

## Exercice2 : Gestion de transactions

A. Soient quatre transactions définies comme suit sur trois granules de base (A, B, C) : (on ne représente ici que les opérations de lecture et écriture) :

T1	T2	T3	T4
DebT1	DebT2	DebT3	DebT4
1:R1(A)	3:W2(C)	6:R3(A)	10:R4(C)
2:W1(A)	4:R2(B)	7:W3(A)	11:W4(C)
FinT1	5:W2(B)	8:R3(B)	12:R4(A)
	FinT2	9:W3(B)	13:W4(A)
		FinT3	FinT4

Et soit l'ordonnancement suivant :

R3(A)W3(A)R1(A)W2(C)R2(B)R4(C)R3(B)W4(C)W3(B)W1(A)W2(B)  
R4(A)W4(A)

1. Sachant qu'il est correct, l'ordonnancement est-il sérialisable ?
2. Si oui, indiquez l'ordonnancement séquentiel équivalent :

*Remarque : Toutes les réponses devront être justifiées*

On rajoute maintenant deux transactions T5 et T6 et on propose l'ordonnancement suivant

T5	T6
DebT5	DebT6
14:R5(A)	18:R6(C)
15:W5(A)	19:W6(C)
16:R5(C)	FinT6
17:W5(C)	
FinT5	

R3(A)W3(A)R1(A)R6(C)W2(C)R2(B)R4(C)R3(B)W4(C)W2(B)W6(C)R  
5(A)W5(A)W1(A)R5(C)W5(C)W3(B)R4(A)W4(A)

3. En utilisant les primitives de verrouillage, et en appliquant le verrouillage à deux phases, construire le graphe d'attente. Que constatez-vous ? Quelles solutions proposez-vous pour rendre l'ordonnancement exécutable ?

*Bonne chance*



EXERCICE 1 :

Soit la base de données d'un club équestre composée des relations suivantes :

Propriétaire (Num-prop, nom, adr)

Jockey (num-jockey, nom, adr)

Cheval (id-cheval, num-prop)

Victoire (nom-course, date, id-cheval, num-jockey)

QUESTIONS :

- 1) L'administrateur de la BD crée un utilisateur AGENT qui peut lire le contenu de la table *Propriétaire* et modifier les tables *Jockey*, *Cheval* et *Victoire*. Donner les requêtes utilisées par l'administrateur pour répondre à ce besoin ?
- 2) Comment modifier cette base de données afin de connaître le nombre de victoires d'un jockey et d'un cheval sans avoir à les calculer ?
- 3) Donner les requêtes permettant ces modifications et quelles sont les conséquences de ces requêtes sur les différents catalogues.
- 4) Que faut-il associer comme actions, à quelle relation et à quel moment, pour que ces nombres soient connus lors de l'interrogation de cette base de données ? Ecrire les requêtes SQL correspondantes.
- 5) Des contrôles de dopage de chevaux sont effectués systématiquement. Dans le cas de contrôle de dopage positif pour un gagnant (jockey, cheval), la victoire déjà enregistrée est annulée et attribuée au gagnant suivant. Que faut-il faire pour que cette réattribution garde la BD cohérente. Ecrire les requêtes correspondantes ?
- 6) Les clés primaires des relations *Propriétaire*, *Jockey*, *Cheval* et *Victoire* sont les attributs soulignés.
  - a. Quel index devrait être créé sur la relation *Jockey* ?
  - b. Si  $n$  est le nombre de pages du fichier correspondant à la relation *Jockey*, et que la hauteur du B-arbre associé à cette relation est 3, quel est le nombre d'accès aux informations sur un jockey quand le numéro est donné ? quand le nom est donné ?
  - c. Si cette dernière requête (par nom) se répète fréquemment, quelle solution proposeriez-vous pour améliorer l'accès. Donner la requête en SQL correspondant à votre solution et quelles sont ses conséquences sur les différents catalogues.
  - d. Quel module du SGBD va prendre en charge ces différents accès.

- 1) Grant Read on Propriétaire to Agent;
- 2) Grant Update on Jockey, cheval, victoire to Agent;

On ajoute dans la relation Jockey NBVJ  
 et dans la relation cheval NBVC qui donne respectivement  
 le Nombre de victoires pour un Jockey (resp cheval).

