

TD transaction suite

Exercice 3

Exercice 03 :

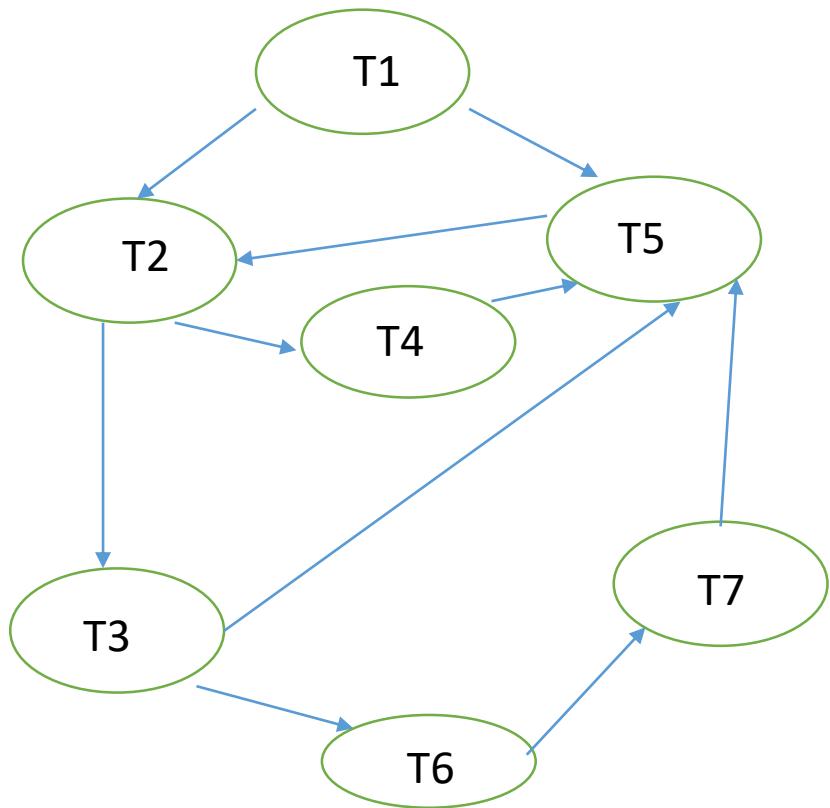
Soit le scénario d'exécution de transactions suivant :

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 3

-



- Sur ce graphe d'attente, on constate la présence de 3 cycles:
- C1: T2 T4 T5 T2
- C2: T2 T3 T5 T2
- C3: T2 T3 T6 T7 T5 T2

Exercice 3

- La solution: casser les cycles en éliminant un minimum de transactions possible, pour cela on va faire l'intersection des cycles pour sélectionner les transactions qui bloquent tous les cycles:
- $C_1 \cap C_2 \cap C_3 : (T_2, T_5)$, on a le choix d'annuler T_5 ou T_2 .
- on choisit la transaction qui a fait le moins de MAJ ou la plus récente, dans notre cas on choisit la plus récente: T_5 ,
- en annulant T_5 , ça permet de casser les 3 cycles et de relancer les transactions une à une à partir de leur point de blocage et à la fin relancer T_5 depuis son début (vu qu'elle a été annulée).
- Si dans un graphe d'attente, on trouve plusieurs cycles mais pas d'intersection entre eux, dans ce cas il faudra éliminer plus d'une transaction (une transaction par cycle pour casser chacun des cycles. Et les relancer depuis leur début à la fin.

