

M2 R - Informatique - Montpellier
Fouille de données

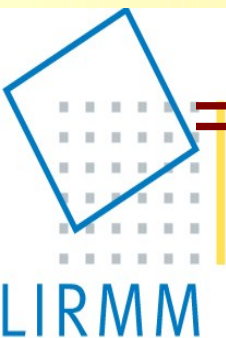
Cours 2 – OLAP - Flou
Anne Laurent

laurent@lirmm.fr

<http://www.lirmm.fr/~laurent>

Objectifs du cours

- Utilisation de la théorie des sous-ensembles flous :
 - BD floues
 - Arbres de décision flous
- SIAD : Systèmes d'Information et d'Aide à la **Décision** :
 - Entrepôts de données
 - OLAP & BD multidimensionnelles



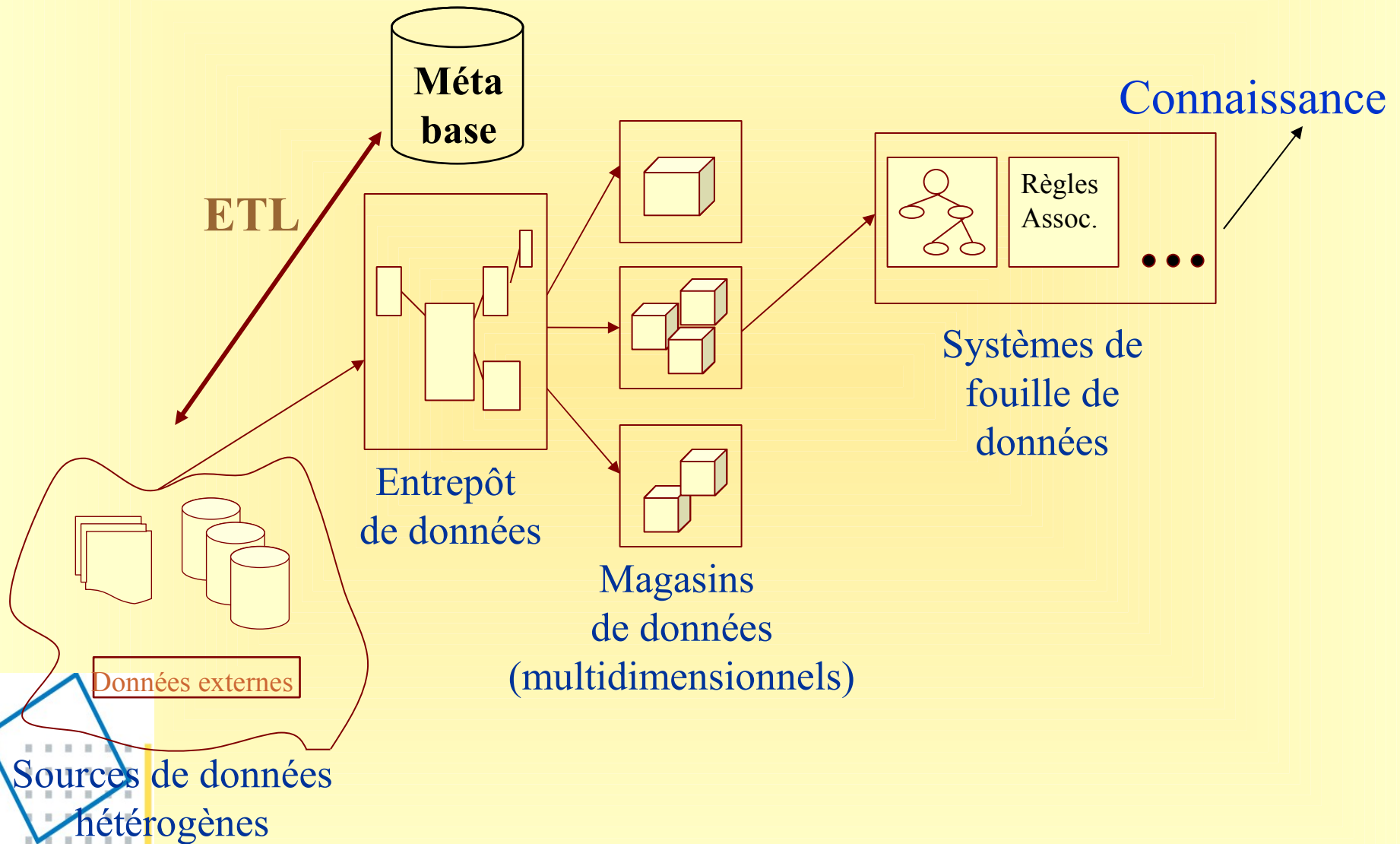
⇒ **BD multidimensionnelles floues, OLAP Mining flou**

Déroulement du cours

- BDM & OLAP
 - OLAP vs. OLTP
 - Bases de données multidimensionnelles
 - Iceberg Cubes
- BDM et data mining : quelques exemples
 - Représentations pertinentes, qualité
 - Recherche de blocs de données
 - Résumés flous de BDM

Bases de données multidimensionnelles et OLAP

Fonctionnement général



ETL : Extract – Transform - Load

Ces outils se chargent principalement de :

- l'extraction (accès aux différentes sources)
- le nettoyage (gestion des inconsistances des données sources)
- la transformation (formats de données etc)
- le chargement
- l'analyse (détection des valeurs non valides)
- l'analyse des méta-données (conception de l'entrepôt de données)

80% du travail !

OLAP (On-Line Analytical Processing)

Solutions OLTP (On Line Transaction Processing) dédiées
aux systèmes *opérationnels*

Non adaptées aux systèmes décisionnels et aux applications OLAP

Terme proposé par Codd (1993)

catégorie d'applications et de technologies permettant de collecter, stocker, traiter et restituer des données multidimensionnelles, à des fins d'analyse.

Requêtes OLAP

Requêtes OLTP : concernent un petit nombre de données et sont très fréquentes

Requêtes OLAP : concernent un grand nombre de données et sont peu fréquentes

Exemples de requêtes OLAP :

- **Requêtes complexes**

Quel est le total des ventes de chaussures ?

Quel est le type des meilleures ventes sur une période donnée, pour une ville précise et un groupe d'âge ?

- **Requêtes *What If***

Que se passerait-il si les prix diminuaient de 10% ?

OLAP vs OLTP

OLAP

vs.

OLTP

(On-Line Analytical Processing)

(On-Line Transaction Processing)

- Requêtes complexes
- Optique décisionnelle
- Vision ensembliste (tendances ...)
- Destiné aux analystes et décideurs (peu nombreux)

- Requêtes simples
- Production et Mise à jour des données
- Vision au niveau individuel
- Destiné aux agents opérationnels (nombreux)

Bases de données multidimensionnelles

Utilisation de *modèles multidimensionnels* par les outils

OLAP pour stocker et présenter l'information :

- Information organisée en hypercubes
- Hypercube : défini sur des dimensions



Modèle multidimensionnel

- OLAP (On-Line Analytical Processing)
- Bases de données multidimensionnelles :

- dimensions \mathcal{D} organisées en hiérarchies
- mesures \mathcal{M}
- ensemble d'hypercubes (*cubes*)

$$\text{Application } D_1 \times \dots \times D_k \rightarrow D_M$$

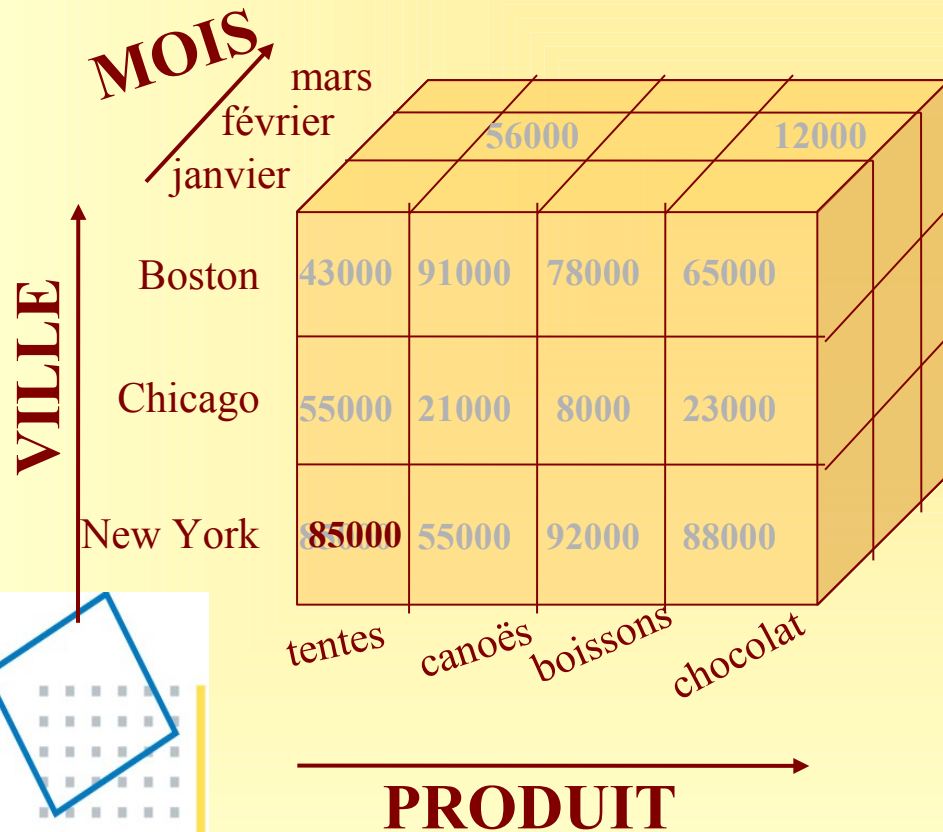
$$\text{avec } \{D_1, \dots, D_k\} \subset \mathcal{D} \text{ et } D_M \subset \mathcal{M}$$

- opérations (sélection, navigation à travers les hiérarchies, ...)

Exemple

Cube des ventes

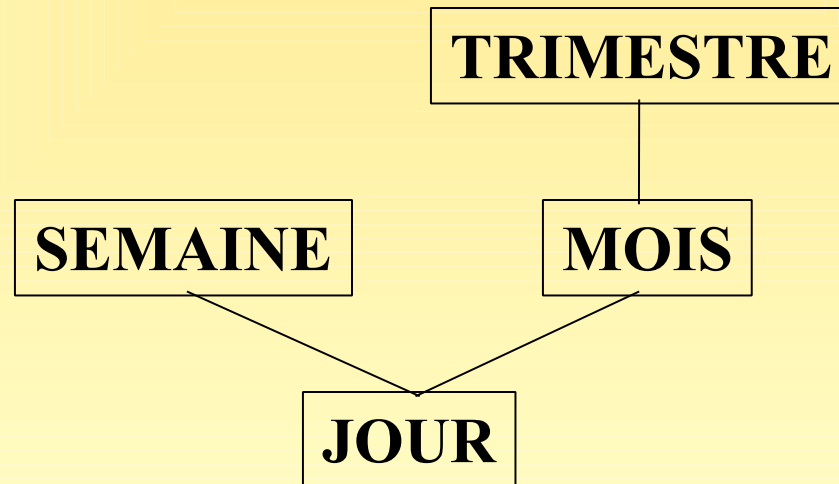
C : Produit x Ville x Mois → Ventes



Dimensions

Organisation des informations selon des dimensions *plates* et des dimensions *hiérarchisées*. Les hiérarchies sont *simples* (arborescentes) ou *complexes* (voir exemple).

