Entrepôts de données et OLAP

Dr Sekhri Arezki
Departement des Sciences de Gestion
Université d'Oran 2

Entrepôt de données (Datawarehouse)

Collection d'informations provenant de **sources diverses** (bases de données existantes), destinées à servir de support en vue de **l'aide à la décision** (Decision Support System, OLAP, Datamining).
Cette BD pour l'aide à la décision est **séparée** des bases de données opérationnelles.

Les données sont extraites, regroupées (agrégées), corrélées avec d'autres informations, transformées, filtrées, de façon à obtenir un **système informationnel** à partir de systèmes opérationnels.

L'entrepôt de données est un **système transversal**, qui complète les systèmes opérationnels.

Marché mondial **en pleine expansion** (source IDC, 1996, en milliard de US$):

1993 : 0.8 1994 : 1.1 1996 : 2.0 1998: 3.7 2000: 5.6

Motivations

- Réconciliation sémantique
  - Dispersion des sources de données au sein d'une entreprise
  - Différents codage pour les mêmes données
  - L'entrepôt rassemble toutes les informations au sein d'un unique schéma
  - Conserve l'historique des données
- Performance
  - Les données d'aide à la décision nécessitent une autre organisation des données
  - Les requêtes complexes de l'OLAP dégradent les performances des requêtes OLTP.
- Disponibilité
  - La séparation augmente la disponibilité
  - Une bonne façon d'interroger des sources de données dispersées
- Qualité des données

Bases de données/Entrepôts (1)

Les **SGBD** sont des systèmes conçus pour l'OLTP (On-Line Transaction Processing).

Permet d'**insérer, modifier, interroger** des informations rapidement, efficacement, en sécurité.

Deux **objectifs** principaux :

- ajouter, retrouver et supprimer des enregistrements repérés par **une clef**

"rechercher une aiguille dans une botte de foin"

- ces opérations doivent pouvoir être effectuées très rapidement, et par de **nombreux utilisateurs simultanément**.

Les systèmes OLTP sont mal adaptés à l'analyse de données.

Bases de données/Entrepôts (2)

Les entrepôts sont des systèmes conçus pour **l'aide à la prise de décision**.

Les objectifs principaux sont

regrouper, organiser, coordonner des informations provenant de sources **diverses**,
les **intégrer** et les **stocker** pour donner à l'utilisateur une vue orientée métier,
retrouver et **analyser** l'information facilement et rapidement.

Questions typiques :

Quels sont les produits qui se vendent le mieux dans chaque région, et quel est l'impact des données démographiques sur ces résultats de vente ?

Bases de Données/Entrepôts (3)

	BD- OLTP	Entrepôts
Objectif	collecte de données opérations au jour le jour	consultation et analyse
Utilisateurs	un département (Employé)	transversal (Gestionnaire)
Types de données	données de gestion (données courantes)	données d'analyse (données historiques)
Informations	détaillées	détaillées + agrégées
n-uplets accédés	dizaines	millions
Opérations	requêtes simples, pré-déterminées sélections et mises à jour nombreuses transactions transactions courtes temps réel recherche d'enregistrements détaillés	requêtes complexes, ad-hoc sélections peu de transactions transactions longues batch agrégations et group by

Bases de données/Entrepôts(4)

Un entrepôt recouvre un horizon bien plus long dans le temps que les systèmes de production.

Il inclut de nombreuses bases de données «travaillées» de façon à définir les données uniformément.

Il est optimisé pour répondre à des questions complexes pour décideurs et analystes.

Bases de données/Entrepôts(5)

Les entrepôts sont physiquement séparés des systèmes de production,
pour des raisons de

Performance : les données des systèmes de production ne sont pas organisées pour pouvoir répondre efficacement aux requêtes des systèmes d'aide à la décision. Même les requêtes simples peuvent dégrader sérieusement les performances.

Accès aux données: un entrepôtdoit pouvoir accéder aux données uniformément, quelle que soit la provenance des données.

Formats des données: les données des entrepôts sont transformées, et doivent être disponibles sous un format simple et unique.

Qualité des données: les données d'un entrepôt sont propres et validées. La qualité des données est vue au sens large du décisionnel, et ne peut être réalisée qu'après comparaison avec d'autres éléments.

Caractéristiques

Dans un entrepôt, les **données** sont

- orientées **par sujets** :

  Les données organisées par sujet (clients, vendeurs, production,etc.) contiennent seulement l'information utile à la prise de décision.
  Les systèmes opérationnels sont plutôt orientés autour des traitements et des fonctions.

- **intégrées** :

  Les données, provenant de **différentes sources** (systèmes légués) sont souvent **structurées et codées de façons différentes**. L'intégration permet d'avoir une représentation **uniforme, cohérente et transparente**. Lorsque les données sont agrégées, il faut s'assurer que l'intégration est correcte.

- **historiques** :

  Un entrepôt contient des données "anciennes", datant de plusieurs années, utilisées pour des comparaisons, des prévisions, etc.

- **non volatiles** :

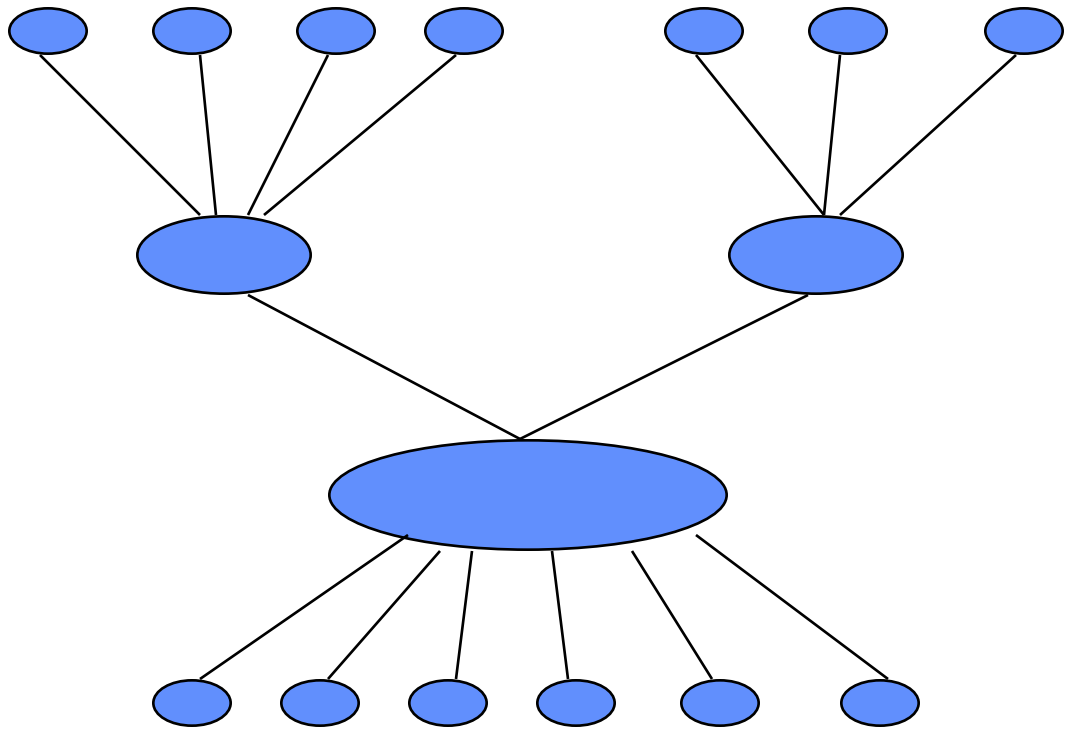
  Une fois chargées dans l'entrepôt, les données ne sont plus modifiables. Elles sont uniquement accessibles en lecture (pour l'instant...).

Fonctions des entrepôts

- **Récupérer** les données existantes des différentes sources
- **Référencer** les données de manière uniforme
- **Stocker** les données (notamment historisées)
- **Mettre à disposition** les données pour :
  - interrogation
  - visualisation
  - analyse

Structure des données

Un entrepôt de données contient 5 types de données :



METADONNEES

fortement résumées

faiblement résumées

données courantes

données anciennes

Structure des données

Données de détail courantes:

reflètent les faits les plus récents (les plus intéressants).
sont généralement stockées sur le disque ==> accès rapide.
peuvent devenir volumineuses (si on a un bas niveau de granularité).
peuvent être une copie (réplique) des données de la transaction de chargement.

Données de détail anciennes :

même niveau de détail que le précédent
stockées sur mémoire de masse ==> accès moins rapide
peu souvent interrogées

Structure de données

Données faiblement résumées :

structurées autour du plus faible niveau de détail des données courantes.
généralement stockées sur le disque.
doivent permettre de répondre rapidement aux questions standards des systèmes d'aide à la décision.
choix des attributs à résumer ?
fréquence des mises à jour ?

Données fortement résumées :

doivent être compactes et facilement accessibles.

Métadonnées

• Les métadonnées jouent un **rôle central** dans l'alimentation de l'entrepôt.

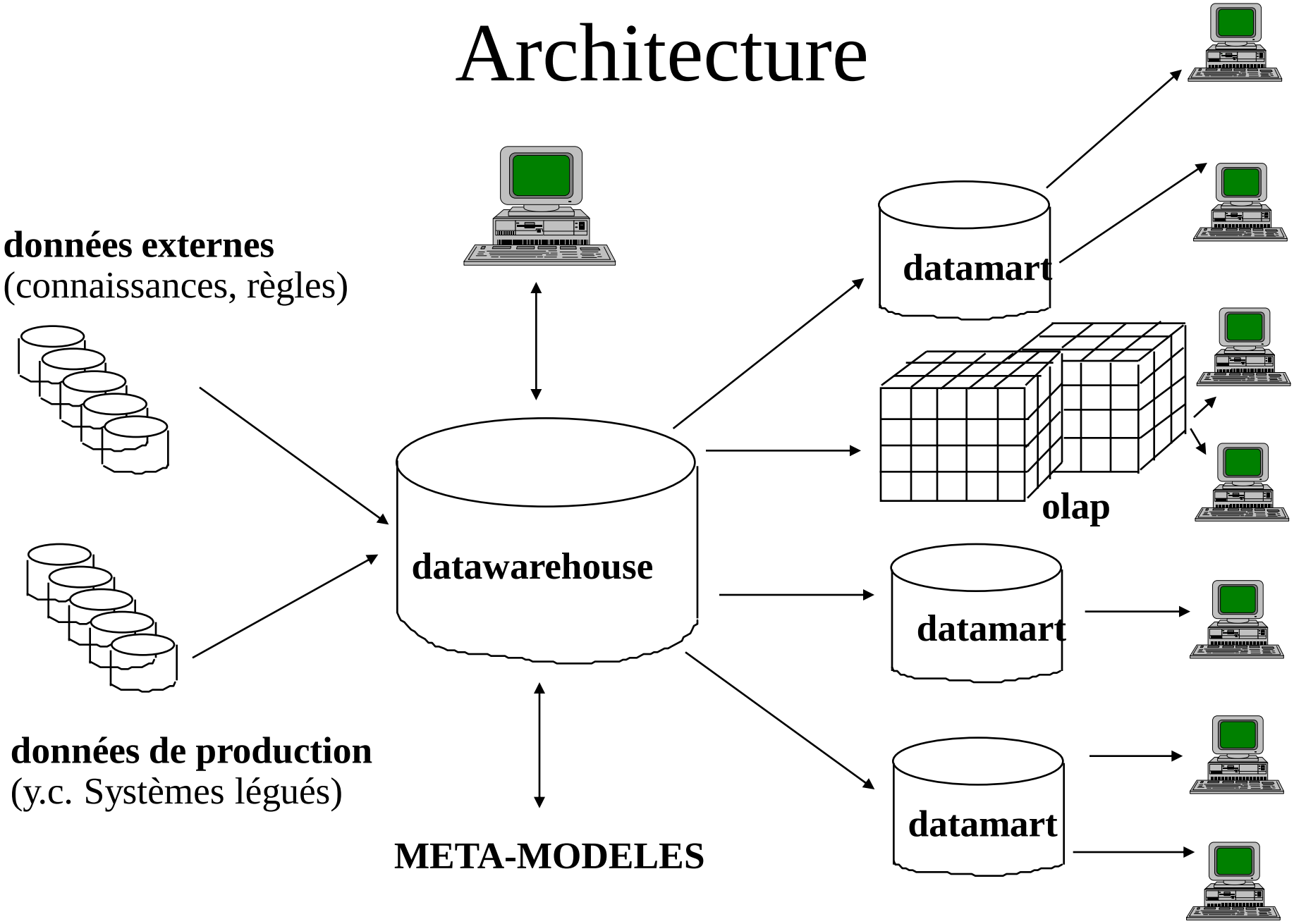
• Ce sont les "**données sur les données**".

