

# **Les entrepôts de données et l'analyse de données**

# Différences entre OLTP et OLAP

Critère	Online Transaction Processing (OLTP)	Online Analytical Processing (OLAP)
But	Contrôler et exécuter les tâches quotidiennes et fondamentales de l'entreprise	Assister dans la planification, la résolution de problème et la prise de décision
Types de données	Données opérationnelles (transactions)	Données historiques consolidées
Sources de données	BD transactionnelles	Entrepôts de données ou magasins de données
Ce que montre les données	Portrait instantané des processus d'affaires de l'entreprise	Vue multidimensionnelle de plusieurs activités d'affaires de l'entreprise
Insertions et mises-à-jour	Courtes requêtes d'insertion et de mise-à-jour lancées par les usagers finaux	Longs traitements en lot servant à rafraichir les données
Requêtes	Simple requêtes retournant quelques enregistrements (lignes) de la BD	Requêtes complexes impliquant souvent plusieurs tables et faisant l'agrégation de valeurs
Temps de réponses	Instantané	Quelques secondes à 1 minute max.

# Différences entre OLTP et OLAP

Critère	Online Transaction Processing (OLTP)	Online Analytical Processing (OLAP)
Espace requis	Relativement petit si les données historiques sont archivées	Grand due aux données historiques et aux données d'agrégation
Modélisation de la BD	Un grand nombre de tables hautement normalisées	Moins de tables, tables typiquement dénormalisées, schémas en étoile ou flocon
Sauvegarde et récupération	Fait de façon régulière, critique pour l'entreprise	Fait de façon irrégulière, peut récupérer des données OLTP
Reporting	Routinier et très ciblé	Ad hoc, multidimensionnel, à plus large portée
Ressources requises	Simple DBMS relationnel	DBMS spécialisé multi-processeurs et à grande capacité
Nombre d'utilisateurs simultanés	Grand	Petit

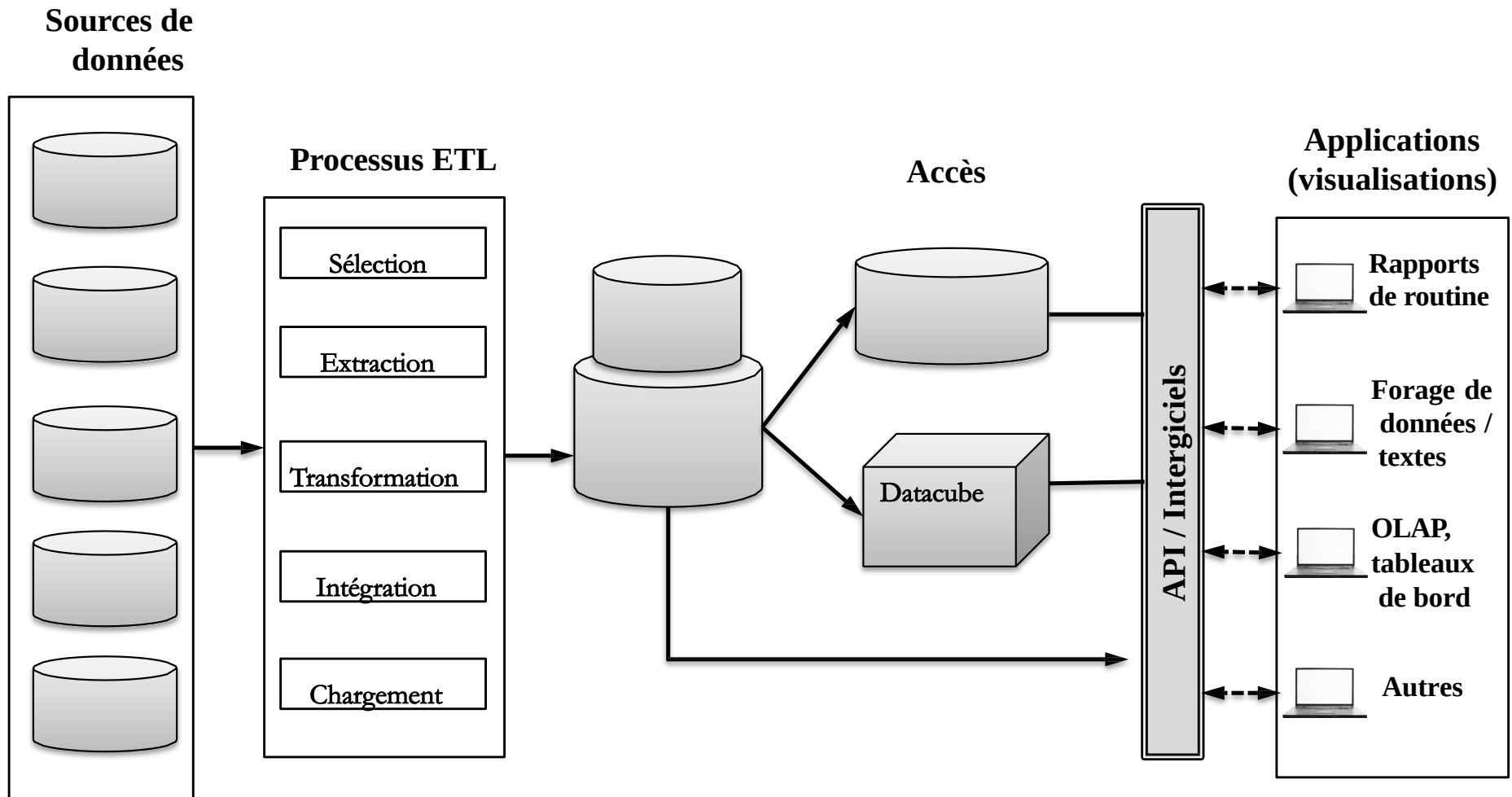
# Entreposage de données (*data warehousing*):

- « La copie **périodique** et **coordonnée** de données provenant de **diverses sources**, internes et externes à l'entreprise, dans un environnement optimisé pour l'**analyse** »
- **Orientés sujet:**
  - Les données sont organisées par sujet (ex: clients, produits, ventes, etc.).
- **Intégrés:**
  - Les données, qui proviennent de diverses sources hétérogènes, sont consolidées et intégrées dans l'entrepôt.
- **Historiques:**
  - Les données ont très souvent une composante temporelle (ex: date et heure d'une transaction).
- **Non-volatiles:**
  - Une fois insérées dans l'entrepôt, les données ne sont jamais modifiées ou effacées; elle sont conservées pour des analyses futures.

# Bénéfices des entrepôts de données

- Permettent de mener des analyses poussées sur différents sujets d'affaires;
- Fournissent une vue consolidée des données de l'entreprise (une seule vérité);
- Procurent de l'information de qualité, plus rapidement;
- Libèrent les ressources (ex: serveurs) dédiées au traitement des transactions des tâches d'analyse;
- Data warehouse Simplifient l'accès aux données. Entrepôt de données

# Solution d'entreposage de données



# Sources de données

- **Enterprise resource planning (ERP):**
  - Gèrent les processus opérationnels d'une entreprise (ex: ressources humaines, finances, distribution, approvisionnement, etc.).
- **Customer relationship management (CRM):**
  - Gèrent les interactions d'une entreprise avec ses clients (ex: marketing, ventes, après-vente, assistance technique, etc.).
- **Systèmes legacy:**
  - Matériels et logiciels obsolètes mais difficilement remplaçables.
- **Point of sale (POS):**
  - Matériels et logiciels utilisés dans les caisses de sorties d'un magasin.
- **Externes:**
  - Ex: données concurrentielles achetées, données démographiques.

# Problèmes des sources de données

- Sources diverses et disparates (ex: BD, fichier texte, etc.);
- Sources sur différentes plateformes et OS;
- Applications *legacy* utilisant des technologies obsolètes;
- Historique de changement non-préservé dans les sources;
- Qualité de données douteuse et changeante dans le temps;
- Structure des systèmes sources changeante dans le temps;
- Incohérence entre les différentes sources;
- Données dans un format difficilement interprétable ou ambigu.

