

Personnalisation de requêtes MDX

Ladjel Bellatreche¹ Arnaud Giacometti² Patrick Marcel²
Hassina Mouloudi²

¹LISI, Université de Poitiers

²LI, Université François Rabelais de Tours

BDA 2006

Plan

Motivation

Les requêtes MDX considérées

Le profil utilisateur

- Les préférences

- Les contraintes de visualisation

La personnalisation

Conclusion

Exemple

```
SELECT { [City].Tours, [City].Orleans },  
       { [Year].2003, [Year].2004, [Year].2005, [Year].2006 },  
       { [Category].shoes, [Category].cloth,  
         [Category].food, [Category].drink }  
FROM   SalesCube  
WHERE  [Measures].quantity
```

Différentes visualisations possibles selon le profil

city = Tours

| | 2003 | 2004 | 2005 | 2006 |
|-------|-------|-------|-------|-------|
| food | 72,00 | 50,00 | 33,00 | 89,00 |
| drink | 26,00 | 20,00 | 25,00 | 77,00 |
| cloth | 56,00 | 30,00 | 32,00 | 60,00 |
| shoes | 45,00 | 50,00 | 32,00 | 51,00 |

| | | 2003 | 2004 | 2005 | 2006 |
|---------|-------|-------|-------|-------|-------|
| Tours | drink | 77,00 | 54,00 | 55,00 | 33,00 |
| | food | 89,00 | 61,00 | 30,00 | 41,00 |
| Orleans | drink | 25,00 | 50,00 | 49,00 | 32,00 |
| | food | 33,00 | 44,00 | 59,00 | 27,00 |

Problématique

Pas de travaux sur la personnalisation de requêtes OLAP

Particularités des requêtes OLAP :

- ▶ la requête porte sur des faits à différents niveaux de détail
- ▶ la requête comporte une partie mise en forme du résultat
 - ▶ qu'est-ce qu'un profil utilisateur pour ce type de requêtes?
 - ▶ peut-on calculer automatiquement la visualisation des faits préférés?
- ▶ ne pas accéder à la table des faits pour personnaliser

Objectif

Étant donné :

- ▶ une **requête** MDX q
- ▶ des **préférences** utilisateur P
- ▶ une **contrainte** de visualisation v

calculer q' telle que :

- ▶ $q' \subseteq q$
- ▶ q' est la plus proche de q satisfaisant v
- ▶ q' est la plus intéressante au regard de P

trouver $q' \in \max_{\subseteq_P} \{q'' \subseteq q \mid v(q'') = true\}$

Les requêtes MDX

Q_{MDX} les requêtes *MDX* de forme :

| | | |
|--------|------------|-------------|
| SELECT | SX_1 | ON AXIS(1), |
| | ... | |
| | SX_K | ON AXIS(K) |
| FROM | C | |
| WHERE | SX_{K+1} | |

soit $q \in Q_{MDX}$, pour $i \in [1, \dots, K + 1]$,

- ▶ $SX_i(q)$ est **une requête de l'algèbre relationnelle**
- ▶ $S_i(q)$ est **le schéma** de cette requête
- ▶ $ref_i(q)$ est **la réponse** à cette requête

| | | |
|------------|-------------|---|
| schéma | $sch(q)$ | $= \bigcup_{i \in [1, \dots, K+1]} S_i(q)$ |
| structure | $struct(q)$ | $= \langle S_1(q), \dots, S_K(q) \rangle$ |
| références | $ref(q)$ | $= ref_1(q) \times \dots \times ref_{K+1}(q)$ |

Exemple

SalesCube de schéma $\langle Product(Item, Category), Time(Day, Month, Year), Location(City, Region, Country), Measure(name), sales(Item, Day, City, name, val) \rangle$

$q =$

```

SELECT    [City].members ON COLUMNS
          CROSSJOIN({[Year].2005,[Year].2006},{[Product].members}) ON ROWS
FROM      SalesCube
WHERE     [Measure].quantity
  
```

- ▶ $sch(q) = \{Location, Time, Product, Measure\}$
- ▶ $struct(q) = \langle \{Location\}, \{Time, Product\} \rangle$
- ▶ $ref(q) = \pi_{City}(Location) \times \{2005, 2006\} \times (\pi_{Category}(Product) \cup \pi_{Item}(Product)) \times \{quantity\}$

Les préférences

Les préférences consistent en :

- ▶ un **ordre partiel sur les dimensions** :

$D <_d D'$ si D' préférée à D

- ▶ pour chaque dimension D_i , un **ordre partiel sur les membres** : $m <_i m'$ si m' préféré à m

Les préférences

Les préférences consistent en :

