

TD 6. Les Transactions

Exercices 1:

Préciser le problème pour chaque ordonnancement ci-dessous.

O1

Temps	T_1	T_2	solde_x
t_1		début_transaction	100
t_2	début_transaction	lire(solde_x)	100
t_3	lire(solde_x)	$\text{solde}_x = \text{solde}_x + 100$	100
t_4	$\text{solde}_x = \text{solde}_x - 10$	écrire(solde_x)	200
t_5	écrire(solde_x)	validation	90
t_6	validation		90

O2

Temps	T_3	T_4	solde_x
t_1		début_transaction	100
t_2		lire(solde_x)	100
t_3		$\text{solde}_x = \text{solde}_x + 100$	100
t_4	début_transaction	écrire(solde_x)	200
t_5	lire(solde_x)	:	200
t_6	$\text{solde}_x = \text{solde}_x - 10$	annulation	100
t_7	écrire(solde_x)		190
t_8	validation		190

O3

Temps	T_5	T_6	solde_x	solde_y	solde_z	somme
t_1		début_transaction	100	50	25	
t_2	début_transaction	somme = 0	100	50	25	0
t_3	lire(solde_x)	lire(solde_x)	100	50	25	0
t_4	$\text{solde}_x = \text{solde}_x - 10$	somme = somme + solde_x	100	50	25	100
t_5	écrire(solde_x)	lire(solde_y)	90	50	25	100
t_6	lire(solde_z)	somme = somme + solde_y	90	50	25	150
t_7	$\text{solde}_z = \text{solde}_z + 10$		90	50	25	150
t_8	écrire(solde_z)		90	50	35	150
t_9	validation	lire(solde_z)	90	50	35	150
t_{10}		somme = somme + solde_z	90	50	35	185
t_{11}		validation	90	50	35	185

Exercice 2 :

Quelles sont parmi les exécutions suivantes celles qui produisent des anomalies ? ai signifie que la transaction i est défaite, ci signifie que la transaction i est validée

E1 : r1(x), w1(x), r2(x), w2(y), a1, c2

E2 : r1(x), w1(x), r2(y), w2(y), a1, c2

E3: r1(x), r2(x), w2(x), w1(x), c1, c2

Exercice 3 :

Construisez les graphes de sérialisation pour les trois exécutions suivantes. Indiquez les exécutions sérialisables.

E1 : w2[x] w3[z] w2[y] c2 r1[x] w1[z] c1 r3[y] c3

E2 : r1[x] w2[y] r3[y] w3[z] c3 w1[z] c1 w2[x] c2

E3 : w3[z] w1[z] w2[y] w2[x] c2 r3[y] c3 r1[x] c1

Exercice 4:

Quelle est l'exécution obtenue par verrouillage à deux phases à partir de H?

H : r1[z] r2[z] w1[x] r3[x] w3[x] c2 w1[y] r3[y] c1 w3[y] c3

Exercice 5 :

L'ordonnancement ci-dessous respecte-t-il le protocole de verrouillage en 2 phases? Justifier.

Temps	Transaction T ₁	Transaction T ₂	État BD
t ₁	Début Transaction		A=60; B=40
t ₂	Ver(A,P)		
t ₃	Lire(A,a)		
t ₄	Dev(A)		
t ₅		DébutTransaction	
t ₆		Ver(A,P)	
t ₇		Lire(A,a)	
t ₈		Dev(A)	
t ₉	Ver(B,X)		
t ₁₀	b:=a		
t ₁₁	Écrire(b,B)		B=60
t ₁₂	Dev(B)		
t ₁₃		Ver(B,X)	
t ₁₄		b:=a+5	
t ₁₅		Écrire(B,b)	B=65
t ₁₆		Dev(B)	
t ₁₇	Ver(A,X)		
t ₁₈	a:= a / 2		
t ₁₉	Écrire(a,A)		A=30
t ₂₀	Dev(A)		
t ₂₁	ConfirmerTransaction		
t ₂₂		ConfirmerTransaction	A=30; B=65

Exercice 6 :

Les trois programmes suivants peuvent s'exécuter dans un système de gestion bancaire. Débit diminue le solde d'un compte c avec un montant donné m . Pour simplifier, tout débit est permis (on accepte des découverts). Crédit augmente le solde d'un compte c avec un montant donné m . Transfert transfère un montant m à partir d'un compte source s vers un compte destination d . L'exécution de chaque programme démarre par un Start et se termine par un Commit (non montrés ci-dessous).

