Entrepôt de données et l’Analyse en ligne

Déroulement du semaine

- 10 fev : veille, cours et TD
- 11 fev : veille, cours et TP
- 12-14 fev : Brief

Plan de cours

1. Systèmes d'information décisionnelles
2. Entrepôt de données (Data Warehouse)
3. Analyse Multidimensionnelle
4. SQL / OLAP Extensions
5. BigQuery

Plan de cours

1. Systèmes d’information décisionnelles
2. Entrepôt de données (Data Warehouse)
3. Analyse Multidimensionnelle
4. SQL / OLAP Extensions
5. BigQuery

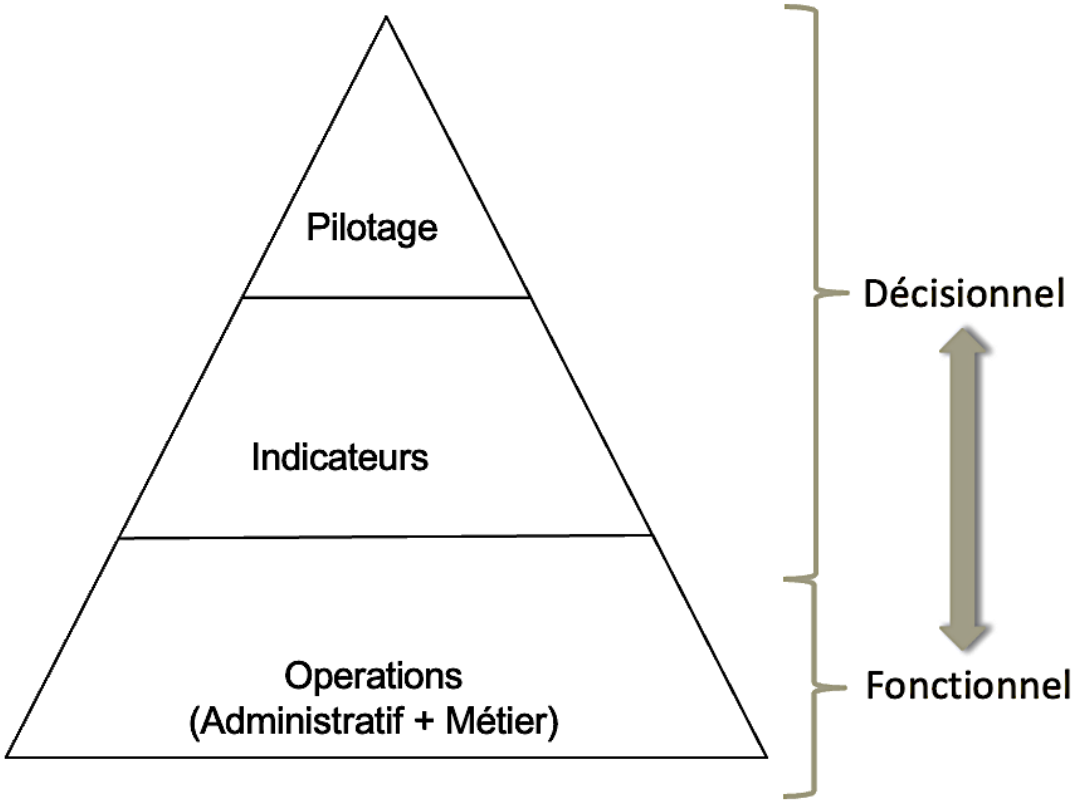
Plan de cours

1. Systèmes d’information décisionnelles
2. Entrepôt de données (Data Warehouse)
3. Analyse Multidimensionnelle
4. SQL / OLAP Extensions
5. BigQuery

Systèmes d’information décisionnelles

Contexte

Un système décisionnel (DW & OLAP) transforme les données opérationnelles en indicateurs pertinents pour permettre un pilotage stratégique efficace de l'entreprise.

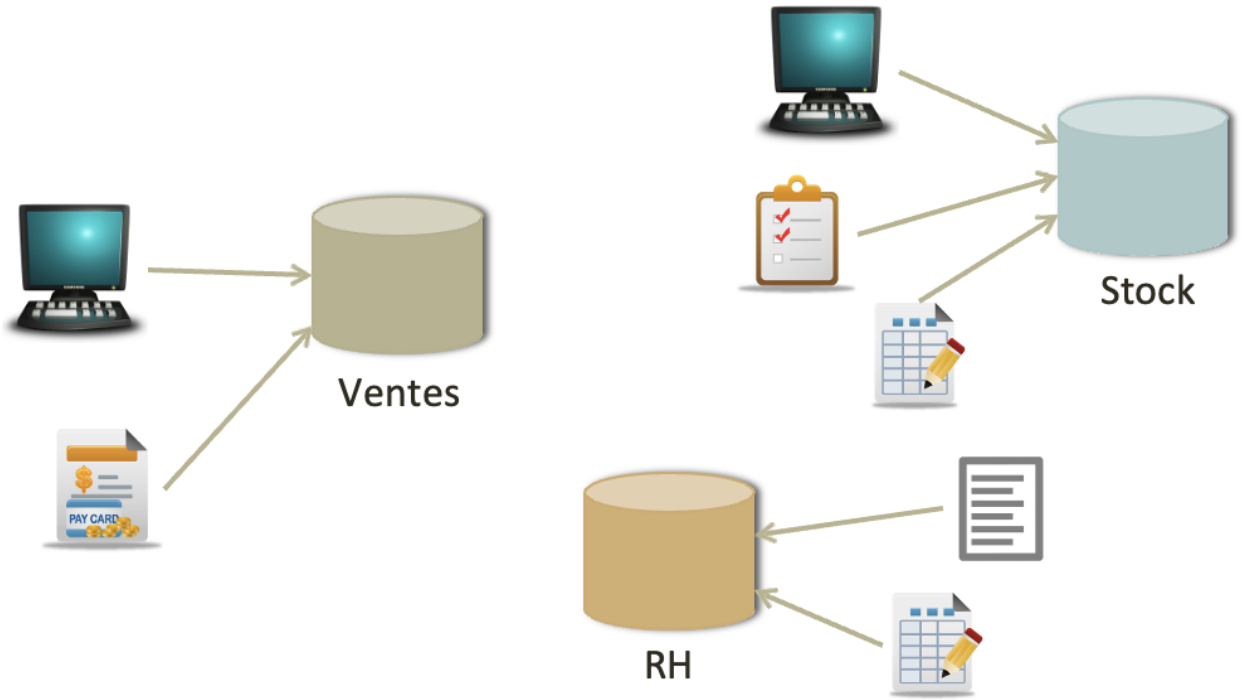


Problématique

- Les entreprises possèdent de nombreuses sources de données potentiellement exploitables

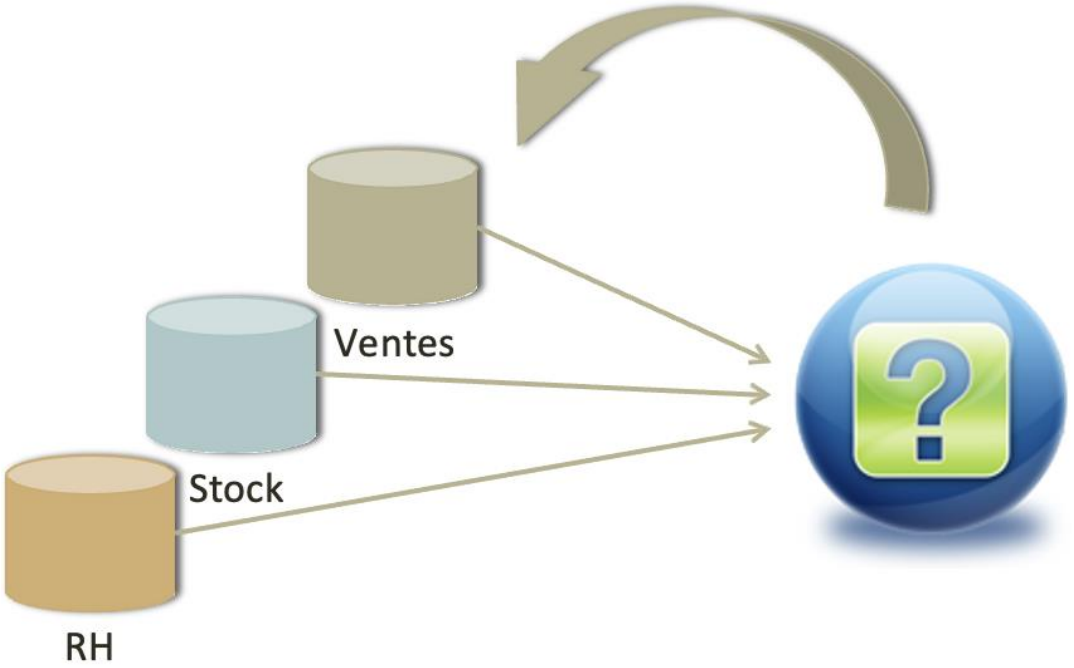
Problématique

- Les sources de données sont disséminées sur diverses bases de données



Problématique

- Le besoin d'analyse exprimé est transversal afin de permettre la prise de décision stratégique

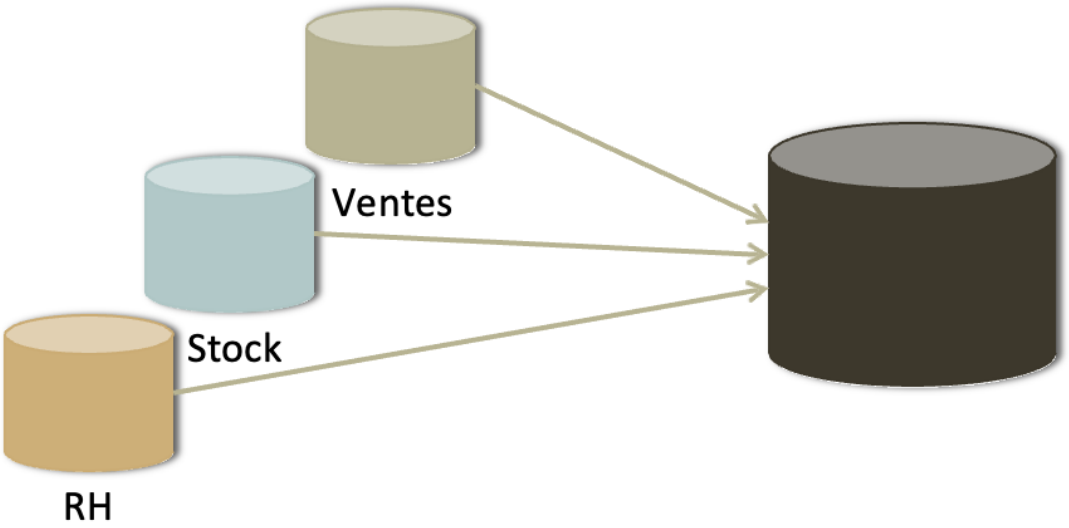


Problématique

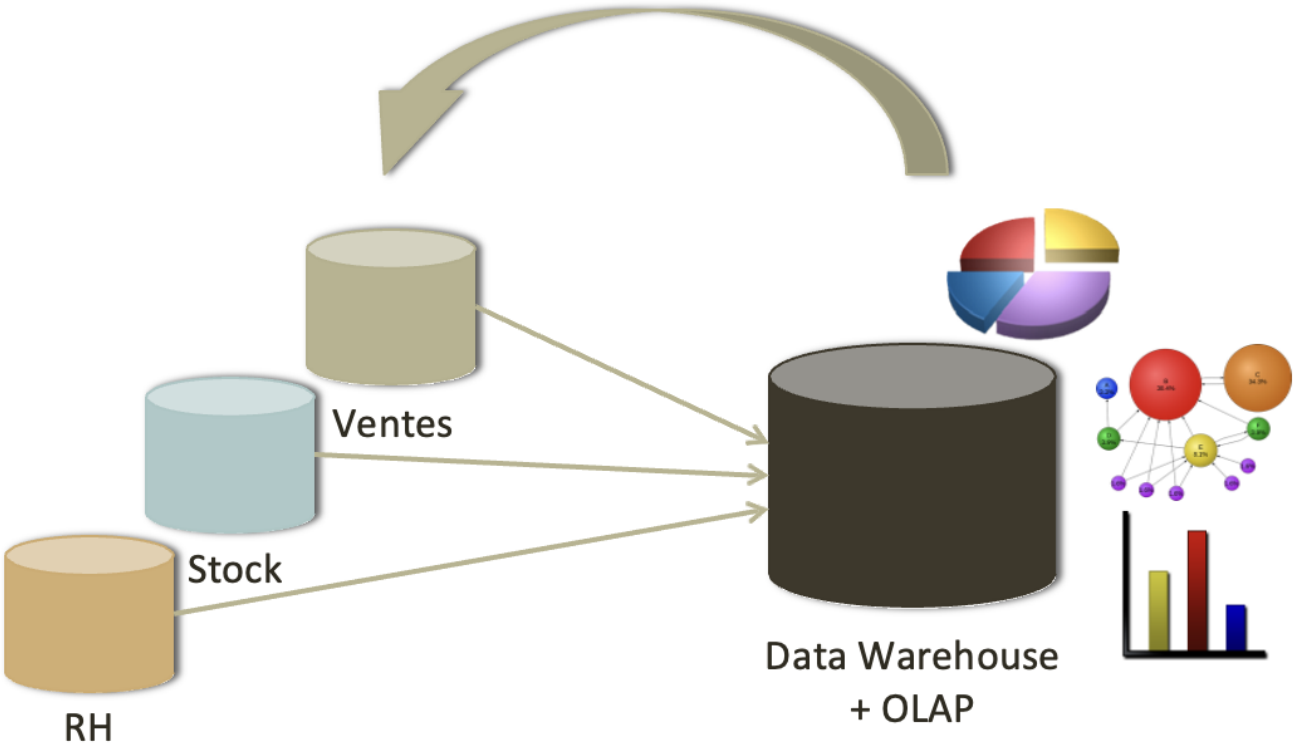
- Type de données : données opérationnelles (de production)
  - Bases de données, fichiers, tickets case, bulletins de paie, …

- Caractéristiques des données :
  - Distribuées : systèmes éparpillés
  - Hétérogènes : systèmes et structures de données différents
  - Détaillées : organisation de données selon les processus fonctionnels et données trop abondantes pour l'analyse
  - Peu/pas adaptées à l'analyse : des requêtes lourdes peuvent bloquer le système transactionnel
  - Volatiles : pas d'historisation systématique

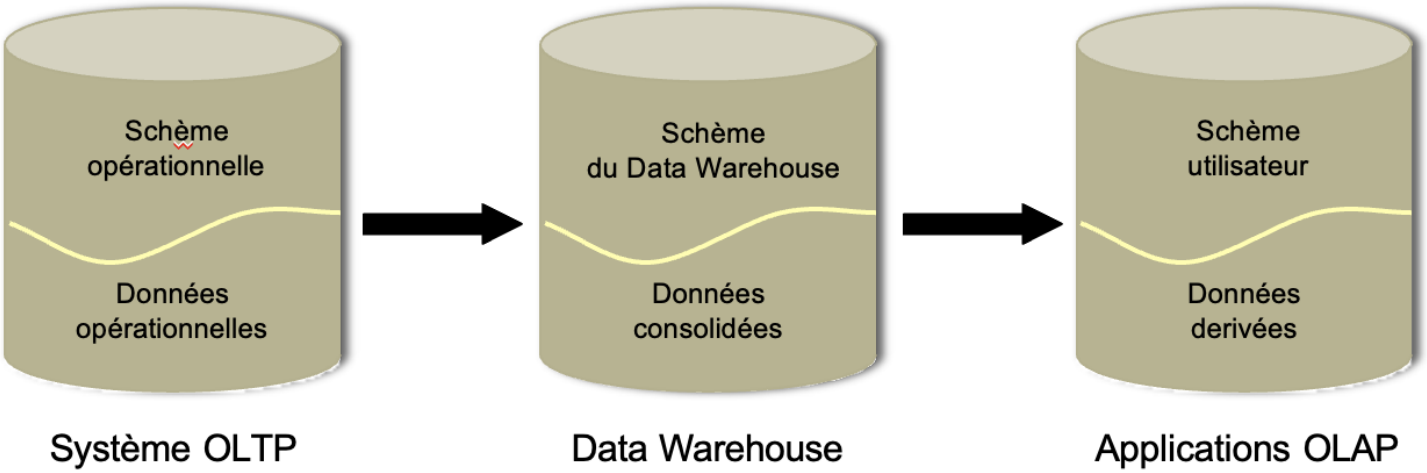
L’idée – centralisation



L’idée – centralisation



Du transactionnel au décisionnel



Les données passent des systèmes opérationnels (OLTP) vers le Data Warehouse où elles sont consolidées, puis vers les outils OLAP où elles deviennent des informations utiles pour la prise de décision.

Pour quoi pas utiliser OLTP?

