

<b>Nom :</b>	<b>Prénom :</b>	<i>page 1</i>
--------------	-----------------	---------------

Université Pierre et Marie Curie - Master d'informatique (SAR)

## Bases de Données Réparties

### Examen du 20 mai 2008

Elements de correction

Les documents ne sont pas 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 : Questions diverses

**3 pts**

**Question 1.** Expliquez pourquoi la plupart des SGBD n'utilisent pas le gestionnaire de cache du système d'exploitation sous-jacent, mais ont leur propre gestionnaire de cache. Quelles sont les fonctionnalités spécifiques de ces gestionnaires de cache ? A quoi servent-elles ?

Le gestionnaire de cache des SE utilise une stratégie de maintien des pages en mémoire et de remplacement de celles-ci lorsqu'il n'y a plus de place qui permet d'optimiser l'espace mémoire en général, mais qui ne convient pas aux SGBD, notamment pour la reprise sur pannes. Les SGBD ont donc leur propre gestionnaire de cache, qui permet de décider eux-mêmes quand une page doit être réécrite, ou qui permet aussi d'empêcher une page d'être écrite sur le disque. Cela permet de gérer les écritures sur le journal. Il y a alors plusieurs stratégies (Fix, flush) qui conditionnent les algorithmes de reprise. On peut ainsi éviter d'utiliser l'algo UNDO ou REDO, selon les stratégies.

**Question 2.** Quelle est la différence entre réplication synchrone et réplication asynchrone ? Donnez les avantages et les inconvénients de ces deux techniques.

Ces deux techniques diffèrent sur la façon de mettre à jour les différentes répliques d'un objet. Synchrones : une transaction qui fait une mise-à-jour doit modifier toutes les répliques avant de valider.

Asynchrone : les mises à jour sont faites périodiquement.

Avantage de synchrone : toutes les copies ont toujours la même valeur

Inconvénient : cher et long

Avantage de asynchrone : moins cher

Inconvénient : données pas toujours fraîches, des incohérences possibles, mais acceptables dans la plupart des cas.

## Exercice 2 : Pannes

2 pts

On considère l'état suivant de la base stocké sur le disque :

ITEM	VALEUR
A	13
B	40
C	35
D	4
E	18

Pour chacun des deux extraits de journal suivants, dites si ce journal pourrait être i) un journal 'UNDO' dont les actions aboutissent à l'état de la base ci-dessus, ii) un journal 'REDO' dont les actions aboutissent) l'état de la base ci-dessus. Justifiez votre réponse si elle est négative.

1. <Begin T1>  
 <T1, C, 35>  
 <T1, D, 450>  
 <Begin T2>  
 <T2, C, 18>  
 <T2, B, 40>  
 <Commit T1>  
 <T2, D, 18>  
 <Begin T3>  
 <T3, C, 35>  
 <T3, E, 18>  
 <T2, A, 13>  
 <Commit T3>  
 <Commit T2>

2. <Begin T1>  
 <T1, D, 4>  
 <Begin T2>  
 <T2, E, 6>  
 <T1, A, 5>  
 <T1, E, 18>  
 <Begin T3>  
 <T3, C, 35>  
 <T3, A 13>  
 <Commit T2>  
 <T3, B, 40>  
 <Commit T3>  
 <T1, A, 11>  
 <Commit T1>

Le journal 1 peut-il être un journal UNDO ?

Si non, pourquoi ?

Le journal 1 peut-il être un journal REDO ?

Si non, pourquoi ?

Le journal 2 peut-il être un journal UNDO ?

Si non, pourquoi ?

Le journal 2 peut-il être un journal REDO ?

Si non, pourquoi ?

1. i) non, la valeur de D ne correspond pas
2. ii) non, il n'y a pas la valeur de D qu'on retrouve dans la base. D vaut 4, et devrait valoir ( 450 ou )18 d'après le journal.
  - i) non, toutes les transactions sont validées, pas besoin de défaire
  - ii) oui

### Exercice 3 : Conception de BD Réparties

4 pts

La base de données d'une entreprise d'informatique, InfoNet, a le schéma global suivant:

SERVICE (#service, nom, chef, site)

PROJET (#projet, #service, nom, chef, budget, durée, date-début)

EMPLOYE (#emp, #service, nom, prénom, fonction, salaire, prime-annuelle, date-naiss, adresse, #tel)

InfoNet est localisé sur trois sites de la région parisienne: Paris centre, Paris sud, et Paris nord. Le site de Paris centre tient également lieu de siège pour l'entreprise.