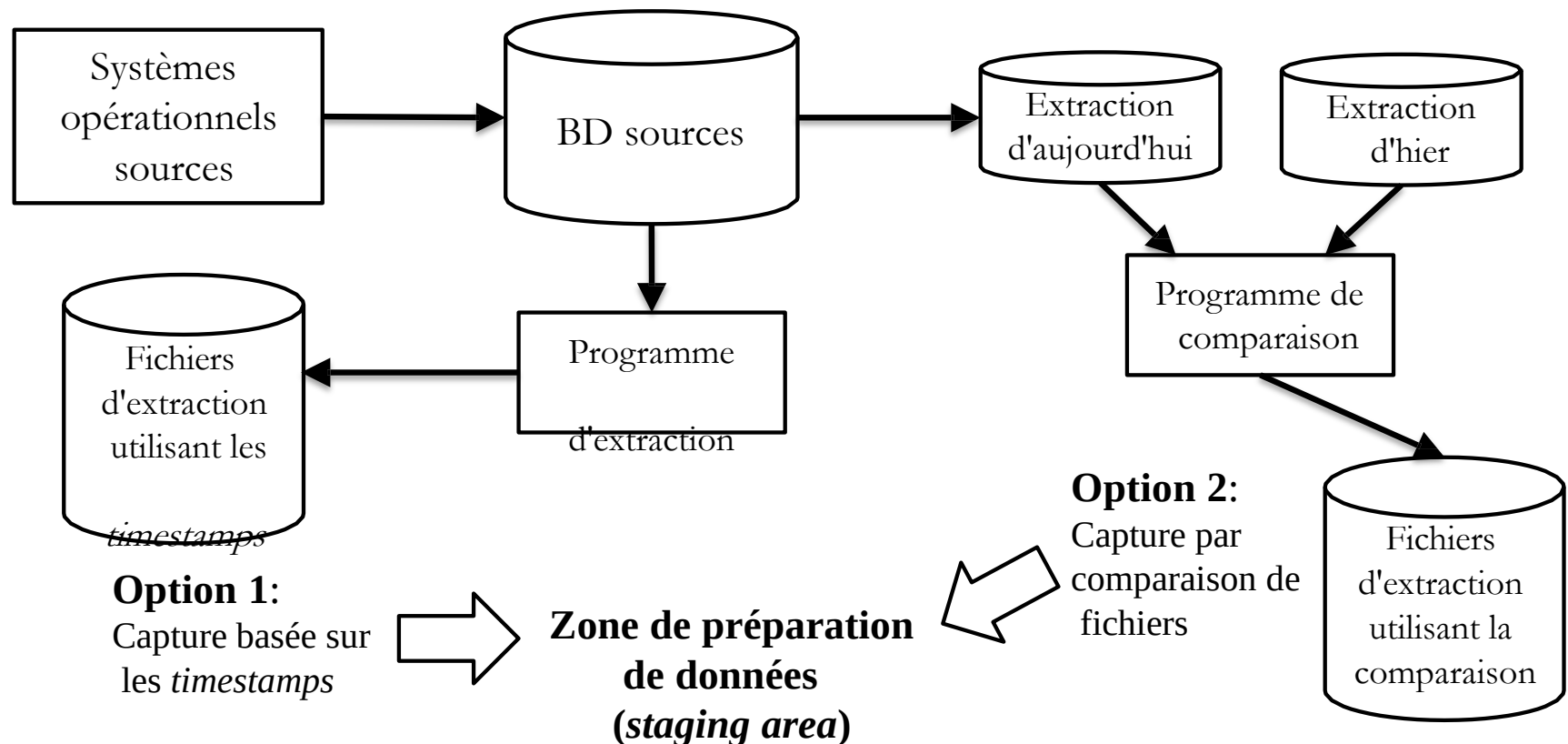


# Processus Extract, Transform, Load (ETL)

- Extraire les données des sources hétérogènes (*extract*)
  - Identifier les données sources utiles
  - Déterminer les données qui ont changé
- Consolider les données (*transform*)
  - Données redondantes, manquantes, incohérentes, etc.
  - Découpage, fusion, conversion, aggrégation, etc.
- Charger les données intégrées dans l'entrepôt (*load*)
  - Mode différé (batch) ou quasi temps-réel.
- Partie la plus longue du développement (jusqu'à 70% du temps total).

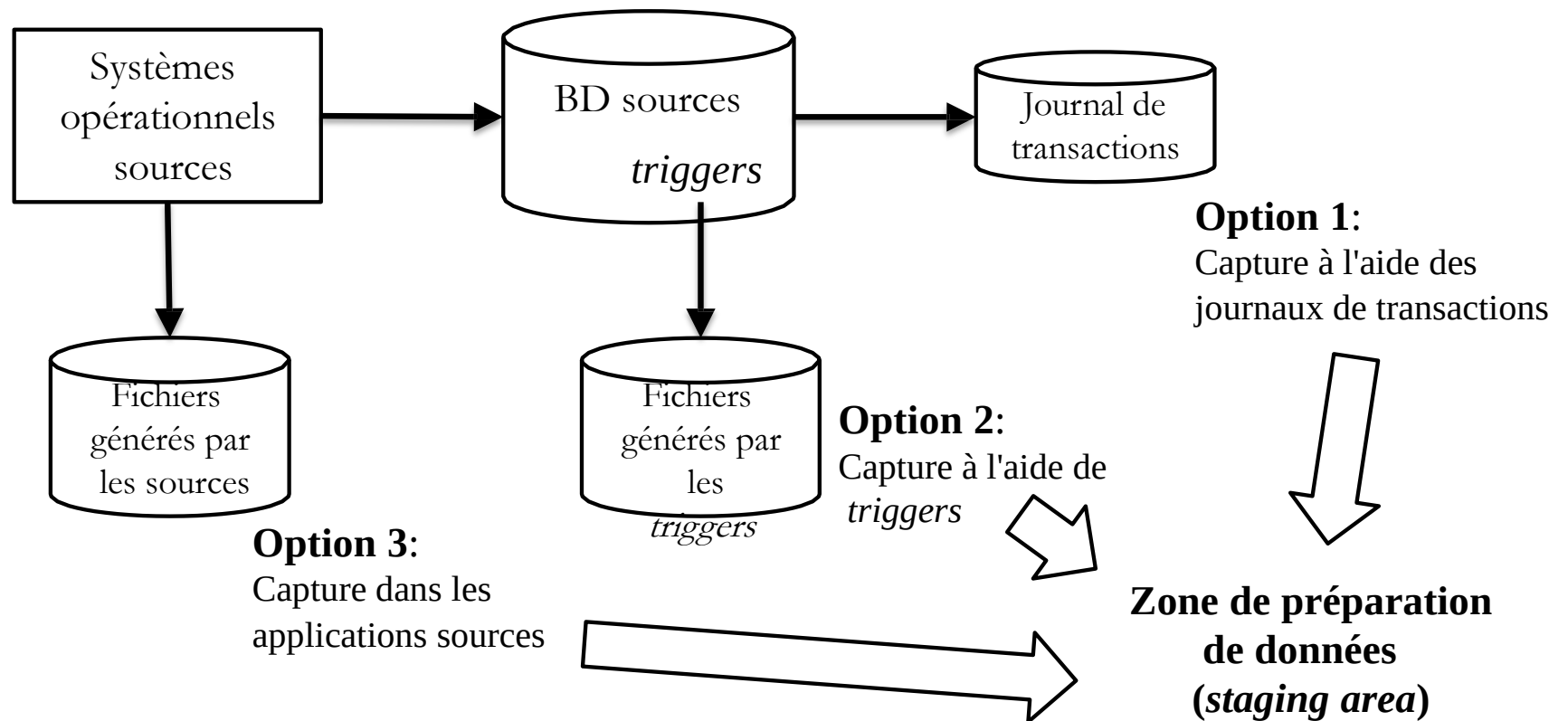
# Extraction des données (différée)

- Extrait tous les changements survenus durant une période donnée (ex: heure, jour, semaine, mois).



# Extraction des données (temps-réel)

- S'effectue au moment où les transactions surviennent dans les systèmes sources.



# Transformation des données

- Révision de format:
  - Ex: Changer le type ou la longueur de champs individuels.
- Décodage de champs:
  - Ex: ['homme', 'femme'] vs ['M', 'F'] vs [1,2].
- Pré-calcul des valeurs dérivées:
  - Ex: *profit* calculé à partir de *ventes* et *coûts*.
- Découpage de champs complexes:
  - Ex: extraire les valeurs *prénom*, *secondPrénom* et *nomFamille* à partir d'une seule chaîne de caractères *nomComple*t.
- Pré-calcul des agrégations:
  - Ex: ventes par produit par semaine par région.
- Déduplication
  - Ex: Plusieurs enregistrements pour un même client

# Chargement des données

- Faire les chargements en lot dans une période creuse (entrepôt de données non utilisé);
- Considérer la bande passante requise pour le chargement;
- Avoir un plan pour évaluer la qualité des données chargées dans l'entrepôt;
- Commencer par charger les données des tables de dimension;
- Désactiver les indexes et clés étrangères lors du chargement.

# Modélisation dimensionnelle

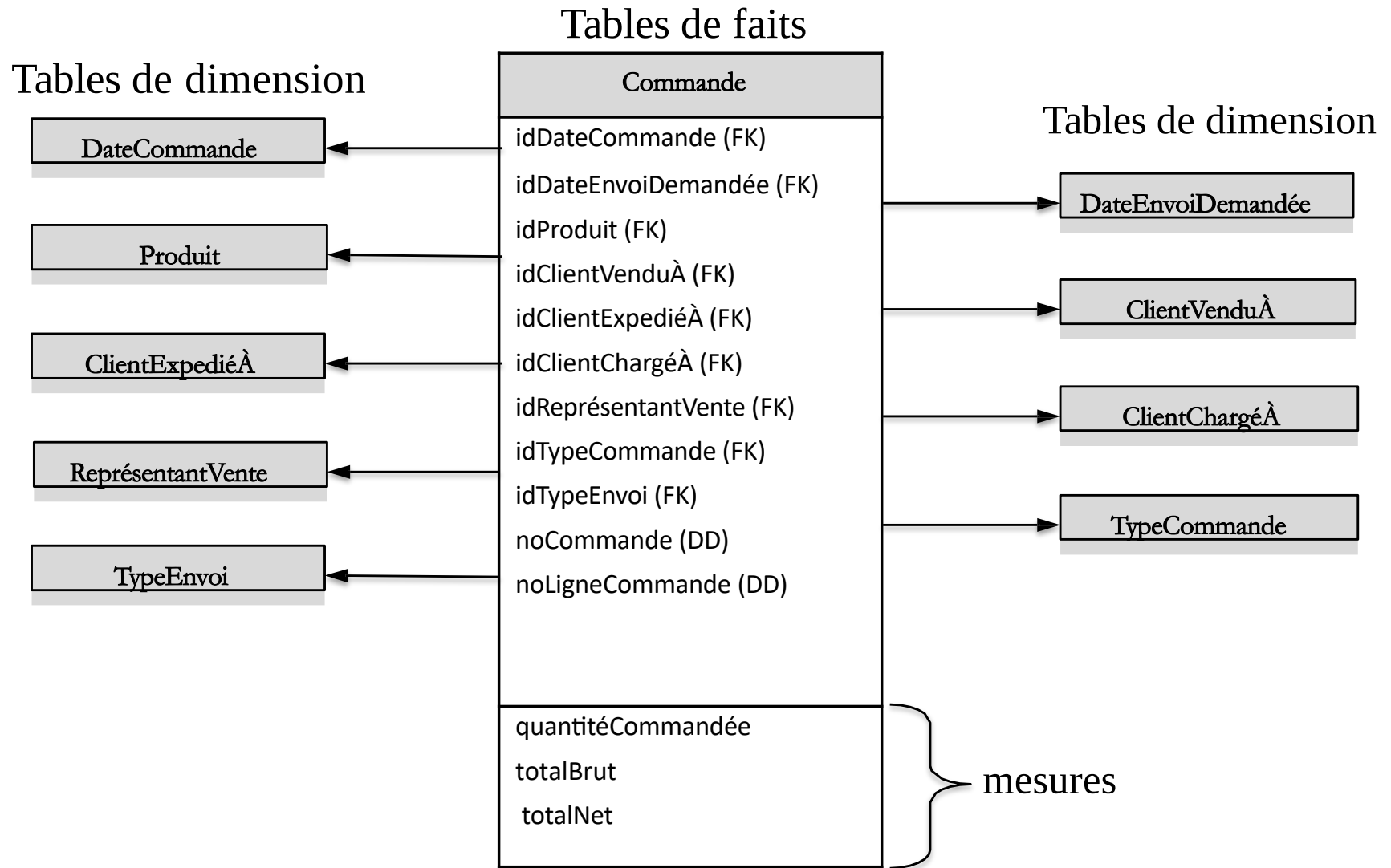
- Représente les données sous la forme d'un **schéma en étoile**:
  - Table de faits entourée de plusieurs tables de dimension (normalement entre 8 et 15)
- Les faits (mesures) sont généralement des valeurs numériques provenant des processus d'affaires;
- Les dimensions fournissent le contexte (*qui, quoi, quand, où, pourquoi et comment*) des faits;
- Les tables ne sont pas normalisées

# Avantages versus modèle entité-relation

- Compréhensibilité:
  - Données regroupées selon des catégories d'affaires qui ont un sens pour les utilisateurs d'affaires;
- Performance:
  - La dénormalisation évite les jointures coûteuses;
  - Autres optimisations (ex: index de jointure en étoile).