	OLTP	Data Warehouse
Données	Atomiques Orientée application A jour Dynamiques	Résumées Orientée sujet Historiques Statiques
Utilisateurs	Employés de bureau Nombreux Concurrents Mises à jour Requêtes prédéfinies Réponses immédiates	Analyst es Peu Non concurrents Interrogations Requetés spécifiques Réponses moins rapides
	Access à peu de données	Access à beaucoup d'information

Métaphore du restaurant

Preparation : the kitchen
Quality, consistency, and integrity

Presentation : the dinning room
Food, decor, service, cost

Raw materials

The kitchen

The dinning room

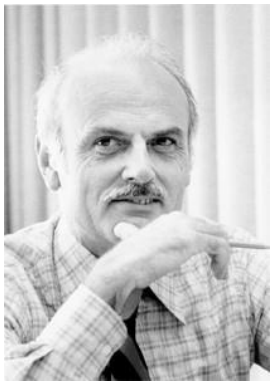
http://www.kimballgroup.com/2004/01/01/data--warehouse--dining--experience/

Plan de cours

1. Systèmes d'information décisionnelles
2. Entrepôt de données (Data Warehouse)
3. Analyse Multidimensionnelle
4. SQL / OLAP Extensions
5. BigQuery

L’entrepôt de données (Data Warehouse)

Les fondateurs



Edgar Frank Codd

- Fondateur du modèle relationnelle (1970)
- Ecrit les douze lois du traitement analytique en ligne (1993)

Bill Inmon

- Formalisé du concept d'entrepôt de données (1994)
- Proposé le modèle Top–down

Ralph Kimball

- Des premiers travaux sur la informatique décisionnelle '70
- Proposé le modèle Botton--up

Définition (B. Inmon) –1994

- « Un entrepôt de données est une collection de données orientées **sujet**, **intégrées**, **non volatiles** et **historiées**, **organisées** pour le support d'un processus d'aide à la décision »

- « Un entrepôt de données ne s'achète pas, il se construit… »

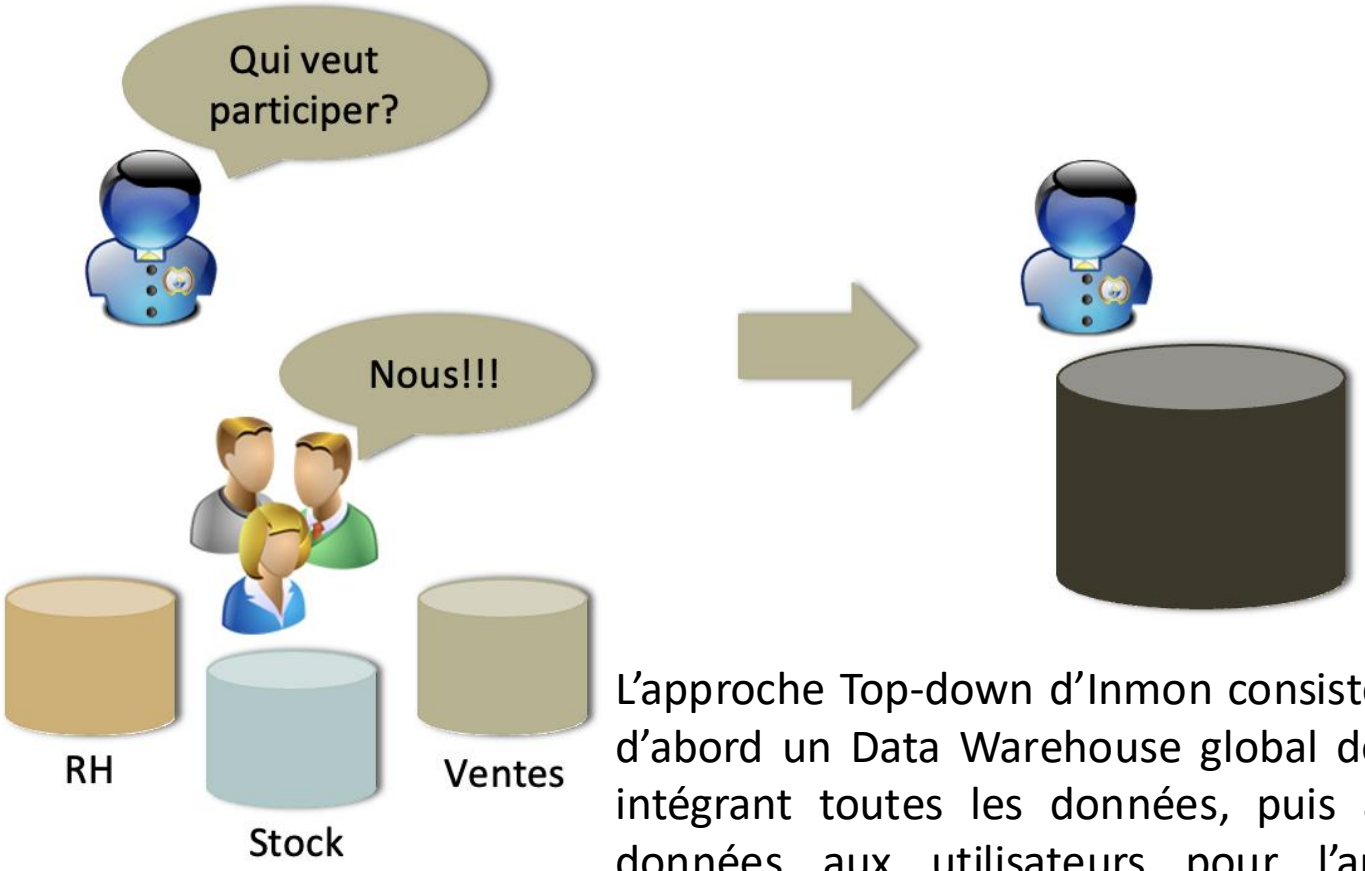
Les approches académiques

	R. Kimball www.kimballgroup.com	**B. Inmon www.inmoncif.com**
Processus	Bottom-up	Top–down
Organisation	Data marts	Data Warehouse
Schématisation	Etoile	Flocon

Deux approches dominent la conception d'un Data Warehouse :

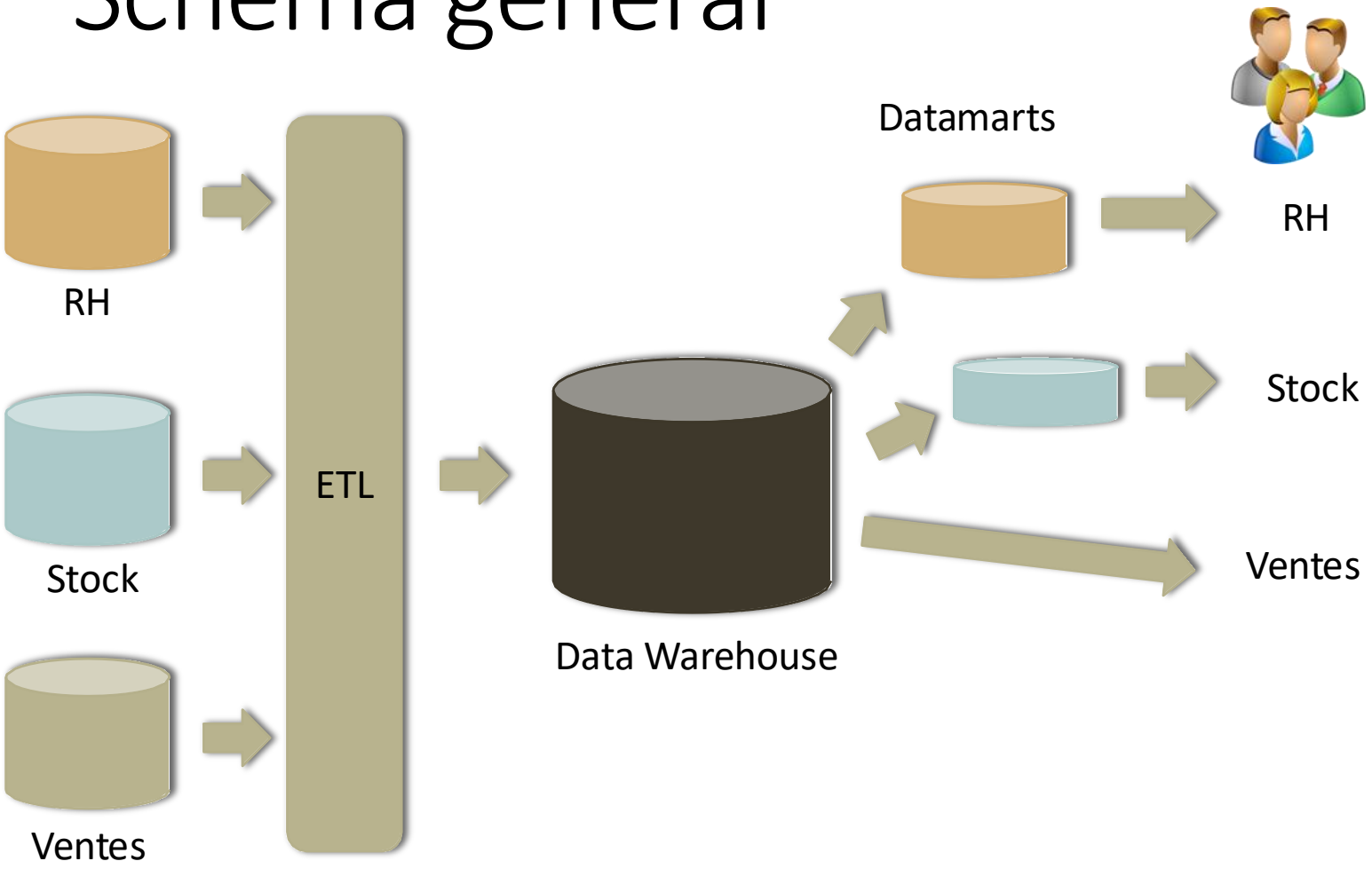
- Kimball (bottom-up, orienté data marts et étoile) et
- Inmon (top-down, orienté Data Warehouse global et flocon).

Approche Top–-down (Inmon)

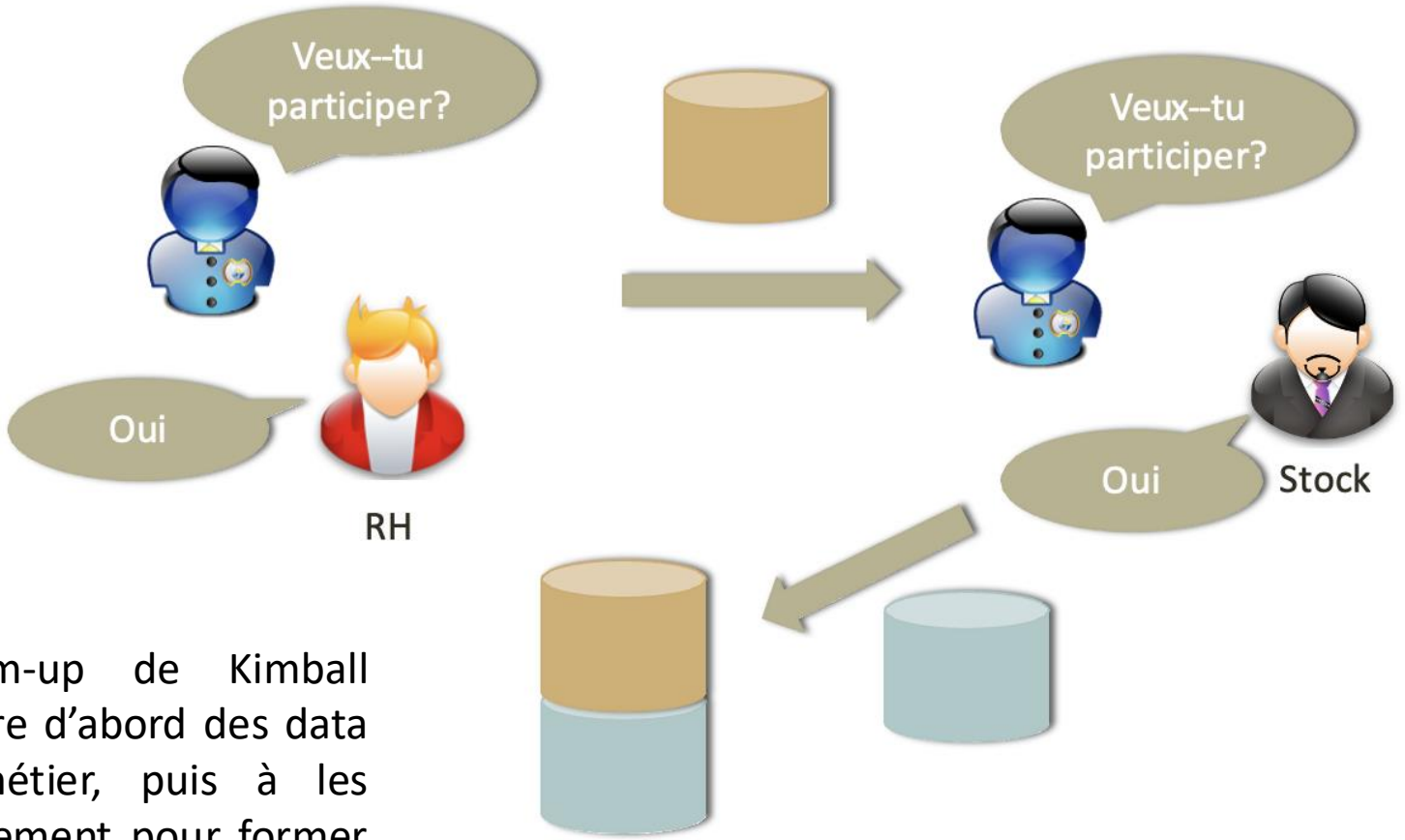


L'approche Top-down d'Inmon consiste à construire d'abord un Data Warehouse global de l'entreprise, intégrant toutes les données, puis à fournir ces données aux utilisateurs pour l'analyse et la décision.

Schéma général

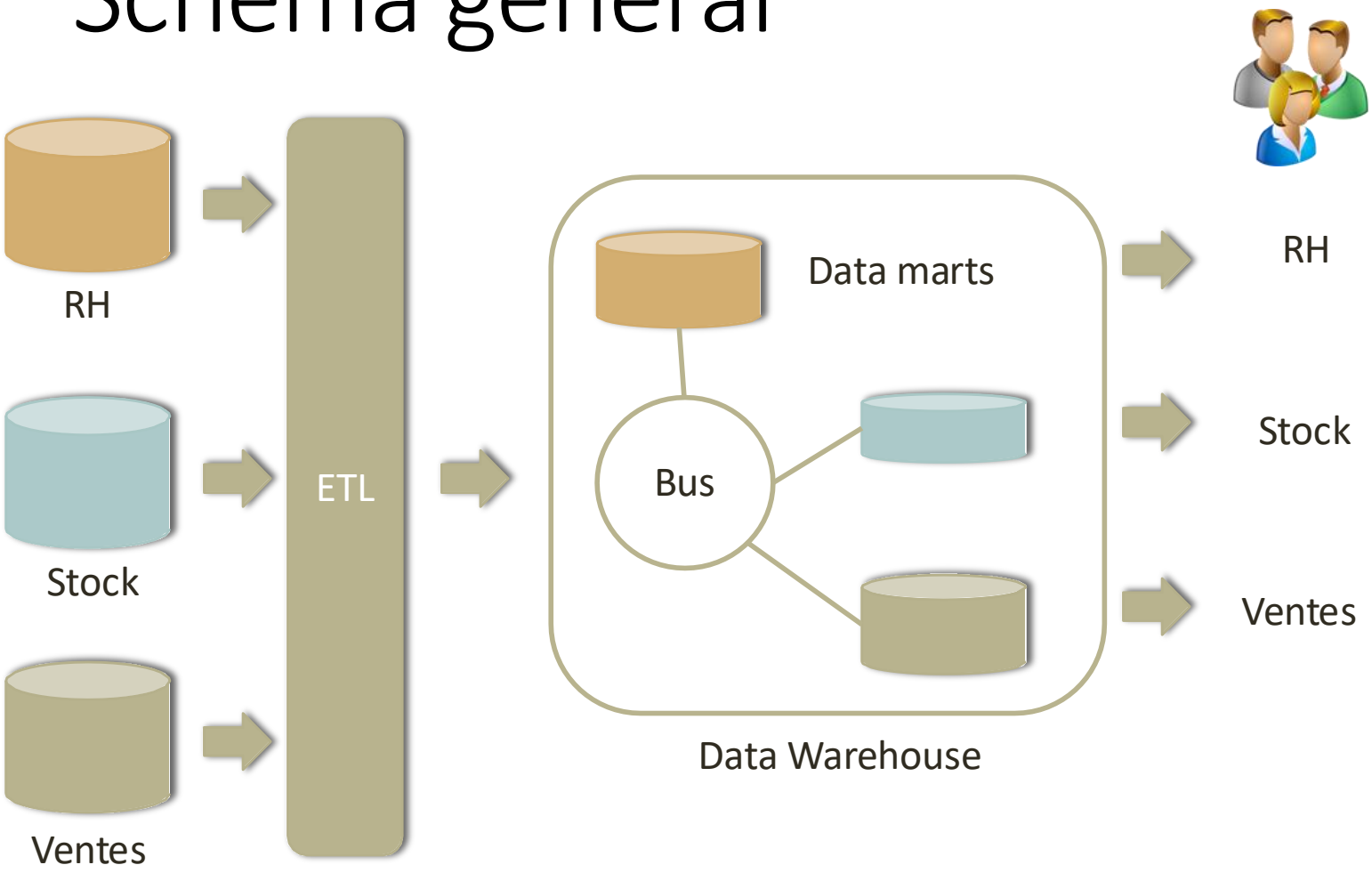


Approche Down--top (Kimball)



L'approche Bottom-up de Kimball consiste à construire d'abord des data marts orientés métier, puis à les intégrer progressivement pour former le Data Warehouse de l'entreprise.