• Elles sont utilisées lors de l'extraction, l'agrégation, la transformation, le filtrage et le transfert des données.

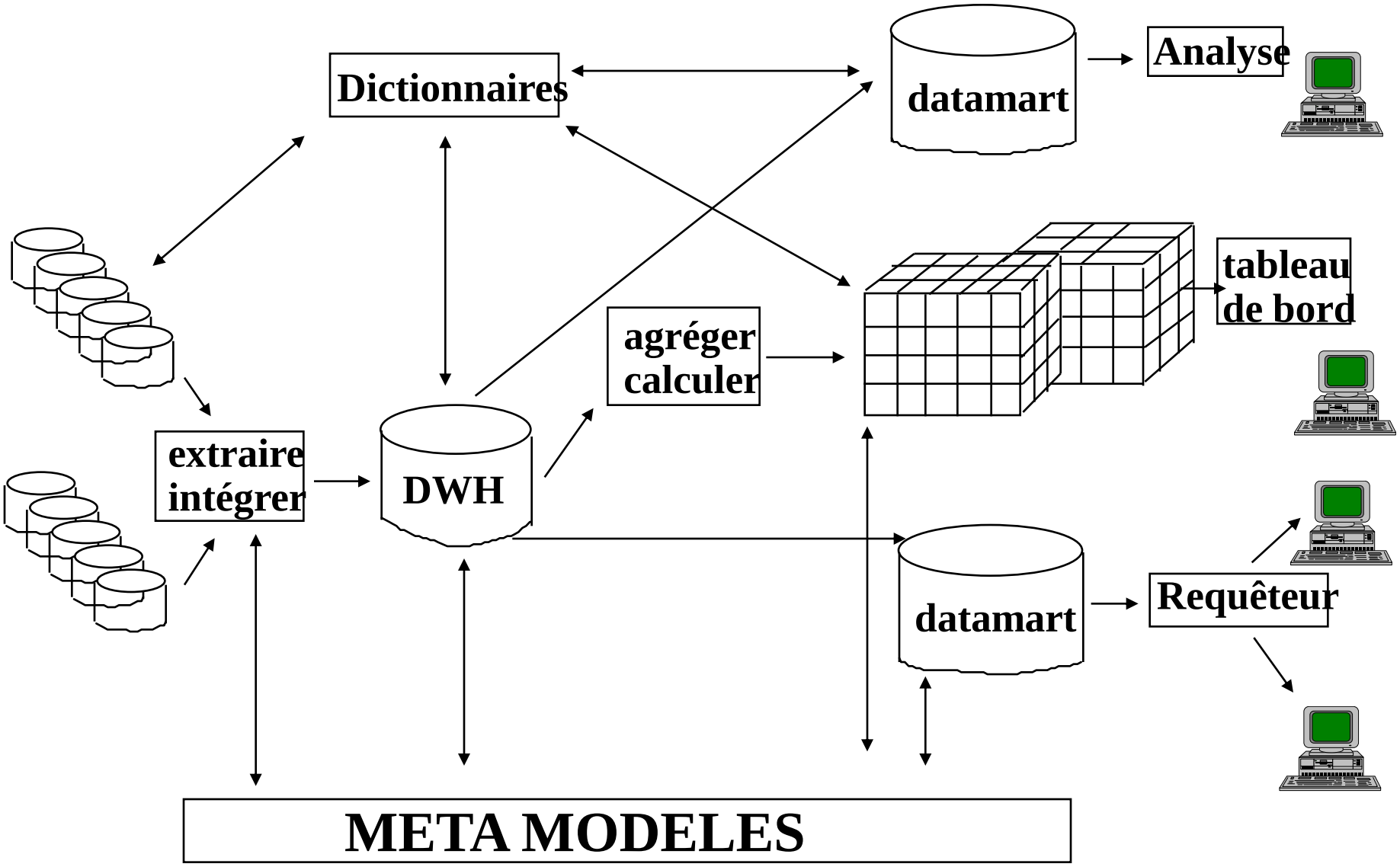
•Le méta-modèle constitue le **référentiel unique**:

- utilisateurs, profils et droits
- applications
- modèles de données, structure des données
- règles d'agrégation et de calcul

Architecture



Architecture



Architecture à 3 niveaux

- Serveur de la BD de l'entrepôt
  - Presque toujours relationnel
- Data marts /serveur OLAP
  - Relationnel (ROLAP)
  - Multidimensionnel (MOLAP)
- Clients
  - Outils d'interrogation et de rapports
  - Outils d'analyse et d'aide à la décision

Conception

- Deux approches:
  - Global as View : intégrer des schémas existants en un schéma global
    - Préintégration : quels schémas intégrer, dans quel ordre, quel modèle choisir, …
    - Comparaison de schémas : déterminer les corrélations, les conflits, etc.
    - Résolution de conflits de schémas hétérogènes
    - Fusion et restructuration
  - Local as View : processus inverse, les schémas sources sont exprimés comme des vues sur un schéma central.

Pré-intégration

En général, les BD à intégrer sont hétérogènes. Pour l'intégration, trois problèmes se posent:

Hétérogénéité des modèles de données :

trouver un modèle commun, et traduire les schémas dans ce modèle.

- **l'orienté objet,** modèle le plus riche sémantiquement, rend l'intégration complexe à cause des différents choix de modélisation des concepteurs.
- **des modèles très simples**, avec un minimum de sémantique (pas d'alternative de modélisation), qui facilitent l'intégration.

Hétérogénéité des puissances d'expression :

certains modèles, pauvres sémantiquement, conduisent à des ambiguïtés sur l'interprétation du schéma (ex: fichiers, modèle relationnel). On peut utiliser la rétro-ingéniérie, qui permet de distinguer dans une relation les objets, les attributs, les associations, les liens de généralisation /spécialisation.

Pré-intégration

Hétérogénéité des modélisations:

le processus de modélisation n'est pas déterministe. On peut réduire les différences en imposant des règles de modélisation, et des règles de normalisation.

Pour les modèles objet, on a des règles de **normalisation syntaxique.**
(ex : un type avec attribut optionnel doit être remplacé par une structure supertype/sous-type, le sous-type contenant cet attribut).

Les règles de **normalisation sémantique** visent à enrichir la sémantique du schéma (ex: s'il existe une dépendance entre deux attributs A et B de même type, et si A n'est pas une clé, remplacer ces attributs par un tuple composé de A et B).
Manque de maturité.

Identification des correspondances

Identifier les éléments communs des bases existantes : considérer ce qui est représenté, plutôt que comment c'est représenté.

Pour **définir une assertion de correspondance inter-schéma** (définition intentionnelle d'une correspondance), il faut

- établir les éléments en correspondance
- voir comment leurs extensions potentielles sont liées (équivalence, inclusion, disjonction, intersection des ensembles)
- déterminer comment identifier les instances en correspondance
- savoir comment les représentations sont liées.

Il faut s'assurer ensuite que l'ensemble d'assertions est **cohérent** et **minimum.**

Intégration

Chaque assertion est analysée pour déterminer la représentation des éléments en correspondance qui sera incluse dans le schéma intégré, et pour définir les règles de traduction entre le schéma intégré et les schémas initiaux.

Souvent, les types en correspondance présentent des différences, créant des conflits:

Classification : les populations du monde réel représentées par les deux types sont différentes.

Description : les types ont des ensembles différents de propriétés.

Structure : les concepts utilisés pour décrire les types sont différents.

Hétérogénéité : les modèles de données utilisés sont différents.

Données : des instances en correspondance ont des valeurs différentes pour des propriétés en correspondance.

Construction d'un entrepôt de données

Trois phases principales

1. Acquisition:

Extraction : collection de données utiles

Préparation : transformation des caractéristiques des données du système opérationnel dans le modèle de l'entrepôt

Chargement : nettoyage (élimination des dupliqués, incomplétudes, règles d'intégrité, etc.) et chargement dans l'entrepôt (trier, résumer, calculs, index).

2. Stockage :

Les données sont chargées dans une base de données pouvant traiter des applications décisionnelles.

3. Restitution des données :

Il existe plusieurs outils de restitution (tableaux de bord, requêteurs SQL, analyse multidimensionnelle, data mining ...)

Processus de chargement

- Extraction de données
  - Snapshots ou différentiels des sources
  - Transferts, cryptage, compressions, etc.
  - Objectif : minimum de changement par rapport aux sources
- Transformation
  - Résolution des conflits au niveau du schéma (différents attributs pour la même information)
  - Identification des valeurs et réconciliation
- Nettoyage
  - Élimination des dupliqués, contraintes d'intégrité, etc.
  - Données incomplètes ou absentes
- Chargement
  - Tris, résumés, calculs divers, etc.
  - Pbs: très grand volume de données, efficacité, quand calculer les index et les tables de résumés, reprise sur panne

Maintenance

- Les données de l'entrepôt sont stockées sous forme de vues matérialisées sur les différentes sources de données.
- Quand répercuter les mises à jour des sources ?
  - À chaque modification ?
  - Périodiquement ?
  - À définir par l'administrateur
- Comment les répercuter ?
  - Tout recompiler périodiquement ?
  - Maintenir les vues de façon incrémentale
    - Détecter les modifications (transactions, règles actives, etc.)
    - Les envoyer à un intégrateur qui détermine les vues concernées, calcule les modifications et les répercute.

Outils d’extraction de données

• Les requêteurs génèrent des requêtes SQL ad hoc (GQL, Reporter, Impromptu).

• Les tableaux de bord prédéfinis, consultables à l’écran, génèrent des états (histogrammes, camemberts, ...)

