

# OLAP queries optimization: A framework for combining Rule-Based and Cost-Based approaches



**H. Mouloudi - A. Giacometti - P. Marcel**

LI - Université François-Rabelais de Tours

**L. Bellatreche** LISI ENSMA - Université de Poitiers

**D. Laurent** LICP - Université de Cergy-Pontoise

# Optimisation des requêtes OLAP

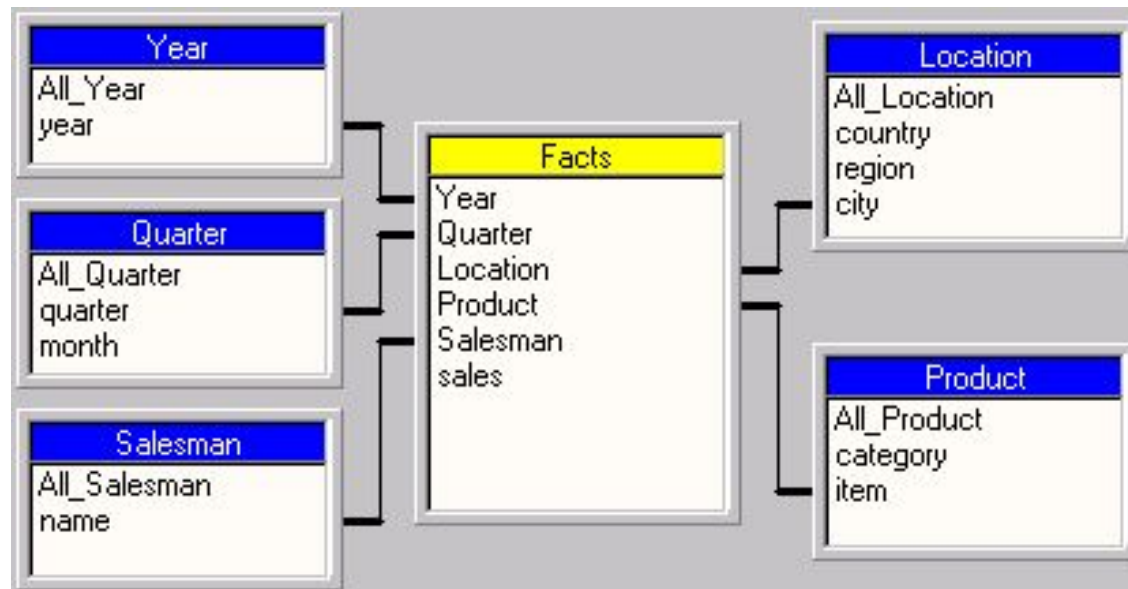
## Existant :

- CBA : Optimisation à base de coût  
(Index, Vues matérialisées, Partitionnement)

## Contribution :

- RBA : Optimisation à base de règles  
(Algèbre multidimensionnelle, Règles de réécriture)
- VBA : Optimisation basée sur la visualisation  
(Calcul des faits visibles par l'utilisateur)

# Motivation : Cube des ventes



Cube des ventes sous forme de schéma en étoile

## Motivation : Requête MDX

```
WITH MEMBER drink.Mydrink AS 'wine + beer'  
SELECT  
  {[quarter].MEMBERS} ON COLUMNS,  
  CROSSJOIN([Name].MEMBERS, [North].CHILDREN) ON ROWS,  
  year.MEMBERS ON PAGES,  
  {Mydrink, Food} ON SECTIONS  
FROM SalesCube  
WHERE [sales]
```

- ▷ Les faits résultats de cette requête ne seront pas tous visibles

# Motivation : Visualisation

<b>A<sub>1</sub></b>								<b>A<sub>3</sub></b>
bill	north	lille	20	30	40	30		<b>drink</b>
		blois	20	10	10	20		↑
		paris	10	40	30	50		beer,wine
⋮								
john	north	lille	70	60	50	50		<b>A<sub>4</sub></b>
		blois	20	20	30	30		<b>1988</b>
		paris	10	20	10	10		
<b>A<sub>2</sub></b>			<i>q</i> <sub>1</sub>	<i>q</i> <sub>2</sub>	<i>q</i> <sub>3</sub>	<i>q</i> <sub>4</sub>		

- ▷ Comment identifier la partie visible de la réponse à une requête OLAP?

