

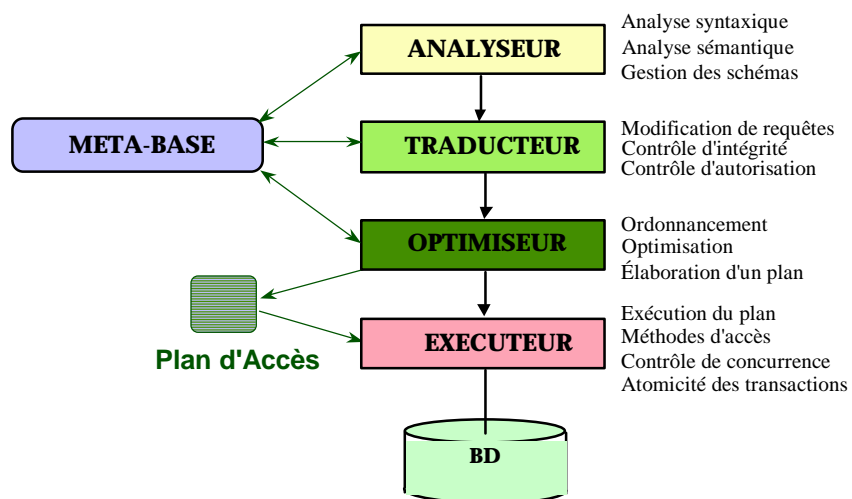
# Vues et Mises à jour

talel@infres.enst.fr

Supports de cours : (1) *Database Management Systems*, R. Ramakrishnan and J. Gehrke, ed. McGrawHill, 2000.  
(2) *Bases de Données*, G. Gardarin, ed. Eyrolles, 2001.

1

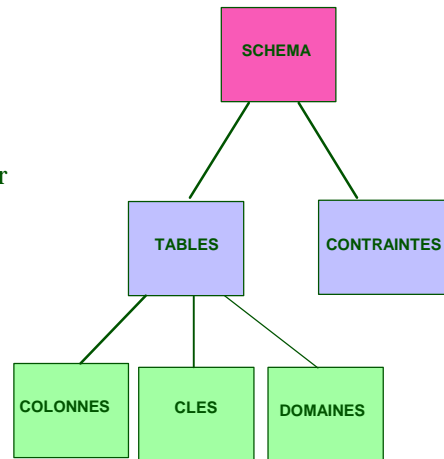
## Interprétation des requêtes



2

## Métabase

- ✗ La métabase (ou catalogue) est gérée comme une base de données relationnelles et contient des informations sur le schéma de chaque base de données
- ✗ SQL2 normalise les tables de la métabase (ou plutôt des vues de ces tables)



3

## Vues (externes)

- ✗ **Objectif**  
Indépendance logique des applications par rapport à la base
- ✗ **Moyen**  
Les vues sont des relations virtuelles dont la définition est stockée dans la métabase.  
Elles sont interrogées et mises à jour comme des relations normales
- ✗ **Avantages/Inconvénients**  
Interrogation efficace  
Mise à jour au travers de vues

4

## *Définition*

### ☞ Vue

- Base de données virtuelle dont le schéma et le contenu sont calculés de la base réelle.

☞ Une vue est donc un ensemble de relations déduites d'une bases de données, par composition des relations de la base.

☞ Par abus de langage, une vue relationnelle est dite table virtuelle

5

## *Création et Destruction*

### ☞ Création

```
CREATE VIEW <NOM DE VUE> [ (LISTE D'ATTRIBUT)]  
AS <QUESTION>  
[WITH CHECK OPTION]
```

La clause WITH CHECK OPTION permet de spécifier que les tuples de la vue **insérés ou mis à jour** doivent satisfaire aux conditions de la question

