



## L3 Miage Bases de données

Durée 2h – documents de CM/TD/TP et dictionnaire bilingue autorisés

<input type="text"/>	0	<input type="text"/>	0	<input type="text"/>	0	<input type="text"/>	0	<input type="text"/>	0	<input type="text"/>	0	<input type="text"/>	0
<input type="text"/>	1	<input type="text"/>	1	<input type="text"/>	1	<input type="text"/>	1	<input type="text"/>	1	<input type="text"/>	1	<input type="text"/>	1
<input type="text"/>	2	<input type="text"/>	2	<input type="text"/>	2	<input type="text"/>	2	<input type="text"/>	2	<input type="text"/>	2	<input type="text"/>	2
<input type="text"/>	3	<input type="text"/>	3	<input type="text"/>	3	<input type="text"/>	3	<input type="text"/>	3	<input type="text"/>	3	<input type="text"/>	3
<input type="text"/>	4	<input type="text"/>	4	<input type="text"/>	4	<input type="text"/>	4	<input type="text"/>	4	<input type="text"/>	4	<input type="text"/>	4
<input type="text"/>	5	<input type="text"/>	5	<input type="text"/>	5	<input type="text"/>	5	<input type="text"/>	5	<input type="text"/>	5	<input type="text"/>	5
<input type="text"/>	6	<input type="text"/>	6	<input type="text"/>	6	<input type="text"/>	6	<input type="text"/>	6	<input type="text"/>	6	<input type="text"/>	6
<input type="text"/>	7	<input type="text"/>	7	<input type="text"/>	7	<input type="text"/>	7	<input type="text"/>	7	<input type="text"/>	7	<input type="text"/>	7
<input type="text"/>	8	<input type="text"/>	8	<input type="text"/>	8	<input type="text"/>	8	<input type="text"/>	8	<input type="text"/>	8	<input type="text"/>	8
<input type="text"/>	9	<input type="text"/>	9	<input type="text"/>	9	<input type="text"/>	9	<input type="text"/>	9	<input type="text"/>	9	<input type="text"/>	9

← codez votre numéro d'étudiant ci-contre, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

**Question 1 ♣** Construire un arbre B+ d'ordre 2 (donc au plus 4 éléments par noeud), au fur et à mesure des insertions suivantes :

[5, 19, 3, 9, 4, 16, 15, 13, 7, 1, 14, 20, 8, 11, 18, 17, 12, 10, 6, 2]

Que contient la racine de cet arbre à la fin de ces insertions ?

- ☐ 15
- ☐ 13
- ☐ 9

- ☐ 11,3,1
- ☐ Aucune de ces réponses n'est correcte.



**Question 2** Dessinez l'arbre B+ **final** ci-dessous (ne **pas** cocher les cases f, p, j ci-contre)

<input type="checkbox"/>	f	<input type="checkbox"/>	p	<input type="checkbox"/>	j
--------------------------	---	--------------------------	---	--------------------------	---

**Question 3 ♣** On suppose que les n-uplets sont stockés aux feuilles de l'arbre B+. Combien faut-il lire de pages disque distinctes (donc, ne pas compter deux fois la même page) pour rechercher l'élément 17 ?

- ☐ 3
- ☐ Aucune
- ☐ 5

- ☐ 2
- ☐ Aucune de ces réponses n'est correcte.

**Question 4 ♣** De même, combien faut-il lire de pages disque distinctes pour tester la présence de l'élément 21 ?

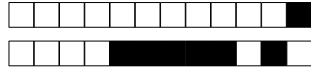
- ☐ 5
- ☐ 3
- ☐ 6

- ☐ 2
- ☐ Aucune de ces réponses n'est correcte.

**Question 5 ♣** De même, combien faut-il lire de pages disque distinctes pour accéder aux n-uplets de l'intervalle  $[8, 16]$  ?

- ☐ 8
- ☐ 10
- ☐ 5

- ☐ 6
- ☐ Aucune de ces réponses n'est correcte.



**Question 6 ♣** On observe le plan d'exécution suivant :

```
SELECT STATEMENT
| 1 | TABLE ACCESS BY INDEX ROWID
| 2 | ..... INDEX UNIQUE SCAN
```

Quelle est la requête SQL pouvant donner ce plan ?

- ☐ select \* from Client where Client.id < 5
- ☐ select \* from Client where Client.id=1254
- ☐ select index from access where index='unique'
- ☐ select \* from Client, Commande where Client.id=Commande.id and Client.id=5
- ☐ Aucune de ces réponses n'est correcte.

**Question 7 ♣** On observe le plan d'exécution suivant :

```
SELECT STATEMENT
| 1 | MERGE JOIN
| 2 | .....SORT JOIN
| 3 | .....TABLE ACCESS FULL
| 4 | .....SORT JOIN
| 5 | .....NESTED LOOP
| 6 | .....TABLE ACCESS FULL
| 7 | .....TABLE ACCESS FULL
```

Quelle est la requête SQL pouvant donner ce plan ?

- ☐ select nom from Client order by montant asc
- ☐ select full from nested1, nested2 where loop is not null
- ☐ select \* from Client, Commande, Produit where Client.id=Commande.cid and Commande.pid=Produit.id
- ☐ select \* from Client natural join Commande where Commande.id=513
- ☐ Aucune de ces réponses n'est correcte.

**Question 8**

Soit l'histoire suivante reçue par un ordonnanceur :

$$r_2[y]r_1[y]w_1[x]r_3[x]w_2[y]r_3[x]r_1[y]r_1[z]c_1w_2[z]w_3[x]c_3c_2$$

Cette histoire est-elle sérialisable ?

- ☐ non
- ☐ oui



**Question 9 ♣**

Cette histoire est :

- ☐ pas recouvrable
- ☐ recouvrable, évitant les annulations en cascade mais pas stricte
- ☐ recouvrable, évitant les annulations en cascade et stricte
- ☐ stricte mais pas recouvrable
- ☐ Aucune de ces réponses n'est correcte.

**Question 10 ♣**

Laquelle des histoires suivantes correspond à l'exécution effective de cette histoire selon l'algorithme de verrouillage à 2 phases (selon la définition vue en cours : sous-verrous en lecture partageable et sous-verrous en écriture exclusif, relâchement de tous les verrous lors du *commit*, redémarrage des transactions bloquées dans l'ordre FIFO).

- ☐  $r_2[y]r_1[y]w_1[x]r_1[y]r_1[z]c_1r_3[x]w_2[y]r_3[x]w_2[z]w_3[x]c_3c_2$
- ☐  $r_2[y]r_1[y]w_1[x]w_2[y]r_3[x]w_1[y]r_3[x]r_1[z]c_1w_2[z]w_3[x]c_3c_2$
- ☐  $r_2[y]r_1[y]w_1[x]r_3[x]r_3[x]w_1[y]w_2[y]r_1[z]c_1w_2[z]w_3[x]c_3c_2$
- ☐ Aucune de ces réponses n'est correcte.

**Question 11** Expliquez pourquoi, si une histoire n'est pas recouvrable, alors elle sera nécessairement modifiée par l'algorithme de verrouillage à deux phases.

☐ f ☐ p ☐ j