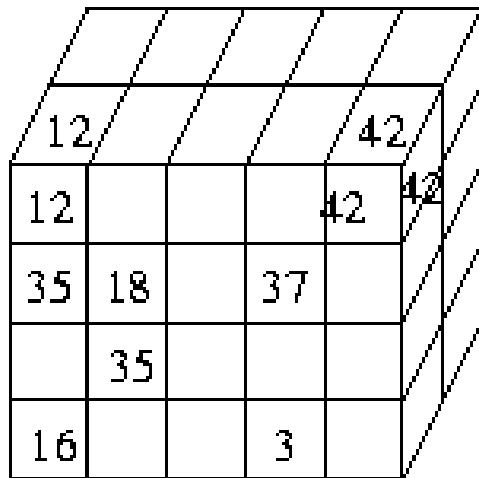
Exemple :



Principales opérations

- Unaires :
 - Visualisation :
 - Rotation
 - Inversion de valeurs sur les dimensions (switch)
 - Modification des données :
 - Sélection sur les cellules (dice)
 - Sélection de tranches (slice)
 - Généralisation/Spécification (Roll-Up/Drill down)
- Binaires :
 - Union
 - Intersection

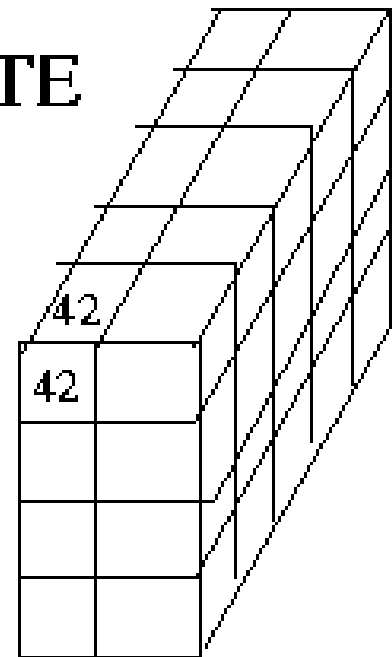
Rotation



A 3D grid representing a 4x5x2 array of numbers. The front face is a 4x5 grid. The top row contains 12 and 42. The second row contains 12 and 42. The third row contains 35, 18, and 37. The fourth row contains 16 and 3. The right face is a 4x2 grid with 42 in the top-right cell.

12				42
12				42
35	18		37	
	35			
16			3	


PIVOT/ROTATE



A 3D grid representing a 4x2x5 array of numbers. The front face is a 4x2 grid. The top row contains 42. The second row contains 42. The other rows are empty. The right face is a 4x5 grid.

42	
42	

Inversion

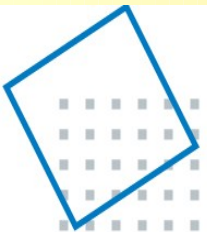


12				42
12				42
35	18		37	
	35			
16			3	

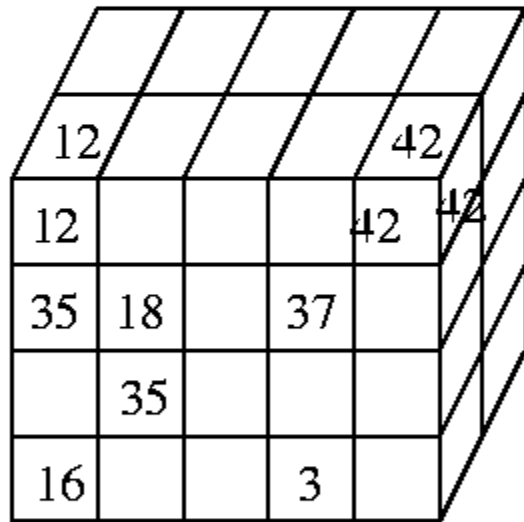
SWITCH

(inversion)

	12			42
	12			42
18	35		37	
35				
	16		3	



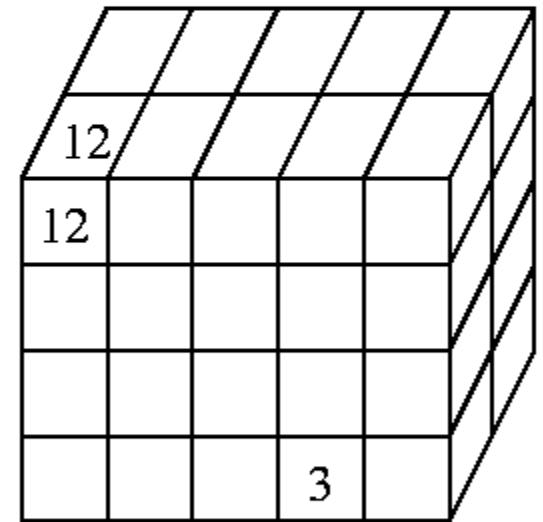
Sélection sur les cellules



A 3D grid representing a 4x5x2 volume. The front face contains the following numbers:

12				42
12				42
35	18		37	
	35			
16			3	

DICE
(restriction)

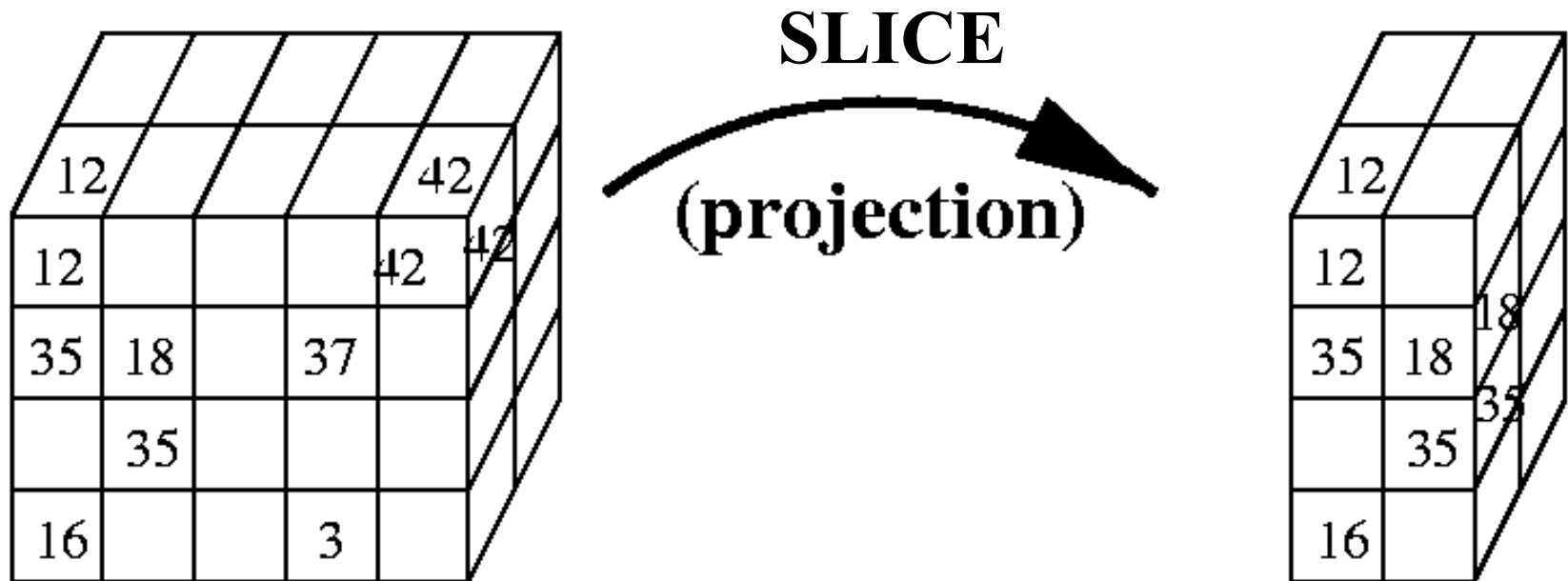


A 3D grid representing a 4x5x2 volume. The front face contains the following numbers:

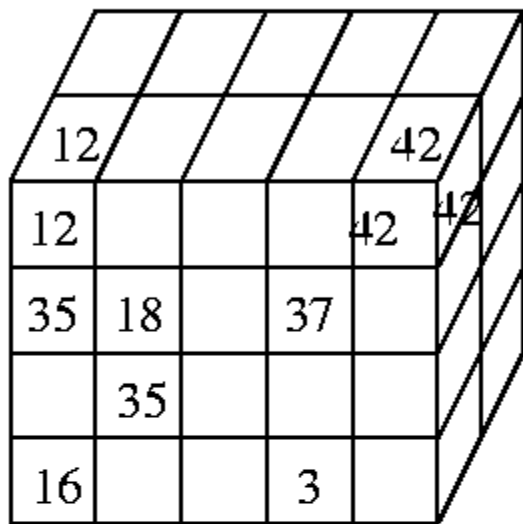
12				
12				
			3	

Critère de sélection : valeurs inférieures à 15

Sélection sur les tranches



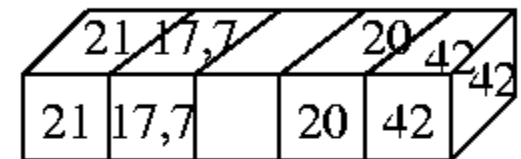
Généralisation/Spécialisation



12				42
12				42
35	18		37	
	35			
16			3	

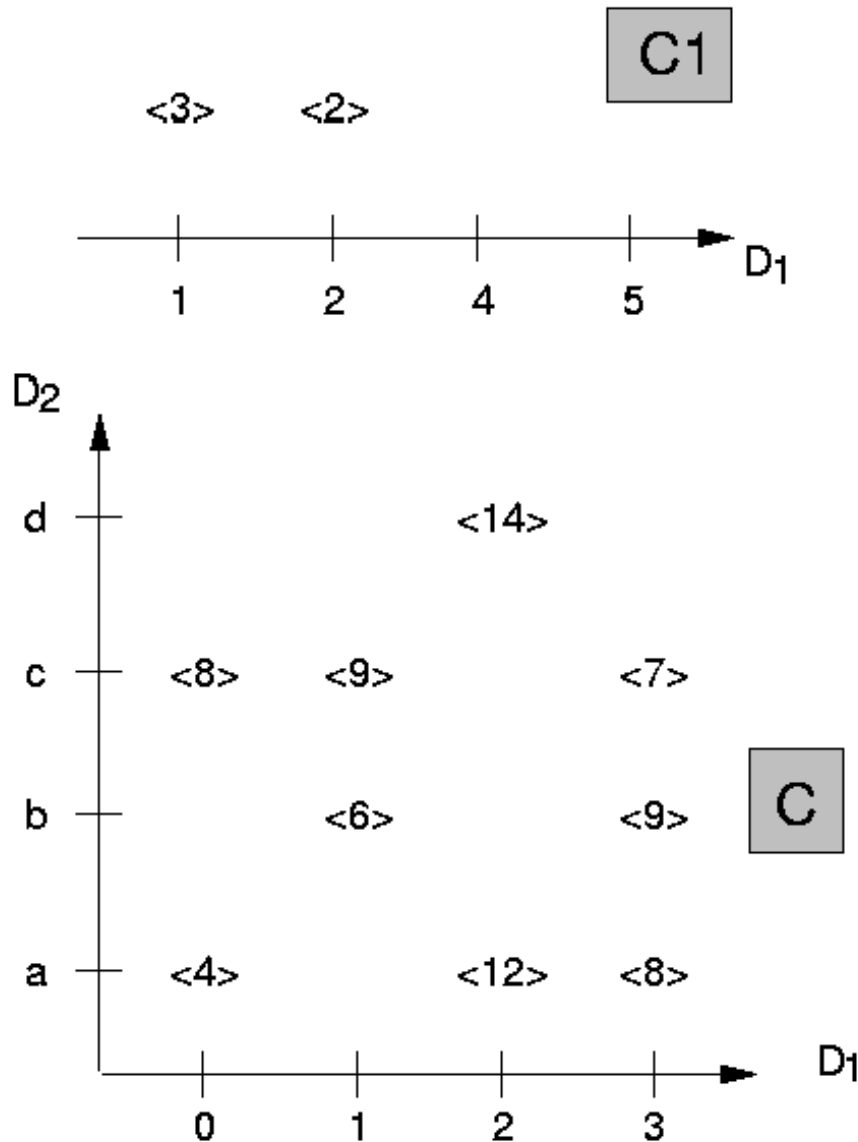
ROLL-UP

(generalisation)

