

Compte Rendu TP3 PL-SQL

Réalisé par Rochdi Mohamed Amine & Majid Ayoub

1. Créant in tableau de nom Transa :

```
1 CREATE TABLE transa (  
2 ID NUMBER(5) PRIMARY KEY,  
3 NOM VARCHAR2(20)  
4 );
```

```
SQL> CREATE TABLE transa (  
2 ID NUMBER(5) PRIMARY KEY,  
3 NOM VARCHAR2(20)  
4 );
```

Table created.

```
SQL> |
```

- 2.

```
SQL> CREATE TABLE transa (  
2 ID NUMBER(5) PRIMARY KEY,  
3 NOM VARCHAR2(20)  
4 );  
  
Table created.  
SQL>
```

DOC	TABLE
PAT	TABLE
RDV	TABLE
ORD	TABLE
MED	TABLE
DET	TABLE
E_TITRE	TABLE
E_SERVICE	TABLE
E_EMPLOYE	TABLE
E_CLIENT	TABLE
E_PRODUIT	TABLE
E_ENTREPOT	TABLE
E_STOCK	TABLE
E_COMMANDE	TABLE
E_LIGNE	TABLE
BIN\$6YZEpXOpTLyLsUy0rB9dGA==\$0	TABLE
→	TRANSA ←
	TABLE

3. La création d'une table est généralement une opération automatiquement validée (commit) dans la plupart des systèmes de gestion de base de données (SGBD), donc vous devriez normalement voir la table immédiatement dans l'autre session.
- 4.

```
SQL> Drop table transa;
Table dropped.
SQL>
```

```
NOV
TABLE
ORD
TABLE
MED
TABLE
DET
TABLE
E_TITRE
TABLE
E_SERVICE
TABLE
E_EMPLOYE
TABLE
E_CLIENT
TABLE
E_PRODUIT
TABLE
E_ENTREPOT
TABLE
E_STOCK
TABLE
E_COMMANDE
TABLE
E_LIGNE
TABLE
BIN$vw9/WsJoSj13c/M0EhV23g==$0
```

5. Vous constaterez probablement que la création et la suppression d'une table sont des opérations automatiquement validées (commit) dans la plupart des SGBD. Cela signifie que les changements sont visibles immédiatement dans d'autres sessions, et ils ne sont pas spécifiquement soumis à la gestion de transactions manuelles. Cependant, la transaction peut être utilisée pour regrouper plusieurs opérations et les valider en bloc.

6.

```
1 INSERT INTO transa (ID, NOM) VALUES (1, 'Amine');
2 INSERT INTO transa (ID, NOM) VALUES (2, 'Ayoub');
3 INSERT INTO transa (ID, NOM) VALUES (3, 'Aymen');
4 INSERT INTO transa (ID, NOM) VALUES (4, 'Hamid');
```

```
SQL> INSERT INTO transa (ID, NOM) VALUES (1, 'Amine');
1 row created.
SQL> INSERT INTO transa (ID, NOM) VALUES (2, 'Ayoub');
1 row created.
SQL> INSERT INTO transa (ID, NOM) VALUES (3, 'Aymen');
1 row created.
SQL> INSERT INTO transa (ID, NOM) VALUES (4, 'Hamid');
1 row created.
SQL> select * from transa;
  ID NOM
----
1|Amine
2|Ayoub
3|Aymen
4|Hamid
4 rows selected.
```

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

7.

a. Modification d'une ligne : Modifions, par exemple, la valeur de la colonne "NOM" pour l'ID 4.

```
1 UPDATE transa SET NOM = 'Hicham' WHERE ID = 4;
```

```
SQL> select * from transa;

  ID|NOM
-----|-----
  1|Amine
  2|Ayoub
  3|Aymen
  4|Hamid

4 rows selected.

SQL> UPDATE transa SET NOM = 'Hicham' WHERE ID = 4;


1 row updated.

SQL> select * from transa;

  ID|NOM
-----|-----
  1|Amine
  2|Ayoub
  3|Aymen
  4|Hicham

4 rows selected.
```

- b. Suppression d'une ligne : Supprimons, par exemple, la ligne avec l'ID 3.



```
1 DELETE FROM transa WHERE ID = 3;
```

```
SQL> select * from transa;

  ID|NOM
-----|-----
  1|Amine
  2|Ayoub
  3|Aymen
  4|Hicham

4 rows selected.

SQL> DELETE FROM transa WHERE ID = 3;

1 row deleted.

SQL> select * from transa;

  ID|NOM
-----|-----
  1|Amine
  2|Ayoub
  4|Hicham ←

3 rows selected.
```