Schéma général



Data Mart

- Un magasin de données (Data mart) est un sous--ensemble de l'entrepôt
- Il correspond à une classe de décideurs intéressés par le même thème
- Son volume réduit permet un accès plus rapide aux données
- Généralement le magasin est modélisé sous forme multidimensionnelle
- Les outils ETL peuvent être utilisés à ce niveau

Les phases

1. La conception (et la phase ETL)
2. La phase de structuration
3. La phase OLAP

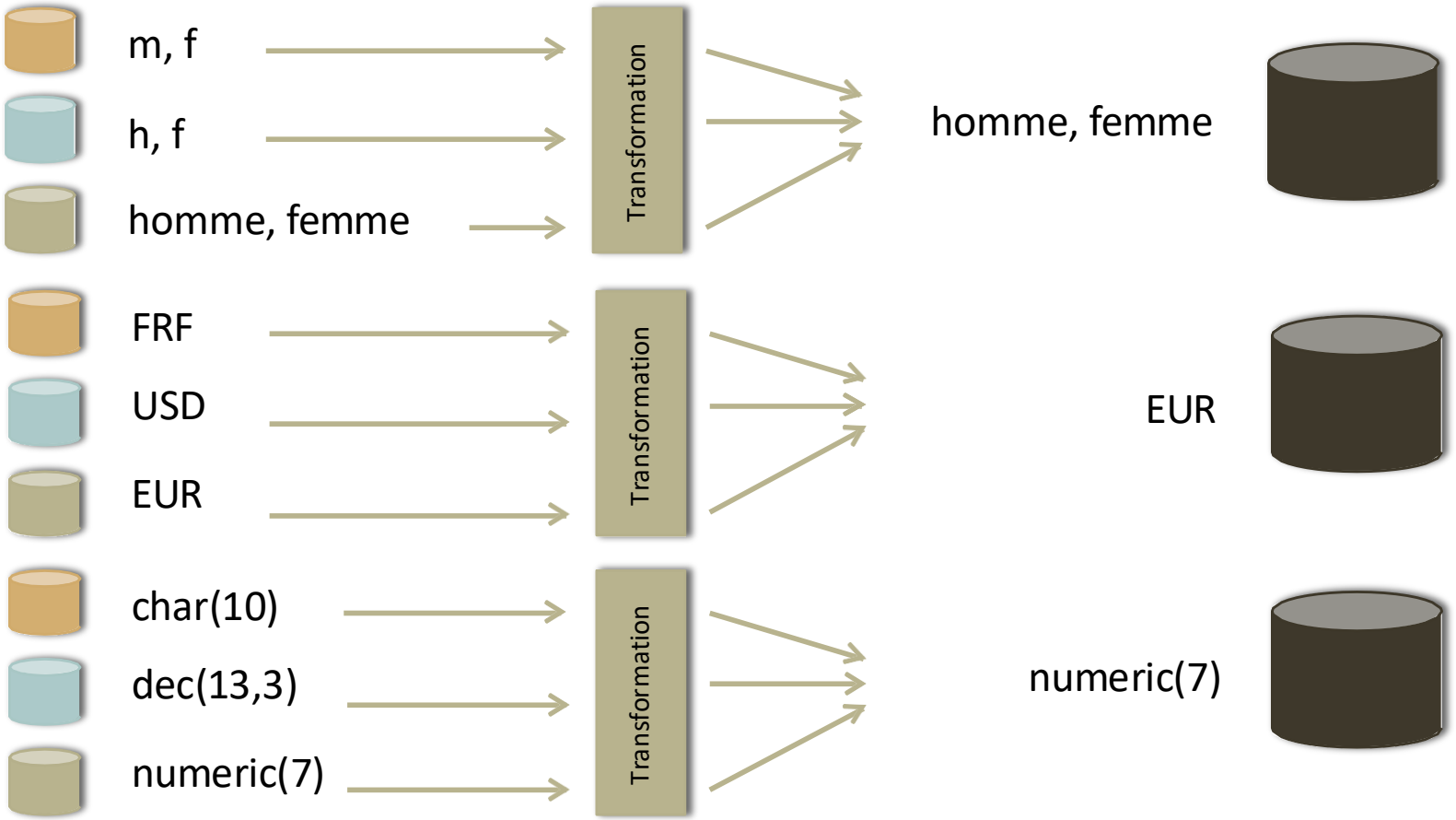
Conception

- Piloter quelle activité de l'entreprise
- Déterminer et recenser les données à utiliser
- Définir les aspects techniques de la réalisation
- Construire les modelés de données
- Mettre au point les démarches d'alimentation (ETL)
- Définir les stratégies d'administration
- Définir des espaces d'analyse
- Définir le mode de restitution
- …

ETL

- Alimentation du Data Warehouse et extraction des Data marts
- **E**xtract
  - Accès aux différentes sources
  - Selon des règles (déclencheurs) ou requêtes
  - Périodique
- **T**ransform
  - Unification des modèles (sources hétérogènes)
  - Gestion des inconsistances des données sources, élimination des doubles, etc.
- **L**oad
  - Chargement dans l'entrepôt ou dans les magasins
- Périodicité parfois longue

Ex. d'intégration des données



La structuration

1. Extraction des données

- Besoin d'outils spécifiques pour :
  - Accéder aux bases de production (requêtes sur des BD hétérogènes)
  - Améliorer la qualité des données : nettoyer, filtrer, …
  - Transformer les données : intégrer, homogénéiser
  - Dater systématiquement les données

La structuration

2. Référentiel
   - La métabase contient des métadonnées : des données sur les données de l'entrepôt de données
     - Quelles sont les données «entreposées», leur format, leur Signification, leur degré d'exactitude
     - Les processus de récupération/extraction dans les bases sources
     - La date du dernier chargement de l'entrepôt
     - L'historique des données sources et de celles de l'entrepôt

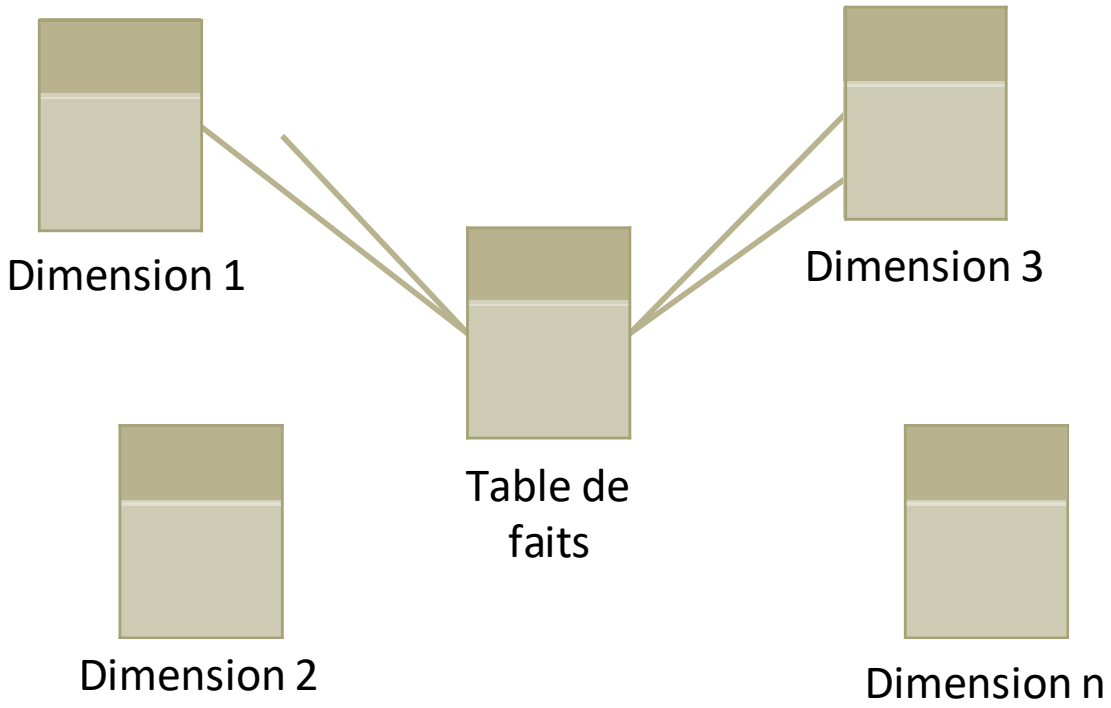
La structuration 3

2. Les modèles (ils sont détaillées ensuite)
   - Modèle en étoile
   - Modèle en flocon
   - Modèle en constellation

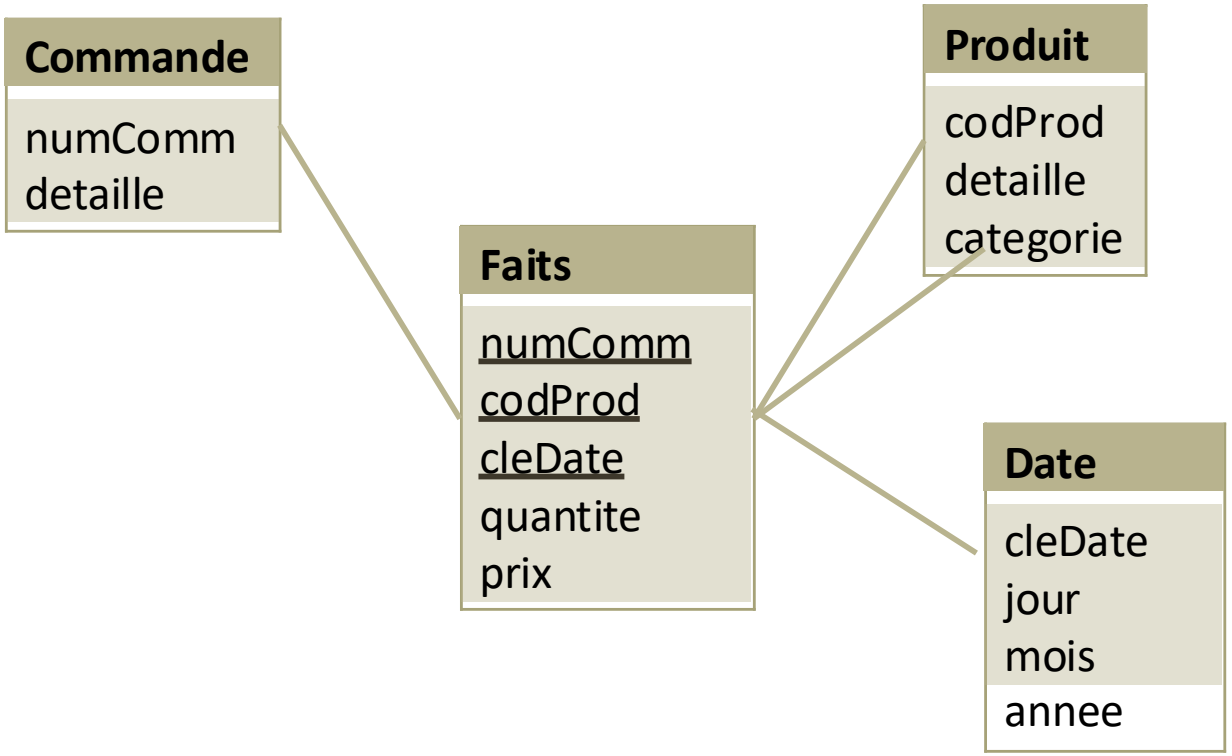
Modèle en étoile

- Une table de faits : identifiants des tables de dimension ; une ou plusieurs mesures
- Plusieurs tables de dimension : descripteurs des dimensions
- Une granularité définie par les identifiants dans la table des faits
- Avantages :
  - Facilité de navigation
  - Performances : nombre de jointures limité ; gestion des données creuses.
  - Gestion des agrégats
  - Fiabilité des résultats
- Inconvénients :
  - Toutes les dimensions ne concernent pas les mesures
  - Redondances dans les dimensions
  - Alimentation complexe

Schéma en étoile



Exemple de schéma en étoile



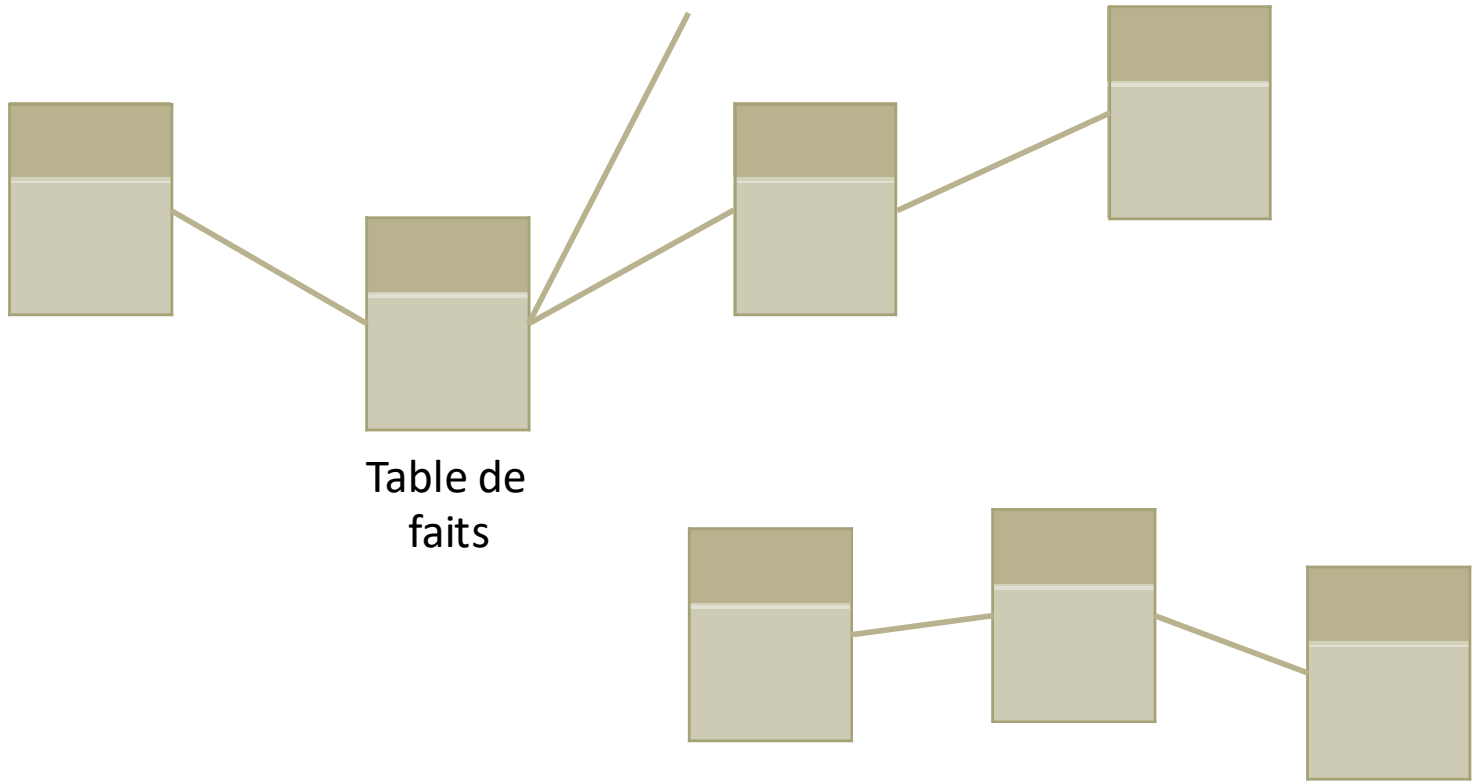
Modèle en flocon

- Le modèle doit être simple à comprendre : on peut augmenter sa lisibilité en regroupant certaines dimensions
- On définit ainsi des hiérarchies : celles-ci peuvent être géographiques ou organisationnelles