- ▶ un **ordre partiel sur les dimensions** :
 $D <_d D'$ si D' préférée à D
- ▶ pour chaque dimension D_i , un **ordre partiel sur les membres** : $m <_i m'$ si m' préféré à m

On en déduit un **ordre sur les références** :

- ▶ $\Delta(t, t')$ les dimensions où t et t' **diffèrent**
- ▶ $M = \max_{<_d}(\Delta(t, t'))$ les dimensions **préférées** où t et t' diffèrent
- ▶ $t \leq t'$ si $\forall D_i \in M, t(D_i) <_i t'(D_i)$

Ordre sur les requêtes

L'ordre sur les références fournit un **ordre partiel sur les requêtes** : pour $q, q' \in Q_{MDX}$:

$q \preceq q'$ si

- ▶ $sch(q) = sch(q')$ et
- ▶ $\forall t \in ref(q), \exists t' \in ref(q'), t \preceq t'$

$q \sqsubseteq q'$ si

- ▶ $sch(q) = sch(q')$ et
- ▶ quelque soit l'instance de C , $ref(q) \subseteq ref(q')$

L'ordre sur les requêtes est tel que si $q \sqsubseteq q'$ alors $q \preceq q'$

Exemple

Soient les préférences :

- ▶ $Location <_d Time$ et $Product <_d Time$
- ▶ $2005 <_{Time} 2006$, $drink <_{Product} food$

$q_1 =$
 SELECT {[Year].2005} ON COLUMNS
 ({[Category].food, [Category].drink},
 {[City].Tours}) ON ROWS
 FROM SalesCube
 WHERE [Measure].quantity

↔

$q_2 =$
 SELECT {[Category].drink} ON COLUMNS
 ({[Year].2006}, {[City].Tours})
 ON ROWS
 FROM SalesCube
 WHERE [Measure].quantity

Exemple

Soient les préférences :

- ▶ $Location <_d Time$ et $Product <_d Time$
- ▶ $2005 <_{Time} 2006$, $drink <_{Product} food$

| | | |
|---|---|--|
| $q_1 =$ SELECT {[Year].2005} ON COLUMNS ({[Category].food, [Category].drink}, {[City].Tours}) ON ROWS FROM SalesCube WHERE [Measure].quantity | } | $q_2 =$ SELECT {[Category].drink} ON COLUMNS ({[Year].2006}, {[City].Tours}) ON ROWS FROM SalesCube WHERE [Measure].quantity |
|---|---|--|

car :

$$ref(q_1) = \{ \langle 2005, food, Tours, quantity \rangle, \langle 2005, drink, Tours, quantity \rangle \}$$

$$ref(q_2) = \{ \langle 2006, drink, Tours, quantity \rangle \}$$

et

$$\langle 2005, food, Tours, quantity \rangle \preceq \langle 2006, drink, Tours, quantity \rangle$$

$$\langle 2005, drink, Tours, quantity \rangle \preceq \langle 2006, drink, Tours, quantity \rangle$$

Contraintes

Une contrainte v est :

- ▶ une **fonction booléenne** définie sur Q_{MDX}
- ▶ **anti-monotone** si $\forall q, q' \in Q_{MDX}$ si $q \sqsubseteq q'$ et $v(q') = true$ alors $v(q) = true$

Utilisation d'un algorithme par niveau pour calculer le plus grand ensemble de références satisfaisant v

Contraintes

Exemple de contrainte de visualisation anti-monotone :

- ▶ pour une structure $T = \langle T_1, \dots, T_K \rangle$
- ▶ et un tuple d'entiers $G = \langle G_1, \dots, G_K \rangle$
- ▶ $v_{T,G}(q) = true$ si
 - ▶ q contient K axes
 - ▶ les dimensions de T_k sont affichées sur l'axe k
 - ▶ l'axe k ne comporte pas plus de G_k positions

Le principe

Contexte :

- ▶ les tables de dimensions sont en mémoire
- ▶ pas d'accès à la table de faits

Pour une **requête** q , des **préférences** P et une **contrainte** v :

1. calcul des sous-ensembles de $ref(q)$
 - ▶ les plus grands possibles
 - ▶ visualisables selon v
 - ▶ contenant les références préférées selon P
2. calcul des structures permettant de visualiser ces ensembles

L'algorithme

Les références de $ref(q)$ peuvent être ordonnées des plus préférées (R_1) au moins préférées (R_n): $ref(q) = R_1 \cup \dots \cup R_n$

