Business Intelligence - M1DAC
DataWarehouse

Laure Soulier
Sorbonne Université
LIP6, Paris - France

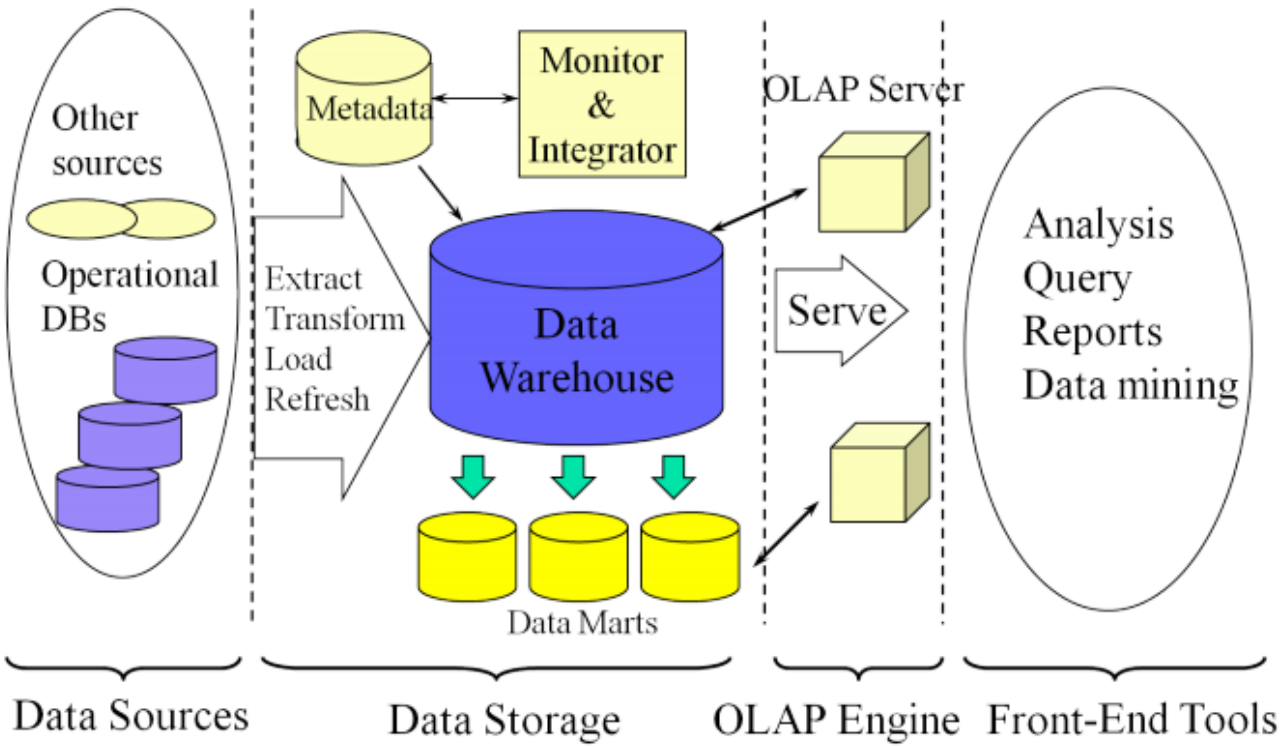
25 janvier 2021

Overview

Data Warehouse - Entrepôt de Données

Système informatique décisionnel

- Système permettant de mesurer un/des phénomènes à analyser.
- Outils : SGBD, ETL, outils d'interrogation, d'analyse et de restitution ("reporting")



Data Warehouse - Entrepôt de Données

Datawarehouse

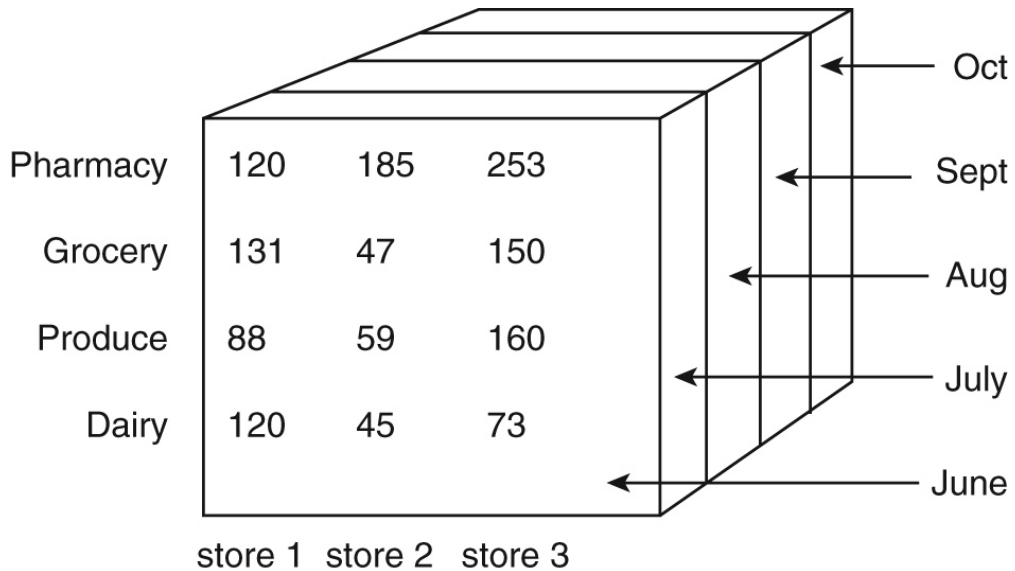
Le terme entrepôt de données (ou base de données décisionnelle, ou encore data warehouse) désigne une base de données utilisée pour collecter, ordonner, journaliser et stocker des informations provenant de base de données opérationnelles et fournir ainsi un socle à l'aide à la décision en entreprise. (source wikipedia)

- **Collecter** : Récupérer l'information produite par l'entreprise
- **Ordonner** : Structurer l'information dans le but de la prise de décision (structure différente des BDs opérationnelles)
- **Journaliser** : Stocker l'historique des données

$\rightarrow$ Données orientées sujet (en fonction du thème, organigramme de la société, services, ...)

Concept d'hypercube

- Données issues de **sources externes** (bases de données, csv, ...)
- Données **agrégées** (somme, moyenne, min, max, ...)
- Données **hiérarchisées**

