

Business Intelligence - M1DAC OLAP

Ludovic Denoyer et Laure Soulier
Université Pierre et Marie Curie
LIP6, Paris - France

6 février 2017

ENTREPÔT DE DONNÉES

Les entrepôts de données (data warehouse) sont :

- Orientés sujet
 - ▶ Les données sont organisées par sujet ou *faits* (ex : clients, produits, ventes, etc.).
 - ▶ Les données sont organisées selon des dimensions
- Intégrés
 - ▶ Les données, qui proviennent de diverses sources hétérogènes, sont consolidées et intégrées dans l'entrepôt.
- Historiques
 - ▶ Les données ont très souvent une composante temporelle (ex : date et heure d'une transaction).
- Non-volatiles
 - ▶ Une fois insérées dans l'entrepôt, les données ne sont jamais modifiées ou effacées ; elle sont conservées pour des analyses futures.

MODÈLE DIMENSIONNEL

Contrairement aux systèmes opérationnels, le stockage des données dans un DW se fait habituellement sous la forme d'un schéma dimensionnel. Un tel schéma nécessite de définir :

- **des dimensions**
- **des faits**

Dimensions

Les dimensions sont les axes sur lesquels on souhaite baser l'analyse des données : la date, la région géographique, le type de produit, etc...

MODÈLE DIMENSIONNEL

Faits

Les faits sont les données que l'on souhaite analyser : On aura des tables de faits pour les ventes (chiffre d'affaire net, quantités et montants commandés, quantités facturées, quantités retournées, volumes des ventes, etc.) par exemple ou sur les stocks (nombre d'exemplaires d'un produit en stock, niveau de remplissage du stock, taux de roulement d'une zone, etc.), ou peut être sur les ressources humaines (performances des employés, nombre de demandes de congés, nombre de démissions, taux de roulement des employés, etc.).

STOCKAGE DE DONNÉES "DIMENSIONNELLES" DANS UNE BD RELATIONNELLE

Différents schéma de stockage :

- Schéma en étoile
- Schéma en flocon
- Schéma en constellation

STOCKAGE DE DONNÉES "DIMENSIONNELLES" DANS UNE BD RELATIONNELLE

Table des faits

La table des faits contient une clef, ainsi que des champs dimension (*foreign keys*) et des champs de mesure. Les champs dimension permettent de relier un fait à ses dimensions, les champs de mesure sont des mesures sur le fait : nombre de vente, etc...

Table des dimensions

Chaque dimension est associée à une table (ou plusieurs dans le cas de dimensions hiérarchiques). Les dimensions contiennent à la fois une clefs, ainsi que des champs descriptifs des dimensions.

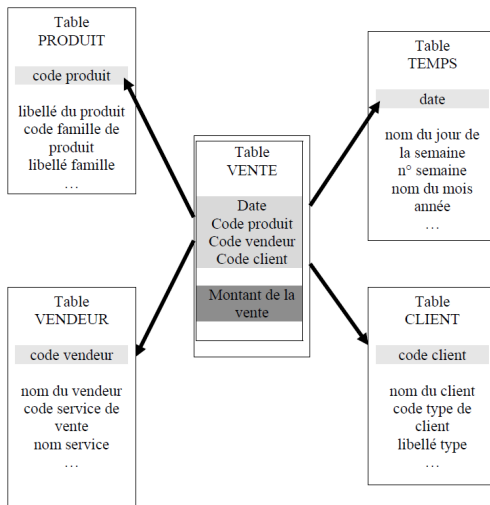
MODÈLE DIMENSIONNEL : ÉTUDE DE CAS

Petit énoncé :

Considérons le cas "classique" d'une société de vente de produit. Elle possède des sources de données sur les produits, les ventes et les clients. On propose de concevoir une ED qui permette de fournir le chiffre d'affaires des ventes d'un produit, par date, client, et vendeur, ainsi que toutes les sommes possibles de chiffre d'affaires.

Dessinez le schéma de BD correspondant

ETUDE DE CAS



DÉFINITION

OLAP

En informatique, et plus particulièrement dans le domaine des bases de données, le traitement analytique en ligne (anglais *online analytical processing*, OLAP) est un type d'application informatique orienté vers l'analyse sur-le-champ d'informations selon plusieurs axes, dans le but d'obtenir des rapports de synthèse tels que ceux utilisés en analyse financière. Les applications de type OLAP sont couramment utilisées en informatique décisionnelle, dans le but d'aider la direction à avoir une vue transversale de l'activité d'une entreprise.

