

# L'informatique des entrepôts de données

Daniel Lemire

## SEMAINE 10 Introduction à MDX

### 10.1. Présentation de la semaine

Tout comme le modèle relationnel utilise SQL comme langage, le modèle OLAP utilise le langage MultiDimensional eXpressions (MDX). Mis au point à la fin des années 90 par Microsoft, MDX est rapidement devenu la norme. Nous allons étudier le langage à partir d'une mise en œuvre open source au sein du moteur OLAP Mondrian.

### 10.2. Un peu de contexte

MDX fut inventé par Mosha Pasumansky. Il a été 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. En superficie, MDX est proche du SQL avec ses instructions **select** et **where**, même si la similarité ne va pas plus loin. Alors que l'objectif visé au départ par MDX était de rendre aisé et intuitif l'accès aux données de différentes dimensions, vous pourrez constater que MDX est tout de même destiné avant tout aux experts OLAP.

On trouve beaucoup de documentation sur le web ayant trait à MDX et vous êtes invité à faire vos propres recherches. Commencez par consulter l'article sur MDX dans wikipedia : [http://fr.wikipedia.org/wiki/Multidimensional\\_Expressions](http://fr.wikipedia.org/wiki/Multidimensional_Expressions).

Par ailleurs, Microsoft rend disponible la documentation concernant MDX en ligne à l'adresse [http://technet.microsoft.com/fr-fr/library/ms145595\(SQL.90\).aspx](http://technet.microsoft.com/fr-fr/library/ms145595(SQL.90).aspx). Vous devriez en prendre rapidement connaissance. Vous trouverez davantage de références en anglais, comme à l'adresse <http://www.databasejournal.com/features/article.php/3593466/MS-SQL-Series.htm#mdx>.

### 10.3. À propos de Mondrian

Mondrian est un serveur OLAP open source disponible depuis 2002. Il est principalement développé par la société Pentaho. Mondrian ne stocke pas les données, mais agit plutôt comme couche OLAP par dessus votre moteur de base de données (par ex. Oracle, Apache Derby, PostgreSQL, MySQL, Microsoft Access, SQL Server, etc.). Mondrian est écrit en Java et fonctionne sous la plupart des systèmes d'exploitation (Windows, Mac, Linux, etc.).

Mondrian a donc le bénéfice d'une certaine neutralité. Il ne présume rien quant à votre infrastructure.

Consultez l'article dans Wikipedia : [https://en.wikipedia.org/wiki/Mondrian\\_OLAP\\_server](https://en.wikipedia.org/wiki/Mondrian_OLAP_server) et <https://db.apache.org/derby/>.

### 10.4. Approche pédagogique

Le meilleur moyen d'apprendre un langage est de mettre la main à la pâte. Nous allons donc travailler directement avec un moteur OLAP supportant MDX. Au lieu d'apprendre la syntaxe formelle du MDX, nous allons plutôt procéder par exemples.

### 10.5. Installation de Mondrian Derby

Avant de débiter, vous devez installer Mondrian Derby. Suivez avec rigueur les instructions suivantes. En cas de problème, faites des saisies d'écran complètes montrant la totalité de vos manipulations techniques depuis le début et transmettez-les à la personne chargée de votre encadrement. Elle pourra vérifier avec vous que vous avez suivi toutes les étapes, point par point. Sur la plupart des PC, on peut prendre une saisie d'écran en appuyant la touche "Print" ou "Scrn" ou "Impr écran Syst", vous pouvez ensuite coller l'image dans votre logiciel favori.

Notez bien : si vous écrivez à la personne qui vous encadre pour demander de l'aide, mais que vous n'incluez pas des saisies d'écran complètes, vous indiquez en même temps que vous n'avez pas bien compris ou suivi les présentes consignes.

Attention : vous devez suivre des consignes techniques détaillées. Il est parfois tentant de passer outre ou de ne pas tout lire. Notez qu'il est impossible de vous aider si c'est ce que vous choisissez de faire. Si vous improvisez, c'est à vos risques!

Encore une fois : lisez attentivement les consignes suivantes.

