EPISEN

Ing2 2021-2022

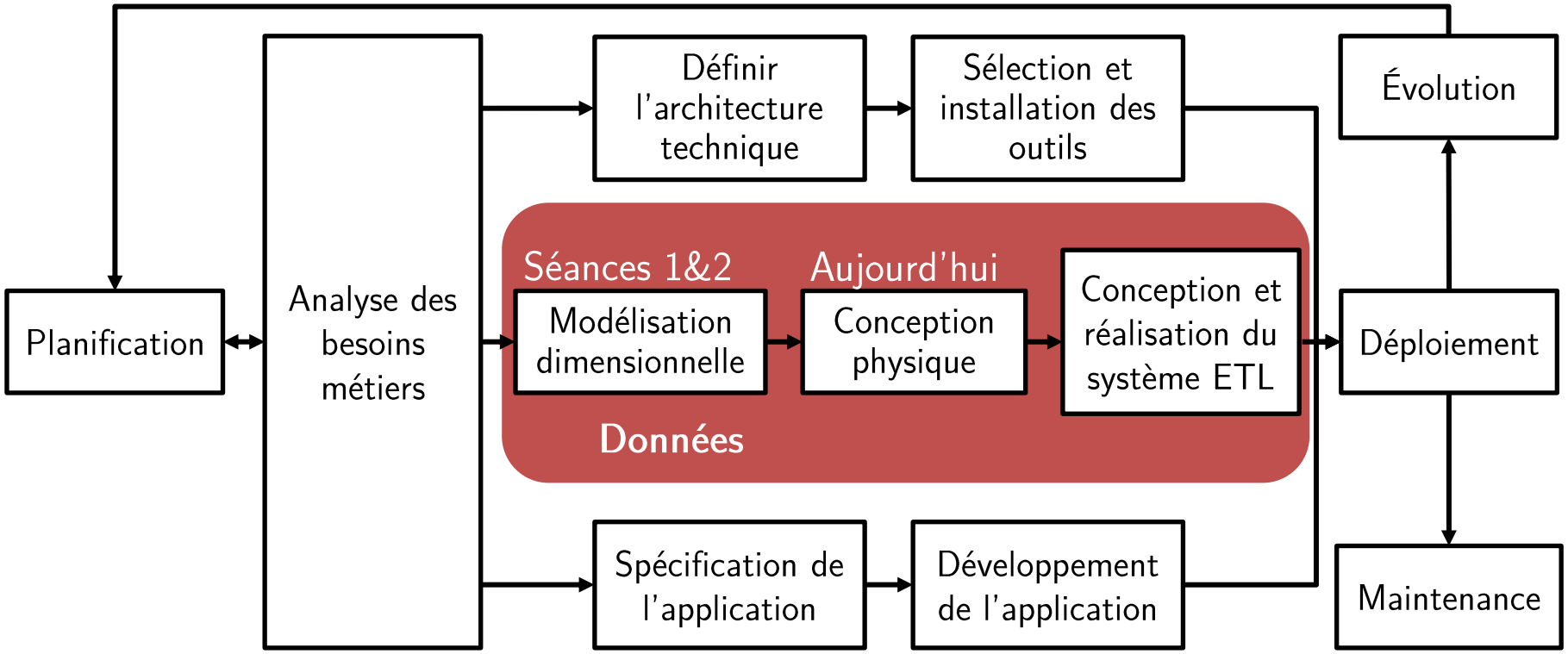
Module BDM

BASES DE DONNÉES MULTIDIMENSIONNELLES ET NOMADES

CONCEPTION PHYSIQUE DES ENTREPÔTS DE DONNÉES

Cycle de vie d'un projet I.D.

Le cycle de vie définit les étapes de conception, développement et déploiement d'un Entrepôt de Données.



(Source Ralph Kimball)

Problématique

- Quels sont les critères à considérer lors de la conception physique?
- Quels sont les choix à faire lors de la conception physique et quel est l'impact de ces choix?

Choix de conception

- Taille de la BD :
  - Tables de faits, index, agrégations, etc.
- Stockage des données:
  - Allocation de mémoire, stockage horizontal/vertical, etc.
- Stratégie d'indexation:
  - Arbre-B, cluster, bitmap, jointure en étoile, etc.
- Stratégie d'agrégation:
  - Dimensions et niveaux d'agrégation, approche MOLAP ou ROLAP, etc.

Taille de la BD

- Quels sont les facteurs à considérer lors de l'estimation de la taille de l'entrepôt de données?
- Comment estime-t-on la taille requise?

Estimation de la taille de la BD

- Taille des lignes des tables de faits et de dimension:
  - Considérer le nombre et le type de chaque colonne.
- Nombre de lignes de ces tables:
  - Estimer le nombre de lignes ajoutées à chaque chargement (ex: chaque mois ou années).
- Stockage des indexes:
  - Environ égal à la taille des données.
- Espace temporaire:
  - Construction des index, opérations de tri (ORDER BY, DISTINCT) et de groupement (GROUP BY), etc.
- Taille des agrégations:
  - Tables d'agrégations, cubes OLAP, etc.
- Autres considérations:
  - Tables de métadonnées;
  - Zone de préparation de données ETL (staging area);
  - Journal des opérations du système (system log);
  - etc

