

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

## MI034 - Bases de Données Réparties Examen du 6 mai 2014

<b>Les documents de cours, TD et TME sont autorisés</b> Durée : 2h.  <b>Les téléphones et tout autre appareil doivent être éteints et rangés.</b>  <b>Répondre aux questions sur la feuille du sujet</b> dans les cadres appropriés. La taille des cadres suggère celle de la réponse attendue. Justifier vos réponses. La qualité de la rédaction sera prise en compte. Le barème est donné à titre indicatif. Ecrire à l'encre bleue ou noire. Ne pas dégrafer le sujet.	<b>Ex1 :</b>	
	<b>Ex2 :</b>	
	<b>Ex3 :</b>	
	<b>Ex4 :</b>	
	<b>Ex5 :</b>	

<b>Exercice 1 : Conception de bases de données réparties</b>	<b>4 pts</b>
--	--------------

Trois communes limitrophes (Romainville, Les Lilas, Noisy le Sec) ont décidé de mutualiser leurs gymnases et les entraîneurs. La gestion globale est effectuée par une base de données répartie, dont le schéma global est le suivant :

**ENTRAINEUR** (Identr, nom, adresse, tél, ville, salaire)

L'attribut *ville* désigne la commune qui emploie l'entraîneur.

**SPORTIF** (Idsportif, nom, adresse, assurance, ville, équipe)

Un sportif habite dans la commune *ville* et peut jouer dans l'équipe *équipe*.

**GYMNASE** (Idgymn, adresse, ville)

Un gymnase est situé dans la commune *ville*.

**EQUIPE** (équipe, sport, niveau)

Une équipe pratique le sport *sport*.

**HORAIRE** (Idgymn, équipe, jour, heure\_début, heure\_fin, Identr)

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

- Chaque commune rémunère ses entraîneurs en envoyant un chèque à leur adresse, mais aussi elle doit pouvoir contacter (par téléphone) tout entraîneur qui utilise ses gymnases.
- Chaque équipe correspond à un sport. Les équipes ont droit à un créneau (jour, heure) dans un des gymnases pour leur entraînement. Cependant, pour le sport «cyclisme », il n'y pas besoin de gymnase.
- Chaque commune gère évidemment ses propres sportifs, ainsi que ses gymnases et les créneaux correspondants.
- Les équipes ne sont associées à aucune commune en particulier. Cependant, pour des questions d'assurance, chaque université doit aussi gérer les sportifs qui utilisent ses gymnases. Pour le cyclisme, c'est Romainville qui en a la charge.

Les relations globales sont fragmentées et réparties sur les différents sites, un site par commune.

Donner la définition des différents fragments ainsi que leur allocation, en utilisant les opérateurs de l'algèbre relationnelle. Répondre en utilisant la variable *i* dont la valeur est dans { Romainville, Les Lilas, Noisy le Sec }.

**Exercice 2 : Questions de cours****3 points**

Répondre aux questions suivantes en argumentant votre réponse.

**Question 1 :** Les systèmes pair à pair structurés (par une table de hachage distribuée sur la clé A) sont-ils adaptés pour les requêtes suivantes? Justifiez votre réponse en une phrase.

***Select \* from R, S where S.A=R.A and R.B = 10 ;***

Adapté - Non-adapté (entourer la bonne réponse)

Justification :

***Select \* from R where A = 2 ;***

Adapté - Non-adapté (entourer la bonne réponse)

Justification :

***Select \* from R where A >= 3 and A <= 10;***

Adapté - Non-adapté (entourer la bonne réponse)

Justification :

**Question 2 :** Quel type de réplication (synchrone ou asynchrone) est le mieux adapté pour les entrepôts de données. Justifiez votre réponse.

**Question 3 :** Qu’est-ce qui distingue une base de données parallèle d’une base de données répartie en termes d’allocation des données (fragments) sur les différents sites.

### Exercice 3 : Optimisation de requêtes réparties

**5 pts**

Soit la base de données répartie sur les sites S1 et S2 telle que :

**Sportif** (ids, nom, âge, ville) est sur S1,  
**Performance** (ids, comp, classement) est sur S2.

L’attribut *comp* identifie une compétition sportive auquel un sportif obtient un classement. Seuls 30% des sportifs de la base ont déjà participé à une compétition, les autres ne pratiquent le sport qu’en tant que loisir. On suppose que la distribution des valeurs des attributs est uniforme.

