

SGBD : BASES DE DONNÉES AVANCÉES [M3106C]

TD N°3 - ACCÈS CONCURRENTS

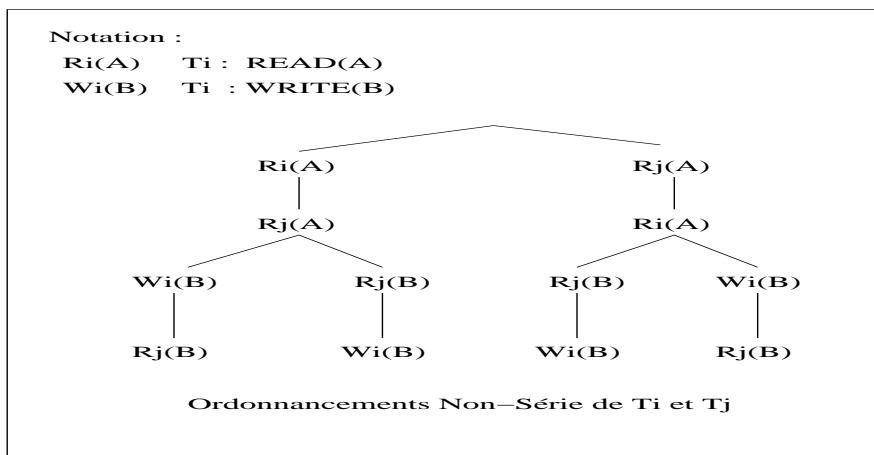
OBJECTIFS

- Ordonnements sérialisables
- Méthode d'accès par verrouillage à deux phases
- Méthodes d'accès par estampillage à l'initialisation

CORRIGÉS

Exercice I :

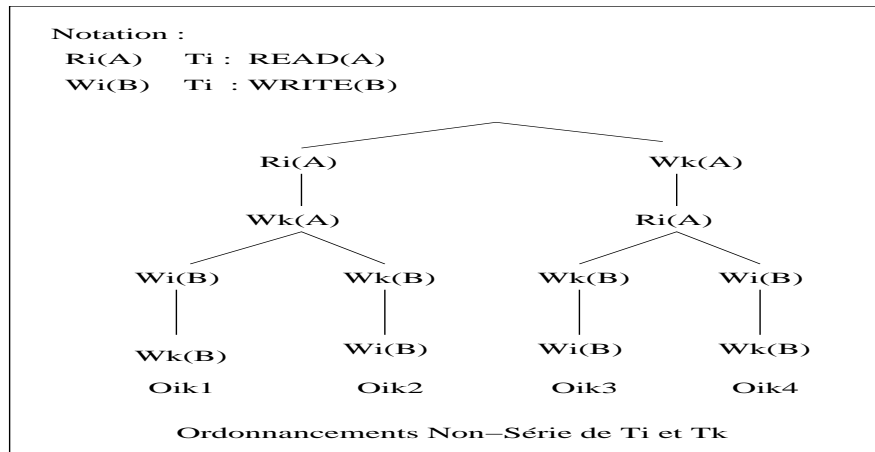
Question 1.1.



Il y a 4 ordonnancements : tous sont sérialisables.

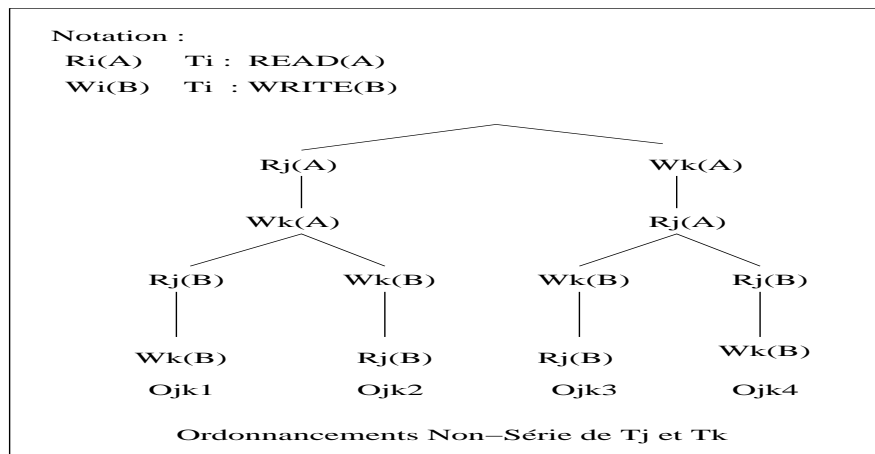
Question 1.2.