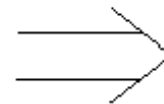


21	17,7	20	42
21	17,7	20	42

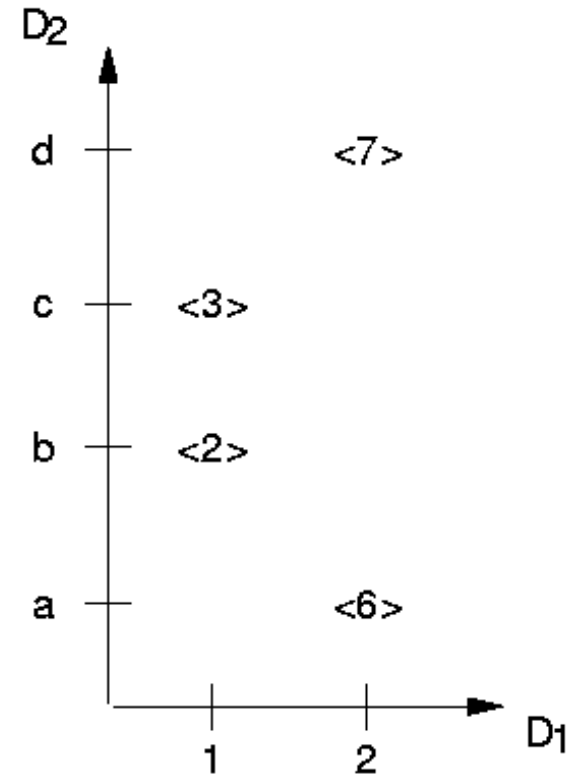
Agrawal et al. : Jointure de 2 cubes



map dimension
 D_1 using the
 identify mapping



elem divides
 the element from
 C by the element
 from C1 if both
 elements exist. Else
 it returns 0



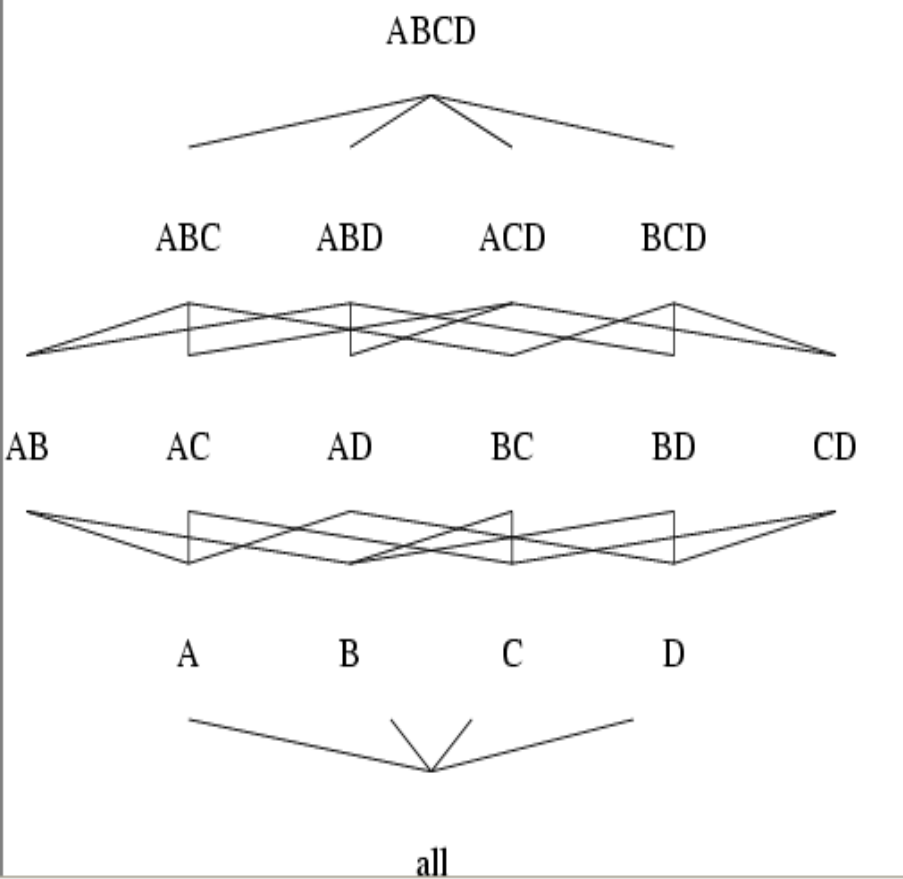
Problématiques associées

- Choix des cubes à construire
- Stockage physique
- Pré-calculs
- Mise à jour
- Types de dimension
- Utilisation : reporting, data mining

Choix des cubes à construire

- Cube : résultat de requête de type *group by* :
SELECT mois, produit, ville, count(*)
FROM Ventes
GROUP BY mois, produit, ville
- Opérateurs Oracle :
 - **GROUP BY CUBE**
 - **GROUP BY ROLLUP**
 - **GROUP BY GROUPING SETS**

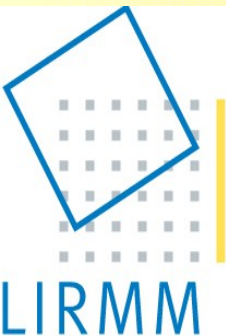
Cubes possibles : treillis des cuboïdes



Stockage physique des cubes

- **MOLAP** (Multidimensional OLAP) : tout le cube est matérialisé physiquement
- **ROLAP** (Relational OLAP) : les données sont stockées dans une base de données relationnelle. Le cube n'est pas matérialisé du tout sauf au moment de la phase de requête
- **HOLAP** (Hybrid OLAP) : seule une partie du cube est matérialisée sous forme multidimensionnelle. Les autres données sont laissées dans la base relationnelle et extraites de manière dynamique au moment des requêtes

Les cubes sont très clairsemés (sparsity)



Modèle sous-jacent

- Tables de faits : entité centrale
 - Objet de l'analyse, taille très importante
- Tables de dimensions : entités périphériques
 - Dimensions de l'analyse, taille peu importante
- Table de faits normalisée (BCNF)
- Tuples de la tables de faits :
 - Clés étrangères formant une clé primaire
 - Valeurs associées à chaque clé primaire (mesures)
- Association de type $(0,n) - (1,1)$ connectant les différentes dimensions aux faits

Normalisation des tables

- Etoile : tables dimensions non normalisées
- Flocon : tables de dimensions normalisées
 - Réduction de la redondance
 - Maintenance simplifiée
 - MAIS navigation coûteuse

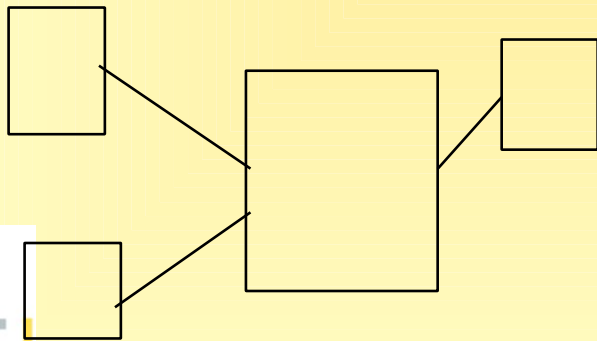


schéma en étoile

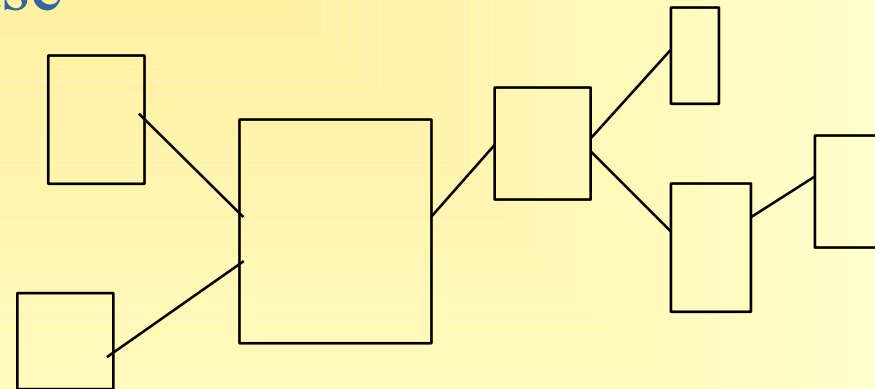
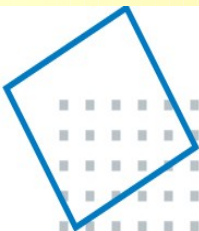
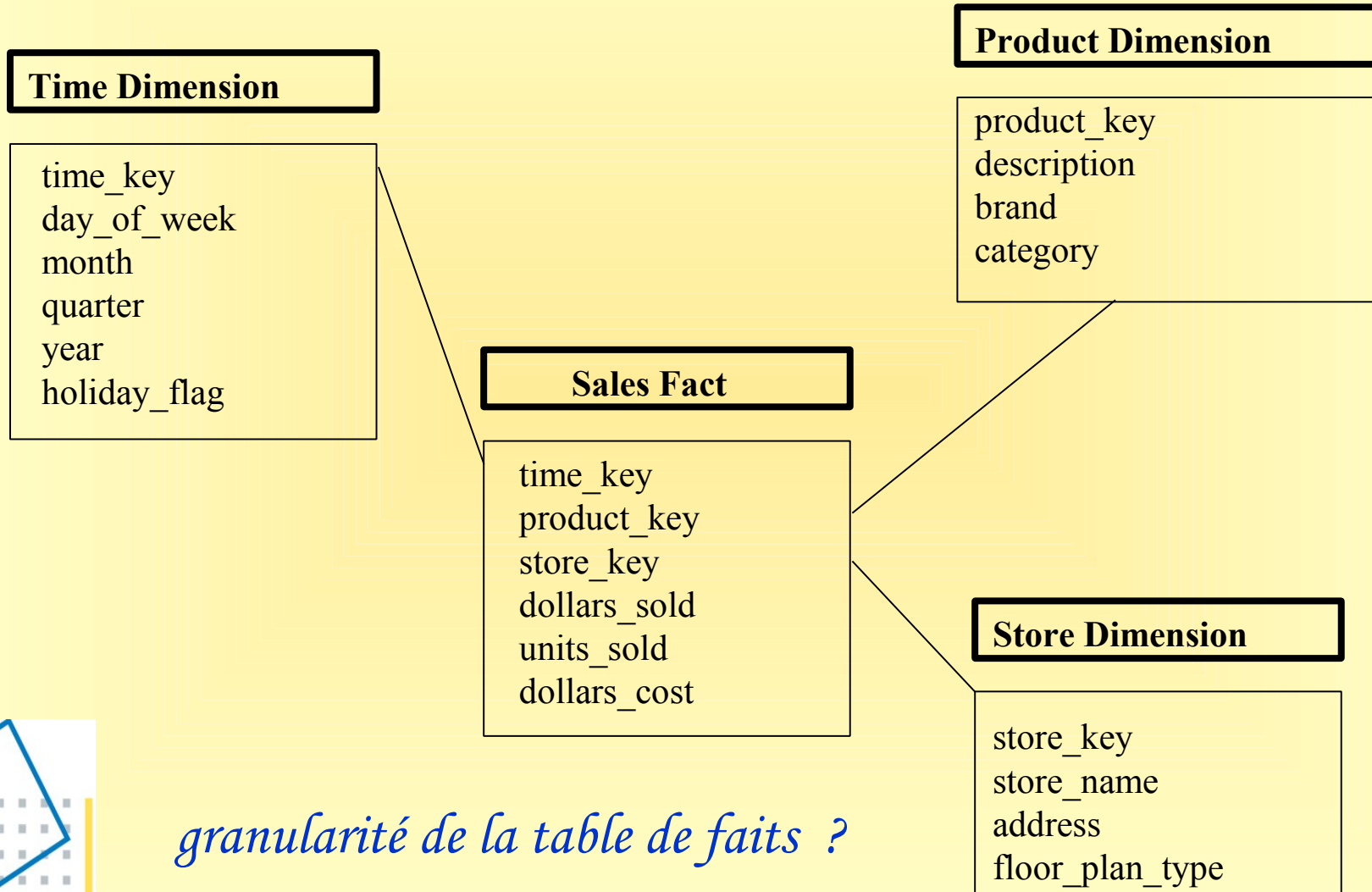


schéma en flocon



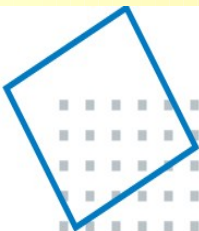
Exemple classique



granularité de la table de faits ?

Pré-calculs

- Pré-calcul de cubes
- Utilisation de cubes déjà calculés pour la construction de nouveaux cubes
- Pré-calcul de requêtes
- Agrégations
- Attention : calculer un cube à un certain niveau rend impossible le drill-down !



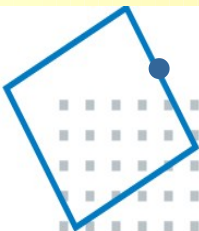
Mise à jour

- Selon la mise à jour de l'entrepôt
- Recalcul de tout le cube ?

