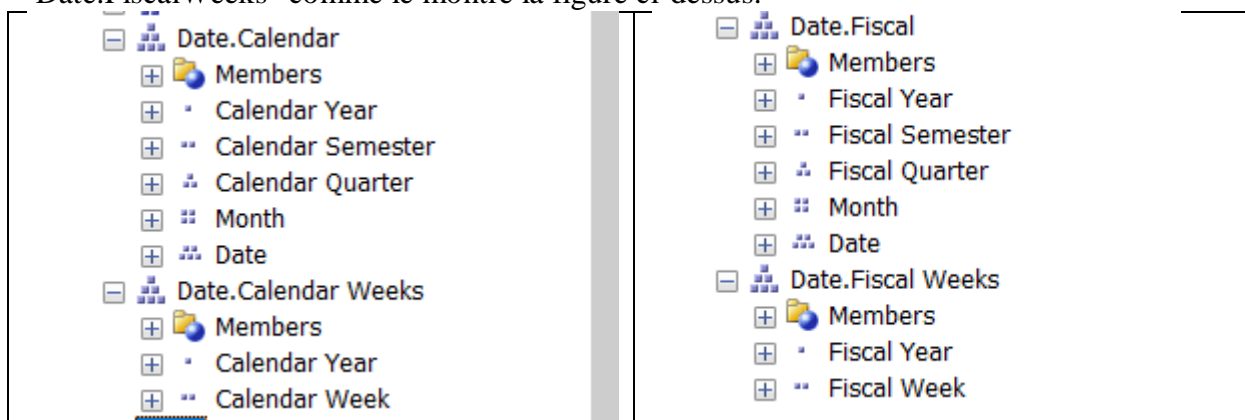
25572,064

II. Concepts de base

MDX est fait pour naviguer dans les bases multidimensionnelles et pour définir des requêtes sur tous les objets (dimensions, hiérarchies, niveaux, membres et cellules) afin d'obtenir (simplement) une représentation sous forme de tableaux croisés.

Il en découle une approche très hiérarchisée. Tout d'abord un cube est composé de dimensions.

Une dimension peut contenir une ou plusieurs hiérarchies. Par exemple, la dimension "Date" contient 4 hiérarchies : "Date.Calendar", "Date.CalendarWeek", "Date.Fiscal" et "Date.FiscalWeeks" comme le montre la figure ci-dessus.



MDX manipule des "sets" en les positionner sur des axes. Il est fastidieux de saisir un par un les membres d'un tuple pour construire des "sets". MDX dispose des fonctions pour retrouver les enfants ou les parents d'un membre, ou pour filtrer les données.

a. Hiérarchie

Une Hiérarchie est composée de niveaux ("levels"). Un niveau correspond à un des attributs de votre base de données. Le plus important est que la séquence des niveaux réponde à une logique de navigation. Par exemple, la hiérarchie "Date.calendar" est composée des niveaux Calendar Year, Calendar Semester, Calendar Quarter, Month et Date. Le sens et l'ordre de navigation sont déterministes.

Un niveau est lui-même composé de membres. Les membres sont les valeurs d'un niveau détectées par le moteur OLAP et stockées dans les métadonnées. Par exemple, les membres du niveau "Calendar Year" sont :

- + Member Properties
 - CY 2005
 - CY 2006
 - CY 2007
 - CY 2008
 - CY 2009
 - CY 2010
 - CY 2011
 - CY 2012
 - CY 2013
 - CY 2014

b. Membres

Un Membre est une instance d'un niveau. Les enfants ("childrens") d'un membre sont tous les membres du niveau immédiatement en dessous de celui-ci. "Food", "Drink" et "Non-Consumable" sont les membres de "Products".

