



Laboratoire d'Informatique Scientifique et Industrielle  
École Nationale Supérieure de Mécanique et d'Aérotechnique  
1, avenue Clément Ader - BP 40109 - 86961 Futuroscope cedex - France

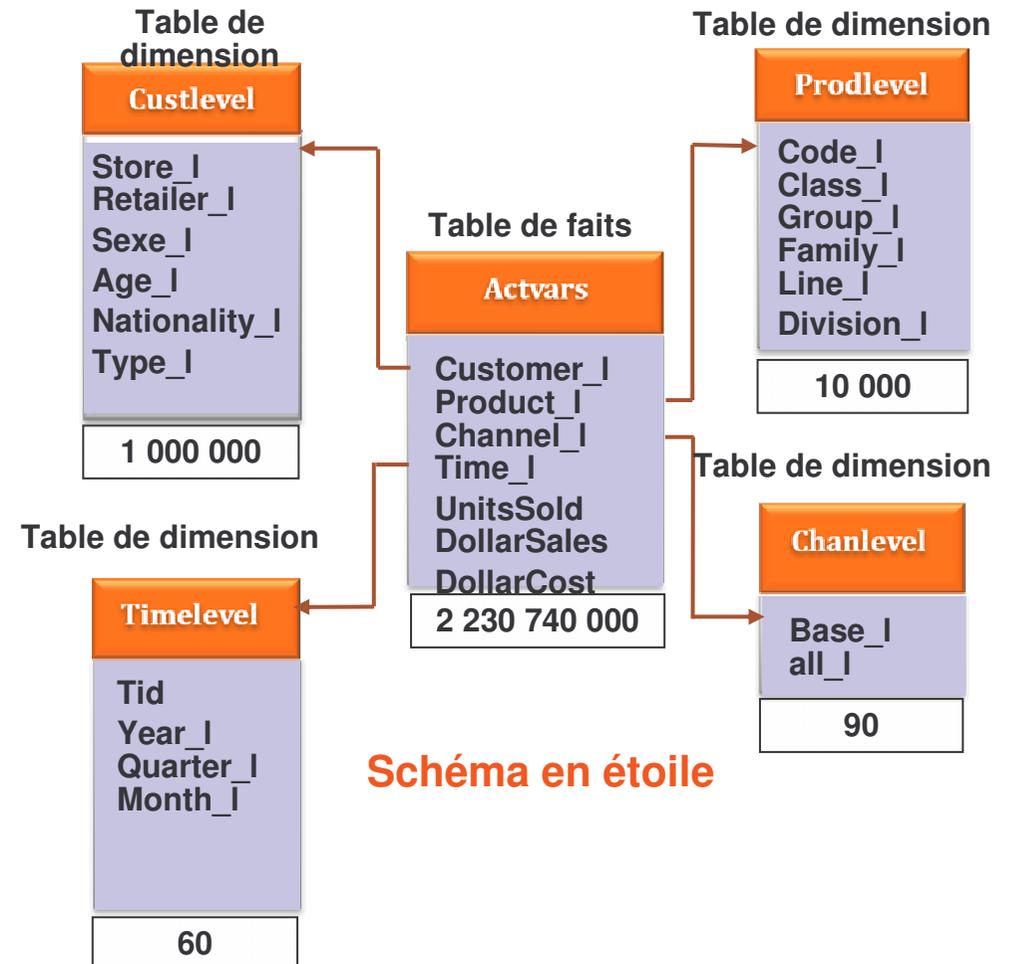


## **Fragmentation Primaire et Dérivée: Étude de Complexité, Algorithmes de Sélection et Validation sous ORACLE10g**

Kamel Boukhalfa, Ladjel Bellatreche et Pascal Richard  
Université de Poitiers - LISI/ENSMA  
{boukhalk, bellatreche, richardp}@ensma.fr

# Contexte: Optimisation des entrepôts de données relationnels

- ❑ Schéma en étoile
  - ❑ Requêtes de Jointure en Etoile (RJE)
    - ❑ Jointures entre la table des faits et les tables de dimension
    - ❑ Sélections sur les tables de dimension
- Fragmentation horizontale est bien adaptée pour les requêtes de jointure en étoile**



