

Le MDX



“

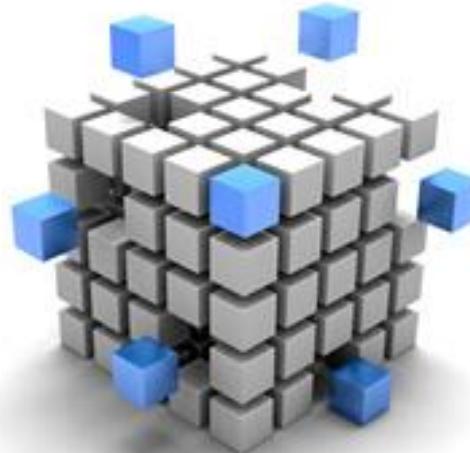
Decouvrir le MDX

Objectifs

3

Présentation

- Définir le MDX
- Les éléments du language
- Les fonctions MDX
- La gestion des erreurs



4.

Le MDX

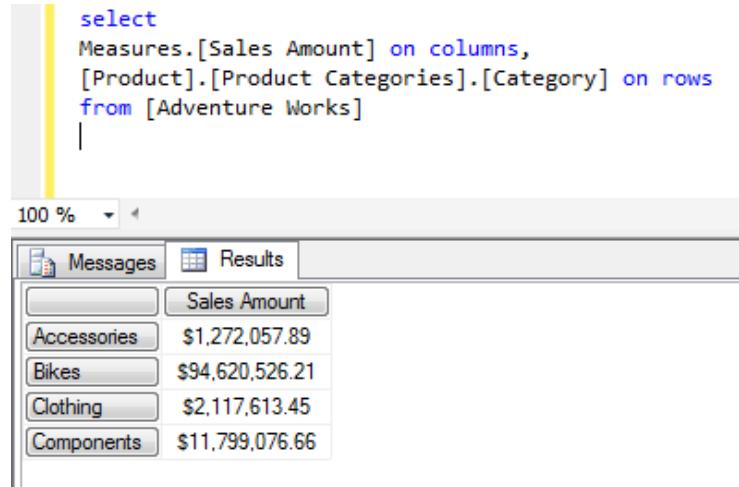
Découvrons le MDX ensemble??!!!!

Définir Le MDX

5

Un langage ou quoi?

- ❑ Le MDX est un langage pour requêter et manipuler des données stockées dans des cubes OLAP
- ❑ Le concept est +- similaire à SQL
SQL → Base de données relationnelle
MDX → Base de données multidimensionnelle



The screenshot shows a Microsoft Analysis Services query interface. At the top, there is a code editor window containing an MDX query:

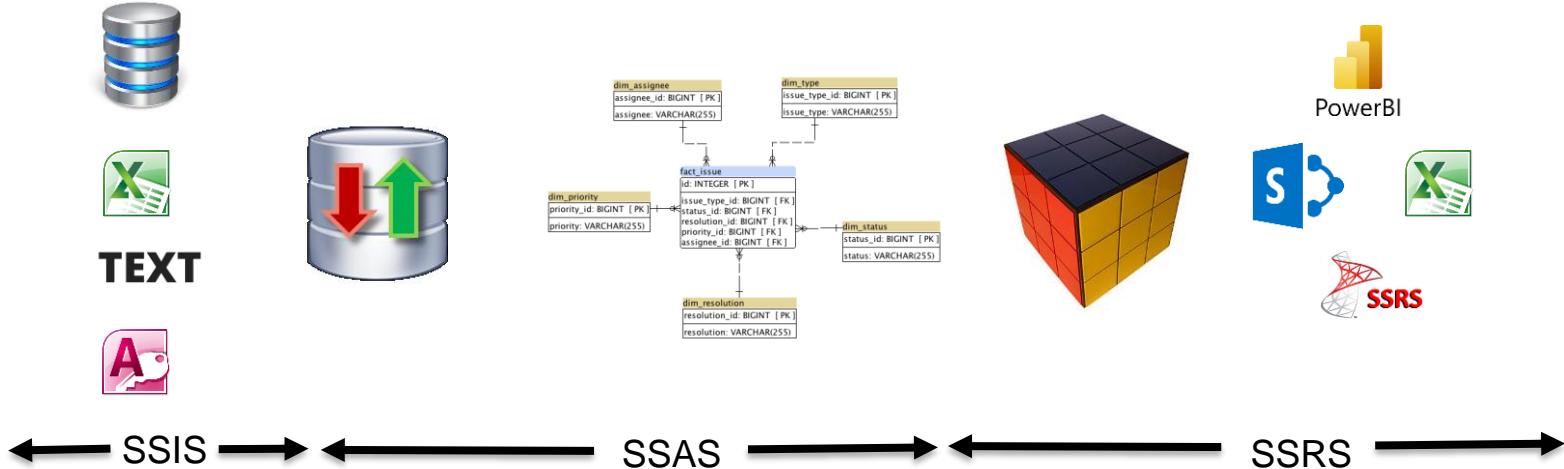
```
select
Measures.[Sales Amount] on columns,
[Product].[Product Categories].[Category] on rows
from [Adventure Works]
```

Below the code editor is a results pane. It has two tabs: "Messages" and "Results". The "Results" tab is selected and displays a table with four rows of data:

	Sales Amount
Accessories	\$1,272,057.89
Bikes	\$94,620,526.21
Clothing	\$2,117,613.45
Components	\$11,799,076.66

Définir le MDX

6



Définir le MDX

7

MDX vs SQL

❑ Select From :

SQL → Table

MDX → Cube

❑ Données:

SQL → (1,N) Colonnes + (0,N) Lignes

MDX → (0,127) Axes + (0,N) données

❑ Where:

SQL → Filtre

MDX → Trancheur

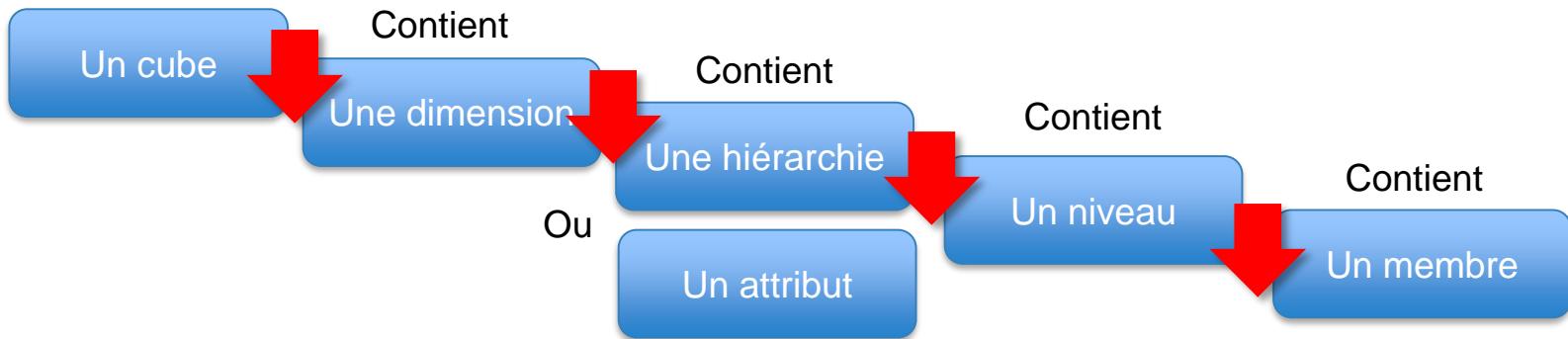
Exemple de requête MDX avec deux axes et deux tranchesurs

```
SELECT  
Measures.AllMembers ON COLUMNS,  
[Date].[Calendar].&[2003].Children ON  
ROWS  
FROM [Cube]  
WHERE  
(  
[Customer].[Customer  
Geography].[France],  
[Product].[Product Categories].[Bikes]  
)
```

Définir le MDX

8

Pour bien raisonner

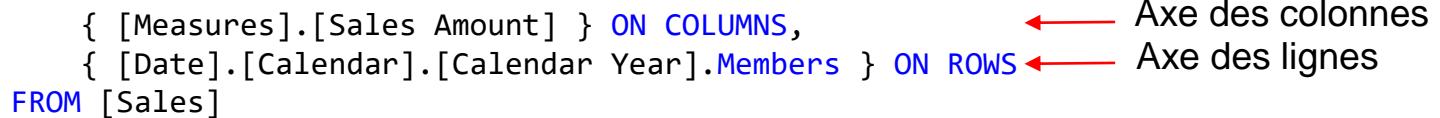


Les axes dans MDX

- Les axes dans MDX définissent le plan de représentation pour les données
- Dans l'instruction SELECT, nous spécifions les données sur les axes du résultat

SELECT

```
{ [Measures].[Sales Amount] } ON COLUMNS,  
{ [Date].[Calendar].[Calendar Year].Members } ON ROWS  
FROM [Sales]
```



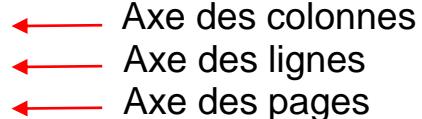
A red arrow points from the text "ON COLUMNS" to the column axis, which is labeled "Axe des colonnes". Another red arrow points from the text "ON ROWS" to the row axis, which is labeled "Axe des lignes".

Les éléments MDX

10

Les axes dans MDX

```
SELECT
    { [Measures].[Sales Amount] } ON COLUMNS,
    { [Date].[Calendar].[Calendar Year].Members } ON ROWS,
    { [Geography].[Country].&[France] } ON PAGES
FROM [Sales]
```



Axe des colonnes
Axe des lignes
Axe des pages

```
SELECT
    { [Measures].[Sales Amount] } ON COLUMNS,
    { [Date].[Calendar].[Calendar Year].Members } ON ROWS
FROM [Sales]
WHERE { [Geography].[Country].&[France] }
```

Les éléments MDX

11

Exemple de requête

```
SELECT  
[Measures].[Sales Amount] ON COLUMNS,  
[Dim Sales Territory].[Sales Territory  
Hierarchy].[Sales Territory Country] ON ROWS  
FROM  
[InternetSalesCube]
```

Messages		Results
		Sales Amount
France		2644017.71430033
Germany		2894312.33820041
United Kingdom		3391712.21090071
Canada		1977844.86209997
United States		9389789.51080357
Australia		9061000.58440184

Les éléments MDX

12

Exemple de requête avec hiérarchie

```
SELECT
{
    [Measures].[Sales Amount], [Measures].[Total Product Cost]
} ON COLUMNS,
{
    [Dim Sales Territory].[Sales Territory Hierarchy].[Sales Territory Country]*
    [Dim Product].[Product Hierarchy].[Product Subcategory Key]
} ON ROWS
FROM
[InternetSalesCube]
```

		Sales Amount	Total Product Cost
United States	Mountain Bikes	3417457.73500016	1866010.18100003
United States	Road Bikes	4289925.89580032	2646951.87159995
United States	Touring Bikes	1292475.9	803403.025400004
United States	Handlebars	(null)	(null)
United States	Bottom Brackets	(null)	(null)
United States	Brakes	(null)	(null)
United States	Chains	(null)	(null)

