

Contrôles d'accès aux données

Droits d'accès
Vues

1. CONTROLE DES DROITS D'ACCES

- **Objectif :**
 - protéger les données de la base contre des accès non autorisés
- **Deux niveaux :**
 - connexion au serveur restreinte aux usagers répertoriés (mot de passe)
 - privilège d'accès aux objets de la base
- **Granule d'objet :**
 - vue
 - relation
 - procédure stockée
 - trigger

Hiérarchie des privilèges

Administrateur Système {users login, ...}



Administrateur de la Base {connexion utilisateurs à BD, gestion des schémas}



Propriétaires d'Objets {créateurs d'objets, distribution des droits}



Autres (PUBLIC) {droits obtenus par GRANT}

Définition des droits

GRANT <droits> **ON** <objet>
TO <usagers>
[WITH GRANT OPTION]

<droits> ::= ALL | SELECT | DELETE |
INSERT [<attribut>]| UPDATE [<attribut>]|
REFERENCE [<attribut>]

<objet> ::= TABLE <relation> * | VIEW <vue> *

<usagers> : ::= PUBLIC| <username> *

Exemple:

GRANT SELECT , UPDATE (DEGRE) **ON** TABLE VINS
TO DUPONT

Suppression des droits

```
REVOKE <droits> ON <objet>  
FROM <usagers>
```

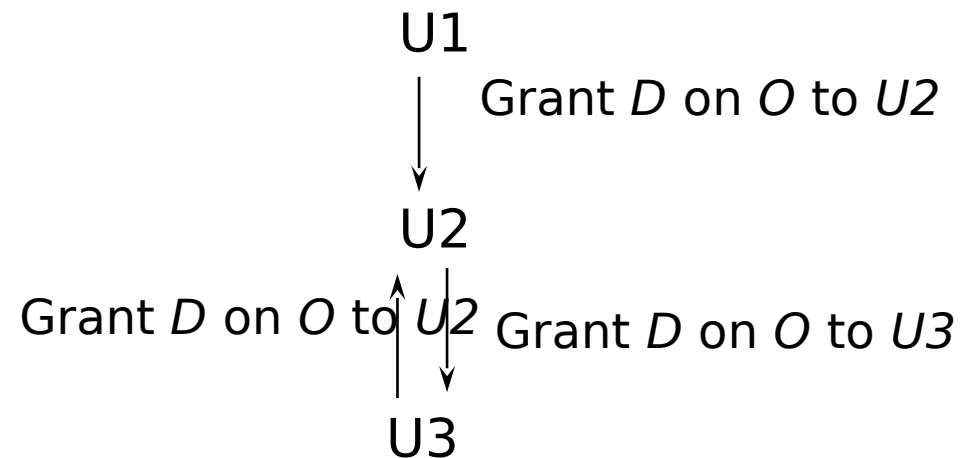
Exemple:

```
REVOKE ALL ON TABLE VINS  
FROM DUPONT
```

Règles d'octroi et de suppression des droits

- On ne peut transmettre que les droits que l'on possède ou qui nous ont été transmis avec "grant option"
- On ne peut supprimer que les droits que l'on a transmis
- La révocation des droits est réursive
- Si un utilisateur U1 a reçu le droit D de la part de plusieurs utilisateurs (U2, U3, ...), il ne perd ce droit que si tous les utilisateurs lui retirent

- Ces règles garantissent la cohérence des retraits même en cas de graphe d'octroi cycliques