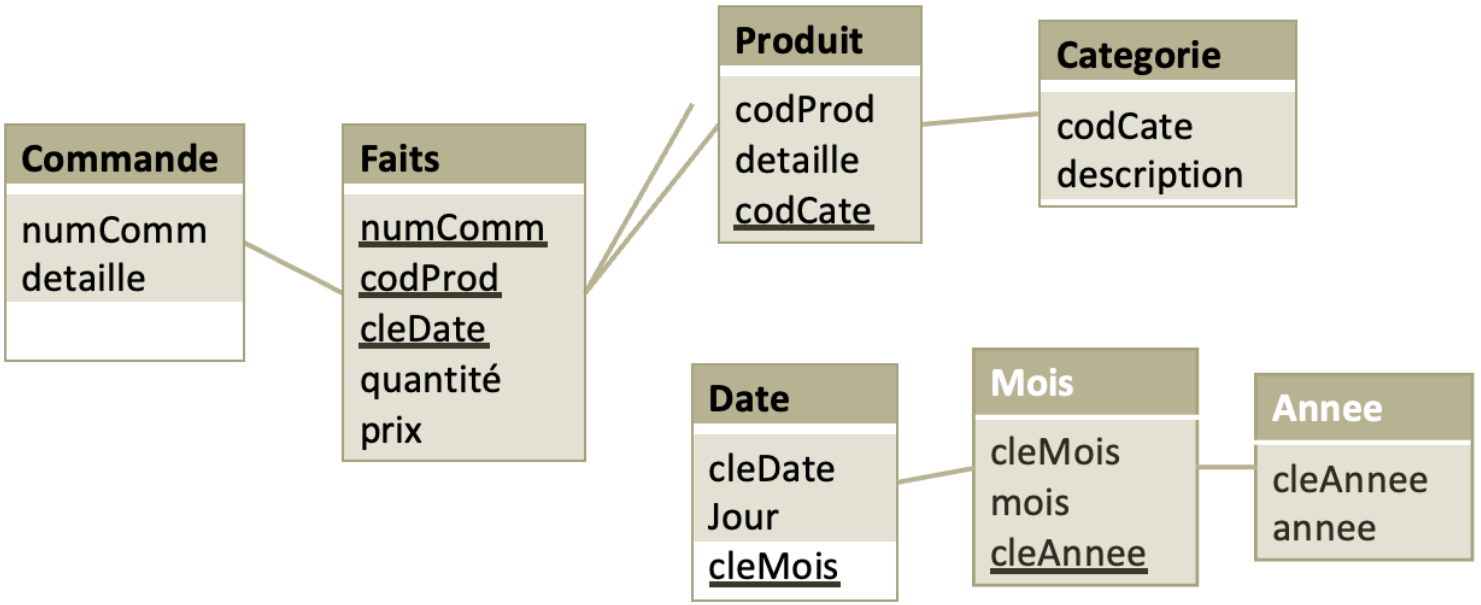
Mod. flocons de neige = Mod. étoile + normalisation des dimension

- Avantages :
  - Réduction du volume
  - Permettre des analyse par pallier (drill down) sur la dimension hiérarchisée
- Inconvénients :
  - Navigation difficile
  - Nombreuses jointures

Schéma en flocon



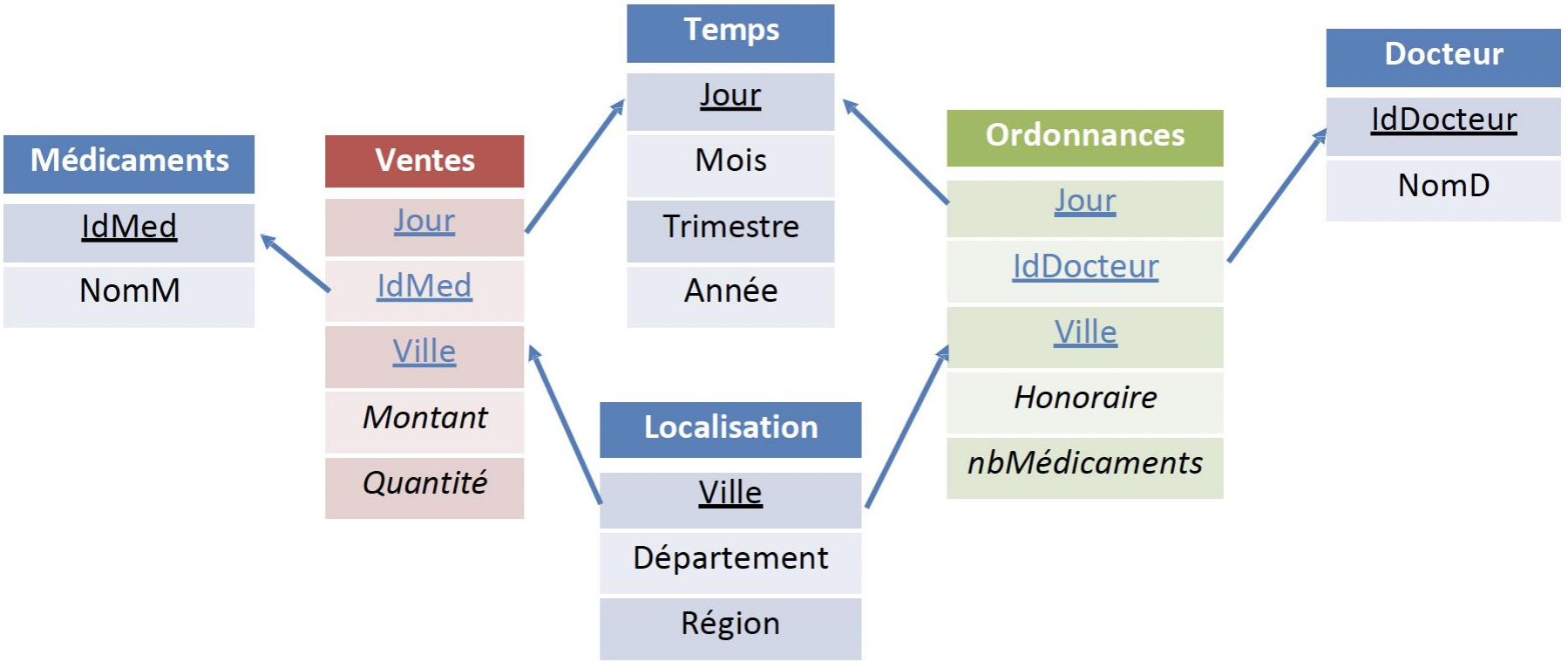
Exemple de schéma en flocon



Modèle en constellation

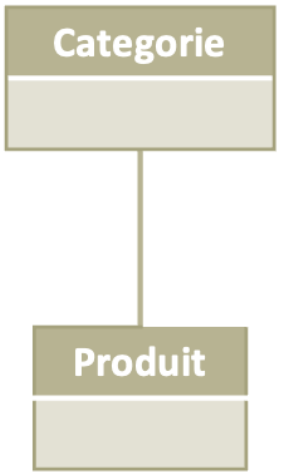
- La modélisation en constellation consiste à fusionner plusieurs modèles en étoile qui utilisent des dimensions communes.
- Un modèle en constellation comprend donc plusieurs tables de faits et des tables de dimensions communes ou non à ces tables de faits.

Modèle en constellation

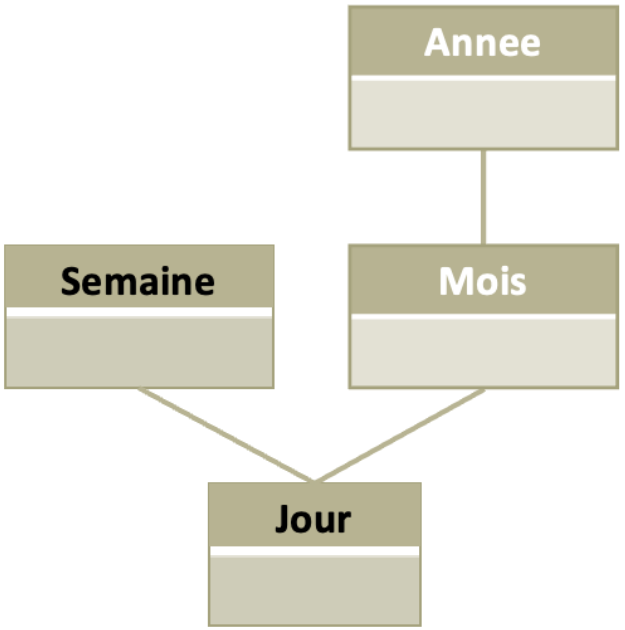


Concernant l’hiérarchie

Hiérarchie Simple



Hiérarchie Multiple



Plan de cours

1. Systèmes d'information décisionnelles
2. Entrepôt de données (Data Warehouse)
3. Analyse Multidimensionnelle
4. SQL / OLAP Extensions
5. BigQuery

L’analyse multidimensionnelle

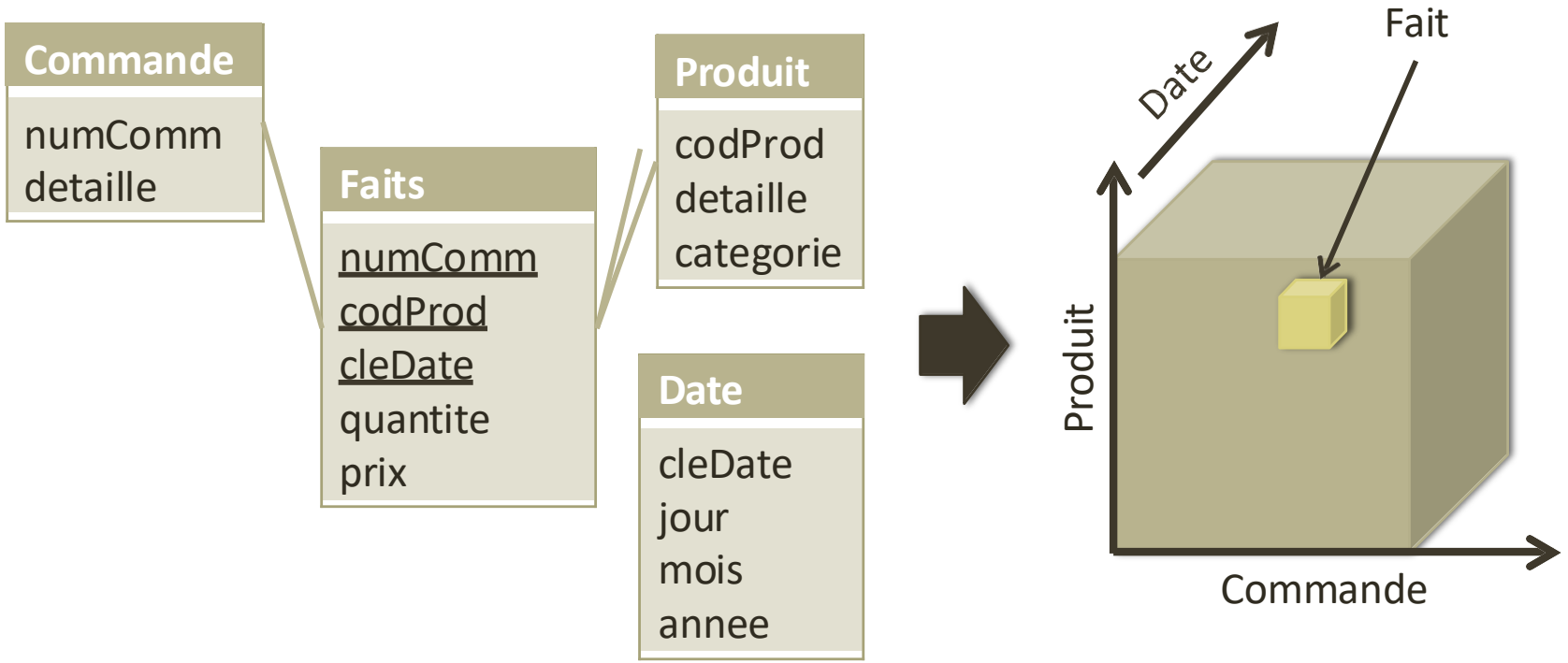
Exploitation d’un entrepôt de données

- Réalisation de rapports (reporting)
- Réalisation de tableaux de bords (dashboards)
- Analyse en ligne OLAP (OnLine Analytical Processing)
- Fouille de données(datamining)
- Visualisations de données
- Etc.

L’analyse multidimensionnelle

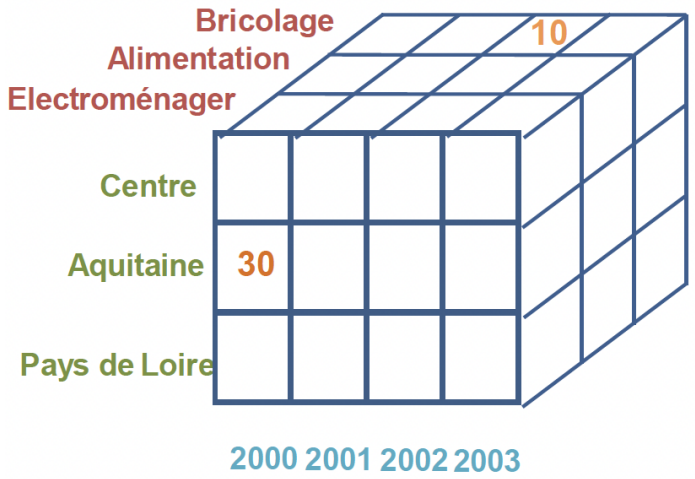
- Objectif : obtenir des informations déjà agrégées selon les besoins de l’utilisateur : simplicité et rapidité d’accès
- HyperCube OLAP : représentation de l’information dans un hypercube à N dimensions
- OLAP (On–Line Analytical Processing) : fonctionnalités qui servent à faciliter l’analyse multidimensionnelle : opérations réalisables sur l’hypercube

(Hyper)Cube de données 1



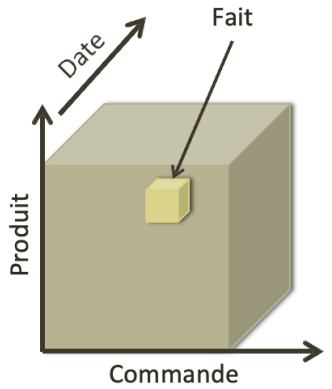
L’analyse multidimensionnelle

- En 2D sous la forme de tableaux croisés



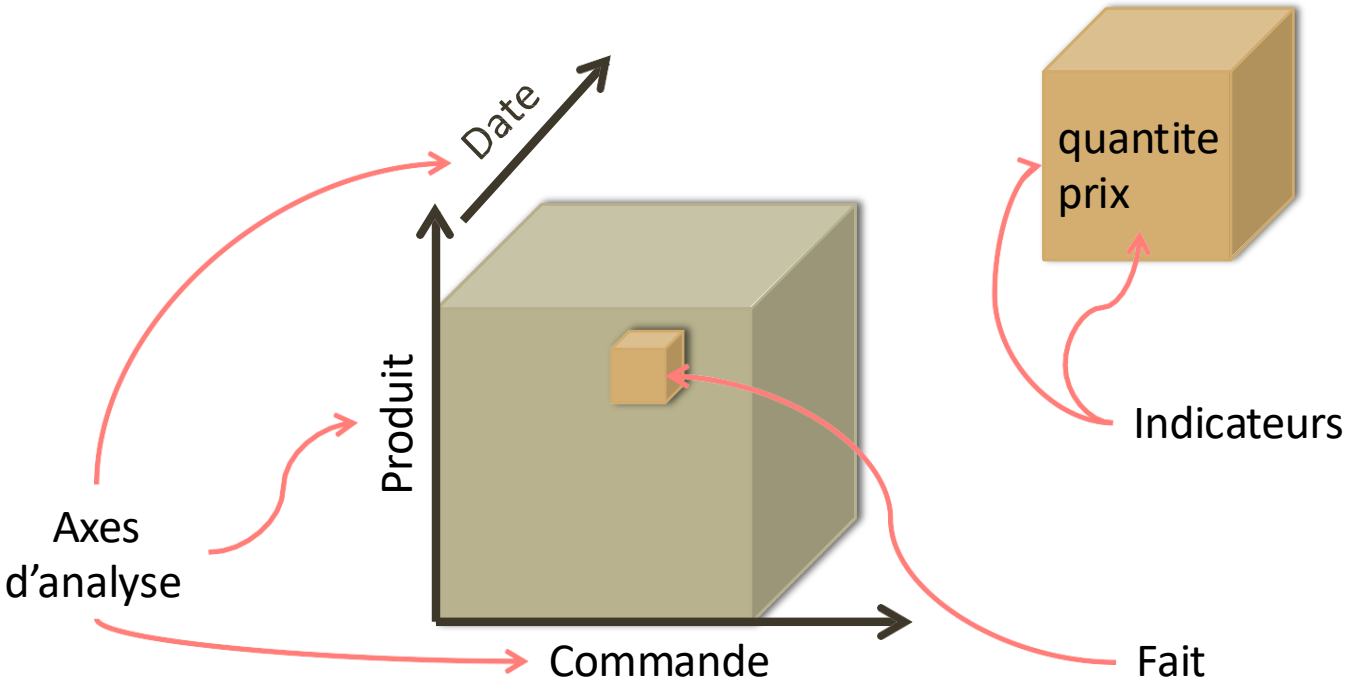
		2000	2001	2002
Centre	Bricolage	20	20	10
	Alimentation	20	20	20
	Electroménager	**30**	20	30
Aquitaine	Bricolage	20	20	20
	Alimentation	20	20	20
	Electroménager	30	20	20

