

SDBD TD

Charles Fallourd

Transaction

Cohérence

$R(X)$; lecture de X
 $W(X)$; écriture de X
 a ; Abort
 C ; Commit
 $R_i(X) \prec T_i \cdot R(X)$

1. $R_1(X); W_1(X); R_2(X); W_2(X); a_1; C_2$
2. $R_1(X); W_1(X); R_2(Y); W_2(Y); a_1; C_2$
3. $R_1(X); R_2(X); R_2(Y); W_2(Y); R_1(Z); a; C_2$
4. $R_1(X); R_2(X); W_2(X); W_1(X); C_1; C_2$
5. $R_1(X); W_1(X); R_2(X); W_2(X); C_1; C_2$

Quels sont les ordonnancements qui produisent des anomalies ?

1. T_2 lit une valeur de X modifiée par T_1 qui ne sera pas validée. \Rightarrow anomalie
2. Pas d'anomalie.
3. Pas d'anomalie.
4. Problème : W_1 modifie une valeur qui a déjà été lue pour mise à jour.
5. Ok car équivalent à T_1 ; T_2

Cohérence ordonnancement

T_1	T_2
$R(\text{soldex});$ $\text{soldex} = \text{soldex} + 100;$ $w(\text{soldex});$	
	$R(\text{soldex});$ $\text{soldex} = \text{soldex} * 1.1;$ $W(\text{soldex});$ $R(\text{soldey});$ $W(\text{soldey});$
$R(\text{soldey});$ $\text{soldey} = \text{soldey} - 100;$ $W(\text{soldey});$ $C;$	$C;$

L'ordonnancement est-il cohérent ? (sérialisable)

$\text{soldex} = 100 \Rightarrow 200 \Rightarrow 220$

$\text{soldey} = 100 \Rightarrow 220 \Rightarrow 120$

$T_1 ; T_2 :$

$\text{soldex} = 100 \Rightarrow 200 \Rightarrow 220$

$\text{soldey} = 200 \Rightarrow 100 \Rightarrow 110$

$T_2 ; T_1 :$

$\text{soldex} = 100 \Rightarrow 110 \Rightarrow 210$

$\text{soldey} = 200 \Rightarrow 220 \Rightarrow 120$

\Rightarrow Non cohérent ! $T_1 ; T_2 \neq T_2 ; T_1$

$$T_1 = R(X); R(Y); W(X); W(Y)$$

$$T_2 = R(Y); W(X)$$

a/ Donner un ordonnancement de T_1 et T_2 sérialisable

b/ Donner un ordonnancement de T_1 et T_2 non sérialisable

Solution :

a/ $R_1(X); R_2(Y); W_1(X); W_2(X); W_1(Y)$

b/ $R_1(X); R_2(Y); W_2(X); W_1(X); W_1(Y)$

Lock

LS(X) : Lock share de X
LX(X) : Lock exclusive de X
UL(X) : unlock de X

T_1 = LX(X) ; R(X) ; $X = X + 100$; W(X); LX(Y); R(Y); $Y = Y - 100$; W(Y); UL(X); UL(Y);
 T_2 = LX(Y); R(Y); $Y = Y + 4$; W(Y); LX(Z); $Z = Z - 500$; W(Z); UL(Y); UL(Z);
 T_3 = LX(Z); $Z = 0$; W(Z); LX(X); $X = X - 500$; W(X); UL(Z); UL(X)

- a) transaction bien formée ? oui pour toutes les transactions
- b) transaction à 2 Φ ?

Bien formée : LS au moins pour lecture, LX pour écriture et libération des verrous
2 Φ (2 phases) : Pose des verrous puis libération. Pas pose + libération + pose
+ libération. (Phase d'extension puis libération).

Que peut-il se passer lors de l'exécution non séquentielle de T_1 , T_2 et T_3 ?
 \Rightarrow Risque de deadlock.

Procédure

```
1 | Procedure credit(solde: x, montant: y)
   |   debut
3 |       LX(X);
   |       R(X);
5 |       X = X+Y;
   |       W(X);
7 |       UL(X);
   |   fin
```

```
Procedure debit(solde:x, montant y)
2 |   debut
   |       LX(x);
4 |       R(x);
   |       x=x-y;
6 |       w(x);
   |       UL(x);
8 |   fin
```

Ecrire la procédure Transfert.

```
Procedure Transfert(compte: x,z, montant y)
2 |   debut
   |       LX(x);
4 |       LX(z);
   |       R(x);
6 |       x=x-y;
   |       W(x);
8 |       R(Z);
   |       z = z+y;
10 |      W(z);
   |      UL(x);
12 |      UL(z);
   |   fin
```

On est obligé de tout refaire ... il faut que la procédure soit à deux phases. Si on appelle les deux procédures précédentes, il y aura extension et libération deux fois.

?

Update, delete et insert \Rightarrow verrouille les tuples accédés en exclusifs.
Commit et Rollback déverrouille tous les verrous.

Q1 : Suffisant pour écrire des transactions bien formées et à 2 Φ ?

\Rightarrow Non il manque le select !