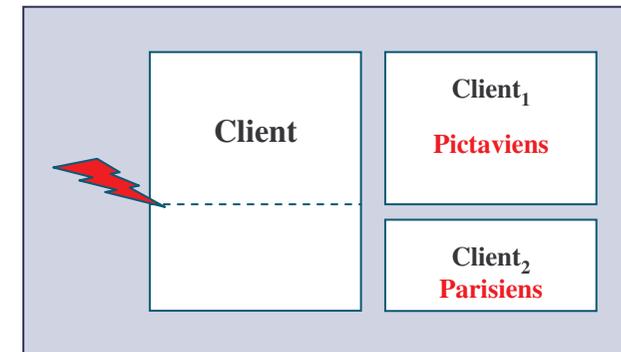
# Fragmentation horizontale et les requêtes de jointure en étoile (I)

## □ Fragmentation horizontale primaire (FHP)

- Fragmenter une table en utilisant les prédicats de sélection définis sur cette table

**Prédicat** : Attribut  $\theta$  Valeur,  $\theta \in \{=, <, \leq, >, \geq\}$  et valeur  $\in$  Domaine(Attribut).

- Exemple: Client (Client\_id, Nom, Ville)
  - Client<sub>1</sub> :  $\sigma_{\text{Ville}='Poitiers'}(\text{Client})$
  - Client<sub>2</sub> :  $\sigma_{\text{Ville}='Paris'}(\text{Client}),$



## □ Impact de la FHP sur les RJE

- Optimisation des sélections définies sur les tables de dimension

# Fragmentation horizontale et les requêtes de jointure en étoile (II)

## ❑ Fragmentation horizontale dérivée (FHD)

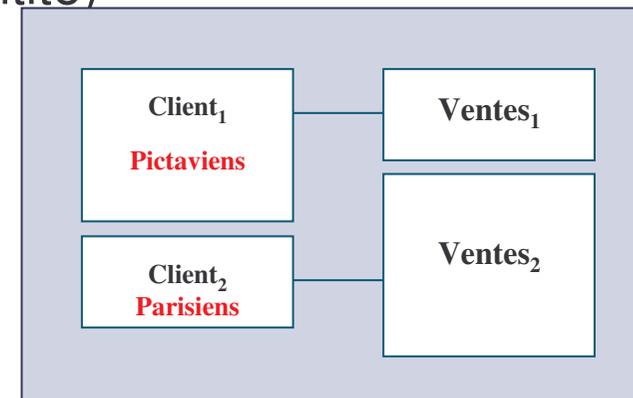
❑ Fragmenter une table (S) selon des attributs d'une autre table (T) : (existence de lien entre S et T)

– Ventes(Client\_id, Produit\_id, Date, Quantité)

