

## Objectifs

Il s'agit de se familiariser avec la notion de transaction.

- 1) Annulation de transaction
- 2) Validation de transaction

Pour le TP, on vous demande de créer la table tran comme présenté ce-après.

create table tran

(a number(5) primary key,

b varchar(5));

## Sommaire des exercices

- 1 - Transaction : create et drop
- 2 - Rollback ...
- 3 - Point de reprise ...
- 4 - Blockage ...
- 5 - Le projet

## Corps des exercices

### 1 - Transaction : create et drop

**Enoncé :**

Considérons les ordres CREATE et DROP.

#### Question 1)

Enoncé de la question

La création et la suppression d'une table est-elle transactionnelle ?

Solution de la question

La création et la suppression d'une table n'est pas transactionnelle.

### 2 - Rollback ...

**Enoncé :**

On vous demande d'insérer les lignes suivantes dans l'une des deux sessions :  
(1, 'ABC') et (2, 'DEF').

#### Question 1)

Enoncé de la question

Juste après l'insertion effectuer un select sur les deux sessions ouvertes.  
que constatez vous ?

Solution de la question

*Insertion des deux lignes :(session 1)*

```
SQL> insert into tran values(1,'ABC') ;
```

*1 row created.*

```
SQL> insert into tran values(2,'DEF');
```

*1 row created.*

*selection des enregistrements dans la table tran (section 1) :*

```
select * from tran;
```

```
  A B  
----  
 1 ABC  
 2 DEF
```

*selection des enregistrements dans la table tran (section 2) :*

```
SQL> select * from tran;
```

*no rows selected*

*=> On constate que seule la session où ont été effectuée les ordres INSERT reconnaît les données dans la table tran. Les enregistrements n'apparaissent pas dans la table tran de la session 2.*

## Question 2)

Enoncé de la question

A partir de la session où vous avez fait les insertions, utiliser l'ordre "rollback", et exécutez le même select.

Constater.

Solution de la question

*Dans la session 1 :*

```
SQL> rollback ;
```

*Rollback complete.*

```
SQL> select * from tran;
```

*no rows selected*

*=> L'ordre rollback annule toute transaction effectuée depuis le dernier commit ou l'ouverture de la session si aucun commit effectué.*

## 3 - Point de reprise ...

Enoncé :

Valider toutes les transactions ouvertes (commit;)

Puis exécuter toutes les commandes suivantes dans l'ordre :

- Insérer la ligne (1, 'ABC')
- Définir un point de reprise
- Insérer la ligne (2, 'DEF')

- Effectuer le select sur les 2 sessions ouvertes.
- Annuler les transactions jusqu'au point de reprise definit plus haut. Executer les selects.
- Valider les transactions, executer les selects et constater.

### Question 1)

Enoncé de la question

Solution de la question

```
SQL> insert into tran values(1,'ABC') ;
```

```
1 row created.
```

```
SQL> commit;
```

```
Commit complete.
```

```
SQL> savepoint reprise1 ;
```

```
Savepoint created.
```

```
SQL> insert into tran values(2,'DEF');
```

```
1 row created.
```

```
SQL> select * from tran;
```

```

      A B
-----
1 ABC
2 DEF
```

Dans la session 2 :

```
SQL> select * from tran;
```

```

      A B
-----
1 ABC
```

=> On remarque que seule la transaction commitée est visible dans les deux sessions.

Au niveau de la session 1, on annule les transaction :

```
SQL> rollback to reprise1;
```

```
Rollback complete.
```

Au niveau de la session 1 :

```
SQL> select * from tran;
```

```

      A B
-----
1 ABC
```

Au niveau de la session 2 :

```
SQL> select * from tran;
```

```

      A B
-----
1 ABC
```

## 4 - Blockage ...

## Enoncé :

Supprimer le contenu de la table tran puis insérer les deux lignes (1,'ABC') et (2,'DEF') et valider la transaction.

### Question 1)

Enoncé de la question

Effectuez dans l'ordre :

- Dans la session 1 mettre à jour la ligne 1 en remplaçant 'abc' par '123'
- Dans la session 2 mettre à jour la ligne 2 en remplaçant 'def' par '456'
- Dans la session 1 mettre à jour la ligne 2 en remplaçant 'def' par 'ijk'
- Dans la session 2 mettre à jour la ligne 1 en remplaçant 'abc' par 'klm'

Constater.

Solution de la question

*Dans la session 1*

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

*Dans la session 2*

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

*Dans la session 1*

```
SQL> update tran set b='ijk' where a=2;  
=> on obtient un fenêtre pop-up avec working..."query is executing ..."  
Dans la session 2
```

```
SQL> update tran set b='klm' where a=1;  
=> on obtient un fenêtre pop-up avec working..."query is executing ..."
```

## 5 - Le projet

### Enoncé :

#### Question 1)

Enoncé de la question

Continuez le projet.

Solution de la question