Composantes d'un cube

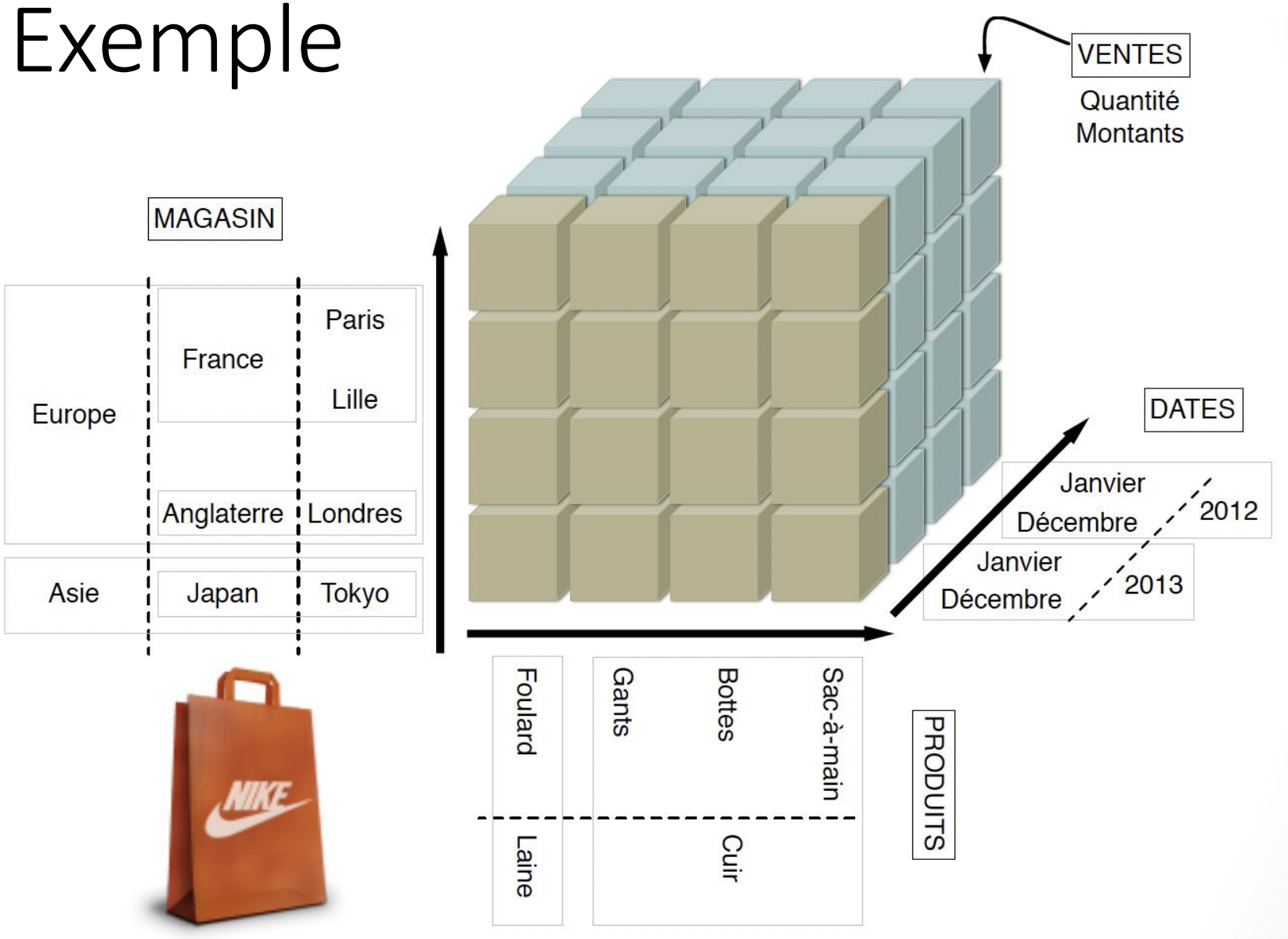


- Chaque **cellule** du cube correspond à une occurrence du fait
- Chaque cellule contient des **indicateurs** (variables, métriques ou mesures)
- Les axes d'analyse, également appelés **dimensions**, contiennent un ensemble de valeurs
- Des **hiérarchies** sont spécifiées sur les dimensions afin de permettre une consolidation des indicateurs
- Chaque indicateur a une **fonction d'agrégat** afin d'être exploité sur la hiérarchie

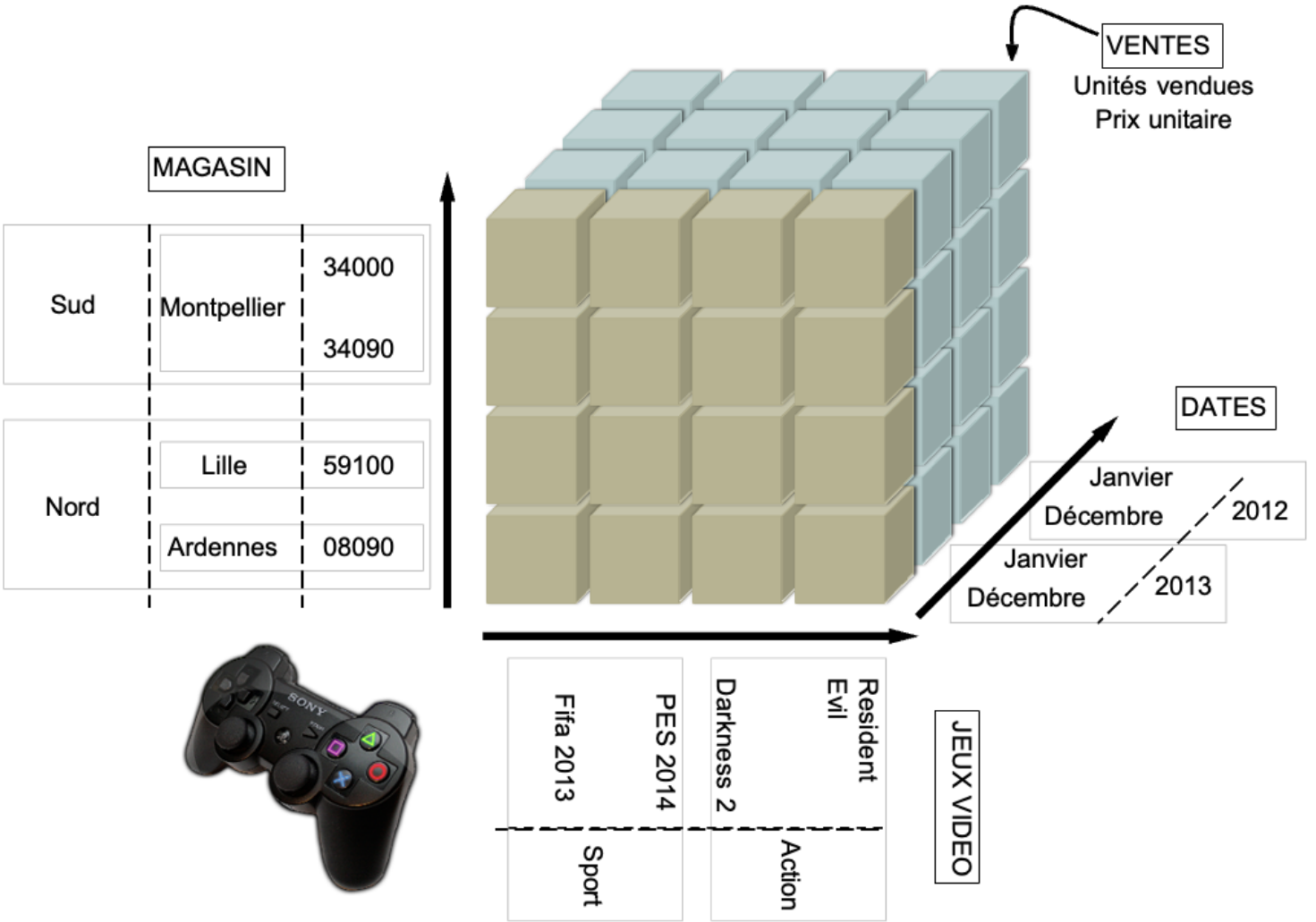
(Hyper)Cube de données



Exemple


VENTES
Quantité
Montants
MAGASIN
Europe
France
Paris
Lille
Anglaterre
Londres
Asie
Japan
Tokyo
DATES
Janvier
Décembre
2012
Janvier
Décembre
2013
PRODUITS
Foulard
Laine
Gants
Bottes
Cuir
Sac-à-main
NIKE

Exemple


VENTES
Unités vendues
Prix unitaire
MAGASIN
Sud
Montpellier
34000
34090
Nord
Lille
59100
Ardennes
08090
DATES
Janvier
Décembre
2012
Janvier
Décembre
2013
JEUX VIDEO
Fifa 2013
PES 2014
Sport
Darkness 2
Resident Evil
Action

L'implémentation du OLAP

- MOLAP (Multidimensional OLAP)
  - Le cube est stocké sous forme propriétaire par un SGBD multidimensionnel dans une matrice
  - On trouve en colonne tous les axes, puis tous les indicateurs
  - Chaque cellule du cube est stockée par une ligne dans la matrice
  - Avantages : adapté aux analyses multidimensionnelles
  - Inconvénients : difficulté de mise en oeuvre (systèmes propriétaires), problème d'éparsité des cubes, etc.

L'implémentation du OLAP

- ROLAP (Relational OLAP)
  - Le stockage peut s'effectuer sur un SGBD relationnel classique
  - Le cube est stocké selon le modèle en étoile (flocon ou constellation)
  - Avantages : faible coût de mise en oeuvre
  - Inconvénients : performance (pour le calcul des jointures & agrégats)

L'implémentation du OLAP

- HOLAP (Hybride OLAP)
  - ROLAP + MOLAP
  - Pour tirer profit des avantages des technologies ROLAP et MOLAP:

    Un système ROLAP pour stocker ,gérer les données détaillées ET

    Un système MOLAP pour stocker, gérer les données agrégées

Operateurs algébriques OLAP

- Modèle relationnel : projection, jointure, restriction, union, division, intersection, etc.
- Modèle OLAP : drill--up, drill--down, slice, dice, pivot, switch, etc.

Catégories d'opérations OLAP

- **Restructuration** : opérations liées à la structure, manipulation et visualisation du cube :
  - **Rotate/pivot :** effectuer à un cube une rotation autour d'un de ses trois axes passant par le centre de 2 faces opposées, de façon à présenter un ensemble de faces différents
  - **Switch :** consiste à inter--changer la position des membres d'une dimension
  - **Split :** consiste à présenter chaque tranche du cube et de passer d'une présentation tridimensionnelle d'un cube à sa présentation sous la forme d'un ensemble de tables
  - **Nest :** imbrication des membres à partir du cube
  - **Push** : combiner les membres d'une dimension aux mesures du cube

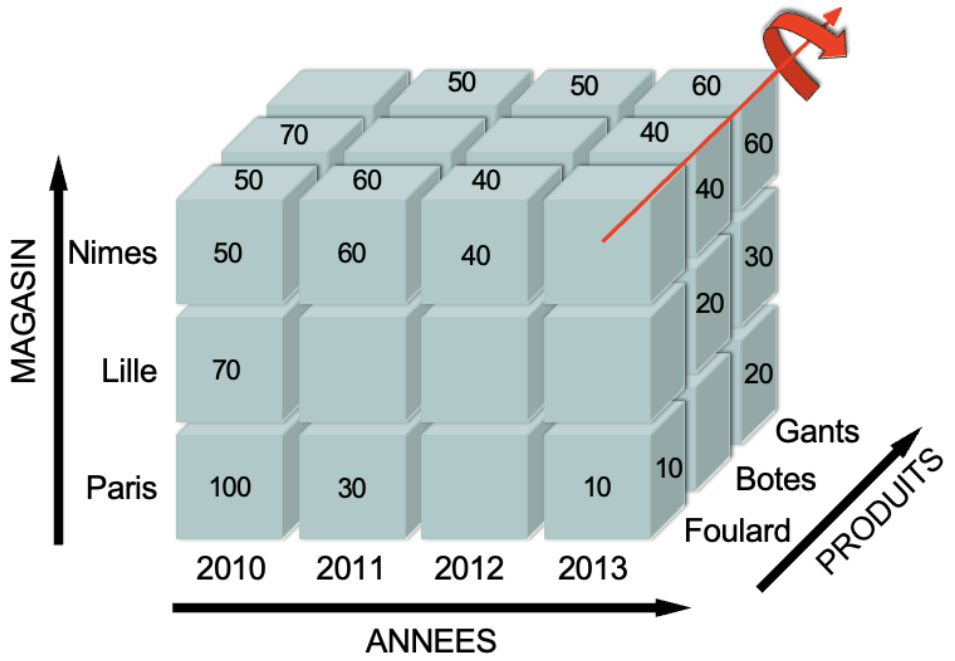
Catégories d'opérations OLAP

- **Granularité** : concerne un changement de niveau de détail : opérations liées au niveau de granularité des données :
  - **Roll-up** : consiste à représenter les données du cube à un niveau de granularité supérieur conformément à la hiérarchie définie sur la dimension. Une fonction d'agrégation (somme, moyenne, etc.) en paramètre de l'opération indique comment sont calculés les valeurs du niveau supérieur à partir de celles du niveau inférieur
  - **Drill-down** : consiste à représenter les données du cube à un niveau de granularité de niveau inférieur, donc sous une forme plus détaillée (selon la hiérarchie définie de la dimension)

Catégories d'opérations OLAP 3

- **Ensembliste** : concerne l'extraction et l'OLTP classique :
  - **Slice** : correspond à une projection selon une dimension du cube
  - **Dice** : correspond à une sélection du cube