# Modèle de cube

$C = \langle \mathcal{D}, F, S \rangle$ :

$\mathcal{D}$  : l'ensemble des tables de **dimensions**

$F$  : la table des **faits** aux niveaux auxquels on voit les données

$S$  : la **structure** du cube  $S = \langle K, axis, pos, depth \rangle$

$K$  : nombre d'axes

$axis(k)$  : dimensions sur l'axe  $k$  et l'ordre d'imbrication

$pos(L)$  : ordre des membres de l'attribut  $L$  sur un axe

$depth(D)$  : niveau de  $D$  auquel on voit les données

## Exemple de structure

$A_1$	bill	north	lille	20	30	40	30	$A_3$
			blois	20	10	10	20	<b>drink</b>
			paris	10	40	30	50	↑
			⋮					<u>beer,wine</u>
	john	north	lille	70	60	50	50	$A_4$
			blois	20	20	30	30	<b>1988</b>
			paris	10	20	10	10	
$A_2$				$q_1$	$q_2$	$q_3$	$q_4$	

$$S = \langle 4, axis, pos, depth \rangle$$

$$axis(1) = (\{Salesman, Location\}, \leq_1), Salesman \leq_1 Location$$

$$depth(Product) = category$$

$$pos(city) = \leq \text{ avec } Lille \leq Blois \leq Paris$$

# Opérateurs de l'algèbre

	Opérateur	modifie		
		<i>D</i>	F	S
Restructuration	Permute			✓
	Switch			✓
	Nest			✓
	Order by			✓
	Project members	✓		✓
Granularité	Aggregate		✓	✓
Filtrage	Select members	✓	✓	✓
	Select measures		✓	
	Project measures		✓	



# Règles de réécriture

**Exemple** : Commutation de la sélection sur les membres et de l'agrégation

- ▷ Garantie par le fait que la sélection peut se faire uniquement :
  - au niveau où on voit les données **ou**
  - au niveau supérieur
  
- ▷ Pas de sélection au niveau inférieur car remet en cause les agrégats

Dans une optimisation **RBA** :

- pousser les sélections avant les agrégations
- ramener en MC moins de faits avant le calcul des agrégats

# Optimisation VBA

## Hypothèse : tables de **dimensions** et **structure** en **mémoire**

**Entrée :** Un cube  $C$  et une requête OLAP  $q$

**Sortie :** Une condition de sélection  $\varphi$  telle que  $\sigma_{\varphi}(q(C))$  est la tranche visible de  $q(C)$

---

1. **Calcul** de la structure  $S'$  du cube  $C' = q(C)$
2. **Calcul** des dimensions  $\mathcal{D}'$  du cube  $C' = q(C)$
3. Pour chaque dimension  $D$  cachée
4.     Pour le niveau  $L$  de  $D$  auquel où on voit les données
5.         **Recherche** du membre  $v$  de ce niveau à la plus petite position
6.         **Ajout** à  $\varphi$  de  $L = v$
7. **Retourner**  $\varphi$

## Exemple d'optimisation VBA

$A_1$								$A_3$
bill	north	lille	20	30	40	30		<b>drink</b>
		blois	20	10	10	20		↑
		paris	10	40	30	50		<u>beer,wine</u>
⋮								
john	north	lille	70	60	50	50		$A_4$
		blois	20	20	30	30		<b>1988</b>
		paris	10	20	10	10		
$A_2$			$q_1$	$q_2$	$q_3$	$q_4$		

Deux dimensions cachées : **Product** sur  $A_3$  et **Year** sur  $A_4$

Conditions ajoutées : (**category = drink**) et (**year = 1988**)

# Plan général d'optimisation

▷ L'optimisation consiste à calculer le plus efficacement possible (RBA et CBA) la partie visible du résultat d'une requête (VBA)

$$\begin{array}{rcl}
 q(C) = \langle q_1(\mathcal{D}), q_2(F, \mathcal{D}), q_3(S) \rangle & & \text{VBA} \\
 \downarrow & & \\
 \sigma_{\varphi}^{member}(q(C)) & & \\
 \downarrow & & \text{RBA} \\
 q'_r(C) = \langle q'_1(\mathcal{D}), q'_2(F, \mathcal{D}), q'_3(S) \rangle & & \\
 \downarrow & & \text{CBA} \\
 F' = q'_2(F, D) & & 
 \end{array}$$