Autorisations sur une vue

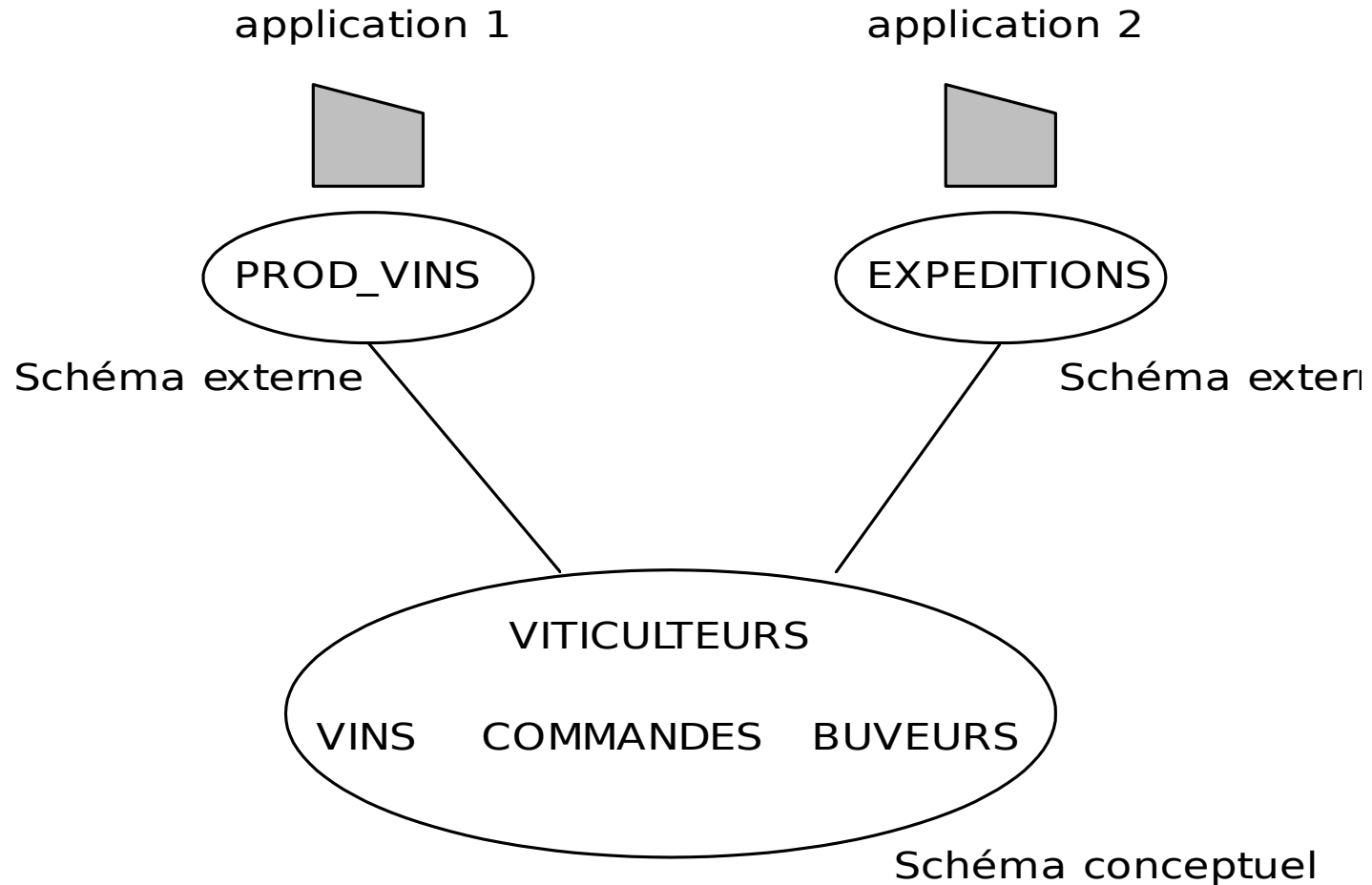
- **Une vue est une relation virtuelle définie à partir des relations de base par une expression SQL**
- **Exemple :**

```
CREATE VIEW GROS_BUVEURS (NB, NOM, QTE_BUE)
AS SELECT B.NB, B. NOM, SUM(A.QTE)
   FROM BUVEURS B, ABUS A
   WHERE B.NB = A.NB
   GROUP BY NB
```
- **AUTORISATIONS SUR LES VUES:**
 - Permet un granule d'autorisation très fin
 - Cependant, les mises à jour au travers des vues sont limitées

2. LES VUES EXTERNES

- **Le concept de vues répond aux besoins suivants :**
 - Adaptation aux applications
 - Intégration des applications existantes
 - Dynamique du schéma de la base
 - Confidentialité et sécurité
 - Décentralisation de l'administration d'une BD
 - Hétérogénéité des modèles
 - Transparence à la localisation (dans le cas de bases réparties)