• Les outils de data mining permettent d’extraire des informations implicites de la base. Ils utilisent des techniques de classification, de segmentation, d’apprentissage symbolique et numérique, des statistiques, des réseaux neuronaux.
(Enterprise Miner, Intelligent Miner, KnowledgeSeeker, STATlab,...)

• Les analyseurs permettent de gérer les données multidimensionnelles (Outils OLAP: Explorer, PowerPlay, Metacube Explorer, ...)

Evolution

Les entrepôts sont amenés à évoluer **souvent** et **considérablement**.
La taille d'un entrepôt **croît rapidement** (de 20giga à 100giga en 2 ans).

Pourquoi ?

- nouvelles données (extension géographique, changement de fréquence des historiques, changement du niveau de détail, etc.)
- ajout de nouveaux éléments de données au modèle (l'ajout d'un attribut pour 2millions de n-uplets représente une augmentation considérable!)
- création de nouveaux index, résumés
- ajout de nouveaux outils (générateurs de requêtes, outils OLAP, etc.)
- nouveaux utilisateurs
- complexité des requêtes

Comment garantir l'**extensibilité**, la **disponibilité**, la **maintenabilité** ?

Evolution

Les prototypes ne sont guère utiles (ne dépassent pas 20giga), les estimations sont souvent erronées...

Trois aspects majeurs sont concernés :

la **base de données** doit être extensible, disponible (plus de batch), facilement gérable (optimisation, indexation, gestion du disque automatiques).

le **middleware** (gestionnaire de transactions, gestionnaire d'accès,...) doit être performant et cohérent. Là aussi, extensibilité, disponibilité, facilité de gestion.

l'intégration des outils doit se faire avec un souci de compatibilité. Les outils doivent être conformes au plus grand nombre de standards.

Problèmes ouverts dans les entrepôts

- **Alimentation des entrepôts :** 1/3 de la taille du projet. Simplifier le processus pour en alléger le coût.
- **Maintenance** : maintenir des vues matérialisées. Pb de cohérence. Pb de mise à jour à travers les vues.
- **Intégration de schéma** : les différentes sources ont des schéma différents, le DW doit avoir un schéma global unique. Comment faire ?
- **Effet taille** : les DW grossissent "`à vue d'oeil" (victimes de leur succès), vers des tera-octets. Comment gérer cela ? Les solutions actuelles seront-elles viables ?
- **Coût d'un DW prohibitif** pour les petites entreprises. Commencer par ne faire que des data-marts et les intégrer peu à peu...

On-Line Analytical Processing (OLAP) & BD multi-dimensionnelles

Introduction

• Evolution des structures de données stockées : fichiers, bases de données hiérarchique, **bases de données relationnelles**, bd. distribuées, bd. objet (multimedia).

⇨ de + en + près de la façon dont les gens *visualisent* et *manipulent* les données

• Bases de données relationnelles [Codd70] :

•Modèle simple : tables bi-dimensionnelles. Facile à mettre en oeuvre.

•Le plus répandu. La plupart des DW ont leurs données venant des BD relationnelles.

•BD relationnelles adaptées à l'OLTP (On-Line Transaction Processing) grâce au standard SQL. Beaucoup de transactions et de requêtes simples, sur peu de données à la fois. Gestion, production.

•BD relationnelles peu adaptées à l'OLAP (On-Line Analytical Processing).. Requêtes moins fréquentes mais plus complexes, longues, nécessitant une reformulation (agrégation) des données de masse. Analyse de données massives (giga, tera octets) stockées dans le DW pour l'aide à la décision.

On-Line Analytical Processing & BD multi-dimensionnelles
Les besoins

•Transformer les données brutes (ex. ventes pour chaque produit, jour, fournisseur) en de l'info. stratégique pour les analyses ultérieures

•Décideurs : finances, ventes, budget, marketing...

•Calculs complexes (analyse de tendance, moyennes mobiles, équations algébriques..) mais requêtes exprimées simplement

"Total des ventes pour chaque produit par trimestre"

"les 5 meilleurs fournisseurs pour chaque catégorie de produit l'an dernier"

"pour chaque produit, sa part de marché dans sa catégorie par rapport à celle de 1994"

"les fournisseurs dont les ventes ont augmenté dans chaque catégorie de produit ces 5 dernières années"

...

On-Line Analytical Processing & BD multi-dimensionnelles
Les besoins (suite)

•Avoir des résultats à temps, pouvoir raffiner, prolonger ou reprendre les analyses → besoin de garder et visualiser les résultats intermédiaires.

•Appréhender (visualiser) les données dans les dimensions naturelles pour les décideurs

Total des ventes pour chaque produit par trimestre

	Trim1	Trim2	Trim3	Trim4
Prod1	12	14	32	22
Prod2	15	26	34	7
...				

⇨ Total des ventes par catégorie de produit par fournisseur par trimestre (+ de détail sur les fournisseurs, - sur les produits)

Table (relation) ***VentesVoiture***

Attributs

nuplets

Modele	**Couleur**	**Ventes**
Clio	Bleu	180
Clio	Rouge	244
Jaguar	Bleu	318
Jaguar	Rouge	204
Espace	Bleu	131
Espace	Blanc	153

On veut des détails sur les vendeurs...

VentesVoiture

Modele	Couleur	Vendeur	Ventes
Clio	Bleu	Toto	12
Clio	Rouge	Toto	23
Clio	Bleu	Titi	34
Clio	Rouge	Titi	45
Clio	Bleu	Tata	56
Clio	Rouge	Tata	67
Clio	Bleu	Tutu	78
Clio	Rouge	Tutu	89
Jaguar	Bleu	Toto	90
Jaguar	Rouge	Toto	09
Jaguar	Bleu	Titi	98
Jaguar	Rouge	Titi	87
Jaguar	Bleu	Tata	76
Jaguar	Rouge	Tata	65
Jaguar	Bleu	Tutu	54
Jaguar	Rouge	Tutu	43
Espace	Bleu	Toto	32
Espace	Blanc	Toto	21
Espace	Bleu	Titi	11
Espace	Blanc	Titi	22
Espace	Bleu	Tata	33
Espace	Blanc	Tata	44
Espace	Bleu	Tutu	55
Espace	Blanc	Tutu	66

Vendeurs

Vendeur	Succursale
Toto	Auto+
Titi	Auto+
Tata	BelleCaisse
Tutu	BelleCaisse

•**BD relationnelle** : présentation peu conviviale. Données brutes. Normalisation

•**OLTP (SQL)** : mises à jour. Requêtes simples "qui, quoi" (ex. les ventes de Toto.

•Pour l'analyse, besoin de données agrégées, synthétisées

•Possibilité d'agréger (fonctions *de base* uniquement) les données sur une seule table (ex. : somme des ventes par modèle), mais très coûteux (parcourir toutes les tables) et recalcul à chaque étape, à chaque utilisation → *tables de résumés*

•Sur plusieurs tables (ex : somme des ventes par succursale), nécessité de faire des jointures coûteuse (ici 24 * 4 = 96 comparaisons) → *dénormalisation*

• **Idée** : stocker les résultats des fonctions agrégées (ex: table des ventes par vendeur, table de la moyenne des ventes par vendeur de chaque succursale,...) les plus fréquemment utilisés.

• **Problèmes** :

•tables de résumés pré-définies. Ce qui n'est *pas prévu* n'est *pas disponible*.

•prolifération des tables de résumés →environnement de décision complexe et confusant

Utilisateurs doivent savoir *quelles tables de résumé* sont disponibles et ce à *quoi elles correspondent*.

•Il faut *rafraîchir* par rapport au données brutes → ne pas interférer la production (concurrence)

le calcul des agrégats nécessite de lire toutes les données d'une table et donc impose le *verrouillage de tous ses n-uplets*.

On-Line Analytical Processing (OLAP) & BD multi-dimensionnelles
Dénormalisation

•**Idée** : Stocker toutes les informations dans une seule table pour éviter les jointures.

Modele	**Couleur**	**Vendeur**	**Ventes**	**Succursale**	**Enfants**
Clio	Bleu	Toto	12	Auto+	Nono
Clio	Bleu	Toto	12	*Auto+*	Nana
Clio	Rouge	Toto	23	*Auto+*	*Nono*
Clio	Rouge	Toto	23	*Auto+*	*Nana*
Clio	Bleu	Titi	34	Auto+	--
Clio	Rouge	Titi	45	*Auto+*	--
Clio	Bleu	Tata	56	BelleCaisse	--
Clio	Rouge	Tata	67	*BelleCaisse*	--
Clio	Bleu	Tutu	78	BelleCaisse	--
Clio	Rouge	Tutu	89	*BelleCaisse*	--
Jaguar	Bleu	Toto	90	*Auto+*	*Nono*
Jaguar	Bleu	Toto	90	*Auto+*	*Nana*
Jaguar	Rouge	Toto	09	*Auto+*	*Nono*
. . .	. . .	. . .	. . .	. . .	. . .
Espace	Blanc	Tutu	66	*BelleCaisse*	--

•**Problèmes** :

- •Grande redondance (ex. succursale et surtout enfants de Toto) → place disque ↑ performances ↓
- •Diminue la densité du stockage des données (colonne Enfants *creuse*) → place disque gaspillée
- •Augmente aussi la taille et le nombre des index
- •Seule alternative → "plus d'acier", i.e.. augmenter les capacités matérielles

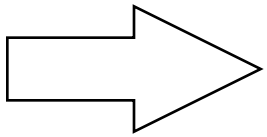
On-Line Analytical Processing (OLAP) & BD multi-dimensionnelles
Vers une autre représentation des données

•*Les bases de données relationnelles* ne sont pas adaptées à l'OLAP car les tables représentent une *vue aplatie de structures naturellement multi-dimensionnelles*.

•Non seulement perte de performances mais aussi nécessité pour les utilisateurs de savoir comment trouver les liens entre les tables pour recréer la vue multi-dimensionnelle.