Exercice

Considérons l'ordonnancement O des transactions T_1 , T_2 et T_3

$O: R_3(B) R_1(A) R_2(C) W_1(C) W_2(B) W_2(C) W_3(A)$

1. Vérifier si O est sérialisable en identifiant les conflits et en construisant le graphe de précédence

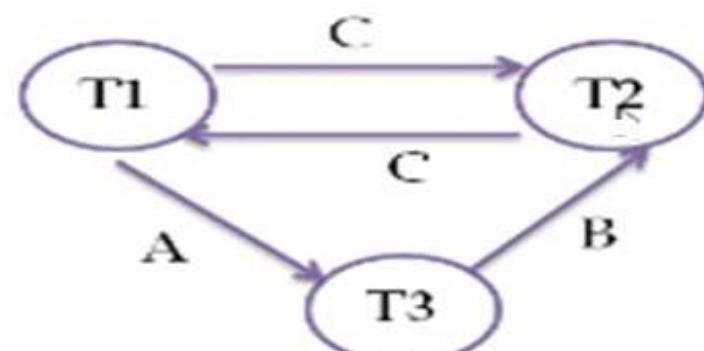
■ Les conflits :

sur B : $R_3(B)$ - $W_2(B)$

sur A : $R_1(A)$ - $W_3(A)$

Sur C : $R_2(C)$ - $W_1(C)$, $W_1(C)$ - $W_2(C)$

■ Graphe de précédence



Le graphe de précédence est cyclique car on a le cycle $\{T_1, T_2, T_1\}$ d'où O n'est pas sérialisable.

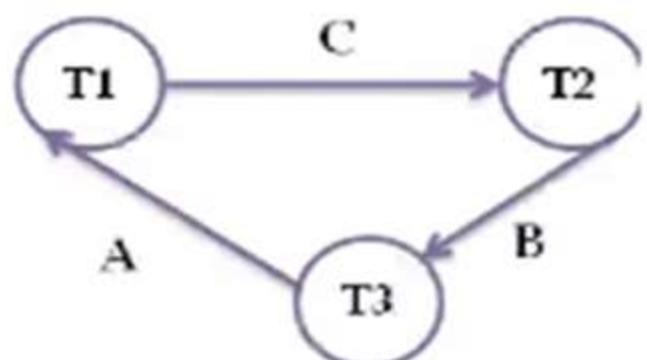
O: R3(B) R1(A) R2(C) W1(C) W2(B) W2(C) W3(A)

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 ?

Transaction	Action	Demande Verrou	Réponse
T3	R3(B)	SLOCK(B)	OK
T1	R1(A)	SLOCK(A)	OK
T2	R2(C)	SLOCK(C)	OK
T1	W1(C)	XLOCK(C)	Non, T1 attend la fin de T2
T2	W2(B)	XLOCK(B)	Non, T2 attend la fin de T3
T2	W2(C)	XLOCK(C)	T2 est en attente
T3	W3(A)	XLOCK(A)	Non, T3 attend la fin de T1

■ Graphe d'attente

- ✓ De graphe d'attente, on constate qu'on a un cycle T1-T2-T3-T1 d'où on a un inter-blockage.
- ✓ T3 est la victime car elle est la plus récente.
- ✓ Rollback(T3) → annuler T3 et libérer les verrous(unlock(B));
- ✓ Reprendre T2 à partir de point de blockage;
- ✓ T3 sera relancée dès son début à la fin.



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

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 ?

Transaction	Action	Demande Verrou	Réponse
Rollback(T3): UNLOCK(B) et reprise de T2			
T2	W2(B)	XLOCK(B)	OK
T2	W2(C)	XLOCK(C)	OK, fin de T2 d'où UNLOCK(C), UNLOCK(B), réveiller T1.
Reprise de T1			
T1	W1(C)	XLOCK(C)	OK, fin de T1, UNLOCK(A), UNLOCK(C).
Relancer T3 dès son début :			
T3	R3(B)	SLOCK(B)	OK
T3	W3(A)	XLOCK(A)	OK, fin de T3, UNLOCK(A), UNLOCK(B).

