

Bases de Données

Travaux dirigés
Gestion de la Concurrency

Exercice 1 :

Soit l'exemple de transfert entre deux comptes bancaires réalisé par la procédure *Dépôt* et l'exemple de calcul de la somme des soldes de deux comptes réalisée par la procédure *ImpSomme*.

```
procedure Dépôt(num, Montant)
begin
Start;
temp := Read(Comptes(num));
temp := temp + Montant;
Write(Comptes(num),temp);
Commit;
end;
```

```
procedure ImpSomme(Compte1,Compte2)
begin
Start;
temp1 := Read(Comptes(Compte1));
print(temp1);
temp2 := Read(Comptes(Compte2));
print(temp2);
temp1 := temp1 + temp2;
print(temp1);
end;
```

a) Supposons le client 1 dépose 1000 dans le compte 13 et que le client 2 y dépose 1500. Au départ le compte 13 a un solde de 10000. Chaque client invoque le programme *Dépôt* en créant une transaction, respectivement *T1* et *T2*. L'exécution simultanée de *T1* et *T2* se passe comme suit (on n'indique que les opérations communiquant avec la base de données).

Exécution 1:

```
Read1(comptes(13));
Read2(comptes(13));
Write2(comptes(13), 11500);
Commit2;
Write1(comptes(13), 11000);
Commit1;
```

??Quel est le résultat de cette exécution ?

??Expliquez pourquoi cette exécution est ou n'est pas recouvrable, sans cascade, stricte.

b) Supposons maintenant que les comptes 7 et 8 ont des soldes de 20000 et que le client 1 imprime les soldes des deux comptes en utilisant la procédure *ImpSomme* (transaction *T1*). En même temps, le client 2 transfère 1000 du compte 7 vers le compte 8 (transaction *T2*) en utilisant la procédure *Transfert* suivante :

```
procedure Transfert(Compte1, Compte2, Montant)
begin
Start;
temp := Read(Comptes(Compte1));
temp := temp - Montant;
Write(Comptes(Compte1),temp);
temp := Read(Comptes(Compte2));
temp := temp + Montant;
Write(Comptes(Compte2),temp);
Commit;
end;
```

L'ordonnancement des opérations est donné par l'exécution suivante.

Exécution 2:

```
Read2(Comptes(7)); Write2(Comptes(7), 19000);
Read1(Comptes(7)); Read1(Comptes(8));
Read2(Comptes(8)); Write2(Comptes(8), 21000);
Commit2;
Commit1.
```

??Quel est le résultat de cette exécution

??Expliquer pourquoi l'exécution est ou n'est pas recouvrable, sans cascade, ou stricte ?

Mêmes questions que pour l'exécution suivante :

Exécution 3:

*Read2(Comptes(7)); Write2(Comptes(7), 19000);
 Read1(Comptes(7));
 Read2(Comptes(8)); Write2(Comptes(8), 21000); Commit2;
 Read1(Comptes(8)); Commit1.*

Exercice 2 :

Soit les transactions *T1* et *T2* suivantes :

*T1 = Read(x); Read(y); Write(x); Write(y).
 T2 = Read(y); Write(x).*

- 1) Donner des exemples d'exécutions non séquentielles de *T1* et *T2* qui soient sérialisables et
 - ?recouvrable mais non sans cascade
 - ?sans cascade mais non stricte
 - ?stricte
- 2) Donner des exécutions non sérialisables et satisfaisant a, b, c.

Exercice 3 :

Soit *T1* et *T2* deux transactions et *x, y* deux granules de la base.
 L'ordonnancement suivant est-il sérialisable ? Est-il accepté par un verrouillage en deux phases ? Est-il accepté par un estampillage ?

Opération	T1	T2
(1)	Read(x)	
(2)	Write(x)	
(3)		Read(x)
(4)	Read(y)	
(5)	Write(y)	
(6)		Write(y)

Exercice 4 :

On considère les deux transactions suivantes:

*T1 : Lk(x), Lk(y), UNLk(x), Lk(z), Lk(t), UNLk(y), UNLk(z), UNLk(t), Lk(u), UNLk(u)
 T2 : Lk(x), Lk(y), UNLk(y), Lk(t), Lk(u), Lk(z), UNLk(u), Lk(t), UNLk(x), UNLk(z)*

NB: On suppose bien sûr que *Ti* accède à un granule entre le moment où elle le verrouille et le moment où elle rend le verrou.

1. Trouver un ordonnancement non-sérialisable de *T1* et *T2*.
2. Trouvez deux ordonnancements non séquentiels et sérialisables de *T1* et *T2*. Dans chaque cas, précisez l'ordonnancement séquentiel équivalent.
3. Une exécution simultanée de *T1* et *T2* peut-elle conduire à un interblocage ? Justifiez votre réponse.

