

Université Pierre et Marie Curie – Paris 6 - Master d'informatique

## Administration des Bases de Données Réparties (ABDR)

Examen du 23 janvier 2006

Sans document – 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. Ne pas dégrafer le sujet. Ecrire à l'encre bleue ou noire.

La **qualité de la rédaction** sera fortement prise en compte. *Illustrer, argumenter et justifier* chaque réponse (une réponse non justifiée est insuffisante). Préciser toutes les hypothèses supplémentaires sur lesquelles repose votre raisonnement.

### Exercice 1 : Index, fragmentation

8 pts

Soit une base de données dont le schéma relationnel est :

**Client** (nom, age, genre, nc)

// le genre vaut 'M' ou 'F', nc est le n° de client

**Commande** (nc, produit, date, lieu, montant)

**Télévente** (nc, vendeur, date, résultat)

// démarchage par téléphone

On donne la taille des données de chaque relation en nombre de blocs disque.

Client : 100 000 blocs

Commande : 2 000 000 blocs

Télévente : 300 000 blocs

On observe la charge suivante sur la base de données, nommée *charge1* :

10%	des requêtes sélectionnent une valeur de	<i>Cient.nc</i> (i.e., de l'attribut <i>nc</i> de <i>Cient</i> )
30%	"	"
35%	"	"
10%	"	"
15%	"	"

*Client.nom*  
*Commande.produit*  
*Télévente.vendeur*  
*Télévente.date*

Toutes les requêtes sont de type ciblé ou multi-point. On suppose que l'usage d'un index réduit le nombre de blocs à lire par rapport à une lecture séquentielle.

#### Question 1

On suppose que l'espace disque disponible permet de créer seulement **deux** index.

Quels sont les index à créer pour apporter un gain d'efficacité maximal ? Justifier et quantifier le gain obtenu.

**Question 2**

On suppose maintenant que l'espace disque disponible est illimité. Quels sont les index à créer pour une efficacité optimale ? Préciser s'ils sont plaçant ou non.

Remarque : La distribution des valeurs des attributs est uniforme. Quantifier votre réponse en fonction, si nécessaire, du nombre de valeurs distinctes d'un attribut noté  $d(\text{attribut})$ .

**Question 3**

La charge est modifiée : l'occurrence de chaque requête est réduite de 5%, et on ajoute une transaction ajoutant  $n$  nouvelles Commandes. La nouvelle charge, nommée *charge2*, est :

5%	des requêtes sélectionnent une valeur de	<i>Cient.nc</i> (i.e., de l'attribut nc de Cient.)
25%	des requêtes sélectionnent une valeur de	<i>Client.nom</i>
30%	“ “ “	<i>Commande.produit</i>
5%	“ “ “	<i>Télévente.vendeur</i>
10%	“ “ “	<i>Télévente.date</i>
25%	de transaction T1	

Une transaction T1 insère  $n$  nouveaux nuplets dans Commande. On suppose que l'insertion d'un nuplet nécessite l'écriture d'une page de donnée et d'une page d'index pour chaque index existant sur cette relation.

A partir de quelle valeur de  $n$  est-il préférable de ne pas créer d'index sur l'attribut produit de Commande ?

**Question 4**

Dans cette question, la charge est celle décrite initialement et nommée *charge1*.

On suppose que l'espace disque disponible ne permet de créer **aucun** index. Par ailleurs, on connecte le SGBD existant à 9 autres. Les 10 SGBD sont tous reliés par un réseau haut débit (coût de communication négligeable) et peuvent stocker des fragments de la base (mais aucun index). Proposer une fragmentation des données qui apporte un gain d'efficacité maximal. Justifier et quantifier le gain obtenu par rapport à une solution centralisée (sans index).

**Question 5**

On donne une liste de requêtes. Pour chaque requête, quels sont les index les plus intéressants ? Préciser le type d'index (arbre, table de hachage, bitmap, multi-attributs, ...) et s'il est couvrant. Si vous proposez plusieurs index pour une requête, donnez l'ordre d'utilisation des index.