Stockage des données

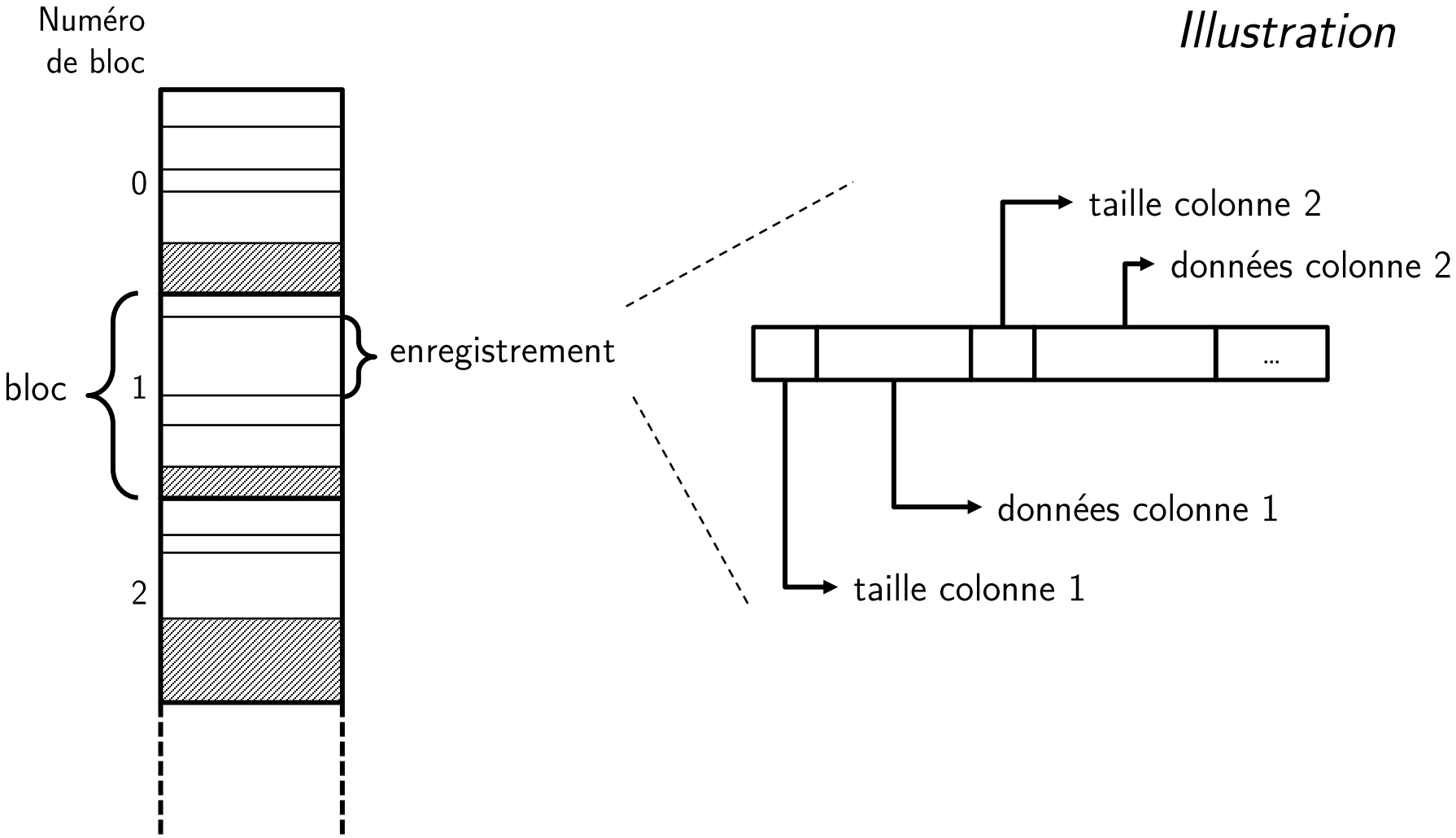
- Comment les données sont-elles stockées physiquement dans l'entrepôt?
- Comment le stockage influence-t-il la performance du système?

Stockage des données

- Fichiers et blocs:
  - Une BD est composée de plusieurs fichiers de données, renfermant des blocs de données;
  - Un bloc referme un certain nombre d'enregistrements (lignes) d'une table de la BD;
  - Un bloc constitue l'unité minimale de transfert entre la mémoire centrale et le disque dur de la BD.
- Principe d'optimisation:
  - L'opération la plus coûteuse au niveau du serveur BD est le transfert de données entre la mémoire centrale et le disque dur;
  - Il faut donc minimiser le nombre de blocs lus et écrits dans la BD.

Stockage des données

Illustration



Stockage des données

Paramètres de stockage (Oracle)

```
CREATE TABLE TableExemple
(
        ...
)
INITRANS 1
MAXTRANS 5
PCTFREE 10
PCTUSED 40
TABLESPACE user_data
STORAGE(
        INITIAL 20480
        NEXT 20480
        MINEXTENTS 1
        MAXEXTENTS 10
        PCTINCREASE 10)
```

Paramètre	Description
INITRANS /MAXTRANS	Espace initial et maximal du bloc pour l'information sur les transactions impliquant le bloc (ex: récupération)
PCTFREE	Pourcentage d'espace dans le bloc réservé pour les UPDATE
PCTUSED	Pourcentage du bloc devant être libre pour le retourner dans la FREE LIST
TABLESPACE	L'espace mémoire dans lequel la table est créée
INITIAL / NEXT	Taille en octets des premier et deuxième extensions mémoire (EXTENT) de la table
MINEXTENTS /MAXEXTENTS	Nombre minimal et maximal d'extensions de mémoire permises pour la table
PCTINCREASE	Facteur de croissance de la taille des extensions de mémoire (après la deuxième)

Ordre de stockage

- L'ordre dans lequel les données sont stockées a-t-il un impact sur la performance?
- L'ordre de stockage est-il différent dans un entrepôt de données que dans une BD transactionnelle?

Stockage vertical / BD en colonnes

- La plupart des RDBMS transactionnels stockent les données horizontalement:
  - Les lignes d'une table sont stockées séquentiellement sur disque:

ligne 1			ligne 2			ligne 3		
col 1	col 2	col 3	col 1	col 2	col 3	col 1	col 2	col 3

  - Facilite les requêtes retournant une ou plusieurs lignes.
- Dans les entrepôts de données, les requêtes portent souvent sur les colonnes (ex: SUM, AVG, MIN, MAX, etc.)
  - Il peut être plus efficace (500x plus rapide dans certains cas) de stocker les données par colonnes:

colonne 1			colonne 2			colonne 3		
ligne 1	ligne 2	ligne 3	ligne 1	ligne 2	ligne 3	ligne 1	ligne 2	ligne 3

Stratégie d'indexation

- Comment peut-on accélérer les opérations de sélection et de jointure dans l'entrepôt de données?
- Quels sont les différents types d'index et leurs avantages respectifs?

