

Travail Pratique
associé au cours
« Entrepôts de données »

Bernard ESPINASSE
2019-2020

Cahier de rendu

Date du TP :

Promotion :

Equipe :

Membres de l'Equipe :

.....
.....
.....
.....

Sommaire

PARTIE A : Installation de l'architecture logicielle	2
A1. Installation du serveur Postgresql.....	2
A2. Création de la base de données de l'entrepôt sous Postgresql.....	2
A3. Installation de JRUBIK/MONDRIAN	2
PARTIE B : Modélisation multidimensionnelle	3
B.1 – Création d'un schéma de cube MONDRIAN en XML	3
B.2 – Analyse du cube « Vente » utilisé pour les requêtes MDX.....	5
PARTIE C : Requêtes MDX sur le cube « Vente ».....	8
C.1 - Requêtes MDX simples	8
C.2 - Requêtes avec usage de fonctions	10
C.3 - Requêtes MDX avec usage de Sets et Members.....	13
C.4 - Requêtes MDX complexes.....	17

PARTIE A : Installation de l'architecture logicielle

A1. Installation du serveur Postgresql

Le télécharger à l'adresse :

<http://www.postgresql.org/download/>

Télécharger aussi le programme d'administration de PostGreSQL : **pgadmin3**

Il existe aussi un programme d'administration : **phpPgadmin** à lancer sur un navigateur.

Il vous est recommandé de **définir un utilisateur administrateur de base de donnée pour l'équipe**.

A2. Création de la base de données de l'entrepôt sous Postgresql

Le script de création de la base de données de l'entrepôt « Polytech'Market » (**esql_create.sql**), le script de remplissage de cette base (**esql_add.sql**), le script de suppression de la base (**esql_del.sql**), sont sur la page Web du cours.

Seul l'utilisateur administrateur peut créer la base de données.

Vous devez lancer ces scripts à partir d'une fenêtre d'un des deux logiciels d'administration de Postgresql (**pgadmin3** ou **pgadmin4** ou **phpPgadmin**).

Une fois la base de données créée, l'utilisateur administrateur peut attribuer des droits aux autres utilisateurs de son équipe (usage de **pgadmin3** ou **pgadmin4** ou **phpPgadmin**).

A3. Installation de JRUBIK/MONDRIAN

Télécharger **Jrubik** ou sur la page Web du cours ou à l'adresse :

<http://rubik.sourceforge.net/jrubik/intro.html>

JRUBIK contient la bibliothèque MONDRIAN.

Installer JRUBIK selon la procédure spécifiée (attention au prérequis), cette installation installe en même temps MONDRIAN.

PARTIE B : Modélisation multidimensionnelle

Cette partie a pour objectif de comprendre la modélisation multidimensionnelle et spécifier des structures multidimensionnelles dont des cubes MONDRIAN en langage XML.

B.1 – Création d'un schéma de cube MONDRIAN en XML

On vous demande d'élaborer un schéma d'un cube MONDRIAN « Stocks » en XML permettant d'analyser les stocks. **Vous utiliserez pour cela un éditeur XML de votre choix.**

1- Déterminez quelle table peut être utilisée comme table des faits pour ces analyses, sachant que les mesures à étudier seront : le nombre de produits commandés, le nombre de produits livrés, les prix d'achat et de vente par lots des produits.

Réponse :

2- Déterminez quelles tables de dimensions seront utilisées pour ces analyses sachant que l'on souhaite une analyse par produit (et type de produits), date et entrepôt de stockage (et type d'entrepôt de stockage).

Réponse :

3- On souhaite maintenant créer un cube utilisant ces tables. Elaborez le schéma de ce cube sachant que :

- Les mesures à ajouter sont : « nombre de produits différents commandés », « total des produits commandés », « nombre de produits livrés », « ventes de l'entrepôt », « coût d'entreposage », « bénéfice de l'entrepôt » (ventes-coûts), « perte de produits » (produit commandés – produits livrés).
- La précision choisie est de 2 chiffres après la virgule.

