

NFP107 et NFP107J
EXAMEN, 2ème session – septembre 2006
(tout document écrit autorisé)

1 Modèle relationnel (8 points)

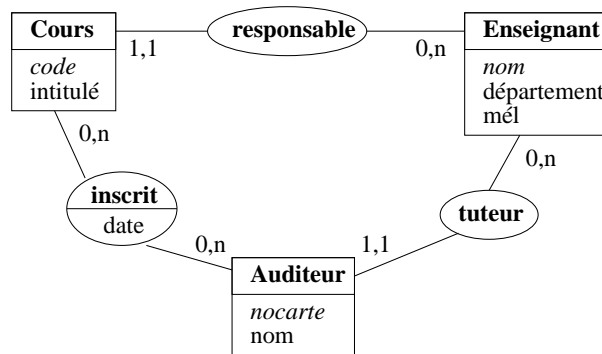
Un système de gestion des inscriptions des auditeurs en cours utilise le schéma relationnel suivant :

- Cours(code, intitulé, responsable) ;
- Enseignant(nom, département, mél) ;
- Auditeur(nocarte, nom, tuteur) ;
- Inscription(nocarte, cours, date) ;

Les clés primaires sont soulignées. Un cours est identifié par son code, il a un intitulé et un enseignant responsable. Un enseignant est identifié par son nom et il a un département et une adresse mél. Un auditeur, identifié par son numéro de carte (nocarte), a un nom et un enseignant qui est son tuteur. Une inscription est identifiée par un couple nocarte auditeur - code cours et a une date.

Modèle conceptuel : (2 points) Donnez un schéma entité-association qui correspond à ce schéma relationnel, en faisant bien apparaître les multiplicités des associations.

Solution :



Requêtes : Exprimez dans chaque cas en algèbre et en SQL les requêtes suivantes sur le schéma précédent :

1. (2 points) Intitulé des cours où l'auditeur Zambon a été inscrit en 2000 ;

Solution : SQL :

```

select intitulé
from Auditeur, Inscription, Cours
where date=2000 and
      nom='Zambon' and
      Inscription.nocarte=Auditeur.nocarte and
      Inscription.cours=Cours.code;
    
```

Algèbre :

$$\Pi_{intitulé}(\sigma_{nom='Zambon'}(Auditeur) \bowtie (\sigma_{date=2000}(Inscription) \bowtie_{cours=code} Cours)).$$

2. (2 points) Noms des enseignants qui ne sont tuteurs d'aucun auditeur ;

Solution : SQL :

```

select nom
from Enseignant
where nom not in (select tuteur from Auditeur);
    
```

Algèbre :

$$\Pi_{nom}(Enseignant) - \Pi_{tuteur}(Auditeur).$$

3. (2 points) Mél des enseignants qui sont responsables d'un cours où sont inscrits en 2006 des auditeurs pour lesquels ces enseignants sont tuteurs ;

Solution : SQL :

```
select mél
from  Inscription,Cours,Auditeur,Enseignant
where date = 2006 and cours=code and Inscription.nocarte=Auditeur.nocarte and
      tuteur=responsable and tuteur=Enseignant.nom;
```

Algèbre :

Solution 1 : $\Pi_{mél}(Enseignant \bowtie_{nom=responsable} (\Pi_{responsable,nocarte}(\sigma_{date=2006}(Inscription)) \bowtie_{cours=code} Cours) \cap \rho_{tuteur \rightarrow responsable}(\Pi_{nocarte,tuteur}(Auditeur)))$

Solution 2 : $\Pi_{mél}(\sigma_{nome=responsable}(\rho_{nom \rightarrow nome}(Enseignant) \bowtie_{nome=responsable} (Cours \bowtie_{code=cours} (\sigma_{date=2006}(Inscription)) \bowtie Auditeur)))$

2 Organisation physique et Optimisation (6 points)

Soit l'ordre SQL :

```
select mél from Enseignant,Cours
where intitulé='BD' and nom=responsable;
```

1. (0,5 point) Que calcule cette requête (en français) ?

Solution : *les méls de responsables de cours intitulés 'BD'*

2. (2 points) Les deux tables sont indexées sur leurs clés seulement (Enseignant sur nom et Cours sur code). Donner un plan d'exécution physique (Explain ou arbre) et l'*expliquer en détail*.

