

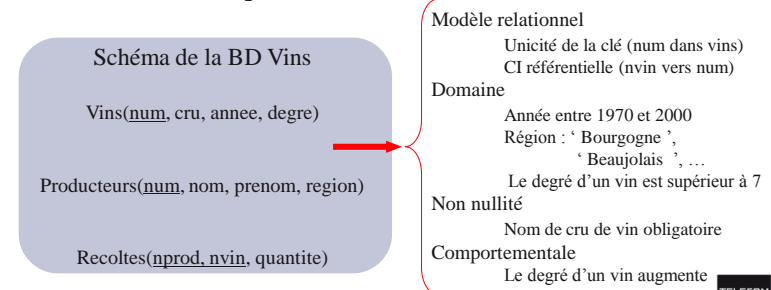
Plan du document

- **Contraintes d'intégrité** slide 177
 - Définition et exemple
 - CIs dans les SGBD
 - Exemple SQL complet
 - Problèmes liés aux CIs
 - Notion de transaction
- **Vues relationnelles** slide 193
- **Gestion des droits** slide 207
- **Conclusion** slide 214
- **Et moi que dois-je faire ?** slide 216

Contraintes d'intégrité

Définition et exemple de contraintes d'intégrité

- Traduction des règles de gestion des données
- Etat cohérent de la base
- Vérifiées en permanence



CI dans les SGBD (1)

■ Comment les exprimer?

- SQL
- Extensions de SQL

■ Quand les déclarer ?

- À la création du schéma de BD : CREATE TABLE
 - À chaque définition d'attribut
 - À la fin de la définition d'une relation
- Au cours de la vie de la BD : ALTER TABLE

CI dans les SGBD (2)

- Non nullité des valeurs d'un attribut
- Unicité de la valeur d'un attribut ou d'un groupe d'attributs
- Valeur par défaut pour un attribut
- Contrainte de domaine
- Clé primaire (un attribut ou un groupe)
- Intégrité référentielle "minimale"

CI dans les SGBD (3)

■ Quelles CI dans les normes

- SQL 86 : unicité, non nullité, vue avec « check option »
- SQL 89 : domaine, clé, intégrité référentielle avec « rejet »
- SQL2 (SQL92) : intégrité référentielle avec « cascade delete » et « cascade UPDATE »

■ Qu'offrent les SGBD ?

- Peu de CI en général
- Autres CI vérifiées par programme

Exemple SQL complet

```
CREATE TABLE Vins (
  num INTEGER PRIMARY KEY,
  cru CHAR (40) NOT NULL,
  annee INTEGER CONSTRAINT Cannee CHECK
    (annee BETWEEN 1970 AND 2010),
  degre NUMBER(4,2) CONSTRAINT Cdegre
    CHECK (degre BETWEEN 9.0AND 15.0)
  ;
```

```
CREATE TABLE Producteurs (
  num INTEGER PRIMARY KEY,
  nom CHAR(40),
  prenom CHAR(40),
  region CHAR(40)
  ;
```

```
ALTER TABLE Producteurs
  ADD CONSTRAINT Cregion CHECK (region IN
    ('Bourgogne', 'Beaujolais', 'Alsace',
    'Jura', 'Corse')) ;
```

```
CREATE TABLE Recoltes(
  nprod INTEGER,
  nvin INTEGER,
  quantite INTEGER)
  ;
```

```
ALTER TABLE Recoltes
  ADD PRIMARY KEY (nprod, nvin) ;
```

```
ALTER TABLE Recoltes
  ADD CONSTRAINT refVIN
  FOREIGN KEY (nvin)
  REFERENCES
  VINS(num) ON DELETE CASCADE ;
```

```
ALTER TABLE Recoltes
  ADD CONSTRAINT refREP
  FOREIGN KEY (nprod)
  REFERENCES
  PRODUCTEURS(num) ON DELETE CASCADE ;
```

Problèmes liés aux CIs

- **Cohérence**
 - Pas de règles contradictoires
- **Redondance**
 - age > 18 et age > 21
- **Optimisation**
 - Limiter au nombre minimal de données mises à jour pour la vérification
 - Vérification sur certains types de mäj uniquement

Plan du document

- **Contraintes d'intégrité** slide 177
 - ...
 - Notion de transaction
 - Gestion des CIs
 - Transaction
 - A
 - C
 - I
 - D
- **Vues relationnelles** slide 19
- **Gestion des droits** slide 216
- **Conclusion** slide 225
- **Et moi que dois-je faire ?** slide 227

Gestion des CIs

- Déclarations de **contraintes d'intégrité**
- Etat **cohérent** de la base
- Vérifiées en **permanence**

Est-ce possible ?



