

TP « Gestion des transactions »

Légende : AS : Access Share, RE : Row Exclusive, RS : Row Share, S : Share
FS : For Share, FU : For Update

Expérience 1 État table t : (0,0), (2,3),(0,1)

Session 1	Affichage	Verrous posés	Session 2	Affichage	Verrous posés
begin ;	BEGIN		begin ;	BEGIN	
select * from t ;	0 0 2 3 0 1	AS sur t			
			update T set A=A+2 ;	UPDATE 3	RE sur t FU sur toutes les données
select * from t ;	0 0 2 3 0 1		select * from t ;	2 0 4 3 2 1	AS sur t
			commit ;	COMMIT	libère les verrous et propage la modification
select * from t ;	2 0 4 3 2 1				
commit ;	COMMIT	libère les verrous			

Quel phénomène apparaît dans la session 1 ?

Après le commit dans la session 2, les changements sont propagés, lecture successives inconsistants

Expérience 2 État table t : (2,0), (4,3),(2,1)

Session 1	Affichage	Verrous posés	Session 2	Affichage	Verrous posés
begin ;	BEGIN		begin ;	BEGIN	
set transaction isolation level Serializable ;	SET		set transaction isolation level Serializable ;	SET	
select * from t ;	2 0 4 3 2 1	AS sur t			
			update t set a=a-1 ;	UPDATE 3	RE sur t FU sur toutes les données
select * from t ;	2 0 4 3 2 1		select * from t ;	1 0 3 3 1 1	AS sur t
			commit ;	COMMIT	Libère les verrous Propage la modification

select * from t ;	2 0 4 3 2 1				
commit ;	COMMIT	Libère les verrous			
select * from t ;	1 0 3 3 1 1				

Est-ce que le phénomène persiste dans la session 1 ?

Le phénomène ne persiste plus dans la session 1, les modifications ne sont visibles qu'après la fin de la transaction

Expérience 3 État table t :

Session 1	Affichage	Verrous posés	Session 2	Affichage	Verrous posés
begin ;	COMMIT				
select * from t ;	1 0 3 3 1 1	AS sur t			
update t set a=a+1	UPDATE 3	RE sur t FU sur les données			
select * from t ;	2 0 4 3 2 1				
rollback ;	ROLLBACK	Libère les verrous Annule les modifications			
select * from t ;	1 0 3 3 1 1				

Quel est le but de rollback ?

Il annule les modifications qui ont été faites dans la transaction

Expérience 4a État table t :

Session 1	Affichage	Verrous posés	Session 2	Affichage	Verrous posés
begin ;	BEGIN		begin ;	BEGIN	
update t set a=a+1	UPDATE 3	RE sur t FU sur les données			
select * from t ;	2 0 4 3 2 1				
			insert into t values (3,7) ;	INSERT 0 1	RE sur t FU sur (3,7)

Feuille1

			select * from t ;	1 0 3 3 1 1 3 7	AS sur t
			delete from t where a=2 ;	DELETE 0	
			delete from t where a=1 ;	bloqué	conflit FU sur (1,0) et (1,1)
commit ;	COMMIT	libère les verrous et propage			
			commit ;	COMMIT	libère les verrous et propage les modifications

Ces deux transactions sont-elles sérialisables ?

Ces transactions ne sont pas sérialisables car elles cherchent à modifier les mêmes données (conflit entre les verrous des 2 sessions)

Expérience 4b État table t : (1,0),(3,3),(1,1) (Attention, modifier la base en conséquence)

Session 1	Affichage	Verrous posés	Session 2	Affichage	Verrous posés
begin ;	BEGIN		begin ;	BEGIN	
set transaction isolation level Serializable ;	SET		set transaction isolation level Serializable ;	SET	
update t set a=a+1	UPDATE 3	RE sur t FU sur les données			
select * from t ;	2 0 4 3 2 1	AS sur t			
			insert into t values (3,7) ;	INSERT 0 1	RE sur t FU sur (3,7)
			select * from t ;	1 0 3 3 1 1 3 7	
			delete from t where a=2 ;	DELETE 0	
			delete from t where a=1 ;	ERROR: could not serialize access due to concurrent update	
commit ;	COMMIT	Libère les verrous Propages les modifications			
			commit ;	ROLLBACK	Libère les verrous Annule les modifications