Définir la partie du schéma associée à la définition de ces mesures.

Réponse :

4- La dimension « Produit » utilise les tables Produit et Type_Produits. Les niveaux « levels » pour cette dimension sont : All, Produit_family, Produit_department, Produit_category, Produit_subcategory, Brand_name et Produit_name.

Définir la partie du schéma associée à cette dimension « Produit ».

Réponse :

5- La dimension « Temps » utilise un seul tableau « Temps_by_day ». Les niveaux pour cette dimensions sont : Year, Trimestre, Month.

Définir la partie du schéma associée à cette dimension « Temps ».

Réponse :

Vous collerez à cet emplacement le code XML du schéma du cube MONDRIAN « Stock ».

B.2 – Analyse du cube « Vente » utilisé pour les requêtes MDX

Soit le cube « Vente » suivant défini sur la base d'entrepôt « Polytech'Market »:

```
<Schema name="Polytech_market">    <Dimension name="Produit">
    <Hierarchy name="defaultHiera" hasAll="true" allMemberName="All Produit"
        primaryKey="produits_id" primaryKeyTable="produits">
        <Join leftKey="type_produits_id" rightKey="type_produits_id">
        <Table name="type_produits" schema="public" />
        <Table name="produits" schema="public" /></Join>
        <Level name="product_family" table="type_produits" column="product_family"
            uniqueMembers="false"/>
        <Level name="product_department" table="type_produits"
            column="product_department"
            uniqueMembers="false"/>
        <Level name="product_category" table="type_produits" column="product_category"
            uniqueMembers="false"/>
        <Level name="product_subcategory" table="type_produits"
            column="product_subcategory"
            uniqueMembers="false"/>
        <Level name="brand_name" table="produits" column="brand_name"
            uniqueMembers="false"/>
        <Level name="product_name" table="produits" column="product_name"
            uniqueMembers="false"/>
    </Hierarchy>
</Dimension>
<Dimension name="Temps">
    <Hierarchy name="defaultHiera" hasAll="true" allMemberName="All Temps"
        primaryKey="temps_id">
        <Table name="temps" schema="public" />
        <Level name="l_annee" table="temps" column="l_annee" uniqueMembers="false"/>
        <Level name="trimestre" table="temps" column="trimestre" uniqueMembers="false"/>
        <Level name="mois" table="temps" column="mois" uniqueMembers="false"/>
    </Hierarchy>
</Dimension>
<Dimension name="Client">
    <Hierarchy name="defaultHiera" hasAll="true" allMemberName="All client"
        primaryKey="client_id">
        <Table name="client" schema="public" />
        <Level name="marital_status" table="client" column="marital_status"
            uniqueMembers="false"/>
        <Level name="fullname" table="client" column="fullname" uniqueMembers="false"/>
    </Hierarchy>
</Dimension>
<Dimension name="Magasin">
    <Hierarchy name="defaultHiera" hasAll="true" allMemberName="All Magasin"
```

```

        primaryKey="magasin_id">
<Table name="magasin" schema="public" />
<Level name="store_country" table="magasin" column="store_country"
    uniqueMembers="false"/>
<Level name="store_state" table="magasin" column="store_state"
    uniqueMembers="false"/>
<Level name="store_city" table="magasin" column="store_city"
    uniqueMembers="false"/>
<Level name="store_name" table="magasin" column="store_name"
    uniqueMembers="false">
    <Property name="store_sqft" column="store_sqft"/>
</Level>
</Hierarchy>
</Dimension>
<Dimension name="Gender">
    <Hierarchy name="defaultHiera" hasAll="true" allMemberName="All Gender"
        primaryKey="client_id">
        <Table name="client" schema="public" />
        <Level name="gender" table="client" column="gender" uniqueMembers="false"/>
        <Level name="fullname" table="client" column="fullname" uniqueMembers="false"/>
    </Hierarchy>
</Dimension>
<Cube name="Vente">
    <Table name="ventes" schema="public" />
    <DimensionUsage name="Produit" source="Produit" foreignKey="produit_id"/>
    <DimensionUsage name="Temps" source="Temps" foreignKey="temps_id"/>
    <DimensionUsage name="Client" source="Client" foreignKey="client_id"/>
    <DimensionUsage name="Magasin" source="Magasin" foreignKey="magasin_id"/>
    <DimensionUsage name="Gender" source="Gender" foreignKey="client_id"/>
    <Measure name="nb ventes" column="produit_id" aggregator="count"
        formatString="#,###"/>
    <Measure name="nb transactions" column="unit_sales" aggregator="sum"
        formatString="Standard"/>
    <Measure name="prix de vente" column="store_sales" aggregator="sum"
        formatString="Standard"/>
    <Measure name="prix achat" column="store_cost" aggregator="sum"
        formatString="Standard"/>
    <Measure name="nb clients" column="client_id" aggregator="distinct-count"
        formatString="Standard"/>
    <CalculatedMember name="Profit" dimension="Measures"
        formula="[Measures].[prix de vente] - [Measures].[prix achat]">
</CalculatedMember>
    <CalculatedMember name="Profit past" dimension="Measures"
        formula="CoalesceEmpty([Measures].[Profit],[Temps].PREVMEMBER),
            [Measures].[Profit]">
</CalculatedMember>
</Cube>