# Exemple de schéma en étoile

(Commande de produits)





# Exemple de schéma en étoile (Commande de produits)

Tables de dimension:

DateCommande
idDate (PK)
date
jourDeSemaine
jourDuMois
jourDeAnnée
jourDansMoisFiscal
jourDansAnnéeFiscale
congéFérié
jourDeTravail
semaineDuMois
...

Produit
idProduit (PK)
description
SKU
marque
sousCatégorie
catégorie
département
poids
taille
couleur
...

ClientExpédiéÀ
idClient (PK)
nomFamille
prénom
sexe
dateNaissance
dateAbonnement
forfaitAbonnemen
t adresseRue
adresseVille
adresseProvince
...

# Tables de faits

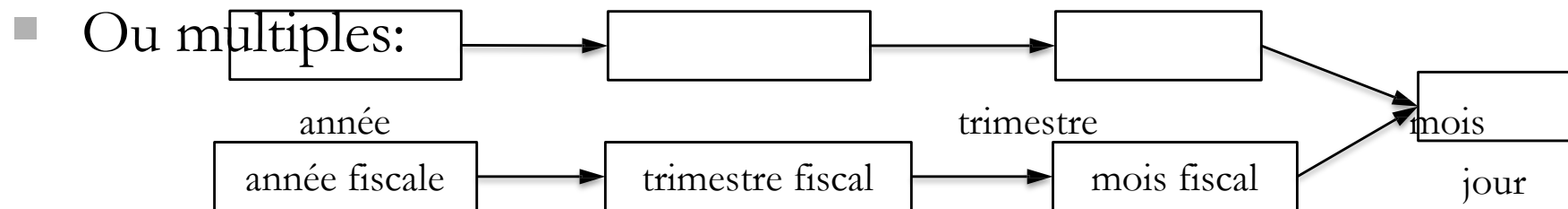
- Correspondent à un événement d'affaires
  - Ex: achat d'un produit par un client, envoi du produit au client, commande de matériaux auprès d'un fournisseur, etc.
- Contiennent deux types de colonnes:
  - Des métriques associées à l'événement d'affaire:
    - Ex: total des ventes, nombre d'items commandés, etc.
  - Des clés étrangères vers les tables de dimension:
    - Ex: ID du client qui fait la commande, ID du produit commandé, etc.
- Contiennent typiquement un très grand nombre de lignes:
  - Jusqu'à plusieurs milliards de lignes;
  - Souvent plus de 90% des données du modèle.

# Tables de dimension

- Ensemble hautement corrélé d'attributs (*jusqu'à plusieurs dizaines*) regroupés selon les objets clés d'une entreprise:
  - Ex: produits, clients, employés, installations, etc.
- Propriétés des attributs:
  - Descriptif (ex: chaînes de caractères);
  - De qualité (ex: aucune valeur manquante, obsolète, erronée, etc.);
  - Valeurs discrètes (ex: jour, âge d'un client);
- Rôles des attributs:
  - Filtrer/agréger les données (ex: ville, catégorie produit, etc.);
  - Étiqueter les résultats (ex: champs descripteurs).

# Hiérarchies dimensionnelles

- Ensemble d'attributs d'une table de dimension ayant une relation hiérarchique ( $x$  est inclus dans  $y$ );
- Correspondent à des relations de type 1 à plusieurs;
- Définissent les chemins d'accès dans les données (*drill-down paths*);
- Peuvent être simples:
  - **Produit** : tous  $\rightarrow$  catégorie  $\rightarrow$  marque  $\rightarrow$  produit;
  - **Lieu** : tous  $\rightarrow$  pays  $\rightarrow$  province  $\rightarrow$  ville  $\rightarrow$  code postal.



# Dimension temporelle

Dimension: Temps
idDate (PK)
date
jourDeSemaine
jourDuMois
jourDeAnnée
jourDansMoisFiscal
jourDansAnnéeFiscale
congéFérié
jourDeTravail
semaineDuMois
...

- Mettre toutes ces valeurs, même si la plupart peuvent être déduites d'une seule colonne;
- Pré-générer les lignes de la table (ex: 10 prochaines années) pour faciliter la référence et éviter les mises à jour

# Dimension temporelle

- Problème: avoir un grain trop fin dans la dimension temporelle (ex: temps du jour) peut causer l'explosion du nombre de rangées:
  - Ex: 31,000,000 secondes différentes dans une année.
- Solution: mettre le temps du jour dans une dimension séparée:
  - **Dimension Date** : année → mois → jour;
  - **Dimension TimeOfDay** : heure → minute → secondes;
  - 86,400 + 365 lignes au lieu de 31,000,000 lignes.
- Note: la dimension TimeOfDay est souvent modélisée comme un simple champs dans la table de faits.

# Dimensions à évolution lente (SCD)

- Slowly Changing Dimensions (SCD);
- Même si elles sont plus statiques que les tables faits, les dimensions peuvent également changer:
  - Ex: adresse d'un client, catégorie d'un produit, etc.
- Stratégies d'historisation:
  - **SCD Type 1**: Écraser l'ancienne valeur avec la nouvelle
  - **SCD Type 2**: Ajouter une ligne dans la table de dimension pour la nouvelle valeur
  - **SCD Type 3**: Avoir deux colonnes dans la table de dimension correspondant à l'ancienne et la nouvelle valeur

# Stratégie SCD Type 1

idProduit	description	code	catégorie
1001	'BébéLala'	'ABC999-Z'	'Éducation'



idProduit	description	code	catégorie
1001	'BébéLala'	'ABC999-Z'	'Stratégie'

- Impossible de faire des analyses sur l'ancienne valeur;
- À utiliser seulement lorsque l'ancienne valeur n'est pas significative pour les besoins d'affaires;
- Exige de mettre à jour les données agrégées avec l'ancienne valeur.



## Stratégie SCD Type 2

idProduit	description	code	catégorie	dateEffective	dateExpirée
1001	'BébéLala'	'ABC999-Z'	'Éducation'	'2007-10-08'	'9999-12-31'



idProduit	description	code	catégorie	dateEffective	dateExpirée
1001	'BébéLala'	'ABC999-Z'	'Éducation'	'2007-10-08'	'2008-10-31'
1002	'BébéLala'	'ABC999-Z'	'Stratégie'	'2008-11-01'	'9999-12-31'

- Permet de faire des analyses historiques;
- Demande l'ajout d'une nouvelle ligne par changement;
- À utiliser lorsque l'ancienne valeur a une signification analytique ou si le changement est une information en soi.

# Stratégie SCD Type 3

idProduit	description	code	oldCatégortie	newCatégorie	dateModif
1001	'BébéLala'	'ABC999-Z'	'Éducation'	'Éducation'	'9999-12-31'

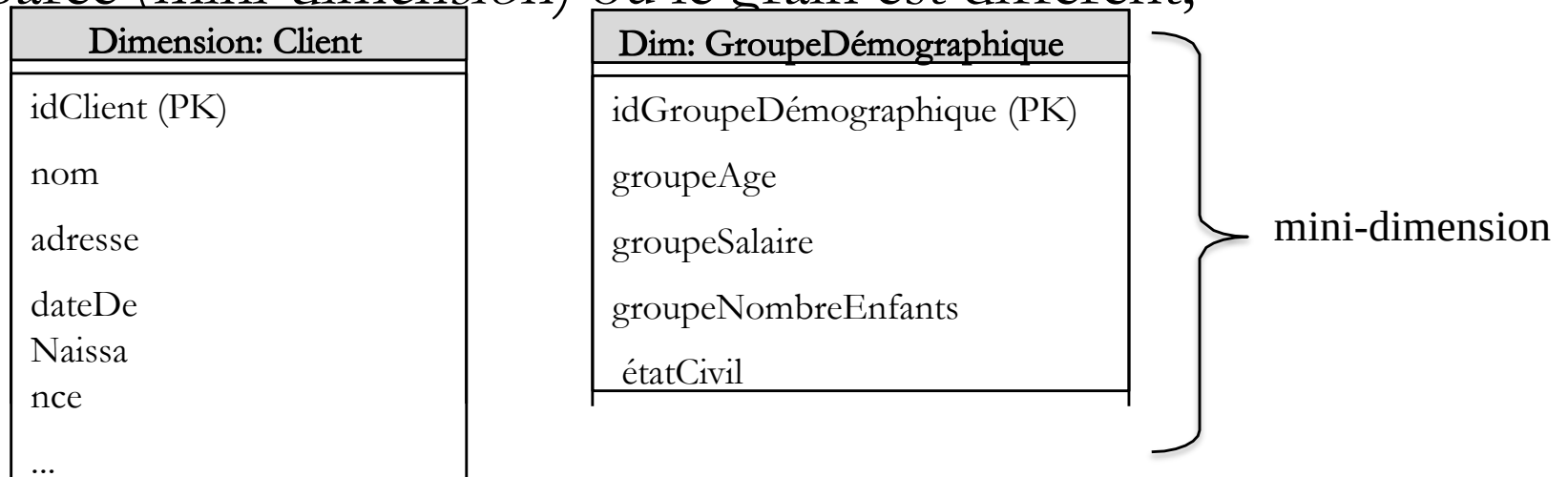


idProduit	description	code	oldCatégortie	newCatégorie	dateModif
1001	'BébéLala'	'ABC999-Z'	'Éducation'	'Stratégie'	'2008-11-01'