- Vérifiez si vous avez une machine virtuelle Java à jour. Allez en ligne de commande. Sous Windows 8 et versions ultérieures, cliquez en bas à gauche avec le bouton droit de la souris et sélectionnez invite de commandes. Sous les versions précédentes de Windows, dans le menu Démarrer, cliquez sur Exécuter; une fenêtre s'ouvre, tapez le nom de programme `cmd` pour ouvrir une fenêtre de commande. Sous Mac OS, lancez le programme **Applications/Utilities/Terminal**. La commande `javac` doit être reconnue. Vous devez vérifier la version de Java dont vous disposez en tapant `javac -version`.
- Si vous n'avez pas le JDK, vous devrez l'installer. Sous Windows, vous pouvez charger la dernière version de Java à partir de l'adresse <https://adoptopenjdk.net>. (Si ce lien ne fonctionne pas, cherchez "java jdk" dans un moteur de recherche.) Assurez-vous d'installer le JDK et non le JRE. Vous devez choisir une version qui corresponde à votre système d'exploitation. Pour Windows, il y a deux version du JDK : la version 32 bits (x86) et la version 64 bits (x64). Vous devez choisir la version 32 bits si votre Windows est une version 32 bits et la version 64 bits si votre version de Windows est une version 64 bits. En cas de doute, ou si vous avez une version de Windows un peu ancienne, choisissez la version 32 bits.
- Si la commande `javac` n'est pas reconnue, mais que vous disposez d'une machine virtuelle Java (peut-être venez vous de l'installez), trouvez l'exécutable `javac` sur votre machine (par exemple `c:\mondirjava\bin\javac.exe`) puis ajoutez le dossier correspondant à la variable PATH en tapant,  
`SET PATH=c:\mondirjava\bin\`  
(Vous devrez peut-être faire cela à chaque fois que vous voudrez démarrer Mondrian. Les utilisateurs avancés peuvent modifier définitivement les variables d'environnement, mais ça ne sera pas nécessaire dans ce cours.)
- Avant de continuer, assurez-vous que lors que vous tapez `java -version` et `javac -version`, vous n'obtenez pas de message d'erreur. Par ailleurs, les deux commandes devraient indiquer la même version de Java. Vous devez avoir la version 1.6 ou

mieux. Si ça ne fonctionne toujours pas, ne continuez pas! Reprenez tout depuis le début.

- Attention: si la commande `java` fonctionne, mais que la commande `javac` ne fonctionne pas, ne continuez pas!
- Sous Windows, vous devez aussi définir la variable `JAVA_HOME` comme ceci : `SET JAVA_HOME=c:\mondirjava`. Notez que l'installateur de `adoptopenjdk` fait ce travail automatiquement dans la mesure du possible à moins d'instruction contraire de votre part. (Vous devrez peut-être faire cela à chaque fois que vous voudrez démarrer Mondrian.) Faites `echo %JAVA_HOME%` en ligne de commande pour vérifier votre travail.
- Sous Windows, la commande `javac -version` vous indiquera si vous avez la version 64 bits. S'il n'y a pas cette mention, c'est que vous avez la version 32 bits.
- Chargez Mondrian Derby 3.5.0 à partir de l'adresse <http://sourceforge.net/projects/mondrian/files/mondrian/mondrian-3.5.0/>. Vous devez trouver une archive zip nommée `mondrian-3.5.0-derby.zip`. Dans cette archive, vous trouverez le fichier `lib/mondrian-embedded.war`. Nous en aurons besoin. **Attention:** Si vous chargez une autre archive que `mondrian-3.5.0-derby.zip`, il est possible que vous n'arriviez pas à faire fonctionner Mondrian. Tous les tests ont été réalisés avec cette version précise de Mondrian. Si vous obtenez une erreur de type `java.lang.NoClassDefFoundError` en exécutant des requêtes MDX, c'est que vous avez installé une mauvaise version de Mondrian. En particulier, la version 3.2 de Mondrian nécessite une étape d'installation supplémentaire. De plus, il faut absolument utiliser la version derby de Mondrian, à moins que vous souhaitiez configurer vous-même une base de données.
- Chargez Tomcat à partir de l'adresse <http://tomcat.apache.org/> (cherchez un lien intitulé Download et chargez un fichier zip). Sous Windows, vous devez charger la version qui correspond à votre JDK (32 bits ou 64 bits). Vous devez extraire le contenu de l'archive dans un dossier, par exemple sur votre bureau. Les tests furent réalisés avec les versions 8.0 de Tomcat.
- Prenez en nom l'emplacement du dossier Tomcat. Pour faciliter le suivi, l'emplacement du dossier ne devrait contenir aucun caractère spécial inutile ni aucun espace. Par exemple, ne placez pas Tomcat dans le dossier "C:/Program Files" ou "C:/Mon Cours", préférez plutôt "C:/MonCours". Choisissez