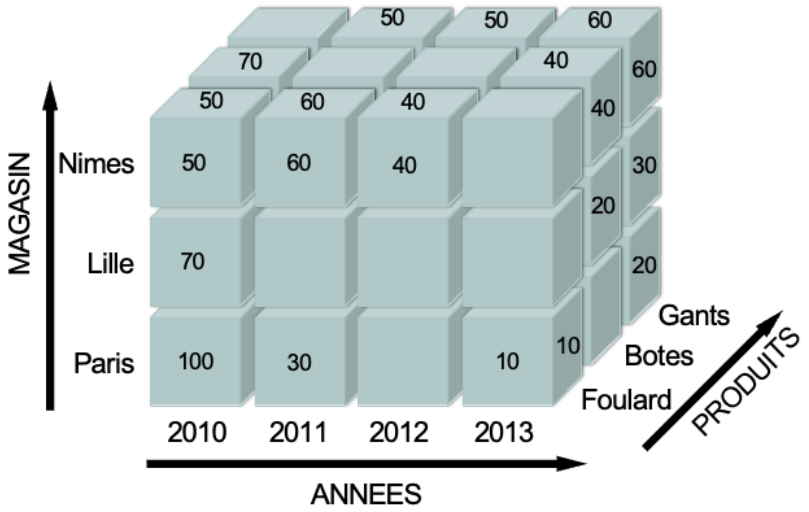
Rotate/pivot

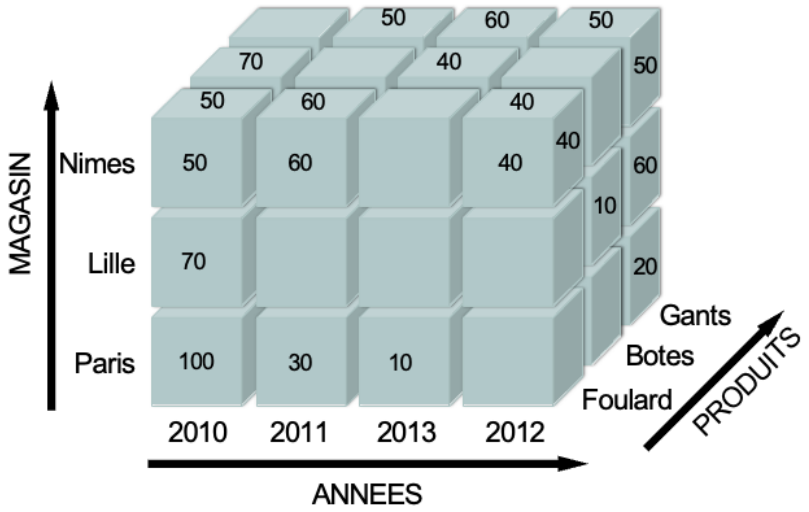


2013	Foulard	Botes	Gants
Nimes		40	60
Lille		20	30
Paris	10		20

Foulard	2010	2011	2012	2013
Nimes	50	60	40	
Lille	70			
Paris	100	30		10

Switch

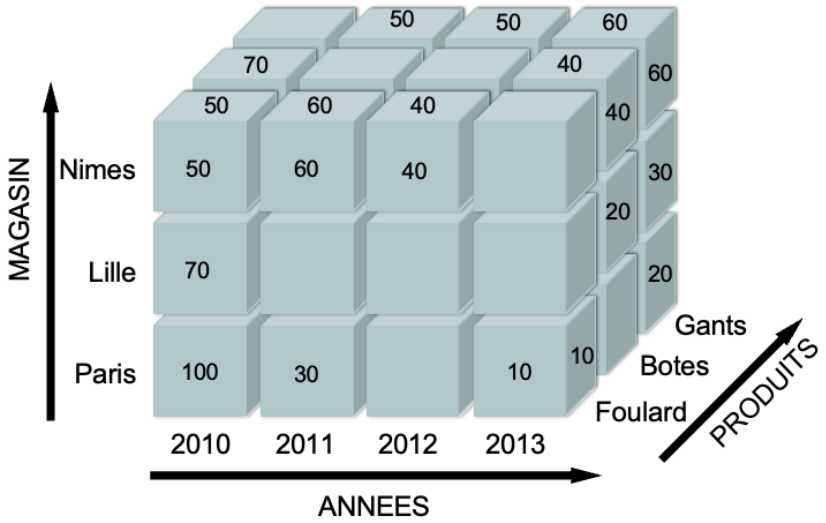




2013	Foulard	Botes	Gants
Nimes		40	60
Lille		20	30
Paris	10		20

2012	Foulard	Botes	Gants
Nimes	40		50
Lille		10	60
Paris			20

Split

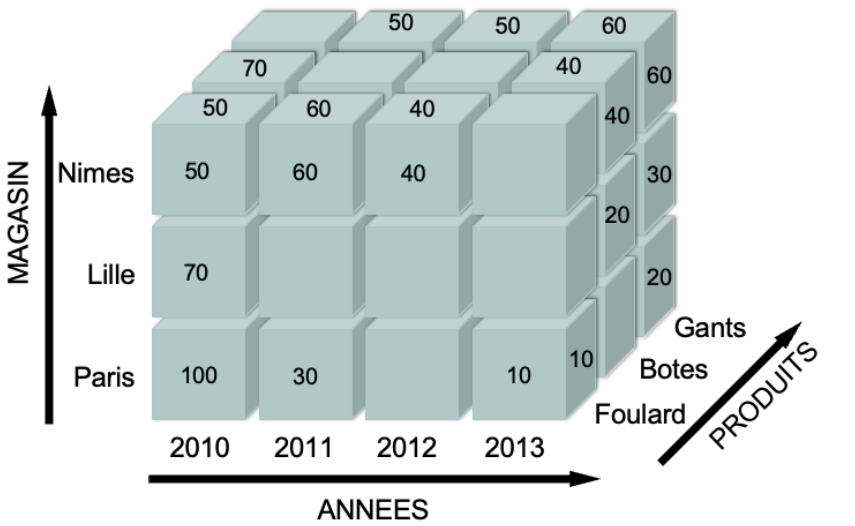


Foulard	2010	2011	2012	2013
Nimes	50	60	40	
Lille	70			
Paris	100	30		10

Botes	2010	2011	2012	2013
Nimes	70			40
Lille	20			20
Paris	30		30	

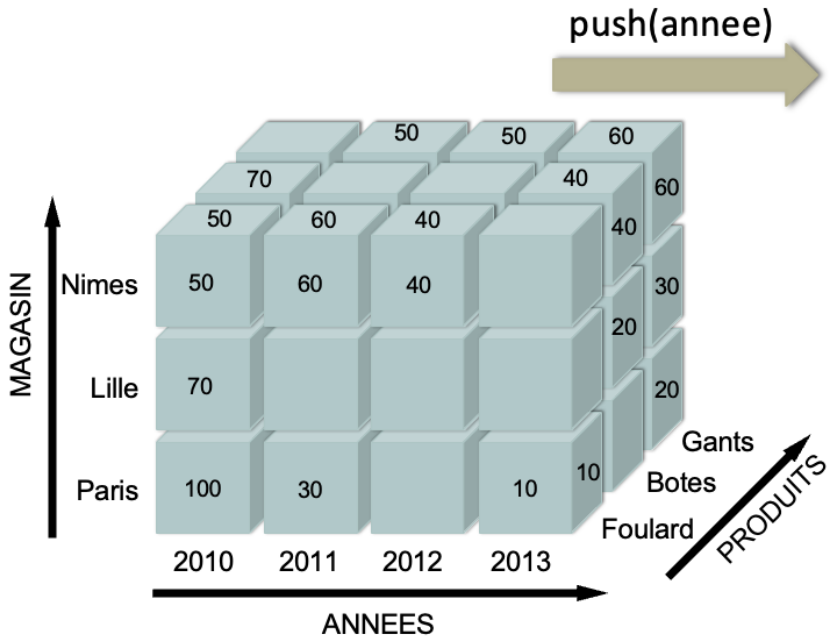
Gants	2010	2011	2012	2013
Nimes		50	50	60
Lille	70	10		30
Paris		30		20

Nest



		Nimes	Lille	Paris
Foulard	2010	50	70	100
	2011	60		30
	2012	40		
	2013			10
Botes	2010	70		
	2011		20	50
	2012			
	2013	40	20	
Gants	2010		20	
	2011	50		10
	2012	50		
	2013	60	30	20

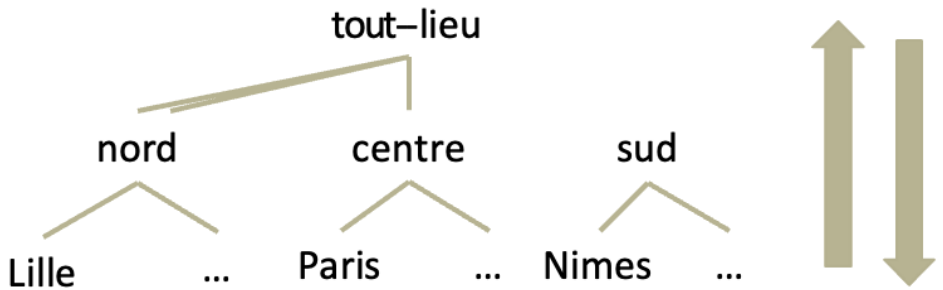
Push



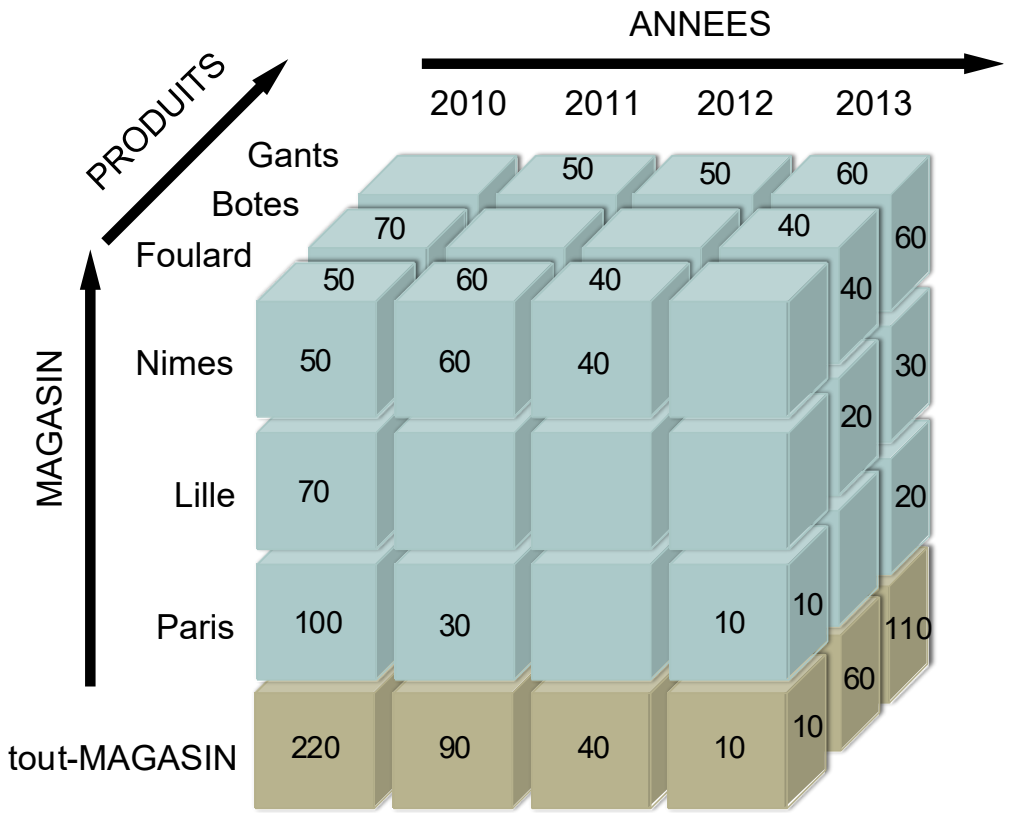
	Foulard	Botes	Gants
Nimes	2010 50 2011 60 2012 40	2010 70 2013 40	 2011 50 2012 50 2013 60
Lille	2010 70	2010 10 2012 50 2013 20	 2011 20 2013 30
Paris	2010 100 2011 30 2013 10	 2012 40	 2011 50 2013 20

Roll--up et Drill--down

- Roll--up : représente les données du cube à un niveau de granularité supérieur conformément à la hiérarchie définie sur la dimension
- Drill--down : représente les données du cube à un niveau de granularité de niveau inférieur, donc sous une forme plus détaillée



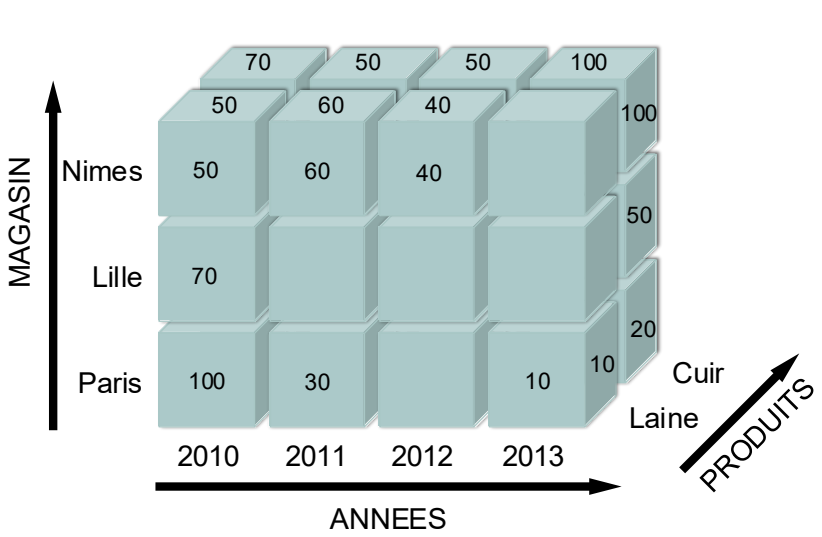
Exemples de Roll--up

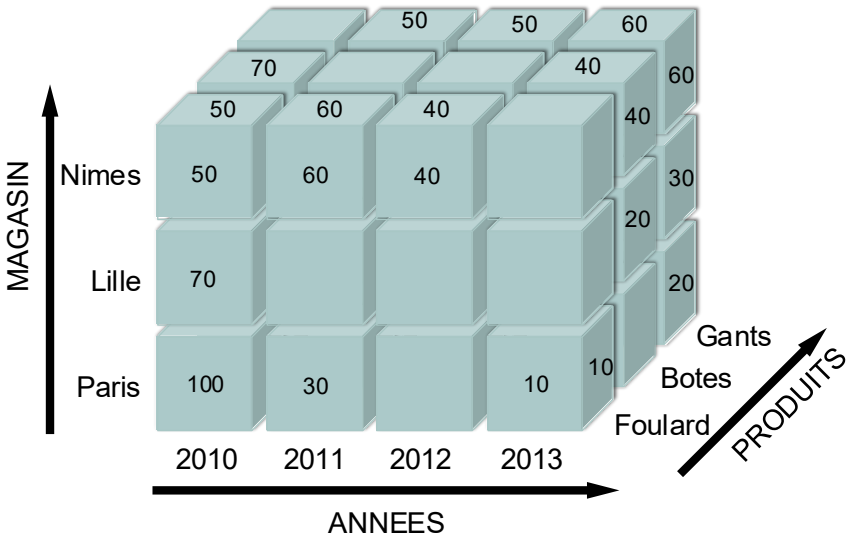


tout–ANNEES

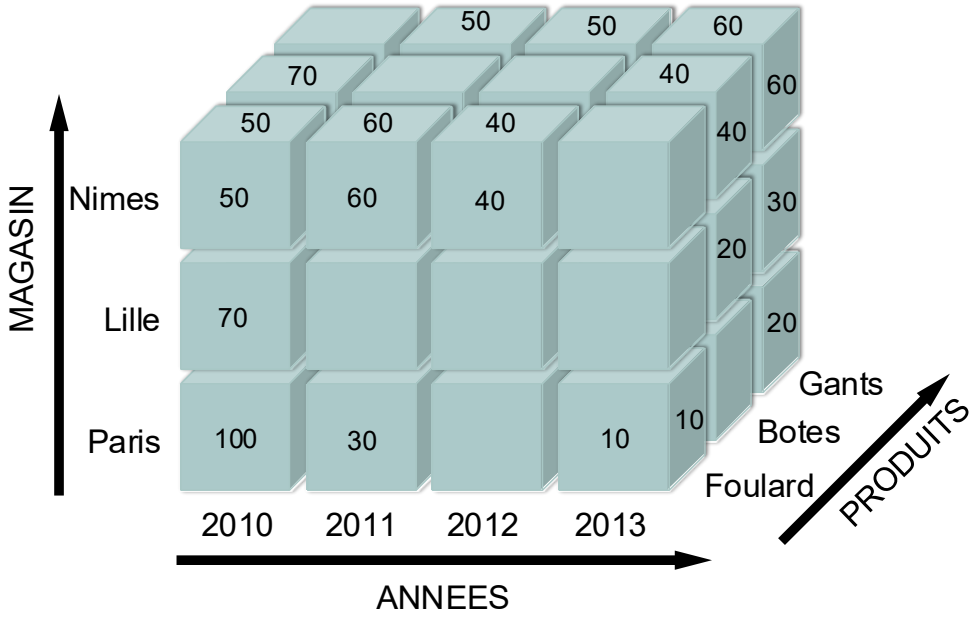
	Foulard	Botes	Gants
Nimes	150	110	160
Lille	70	80	90
Paris	140	20	80

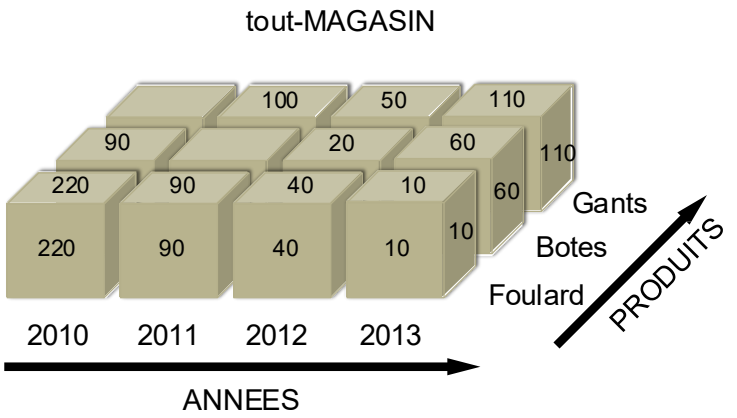
Exemple de Drill--down





Slice (projection)





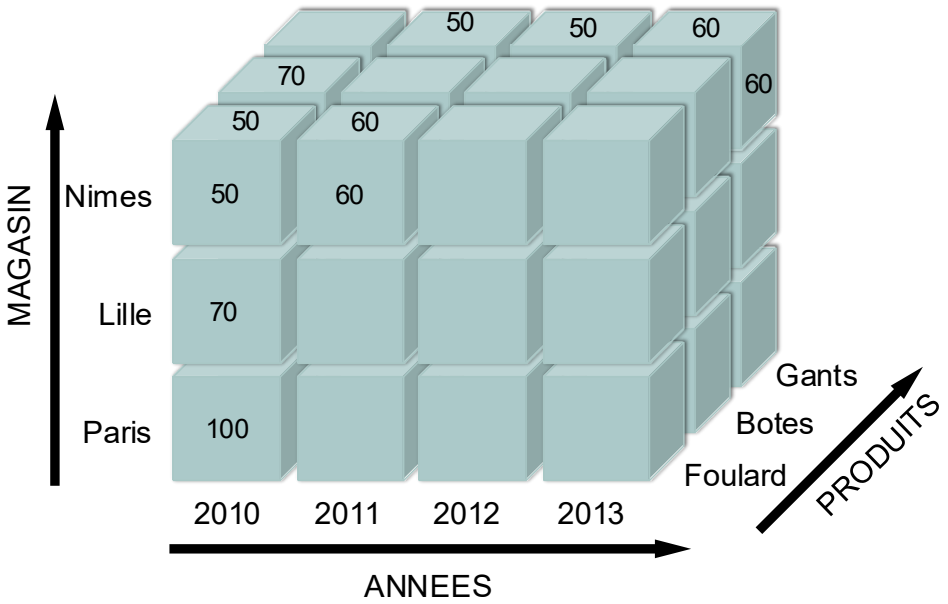
$\pi_{annees,produits}$

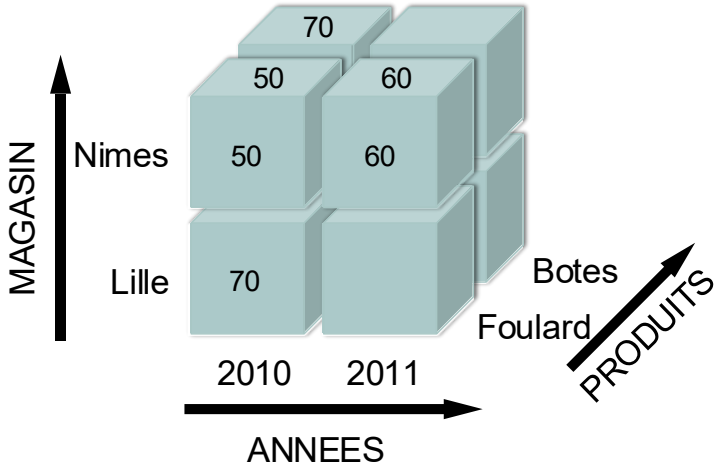
	2010	2011	2012	2013
Foulard	220	90	40	10
Botes	90		20	60
Gants		100	50	110

Dice (selection)