Rien ne se perd,
rien ne se crée



900



Virer 100 €



100 100

Transaction

- Regrouper des opérations élémentaires (lire, écrire)

Virement (cpté source, montant, cpté destinataire)

Dire « je commence »

Enlever montant au cpté source

Ajouter montant au cpté destinataire

Dire « j'ai fini »

Propriétés

A
C
I
D



© jojje11 - Fotolia.com

A ?

Virement (cochon rose, 100, cochon vert)
 Dire « je commence »
 Enlever 100 au cochon rose
 Ajouter 100 au cochon vert
 Dire « j'ai fini »



C ?

Virement (cochon rose, 100, cochon vert)
 Dire « je commence »
 Enlever 100 au cochon rose
 Ajouter 100 au cochon vert
 Ajouter 0,1 a mon compte
 Dire « j'ai fini »



I ?

Virement (cochon rose, 100, cochon vert)
 ① Dire « je commence »
 ③ Enlever 100 au cochon rose
 ⑦ Ajouter 100 au cochon vert
 Dire « j'ai fini »

Virement (cochon rose, 500, cochon vert)
 ② Dire « je commence »
 ④ Enlever 500 au cochon rose
 ⑤ Ajouter 500 au cochon vert
 ⑥ Dire « j'ai fini »



D ?

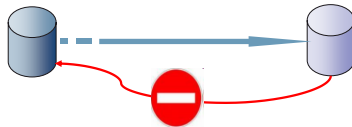
Virement (cochon rose, 100, cochon vert)

Dire « je commence »

Enlever 100 au cochon rose

Ajouter 100 au cochon vert

Dire « j'ai fini »



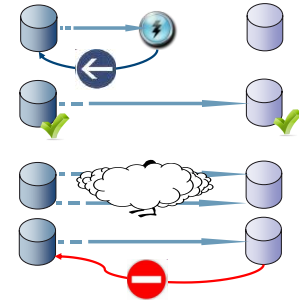
Une transaction est ACID

A : atomicity

C : consistency

I : isolation

D : durability



Plan du document

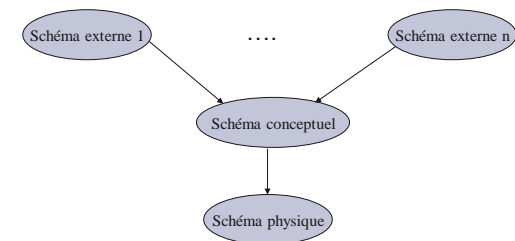
- Contraintes d'intégrité
- Vues relationnelles
 - Rappel d'architecture
 - Exemple de vues
 - Principe
 - Objectif
 - Syntaxe
 - Exemples de création de vues
 - Manipulations de la BD au travers des vues
 - Suppression et renommage
 - Critique des vues
- Gestion des droits
- Conclusion
- Et moi que dois-je faire ?

Rappel d'architecture

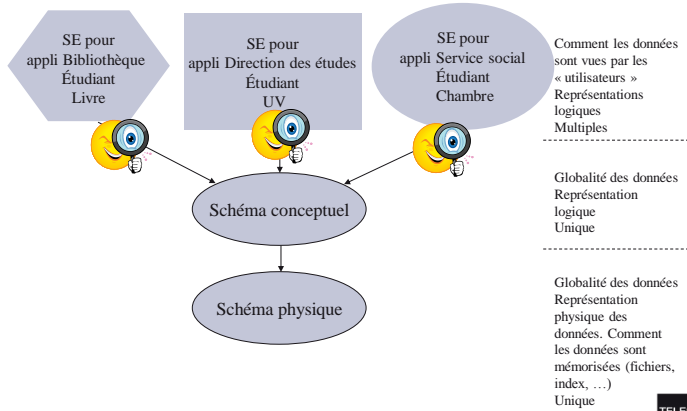
Description des données : 3 niveaux d'abstraction

Vues

■ Groupe ANSI/X3/SPARC (1975)

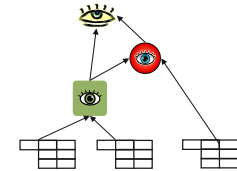


Exemple de vues



Principe

- **Relation virtuelle**
 - Ensemble de tuples qui n'existe pas physiquement
 - Calculable à l'exécution
- **Définie par une requête SQL**
- **Utilisable comme une relation**
- **Utilisable pour définir une autre vue**



Objectifs

- **Indépendance logique**
- **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**

Syntaxe

```
CREATE VIEW nom_vue [{nom_attribut}]
AS requête
```

■ Exemple

```
CREATE VIEW vins_beaujolais (no, nom, degre)
AS SELECT num, cru, degre
FROM vins
WHERE cru='Beaujolais ';
```

Noms différents ...