Types de mesure

- Additive
- Semi-additive
- Non -additive

- Distributive (count, sum, min, max)
- Algébrique (avg)
- Holistique (rank, median)



Iceberg Cubes / iceberg queries

- Très actif depuis 1998 environ
- **But** : requêtes permettant de ne récupérer que les combinaisons dimensions/mesure ayant une valeur supérieure à un seuil fixé par l'utilisateur (le reste étant considéré comme trivial). On ne récupère pas grand chose → iceberg

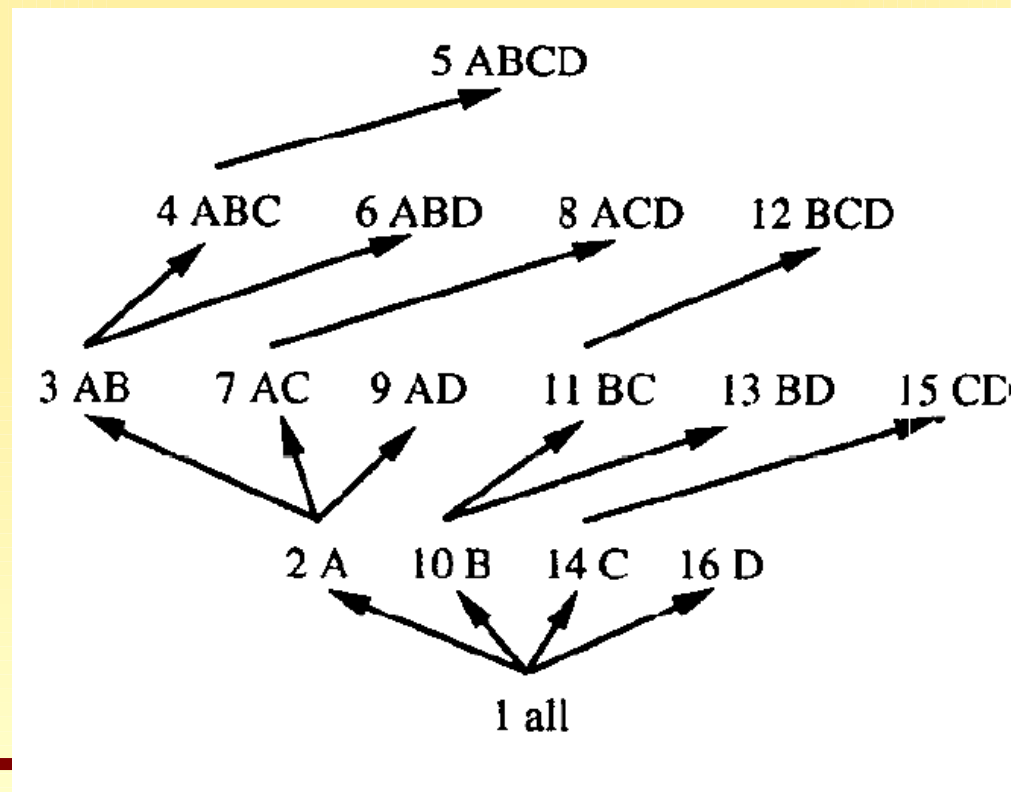
```
select D1, ..., Dk, count(mes)
from R
group by D1, ..., Dk
having count(mes) >  $\sigma$  ;
```

Problème : comment calculer ?

- comment choisir les combinaisons d'attributs (group by) pour que la fonction d'agrégat soit au dessus du seuil fixé ?
- Le plus simple : avoir en mémoire les combinaisons et un compteur qu'on renseigne en 1 passe sur la base ... impossible car grosses bases !
- Tri de sous-parties de la base avec maintenance de l'agrégat ... mais plusieurs passes sur la base (qui n'est peut-être pas matérialisée) !

Algorithmes pour calcul des icebergs

- Parcours du treillis des dimensions
- Algorithme BUC (Bottom Up Computation)
- [Beyer, Ramakrishnan, ACM-SIGMOD, 1999]



- Algorithme TDC (Top Down Computation)
- Recherches les plus récentes : Cho, Pei, Cheung, Cross Table Cubing: Mining Iceberg Cubes from Data Warehouses, *SDM'05*. Plusieurs tables (flocon, étoile)
- Et quand la fonction d'agrégat n'est pas anti-monotone ...
 - Efficient computation of Iceberg cubes with complex measures. Jiawei Han, Jian Pei, Guozhu Dong, Ke Wang, *SIGMOD 2001*.
 - Extension de BUC sur les mesures de type « moyenne » ou « profit » = prix - coût
 - Propriété anti-monotone : top-k average, top-k apriori, top-k BUC (calcul à partir des k plus grandes valeurs de la base)

OLAP & Data Mining

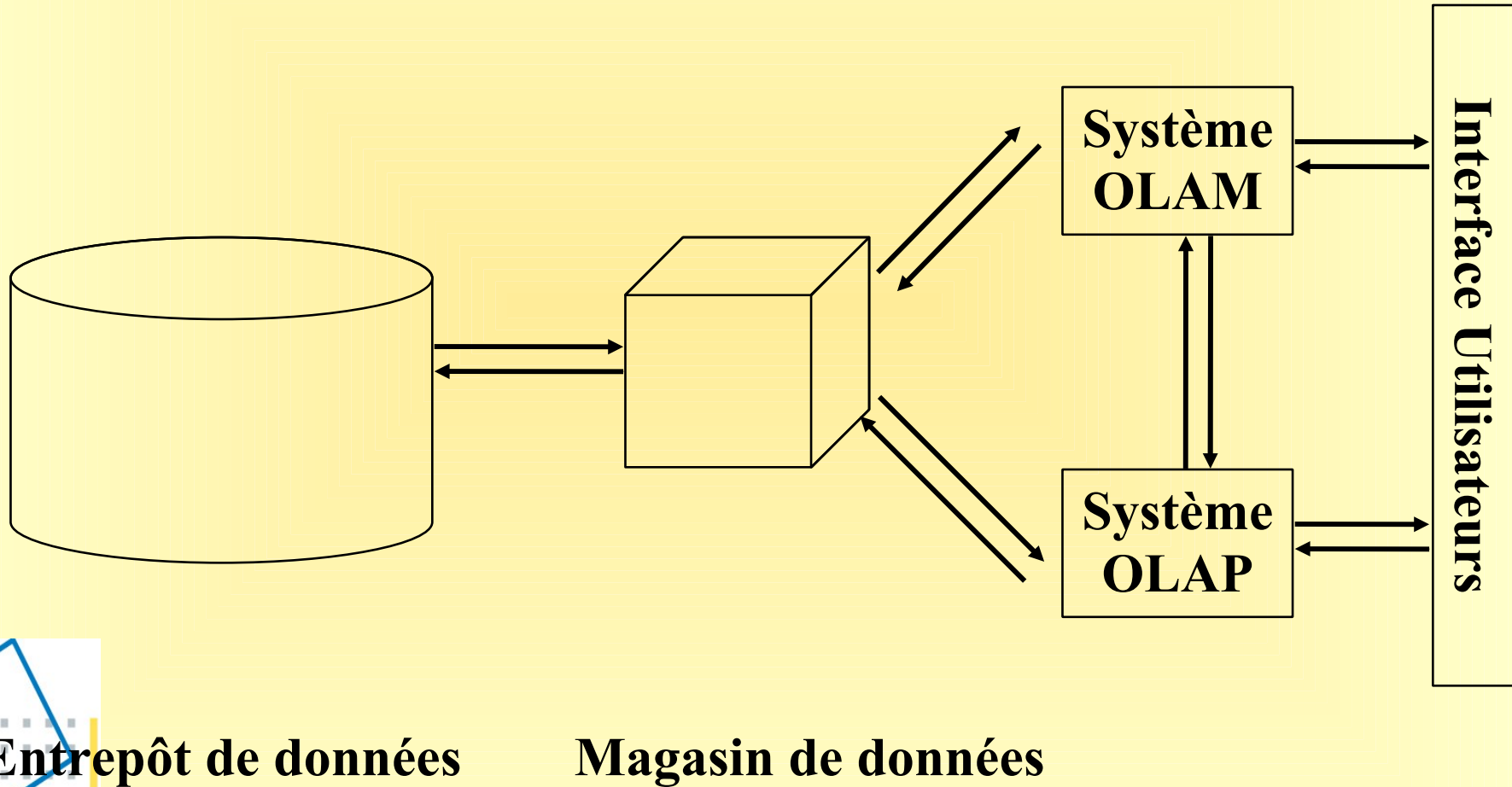
OLAP Mining (OLAM)

J. HAN (1997) : intégration du processus de data mining dans les bases de données multidimensionnelles (système DBMiner).

Intérêt des outils OLAP pour le data mining :

- Les données provenant des entrepôts ont été *nettoyées*
- Niveaux de granularité facilement exploitables
- Optimisation des requêtes complexes et pré-calculs
- Visualisation des données

Architecture générale



Modules d'analyse des données

J. Han (1997) :

- Caractérisation (résumé des données)
- Classification supervisée et non supervisée (*clustering*)
- Comparaison
- Extraction de règles d'association
- Prédiction
- Analyse de séries temporelles

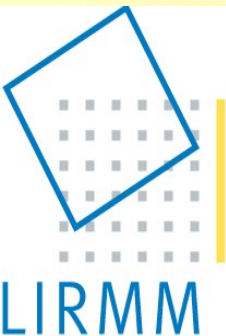
Représentations de cubes

- Qualité des représentations/pertinence
- Travaux de Yeow Wei Choong et Patrick Marcel
- Recherche de blocs homogènes

DOLAP 2001

Computing Appropriate
Representations
for Multidimensional Data

Yeow Wei Choong et al. , Proceedings of the *4th ACM international workshop on Data warehousing and OLAP (DOLAP)*, Atlanta, Georgia, USA, pages 16 – 23, 2001.

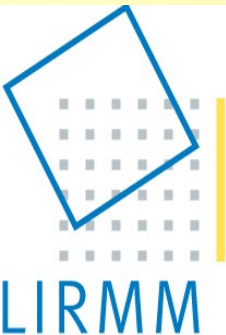


Before Restructuring

World Consumption (US\$bil)
Sales: 2000

	Beer	Water	Soda	Wine	Milk
Europe	4	4	7	6	5
America	4	5	7	7	6
Asia	3	3	6	5	5
Africa	2	2	6	5	4

Representation 1



After Restructuring

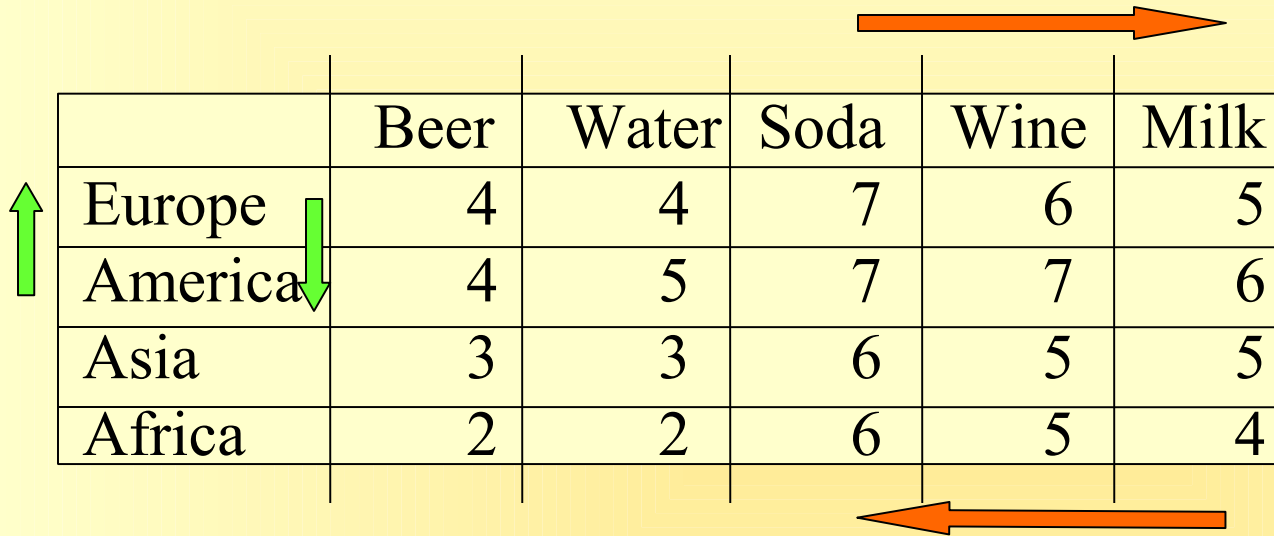
World Consumption (US\$bil)