```

</Schema>

On vous demande de donner pour ce cube « Sales » :

1- les tables de la base « Polytech'market » concernées par le cube

Réponse :

2- les dimensions de ce cube avec leur(s) hiérarchie(s)

Réponse :

3- les membres impliqués dans ce cube

Réponse :

4- les mesures impliquées dans ce cube

Réponse :

PARTIE C : Requêtes MDX sur le cube « Vente »

Cette partie a pour objectif d'écrire des requêtes OLAP en langage MDX en utilisant l'outil logiciel JRUBIK, requêtes sur le cube « Ventes » de la base « Polytech'market » donné en partie A-2.

C.1 - Requêtes MDX simples

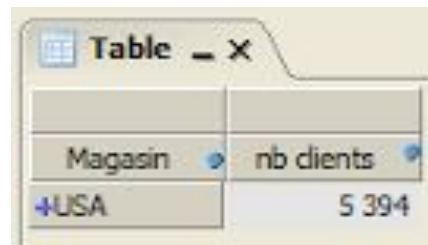
5- Afficher le nombre de clients des magasins qui se trouvent aux USA.

Réponse EXEMPLE :

- **Code MDX généré par JRUBIK:**

```
SELECT
    [Measures].[Nombre de clients] ON COLUMNS,
    {[Store].[All Stores].[USA]} ON ROWS
FROM [Vente]
```

- **Tableau résultats généré par JRUBIK:**



Magasin	nb clients
+USA	5 394

6- Afficher pour tous les magasins « Magasin » les mesures suivantes : Nb ventes, Magasin Cost, Magasin Vente, Vente Count et Nombre de clients.

Structure du tableau de résultats attendu :



Magasin	nb clients
+USA	

Réponse :

Vous “collerez” ici :

- ***le code MDX généré par JRubik.***
- ***Le tableau de résultats obtenu***

7- Modifier la requête précédente pour afficher le nombre de clients des magasins qui se trouvent à Seattle dans l'état de Washington (WA) aux USA.

Structure attendue du tableau de résultats:

Magasin	nb clients
Seattle	
Store 15	

Réponse :

Vous “collerez” ici :

- ***le code MDX généré par JRubik.***
- ***Le tableau de résultats obtenu***

8- Modifier la requête précédente pour afficher le nombre de clients des magasins en fonction des villes qui se trouvent dans l'état de Washington (WA) aux USA.

Recommandation : Vous pouvez utiliser la fonction « Descendant » (voir l'annexe).

