

7.3 Série de TD N 03

Exercice 1 :

1. Commenter et donner le résultat de l'exécution des deux transactions ci-dessous ?

<pre>BEGIN TRANSACTION TRAN1; CREATE TABLE T1 (A INTEGER); CREATE TABLE T2 (A INTEGER); CREATE TABLE T3 (A INTEGER); INSERT INTO T1 VALUES (1); INSERT INTO T2 VALUES (1); INSERT INTO T3 VALUES (1); ROLLBACK;</pre>	<pre>BEGIN TRANSACTION Tr1; CREATE TABLE T1 (A INTEGER); CREATE TABLE T2 (A INTEGER); SAVE TRAN P1; CREATE TABLE T3 (A INTEGER); INSERT INTO T1 VALUES (1); INSERT INTO T2 VALUES (1); INSERT INTO T3 VALUES (1); ROLLBACK TRAN P1; COMMIT;</pre>
---	---

2. Soit la table Film : Film(idF, entrees) définie en relationnel permettant d'enregistrer le nombre d'entrées des films identifiés par leur idF.

Temps	Transaction Tr1	Transaction Tr2
t0	BEGIN TRANSACTION Tr1	
t1		BEGIN TRANSACTION Tr2
t2	UPDATE Film SET entrees=entrees+1 WHERE idF ='3'	
t3		UPDATE Film SET entrees=entrees+1 WHERE idF ='3'
t4	COMMIT	
t5		COMMIT

- a. Du point de vue de la transaction Tr1, puis Tr2, de combien les entrées du film 3 ont-t-elles été augmentées à t2, t3, t4 et t5.
- b. Après exécution des deux transactions, la base de données est-elle dans un état cohérent ? est-elle dans un état correct ? expliquer ?.

Exercice 2 :

1. Soit une table Gardien ayant initialement 0 tuples. Commenter chaque action dans les deux transactions ci-dessous, et dire quel sera le nombre final des tuples après leur exécution ?

BEGIN TRAN ; INSERT INTO GARDIEN VALUES (80,'Durand','aa','16/01/1990') ; SELECT * FROM GARDIEN ; ROLLBACK ; SELECT * FROM GARDIEN ;	BEGIN TRAN ; INSERT INTO GARDIEN VALUES (80,'Ali','aa','16/01/1990') ; SELECT * FROM GARDIEN ; COMMIT ; SELECT * FROM GARDIEN ;
---	--

2. Ecrire une transaction contenant un BEGIN TRAN, puis 2 insertions dans Gardien (avec nom= Nom1 et Nom2), suivies d'un COMMIT, suivi d'une insertion (nom=Nom3), une mise à jour (sur date de naissance) et une destruction d'un tuple, suivie d'un BEGIN TRAN, suivi d'une sélection complète sur la table, suivi d'un point d'arrêt P1, suivi d'une mise à jour (sur date de naissance), suivie d'une insertion d'un tuple, suivi d'une sélection complète sur la table, suivie d'un ROLLBACK P1, suivi d'une sélection complète sur la table.
 - a. Donner le nombre de tuples dans la table Gardien à chaque sélection ?
 - b. Donner le nombre final des tuples dans la table Gardien ?
 - c. Donner le nombre de mises à jour validées.

Exercice 3 :

On considère 04 transactions concurrentes qui veulent mettre à jour deux tables Tab1 et Tab2.

T0	UPDATE Tab1 SET nb1 = 15
	UPDATE Tab2 SET nb2 = 20
T1	UPDATE Tab1 SET nb1 = nb1 + 1
	UPDATE Tab2 SET nb2 = nb2 - 1
T2	UPDATE Tab1 SET nb1 = nb1 + 3
	UPDATE Tab2 SET nb2 = nb2 - 3
T3	SELECT nb1 FROM Tab1
	SELECT nb2 FROM Tab2

1. Donner les valeurs finales de nb1 et nb2 dans le cas d'exécution des transactions succession T0, T1, T2 puis T3.

On envisage trois schémas d'exécutions Exécution 1, Exécution 2 et Exécution 3 :

Exécution 1	Exécution 2	Exécution 3
T0 écriture Tab1	T0 écriture Tab1	T0 écriture Tab1
T0 écriture Tab2	T1 lecture Tab1	T0 écriture Tab2
T1 lecture Tab1	T1 écriture Tab1	T3 lecture Tab1
T1 écriture Tab1	T0 écriture Tab2	T3 lecture Tab2
T2 lecture Tab1	T2 lecture Tab1	T1 lecture Tab1
T2 écriture Tab1	T2 écriture Tab1	T1 écriture Tab1
T3 lecture Tab1	T1 lecture Tab2	T2 lecture Tab2
T1 lecture Tab2	T2 lecture Tab2	T2 écriture Tab2
T1 écriture Tab2	T1 écriture Tab2	T1 lecture Tab2
T2 lecture Tab2	T2 écriture Tab2	T1 écriture Tab2
T2 écriture Tab2	T3 lecture Tab1	T2 lecture Tab1
T3 lecture Tab2	T3 lecture Tab2	T2 écriture Tab1

2. Lesquels de ces exécutions sont sérialisables par rapport à la succession T0, T1, T2, T3 ?
3. Donnez les graphes de précédences de ces exécutions et retrouvez les résultats précédents.
4. Intéressons-nous aux exécutions non sérialisables. Est-ce qu'elles peuvent être exécutées par une procédure de verrouillage à deux phases (2PL) utilisant deux

modes : S-lock (lecture partagée) et X-lock (écriture exclusive).

Exercice 4 :

Un ordonnanceur avec verrouillage à 2 phases reçoit l'exécution ci-dessous, de trois transactions T1, T2 et T3, sur deux variables différentes x et y (L = Lecture, E = Ecriture, R = relâcher tous les verrous détenus par la transaction).

Exe : L1[x] L2[y] E3[x] E1[y] E1[x] E2[y] R2 L3[y] L1[y] R1 E3[y] R3

Indiquer l'ordre d'exécution établi par l'ordonnanceur, en considérant que les opérations bloquées en attente d'un verrou seront exécutées en priorité dès que le verrou devient disponible, dans l'ordre de leur blocage. On suppose que les verrous d'une transaction sont relâchés au moment du Commit.