### ☞ Destruction

```
DROP VIEW <nom de vue>
```

6

## Exemples

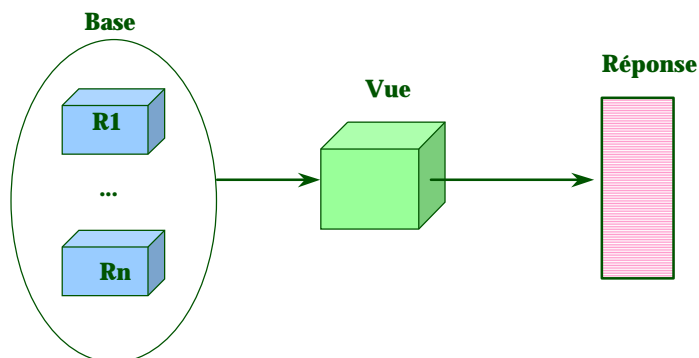
BUVEURS (NB, Nom, Prénom, Adresse, Type)  
VINS (NV, Cru, Région, Millésime, Degré)  
ABUS (NV, NB, Date, Quantité)

- ⌘ (V1) Les vins de Bordeaux :  
CREATE VIEW vinsbordeaux (nv, cru, mill, degré) AS  
SELECT nv, cru, millésime, degré  
FROM vins WHERE Région = "Bordelais"
- ⌘ (V2) Les gros buveurs :  
CREATE VIEW grosbuveurs AS  
SELECT nb, nom, prénom, adresse  
FROM buveurs b, abus a  
WHERE b.nb = a.nb and a.quantité > 10
- ⌘ (V3) Les quantités de vins bues par cru :  
CREATE VIEW vinsbus (cru, mill, degré, total) AS  
SELECT cru, millésime, degré, SUM(quantité)  
FROM vins v, abus a WHERE v.nv = a.nv  
GROUP BY Cru

7

## Interrogation de Vues

- ⌘ Transparence pour l'utilisateur  
manipulées comme des tables de la base
- ⌘ Simplifier les requêtes utilisateurs
- ⌘ Utiles en architecture client-serveur



8

## *Techniques d'évaluation*

### ☞ Réécriture de requête

Peut être effectuée au niveau du source ou par concaténation d'arbres

- Au niveau du source, en remplaçant certaines relations du FROM par leurs sources et en enrichissant les conditions de la clause WHERE pour obtenir le résultat de la question initiale.
- Concaténation d'arbre : mécanisme consistant à remplacer un nœud pendant dans un arbre relationnel par un autre arbre calculant le nœud remplacé.

9

## *Exemple*

### ☞ (1) Question

```
SELECT nom, prénom FROM grosbuveurs  
WHERE adresse LIKE "Paris".
```

### ☞ (2) Définition de vue

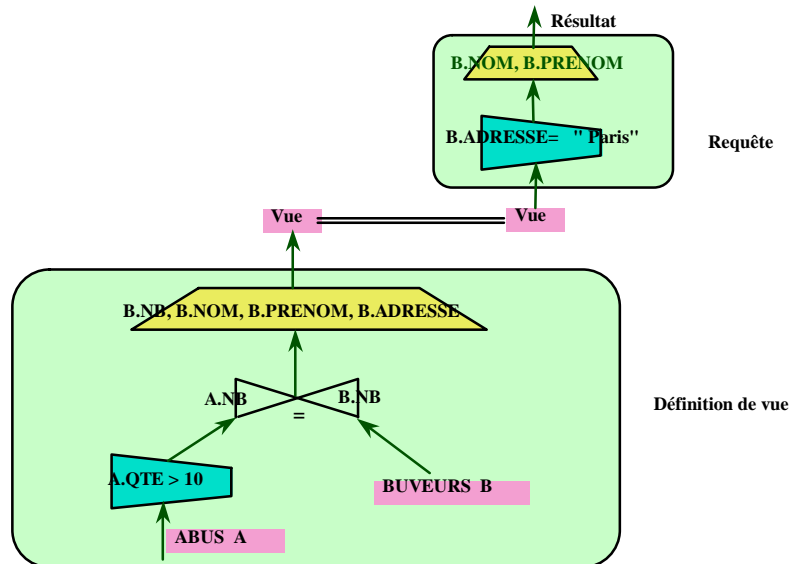
```
CREATE VIEW grosbuveurs AS  
SELECT nb, nom, prénom, adresse FROM buveurs b, abus a  
WHERE b.nb = a.nb AND a.quantité > 10.
```