Exemples

```
CREATE VIEW crus (nom) AS
SELECT DISTINCT cru
FROM vins ;
```

```
CREATE VIEW Clients_beaujolais_paris (num, nom,
qte_cdee) AS
SELECT Cl.num, Cl.nom, sum(qte)
FROM Clients Cl, Commandes C, Vins V
WHERE Cl.num = C.ncli AND C.nvin=V.num
AND V.cru = ' Beaujolais'
AND Cl.ville = 'Paris'
GROUP BY Cl.num, Cl.nom;
```

Renommage
obligatoire

Exemples (2)

Relation : Parent (asc, dsc)

```
CREATE VIEW grand_parent (asc, dsc) as
SELECT P1.asc, P2.dsc
FROM parent P1, parent P2
WHERE P1.dsc = P2.asc;
```

```
CREATE VIEW arr_grd_parent (asc, dsc) AS
SELECT P.asc, GP.dsc
FROM parent P, grand_parent GP
WHERE P.dsc=GP.asc;
```

Manipulation de la BD au travers des vues

■ Consultation

- Toujours possible
- Nom de la vue dans la clause FROM

```
SELECT *
FROM vins_beaujolais;
```

■ Mises à jour

- Rarement possible ...

Mises à jour

- **DISTINCT** ne doit pas être utilisée dans la requête
- **Clause FROM** ne doit contenir qu'une seule relation; si la relation source est une vue, celle-ci doit vérifier cette contrainte
- **Clause SELECT** ne doit pas faire référence à des expressions, ou des fonctions
- **Clause WHERE** ne doit pas contenir de requête imbriquée
- **Requête** ne doit pas contenir de clause **GROUP BY** ni de clause **HAVING**

Mises à jour (2)

```
CREATE VIEW Vins_bordeaux AS
SELECT num, annee, degre
FROM vins
WHERE cru = 'Bordeaux ' ;
```

😊
Tuples identifiables

```
CREATE VIEW deg_moy_par_cru AS
SELECT cru, avg (degre)
FROM vins
GROUP BY cru ;
```

☹️
Augmenter la moyenne des degrés de 1?

Mises à jour (3)

```
CREATE VIEW Clients_beaujolais_paris
(num, nom, qte_cdee) AS
SELECT Cl.num, Cl.nom, sum(qte)
FROM Clients Cl, Commandes C, Vins V
WHERE Cl.num = C.ncli
AND C.nvin=V.num
AND V.cru = 'Beaujolais'
AND Cl.ville = 'Paris'
GROUP BY Cl.num, Cl.nom ;
```

☹️
Ajouter 10 à la quantité commandée par le buveur 10?

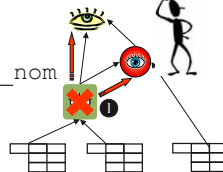
Suppression et renommage

■ Destruction

- DROP VIEW
- Toutes les vues qui utilisent cette vue sont automatiquement détruites
- Relation de base détruite => toutes les vues définies sur cette relation sont automatiquement détruites

■ Renommage

- RENAME ancien_nom TO nouveau_nom



Critiques des vues

😊

- ⊗ Sécurité
- ⊗ Différentes façons de voir la BD
- ⊗ Indépendance logique
- ⊗ Manipulation facile

☹️

- ⊗ Mises à jour impossibles en général

Plan du document

- Contraintes d'intégrité
- Vues relationnelles
- Gestion des droits
 - Techniques pour assurer la confidentialité
 - Concepts SQL
 - Privilèges
 - Syntaxe
 - Rôles
 - Suppression de droit
 - Droits et vues
- Conclusion
- Et moi que dois-je faire ?