Structure attendue du tableau de résultats:

Magasin	Client
Bellingham	+ All client
Store 2	
Bremerton	
Store 3	
Seattle	
Store 15	
Spokane	
Store 16	
Tacoma	
Store 17	
Walla Walla	
Store 22	
Yakima	
Store 23	

Réponse :

Vous “collerez” ici :

- ***le code MDX généré par JRubik.***
- ***Le tableau de résultats obtenu***

C.2 - Requêtes avec usage de fonctions

9- Afficher toutes les mesures calculées et non calculées (ex : “Profit Last Period”) des magasins de l’état de Californie (CA) et de chaque magasin des villes de cet état.

Recommandation : utiliser la fonction « AddCalculateMembers » (Voir l’annexe)

Structure attendue du tableau de résultats:



Magasin	nb ventes	prix de vente	prix achat	Calculo
Alameda				
HQ				
Beverly Hills				
Store 6				
Los Angeles				
Store 7				
San Diego				
Store 24				
San Francisco				
Store 14				

Reponse :

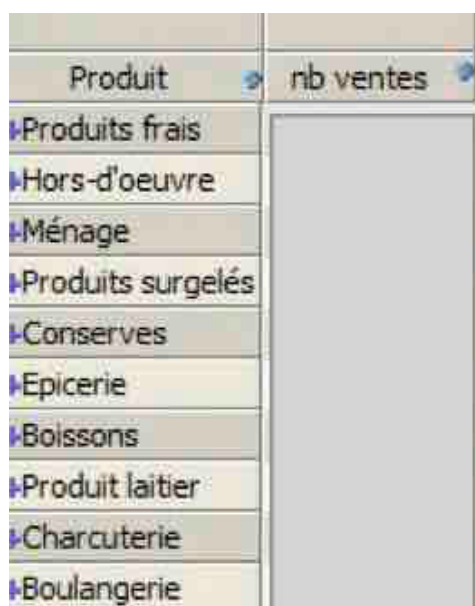
Vous “collerez” ici :

- le code MDX généré par JRubik.
- Le tableau de résultats obtenu

10- Déterminer les tops 10 des catégories de produits vendus.

Recommandation : Voir l’utilisation de la fonction TopCount dans l’annexe.

Structure attendue du tableau de résultats:



Produit	nb ventes
Produits frais	
Hors-d'oeuvre	
Ménage	
Produits surgelés	
Conserves	
Epicerie	
Boissons	
Produit laitier	
Charcuterie	
Boulangerie	

Réponse :

Vous “collerez” ici :

- ***le code MDX généré par JRubik.***
- ***Le tableau de résultats obtenu***

11- Déterminer les marques qui constituent 70 % des ventes.

Recommandation : Utiliser la fonction TopPercent (voir l'annexe) ;

Structure attendue du tableau de résultats:

Product	Unit Sales
+Hermanos	
+Tell Tale	
+Ebony	
+Tri-State	
+High Top	
+Hermanos	
+Tell Tale	
+Ebony	
+Tri-State	
+High Top	
+Just Right	
+Fort West	
+Nationaleel	
+Booker	
+Even Better	
+Better	
+Carlson	
+Blue Label	
+Pleasant	
+Bravo	
+Gorilla	
+Fast	

Réponse :

Vous “collerez” ici :

- ***le code MDX généré par JRubik.***
- ***Le tableau de résultats obtenu***

12- Déterminer les marques qui composent la tranche inférieure des 20 % des ventes.

Recommandation : vous pouvez utiliser la fonction BottomPercent (voir l'annexe)

Structure attendue du tableau de résultats:

Product	Unit Sales
+Choice	
+Bird Call	
+Golden	
+Black Tie	
+Cormorant	
+Bravo	
+Tip Top	
+Mighty Good	
+Better	
+Blue Label	
+CDR	
+Denny	
+Faux Products	

Réponse :

Vous “collerez” ici :

- *le code MDX généré par JRubik.*
- *Le tableau de résultats obtenu*

13- Déterminer les tops 5 clients dans les top 5 magasins

Recommandation : Utiliser les fonctions Generate, TopCount (voir annexe)