Sales: 2000

	Beer	Water	Milk	Wine	Soda
America	4	5	6	7	7
Europe	4	4	5	6	7
Asia	3	3	5	5	6
Africa	2	2	4	5	6

Representation 2

"Switch"

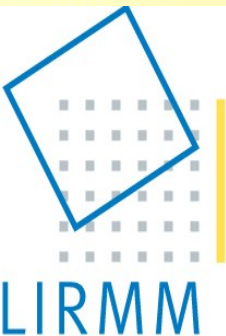


	Beer	Water	Soda	Wine	Milk
Europe	4	4	7	6	5
America	4	5	7	7	6
Asia	3	3	6	5	5
Africa	2	2	6	5	4

	Beer	Water	Milk	Wine	Soda
America	4	5	6	7	7
Europe	4	4	5	6	7
Asia	3	3	5	5	6
Africa	2	2	4	5	6

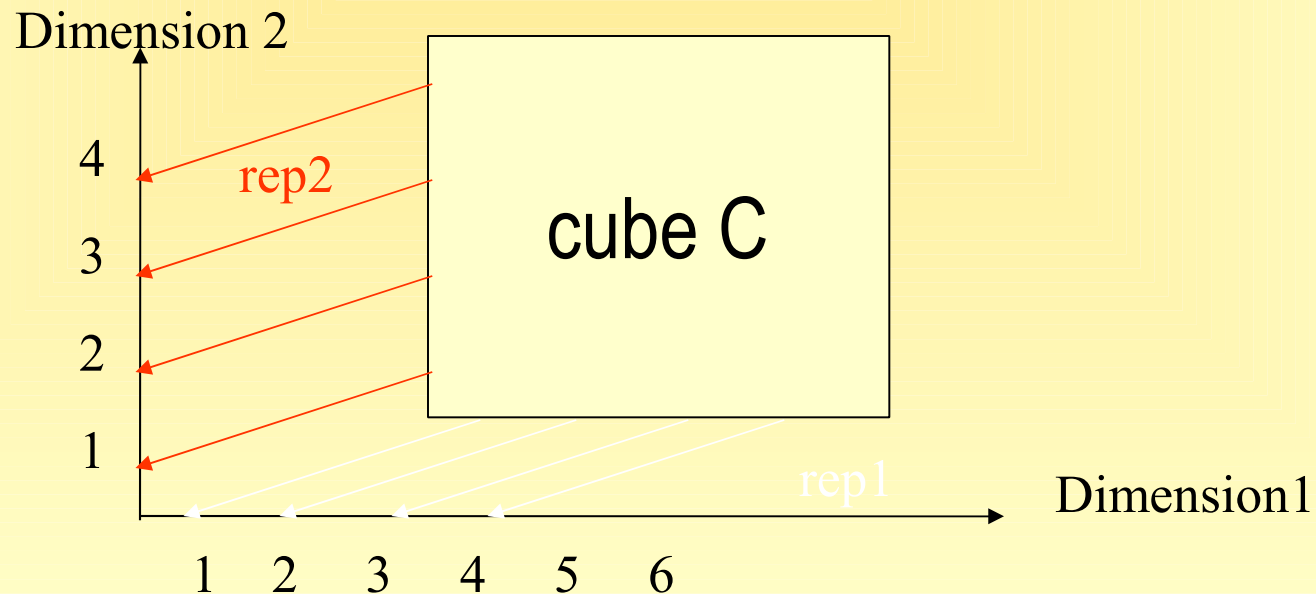
Motivation

- Many representations of a given cube
 - Constructed by the User
- What are the most appropriate representations
 - Quality of a Representation
- How to compute these Representations
 - "switch" operator



Representation

- Given a n -dimensional cube C , a representation of C is a set of n mappings:
 - one mapping per dimension
 - associates each member to an integer



Example: 2-dimensional cube

$\langle C, \{x, y\}, \{a, b\}, \{1, 2, 3, 4\}, m_c \rangle$ where
 $m_c(x, a) = 1, m_c(y, a) = 2, m_c(x, b) = 3, m_c(y, b) = 4$

Four possible representations:

R_1

2 a	1	2
1 b	3	4
	x	y
	1	2

R_2

2 b	3	4
1 a	1	2
	x	y
	1	2

R_3

2 b	4	3
1 a	2	1
	y	x
	1	2

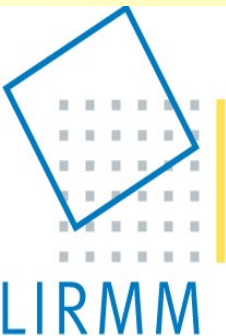
R_4

2 a	2	1
1 b	4	3
	y	x
	1	2

Note:

a	1	3
b	2	4

Not a representation of C



Position of a Cell

		R_1	
2 a	1	2	
1 b	3	4	
	x	y	
	1	2	

The position of cell $c_1 = \langle x, a, 1 \rangle$ is $\langle 1, 2 \rangle$

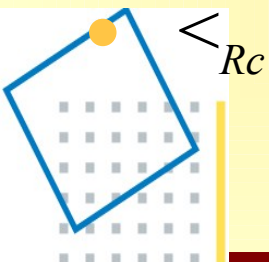
The position of cell $c_2 = \langle y, b, 4 \rangle$ is $\langle 2, 1 \rangle$



Cell Ordering

- A cube $\mathbf{C} = \langle C, dom_1, \dots, dom_n, dom_m, m_c \rangle$
- A representation $\mathbf{R}_c = \{rep_1, \dots, rep_n\}$
- Cells $\mathbf{c} = \langle m_1, \dots, m_n, m \rangle$
 $\mathbf{c}' = \langle m_1', \dots, m_n', m' \rangle$

$$\mathbf{c} <_{Rc} \mathbf{c}' \text{ iff } \forall i \in [1, \dots, n], rep_i(m_i) \leq rep_i(m_i')$$



is a partial ordering

Cell Ordering – an example

$$c_1 = \langle x, a, 1 \rangle$$

$$c_3 = \langle x, b, 3 \rangle$$

$$c_2 = \langle y, a, 2 \rangle$$

$$c_4 = \langle y, b, 4 \rangle$$

	R	
a	1	2
b	3	4
	x	y

We have:

$$c_3 <_R c_1$$

$$c_3 <_R c_2$$

$$c_3 <_R c_4$$

$$c_1 <_R c_2$$

$$c_4 <_R c_2$$

Note that c_1 cannot be compared to c_4

Misplaced Cell

- Given a representation R_c of a cube C
- $c = \langle m_1, \dots, m_n, m \rangle$ is misplaced w.r.t. R_c if

1. $m \neq \perp$

and

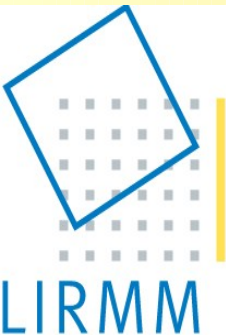
2. $\exists c_1 = \langle m_1', \dots, m_n', m' \rangle \in C$ such that

$$c <_{R_c} c_1 \text{ and } m > m'$$

or

$\exists c_2 = \langle m_1'', \dots, m_n'', m'' \rangle \in C$ such that

$$c_2 <_{R_c} c \text{ and } m'' > m$$



Characterizing the Representation

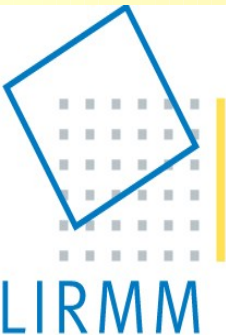
Given a representation R_C of a cube C

$M_{R_C}(C)$: number of misplaced cells in C w.r.t. R_C

- R_C is a **Perfect Representation (PR)** if
$$M_{R_C}(C) = 0$$

(i.e. there are no misplaced cells w.r.t. R_C)
- R_C is an **Optimal Representation (OR)** if
$$\forall R'_C, M_{R'_C}(C) \geq M_{R_C}(C)$$

(i.e. there is no other 'better' representation)



Switching

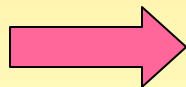
$$\text{switch}(j, p, q)(R_c) = R'_c$$

R'_c is obtained from R_c by permutation of rows p and q of dimension j

Example

R_1

a	1	2
b	3	4
	x	y



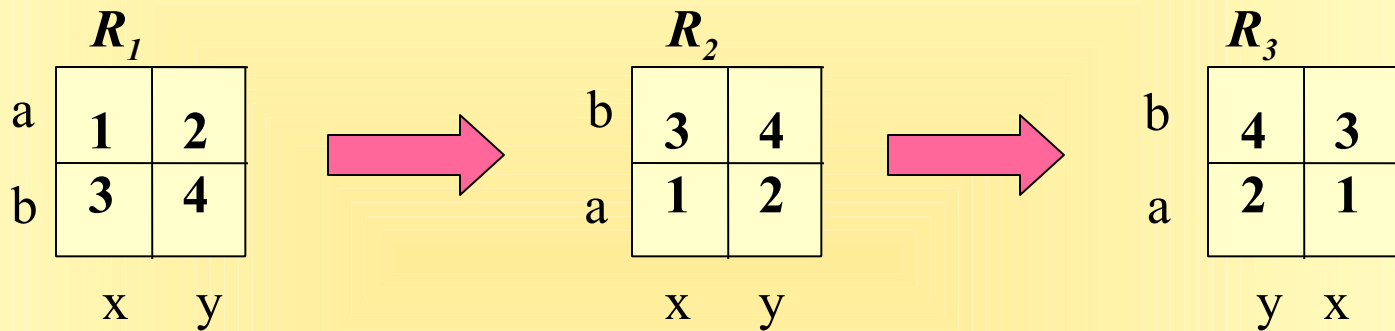
R_2

b	3	4
a	1	2
	x	y

$$\text{switch}(1, a, b)(R_1)$$

Arrangement

An **arrangement** is a finite composition of switches



$$\text{switch}(1, a, b)(R_1) = R_2 \quad \text{switch}(2, x, y)(R_2) = R_3$$

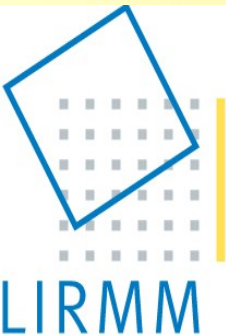
$$\text{switch}(2, x, y)(\text{switch}(1, a, b)(R_1)) = R_3$$



PR Problem

For a given cube and a given representation of this cube,

- **Test whether there exists at least one PR**
 - **If so, compute one PR**
- **If there are more than one PR**
 - **Compute the number of PRs**
 - **List all the arrangements leading to these PRs**



Basic Theorem

A representation of a cube is a PR if and only if every row in every dimension is sorted

Sketch of the proof:

- *If*: Trivial since if a representation is a PR then every row in every dimension must be sorted
- *Only if*: Consider the following example which

can *not* be a PR. If every row is sorted then

we must have: $X \geq 1, Y \geq 1, X \leq 0, Y \leq 0$

impossible

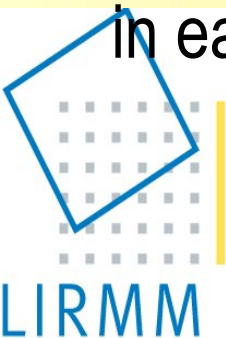
	R	
a	X	0
b	1	Y
	x	y



Case 1

No duplicates and no null values in each row

- There exists *at most* one *PR* of a given cube \mathcal{C}
- If there exists a representation such that for one dimension, a row r is sorted and another row r' is *not* sorted, then there exists no *PR*
- If a *PR* exists, then it can be obtained by sorting *only* one row in each dimension



Case 1

No duplicates and no null values in each row

input: The representation of a cube C

output: The PR of C or the indication “no PR”

For each dimension k of C do:

 choose a row r in k

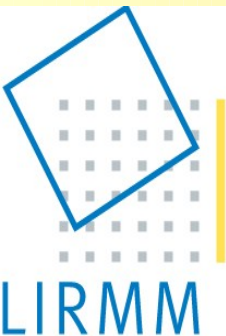
 sort r

 for every row r' in k do

 check if r' is sorted

 if r' is unsorted then

 exit with output “non PR”



Case 2

Dealing with duplicates & no null values

$$R_1$$

b	4	3
a	1	1
	x	y

$$R_2$$

b	3	4
a	1	1
	y	x

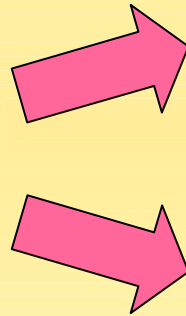
Sorting row $\langle \mathbf{a} \rangle$ may lead to representation R_1 which is **not** perfect since row $\langle \mathbf{b} \rangle$ is not sorted

Sorting row $\langle \mathbf{b} \rangle$ leaves row $\langle \mathbf{a} \rangle$ **unchanged** and gives a PR

Case 3

Dealing with null values

b	1	⊥	4	2	⊥
a	1	2	3	⊥	⊥
	v	y	w	x	z



b	1	2	⊥	4	⊥
a	1	⊥	2	3	⊥
	v	x	y	w	z

b	1	⊥	2	4	⊥
a	1	2	⊥	3	⊥
	v	y	x	w	z

Null values

		R_1		
		x	y	
b	\perp	$\mathbf{0}$		sorted
a	$\mathbf{1}$	\perp		sorted

sorted sorted

		R_2		
		y	x	
b	$\mathbf{0}$	\perp		
a	\perp	$\mathbf{1}$		

But R_1 is not a PR !