Source : wikipedia

OLAP s'oppose au traitement de transactions en ligne (*online transaction processing abr. OLTP*) qui s'inscrit dans un système opérationnel (en production).

OLAP

- OLAP fournit donc un cadre générique permettant l'analyse des données sur plusieurs dimensions.
- OLAP définit principalement **des opérations génériques** sur les hypercubes permettant l'analyse des données (y compris par des non experts).
- OLAP repose sur **des technologies** permettant le calcul et la mise à jour des hypercubes.
- OLAP définit aussi **un langage de requête** permettant l'interrogation d'un hypercube (langage MDX)

DIFFÉRENTES TECHNOLOGIES

Plusieurs technologies/philosophies sont liées à l'implémentation d'un système OLAP :

- **ROLAP - Relational OLAP** : OLAP sur du relationnel
- **MOLAP - Multidimensional OLAP** : OLAP sur un DW dimensionnel
- **HOLAP - Hybrid OLAP** : Mélange des deux (relationnel si nécessaire)

Mais aussi : Spatial OLAP (SOLAP), Desktop OLAP (DOLAP), ...

TECHNOLOGIES

Nom	Editeur	Technologie
DB2 UDB Server	<i>IBM</i>	ROLAP
Oracle9i	<i>Oracle</i>	ROLAP
SQL Server 2000	<i>Microsoft</i>	ROLAP
DSS	<i>Microstrategy</i>	ROLAP
TeraData	<i>Teradata Corporation</i>	ROLAP massivement parallèle
Informix Metacube	<i>Informix</i>	MOLAP
Essbase	<i>Arbor Software/Hyperion</i>	MOLAP
SAS OLAP Server	<i>SAS</i>	MOLAP
Metacube	<i>Informix</i>	ROLAP
SQL Server	<i>Microsoft</i>	HOLAP
MDDB	<i>SAS Institute</i>	MOLAP/ROLAP
Oracle Express-server	<i>Oracle</i>	MOLAP/ROLAP
DB2 OLAP Server	<i>IBM</i>	MOLAP/ROLAP
Crystal	<i>Seagate Software</i>	Serveur d'application OLAP unique pour tous les déploiements
PowerPlay	<i>Cognos</i>	idem

Source : B. Espinasse

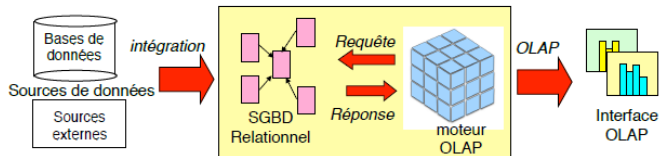
ROLAP

- ROLAP est la technologie la plus utilisée en OLAP car les SGBD relationnels sont très largement répandus
- Cependant les SGBDs relationnels "tout seuls" ne sont pas adaptés à des analyses OLAP \Rightarrow nécessité d'étendre les fonctionnalités d'un SGBD

Un moteur ROLAP :

- Permet de faire les calculs adaptés aux requêtes OLAP sur le SGBD relationnel
- Il permet aussi de générer des requêtes adaptées au schéma de l'entrepôt

ROLAP



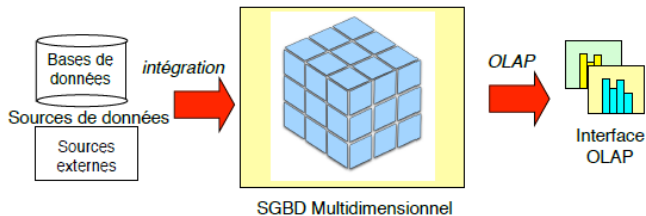
Source : B. Espinasse

MOLAP

La philosophie MOLAP consiste au stockage des données directement dans une structure de cube multidimensionnel :

- MOLAP nécessite le pré-calcul et le stockage des informations du cube,
- mais il permet des extractions très rapides et optimisées.