Expérience 5 État table t : (2,0),(4,3),(2,1),(3,7) (Attention, ne pas oublier d'ajouter la ligne (3,7))

Session 1	Affichage	Verrous posés	Session 2	Affichage	Verrous posés
begin ;	BEGIN		begin ;	BEGIN	
update t set a=3 where a=2 ;	UPDATE 2	RE sur t FU sur (2,0) et (2,1)			
			update t set b=2 where b=3 ;	UPDATE 1	RE sur t FU sur (4,3)

Feuille1

			select * from t ;	2 0 2 1 3 7 4 2	AS sur t
select * from t ;	4 3 3 7 3 0 3 1	AS sur t			
			update t set a=3 where a=2 ;	Bloqué	conflit FU sur (2,0) et (2,1)
update t set b=2 where b=3	ERROR: deadlock detected	Conflit FU sur (4,3)			
commit ;	ROLLBACK	Libère les verrous Annule les modifications	commit ;	COMMIT	Libère les verrous Propage les modifications

Comment se terminent les transactions ?

La première transaction lève une erreur deadlock, c'est à dire que les 2 transactions ont des conflits, donc la première fait un rollback et donc libère les verrous nécessaires à l'exécution de la deuxième

Expérience 6 État table t : (3,0),(4,2),(3,1),(3,7)

Session 1	Affichage	Verrous posés	Session 2	Affichage	Verrous posés
begin ;	BEGIN		begin ;	BEGIN	
select * from t where b=7 for update ;	3 7	FU sur (3,7)			
			update t set b=10 where b=0 ;	UPDATE 1	RE sur t FU sur (3,0)
			select * from t ;	3 7 4 2 3 1 3 10	AS sur t
select * from t ;	3 7 4 2 3 0 3 1	AS sur t	update t set a=5 where a=3 ;	Bloqué puis UPDATE 3	
update t set b=6 where b=7 ;	UPDATE 1	RE sur t FU sur (3,7)			
commit ;	COMMIT	Libère les verrous Propages les modifications			
			commit ;	COMMIT	Libère les verrous Propage les modifications
			select * from t ;	4 2 5 6 5 1 5 10	

Quel type de verrouillage est effectué lors du select de la session 1 ?

Le select de la session 1 verrouille la ligne sélectionné avec le for update

Expérience 7 État table t : (5,10),(4,2),(5,1),(5,6)

Session 1	Affichage	Verrous posés	Session 2	Affichage	Verrous posés
begin ;	BEGIN		begin ;	BEGIN	
lock table t in share mode ;	LOCK TABLE				
			select * from t ;	4 2 5 6 5 1 5 10	AS sur t
			insert into t values (5,1) ;	Bloqué INSERT 0 1	Conflit sur a table t
commit ;	COMMIT	Libère les verrous			
			commit ;	COMMIT	Libère les verrous Propage la modification
begin ;	BEGIN		begin ;	BEGIN	
lock table t in row share mode ;					
update t set b=b+1 ;	UPDATE 5	RE sur t FU sur les données	insert into t values (5,2) ;	INSERT 0 1	FU sur (5,2)
			update t set b=b+1	Bloqué UPDATE 6	Conflit FU sur les données FU sur les données
commit	COMMIT	Libère les verrous Propages les modifications			
			commit ;	COMMIT	Libère les verrous Propage les modifications

Comment expliquez-vous ces résultats ?

Le share mode bloque toutes modifications de la table dans les autres transactions, tandis que le row share mode ne bloque que les lignes modifiés par la transaction

Expérience 8 État table t : (5,12),(4,4),(5,3),(5,8),(5,3),(5,3)

Session 1	Affichage	Verrous posés	Session 2	Affichage	Verrous posés
begin ;	BEGIN		begin ;	BEGIN	
lock table t in exclusive mode ;	LOCK TABLE				
Select * from t ;	4 4 5 8 5 3 5 12 5 3 5 3	AS sur t	select * from t ;	4 4 5 8 5 3 5 12 5 3 5 3	AS sur t
			select * from t for update ;	Bloqué	Conflit FU sur les données
commit ;	COMMIT	Libère les verrous			
			commit ;	COMMIT	Libère les verrous

Comment interprétez vous ces réponses ?

Le exclusive mode permet d'autoriser la lecture mais pas la modification des données sur la table