- L'exécution finale obtenue est T2, T1, T3

Exercice 2

Exercice 02 :

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	Ecrire (a, A);		
t13	Lire (B, b);		
t14	b:=(b*2)-10;		
t15	Ecrire (b, B);		
t16	Confirmer		
t17			tmp:= a;
t18			a:= b;
t19			b:= tmp;
t20			Ecrire (a, A);
t21			Ecrire (b, B);
t22			Confirmer

1. Donner tous les résultats corrects.
2. L'ordonnancement est-il correct? Sérialisable?

Exercice 2 A=5, B=10; T1,T3,T2

Temps	Transaction T1	Transaction T3	Transaction T2
t1	Début Transaction		
t2	Lire (A, a); (a= 5)		
t3	a:=(a*2)-5; (a= 5)		
t4			Début Transaction
t5			Lire (A, a); (a= 5)
t6			Lire (B, b); (b=10)
t7		Début Transaction	
t8		Lire (A, a); (a= 5)	
t9		Lire (B, b); (b=10)	
t10		Afficher (a, b);	
t11		Confirmer	
t12	Écrire (a, A); (A=5)		
t13	Lire (B, b); (b=10)		
t14	b:=(b*2)-10; (b=10)		
t15	Écrire (b, B); (B=10)		
t16	Confirmer		
t17			tmp:= a; (temp=5)
t18			a:= b; (a= 10)
t19			b:= tmp; (b=5)
t20			Écrire (a, A); (A=10)
t21		K.KESSI ASGBD M1	Écrire (b, B); (B=5)
t22			Fin

Exercice 2

T2, T3, T1

Temps	Transaction T1	Transaction T3	Transaction T2
t1	Début Transaction		
t2	Lire (A, a); (a=10)		
t3	a:=(a*2)-5; (a= 15)		
t4			Début Transaction
t5			Lire (A, a); (a= 5)
t6			Lire (B, b); (b=10)
t7		Début Transaction	
t8		Lire (A, a); (a= 10)	
t9		Lire (B, b); (b=5)	
t10		Afficher (a, b);	
t11		Confirmer	
t12	Écrire (a, A); (A=15)		
t13	Lire (B, b); (b=5)		
t14	b:=(b*2)-10; (b=0)		
t15	Écrire (b, B); (B=0)		
t16	Confirmer		
t17			tmp:= a; (temp=5)
t18			a:= b; (a= 10)
t19			b:= tmp; (b=5)
t20			Écrire (a, A); (A=10)
t21		K.KESSI ASGBD M1	Écrire (b, B); (B=5)
t22			Confirmer

Exercice 2

	A	B
T1 T2 T3	10	5
T1 T3 T2	10	5
T2 T1 T3	15	0
T2 T3 T1	15	0
T3 T1 T2	10	5
T3 T2 T1	15	0

Les résultats corrects sont obtenus à partir des exécutions série des transactions, si on a n transactions, on aura alors $n!$ résultats corrects possibles. On a 3 transactions on a alors $3! = 3*2*1=6$ résultats corrects possibles.

T3 ne modifie pas la base de données, donc c'est comme si qu'on avait à chaque fois soit T1 T2 ou T2 T1.

Exercice 2

- L'ordonnancement est-il correct? Sérialisable ?

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

Exercice 2

- L'ordonnancement nous donne A=10, B=5, ce résultat appartient au aux résultats corrects, cet ordonnancement est alors correct
- L'ordonnancement est-il Sérialisable ?
- R1(A) R2(A) R2(B) R3(A) R3(B) W1(A) R1(B) W1(B) W2(A) W2(B)
- Dessiner le graphe de précédence en listant les conflits sur les granules A et B.

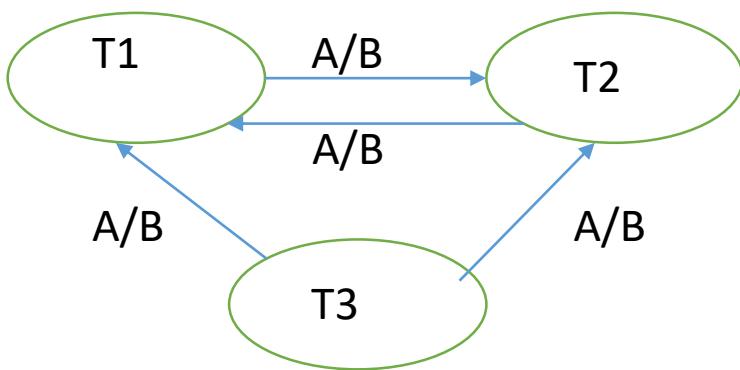
Exercice 2

- R1(A) R2(A) R2(B) R3(A) R3(B) W1(A) R1(B) W1(B) W2(A) W2(B)
- Conflits sur A: R1(A) – W2(A) T1 précède T2
- R2(A) – W1(A) T2 précède T1
- R3(A) – W1(A) T3 précède T1
- R3(A) – W2(A) T3 précède T2
- W1(A) – W2(A) T1 précède T2