aussi des noms de dossier courts (par ex., moins de 32 caractères).

- Copiez le fichier `mondrian-embedded.war` dans le dossier `webapps` de Tomcat. Avant de continuer, assurez-vous que le fichier a bien été copié au bon endroit.
- Visitez l'adresse `http://localhost:8080` dans votre navigateur web (par ex. Firefox). Vous devez obtenir un message d'erreur. En effet, il ne devrait pas y avoir de serveur sur votre machine utilisant le port 8080. Si jamais c'est le cas, prenez-en note. Il vous faudra modifier le port sur lequel le serveur Tomcat fonctionne. Ouvrez le fichier `conf/server.xml` de Tomcat avec un éditeur de texte et remplacez le port 8080 par le port 8090 (ou tout autre port disponible). Nous supposons dans toutes les instructions qui suivent que vous utilisez le port 8080 : c'est à vous d'interpréter les instructions en remplaçant 8080 par le numéro du port que vous utilisez.
- Tomcat est un serveur web écrit en Java. Quand vous allez le démarrer, il ne va rien se passer en apparence. Cependant, le serveur web va rouler sur votre machine, localement, jusqu'à ce que vous y mettiez fin. Demarrez Tomcat maintenant. Sous Windows, il faut exécuter le script `bin\startup.bat`. Allez donc en ligne de commande dans le dossier `bin` et tapez `startup.bat`. Sous Linux et Mac OS, exécutez plutôt le script `bin/startup.sh`. (Allez donc en ligne de commande dans le dossier `bin` et tapez `./startup.sh`.) Toujours sous Linux et Mac OS, il peut être utile de changer la permission des fichiers à l'extension `sh` pour les rendre exécutables : allez dans le dossier `bin`, puis tapez `chmod +x *.sh`. Si vous obtenez une erreur portant sur les variables d'environnement (`JAVA_HOME` ou `JRE_HOME`), votre environnement Java est mal configuré: il faut peut-être réinstaller votre environnement Java. Si vous obtenez une erreur mentionnant un conflit entre une version 32 bits et une version 64 bits, c'est que votre version de Tomcat ne correspond pas à votre version du JDK : ils doivent toutes les deux être 32 bits ou toutes les deux être 64 bits. Avant de continuer, assurez-vous que Tomcat fonctionne en visitant le site `http://localhost:8080` dans votre navigateur web. Vous devriez voir une page web avec la mention **Apache Tomcat**, puis ce message `If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!`. Si Tomcat ne fonctionne pas, il est inutile de continuer. Vous devez recommencer.