### ☞ (3) Question modifiée

```
SELECT nom, prénom FROM buveurs b, abus a  
WHERE b.adresse LIKE " Paris" AND b.nb = a.nb AND  
a.quantité > 10.
```

10

## Concaténation d'arbres



11

## Mises à jour de Vues

### ☞ Vue mettable à jour

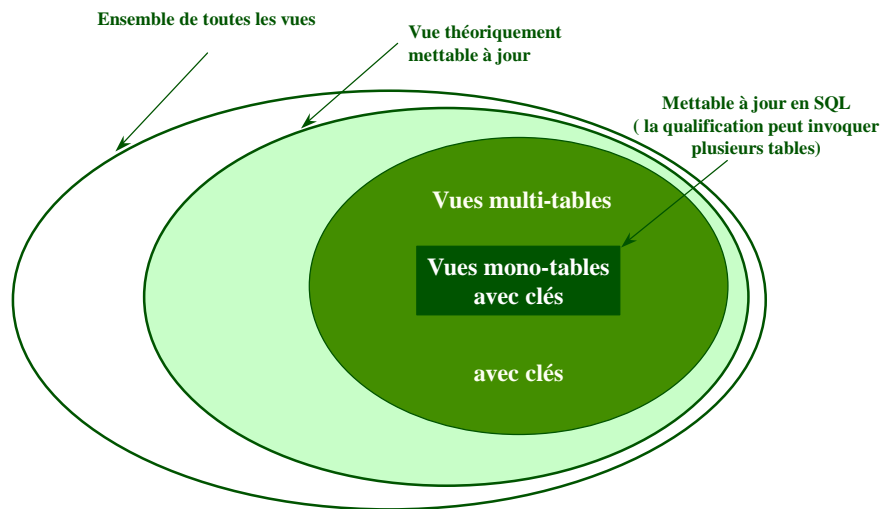
Vue comportant suffisamment d'information pour permettre un report des mises à jour dans la base sans ambiguïté.

### ☞ Conditions suffisantes

- Clés des tables participantes déductibles de la vue
- Vues mono-table avec clé de la table

12

## Classification

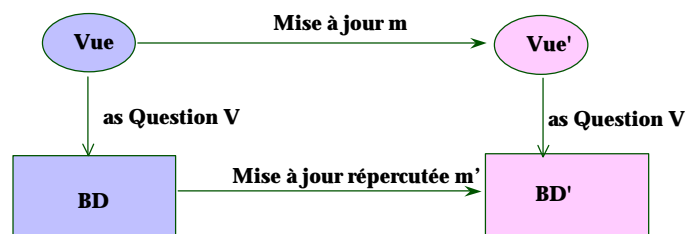


13

## Principe

### ⌘ Problème

- Il peut manquer des données dans la vue pour effectuer le report dans la BD
- Comment définir une stratégie de report cohérente ?



- La mise à jour sur le contenu de la vue doit correspondre au contenu de la vue après le report de la mise à jour sur la BD.

14

## *Exemple de besoin de vues*

### ⌘ **Besoin des entreprises**

- accéder à toutes les données de l'entreprise
- regrouper les informations disséminées dans les bases
- analyser et prendre des décisions rapidement (OLAP)

### ⌘ **Exemples d' applications concernées**

- bancaire : regrouper les infos sur un client
  - ⌘ réponse à ses demandes
  - ⌘ mailing ciblés pour le marketing
- grande distribution : regrouper les infos ventes
  - ⌘ produits à succès
  - ⌘ modes, habitudes d'achat
  - ⌘ préférences par secteurs géographiques

15

## *Entrepôt de données*

### ⌘ **DataWarehouse**

Ensemble de données historisées variant dans le temps, organisé par sujets, consolidé dans une base de données unique, géré dans un environnement de stockage particulier, aidant à la prise de décision dans l'entreprise.