Les membres sont des items accessibles dans les hiérarchies. On y fera référence de différentes manières. En voici quelques exemples suivant la profondeur dans la hiérarchie :

[Date].[Calendar Year].[Calendar Year]; [Organization].[Currency Code].&[6];

Nous pouvons illustrer l'utilisation des membres à l'aide des requêtes simples ci-dessus. Dans ces requêtes, nous utilisons d'affichage le mot clé on columns pour afficher les résultats obtenus dans les colonnes du tableau multidimensionnel.

```
Select [Measures].[Reseller Total Product Cost] on 0,  
[Date].[Calendar].[Calendar Year] on 1  
from [Adventure Works]
```

Calendar Year	Reseller Total Pro...
CY 2005	(null)
CY 2006	(null)
CY 2007	(null)
CY 2008	(null)
CY 2009	(null)
CY 2010	472295,1427
CY 2011	18163382,5768
CY 2012	27277732,4957
CY 2013	34066704,1638
CY 2014	(null)

c. Tuples

Un tuple est une suite de plusieurs membres entre parenthèses séparés par une virgule : (([Product].[Product].[Product],[Date].[Calendar].[Calendar Year]) est un exemple de tuple. Nous pouvons omettre les parenthèses s'il s'agit d'un tuple ne contenant qu'un seul membre. Un tuple donne la liste des membres qui identifie une ou plusieurs cellules dans un cube. Rappelons qu'une cellule est l'intersection des dimensions d'un cube de données. Dans un tuple, les mesures sont traitées comme une dimension particulière, nommée Measures. Dans l'exemple suivant, nous affichons la valeur de la mesure Reseller Total Product pour les différents canaux de vente et uniquement pour l'année 2010.

```
Select [Measures].[Reseller Total Product Cost] on 0,
([Sales Channel].[Sales Channel].[Sales Channel],[Date].[Calendar].[Calendar Year].&[2010]) on 1
from [Adventure Works]
```

		Coût Total Produit (Revendeur)
Internet	CY 2010	\$472,295.14
Reseller	CY 2010	\$472,295.14

d. Sets

Un set est un ensemble ordonné de tuples. Un set peut être vu comme une plage de valeurs. Le set commence par une accolade "{", dans laquelle sont énumérés les tuples séparés par des virgules, et se termine par une accolade appariée "}".

Ci-après, un exemple d'un set qui affiche deux mesures différentes et le même membre.

```
Select
{
([Measures].[Reseller Sales Amount],[product].[product categories].[category]),
([measures].[reseller tax amount],[product].[product categories].[category])} on 0
from [adventure works]
```

Montant des Ventes (Revendeur)	Montant des Ventes (Revendeur)	Taxes (Revendeur)	Taxes (Revendeur)	Taxes (Revendeur)
Vêtements	Composant	Accessoire	Vélo	Vêtements
\$1,777,840.84	\$11,799,076.66	\$45,703.83	\$5,304,190.40	\$142,227.25

L'exemple suivant permet d'afficher les différentes catégories des produits ainsi que le total des produits sur les colonnes. La mesure à afficher est reseller tax amount.

```
--SPECIFYING MEASURES IN WHERE CLAUSE
```

```
--WHERE CLAUSE TO GET PRODUCT WISE SALE OF SPECIFIC MEASURE
```

```
Select
{[product].[product categories].[category],[product].[product categories]} on columns
from [adventure works]
where
[measures].[reseller tax amount]
```

Accessoire	Vélo	Vêtements	Composant	Tous les Produits
\$45,703.83	\$5,304,190.40	\$142,227.25	\$943,926.12	\$6,436,047.61

e. Slicer /Dicer

Pour sélectionner une partie du cube, nous pouvons utiliser les slicer. Ceci permet d'afficher les valeurs pour une partie du cube. Il se fait d'une manière explicite en utilisant la clause where.

Par exemple, la requête suivante affiche les catégories de produits vendus avec la mesure par défaut Reseller Sales Amount.

```
Select
{
[product].[product categories].[category], [product].[product categories]
} on columns
```

```
from [adventure works]
```

Accessoire	Vélo	Vêtements	Composant	Tous les Produits
\$571,297.93	\$66,302,381.56	\$1,777,840.84	\$11,799,076.66	\$80,450,596.98

Quand un seul membre d'une dimension est spécifié en paramètre, il s'agit d'un Slice. S'il s'agit de plusieurs, il s'agit de Dice.