Les références de membres

- ❑ Une référence de membre est tout simplement une succession hiérarchique membre dans la collection des valeurs d'un attribut
- ❑ Exemple

[Date].[Annee].[Annee].[2017]

ou

[Date].[Annee].[Annee].&[2017]

SELECT

```
{[Measures].[Sales Amount] } ON COLUMNS,  
{[DimProduct].[Product Name] } ON ROWS  
FROM [Sales]  
WHERE ( [Date].[Annee].[Annee].&[2017] )
```

Les SET inline {...}

□ La SET est une liste de membres

- La set en dessous est dite inline
- Définie directement dans la requête
- Portée locale
- N'a aucun nom réutilisable

```
SELECT
{
[Dim Employee].[Hierarchy].[Gender],
[Dim Employee].[Hierarchy].[Title],
[Dim Employee].[Hierarchy].[FullName]
} ON 0

FROM [DSV]
```

Les SET dynamiques {...}

La set en dessous est une set calculée implicitement

- Générée par la fonction Members
- Toujours sans nom
- Résolue dynamiquement par SSAS

```
SELECT
{
[Dim Employee].[Hierarchy].members
} ON 0

FROM [DSV]
```

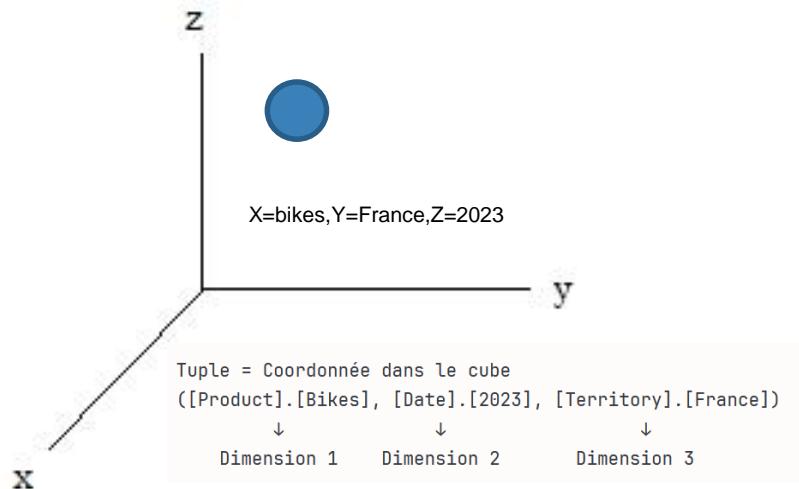
Les SET nommées {...}

- ❑ Les SET nommées permettent de prédéfinir les groupes dynamiques des membres à afficher sur des colonnes ou des lignes.
 - Réutilisable
 - Lisible
 - Maintenable

```
WITH
SET [Employee Attributes] AS
{
    [Dim Employee].[Hierarchy].[Gender],
    [Dim Employee].[Hierarchy].[Title],
    [Dim Employee].[Hierarchy].[FullName]
}
SELECT
    [Employee Attributes] ON 0
FROM [DSV]
```

Les Tuples (...)

- ❑ Chaque cellule montre des données pour un membre particulier de chaque hiérarchie
- ❑ Un Tuple définit quel membre devrait être utilisé de chacune des hiérarchies
- ❑ Un Tuple peut contenir des membres issus de plusieurs dimensions à la fois



Les Tuples (...)

- Voici un exemple de Tuple

```
SELECT
{
  (
    [Dim Product].[Hierarchy].[Product Name],
    [Date].[Calendar].[Calendar Year].&[2023]
  )
} ON 0
FROM [DSV]
```

Les éléments MDX

18

Les membres calculés

- ❑ MDX offre le moyen de créer et d'interroger des membres calculés dans le jeu de résultats
- ❑ **WITH MEMBER** précède l'instruction **SELECT**
- ❑ Special Period est un **membre artificial**
- ❑ Special Period est un **membre artificial**, il regroupe 2022 + 2023

```
WITH
MEMBER
    [Date].[Calendar].[Special Period]
AS
    Aggregate(
        {
            [Date].[Calendar].[Calendar Year].&[2022],
            [Date].[Calendar].[Calendar Year].&[2023]
        }
    )
SELECT
    { [Measures].[Sales Amount] } ON COLUMNS,
    {
        [Date].[Calendar].[Calendar Year].Members,
        [Date].[Calendar].[Special Period]
    } ON ROWS
FROM [Sales]
```

Les éléments MDX

19

Les membres calculés

- ❑ Profit Margin est une **mesure calculée à la volée**
- ❑ Il n'apparaît qu'aux colonnes