–  $Ventes_1 = Ventes \leftrightarrow Client_1$

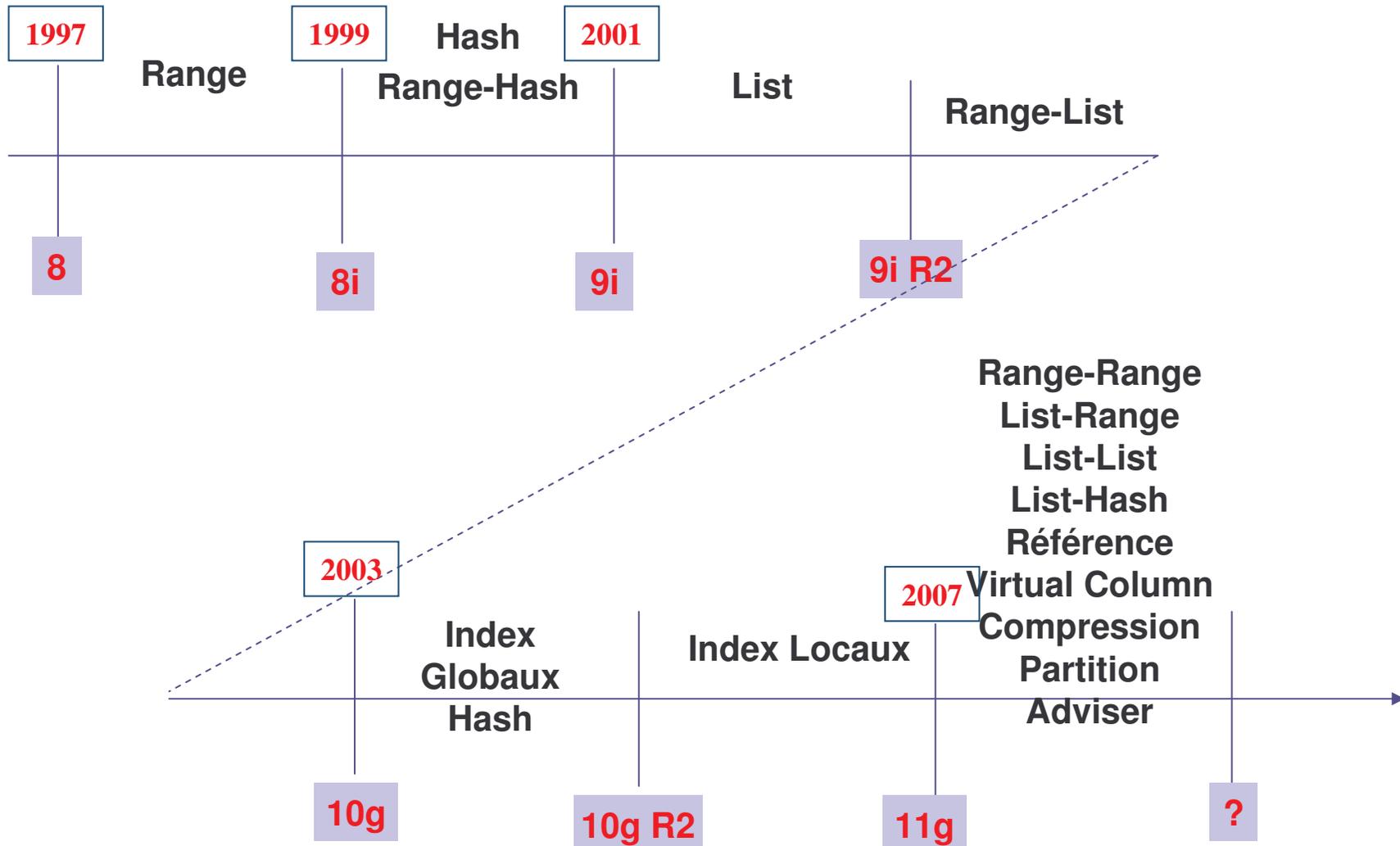
–  $Ventes_2 = Ventes \leftrightarrow Client_2$