- Rarement employée;
- Profondeur de l'historique est de un seul changement;
- Utilisé lorsqu'on veut vouloir comparer les faits avec l'ancienne ou la nouvelle valeur;

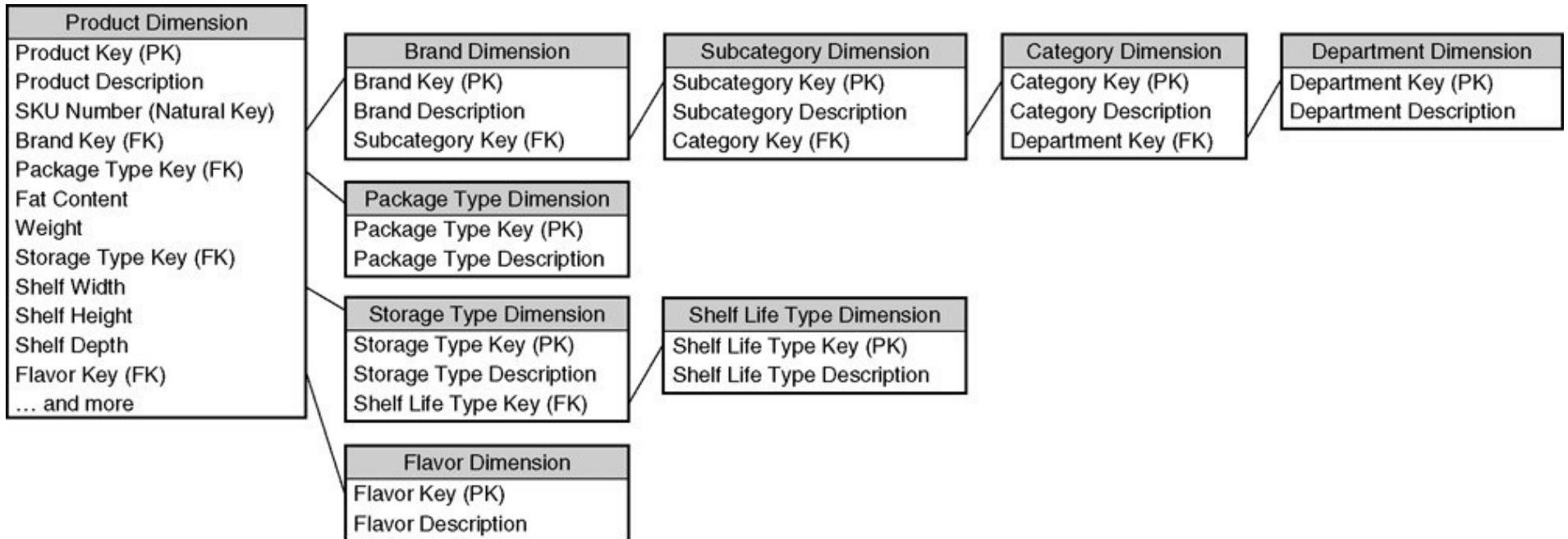
# Mini-dimensions

- Sert lorsque qu'une dimension renferme des attributs qui peuvent changer souvent:
  - Ex: le profil démographique des clients (âge, revenu, etc.).
- Solution: mettre les attributs plus volatiles dans une dimension séparée (*mini-dimension*) où le grain est différent;



# Schéma en flocon

- Provient de la normalisation des tables de dimension;
- Exemple:



# Schéma en flocon

- Avantages:
  - Petite économie d'espace;
  - Plus facile de mettre à jour les dimensions en cas de changement.
- Inconvénients:
  - Schéma moins intuitif aux utilisateurs d'affaires;
  - Dégradation de la performance à cause des jointures additionnelles.
- En général, on préfère ne pas normaliser les tables de dimension.

# Types d'entrepôts de données

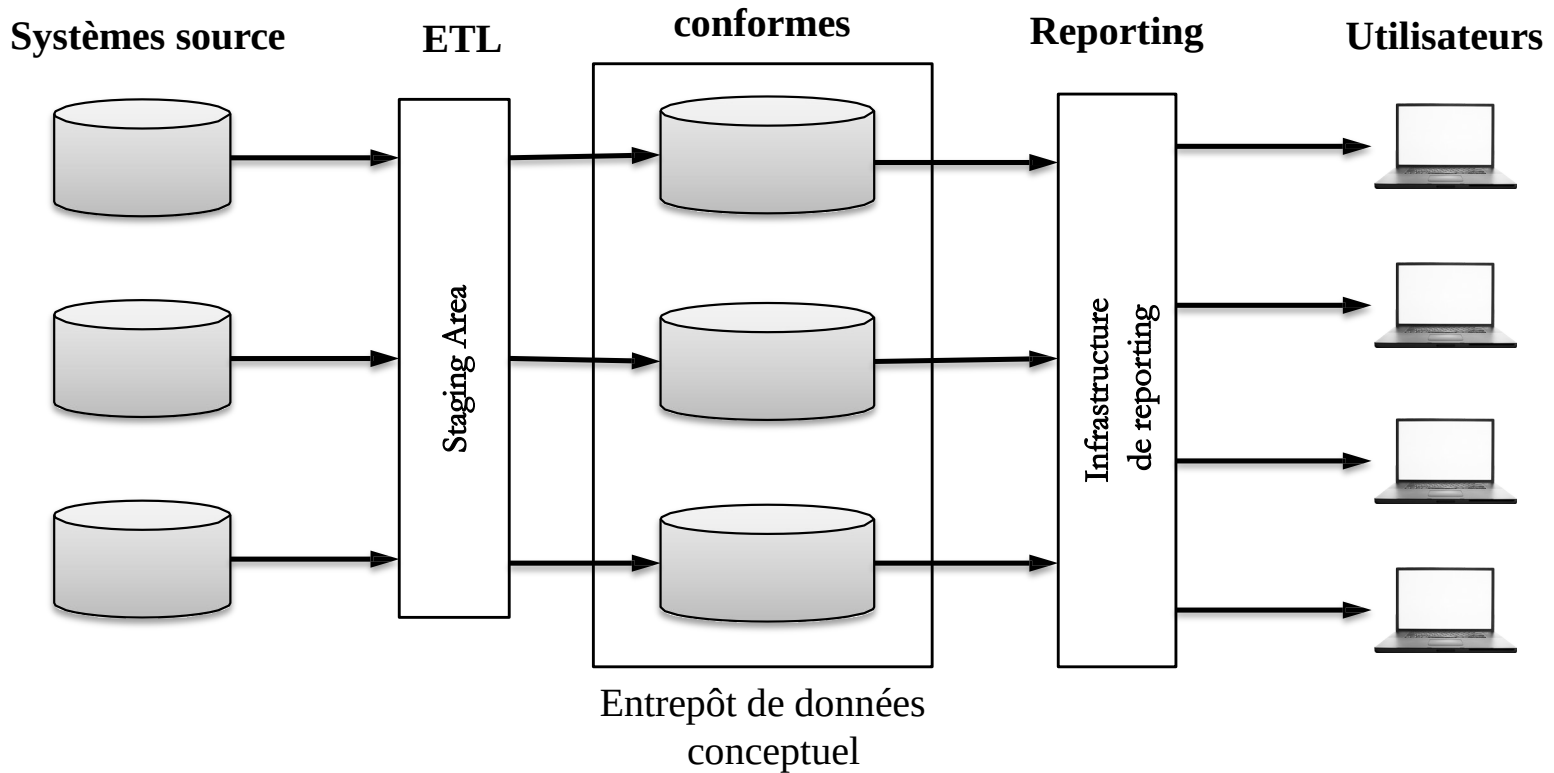
1. Magasins de données
2. Entrepôts de données d'entreprise (EDW)
  - Bus de magasins de données (datamart bus)
  - Hub-and-spokes
  - Entrepôts de données fédérés

# Magasins de données (datamart)

- Contiennent une portion du contenu de l'entrepôt de données;
- Se concentre sur un seul sujet d'analyse (ex: les ventes OU l'inventaire, mais pas les deux);
- Servent à faire des analyses simples et spécialisées (ex: fluctuations des ventes par catégorie de produits);
- Nombre de sources limitées, provenant la plupart du temps d'un même département;
- Modélisés sous la forme d'un schéma en étoile.