Dans l'exemple suivant, nous limitons les résultats aux produits vendus au Canada. Les colonnes affichées sont les mêmes mais les valeurs sont différentes.

```
Select
{
[product].[product categories].[category],[product].[product categories]} on columns
from [adventure works]
where
(
[geography].[geography].[country].[canada])
```

Accessoire	Vélo	Vêtements	Composant	Tous les Produits
\$118,127.35	\$11,636,380.59	\$378,947.63	\$2,244,470.02	\$14,377,925.60

Dans l'exemple suivant, nous limitons les résultats aux produits vendus au Canada et affichons la mesure internet sales amount. L'affichage est le même mais les valeurs sont différentes.

```
Select
{
[product].[product categories].[category],[product].[product categories]} on columns
from [adventure works]
where
(
[geography].[geography].[country].[canada],
[measures].[internet sales amount]
)
```

Accessoire	Vélo	Vêtements	Composant	Tous les Produits
\$700,759.96	\$28,318,144.65	\$339,772.61	(null)	\$29,358,677.22

La requête suivante effectue un slice pour afficher la mesure Internet Sales Amount des produits vendus mais uniquement pour le Canada et l'Australie.

```
Select
{
[product].[product categories].[category],
[product].[product categories]
} on columns
from [adventure works]
where
(
{
[customer].[customer geography].[country].[canada],
[customer].[customer geography].[country].[australia]
},
[measures].[internet sales amount]
)
```

Accessoire	Vélo	Vêtements	Composant	Tous les Produits
\$242,068.48	\$10,673,352.40	\$123,424.57	(null)	\$11,038,845.45

Notez que si on veut afficher les champs Canada, et Australie, il faut déplacer le filtre vers l'un des axes à afficher.

```
Select
{
```

```
[product].[product categories].[category],
[product].[product categories]
} on columns,
({
[customer].[customer geography].[country].[canada],
[customer].[customer geography].[country].[australia]
}),
[measures].[internet sales amount]) on 1
from [adventure works]
```

		Accessoire	Vélo	Vêtements	Composant	Tous les Produits
Canada	Montant des Ventes (Internet)	\$103,377.85	\$1,821,302.39	\$53,164.62	(null)	\$1,977,844.86
Australie	Montant des Ventes (Internet)	\$138,690.63	\$8,852,050.00	\$70,259.95	(null)	\$9,061,000.58

On peut aussi utiliser les opérateurs – (Moins) ou son équivalent Except pour enlever une partie des résultats

```
Select
{
[product].[product categories].[category], [product].[product categories]} on columns
from [adventure works]
where
(
{[Sales Territory].[Sales Territory].[country]
-
[Sales Territory].[Sales Territory].[country].[United Kingdom]},
[measures].[internet sales amount]
)
```

Accessoire	Vélo	Vêtements	Composant	Tous les Produits
\$624,129.92	\$25,035,301.99	\$307,533.10	(null)	\$25,966,965.01

```
Select
{
[product].[product categories].[category],
[product].[product categories]
}
on columns
from [adventure works]
where
([measures].[internet sales amount],
{
Except(
[Sales Territory].[Sales Territory].[country],
[Sales Territory].[Sales Territory].[country].[United Kingdom])
})
```

Accessoire	Vélo	Vêtements	Composant	Tous les Produits
\$624,129.92	\$25,035,301.99	\$307,533.10	(null)	\$25,966,965.01

f. Filter

La fonction Filter permet de filtrer les valeurs selon une condition particulière.

```
Filter(Set_Expression, Logical_Expression )
```

La requête suivante affiche uniquement les catégories de produit dont le montant des ventes est supérieur à 0.

```
--apply filter using filter function
```

```

Select
{
[measures].[internet sales amount]} on columns,
{
filter([product].[product categories].[category],
[measures].[internet sales amount] >0) } on rows
From [adventure works]

```

	Montant des Ventes (Internet)
Accessoire	\$700,759.96
Vélo	\$28,318,144.65
Vêtements	\$339,772.61

Le même résultat peut être obtenu en utilisant la fonction Having.

```

Select {[measures].[internet sales amount] } on columns,
{[product].[product categories].[category]}
having [measures].[internet sales amount] > 0 on rows
from [adventure works]

```

g. Opérateurs, AND, >

On peut filtrer les résultats en spécifiant deux conditions avec AND (ou OR) ou en comparant deux valeurs avec >, < et =. La requête suivante affiche les produits qui ont un montant des ventes par internet compris entre 500000 et 750000 \$.

```

Select { [measures].[internet sales amount] } on columns,
{filter([product].[product categories].[category],[measures].[internet sales amount]
>500000 AND [measures].[internet sales amount] <750000) } on rows
From [adventure works]