- i) Alter table Jockey <sup>AS</sup> ADD Field (NBVJ Integer);
- Alter table cheval <sup>AS</sup> add Field (NBVC Integer);

Conséquences sur le Catalogue:

1/ insertion de 2 tuple dans le catalogue Attribut  
 correspondant à la description des attributs NBVJ <sup>et NBVC</sup>

2/ Cardinalité de la relation "Attribut" de le catalogue  
relation augmente de 2.

3/ le degré de la relation Jockey augmente de 1 <sup>et</sup>  
 = cheval = 1

4/ la taille d'un tuple de la relation Jockey augmente <sup>et</sup>  
 de 2 car (si on considère que la taille d'un entier est 2)

= cheval =

Il faut créer un Trigger lors de l'ajout (insert) d'un  
 tuple dans la relation victoire, on augmente de 1 NBVJ de  
 jockey et NBVC dans cheval

Define Trigger Victoires on Insert of victoire

(update Jockey set NBVJ = NBVJ + 1

where NumJockey = New.Victoire  
 NumJockey)

Idem avec Cheval



⑤ Il faut faire un Delete dans la relation Victoire pour suppr la victoire au couple (Jockey, cheval) correspondant

Associer un Trigger au même moment faire un Insert pour donner la victoire au couple (Jockey, cheval) arrivé à 2<sup>e</sup> position par lequel le Trigger de la question 4 met à jour NBVJ et NBVC le Trigger de suppression est :

Defme Trigger Victoir Sup on Delet of Victoir  
(Update Jockey set NBVJ = NBVJ - 1  
where NumJockey = 0 and Victoire.  
NumJockey)  
idem pour cheval

⑥ Pour la relation Jockey on cree un index primaire sur le  
a) NumJockey comme suit :

c.s. Create unique Index XJockey on Jockey (NumJockey)

b) Pour le nbre d'accès à un jockey quand le nom est donné = 3  
c.s. quand le nom est donné, et seulement que les 2 accès  
(seq et direct) sont permis dans un B. arbre le nbre d'accès  
est  $n + 3$  c.s

c) Si une recherche par nom est fréquente on cre un index  
secondaire - Create index Xnonjockey on Jockey (Nom); c.s

Cette requête a par conséquent, l'insertion d'un tuple c.s  
dans le catalogue index et la cardinalité de l'index dans (c.s)  
le 2d index augmente de 1.

d) C'est l'administrateur qui définit les différents accès aux  
relations d'une BD et l'optimiseur qui choisit parmi les  
chemins existants le meilleur chemin.



## Correction exercice 2 :

Première méthode : Juste en se basant sur l'ordre chronologique des actions de chaque transaction

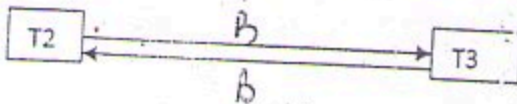
Temps	T1	T2	T3	TA
T			R3(A)	
T+1			W3(A)	
T+2	R1(A)			
Etc.		W2(C)		
		R2(B)		
			R3(B)	R4(C)
			W3(B)	W4(C)
	W1(A)			
		W2(B)		
T+12				R4(A)
				W4(A)

Nous constatons qu'il y a recouvrement de T2 par T3, donc ordonnancement non sérialisable

Deuxième méthode : Graphe de précédence :

R3(A)W3(A)R1(A) W2(C)R2(B)R4(C)R3(B)W4(C) W3(B) W1(A) W2(B)  
R4(A) W4(A)

R2(B) precede W3(B) R3(B) precede W2(B)



3ème Méthode : Par permutation des opérations compatibles et permutable

R3(A)W3(A)R1(A) W2(C)R2(B)R4(C)R3(B)W4(C) W3(B) W1(A) W2(B)  
R4(A) W4(A) : permutation de R4(C) et R3(B) granules différents