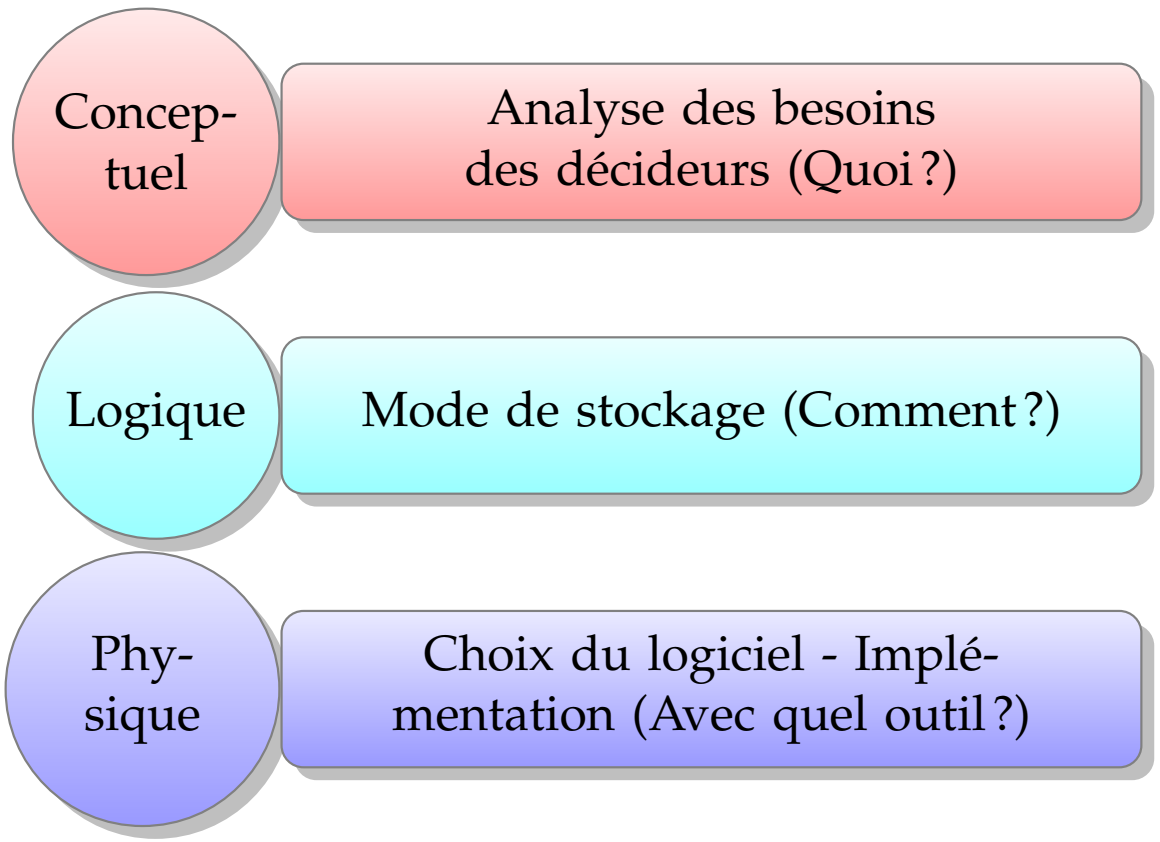


Hypercube : Espace à n dimensions

Overview

MODÉLISATION MULTIDIMENSIONNELLE

- Modélisation des données pour supporter efficacement les processus OLAP ("On Line Analytic Processing")
- Niveaux d'abstraction identiques à la modélisation relationnelle



Modélisation conceptuelle

Caractéristiques du modèle relationnel :

- Normalisation (3NF)
- Répond aux besoins transactionnels (OLTP)
- Avantages :
  - Réduction de l'entrée de données
  - Réduction du nombre d'index
  - Ajouts/destructions/modifications plus rapides
- Inconvénients :
  - Peu efficace pour l'extraction de données analytiques
  - Beaucoup de relations
  - Trop complexe pour l'utilisateur BI

Modélisation conceptuelle

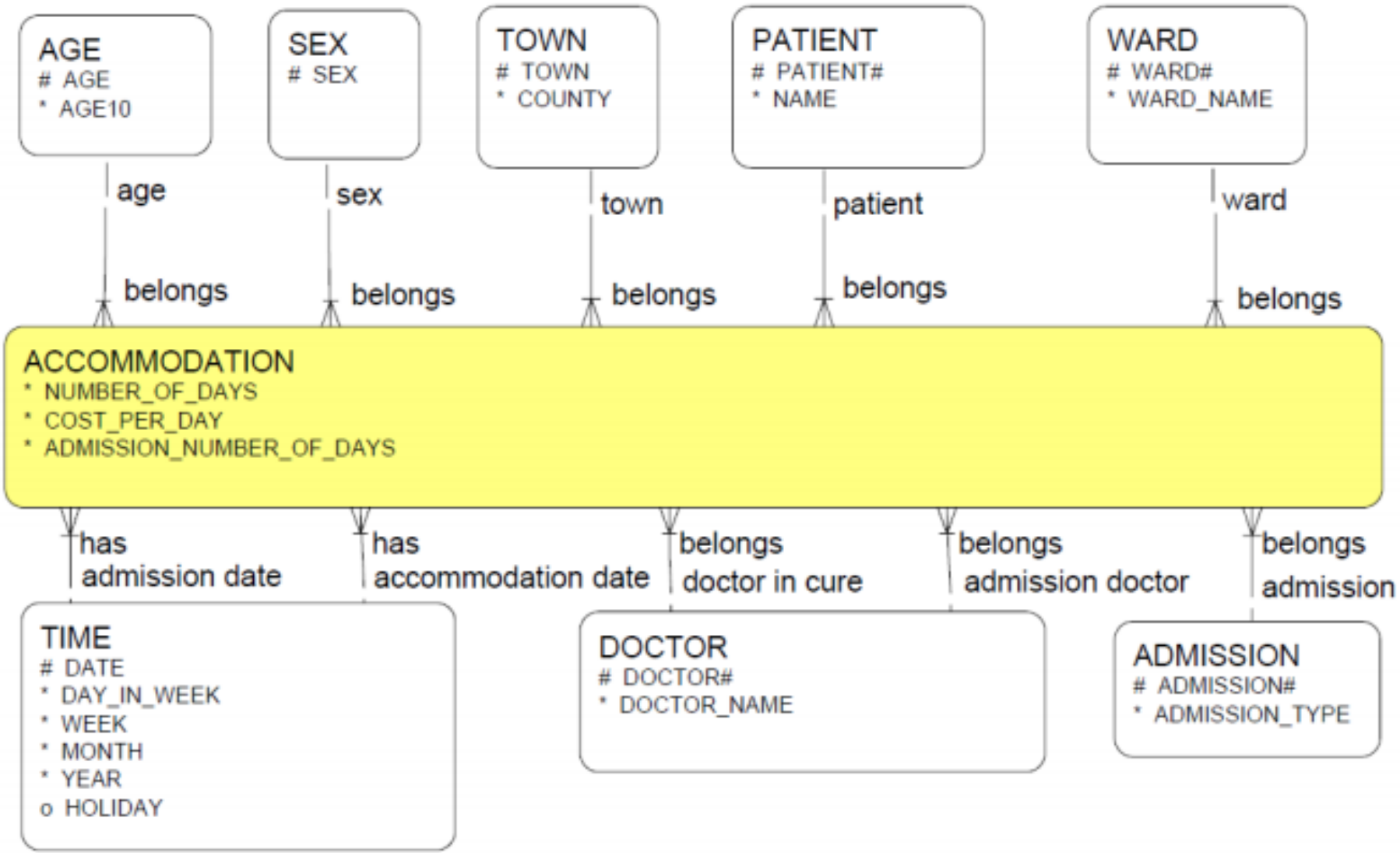
3 types de modélisation :

- Schéma en étoile
- Schéma en flocon
- Schéma en constellation

Schéma en étoile

- Schéma $S = (Nom_S, F, D_1, D_2, ..., D_x)$
- Fait $F = (Nom_F, m_1, m_2, ..., m_y)$
  - Sujet d'analyse (cellule de l'hypercube)
  - Orienté selon les besoins du décideur
  - Les mesures m_i sont les critères sur lesquels on souhaite faire l'analyse
- Dimension $D = (Nom_D, a_1, ..., a_z)$
  - Axe d'analyse (arrête d'un hypercube)
  - $a_1, ..., a_z$ attributs d'analyse