- Vérifiez que Mondrian a été reconnu par Tomcat. Allez dans le dossier `logs` de Tomcat et ouvrez le fichier `catalina.out` dans un éditeur de texte ou dans un navigateur web. Vous devriez voir une ligne ressemblant à celle-ci : `Deploying web application archive mondrian-embedded.war`. Cette ligne devrait être suivie peu après d'une autre ligne qui indique : `Deployment of web application archive mondrian-embedded.war has finished`. Si ce n'est pas le cas, c'est que vous avez omis une étape. Vous devez alors fermer Tomcat et recommencer. Pour terminer Tomcat, exécutez le script `bin/shutdown.bat` sous Windows ou `bin/shutdown.sh` sous Linux ou Mac OS. À tout moment, en cas de problème inexpliqué, vous pouvez toujours consulter le fichier `catalina.out` pour obtenir un diagnostic.
- Rendez-vous à l'adresse `http://localhost:8080/mondrian-embedded/`, vous devez voir une page intitulée **Mondrian OLAP Server**.
- Afin de tester votre installation, suivez le lien **JPivot pivot table**. Vous devriez voir une interface graphique se charger. Si ça ne fonctionne pas, et en particulier si vous obtenez une erreur Java de type `java.lang.NoClassDefFoundError`, assurez-vous d'avoir installé la bonne version de Mondrian. Dans le cas contraire, vous devez recommencer l'installation.
- Allez dans le répertoire `webapps` qui se trouve dans le répertoire de Tomcat. Vous devriez y trouver un répertoire `mondrian-embedded`. Dans celui-ci, vous devriez trouver plusieurs fichiers à l'extension `.jsp`. Ceux-ci proviennent de l'archive `mondrian-embedded.war` (qui est en fait un fichier compressé zip). Au démarrage de Tomcat, celui-ci découvre le fichier `mondrian-embedded.war` et décompresse l'archive, copiant de fait les fichiers de l'archive vers le répertoire `mondrian-embedded` sur votre disque. Vérifiez aussi qu'il s'y trouve un dossier `WEB-INF` contenant des fichiers. Vérifiez-en le contenu. Vous devriez avoir notamment les fichiers suivants :
  - `jpivot/chart/chart.xml`
  - `jpivot/navi/hierarchy-navigator.xml`
  - `jpivot/navi/navigator.xml`
  - `jpivot/showxml.xml`
  - `jpivot/table/fo_mdxtable.xml`
  - `jpivot/table/mdxcell.xml`
  - `jpivot/table/mdxslicer.xml`
  - `jpivot/table/mdxtable.xml`

The screenshot shows the Mondrian web interface. At the top, there is a dropdown menu labeled 'Sample Query #0' and a 'show query' button. Below this is a text area containing the MDX query: `select [Time].[1997] on columns from [Sales]`. Under the text area is a 'process MDX query' button. Below the button, the word 'Results:' is displayed. Under 'Results:', there is a table with two rows: the first row contains the header `[Time].[1997]` and the second row contains the value `266,773`.

**Figure 1.** Exemple de requête en ligne avec Mondrian

- `jpivot/table/xls_mdxtable.xsl`
- `jpivot/toolbar/htoolbar.xsl`
- `jpivot/toolbar/vtoolbar.xsl`

Ceux-ci doivent être présents puisqu'ils sont dans l'archive `mondrian-embedded.war`.

- Revenez ensuite à l'adresse `http://localhost:8080/mondrian-embedded/`. Suivez le lien intitulé **Basic interface for ad hoc queries**<sup>1</sup>. Il vous donne accès à une interface permettant de saisir des requêtes MDX. (Normalement, nous devrions aussi installer une base de données et nos propres données, mais nous utiliserons ici les données test fournies par défaut avec Mondrian.) Les requêtes MDX vont dans la boîte de saisie. On exécute la commande en pressant le bouton **process MDX query**. Testez que votre système est fonctionnel en saisissant la requête `select [Time].[1997] on columns from [Sales]`. Vous devriez obtenir un résultat semblable à la Fig. 1. Vous pouvez ignorer le reste de l'interface (Sample Query, show query, etc.).
- Lorsque vous avez terminé votre travail, exécutez le script `bin/shutdown.bat` sous Windows ou `bin/shutdown.sh` sous Linux ou Mac OS afin de fermer Tomcat. Si vous ne faites pas cette opération, et ne fermez pas votre ordinateur, Tomcat continuera de fonctionner, utilisant ainsi de la mémoire sur votre ordinateur. De plus, si vous tentez de repartir Tomcat sans d'abord l'avoir fermé, vous obtiendrez un message d'erreur.

<sup>1</sup>Si vous suivez d'autres liens, vous obtiendrez peut-être une erreur.

## 10.6. Démarche recommandée

Vous devez reproduire tous les exemples, un à un, et vous assurer de les comprendre. Ne copiez pas trop rapidement : prenez plutôt le temps de réécrire les commandes vous-même.

Mondrian n'est pas toujours très convivial. Lorsque vous commettez une erreur dans une expression MDX, il est possible que le message d'erreur vous paraisse incompréhensible. Soyez patient!