•Il est donc nécessaire de disposer d'une *structure de stockage adaptée à l'OLAP*, i.e. permettant de

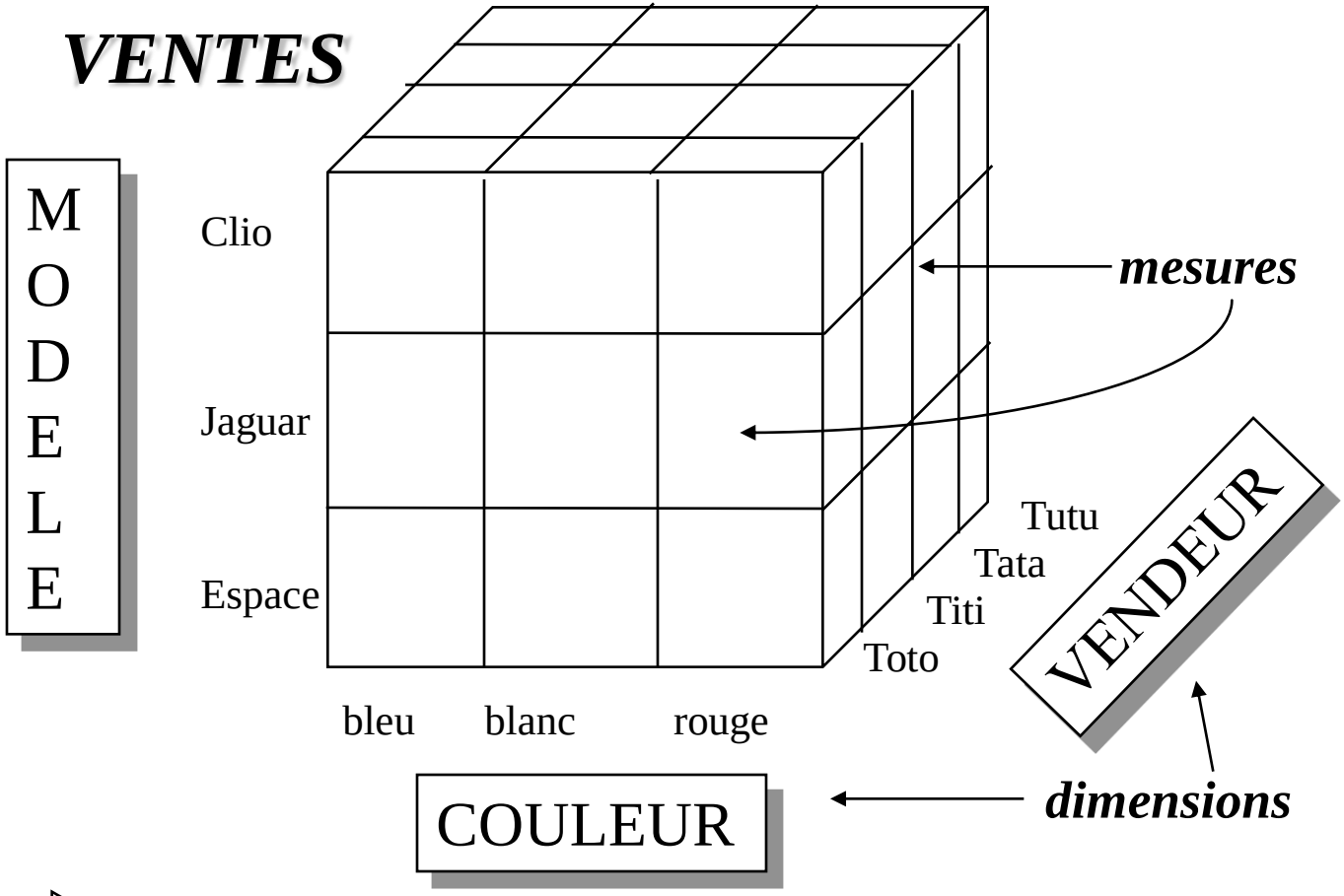
- •*visualiser* les données dans *plusieurs dimensions* naturelles,
- •de pouvoir *définir et ajouter des dimensions* facilement
- •de *manipuler* les données ainsi représentées facilement et efficacement.



Bases de données multi-dimensionnelles ("Cube")

On-Line Analytical Processing (OLAP) & BD multi-dimensionnelles
Bases de données multidimensionnelles (1)

- Une *BD multidimensionnelle* [Gray & al. - VLDB'96] est un hyper-cube :
  - Les axes sont appelés *dimensions* définies par l'utilisateur
  - Les points dans l'espace (cellules) contiennent des *mesures* calculées à partir de formules plus ou moins complexes.
  - Les *opérateurs* sur le cube sont *algébriques* (retournent un cube) et peuvent ainsi être *combinés*



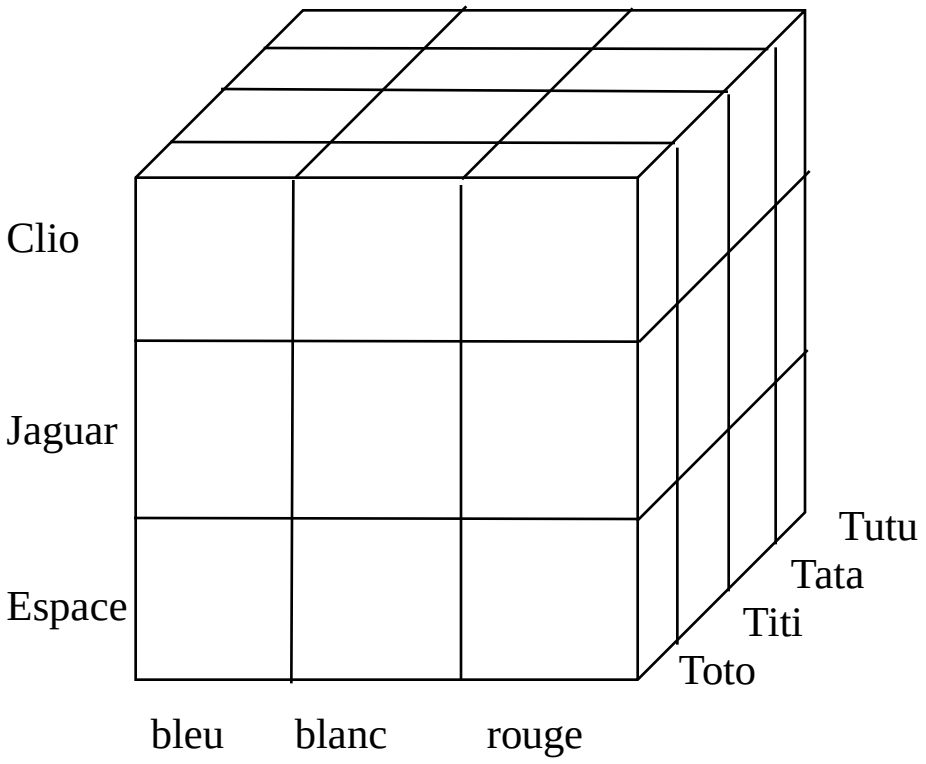
⇨ Base de données multi-dimensionnelle = "super-tableur"

On-Line Analytical Processing (OLAP) & BD multi-dimensionnelles
Bases de données multidimensionnelles (2)

•Représenter ce cube nécessiterait en relationnel une table de 4*3*3 = 36 nuplets à "balayer"

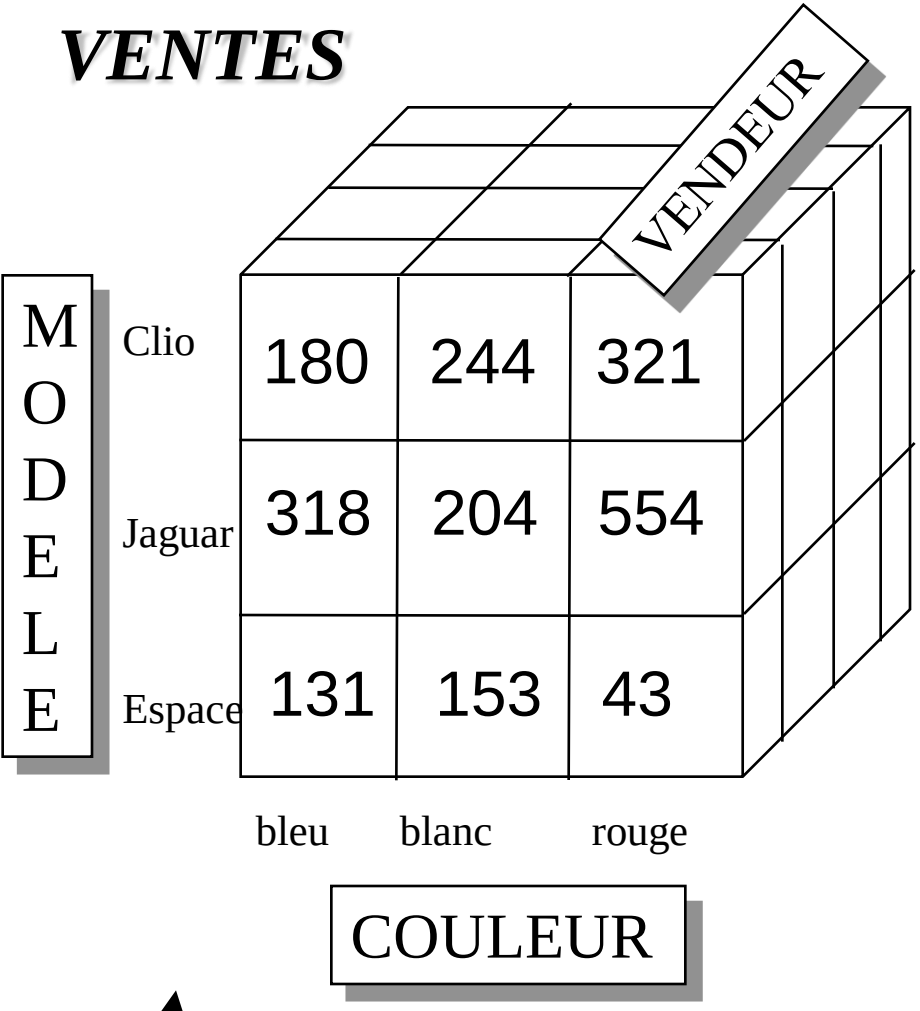
•Ici, pour retrouver une valeur dans une cellule, il faut faire 4+3+3 = 10 recherches seulement

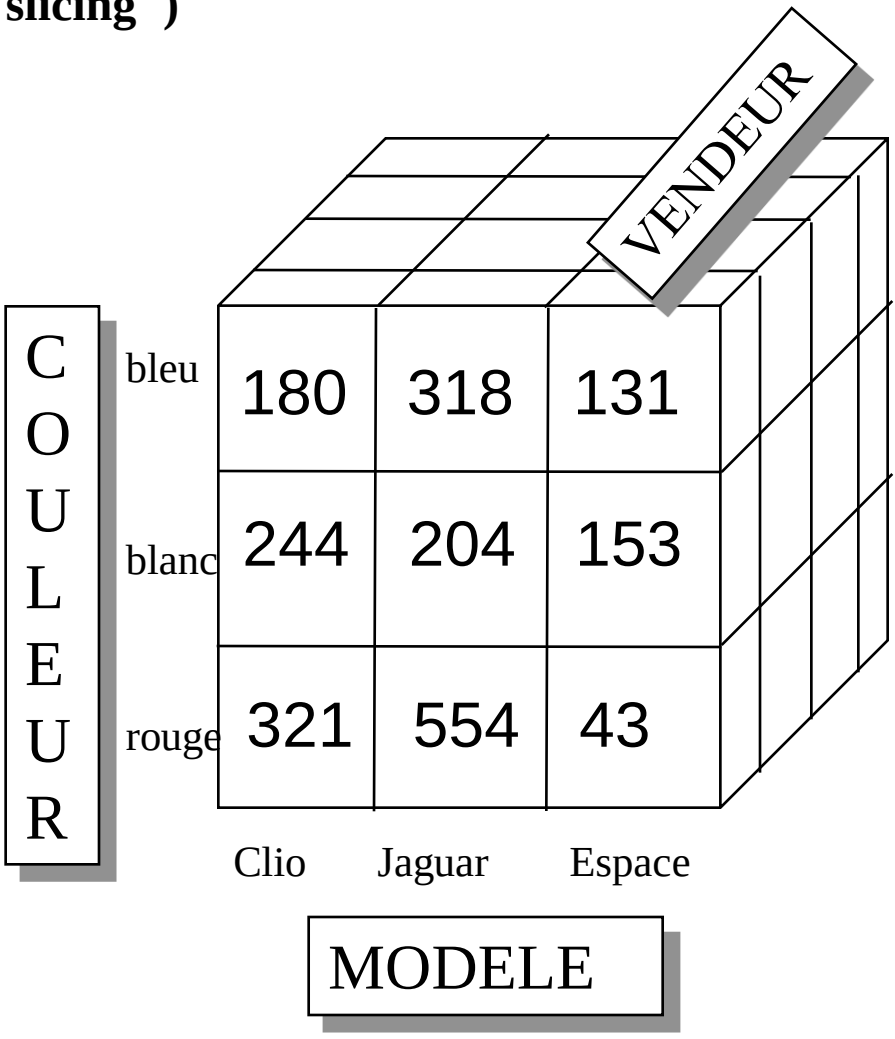
Modele	**Couleur**	**Vendeur**	**Ventes**	*nuplet*
Clio	Bleu	Toto	12	*1*
Clio	Rouge	Toto	23	*2*
Clio	Blanc	Toto	22	*3*
Clio	Bleu	Titi	34	*4*
Clio	Rouge	Titi	45	*5*
Clio	Blanc	Titi	48	*6*
Espace	Bleu	Tutu	55	*7*
Espace	Blanc	Tutu	66	*8*
...	...	...	...	...
Espace	Bleu	Tutu	55	*35*
Espace	Blanc	Tutu	66	*36*



