Entrepôts de données : Systèmes ROLAP, MOLAP et HOLAP (5)

Bernard ESPINASSE
Professeur à Aix-Marseille Université (AMU)
Ecole Polytechnique Universitaire de Marseille

Janvier 2016

- **Introduction aux systèmes OLAP**
- **Systèmes ROLAP**
- **Systèmes MOLAP**
- **Systèmes HOLAP**

Plan

1. **Introduction aux systèmes OLAP**
2. **Systèmes ROLAP**
   - Introduction à la technologie ROLAP
   - Techniques d'indexation ROLAP
   - Sélection et matérialisation de vues
   - Fragmentation de tables
   - Forces et faiblesses de la technologie ROLAP
   - Quelques produits de technologie ROLAP
3. **Systèmes MOLAP**
   - Introduction à la technologie MOLAP
   - Techniques de stockage
   - Densité et compression
   - Agrégation et calcul des agrégats
   - Forces et faiblesses de la technologie MOLAP
   - Quelques produits de technologie MOLAP
4. **Systèmes HOLAP**
   - Introduction à la technologie HOLAP
   - Quelques produits de technologie HOLAP

Bibliographie et sources du cours

Ouvrages :

- Benitez-Guerrero E., C. Collet, M. Adiba, « Entrepôts de données : Synthèse et analyse », Rapport de recherche IMAG N°IMAG-RR - 99-1017-I, 1999.
- Vaisman A., Zimányi E., « Data Warehouse Systems: Design and Implementation », Springer-Verlag, 2014, ISBN 978-3-642-54654-9.
- Golfarelli M., Rizzi S., « Data Warehouse Design : Modern Principles and Methodologies », McGrawHill, 2009.
- Kimball R., Ross, M., « Entrepôts de données : guide pratique de modélisation dimensionnelle », 2°édition, Ed. Vuibert, 2003, ISBN : 2-7117-4811-1.

Cours :

- Cours de F. Bentayeb, O. Boussaid, J. Darmont, S. Rabaseda, Univ. Lyon 2
- Cours de P. Marcel, Université de Tours
- Cours de G. Gardarin, Université de Versailles
- Cours de M. Adiba et M.C. Fauvet, Université Grenoble
- Cours de H. Garcia-Molina, Stanford Universitry.

1 – Introduction aux systèmes OLAP

- **Caractéristiques des produits OLAP**
- **Différents types de systèmes OLAP**

Les règles de Codd pour les produits OLAP

En 1993 Codd définit les bases du modèle OLAP : 12 règles de Codd définissent l'évaluation des produits OLAP :

1. **Vue multidimensionnelle** : Une base OLAP offre une vue multidimensionnelle des données
2. **Transparence**: éléments techniques mis en œuvre invisibles pour l'utilisateur
3. **Accessibilité**: la complexité et l'hétérogénéité des données sont masquées par les outils OLAP
4. **Stabilité**: performances stables indépendamment du contexte d'analyse
5. **Architecture Client/Serveur** : le coté serveur a en charge l'homogénéisation des données, les clients se connectent simplement au serveur
6. **Traitement générique des dimensions** : une seule structure logique pour toutes les dimensions. Tout calcul effectué sur une dimension peut l'être sur les autres
7. **Gestion dynamique des matrices creuses** : gestion dynamique de la mémoire physique nécessaire pour stocker les données non nulles
8. **Support multi-utilisateurs** : gestion des accès concurrents aux données
9. **Croisement des dimensions**
10. **Manipulation intuitive des données**
11. **Flexibilité des restitutions**
12. **Nombre illimité de niveaux d'agrégations et de dimensions**

Caractéristiques majeures des produits OLAP

L'acronyme FASMI (Fast Analysis of Shared Multidimensional Information) permet de **résumer la définition des produits OLAP** (http ://www.olapreport.com/fasmi.htm, The OLAP Report - 2004) :

- **Fast** : temps de réponse aux demandes des utilisateurs entre 1 et 20 secondes : utilisation dans les produits OLAP de **pré-calculs** pour réduire les durées des requêtes,
- **Analysis** : faire face à toutes les logiques d'affaire et de statistiques, ainsi que fournir la possibilité aux utilisateurs de construire leurs calculs et leurs analyses sans avoir à programmer : **outils fournis** avec les produits OLAP
- **Shared** : le système doit créer un contexte où la **confidentialité** est préservée et doit gérer les cas où plusieurs utilisateurs ont des droits en écritures (plutôt une faiblesse des produits OLAP actuels)
- **Multidimensional** : caractéristique majeure, les produits OLAP doivent fournir des **vues conceptuelles multidimensionnelles** des données et supporter des **hiérarchies de dimensions**
- **Informations** : ensemble des données et les informations nécessaires pour un produit OLAP.

Différents types de systèmes OLAP

3 principales stratégies d'implémentation d'ED et d'analyse OLAP possibles dans les produits OLAP :

1 - Systèmes ROLAP (Relational OLAP) :
- utilisent un SGBD relationnel classique avec des adaptations spécifiques à l'OLAP
- la base relationnelle de l'entrepôt est organisée pour réagir comme une base OLAP
- lents et peu performants mais sans limites de taille

2 – Systèmes MOLAP (Multidimensionnal OLAP) :
- utilisent un SGBD multidimensionnel (MOLAP), ils sont l'application physique du concept OLAP (réellement d'une structure multidimensionnelle)
- très rapides et performants mais limité au gigaoctet.

3 – Systèmes HOLAP (Hybrid OLAP) :
- c'est un compromis :
  - une base MOLAP pour les données souvent consultées (la minorité selon Pareto),
  - une base ROLAP pour les autres (la majorité).

4 – Base DOLAP (Desktop OLAP) : base OLAP très limitée en taille, hébergée sur le poste client, et très rapide

5 - Base OOLAP (Object OLAP) : utilise un SGBD Orienté Object : peu utilisé.