## 10.7. Données

Normalement, dans une entreprise, il faut créer ses propres cubes. La collecte de données et la configuration des dimensions, qui forment les cubes est une opération un peu fastidieuse. Il faut aussi installer et choisir un moteur de bases de données, il faut charger les données, configurer les index, et ainsi de suite. Pour nos besoins, nous allons utiliser les données déjà présentes dans votre installation de Mondrian, sous le nom de FoodMart. FoodMart est composé des cubes suivants :

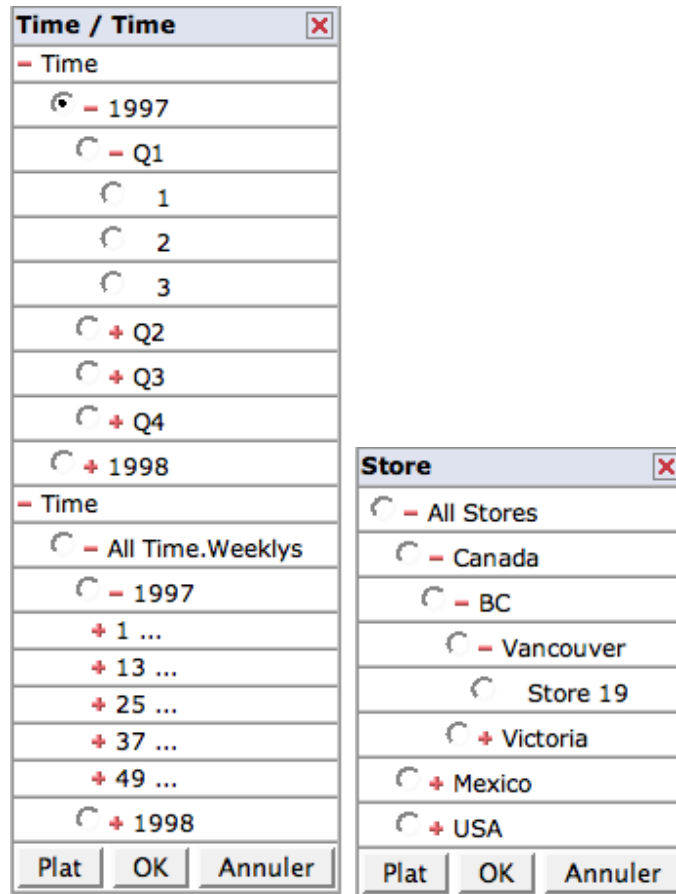
- Le cube Warehouse comprend les mesures Store Invoice, Supply Time, Warehouse Cost, Warehouse Sales, Units Shipped, Units Ordered, Warehouse Prot, et les dimensions Store, Store Size in SQFT, Store Type, Time, Product, Warehouse.
- Le cube Store comprend les mesures Store Sqft, Grocery Sqft et les dimensions Store, Store Type, Has coffee bar, Grocery sqft
- Le cube HR comprend les mesures Org Salary, Count, Number of Employees, Employee Salary, Avg Salary et les dimensions Time, Store, Pay Type, Store Type, Position, Department, Employees
- Le cube Sales comprend les mesures Unit Sales, Store Cost, Store Sales, Sales Count, Customer Count et les dimensions Store, Store Size in SQFT, Store Type, Time, Product, Promotion Media, Promotions, Customers, Education Level, Gender, Marital Status, Yearly Income, Has bought dairy

Le cube Sales nous intéresse plus particulièrement à cause de son grand volume.

## 10.8. Concepts de base

A lire : [http://fr.wikipedia.org/wiki/Arbre\\_enraciné](http://fr.wikipedia.org/wiki/Arbre_enraciné)





**Figure 2.** Illustration de hiérarchies MDX

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 une représentation sous forme de tableaux. En MDX, un cube est composé de dimensions. Une dimension peut contenir une ou plusieurs hiérarchies. Par exemple, la dimension Time contient deux hiérarchies : Year, Quarter, Month et Year, Week, Day, comme le montre la Fig. 2. Dans cet exemple, la première hiérarchie est celle par défaut alors que la seconde se nomme Weekly. Dans la pratique, du moins pour le contexte de ce cours, on peut généralement ignorer le fait qu'une dimension peut avoir plus d'une hiérarchie et faire comme s'il n'y avait qu'une hiérarchie par dimension qui s'applique par défaut.