## **CBA : mode d'utilisation**

**Contexte** : VBA ajoute des conditions de sélection aux requêtes

**Utilisation** : comment exploiter ces conditions supplémentaires  
pour réduire le coût d'E/S ?

**Optimisation** : utiliser les index de jointure :

- sur les attributs des conditions supplémentaires
- si les facteurs de sélectivité de ces conditions sont importants

## Conclusion & Perspectives

- Définition d'un modèle de cube
- Définition d'une algèbre et des règles de réécriture
- Technique d'optimisation visuelle
  
- Implémentation de notre approche
- Optimisation d'une séquence de requêtes
- Sélection des index de jointure dirigée par VBA

# Domaine de définition de la sélection

▷ Les axes montrent à partir de quels membres les agrégats sont calculés

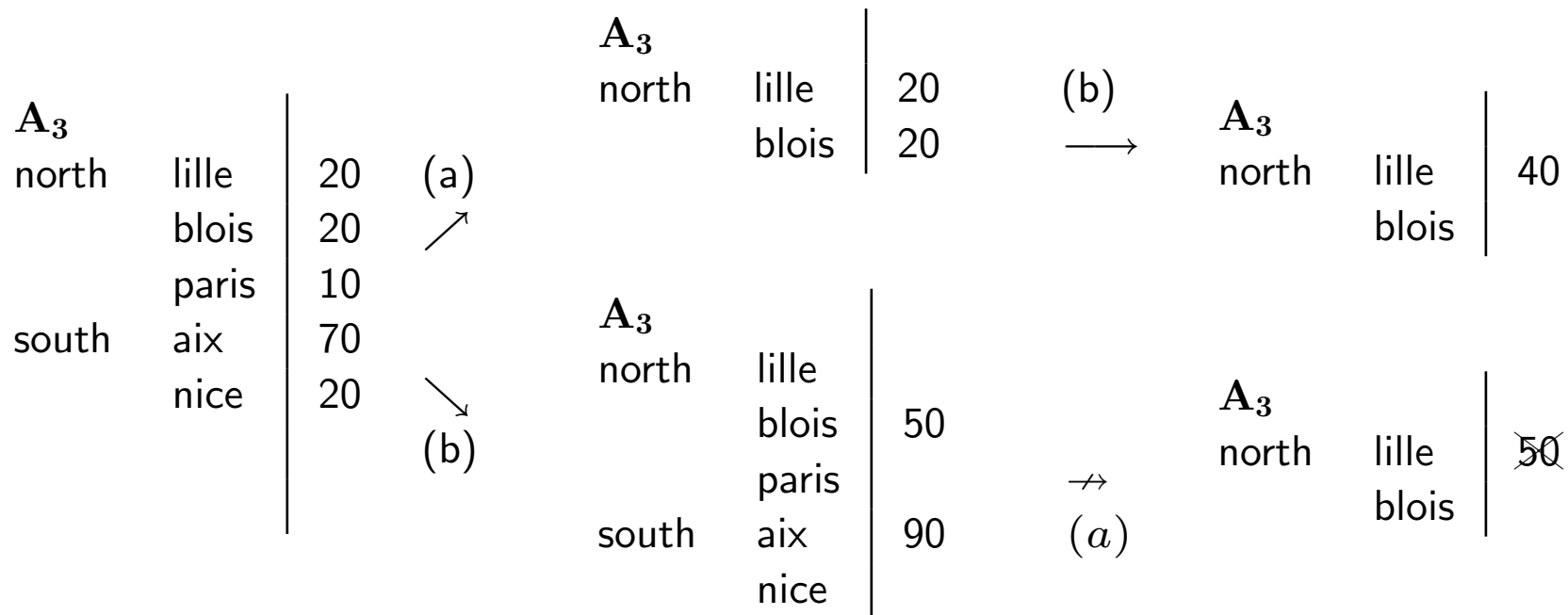
$A_3$	north	lille	20	(a) ↗	$A_3$	north	lille	20	(b) →	$A_3$	north	lille	40
		blois	20					blois		20			
	south	paris	10										
		aix	70										
		nice	20										

(a) :  $\sigma_{ville=lille \vee ville=blois}^{membre}$

(b) :  $Aggregate_{city \rightarrow region; sum(qty)}$

# Domaine de définition de la sélection

▷ La sélection ne doit pas nécessiter de nouvelles opérations d'agrégation



(a) :  $\sigma_{ville=lille \vee ville=blois}^{membre}$

(b) :  $Aggregate_{city \rightarrow region; sum(qty)}$