Stratégie d'indexation

- Index:
  - Structure de données résidant dans la BD;
  - Permet d'améliorer la performance d'opérations comme la sélection et la jointure.
- Types d'index:
  - Index arbre-B (B-tree);
  - Index groupant (cluster);
  - Index bitmap;
  - Index de jointure en étoile (star-join)

Stratégie d'indexation

Considérations:

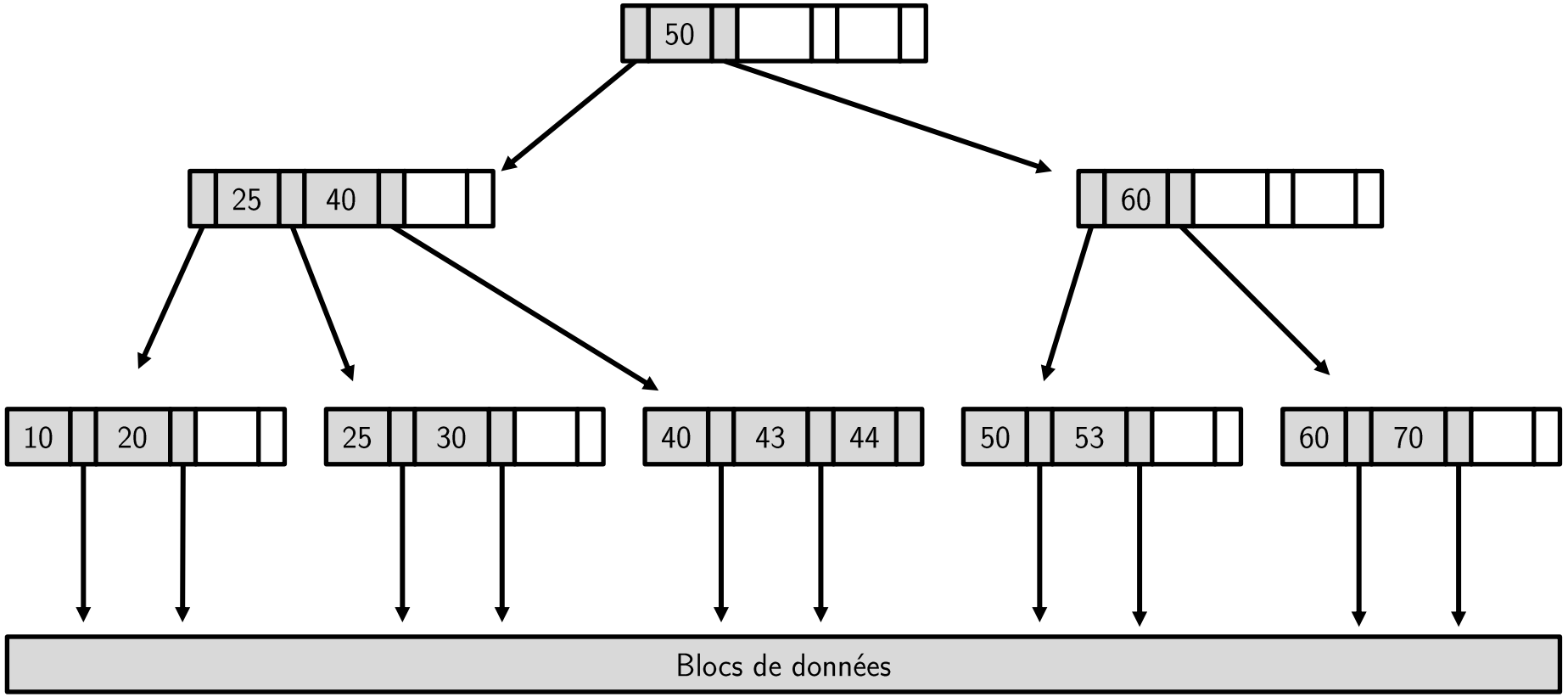
- Stockage:
  - Peut prendre autant d'espace que les données à indexer.
- Mise à jour suite à une insertion/suppression:
  - Moins significatif dans les entrepôts de données.

Index arbre-B

- Index hiérarchique sous la forme d'un arbre balancé;
- Réorganisation dynamique à la suite d'une insertion, suppression ou mise à jour de la clé;
- Sélection d'une ligne en $\mathcal{O}(\log N)$ dans le pire des cas, où N est le nombre de lignes;
- Propriétés:
  - Chaque bloc doit être rempli à moitié au minimum;
  - Les clés sont triées dans un bloc;
  - Toutes les clés contenues dans le sous-arbre d'une clé sont strictement inférieures à cette clé.

Index arbre-B

Exemple

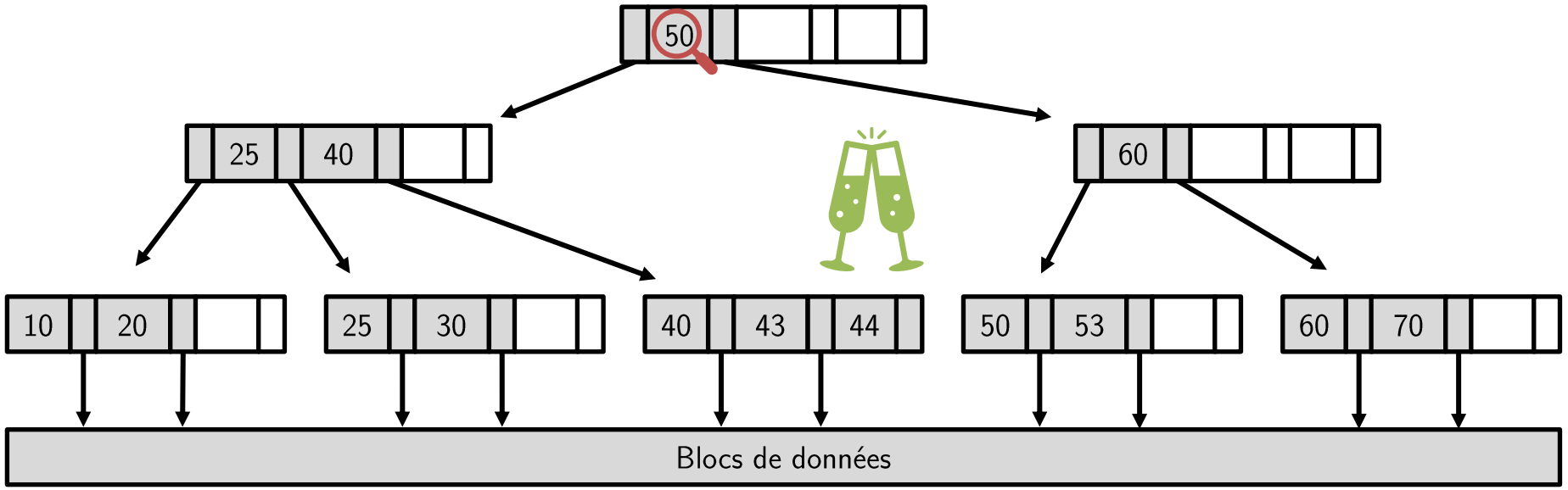


Index arbre-B

Recherche avec clé C dans un arbre-B:

- Bloc interne:
  - On cherche le plus petit i tel que $C < C_i$
  - On suit le pointeur à la gauche de C_i
- Feuille:
  - On parcourt le bloc jusqu'à ce qu'on trouve la clé ou qu'on dépasse sa valeur

Exemple: Recherche de $C = 43$



Index arbre-B

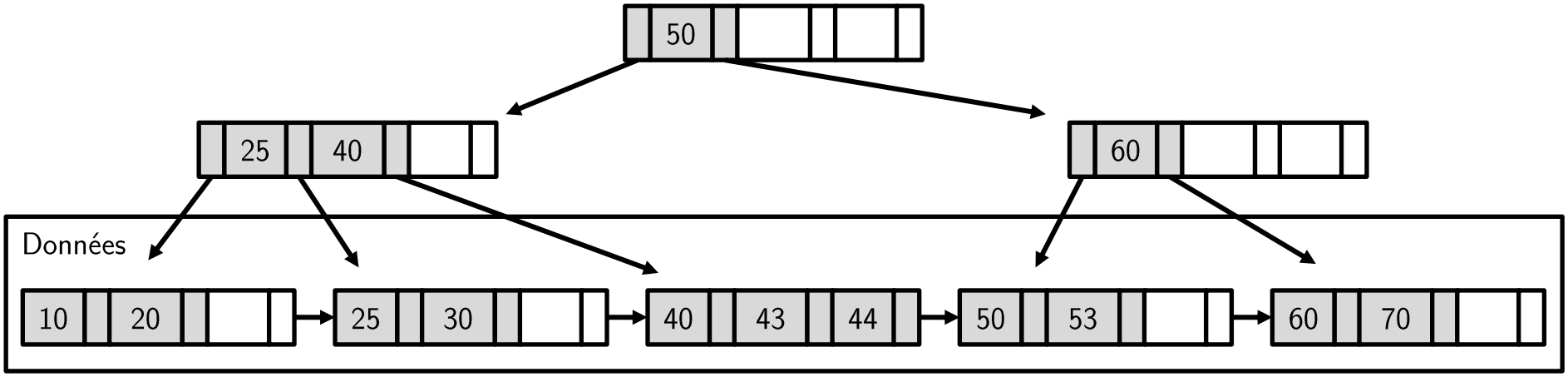
- Exemple de création (Oracle):

```
CREATE INDEX nomIndex ON TableExemple(col1, col2, ...)
[paramètres de stockage]
```

- Consignes d'utilisation:
  - Ne pas créer d'index si plus de 15% de la table peut être retournée (sinon le balayage est plus efficace);
  - Ne pas créer d'index sur les colonnes dont la valeur est souvent nulle;
  - Créer des index sur les clé uniques ou sur les colonnes utilisées pour filtrer;
  - Clés composées: mettre d'abord les colonnes les plus utilisées car l'index sert également pour les préfixes.

Organisation par index (TOI)

- Les données sont stockées physiquement dans l'index



- Les lignes sont physiquement ordonnées selon la clé
- Accélère les sélections et les jointures

Organisation par index (TOI)

- Exemple (Oracle) :

```
CREATE TABLE ExempleTable
(
        idClePrimaire INTEGER NOT NULL,
        col1 DATE NOT NULL,
        col2 VARCHAR NOT NULL,
        ...
        PRIMARY KEY (idClePrimaire)
)
ORGANIZATION INDEX
```

- La clé primaire est automatiquement utilisée comme clé de l'index;
- Un seul TOI par table, mais peut être combiné avec plusieurs indexes arbre-B

Index groupant (cluster)

- Regroupe les lignes d'une ou plusieurs tables selon une clé (index cluster) ou une valeur dérivée de la clé (hash cluster);
- Un seul possible par table.
- Cluster sur plusieurs tables:
  - Accélère la jointure en regroupant physiquement les lignes de plusieurs tables selon la clé de jointure (pré-jointure);
  - Ralentit le balayage d'une table (full table scan) car les lignes de la table sont plus distancées.

Index groupant (cluster)

Cluster sur deux tables

idClient	nom	noTéléphone
1001	'Jean Bon'	'111-111-1111'
1002	'Charles Latan'	'222-222-2222'
1003	'Paul Lisse'	'333-333-3333'

idCompte	idClient	Montant
1	1001	1000$
2	1001	100$
3	1002	500$
4	1002	2000$
5	1003	5000$

Clé de l'index: idClient

Bloc 0

1003	'Paul Lisse'	'333-333-3333'
5	1003	5000$

Bloc 1

1001	'Jean Bon'	'111-111-1111'
1	1001	1000$
2	1001	100$

Bloc 2

1002	'Charles Latan'	'222-222-2222'
3	1002	500$
4	1002	2000$

Index groupant (cluster)

- Exemple (Oracle) :

```
CREATE CLUSTER MonCluster(idClient INTEGER)
[paramètres de stockage]

CREATE TABLE Client(
        idClient INTEGER NOT NULL PRIMARY KEY,
        nom VARCHAR2(50) NOT NULL,
        noTéléphone VARCHAR2(12) NOT NULL)
CLUSTER MonCluster(idClient)

CREATE TABLE Compte(
        idCompte INTEGER NOT NULL PRIMARY KEY,
        idClient INTEGER NOT NULL,
        montant NUMBER(7,2) NOT NULL,
        FOREIGN KEY (idClient) REFERENCES Client)
CLUSTER MonCluster(idClient)
```

- Note: peu employé dans les entrepôts de données

Index bitmap

- Crée une colonne (bit) pour chaque valeur possible des colonnes indexées
  - ➢ Ex: colonne sexe {homme, femme} se traduit par deux bits;
- Sert à accélérer les sélections ayant des contraintes d'égalités de valeur sur plusieurs colonnes.
- Consignes d'utilisation:
  - Lorsque le domaine de la colonne est réduit et varie peu;
  - Pour les sélections pouvant retourner un pourcentage élevé du nombre de lignes;
  - Exemples:
    - ✓ Colonne `sexe` est un bon choix;
    - ✗ Colonne `noTéléphone` est un mauvais choix.