1. $M_0 = \{\emptyset\}$
2. pour i de 1 à n faire
 - 2.1 prendre R_i les préférées à l'étape i
 - 2.2 pour chaque R de M_{i-1}
 - ▶ ajouter à M_i l'ensemble $R \cup R'_i$ tel que:
 - ▶ R'_i est le plus grand sous-ensemble de R_i tel que
 - ▶ $R \cup R'_i$ satisfait v
3. pour chaque R de M_n
 - ▶ calculer les structures S permettant de visualiser R
4. construire les requêtes MDX à partir des couples $\langle R, S \rangle$

Exemple

La requête :

```
SELECT    {[City].Tours, [City].Orleans},
          {[Year].2003,[Year].2004,[Year].2005,[Year].2006},
          {[Category].shoes,[Category].cloth,[Category].food,[Category].drink}
FROM      SalesCube
WHERE     [Measures].quantity
```

Les préférences :

(Time <_d Location) et (Product <_d Location)
 2002 <_{Time} 2003 <_{Time} 2004 <_{Time} 2005 <_{Time} 2006
 electronics <_{Product} shoes <_{Product} cloth <_{Product} food <_{Product} drink
 (Centre <_{Location} Tours) et (Centre <_{Location} Orleans)
 quantity <_{Measures} price

La contrainte de visualisation : $v_{T,G} : T = \langle \emptyset, \emptyset \rangle$ et $G = \langle 4, 4 \rangle$

Exemple

Etape 1 :

$$R_1 = \{ \langle 2006, \textit{drink}, \textit{Orleans}, \textit{quantity} \rangle, \langle 2006, \textit{drink}, \textit{Tours}, \textit{quantity} \rangle \}$$

| | | |
|-------|---------|------|
| | | 2006 |
| drink | Orleans | |
| | Tours | |

Exemple

Etape 2 :

$$R_2 = \{ \langle 2006, \text{food}, \text{Orleans}, \text{quantity} \rangle, \langle 2006, \text{food}, \text{Tours}, \text{quantity} \rangle, \langle 2005, \text{drink}, \text{Orleans}, \text{quantity} \rangle, \langle 2005, \text{drink}, \text{Tours}, \text{quantity} \rangle \}$$

| | | 2006 | 2005 |
|-------|---------|------|------|
| drink | Orleans | | |
| | Tours | | |
| food | Orleans | | |
| | Tours | | |

Exemple

Etape 3 :

$R_3 =$

$\{ \langle 2005, \text{food}, \text{Orleans}, \text{quantity} \rangle, \langle 2004, \text{drink}, \text{Orleans}, \text{quantity} \rangle, \langle 2005, \text{food}, \text{Tours}, \text{quantity} \rangle, \langle 2004, \text{drink}, \text{Tours}, \text{quantity} \rangle, \langle 2006, \text{cloth}, \text{Orleans}, \text{quantity} \rangle, \langle 2006, \text{cloth}, \text{Tours}, \text{quantity} \rangle \}$

or on ne peut pas visualiser 3 années, 3 catégories et 2 villes

| | | 2006 | 2005 | 2004 |
|-------|---------|------|------|------|
| drink | Orleans | | | |
| | Tours | | | |
| food | Orleans | | | |
| | Tours | | | |

| | | food | drink | cloth |
|---------|------|------|-------|-------|
| Tours | 2005 | | | |
| | 2006 | | | |
| Orleans | 2005 | | | |
| | 2006 | | | |

Exemple

Etc.

Au final une des requêtes générées est :

```
SELECT    {[Year].2003,[Year].2004,[Year].2005,[Year].2006} ON COLUMNS
          CROSSJOIN({[City].Tours,[City].Orleans},
                    {[Category].food,[Category].drink}) ON ROWS
FROM      SalesCube
WHERE     [Measures].quantity
```

| | | 2003 | 2004 | 2005 | 2006 |
|---------|-------|-------|-------|-------|-------|
| Tours | drink | 77,00 | 54,00 | 55,00 | 33,00 |
| | food | 89,00 | 61,00 | 30,00 | 41,00 |
| Orleans | drink | 25,00 | 50,00 | 49,00 | 32,00 |
| | food | 33,00 | 44,00 | 59,00 | 27,00 |

Conclusion et perspectives

- ▶ Contribution :
 - ▶ un cadre pour la personnalisation de requêtes OLAP
 - ▶ développement d'un prototype pour applications mobiles

Conclusion et perspectives

- ▶ Contribution :
 - ▶ un cadre pour la personnalisation de requêtes OLAP
 - ▶ développement d'un prototype pour applications mobiles
- ▶ Perspectives :
 - ▶ utiliser plutôt un langage clos !
 - ▶ ajouter au profil des préférences sur la structure
 - ▶ prendre en compte des contraintes telles que :
 - ▶ le coût d'évaluation de la requêtes
 - ▶ la cardinalité de la réponse