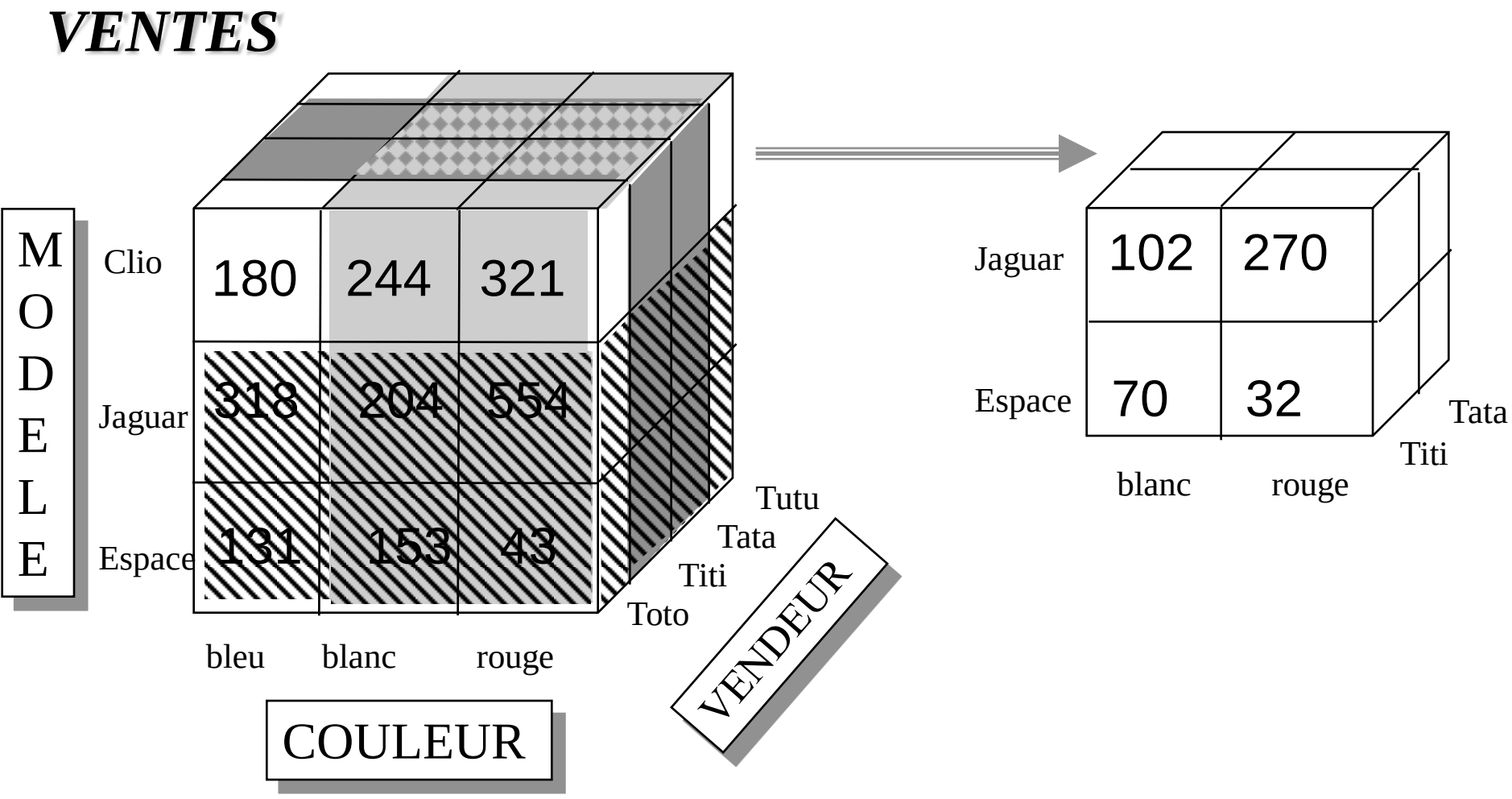
Opérateurs : **rotation ("slicing")**

VENTES



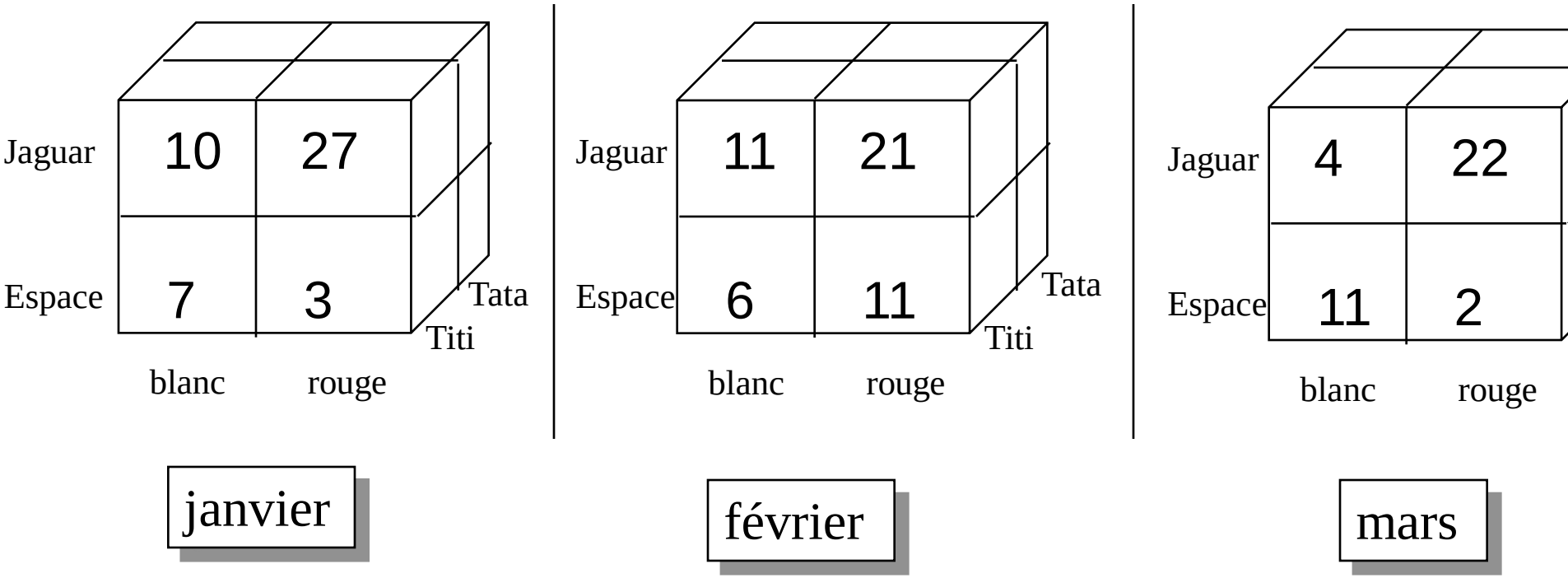


Aucun réarrangement des données. 6 vues possibles directement



Recherche des cellules rapide. Recalcul plus rapide sur les faces

VENTES



Possibilité d'ajouter une dimension à tout moment

Hiérarchies de dimensions

•Les dimensions peuvent être vues à différents niveaux de granularité, qui correspondent à différents niveau de consolidation des données.

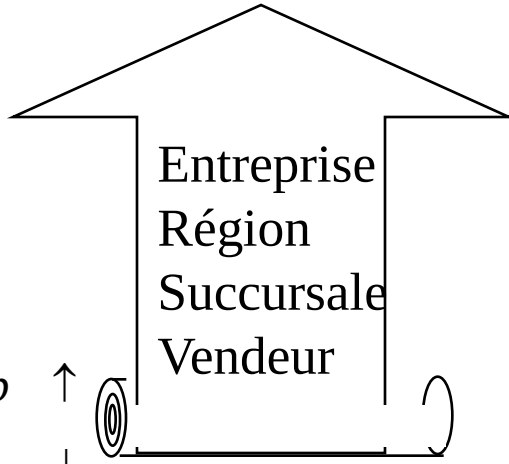
•Exemples :

- •Vendeur → Succursale → Région → Entreprise
- •Jour → Mois → Trimestre → Année → Décennie

•Ces "chemins de consolidation de données" correspondent aux hiérarchies naturelles de l'entreprise.