Techniques pour assurer la confidentialité

- Définition de contraintes d'autorisation
- Contrôle de flux de données
- Contrôle d'inférence
- Cryptographie

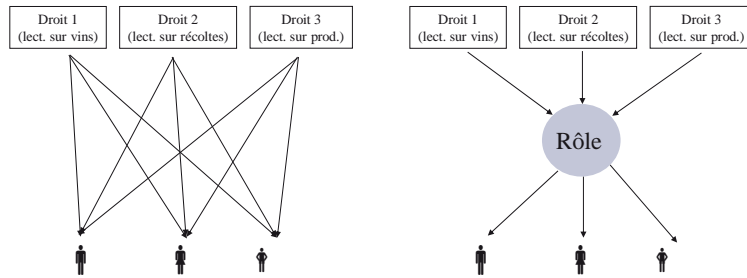
Privilèges

- Créateur d'une relation ou d'une vue = propriétaire
- Peut donner un droit
- Peut autoriser à transmettre un droit
- Droits sur un objet
 - SELECT : consultation
 - DELETE : suppression
 - INSERT : insertion
 - UPDATE : modification

Syntaxe

	defude	lecocq
<pre>GRANT <privilège> ON <objet> TO <utilisateur> [WITH GRANT OPTION];</pre>	<pre>GRANT SELECT ON vins TO lecocq;</pre>	<pre>SELECT * ON defude.vins ;</pre>
	<pre>GRANT SELECT, UPDATE (cru) ON vins TO lecocq, bouzeghoub WITH GRANT OPTION;</pre>	<pre>UPDATE defude.vins SET degre = degre * 1.1 ;</pre>
	<pre>GRANT ALL ON vins TO lecocq;</pre>	<pre>Si DBA défini un alias : SELECT * FROM vins ;</pre>

Rôles



Suppression de droit

```

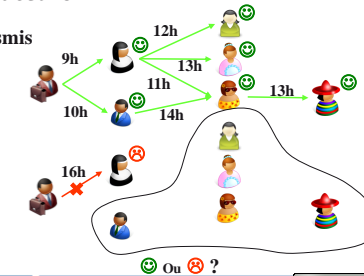
■ REVOKE <privilèges>
ON <objet>
FROM <utilisateurs>

REVOKE ALL
ON vins
FROM defude;

```

■ Retire les droits transmis

- En cascade



Droits et vues

- Donner des droits de lecture non pas sur la relation entière, mais sur une vue basée sur la relation

- => Restreindre l'accès à quelques attributs

```

CREATE VIEW vins_beaujolais AS
SELECT num, cru, degre
FROM vins
WHERE cru='Beaujolais';

GRANT SELECT
ON vins_beaujolais
TO producteurs_beaujolais;

```

- Droits + vues => bonne protection des données

Plan du document

- Contraintes d'intégrité
- Vues relationnelles
- Gestion des droits
- Conclusion
- Et moi que dois-je faire ?

Mécanismes complémentaires

- **Contraintes d'intégrité**
 - Garantissent la cohérence des données
 - Sécurité « sémantique », applicative
- **Vues**
 - Délimitent des frontières d'accès, avec une granularité très fine
 - Masquent la complexité interne
- **Droits**
 - Spécifient les types d'accès, avec un système de droits riche

Page 215

Département INFormatique



Plan du document

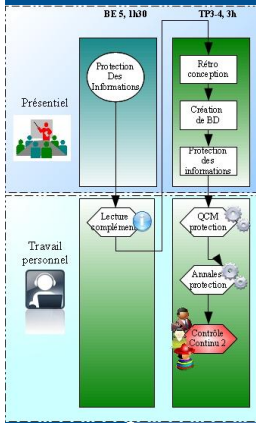
- Contraintes d'intégrité
- Vues relationnelles
- Gestion des droits
- Conclusion
- **Et moi que dois-je faire ?**

Page 216

Département INFormatique



Et moi que dois-je faire ?



- **Relire les transparents**
- **Lire la documentation complémentaire :**
 - Notre cours rédigé : <http://www-inf.it-sudparis.eu/COURS/bd/index.php?idr=45>
 - Cours de Télécom ParisTech : <http://perso.telecom-paristech.fr/~talel/cours/inf225/wwwbd/polyv7/chap6.htm>
- **Approfondir la notion de transaction :**
 - Cours de télécom ParisTech : <http://perso.telecom-paristech.fr/~talel/cours/inf225/wwwbd/polyv7/chap8.htm>
- **Faire le QCM lié à ce cours sur moodle**
- **Pratiquer**
 - ⇨ faire les questions liées à la protection de l'information dans les annales récentes (ex création de table)
 - ⇨ rendre le **contrôle continu 2** en début de CII
- **Mots-fléchés**

Page 217

Département INFormatique