```

	Montant des Ventes (Internet)
Accessoire	\$700,759.96

La seconde affiche les produits dont le montant des ventes par internet est supérieur montant des ventes par revendeur.

```

Select { [measures].[internet sales amount]} on columns,
{filter([product].[product categories].[category], [measures].[internet sales amount]
> [measures].[reseller sales amount]) } on rows
From [adventure works]

```

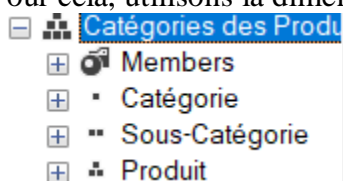
	Montant des Ventes (Internet)
Accessoire	\$700,759.96

h. Fonctions sur les membres et les dimensions

Les fonctions Members, CurrentMember et Children permettent de rechercher les membres d'une dimension, le membre en cours et les descendants directs (fils) d'un membre, respectivement. La fonction Members s'applique sur une dimension et la fonction Children sur un membre. La fonction CurrentMember s'applique à la fois sur une dimension ou un niveau d'une dimension:

<dimension>.CurrentMember ou < niveau >.CurrentMember. Il existe d'autres fonctions qui s'appliquent sur les membres comme Parent, NextMember, PrevMember, FirstSibling, LastSibling, FirstChild, LastChild, etc.

Pour cela, utilisons la dimension Produit et la hiérarchie correspondante.



La requête suivante affiche les membres de Category selon la hiérarchie Product Categories.

```
Select { [measures].[internet sales amount], [measures].[reseller sales amount]} on 0,
([Product].[Product Categories].[Category].MEMBERS) on 1
From [adventure works]
```

	Montant des Ventes (Internet)	Montant des Ventes (Revendeur)
Accessoire	\$700,759.96	\$571,297.93
Vélo	\$28,318,144.65	\$66,302,381.56
Vêtements	\$339,772.61	\$1,777,840.84
Composant	(null)	\$11,799,076.66

La fonction **Children** affiche les membres du niveau de hiérarchie immédiatement inférieur. Par exemple, dans la hiérarchie de la dimension Produit, elle affiche les membres de Catégorie, c'est-à-dire Accessoire, Vélo, Vêtements et Composant. Ceci permet de faire une opération Drill Through.

```
Select
{
[measures].[internet sales amount],
[measures].[reseller sales amount]
} on columns,

([Product].[Product Categories].children ) on rows
from
[adventure works]
```

	Montant des Ventes (Internet)	Montant des Ventes (Revendeur)
Accessoire	\$700,759.96	\$571,297.93
Vélo	\$28,318,144.65	\$66,302,381.56
Vêtements	\$339,772.61	\$1,777,840.84
Composant	(null)	\$11,799,076.66

De la même manière, elle affiche les membres de la Sous Catégorie pour la catégorie numéro 4, à savoir Accessoire.

```
Select {[measures].[internet sales amount],[measures].[reseller sales amount]} on 0,
([Product].[Product Categories].[Category].&[4].children) on 1
From [adventure works]
```

	Montant des Ventes (Internet)	Montant des Ventes (Revendeur)
Porte-vélo	\$39,360.00	\$197,736.16
Range-vélo	\$39,591.00	(null)
Bidon et porte-bidon	\$56,798.19	\$7,476.60
Nettoyant	\$7,218.60	\$11,188.37
Garde-boue	\$46,619.58	(null)
Casque	\$225,335.60	\$258,712.93
Sac d'hydratation	\$40,307.67	\$65,518.75
Éclairage	(null)	(null)
Antivol	(null)	\$16,225.22
Sacoche	(null)	(null)
Pompe	(null)	\$13,514.69
Pneu et chambre à air	\$245,529.32	\$925.21

La fonction Members peut-être combinée avec Levels(n) pour retrouver tous les membres du niveau n d'une dimension. Ci-dessus un exemple d'utilisation de la fonction Levels.

i. Is

Pour sélectionner les produits à afficher dans MDX, on utilise la fonction IS. La requête suivante permet d'afficher les mesures Internet et reseller sales amount pour les produits de la catégorie Bikes ou accessoires.