# Architecture Datamart bus

Datamarts liés  
par  
dimensions

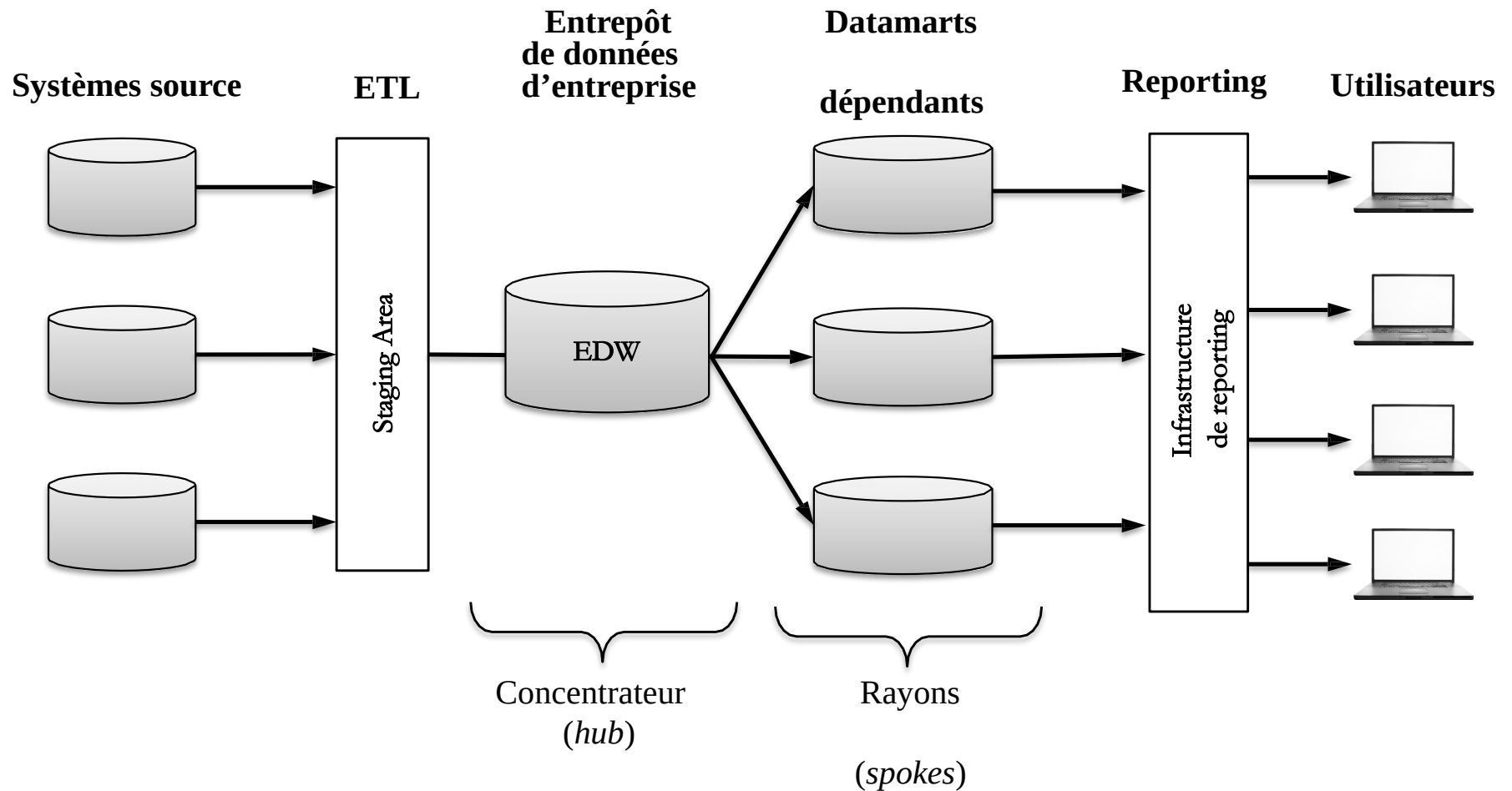




# Architecture Datamart bus

- Approche *bottom-up*, où on construit l'entrepôt un datamart à la fois;
- Modélisation dimensionnelle (*schéma en étoile*) des datamarts, au lieu du diagramme entité-relation;
- Entrepôt de données conceptuel, formé de magasins de données inter-reliés à l'aide d'une couche d'intergiciels (*middleware*).
- Intégration des données assurée par les dimensions partagées entre les datamarts (i.e., *dimensions conformes*);
- Approche incrémentale qui donne des résultats rapidement (développement agile);

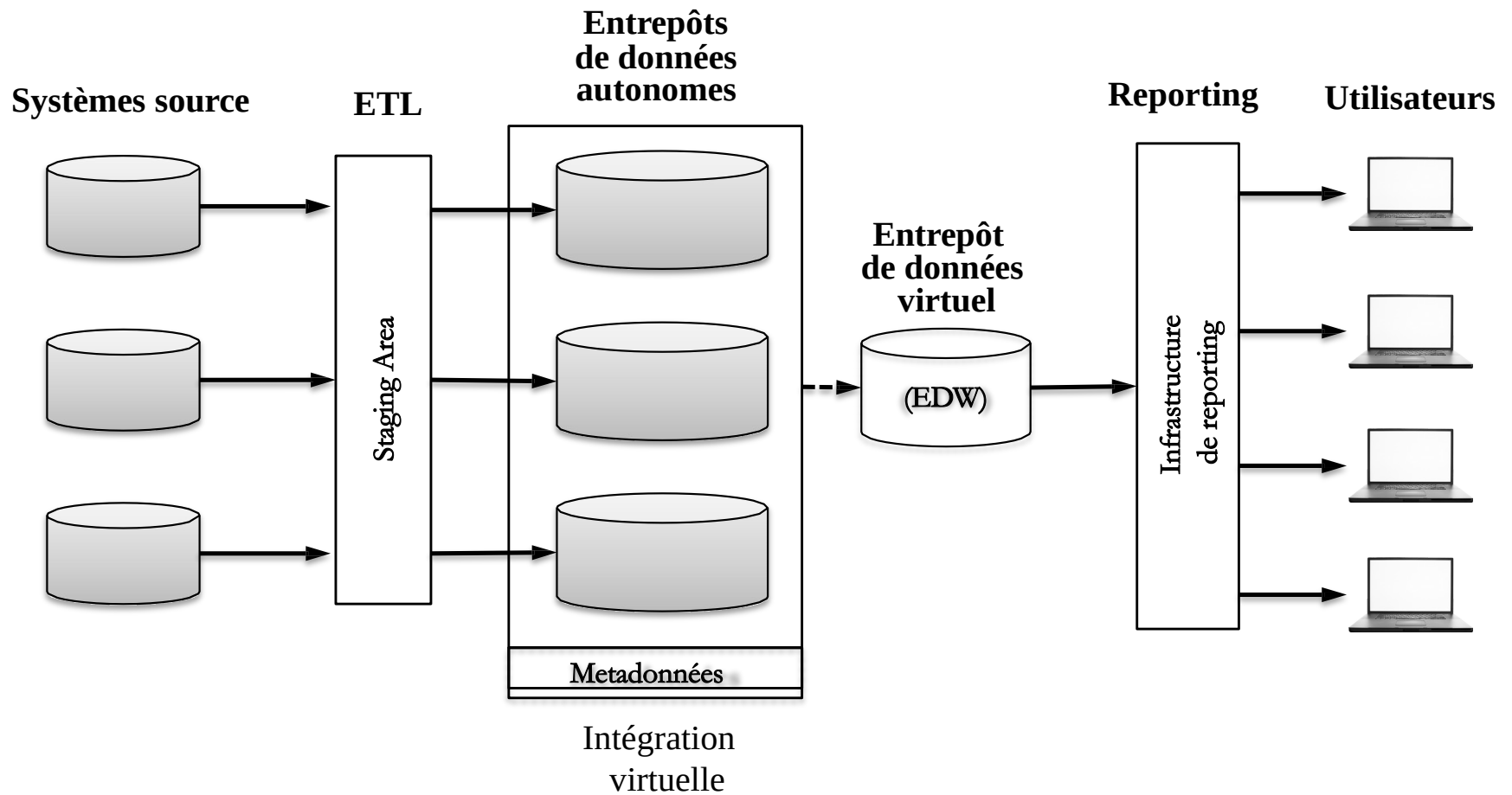
# Architecture Hub-and-spokes



# Architecture Hub-and-spoke

- Approche *top-down*, favorisant l'intégration et consolidation complète des données de l'entreprise
- Entrepôt (*hub*) contient les données **atomiques** (niveau de détail le plus fin) et **normalisées** (3FN);
- Les datamarts (*spokes*) contiennent principalement des données **agrégées** (pas atomique) et suivant le **modèle dimensionnel**;
- La plupart des requêtes analytiques sont faites sur les datamarts;
- Développement plus long, dû à la complexité du processus ETL et de la modélisation;
- Meilleure qualité de données que l'architecture par bus de datamarts.

# Architecture fédérée



# Architecture fédérée

- Entrepôt de données distribué sur plusieurs systèmes hétérogènes;
- Données intégrées logiquement ou physiquement à l'aide de méta-données (ex: XML);
- Opère de manière transparente (l'utilisateur ne voit pas que les données sont réparties);
- Utile lorsqu'il y a déjà un entrepôt en place (ex: acquisitions ou fusions de compagnies);
- Très complexe (synchronisation, parallélisme, concurrence, etc.) et faible performance.

# Stockage vertical

- La plupart des RDBMS transactionnels stockent les données horizontalement:

ligne 1			ligne 2			ligne 3
col 1	col 2	col 3	col 1	col2	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:

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

# Index de jointure en étoile (star join)

- Pré-calcule les lignes des tables de dimension pouvant être jointes avec la table de faits;
- Évite de joindre les tables de dimension les unes après les autres.
- Bitmap join index (Oracle)
  - Index bitmap sur des colonnes situées dans des tables de dimension différentes;
  - Les colonnes à pré-joindre doivent avoir un domaine restreint (comme pour les index bitmap standards);
  - Peut accélérer jusqu'à 10 fois la jointure (benchmarks Oracle).

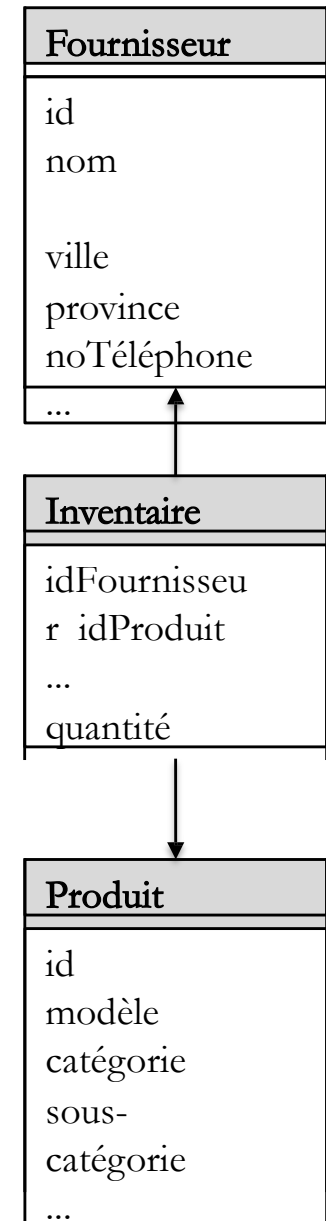
# Bitmap join index (Oracle)