Solution :

Plan d'exécution

```
-----
0      SELECT STATEMENT Optimizer=RULE
1      0      NESTED LOOPS
2      1      TABLE ACCESS (FULL) OF 'COURS' (TABLE)
3      1      TABLE ACCESS (BY INDEX ROWID) OF 'ENSEIGNANT' (TABLE)
4      3      INDEX (UNIQUE SCAN) OF 'SYS_C0013137' (INDEX (UNIQUE))
```

Jointure par boucle imbriquée avec traversée d'index : parcours séquentiel de la table Cours, sélection sur intitulé, pour chaque cours d'intitulé BD, la valeur de l'attribut responsable sert de clé d'accès à l'index sur le nom de la table Enseignant. La traversée de l'index donne les rowid des nuplets d'Enseignant. Après accès par rowid à ces nuplets, projection sur les méls.

3. (2 points) On ajoute un index sur l'intitulé. Donner le nouveau plan d'exécution physique.

Solution :

```
SQL> create index cours_idx on COURS(intitule);
Index cree.
```

Plan d'exécution

```
-----
0      SELECT STATEMENT Optimizer=RULE
1      0      NESTED LOOPS
```

```

2      1      TABLE ACCESS (BY INDEX ROWID) OF 'COURS' (TABLE)
3      2      INDEX (RANGE SCAN) OF 'COURS_IDX' (INDEX)
4      1      TABLE ACCESS (BY INDEX ROWID) OF 'ENSEIGNANT' (TABLE)
5      4      INDEX (UNIQUE SCAN) OF 'SYS_C0013137' (INDEX (UNIQUE))

```

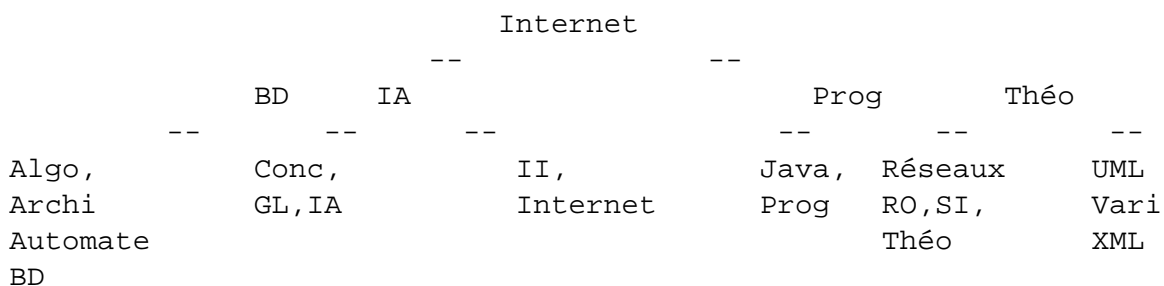
On remplace le balayage séquentiel de Cours par une traversée de l'index sur les intitulés qui donne les rowid des nuplets cours accédés par rowid. Pour chacun d'eux faire comme dans le plan précédent

4. (1,5 points) Cet index est implanté par un arbre B+ d'ordre 2 vu en cours (ordre lexicographique ascendant). Voici la liste des intitulés :

Algorithmique, BD, Automates, Programmation, Vari, Réseaux, Architecture, Java, Internet, XML, Théorie, SI, UML, Génie logiciel, Informatique industrielle, Informatique artificielle, Recherche opérationnelle, Conception

Donner l'arbre correspondant sous la forme vue en cours.

Solution :



3 Concurrence (6 points)

Les trois opérations suivantes s'exécutent dans le système de gestion des inscriptions :

- *Nouvel_enseignant*(nom, dept, mél, nocarte), qui rajoute à la relation Enseignant un enseignant [nom, dept, mél] et le marque comme tuteur de l'auditeur identifié par nocarte.
- *Modif_tuteur_inf*(nocarte, nom-tuteur), qui modifie le tuteur de l'auditeur identifié par nocarte avec nom-tuteur, seulement si l'ancien tuteur n'est pas au département informatique.
- *Auditeurs_tuteur*(nom-tuteur), qui trouve les noms de tous les auditeurs qui ont pour tuteur nom-tuteur.

1. (1,5 points) Montrer que l'exécution $\mathbf{H} : r_1[x]r_2[y]r_1[z]w_3[u]r_3[x]w_1[x]c_1w_3[x]c_2c_3$ peut représenter une exécution concurrente des trois opérations ci-dessus, en donnant la signification de x, y, z, u .