Quelques solutions commerciales

Nom	Editeur	Technologie
DB2 UDB Server	*IBM*	ROLAP
Oracle9i	*Oracle*	ROLAP
SQL Server 2000	*Microsoft*	ROLAP
DSS	*Microstrategy*	ROLAP
TeraData	*Teradata Corporation*	ROLAP massivement parallèle
Informix Metacube	*Informix*	MOLAP
Essbase	*Arbor Software/Hyperion*	MOLAP
SAS OLAP Server	*SAS*	MOLAP
Metacube	*Informix*	ROLAP
SQL Server	*Microsoft*	HOLAP
MDDB	*SAS Institute*	MOLAP/ROLAP
Oracle Express-server	*Oracle*	MOLAP/ROLAP
DB2 OLAP Server	*IBM*	MOLAP/ROLAP
Crystal	*Seagate Software*	Serveur d'application OLAP unique pour tous les déploiements
PowerPlay	*Cognos*	idem

2 – Systèmes ROLAP

- **Stratégie ROLAP d'implantation d'un ED**
- **Modèles logiques d'un ED « ROLAP »**
- **Introduction à la technologie ROLAP**
- **Techniques d'indexation ROLAP**
- **Sélection et matérialisation des vues en Rolap**
- **Fragmentation des tables**
- **Forces et faiblesses de la technologie ROLAP**
- **Quelques produits de technologie ROLAP**

Stratégie ROLAP d'implantation d'un ED

- les **SGBD relationnels représentant plus de 80% des SGBD** : c'est la **stratégie la plus couramment utilisée** pour implanter un ED
- les SGBD relationnels doivent cependant **être adaptés** car ils n'ont pas les caractéristiques adéquates pour répondre aux besoins des ED :
  - **Extensions du langage SQL à de nouveaux opérateurs**
  - Usage de **vues matérialisées**
  - **Indexation binaire** pour améliorer les performances
  - …
- ils réalisent des **calculs** de **données dérivés** et **agrégations** à différents niveaux
- ils génèrent des requêtes adaptées au schéma relationnel de l'ED et tirent profit des **vues matérialisées** existantes (facteur principal de performance)

Modèles logiques d'un ED « ROLAP » (1)

Le modèle multidimensionnel est traduit ainsi :

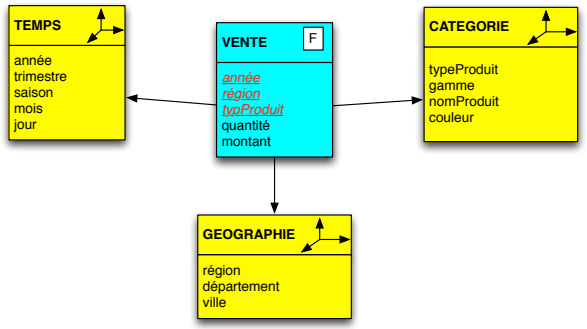
- **chaque fait** correspond à une table, appelée **table de fait**,
- **chaque dimension** correspond à une table, appelée **table de dimension**.

Ainsi :

- la **table de fait** est constituée :
  - d'attributs représentant les mesures d'activité et
  - les attributs clés étrangères de chacune des tables de dimension.
- les **tables de dimension** contiennent :
  - les paramètres et
  - une clé primaire permettant de réaliser des jointures avec la table de fait.

Modèles logiques d'un ED « ROLAP » (2)

Soit le schéma en étoile :

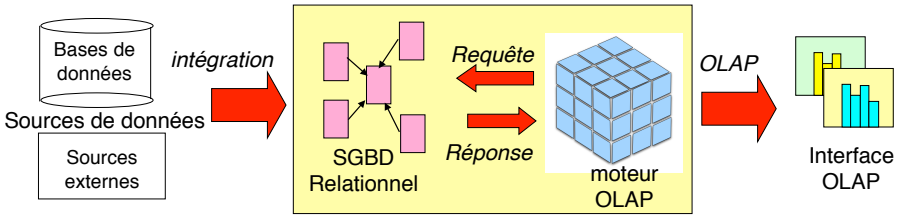


Modélisation logique ROLAP :

- VENTE(CleTps#,CleGeo#,CleCat#,Quantite,Montant) - ***table des faits***
- TEMPS(CleTps,Annee,Trimestre,Saison,Mois,Jour)) - ***table de dimension***
- GEOGRAPHIE(CleGeo,Region,Departement,Ville) - ***table de dimension***
- CATEGORIE(CleCat,TypeProd,Gamme,NomProd,Couleur) - ***table de dimension***

Introduction à la technologie ROLAP (1)

- un **SGBR relationnel** est utilisé pour stocker l'ED (en étoile ou flocon)
- le **moteur OLAP** est un **élément complémentaire** qui :
  - **fournit une vision multidimensionnelle** de l'ED,
  - **fait des calculs de données dérivés et des agrégations** à différents niveaux
  - **génère des requêtes SQL adaptées** au schéma relationnelle de l'ED en profitant de vues matérialisées existantes



Introduction à la technologie ROLAP (2)

- **Systèmes ROLAP = technologie de stockage relationnelle**
- Le **modèle relationnel demande des extensions** pour supporter les requêtes d'analyses multidimensionnelles du niveau d'application :
- **Extensions du langage SQL :**
  - de **nouveaux opérateurs** tels que « **cube** » et
  - de **nouvelles fonctions** comme « **rank** » et « **percentile** » complètent les fonctions classiques de SQL comme « count », « sum » et « avg »

=> Cf Cours Extension SQL pour l'OLAP

Introduction à la technologie ROLAP (3)

- **Le moteur OLAP :**
  - **traduit dynamiquement** le *modèle logique de données multidimensionnel* **M** en *modèle de stockage relationnel* **R** (en étoile ou en flocon)
  - techniquement, **il transforme** *une requête multidimensionnelle* **m sur M en une** *requête relationnelle* **r sur R**
- **L'efficacité** de la requête détermine la performance et le passage à l'échelle global du système :

  => choix de technique/stratégies d'optimisation spécifiques distinguant les produits ROLAP :

  1. **techniques d'indexation spécifiques**
  2. **sélection** et **matérialisation de vues**
  3. **fragmentation** des tables de l'ED

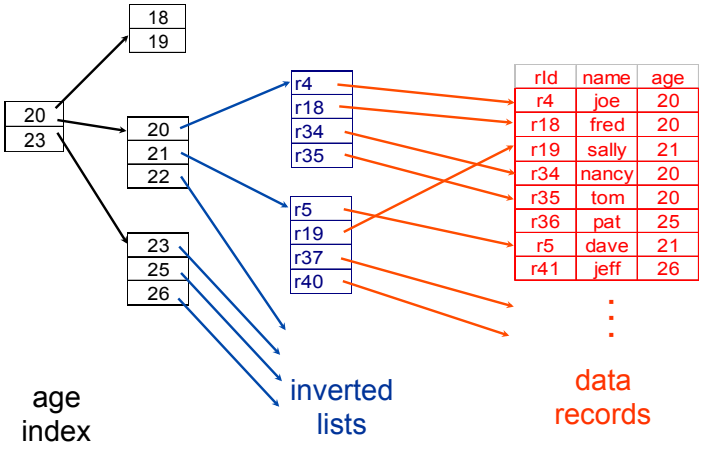
1. Rolap & Indexation

Principales techniques d'indexation ROLAP :