Schéma en étoile



Exemple 1

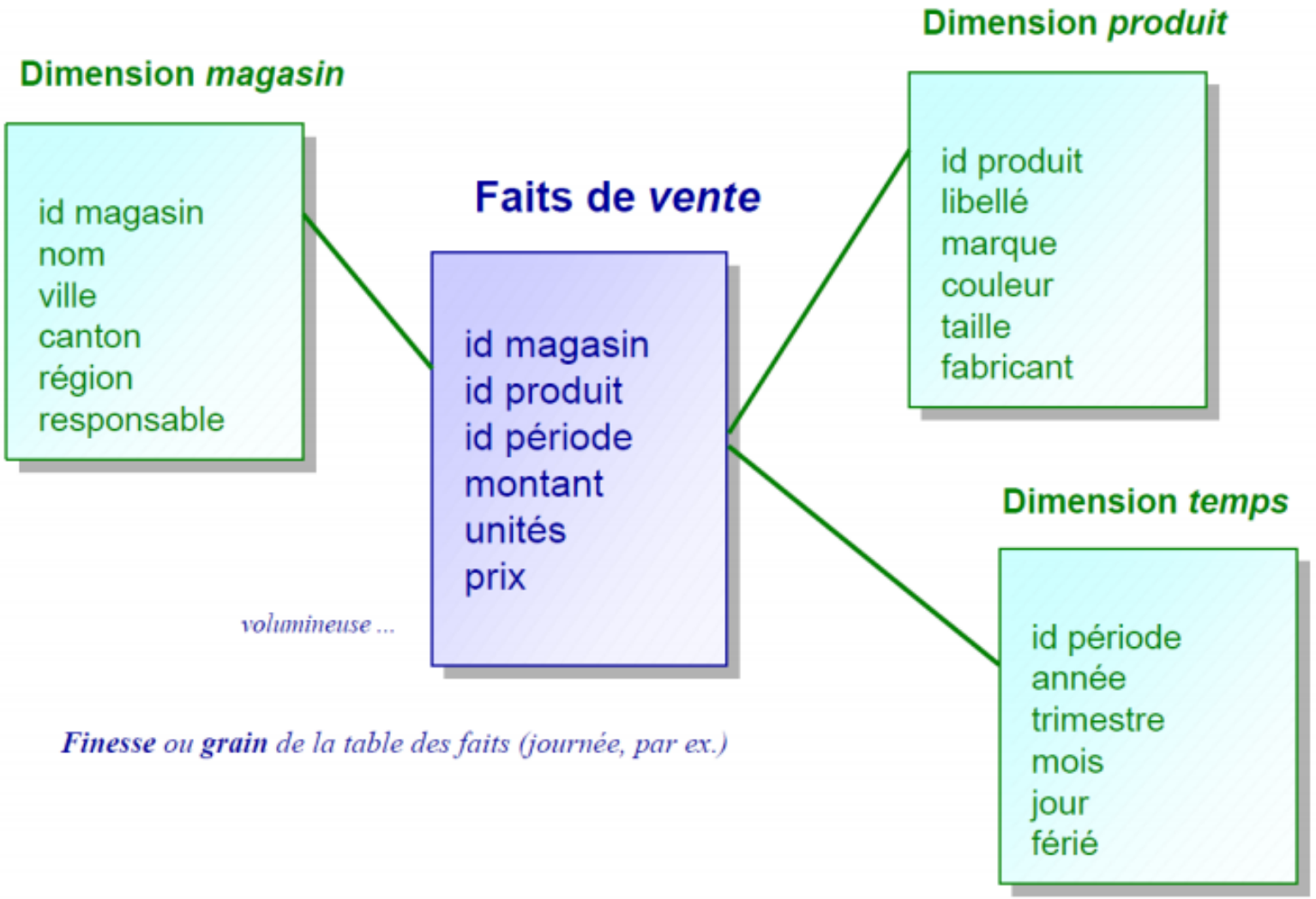
Notre société vend des produits dans des magasins. Les produits sont décrits par un nom, une marque, un type, une couleur, une taille et un fabricant. Un client (nom, prénom, adresse) peut acheter pour un certain montant plusieurs unités d'un même produit (une vente = 1 ou plusieurs unités).
Un magasin possède un responsable et une région.

Le service marketing souhaite être en mesure d'obtenir les réponses aux questions suivantes :

- Je veux un diagramme des ventes globales de l'entreprise dans le temps
- Je veux un diagramme des ventes par magasin
- Je veux un diagramme des ventes par produit, et par magasin

Question : Dessinez le schéma du DW sous-jacent. Donnez un exemple de 'valeurs' pour chacune des tables créées.

EXEMPLE 1



***Finesse** ou **grain** de la table des faits (journée, par ex.)*

Schéma en flocon (1/2)

- Schéma $S = (Nom_S, F, D_1, D_2, ..., D_x)$
- Fait $F = (Nom_F, m_1, m_2, ..., m_y)$
  - Sujet d'analyse (cellule de l'hypercube)
  - Orienté selon les besoins du décideur
  - Les mesures sont les critères sur lesquels on souhaite faire l'analyse
- Dimension $D = (Nom_D, a_1, ..., a_z, \mathbf{H_1}, ..., \mathbf{H_w})$
  - Axe d'analyse (arrête d'un hypercube)
  - $a_1, ..., a_z$ attributs d'analyse
  - $\mathbf{H_1}, ..., \mathbf{H_w}$ **hiérarchie, organisation de graduation des attributs sur un axe d'analyse**
- **Hiérarchie** $H = (Nom_H, < p_1, ..., p_n >)$
  - **Niveau de granularité d'analyse**
  - $< p_1, ..., p_n >$ **: liste des attributs ordonnés**

Schéma en flocon (2/2)

Les mesures ont différentes caractéristiques suivant leur exploitation de la dimension :

- Mesures additives : additionnables le long de toutes les dimensions (ex : chiffre d'affaires par rapport aux clients/pays/mois/...)
- Mesures semi-additives : additionnables le long de certaines dimensions (ex : pas de sens de cumuler le niveau des stocks au fil du temps ;mais par magasin, cela a du sens)
- Mesures non additives : ne peut être additionnables selon aucune dimension (ex : prix d'un produit)

Exemple 2

On considère maintenant les requêtes suivantes :

- Je veux un diagramme des ventes globales de l'entreprise par jour
- Je veux un diagramme des ventes par mois
- Je veux un diagramme des ventes par saison (été, hiver, automne, printemps)
- Je veux un diagramme des ventes par couleur de produit
- Je veux un diagramme des ventes par type de produit
- Je veux un diagramme des ventes par gamme de produit
- Je veux un diagramme des ventes par ville
- Je veux un diagramme des ventes par région

Question : Dessinez le schéma du DW sous-jacent. Donnez un exemple de 'valeurs' pour chacune des tables créées.