Exercice 2

- R1(A) R2(A) R2(B) R3(A) R3(B) W1(A) R1(B) W1(B) W2(A) W2(B)
- Conflits sur B: R2(B) – W1(B) T2 précède T1
- R3(B) – W1(B) T3 précède T1
- R3(B) – W2(B) T3 précède T2
- R1(B) – W2(B) T1 précède T2
- W1(B) – W2(B) T1 précède T2

Exercice 2



- On a un cycle dans ce graphe de précédence: T1 T2 T1, cet ordonnancement n'est pas sérialisable.
- Si un ordonnancement est sérialisable, il est alors correct.
- Si un ordonnancement est correct, il n'est pas forcément sérialisable.
- Si un ordonnancement n'est pas correct alors il n'est pas sérialisable.

Exercice 2

- Donner le scenario 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.
- Donner le scénario d'exécution de cet ordonnancement en utilisant la technique d'estampillage à deux estampilles.

Exercice 2

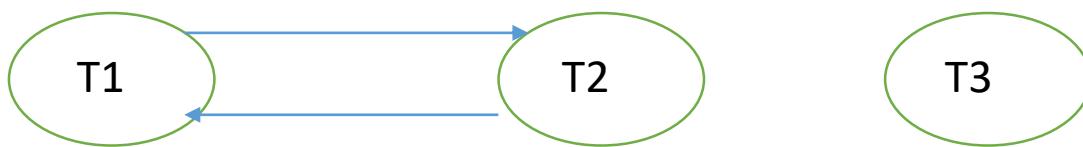
- Donner le scenario d'exécution de cet ordonnancement dans le cas du verrouillage à deux phases. Construire le graphe d'attente de cet ordonnancement. Existe-t-il un deadlock ? Si oui, proposer une solution à ce problème.
- R1(A) R2(A) R2(B) R3(A) R3(B) (**fin T3**) W1(A) R1(B) W1(B) (**fin T1**) W2(A) W2(B) (**fin T2**)

Exercice 2

- R1(A) R2(A) R2(B) R3(A) R3(B) (**fin T3**) W1(A) R1(B) W1(B) (**fin T1**) W2(A) W2(B) (**fin T2**)
- T1: R1(A) Slock (A), l'obtient.
- T2: R2(A) Slock (A), l'obtient.
- T2: R2(B) Slock (B), l'obtient.
- T3: R3(A) Slock (A), l'obtient.
- T3: R3(B) Slock (B), l'obtient, T3 se termine et libère les verrous.
- T1: W1(A) Xlock (A), Non T1 attend T2
- T1: R1(B) Slock(B), T1 est en attente
- T1: W1(B) Xlock(B), T1 est en attente
- T2: W2(A) Xlock(A), Non T2 attend T1
- T2: W2(B) Xlock(B), T2 est en attente

Exercice 2

- Graphe d'attente:



- Dans ce graphe d'attente, on a un cycle $T1 \rightarrow T2 \rightarrow T1$, il y a alors un inter blocage (deadlock).
- La solution c'est de casser tous les cycles en annulant un minimum de transaction,
- ici on a un cycle, on va alors annuler la transaction la plus récente de ce cycle : $T2$ afin de pouvoir reprendre $T1$ à partir de son point de blocage relancer $T2$ depuis son début (vu qu'elle a été annulée).