- **listes inversées** (inverted lists)
- **indexation binaire** (index de vecteurs de bits - bitmap indexing) :
  - oracle 9i
  - DB2
  - microsoft SQL server 2000
  - sybase IQ
- **index de jointure** (join indexing)
  - oracle 9i

1. Rolap & Indexation : listes inversées/inverted lists (1)

listes inversées (inverted lists) :



D'après H. Garcia-Molina

1. Rolap & Indexation : listes inversées/inverted lists (2)

Exemple d'utilisation :

- **Requête :**

```
GET people WITH age = 20 AND name = ''Fred''
```

- Soit L1 = liste pour attribut « age = 20 » : {r4, r18, r34, r35}
- Soit L2 = liste pour attribut « name = Fred » : {r18, r52}

- **Réponse à la requête :**

  c'est l'intersection de L1 avec L2 :

  L1 Π L2 = {r4, r18, r34, r35} **Π** {r18, r52} = **{r18} = r18**

1. Rolap & Indexation : indexation binaire/bitmap index (1)

indexation binaire = index de vecteurs de bits = bitmap index

Principe général :

- dans une table, un **index** associe, pour chaque valeur possible d'un **attribut** (ou groupe d'attributs), la **liste des tuples contenant cette valeur**
- un **index binaire** utilise un **vecteur de bits** pour représenter une telle liste : chaque tuple d'une table est associé à un bit qui prend la valeur :
  - **1 si le tuple associé fait partie de la liste** ou
  - **0 dans le cas contraire**
- **index binaire** = structure de **taille réduite** qui peut être gérée totalement en **mémoire centrale** améliorant les performances des SGBDR
- indexation **adaptée** lorsque le nombre de **valeurs possibles d'un attribut est faible**

1. Rolap & Indexation : indexation binaire/bitmap index (2)

Intérêts de l'indexation binaire :

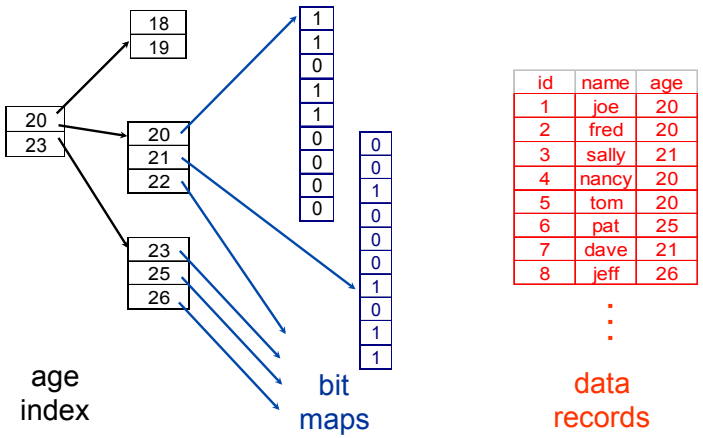
- **opérations sur les bits très rapides** (AND, OR, XOR, NOT)
- permet **l'optimisation de requêtes** de types sélection, comparaison, jointure, agrégation …
- un vecteur de bits pour **chaque valeur d'attribut**
- la **longueur du vecteur de bits = nb de tuples** de la table
- **plus compact que les B arbres**

Inconvénients de l'indexation binaire :

- le **coût de maintenance peut être important** car tous les index binaires d'une table doivent être **actualisés** lors de l'insertion d'un nouveau tuple dans la table
- **espace important pour leur stockage :** cependant vecteurs avec principalement des bits de valeur 0, d'où usage de techniques de compression (run-lengh encoding)

1. Rolap & Indexation : indexation binaire/bitmap index (3)

D'après H. Garcia-Molina



1. Rolap & Indexation : indexation binaire/bitmap index (4)

Exemple : soit le cube Ventes :

id	produit	ville
id1	clous	lyon
id2	vis	paris
id3	clous	paris
id4	écrous	lyon

index Bitmap associés :

Index Bitmap sur l'attribut **produit** :

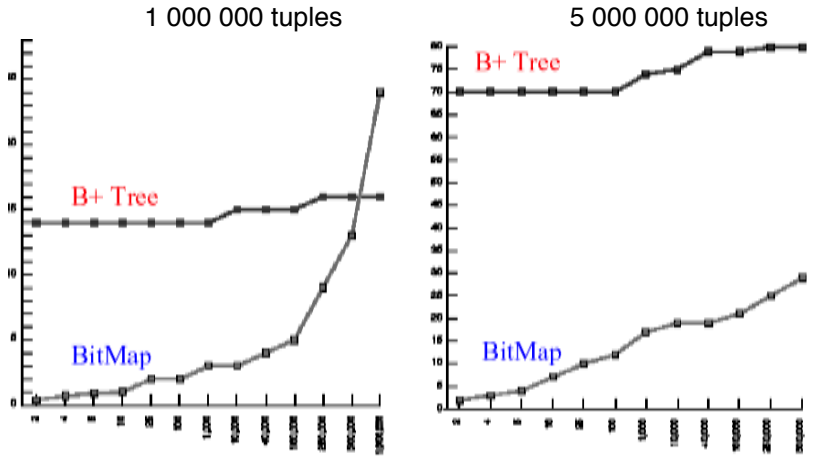
id	clous	vis	écrous
id1	1	0	0
id2	0	1	0
id3	1	0	0
id4	0	0	1

Index Bitmap sur l'attribut **ville** :

id	paris	lyon
id1	0	1
id2	1	0
id3	1	0
id4	0	1

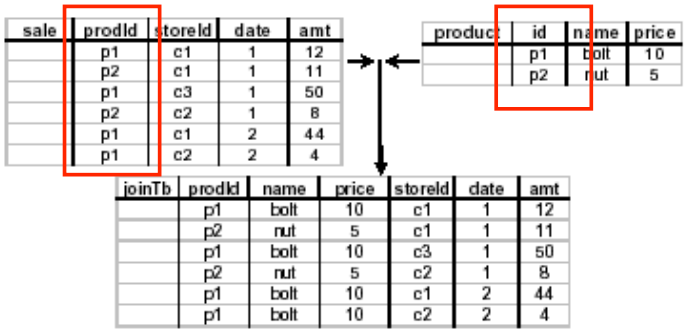
1. Rolap & Indexation : indexation binaire/bitmap index (5)

Occupation comparative B-tree / bitmap index :