### ⌘ **Trois fonctions essentielles :**

- collecte de données de bases existantes et chargement
- gestion des données dans l'entrepôt
- analyse de données pour la prise de décision

### ⌘ **Vue concrète :**

- idéale pour modéliser un sous-ensemble de données extrait du datawarehouse et ciblé sur un sujet unique

16

## *Vue concrète*

### ☞ Vue concrète

**Table** calculée à partir des tables de la base par une question et matérialisée sur disques par le SGBD.

### ☞ Exemple : vues concrètes avec agrégats de la table

VENTES(NUMV, NUMPRO, NUMFOU, DATE, QUANTITÉ, PRIX)

### ☞ Etendue selon les dimensions PRODUITS et

FOURNISSEURS par les tables :

PRODUITS (NUMPRO, NOM, MARQUE, TYPE, PRIX)

FOURNISSEURS (NUMFOU, NOM, VILLE, RÉGION, TEL.)

17

## *Exemple*

### ☞ Vue des ventes totalisées par produit, fournisseurs, dates

```
CREATE CONCRETE VIEW VENTESPFDF
(numpro,numfou,date,compte,quantot)AS
SELECT numpro,numfou,date,COUNT(*) AS compte,
SUM(quantité) AS quantot
FROM ventes
GROUP BY Numpro, Numfou, Date
```

### ☞ Dérivable de la vue précédente :

```
CREATE CONCRETE VIEW ventespd (numpro, date,
compte,quantot) AS
SELECT numpro, date,SUM(compte)AS compte,SUM(quantot) AS
quantot
FROM ventespdf
GROUP BY numfou
```

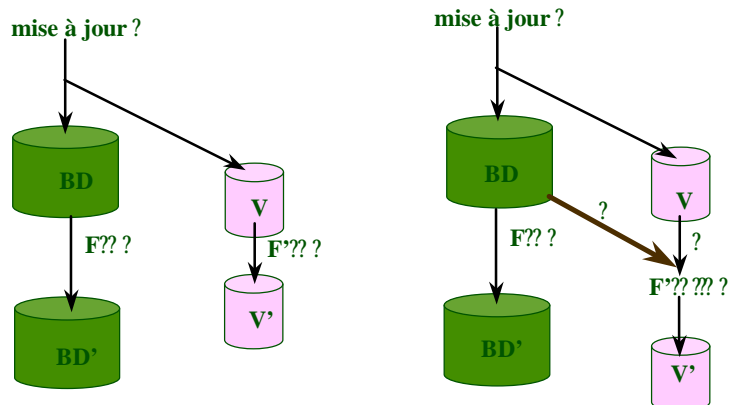
18

## Problème de mise à jour

- ☞ La vue (table) peut être distante
  - on envoie seulement le « delta »
  - les tuples ajoutés et les tuples supprimés
- ☞ Vue auto-maintenable
  - Vue contenant suffisamment d'information pour être mise à jour à partir des ordres de mises à jour effectués sur les relations de base, sans accès aux tuples des relations de la base.
- ☞ Agrégats auto-maintenable
  - Ensemble d'agrégats pouvant être calculés à partir des anciennes valeurs des fonctions d'agrégats et des mises à jour des données de base servant au calcul de la vue.

19

## Vue auto-maintenable ou non



Vue auto-maintenable : lors d'une mise à jour ??  
ce ? suffit pour mettre à jour la vue

Vue non-auto-maintenable : lors d'une mise à jour ?  
celle-là ne suffit pas pour mettre à jour la vue; il faut  
en plus les tuples ?? de la base.

20

## *Conclusion*

- ✍ Techniques uniformes bien maîtrisées
- ✍ Support de l'architecture client/serveur
- ✍ Support des datawarehouse