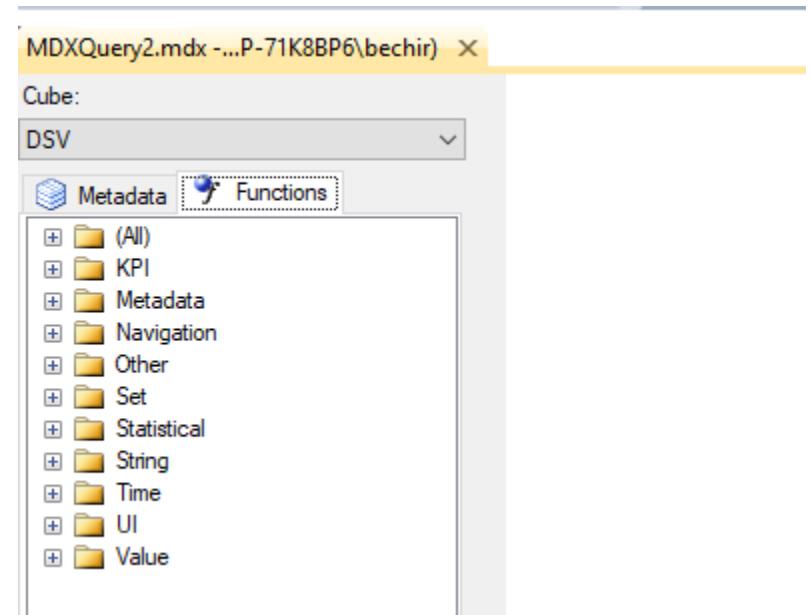
```
WITH
MEMBER
    [Measures].[Profit Margin]
AS
    IIF(
        [Measures].[Sales Amount] = 0,
        NULL,
        [Measures].[Net Profit] / [Measures].[Sales Amount]
    )
SELECT
{
    [Measures].[Sales Amount],
    [Measures].[Net Profit],
    [Measures].[Profit Margin]
} ON COLUMNS,
{ [Date].[Calendar].[Calendar Year].Members } ON ROWS
FROM [Sales]
```

Les éléments MDX

20

Les fonctions

- ❑ Elles représentent un ensemble d'instructions qui permettent de retourner un élément MDX
- ❑ Les fonctions sont partagées sur 9 catégories
- ❑ Chaque catégorie comprend un ensemble de fonctions qui peuvent être utilisées dans ce contexte



Les fonctions

Récapitulatif des 9 catégories :

1. KPI - 6 fonctions
2. Metadata - 18 fonctions
3. Navigation - 18 fonctions
4. Set - 24 fonctions
5. Statistical - 21 fonctions
6. String - 24 fonctions
7. Time - 20 fonctions
8. Value - 25 fonctions
9. Other - 31 fonctions

Les fonctions KPI

1. `KPIValue()` : Retourne la valeur actuelle du KPI
2. `KPIGoal()` : Retourne l'objectif cible du KPI
3. `KPIStatus()` : Retourne le statut du KPI (-1, 0, 1)
4. `KPITrend()` : Retourne la tendance d'évolution du KPI
5. `KPIWeight()` : Retourne le poids/importance du KPI
6. `KPICurrentTimeMember()` : Retourne le membre temporel actuel du KPI

Les fonctions Metadata

1. Members : Retourne tous les membres d'une dimension, hiérarchie ou niveau
2. AllMembers : Retourne tous les membres incluant les membres calculés
3. Dimensions() : Retourne la dimension à l'index spécifié
4. Hierarchy() : Retourne la hiérarchie d'un membre
5. Level() : Retourne le niveau d'un membre
6. Levels() : Retourne le niveau à l'index spécifié dans une hiérarchie
7. CurrentMember : Retourne le membre actuel dans une dimension/hiérarchie
8. DefaultMember : Retourne le membre par défaut d'une dimension/hiérarchie
9. Item() : Retourne un membre spécifique d'un tuple ou d'un set

Les fonctions Metadata

9. UnknownMember : Retourne le membre inconnu d'une dimension
10. MemberValue : Retourne la valeur d'un membre
11. Name : Retourne le nom d'un membre
12. UniqueName : Retourne le nom unique d'un membre
13. Caption : Retourne le libellé d'affichage d'un membre
14. Ordinal : Retourne la position ordinaire d'un niveau
15. Count : Retourne le nombre d'éléments dans un set ou tuple
16. Dimension : Retourne la dimension d'un membre ou hiérarchie
17. Properties() : Retourne la propriété d'un membre

Les fonctions Navigation

1. Parent : Retourne le membre parent d'un membre
2. Children : Retourne les enfants d'un membre
3. FirstChild : Retourne le premier enfant d'un membre
4. LastChild : Retourne le dernier enfant d'un membre
5. Descendants : Retourne les descendants d'un membre à un niveau spécifié
6. Ancestor : Retourne l'ancêtre d'un membre à un niveau spécifié
7. Ancestors : Retourne tous les ancêtres d'un membre
8. Siblings : Retourne les frères/sœurs d'un membre (même parent)
9. FirstSibling : Retourne le premier frère/sœur d'un membre
10. LastSibling : Retourne le dernier frère/sœur d'un membre
11. Cousin : Retourne le membre cousin à la même position relative

Les fonctions Naviguation

- 13. PrevMember : Retourne le membre précédent au même niveau
- 14. NextMember : Retourne le membre suivant au même niveau
- 15. Lag() : Retourne le membre n positions avant le membre actuel
- 16. Lead() : Retourne le membre n positions après le membre actuel
- 17. Ascendants : Retourne l'ensemble des descendants d'un membre
- 18. DrilldownLevel : Descend d'un niveau dans la hiérarchie
- 19. DrillupLevel : Remonte d'un niveau dans la hiérarchie

Les fonctions Set

1. CrossJoin() : Produit cartésien de deux ou plusieurs sets
2. Union() : Combine deux sets en éliminant ou conservant les doublons
3. Intersect() : Retourne l'intersection de deux sets
4. Except() : Retourne les éléments du premier set absents du second
5. Distinct() : Élimine les doublons d'un set
6. Filter() : Filtre un set selon une condition
7. TopCount() : Retourne les N premiers éléments d'un set
8. BottomCount() : Retourne les N derniers éléments d'un set
9. TopPercent() : Retourne les éléments représentant un pourcentage du total
10. BottomPercent() : Retourne les éléments du bas représentant un pourcentage
11. TopSum() : Retourne les éléments dont le cumul atteint une somme
12. BottomSum() : Retourne les éléments du bas dont le cumul atteint une somme
13. Order() : Trie un set selon une expression