R1 : `select count(*) from Client where genre='M' and age between 18 and 25`

R2 : `select lieu, produit, avg(age) from Client c1, Commande c2  
where c1.nc = c2.nc and genre = 'F'  
group by lieu, produit  
having max(montant) / min(montant) > 1.5`

R3 : `select cl.nc, cl.nom, t.vendeur  
from Télévente t, Client c  
where t.nc = c.nc and c.nc in (select nc from Commande where produit like '%ette')`

**Exercice 2 : Exécution de requêtes****4 pts****Question 1**

Soient les tables :

```
create table R (a number(5), b number(5));
create table S (a number(5), b number(5));
```

Aucune contrainte d'intégrité n'est définie pour les tables R et S. Les attributs sont des nombres entiers inférieurs à 10000. Il y a un million de nuplets dans chaque relation. On suppose qu'il existe un seul index : arbre B+ sur S.b. On considère les requêtes suivantes avec leur plan d'exécution (l'indentation représente la structure arborescente d'un plan). Pour chaque plan, décrire le problème de performance potentiel, proposer un meilleur plan d'exécution. Justifier votre réponse.

**Requête R1**

```
SELECT R.a
FROM R, S
WHERE R.a = S.a and S.b > 5
```

**Plan pour R1**

SELECT STATEMENT	// projection
MERGE JOIN	// jointure par fusion
SORT (JOIN)	// tri avant jointure
TABLE ACCESS (FULL) R	// lecture séquentielle de R
SORT (JOIN)	// tri avant jointure
TABLE ACCESS (FULL) S	// lecture séquentielle de S et sélection tq b>5

**Requête R2**

```
SELECT R.a
FROM R
WHERE EXISTS (
    SELECT *
    FROM S
    WHERE R.a = S.a AND S.b = 5)
```

**Plan pour R2**

SELECT STATEMENT	// projection
INDEX NESTED LOOP	// jointure par boucles imbriquées
TABLE ACCESS ( FULL) R	// lecture séquentielle de R
TABLE ACCESS (BY ROWID) S	// accès aux tuples de S depuis l'index
INDEX (RANGE SCAN)	// traversée de l'index (avec b=5)

**Question 2**

a) Soient les 3 requêtes R2 (définie ci-dessus), R3 et R4

R3 :

```
select R.a
from S, R
where S.b =5 and S.a = R.a
```

R4 :

```
select distinct R.a
from R
where R.a in (select S.a from S where S.b =5)
```

Quelles sont les requêtes équivalentes ? Justifier.

R2 $\Leftrightarrow$ R3 ? oui ☐

non ☐

Justif. :

R2 $\Leftrightarrow$ R4 ? oui ☐

non ☐

Justif. :

R3 $\Leftrightarrow$ R4 ? oui ☐

non ☐

Justif. :

**Exercice 3 : Stockage****2 pts****Question 1**

On souhaite installer un SGBD sur une machine disposant de 3 disques D1 à D3. Lors de l'installation, le logiciel du SGBD est stocké sur D1. Il est demandé de choisir un emplacement pour les données contenues dans la base, et le journal des transactions. Quelle est la proposition la plus intéressante ? Justifier en indiquant les avantages et inconvénients de chaque proposition P1 à P4.