## ❑ Impact de la FHD sur RJE



❑ Optimisation de la jointure entre S et T

# Forte adoption de la FH par les SGBD commerciaux



# PLAN

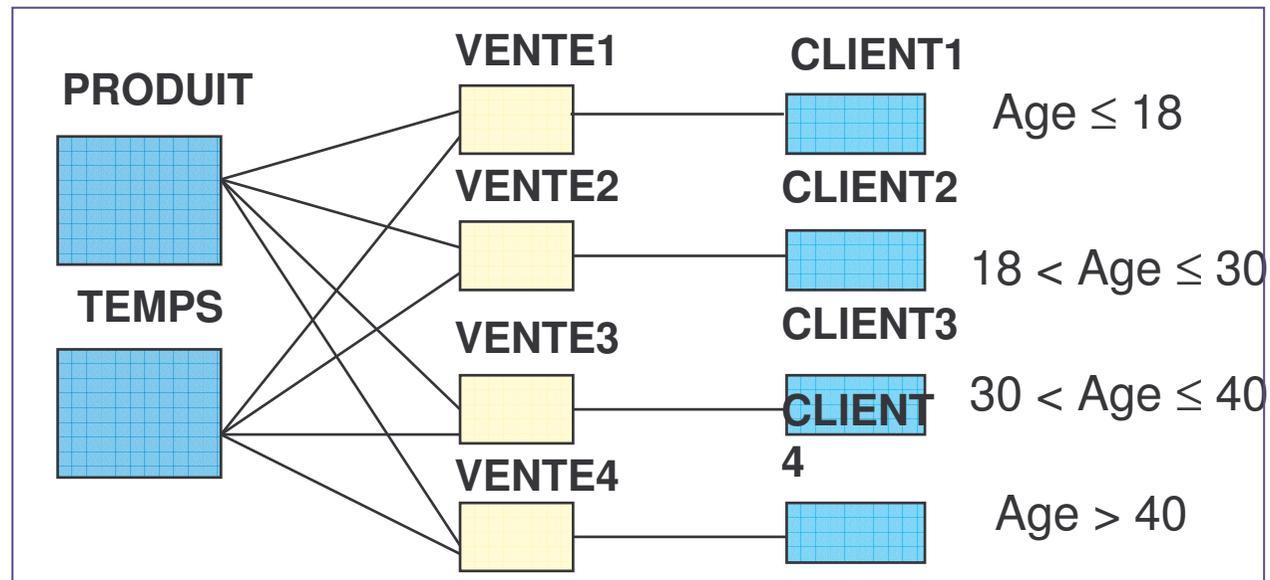
---

- Démarche de fragmentation d'un entrepôt
- Problème de sélection d'un schéma de fragmentation
  - Formalisation
  - Complexité
- Algorithme de Hill Climbing
- Expérimentations
- Validation sous Oracle
- Conclusion et perspectives

# Notre démarche de fragmentation

1. Fragmenter (virtuellement/physiquement) des tables de dimension en utilisant la fragmentation primaire
2. Fragmenter la tables des faits (en utilisant les schémas de fragmentation des tables de dimension)

## Exemple



## Explosion du nombre de fragments

$$N = \prod_{i=1}^k M_i$$

$M_i$  : nombre de fragments de la table de dimension  $D_i$   
 $k$  : nombre de tables de dimension fragmentées

# Représentation d'un schéma de fragmentation

- ❑ Décomposition des domaines des attributs de fragmentation en sous domaines
- ❑ Codage d'un schéma de fragmentation

## Exemple

Trois attributs de dimension:  
Age, Genre, Saison

	Age<18	18-30	30-45	45-60	>60
Age	0	1	0	1	2
Genre	M		F		
	0	0			
Saison	été		automne	hiver	printemps
	0	0	0	1	
	Partition P0			Partition P1	

Les tables sont fragmentées comme suit

- ❑ Table Client en 3 fragments sur Age (Genre n'est pas utilisé)

- Client1 : Age <18 **OU**  $30 \leq \text{Age} < 45$
- Client2 :  $18 \leq \text{Age} < 30$  **OU**  $45 \leq \text{Age} < 60$
- Client 3 : Age  $\geq 60$



Table Ventes fragmentée en :  
**3x2=6 fragments**

- ❑ Table temps en 2 fragments sur Saison

- Temps 1 : Saison=été **OU** automne **OU** hiver
- Temps2 : Saison=printemps

# Fragmentation Dirigée par le Nombre de Fragments

## ~~: Formalisation~~

---

### Entrées

- Entrepôt de données composé de
  - Un ensemble  $D$  de tables de dimension  $D = \{D_1, D_2, \dots, D_d\}$
  - Une table des faits  $F$
- Charge de requêtes les plus fréquentes  $Q = \{Q_1, Q_2, \dots, Q_m\}$
- $W$  : seuil (fixé par l'administrateur)

### Sorties :

- Ensemble  $D' \subseteq D$  des tables de dimension fragmentées
- Ensemble de  $N$  fragments de faits  $F_1, F_2, \dots, F_N$

### Objectifs :

- Réduire le temps d'exécution de  $Q$
- $N \leq W$

## NP-complétude de notre Problème

---

- Problème de fragmentation horizontale à un seul domaine (PFHSD)
  - Une seule table de dimension
  - Un seul attribut A
- Réduction à partir du problème 3-Partition
  - 3-Partition NP-Complet
  - PFHSD NP-Complet
- Notre problème de fragmentation est plus compliqué
  - Plusieurs tables de dimension
  - Plusieurs attributs par table de dimension