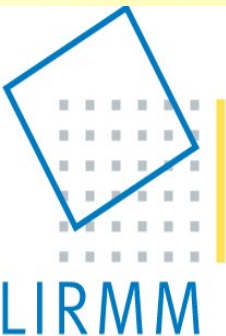
R_2 is a PR

R_1 is a Weak PR (WPR)

Current & Future Work

- PR Problem with Null Values
- Introducing an Efficient Implementation
- Identifying all ORs and their Arrangements
- Use other OLAP operations (e.g. roll-up)
- t -OR Problem: given a threshold t , find R_c of C such that

$$M_R^c(C) \leq t$$



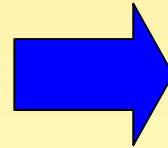
Recherche de blocs homogènes

Conférences EGC 2004, IPMU'04, IPMU'06,
VIEW'06, chapitre IDEA

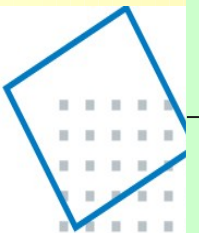
Représentations parfaites

- Recherche efficace d'organisations pertinentes des valeurs des dimensions pour l'organisation de la valeur de la mesure

3	5	2	4
2	4	1	3
5	7	4	6
4	6	3	5



4	5	6	7
3	4	5	6
2	3	4	5
1	2	3	4

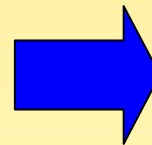


Notre approche

- Détection de *blocs* contenant une grande proportion de valeurs identiques

PRODUIT						
P1	6	6	8	5	5	2
P2	6	8	5	5	6	75
P3	8	5	5	2	2	8
P4	8	8	8	2	2	2
	V1	V2	V3	V4	V5	V6

VILLE



PRODUIT						
P1	6	6	8	5	5	2
P2	6	8	5	5	6	75
P3	8	5	5	2	2	8
P4	8	8	8	2	2	2
	V1	V2	V3	V4	V5	V6

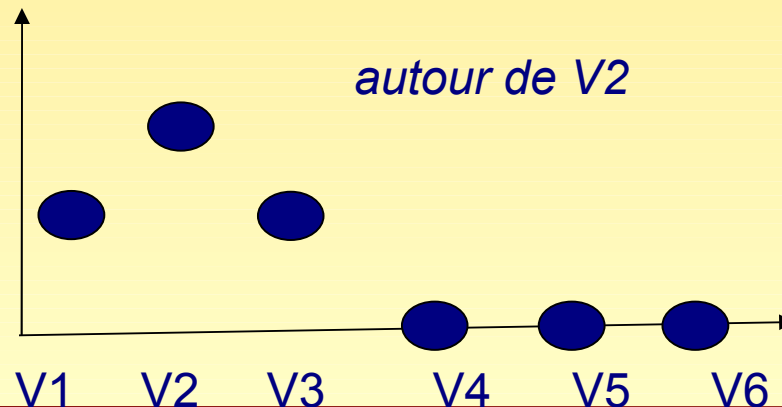
VILLE

Description des blocs par des règles

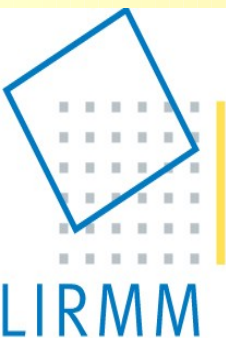
- Une règle pour chaque bloc
- Règles maximales
- Nécessité de représenter les recouvrements

⇒ Règles floues

si valeur de dimension
ville =



Alors valeur de
mesure = 8



Génération des blocs

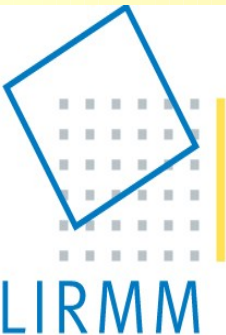
- Algorithme par niveaux, dimension par dimension
- Support d'un bloc b pour une valeur de mesure m :

$$\text{supp}(b, m) = \frac{\# \text{ occurrences de } m \text{ dans } b}{\# \text{ cellules de } C}$$

- Confiance d'un bloc b pour une valeur de mesure m

⋮

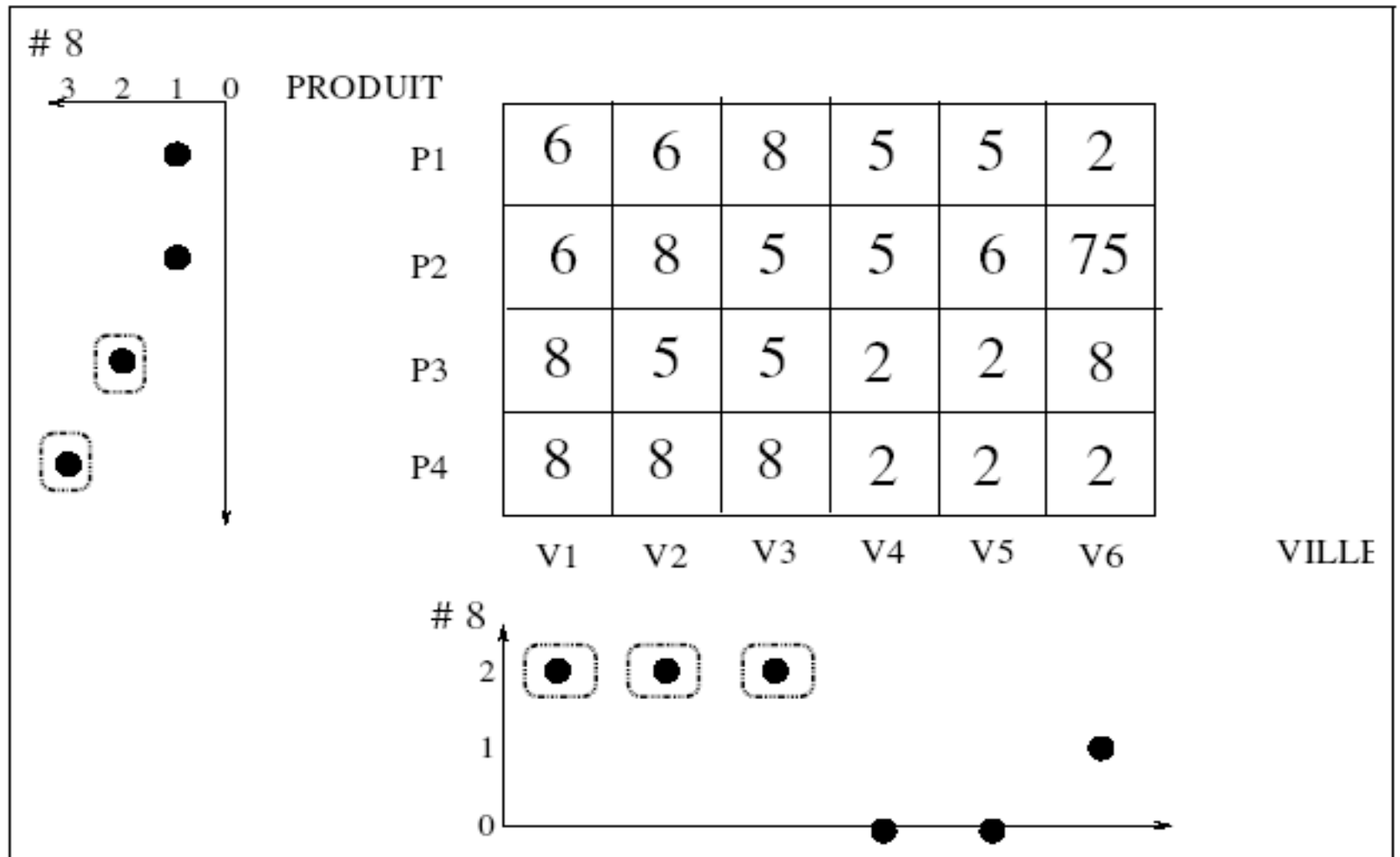
$$\text{conf}(b, m) = \frac{\# \text{ occurrences de } m \text{ dans } b}{\# \text{ cellules de } b}$$



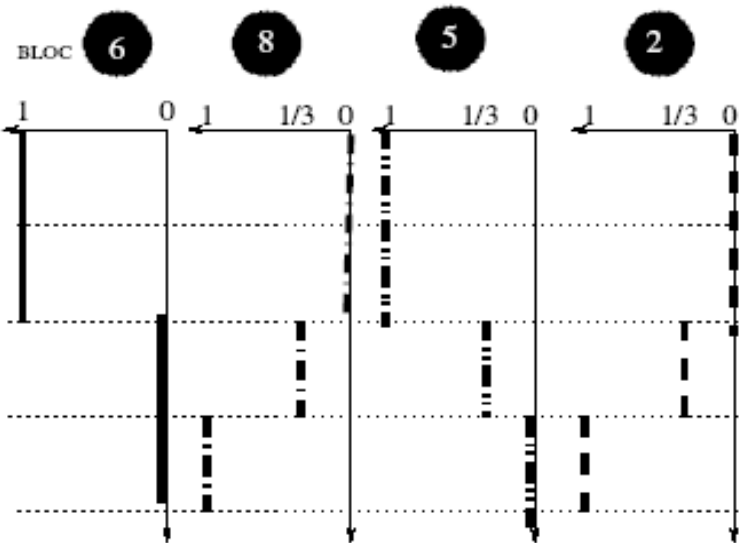
Algorithme de recherche (simplifié)

- **ENTRÉE** : cube de données C , k dimensions, σ seuil de support minimum, γ seuil de confiance minimale
- **SORTIE** : ensemble des blocs associés à C
- **Pour chaque valeur de mesure m**
 - Pour chaque dimension d_i ($i = 1$ à k)
 - Construction des intervalles maximaux de valeurs de d_i formant des blocs b tels que $supp(b,m) > \sigma$
 - Pour chaque dimension d_p ($p = 2$ à k)
 - Génération des candidats à partir des fréquents de taille $p-1$
 - Coupure, puis calculs des supports des candidats restants
 - Suppression des blocs b tels que $conf(b,m) < \gamma$
- Si visualisation alors remplacement des valeurs de cellules