Les fonctions Set

14. Hierarchize() : Organise un set selon l'ordre hiérarchique
15. Head() : Retourne les N premiers éléments d'un set
16. Tail() : Retourne les N derniers éléments d'un set
17. Subset() : Retourne un sous-ensemble à partir d'une position
18. Generate() : Génère un set en appliquant un set à chaque membre
19. Extract() : Extrait des éléments d'un set selon des dimensions
20. NonEmpty() : Retourne les membres non vides d'un set
21. NonEmptyCrossjoin() : CrossJoin optimisé éliminant les cellules vides
22. Exists() : Retourne les membres existant dans un autre set
23. DrilldownMember() : Développe les membres spécifiés dans un set
24. ToggleDrillState() : Bascule l'état développé/réduit des membres

Les fonctions Set

14. Hierarchize() : Organise un set selon l'ordre hiérarchique
15. Head() : Retourne les N premiers éléments d'un set
16. Tail() : Retourne les N derniers éléments d'un set
17. Subset() : Retourne un sous-ensemble à partir d'une position
18. Generate() : Génère un set en appliquant un set à chaque membre
19. Extract() : Extrait des éléments d'un set selon des dimensions
20. NonEmpty() : Retourne les membres non vides d'un set
21. NonEmptyCrossjoin() : CrossJoin optimisé éliminant les cellules vides
22. Exists() : Retourne les membres existant dans un autre set
23. DrilldownMember() : Développe les membres spécifiés dans un set
24. ToggleDrillState() : Bascule l'état développé/réduit des membres

Les fonctions Statistical

- 1 . Sum() : Calcule la somme d'une expression sur un set
- 2 . Avg() : Calcule la moyenne d'une expression sur un set
- 3 . Min() : Retourne la valeur minimale d'une expression sur un set
- 4 . Max() : Retourne la valeur maximale d'une expression sur un set
- 5 . Count() : Compte le nombre d'éléments dans un set
- 6 . DistinctCount() : Compte le nombre d'éléments distincts
- 7 . Median() : Calcule la médiane d'une expression numérique
- 8 . Var() : Calcule la variance d'une expression (population)
- 9 . VarP() : Calcule la variance d'une expression (échantillon)
- 10 . StDev() : Calcule l'écart-type d'une expression (population)
11. StDevP() : Calcule l'écart-type d'une expression (échantillon)

Les fonctions Statistical

- 12 . Correlation() : Calcule la corrélation entre deux expressions
- 13 . Covariance() : Calcule la covariance entre deux expressions
- 14 . CovarianceN() : Calcule la covariance normalisée
- 15 . LinRegIntercept() : Retourne l'ordonnée à l'origine d'une régression linéaire
- 16 . LinRegSlope() : Retourne la pente d'une régression linéaire
- 17 . LinRegR2() : Retourne le coefficient R² d'une régression linéaire
- 18 . LinRegPoint() : Calcule un point sur la ligne de régression
- 19 . LinRegVariance() : Retourne la variance de la régression linéaire
- 20 . Rank() : Retourne le rang d'un tuple dans un set
- 21 . Percentile() : Calcule le percentile d'une valeur dans un set

Les fonctions String

1. Name() : Retourne le nom d'un membre sous forme de chaîne
2. UniqueName() : Retourne le nom unique complet d'un membre
3. Caption() : Retourne le libellé d'affichage d'un membre
4. Format() : Formate une valeur selon un format spécifié
5. Instr() : Recherche une sous-chaîne et retourne sa position
6. Left() : Retourne les n premiers caractères d'une chaîne
7. Right() : Retourne les n derniers caractères d'une chaîne
8. Mid() : Extrait une sous-chaîne à partir d'une position
9. Len() : Retourne la longueur d'une chaîne
10. LCase() : Convertit une chaîne en minuscules
11. UCase() : Convertit une chaîne en majuscules

Les fonctions String

12. Trim() : Supprime les espaces au début et à la fin
13. LTrim() : Supprime les espaces au début
14. RTrim() : Supprime les espaces à la fin
15. Replace() : Remplace une sous-chaîne par une autre
16. StrToMember() : Convertit une chaîne en membre MDX
17. StrToSet() : Convertit une chaîne en set MDX
18. StrToTuple() : Convertit une chaîne en tuple MDX
19. StrToValue() : Convertit une chaîne en valeur numérique
20. MemberToStr() : Convertit un membre en chaîne
21. SetToStr() : Convertit un set en chaîne
22. TupleToStr() : Convertit un tuple en chaîne
23. UserName : Retourne le nom de l'utilisateur actuel
24. Generate() : Génère une chaîne concaténée à partir d'un set

Les fonctions Time

1. Ytd() : Retourne les membres depuis le début de l'année (Year-To-Date)
2. Qtd() : Retourne les membres depuis le début du trimestre (Quarter-To-Date)
3. Mtd() : Retourne les membres depuis le début du mois (Month-To-Date)
4. Wtd() : Retourne les membres depuis le début de la semaine (Week-To-Date)
5. PeriodsToDate() : Retourne les membres depuis le début d'une période spécifiée
6. ParallelPeriod() : Retourne le membre de la période parallèle
7. OpeningPeriod() : Retourne le premier descendant dans une période
8. ClosingPeriod() : Retourne le dernier descendant dans une période
9. LastPeriods() : Retourne un set des N dernières périodes
10. DateAdd() : Ajoute un intervalle de temps à une date
11. DateDiff() : Calcule la différence entre deux dates
12. DatePart() : Extrait une partie d'une date (année, mois, jour)

