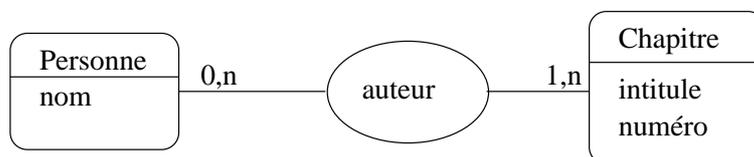


## SGBD B7 - UV 19786 - Cours du soir

### EXAMEN, 1ère Session - 18 Juin 2003

#### 1 Conception de schéma (3 points)

Voici un schéma conceptuel avec des personnes qui sont des auteurs de chapitres de livres :



#### Questions :

- Est-ce que dans ce schéma un chapitre peut avoir plusieurs auteurs ?  
**Solution :** Oui, à cause de la cardinalité 1,n de l'association auteur pour Chapitre.
- Est-ce qu'il peut y avoir des personnes qui ne sont pas des auteurs ?  
**Solution :** Oui, à cause de la cardinalité 0,n de l'association auteur pour Personne.
- Le schéma précédent est incomplet et on voudrait y ajouter des informations concernant les livres qui contiennent les chapitres. Ajoutez les entités, associations et cardinalités manquantes pour décrire les informations et contraintes suivantes :
  - Chaque livre a un titre et un numéro ISBN.
  - Un livre est composé d'au moins un chapitre.
  - Chaque chapitre fait partie d'un seul livre.

#### 2 Langages de Requêtes (6 points)

Un organisme international maintient des statistiques concernant l'éducation dans différents pays. Voici deux tables, **Dépenses** et **Niveau** avec des informations concernant les dépenses et le niveaux d'études de la population pour chaque pays (les attributs qui forment une clé sont soulignés) :

**Dépenses** (pays, degré, duréeMoy, budgetEl)

- pays : le nom du pays
- degré : 1 (primaire), 2 (secondaire), 3 (supérieur)
- duréeMoy : durée moyenne des études
- budgetEl : budget par élève en Euros (sur la durée moyenne des études)

Par exemple, Dépenses('France', 3, 36000, 4.7) signifie qu'en France les dépenses publiques par élève dans l'enseignement supérieur sont de 36000 Euros pour une durée moyenne de 4.7 années d'études.

**Niveau** (pays, degré, âge, taux)

- âge : borne inférieure de la tranche d'âge [age, age + 10]
- taux : pourcentage de la population dans la tranche d'âge indiquée par l'attribut âge qui ont un niveau d'études degré ou supérieur.

Par exemple, Niveau('Autriche', 2, 20, 76) signifie qu'en Autriche 76 pourcents de la population entre 20 et 30 ans ont obtenu un diplôme du degré secondaire.

**Questions :** Écrivez les requêtes suivantes dans les langages indiqués (pour les requêtes en calcul relationnel vous pouvez choisir entre le calcul n-uplet et le calcul domaine) :

- Les noms des pays où la durée moyenne des études supérieures dépasse 5 ans (en algèbre).

**Solution :**

$$\pi_{nom}(\sigma_{duree > 5 \wedge degre = 3} Dépenses)$$

- On cherche pour chaque pays et chaque degré d'études : le nom du pays, les dépenses par élève et le pourcentage des personnes entre 20 et 30 ans qui ont atteint au moins ce degré (SQL et calcul).

**Solution :**

```

select Dépenses.pays, Dépenses.degré, budgetEl, taux
from Dépenses, Niveau
where Dépenses.pays=Niveau.pays
and Dépenses.degre=Niveau.degré
and age = 20
  
```

$$\{P,D,B,T \mid \exists M : Depenses(P,D,M,B) \wedge Niveau(P,D,20,T)\}$$

3. Tous les pays où, pour chaque degré d'études, le budget par élève est supérieur à 30000 Euros (en SQL et algèbre).

**Solution :**

```
select pays
from Dépenses
where pays not in (select pays
                  from Dépenses
                  where budgetEl < 30000)
                 $\pi_{pays}(Depenses) - \pi_{pays}(\sigma_{budgetEl < 30000} Depenses)$ 
```

4. Tous les pays où le budget par élève du secondaire est supérieur à la moyenne sur tous les pays.

**Solution :**

```
select pays
from Dépenses
where degré = 2
  and budgetEl > (select avg(budgetEl)
                  from Dépenses
                  where degré = 2)
```

### 3 Index (2 points)

- Soit le fichier séquentiel suivant (on ne donne pour chaque article du fichier que la clé sur laquelle on construit l'arbre): 1 6 8 15 32 7 9 23 18 17 16 21 29 12 11. Dessinez un arbre B+ d'ordre 2 après l'insertion de toutes les clés dans l'ordre de leur apparition dans le fichier.
- Expliquez la différence entre un index dense et un index non-dense. Quels sont les avantages et les inconvénients de chacun de ces deux types d'index?