**Question 1** En supposant que la base est répartie sur les trois sites informatiques du centre, du sud, et du nord, proposer une bonne décomposition de la base InfoNet sur ces trois sites en se basant sur les hypothèses suivantes.

- L'attribut site de SERVICE prend une des valeurs suivantes: "centre", "sud", et "nord".
- L'attribut nom de SERVICE prend une des valeurs suivantes: "commercial", "financier", "technique", "maintenance", "recherche et développement", etc.
- #service (resp. #emp) est clé primaire de SERVICE (resp. EMPLOYE).
- #projet est local à un service.
- Le chef d'un service ou d'un projet est un employé désigné par son numéro.
- Les employés sont affectés à un site donné, sauf pour les employés du service de maintenance qui interviennent dans tous les sites.

- Les données concernant la rémunération ainsi que les informations personnelles des employés sont regroupées et centralisées au siège de l'entreprise.

Donner la définition des différents fragments en utilisant les opérateurs de l'algèbre relationnelle.

Réponse :

- Fragmentation horizontale primaire de SERVICE en fonction de la valeur de l'attribut site:

$$\text{SERVICE}_{\text{centre}} = \Pi_{\# \text{service, nom, chef}} ( \sigma_{\text{site} = \text{"centre"}} (\text{SERVICE}) )$$

$$\text{SERVICE}_{\text{sud}} = \Pi_{\# \text{service, nom, chef}} ( \sigma_{\text{site} = \text{"sud"}} (\text{SERVICE}) )$$

$$\text{SERVICE}_{\text{nord}} = \Pi_{\# \text{service, nom, chef}} (\sigma_{\text{site} = \text{"nord"}} (\text{SERVICE}))$$

- Fragmentation horizontale dérivée de PROJET selon le service correspondant:

$$\text{PROJET}_{\text{centre}} = \text{PROJET} \bowtie \text{SERVICE}_{\# \text{service} \text{ centre}}$$

$$\text{PROJET}_{\text{sud}} = \text{PROJET} \bowtie \text{SERVICE}_{\# \text{service} \text{ sud}}$$

$$\text{PROJET}_{\text{nord}} = \text{PROJET} \bowtie \text{SERVICE}_{\# \text{service} \text{ nord}}$$

- EMPLOYE sera fragmentée en plusieurs étapes:

a) Fragmentation verticale pour séparer les données concernant la rémunération ainsi que les informations personnelles des informations professionnelles.

$$\text{EMPLOYE}_{\text{personnelle}} = \Pi_{\# \text{emp, nom, prénom, salaire, prime-annuelle, date-naiss, adresse, \#tel}} (\text{EMPLOYE})$$

$$\text{EMPLOYE}_{\text{professionnelle}} = \Pi_{\# \text{emp, \#service, nom, prénom, fonction}} (\text{EMPLOYE})$$

Le fragment  $\text{EMPLOYE}_{\text{personnelle}}$  sera stocké sur le site de Paris centre.

b) Le fragment  $\text{EMPLOYE}_{\text{professionnelle}}$  suivra une fragmentation horizontale dérivée en fonction de deux critères: (1) les employés de maintenance interviennent dans tous les sites, et donc leurs données seront répliquées sur les trois sites, et (2) les autres employés seront gérés par le site où ils sont affectés.

$$\text{EMPLOYE}_{\text{maintenance}} = \text{EMPLOYE}_{\text{professionnelle}} \bowtie (\sigma_{\text{nom} = \text{"maintenance"}} (\text{SERVICE}))_{\# \text{service}}$$

$$\text{EMPLOYE}_{\text{autre}} = \text{EMPLOYE}_{\text{professionnelle}} \bowtie (\sigma_{\text{nom} \neq \text{"maintenance"}} (\text{SERVICE}))_{\# \text{service}}$$