1. Rolap & Indexation : index de jointure/join indexing (1)

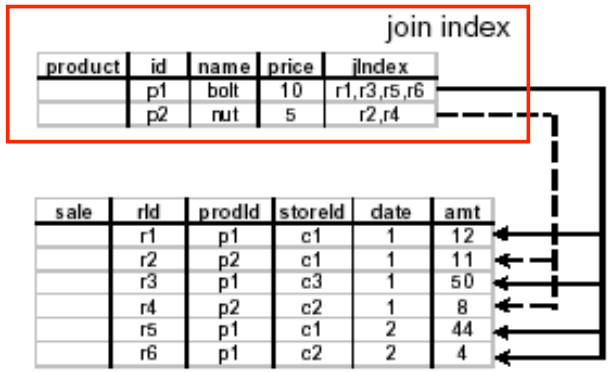
- **pré-calcul de jointure binaire**
- utilisé avec les **schéma en étoile** (stars schemas)
- **évite de calculer la jointure**
- **maintient les relations** entre :
  - une ***clé étrangère***
  - les ***clés primaires*** qui la contienne



(d'après H. Garcia-Molina)

1. Rolap & Indexation : index de jointure/join indexing (2)

Suite …



(d'après H. Garcia-Molina)

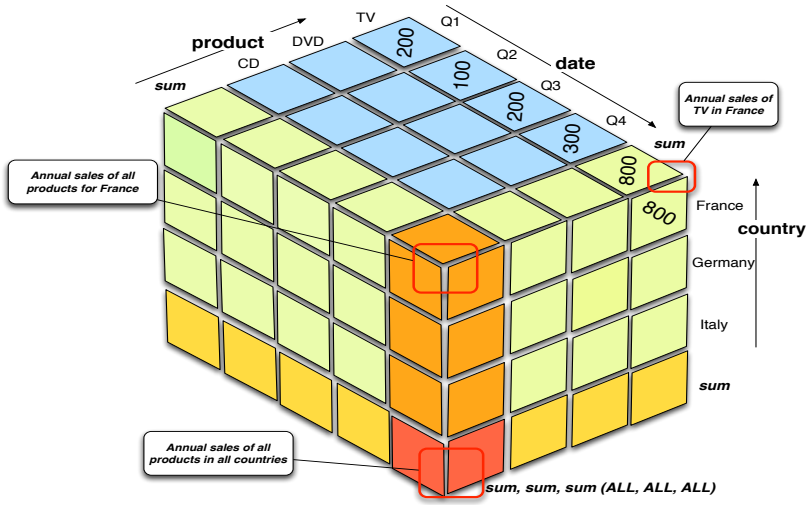
Remarque : les techniques **bitmap index** et **join index** peuvent être **combinées**

2. Sélection et matérialisation de vues : cube de données (1)

- Un DW est basé sur un modèle multidimensionnel où les données sont vues comme des **cubes de données (data cubes)**
- ex: Cube « **Ventes** », permet de voir les données selon plusieurs dimensions :
- Les tables de **dimension** : par ex:
  - Produit (nom_produit, marque, type)
  - Date (jour, semaine, mois, trimestre, année)
- La table de **faits** contient :
  - des mesures (ex: unités_vendues)
  - les clés externes faisant référence à chaque table de dimension
- un cube de dimension n est dit **cuboïde**
- le treillis des cuboïdes d'un entrepôt forme un **data cube**.

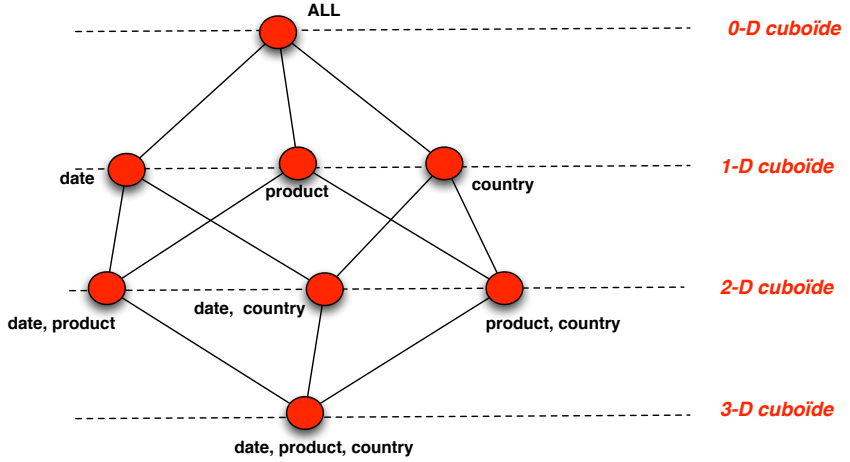
2. Sélection et matérialisation de vues : cube de données (2)

Exemple de Cube « Ventes » avec tous les agrégats possibles :



2. Sélection et matérialisation de vues : Cube et Cuboïdes

Cube = treillis de Cuboïdes :



2. Sélection et matérialisation de vues : pré-calcul d'agrégats

3 possibilités :

1. ne **pas stocker d'agrégat** : coûteux en temps
2. **stocker tous les agrégats** : coûteux en espace
3. ne **stocker qu'une partie des agrégats** : ... mais lesquels ?

Datacube = faits + tous les cuboïdes possibles :

- si le **cube est dense** : taille du datacube = taille de la table de faits
- si le **cube est creux** : chaque cuboïde = taille de la table de faits

Matérialiser des cuboïdes choisis en fonction :

- du **grain** (niveau d'agrégation) *le grain doit être suffisamment fin pour pouvoir répondre aux requêtes*
- des **requêtes utilisateurs** (frequently asked queries)

=> Matérialisation des cuboïdes grâce à des vues matérialisées

2. Sélection et matérialisation de vues : vues matérialisées (1)

- **vue matérialisée** = résultat du **calcul d'une vue stockée sur disque**
- elles sont utilisées pour **représenter les agrégations des tables d'un schéma en étoile**
- les **requêtes peuvent utiliser ces vues** (données pré-agrégées) pour **augmenter les performances**
- une vue matérialisée peut servir à **en construire d'autres**
- mais il est souvent **impossible de matérialiser toutes les vues**
  => sélection des vues à matérialiser en tenant compte :
  - du coût d'exécution des requêtes
  - du coût de maintenance (rafraîchissement)
  - du coût de calcul
  - de l'espace disque requis
- pour utiliser des vues matérialisées, une requête doit être **réécrite** : **la réécriture peut être difficile.**