- c. Annulation des mises à jour avec ROLLBACK : Utilisons ROLLBACK pour annuler les modifications apportées dans cette session.



```
1 ROLLBACK;
```

```
SQL> select * from transa;

      ID|NOM
-----|-----
      1|Amine
      2|Ayoub
      4|Hicham

3 rows selected.

SQL> rollback;

Rollback complete.

SQL> select * from transa;

no rows selected

SQL> |
```

Après l'exécution de la commande ROLLBACK, toutes les modifications (UPDATE, DELETE) seront annulées, ramenant la base de données à l'état précédent le début de la transaction.

8. Structure. sa et table la de contenu le de contenu le Vérifier.


```
SQL> select * from transa;

no rows selected

SQL> |
```

9. En conclusion, l'utilisation de transactions peut être cruciale pour garantir la cohérence des données et pour permettre l'annulation d'un ensemble d'opérations si nécessaire. Il est important de comprendre comment les transactions fonctionnent dans le système de gestion de base de données spécifique que vous utilisez.
10. Insérer à nouveau trois ou quatre lignes, les modifier et les détruire partiellement, puis valider (en écrivant « COMMIT ; ») ces mises à jour, D'accord, poursuivons avec l'insertion, la modification, la suppression partielle, et la validation des mises à jour en utilisant COMMIT. Voici comment vous pouvez le faire :

- a. **Insertion de trois ou quatre lignes** : Ajoutons, par exemple, trois nouvelles lignes à la table "transa".



```
1 INSERT INTO transa (ID, NOM) VALUES (1, 'Amine');
2 INSERT INTO transa (ID, NOM) VALUES (2, 'Ayoub');
3 INSERT INTO transa (ID, NOM) VALUES (3, 'Aymen');
4 INSERT INTO transa (ID, NOM) VALUES (4, 'Hamid');
```

```
SQL> INSERT INTO transa (ID, NOM) VALUES (1, 'Amine');
1 row created.

SQL> INSERT INTO transa (ID, NOM) VALUES (2, 'Ayoub');
1 row created.

SQL> INSERT INTO transa (ID, NOM) VALUES (3, 'Aymen');
1 row created.

SQL> INSERT INTO transa (ID, NOM) VALUES (4, 'Hamid');
1 row created.

SQL> select * from transa;

      ID|NOM
-----|-----
      1|Amine
      2|Ayoub
      3|Aymen
      4|Hamid

4 rows selected.
```

- b. **Modification de lignes** : Modifions, par exemple, la valeur de la colonne "NOM" pour l'ID 4.



```
1 UPDATE transa SET NOM = 'Hicham' WHERE ID = 4;
```

```
SQL> select * from transa;

      ID|NOM
-----|-----
      1|Amine
      2|Ayoub
      3|Aymen
      4|Hamid

4 rows selected.

SQL> UPDATE transa SET NOM = 'Hicham' WHERE ID = 4;
1 row updated.

SQL> select * from transa;

      ID|NOM
-----|-----
      1|Amine
      2|Ayoub
      3|Aymen
      4|Hicham

4 rows selected.
```

- c. **Suppression partielle** : Supprimons, par exemple, la ligne avec l'ID 3.



```
1 DELETE FROM transa WHERE ID = 3;
```

```
SQL> select * from transa;

  ID|NOM
-----|-----
  1|Amine
  2|Ayoub
  3|Aymen
  4|Hicham

4 rows selected.

SQL> DELETE FROM transa WHERE ID = 3;

1 row deleted.

SQL> select * from transa;

  ID|NOM
-----|-----
  1|Amine
  2|Ayoub
  4|Hicham

3 rows selected.
```

- d. **Validation des mises à jour avec COMMIT** : Utilisons COMMIT pour valider les modifications apportées dans cette session.



```
1 COMMIT;
```

```
SQL> commit;

Commit complete.
```

11.



```
1 ROLLBACK;
```

```
SQL> rollback;

Rollback complete.
```

12. Maintenant detruire les donnees de votre table et valider.

```

SQL> select * from transa;

      ID|NOM
-----|-----
      1|Amine
      2|Ayoub
      4|Hicham

3 rows selected.

SQL> DELETE FROM transa;

3 rows deleted.

SQL> commit;

Commit complete.

SQL> select * from transa;

no rows selected

```

La première commande DELETE FROM transa supprime toutes les lignes de la table "transa". Ensuite, la commande COMMIT valide la suppression, rendant les changements permanents dans la base de données.