- 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
```





# Requêtes analytiques

- Exemple:
  - 300 magasins;
  - 40,000 produits;
  - 80 marques (500 produits par marque);
  - Environ une 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.

# Pré-agrégation des données

- Accélère les requêtes analytiques en pré-calculant l'agrégation de faits à différents niveaux des hiérarchies dimensionnelles
- Duplique l'information contenue dans la table de faits atomique (niveau le plus granulaire)
- Exemple (suite):
  - Table pré-agrégée 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 7.8 millions.

# Stratégie d'agrégation

- Tenir compte du type et de la fréquence des requêtes faites à l'entrepôt (profilage de requêtes)
- Choisir un niveau de hiérarchie offrant un bon compromis entre l'utilité et le gain en performance
  - Règle: Chaque nouvelle ligne doit agréger au moins 10 lignes de la table de faits atomiques
- Approches d'agrégation:
  - BD relationnelle avec vue matérialisées (ROLAP)
  - Cube de données multidimensionnelles (MOLAP)

# Agrégation par vues matérialisées (ROLAP)

- Table physique synchronisée avec les résultats d'une requête
- Synchronisation temps-réel, en lot ou sur demande
- Permet les index, 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 REWRITE  
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

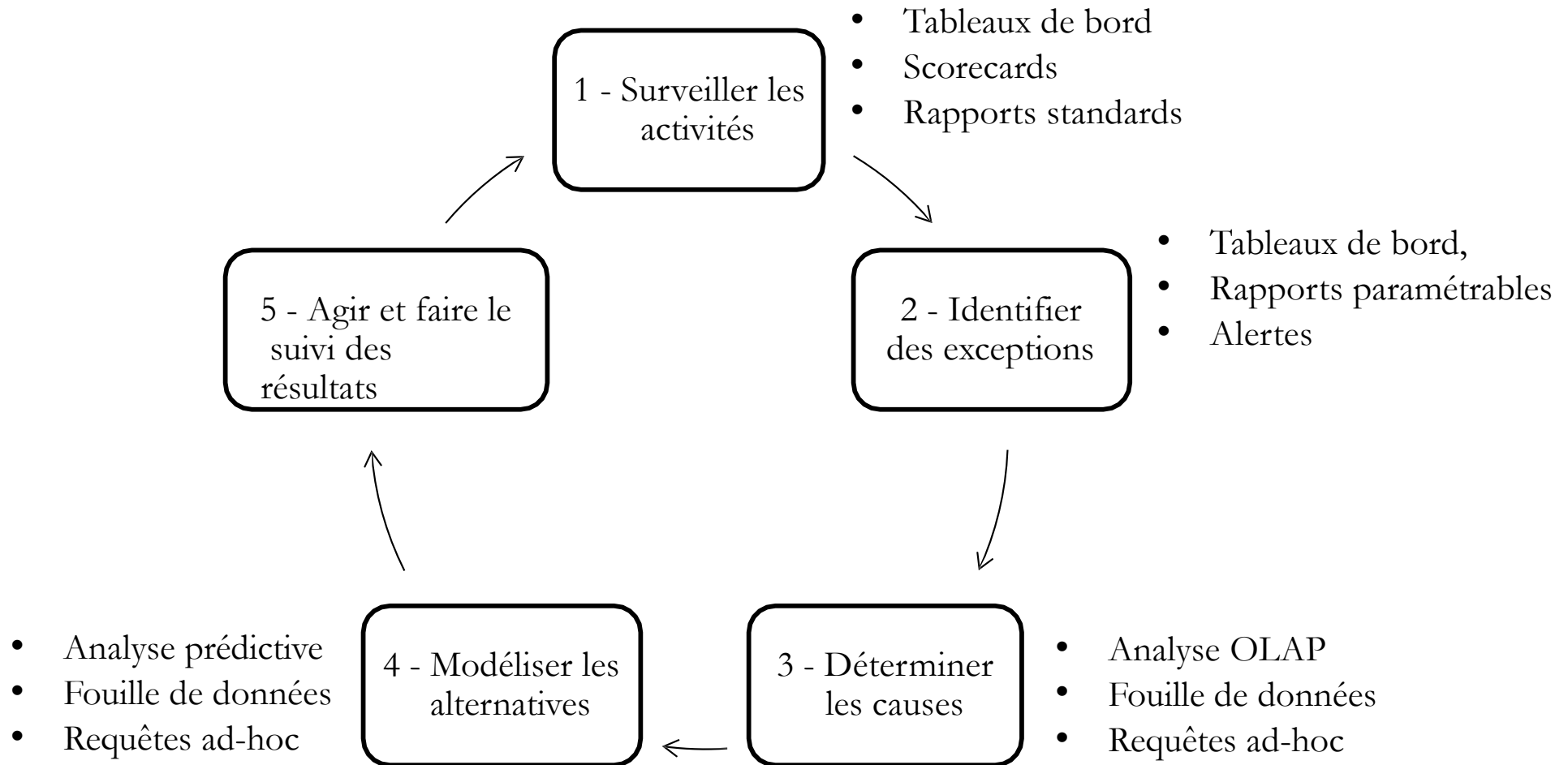
# Cubes multi-dimensionnels (MOLAP)

- 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*);
- Se basent beaucoup sur le pré-calcul d'agrégations selon les hiérarchies dimensionnelles

# Cubes multi-dimensionnels (MOLAP)

- Offrent une meilleure performance pour les opérations d'analyse multidimensionnelle (ex: *slicing* et *dicing*) que ROLAP;
- Facilitent la gestion des agrégations;
- Peuvent être limités en terme du nombre de dimensions et de valeurs possibles pour ces dimensions (explosion de la mémoire pour les agrégations);
- Les mises à jour peuvent être plus coûteuses que ROLAP;
- La technologie est plus fermée que les ROLAP.
- Plusieurs produits commerciaux offrent une approche hybride (HOLAP) combinant les avantages du MOLAP et du ROLAP.

# Cycle analytique et types d'application



# Fonctions analytiques Oracle

```
SELECT col1, col2, ..., FctAnalytique(args) OVER  
([PARTITION BY <...>]          -- partitionnement des  
lignes  
[ORDER BY <...>]              -- ordre des lignes dans  
les partitions  
[<fenetre>]                   -- fenetre autour de la  
ligne courante  
)
```

- Pour chaque partition, les lignes sont parcourues selon l'ordre défini par le ORDER BY
- Pour chaque ligne, la fonction est évaluée sur l'ensemble des lignes autour de celle-ci, défini par la fenêtre
- Exemples de fonctions:
  - AVG, CORR, FIRST, RANK, LAG, LAST, MAX, MIN, etc.
- Exemples de fenêtres:
  - ROWS BETWEEN <x> PRECEDING and <y> FOLLOWING
  - ROWS UNBOUNDED PRECEDING



# Fonctions analytiques Oracle