Les fonctions Time

14. DatesBetween() : Retourne les dates entre deux bornes
15. DateSerial() : Crée une date à partir d'année, mois, jour
16. FirstInMonth() : Retourne le premier jour du mois
17. LastInMonth() : Retourne le dernier jour du mois
18. LastChild : Retourne le dernier enfant (souvent utilisé pour périodes)
19. FirstChild : Retourne le premier enfant (souvent utilisé pour périodes)
20. PreviousMember : Retourne la période précédente
21. NextMember : Retourne la période suivante

Les fonctions Value

1. Value : Retourne la valeur numérique d'un membre ou d'une cellule
2. MemberValue : Retourne la valeur associée à un membre
3. Aggregate() : Agrège les valeurs d'un set selon le contexte
4. CalculationCurrentPass() : Retourne le numéro de passe de calcul actuel
5. CalculationPassValue() : Retourne la valeur à un passe de calcul spécifique
6. CoalesceEmpty() : Retourne la première valeur non vide d'une liste
7. IIf() : Retourne une valeur conditionnelle (If-Then-Else)
8. IsEmpty() : Vérifie si une valeur est vide (NULL)
9. Abs() : Retourne la valeur absolue d'un nombre
10. Ceiling() : Arrondit au plus petit entier supérieur
11. Floor() : Arrondit au plus grand entier inférieur
12. Round() : Arrondit à un nombre de décimales spécifié
13. Exp() : Retourne e élevé à une puissance

Les fonctions Value

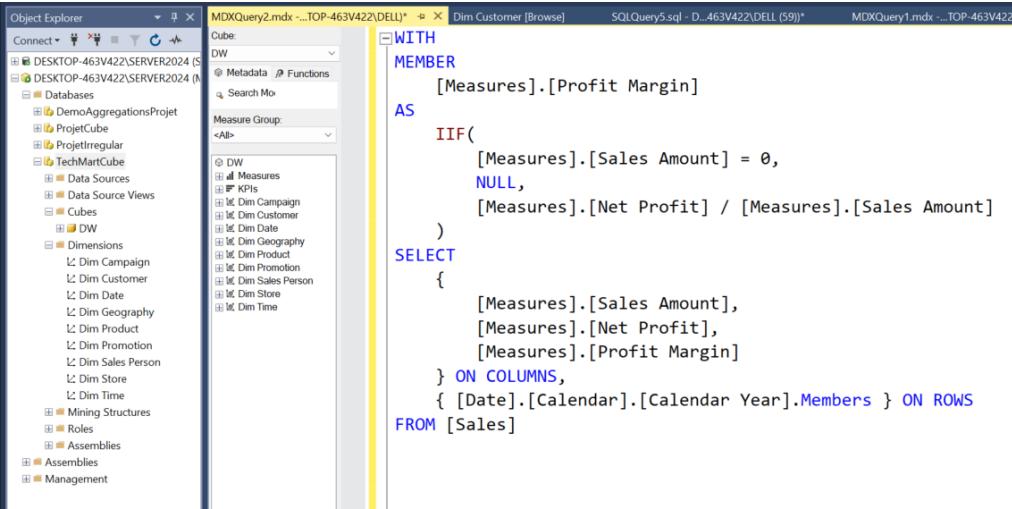
12. 14. Ln() : Retourne le logarithme naturel
15. Log() : Retourne le logarithme dans une base spécifiée
16. Log10() : Retourne le logarithme en base 10
17. Sqrt() : Retourne la racine carrée
18. Power() : Élève un nombre à une puissance
19. IsAncestor() : Vérifie si un membre est ancêtre d'un autre
20. IsGeneration() : Vérifie si un membre appartient à une génération
21. IsLeaf() : Vérifie si un membre est une feuille (sans enfants)
22. IsSibling() : Vérifie si deux membres sont frères/sœurs
23. ValidMeasure() : Retourne une mesure valide dans le contexte
24. Null : Représente une valeur nulle
25. Error() : Génère une erreur avec un message personnalisé

Les éléments MDX

38

Où utiliser MDX?

- ❑ Le meilleur endroit est le Query Editor de SSMS
- ❑ MDX Studio
- ❑ Au niveau de MDX Script au sein de SSDT (Mais pas agréable)



The screenshot shows the Microsoft Data Modeler interface. On the left, the Object Explorer displays a tree structure of database objects including Databases, Dimensions, and Measures. In the center, the MDXQuery2.mdx tab is active, showing the following MDX script:

```
WITH MEMBER [Measures].[Profit Margin]
AS
  IIF(
    [Measures].[Sales Amount] = 0,
    NULL,
    [Measures].[Net Profit] / [Measures].[Sales Amount]
  )
SELECT
{
  [Measures].[Sales Amount],
  [Measures].[Net Profit],
  [Measures].[Profit Margin]
} ON COLUMNS,
{ [Date].[Calendar].[Calendar Year].Members } ON ROWS
FROM [Sales]
```

“

Lab:Opérations basiques

Afficher les membres sans données

40

- Il est possible à travers une SET vide au niveau des colonnes `SET{} ON 0`

```
SELECT
{
} ON COLUMNS,
[Dim Customer].[Customer].members ON 1
FROM
[InternetSalesCube]
```



The screenshot shows a results grid from SSMS with two tabs at the top: 'Messages' and 'Results'. The 'Results' tab is selected and displays a list of names under the heading 'All'. The names listed are: Jon Yang, Eugene Huang, Ruben Torres, Christy Zhu, Elizabeth Johnson, Julio Ruiz, and Janet Alvarez.