•Il peut en exister plusieurs pour la même dimension

- •Jour → Semaine → Année → Décennie



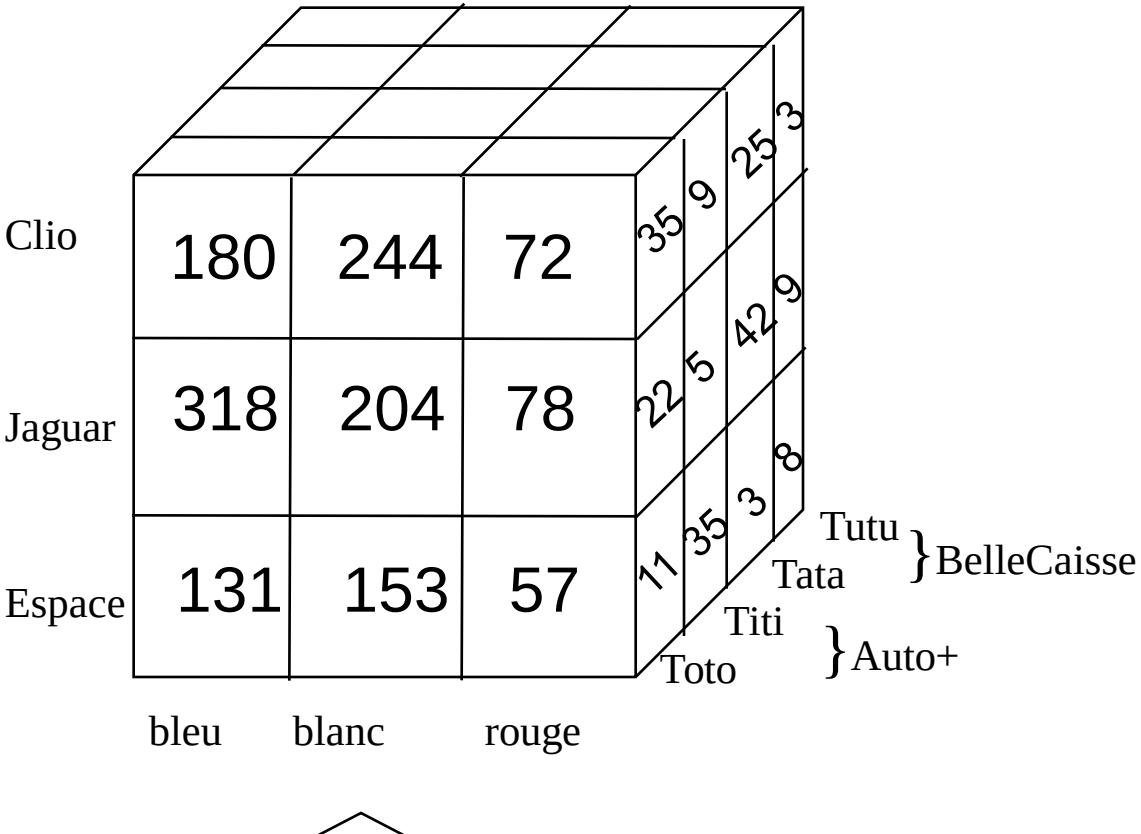
•Faire passer d'une vue à une vue moins détaillée : opérateur *roll-up* ↑

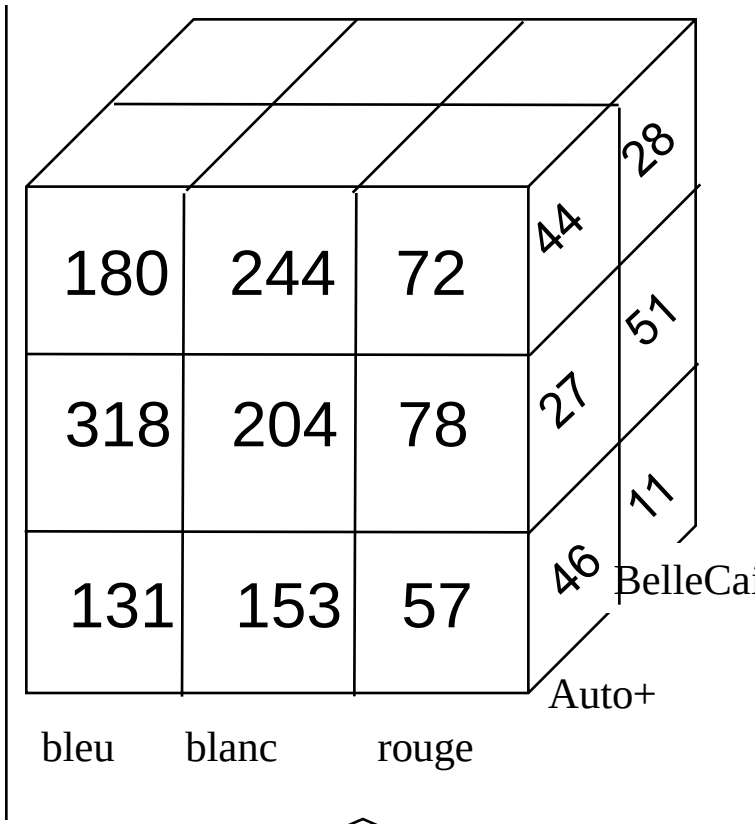
• " " plus " " *drill-down* ↓

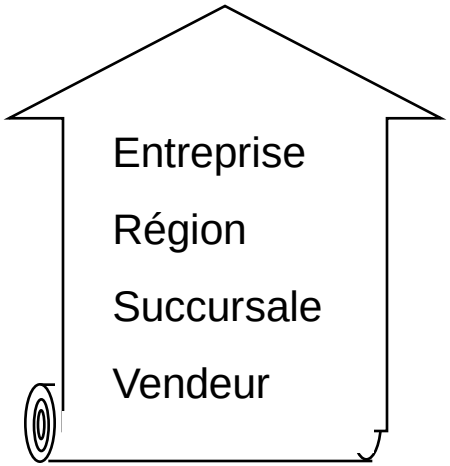
•Ces opérateurs doivent être quasi-instantanés

On-Line Analytical Processing (OLAP) & BD multi-dimensionnelles

Opérateurs : **Roll-up** et **Drill-down**

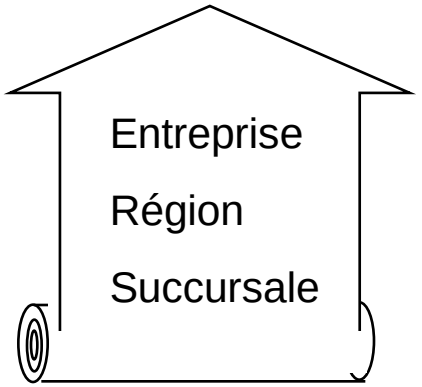






Roll-up

Drill-down



Requêtes sur une BD multidimensionnelle

- Les BD multidimensionnelles étant plus "naturelles", leur interrogation l'est aussi.
- Ex : "ventes par modèle et par succursale"
  - sur le cube : *print total.(ventes keep modèle, succursale)*

Modèle \ Succursale	Auto+	BelleCaisse
Clio	44	28
Jaguar	27	51
Espace	46	11

  - sur les tables relationnelles :

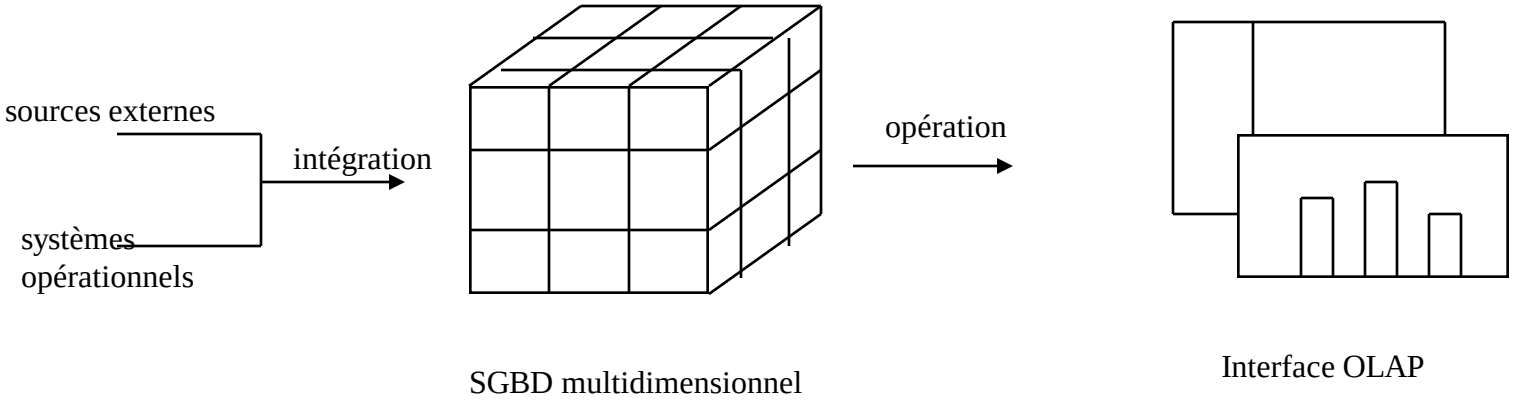
```
Select modèle, succursale, sum(ventes)
from VentesVoitures, Vendeurs
where VentesVoitures.vendeur = Vendeurs.vendeur
Group by modèle, succursale Order by modèle, succursale ;
```

Modele	**Succursale**	**Sum(ventes**
Clio	Auto+	44
Clio	BelleCaisse	28
Jaguar	Auto+	27
Jaguar	BelleCaisse	51
Espace	Auto+	46
Espace	BelleCaisse	11

Représentation des données

Les systèmes spécialisés : MOLAP (OLAP multidimensionnel)

Les données multidimensionnelles sont stockées dans un SGBD dont les structures sont optimisées pour le stockage et le traitement des données.
Accès rapide en lecture/écriture pour de grandes quantités de données.
Les données sont séparées en plusieurs cubes denses de petite taille, pour traiter la dispersion des données (seules un petit nombre de cellules possibles ont une valeur).

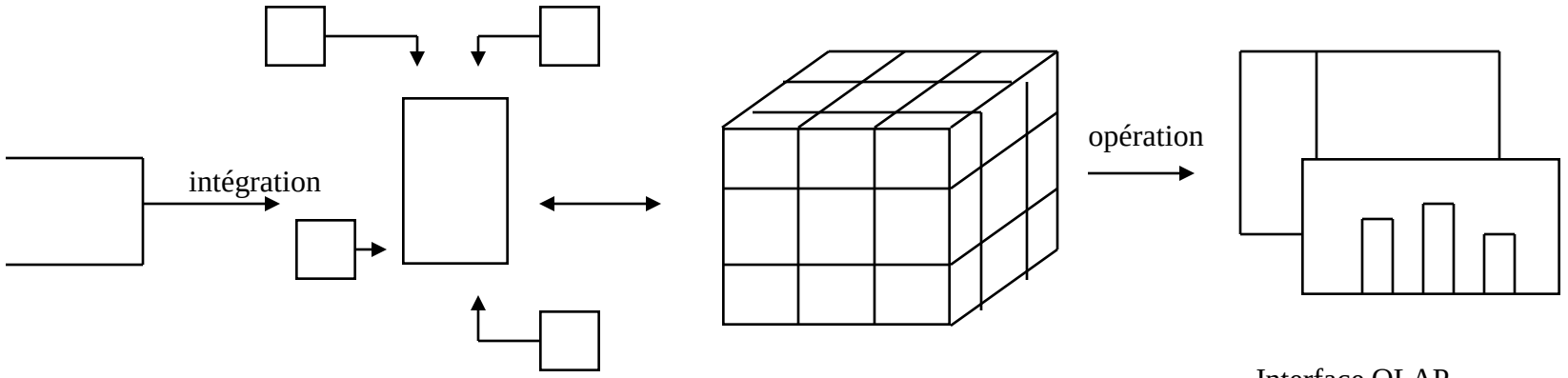


- *Arbor Essbase, IRI Express, Pilot (Pilot Software), ...*

Représentation des données

Les sytèmes relationnels : ROLAP

Les données multidimensionnelles sont stockées dans un SGBD relationnel. Elles sont organisées en schémas en forme d'étoiles ou de flocon. Accès en mode lecture.



- *Redbrick, Microstrategy, MetaCube (Informix)...*

Les sytèmes hybrides : HOLAP

Les données multidimensionnelles sont stockées soit dans un SGBD relationnel, soit dans un SGBD multidimensionnel, afin d'éviter les problèmes des systèmes MOLAP et ROLAP.