Exemple de calcul



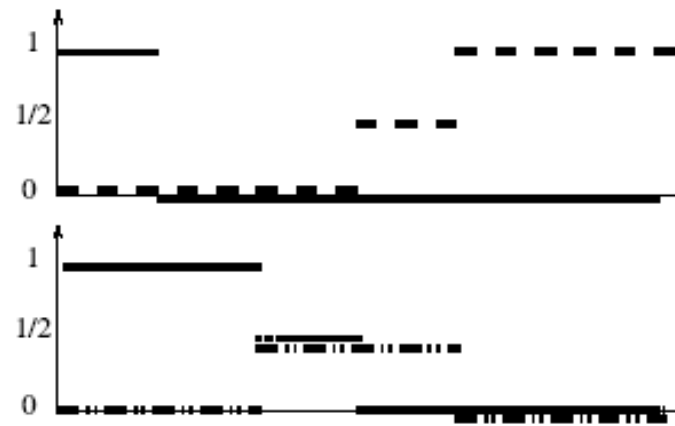
Exemple de résultat



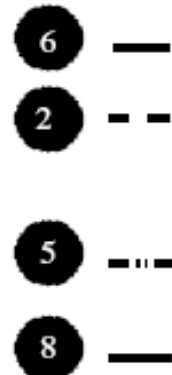
PRODUIT

	V1	V2	V3	V4	V5	V6
P1	6		5	5		
P2	6		5	5		
P3	8	2	48	76	2	8
P4	8	8	8	2	2	2

VILLE



BLOCS



Blocs intervalles et « flous »

- Problème souvent rencontré : pas ou peu de blocs trouvés (ou petits)
- ⇒ Redéfinition du support et de la confiance pour prendre en compte des valeurs *presque* égales
- $i_{sup} \ i_{conf}$
 - $f_{sup} \ f_{conf}$
 - Blocs construits avec intervalles (flous) plus grands que les blocs « crisp »



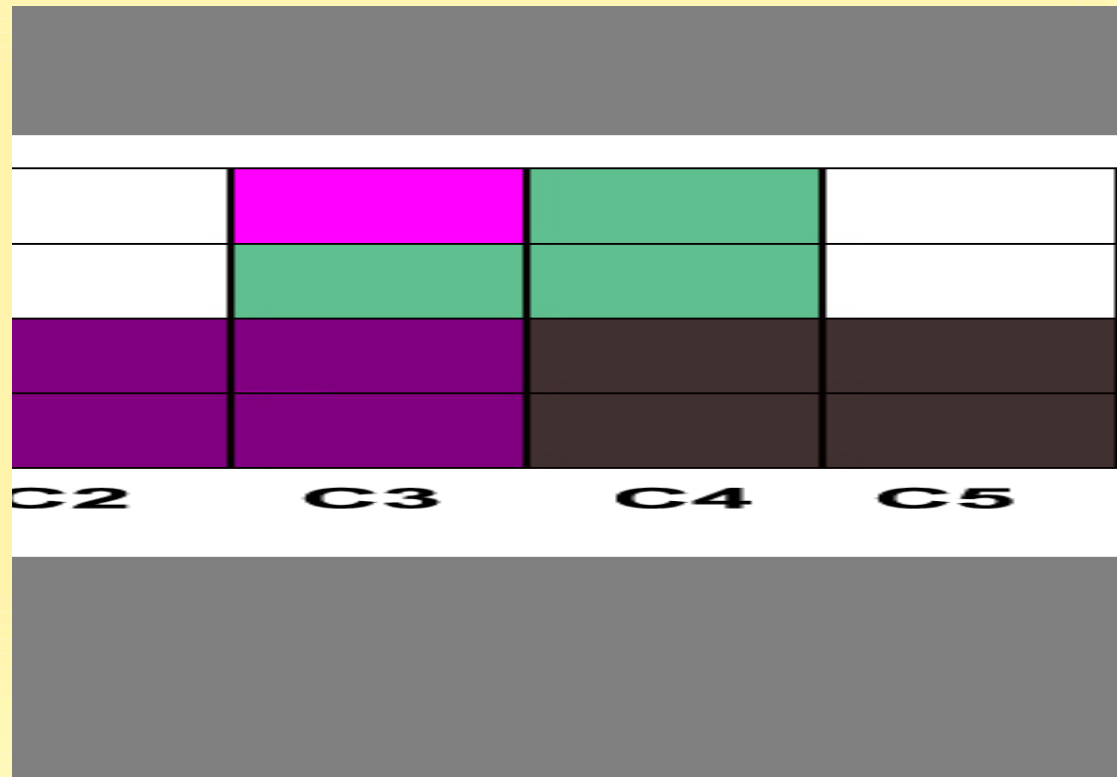
intervalles

6	5.9	7.8	4.8	5
6.1	8	5.1	4.7	5.3
8.1	5	4.9	2.4	1.8
7.9	8.1	8.2	2.2	1.9

Intervalles flous

6	5.9	7.8	4.8	5
6.1	8	5.1	4.7	5.3
8.1	5	4.9	2.4	1.8
7.9	8.1	8.2	2.2	1.9

Visualisation des blocs



Qualité des représentations

La qualité s'exprime en fonction de :

- la proportion de cellules incluses dans des blocs,
- du nombre de blocs construits,
- du nombre de blocs par rapport au nombre de valeurs de mesure,
- du nombre de recouvrements entre blocs et de leur taille.

Travaux connexes

- Bases de données géographiques
- Segmentation
- Segmentation floue
- Résumés de cubes/quasi cubes

Conclusion

- Méthode de résumé et de visualisation de cubes de données
- Méthode fondée sur un algorithme par niveaux
- Découverte de blocs de données
- Description des blocs par des règles
- théorie des sous-ensembles flous

OLAP & OLAP Mining : Exemples possibles d'extension au flou

- Requêtes OLAP floues

Les ventes de chaussures sont-elles *fortes* ?

Les ventes à l'*est* sont-elles *fortes* ?

- Données imparfaites :

- imprécises (les ventes ont été *fortes*)

- et/ou incertaines (les ventes sont *sans doute* de 88000 unités)

- Apprentissage flou, notamment arbres de décision flous (Salammbô) et résumés flous

Extension réalisées

- **Modèle multidimensionnel :**
 - Représentation
 - Manipulation
- **Algorithmes d'extraction d'information :**
 - Arbres de décision flous
 - Résumés flous
 - Résumés simples (*La plupart des ventes sont faibles*)
 - Résumés complexes (*La plupart des ventes de canoës en février sont faibles*)
 - Résumés intra-dimensions (*La plupart des ventes fortes de chaussures de l'est proviennent de **Boston***)
 - Recherche de cellules anormalement vides

Cas multidimensionnel flou

- **Prise en compte du flou :**
 - données
 - termes de résumés
- **Résumés multi-niveaux :**
 - prise en compte de chaque dimension à tous les niveaux possibles
 - raffinement possible sur une ou plusieurs dimensions

Prise en compte de la mesure

- Mesure additive :
support de règles
- Mesure partitionnée (partition floue) :
nouvelle dimension

exemple : ventes faibles, moyennes, fortes

Types de résumés flous générés

- résumés *inter-dimensions* :
la plupart des ventes de *canoës* à *Boston* sont *faibles*
- résumés *intra-dimensions* :
la plupart des ventes de *l'EST* sont effectuées à *Boston*
- *raffinement* de résumés :
 1. production d'un résumé à niveau de granularité élevé :
peu de ventes au **deuxième trimestre** 1995 concernent les **produits de camping**
 1. raffinement sur une ou plusieurs dimensions :
peu de ventes au **deuxième trimestre** 1995 concernent des **tentes**

Génération efficace des résumés

- Méthode *naïve* inapplicable
- Méthodes de réduction de la complexité :
 - Algorithmes par niveaux
 - Choix utilisateur :
 - supports et degrés de vérité minimaux
 - quantificateurs
 - dimensions
 - termes de résumé
 - niveaux de granularité
 - dimensions à raffiner
- Propriétés des résumés
- Propriétés des bases multidimensionnelles

Cellules vides

- Cubes souvent très peu denses : problème de stockage efficace
- Pour l'analyse des données :
 - remplacement des cellules vides par une valeur par défaut
 - non prise en compte des cellules vides
- Sémantique :
 - **pas de valeur (position inapplicable)**
 - **valeur 0**
 - **donnée manquante**

Motivations

Produit	Ville	Ventes
canoes	L.A.	30
canoes	S.F.	29
canoes	N.Y.	10
tentes	L.A.	48
tentes	N.Y.	2
boissons	L.A.	123
boissons	S.F.	145
boissons	N.Y.	47
chocolat	L.A.	152
chocolat	S.F.	100
chocolat	N.Y.	59

Tentes S.F. ?

		L.A.	S.F.	N.Y.
équipements	canoes	30	29	10
	tentes	48	--	2
nourriture	boissons	123	145	47
	chocolat	152	100	59

Raisons possibles de l'absence de valeur

- Aucun n-uplet relatif aux ventes de tentes à San-Francisco
- n-uplet avec valeur manquante :

Tentes S.F. ?

Tentes ? 60

? S.F. 60

Intérêt potentiel pour l'analyste

- Identification de biais possibles introduits dans l'analyse des données
- Identification de nouveaux marchés porteurs
- Identification de défauts commerciaux pour certains produits, certains magasins, ...
- Évaluation de la qualité des données

Bases de données multidimensionnelles

- Problème des cellules vides :
 - Traité comme un problème de stockage efficace
 - Pour l'analyse des données :
 - Remplacement des cellules vides par une valeur par défaut
 - Non prise en compte des cellules vides

Détection des cellules vides potentiellement intéressantes

- Grand nombre de cellules vides
- Mise en évidence de toutes les cellules potentiellement intéressantes et seulement elles
 - Définition de la notion de cellule vide intéressante dans ce cadre
 - Méthodes de parcours efficaces (passage à l'échelle)

Notion de cellule vide intéressante

- On appelle cellule vide intéressante une cellule vide qui **dénote** à côté de cellules non vides proches (au sens des hiérarchies).

	L.A.	S.F.	N.Y.
équipements < canoes	30	29	10
équipements < tentes	48	--	2
nourriture < boissons	123	145	47
nourriture < chocolat	152	100	59

Détection des cellules anormalement vides

- **Méthodes d'apprentissage (supervisé) :**
 - transformation des données (2 classes : cellule vide / cellule non vide)
- **Étude des voisinages (au sens des hiérarchies) :**
 - construction des voisinages de chaque cellule vide
 - calcul du degré d'intérêt de chaque cellule vide

exemple : proportion de cellules vides dans le voisinage

Voisinage de cellule

2 formes de voisinage

- *bloc*
 - Ensemble des cellules pour lesquelles les positions sur **toutes les dimensions** ont même parent

⇒ 1 seul voisinage par bloc
- Colonne / tranche
 - Ensemble des cellules telles qu'il existe au moins une dimension dont les positions ont même parent dans la hiérarchie

Exemples de voisinages

Ouest

L.A.

S.F.

N.Y.