2. Sélection et matérialisation de vues : vues matérialisées (2)

- **Exemple de Cube :**
  Ventes (modele, couleur, date, vendeur, prix, quantite)
- **Vue matérialisée des agrégats par** modele, couleur, mois, ville **:**

```
INSERT INTO ventesVue1
SELECT modele, couleur, mois, ville,
   SUM(prix) as prix, SUM(quantite) as quantite
FROM ventes, vendeur, temps
WHERE ventes.vendeur = vendeur.nom
AND ventes.date = temps.jour
GROUP BY modele, couleur, mois, ville
```

- **Vue matérialisée des agrégats par** modele, semaine, departement **:**

```
INSERT INTO ventesVue2
SELECT modele, semaine, departement
   SUM(prix) as prix, SUM(quantite) as quantite
FROM ventes, vendeur, temps
WHERE ventes.vendeur = vendeur.nom
AND ventes.date = temps.jour
GROUP BY modele, semaine, departement
```


2. Sélection et matérialisation de vues : vues matérialisées (3)

- **La requête Q1 :**

```
SELECT modele, SUM(prix)
FROM ventes
GROUP BY modele
```

- **peut être traitée par :**

```
SELECT modele, SUM(prix)
FROM ventesVue1
GROUP BY modele
```

ou

```
SELECT modele, SUM(prix)
FROM ventesVue2
GROUP BY modele
```

2. Sélection et matérialisation de vues : vues matérialisées (4)

- **La requête Q1 :**

```
SELECT modele, annee, departement, SUM(prix)
FROM ventes, vendeur, temps
WHERE ventes.vendeur = vendeur.nom
AND ventes.date = temps.jour
GROUP BY modele, annee, departement
```

peut être traitée par :

```
SELECT modele, annee, departement, SUM(prix)
FROM ventesVue1, vendeur, temps
WHERE ventesVue1.ville = vendeur.ville
AND ventesVue1.mois = temps.mois
GROUP BY modele, annee, departement
```

- **La requête Q1 :**

```
SELECT modele, couleur, date, SUM(prix)
FROM ventes
GROUP BY modele, couleur, date
```

Q3 ne peut être traitée ni via ventesVue1, ni via ventesVue2 :

- grain des vues trop gros
- ventesVue2 ne regroupe pas par couleur

3. Rolap & Fragmentation

- **Afin d'optimiser les requêtes OLAP , on peut fragmenter les tables du schéma en étoile :**
- **Découpage des tables du schéma en étoile :**
  - Par **fragmentation horizontale** : par **sélection**
  - Par **fragmentation verticale** : par **projection**
- Les **requêtes** sont **évaluées sur chaque fragment**
- L'obtention de la réponse **nécessite une requête de reconstruction :**
  - Par **union** pour la **fragmentation horizontale**
  - par **jointure** pour la **fragmentation horizontale**

Forces et faiblesses de la technologie ROLAP

Forces :

- s'appuie sur la **maturité de la technologie relationnelle**
- permet de **stoker de très grands volumes** de données
- permet la définition de données complexes et multidimensionnelles en utilisant un **modèle relativement simple**,
- réduit le nombre de jointures à réaliser dans l'exécution d'une requête

Faiblesses :

- peut conduire à des **temps de réponses élevés** : génération de SQL encore peu efficace
- ne peut effectuer des requêtes OLAP avec des **calculs complexes**
- ED structuré en **étoile ou flocon seulement**

Quelques produits de technologie ROLAP (1)

IBM DB2 UDB:

- SGBD fonctionne sur de nombreuses plate-formes
- il supporte le concept de « fédération » de bases de données relationnelles *(shared nothing database)*
- il dispose d'une gamme d'outils, notamment :
  - DB2 Performance Expert : permet la création de rapports, d'analyses et recommande des changements pour améliorer la performance.
  - DB2 Data Joiner: pour l'optimisation des requêtes SQL.
  - DB2 Integrated Cluster Environnement : pour le passage à l'échelle.

Oracle9i :

- SGBD fonctionne sur de nombreuses plate-formes
- il supporte des partitions de *hash*, *range* et *list*, ainsi que la consolidation sur une base de données centralisée (shared disk data base)
- il dispose d'une gamme d'outils, notamment :
  - Real Application Clusters : permet de désigner certains processeurs comme processeurs OLAP et d'autres comme processeurs de requêtes.
  - optimiser : basé sur les coûts ou sur les règles.

Quelques produits de technologie ROLAP (2)

SQL Server 2000 :

- Offre Microsoft
- il supporte le concept de « fédération » de bases de données relationnelles *(shared nothing database)*, et la liaison entre bases de données distribuées et hétérogènes
- il dispose d'une gamme d'outils, notamment :
  - L'optimiser de SQL Server : basé sur les coûts avec création automatique de statistiques et leur rafraîchissement
  - Le Query Processor : supporte des requêtes multidimensionnelles, ainsi que les index composites et semi-jointures.
  - Le SQL Query Analyzer : peut faire des suggestions par rapport à l'implantation des index additionnels et des statistiques complémentaires.
  - Microsoft DTS (Data Transformation Services) : outil ETL intégré dans Microsoft SQL Server.

Autres produits :

- Sybase IQ, DSS Agents de MicroStrategy, MetaCube de Informix, …

3 – Systèmes MOLAP

- **Introduction à la technologie MOLAP**
- **Techniques de stockage**
- **Densité et compression**
- **Agrégation et calcul des agrégats**
- **Forces et faiblesses de la technologie MOLAP**
- **Quelques produits de technologie MOLAP**

Stratégie « MOLAP » d'implantation d'un ED

- Utilise un SGBD Multidimensionnel (SGBDM) **capable de stocker et traiter des données multidimensionnelles**
- A ce jour **pas de cadre technologique commun** pour ces systèmes : chaque produit a sa **version du modèle multidimensionnel** et ses **stratégies** de stockage
- ont des **bonnes performances** du fait qu'ils effectuent la **pré-agrégation** et le **pré-calcul** des données sur tous les niveaux des hiérarchies du modèle de l'entrepôt
- **génèrent** de très **grands volumes d'information**,
- les **techniques incrémentales de rafraîchissement** associées sont encore limitées, conduisant à **reconstruire périodiquement** l'ED
- sont **adaptés** à de **petits ED** (quelques Go) et lorsque **le modèle multidimensionnel ne change pas beaucoup**

Introduction à la technologie « MOLAP » (1)

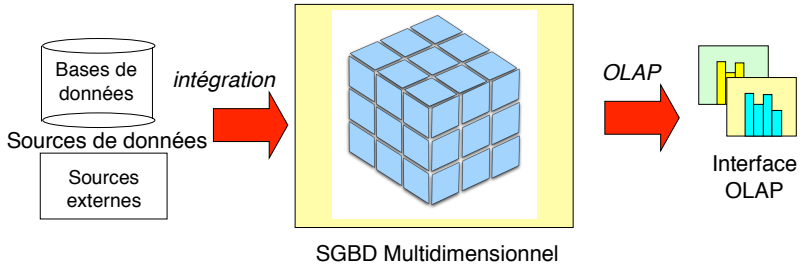
- MOLAP permet de stocker les données de manière multidimensionnelle
- Le calcul des agrégats dans un tel tableau se fait en colonne ou en ligne et est par conséquent très **rapide** : ***Pas de jointure à faire***
- **Taille limitée**
- **Pas de langage d'interrogation des données** : nécessité de redéfinir les opérations de manipulation de structures multidimensionnelles

Intérêt : les temps d'accès sont optimisés,

Inconvenient : nécessite de redéfinir des opérations pour manipuler ces structures multidimensionnelles.