EXEMPLE 2

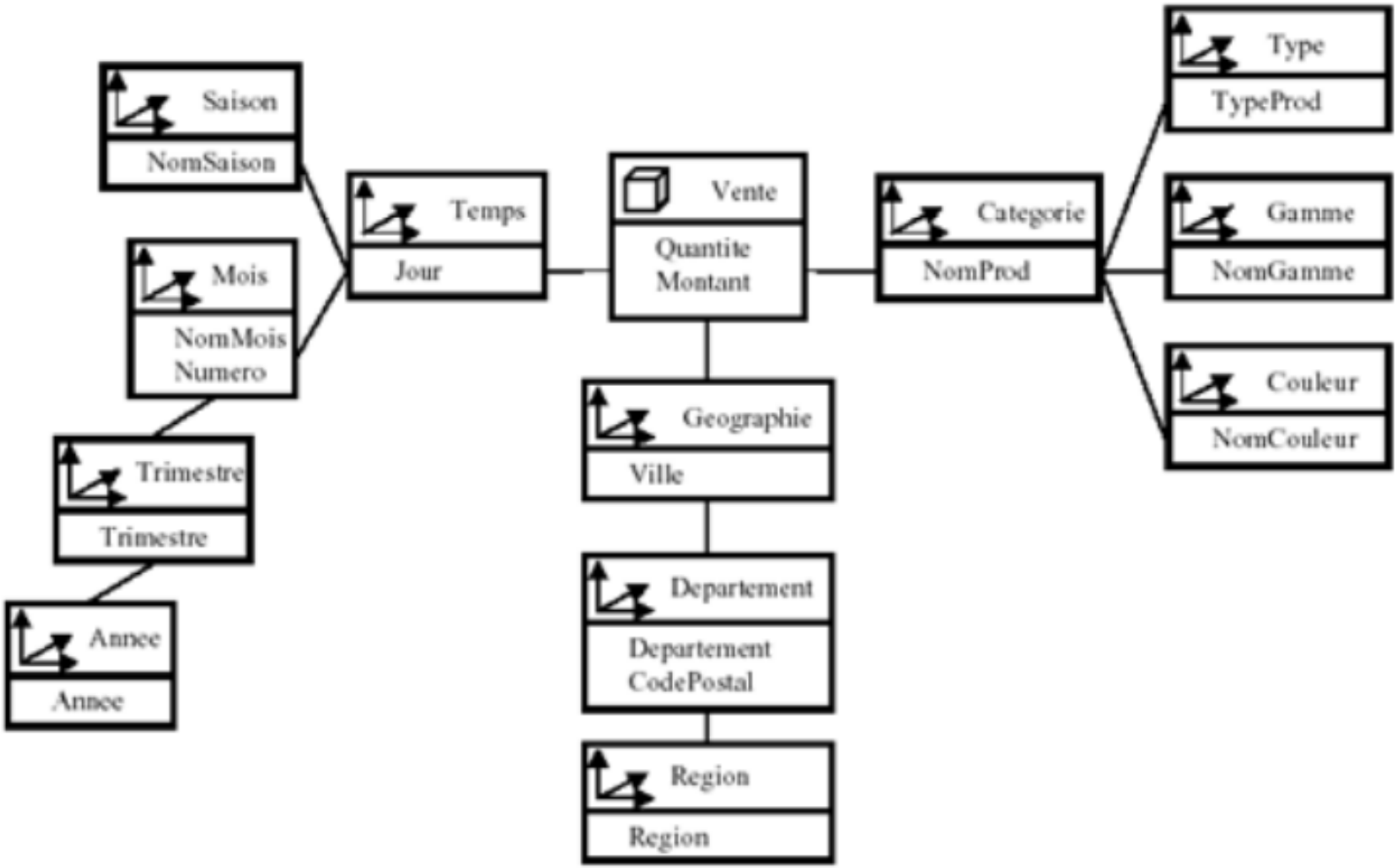
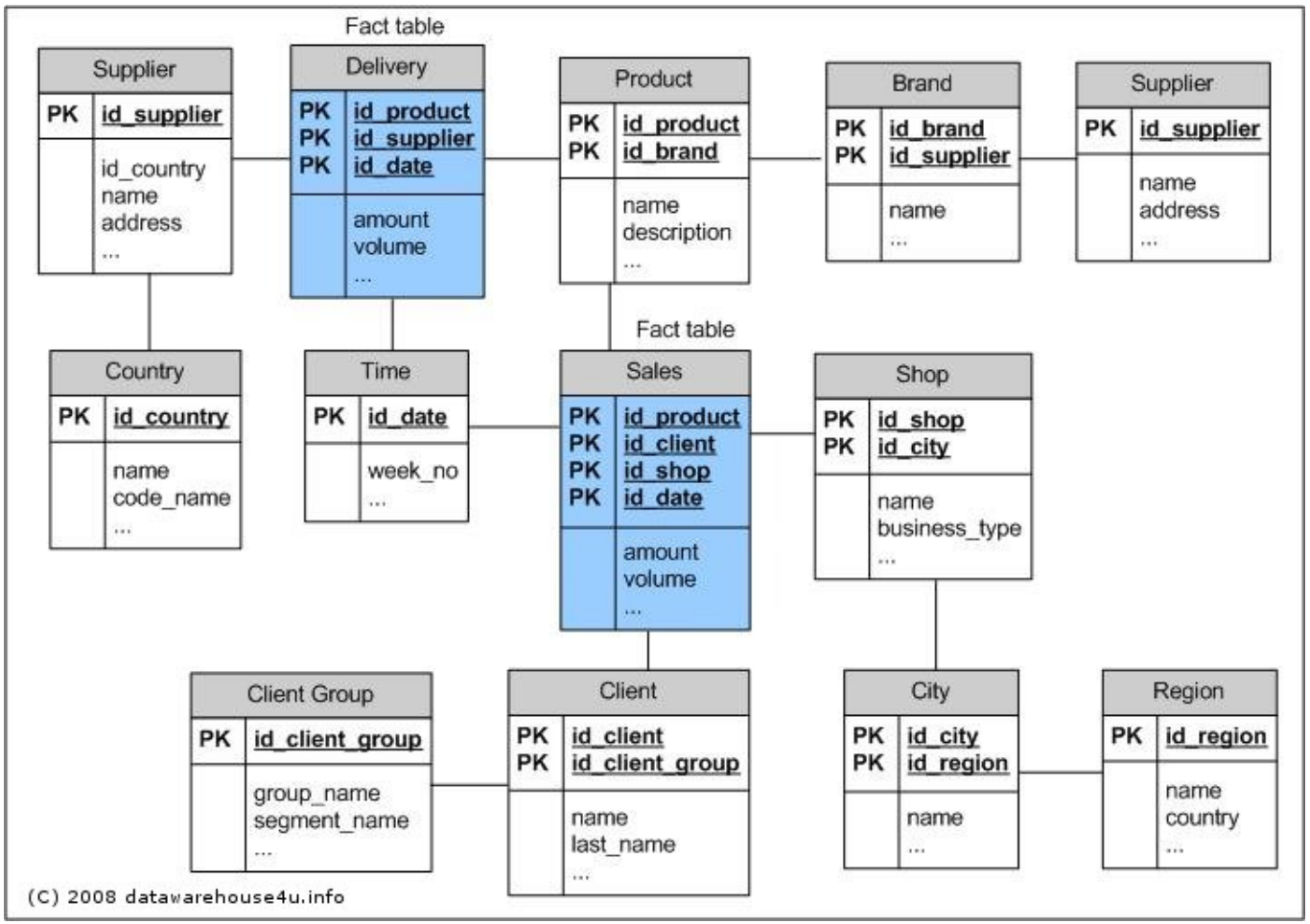

Saison
NomSaison
Mois
NomMois
Numero
Trimestre
Trimestre
Annee
Annee
Temps
Jour
Vente
Quantite
Montant
Geographie
Ville
Departement
Departement
CodePostal
Region
Region
Categorie
NomProd
Type
TypeProd
Gamme
NomGamme
Couleur
NomCouleur

SCHÉMA EN CONSTELLATION

- Généralisation du schéma en étoile
- N faits et N dimensions
- Partage de dimensions entre les faits