Débit (c:Compte; m:Montant)	Crédit (c:Compte; m:Montant)	Transfert(s,d:Compte; m:Montant)
Begin	Begin	Begin
$t := \text{Read}(c);$	$t = \text{Read}(c);$	Débit(s,m);
Write(c,t-m);	Write(c,t+m);	Crédit(d,m);
End	End	End

Le système exécute en même temps les trois opérations suivantes : (1) un transfert de montant 100 du compte A vers le compte B (2) un crédit de 200 pour le compte A (3) un débit de 50 pour le compte B.

3.1. Écrire les transactions T1, T2 et T3 qui correspondent à ces opérations.

3.2. Montrer que l'histoire H: $r1[A] \ r3[B] \ w1[A] \ r2[A] \ w3[B] \ r1[B] \ c3 \ w2[A] \ c2 \ w1[B] \ c1$ est une exécution concurrente de T1, T2 et T3.

3.3. Mettre en évidence les conflits dans H et construire le graphe de sérialisation de cette histoire. H est-elle sérialisable?

3.4. Quelle est l'exécution H' obtenue à partir de H par verrouillage à deux phases?

Exercice 7 :

Une agence environnementale utilise une base de données relationnelle pour gérer des informations sur les véhicules.

Le schéma est le suivant (les clés sont soulignées) :

propriétaire (idprop, nom, âge)

véhicule (immatriculation, kilométrage, #idprop, #idmodèle)

modèle (idmodèle, crashtest, consommation)

carburant (désignation, cours, pollution)

véhicule_Carb(#idmodèle, #désignation)

L'exécution suivante est reçue par le système de l'agence environnementale :

H : $r1[x]r2[y]r3[x]w2[y]w1[x]r3[y]r1[z] \ r2[z]w2[z]w1[y]c1c2w3[z]c3$

1. Parmi les programmes qui s'exécutent dans le système, il y a $\text{KilométrAge}(i, k, a)$, qui fixe pour le véhicule d'immatriculation i le kilométrage à k et l'âge du propriétaire à a . Montrez quelle transaction de H pourrait provenir de KilométrAge .

2. Identifiez tous les conflits dans H et vérifiez si l'exécution est sérialisable en construisant le graphe de sérialisation.

3. Quelle est l'exécution obtenue par verrouillage à deux phases à partir de H?

Exercice 8 : (source : examen SGBD 2018)

Soit un système de vente en ligne qui gère plusieurs opérations commerciales, entre autre la procédure **Promotion_Art(A,P)** qui met à jour le prix de vente de l'article A qui est en promotion (P pourcentage de la remise).

```
CREATE OR REPLACE PROCEDURE Promotion_Art (ref article.refart%TYPE, pourct
remise.pourcentage%TYPE ) IS
Vref article.refart%TYPE ;
BEGIN

    SELECT refart INTO vref FROM Article WHERE refart= ref ;

    UPDATE Article SET prixv = ((1-pourct/100)* prixv) WHERE refart=ref ;
    COMMIT ;

    EXCEPTION
    WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.PUT_LINE('Cet article n"existe
pas !') ;
END ;
```

Cette procédure touche la table de la base de données du système de vente :

Article(refArt, libellé, prixA, prixV, #idremise)

1. Avant la mise à jour de la table Article on doit vérifier que le nouveau prix de vente de l'article en promotion n'est pas inférieur à son prix d'achat. Créer un trigger **trig_verif_prixv** pour vérifier cette contrainte.
2. Soit H une exécution reçue par le serveur de données du système de vente en ligne entre dateD et dateF de la période promotionnelle.

H : r1[x] r2[y] r2[t] w1[x] r2[z] r3[z] r3[x] w2[z] r1[z] c2 w1[x] c1 w3[x] c3

- 2.1. Identifier les transactions concurrentes de H.
- 2.2. Déterminer en justifiant votre réponse si la procédure **Promotion_Art** puisse provenir de l'une des transactions de H.
- 2.3. Vérifier si H est sérialisable en identifiant les conflits et en construisant le graphe de sérialisation.
- 2.4. Quelle est l'exécution obtenue par verrouillage à deux phases à partir de H?