Assurez-vous de comprendre les implications de cette opération, car elle effacera toutes les données de la table. Si vous avez besoin de conserver une copie des données avant la suppression, vous devriez envisager de faire une sauvegarde au préalable.

13.

```

SQL> INSERT INTO transa (ID, NOM) VALUES (1, 'Amine');

1 row created.

SQL> INSERT INTO transa (ID, NOM) VALUES (2, 'Ayoub');

1 row created.

SQL> INSERT INTO transa (ID, NOM) VALUES (3, 'Aymen');

1 row created.

SQL> INSERT INTO transa (ID, NOM) VALUES (4, 'Hamid');

1 row created.

SQL> select * from transa;

      ID|NOM
-----|-----
      1|Amine
      2|Ayoub
      3|Aymen
      4|Hamid

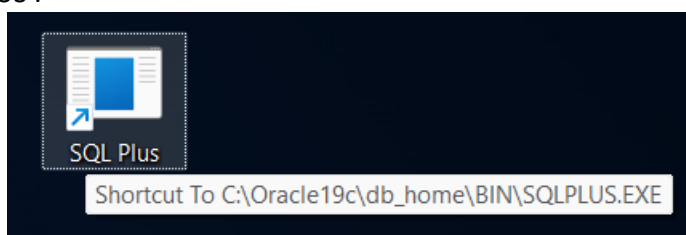
4 rows selected.

SQL> EXIT

```

DISCONNECTED FROM ORACLE

14. Ouvrir à nouveau SQL PLUS :



```

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> select * from transa;

   ID NOM
-----
    1 Amine
    2 Ayoub
    3 Aymen
    4 Hamid

```

Conclusion : EXIT exécute automatiquement COMMIT

15.

```

1  INSERT INTO transa (ID, NOM) VALUES (5, 'Deae');
2  INSERT INTO transa (ID, NOM) VALUES (6, 'Noura');

SQL> select * from transa;

   ID NOM
-----
    1 Amine
    2 Ayoub
    3 Aymen
    4 Hamid

SQL> INSERT INTO transa (ID, NOM) VALUES (5, 'Deae');
1 row created.

SQL> INSERT INTO transa (ID, NOM) VALUES (6, 'Noura');
1 row created.

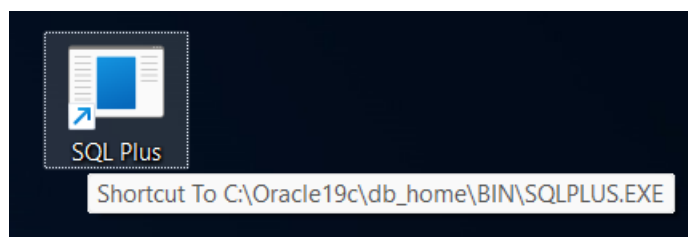
SQL> select * from transa;

   ID NOM
-----
    6 Noura
    1 Amine
    2 Ayoub
    3 Aymen
    4 Hamid
    5 Deae

6 rows selected.

```

16.




```
Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0
```

```
SQL> select * from transa;
```

ID	NOM
1	Amine
2	Ayoub
3	Aymen
4	Hamid

Conclusion : on constate quand on ferme la fenêtre brutalement la transaction ne commit pas.

17.

```
SQL> select * from transa;
```

ID	NOM
1	Amine
2	Ayoub
3	Aymen
4	Hamid

```
SQL> INSERT INTO transa (ID, NOM) VALUES (5, 'Deae');
```

1 row created.

```
SQL> INSERT INTO transa (ID, NOM) VALUES (6, 'Noura');
```

1 row created.

```
SQL> select * from transa;
```

ID	NOM
5	Deae
6	Noura
1	Amine
2	Ayoub
3	Aymen
4	Hamid

6 rows selected.

```
SQL> select * from transa;
```

ID	NOM
5	Deae
6	Noura
1	Amine
2	Ayoub
3	Aymen
4	Hamid

6 rows selected.

```
SQL> ALTER TABLE transa ADD (AGE Number(2));
```

Table altered.

```
SQL> select * from transa;
```

ID	NOM	AGE
5	Deae	
6	Noura	
1	Amine	
2	Ayoub	
3	Aymen	
4	Hamid	

6 rows selected.

```

SQL> select * from transa;

      ID NOM                AGE
-----
      5 Deae
      6 Noura
      1 Amine
      2 Ayoub
      3 Aymen
      4 Hamid

6 rows selected.

SQL> ROLLBACK;

Rollback complete.

SQL> select * from transa;

      ID NOM                AGE
-----
      5 Deae
      6 Noura
      1 Amine
      2 Ayoub
      3 Aymen
      4 Hamid

6 rows selected.

SQL> DESC transa;
Name                               Null?    Type
-----
ID                                NOT NULL NUMBER(5)
NOM                               VARCHA2(20)
AGE                               NUMBER(2)