MOLAP



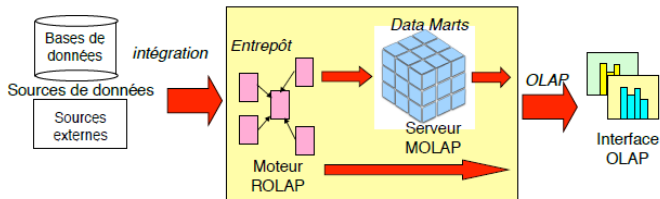
Source : B. Espinasse

HOLAP

Les systèmes HOLAP tente d'exploiter le meilleur des deux mondes :

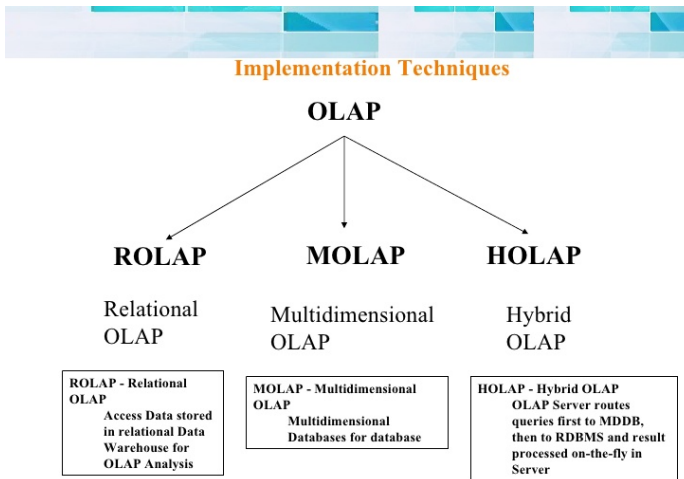
- La structure du moteur SGBD pour le stockage des données détaillées
- Un système de type MOLAP comme structure de données pour un certain nombre de requêtes (données agrégées)

HOLAP



Source : B. Espinasse

HOLAP



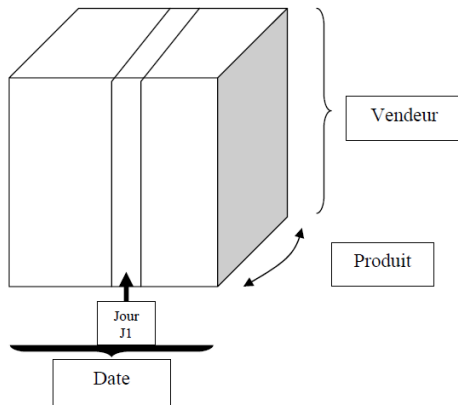
HOLAP

	Data storage	Aggregations storage	Query performance	Latency
MOLAP	Cube	Cube	High	High
HOLAP	Relational database	Cube	Medium	Low (none)
ROLAP	Relational database	Relational database	Low	Low (none)

OPÉRATIONS GÉNÉRIQUES

IDÉE GÉNÉRALE

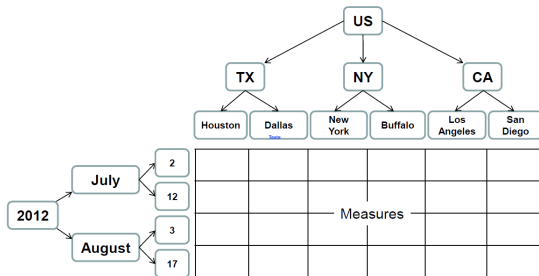
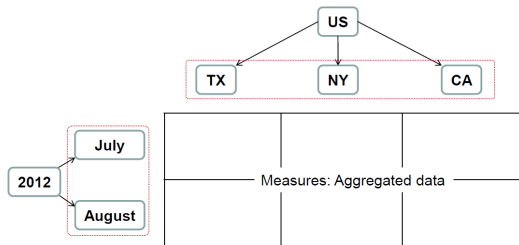
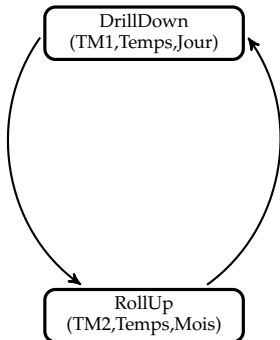
A partir du schéma en étoile précédent, il est possible de construire des tableaux multi-dimensionnels (ou hypercubes) permettant l'analyse des données du DW.