Structure attendue du tableau de résultats:

Store	Customers	Unit Sales
Store 13	Bobby Lundin	
	Timothy Beard	
	Barbara Chan	
	Erin Smith	
	LeeAnn Otte	
Store 17	Jeremy Styers	
	Patricia Gervasi	
	William Wade	
	Diane Brown	
	Tammy Wilkinson	
Store 11	Janeen Broxton	
	Jim Velasquez	
	Rosemary Stephens	

Réponse :

Vous “collerez” ici :

- le code MDX généré par JRubik.
- Le tableau de résultats obtenu

C.3 - Requêtes MDX avec usage de Sets et Members

14- Déterminer les marques vendues au cours des 3 derniers trimestres.

Recommandation : vous pouvez utiliser les fonctions suivantes : Filter, IsEmpty, Tail, Union, Descendants (Voir annexe).

Structure attendue du tableau de résultats:

Product	Time		
	+ Q2	+ Q3	+ Q4
+ Colony			
+ Fantastic			
+ Great			
+ Modell			
+ Sphinx			
+ Colony			
+ Fantastic			
+ Great			
+ Modell			
+ Sphinx			
+ Colony			
+ Fantastic			
+ Great			
+ Modell			
+ Sphinx			
+ BBB Best			
+ CDR			
+ Landslide			
+ Plato			
+ Super			

Réponse :

Vous “collerez” ici :

- le code MDX généré par JRubik.
- Le tableau de résultats obtenu

15- Déterminer les tendances récentes des meilleures marques les plus vendues.

Recommandation :

- Déclarer un ensemble « SET » qui va contenir les 10 TOP Marque vendues*
- Déclarer un ensemble « SET » qui va contenir le dernier mois.*
- Déclarer un ensemble qui va contenir les derniers six mois. (en utilisant la fonction Lag appliquer sur l'ensemble dernier mois)*

d. Fonctions à utiliser : Lag, TopCount, ... (Voir l'annexe)

Structure attendue du tableau de résultats:

Product	Time						
	6	7	8	9	10	11	12
→Hermanos							
→Tell Tale							
→Ebony							
→Tri-State							
→High Top							
→Hermanos							
→Tell Tale							
→Ebony							
→Tri-State							
→High Top							

Réponse :

Vous “collerez” ici :

- *le code MDX généré par JRubik.*
- *Le tableau de résultats obtenu*

16- Détermination des ventes qui dépassent les coûts du magasin de 155 %.

Recommandation :

- Définir un membre calculé en utilisant WITH MEMBER qui correspond à : $(MagasinVente - MagasinCost)/MagasinCost$.*
- Vous pouvez utiliser la fonction Filter (voir annexe).*

Structure attendue du tableau de résultats:

Product	Store Sales	Store Cost	SalesRatio
	→Plato		
→Sphinx			
→Super			
→Better			
→Pleasant			
→Pleasant			
→High Top			
→ADJ			
→Hilltop			
→Cormorant			

Reponse :

Vous “collerez” ici :

- *le code MDX généré par JRubik.*
- *Le tableau de résultats obtenu*

17- Déterminer les Tops 10 et les 10 dernières marques de produit.

Recommandation :

- Définir un SET qui contient les marques triées (par la fonction Order).*
- Utiliser les fonctions suivantes : Order, Rank, Head, Tail, Union, ... (voir l'annexe).*

Structure attendue du tableau de résultats:

Product	Brand Rank	Unit Sales
+Hermanos		
+Tell Tale		
+Ebony		
+Tri-State		
+High Top		
+Hermanos		
+Tell Tale		
+Ebony		
+Tri-State		
+High Top		
+Blue Label		
+Better		
+Mighty Good		
+Bravo		
+Tip Top		
+Cormorant		
+Black Tie		
+Golden		
+Bird Call		
+Choice		

Réponse :

Vous “collerez” ici :

- le code MDX généré par JRubik.
- Le tableau de résultats obtenu

18- Ecrire une requête qui affiche les familles de produits (Produit Family) sur les lignes, et les ventes des magasins [Magasin Vente] et pourcentage de la contribution totale [Percent of total] (pour stocker les ventes) sur les colonnes.

