

SGBD TP8: CORRECTION-LES TRANSACTIONS (1)

Accès concurrents

- Durée 1h15mn.
- Pour ce TP vous avez besoin de la BD GestAsso.

④ Pour simuler l'exécution de transactions concurrentes, ouvrez 2 fenêtres terminal (l'une sera identifiée par *TR1* et l'autre par *TR2*), connectez vous sur chacune de ces consoles à la base oracle avec sql*plus. ORACLE considère chaque session comme un utilisateur. Ainsi même si une personne est connectée deux fois à la base, ORACLE considère qu'il s'agit de 2 personnes différentes. Nous avons donc deux utilisateurs notés 1 et 2 réalisant des opérations de lecture, écriture, validation, annulation, en concurrence sur la base GestAsso.

Travail à faire :

④ Avant de commencer les tests exécuter les commandes ci dessous :

```
SQL> INSERT INTO typeact VALUES (107, 'Collecte solidaire', 'collecte de vêtements, de nourriture');
SQL> INSERT INTO Action VALUES ('Act23', 'vetements', '02/04/2018', '03/11/2018', 'collecte de vetement pour fille, age 10ans', 107);
SQL> COMMIT ;
```

1. Faites les modifications dans la session TR1 et vérifiez si les modifications sont connus de l'autre session TR2.

TR1 : *1 Row updated*

```
SQL> UPDATE Action SET
datef='03/09/2018'
WHERE noact='Act23' ;
```

- La modification faite par la TR1 sur datef est invisible pour la TR2

- blocage de la mise à jour lancée par la TR2 sur datef.

TR2 :

```
SQL> SELECT * FROM Action
WHERE noact='Act23' ;
```

```
SQL> UPDATE Action SET
datef='07/09/2018'
WHERE noact='Act23' ;
```

2. Faites un COMMIT dans la session TR1 et voyez si les modifications sont connues de l'autre session.

TR1 : *Commit complete pour la TR1*

```
SQL> COMMIT ;
```

*- 1 row update pour la TR2
- l'attribut datef a la valeur mis à jour par la TR2*

TR2 :

```
SQL> SELECT * FROM Action
WHERE noact='Act23' ;
```

TR1 : *- la TR1 ne peut pas voir la nouvelle valeur de datef mis à jour par TR2. Datef est toujours = 03/09/18*

```
SQL> SELECT * FROM Action
WHERE noact='Act23' ;
```

3. Faites un COMMIT dans la session TR2.

- commit complete

TR2 :

```
SQL> COMMIT ;
```

4. Faites un SELECT dans les deux sessions pour voir la modification.

TR1 : *- après le commit de la TR2, la TR1 peut voir la nouvelle valeur de datef*

```
SQL> SELECT * FROM Action
WHERE noact='Act23' ;
```

TR2 :

```
SQL> SELECT * FROM Action
WHERE noact='Act23' ;
```

5. Utilisez un SELECT FOR UPDATE sur la session TR1 et essayer de modifier les lignes bloquées avec la session TR2. Dites ce qui se passe.

TR1 :

```
SQL> SELECT * FROM Action
WHERE noact='Act23'
FOR UPDATE OF datef;
```

- blocage de la mise à jour lancée par la TR2 sur le tuple dont la clé primaire est 'act23' sur datef car la TR1 avec le for update l'a verrouillé en écriture.

TR2 :

```
SQL> UPDATE action SET
datef='09/10/2018'
WHERE noact='Act23' ;
```

TR1 : *- 1 row update*

```
SQL> UPDATE action SET
datef='01/10/2018'
WHERE noact='Act23' ;
```

-commit complete

- La valeur de l'attribut datef est 09/10/18

- Rollback complete

- La valeur de l'attribut datef reprend la valeur précédant : 01/10/18

TR2 :

```
SQL> SELECT * FROM Action
WHERE noact='Act23' ;
```

SQL> ROLLBACK ;

```
SQL> SELECT * FROM Action
WHERE noact='Act23' ;
```

6. En travaillant sur la table Famille, commentez le mode de fonctionnement par défaut d’Oracle :

```
Ø Avant de commencer les tests exécuter les commandes ci dessous :
SQL> ALTER TABLE famille ADD (montantaide NUMBER (6,2)) ;
SQL> UPDATE famille SET montantaide=100 ;
SQL> COMMIT ;
```

6.1.

```
TR1:                                - table(s) locked
SQL> LOCK TABLE famille IN
EXCLUSIVE MODE ;
```

```
- le select s'exécute sans problème
TR2:
SQL> SELECT * FROM famille
WHERE nofam=1 ;
```

6.2.

```
TR1:                                1 row updated
SQL> UPDATE famille
SET montantaide = montantaide + 10
WHERE nofam=1;
```

```
La mise à jour faite par la TR1 n'est
pas visible pour la TR2=> la valeur
de montantaide est toujours =100
TR2:
SQL> SELECT * FROM famille
WHERE nofam=1 ;
```

```
TR1:                                -commit complete
SQL> COMMIT ;
```

6.3.

```
TR1:                                Montantaide =110
SQL> SELECT * FROM famille
WHERE nofam=1 ;
```

```
Montantaide =110
TR2:
SQL> SELECT * FROM famille
WHERE nofam=1 ;
```

```
TR1:                                - 1 row updated
SQL> UPDATE famille                                - montantaide=140
SET montantaide = montantaide + 30
WHERE nofam=1;
```

```
SQL> SELECT * FROM famille
WHERE nofam=1 ;
```

```
-pour TR2 : montantaide=110
TR2:
SQL> SELECT * FROM famille
WHERE nofam=1 ;
```

```
SQL> UPDATE famille
SET montantaide = montantaide - 15
WHERE nofam=1;
```

```
TR1:                                -commit complete
SQL> COMMIT ;
```

```
- TR2 est débloquée et exécute la mise à jour (montantaide=140-15).
1 row updated
TR2:
??????
```

```
- la valeur de l'attribut montantaide =125
SQL> SELECT * FROM famille
WHERE nofam=1 ;
```

6.4.

```
TR1:                                COUNT(TYPEACT)
SQL> SELECT COUNT (typeact)
FROM action                                -----
WHERE typeact=100 ;                                2
```

```
- 1 row updated
TR2:
SQL> UPDATE Action
SET typeact = 100
WHERE noact='Act23';
```

```
-commit complete
```

```
SQL>COMMIT ;
TR1:                                COUNT(TYPEACT)
SQL> SELECT COUNT (typeact)
FROM action                                -----
WHERE typeact=100 ;                                3
```