Exemple (1)



Exemple (2)

- **Schéma conceptuel :**
- **VINS (NV, CRU, ANNEE, DEGRE, NVIT)**
- **VITICULTEURS (NVIT, NOM, PRENOM, VILLE)**
- **BUVEURS (NB, NOM, PRENOM, VILLE)**
- **COMMANDE (NB, NV, QTE, DATE)**
- **on peut définir de nombreuses applications particulières qui n'utilisent qu'une partie des données du schéma.**

Exemple (3)

APPLICATION = INFLUENCE GEOGRAPHIQUE SUR LA CONSOMMATION DE L'ANNEE 1996

- **VUES = RESTRICTION-PROJECTION DES RELATIONS DE BASE**

BUVEURS-1 (NB, VILLE)

ACHETE-1 (NB, NV, QTE, DATE) // restreint aux dates de l'année 96

VINS-1 (NV, CRU)

- **VUE = RESTRUCTURATION DU SCHEMA LOGIQUE**

ACHAT-2 (NB, VILLE, CRU, QTE, DATE)

- **VUE = RESTRUCTURATION et AGREGATION**

CONSOMMATION-PAR-VILLE

CPV (VILLE, CRU, QTE)

Définition des vues

```
CREATE VIEW <nom_vue> [(liste_attributs)]  
AS <expression_de_sélection>  
[WITH CHECK OPTION]
```

- L'expression de sélection peut porter sur des tables de base et/ou des vues
- Dans le cas de vues modifiables, la clause **WITH CHECK OPTION** garantit que les tuples insérés (ou modifiés) dans la vue vérifient bien le critère de la vue

Exemple de définition (1)

```
CREATE VIEW BUVEURS-1  
AS SELECT NB, VILLE  
FROM BUVEURS
```

```
CREATE VIEW ACHETE-1  
AS SELECT NB, NV, QTE, DATE  
FROM COMMANDE  
WHERE DATE BETWEEN '01.01.96' AND '31.12.96'
```

```
CREATE VIEW ACHAT-2 (NB, VILLE, CRU, QTE, DATE)  
AS SELECT B.NB, B.VILLE, V.CRU, C.QTE, C.DATE  
FROM BUVEURS B, VINS V, COMMANDES C  
WHERE C.NB = B.NB AND  
C.NV = V.NV AND  
C.DATE BETWEEN '01.01.96' AND '31.12.96'
```