Soit  $t(\text{attr})$  la taille en octets d’un attribut, on a :

$t(d) = 10$ ,  $t(\text{nom}) = 45$ ,  $t(\text{âge}) = 10$ ,  $t(\text{ville}) = 35$ ,  $t(\text{compétition}) = 30$ ,  $t(\text{classement}) = 10$

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

$T(\text{Sportif}) = 1$   $T(\text{Performance}) = 5$

Le **coût** d’un plan d’exécution d’une requête est la **somme des transferts** de données, mesuré en millions d’octets.

Soit la requête **R1** posée sur S3 (le résultat doit être délivré sur S3)

```
R1 :  select e.d, nom, âge, ville, comp, classement
      from Sportif s, Performance p
      where p.ids = s.ids ;
```

On considère les plans d’exécution de R1 suivants :

P1 : transférer les deux relations sur S3 puis traiter la jointure sur S3.

P2 : transférer Sportif sur S2, puis jointure sur S2, puis transfert final vers S3.

P3 : transférer Performance sur S1, puis jointure sur S1, puis transfert final vers S3.

P4 : semi-jointure sur S1, puis jointure sur S2, puis transfert final vers S3.

a) Quelle est l'expression algébrique de la requête R1

b) Quelle est la cardinalité de Sportif?

c) Quelle est la cardinalité de la requête R1 ?

d) Combien vaut T(R1) ?

e) Quel est le coût de P1 ?

Coût(P1) =

f) Quel est le coût de P2 ?

Coût(P2) =

g) Quel est le coût de P3 ?

Coût(P3) =

h) Rappelez en quelques lignes les différentes étapes de P4 (utiliser l'algèbre relationnelle autant que possible)

P4 :

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.

i) Quel est le coût de P4 ?

P4

j) Expliquer en quelques lignes pourquoi il est inutile d'envisager un plan P5 où la semi-jointure se ferait sur S2 ?

**Exercice 4 : Transactions réparties****4 points**

Soit une base de données répartie telle que les granules A et B sont sur le site S1, les granules C, D, E sont sur le site S2. Soit la séquence Seq suivante des transactions T1, T2, T3, T4 (Li(g) signifie lecture du granule g par la transaction ti, Ei(g) signifie écriture par la transaction Ti du granule g, Vi signifie validation de la transaction Ti).

Seq : L3(A), L2(C), L1(A), E3(A), L2(A), L2(D), L3(D), E4(D), L4(C), L1(D), L3(B), E2(B)

**Question 1**

On suppose un contrôle de concurrence optimiste où le graphe de précédence global de l'exécution est construit sur S1, progressivement, à chaque nouvelle précédence générée sur l'un ou l'autre des sites. Les messages sont de la forme (Ti, Tj, g) si Ti précède Tj sur le granule g.

Représenter ce graphe à la fin de Seq. L'exécution sera-t-elle validée ? Justifiez votre réponse. Cette décision aurait-elle pu être prise en ne considérant que les graphes de précédences locaux ? Quels messages ont-été envoyés par S2 jusqu'à ce moment ?

**Au moment de la première demande de validation**

Graphe de précédence sur S1 (indiquer sur chaque arc le(s) granule(s) correspondants(s))

Graphe de précédence sur S2 (indiquer sur chaque arc le(s) granule(s) correspondants(s))

Graphe de précédence global (indiquer sur chaque arc le(s) granule(s) correspondants(s))

Validation acceptée (oui/non) ?

Justification :

Décision possible en ne considérant que les graphes de précédence locaux (oui/non) ?

Justification :

Messages envoyés de S2 à S1 (dans l'ordre d'émission):

**Question 2**

On suppose maintenant un contrôle de concurrence par verrouillage deux-phases sur la séquence Seq. Les transactions lectrices obtiennent toujours le verrou partagé sur un granule si c'est possible, même si des transactions écrivaines sont en attente sur le granule.

Assiste-t-on à un interblocage ? Si oui, montrez l'état de la table des verrous à l'instant où se produit l'interblocage (en indiquant quelle est l'opération de Seq qui génère l'interblocage. Sinon, montrez l'ordre produit par le contrôleur de concurrence.

Remarque : la promotion de verrou est possible. Rappel : lorsqu'une transaction T demande un verrou exclusif sur un granule g, qu'elle possède déjà un verrou partagé sur g, et qu'aucune autre transaction ne verrouille g, alors le verrou partagé est promu en verrou exclusif afin de satisfaire la demande de T.