Index bitmap

idClient	nom	étatCivil	sexe	revenuNet
1001	'Jean Bon'	'marié'	'H'	55320
1002	'Lucie Fer'	'N/A'	'F'	38145
1003	'Sylvie Agra'	'célibataire'	'F'	152546
1004	'Charles Latan'	'N/A'	'H'	25100

```
CREATE BITMAP INDEX bitmapEtatCivil ON Client(etatCivil)
CREATE BITMAP INDEX bitmapSexe ON Client(sexe
```

BitMap étatCivil

marié	célibataire	N/A
1	0	0
0	0	1
0	1	0
0	0	1

BitMap sexe

H	F
1	0
0	1
0	1
1	0

Index de jointure en étoile

- Pré-calcule les lignes des tables de dimension pouvant être jointes avec la table de faits;
- Permet d'utiliser l'information des tables de dimensions sans devoir joindre ces tables à la table de faits.
- Bitmap join index (Oracle):
  - Calcule l'index bitmap sur la table résultant de la jointure des tables de dimensions avec la table de faits;
  - Les colonnes des tables de dimension employée dans l'index bitmap doivent avoir un domaine restreint;
  - Peut accélérer jusqu'à 10 fois la jointure (benchmarks Oracle).

Index de jointure en étoile

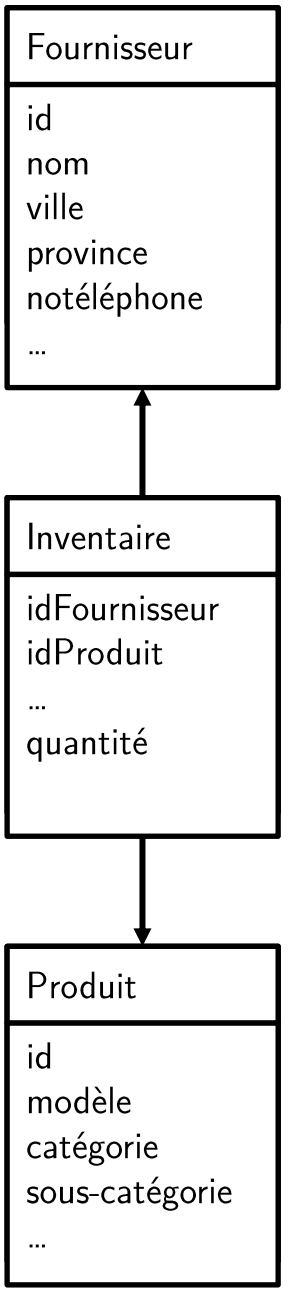
Exemple bitmap join index:

- Requête à optimiser:

```
SELECT SUM(Inventaire.quantité)
FROM Inventaire, Produit, Fournisseur
WHERE Inventaire.idProduit = Produit.id AND
      Inventaire.idFournisseur = Fournisseur.id AND
      Produit.catégorie='moteur_mazda' AND
      Fournisseur.province='QC'
```

- Index à créer

```
CREATE BITMAP INDEX indexJointure ON
Inventaire(Fournisseur.province, Produit.catégorie)
FROM Inventaire, Fournisseur, Produit
WHERE Inventaire.idProduit = Produit.id AND
      Inventaire.idFournisseur = Fournisseur.id
```