```

Conclusion :

Les nouvelles insertions seront annulées par la commande ROLLBACK, car elles n'ont pas été validées par un COMMIT avant d'exécuter ROLLBACK.

L'ajout d'une nouvelle colonne à la table avec ALTER TABLE n'est pas affecté par le ROLLBACK. L'ajout de colonnes est une opération de définition de schéma qui n'est pas annulée par ROLLBACK.

En exécutant DESC transa, vous devriez voir la nouvelle colonne ajoutée, même si les insertions précédentes ont été annulées.

Il est essentiel de comprendre que les commandes ROLLBACK n'annulent que les modifications de données non validées, et non les changements de schéma tels que l'ajout de colonnes.

18.



```

1  DELETE FROM transa;

```

19.

```

SQL> INSERT INTO transa (ID, NOM) VALUES (1, 'Amine');
1 row created.

SQL> INSERT INTO transa (ID, NOM) VALUES (2, 'Ayoub');
1 row created.

SQL> INSERT INTO transa (ID, NOM) VALUES (3, 'Deae');
1 row created.

SQL> INSERT INTO transa (ID, NOM) VALUES (4, 'Noura');
1 row created.

SQL> select * from transa;

```

ID	NOM	AGE
1	Amine	
2	Ayoub	
3	Deae	
4	Noura	

20.

```

1 row created.

SQL> INSERT INTO transa (ID, NOM) VALUES (6, 'hicham');
1 row created.

SQL> INSERT INTO transa (ID, NOM) VALUES (7, 'kamal');
1 row created.

SQL> SAVEPOINT p1;
Savepoint created.

SQL> select * from transa;

```

ID	NOM	AGE
1	Amine	
2	Ayoub	
3	Deae	
4	Noura	
5	rocmine	
6	hicham	
7	kamal	

```

7 rows selected.

```

21.

```

SQL> INSERT INTO transa (ID, NOM) VALUES (8, 'nora');
1 row created.

SQL> INSERT INTO transa (ID, NOM) VALUES (9, 'zineb');
1 row created.

SQL> INSERT INTO transa (ID, NOM) VALUES (10, 'adam');
1 row created.

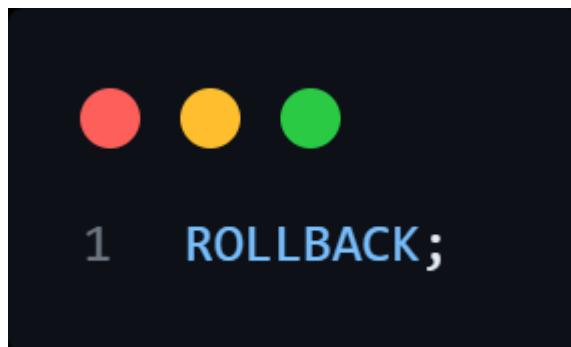
SQL> SAVEPOINT p2;
Savepoint created.

SQL> select * from transa;

      ID NOM                AGE
-----
      1 Amine
      2 Ayoub
      3 Deae
      4 Noura
      5 rocmine
      6 hicham
      7 kamal
      8 nora
      9 zineb
     10 adam
10 rows selected.

```

22.



```

SQL> ROLLBACK;
Rollback complete.

SQL> select * from transa;

      ID NOM                AGE
-----
      5 Deae
      6 Noura
      1 Amine
      2 Ayoub
      3 Aymen
      4 Hamid
6 rows selected.

```

23.

```

SQL> INSERT INTO transa (ID, NOM) VALUES (7, 'nora');
1 row created.

SQL> INSERT INTO transa (ID, NOM) VALUES (8, 'zineb');
1 row created.

SQL> INSERT INTO transa (ID, NOM) VALUES (9, 'adma');
1 row created.

SQL> SAVEPOINT p3;
Savepoint created.

SQL> SAVEPOINT p3;
Savepoint created.

SQL> select * from transa;

      ID NOM                                AGE
-----
      5 Deae
      6 Noura
      1 Amine
      7 nora
      2 Ayoub
      3 Aymen
      4 Hamid
      8 zineb
      9 adma

9 rows selected.