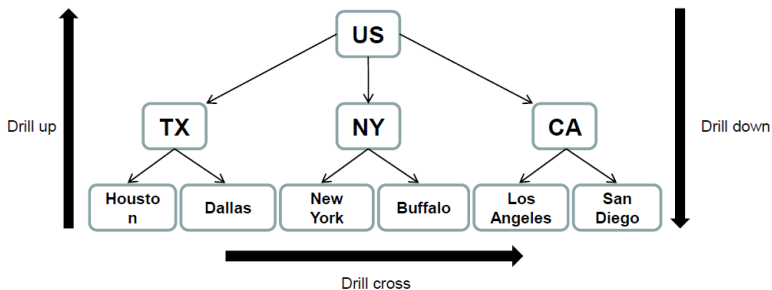
OPÉRATIONS DE MANIPULATION MULTIDIMENSIONNELLES

Catégorie		Opérateur	Description
Paramétrage d'une TM	Forage	DrillDown	Forage vers le bas
		RollUp	Forage vers le haut
	Rotation	Frotate	Rotation des faits
		Drotate	Rotation des dimensions
		Hrotate	Rotation des hiérarchies d'une dimension
Présentation d'une TM	Sélection	Select	Sélection des valeurs des paramètres ou mesures
		UnSelect	Désélection
	Permutation des paramètres	Switch	Permutation de deux valeurs d'un même paramètre
		Order	Ordonnancement de valeurs de paramètres
		Nest	Permutation de deux paramètres
Transformation d'une TM	Calcul	Cube	Permet de spécifier une fonction d'agrégation
		UnCube	Permet de supprimer une fonction d'agrégation
	Modification du fait	DelM	Suppression d'une mesure
		AddM	Ajout d'une mesure
		Push	Transformation d'un paramètre en mesure
	Modification dimension	Pull	Transformation d'une mesure en paramètre

OPÉRATIONS DRILLDOWN ET ROLLUP



OPÉRATIONS - DRILL(S)

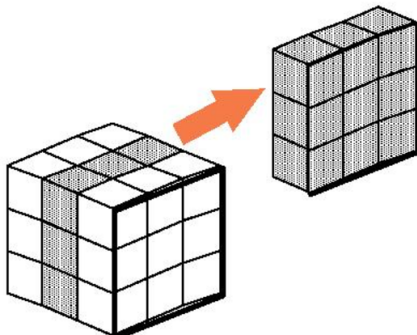


OPÉRATIONS - SLICE/SELECTION

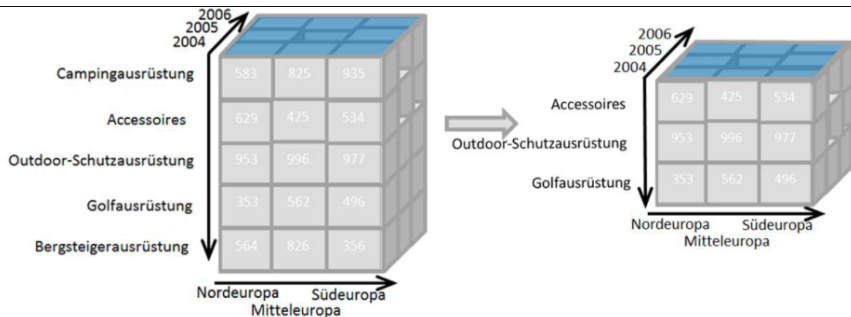
		1995	1996	1997
Frais	IdF	220	265	284
	Province	225	245	240
Liquide	IdF	163	152	145
	Province	187	174	184



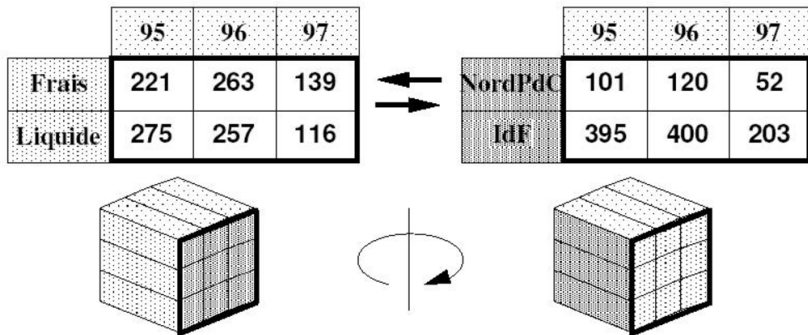
		1996
Frais	IdF	265
	Province	245
Liquide	IdF	152
	Province	174