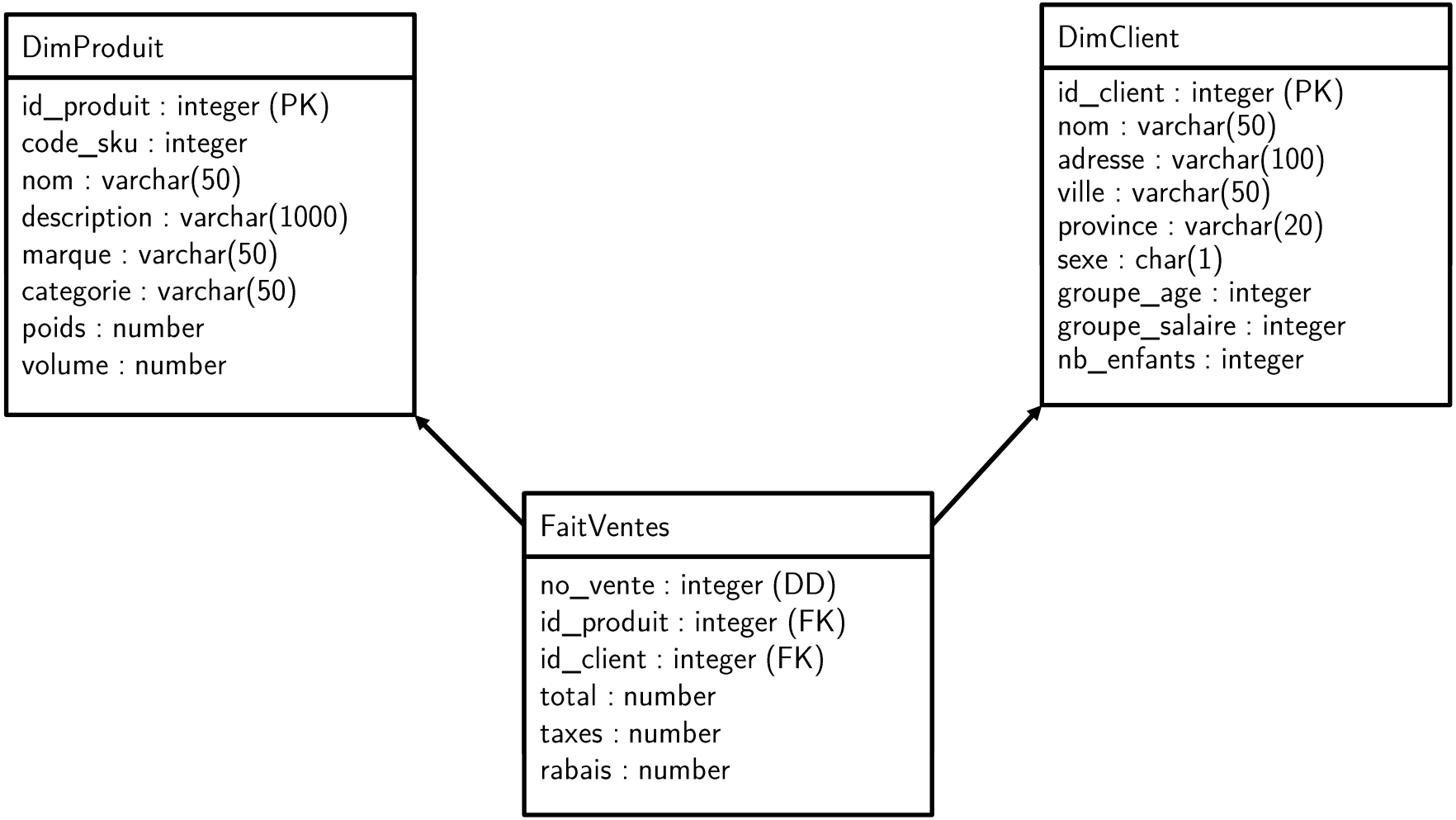


Recommandations

- Tables de dimension:
  - Mettre un index arbre-B sur les colonnes employées pour filtrer et ayant un grand domaine;
  - Mettre un index bitmap sur les colonnes dont le domaine est restreint et invariant, souvent utilisées pour filtrer les sélections (ex: sexe).
- Tables de faits:
  - Utiliser une TOI si les lignes sont souvent accédées par intervalle de la clé primaire (ex: $ total pour transactions 10000 à 20000);
  - Mettre un index arbre-B sur les clés étrangères;
  - Créer des index de jointure en étoile sur les combinaisons d'attributs de dimension les plus souvent utilisées.

Exercice

- Proposez des indexes pour le schéma suivant:



Pré-agrégation

- Qu'est-ce que la pré-agrégation des tables?
- À quoi sert cette stratégie?
- Comment peut-on implémenter concrètement la pré-agrégation?

Stratégie d'agrégation

- Exemple d'agrégation:
  - Contexte:
    - 300 magasins et 40,000 produits;
    - Environ 500 produits par marque;
    - Environ 1 vente à chaque semaine, pour chaque produit, dans chaque magasin.
  - Requête sur 1 produit, 1 magasin, 1 semaine:
    - Agrégation de 1 ligne de la table de faits.
  - Requête sur 1 produit, tous les magasins, 1 semaine:
    - Agrégation de 300 lignes de la table de faits.
  - Requête sur 1 marque, tous les magasins, 1 année:
    - Agrégation de 7,800,000 lignes de la tables de faits.

Tables de faits agrégées

- Accélèrent les requêtes analytiques en pré-calculant l'agrégation de faits à différents niveaux des hiérarchies dimensionnelles;
- Dupliquent l'information contenue dans la table de faits atomique (niveau le plus granulaire).
- Exemple (suite):
  - Table de faits agrégés où chaque ligne donne le total des ventes durant une semaine, pour une marque de produits dans un certain magasin;
  - Requête sur 1 marque, tous les magasins, 1 année:
    - Agrégation de 15,600 lignes (au lieu de 7 millions).

Niveaux d'agrégation

Magasin		Produit		Temps	
magasin	x	produit		**jour**	x
ville		**marque**	x	semaine	
région		catégorie		mois	
province		département		trimestre	
tous		tous		année	

Agrégation sur une seule dimension (one-way agregate)

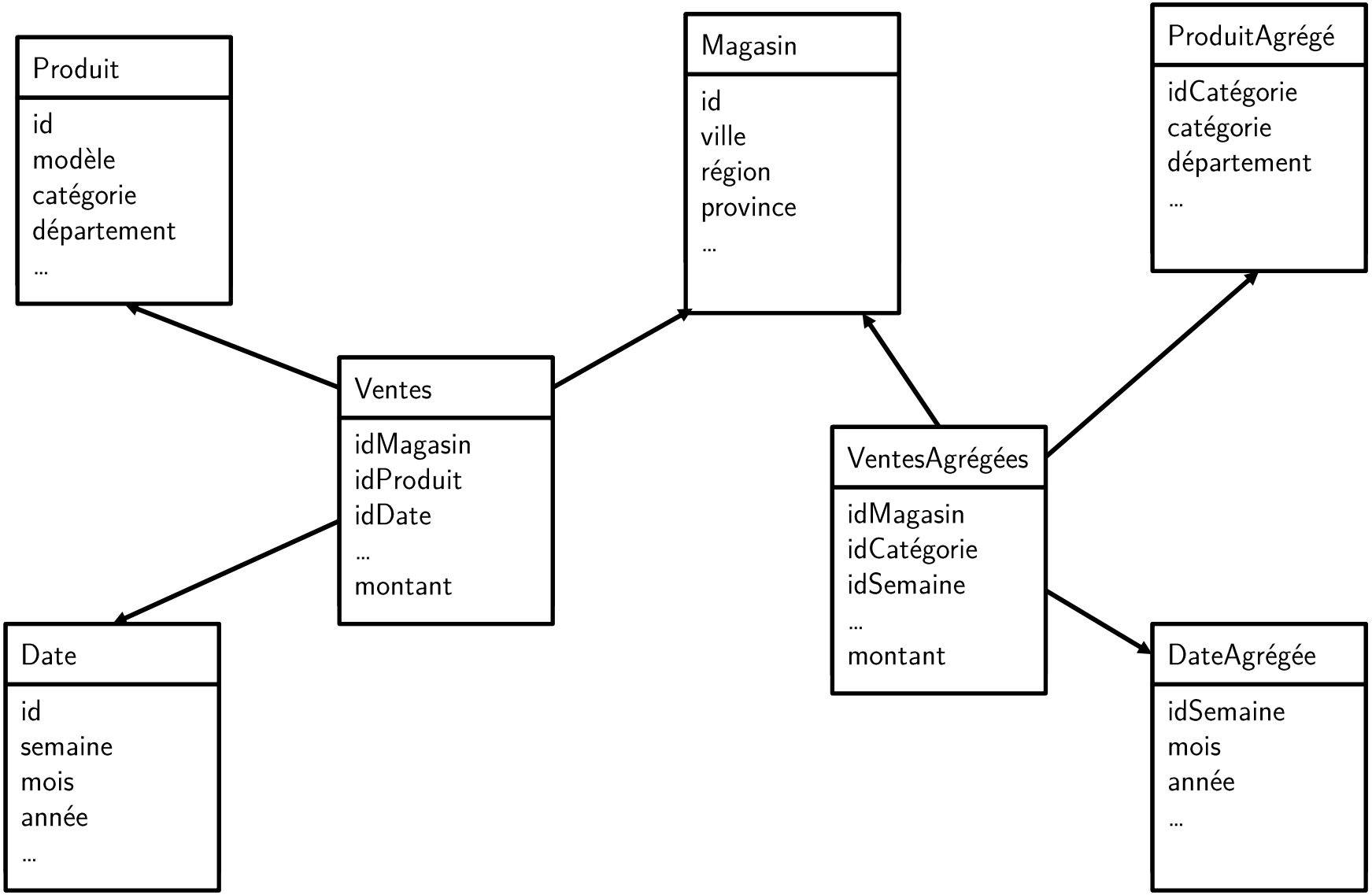
Magasin		Produit		Temps	
magasin		produit		**jour**	x
ville	x	**marque**	x	semaine	
région		catégorie		mois	
province		département		trimestre	
tous		tous		année	