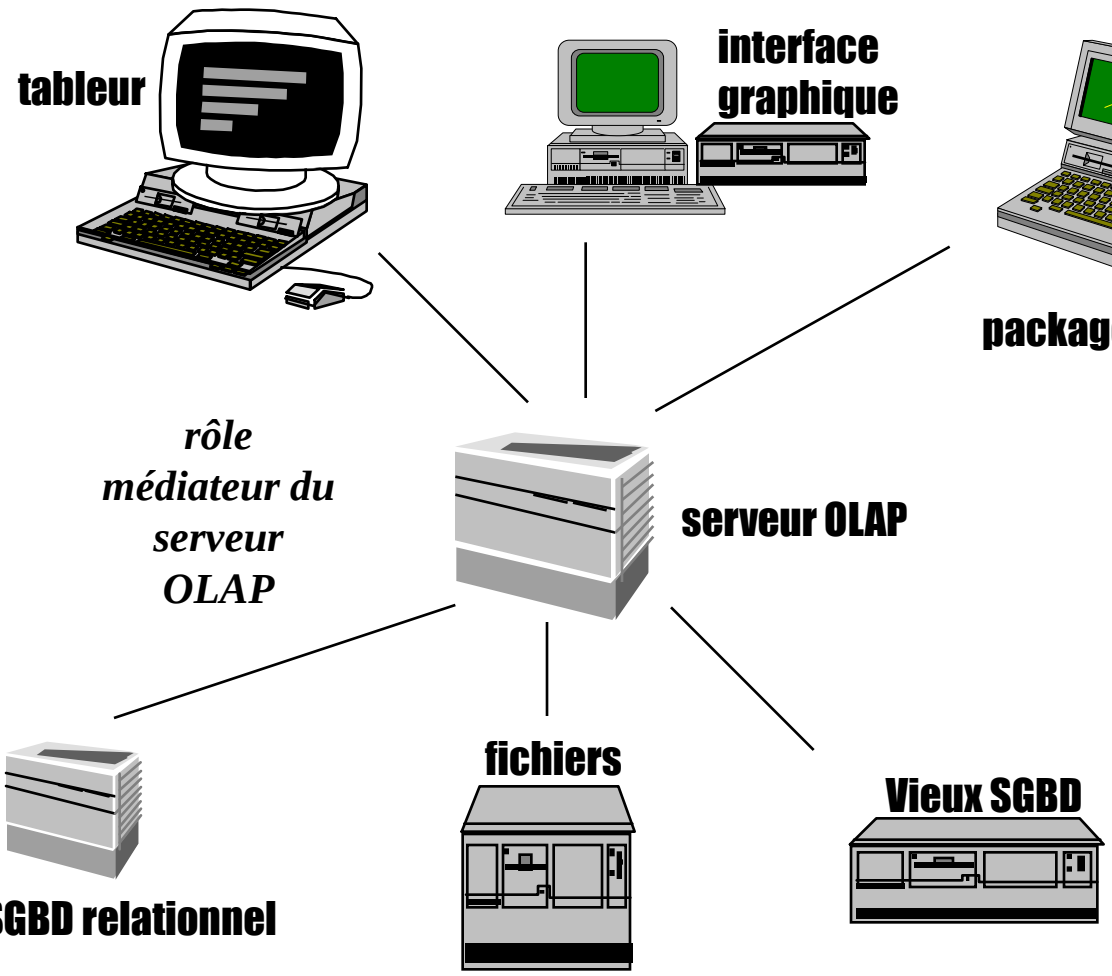
Bilan

Les systèmes MOLAP ont de bons temps de réponse, et peuvent effectuer des calculs complexes, mais ne peuvent pas traiter de grandes quantités de données.

Les systèmes ROLAP peuvent stocker de grandes quantités de données, mais ne peuvent effectuer des calculs complexes, et sont plus lents.

On-Line Analytical Processing (OLAP) & BD multi-dimensionnelles

Architectures possibles (1)



•***approche dédiée*** : le serveur OLAP stocke la BD multidimensionnelle dans des structures non relationnelles. Agrégations, roll-ups précalculés.

•***approche supportée*** : le serveur OLAP traduit les opérations sur le cube en des opérations relationnelles. Les données sont stockées sur le SGBD relat., avec des vues matérialisée et des index supplémentaires.

Architectures possibles (2)

•***Dans les 2 approches* :** les efforts de recherche portent à la fois sur le *calcul du cube* à partir de tables relationnelles (alimentation du serveur), et sur le *calcul d'opérations complexes à partir du cube* (principalement ré-agrégation de données agrégées)

•***Calcul du cube*** : par exemple, on veut que l'agrégation sur 3 dimensions soit la somme.

Equivaut a 2^3 = 8 "group-by" de SQL (toutes les combinaisons de dimensions)
2 types d'optimisation :

- trier les données pour calculer plusieurs agrégats à partir du même tri
- enchaîner les calculs de manière à utiliser certains résultats intermédiaires à la volée (pipe-line)

•***calcul d'opérations complexes*** :

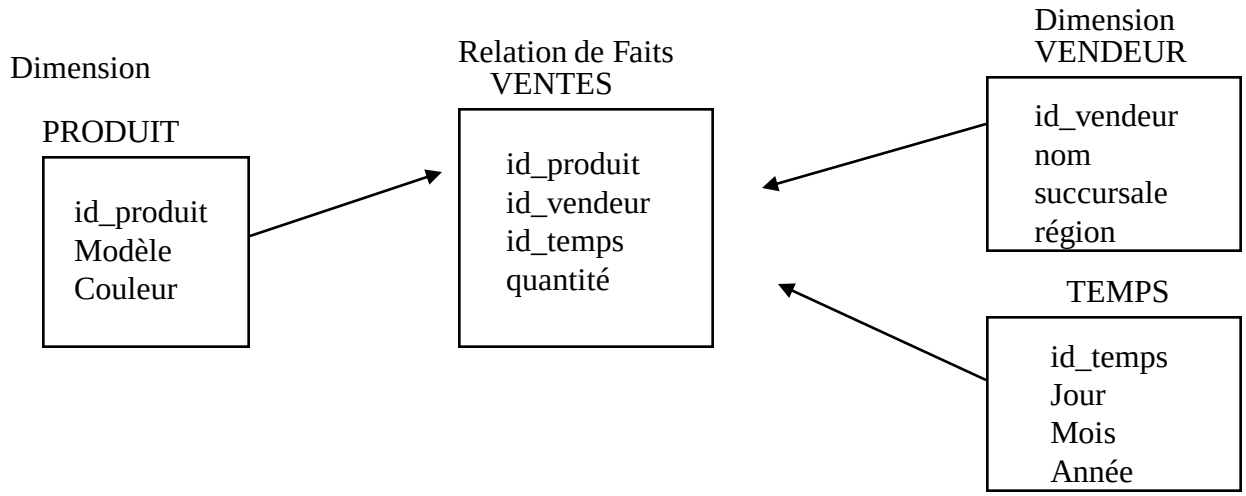
- précalculer certaines informations auxiliaires (pas plus que la taille du cube)
- mettre à jour ces informations incrémentalement en batch
- utilisation de techniques de codages couvrants...

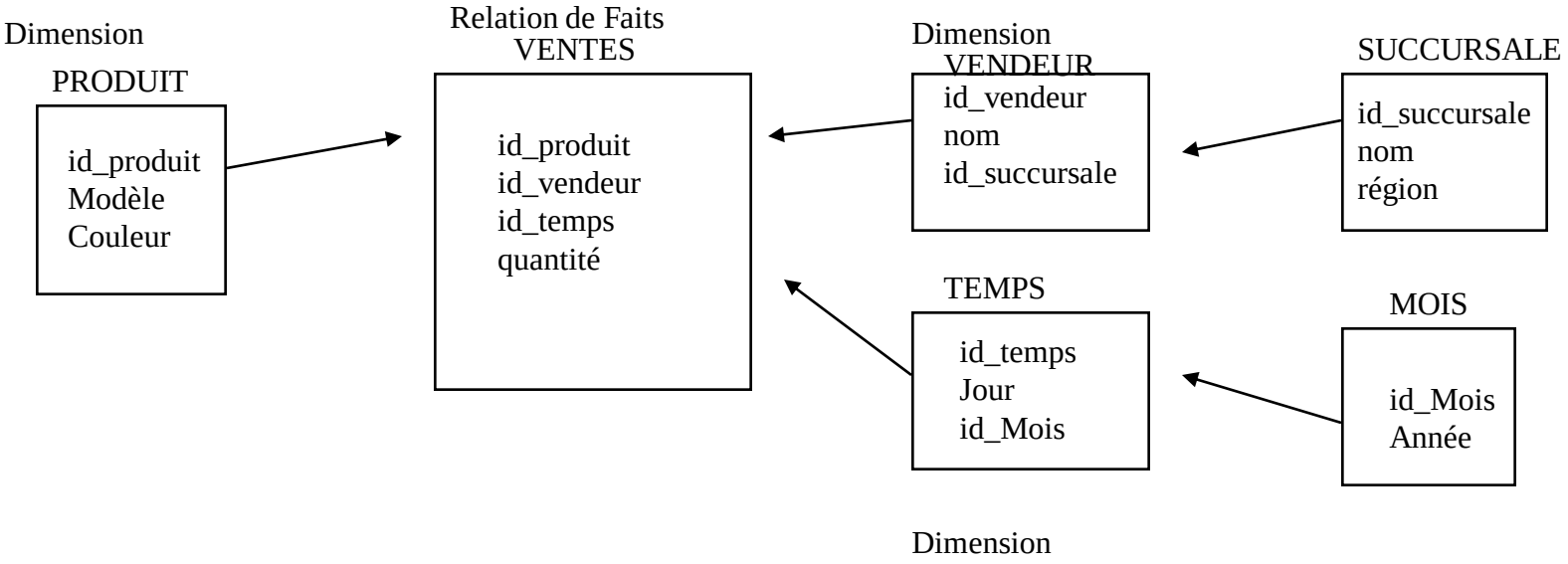
Schémas

- Schéma en étoile :
  - 1 table de faits centrale et plusieurs tables de dimensions dénormalisées
  - Les mesures sont stockées dans la table de faits
  - Il existe une table de dimension pour chaque dimension avec tous les niveaux d'agrégation
- Schéma en flocon
  - Version normalisée du schéma en étoile
  - Traitement explicite des hiérarchies de dimension (chaque niveau est représenté dans une table différente)
  - Plus facile à maintenir, plus lent lors de l'interrogation.

On-Line Analytical Processing (OLAP) & BD multi-dimensionnelles

Schémas en étoile et en flocon





Modèles de cube

Il existe plusieurs modèles de cube, et un ensemble d'opérateurs de base permettant de construire d'autres opérateurs plus complexes. Certaines fonctionnalités n'existent pas dans les produits commerciaux:

- Traitement symétrique des dimensions et mesures.

Les sélections et les agrégrations doivent pouvoir être faites sur les dimensions comme sur les mesures. Ex: *quantité totale des articles vendus par tranche de prix (0-499, 500-999, etc.).* La mesure (vente) est aussi l'attribut de group-by.

