Bases de données – ING1
TD 10 : Transactions

Il s'agit de se familiariser avec la notion de transaction à travers :

- Annulation de transaction;
- Validation de transaction;
- Jalonnage d'une transaction;
- Transaction en lecture consistante;
- Gestion d'accès concurrents.

Exercice 1 : Rappels de cours

a) Qu'est-ce qu'une transaction ?

Solution de la question

Un ensemble ordonné d'opérations de mise à jour (insert, update, delete) d'une base de données qui sera entièrement validé ou entièrement annulé.

b) Quel sont les intérêts des quatre propriétés (ACID) d'une transaction ?

Solution de la question

Atomicité : Une transaction est entièrement validée ou entièrement annulé. Cela découle de la cohérence d'une transaction.

Cohérence : Une transaction permet d'assurer l'intégrité de la base de données en regroupant des opérations qui doivent impérativement l'être.

Isolation : Dans un environnement multi-utilisateur, tant qu'une transaction n'est pas terminée elle ne doit pas être visible par d'autres utilisateurs. Autrement, les autres utilisateurs pourraient constater des incohérences entre les données. Une autre raison est que tant qu'une transaction n'est pas terminée, on ne sait pas si elle va être validée ou annulée.

Durabilité : Quand une transaction s'est bien terminée, des données ont été mises à jour. C'est l'un des principes fondamentaux d'une base de données que de savoir conserver des données validées.

c) Créer ou détruire une table sont-elles des opérations d'une transaction ?

Solution de la question

Non, seules les opérations de type INSERT, UPDATE ou DELETE apparaissent dans une transaction. En conséquence, les opérations CREATE, ALTER et DROP sont prises en compte immédiatement par toutes les transactions en cours.

Exercice 2 : ROLLBACK

Pour la suite de ce TD, on vous demande de créer la table `transac` comme présentée ci-après :

```
CREATE TABLE transac (a NUMBER(5), b VARCHAR2(5), CONSTRAINT pk_transac PRIMARY KEY (a));
```

En suite, ouvrez deux sessions de SQLPLUS et insérez dans la table `transac` les lignes suivantes dans la première session sans valider ou annuler :

(1, 'ABC') et (2, 'DEF').

```
SQL> insert into transac values(1,'ABC');
1 row created.
SQL> insert into transac values(2,'DEF');
1 row created.
```

Question 1 : Juste après l'insertion effectuer un SELECT sur les deux sessions. Que constatez-vous ?

Solution de la question

Sélection des enregistrements dans la table transac (session 1) :

```
SQL> select * from transac; A B ---- ----- 1 ABC 2 DEF
```

Sélection des enregistrements dans la table transac (session 2) :

```
SQL> select * from transac; no rows selected
```

On constate que seule la session où ont été effectués les ordres INSERT reconnaît les données dans la table transac. Les enregistrements n'apparaissent pas dans la table transac de la session 2.

Question 2 : A partir de la session où vous avez fait les insertions, utiliser l'ordre ROLLBACK, et exécutez le même SELECT. Que constatez-vous ?

Solution de la question

Dans la session 1 :

```
SQL> rollback ; Rollback complete.
SQL> select * from transac; no rows selected
```

L'ordre ROLLBACK annule toutes les opérations de la transaction de la session 1.

Exercice 3 : Point de reprise ou jalon

Valider toutes les transactions ouvertes (COMMIT;)

Puis exécuter toutes les commandes suivantes dans l'ordre :

1. Insérer la ligne (1, 'ABC') dans la première session;
2. Valider la transaction dans la première session;
3. Insérer la ligne (2, 'DEF') dans la première session;
4. Définir un point de reprise dans la première session;
5. Insérer la ligne (3, 'GHI') dans la première session;
6. Effectuer le SELECT sur les 2 sessions ouvertes. Que constatez-vous ?

7. Annuler dans la première session, la transaction courante jusqu'au point de reprise défini plus haut;
8. Exécuter les SELECTs dans les deux sessions;
9. Valider la transaction de la session 1, exécuter les SELECTs dans les deux sessions et constater.

Solution de la question

Dans la session 1:

```
SQL> insert into transac values(1,'ABC'); 1 row created.
SQL> commit; Commit complete.
SQL> insert into transac values(2,'DEF'); 1 row created.
SQL> savepoint reprise1 ; Savepoint created.
SQL> insert into transac values(3,'GHI'); 1 row created.
SQL> select * from transac; --------- ----- 1 ABC 2 DEF 3 GHI
```

Dans la session 2 :

```
SQL> select * from transac; A B --------- ----- 1 ABC
```

On remarque que seule la transaction validée est visible dans les deux sessions.

Au niveau de la session 1, on annule les opérations jusqu'au point de reprise.

```
SQL> rollback to reprise1; Rollback complete.
```

Au niveau de la session 1 :

```
SQL> select * from transac; A B --------- ----- 1 ABC 2 DEF
```

Au niveau de la session 2 :

```
SQL> select * from transac; A B --------- ----- 1 ABC
```

On valide la transaction courante de la session 1.

```
SQL> commit; Commit complete.
```

Dans les sessions 1 et 2, on constate

```
SQL> select * from transac; A B --------- ----- 1 ABC 2 DEF
```

Exercice 4 : Deadlock

Dans la session 1, supprimer le contenu de la table `transac` puis insérer les deux lignes (1,'abc') et (2,'def') et valider la transaction.

Effectuez dans l'ordre :

1. Dans la session 1, mettre à jour la ligne 1 en remplaçant 'abc' par '123';
2. Dans la session 2, mettre à jour la ligne 2 en remplaçant 'def' par '456';
3. Dans la session 1, mettre à jour la ligne 2 en remplaçant 'def' par 'ijk';
4. Dans la session 2, mettre à jour la ligne 1 en remplaçant 'abc' par 'klm';
5. Dans la session 1, faites une sélection sur la table `transac`;
6. Dans la session 1, valider la transaction;
7. Dans les sessions 1 et 2, faites une sélection sur la table `transac`;

Que constatez-vous ?

Solution de la question

Dans la session 1

```
SQL> update transac set b='123' where a=1; 1 row updated.
```

Dans la session 2

```
SQL> update transac set b='456' where a=2; 1 row updated.
```

Dans la session 1

```
SQL> update transac set b='ijk' where a=2;
```

On obtient aucun message (même pas le prompt SQL>) ou le message "query is executing"

Dans la session 2

```
SQL> update transac set b='klm' where a=1;
```

On constate que la session 2 est bloquée. Après un certain temps d'attente, dans la session 1, on a le message suivant :

```
ERREUR à la ligne 1 : ORA-00060: deadlock detected while waiting for
resource
```

La session 1 est débloquée

Dans la session 1

```
SQL> select * from transac; A B ---------- ---- 1 123 2 def
```

On constate que l'ordre **update transac set b='ijk' where a=2;** a été retiré de la transaction.

Dans la session 1

```
SQL> commit; Validation effectuée.
```

La session 2 est débloquée.

Dans les deux sessions

```
SQL> select * from transac; A B ---------- ---- 1 klm 2 456
```

On constate les dernières mises à jour des deux lignes, faites dans la session 2 ont été prises en compte. En fait la validation de la transaction de la session 1 a servi à débloquer la transaction de la session 2. C'est pour cela qu'en définitive, seules les opérations de la transaction de la session 2 ont eu un effet.