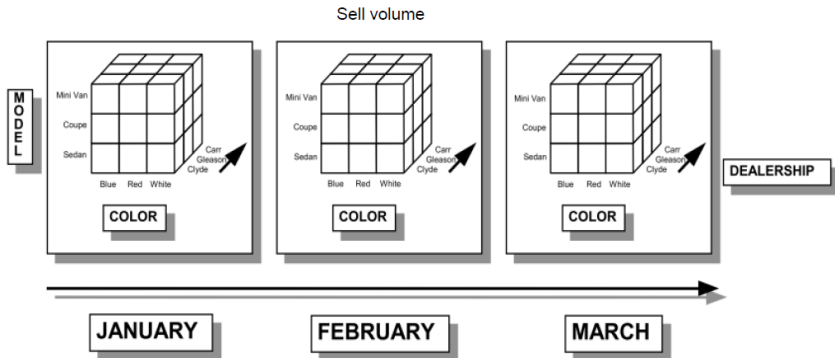
OPÉRATIONS - DICE / SÉLECTION D'UN CUBE



OPÉRATIONS - ROTATION



OPÉRATIONS - DELM/ADDM : AJOUT ET SUPPRESSION DE DIMENSION



MONDRIAN

Nous allons nous intéresser concrètement au système ROLAP Mondrian :

- Serveur OLAP écrit en JAVA
- Supporte le langage MDX
- Supporte l'interface *olap4j* permettant l'utilisation de plusieurs technologies OLAP en JAVA



Attention : ceci est un exemple illustratif....

MONDRIAN

Mondrian est utilisé pour :

- L'analyse interactive haute performance de grands ou de petits volumes d'informations.
- L'exploration de données multi-dimensionnelles, par exemple l'analyse des ventes par ligne de produits, par région, par période de temps
- Le calcul avancé en utilisant les expressions de calcul du langage MDX
- La transformation de requêtes MDX en Structured Query Language (SQL) pour récupérer des réponses aux requêtes dimensionnelles
- La formulation de requêtes à grande vitesse grâce à l'utilisation des tableaux agrégés stockés dans le SGBDR

MONDRIAN

Le schéma OLAP est écrit en XML. Ce XML permet la description de :

- La base physique (relationnelle) sur laquelle seront basés les calculs
- La définition sur cette base des **table de faits** et **tables de dimensions**
- La définition sur cette base des **mesures**
- La définition de tables agrégées

MONDRIAN

```

<Schema>
  <Cube name="Sales">
    <Table name="sales_fact_1997"/>
    <Dimension name="Gender" foreignKey="customer_id">
      <Hierarchy hasAll="true" allMemberName="All Genders" primaryKey="customer_id">
        <Table name="customer"/>
        <Level name="Gender" column="gender" uniqueMembers="true"/>
      </Hierarchy>
    </Dimension>
    <Dimension name="Time" foreignKey="time_id">
      <Hierarchy hasAll="false" primaryKey="time_id">
        <Table name="time_by_day"/>
        <Level name="Year" column="the_year" type="Numeric" uniqueMembers="true"/>
        <Level name="Quarter" column="quarter" uniqueMembers="false"/>
        <Level name="Month" column="month_of_year" type="Numeric" uniqueMembers="false"/>
      </Hierarchy>
    </Dimension>
    <Measure name="Unit Sales" column="unit_sales" aggregator="sum"
      formatString="#,###"/>
    <Measure name="Store Sales" column="store_sales" aggregator="sum"
      formatString="#,###.##"/>
    <Measure name="Store Cost" column="store_cost" aggregator="sum"
      formatString="#,###.00"/>
    <CalculatedMember name="Profit" dimension="Measures"
      formula="[Measures].[Store Sales] - [Measures].[Store Cost]">
      <CalculatedMemberProperty name="FORMAT_STRING" value="$#,##0.00"/>
    </CalculatedMember>
  </Cube>
</Schema>

```


MONDRIAN

Le schema physique donne des informations sur la structuration de la BD :

```
<Schema>
<PhysicalSchema>
  <Table name="sales_fact_1997"/>
  <Table name="customer"/>
  <Table name="time_by_day"/>
  ...
</PhysicalSchema>
<Cube>
  ...
</Cube>
</Schema>
```