Une hiérarchie est composée de niveaux (*levels*). Par exemple, la hiérarchie Time est composée des niveaux Year, Quarter et Month; la hiérarchie Store est composée des niveaux Country, State, City et Store

Name. On numérote ces niveaux en commençant par la valeur unitaire (Year est 1, Quarter est 2, et ainsi de suite).

Un niveau est lui-même composé de membres (ou valeurs). Par exemple, les membres du niveau Country sont Canada, Mexico et USA (voir Fig. 2). Il y a toujours un niveau zéro qui contient un seul membre (All) qui contient tous les membres du premier niveau (celui-ci est toujours présent par défaut). Les membres forment un arbre: chaque membre peut contenir d'autres membres d'un niveau inférieurs (ses enfants ou *children*). Par exemple, les enfants du membre USA sont les différents états américains. Les enfants d'un membreAm appartiennent au prochain niveau (les états Américain seront du niveau 2).

Nous pouvons illustrer l'utilisation des membres à l'aide des requêtes simples ci-dessus. Dans ces requêtes, nous utilisons l'instruction `on columns` qui stipule que la dimension sélectionnée détermine le contenu des colonnes. Nous reviendrons sur cette instruction par la suite.

```
select [Time].[1997] on columns
from [Sales]
```

```
select [Product].[Food] on columns
from [Sales]
```

```
select [Product].[Food].[Baked Goods] on columns
from [Sales]
```

Notez comment on spécifie le membre `[Product].[Food].[Baked Goods]` en spécifiant un membre du premier niveau (Food) puis du second niveau (Baked Goods). On omet généralement de spécifier le membre de niveau zéro car il n'y en a qu'un. Néanmoins, on pourrait écrire `[Product].[All Products].[Food].[Baked Goods]` au lieu de `[Product].[Food].[Baked Goods]`.

La mesure par défaut dans le cube Sales est Unit Sales. Nous verrons bientôt comment stipuler la mesure.

### 10.8.1. Tuples

Un *tuple* est une suite de plusieurs membres entre parenthèses séparés par une virgule :

```
([Time].[1997], [Product].[Food])
```

est un exemple de tuple. Nous pouvons omettre les parenthèses s'il s'agit d'un tuple ne contenant qu'un seul membre.

On peut utiliser une requête avec un tuple pour stipuler que nous souhaitons obtenir le résultat des ventes durant l'année 1997, mais seulement en ce qui concerne les Baked Goods :

```
select ([Product].[All Products].[Food].[Baked Goods],
```

```
[Time].[1997]) on columns
from [Sales]
```

Avec un tuple, on peut spécifier une autre mesure. Par exemple, si on s'intéresse plutôt au nombre de clients (mesure Customer Count), on peut utiliser cette requête :

```
select ([Product].[All Products].[Food].[Baked Goods],
[Time].[1997],
[Measures].[Customer Count]) on columns
from [Sales]
```

En pratique, les mesures se traitent comme une dimension, mais dont le nom est toujours Measures.

### 10.8.2. Sets

Il est un peu pénible de construire une requête MDX pour chaque tuple. On peut faire mieux.

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 }.

Ainsi, il est possible de traiter deux mesures distinctes dans une même requête :

```
select
{
  ([Measures].[Unit Sales],
  [Product].[All Products].[Food].[Baked Goods]),
  ([Measures].[Store Sales],
  [Product].[All Products].[Food].[Baked Goods])
}
on columns
from [Sales]
```

Bien sûr, on peut aussi, par exemple, énumérer des catégories de produits :

```
select
{
  ([Measures].[Unit Sales], [Product].[Food]),
  ([Measures].[Unit Sales], [Product].[Drink])
}
on columns
from [Sales]
```

On peut varier à la fois la mesure, mais aussi les dimensions et les niveaux hiérarchiques :

```
select
{
  ([Measures].[Unit Sales], [Product].[Food]),
  ([Measures].[Store Sales], [Product].[Food].[Baked Goods])
}
on columns
from [Sales]
```

Bien sûr, un set peut ne contenir qu'un seul élément :

```
select
{
  ([Time].[1997])
}
on columns
from [Sales]
```