Date: 9 septembre 2014.
Hocine ABIR - IUT Villetaneuse .



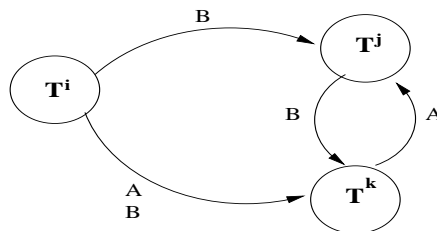
Il y a 4 ordonnancements : Seuls Oik1 et Oik3 sont sérialisables.

Question 1.3.



Il y a 4 ordonnancements : Seuls Ojk1 et Ojk3 sont sérialisables.

Question 1.4.



Question 1.5.

On note \llbracket les verrous acquis et $\{\}$ les verrous non-acquis

T_i	T_j	T_k	A	B
READ(A)			$[T_i, R]$	
		WRITE(A)	$[T_i, R]$ $\{T_k, W\}$	
	READ(A)		$[T_i, R]$ $[T_j, R]$ $\{T_k, W\}$	
WRITE(B)			$[T_i, R]$ $[T_j, R]$ $\{T_k, W\}$	$[T_i, W]$
commit			$[T_j, R]$ $\{T_k, W\}$	
	READ(B)		$[T_j, R]$ $\{T_k, W\}$	$[T_j, R]$
	commit		$[T_k, W]$	
		WRITE(A)	$[T_k, W]$	
		WRITE(B)	$[T_k, W]$	$[T_k, W]$
		commit		

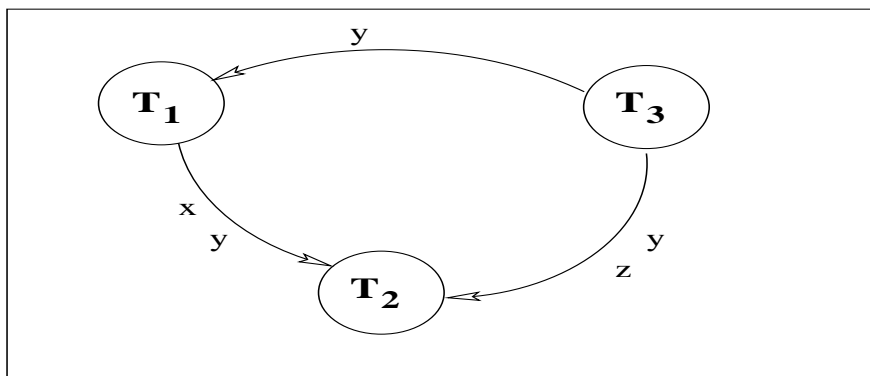
L'ordonnancement série est : T_i, T_j, T_k

Question 1.6.

l'accès aux ressources A et B se fait dans le même ordre pour les 3 transactions.

Exercice II :

Question 2.1.



Question 2.2.

- S est sérializable : car son graphe de précédence ne contient pas de cycle.
- S est équivalent à l'ordonnancement série $\langle T_3, T_1, T_2 \rangle$

Question 2.3.

Temps	T_1	T_2
1	READ(X)	
2	WRITE(X)	
3		READ(Z)
4		READ(Y)
5		WRITE(Y)
6	READ(Y)	
7	WRITE(Y)	
8		READ(X)
9		WRITE(X)

Exercice III :

Question 3.1.