- Hiérarchies multiples dans les dimensions.

Jour --> Mois --> Trimestre --> Année --> Décennie

Jour --> Semaine --> Année --> Décennie

- Agrégations ad-hoc

On peut avoir besoin d'autres agrégats que ceux prévus initialement (ex: somme totale des ventes d'un produit, ou moyenne par succursale).

- Modèle de requêtes

Actuellement, pas de composition de requêtes ==> résultats intermédiaires.

Une approche "bases de données" (R. Agrawal, Ashish Gupta, S. Saragawi - IBM San-Jose)

- En marge du système Quest (Data mining). Intégration prévue.
- Modèle de données "rigoureux", proche algèbre relationnelle, moins d'opérateurs possibles.
- Les **opérateurs** permettent de manipuler un ou plusieurs hypercubes
  - Push et pull permettent de transformer une dimension en mesure et vice-versa
  - destroy dimension
  - restrict : sélectionne le long d'une dimension
  - join : associe deux hypercubes
  - merge : agrège un cube selon une ou plusieurs dimension- fonction d'agreg.
- Permettent d'exprimer roll-up (merge) et drill-down (join avec le cube contenant les details).
- Peuvent être traduits en un SQL un peu amélioré (extension proposée)
- Permettent d'exprimer les requêtes "Olap"

"pour chaque produit, sa part de marché dans sa catégorie par rapport à celle de 1994"

Restrict dimension Temps au dernier mois.
Merge la dimension fournisseurs en un seul point avec la fonction somme.
Push la dimension produit pour avoir <ventes, produits> comme mesure.
Merge la dimension produit par catégories avec la fonction maximum
Pull produit à la place de catégorie -> nouveau cube C1....etc.

Cube multidimensionnel de Li et Wang

Coexistence de cubes multidimensionnels et de relations.

Les hiérarchies sont représentées par des relations de groupement (ex: les villes sont regroupées dans des régions).

L'algèbre comprend les opérateurs relationnels et des opérateurs multidimensionnels permettant de manipuler les relations, de générer et réarranger des cubes à partir de relations et de cubes.

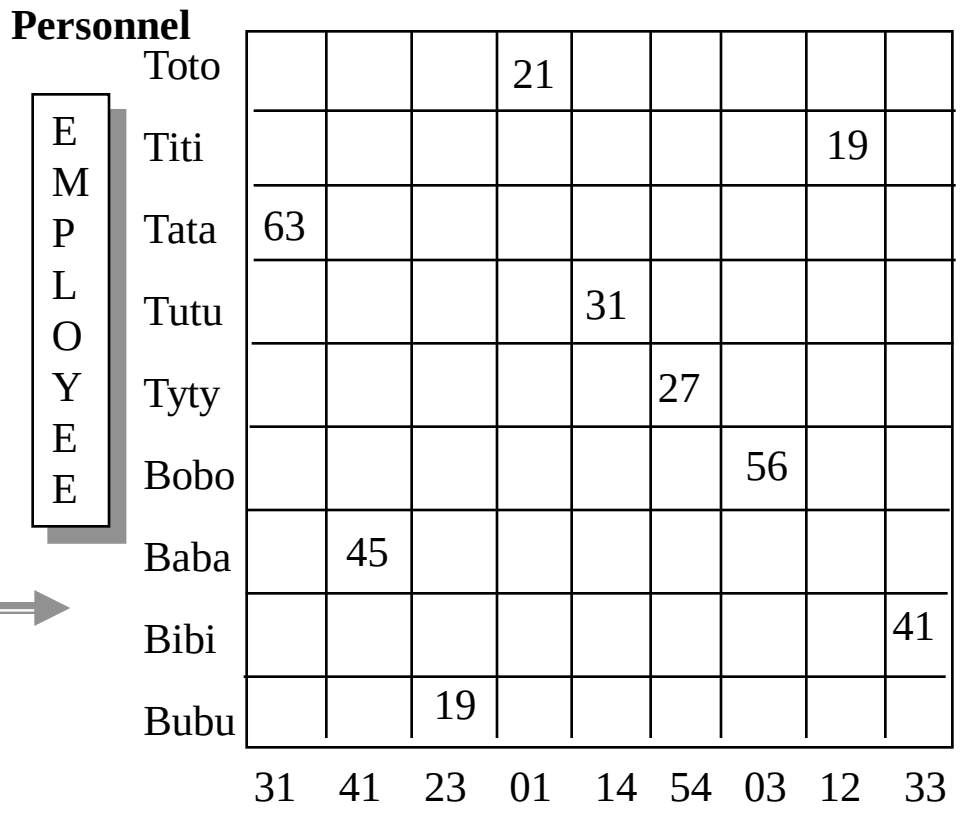
+ grande flexibilité dans la manipulation des hiérarchies et pour la construction de cubes à partir de relations.
- les modalités et les mesures ne sont pas traitées symétriquement.

BD multidimensionnelles ≠ panacée.

- La technologie multidimensionnelle n'est pas toujours adaptée et ne doit pas être utilisée "`à toutes les sauces"
- Exemple :

Personnel

Employé	#employé	Age
Toto	01	21
Titi	12	19
Tata	31	63
Tutu	14	31
Tyty	54	27
Bobo	03	56
Bibi	41	45
Baba	33	41
Bubu	23	19



Les données sur le personnel **ne sont pas multidimensionnelles** : pas de relation entre les éléments des différents nuplets

Les précurseurs des BD multidimensionnelles : BD temporelles, BD spatiales, BD statistiques

- **Bases de données temporelles** : stockage et interrogation des valeurs des différents item de la base selon une ou plusieurs dimensions temporelles (les plus utiles : temps de validité, temps de transaction)

  seul le temps est traité comme une dimension

- **Bases de données spatiales** : représentation des item selon leur forme et position dans l'espace.

  opérateurs développés géométriques, loin des opérateurs OLAP

  possibilité d'utiliser les travaux sur l'indexation de telles bases

- **Bases de données statistiques** : bcp de points communs, mais sans construire un nouveau modèle. Extension du modèle relationnel pour supporter les tables de résumé et les *traitements* statistiques.

  Traitement non-uniforme des dimensions et des mesures.

  Récupérer les techniques d'implémentation (notamment vues agrégées)

Règles d'or de Codd

•1993 : E.F. Codd formule 12 *règles d'or* (à la demande de Arbor soft. !!)

•1995 : 18 règles en 4 groupes :

Basiques :

1. vue multidimensionnelle
2. manipulation directe
3. médiation (accessibilité)
4. intégration d'approche dédiée **et** d'approche supportée.
5. support de tous les modèles d'analyse des entreprises
   (seuls les plus simples sont habituellement supportés)
6. Client/serveur
7. Transparence (ne pas avoir à savoir d'où viennent les données,
   même si elles viennent de sources externes).
8. Multi-utilisateurs (lecture seule ?)

Règles d'or de Codd

Caractéristiques spéciales

9. Traitement des données dénormalisées
10. Stockage des résultats à part (ne pas interférer avec les mise à jour des transactions de production)
11. Représentation des valeurs manquantes
12. Traitement des valeurs manquantes.

Présentation des rapports:

13. Flexibilité (ajout de dimension...)
14. Performances non dégradées si nb. dim. ou taille BD augmente.
15. Ajustement de la représentation physique

Contrôle des dimensions :

16. Généricité : traitement équivalent de chaque dimension.
17. Nombre et profondeur illimités (actuellement, max. = 10 et 6)
18. Calculs à travers n'importe quelles dimensions.

Test FASMI
(Fast Analysis of Shared Multidimensional Information)

- Voulu plus simple, réaliste et général que les règles de Codd
- **Fast :** 1 seconde pour les analyses de bases, - de 5 secondes pour la plupart, très peu au dessus de 20 secondes (au-delà de 30 secondes, CTRL-ALT-DEL!). Même si on a maintenant en 5 minutes ce qui durait des heures, l'utilisateur perd le fil de son raisonnement...
- **Analysis** : doit servir pour n'importe quelle analyse logique ou statistique assez facilement (sans programmer), que ce soit par des outils internes ou des appels a des outils externes (ex. tableur).
- **Shared** : les bonnes propriétés habituelles "multi-utilisateurs" des SGBD : concurrence d'accès en écriture, confidentialité, sécurité.
- **Multidimensional** (view)
- **Information :** toute l'information nécessaire doit pouvoir être produite (rapport de 1 à 1000 entre le produit le moins puissant et le plus puissant). Benchmarking...

Olap Council (http://www.olapcouncil.org)

- Fondé pour le développement et la standardisation de l'OLAP
- Regroupe la plupart des vendeurs d'OLAP (mais pas tous!)
- OLAP MDAPI : Interface standard que doivent fournir les serveurs OLAP, de manière à ce que différents outils d'analyse puissent se développer par rapport à ces spécifications.
- Interopérabilité : le même outil d'analyse pourra alors utiliser simultanément des données provenant de différents serveurs OLAP.
- MDAPI V.5 disponible sur WWW pour commentaires. Uniquement pour des analyses de consultation. Prochaines versions incluront les mises à jour rétroactives.
- Même stratégie pour le benchmarking : package complet APB-1

Problèmes ouverts en OLAP

- Langage et optimisation de requêtes
- Stockage et indexation des BD multidimensionnelles
- Utilisation pour la fouille de données
- Mises à jour directe sur le cube. Rétro-action sur les données brutes.

Quelques références

R. Agrawal, A. Gupta, S. Saragawi: Modeling multidimensional databases, *IEEE Conference on Data Engineering, 1997.*

L. Cabibbo , R. Torlone: A logical approach to multidimensional databases, *EDBT International Conference, 1998.*

M. Gyssens, LVS. Lakshamanan, I. Subramanian : tables as a paradigm for querying and restructuring, *ACM PODS International Conference, 1996.*

M. Gyssens, LVS. Lakshamanan: A foundation for multidimensional databases, *VLDB International Conference, 1997.*

MS. Hacid, P. Marcel, C. Rigotti : A rule-based language for ordered Multidimensional databases, *5Th Workshop on Deductive Databases and Logic Programming, 1997.*

C. Li, XS. Wang : A data model for supporting on-line analytical processing, *5th International Conference on Information and Knowledge Representation, 1996.*

Systèmes légués

- **Système légué** :
  gros système, *critique*, sur environnement ancien. Souvent *peu documenté*. *Interactions* entre les différents modules *peu claires*. *Très cher* à maintenir.
- Il faut l'intégrer (migration) au système actuel (Entrepôt) = architecture cible.
- Contraintes : migration sur place, garder opérationnel, corriger et améliorer pour anticiper, le moins de changements possibles (diminuer le risque), flexible sur les évolutions futures, utiliser les technologies modernes.
- Approche classique : tout réécrire dans l'architecture cible
  - promesses à tenir dans des conditions changeantes
  - problème de transfert de très gros fichiers (plusieurs jours) dans système critique
  - gros projet, retard mal vus, risque d'abandon
- Approche incrémentale :
  - *isoler* des sous-systèmes a migrer
  - établir des *passerelles* pour que les modules déjà migrés puissent communiquer avec les modules encore dans le système légué (traducteur de requêtes et de données).
  - coordonner les mises à jour pour garder la cohérence.