## 10.9. Syntaxe de base

Maintenant que nous connaissons les tuples et les sets, nous allons explorer davantage les requêtes MDX. Voici le prototype d'une requête :

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

### 10.9.1. Spécification d'axes

Sous Mondrian, il y a deux axes, les colonnes (columns) et les rangées (rows). On spécifie l'axe avec le mot-clé **on**. La requête suivante donne les unités vendues "[Measures].[Unit Sales]" par an pour les produits "Drink" et "Food".

```
select
{
  ([Measures].[Unit Sales], [Product].[Food]),
  ([Measures].[Unit Sales], [Product].[Drink])
} on columns,
{
  ([Time].[1997]),
  ([Time].[1998])
}
```

```
} on rows  
from [Sales]
```

(Quand Mondrian n'a pas de données correspondant à une cellule, il n'affiche rien dans la cellule correspondante.)

### 10.9.2. Spécification de filtres (Slicers)

On utilise la clause **where** pour filtrer les cellules. Imaginons que nous souhaitions n'avoir que les ventes concernant les consommateurs de sexe masculin. Nous obtenons ce résultat avec la requête suivante :

```
select  
{  
    ([Measures].[Unit Sales], [Product].[Food]),  
    ([Measures].[Unit Sales], [Product].[Drink])  
  
}  
on columns,  
{  
    ([Time].[1997]),  
    ([Time].[1998])  
}  
on rows  
from [Sales]  
where  
{  
    ([Gender].[M])  
}
```

On peut aussi utiliser la clause **where** pour stipuler une autre mesure que la mesure par défaut.

```
select  
{([Product].[Drink])}on columns,  
{([Time].[1997])} on rows  
from [Sales]  
where {[Measures].[Store Cost]}
```

### 10.9.3. Insertion de commentaires

On peut insérer les commentaires dans du code MDX de trois façons différentes.

```
// Commentaire en fin de ligne  
-- Commentaire en fin de ligne
```

```

/* Commentaire
   sur plusieurs lignes
*/

```

## 10.10. Remerciements

Mondrian n'est pas parfait, mais il a l'avantage d'être gratuit ce qui est bien pratique pour apprendre MDX sans dépenser une fortune. Il faut donc remercier l'auteur de Mondrian, Julian Hyde, et tous ses collaborateurs. Mondrian est un produit de la société d'intelligence d'affaires Pentaho. Julian Hyde a un blogue : <http://julianhyde.blogspot.com/>.

## 10.11. Questions d'approfondissement

- (a) Calculez la mesure Store Cost pour les produits [Product].[All Products].[Drink].[Alcoholic Beverages] et [Product].[All Products].[Drink].[Dairy], et pour les deux sexes ([Gender].[M], [Gender].[F]). Votre réponse doit comprendre quatre cellules.
- (b) Pour l'année 1997, trouvez les valeurs des mesures Store Cost et Unit Sales pour les produits de type Drink. Formulez votre requête de deux manières distinctes : une fois avec l'instruction `where` et une fois sans.

## 10.12. Réponses suggérées

```

(a) select
    [Product].[All Products].[Drink].[Alcoholic Beverages],
    [Product].[All Products].[Drink].[Dairy]} on columns,
    {[Gender].[M], [Gender].[F]} on rows
from [Sales]
where {[Measures].[Store Cost]}

```

```

select
  {[Product].[Drink]} on columns,
  {[Measures].[Store Cost],
  {[Measures].[Unit Sales]} on rows
from [Sales]
([Time].[1997], [Measures].[Unit Sales]) } on rows
from [Sales]

/**
 * on
 **/

select
  {[Product].[Drink]} on columns,
  {[Time].[1997], [Measures].[Store Cost],
  {[Measures].[Unit Sales]} on rows
from [Sales]
where [Time].[1997]

```

### 10.13. En terminant

N'oubliez pas de fermer Tomcat lorsque vous aurez terminé. Il faut exécuter le script `bin/shutdown.bat` sous Windows ou `bin/shutdown.sh` sous Linux ou Mac OS. Si vous ne le faites pas, Tomcat continuera de fonctionner. Par contre, conservez votre installation de Mondrian, vous en aurez encore besoin.