# Algorithme de Sélection: Hill Climbing

## Principe

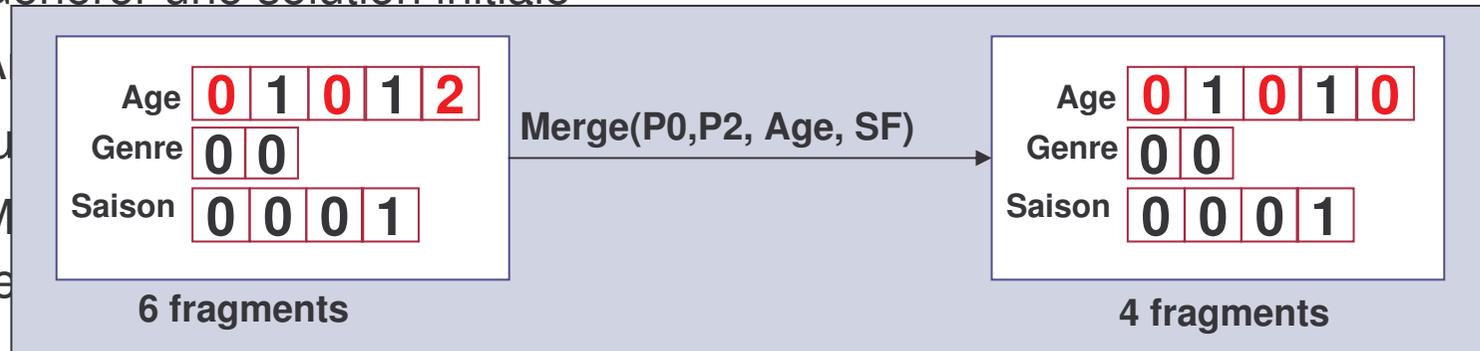
1. Générer une solution initiale

2. A

Mesur

M

re



## Solution initiale

A

Amo

M



Fusionner deux partitions en une seule.

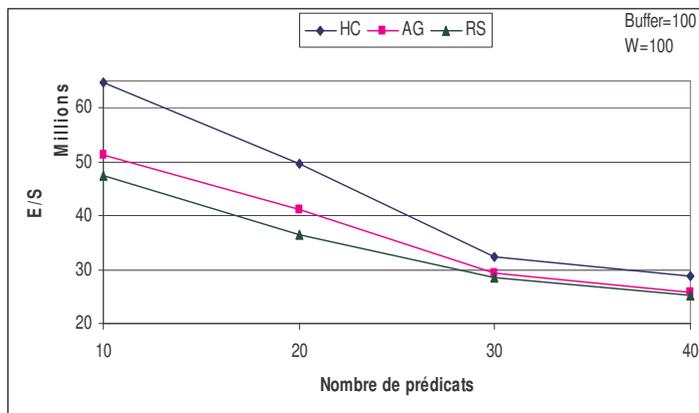
Split

Eclater une partition de sous domaines en deux partitions

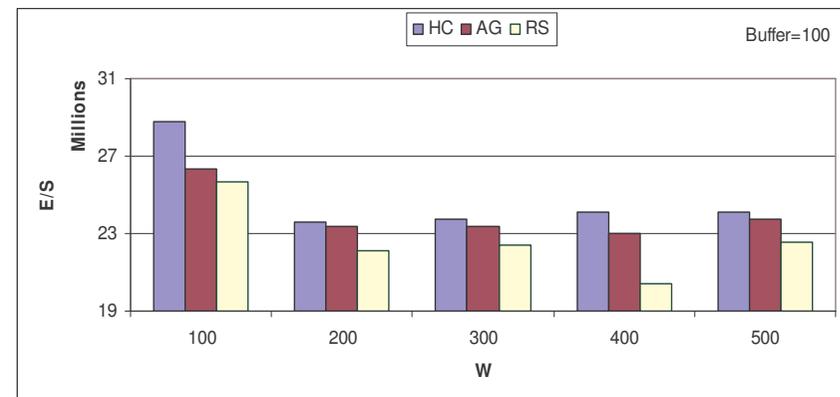
# Expérimentation(I)