**Solution :** voir poly du cours.

### 4 Optimisation (4 points)

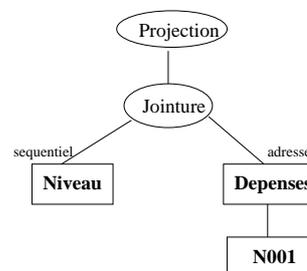
On donne ci-dessous une requête SQL et un plan d'exécution sous sa forme arborescente et EXPLAIN :

**Requête SQL :**

```
select Depenses.pays, budgetEl
from Depenses, Niveau
where Depenses.pays = Niveau.pays
  and Depenses.degre = Niveau.degre
  and Niveau.taux > 80;
```

**Plan d'exécution :**

```
0 SELECT STATEMENT
1 NESTED LOOPS
2 TABLE ACCESS FULL NIVEAU
3 TABLE ACCESS BY INDEX ROWID DEPENSES
4 INDEX RANGE SCAN N001
```



**Questions :**

- Est-ce que ce plan d'exécution utilise un index ? Si oui, il est défini sur quel(s) attribut(s) de quel(s) table(s) ?

**Solution :** Plusieurs réponses sont possibles: Il existe un index sur l'attribut

- pays
- degre
- pays et degre

de la table *Dépenses*

- Expliquer en détail le plan d'exécution (accès aux tables, sélections, jointure, projections)
- On ajoute un index sur l'attribut *taux* de la table **Niveau**. Expliquez les améliorations en terme de plan d'exécution apportées par la création de cet index.

**Solution :** après création d'index le plan est :

```
0 SELECT STATEMENT
  1 NESTED LOOPS
    2 TABLE ACCESS BY INDEX ROWID NIVEAU
      3 INDEX RANGE SCAN NIV_POURC
        4 TABLE ACCESS BY INDEX ROWID DEPENSES
          5 INDEX RANGE SCAN DEP_PAYS
```

- Est-ce qu'on peut encore améliorer ce plan d'exécution en ajoutant un troisième index?

**Solution :** La réponse dépend de la réponse à la première question. Si l'index est défini seulement sur un attribut de la table *Depenses*, il est possible de définir un deuxième index sur l'autre attribut et de faire une jointure d'index.

## 5 Concurrency (5 points)

Une application de commerce électronique utilise une base de données qui stocke des informations sur les produits et sur les commandes des clients. Pour chaque produit, deux enregistrements sont nécessaires : un pour les informations sur le produit (nom et prix) et un autre pour la quantité en stock. Un seul enregistrement est utilisé pour la commande d'un client, qui est construite à partir du contenu du panier du client. Le panier, qui n'est pas stocké dans la base de données, accumule au fur et à mesure la liste des achats du client

Le programme d'achat pour un client a la forme suivante :

Initialisation panier vide

pour chaque achat d'un produit répéter

Demander quel produit acheter et en quelle quantité  
Lecture nom et prix produit  
Calcul ligne d'achat à rajouter au panier  
Rajout ligne au panier  
Mise à jour stock produit

fin répéter

Ecriture de la commande à partir du contenu du panier

Trois opérations s'exécutent en même temps, en tant que transactions dans le système : (i) un client achète le produit A, ensuite le produit B, (ii) un autre client achète seulement le produit B, et (iii) une promotion de 20% de réduction de prix est appliquée au produit B.

- (1 point) L'exécution ci-dessus représente-t-elle une exécution concurrente des trois opérations? Justifiez votre réponse en expliquant la signification de tous les enregistrements et des opérations qui y interviennent.

**H:**  $r_1[x] r_1[y] r_2[u] r_3[u] w_1[y] w_3[u] r_1[u] r_2[v] c_3 w_2[v] w_2[s] r_1[v] w_1[v] c_2 w_1[z] c_1$

**Solution :** Les transactions extraites de **H** sont:

**T<sub>1</sub> :**  $r_1[x] r_1[y] w_1[y] r_1[u] r_1[v] w_1[v] w_1[z] c_1$

**T<sub>2</sub> :**  $r_2[u] r_2[v] w_2[v] w_2[s] c_2$

**T<sub>3</sub> :**  $r_3[u] w_3[u] c_3$

Elles représentent effectivement les trois opérations exécutées.  $x$  contient les informations sur le produit A,  $y$  le stock de produit A,  $u$  les informations sur le produit B,  $v$  le stock de produit B,  $z$  la commande du premier client,  $s$  la commande du second client.

