Thomas Lerchundi
7 octobre 2015

Base de données

Rapport du TP1

Transactions

UJF Imag

Exercice 1 : Annulation des transactions

On insère dans la table deux comptes pour Paul :

```
insert into Comptes values (1, 'Paul', 1000);
insert into Comptes values (2, 'Paul', 3);
```

Après affichage, la table contient bien les 2 nouveaux comptes.

Après avoir fait un rollback, la table ne contient plus rien, la transaction est bien annulée.

Exercice 2 : Validation de transaction

Cette fois-ci, on insère dans la table deux comptes pour Pierre, puis on commit en suivant:

```
insert into Comptes values (1, 'Pierre', 1500);
insert into Comptes values (2, 'Pierre', 30);
commit;
```

Comme la transaction est validée par le commit, le rollback n'a aucun effet sur la table.

Exercice 4 : Points de sauvegarde

On insère un compte pour Paul, puis on place un point de sauvegarde :

```
insert into Comptes values (3,'Paul',300);
savepoint UnInsert;
```

On insère un autre compte, puis on rollback jusqu'au point de sauvegarde.

```
insert into Comptes values (4, 'Paul', 4333);
rollback to UnInsert;
```

Le compte numero 4 est bien enlevé. Et le compte 3 persiste. Après un rollback, la transaction est totalement annulée et le Compte 3 n'est plus dans la table.

Partie 2 : Cohérence

Exercice 1 :

```
insert into Comptes values (3, 'Claude', 100);
insert into Comptes values (4, 'Henri', 200);
set constraint SoldePositif IMMEDIATE;
Update Comptes set Comptes.Solde=Solde+50 where Comptes.Nom='Henri';
Update Comptes set Comptes.Solde=Solde-150 where Comptes.Nom='Claude';
->check constraint (LERCHUNT.SOLDEPOSITIF) violated
```

On constate que comme le mode est Immediate, la contrainte est de suite vérifiée au moment de la mise à jour, avant même la validation de la transaction.

En mode Deferred, la vérification de la contrainte ne se fait qu'au moment de la validation de la transaction.

Partie 3 : Isolation

Exercice 1 : niveau d'isolation READ COMMITED

On ajoute 1000 sur le compte numéro 2 de Pierre.

```
Update Comptes set Comptes.Solde=Solde+1000 where Comptes.NC='2';
```

En affichant sur S1, le compte 2 contient 1030 euros, alors que sur S2, il n'en contient que 30.

Tant que S1, n'a pas validé la transaction, les lectures de S2 se font depuis la dernière valeur validée.

Exercice 2

Dans S1 :

```
insert into Comptes values (3,'Paul',200);
```

Dans S2 :

```
set transaction isolation level serializable;
```

On constate que les autres transactions n'ont pas d'effet sur une transaction en mode Serializable, si cette même transaction n'a pas commit.

Exercice 3: Verrouillage

Dans S1 :Delete From Comptes Where Nom in ('Pierre');

Dans S2 : Update Comptes set Comptes.Nom='Pierre';

On constate un blocage!

On commit alors S1, ce qui débloque S2;

Dans S2 :

SQL> Update Comptes set Comptes.Nom='Pierre';

Le contenu de la table est donc:

NC NOM SOLDE

---------- ---------- ----------

1 Pierre 1500

2 Pierre 1030

3 Pierre 200

commit de S2

Dans S1:

SQL> select * from Comptes;

NC NOM SOLDE

---------- ---------- ----------

1 Pierre 1500

2 Pierre 1030

3 Pierre 200

C'est donc la dernière transaction à valider qui affecte l'ensemble de la table. Comme S2 a été lancée avant que la transaction de S1 ait eu le temps de commit, elle n'a pas évalué les données qui ont été supprimées par S1.

Exercice 4 : Interblocage

Création du Deadlock:

Etat initial de la table:

NC	NOM	SOLDE
1	Pierre	1500
2	Pierre	1030
3	Pierre	200

Voici donc comment je crée l'interblocage :

T1: SQL> update Comptes set Nom = 'thomas' where solde =1500;

T2: SQL> update Comptes set Nom = 'lucas' where solde =200;

T1: SQL> Update Comptes set Nom='Renaud' where solde=200;

(T1 attend que T2 ait fini de travailler sur le compte 3)

T2: SQL> update Comptes set Nom='Jaki' where solde=1500;

(T2 attend que T1 ait fini de travailler sur le compte 1)

Voici l'affichage de oracle qui arrive dans le terminal de T1:

ERROR at line 1:

ORA-00060: deadlock detected while waiting for resource

Oracle détecte donc l'interblocage.

T1 se débloque mais si j'affiche l'état de la table dans T1 : je vois qu'il a écrit thomas mais pas Renaud dans la table.

T2 est toujours en attente d'un commit de T1.