1. Théorique : Modèle de coût
2. Validation sous Oracle

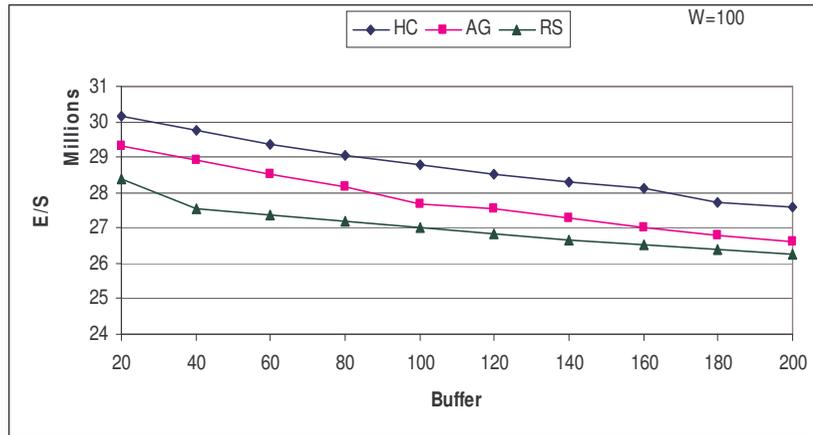
DataSet :	Requêtes :	Algorithmes
Benchmark APB-1	60 requêtes	
<ul style="list-style-type: none"> <li>• 1 Table de faits (24 786 000 )</li> <li>• 4 tables de dimension</li> </ul>	<ul style="list-style-type: none"> <li>• 40 prédicats de sélection</li> <li>• 12 attributs de sélection</li> </ul>	<ul style="list-style-type: none"> <li>• Hill Climbing</li> <li>• Algorithme génétique</li> <li>• Recuit Simulé</li> </ul>