P1) Données D2, journal sur D3

P2) Données réparties sur D1, D2, D3, journal répliqué sur D1, D2 et D3

P3) Données réparties sur D1, D2, journal sur D3

P4) Données sur D3, journal répliqué sur D1 et D2

Entourer la proposition retenue :      P1          P2          P3          P4

**Exercice 4 : Cas des bases de données astronomiques****6 pts**

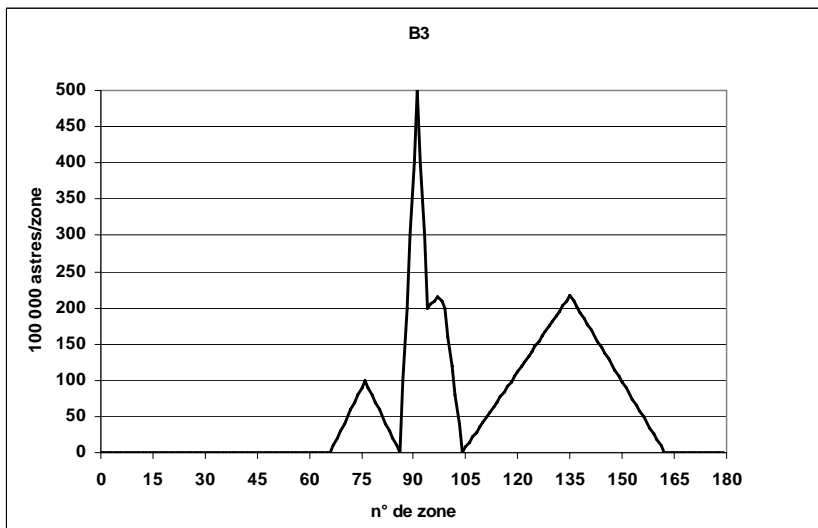
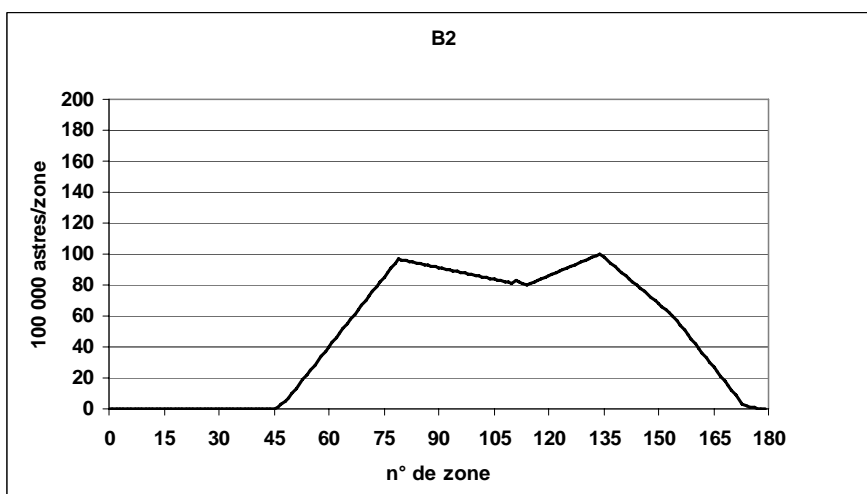
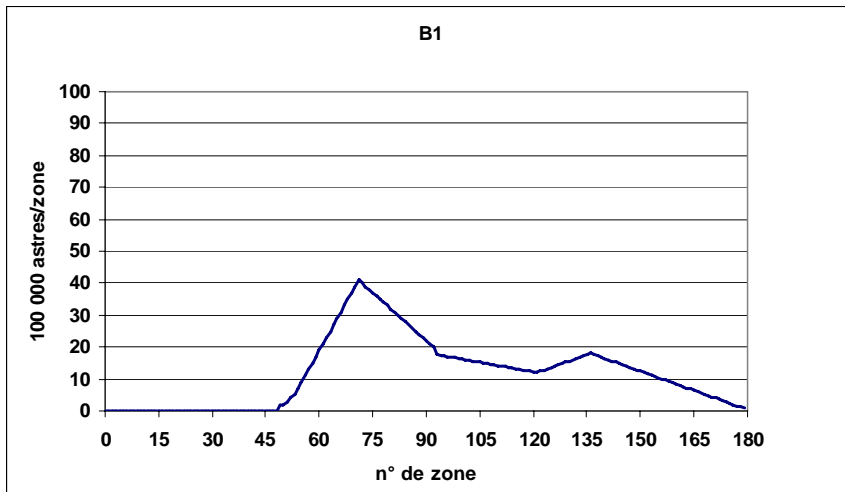
Cet exercice étudie l'interrogation en SQL des données astronomiques de l'observatoire virtuel ([www.usvo.org](http://www.usvo.org)). Soient 3 bases de données astronomiques B1, B2, B3 dont le schéma est identique. Chaque base contient une seule relation *Astre*(num, a, d, p<sub>1</sub>, ..., p<sub>500</sub>). L'attribut *num* est l'identifiant d'un astre (taille : 8 octets). Un identifiant est local à une base et ne permet pas de comparer des astres de deux bases différentes. Un astre est repéré par deux coordonnées sur la sphère céleste : l'ascension droite *a* et la déclinaison *d* (taille : 4 octets par coordonnée). La relation *Astre* possède également 460 attributs *p<sub>i</sub>* décrivant diverses propriétés astrophysiques et astrométriques d'un astre. La taille d'un nuplet est de 1862 octets. Les bases B1, B2 et B3 contiennent respectivement 2, 9 et 120 millions d'astres.