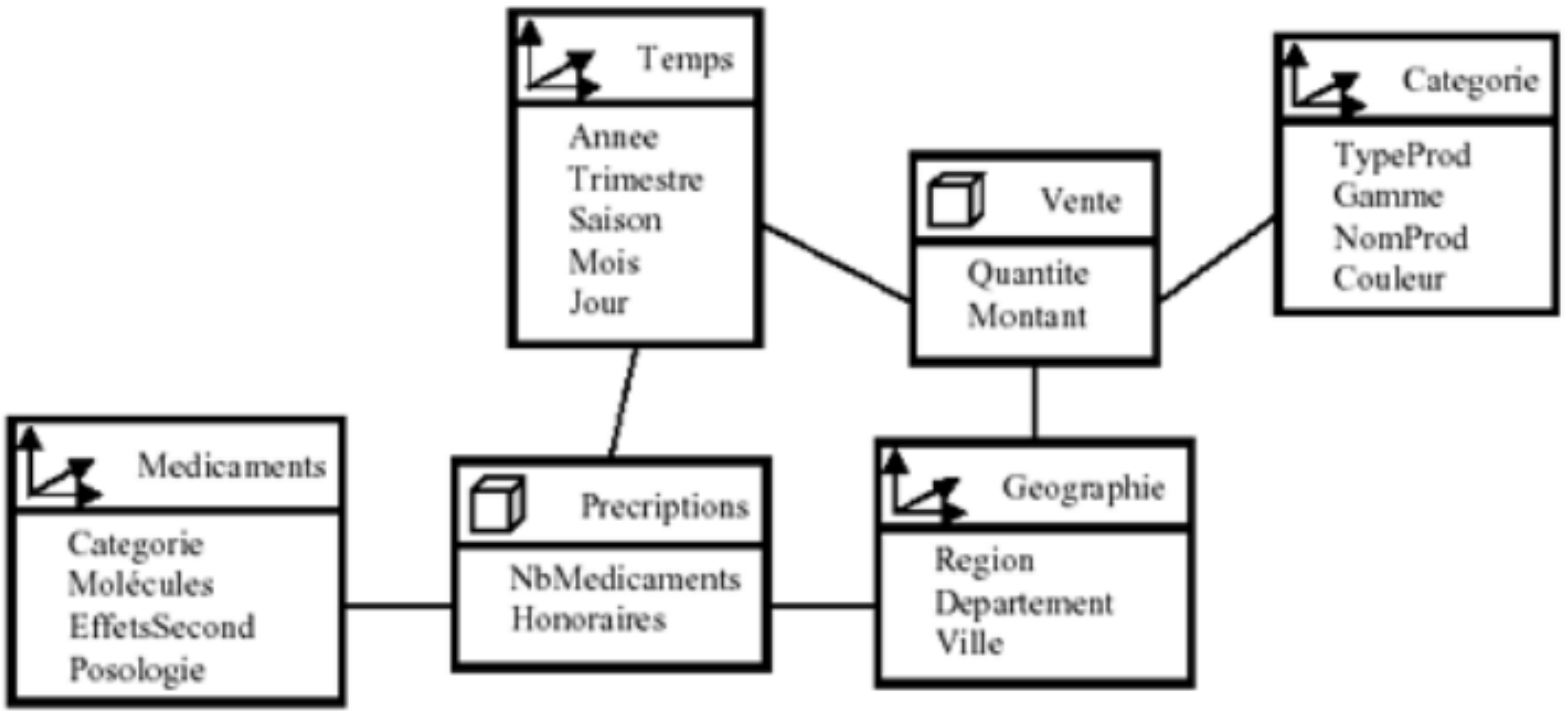
Exemple 3

On considère maintenant que nos magasins vendent en plus des médicaments.
En plus des demandes précédentes (exercice 2), le service du marketing souhaite aussi obtenir :

- Les prescriptions par médicaments (les médicaments ne sont pas des produits, il faut les considérer à part)
- Les prescriptions par médicaments par mois prescriptions par médicaments magasin

Question : Dessinez le schéma du DW sous-jacent (sans les hiérarchies). Donnez un exemple de 'valeurs' pour chacune des tables créées.

EXEMPLE 3


Temps
Annee
Trimestre
Saison
Mois
Jour
Vente
Quantite
Montant
Categorie
TypeProd
Gamme
NomProd
Couleur
Medicaments
Categorie
Molécules
EffetsSecond
Posologie
Precriptions
NbMedicaments
Honoraires
Geographie
Region
Departement
Ville

Modélisation logique - Relationnel OLAP

Règles de transformation pour un schéma logique **dénormalisé** :

- R1 : Toute dimension est transformée en relation (table) où :
  - Tous les attributs de la dimension deviennent des attributs de la relation
  - L'attribut le plus bas (granularité fine) devient la clé primaire
- R2 : Tout fait est transformé en une relation où :
  - La clé primaire est la concaténation des clés étrangères référençant les dimensions (ou une clé synthétique)
  - Les attributs sont les mesures du fait

Avantages/Inconvénients

Modélisation logique - Relationnel OLAP

Règles de transformation pour un schéma logique **dénormalisé** :

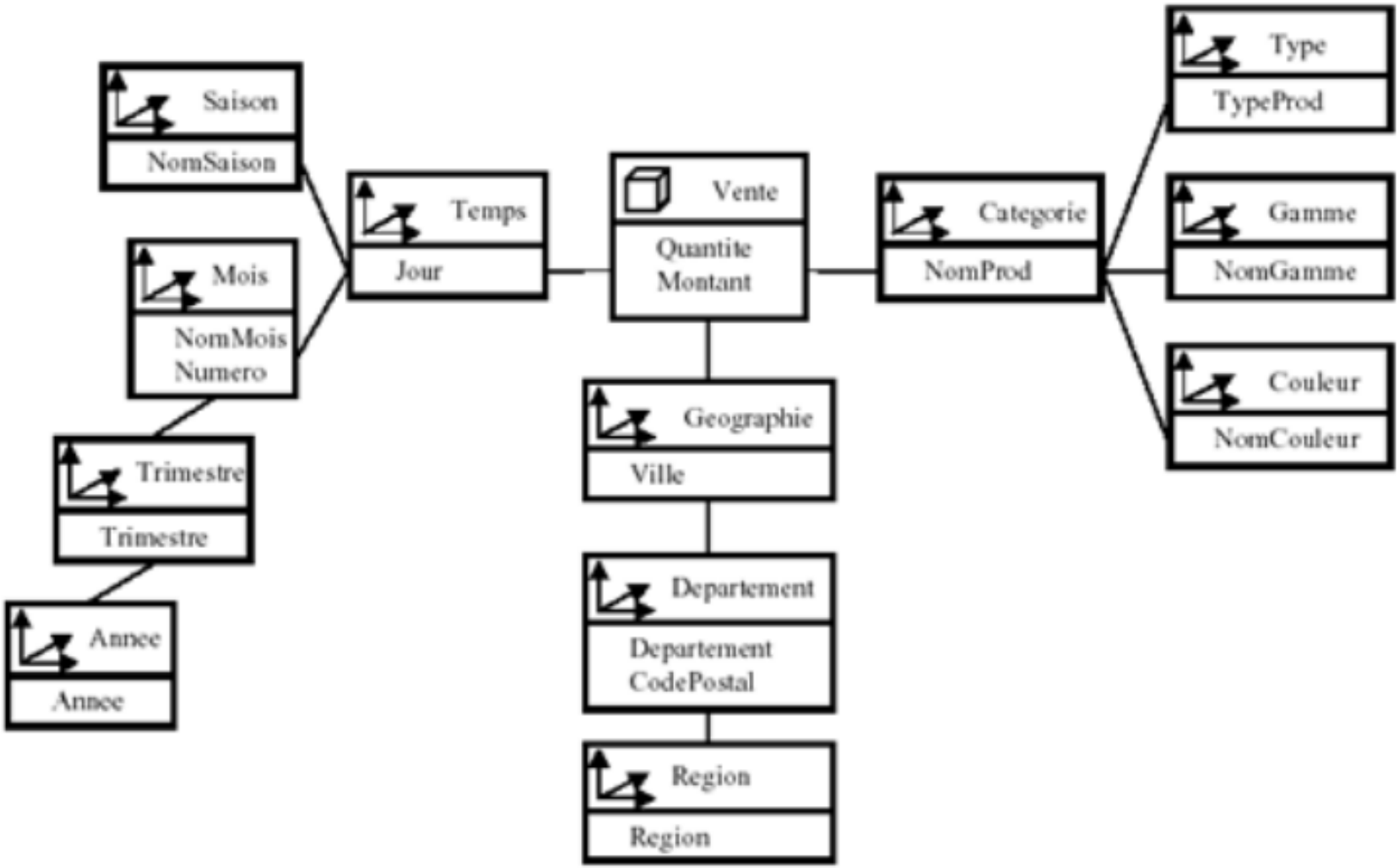
- R1 : Toute dimension est transformée en relation (table) où :
  - Tous les attributs de la dimension deviennent des attributs de la relation
  - L'attribut le plus bas (granularité fine) devient la clé primaire
- R2 : Tout fait est transformé en une relation où :
  - La clé primaire est la concaténation des clés étrangères référençant les dimensions (ou une clé synthétique)
  - Les attributs sont les mesures du fait

Avantages/Inconvénients

- Perte de la notion de hiérarchie
+ Peu de jointures, peu conduire à des redondances
→ Schéma performant