Introduction à la technologie « MOLAP » (2)

- un **SGBD Multidimensionnel** est utilisé pour stocker l'ED :



- les systèmes MOLAP réalisent des pré-agrégations et des pré-culculs de données sur tous les niveaux de hiérarchies des dimensions de l'ED

Introduction à la technologie MOLAP (3)

Technologie des **bases de données multidimensionnelles** :

- structure de stockage = **tableaux**
- correspondance directe avec la vue multidimensionnelle
- les **membres** sont implicites :
  - constituent **l'adresse de la cellule**
  - sont **normalisés** (vis = 0, clous = 1, ...)

Gestion de la **faible densité** (sparsity) :

- techniques de **compression** spécifiques
- structure **d'index spécifiques**
- si le tableau est **dense**, la mémoire ne contiendra que les mesures

Introduction à la technologie MOLAP (4)

Exemple de tableau MOLAP :

- 1460 jours
- 200 000 produits ($2x10^5$)
- 300 magasins ($3x10^2$)
- promotion : 1 valeur booléenne (2)

Nombres de cellules du cube :

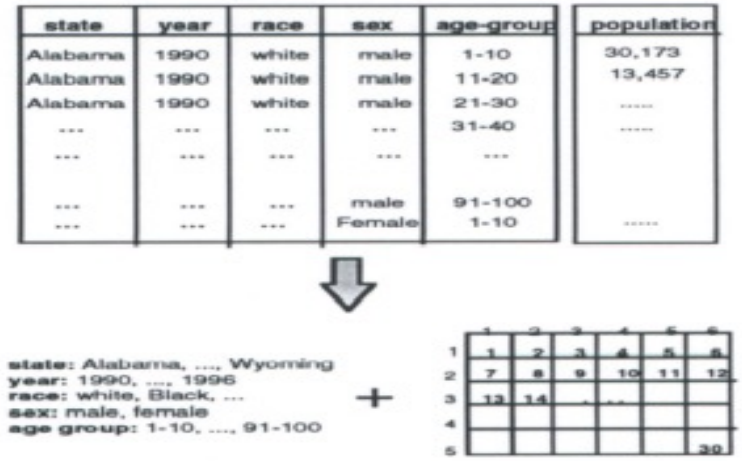
$1460 \times 2x10^5 \times 3x10^2 \times 2 = 1{,}75 \times 10^{11}$ cellules

Densité du cube :

- seulement 10% de produits vendu par jour :
- densité = $1{,}75 \times 10^{10} / 1{,}75 \times 10^{11} = 0{,}1$

Technologie MOLAP : stockage (1)

(D'après P. Marcel)



Technologie MOLAP : techniques de stockage (2)

(D'après P. Marcel)

87		73
	25	95
	89	62

implantation "row major"

87		73		25	95		89	62

a[0][0] . . . a[2][2]

- d dimensions, N_k membres dans la dimension k
- la fonction *p* donne la position dans le tableau en fonction de l'indice i_d :

$$p(i_1, \ldots i_d) = \sum_{j=1}^{d} (i_j \times \Pi_{k=j+1}^{d} N_k)$$

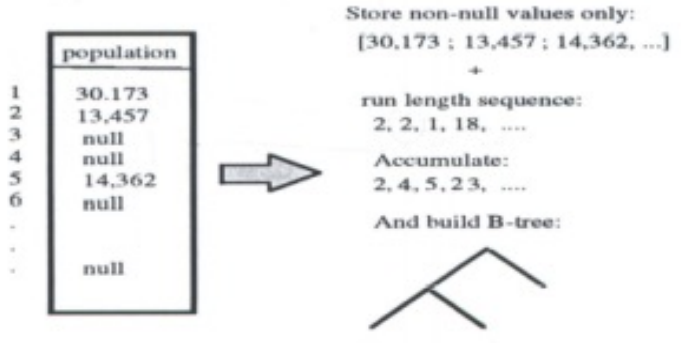
Exemple :

a[2][3][4] avec 3 dimensions de 8, 9 et 10 membres :

$$p(2,3,4) = 2 \times 9 \times 10 + 3 \times 10 + 4 = 214$$

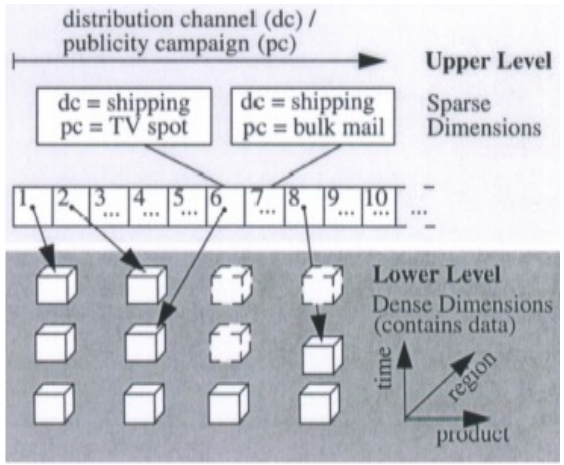
Technologie MOLAP : densité et compression

- Typiquement, jusqu'à **90 % de cellules vides**
- Stocker les données en **blocs denses**, pour cela on utilise des **techniques de compression** (similaires pour certaines à celles utilisées en relationnel)
- Ces techniques fonctionnant **bien pour 2 ou 3 dimensions échouent en 20 dimensions** :



Technologie MOLAP : indexation

Techniques d'indexation spécifiques :



(D'après P. Marcel)

Coût MOLAP des opérations Technologie MOLAP : agrégation et calcul des agrégats

Agréger = parcourir et appliquer la fonction d'agrégat sur des lignes du tableau :