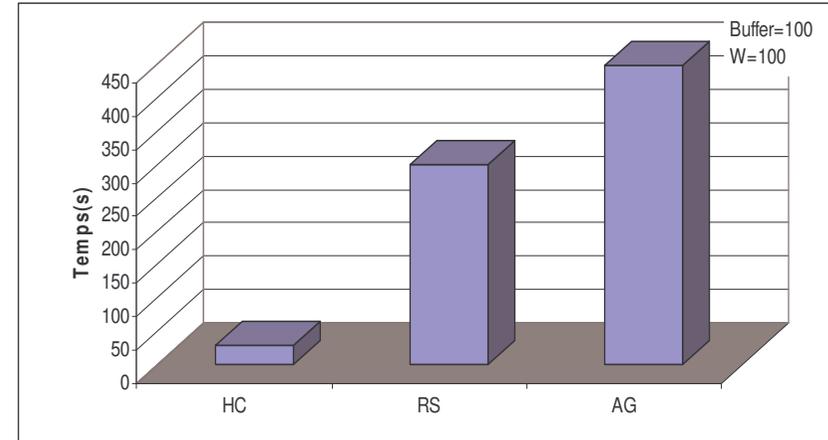
Performance vs Nombre de prédicats



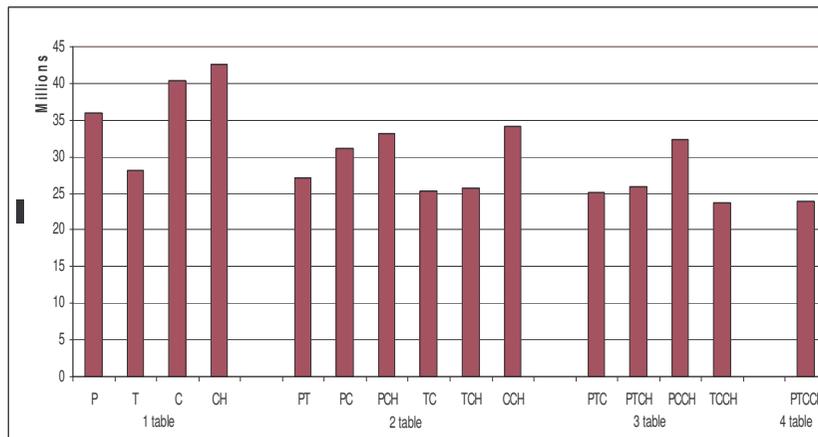
Effet du seuil W



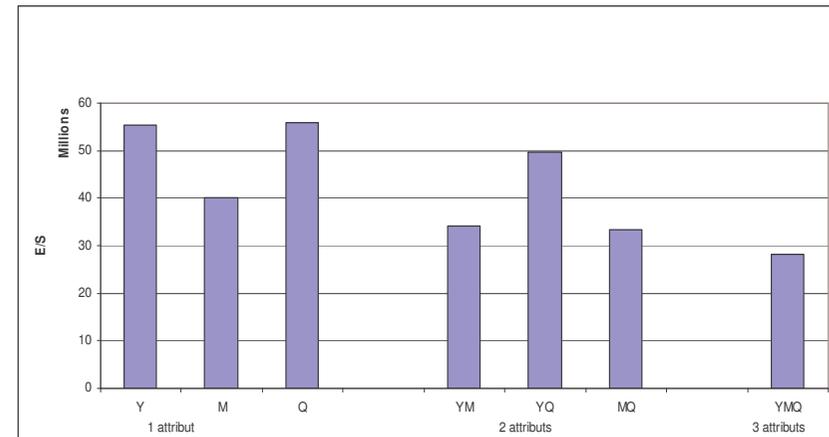
Effet du buffer sur la performance



Temps d'exécution



Choix des tables de dimension



Choix des attributs de dimension

# Validation sous Oracle(I)

---

## ❑ Problèmes rencontrés

- ❑ FHP sur  $n$  ( $n > 2$ ) attributs n'est pas supportée.
- ❑ FHD sur ( $m > 1$ ) tables de dimension n'est pas supportée.

## ❑ Solutions

1. Méthode d'implémentation de la FHP (ajout d'une nouvelle propriété)
  2. Méthode d'implémentation de la FHD (vues matérialisée)
- ⇒ Nécessité de réécrire les requêtes sur les schémas fragmentés
- ❑ Identifier les fragments valides pour chaque requête
  - ❑ Réécrire la requête sur ces fragments

# Validation Sous Oracle(II)

- ❑ Implémentation de la FHP sur plusieurs attributs
  - ❑ Fragmentation de la table Client sur Age, Sexe, Ville
  - ❑ Age : **Age**<26, 26≤Age≤60, Age>60 → **Range**
  - ❑ Sexe : **M**, **F** → **Range-List**
  - ❑ Ville : **Poitiers**, Paris, Nantes → **?**

Client

RID <sup>c</sup>	CID	Nom	Age	Sexe	Ville	Col <sub>c</sub>
6	616	Gilles	15	M	Poitiers	1
5	515	Yves	25	F	Paris	4
4	414	Patrick	33	M	Nantes	9
3	313	Didier	50	M	Nantes	9
2	212	Eric	40	F	Poitiers	10
1	111	Pascal	20	M	Poitiers	1

Clients masculins ayant moins de 26 ans habitant Poitiers

RID <sup>c</sup>	CID	Nom	Age	Sexe	Ville	Col <sub>c</sub>
6	616	Gilles	15	M	Poitiers	1
1	111	Pascal	20	M	Poitiers	1

RID <sup>c</sup>	CID	Nom	Age	Sexe	Ville	Col <sub>c</sub>
5	515	Yves	25	F	Paris	4

RID <sup>c</sup>	CID	Nom	Age	Sexe	Ville	Col <sub>c</sub>
4	414	Patrick	33	M	Nantes	9
3	313	Didier	50	M	Nantes	9

RID <sup>c</sup>	CID	Nom	Age	Sexe	Ville	Col <sub>c</sub>
2	212	Eric	40	F	Poitiers	10



# Validation sous Oracle(II)

- ❑ Implémentation de la FHD en utilisant plus d'une table de dimension

Vente

RID <sup>S</sup>	CID	PID	TiD	Montant	Col <sub>F</sub>
1	616	106	11	25	1-1
2	616	106	66	28	1-2
3	616	104	33	50	1-1
4	515	104	11	10	4-1

RID <sub>C</sub>	CID	N
6	616	G
5	515	Y
4	414	P
3	313	K
2	212	D
1	111	P

Te

RID <sup>T</sup>	TID
6	11
5	22
4	33
3	44
2	55
1	66



```
CREATE MATERIALIZED VIEW V
AS
```

```
SELECT v.CID, v.PID, v.TID, Montant, ColC||'-'||ColT as ColF
FROM Ventes v, Client c, Temps t
WHERE v.CID= c.CID
AND v.TID = t.TID
```