<pre> Select {[measures].[internet sales amount],[measures].[reseller sales amount]} on columns, { filter([product].[product categories].[category], ([product].[product categories].[CURRENTMEMBER IS [product].[product categories].[category].[accessories]) OR ([product].[product categories].[CURRENTMEMBER IS [product].[product categories].[category].[Bikes])) } on rows From [adventure works] </pre>		
	Montant des Ventes (Internet)	Montant des Ventes (Revendeur)
Accessoire	\$700,759.96	\$571,297.93
Vélo	\$28,318,144.65	\$66,302,381.56

j. Topcount/Bottomcount

La fonction Topcount/Bottomcount permet d'afficher les X premières/dernières valeurs. Appliquées à une mesure, elle affiche les x valeurs les plus grandes/petites.

Dans la première requête, elle affiche les 5 premières lignes du résultat, dans la seconde, les 5 plus grandes valeurs correspondant à la mesure Internet Sales Amount.

<pre> Select [measures].[internet sales amount] on columns, topcount([product].[product categories].[subcategory],5) on rows from [adventure works]; </pre>		<pre> Select [measures].[internet sales amount] on columns, topcount([product].[product categories].[subcategory],5,[measures].[internet sales amount]) on rows from [adventure works] </pre>	
	Montant des Ventes (Internet)		Montant des Ventes (Internet)
Porte-vélo	\$39,360.00	Vélo de route	\$14,520,584.04
Range-vélo	\$39,591.00	VTT	\$9,952,759.56
Bidon et porte-bidon	\$56,798.19	Vélo de randonnée	\$3,844,801.05
Nettoyant	\$7,218.60	Pneu et chambre à air	\$245,529.32
Garde-boue	\$46,619.58	Casque	\$225,335.60

La fonction bottomcount affiche les dernières valeurs. La requête suivante affiche les 10 produits qui ont la plus valeur non nulle de Internet Sales Amount.

<pre> Select {[measures].[internet sales amount],[measures].[reseller sales amount]} on columns, bottomcount(filter([product].[product categories].[subcategory],[measures].[internet sales amount]>0) ,10, [measures].[internet sales amount]) on rows from [adventure works] </pre>	
---	--

	Montant des Ventes (Internet)	Montant des Ventes (Revendeur)
Chaussettes	\$5,106.32	\$24,638.81
Nettoyant	\$7,218.60	\$11,188.37
Casquette	\$19,688.10	\$31,541.35
Gants	\$35,020.70	\$207,775.17
Veste	\$35,687.00	\$223,801.37
Porte-vélo	\$39,360.00	\$197,736.16
Range-vélo	\$39,591.00	(null)
Sac d'hydratation	\$40,307.67	\$65,518.75
Garde-boue	\$46,619.58	(null)
Bidon et porte-bidon	\$56,798.19	\$7,476.60

k. NonEmpty()

La fonction NonEmpty permet d'optimiser le temps d'exécution des requêtes en réduisant la taille des sets et ainsi accélérer les résultats de calcul en ne traitant que les cellules qui présentent des résultats. Ceci est visible pour les dimensions qui présentent de grandes valeurs comme Client et Date.

Testons avec l'exemple suivant. Il permet d'afficher les montants des ventes des clients par dates, clients, si les montants sont supérieur à 1000.

Ici, l'utilisation de la fonction Non Empty a permis d'afficher uniquement les valeurs filtrées non vides. Lors de l'exécution, toutes les combinaisons (Client, Date) sont créés même si elles ne présentent pas de ventes (cellules vides) avant d'être filtrées. Le temps d'exécution de la requête est de 5 minutes 34 secondes.

```
SELECT { [Measures].[Internet Sales Amount] } ON 0,
NON EMPTY Filter( ([Customer].[Customer].[Customer].MEMBERS
, [Date].[Date].[Date].MEMBERS ), [Measures].[Internet Sales Amount] > 1000 ) ON 1
FROM [Adventure Works]
```

Messages Results		
		Montant des Ventes (Internet)
Aaron A. Allen	December 2, 2011	\$3,399.99
Aaron A. Hayes	September 28, 2013	\$2,329.98
Aaron C. Campbell	September 10, 2013	\$1,155.48
Aaron C. Diaz	June 1, 2011	\$3,578.27
Aaron C. Diaz	March 25, 2013	\$2,451.30
Aaron C. Scott	July 20, 2013	\$2,492.32
Aaron E. Baker	September 9, 2013	\$1,750.98
Aaron E. Evans	June 11, 2013	\$2,433.04
Aaron Foster	July 7, 2013	\$2,430.44
Aaron Foster	December 7, 2013	\$2,482.03
Aaron Gonzales	December 16, 2013	\$1,810.46
Aaron J. Hughes	July 16, 2012	\$2,049.10
Aaron J. Hughes	September 1, 2013	\$2,407.04
Aaron L. King	January 13, 2013	\$2,359.98
Aaron L. King	April 16, 2013	\$2,398.05

