**Les accès concurrents sous Oracle**

1. Ouvrez deux sessions sous le même nom. Créez et peuplez la table Produit

```
CREATE TABLE PRODUIT (
numprod number(6) constraint produit_numprod_pk primary key,
libelléProd varchar2(20) not null,
qté number(5));
--
Insert into PRODUIT  values (1,'chaise',2500);
Insert into PRODUIT  values (2,'table',1000);
Insert into PRODUIT  values (3,'bureau',860);
commit;
```

2. Faites des modifications (de type LMD) dans une des sessions et voyez si les modifications sont connues de l'autre session.

   SESSION 1 :

   ```
   select * from produit;
   update produit set qté=7000 where numprod=1;
   ```

   SESSION 2 :

   ```
   select * from produit;
   ```

3. Faites un COMMIT et vérifiez si les modifications sont connues de l'autre session.

   SESSION 1 : commit;

   SESSION 2 : select * from produit;

4. Modifiez la quantité **d'un** produit dans les deux sessions (avec des valeurs différentes). Que se passe-t-il ?

   SESSION 1 :

   ```
   select * from produit;
   update produit set qté=5000 where numprod=1;
   ```

   SESSION 2 :

   ```
   select * from produit;
   update produit set qté=9000 where numprod=1;
   ```

   ➔ blocage (car verrou sur la ligne)

5. Faites un COMMIT dans la session qui a fait les modifications en premier (SESSION 1). Que se passe-t-il ? Faites un select dans les 2 sessions pour voir la modification.

   SESSION 1: commit;

   SESSION 2 :

   ```
   select * from produit;
   ```

   SESSION 1 :

   ```
   select * from produit;
   ```

6. Faites un COMMIT dans la deuxième session. Faites un select dans les 2 sessions pour voir la modification.

   SESSION 2 :

   ```
   commit;
   select * from produit;
   ```

   SESSION 1 :

   ```
   select * from produit;
   ```

7. Utilisez un SELECT **FOR UPDATE** sur une des sessions et essayez de modifier les lignes bloquées avec l'autre session.
   SESSION 1 :
   select numprod, libelléprod from produit where numprod <3 for update of qté;
   SESSION 2 :
   update produit set qté=9000 where numprod=3;
   update produit set qté=9000 where numprod=1;
   commit;
   SESSION 1 :
   select * from produit;
   update produit set qté=2000 where numprod=1;
   commit;
   select * from produit;
   SESSION 2 :
   select * from produit;

8. **Situation de DEADLOCK (verrous mortels)** : 2 (ou plusieurs) sessions se bloquent mutuellement. Chacune des sessions attend des données verrouillées par l'autre session. Oracle détecte automatiquement les situations de deadlock et les résout en invalidant (ROLLBACK) l'ordre SQL associé à la transaction qui détecte le deadlock.
   Exemple :

   SESSION 1 :
   update produit set libelléProd='table basse' where numprod=2;
   SESSION 2 :
   update produit set qté=2000 where numprod=1;
   SESSION 1 :
   update produit set libelléProd='chaise pliante' where numprod=1;
   SESSION 2 :
   update produit set qté=5000 where numprod=2;

   **DEADLOCK détecté au niveau de SESSION1 :**
   Erreur commençant à la ligne 1 de la commande :
   update produit set libelléProd='chaise pliante' where numprod=1
   Rapport d'erreur :
   Erreur SQL : ORA-00060: détection d'interblocage pendant l'attente d'une ressource
   00060. 00000 - "deadlock detected while waiting for resource"
   *Cause: Transactions deadlocked one another while waiting for resources.
   *Action: Look at the trace file to see the transactions and resources
   involved. Retry if necessary.
   ➔ **ROLLBACK de update produit set libelléProd='chaise pliante' where numprod=1;**

   SESSION 2 en attente.
   SESSION 1 :
   commit;
   SESSION 2 :
   commit;
   select * from produit;

Résultats :

| | NUMPROD | LIBELLÉPROD | QTÉ |
|---|---|---|---|
| 1 | 1 | chaise | 2000 |
| 2 | 2 | table basse | 5000 |
| 3 | 3 | bureau | 9000 |