MONDRIAN

Définition du cube et du fait

```
<Schema>
```

```
....
```

```
  <Cube name="Sales">
```

```
    <Table name="sales_fact_1997"/>
```

```
    ....
```

```
  </Cube>
```

```
</Schema>
```

MONDRIAN

Définition des mesures

```
<Measure name="Unit Sales" column="unit_sales" aggregator="sum"
  formatString="#,###"/>
<Measure name="Store Sales" column="store_sales" aggregator="sum"
  formatString="#,###.##"/>
<Measure name="Store Cost" column="store_cost" aggregator="sum"
  formatString="#,###.00"/>
<CalculatedMember name="Profit" dimension="Measures"
  formula="[Measures].[Store Sales] - [Measures].[Store Cost]">
  <CalculatedMemberProperty name="FORMAT_STRING" value="$#,##0.00"/>
</CalculatedMember>
```

MONDRIAN

Définition des dimensions (membres, hiérarchie, niveau) (1/2)

```
<Dimension name="Gender" foreignKey="customer_id">
  <Hierarchy hasAll="true" allMemberName="All Genders" primaryKey="customer_id"
    <Table name="customer"/>
    <Level name="Gender" column="gender" uniqueMembers="true"/>
  </Hierarchy>
</Dimension>
<Dimension name="Time" foreignKey="time_id">
  <Hierarchy hasAll="false" primaryKey="time_id">
    <Table name="time_by_day"/>
    <Level name="Year" column="the_year" type="Numeric" uniqueMembers="true"/>
    <Level name="Quarter" column="quarter" uniqueMembers="false"/>
    <Level name="Month" column="month_of_year" type="Numeric" uniqueMembers="f
  </Hierarchy>
</Dimension>
```

MONDRIAN

Définition des dimensions (2/2)

```
<Dimension name='Promotion' table='promotion' key='Promotion Id'>
  <Attributes>
    <Attribute name='Promotion Id' keyColumn='promotion_id' hasHierarchy='false' />
    <Attribute name='Promotion Name' keyColumn='promotion_name'
      hasHierarchy='false' />
    <Attribute name='Media Type' keyColumn='media_type'
      hierarchyAllMemberName='All Media' hasHierarchy='false' />
  </Attributes>
  <Hierarchies>
    <Hierarchy name='Media Type' allMemberName='All Media'>
      <Level attribute='Media Type' />
    </Hierarchy>
    <Hierarchy name='Promotions' allMemberName='AllPromo'>
      <Level attribute='Promotion Name' />
    </Hierarchy>
  </Hierarchies>
</Dimension>
```

MONDRIAN

Plusieurs outils permettant de réaliser ces hypercubes sont disponibles :

- Eclipse
- Pentaho Cube Designer
- Mondrian Workbench

MDX

Définition

Le MDX (de l'anglais Multidimensional Expressions, « expressions multidimensionnelles ») est un langage de requête pour les bases de données OLAP, analogue au rôle de SQL pour les bases de données relationnelles. C'est aussi un langage de calcul avec une syntaxe similaire à celle des tableurs.

Le langage des expressions multidimensionnelles possède une syntaxe appropriée à l'interrogation et manipulation des données multidimensionnelles mémorisées dans un cube OLAP¹. Bien qu'il soit possible de traduire certaines expressions dans le langage SQL traditionnel, cela nécessite une syntaxe SQL souvent maladroite même pour des expressions MDX très simples. MDX a été adopté par une large majorité de fournisseur de la technologie OLAP et est devenu un standard de facto pour les systèmes OLAP.

Source : wikipedia

MDX - EXEMPLE

```
SELECT
    { [Measures].[Store Sales] } ON COLUMNS,
    { [Date].[2002], [Date].[2003] } ON ROWS
FROM Sales
WHERE ( [Store].[USA].[CA] )
```


MDX - EXEMPLE

```
SELECT
    { [Measures].[Store Sales] } ON COLUMNS,
    { [Date].[2002], [Date].[2003] } ON ROWS
FROM Sales
WHERE ( [Store].[USA].[CA] )
```

- **Sales** est le cube sur lequel la requête est faite

MDX - EXEMPLE

```
SELECT
    { [Measures].[Store Sales] } ON COLUMNS,
    { [Date].[2002], [Date].[2003] } ON ROWS
FROM Sales
WHERE ( [Store].[USA].[CA] )
```

- **Sales** est le cube sur lequel la requête est faite
- **[Measures].[Store Sales]** et **[Date].[2002]**, **[Date].[2003]** sont les dimensions conservées
- **[Store].[USA].[CA]** est le "slicer"

MDX

```
SELECT Measures.[Unit Sales] ON COLUMNS,  
[Store].[All Stores] ON ROWS  
FROM [Sales]
```

Délimiteurs

Les délimiteurs [et] peuvent ne pas être mis si pas d'ambiguïté.