L'utilisation de la fonction NonEmpty() permet de ne filtrer que les cellules où il ya une combinaison, c'est-à-dire un client a acheté et donc les cellules du cube qui contiennent une valeur. Toutes les autres combinaisons ne sont pas considérées qui sont Le temps d'exécution est égal à 5 secondes

```
SELECT { [Measures].[Internet Sales Amount] } ON 0,
```

```

NON EMPTY Filter( NonEmpty( ( [Customer].[Customer].[Customer].MEMBERS
[Date].[Date].[Date].MEMBERS ), { [Measures].[Internet Sales Amount] } ),
[Measures].[Internet Sales Amount] > 1000 ) ON 1
FROM [Adventure Works]

```

Messages Results		
		Montant des Ventes (Internet)
Aaron A. Allen	December 2, 2011	\$3,399.99
Aaron A. Hayes	September 28, 2013	\$2,329.98
Aaron C. Campbell	September 10, 2013	\$1,155.48
Aaron C. Diaz	June 1, 2011	\$3,578.27
Aaron C. Diaz	March 25, 2013	\$2,451.30
Aaron C. Scott	July 20, 2013	\$2,492.32
Aaron E. Baker	September 9, 2013	\$1,750.98
Aaron E. Evans	June 11, 2013	\$2,433.04
Aaron Foster	July 7, 2013	\$2,430.44
Aaron Foster	December 7, 2013	\$2,482.03
Aaron Gonzales	December 16, 2013	\$1,810.46
Aaron J. Hughes	July 16, 2012	\$2,049.10
Aaron J. Hughes	September 1, 2013	\$2,407.04
Aaron L. King	January 13, 2013	\$2,359.98
Aaron L. King	April 16, 2013	\$2,398.05

DESKTOP-2NH09QF | DESKTOP-2NH09QF\DELL | AdventureWorksDW2014Mu... | 00:00:05

```

SELECT
    [Measures].[Internet Sales Amount] ON COLUMNS
    ,Non Empty([Sales Territory].[Sales Territory Group].children) ON ROWS
FROM [Adventure Works]
WHERE [Product].[Category].[Bikes]

```

	Montant des Ventes (Internet)
Europe	\$8,644,932.72
North America	\$10,821,161.92
Pacific	\$8,852,050.00

IV. Fonctionnalités avancées de MDX

a. With Set

```

with
set TopSellingCities
as {
    [Geography].[Geography].[City].&[Toronto]&[ON]
    ,[Geography].[Geography].[City].&[London]&[ENG]
    ,[Geography].[Geography].[City].&[seattle]&[WA]
}
select
measures.[Reseller Sales Amount] on Columns
,TopSellingCities on rows
from [Adventure Works]

```

	Montant des Ventes (Revendeur)
Toronto	\$4,460,477.69
London	\$1,935,729.92
Seattle	\$1,494,860.53

a. With Member

Si un membre calculé n'est nécessaire que pour une seule requête MDX (Multidimensional Expressions), vous pouvez le définir à l'aide du mot clé WITH. Un membre calculé créé à l'aide du mot clé WITH n'existe plus une fois que l'exécution de la requête est terminée.

```
With
Member [measures].[customer sales] as [measures].[internet sales amount]
Member [measures].[retailer sales] as [measures].[reseller sales amount]
Member [measures].[Total Sales] as [measures].[internet sales amount]+
[measures].[reseller sales amount]
Select
{
[measures].[customer sales],
[measures].[retailer sales],
[measures].[total sales]
} on columns,
non empty( {
[date].[calendar].[calendar year]
}) on rows
from [adventure works]
```