(MAGASIN = Nimes OR MAGASIN = Lille) AND
(ANNEE = 2010 OR ANNEE = 2011) AND
(PRODUITS = Foulard OR PRODUITS = Botes)

ventes ≥ 50





Langages pour OLAP

- SQL étendu (Extensions de SQL--3 / SQL--99 pour OLAP) :
  - Nouvelles fonctions SQL d'agrégation: Rank, N_tile
  - Nouvelles fonctions de la clause GROUP BY :
    - ROLLUP
    - CUBE
    - GROUPING SETS (multiple GROUP Bys)
  - Fenêtre glissante :
    - WINDOWS/OVER/PARTITION
- MDX (Multi Dimensional eXpression) :
  - Langage de requêtes OLAP
  - Proposé par Microsoft (1997)

Edgar F. Codd (1993) : définition des bases du modèle OLAP

12 règles de Codd définissant l'évaluation des produits OLAP :

1. Vue multidimensionnelle des données dans une base OLAP
2. Transparence : éléments techniques invisibles pour l'utilisateur
3. Accessibilité : complexité et l'hétérogénéité des données masquées par les outils OLAP
4. Stabilité : performances stables indépendamment du contexte d'analyse
5. Architecture client/serveur : le serveur homogénéise les 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

Quelques solutions commerciales OLAP

Solutions ROLAP (basées sur bases relationnelles)

- IBM Db2 Warehouse : ROLAP (analyse sur entrepôts de données relationnels)
- Oracle Autonomous Data Warehouse + Oracle Analytics : ROLAP
- Microsoft Azure Synapse Analytics : ROLAP cloud moderne
- Snowflake : ROLAP cloud (très utilisé en BI moderne)
- Google BigQuery : ROLAP cloud pour analyses massives

Quelques solutions commerciales OLAP

Solutions MOLAP (multidimensionnelles)

- Oracle Essbase : MOLAP (toujours une référence en cubes multidimensionnels)
- IBM Planning Analytics (TM1) : MOLAP pour finance et planification
- SAS Viya OLAP / SAS Analytics : MOLAP avancé

Solutions HOLAP (hybrides)

- Microsoft SQL Server Analysis Services (SSAS) : HOLAP / MOLAP / ROLAP
- SAP BW/4HANA : HOLAP (hybride mémoire + relationnel)
- MicroStrategy : HOLAP avec moteur hybride
- Oracle Analytics Cloud : hybride MOLAP/ROLAP

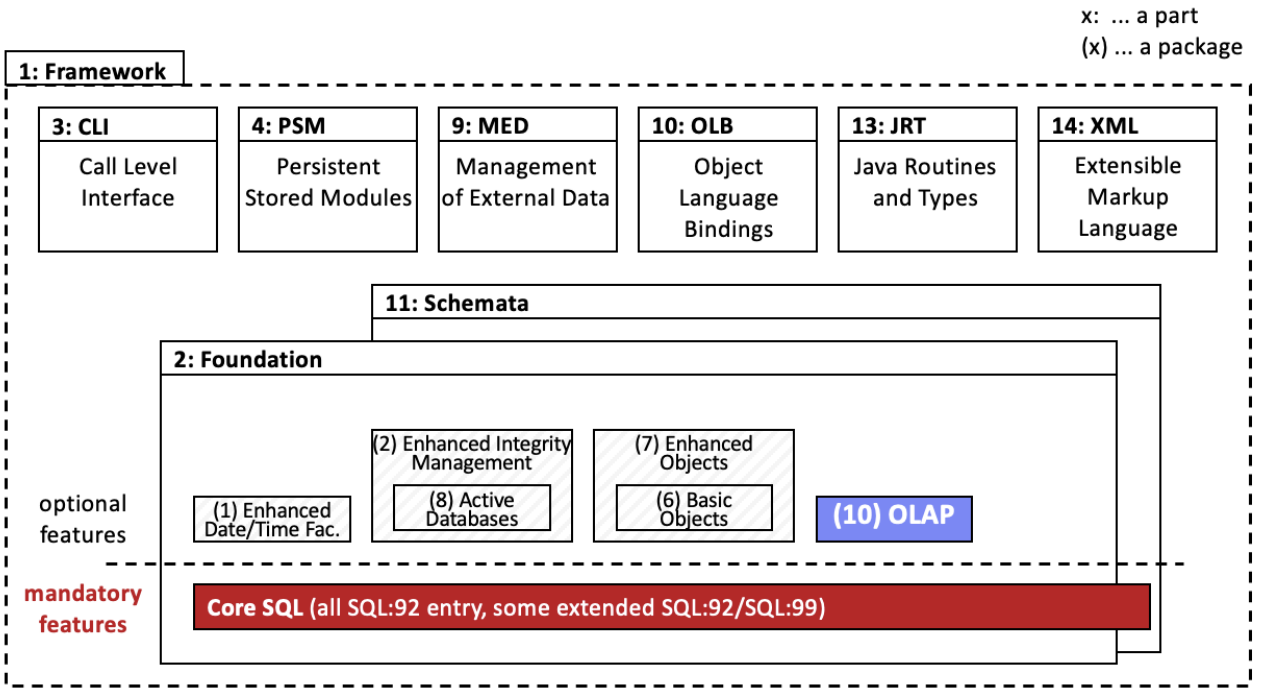
Aujourd'hui, l'OLAP est majoritairement : dans le cloud (Snowflake, BigQuery, Azure Synapse), hybride (HOLAP) et intégré aux outils BI modernes (Power BI, Tableau, Looker)

TD : Conception conceptuelle d’un Data Warehouse

Plan de cours

1. Systèmes d'information décisionnelles
2. Entrepôt de données (Data Warehouse)
3. Analyse Multidimensionnelle
4. SQL / OLAP Extensions
5. BigQuery

Standard SQL (ANSI / ISO / IEC)



- SQL/OLAP (Part 10)
- Le bloc (10) OLAP fait partie officiellement du standard SQL
- Il introduit : les fonctions analytiques, les fenêtres (OVER, PARTITION BY, ORDER BY)

  Ces fonctionnalités sont essentielles pour l'analyse décisionnelle

  BigQuery, PostgreSQL, Oracle, SQL Server implémentent largement cette partie

L'OLAP n'est pas un langage à part :
c'est une extension normalisée du SQL, intégrée officiellement depuis longtemps,
et exploitée massivement dans les entrepôts de données modernes comme BigQuery

Multi-groupements (SQL / OLAP)

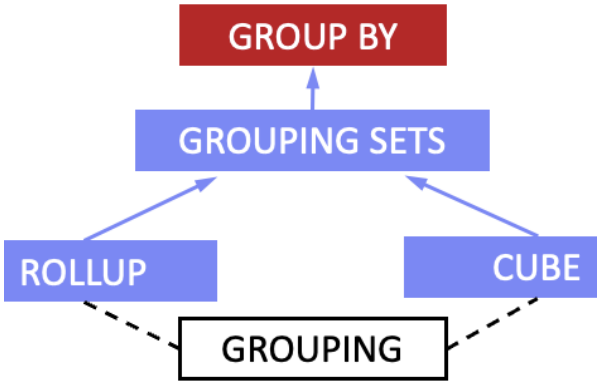
GROUP BY (SQL classique) permet de :

- regrouper des tuples selon une ou plusieurs variables catégorielles
- calculer des agrégats par groupe (SUM, COUNT, AVG, …)
- Exemple illustré :
- données de ventes par année et trimestre
- requête :

Year	Quarter	Revenue
2004	1	10
2004	2	20
2004	3	10
2004	4	20
2005	1	30

```
SELECT Year, SUM(Revenue)
  FROM Sales
  GROUP BY Year
```

Year	SUM
2004	60
2005	30



Problème en analytique décisionnelle

- En OLAP, on veut souvent répondre à plusieurs niveaux d'analyse en une seule requête, par exemple :
- chiffre d'affaires : par année, par année + trimester, total global
- sans écrire plusieurs requêtes SQL

extensions de groupement du standard SQL.

Multi-groupements : GROUPING SETS

Sémantique de GROUPING SETS

```
GROUP BY GROUPING SETS
  ((<attribute-list>), …)
```

- GROUPING SETS permet de définir plusieurs regroupements dans une seule requête SQL
- Chaque grouping set est une liste d'attributs de regroupement
- La fonction d'agrégation est la même pour tous les regroupements (SUM, COUNT, etc.)
- GROUPING SETS ≡ plusieurs GROUP BY concaténés par UNION ALL

Year	Quarter	Revenue
2004	1	10
2004	2	20
2004	3	10
2004	4	20
2005	1	30

```
SELECT Year, Quarter, SUM(Revenue)
  FROM R
  GROUP BY GROUPING SETS
    ((), (Year), (Year,Quarter))
```


`()`
→ **total global** (aucun attribut de regroupement)
`(Year)`
→ agrégation par **année**
`(Year, Quarter)`
→ agrégation par **année et trimestre**

Year	Quarter	SUM
-	-	90
2004	-	60
2005	-	30
2004	1	10
2004	2	20
2004	3	10
2004	4	20
2005	1	30

Multi-groupements : GROUPING SETS

Sémantique de ROLLUP

- ROLLUP permet de réaliser un regroupement hiérarchique
- Il s'appuie sur une hiérarchie naturelle de dimension
  - Temps : Jour → Mois → Trimestre → Année
  - Géographie : Ville → Département → Région → Pays
- Il calcule automatiquement :
  - le détail
  - les sous-totaux
  - le total global

ROLLUP est une opération OLAP de type roll-up (agrégation ascendante)

```
GROUP BY ROLLUP
  (<attribute-list>)
```

Year	Quarter	Revenue
2004	1	10
2004	2	20
2004	3	10
2004	4	20
2005	1	30

```
SELECT Year, Quarter, SUM(Revenue)
  FROM R
  GROUP BY ROLLUP(Year,Quarter)
```

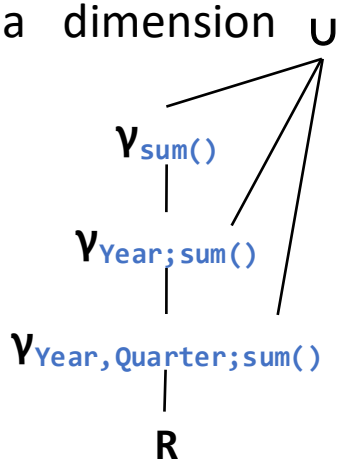
Year	Quarter	SUM
-	-	90
2004	-	60
2005	-	30
2004	1	10
2004	2	20
2004	3	10
2004	4	20
2005	1	30

Multi-groupements : Rollup, cont. et Groupement

Implémentation par l'opérateur

- ROLLUP est implémenté par le moteur SQL comme une suite de tours d'agrégation
- À chaque tour, le moteur : réduit le niveau de detail, recalcule les fonctions d'agrégation
- Cela fonctionne pour :
  - mesures additives (SUM, COUNT)
  - mesures semi-additives (avec précautions sur la dimension temps)
  - Exemple :

```
SELECT Year, Quarter,
SUM(Revenue)
   FROM R
   GROUP BY ROLLUP(Year,Quarter)
```



- ROLLUP n'est pas une vue matérialisée, mais une évaluation **logique** par le moteur SQL.

Multi-groupements : Rollup, cont. et Groupement

Sémantique du GROUPEMENT

- Avec ROLLUP (ou CUBE), on obtient des lignes contenant des valeurs NULL.
  - Mais tous les NULL n'ont pas la même signification :
  - NULL présent dans les données
  - NULL introduit par l'agrégation (total / sous-total)
  - Sans information supplémentaire, on ne peut pas les distinguer.
- GROUPING(attribut) permet de savoir pourquoi un attribut vaut NULL
- Valeurs possibles :
  - 0 → valeur réelle (donnée d'origine)
  - 1 → valeur générée par une agrégation OLAP
  - C'est un test sur le niveau d'agrégation.

```
SELECT Team, SUM(Revenue),
    GROUPING(Team) AS Agg
  FROM R
  GROUP BY ROLLUP (Team)
```

Team	Revenue	Agg
NULL	10	0
Sales	40	0
Tech	20	0
NULL	70	1

Multi-groupements : Cube

```
GROUP BY CUBE(<attribute-list>)
```

- **Sémantique de CUBE**
  - CUBE calcule les agrégats pour toutes les combinaisons possibles des attributs de regroupement
  - Pour n attributs, il produit 2^n groupings
  - C'est l'opérateur SQL le plus général des multi-groupings
- CUBE correspond à la construction complète d'un cube OLAP

Year	Quarter	Revenue
2004	1	10
2004	2	20
2004	3	10
2004	4	20
2005	1	30

```
SELECT Year, Quarter,
       SUM(Revenue)
  FROM R
  GROUP BY CUBE(Year,Quarter)
```

Year	Quarter	SUM
-	-	90
2004	-	60
2005	-	30
-	1	40
-	2	20
-	3	10
-	4	20
2004	1	10
2004	2	20
2004	3	10
2004	4	20
2005	1	30

Plan de cours

1. Systèmes d’information décisionnelles
2. Entrepôt de données (Data Warehouse)
3. Analyse Multidimensionnelle
4. SQL / OLAP Extensions
5. BigQuery

BigQuery

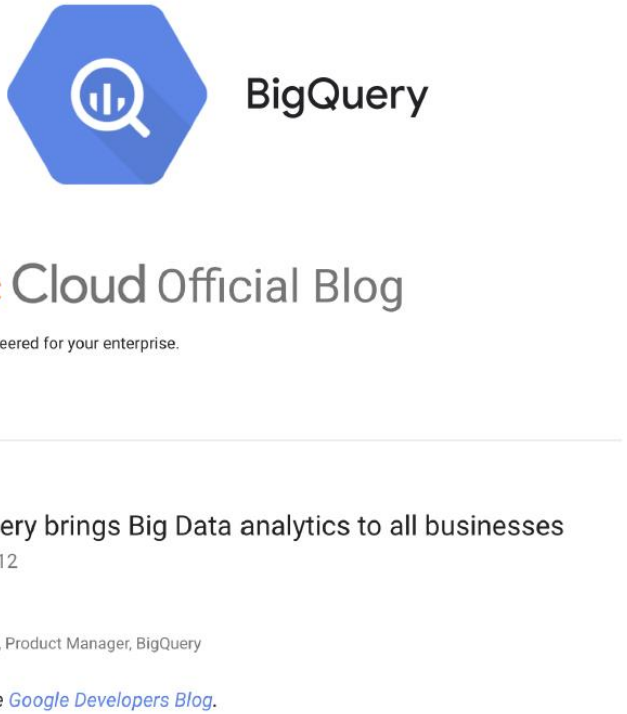
Intro BigGuery

Qu’est-ce que BigQuery ?

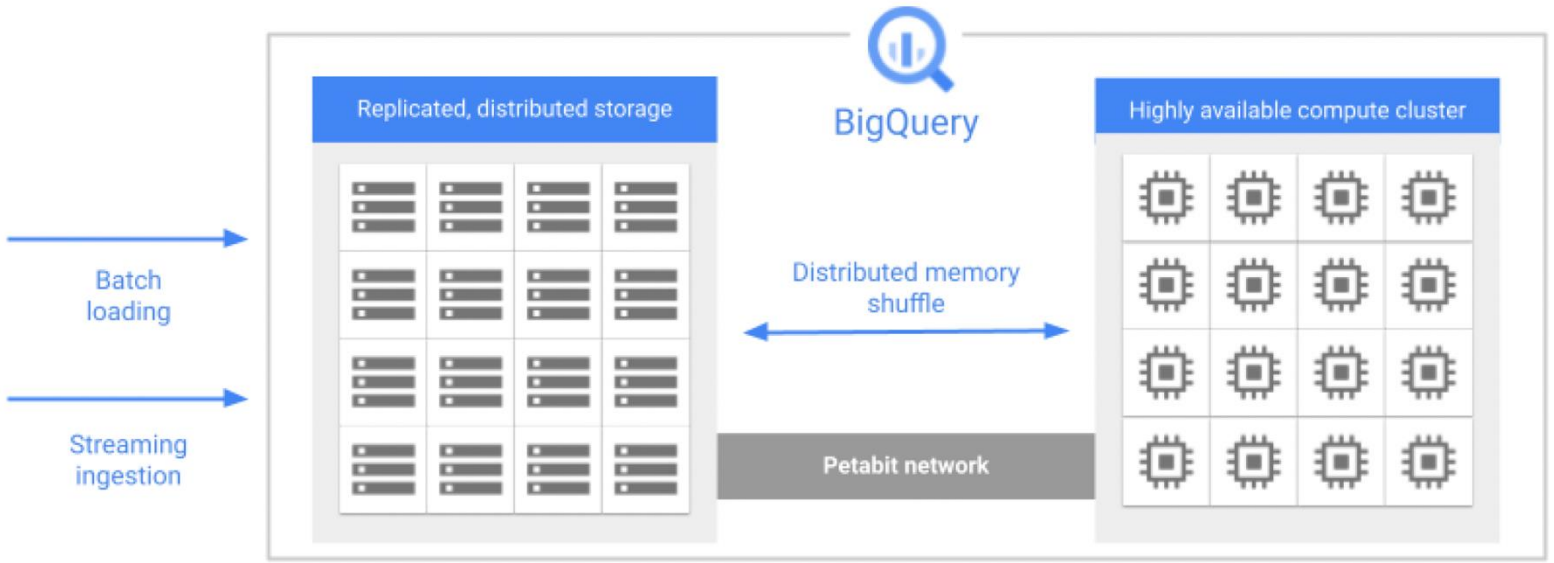
- Utilise SQL
- Évolutif pour analyser d’immenses ensembles de données
- Entrepôt de données d’entreprise
- Lancé en 2012 en utilisant les mêmes outils que
- Utilisations de Google

Qu’est-ce qui rend BigQuery unique ?