```

24.

```

SQL> INSERT INTO transa (ID, NOM) VALUES (10, 'adma');
1 row created.

SQL> INSERT INTO transa (ID, NOM) VALUES (11, 'adma');
1 row created.

SQL> INSERT INTO transa (ID, NOM) VALUES (12, 'adma');
1 row created.

SQL> SAVEPOINT p4;
Savepoint created.

SQL> select * from transa;

      ID NOM                                AGE
-----
      5 Deae
      6 Noura
      1 Amine
      7 nora
      2 Ayoub
      3 Aymen
      4 Hamid
      8 zineb
      9 adma
     10 adma
     11 adma

      ID NOM                                AGE
-----
     12 adma

12 rows selected.

```

25.



```
1  
2 COMMIT;  
3 select * from transa;
```

```
SQL> COMMIT;  
Commit complete.  
SQL> select * from transa;  
  
      ID NOM                AGE  
-----  
      5 Deae  
      6 Noura  
      1 Amine  
      7 nora  
      2 Ayoub  
      3 Aymen  
      4 Hamid  
      8 zineb  
      9 adma  
     10 adma  
     11 adma  
  
      ID NOM                AGE  
-----  
     12 adma  
  
12 rows selected.
```

La commande COMMIT valide toutes les modifications apportées au cours de la transaction, les rendant permanentes dans la base de données

26.



```
1 delete from transa;
```

```
SQL> delete from transa;

12 rows deleted.

SQL> select * from transa;

no rows selected
```

27.

```
SQL> INSERT INTO transa (ID, NOM) VALUES (13, 'kamale');
1 row created.

SQL> INSERT INTO transa (ID, NOM) VALUES (14, 'bader');
1 row created.

SQL> INSERT INTO transa (ID, NOM) VALUES (15, 'nora');
1 row created.

SQL> SAVEPOINT p5;
Savepoint created.

SQL> select * from transa;

      ID NOM                AGE
-----
      13 kamale
      14 bader
      15 nora
```

28.

```
SQL> INSERT INTO transa (ID, NOM) VALUES (16, 'hatim');
1 row created.

SQL> INSERT INTO transa (ID, NOM) VALUES (17, 'ayoub');
1 row created.

SQL> INSERT INTO transa (ID, NOM) VALUES (18, 'anwaer');
1 row created.

SQL> SAVEPOINT p6;
Savepoint created.
```

29.



```
1 rollback to p6;
```

30.

```
SQL> rollback to p6;

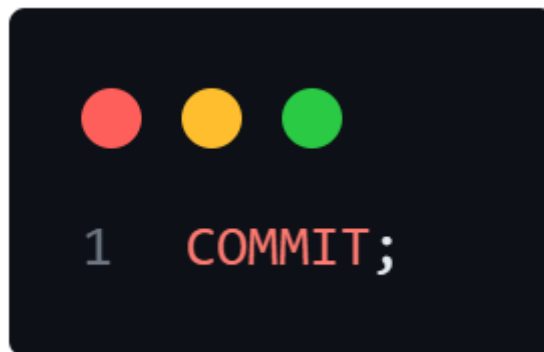
Rollback complete.

SQL> select * from transa;

      ID NOM      AGE
-----
      13 kamale
      14 bader
      15 nora
      16 hatim
      17 ayoub
      18 anwaer

6 rows selected.
```

31.



```
1 COMMIT;
```

32.



```
1 delete from transa;
```

33.

```
SQL> INSERT INTO transa (ID, NOM) VALUES (1, 'hatim');

1 row created.

SQL> INSERT INTO transa (ID, NOM) VALUES (2, 'ayoub');

1 row created.

SQL> INSERT INTO transa (ID, NOM) VALUES (3, 'anwaer');

1 row created.
```

34.

```
SQL> ALTER TABLE transa RENAME COLUMN age TO salary;

Table altered.
```

35.



```
1 desc transa;
```

36.



```
1 select * from transa;
```

```
SQL Plus
SQL> select * from transa;

      ID NOM                SALARY
-----
1 hatim
2 ayoub
3 anwaer

SQL> |
```