

TP N°4

Transactions

Créer la table suivante:

RESERVATION (NUMERO SEANCE, NUMERO PLACE, NOM SPECTATEUR)

Exercice 1. Effectuez successivement les opérations suivantes.

1. Affichez toutes les réservations.
2. Insérez une réservation:
insert into RESERVATION values (1 , 1 , 'Alice');
3. Affichez toutes les réservations.
4. Tapez l'instruction commit;
5. Affichez toutes les réservations.
6. Insérez une nouvelle réservation.
7. Affichez toutes les réservations.
8. Tapez l'instruction rollback;.
9. Affichez toutes les réservations.
10. Concluez!
L'insertion est effectuée dans la base de données mais peut être défaire définitivement par un rollback.

Exercice 2. Lancez une seconde session sqlplus et procédez comme suit.

1. Insérez une réservation dans cette seconde session.
La réservation est créée dans la seconde session.
2. Affichez cette même réservation pour chacune des deux sessions. Commentez!
La réservation n'apparaît que dans la seconde session (et pas dans la première); propriété d'isolation des transactions.
3. Insérez cette même réservation dans la première session. Que se passe-t-il dans cette première session?
La première session est bloquée sur cette insertion ce qui suppose un mécanisme de verrouillage des enregistrements (par exemple : l'utilisation de mutex).

4. Effectuez un rollback dans la seconde session.
La seconde session défait l'insertion et termine sa transaction, et ainsi débloquent la première session qui peut alors effectuer son insertion.
5. Affichez cette même réservation pour chacune des deux sessions.
La réservation n'apparaît que dans la première session (et non dans la seconde).
6. Effectuez un commit dans la première session.
La seconde session valide définitivement la réservation.
7. Affichez cette même réservation pour chacune des deux sessions.
La réservation apparaît dans les deux sessions.
8. Concluez!
Les sessions ne voient que leurs propres mises à jour tant que les autres transactions ne sont pas achevées (validées ou invalidées). De plus, en cas d'accès concurrents, le dernier arrivé est mis en attente.

Effectuez successivement les opérations suivantes.

1. Modifiez une réservation R1 dans la première session.

```
1 update RESERVATION
   set NOMSPECTATEUR = 'Alice'
3 where NUMEROSEANCE = 1
   and NUMEROPLACE = 1;
```

La réservation R1 est modifiée dans la première session.

2. Modifiez une réservation R2 dans la seconde session.
La réservation R2 est modifiée dans la seconde session.
3. Modifiez la réservation R1 dans la seconde session.
La seconde session est bloquée sur la modification de la réservation R1.
4. Modifiez la réservation R2 dans la première session.
La demande de la modification de la réservation R2 par la première session est annulée par Oracle suite à la détection d'un interblocage d'où le message d'erreur ORA-00060: deadlock detected while waiting for resource. On est donc ramené à la même situation que dans l'exercice précédent.
5. Effectuez un rollback dans la première session.
La première session défait son travail (uniquement sur la réservation R1), termine sa transaction et ainsi débloquent la seconde session qui modifie alors la réservation R1.
6. Effectuez un rollback dans la seconde session.
La seconde session défait son travail (sur les réservations R2 et R1) et termine sa transaction.

7. Concluez!

Oracle a un mécanisme de détection (et de résolution) d'interblocage.