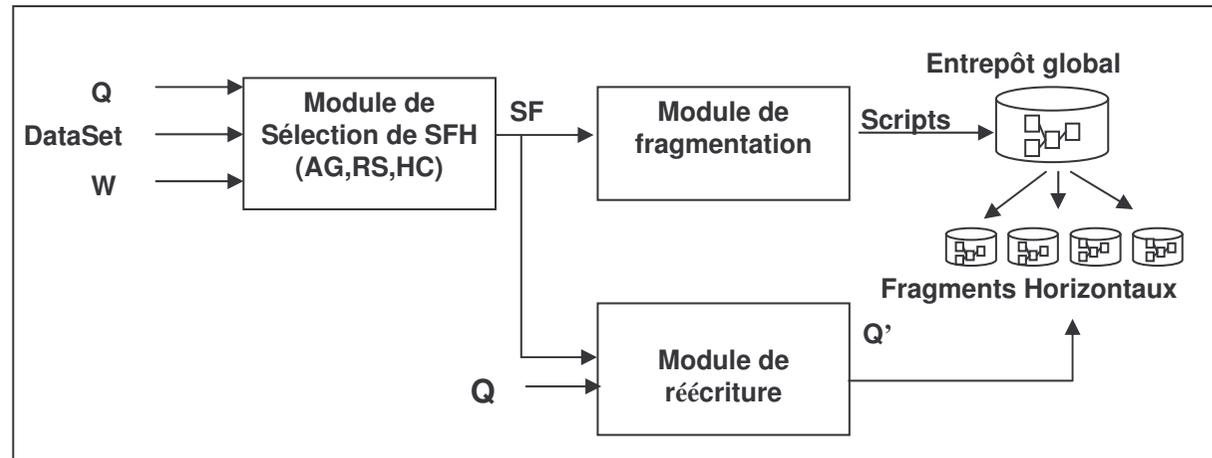
Ventes effectuées durant le premier trimestre par des clients masculins ayant moins de 26 ans et habitant Poitiers

RID <sup>S</sup>	CID	PID	TID	Montan <sup>+</sup>	Col <sub>F</sub>
1	616	106	11	25	1-1
3	616	104	33	50	1-1
8	111	101	33	27	1-1
19	616	104	22	20	1-1

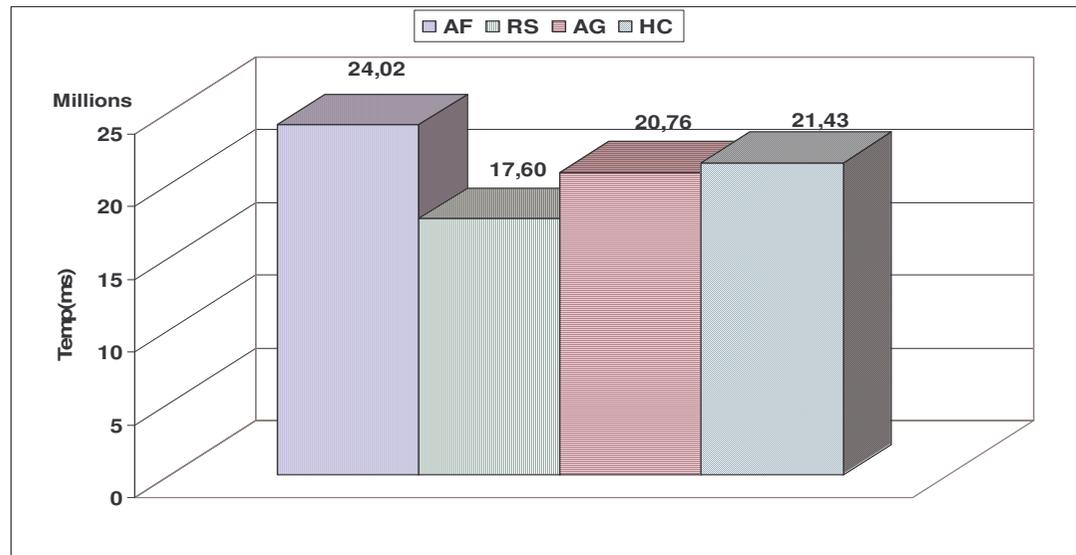
26	313	105	22	17	9-1
27	313	106	11	15	9-1

# Validation Sous Oracle (III)

## Architecture



## Résultats



## Conclusion et perspectives

---

- Proposition d'une approche de fragmentation dans les ED
- Étude de complexité
- Proposition d'un Algorithme Hill Climbing
- Validation Théorique
- Validation sous Oracle
  
- Fragmentation mixte.