Nom :	Prénom :	*page 1*

UPMC - UFR 919 - Master d'informatique - M1

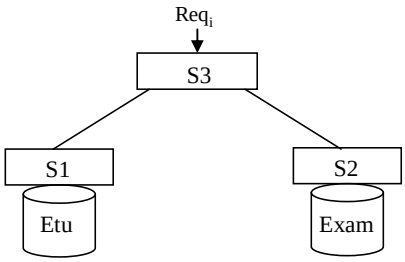
Bases de Données Réparties
Examen du 4 juin 2009

Les documents de cours, TD et TME sont autorisés – Durée : 2h.

Répondre aux questions sur la feuille du sujet dans les cadres appropriés. La taille des cadres suggère celle de la réponse attendue. Utiliser le dos de la feuille précédente si la réponse déborde du cadre. Le barème est donné à titre indicatif. La qualité de la rédaction sera prise en compte. Ecrire à l'encre bleue ou noire. Ne pas dégrafer le sujet.

Exercice 1 : Optimisation de requêtes réparties — 10 pts

On considère l'architecture suivante :



On a **Etu** (<u>d</u>, nom, prénom) sur S1,
Exam (<u>d, épreuve</u>, note) sur S2.

L'attribut *d* de Etu et Exam est le numéro de dossier d'un étudiant. Les attributs *d* et *note* sont des entiers positifs. La distribution des valeurs des attributs est uniforme.
Soit t(a) la taille en octets d'un attribut, on a :

t(d) = 10, t(nom) = t(prenom) = 45, t(epreuve) = 30, t(note) = 10

Soit T(R) la taille d'une relation R, exprimée en millions (10^6) d'octets. On a les valeurs suivantes :

T(Etu) = 1 T(Exam) = 5

Toutes les requêtes Req$_i$ sont posées sur S3 (leur résultat est affiché sur S3).
Le **coût** d'un plan d'exécution d'une requête est la **somme des transferts** de données, mesuré en millions d'octets.

Question 1

Soit la requête Req1

Req1 : select e.d, nom, prénom, épreuve, note
from Etu e, Exam x
where e.d = x.d

On considère les plans d'exécution de Req1 suivants :

P1 : transférer les deux relations sur S3 puis traiter la jointure sur S3.
P2 : transférer Etu sur S2, puis traiter la jointure sur S2, puis transférer le résultat sur S3.
P3 : transférer Exam sur S1, puis traiter la jointure sur S1, puis transférer le résultat sur S3.

a) Quelle est l'expression algébrique de la requête Req1

b) Quelle est la cardinalité de Etu?

c) Quelle est la cardinalité de la requête Req1 ?

d) Combien vaut T(Req1) ?

e) Quel est le coût de P1 ?

P1

f) Quel est le coût de P2 ?

P2

g) Quel est le coût de P3 ?

P3

h) En moyenne, combien d'épreuves a passé un étudiant ? Donner la valeur.

Question 2

On veut connaître le nom des étudiants dont la note totale est inférieure à 100. On sait que 1% des étudiants sont dans ce cas (on suppose qu'il s'agit d'une université d'excellence).

Soit la requête Req2

Req2 : select e.d, nom, prénom
from Etu e, Exam x
where e.d = x.d
group by e.d
having sum(note) < 100

Soit la sous-requête T2 :

Select d, sum(note) as total
From Exam
Group by d

On considère le plan d'exécution suivant pour traiter Req2 :

P4 : Etape 1) transférer Etu sur S3,
Etape 2) traiter T2 sur S2 et transférer son résultat sur S3
Etape 3) finir le traitement sur S3.

a) Quelle est la taille (en octets) d'un nuplet de T2 ?

b) Combien vaut card(T2) ?

c) Combien vaut T(T2) ?

d) Pendant l'étape 3 du plan P4, une requête est traitée sur S3 à partir des données provenant de Etu et T2 afin d'obtenir le résultat de Req2. Exprimer cette requête en SQL (la clause FROM doit mentionner Etu et T2).

e) Quel est le coût de P4 ?

f) Décrire le plan d'exécution P5 de Req2 qui **minimise** les transferts. Pour chaque étape, préciser le site, la requête traitée et les transferts.