Exercice 5 :

Montrer que toute exécution de transactions respectant toutes le protocole V2P est sérialisable.

Exercice 6 :

Soit $T1$, $T2$ et $T3$ trois transactions et x , y , z trois granules de la base.

L'ordonnancement suivant est-il sérialisable ? Si oui, indiquez l'ordonnancement séquentiel équivalent.

Opération	T1	T2	T3
(1)	Lk_S(x)		
(2)		Lk_X(y)	
(3)		Lk_S(z)	
(4)	UNLk(x)		
(5)			Lk_X(x)
(6)			Lk_S(z)
(7)		UNLk(y)	
(8)	Lk_X(y)		
(9)			UNLk(x)
(10)		UNLk(z)	
(11)	UNLk(y)		
(12)			Lk_X(z)
(13)			Lk_S(y)
(14)		Lk_S(y)	
(15)			UNLk(y)
(16)		UNLk(y)	
(17)			UNLk(z)

Exercice 7 :

Soit $T1$, $T2$, $T3$ et $T4$ quatre transactions et x , y deux granules de la base.

L'ordonnancement suivant est-il sérialisable ? Si oui, indiquez l'ordonnancement séquentiel équivalent.

Opération	T1	T2	T3	T4
(1)		Lk_S(x)		
(2)			Lk_S(x)	
(3)		Lk_X(y)		
(4)		UNLk(x)		
(5)			Lk_S(x)	
(6)		UNLk(y)		
(7)	Lk_S(y)			
(8)			UNLk(x)	
(9)				Lk_S(y)
(10)	Lk_S(x)			
(11)				UNLk(y)
(12)	Lk_X(z)			
(13)	UNLk(x)			
(14)				Lk_S(x)
(15)				UNLk(x)
(16)	UNLk(y)			
(17)	UNLk(z)			

Exercice 8 :

Soit T1, T2, T3, T4 et T5 cinq transactions et x, y, z, t quatre granules de la base.

L'ordonnancement suivant est-il sérialisable ? Trouvez un autre ordonnancement des mêmes transactions qui ne soit pas sérialisable.

Opération	T1	T2	T3	T4	T5
(1)				Lk(z)	
(2)					Lk(x)
(3)				UNLk(z)	
(4)					Lk(y)
(5)			Lk(t)		
(6)					UNLk(x)
(7)					UNLk(y)
(8)				Lk(x)	
(9)			UNLk(t)		
(10)			Lk(z)		
(11)				UNLk(x)	
(12)			UNLk(z)		
(13)	Lk(z)				
(14)	Lk(x)				
(15)		Lk(y)			
(16)	UNLk(z)				
(17)		UNLk(y)			
(18)		Lk(t)			
(19)	UNLk(x)				
(20)		Lk(x)			
(21)		UNLk(t)			
(22)		UNLk(x)			

Exercice 9 :

Soit T1, T2, T3 et T4 quatre transactions et x, y, z trois granules de la base.

Opération	T1	T2	T3	T4
(1)		L(x)		
(2)			L(x)	
(3)		E(y)		
(4)			E(x)	
(5)		L(z)		
(6)	L(y)			
(7)				L(y)
(8)	L(x)			
(9)	E(z)			
(10)				E(x)

1. Cet ordonnancement est-il sérialisable ? Si oui, donner l'ordre séquentiel des transactions équivalent.
2. Cet ordonnancement est-il accepté par un verrouillage en deux phases ?
3. Cet ordonnancement est-il accepté par un estampillage ? Motivez et illustrez votre réponse.

On suppose qu'avant le début des transactions les estampilles de lecture et d'écriture de x, y et z sont toutes égales à 10 et que les transactions T1, T2, T3, T4 se sont vu affecter par le SGBD respectivement les estampilles 15, 13, 12 et 14. Représenter par un tableau l'évolution du contenu de la base de données au fur et à mesure de l'exécution. Ce tableau aura 10 colonnes. Dans la première figurera le numéro de l'opération, puis trois colonnes pour chaque item de la base, contenant respectivement la valeur de l'item, son estampille de lecture, son estampille d'écriture. Les valeurs correspondent à la situation **après** l'opération. Ainsi, si on note (0) l'initialisation, la première ligne du tableau sera:

(0) x0 10 10 y0 10 10 z0 10 10

où x0, y0, z0 sont les valeurs de x, y et z à la suite de l'initialisation.

Un commentaire expliquera le passage de chaque ligne du tableau à la suivante. En cas de conflit entre deux transactions, la plus jeune, c'est-à-dire celle dont l'estampille est la plus grande, sera défaite, et ne sera plus considérée dans la suite de l'exercice.