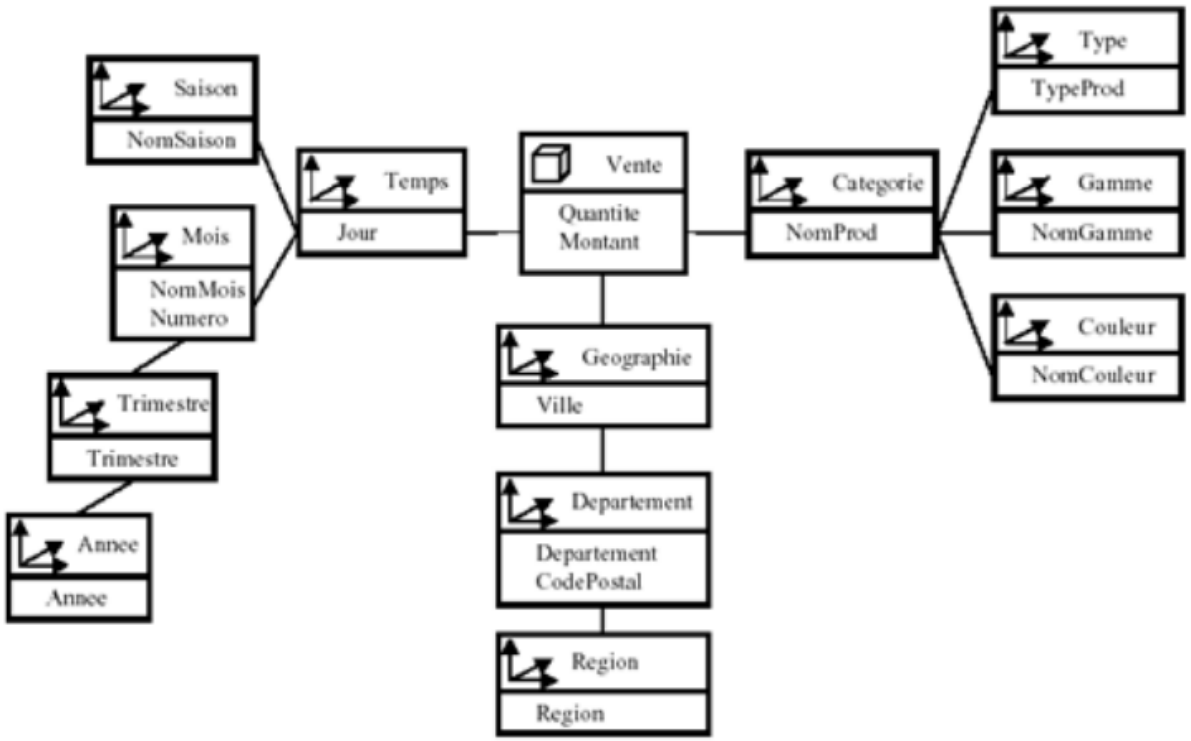
EXEMPLE 4

Transformer le schéma conceptuel suivant en schéma logique dénormallisé.



EXEMPLE 4

Transformer le schéma conceptuel suivant en schéma relationnel.



- Lieu (**Ville**, Code Postal, Département, Region)
- Date (**Jour**, NomSaison, NomMois, NumeroMois, Trimestre, Année)
- Produit (**NomProd**, TypeProd, NomGamme, NomCouleur)
- Vente (**#Ville, #Jour,# NomProd**, Quantité, Montant)

Modélisation logique - Relationnel OLAP

Règles de transformation pour un schéma logique **normalisé** :

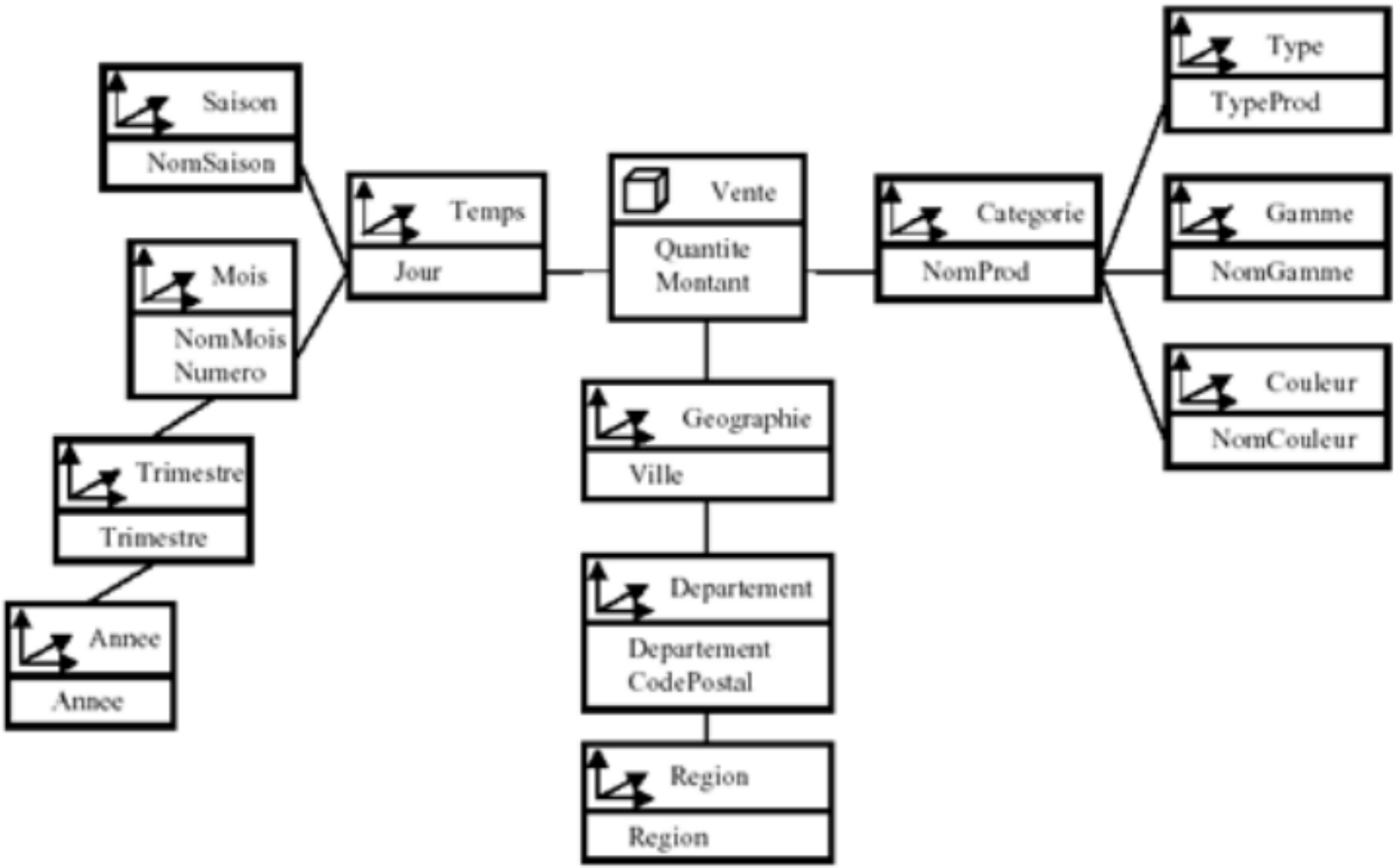
- R1 : une dimension est représentée par plusieurs tables :
  - Chaque table représente un niveau d'agrégation
  - Chaque table est composée de : 1 clé primaire (paramètre du niveau d'agrégation), une clé étrangère qui permet de faire le lien avec le niveau d'agrégation supérieur et éventuellement un ensemble d'attributs associés.
- R2 : Tout fait est transformé en une relation où :
  - La clé primaire est la concaténation des clés étrangères référençant les dimensions (ou une clé synthétique)
  - Les attributs sont les mesures du fait

Avantages/Inconvénients

+ Hiérarchie explicite
- Nombreuses jointures

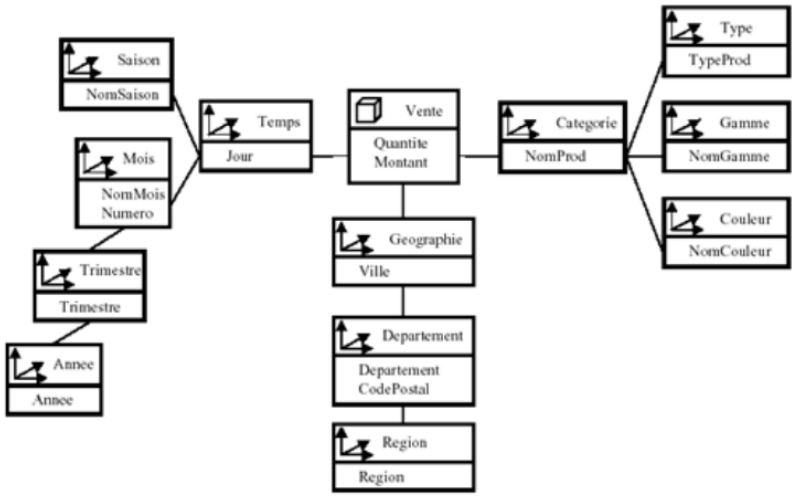
EXEMPLE 5

Transformer le schéma conceptuel suivant en schéma logique normallisé.