- les agrégats peuvent être **calculés à la demande**
- **précalculés** et **stockés comme des lignes du tableau**

Ex cube c de dimension A,B,C group by A,C
naïvement :

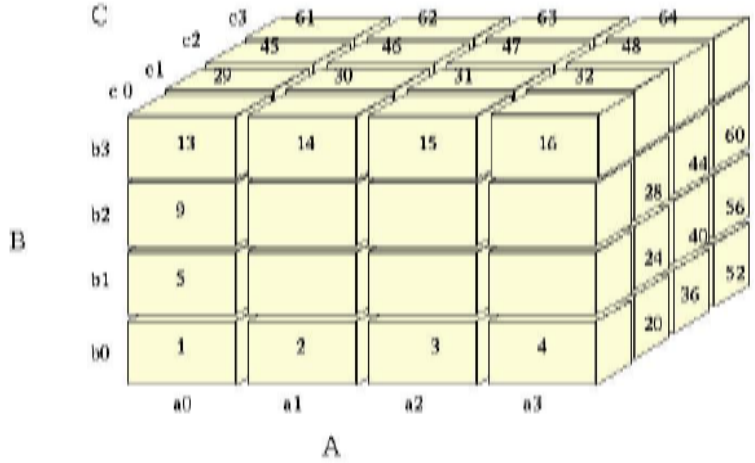
```
for(a=0;a<a max;a++)
   for(b=0;b<b max;b++)
      for(c=0;c<c max;c++)
         res[a][c] += c[a][b][c]
```

Calcul des agrégats en MOLAP :

1. **partitionner le tableau physique n dimensions en sous-cubes (chunks) :**
   - de n dimensions
   - tenant en mémoire principale
   - compressés pour gérer la faible densité
2. **calculer les agrégats :**
   - visiter chaque cellule de chaque sous-cube
   - calculer l'agrégat partiel impliquant cette cellule

Technologie MOLAP : agrégation et calcul des agrégats

(D'après P. Marcel)



Technologie MOLAP : agrégation et calcul des agrégats

minimiser le nombre de visite par cellule dans le calcul d'agrégats : exploiter l'ordre de visite pour calculer simultanément des **agrégats partiels** (réduction des accès mémoire et des coûts de stockage)

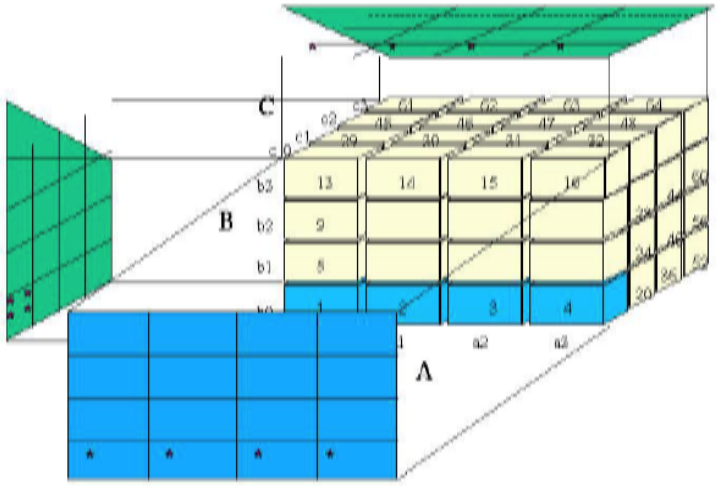
Exemple :

- cube à 3 dimension A, B, C
- tailles :

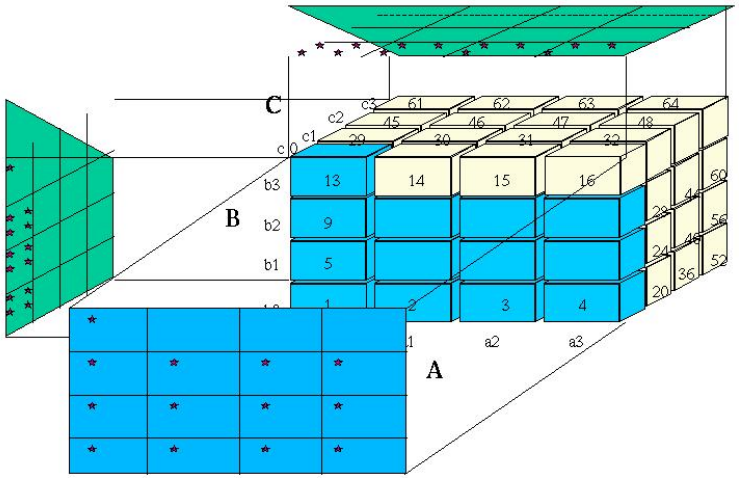
	taille
A	40
B	400
B	4000
BC	1 600 000
AC	160 000
AB	16 000

- dimensions partitionnées en 4 sous-cubes de même taille
  scan dans l'ordre 1, 2, 3, ..., 64 (BC, AC, AB) :
  - le calcul de b0c0 demande 4 scans (1, 2 , 3 , 4)
  - le calcul de a0c0 demande 13 scans (1, 5, 9, 13)
  - le calcul de a0b0 demande 49 scans (1, 17, 33, 49)

Technologie MOLAP : agrégation et calcul des agrégats



Technologie MOLAP : agrégation et calcul des agrégats



Technologie MOLAP : agrégation et calcul des agrégats

- **Occupation mémoire minimum :**

	16000	AB
+	10 × 4000	une colonne de AC
+	100 × 1000	un sous-cube de BC
=	156 000	

- **Scan dans l'ordre 1, 17, 33, 49, 5, 21, ... (AB, AC, BC)**
  le calcul de b0c0 demande 49 scans
  le calcul de a0c0 demande 13 scans
  le calcul de a0b0 demande 4 scans
- **Occupation mémoire minimum :**

	1 600 000	BC
+	10 × 4000	une colonne de AC
+	10 × 100	un sous-cube de AB
=	1 641 000	

- **Méthode** (valable pour un petit nombre de dimensions) **:**
  les cuboïdes doivent être calculés par taille croissante
  garder le plus petit cuboïdes en mémoire principale
  rechercher et calculer seulement un sous-cube à la fois pour le plus grand cuboïde

Coût MOLAP des opérations typiques

Coût MOLAP des opérations typiques :

opération	coût
roll-up drill-down	dépend de l'utilisation d'un cache
rotate slice-dice	élevé : accès au cube

(D'après P. Marcel)

Forces et faiblesses de la technologie MOLAP

Forces et faiblesses majeures de la technologie MOLAP :

Forces :