**Interblocage : oui non**

Table des verrous (si oui), ordre produit (si non) :

**Exercice 5 : Requêtes réparties avec JDBC****5 pts**

On considère des étudiants qui suivent plusieurs UE de math et de langue. Un étudiant a au moins une note dans une UE de math et une autre dans une UE de langue. La base de données est répartie sur 3 sites S1 à S3 comme suit :

S1 : **Etu** (d, nom, age) // d est le numéro de dossier de l'étudiant  
 S2 : **Math** (d, ue, note) // d fait référence à Etu.d  
 S3 : **Langue** (d, ue, note) // d fait référence à Etu.d

Les programmes java, proposés par la suite, sont des extraits de programmes complets, suffisants pour comprendre ce que fait le programme. Ils s'exécutent sur un site S0 pouvant contacter les sites S1 à S3.

**Question 1 : Qu'affiche la procédure A1(N) ?**

```

1 void A1(int N) {
2     s2 = DriverManager.getConnection(url2, u2, pw2); // site S2
3
4     PreparedStatement q2 = s2.prepareStatement("select d from Math group
5 by d having min(note) > ? ");
6     q2.setInt(1, N);
7     ResultSet r2 = q2.executeQuery();
8
9     while(r2.next()) {
10        out.println(r2.getString(1));
    }
    q2.close();s2.close();
}

```

A1 affiche ...

**Question 2**

Le programme suivant affiche le résultat de la requête *globale* R1 définie ci-dessous. Le résultat est **trié** par numéro de dossier d'étudiant. Pour un étudiant, les notes ne sont pas triées.

```
R1 : Select * from (
    select e.d, e.nom, i.ue, i.note
    from Etu e, Math i
    where e.d = i.d
    union
    select e.d, e.nom, i.ue, i.note
    from Etu e, Langue i
    where e.d = i.d )
order by d
```

a) Compléter les trous :

```
1  s1 = DriverManager.getConnection(url1, u1, pw1); // site S1
2  s2 = DriverManager.getConnection(url2, u2, pw2); // site S2
3  s3 = DriverManager.getConnection(url3, u3, pw3); // site S3
4
5  Statement q1 = s1.createStatement();
6  ResultSet r1 = q1.executeQuery("select TROU1 from Etu TROU2");
7
8  // q2 et q3 sont des PreparedStatement
9  q2 = s2.prepareStatement("select ue, note from Math TROU3");
10 q3 = s3.prepareStatement("select ue, note from Langue TROU4");
11
12 while(r1.next()) {
13     q2.setString(1, TROU5);
14     ResultSet r2 = q2.executeQuery();
15     while(r2.next()) {
16         out.println( TROU6 + r2.getString(1) + " " + r2.getString(2));
17     }
18     q3.setString(1, TROU7);
19     ResultSet r3 = q3.executeQuery();
20     while(r3.next()) {
21         out.println(TROU8);
22     }
23 }
24 }
```

Trous1 et 2 : `SELECT _____ FROM Etu _____`

Trou3 : \_\_\_\_\_ Trou4 : \_\_\_\_\_

Trou5 : \_\_\_\_\_ Trou6 : \_\_\_\_\_

Trou7 : \_\_\_\_\_ Trou8 : \_\_\_\_\_

b) Que faudrait-il modifier pour que le résultat soit trié par moyenne générale décroissante ?

**Question 3**

On veut connaître les étudiants qui compensent les mathématiques grâce aux langues. Soit *M1* la moyenne en math d'un étudiant (sur les UE de math qu'il suit), et *M2* sa moyenne générale sur toutes les UE de math et de

langue qu'il suit. Le programme A3 affiche les étudiants tels que  $M1 < 10$  et  $M2 \geq 10$ . Pour chaque étudiant, on affiche une ligne du résultat formée des trois champs :  $d$ ,  $M1$ ,  $M2$ .

a) Ecrire A3 en pseudo-code (**pas** en java). Préciser les requêtes SQL, le calcul de  $M2$  et l'affichage du résultat.

```

Connexion à S2, traiter la requête R2 suivante :
SELECT ...

Pour chaque nuplet  $t$  de R2

    Connexion à S3 pour traiter la requête R3 suivante :
    SELECT ...

    Pour chaque nuplet  $t'$  de R3

        Si(
            )
        Alors
            afficher
        Fin si
    Fin pour
Fin pour

```

b) A3 utilise maintenant un PreparedStatement pour accéder à S3 avec la requête suivante : « récupérer la moyenne générale  $M2$  de l'étudiant  $d$  pour lequel  $M2 \geq 10$  ». Compléter l'instruction ci-dessous qui définit cette requête. Ne pas oublier le(s) paramètre(s) marqué(s) par un point d'interrogation. Remarque : si l'étudiant  $d$  n'a pas la moyenne générale, alors le résultat de la requête est vide. Autre remarque : on « délègue » le calcul de la moyenne générale sur le site S3.

```

PreparedStatement
q3 = s3.prepareStatement(" SELECT

                                FROM Langue

                                ...

                                ...

);

```