EXEMPLE 5

Transformer le schéma conceptuel suivant en schéma relationnel.



- Lieu (**idVille**, nomVille, #idCP)
- CodePostal(**idCP**, codePost, NomDepartement, #idRegion)
- Region(**idRegion**, NomRegion)
- DateJour(**idJour**, Jour, #idSaison, #NumeroMois)
- DateSaison(**idSaison**, NomSaison)
- DateMois(**NumeroMois**, NomMois, #idTrimestre)
- DateTrimestre(**idTrimestre**, Trimestre, #Annee)
- DateAnnee(**Annee**)
- Produit (**NomProd**, #idTypeProd, #idNomGamme, #idNomCouleur)

Modélisation logique - Relationnel OLAP

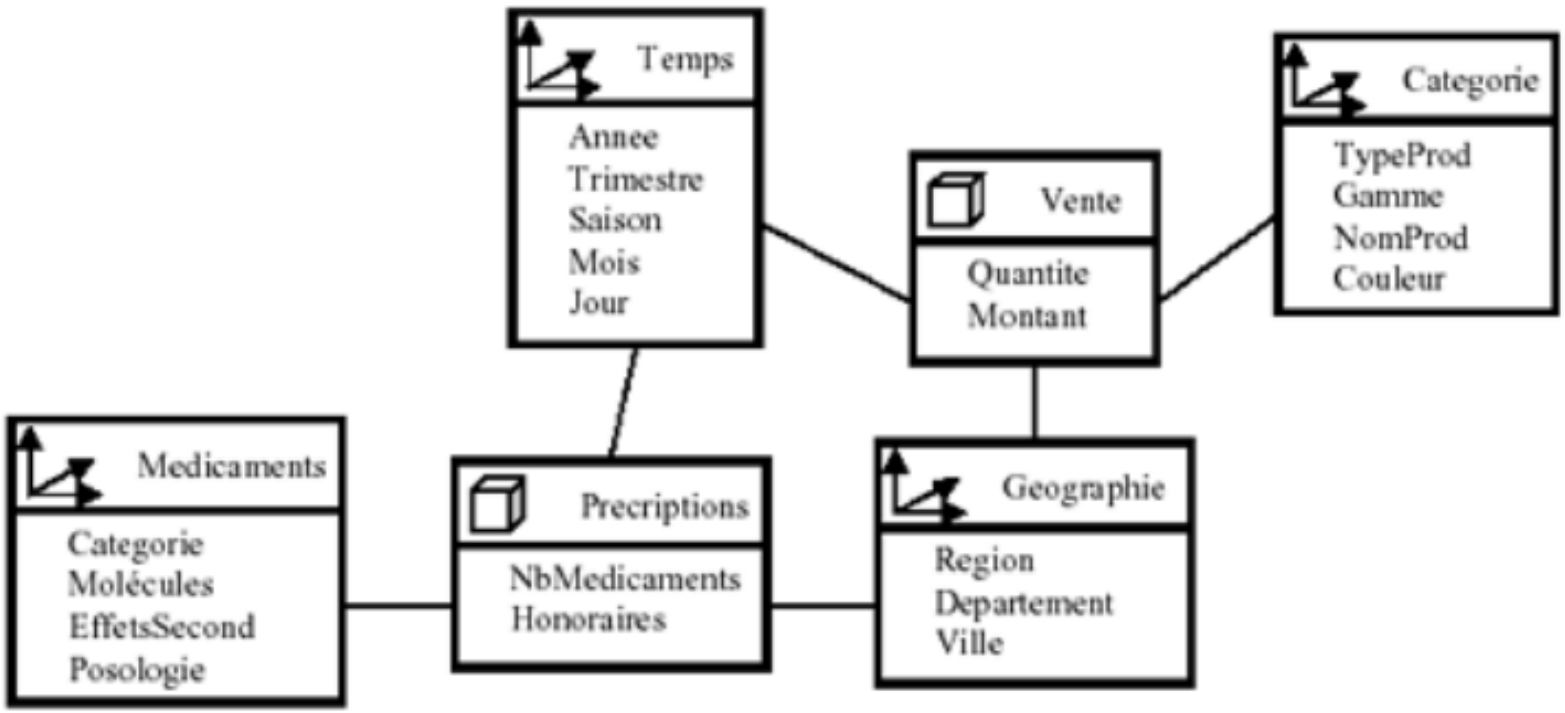
Règles de transformation pour un schéma logique hybride (partiellement normalisé) :

- R1 : Une dimension monofait est représentée par une table
- R2 : Une dimension multifaits partagée à un même niveau de granularité par plusieurs faits est représenté par une table (on n'explose pas la dimension)
- R3 : toute dimension partagée à des niveaux de granularité différent par plusieurs faits est représentée par plusieurs tables
- R4 : tout fait est représenté par une table