- Très bonnes performances dues à :
  - la réalisation de nombreuses **pré-agrégations** et de **pré-calculs** de données sur tous les niveaux de hiérarchies des dimensions de l'ED
  - **accès rapide** à une position d'un tableau (par son indice)

Faiblesses :

- Les pré-agrégations et les pré-calculs de données réalisés génèrent de très **importants volumes de données**
- quand la **taille** de l'ED **dépasse quelques Go** et que le **modèle multidimensionnel évolue, les performances se dégradent vite**
- Techniques incrémentales de rafraîchissement limitées : nécessité de **reconstruire périodiquement l'ED**

Quelques produits de technologie MOLAP (1)

Essbase (Arbor Software) :

- SGBD multidimensionnel, multi-utilisateurs,
- il dispose d'une gamme d'outils, notamment :
  - Hyperion Essbase Application Manager : fournit des outils graphiques, des modules pour la construction et le chargement des structures OLAP, pour le chargement des données, pour la définition des processus de calcul, pour la gestion des partitions de la base de données,...
  - Hyperion Essbase Query Designer : facilite la navigation des utilisateurs finaux.
  - Hyperion Essbase Partition Option : permet aux développeurs des applications, la création logique et physique de sous-ensembles des bases de données. 2 types de partitions sont supportés : transparente et liée.

SAS OLAP Server (SAS) :

- SGBD multidimensionnel
- il dispose d'une gamme d'outils, notamment :
  - SAS OLAP Cube Studio : pour la construction des cubes et qui peut être facilement utilisé pour la définition des mesures, des dimensions et des agrégations.
  - SAS Metadata Server : conteneur des métadonnées.

Quelques produits de technologie MOLAP (2)

Informix MetaCube (Informix) :

- SGBD multidimensionnel
- il dispose d'une gamme d'outils, notamment :
  - MetaCube Analysis Engine : Il fournit une interface multidimensionnel et étend la fonctionnalité d'analyse de la base de données avec l'incorporation des opérations OLAP, par exemple l'opération rotation (rotate).
  - MetaCube Explorer : C'est une interface graphique qui permet aux utilisateurs d'exécuter des requêtes au travers du MetaCube Analysis Engine.
  - MetaCube Warehouse Manager : C'est une interface graphique pour créer une description multidimensionnelle des tables et des colonnes dans l'entrepôt de données.
  - MetaCube Analysis Engine : stocke cette description multidimensionnelle comme un ensemble de tables.

Autres systèmes MOLAP :

- Pilot (Pilot Software) :
- TMI (Applix) :
- …

4 – Systèmes HOLAP

- **Introduction à la technologie HOLAP**
- **Quelques produits de technologie HOLAP**

Stratégie « HOLAP » d'implantation d'un ED (1)

Dans l'approche relationnelle (ROLAP) :

- 30% du temps est consacré aux entrées/sorties

Dans l'approche multidimensionnelle (MOLAP) :

- 20% du temps est consacré aux entrées/sorties

  (70% calculs et 10% décompression)

L'approche HOLAP (Hybride OLAP) :

- consiste à **utiliser les tables comme structure permanente de stockage des données**
- et les **tableaux comme structure pour les requêtes**.

Stratégie « HOLAP » d'implantation d'un ED (2)

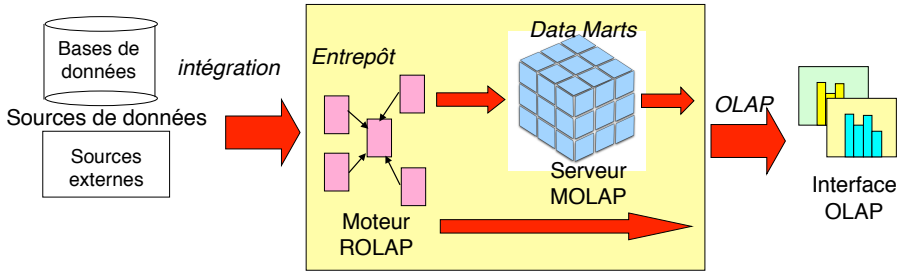
- **Stratégie hybride** : tire profit des avantages des technologies **ROLAP** et **MOLAP** en utilisant :
  - un **SGBD Relationnel** pour **stocker**, **gérer** les **données détaillées**
  - un **SGBD Multidimensionnel** pour **stocker**, **gérer** les **données agrégées**
- Permettent :
  - de gérer de **très grandes quantités de données**
  - d'avoir des **temps de réponses acceptables** en analyse OLAP
- **Produits** : Express d'Oracle, Media/MR de Speedware, Holos de Seagate Technology, …

Introduction à la technologie HOLAP (1)

- Les **systèmes HOPAL** essaient de **combiner** les bons cotés des systèmes ROLAP et MOLAP :
  - En stockant les **données détaillées** de **l'ED** dans un SGBD Relationnel - **ROLAP**
  - En stockant les **données agrégées**, souvent des **magasins de données** (data marts) de l'ED dans un SGBD Multidimensionnel - **MOLAP** :
    - **granularité moins fine**
    - **index en mémoire centrale**
- Ils permettent ainsi d'avoir des **ED de taille importante** tout en ayant des **temps de réponse satisfaisants**.

Introduction à la technologie HOLAP (2)

- De nombreux systèmes commerciaux utilisent l'approche HOLAP :
  - ils manipulent les informations de l'entrepôt de données avec un **moteur ROLAP** et
  - ils exploitent les « data marts » avec **une approche multidimensionnelle MOLAP**
- architecture générale de ces systèmes :



Quelques produits de technologie HOLAP

DB2 OLAP Server :

- Il permet de calculer, de consolider et d'accéder l'information à partir des bases de données multidimensionnelles, relationnelles ou les deux
- ses composants sont :
  - DB2 OLAP Integration Server : Il utilise des outils graphiques et des services pour l'intégration des données.
  - DB2 OLAP Server Administration Services : Il fournit des outils pour améliorer et faciliter des tâches d'administration.

Oracle Express Server :

- SGBD exploitant un modèle de données multidimensionnel.
- Il gère un ensemble d'indicateurs à n dimensions, dont les valeurs sont stockées ou calculées dynamiquement
- stockage des données : dans BD multidimensionnelle ou relationnelle :
  - La base Oracle Express Server : Stocke les agrégats multidimensionnels, tandis que les données de détail sont stockées dans la base relationnelle.
  - En utilisant un 4GL Express Server propose des fonctions avancées pour la présentation et l'analyse des résultats

Autres produits HOLAP : Media/MR (Speedware), Hollos (Seagate), …