```
SELECT noClient, dateCommande, montant, SUM(montant) OVER (
  PARTITION BY noClient
  ORDER BY dateCommande
  ROWS UNBOUNDED PRECEDING) AS montantCumulatifDuClient
FROM Vente
ORDER BY noClient, dateCommande
```

NOCLIENT	DATECOMMAN	MONTANT	MONTANTCUMULATIFDUCLIENT
1	10/01/2000	100	100
1	15/01/2000	300	400
1	25/02/2000	100	500
1	04/03/2000	200	700
1	10/04/2000	200	900
1	15/04/2000	100	1000
1	05/06/2000	200	1200
2	10/01/2000	200	200
2	16/01/2000	200	400
2	20/02/2000	200	600
2	15/03/2000	500	1100
2	20/05/2000	200	1300
2	05/06/2000	100	1400
3	10/01/2000	500	500
3	15/01/2000	100	600
3	05/04/2000	200	800
3	06/04/2000	400	1200
3	25/05/2000	500	1700
4	20/02/2000	400	400
4	04/03/2000	300	700
4	15/03/2000	100	800

## Requête:

Montant cumulatif par client.

## Partition:

Fonction appliquée par groupes de clients;

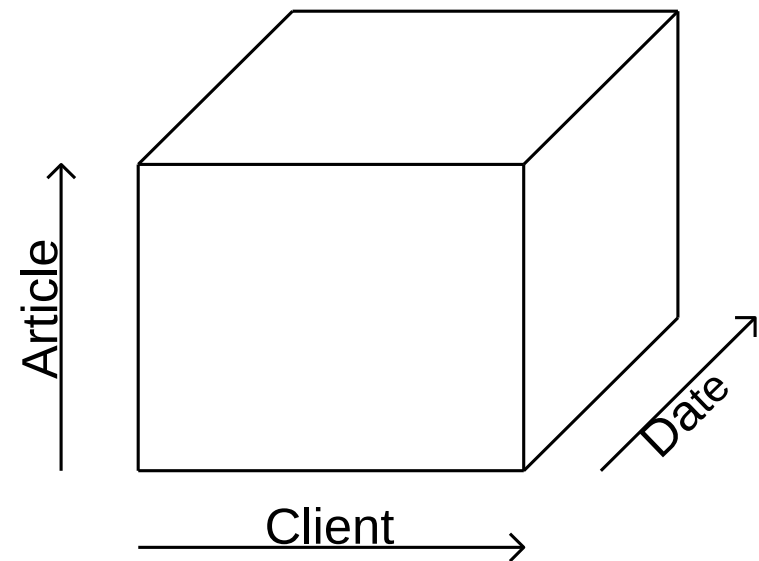
## ORDER BY et ROWS

## UNBOUNDED PRECEDING:

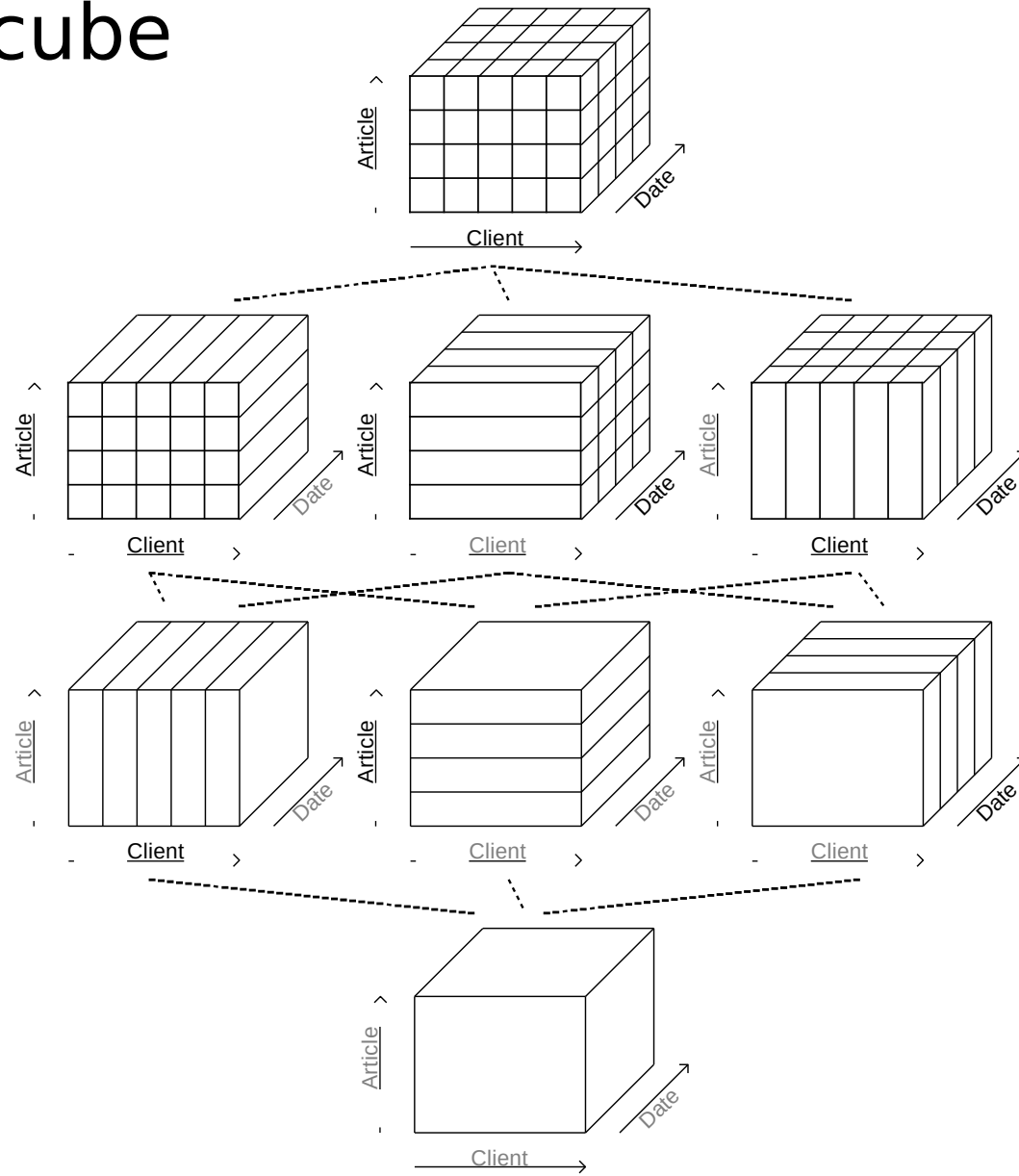
Fonction appliquée dans une fenêtre allant de la ligne courante jusqu'au début de la table, où les lignes sont en ordre de date.

# Exemple de cube OLAP

Table Vente			
noClient	noArticle	dateVente	montant
1	10	10/01/2000	100
2	20	10/01/2000	200
3	10	10/01/2000	500
1	10	15/01/2000	300
3	40	15/01/2000	100
2	60	16/01/2000	200
4	60	20/02/2000	400
2	10	20/02/2000	200
1	40	25/02/2000	100
4	10	04/03/2000	300
1	20	04/03/2000	200
...			



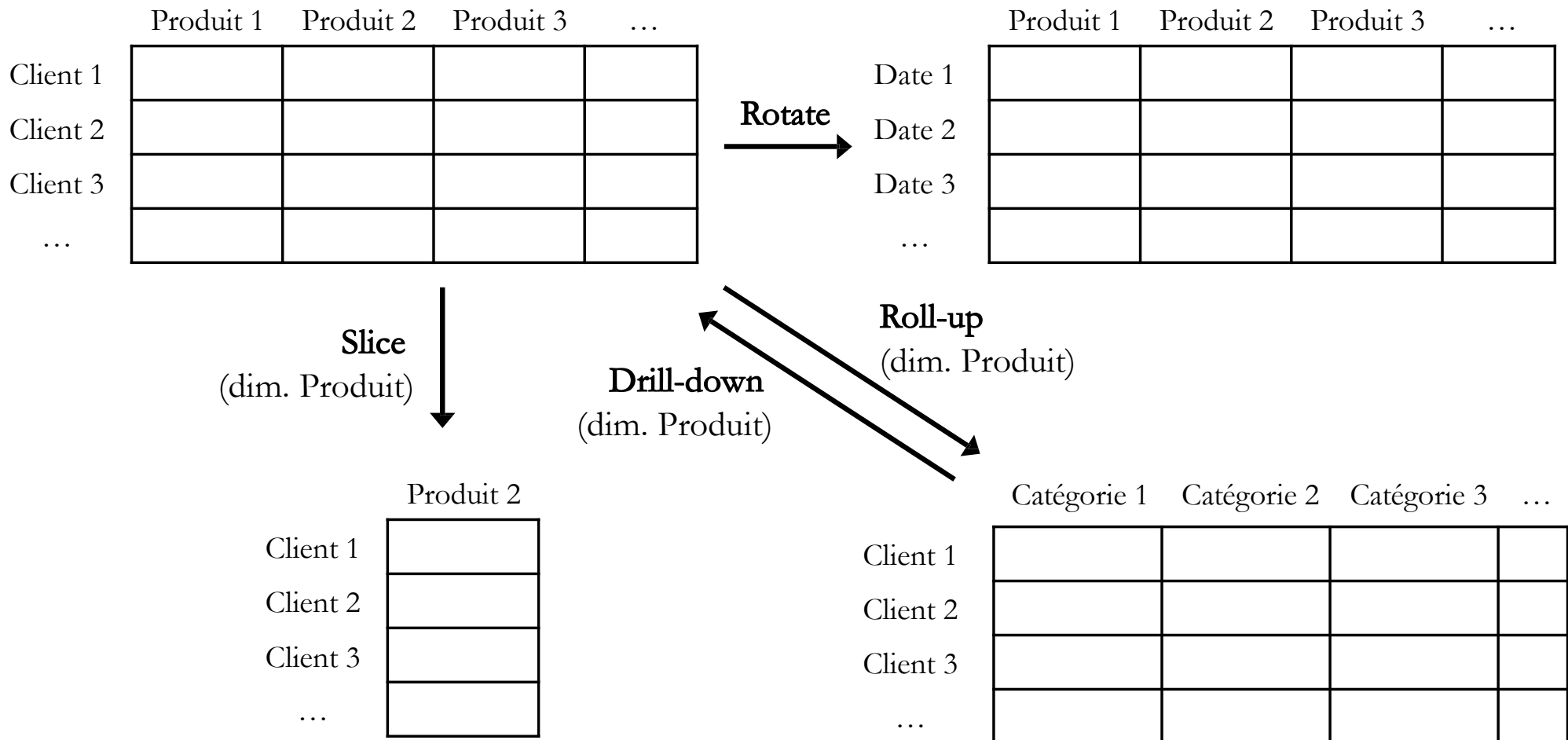
# Hiérarchie du cube



# Analyse multi-dimensionnelle (OLAP)

- Représente les données en cube multi-dimensionnel où chaque côté est une dimension d'analyse et chaque case une métrique
- Opérations sur le cube:
  - Rotate: sélection du pivot d'analyse en faisant tourner le cube;
  - Slice: extrait une tranche du cube, c'est-à-dire, un sous-ensemble des valeurs du cube correspondant à une certaine valeur d'une des dimensions non-découpées;
  - Drill-down: détaille les données en descendant le niveau hiérarchique d'une dimension;
  - Roll-up: agrège les données en augmentant le niveau hiérarchique d'une dimension;

# Analyse multi-dimensionnelle (OLAP)

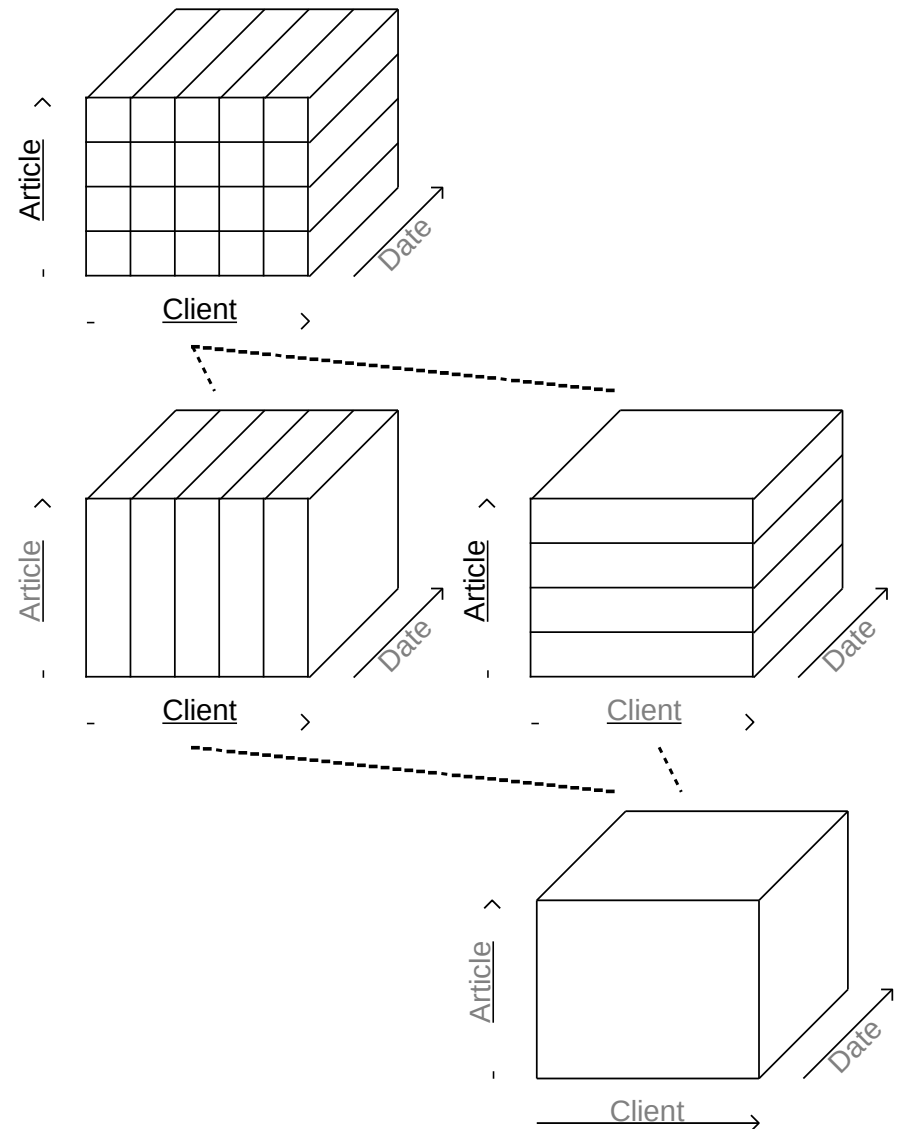


Note: Catégorie correspond à la catégorie d'un produit (hiérarchie de la dim. Produit)

# Cause CUBE (SQL99)