Exercice 2

- R1(A) R2(A) R2(B) R3(A) R3(B) W1(A) R1(B) W1(B) W2(A) W2(B)
- Annuler T2, Rollback T2, Unlock(A), Unlock (B)
- Reprise de T1:
 - T1: W1(A) Xlock (A), l'obtient
 - T1: R1(B) Slock(B), l'obtient
 - T1: W1(B) Xlock(B), l'obtient, fin de T1, libère les verrous.
- Relancer T2 depuis son début:
 - T2: R2(A) Slock (A), l'obtient.
 - T2: R2(B) Slock (B), l'obtient.
 - T2: W2(A) Xlock(A), l'obtient.
 - T2: W2(B) Xlock(B), l'obtient.
- L'exécution finale obtenue est: T3 T1 T2

Exercice 2

- Donner le scénario d'exécution de cet ordonnancement en utilisant la technique d'estampillage à deux estampilles.
- Principe: Si une transaction tente de lire ou d'écrire une donnée, alors la lecture ou l'écriture n'est autorisée que si la dernière mise à jour de la donnée a été effectuée par une transaction plus **ancienne**. Sinon la transaction qui a demandé la lecture ou l'écriture est reprise avec une nouvelle estampille.

Exercice 2

- EE(A): Estampille d'écriture de A
 - EL(A): Estampille de lecture de A
 - E(T): Estampille de la transaction T
-
- T read A R(A):
 - Si $E(T) < EE(A)$ rejeter T et reprise avec nouvelle estampille.
 - Sinon $E(T) \geq EE(A)$, exécution de R(A) et $EL(A) = \max(E(T), EL(A))$

 - T write A W(A):
 - Si $E(T) < EL(A)$ rejeter T et reprise avec nouvelle estampille
 - Si $E(T) < EE(A)$ rejeter T et reprise avec nouvelle estampille
 - Sinon ($E(T) \geq EL(A)$ et $E(T) \geq EE(A)$) exécution de W(A) et $EE(A) = E(T)$

Exercice 2

- R1(A) R2(A) R2(B) R3(A) R3(B) W1(A) R1(B) W1(B) W2(A) W2(B),
- Au début EE(A), EL(A), EE(B), EL(B) sont tous à zéro.

		EL(A)	EE(A)	EL(B)	EE(B)
		0	0	0	0
E(T)>EE(A)(1>0), EL(A)=1	R1(A)	1	0	0	0
E(T)>EE(A)(2>0), EL(A)=2	R2(A)	2	0	0	0
E(T)>EE(B)(2>0), EL(B)=2	R2(B)	2	0	2	0
E(T)>EE(A)(3>0), EL(A)=3	R3(A)	3	0	2	0
E(T)>EE(B)(3>0), EL(B)=3	R3(B)	3	0	3	0
E(T) < EL(A) (1<3): T1 annulée reprise avec nouvelle estampille T4	W1(A)	1<3	0	3	0

Exercice 2

R1(A) R2(A) R2(B) R3(A) R3(B) W1(A) R1(B) W1(B) W2(A) W2(B)

		EL(A)	EE(A)	EL(B)	EE(B)
T1 annulée	R1(B)	3	0	3	0
T1 annulée	W1(B)	3	0	3	0
E(T) < EL(A) (2<3): T2 annulée reprise avec nouvelle estampille T5	W2(A)	2<3	0	3	0
T2 annulée	W2(B)	3	0	3	0
Relancer T1 avec T4 E(T) >= EE(A), 4>0, EL(A)=4	R4(A)	4	0	3	0
4>= EL(A)=4 et 4> EE(A)=0, EE(A)=4	W4(A)	4	4	3	0
4>0, EL(B)=4	R4(B)	4	4	4	0
4>= EL(B)=4 et 4> EE(B)=0, EE(A)=4	W4(B)	4	4	4	4
Relancer T2 avec T5 E(T) >= EE(A), 5>4, EL(A)=5	R5(A)	5	4	4	4
5>4, EL(B)=5	R5(B)	5	4	5	4
5>= EL(A)=5 et 5>= EE(A)=4, EE(A)=5	W5(A)	5	5	5	4
5>= EL(B)=5 et 5>= EE(B)=4, EE(B)=5	W5(B)	5	5	5	5