On considère que la lecture de tous les nuplets d'une relation est représentée par une seule lecture de la relation. Aussi, l'insertion d'un nuplet dans une relation est représenté par une écriture de la relation. \mathbf{H} contient donc à la fois des opérations sur des nuplets et sur des relations. On considère également que l'accès à un nuplet dont on connaît la clé est direct.

Solution : Les transactions de \mathbf{H} :

$T_1 : r_1[x]r_1[z]w_1[x]c_1$

$T_2 : r_2[y]c_2$

$T_3 : w_3[u]r_3[x]w_3[x]c_3$

T_2 n'a qu'une lecture, c'est donc *Auditeurs_tuteur*, qui lit toute la relation Auditeur, donc $y =$ relation Auditeur.

T_3 commence par une écriture, c'est donc *Nouvel_enseignant*, qui insère un nouvel enseignant, ensuite met à jour l'auditeur avec le nouveau tuteur (lecture-écriture auditeur). Donc $u =$ relation Enseignant, $x =$ un nuplet auditeur.

T_1 est donc *Modif_tuteur_inf*, avec lecture de l'auditeur, lecture de l'enseignant tuteur pour vérifier son département et écriture auditeur, donc $z =$ nuplet enseignant.

2. (1,5 points) Donner l'exécution de \mathbf{H} par verrouillage hiérarchique, en considérant que les verrous sont relâchés au Commit et que les opérations bloquées en attente de verrou s'exécutent en priorité quand le verrou devient disponible, en respectant l'ordre de blocage.

Solution : $r_1[x]$ s'exécute en prenant le verrou S sur le nuplet auditeur x et IS sur la relation Auditeur y
 $r_2[y]$ prend le verrou S sur la relation Auditeur y (pas de conflit $S-IS$) et s'exécute
 $r_1[z]$ prend le verrou S sur le nuplet enseignant z et IS sur la relation Enseignant u et s'exécute
 $w_3[u]$ est bloquée, car ne peut pas prendre X sur u (conflit $X-IS$), donc T_3 bloquée
 $r_3[x]$ bloquée, car T_3 bloquée
 $w_1[x]$ ne peut pas prendre IX sur la relation Auditeur y (conflit $IX-S$), donc T_1 bloquée
 c_1 bloquée, car T_1 bloquée
 $w_3[x]$ bloquée, car T_3 bloquée
 c_2 s'exécute et relâche le verrou S sur $y \rightarrow w_1[x]$ peut s'exécuter, ainsi que $c_1 \rightarrow$ toute la transaction T_3
s'exécute $w_3[u]r_3[x]w_3[x]c_3$
Résultat : $r_1[x]r_2[y]r_1[z]c_2w_1[x]c_1w_3[u]r_3[x]w_3[x]c_3$

3. (1,5 points) Donner l'exécution de \mathbf{H} par verrouillage simple, en considérant que x, y, z, u se trouvent au même niveau de granularité. Les mêmes conditions qu'au point précédent s'appliquent pour le relâchement de verrous et le déblocage prioritaire des opérations en attente.

Solution : $r_1[x], r_2[y], r_1[z]$ s'exécutent en prenant les verrous de lecture sur x, y, z
 $w_3[u]$ s'exécute en prenant le verrou d'écriture sur u
 $r_3[x]$ partage le verrou de lecture avec $r_1[x]$ et s'exécute
 $w_1[x]$ bloquée par $r_3[x]$, donc T_1 bloquée
 c_1 bloquée, car T_1 bloquée
 $w_3[x]$ bloquée par $r_1[x]$, donc T_3 bloquée
Attente circulaire entre T_1 et $T_3 \rightarrow$ interblocage

4. (1,5 points) En considérant toujours que x, y, z, u se trouvent au même niveau de granularité, vérifier si \mathbf{H} est sérialisable, en identifiant les conflits et en construisant son graphe de sérialisation. \mathbf{H} est-elle stricte ?

Solution : Conflits : sur $x : r_1[x] - w_3[x], r_3[x] - w_1[x], w_1[x] - w_3[x]$

Graphe de sérialisation : $T_1 \xrightarrow{\text{c}} T_3$.

Le graphe contient un cycle entre T_1 et T_3 , donc \mathbf{H} n'est pas sérialisable.

Seule l'écriture $w_1[x]$ est suivie d'une autre opération sur la même donnée ($w_3[x]$). Mais comme au moment de $w_3[x]$, $w_1[x]$ est déjà validée, \mathbf{H} est stricte.