MDX

```
SELECT Measures.MEMBERS ON COLUMNS,  
Product.Style.CHILDREN ON ROWS  
FROM [Adventure Works]
```

équivalent à :

```
SELECT [Measures].MEMBERS ON COLUMNS,  
[Product].[Style].CHILDREN ON ROWS  
FROM [Adventure Works]
```

MDX

```
SELECT [Measures].MEMBERS ON COLUMNS,  
[Date].[Calendar Year].MEMBERS ON ROWS  
FROM  
(SELECT [Measures].[Internet Sales Amount] ON COLUMNS,  
[Date].[Calendar Year].[2004] ON ROWS  
FROM [Adventure Works])
```

Expressions de sous-cube

Attention : ne marche pas dans tous les systèmes OLAP

MDX

```
SELECT Measures.MEMBERS ON COLUMNS,  
[Store].MEMBERS ON ROWS  
FROM [Sales]
```

MEMBERS

Retourne le jeu des membres d'une dimension, d'un niveau ou d'une hiérarchie.

MDX

```
SELECT Measures.MEMBERS ON COLUMNS,  
{[Store].[Store State].[CA], [Store].[Store State].[WA]}  
  ON ROWS  
FROM [Sales]
```

Exercice

Dessinez la sortie de cette requête ?

MDX

```
SELECT Measures.MEMBERS ON COLUMNS,  
{[Store].[Store State].[CA].CHILDREN,  
[Store].[Store State].[WA].CHILDREN}  
  ON ROWS  
FROM [Sales]
```

CHILDREN

Retourne le jeu des enfants d'un membre spécifié.

MDX

```
SELECT Measures.MEMBERS ON COLUMNS,  
{ [Store].[Store State].[CA],  
  DESCENDANTS([Store].[Store State].[CA], [Store City]) }  
  ON ROWS  
FROM [Sales]
```

DESCENDANTS

```
DESCENDANTS(member, level [, flags])
```

Retourne le jeu de descendants d'un membre à un niveau spécifié ou à une distance spécifiée, en incluant ou en excluant éventuellement des descendants dans d'autres niveaux.

MDX

DESCENDANTS

Le "flag" peut être BEFORE, AFTER, SELF ou bien BEFORE_AND_AFTER selon le niveau désiré

```
SELECT Measures.MEMBERS ON COLUMNS,  
  {[Store].[Store State].[CA],  
  DESCENDANTS([Store].[Store State].[CA], [Store City],AFTER)}  
  ON ROWS  
FROM [Sales]
```

```
SELECT Measures.MEMBERS ON COLUMNS,  
  {[Store].[Store State].[CA],  
  DESCENDANTS([Store].[Store State], [Store City]  
  ,BEFORE_AND_AFTER)} ON ROWS FROM [Sales]
```

MDX

```
select AddCalculatedMembers([Measures].Members) ON COLUMNS,  
      {[Store].[USA].[CA],  
Descendants([Store].[USA].[CA], [Store].[Store City])}  
ON ROWS  
from [Sales]
```

AddCalculatedMembers

Retourne un ensemble généré par l'ajout de membres calculés à un ensemble spécifié.