équipements

canoës

30

29

10

tentes

48

--

2

nourriture

boissons

123

145

47

chocolat

152

100

59

Utilisation de mesures floues :

- Définition de relations floues R, par exemple *approximativement égal à* ou *environ la moitié de*
- On a alors
 - $0 \leq \delta \leq 1$
 - Calcul de d :

$$\delta = T (f_R(v_{c'}, v_c))$$

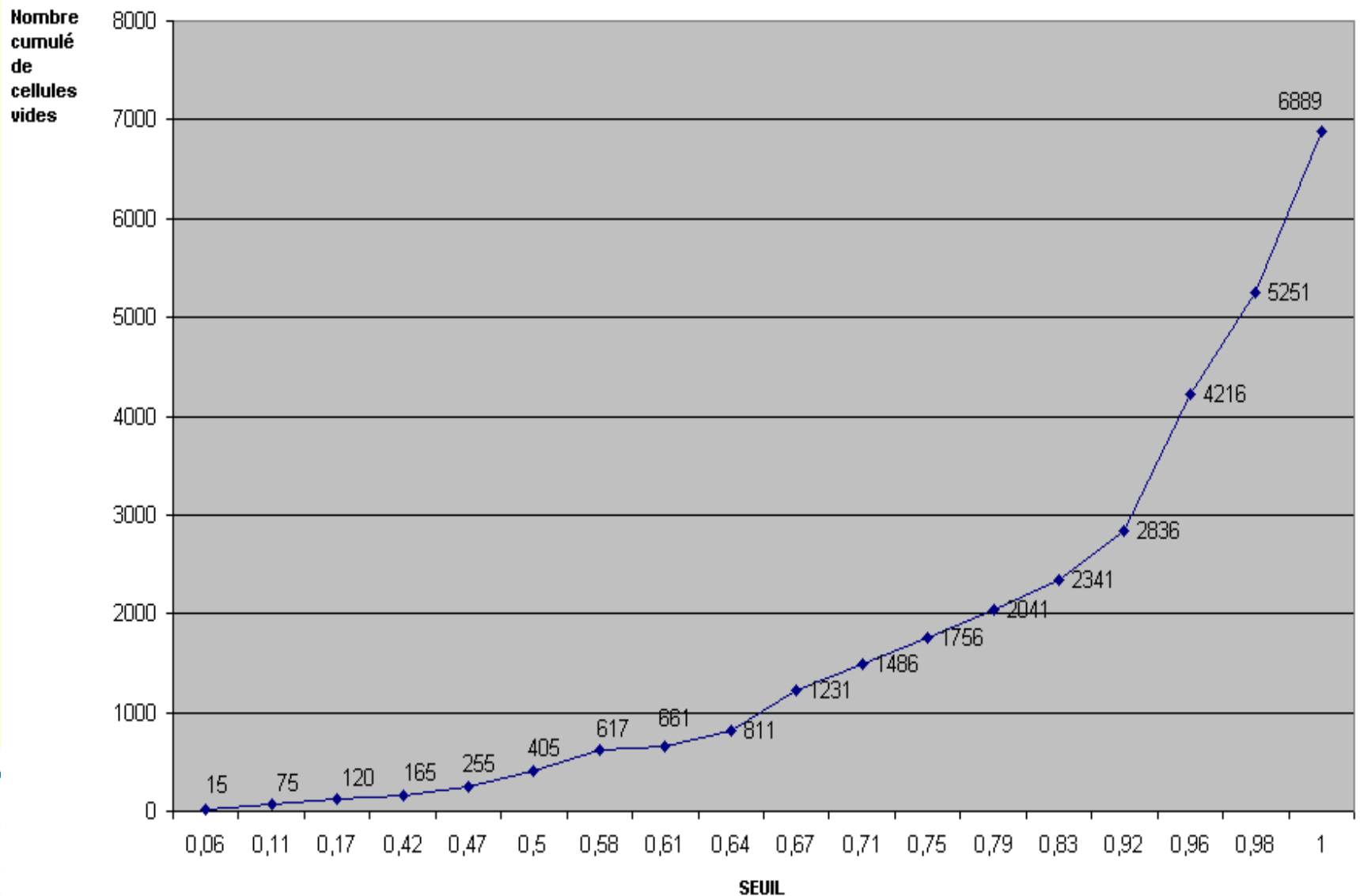
$$c' \in V(c)$$

$$c' = (d_1, \dots, \mathbf{d}_i, \dots, d_k)$$

Prise en compte de tous les niveaux de hiérarchie

- Faible taux de cellules vides quand on agrège
- Possibilité de considérer toutes les combinaisons niveau fils – niveau parent (pas forcément parent immédiat)
- Seuils à fixer en fonction du niveau de hiérarchie

Résultats obtenus :

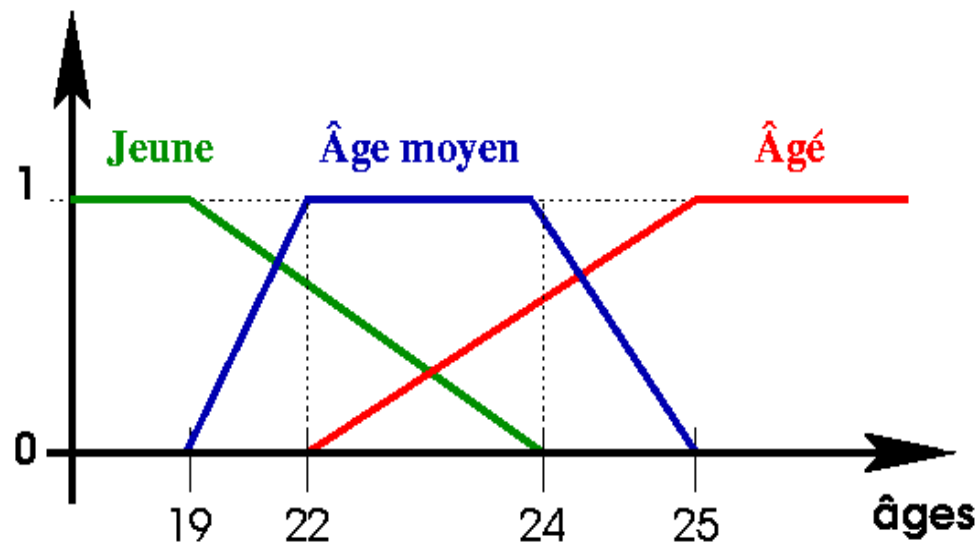


Bases de données test

- **Base Oracle** : résultat de ventes de produits
- **Base BAC** :
 - fournie par le MENRT
 - résultats au baccalauréat pour deux années consécutives :
1 million de n-uplets
 - base relationnelle (Oracle 8)
 - exemple de cube construit :
proportion d'admis en fonction de 9 dimensions
17 millions de cellules

Résultats obtenus : arbres de décision

- R1:** Chez les jeunes*, sans spécialité, la proportion de candidats reçus avec mention (AB, B ou TB) est supérieure à 25%.
- R2:** Chez les lycéens d'âge moyen*, sans spécialité, la proportion de candidats reçus avec mention (AB, B ou TB) est inférieure à 25%.
- R3:** Chez les jeunes*, avec la spécialité "math", passant le bac série "S", la proportion de candidats reçus avec mention (AB, B ou TB) est supérieure à 25%.
- R4:** Chez les lycéens âgés*, passant le bac série "S", la proportion de candidats reçus avec mention (AB, B ou TB) est inférieure à 25%.



Résultats obtenus : résumés flous

- Exemples de résumés produits :

peu de *publicité* sur les *canoës* est faite à *Boston*

peu de *ventes* dans l'*Ouest* concernent des produits de *camping*

la plupart des *très jeunes* bacheliers ont *16 ans*, très peu ont *15 ans*

Exemples de résumés flous

Peu de Ventes sont *moyennes* : 0

Environ la moitié des Ventes sont *moyennes* : 0,84

La plupart des Ventes sont *moyennes* : 0,16

La plupart(1) des Ventes sont *moyennes* : 0,48

La plupart(2) des Ventes sont *moyennes* : 0

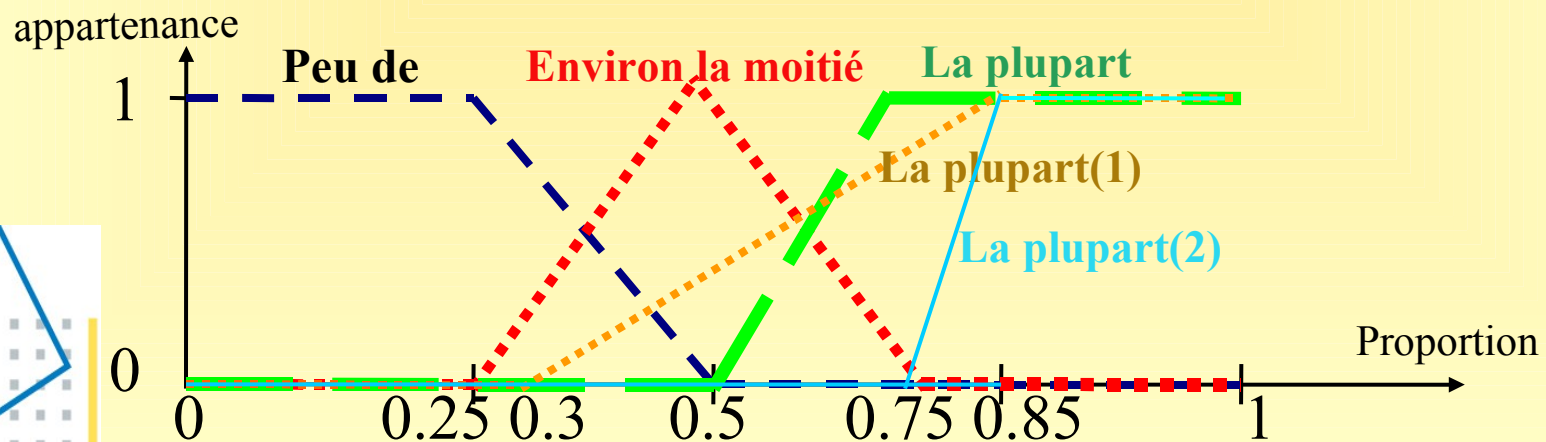
Peu de Ventes sont *fortes* : 0,40

Environ la moitié des Ventes sont *fortes* : 0,49

La plupart des Ventes sont *fortes* : 0

La plupart(1) des Ventes sont *fortes* : 0,19

La plupart(2) des Ventes sont *fortes* : 0



Références (1/3)

BDM et OLAP

E.F. Codd, S.B. Codd, and C.T. Salley, Providing OLAP (on-line analytical processing) to user-analysts: An IT mandate. *Technical report*, 1993

Pilot White Paper. *An introduction to OLAP, Multidimensional Terminology and Technology*, 1995 <http://www.pilotsw.com/olap/olap.htm#dsgl>

P. Vassiliadis, T. Sellis. A Survey on Logical Models for OLAP Databases. *SIGMOD Record*, volume 28, numéro 4, 1999.

OLAP Mining

J. Han, OLAP Mining: An Integration of OLAP with Data Mining, *IFIP Conference on Data Semantics*, 1997, pages 1-11.

<http://www.cs.sfu.ca/people/Faculty/Han/han.html>

S. Sarawagi, R. Agrawal et N. Megido, Discovery-driven exploration of OLAP Data Cubes. *Proc. Of EDBT'98*.

H. Guenzel, J. Albrecht et W. Lehner, Data Mining in a Multidimensional Environment, in *Proc. Of ADBIS'99*, 1999.

Références (2/3)

OLAP Mining Flou

- A. Laurent, « De l'OLAP Mining au F-OLAP Mining », *Actes des Journées Francophones d'Extraction et Gestion des Connaissances, Revue Extraction des Connaissances et Apprentissage*, vol.1, numéro 1-2, pages 189-200, Hermès, 2001.
- A. Laurent, B. Bouchon-Meunier, A. Doucet, S. Gancarski et C. Marsala, « Fuzzy Data Mining from Multidimensional Databases », *Int. Symp. on Computational Intelligence, Studies in Fuzziness and Soft Computing*, J. Kacprzyk ed., vol. 54, Springer-Verlag, pages 278-283, 2000.
- A. Laurent, S. Gancarski et C. Marsala, «Coopération entre un système d'extraction de connaissances floues et un système de gestion de bases de données multidimensionnelles», *Rencontres Francophones sur la Logique Floue et ses Applications*, Cepaduès éd., pages 325-332, 2000.

Sites internet

AltaPlana Online Analytical Processing : <http://www.altaplana.com/olap>

Data Warehousing and OLAP Bibliography : <http://www.cs.toronto.edu/~mendel/dwbib.html>

OLAP Report : <http://www.olapreport.com>

Références (3/3)

Logique Floue

- B. Bouchon-Meunier, *La logique floue*, Que Sais-je, PUF, 1994.
- Kaufmann, Introduction à la théorie des sous-ensembles flous, vol. 1 & 3, Masson, 1973.

BD Floues

- D. Dubois and H. Prade, Using fuzzy sets in flexible querying: Why and how?, Flexible Query Answering Systems (FQAS), Kluwer Academic Publishers, pages 45-60, 1997

<ftp://ftp.irit.fr/pub/IRIT/RPDMP/UFSFQW.ps.gz>

- P. Bosc and O. Pivert, Fuzzy Databases, Handbook of Fuzzy Computation, P. Bonissone and W. Pedrycz ed., 1998, chapter 4.1, pages 1-12, Oxford University Press
- P. Bosc and D. Dubois and H. Prade, Fuzzy Functional Dependencies -An Overview and a critical discussion, *Jal of the American Society for Information Science*, vol 49, p 217-235, 1998
- E.A. Rundensteiner and L. Bic, Aggregates in Possibilistic Databases, *Int. Conf. on Very Large Data Bases*, Morgan Kaufmann, 1989

Résumés flous

- J. Kacprzyk and R.R. Yager and S. Zadrozny, A fuzzy logic based approach to linguistic summaries of databases, *Int. Journal of Applied Mathematics and Computer Science*, volume 10, pages 813-834, 2000.