(1) Solution 1

Dans l'ordonnement série $\langle T_{i1}, T_{i2} \rangle$, l'opération de (2) de T_{i1} et l'opération (3) T_{i2} (voir tableau ci-dessous) ne sont pas permutable donc aucun ordonnancement n'est sérialisable.

Opération	T_{i1}	T_{i2}
1	READ(Y)	
2	WRITE(Y)	
3		READ(Y)
4		WRITE(Y)

idem pour $\langle T_{i2}, T_{i1} \rangle$

(2) Solution 2

Enumérer tous les ordonnancements non-séries et vérifier que chacun d'eux comporte un cycle dans le graphe de précedence :

#	Ordonnement	Cycle
1	$R_{i1}(Y), R_{i2}(Y), W_{i1}(Y), W_{i2}(Y)$	$R_{i1}(Y) < W_{i2}(Y)$ et $R_{i2}(Y) < W_{i1}(Y)$
2	$R_{i1}(Y), R_{i2}(Y), W_{i2}(Y), W_{i1}(Y)$	$R_{i1}(Y) < W_{i2}(Y)$ et $R_{i2}(Y) < W_{i1}(Y)$
3	$R_{i2}(Y), R_{i1}(Y), W_{i2}(Y), W_{i1}(Y)$	$R_{i2}(Y) < W_{i1}(Y)$ et $R_{i1}(Y) < W_{i2}(Y)$
4	$R_{i2}(Y), R_{i1}(Y), W_{i1}(Y), W_{i2}(Y)$	$R_{i2}(Y) < W_{i2}(Y)$ et $R_{i1}(Y) < W_{i2}(Y)$

Question 3.2.

De chaque ordonnancement série de T_{j1} et T_{j2} , on peut construire deux ordonnancements non-séries sérialisables équivalents.

#	Ord Série	Ordonnements Non-Série Sérialisables
1	$\langle T_{j1}, T_{j2} \rangle$	$w_{j1}(X), R_{j1}(Y), w_{j2}(X), W_{j1}(Y), R_{j2}(Y), W_{j2}(Y)$
2		$w_{j1}(X), w_{j2}(X), R_{j1}(Y), W_{j1}(Y), R_{j2}(Y), W_{j2}(Y)$
3	$\langle T_{j2}, T_{j1} \rangle$	$w_{j2}(X), R_{j2}(Y), w_{j1}(X), W_{j2}(Y), R_{j1}(Y), W_{j1}(Y)$
4		$w_{j2}(X), w_{j1}(X), R_{j2}(Y), W_{j2}(Y), R_{j1}(Y), W_{j1}(Y)$

Autre Solution :

Enumérer tous les ordonnancements (Il y a au total 18) et vérifier pour chacun d'eux la sérialisabilité.

Question 3.3.

T_{j1}	T_{j2}	X	Y
WRITE(X)	WRITE(X)	[j1,W]	
READ(Y)		[j1,W] , {j2,W}	[j1,R]
WRITE(Y)		[j1,W] , {j2,W}	[j1,R] , [j1,W]
COMMIT		[j2,W]	
	WRITE(X)	[j2,W]	
	READ(Y)	[j2,W]	[j2,R]
	WRITE(Y)	[j2,W]	[j2,R] , [j2,W]
	COMMIT		

Question 3.4.

- La transaction T_j débute avec un WRITE(X),
- L'occurrence de T_j qui débute obtient un verrou WRITE (exclusif) sur X et bloque la deuxième occurrence jusqu'à son COMMIT.

Question 3.5.

- tous les ordonnancements non série entraînent un blocage mutuel
- La transaction T_k débute avec un READ(Y),
- Chaque occurrence de T_k obtient un verrou READ (partagé) sur Y et les deux occurrences s'attendent mutuellement.
- Exemple :

T_{k1}	T_{k2}	X	Y
READ(Y)	READ(Y)		[j1,R]
WRITE(Y)			[j1,R] , [j2,R]
	WRITE(Y)		[j1,R] , [j2,R] , {j1,W}
...			[j1,R] , [j2,R] , {j1,W} , {j2,W}
			blocage mutuel