

# TME - Transactions

## Objectifs

- Mettre en place des situations de concurrence entre transactions.
- Comprendre les mécanismes de contrôle de concurrence.

## Préparation

- Pour se connecter à Oracle, lisez et effectuez les instructions de la page web « ConnexionOracle » disponible sur le site web de l'UE dans la partie des TME
- Pour récupérer les fichiers du TME, il faut décompresser l'archive du TME dans votre répertoire principal. Pour cela, taper les commandes suivantes :
  - cd
  - tar xzvf \$BD\_TOOL/li345/trans-etv.tgz
  - cd trans-etv
- Pour exécuter les instructions SQL contenu dans un fichier monfichier.sql, il suffit de taper :
  - SQL > @monfichier

Une BD contient un stock de produits. Pour chaque produit, on a le n°, le nom et la quantité restant en stock.

- **Produit**(num, nom, reste)

Depuis sqlplus, exécuter le fichier nommé *produit.sql* pour créer la base

- SQL> @produit

Chaque transaction effectue des opérations parmi les suivantes.

- Lecture d'un tuple:
  - L(A) : select \* from Produit where num = 1;
  - L(B) : select \* from Produit where num = 2;
- Ecriture d'un tuple :
  - E(A) : update Produit set reste= &1 where num = 1;
  - E(B) : update Produit set reste=&1 where num = 2;
  - E(C) : update Produit set reste=&1 where num = 3;
- Lecture/écriture d'un tuple :
  - D(A) : update Produit set reste = reste\*2 where num = 1;L(A)
- Modification du schéma d'une relation
  - SR : alter table Produit modify (nom varchar2(15));

Ouvrir 2 fenêtres de terminal, juxtaposées l'une en haut à gauche de l'écran, l'autre en bas à gauche (certaines questions peuvent nécessiter une troisième fenêtre).

Dans **chaque** fenêtre :

- `cd trans-etu` (aller dans le répertoire de travail)
- Récupérer le numéro de terminal par la commande `tty`
- `sqlplus E123456/E123456@oracle` (se connecter au SGBD) où 123456 doit être remplacé par votre numéro étudiant

Prendre connaissance du contenu des fichiers SQL situés dans votre répertoire de travail.

## Exercice 1 : Verrouillage

Pour cet exercice, le niveau d'isolation sera *read committed*

- `SQL> alter session set isolation_level = read committed;` (voir fichier `isol1.sql`)
- le niveau d'isolation est fixé pour toutes les transactions d'une session (= un terminal)
- il est possible de changer le niveau d'isolation uniquement en **début** de transaction (au début de la session ou juste après un commit).

Réaliser les expériences montrant le fonctionnement du gestionnaire de verrous. Ci-après, chaque séquence d'opérations est représentée par un tableau. L'axe horizontal représente le temps. L'opération de la colonne  $n+1$  débute **après** l'opération de la colonne  $n$ . Exécuter **T1** dans le 1er terminal, et **T2** dans le 2ème.

### a) Une lecture peut-elle être bloquante ?

- Vérifier que L1(A) ne bloque pas L2(A)

T1	L(A)	
T2		L(A)

(ce tableau signifie : taper `@la` dans le terminal 1, puis `@la` dans le terminal 2)

- Vérifier que L1(A) ne bloque pas E2(A)

T1	L(A)	
T2		E(A)

### b) Une lecture peut-elle être bloquée ?

- L2(A) n'est pas bloquée par L1(A) (voir ci-dessus)
- Vérifier que L2(A) n'est pas bloquée par E1(A)

T1	E(A)		commit	
T2		L(A)		commit

- La transaction T2 voit-elle l'écriture faite précédemment par T1 ? Quel est l'ordre en série équivalent à cette exécution ? Est-ce le même que l'ordre chronologique des commit, pourquoi ?

### c) Une écriture peut-elle être bloquante ou bloquée ?

- E2(A) bloquée par E1(A): Vérifier que T2 est en attente d'un verrou exclusif sur A.

T1	E(A)	
T2		E(A): attente

- Que se passe-t-il : lorsque T1 valide ? lorsque T1 abandonne (rollback) ?

<b>T1</b>	E(A)		commit ou rollback
<b>T2</b>		E(A): attente	....

**d) Granularité du verrouillage**

- Vérifier qu'E2(B) n'est pas bloquée par E1(A).
- Une lecture est-t-elle bloquée par une écriture SR qui modifie le **schéma** d'une relation ?
- Une écriture E2(A) est-elle bloquée par SR1 ?

<b>T1</b>	SR	
<b>T2</b>		E(A) attente ?

- Une écriture SR2 est-elle bloquée par E1(A) ?

<b>T1</b>	E(A)	
<b>T2</b>		SR attente ?

- Quels sont les verrous demandés avant de traiter E(A) et SR ?

**e) Interblocage**

- Vérifier l'interblocage suivant:

<b>T1</b>	E(A)	E(B)	
<b>T2</b>	E(B)		E(A)

- Comment résoudre l'interblocage sans abandonner T1 ?

**Exercice 2 : Sérialisation**

Pour cet exercice, le niveau d'isolation des sessions est *serializable*

- SQL> alter session set isolation\_level = serializable; (ou @isol2)

**a)** La séquence suivante est-elle sérialisable ? Si oui, quel est l'ordre en série équivalent ?

<b>T1</b>	L(A)			<b>D(A)</b>
<b>T2</b>		E(A)	commit	

Que se passe-t-il ?

**b)** La séquence suivante est-elle sérialisable ? Si oui, quel est l'ordre en série équivalent ?

<b>T1</b>	L(A)			E(A)
<b>T2</b>		E(A)	commit	

Pourquoi cette séquence est-t-elle refusée ?

**c)** La séquence suivante est-elle sérialisable ? Si oui, quel est l'ordre en série équivalent ?

<b>T1</b>	L(B)			E(A)
<b>T2</b>		E(A)	commit	

Pourquoi cette séquence est-t-elle refusée ?

d) Recommencer en modifiant les niveaux d'isolation de T1 et de T2. (voir fichiers *isol1* et *isol2*) Est ce possible de garantir le niveau d'exécution *sérializable* pour T2, lorsque T1 est au niveau *read committed* ? Commenter les résultats obtenus.

### Exercice 3 : Interrogation du dictionnaire d'Oracle

**Question 1 :** On considère les vues du dictionnaire d'Oracle suivantes : `dba_waiters`, `v$locked_object`, `v$session`, `user_objects` et `v$transaction`. Utilisez la documentation sur le site web des TME de l'UE (DocumentationOracle ->le dictionnaire du SGBD) pour voir la description de leurs attributs. Ecrivez les requêtes SQL suivantes :

1. la liste des tables pour lesquelles une session en attend une autre, avec les numéros des sessions concernées, le mode de verrou demandé et celui accordé. Utiliser `dba_waiters`, `v$locked_object`, `v$session` et `user_objects`.
2. la liste des estampilles (SCN) des transactions actives (ayant déjà fait au moins une écriture) ainsi que le numéro de session correspondant, triées par nom d'utilisateur. Utiliser `v$transaction` et `v$session` (voir les attributs `addr` et `taddr`)
3. la liste des sessions et le terminal associé. Utiliser `v$session`.

**Question 2 :** Définir une situation d'interblocage qui implique 3 transactions (sans attente mutuelle directe entre 2 transactions). On souhaite abandonner la transaction la plus récente. Quelle requête du dictionnaire affiche la date de début des transactions actives ? Voir l'estampille *system change number* (SCN) d'une transaction.