

TD Accès concurrents

Les exercices qui seront traités en TD sont 01, 02 et 04

Exercice 01 :

On considère 3 transactions T_i , T_j , T_k :

T_i	T_j	T_k
READ(A)	READ(A)	WRITE(A)
WRITE(B)	READ(B)	WRITE(B)

L'exécution de la dernière requête (READ ou WRITE) termine la transaction qui a effectué cette requête.

Questions :

1. Décrire les ordonnancements non-série que l'on peut constituer avec T_i et T_j ? Parmi ces ordonnancements, quels sont ceux qui sont sérialisables (justifier) ?
2. Décrire les ordonnancements non-série que l'on peut constituer avec T_i et T_k ? Parmi ces ordonnancements, quels sont ceux qui sont sérialisables (justifier) ?
3. Décrire les ordonnancements non-série que l'on peut constituer avec T_j, T_k ? Parmi ces ordonnancements, quels sont ceux qui sont sérialisables (justifier) ?
4. On considère l'ordonnement de T_i, T_j, T_k suivant :

T_i	T_j	T_k
READ(A)		
		WRITE(A)
	READ(A)	
WRITE(B)		
	READ(B)	
		WRITE(B)

- a. Est-ce cet ordonnancement est sérialisable ?
- b. Décrire comment le mécanisme d'accès par verrouillage à deux phases sérialise cet ordonnancement ?
Déduire l'ordonnement série équivalent à cette sérialisation ?
- c. Montrer qu'en utilisant le mécanisme d'accès par verrouillage à deux phases, aucun ordonnancement de T_i, T_j, T_k ne peut engendrer un blocage mutuel ?

Exercice 02 :

Soit le scénario d'exécution de transactions suivant :

Transaction	Données verrouillées par la transaction	Données que la transaction attend
T1	X2	X1, X3
T2	X3, X10	X7, X8
T3	X8	X4, X5
T4	X7	X1
T5	X1, X5	X3
T6	X4, X9	X6
T7	X6	X5

1. Produisez un graphe d'attente pour ce scénario
2. Déterminez si des verrous indéfinis (deadlock) apparaissent
3. Donner dans ce cas la solution préconisée.

Exercice 03 :

Considérons l'ordonnancement **O** des transactions T1, T2 et T3 initialement arrivées aux temps 10, 20, 30, respectivement.

O: R3(B) R1(A) R2(C) W1(C) W2(B) W2(C) W3(A)

1. Vérifier si O est sérialisable en identifiant les conflits et en construisant le graphe de précédence.
2. Appliquer le protocole de verrouillage à deux phases sur O jusqu'à l'exécution de toutes les actions. Quelle est l'exécution finale obtenue ?
3. Si la stratégie wait-die est utilisée pour gérer les demandes de verrouillage. Quelle est l'exécution finale obtenue ?
4. Si la stratégie wound-wait est utilisée pour traiter les demandes de verrouillage, dans quel ordre les transactions sont finalement validées?

Exercice 04 :

Considérer l'ordonnancement de transactions suivant : A=5; B=10;

Temps	Transaction T1	Transaction T3	Transaction T2
t1	Début Transaction		
t2	Lire (A, a);		
t3	a:=(a*2)-5;		
t4			Début Transaction
t5			Lire (A, a);
t6			Lire (B, b);
t7		Début Transaction	
t8		Lire (A, a);	
t9		Lire (B, b);	
t10		Afficher (a, b);	
t11		Confirmer	
t12	Écrire (a, A);		
t13	Lire (B, b);		
t14	b:=(b*2)-10;		
t15	Écrire (b, B);		
t16	Confirmer		
t17			tmp:= a;
t18			a:= b;
t19			b:= tmp;
t20			Écrire (a, A);
t21			Écrire (b, B);
t22			Confirmer

1. Donner tous les résultats corrects.
2. L'ordonnancement est-il correct? Sérialisable ?
3. Donner le scénario d'exécution de cet ordonnancement dans le cas du verrouillage à deux phases. Construire le graphe d'attente de cet ordonnancement. Existe-t-il un deadlock ? Si oui, proposer une solution à ce problème.
4. Donner le scénario d'exécution de cet ordonnancement en utilisant la technique d'estampillage à deux estampilles.
5. Si la stratégie wait-die est utilisée pour gérer les demandes de verrouillage. Quelle est l'exécution finale obtenue ?
6. Si la stratégie wound-wait est utilisée pour traiter les demandes de verrouillage, dans quel ordre les transactions sont finalement validées?

Exercice 05 :

Un SGBD reçoit la séquence d'opérations suivante, en provenance de trois transactions :

O1: R1(X) R2(X) W2(Y) R3(Y) W3(Z) W1(Z) W2(X) R3(X) W3(X)

L'un des programmes qui s'exécute dans ce SGBD gère les connexions à un système informatique. Ce programme lit le mot de passe de l'utilisateur, le compare avec celui stocké sous forme cryptée dans la base de données écrit les informations de connexion dans le journal de connexions et incrémente la valeur du nombre d'accès réalisés par l'utilisateur dans le système. Le journal de connexions et le nombre d'accès des utilisateurs sont stockés dans la base de données.

1. Est-ce que l'une des transactions ci-dessus peut représenter l'exécution de ce programme de gestion des connexions? Justifier votre réponse.
2. Construire le graphe de précédence de cet ordonnancement. O1 est-il sérialisable ?
3. Donner le scénario d'exécution de cet ordonnancement en appliquant le protocole de verrouillage à deux phases. Justifier.
4. L'algorithme d'estampillage (à deux estampilles) accepte-t-il cet ordonnancement sans rejets ?
5. Soit maintenant l'ordonnancement O2 : **R1(X) R2(X) R3(Y) W2(Y) W3(Z) W1(Z) R3(X) W2(X) W3(X).** Construire le graphe d'attente de cet ordonnancement. Existe-t-il un deadlock ? Si oui, proposer une solution à ce problème.