P5

g) On suppose maintenant que S3 est une application écrite en java et connectée aux bases S1 et S2 par JDBC. On dispose pour cela des objets de type *Connection* nommés connectS1 et connectS2.

Compléter la méthode *traiteRequete* du programme java (exécuté sur S3) qui traite Req2 en minimisant les transferts (selon le plan P5) et affiche son résultat.

Remarque : si nécessaire, JDBC permet à une application de transmettre des données à destination d'un SGBD, en utilisant des instructions SQL *insert*.

```
public void traiteRequete() {
       Connection connectS1 = DriverManager.getConnection(…); // accès à S1
       Connection connectS2 = DriverManager.getConnection(…); // accès à S2
```

}

Exercice 2 : Conception de bases de données réparties 10 pts

Question 1. Trois universités parisiennes de prestige (Jussieu, Sorbonne, Dauphine) ont décidé de mutualiser leurs bibliothèques et leur service de prêts, afin de permettre à l'ensemble des étudiants d'excellence d'emprunter des ouvrages dans toutes les bibliothèques des universités participantes. Par exemple, un étudiant de Jussieu pourra emprunter des ouvrages à la bibliothèque de la Sorbonne, en se connectant sur le site de la Sorbonne. Pour cela, il lui faudra verser une certaine somme : une provision est prévue et ajoutée dans les droits d'inscription. Pour compenser les efforts financiers faits par une université concurrente, cette somme sera plus importante si le prêt se fait dans une autre université que celle où est inscrit l'étudiant. Lorsqu'il emprunte dans sa propre université, le tarif est moins cher.

La gestion commune des bibliothèques et des emprunts est effectuée par une base de données répartie, dont le schéma global est le suivant :

EMPLOYE (Id_pers, nom, adresse, statut, affectation)

L'attribut affectation désigne ici l'université où travaille l'employé.

ETUDIANT (Id_etu, nom, adresse, université, cursus, reste_provision)

L'attribut université indique l'université où est inscrit l'étudiant. L'attribut reste_provision indique la somme restante dont l'étudiant dispose pour emprunter des livres. Lorsque cette somme est insuffisante, l'étudiant doit alors effectuer de nouveaux versements à son université pour continuer à pouvoir emprunter.

OUVRAGES (Id_ouv, titre, éditeur, année, domaine, stock, site)

L'attribut site indique l'université dont la bibliothèque gère cet ouvrage. L'attribut domaine permet de classer les ouvrages en catégories (physique, maths, informatique, médecine, etc.). L'attribut stock désigne le nombre d'ouvrages restant disponibles au prêt

TARIF (université, tarif_entreprise, tarif_concurrence)

Tarif_entreprise indique le coût de l'emprunt pour un étudiant de l'université correspondant à l'attribut université. Tarif_concurrence indique le coût pour un étudiant d'une autre université.

AUTEURS (Id_ouv, nom_auteur)

PRETS (Id_ouv, Id_etu, date_emprunt, date_retour)

La gestion de cette application s'appuie sur les hypothèses suivantes :

- un employé est affecté à un seul site
- un étudiant est inscrit dans une seule université, mais peut emprunter dans toutes les bibliothèques.
- un ouvrage emprunté dans une bibliothèque est rendu dans la même bibliothèque.
- Pour garantir le paiement entre universités, le champ reste_provision de la relation ETUDIANT est mis à jour lors de chaque emprunt, quelle que soit la bibliothèque d'emprunt.
- Chaque université d'excellence gère évidemment ses propres étudiants
- Chaque bibliothèque gère son personnel et les ouvrages qu'elle détient.

Les relations globales sont fragmentées et réparties sur les différents sites.

Donner la définition des différents fragments en utilisant les opérateurs de l'algèbre relationnelle ainsi que le schéma d'allocation des fragments.

Question 2. Donner les opérations de reconstruction des relations globales