Les séquences lecture-écriture sur  $y$ ,  $v$  et  $u$  sont les mises-à-jour respectivement du stock de produit A, du stock de produit B et du prix du produit B. Les écritures de  $z$  et  $s$  sont les créations des commandes des deux clients à partir des paniers respectifs.

Donc oui, **H** est une exécution concurrente des trois opérations.

- (1 point) Vérifiez si **H** est sérialisable en identifiant les conflits et en construisant le graphe de sérialisation.

**Solution :** Les conflits :

sur  $x$ ,  $y$ ,  $z$ ,  $s$  : pas de conflits

sur  $u$  :  $r_2[u]-w_3[u]$ ,  $w_3[u]-r_1[u]$

sur  $v$  :  $r_2[v]-w_1[v]$ ,  $w_2[v]-r_1[v]$ ,  $w_2[v]-w_1[v]$

Le graphe de sérialisation sera alors :  $T_1 \leftarrow T_2 \rightarrow T_3 \rightarrow T_1$

Il ne contient pas de cycle, donc **H** est sérialisable.

- (1 point) Montrez que l'exécution **H** n'est pas stricte. Est-il possible, en modifiant juste la position des **Commit** de la rendre stricte?

**Solution :** Après  $w_3[u]$ , on a  $r_1[u]$  avant la fin de  $T_3$ , donc **H** n'évite pas les annulations en cascade, donc elle n'est pas stricte non plus. De plus, après  $w_2[v]$ , on a  $r_1[v]$  et aussi  $w_1[v]$  avant la fin de  $T_2$ . On peut éviter tous ces problèmes en déplaçant  $c_3$  juste après  $w_3[u]$  et  $c_2$  juste après  $w_2[s]$ .

4. (2 points) Quelle est l'exécution obtenue par verrouillage à deux phases à partir de  $\mathbf{H}$ ? Quel prix du produit B sera appliqué pour les deux commandes, le prix initial ou le prix réduit?

On considère que le relâchement des verrous d'une transaction se fait au *Commit* et qu'à ce moment on exécute en priorité les opérations bloquées en attente de verrou, dans l'ordre de leur blocage.

**Solution :  $\mathbf{H}$  :**  $r_1[x] r_1[y] r_2[u] r_3[u] w_1[y] w_3[u] r_1[u] r_2[v] c_3 w_2[v] w_2[s] r_1[v] w_1[v] c_2 w_1[z] c_1$

$r_1[x], r_1[y], r_2[u]$  s'exécutent, en prenant les verrous de lecture

$r_3[u]$  partage le verrou de lecture sur  $u$  avec  $r_2[u]$  et s'exécute

$w_1[y]$  s'exécute, pas de conflit avec  $r_1[y]$  (même transaction)

$w_3[u]$  bloquée par  $r_2[u]$ , donc  $T_3$  bloquée

$r_1[u]$  partage le verrou de lecture sur  $u$  avec  $r_2[u]$  et  $r_3[u]$  et s'exécute

$r_2[v]$  prend le verrou de lecture sur  $v$  et s'exécute

$c_3$  bloquée car  $T_3$  bloquée

$w_2[v]$  s'exécute, pas de conflit avec  $r_2[v]$  (même transaction)

$w_2[s]$  prend le verrou d'écriture sur  $s$  et s'exécute

$r_1[v]$  bloquée par  $w_2[v]$ , donc  $T_1$  bloquée

$w_1[v]$  bloquée également, car  $T_1$  bloquée

$c_2$  s'exécute et relâche les verrous de  $T_2$   $\rightarrow w_3[u]$  reste bloquée à cause du verrou de  $r_1[u]$ , donc  $T_3$  aussi, par contre  $r_1[v]$  et  $w_1[v]$  peuvent prendre les verrous sur  $v$  et s'exécuter, donc  $T_1$  n'est plus bloquée

$w_1[z]$  prend le verrou d'écriture sur  $z$  et s'exécute

$c_1$  s'exécute et relâche les verrous de  $T_1$   $\rightarrow$  toutes les opérations de  $T_3$  bloquées peuvent prendre les verrous et s'exécuter :  $w_3[u]$  et ensuite  $c_3$ .

Le résultat final est donc  $\mathbf{H}'$  :  $r_1[x] r_1[y] r_2[u] r_3[u] w_1[y] r_1[u] r_2[v] w_2[v] w_2[s] c_2 r_1[v] w_1[v] w_1[z] c_1 w_3[u] c_3$

Le prix du produit B n'est modifié qu'à la fin de l'exécution  $\mathbf{H}'$ , donc les deux commandes se feront avec le prix initial.