MDX

```
SELECT {[Store Type].[Store Type].MEMBERS} ON COLUMNS,  
        {[Store].[Store State].MEMBERS} ON ROWS  
FROM [Sales]  
WHERE (Measures.[Unit Sales])
```

Slicer

WHERE spécifie une "tranche dimensionnelle" dans le cube

MDX

```
SELECT {[Store Type].[Store Type].MEMBERS} ON COLUMNS,  
        {[Store].[Store State].MEMBERS} ON ROWS  
FROM [Sales]  
WHERE (Measures.[Unit Sales], [Time].[Year].[1997])
```

Slicer

WHERE spécifie une "tranche dimensionnelle" dans le cube



WITH SET

WITH SET permet la création d'ensembles

```
WITH SET [ChardonnayChablis] AS
    {[Product].[All Products].[Drink].[Good Chardonnay],
    [Product].[All Products].[Drink].[Pearl Chardonnay],
    [Product].[All Products].[Drink].[Portsmouth],
    [Product].[All Products].[Drink].[Walrus Chardonnay],
SELECT
    [ChardonnayChablis] ON COLUMNS,
    {Measures.[Unit Sales]} ON ROWS
FROM Sales
```

CALCULATED MEMBERS

WITH MEMBER

MDX permet de rajouter des calculs directement dans les requêtes MDX. Ceci s'effectue grâce à la syntaxe :

```
WITH MEMBER parent.name AS 'expression'
```

```
WITH MEMBER Measures.ProfitPercent AS  
'(Measures.[Store Sales] - Measures.[Store Cost]) /  
(Measures.[Store Cost])', FORMAT_STRING = '#.00%'
```



```
WITH
    MEMBER [Measures].[Special Discount] AS
        [Measures].[Discount Amount] * 1.5
SELECT
    [Measures].[Special Discount] on COLUMNS,
    NON EMPTY [Product].[Product].MEMBERS ON Rows
FROM [Adventure Works]
WHERE [Product].[Category].[Bikes]
```


CALCULATED MEMBERS

WITH MEMBER

MDX permet de rajouter des calculs directement dans les requêtes MDX. Ceci s'effectue grâce à la syntaxe :

```
WITH MEMBER parent.name AS 'expression'
```

```
WITH MEMBER [Time].[First Half 97] AS  
'[Time].[1997].[Q1] + [Time].[1997].[Q2]'  
MEMBER [Time].[Second Half 97] AS  
'[Time].[1997].[Q3] + [Time].[1997].[Q4]'
```

CALCULATED MEMBERS

```
WITH MEMBER [Time].[First Half 97] AS  
'[Time].[1997].[Q1] + [Time].[1997].[Q2]'  
MEMBER [Time].[Second Half 97] AS  
'[Time].[1997].[Q3] + [Time].[1997].[Q4]'  
SELECT {[Time].[First Half 97], [Time].[Second Half 97],  
[Time].[1997].CHILDREN} ON COLUMNS,  
{[Store].[Store Name].MEMBERS} ON ROWS  
FROM [Sales]
```



```
WITH MEMBER measures.X AS  
    [Product].children.count  
SELECT Measures.X ON 0  
FROM [Sales]
```

COUNT

Retourne le nombre de cellules d'un ensemble



```
SELECT Measures.MEMBERS ON COLUMNS,  
TOPPERCENT({[Store].[Store City].MEMBERS}, 50,  
Measures.[Sales Count]) ON ROWS  
FROM [Sales]
```

TOPPERCENT

Trie un jeu en ordre décroissant et retourne un jeu de tuples avec les valeurs les plus élevées dont le total cumulé est égal ou supérieur à un pourcentage spécifié.



```
WITH MEMBER Measures.[Maximum Sales] AS
'MAX(DESCENDANTS([Time].CURRENTMEMBER, [Time].[Month]),
Measures.[Unit Sales])'
SELECT {[Time].[1997]} ON COLUMNS,
[Product].[Product Category].MEMBERS ON ROWS
FROM [Sales]
WHERE (Measures.[Maximum Sales])
```

Autres mesures

Les fonctions disponibles sont AVG, MEDIAN, MAX, MIN, VAR, and STDDEV

REFERENCES

- MDX Reference @ MSDN :
<http://msdn2.microsoft.com/en-us/library/ms145506.aspx>
- The Baker's Dozen : 13 Tips for Querying OLAP Databases with MDX :
<http://www.devx.com/codemag/Article/37460/1954?pf=true>
- Tutorial : Introduction to Multidimensional Expression (MDX).
http://www.fing.edu.uy/inco/grupos/csi/esp/Cursos/cursos_act/2005/DAP_SistDW/Material/2-SDW-Laboratoriol-2005.pdf
- MDX resources : <http://www.mosha.com/msolap/mdx.htm>
- <http://www-igm.univ-mlv.fr/~dr/XPOSE2005/entrepot/multidim.html>