	customer sales	retailer sales	Total Sales
CY 2010	\$43,421.04	\$489,328.58	\$532,749.62
CY 2011	\$7,075,525.93	\$18,192,802.71	\$25,268,328.64
CY 2012	\$5,842,485.20	\$28,193,631.53	\$34,036,116.73
CY 2013	\$16,351,550.34	\$33,574,834.16	\$49,926,384.50
CY 2014	\$45,694.72	(null)	\$45,694.72

```
WITH MEMBER [Measures].AnnualResellerSales As
CASE
    WHEN [Measures].[Reseller Sales Amount] > 750000 THEN 'Very High Sales'
    WHEN [Measures].[Reseller Sales Amount] > 50000 THEN 'High Sales'
    WHEN [Measures].[Reseller Sales Amount] > 25000 THEN 'Medium Sales'
    WHEN [Measures].[Reseller Sales Amount] > 0 THEN 'Low Sales'
    ELSE 'No Sale'
END
SELECT [Date].[Calendar].[Calendar Year].members on 0
,NON EMPTY [Reseller].[Reseller].children on 1
FROM [Adventure Works]
WHERE [Measures].AnnualResellerSales
```

	CY 2005	CY 2006	CY 2007	CY 2008	CY 2009	CY 2010	CY 2011	CY 2012	CY 2013	CY 2014
A Bicycle Association	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale
A Bike Store	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	High Sales	No Sale	No Sale	No Sale
A Cycle Shop	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale
A Great Bicycle Company	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	Low Sales	Low Sales	Low Sales	No Sale
A Typical Bike Shop	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	High Sales	No Sale	No Sale	No Sale
Acceptable Sales & Service	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	Low Sales	No Sale	No Sale	No Sale
Accessories Network	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	Low Sales	No Sale
Acclaimed Bicycle Company	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	Low Sales	No Sale
Ace Bicycle Supply	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	Low Sales	Low Sales	No Sale
Action Bicycle Specialists	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	High Sales	No Sale
Active Cycling	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	Low Sales	No Sale	No Sale
Active Life Toys	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	High Sales	No Sale	No Sale
Active Systems	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	Low Sales	Low Sales	No Sale	No Sale
Active Transport Inc.	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	High Sales	Low Sales	Low Sales	No Sale
Activity Center	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	Low Sales	Low Sales	Low Sales	No Sale
Advanced Bike Components	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	High Sales	High Sales	High Sales	No Sale
Aerobic Exercise Company	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	Low Sales	No Sale
Affordable Sports Equipment	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	High Sales	High Sales	No Sale	No Sale
All Cycle Shop	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	Low Sales	Low Sales	Low Sales	No Sale
All Seasons Sports Supply	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	Low Sales	Low Sales	No Sale
Alpine Ski House	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	Low Sales	Low Sales	No Sale	No Sale
Alternative Vehicles	No Sale	No Sale	No Sale	No Sale	No Sale	No Sale	Low Sales	No Sale	No Sale	No Sale

b. ParallelPeriod

Retourne un membre d'une période antérieure dans la même position relative que le membre spécifié.

La fonction ParallelPeriod prend l'ancêtre du membre spécifié au niveau spécifié, recherche le frère de l'ancêtre avec le décalage spécifié, puis retourne la période parallèle du membre spécifié parmi les descendants du frère.

Dans l'exemple suivant, on affiche la valeur de deux niveaux avant le trimestre actuel (4-2) ce qui nous renvoie au semestre 2 qui commence par avril. retourne de niveau

```
SELECT ParallelPeriod ([Date].[Calendar].[Calendar Quarter]
, 2
, [Date].[Calendar].[Month].[October 2013])
ON 0
FROM [Adventure Works]
```

April 2013

\$3,483,161.40

<pre>SELECT ParallelPeriod ([Date].[Calendar].[Calendar Semester] , 3 , [Date].[Calendar].[Month].[October 2013]) ON 0 FROM [Adventure Works]</pre>		
<table> <tr> <td>April 2012</td> </tr> <tr> <td>\$3,053,816.33</td> </tr> </table>	April 2012	\$3,053,816.33
April 2012		
\$3,053,816.33		

Pour l'examen.

<https://docs.microsoft.com/en-gb/sql/analysis-services/multidimensional-models/mdx/mdx-named-sets-creating-query-scoped-named-sets?view=sql-server-2014>

https://github.com/PacktPublishing/MDX-with-Microsoft-SQL-Server-2016-Analysis-Services-Cookbook/blob/master/Chapter%201/B05651_CH1_R4_001.mdx

<https://www.codeproject.com/Articles/802799/Reading-OLAP-Cube-using-MDX-Part-2>

https://books.google.tn/books?id=rqDcDgAAQBAJ&printsec=frontcover&hl=fr&source=gbv_atb#v=onepage&q&f=false