Les utilisateurs posent deux catégories de requêtes :

- Une requête dite *à large échelle* interroge une seule base. Elle affiche les identifiants des astres qui ont une certaine propriété astrophysique. Par exemple, quels sont les astres de B1 dont la magnitude est comprise entre 9 et 10 ?
- Une requête dite *de mise en correspondance* interroge les 3 bases. Elle affiche l'identifiant et les coordonnées des objets présents dans les 3 bases qui se situent dans une même région du ciel, spécifiée par un point et un rayon, et qui satisfont certaines propriétés astrophysiques. On suppose que la requête est très sélective : moins de 5000 astres par base satisfont la requête.

Chaque base contient les observations d'un certain télescope. Les trois télescopes n'observent pas tous les mêmes zones du ciel. Les observations sont plus ou moins fines selon le télescope, c'est pourquoi tous les télescopes n'observent pas autant d'astres dans chaque zone.

On donne le nombre d'astres observés pour chaque zone et chaque télescope. Il y a 180 zones allant de la latitude -90° à la latitude +90°. Chaque zone représente un «anneau» de la sphère céleste située entre 2 cercles espacés de 1 degré de latitude. On donne le nombre d'astres par zone, en centaine de milliers :



### Question 1

L'exécution d'une requête à large échelle sur la base B3 dure en moyenne 11 minutes, bien qu'aucune autre requête ne s'exécute simultanément. On constate que 95% du temps de traitement est passé à lire des données du disque

a) La relation Astre de B3 est-elle lue en totalité pour traiter une requête à large échelle ? Si oui, donner une approximation du temps de lecture moyen pour un bloc disque de 8KO et vérifier que la valeur obtenue est réaliste. Sinon, décrire l'organisation interne des données permettant d'obtenir un tel temps de réponse.

b) La durée de 11 minutes n'est pas acceptable pour les utilisateurs souhaitant une réponse en moins de 2 minutes. Pour améliorer la situation, on décide de répartir B3 sur 8 SGBD. Proposer une fragmentation optimale pour le traitement d'une requête à large échelle.

### Question 2

L'exécution d'une requête de mise en correspondance nécessite l'accès aux 3 bases. On dispose pour chaque base, de 8 SGBD.

a) Proposer une fragmentation optimale pour le traitement d'une requête de mise en correspondance.

b) Dans quel ordre il faut accéder à chaque base ? Expliquer comment déterminer l'ordre d'accès le plus efficace aux 3 bases.

c) Proposer brièvement une structure d'index permettant de retrouver tous les astres d'une base localisés dans une certaine région spécifiée par un point (coordonnées  $a$  et  $d$ ) et un rayon  $r$ .