Recommendation :

- a. Définir un nouveau membre calculée qui va correspondre à la contribution totale $(([\text{Measures}].[\text{Magasin Vente}], [\text{Produit}].\text{CurrentMember}) / ([\text{Measures}].[\text{Magasin Vente}], [\text{Produit}].[\text{All Produits}]))$

Structure attendue du tableau de résultats:

Product	Store Sales	Percent of Total
+Drink		
+Food		
+Non-Consumable		

Réponse :

Vous “collerez” ici :

- le code MDX généré par JRubik.
- Le tableau de résultats obtenu

19- Ecrire une requête MDX qui affiche uniquement les catégories de produits qui contiennent des produits qui sont dans le top cinq en fonction des ventes en magasin sur les lignes. Sur les colonnes, afficher uniquement les stocks des ventes.

Recommandation :

- a. Déclarer un SET avec les top des produits, et un SET avec les top des catégorie.
- b. Vous pouvez utiliser les fonctions Ancestor, Generate et TopCount (voir annexe).

Structure attendue du tableau de résultats:

Product	Store Sales
Paper Products	
Hygiene	
Bread	
Vegetables	

Réponse :

Vous “collerez” ici :

- le code MDX généré par JRubik.
- Le tableau de résultats obtenu

20- Ecrire la requête qui affiche les 5 plus importants produits dans chaque département.

Recommandation : vous pouvez utiliser les fonctions Generate et TopCount (voir l'annexe).

Structure attendue du tableau de résultats:

Product	Time			
	+ Q1	+ Q2	+ Q3	+ Q4
Alcoholic Beverages				
Portsmouth Light Beer				
Portsmouth Chardonnay Wine				
Top Measure Light Wine				
Good White Zinfandel Wine				
Portsmouth Chardonnay				
Beverages				
Washington Apple Drink				
BBB Best French Roast Coffee				
CDR Regular Coffee				
Fabulous Apple Juice				
Skinner Diet Soda				
Dairy				
Even Better 1% Milk				
Gorilla 2% Milk				
Booker Whole Milk				
Club Whole Milk				
Carlson 2% Milk				

Réponse :

Vous “collerez” ici :

- le code MDX généré par JRubik.
- Le tableau de résultats obtenu

C.4 - Requêtes MDX complexes

Pour les requêtes de ce groupe, on vous demande, en quelques lignes :

- d'expliquer l'analyse que permet de faire la requête
- de décrire les dimensions, membres, mesures, ... traités par la requête
- d'expliquer les opérations réalisées par la requête sur le cube Sales (dimensions, mesures, ...)

21- Requête utilisant la fonction Hierarchize, Union, CrossJoin (Voir annexe)

Soit la requête MDX:

```
SELECT
  Hierarchize(Union(Union(Crossjoin({[client].[All client]},
    {[Gender].[All Gender]}), Crossjoin({[client].[All client]},
    [Gender].[All Gender].Children)),
  Union(Crossjoin([client].[All client].Children, {[Gender].[All Gender]}),
  Crossjoin([client].[All client].Children, [Gender].[All Gender].Children)))
  ON COLUMNS,
  {[Measures].[Nb ventes]} ON ROWS
FROM [Vente]
```

Le résultat obtenu :

Mesures	Client								
	- All client			+ M			+ S		
	Gender			Gender			Gender		
	- All Gender	+ F	+ M	- All Gender	+ F	+ M	- All Gender	+ F	+ M
nb ventes	↓ 164 558	↓ 83 872	↓ 80 686	↓ 81 913	↓ 42 714	↓ 39 199	↓ 82 645	↓ 41 158	↓ 41 487

Que fait cette requête ?