- Traitement analytique en ligne (OLAP)
- Calcul et stockage séparés
- Serverless



Calcul et stockage



Snowflake et BigQuery

Snowflake

- Populaire chez les développeurs
- Fonctionne sur n'importe quel cloud
- Des niveaux spécifiques de ressources de calcul (petites, Moyenne, etc.)

BigQuery

- Populaire pour les requêtes analytiques (rapports)
- Ne fonctionne que sur Google Cloud
- Entièrement serverless : l'infrastructure est totalement gérée par Google

Redshift et BigQuery

Redshift

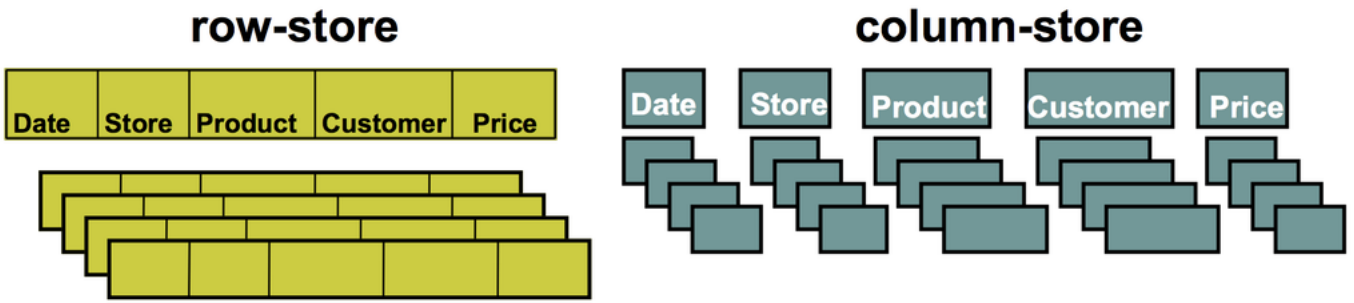
- Calculs constants ou mode serverless
- Adapté aux tableaux de bord en temps réel (live dashboarding)

BigQuery

- Uniquement serverless
- Analyse à un instant donné (point-in-time), typiquement par jour ou par heure

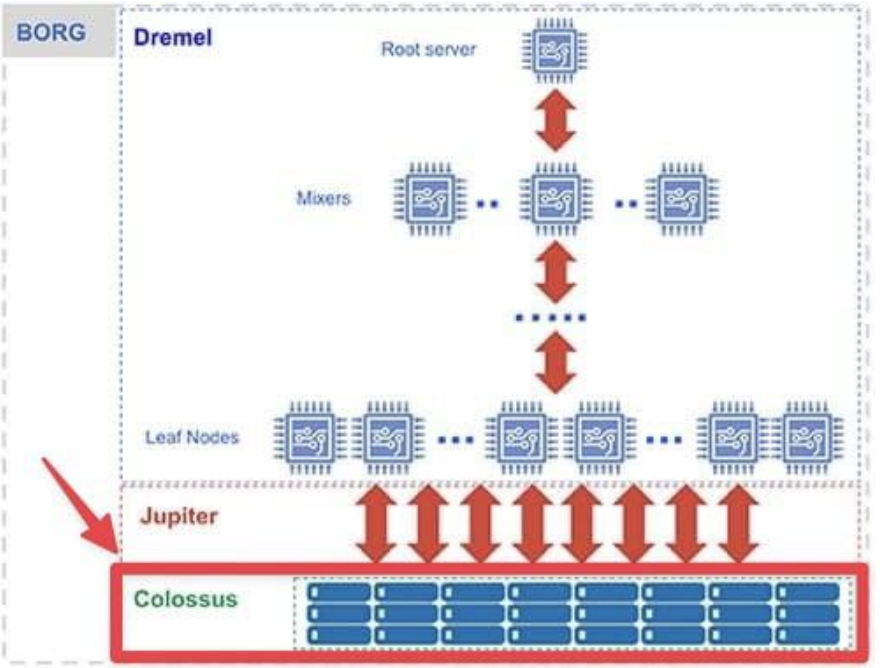
BigQuery

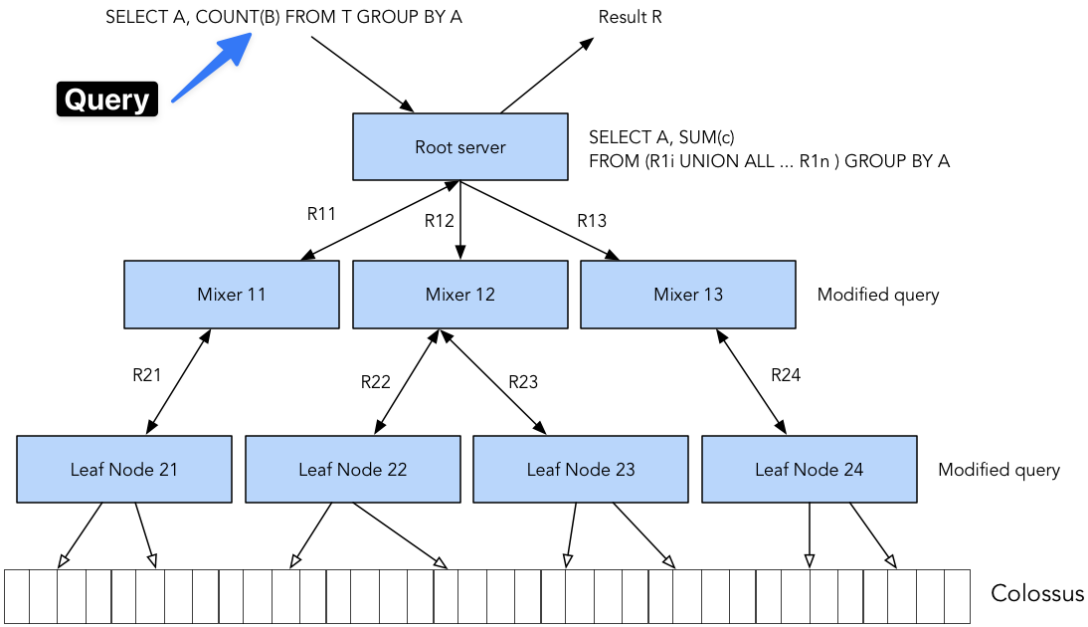
- BigQuery est un SGBD analytique **orienté colonnes**
- Explique :
  - la rapidité des requêtes OLAP
  - le coût basé sur les colonnes lues
  - l'importance de sélectionner uniquement les colonnes utiles



Architecture BigQuery

- **Colossus** : système de fichiers distribué (stockage physique)
- **Capacitor** : format colonne au-dessus de Colossus
- **Dremel** : moteur d'exécution distribué qui pousse le calcul vers les données
- **Borg** est le système interne de Google qui orchestre les ressources de calcul utilisées par BigQuery.
- **Jupiter** : réseau de communication interne de Google, conçu pour supporter des échanges massifs, rapides et fiables entre les composants de BigQuery





BigQuery — Organisation des données

- Le projet constitue la base organisationnelle de BigQuery, définissant le périmètre de vos ressources et de votre facturation.
  - Unité de facturation et d'organisation
  - Contient des datasets
  - Gestion des permissions
  - Isolation des ressources
  - Projet par défaut dans le sandbox

```
SELECT *
FROM `project.dataset.table`
```

BigQuery — Organisation des données

- Les datasets agissent comme des conteneurs logiques, organisant vos tables et définissant les règles d'accès et de localisation des données.
  - Conteneur logique pour les tables
  - Définit la localisation des données
  - Gestion des permissions par dataset
  - Conventions de nommage
  - Datasets publics disponibles

```
SELECT *
FROM `project.dataset.table`
```

BigQuery — Organisation des données

- Les tables BigQuery offrent une flexibilité remarquable avec différents types et options de configuration pour optimiser vos données.
  - Structure des données (schéma)
  - Types de tables : native, externe, vue
  - Partitionnement et clustering
  - Gestion des versions
  - Contrôle d'accès par table

```
SELECT *
FROM `project.dataset.table`
```

BigQuery — Regions

Dans BigQuery, la région est une contrainte majeure :
elle est définie lors de la création du dataset,
les données ne peuvent être déplacées que par duplication,
et une requête ne peut pas accéder à des données situées dans plusieurs régions.

Avantages de BigQuery

- Les avantages de BigQuery en font une solution de choix pour les entreprises cherchant à exploiter leurs données efficacement et sans contraintes techniques.
    - Scalabilité automatique
    - Performance exceptionnelle
    - Coût optimisé (pay-per-query)
    - Sécurité enterprise
    - Intégration ML native
    - Support SQL standard

Le Sandbox BigQuery

- Qu'est-ce que le Sandbox ?
- Le Sandbox BigQuery est votre porte d'entrée gratuite vers l'apprentissage de cette technologie, sans engagement financier ni configuration complexe.
    - Environnement gratuit pour apprendre
    - 1 TB de requêtes par mois
    - 10 GB de stockage
    - Pas de carte bancaire requise
    - Accès aux datasets publics
    - Idéal pour la formation

Accès au Sandbox

- Comment accéder au Sandbox
- L'accès au Sandbox est simple et rapide, vous permettant de commencer à expérimenter avec BigQuery en quelques minutes seulement.
    - https://console.cloud.google.com/bigquery

Interface BigQuery

- L'interface BigQuery est conçue pour être intuitive et productive, regroupant tous les outils nécessaires à l'analyse de données dans un environnement unifié.
  - Panneau de navigation (Explorer)
  - Éditeur de requêtes
  - Résultats des requêtes
  - Historique des requêtes
  - Informations sur les jobs

Navigation dans l'Explorer

- Le panneau Explorer est votre boussole dans l'univers BigQuery, vous permettant de naviguer facilement entre les différentes ressources disponibles.
  - Projets et datasets
  - Tables et vues
  - Fonctions et procédures
  - Datasets publics
  - Favoris et recherche

Types de données BigQuery

- BigQuery supporte une riche variété de types de données, permettant de gérer efficacement tous types d'informations, des plus simples aux plus complexes.
  - Numériques : INT64, FLOAT64, NUMERIC, BIGNUMERIC
  - Chaînes : STRING, BYTES
  - Dates : DATE, TIME, DATETIME, TIMESTAMP
  - Booléens : BOOL
  - Structures : ARRAY, STRUCT
  - Géographiques : GEOGRAPHY

Datasets publics

- Les datasets publics constituent une ressource inestimable pour l'apprentissage et l'expérimentation avec des données réelles et variées.
  - bigquery-public-data.samples
  - bigquery-public-data.usa_names
  - bigquery-public-data.chicago_crime
  - bigquery-public-data.stackoverflow
  - Plus de 200 datasets publics

Première requête SQL

- Commençons par découvrir la syntaxe SQL de base dans BigQuery, similaire au SQL standard mais avec quelques spécificités importantes.
  - SELECT column1, column2
  - FROM `project.dataset.table`
  - WHERE condition
  - ORDER BY column1
  - LIMIT 1000
- Cette structure de base vous permettra de construire toutes vos requêtes futures en BigQuery.

Exploration d'une table

- Avant d'analyser des données, il est essentiel de bien comprendre leur structure et contenu grâce à ces commandes exploratoires.
  - DESCRIBE `project.dataset.table`;
  - -- Compter les lignes
  - SELECT COUNT(*) FROM `project.dataset.table`;
  - -- Aperçu des données
  - SELECT * FROM `project.dataset.table` LIMIT 10;
- Ces commandes sont vos outils de base pour explorer et comprendre vos données.

Requête SELECT de base

- Découvrons maintenant comment sélectionner et ordonner des données spécifiques avec notre premier exemple pratique.
    - SELECT name, number, year
    - FROM `bigquery-public-data.usa_names.usa_1910_2013`
    - WHERE year = 2013
    - ORDER BY number DESC
    - LIMIT 10;
- Cette requête nous montre les 10 prénoms les plus populaires en 2013.

Filtrage avec WHERE

- Le filtrage est essentiel pour extraire uniquement les données pertinentes de vos analyses.
  - SELECT name, number, year
  - FROM `bigquery-public-data.usa_names.usa_1910_2013`
  - WHERE year BETWEEN 2010 AND 2013
  - AND gender = 'F'
  - AND number > 1000
  - ORDER BY number DESC;
- Cette requête combine plusieurs conditions pour filtrer les prénoms féminins populaires de la dernière décennie.

Fonctions d'agrégation

- Fonctions courantes
- Les fonctions d'agrégation permettent de synthétiser vos données et d'obtenir des statistiques significatives.
    - SELECT gender, COUNT(*) as total_records, SUM(number) as total_births, AVG(number) as avg_births, MAX(number) as max_births
    - FROM `bigquery-public-data.usa_names.usa_1910_2013`
    - GROUP BY gender;
- Ces statistiques nous donnent un aperçu global des différences entre prénoms masculins et féminins.

Regroupement et filtrage GROUP BY et HAVING

- La combinaison GROUP BY et HAVING permet d'analyser des données par groupes et de filtrer les résultats agrégés.
    - SELECT year, COUNT(DISTINCT name) as unique_names, SUM(number) as total_births
    - FROM `bigquery-public-data.usa_names.usa_1910_2013`
    - GROUP BY year
    - HAVING total_births > 1000000
    - ORDER BY year;
- Cette analyse révèle l'évolution démographique à travers les années avec un seuil minimum de naissances.

Fonctions de chaînes

- Les fonctions de chaînes offrent de puissantes capacités de manipulation et transformation du texte.
  - SELECT UPPER(name) as name_upper, LOWER(name) as name_lower, LENGTH(name) as name_length, SUBSTR(name, 1, 3) as name_prefix
  - FROM `bigquery-public-data.usa_names.usa_1910_2013`
  - WHERE year = 2013
  - LIMIT 10;
- Ces transformations sont essentielles pour nettoyer et standardiser vos données textuelles.

Fonctions de dates

- Les fonctions de dates permettent d'extraire, formater et manipuler les informations temporelles efficacement.
  - SELECT year, EXTRACT(DECADE FROM DATE(year, 1, 1)) as decade, DATE(year, 1, 1) as year_date, FORMAT_DATE('%Y-%m-%d', DATE(year, 1, 1)) as formatted_date
  - FROM `bigquery-public-data.usa_names.usa_1910_2013`
  - WHERE year >= 2010
  - GROUP BY year
  - ORDER BY year;
- Cette approche vous permettra d'analyser vos données selon différentes granularités temporelles.

Expressions conditionnelles CASE WHEN

- Les expressions conditionnelles permettent de créer des catégories et d'appliquer une logique métier directement dans vos requêtes.
- SELECT name, number,
- CASE
- WHEN number > 10000 THEN 'Very Popular'
- WHEN number > 1000 THEN 'Popular'
- WHEN number > 100 THEN 'Common'
- ELSE 'Rare'
- END as popularity_category
- FROM `bigquery-public-data.usa_names.usa_1910_2013`
- WHERE year = 2013
- ORDER BY number DESC;
- Cette catégorisation automatique simplifie l'analyse et la présentation des résultats.

Requêtes imbriquées

- Les sous-requêtes permettent de construire des analyses complexes en combinant plusieurs niveaux de traitement.
  - SELECT name, total_births, (total_births / max_births) * 100 as percentage_of_max
  - FROM (
  - SELECT name, SUM(number) as total_births, MAX(SUM(number)) OVER() as max_births
  - FROM `bigquery-public-data.usa_names.usa_1910_2013`
  - GROUP BY name
  -)
  - ORDER BY total_births DESC
  - LIMIT 10;
- Cette technique permet de calculer des pourcentages relatifs et d'effectuer des analyses comparatives sophistiquées.

Bonnes pratiques - Optimisation

- Optimisation des requêtes
- L'optimisation est cruciale pour maintenir des performances élevées et contrôler les coûts dans BigQuery.
    - Utilisez LIMIT pour les tests
    - Sélectionnez seulement les colonnes nécessaires
    - Utilisez WHERE pour filtrer tôt
    - Évitez SELECT *
- Préférez les filtres sur les colonnes partitionnées

Bonnes pratiques - Nommage

- Des conventions de nommage cohérentes améliorent la lisibilité et la maintenabilité de vos requêtes.
  - Utilisez des noms explicites
  - Préférez snake_case
  - Évitez les mots-clés SQL
  - Utilisez des alias pour les expressions complexes
  - Documentez vos requêtes

Gestion des erreurs courantes

- Connaître les erreurs les plus communes vous permettra de les éviter et de diagnostiquer rapidement les problèmes.
  - Noms de table incorrects
  - Limites de quota dépassées
  - Erreurs de syntaxe SQL
  - Types de données incompatibles
  - Fonctions non supportées

Historique et monitoring

- Le monitoring de vos requêtes est essentiel pour optimiser les performances et contrôler les coûts.
  - Onglet "Historique des requêtes"
  - Temps d'exécution
  - Données traitées
  - Coût estimé
  - Erreurs et optimisations

TP : Informatique Décisionnelle & OLAP avec BigQuery

Références

- Cours Systèmes d'information décisionnels, E. GRISLIN--LE STRUGEON et D. DONSEZ
- Cours de Entrepôts de données et analyse en ligne, Bernard ESPINASSE
- http://www.kimballgroup.com/
- SQL Server Microsoft et OLAP
- Cours de Introduction aux systèmes d'information décisionnelle, O. Boussaid