

NOM : _____ Prénom : _____ Groupe : _____	<i>R3.07 - BD dans les langages de programmation</i>  <b>Exemple d'Examen final</b>
---	---

### **Exercice 1 : Transaction**

On considère la relation compte suivante :

**Compte** (numcompte, solde, numclient)

**Client**(numclient, nomclient, prenomclient)

1. Soit la transaction suivante :

Begin;

INSERT INTO COMPTE VALUES (456, 1000, 123);

INSERT INTO COMPTE VALUES (457, 10000, 123);

Commit;

**Une panne logicielle se produit. Que doit faire le SGBD dans les cas suivants pour assurer la propriété de Durabilité :**

- a) 1<sup>er</sup> cas : si un point de contrôle a été fait après le COMMIT et que la panne se produit après le COMMIT ?

--

- b) 2<sup>ème</sup> cas : si un point de contrôle a été fait après l'insertion du 1er compte et que la panne se produit après le COMMIT ?

--

- c) 3<sup>ème</sup> cas : si un point de contrôle a été fait avant le début de la transaction et que la panne se produit avant le COMMIT ?

--

- a) 4<sup>ème</sup> cas : si un point de contrôle a été fait après l'insertion du 1er compte et que la panne se produit avant le COMMIT ?

Dans les questions suivantes, deux utilisateurs différents (user1 et user2) lancent des instructions en parallèle et l'axe du temps est représenté de façon verticale ( $T1 < T2 < \dots < Tn$ ) dans le mode de gestion de transactions par défaut.

Pour chacune des questions suivantes, répondez au QCM.

2. On suppose que la valeur initiale du solde du compte numéro 123 est de 100€. Considérez l'exécution suivante.

Temps	User1	User2
T1	Begin;	
T2	select solde from compte where numcompte = 123;	
T3		Begin;
T4		select solde from compte where numcompte = 123;
T5	update compte set solde = solde + 500 where numcompte = 123;	
T6		update compte set solde = solde + 500 where numcompte = 123;
T7	select solde from compte where numcompte = 123;	
T8	Commit;	
T9		select solde from compte where numcompte = 123;
T10		Commit;

- Au temps T8, le résultat de la requête pour user1 est égal à 600 ☐ oui ☐ non
- Au temps T10, le résultat de la requête pour user2 est égal à 1100 ☐ oui ☐ non
- Il y a un problème de lecture non reproductible ☐ oui ☐ non
- Il y a un problème de lecture impropre ☐ oui ☐ non
- Il y a un problème de perte de mise à jour ☐ oui ☐ non

3. Considérez l'exécution ci-dessous.

Temps	User1	User2
T1	begin;	
T2		begin;
T3	update compte set solde = solde + 500 where numcompte = 123;	
T4		update compte set solde = solde - 200 where numcompte = 245;
T5	commit;	
T6		commit;

- L'utilisateur user2 est mis en attente ☐ oui ☐ non  
 Il y a interblocage ☐ oui ☐ non  
 Il y a un assassinat ☐ de user1 ☐ de user2 ☐ non

4. On suppose que la valeur initiale du solde du compte numéro 123 est de 100€. Considérez l'exécution ci-dessous.

Temps	User1	User2
T1	begin;	
T2		begin;
T3	update compte set solde = solde + 500 where numcompte = 123;	
T4		update compte set solde = solde + 500 where numcompte = 123;
T5	commit;	
T6		select solde from compte where numcompte = 123;
T7		commit;

- L'utilisateur user2 est mis en attente ☐ oui ☐ non  
 Il y a interblocage ☐ oui ☐ non  
 Il y a un assassinat ☐ de user1 ☐ de user2 ☐ non  
 Au temps T6, le résultat de la requête pour user2 est égal à 1100 ☐ oui ☐ non

5. Considérez l'exécution ci-dessous.

Temps	User1	User2
T1	begin;	
T2		begin;
T3	insert into compte values (245, 500, 41);	
T4		insert into compte values (250, 1000, 42);
T5	insert into compte values (250, 1000, 42);	
T6		insert into compte values (245, 500, 41);

- L'utilisateur user2 est mis en attente ☐ oui ☐ non  
 Il y a interblocage ☐ oui ☐ non  
 Il y a un assassinat ☐ de user1 ☐ de user2 ☐ non

1. Soient les 2 transactions suivantes :

Begin; Select sum(solde) from Compte where numclient=2; Update Compte set solde = solde+100 where numclient=2 ; Select * From Client where numClient=2; Commit;	Begin; Select * from Client where numClient=2; Insert into Compte values (4, 1000, 2); Commit;
---	---

Proposez un ordonnancement non séquentiel de l'exécution des deux transactions en appliquant le protocole du verrouillage à 2 phases avec des verrous binaires. Notez les opérations de verrouillage et libération des verrous.

*Durée : 1h30 – Documents autorisés : Une feuille A4 recto-verso manuscrite*

Temps	Transaction 1	Transaction 2
T1		
T2		
T3		
T4		
T5		
T6		
T7		
T8		
T9		
T10		
T11		
T12		

*Durée : 1h30 – Documents autorisés : Une feuille A4 recto-verso manuscrite*

T13		
T14		
T15		
T16		
T17		
T18		
T19		
T20		

### Exercice 3 : Optimisation

Une école de musique a mis en place une application pour gérer ses élèves et ses cours. Le schéma relationnel de la base de données de l'application est le suivant :

**prof** (numprof, nomprof)

par exemple, le  $n$ -uplet *prof* (1, gauduel) signifie que le *prof* de numéro 1 a pour nom gauduel. Un *prof* est identifié par son numéro.