Exemple de définition (2)

```
CREATE VIEW CPV (VILLE, CRU, QTE)

AS   SELECT  B.VILLE, V.CRU, SUM (C.QTE)

      FROM    BUVEURS B, COMMANDES C, VINS V

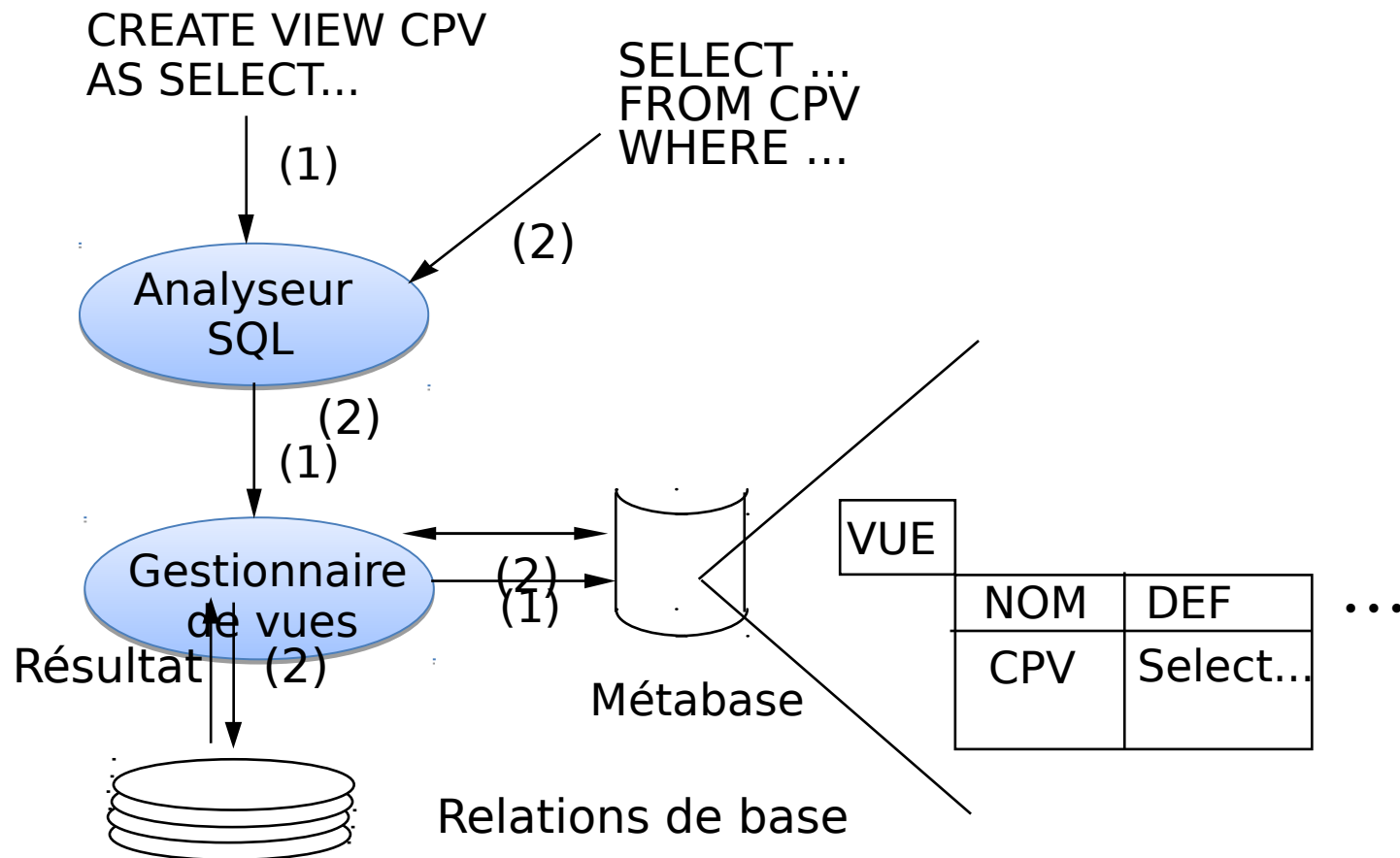
      WHERE   B.NB = C.NB AND

              C.NV = V.NV AND

              C.DATE BETWEEN '01.01.96' AND '31.12.96'

      GROUP BY B.VILLE, V.CRU
```

Déclaration et interrogation



Interrogation des vues (1)

MODIFICATION DE QUESTION PAR RESTRUCTURATION SYNTAXIQUE

EXEMPLE :

```
SELECT CRU, QTE  
FROM CPV  
WHERE VILLE = "Douala"
```

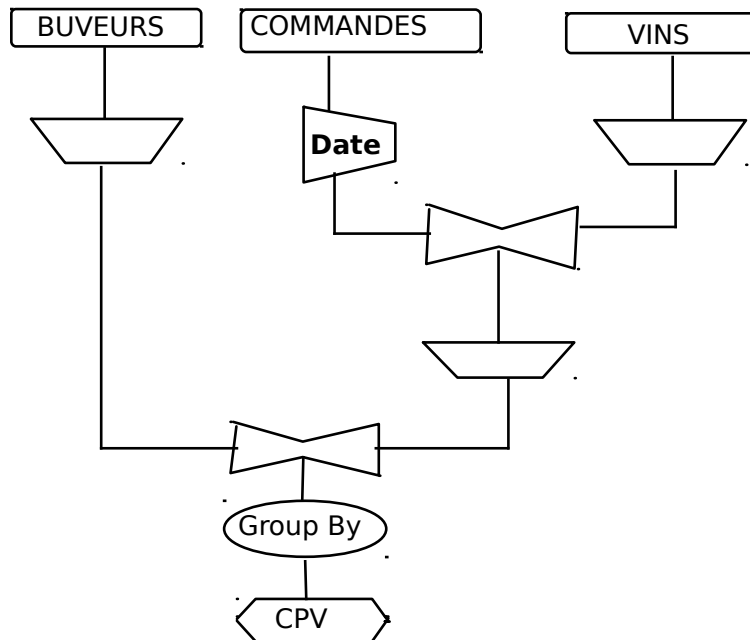
DEVIENT :

```
SELECT V.CRU, SUM (C.QTE)  
FROM BUVEURS B, COMMANDES C, VINS V  
WHERE B.NB = C.NB AND  
C.NV = V.NV AND  
C.DATE BETWEEN '01.01.96' AND '31.12.96' AND  
B.VILLE = 'Douala'  
GROUP BY VILLE, CRU
```