--- sur le site Paris centre, on aura, en plus du fragment  $\text{EMPLOYE}_{\text{personnelle}}$ , le fragment suivant:

$$\text{EMPLOYE}_{\text{professionnelle-centre}} = \text{EMPLOYE}_{\text{maintenance}} \cup (\text{EMPLOYE}_{\text{autre}} \bowtie \text{SERVICE}_{\text{centre}})$$

--- sur le site Paris sud, on aura le fragment suivant:

$$\text{EMPLOYE}_{\text{professionnelle-sud}} = \text{EMPLOYE}_{\text{maintenance}} \cup (\text{EMPLOYE}_{\text{autre}} \bowtie \text{SERVICE}_{\text{sud}})$$

--- enfin, sur le site paris nord, on aura le fragment suivant:

$$\text{EMPLOYEE}_{\text{professionnelle-nord}} = \text{EMPLOYEE}_{\text{maintenance}} \cup (\text{EMPLOYEE}_{\text{autre}} \bowtie \text{SERVICE}_{\text{nord}})$$

Remarque: Une autre solution consiste à stocker la totalité de la relation EMPLOYEE sur le site de Paris centre. Pour les autres sites, on suivra la même fragmentation donnée ci-dessus.

**Question 2.** Pour chacune des décompositions, donnez les requêtes de reconstruction.

On crée les 3 relations temporaires suivantes :

1. Tcentre (site) qui contient un n-uplet ayant la valeur centre.

Tsud(site) qui contient un n-uplet ayant la valeur sud.

Tnord(site) qui contient un n-uplet ayant la valeur nord.

Service =  $\bigcup_i (\text{Service}_i \times T_i)$ , avec  $i = \{\text{centre, sud ; nord}\}$

2. PROJET =  $\bigcup_i \text{PROJET}_i$

3.

EMPLOYEE prof =  $\bigcup_i \text{EMPLOYEE}_{\text{prof-I}}$

Rmq : l'union élimine les employés de maintenance en double provenant de la réplication

EMPLOYEE = EMPLOYEEprof JOINTURE EMPLOYEE perso

**Question 3.** Montrez que votre décomposition est correcte.

Une décomposition est correcte si la requête de reconstruction donne le même résultat que le résultat initial (pas de n-uplet 'perdu', pas de n-uplet 'en trop').

Service : tous les n-uplets se retrouvent( décomposition horizontale complète : l'attribut site prend 3 valeurs, la décomposition suit ces trois valeurs) Ensuite on fait l'union.

PROJET : idem

EMPLOYE : les deux premiers sont des unions, suite à des décomposition horizontales ou horizontale dérivée. Le dernier est une jointure sur la clé (#emp), et tous les attributs sont présents. Donc ; OK

#### Exercice 4 : Evaluation de requêtes

6 pts

On considère le schéma de bases de données suivant :

**Employes** (*emp-id* : integer, *serv-id* : integer, *salaire* : real)

**Services** (*serv-id* : integer, *dir-id* : integer, *budget* : integer)

Le champ *dir-id* de la relation Services est l'*emp-id* du directeur du service.

Les n-uplets de ces deux relations ont une taille de 20 octets. La relation **Employes** a 100 000 pages, et la relation **Services** a 5000 pages. Une page a une taille de 4K octets, et comprend 200 n-uplets.

Ces relations sont stockées sur un SGBD parallèle de 10 processeurs dans lequel les relations sont partitionnées par intervalles de valeurs (range partitioning) sur les attributs *salaire* et *budget*. On suppose que ces attributs contiennent des valeurs dans l'intervalle [0, 1 000 000] uniformément distribuées.

Le coût de lecture d'une page du disque (ou d'écriture sur le disque) est  $Ce/s$  et le coût de transfert d'une page d'un processeur vers un autre est  $Ct$ .

**Question 1.** Pour chacune des requêtes suivantes, décrivez le plan d'évaluation, et donnez le coût en termes de  $Ce/s$  et  $Ct$ .

On donnera le coût total de la requête, ainsi que le coût en temps écoulé (si plusieurs opérations sont effectuées en parallèle, le temps écoulé est le maximum du temps pris par chacun des processeurs pour faire son travail).