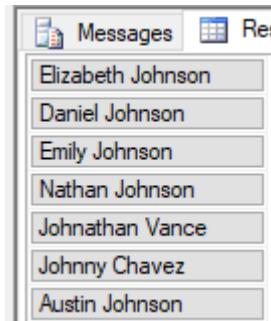
All
Jon Yang
Eugene Huang
Ruben Torres
Christy Zhu
Elizabeth Johnson
Julio Ruiz
Janet Alvarez

Techniques de filtres

41

- Si nous voulons encore filtrer les membres pour avoir des valeurs de membres bien déterminés à ne pas confondre avec WHERE

```
SELECT
{ } ON 0,
{ Filter(
[Dim Customer].[Customer].members,
-- Recherche insensible à la casse recommandée
InStr(1,
[Dim Customer].[Customer].CurrentMember.Name,
'John',1
) > 0
)
} ON 1
FROM
[InternetSalesCube]
```



A screenshot of a Microsoft Excel ribbon interface. The ribbon tabs visible are 'Messages' and 'Re...'. Below the ribbon, there is a table containing seven rows of names. The names listed are Elizabeth Johnson, Daniel Johnson, Emily Johnson, Nathan Johnson, Johnathan Vance, Johnny Chavez, and Austin Johnson.

Elizabeth Johnson
Daniel Johnson
Emily Johnson
Nathan Johnson
Johnathan Vance
Johnny Chavez
Austin Johnson

Utiliser des valeurs par défaut

42

- Il est possible à travers l'utilisation d'une variable dummy

```
WITH
MEMBER [Measures].[Dummy] AS -1
SELECT
{[Measures].[Dummy]} ON 0,
{[Dim Customer].[Customer].[Customer].MEMBERS} ON
1
FROM
[InternetSalesCube]
```

Messages		Results
		Dummy
Jon Yang		-1
Eugene Huang		-1
Ruben Torres		-1
Christy Zhu		-1
Elizabeth Johnson		-1
Julio Ruiz		-1
Janet Alvarez		-1
...

L'utilisation de la clause Where

43

- Nous voulons restreindre la vue à des membres particuliers nous utilisons la clause WHERE comme en SQL

```
SELECT
{[Dim Customer].[Customer
Hierarchy].[Country].members} ON 0,
{[Dim Product].[Product Category].members} ON 1
FROM
[InternetSalesCube]
WHERE
(
[Measures].[Sales Amount],
[Due Date].[Calendar Hierarchy].[Calendar
Year].&[2005]
)
```

	Australia	Canada
All	1225145.275	128938.4574
Bikes	1225145.275	128938.4574
Components	(null)	(null)
Clothing	(null)	(null)
Accessories	(null)	(null)
Unknown	(null)	(null)

L'utilisation de la fonction NonEmpty()

44

Problématique

- Nous voulons restreindre la vue à des membres non nul pour optimiser les performances de la requête

Solution

- Nous utilisons la fonction non empty

Remarque

- Il existe deux fonctions NonEmpty() et NON EMPTY, les deux éliminent les sorties nulles
- NON EMPTY est plus rapide mais elle ne peut pas être utilisée qu'au niveau des axes seulement, elle est considérée comme un opérateur

L'utilisation de la fonction NonEmpty()

45

- Voici un exemple

```
SELECT  
{[Measures].[Sales Amount]} ON 0,  
NON EMPTY ←  
Filter(  
    {[Dim Customer].[Customer].members}*  
    {[Due Date].[Month Number Of Year].members},  
    [Measures].[Sales Amount]>1000  
) ON 1
```

```
FROM  
[InternetSalesCube]
```

Utilisée au niveau de l'axe
des lignes

	Sales Amount
All	29358677.2206504
April	642273.966600001
April	492819.5737
April	1457599.46999999
August	501182.445800001
August	513690.3892
August	880663.299999989
August	20631.3
December	666111.662200001

L'utilisation de la fonction NonEmpty()

46

- ❑ NonEmpty() : Évaluation après récupération des données (plus lent)

```
SELECT
    { [Measures].[Sales Amount] } ON 0,
    Filter(
        NonEmpty(                                     ← Utilisée à l'intérieur de l'axe
            { [Dim Customer].[Customer].Members }
            *
            { [Due Date].[Month Number Of Year].Members },
            { [Measures].[Sales Amount] }
        ),
        [Measures].[Sales Amount] > 1000
    ) ON 1
FROM [InternetSalesCube]
```

L'utilisation de la fonction Properties()

47

- ❑ Nous voulons afficher les propriétés des membres au niveau de la requête

L'utilisation de la fonction properties()

48

- Nous volons afficher les propriétés des membres au niveau de la requête

```
WITH MEMBER [Measures].[Subcategories] AS  
[Dim Product].[Product].Properties(  
"Product Subcategory" )
```

```
SELECT  
{[Measures].[Sales  
Amount],[Measures].[Subcategories]} ON 0,  
{[Dim Customer].[Customer].members} ON 1  
  
FROM  
[InternetSalesCube]
```

La gestion des erreurs

49

- Voici un exemple classique pour éviter la division par zéro

```
WITH MEMBER [Measures].[Safe Ratio] AS
  IIf(
    [Measures].[Quantity] = 0,
    NULL, -- ou 0, ou "N/A"
    [Measures].[Sales] / [Measures].[Quantity]
  )

SELECT
  {[Measures].[Sales], [Measures].[Quantity],
  [Measures].[Safe Ratio]} ON 0,
  [Product].[Product].Members ON 1
FROM [Cube]
```