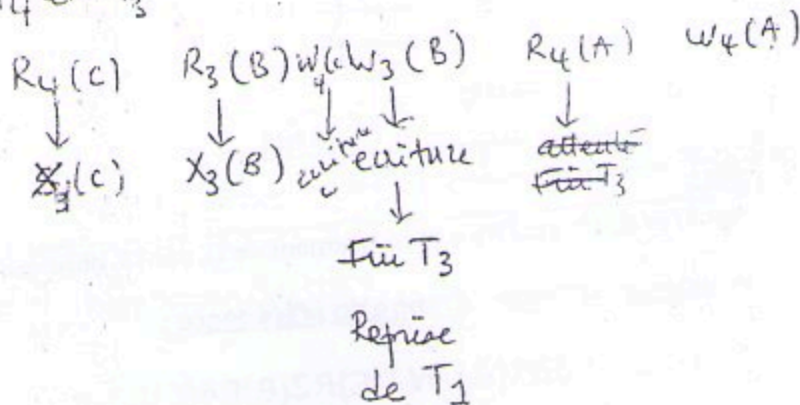
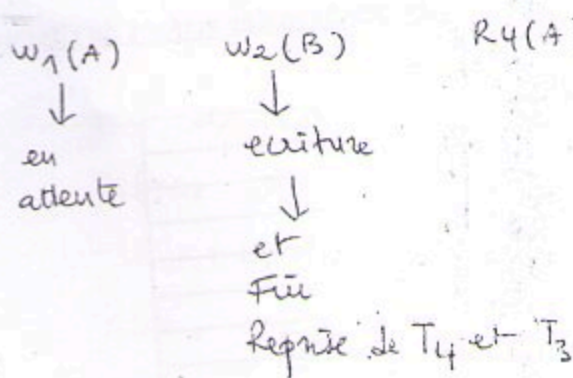
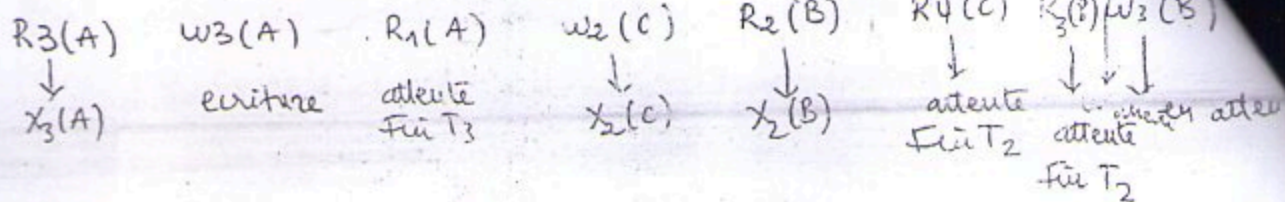
R3(A)W3(A)R1(A) W2(C)R2(B) R3(B) R4(C)W4(C) W3(B) W1(A) W2(B)  
R4(A) W4(A) : permutation de W3(B) avec W4(C) et R4(C)

R3(A)W3(A)R1(A) W2(C)R2(B) R3(B) W3(B) R4(C)W4(C) W1(A) W2(B)  
R4(A) W4(A) : On ne peut plus faire remonter W3(B) pour reconstruire la séquence de

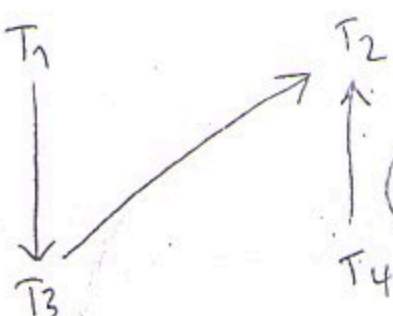
T3, car R2(B) et W3(B) non permutable, et on ne peut pas permuter W3(A) et R1(A) pour faire descendre les opérations de T3, donc on ne peut pas retrouver un ordonnancement séquentiel.

1)