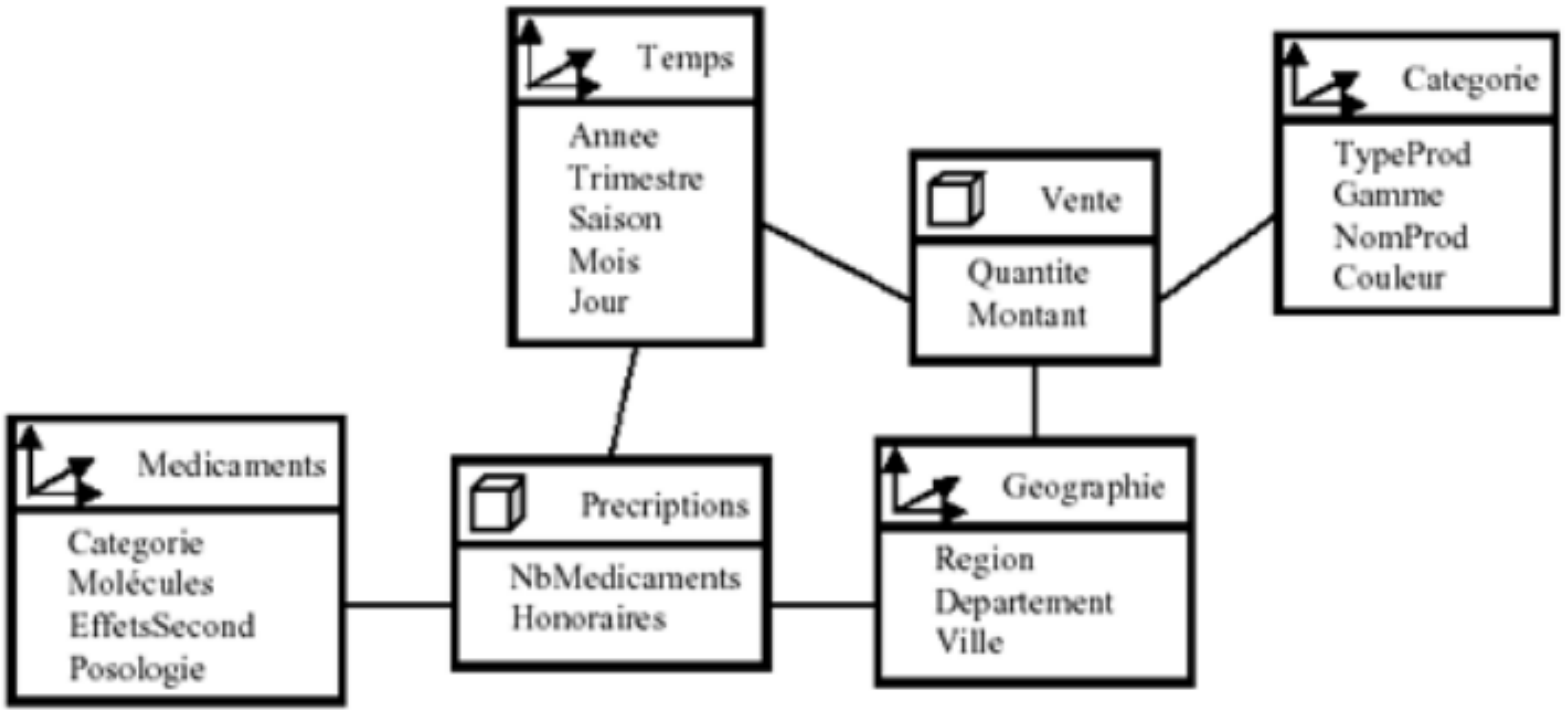
Avantages/Inconvénients

+ Permet d'éviter les redondances

Exemple 6


Temps
Annee
Trimestre
Saison
Mois
Jour
Vente
Quantite
Montant
Categorie
TypeProd
Gamme
NomProd
Couleur
Medicaments
Categorie
Molécules
EffetsSecond
Posologie
Precriptions
NbMedicaments
Honoraires
Geographie
Region
Departement
Ville

Exemple 6



- Categorie(**idC**,TypeProd, Gamme, NomProd, Couleur)
- Medicament(**idMed**, categorie, molecules, effetsSecond, Posologie)
- DateMois(**idDateMois**, jour, mois,#idD)
- DateAutre(**idD**, saison, trimestre, annee)
- Geographie(**idG**, ville, departement, region)
- Vente(#idC, #idDateMois, #idD, #idG quantite, montant)
- Prescription(#idMed, #idDateMois, #idG, NbMedicaments, Honoraires)

Modélisation physique

- Limites des commandes SQL
  - CREATE TABLE NomTable AS SELECT ... : recopie physique d'une table. Si mise à jour de la table source, pas de mise à jour de la table "NomTable"
  - CREATE VIEW NomView AS SELECT ... : la vue est recalculé à chaque requête
- Implémentation dépendante du logiciel. En principe : vues matérialisées.

Exemple d'implémentation en Oracle 10g

- Création de l'alias de la BD

```
CREATE DATABASE LINK myRelDB CONNECT TO user
IDENTIFIED BY user USING localDB;
```

- Création du fait

```
CREATE MATERIALIZED VIEW salesMv
BUILD IMMEDIATE|DEFERRED
REFRESH FAST|COMPLETE|FORCE ON COMMIT|ON DEMAND
AS SELECT t.calendarYear, p.prodId,
SUM(s.amountSold) AS sumSales
FROM times@myRelDB t, products@myRelDB p,
sales@myRelDB s
WHERE t.timeId = s.timeId AND p.prodId = s.prodId
GROUP BY t.calendarYear, p.prodId;
ALTER TABLE salesMv ADD CONSTRAINT pk_salesmv
PRIMARY KEY (prodId,calendarYear);
```

Exemple d'implémentation en Oracle 10g

- Création des dimensions

```
CREATE MATERIALIZED VIEW location
BUILD IMMEDIATE|DEFERRED
REFRESH FAST|COMPLETE|FORCE ON COMMIT|ON DEMAND
AS SELECT g.ville, g.pays, g.continent
FROM geographie@myRelDB g;
ALTER TABLE location ADD CONSTRAINT pk_location
PRIMARY KEY (ville);
```

- Création des hiérarchies

```
CREATE DIMENSION geo LEVEL n1 is (ville) LEVEL n2
is (pays) LEVEL n3 is (continent) HIERARCHY H1 (n1
CHILD OF n2 CHILD OF n3 )
```

Overview

Démarche de construction d'un ED

- Top-Down
  - Conception de tout l'entrepôt (tous les faits et toutes les dimensions et hiérarchies). Vision très claire et très conceptuelle des données de l'entreprise.
  - Lourd, contraignant mais la plus complète. Demande une connaissance globale de l'entreprise.
- Bottom-Up
  - Approche inverse : créer les étoiles une par une, puis les regrouper jusqu'à obtenir la vision globale de l'entreprise.
  - Méthode simple à réaliser mais travail d'intégration important et redondance possible entre les étoiles.
- Middle-Out
  - Approche hybride, et conseillée par les professionnels du BI.
  - Conception totale de l'entrepôt de données, puis création des divisions plus petites et plus gérables.
  - NB : possibles compromis de découpage (dupliquer des dimensions identiques pour des besoins pratiques).

Modification de l'entrepôt de données

On considère qu'un produit a changé de nom en 2014 (même si cela reste en fait le même produit). On veut continuer à être en mesure de calculer les ventes de ce produit.
Quel est le problème ? Proposez une solution. Donnez un exemple de 'valeurs' pour chacune des tables créées.

Dimensions à évolution lente

Slowly Changing Dimensions (SCDs)

On parle d'une dimension à évolution lente (slowly changing dimension) lorsqu'une dimension peut subir des changements de description des membres. Un client peut changer d'adresse, se marier, ... Un produit peut changer de nom, de formulation "Yaourt à la vanille" en "Saveur Vanille".

Comment gérer cette situation :

- Type 0 - La méthode passive
- Type 1 - Réécriture sur l'ancienne valeur
- Type 2 - Création d'un enregistrement supplémentaire
- Type 3 - Ajour d'une nouvelle colonne
- Type 4 - Utilisation d'une table historique
- Type 6 - Combinaison des approches de type 1,2,3 (1+2+3=6)

Dimensions à évolution lente

Type 0 - La méthode passive

- Pas de prise en compte des SCDs

Dimensions à évolution lente

Type 1 - Réécriture sur l'ancienne valeur :

- La table...

Supplier Key	Supplier Code	SupplierName	Supplier State
123	ABC Acme	Supply Co	**CA**

- ... devient

Supplier Key	Supplier Code	SupplierName	Supplier State
123	ABC Acme	Supply Co	**IL**

- Pas d'historique
- Maintenance facile

Dimensions à évolution lente

Type 2 - Création d'un enregistrement supplémentaire

- Numérotation de version

SupplierKey	SupCode	SupName	SupState	Version
123	ABC	Acme Supply	CA	0
124	ABC	Acme Supply	IL	1

- Plage de validité

SupplierKey	SuprCode	SupName	SupState	StartDate	End Date
123	ABC	Acme Supply	CA	01-Jan-2000	21-Dec-2004
124	ABC	Acme Supply	IL	22-Dec-2004	NULL

Dimensions à évolution lente

Type 3 - Ajour d'une nouvelle colonne

Key	Code	Sup Name	Original Sup State	Effective Date	Current Sup State
123	ABC	Acme Supply	CA	22-Dec-2004	IL

- Ne peut pas suivre l'ensemble des changements

Dimensions à évolution lente

Type 4 - Utilisation d'une table historique

- Supplier

key	Code	Supplier Name	Supplier State
124	ABC	acme Supply	IL

- Supplier History

key	Code	Supplier Name	Supplier State	Create Date
123	ABC	Acme Supply	CA	14-June-2003
124	ABC	Acme Supply Co	IL	22-Dec-2004