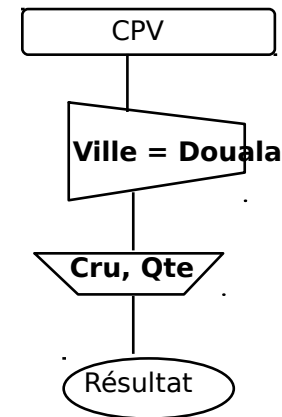

Interrogation des vues (2)

MODIFICATION DE QUESTION PAR RESTRUCTURATION D'ARBRES ALGEBRIQUES

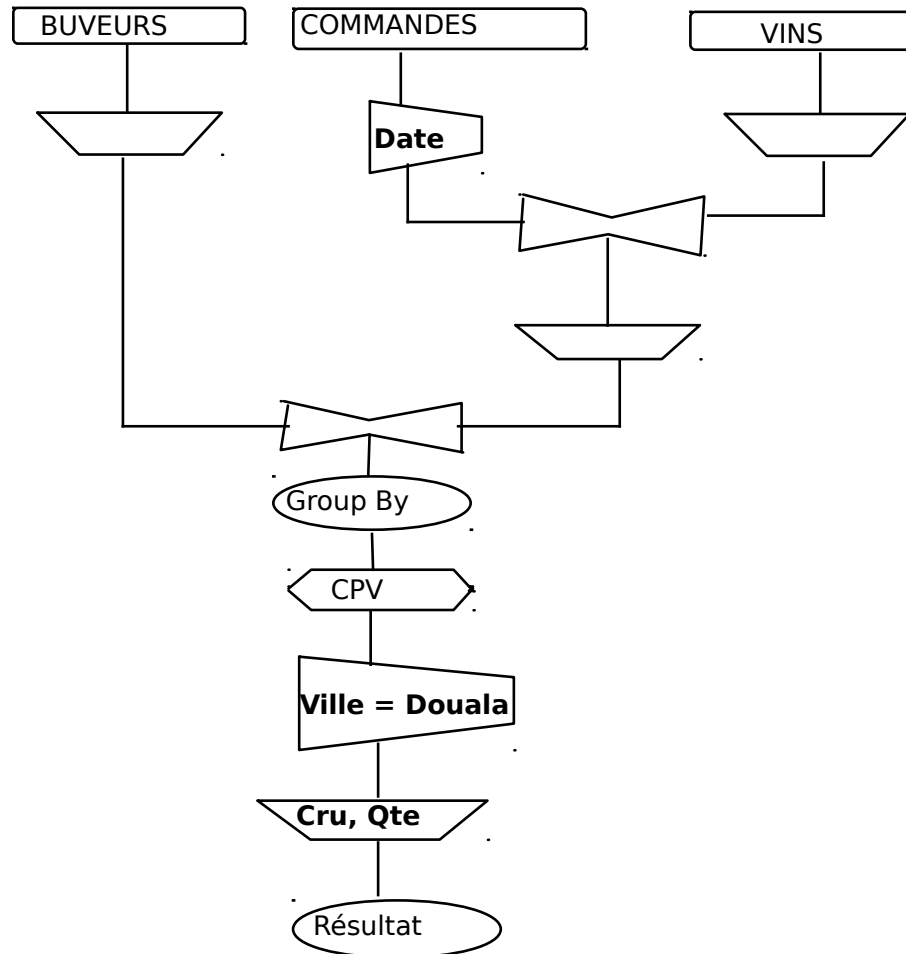
Arbre algébrique de la vue CPV



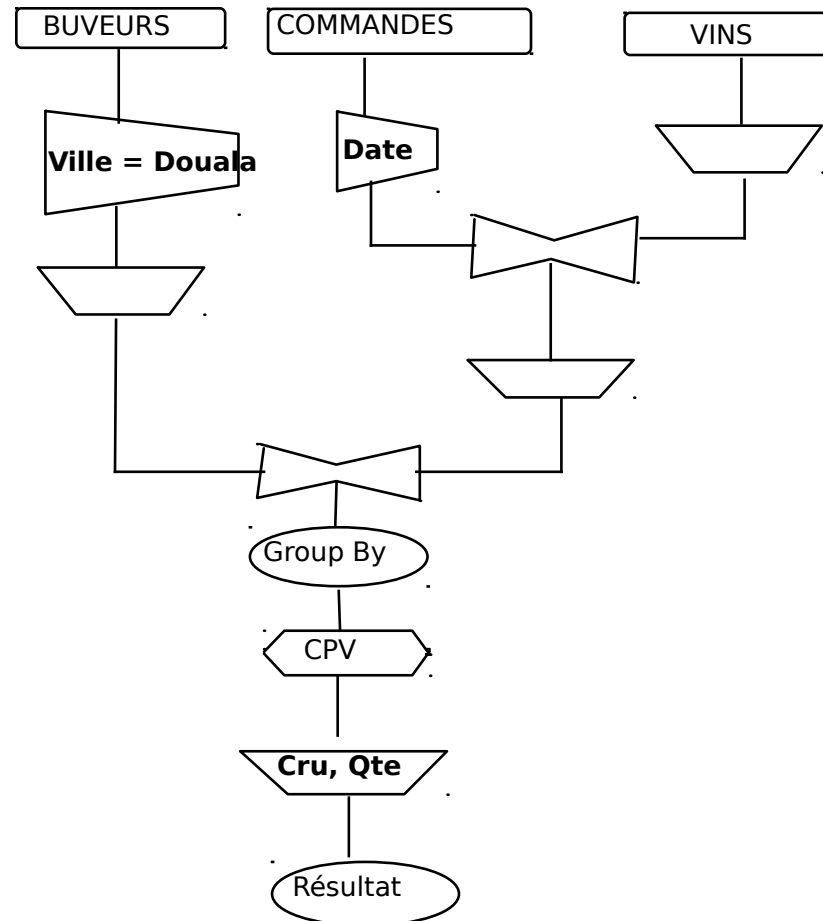
Arbre algébrique de la requête



Les arbres sont mis bout à bout



Puis l'arbre résultat est optimisé



Mises à jour au travers des vues

- **Les mises à jour au travers des vues sont rarement définies**
 - Ex: Comment reporter une mise à jour sur CPV dans les relations de base ?
- **Pour rendre le report de mises à jour possible, la définition de la vue doit respecter certaines contraintes:**
 - la clause SELECT doit conserver les clés de toutes les relations de base
 - la clause SELECT ne doit pas contenir de calcul d'agrégat
 - la définition ne doit pas contenir de clause GROUP BY ni HAVING
- **Dans SQL2, en plus des conditions précédentes, seules les vue définies à partir d'une relation unique seront considérées comme modifiables**

Vues concrètes

- **Vue concrète :**
 - Vue dont on a demandé l'implantation dans la base de données (virtuel ---> réel)
- **Intérêts :**
 - interrogations fréquentes,
 - systèmes répartis
- **Problèmes :**
 - Maj des données de base à répercuter sur la vue
- **L'objectif est d'éviter de réévaluer complètement la vue à chaque mise a jour**

3. CONCLUSION

- **Le modèle relationnel offre**
 - des contraintes d'intégrités riches
 - des droits d'accès intégrés
 - un concept de vue très puissant
 - des méthodes d'interrogation simples et efficaces
- **Problèmes difficiles :**
 - maj au travers des vues
 - vues concrètes