Agrégation sur deux dimensions (two-way agregate)

Magasin		Produit		Temps	
magasin		produit		jour	
ville	x	**marque**	x	**semaine**	x
région		catégorie		mois	
province		département		trimestre	
tous		tous		année	

Agrégation sur trois dimensions (three-way agregate)

Stratégie d'agrégation



Choix d'agrégation

Comment décide-t-on les pré-agrégations à faire (dimensions et grain) ?

- Tenir compte du type et de la fréquence des requêtes faites à l'entrepôt (profilage de requêtes);
- Choisir les dimensions les plus souvent utilisées;
- Choisir un niveau de hiérarchie offrant un bon compromis entre l'utilité et le gain en performance
  - Ex: l'agrégation des magasins Ikea au niveau "ville" n'est pas utile car il y a peu de magasins dans une ville (max. 3);
- Chaque nouvelle ligne doit agréger au moins 10 lignes de la table de faits atomiques.

Approches d'agrégations

- Relational OLAP (ROLAP):
  - Utilise les BD relationnelles standards (ex: vues matérialisées).
- Multidimensional OLAP (MOLAP):
  - BD multidimensionnelles (cubes de données);
  - Technologie différente des BD relationnelles (ex: tableaux compressés).
- Approche hybride (HOLAP):
  - Combine les avantages du MOLAP et ROLAP;
  - Implémentée dans la plupart des produits commerciaux.

Agrégation par vues matérialisées (ROLAP)

- Table physique synchronisée avec les résultats d'une requête;
- Synchronisation temps-réel, en lot ou sur demande;
- Permet les indexes, le partitionnement, contrôle d'accès, etc.
- Hiérarchie d'agrégations possible en créant une nouvelle vue à partir d'autres vues.
- Exemple (Oracle):

```
CREATE MATERIALIZED VIEW TransactionAgrégée
REFRESH FORCE ENABLE QUERY REWITE
AS
SELECT idMagasin,
       P.catégorie AS catégorie,
       D.semaine AS semaine,
       SUM(T.montant) AS montantAgrégé
FROM Transaction T, Produit P, Date D
WHERE T.idProduit = P.id AND T.idDate = D.id
GROUP BY idMagasin, P.catégorie, D.semaine
```

Paramètre	Description
REFRESH FORCE	Synchronisation incrémentale lorsque possible, sinon complète
ENABLE QUERY REWRITE	Permet de réécrire la requête si cela améliore la performance

Agrégation par vues matérialisées (ROLAP)

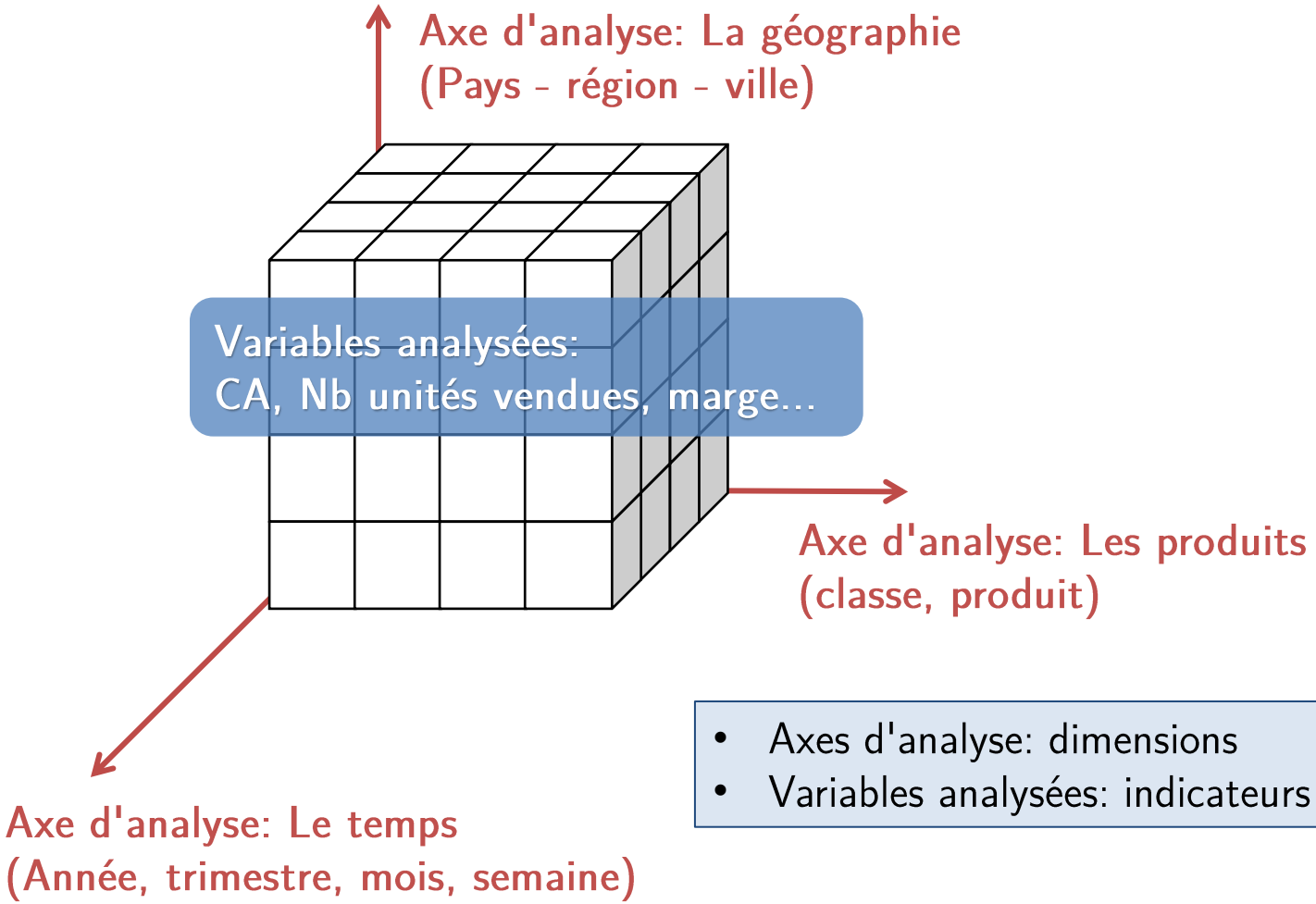
- Avantages:
  - Utilise la même technologie que l'entrepôt de données (BD relationnelle standard);
  - Langage établi de requêtes (SQL).
- Inconvénients:
  - Gestion des agrégations plus complexe que MOLAP;
  - Moins bonne performance que MOLAP, mais la technologie s'améliore.