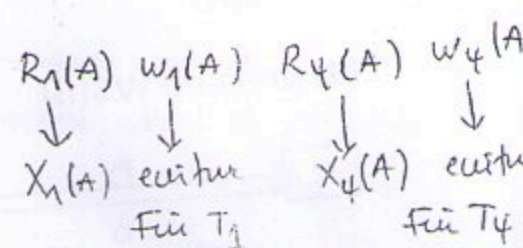
verrouillage à 2VR



2pts



2pts





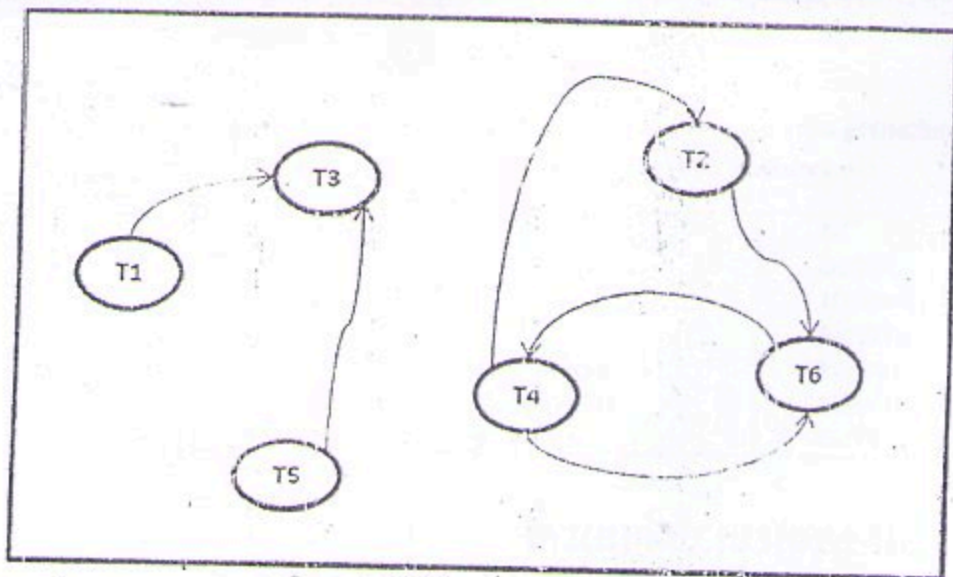
2

Ancien VLP

Transaction	Action	Verrou	Résultat
T3	R3(A)	SLOCK(A)	OK
T3	W3(A)	XLOCK(A)	OK
T1	R1(A)	SLOCK(A)	OK
T6	R6(C)	SLOCK(C)	T1 attend T3
T2	W2(C)	XLOCK(C)	OK
T2	R2(B)	SLOCK(B)	T2 attend T6
			T2 attend T6 puisque T2 est bloquée sur l'action W2(C) OK
T4	R4(C)	SLOCK(C)	Non, R4(C) sera exécutée parce que T6 possède un verrou partagé (R6(C)) et non pas exclusif
T3	R3(B)	SLOCK(B)	OK
T4	W4(C)	XLOCK(C)	OK
T2	W2(B)	XLOCK(B)	T4 attend T6
			Ici, T2 attend T6 puisqu'il est toujours bloqué dans W2(C)
T6	W6(C)	XLOCK(C)	Non, ici W6(C) ne peut pas être exécuté parce que T4 possède un verrou partagé (R4(C)) Donc, T6 Attend T4
T5	R5(A)	SLOCK(A)	Ici, T5 attend T3 seulement. Puisque T1 ne possède qu'un verrou partagé
T5	W5(A)	XLOCK(A)	T5 attend T3
T1	W1(A)	XLOCK(A)	T1 attend T3
T5	R5(C)	SLOCK(C)	T5 est bloqué dans R5(A), T5 attend T3
T5	W5(C)	XLOCK(C)	T5 est bloqué dans R5(A), T5 attend T3
T3	W3(B)	XLOCK(B)	T3 attend T2
T4	R4(A)	SLOCK(A)	OK, Fin T3, Unlock(A,B)
T4	W4(A)	XLOCK(A)	T4 est bloquée dans W4(C), T4 attend T6
			T4 est bloquée dans W4(C), T4 attend T6







lution 1 avec ancien V2P

$R_3(A) \downarrow S_3(A)$    
  $W_3(A) \downarrow X_3(A)$    
  $R_1(A) \downarrow \text{attente Fin } T_3$    
  $W_2(C) \downarrow X_2(C)$    
  $R_2(B) \downarrow S_2(B)$    
  $R_4(C) \downarrow \text{attente Fin } T_2$

$R_3(B) \downarrow S_3(B)$    
  $W_4(C) \downarrow \text{en attente}$    
  $W_3(B) \downarrow \text{attente Fin } T_2$    
  $W_1(A) \downarrow \text{attente Fin } T_2$    
  $W_2(B) \downarrow \text{attente Fin } T_3$    
  $R_4(A) \downarrow \text{attente Fin } T_3$    
  $W_4(A) \downarrow \text{attente Fin } T_3$