Réponse :

22- Requête avec définition de Set

Soit la requête MDX:

```
WITH SET [Months] AS 'Except(Descendants([Temps].[1998],
  [Temps].[mois]),
  {[Temps].[1998].[Q1].[01_Janvier],
  [Temps].[1998].[Q4].[12_Decembre]})'
member [Measures].[Growth] AS '((([Temps].CurrentMember,
  [Measures].[prix de vente]) - ([Temps].PrevMember,
  [Measures].[prix de vente])) / ([Temps].PrevMember,
  [Measures].[prix de vente]))', FORMAT_STRING = "#.00%"
MEMBER [Measures].[Avg Growth] as 'Avg([Months],
  [Measures].[Growth])', FORMAT_STRING = "#.00%"
```

```

SELECT
  {[Measures].[Avg Growth]} ON COLUMNS,
  Order([Produit].[product_category].Members, [Measures].[Avg Growth],
  BDESC) ON ROWS
FROM [Vente]

```

Le résultat obtenu :

Produit	Avg Growth
+Divers	10,21%
+Salle de bain	7,54%
+Décongestionnant	7,23%
+Huîtres en conserve	5,80%
+Divers	5,44%
+Légumes sous vide	5,02%
+Produits d'entretien	4,85%
+Petit déjeuner	4,64%
+Sardines en conserve	4,56%
+Palourdes en conserve	4,51%
+Anchois en conserve	4,21%
+Viande	3,81%
+Boissons gazeuses	3,51%
+Epicerie	3,41%
+Viande	3,38%
+Produit laitier	3,32%
+Desserts congelés	3,02%
+Matériel	3,01%
+Produits en plastique	2,99%
+Rhume	2,97%
+Oeufs	2,93%
+Thon en conserve	2,90%
+Produit laitier	2,73%
+Pain	2,68%
+Bougies	2,67%

Que fait cette requête ?

Réponse :

23- Requête avec IIF

Soit la requête MDX:

```

WITH MEMBER [Measures].[StoreSqftVal] AS
  '([Magasin].CurrentMember.Properties("Store_Sqft"))'
  MEMBER [Measures].[StoreSqft] AS
    'IIf(([Magasin].CurrentMember.Level.Name = "Store_name"),
    [Measures].[StoreSqftVal],
    Sum(Descendants([Magasin].CurrentMember,
    [Magasin].[Store_Name]), [Measures].[StoreSqftVal]))'

```

```

SELECT
    {[Measures].[StoreSqft]} ON COLUMNS,
    Filter(Descendants([Magasin].[All Magasin], [Magasin].[Store_State]),
        ([Measures].[StoreSqft] > 0.0)) ON ROWS
FROM [Vente]

```

Le résultat obtenu :

Magasin	StoreSqft
+BC	57 564
+DF	65 265
+Guerrero	23 593
+Jalisco	24 597
+Veracruz	34 791
+Yucatan	30 797
+Zacatecas	92 725
+CA	121 718
+OR	48 013
+WA	190 991

Que fait cette requête ?

Réponse :

24- Requête avec filtre

Soit la requête MDX:

```

WITH SET [Recent Months] AS
    'Filter([Temps.defaultHiera].[mois].Members,
        (NOT IsEmpty([Temps].CurrentMember)))'
    SET [Recent Month] AS
        '{[Recent Months].Item((Count([Recent Months]) - 1.0))}'
SELECT
    {[Recent Month],
    [Recent Month].Item(0.0).Lag(1.0),
    [Recent Month].Item(0.0).Lag(2.0)}
    ON COLUMNS,
    {[Magasin.defaultHiera].[All Magasin].[USA].Children}
    ON ROWS
FROM [Vente]

```

Le résultat obtenu :

Magasin	Temps		
	11	10	9
+CA	↓ 3 130	↓ 2 067	↓ 2 198
+OR	↓ 2 221	↓ 1 605	↓ 2 013
+WA	↓ 4 743	↓ 3 342	↓ 3 517

Que fait cette requête ?

Réponse :