BD multi-dimensionnelles (MOLAP)

- Représentent les données sous la forme d'un tableau multidimensionnel (cube) où:
  - Les coordonnées d'une case correspondent à une combinaison de valeurs des dimensions du cube;
  - Le contenu d'une case correspond aux faits (mesures) pour ces valeurs;
- Utilisent des techniques de compression pour gérer le fait que la plupart des cases du cubes sont vides (sparse array compression);
- Pré-calculent certaines agrégations, en se basant sur les hiérarchies dimensionnelles.

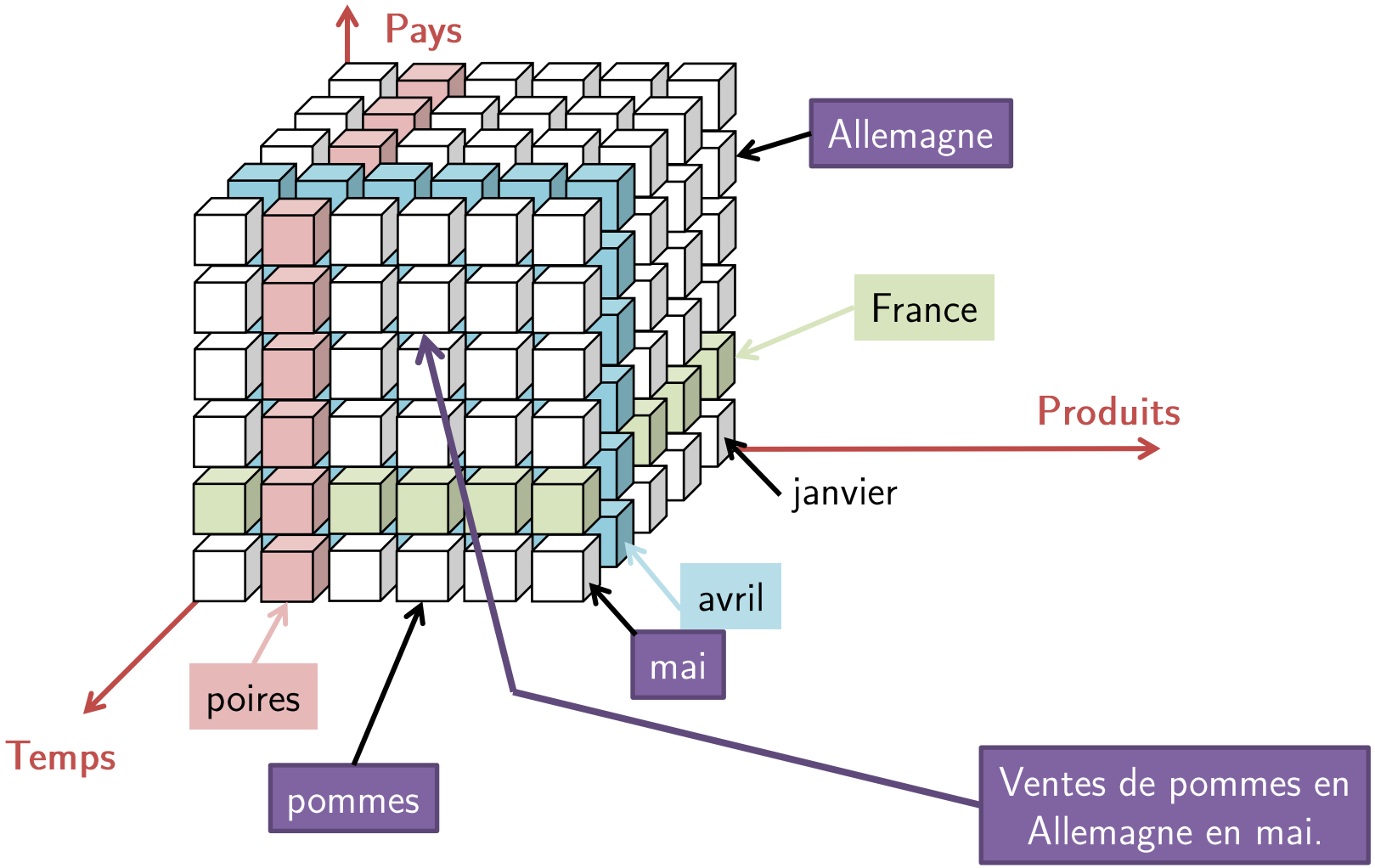
Le cube

Un cube est défini par des dimensions ; au croisement de ces dimensions se trouvent des mesures.



- Axes d'analyse: dimensions
- Variables analysées: indicateurs de mesure

Le cube


Pays
Allemagne
France
Produits
janvier
avril
mai
poires
Temps
pommes
Ventes de pommes en Allemagne en mai.

BD multi-dimensionnelles (MOLAP)

- Avantages:
  - Bonne performance pour les opérations d'analyse multi-dimensionnelle (ex: slicing et dicing);
  - Gestion efficace des agrégations;
- Inconvénients:
  - Demande beaucoup de mémoire, surtout si le nombre de dimensions est important;
  - Les mises à jour peuvent être plus coûteuses que ROLAP.

OLAP & le Data Mining

- Les applications OLAP : utilisées pour analyser les performances de l'entreprise (technologie OLAP)
  - Ventes réalisées en quantité et valeur par point de vente pour chaque collection d'ouvrages?
- Le Data Mining (fouille de données) utilise des algorithmes de reconnaissance de modèles afin de détecter des **comportements particuliers**, des **corrélations** ou des **tendances** dans les données
  - Les informations obtenues servent à des fins de prédiction telles que des prévisions de ventes, segmentation de population d'individus au comportement similaire etc