La gestion des erreurs

50

- Voici un autre exemple pour détourner les valeurs négatives non tolérées

```
WITH MEMBER [Measures].[Positive Sales] AS  
IIF(  
    [Measures].[Sales Amount] < 0,  
    0,  
    [Measures].[Sales Amount]  
,  
FORMAT_STRING = "#,##0.00"
```

```
SELECT  
    [Measures].[Positive Sales] ON 0  
FROM [Cube]
```

La gestion des erreurs

51

- La detection des valeurs vides avec IsEmpty

```
WITH MEMBER [Measures].[Safe Calculation] AS  
IIf(  
    IsEmpty([Measures].[Sales Amount]),  
    0, -- Valeur par défaut  
    [Measures].[Sales Amount] * 1.1  
)
```

```
SELECT  
    [Measures].[Safe Calculation] ON 0,  
    [Date].[Calendar].Members ON 1  
FROM [Cube]
```

La gestion des erreurs

52

- L'utilisation de la première valeur non vide avec CoalesceEmpty

```
WITH MEMBER [Measures].[Reliable Value] AS  
CoalesceEmpty(  
    [Measures].[Internet Sales],  
    [Measures].[Reseller Sales],  
    [Measures].[Previous Year Sales],  
    0 -- Valeur par défaut si tout est vide  
)
```

```
SELECT  
    [Measures].[Reliable Value] ON 0,  
    [Product].[Category].Members ON 1  
FROM [Cube]
```

La gestion des erreurs

53

- L'utilisation de la fonction Error() pour gérer les erreurs

```
WITH MEMBER [Measures].[Validated Ratio] AS
  IIf(
    [Measures].[Denominator] = 0,
    Error("Division impossible : dénominateur nul"),
    [Measures].[Numerator] /
    [Measures].[Denominator]
  )

SELECT
  [Measures].[Validated Ratio] ON 0
FROM [Cube]
```

La gestion des erreurs

54

- Validation des membres et hiérarchies

```
WITH MEMBER [Measures].[Member Exists] AS
    IIF(
        [Product].[Category].CurrentMember.UniqueName =
        "[Product].[Category].[Unknown]",
        "Membre invalide",
        "OK"
    )
SELECT
    [Measures].[Member Exists] ON 0
FROM [Cube]
```

Performances

❑ L'optimisation des performances MDX est **cruciale** car une requête mal écrite peut :

- Prendre plusieurs minutes au lieu de quelques secondes
- Consommer toute la RAM du serveur
- Bloquer d'autres utilisateurs
- Créer une mauvaise expérience utilisateur

Performances

Comment MDX exécute les requêtes

1. Analyse syntaxique (Parsing)
2. Résolution des membres/dimensions
3. Construction du sous-cube (Subspace)
4. Évaluation des formules
5. Application des filtres
6. Tri et formatage
7. Retour des résultats

Minimiser les étapes 3, 4 et 5 qui sont les plus coûteuses.

Optimisation des performances

57

Performances

Limiter la taille des SET dès le départ

```
-- LENT : Génère 1M x 1M = 1 trillion de tuples
SELECT ON 0,
{[Customer].[Customer].Members * [Product].[Product].Members} ON 1
FROM [Cube]

-- RAPIDE : Filtre AVANT le CrossJoin
SELECT ON 0,
NON EMPTY
{
    TopCount([Customer].[Customer].Members, 100, [Measures].[Sales])
    *
    TopCount([Product].[Product].Members, 50, [Measures].[Sales])
} ON 1
FROM [Cube]
```

Optimisation des performances

58

Performances

Toujours filtrer **avant** les opérations de CrossJoin

```
-- LENT : Order() trie tout le set
SELECT
    [Measures].[Sales Amount] ON 0,
    Order(
        [Product].[Product Hierarchy].Members,
        [Measures].[Sales Amount],
        DESC
    ) ON 1
FROM [Cube]

-- RAPIDE : Hierarchize() utilise l'ordre naturel
SELECT
    [Measures].[Sales Amount] ON 0,
    Hierarchize(
        [Product].[Product Hierarchy].Members
    ) ON 1
FROM [Cube]
```

Optimisation des performances

59

Performances

Utiliser SCOPE pour les calculs en masse

```
-- LENT : WITH MEMBER recalcule à chaque cellule
WITH MEMBER [Measures].[Adjusted Sales] AS
    [Measures].[Sales] * 1.1

SELECT
    [Measures].[Adjusted Sales] ON 0,
    [Product].[Product].Members ON 1
FROM [Cube]

-- RAPIDE : SCOPE calcule une seule fois dans le script MDX
-- (Dans le MDX Script du cube)
SCOPE([Measures].[Adjusted Sales]);
    THIS = [Measures].[Sales] * 1.1;
END SCOPE;

-- Requête simplifiée
SELECT
    [Measures].[Adjusted Sales] ON 0,
    [Product].[Product].Members ON 1
FROM [Cube]
```

Optimisation des performances

60

Performances

Éviter les requêtes trop larges

```
-- DANGEREUX : Peut consommer 10+ GB de RAM
SELECT
    [Measures].Members ON 0,
    [Customer].[Customer].Members *
    [Product].[Product].Members *
    [Date].[Date].Members ON 1
FROM [Cube]

-- RAISONNABLE : Limite le résultat
SELECT
    {[Measures].[Sales Amount], [Measures].[Order
Quantity]} ON 0,
    NON EMPTY
    TopCount([Customer].[Customer].Members, 1000,
    [Measures].[Sales Amount]) ON 1
FROM [Cube]
WHERE [Date].[Year].&[2026]
```