**R1 .** Quel est l'employé le mieux payé ?

Plan :

Coût total :

Temps écoulé :

Réponse :

Plan : Le n-uplet ayant le plus haut salaire se trouve sur le processeur 10. On parcourt tous les n-uplets de la relation Employe et on garde celui qui a le salaire le plus élevé.

Coût total : nb de pages de Emp sur le processeur 10) \* Ce/s

= 10 000 \* Ce/s

Temps écoulé : 10000 \* Ce/s

**R2 .** Quel est l'employé le mieux payé du service 421 (serv-id=421)?

Plan :

Coût total :

Temps écoulé :

Réponse :

Plan : Ce n-uplet peut se trouver sur n'importe quel processeur. Il faut donc parcourir tous les n-uplets de Employes et garder celui du service 421 qui a le salaire le plus élevé. Tous les processeurs sauf un envoient leur résultat sur un processeur qui gardera le n-uplet ayant le meilleur salaire parmi les 9 qui ont été envoyés et celui qu'il a sélectionné.

Coût total : 10 (chaque processeur) \* nb de pages de Emp \* Ce/s + 9 Ct

= 10 \* 10 000 \* Ce/s + 9 Ct = 100 000 Ce/s + 9 Ct

Temps écoulé : 10000 \* Ce/s + Ct

**R3.** Quel est l'employé le mieux payé de tous les services dont le budget est inférieur à 100 000 ?

(On utilisera l'algorithme de tri-fusion pour la jointure, dont le coût est 3 (M + N), M et N étant les tailles des deux relations à joindre).

Plan :



Coût :

Temps écoulé :

Réponse :

Plan : Les n-uplets de Services dont le budget est inférieur à 100 000 se trouvent sur le processeur 1. Les n-uplets des employés les mieux payés se trouvent sur les derniers processeurs, mais on ne peut pas savoir lequel. Il faut donc parcourir tous les n-uplets de Employes (comme dans la requête précédente).

Donc, le processeur 1 parcourt tous les services en retenant uniquement les attributs serv-id. le résultat est envoyé sur tous les autres processeurs et on fait la jointure avec les n-uplets de Employes et on prend celui qui a le salaire le plus élevé. Ensuite, tous les processeurs sauf un renvoient le tuple sélectionné. Le processeur qui reçoit ces n-uplets sélectionne celui qui a le salaire le plus élevé.

Coût :

Parcours de Service sur le processeur 1 pour récupérer le champ serv\_id (500 Ce/s) +

Envoi de serv\_id aux 9 autres processeurs ( $9 * 167 \text{ Ct}$ ) (500 pages de services, mais le champ serv-id fait 1/3 du n-uplet environ, donc  $500/3 = 167$ ) +

Stockage de ces tuples sur les processeurs ( $10 * 167 \text{ Ce/s}$ ) +

Joindre ces tuples avec employes ( $10 * 3 * (167 + 10\,000) * \text{Ce/s} = 305\,010 \text{ Ce/s}$ ) +

Envoi des résultats locaux au processeur initial ( $9 * \text{Ct}$ ).

Total =  $307\,180 \text{ Ce/s} + 1512\text{Ct}$

Temps écoulé :  $500 \text{ Ce/s} + 167 \text{ Ct} + 167 \text{ Ce/s} + 30\,501 \text{ Ce/s} + \text{Ct} = 31\,168 \text{ Ce/s} + 168 \text{ Ct}$

**Question 2.** Donnez le plan d'exécution de la requête suivante :

**R4.** Salaire des directeurs de services dont le budget est inférieur à 300000 et qui gagnent plus de 100000.

Réponse :

Plan. Les n-uplets de Service sont stockés sur les 3 premiers processeurs. Les tuples d'Employes dont le salaire est supérieur à 100 000 sont stockés sur les processeurs 2 à 10. On parcourt tous les n-uplets de Services en retenant le champ dir-id des n-uplets ayant un budget inférieur à 300000. On envoie le champ dir-id aux processeurs 2 à 10 qui les stockent. On fait la jointure pour obtenir le résultat.

Coût :

Temps écoulé :