```
SELECT noClient,noArticle,SUM(montant)
FROM Vente
GROUP BY CUBE(noClient,noArticle)
```

NOCLIENT	NOARTICLE	SUM(MONTANT)
1	10	500
1	20	200
1	40	100
1	50	200
1	60	200
1		1200
2	10	700
2	20	300
2	60	400
2		1400
3	10	1000
3	20	400
3	40	100
3	50	200
3		1700
4	10	300
4	60	500
4		800
	10	2500
	20	900
	40	200
	50	400
	60	1100
		5100



# Cause CUBE (SQL99)

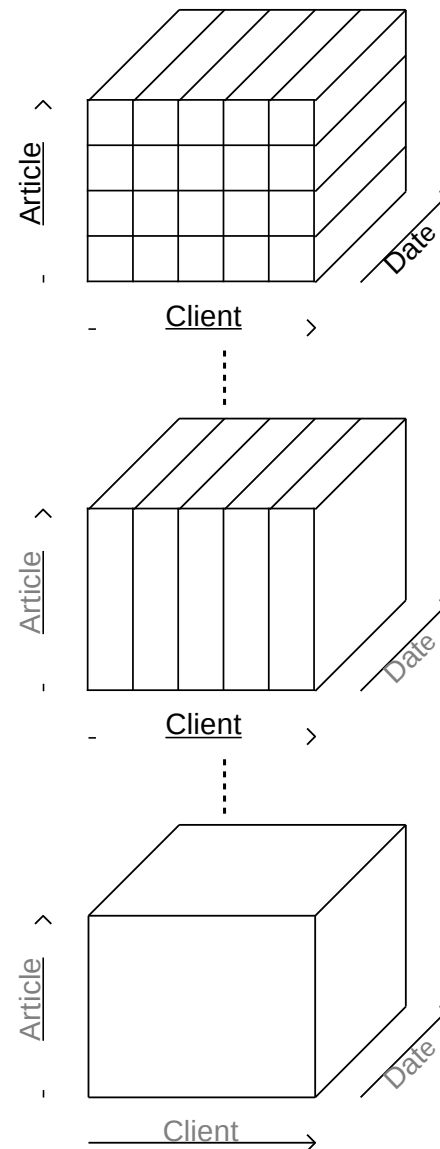
## ■ Résultat:

		<i>noArticle</i>					
		10	20	40	50	60	total
<i>noClient</i> <i>t</i>	1	500	200	100	200	200	<b>1200</b>
	2	700	300	0	0	400	<b>1400</b>
	3	1000	400	100	200	0	<b>1700</b>
	4	300	0	0	0	500	<b>800</b>
	total	<b>2500</b>	<b>900</b>	<b>200</b>	<b>400</b>	<b>1100</b>	<b>5100</b>

# Cause ROLLUP (SQL99)

```
SELECT noClient,noArticle,SUM(montant)
FROM Vente
GROUP BY ROLLUP(noClient,noArticle)
```

NOCLIENT	NOARTICLE	SUM(MONTANT)
1	10	500
1	20	200
1	40	100
1	50	200
1	60	200
1		1200
2	10	700
2	20	300
2	60	400
2		1400
3	10	1000
3	20	400
3	40	100
3	50	200
3		1700
4	10	300
4	60	500
4		800
		5100





# OLAP graphique (Palo-pivot)

File Manager

Files New Spreadsheet New Palo Pivot

Filter Products Datatypes  
All Products All Datatypes

Months Years

Regions  
Measure

		Jan		Feb	
		2005	2006	2005	2006
Europe	Units	381.312,00	569.041,03	318.205,00	474.714,83
	Gross Profit	348.924,19	249.042,00	324.050,64	227.555,49
West	Units	160.272,00	240.925,06	138.157,00	208.832,75
	Gross Profit	149.972,98	107.441,21	149.845,49	102.522,80
East	Units	80.196,00	120.231,47	64.564,00	96.071,91
	Gross Profit	69.805,86	51.261,45	56.147,94	59.174,49
South	Units	96.021,00	141.146,48	72.882,00	106.172,23
	Gross Profit	84.725,28	61.196,57	82.770,73	40.574,01
North	Units	44.823,00	66.738,03	42.602,00	63.637,95
	Gross Profit	44.420,08	29.142,78	35.286,48	25.284,19

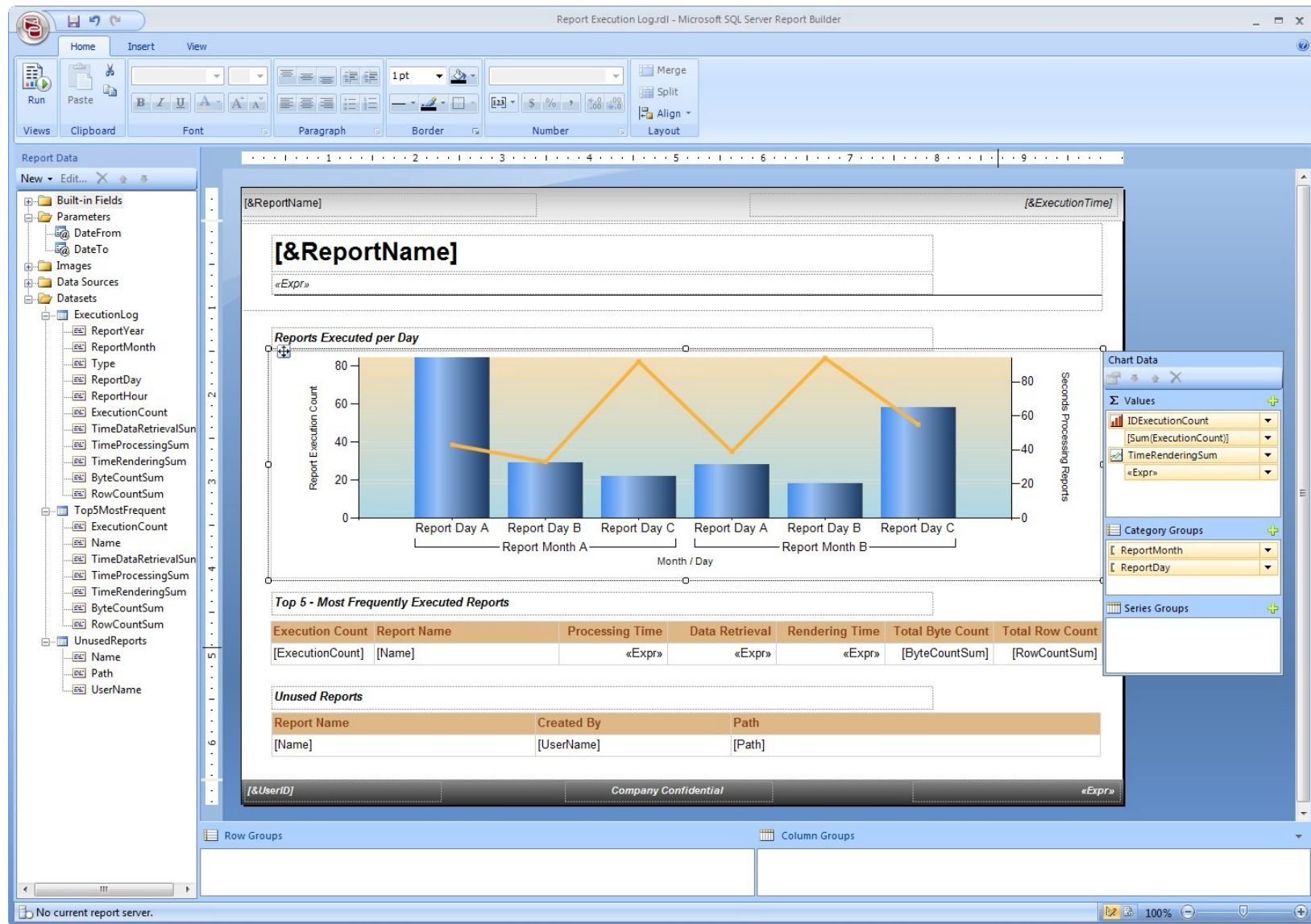
# Reporting

- Rapports préformatés:
  - Librairie de rapports fréquemment employés avec une description de leur contenu
  - Exemples:
    - Ventes courantes versus ventes de l'année précédente;
    - Taux mensuel d'attrition par plan de service;
    - Taux de réponses courrier par promotion par produit;
- Rapports paramétrables:
  - Permettent de modifier facilement le contenu à l'aide de paramètres (ex: choix d'un produit, d'une région, etc.)

# Reporting

- Outils de conception:
  - Permettre aux utilisateurs de pouvoir créer facilement de nouveaux rapports à l'aide d'une interface conviviale.
- Exécution sur le serveur:
  - Éviter de surtaxer l'ordinateur de l'utilisateur en exécutant le rapport sur un serveur dédié.
- Rapports cédulés:
  - Permettre aux utilisateurs de cédule l'envoi de rapports à des intervalles ou lors d'événements prédéterminés.

# MS SQL Server Reporting Services



# Tableaux de bords

- Montrent les indicateurs de performance à l'aide d'éléments visuels:
  - Graphiques (ex: courbes, tarte, etc.)
  - Jauges (ex: profits comparés aux objectifs d'affaires)
  - Feux de circulation (ex: rouge signifie problème)
- Requièrent très peu de connaissances techniques de la part des utilisateurs
- Combinent les données de divers systèmes pour offrir une vue unifiée de haut niveau de l'entreprise
- Tiennent souvent compte des changements ponctuels des données (ex: rafraichissement des éléments visuels à intervalles réguliers)

# Tableau de bord (MS Powerpivot)