**cours** (numcours, discipline, niveau)

par exemple, le  $n$ -uplet cours  $(1, \text{piano}, d)$  signifie que le cours de numéro 1 correspond à la discipline piano et a pour niveau  $d$  (débutant). Un cours est identifié par son numéro.

**enseigne** (numprof, numc, datecours)

par exemple, le  $n$ -uplet  $\text{enseigne}(1, 1, 2020)$  signifie que le cours numéro 1 est enseigné par le prof numéro 1 durant l'année 2020.

1. Expliquer avec vos propres mots ce qu'est un index ET à quoi sert un index

--

2. Dans chacune des tables du schéma donné en début d'exercice, sur quels attributs Postgres a-t-il automatiquement créé des index ?

3. Soit le plan d'exécution ci-dessous.

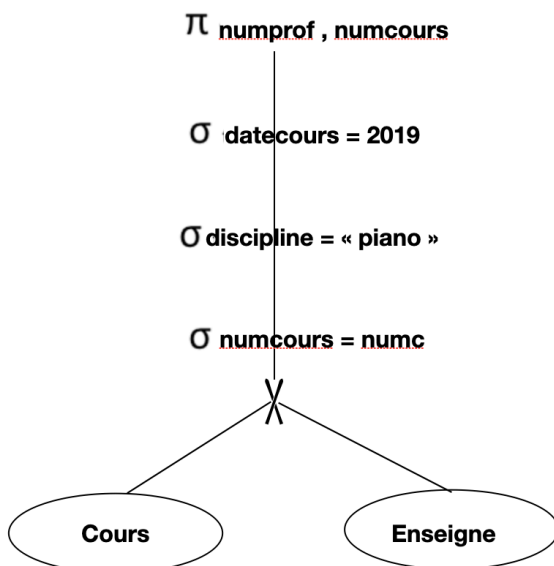
a) donnez la requête SQL correspondant à ce plan

```
QUERY PLAN
-----
Seq Scan on prof (cost = 0.00..578.87 rows= 1 width=54)
    Filter: ( nomprof = 'Bupre' :text)
(2 rows)
```

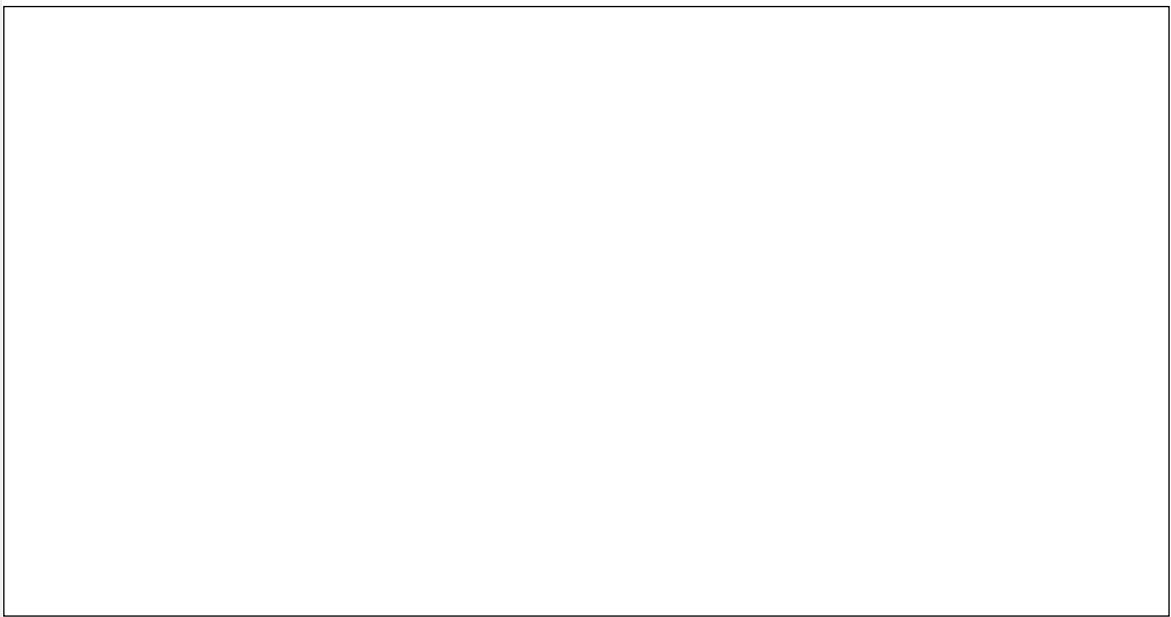
b) Suivant ce plan d'exécution, y a-t-il un index sur l'attribut nomprof ? Justifier.

3. Soit l'arbre algébrique suivant

a) A quelle requête correspond cet arbre (en langage naturel et en SQL)



b) Optimisez cet arbre.



**12.** Donnez un arbre algébrique **optimisé** pour la requête suivante : « **les numéros et les disciplines de cours de niveau débutant enseignés par Bertin après 2000** ».

**N.B.** La requête devra être réalisée **sans requête imbriquée**

