



**ECOLE MAROCAINE DES
SCIENCES DE L'INGENIEUR**
Membre de **HONORIS UNITED UNIVERSITIES**

INGÉNIERIE INFORMATIQUE ET RÉSEAUX
EMSI SOUSSI – 3^e IIR G12

TRANSACTION SOUS ORACLE

DECEMBRE 2022

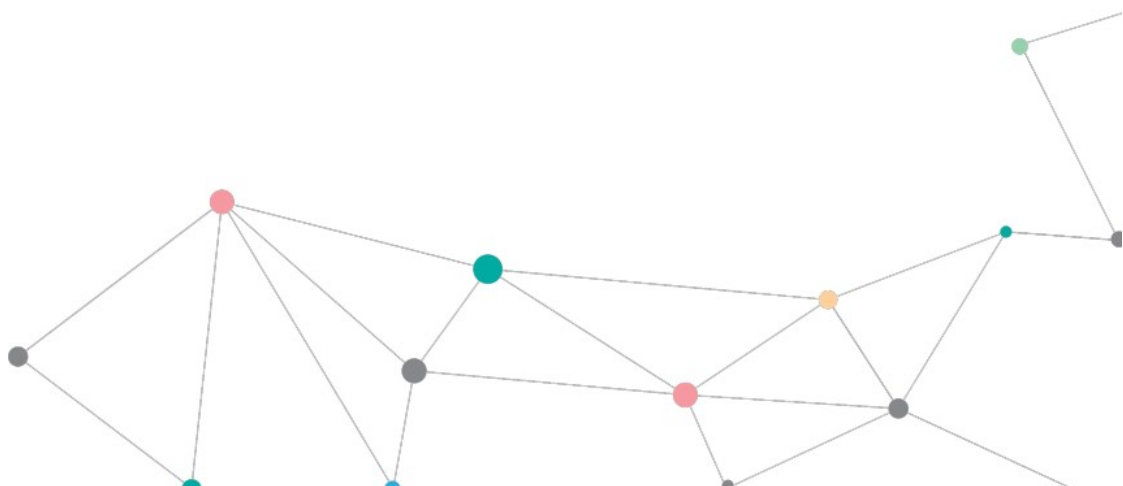
Professeur :

Mme.EL HAIBA Maria

Réalisé par :

BAZTAMI Ismail

EL ALAMI Khalil



BASES DE DONNEES RELATIONNELLES :

Partie 1: Transaction et ordre CREATE TABLE ET DROP TABLE

Pour réaliser cette première partie du TP, vous devez avoir deux sessions différentes ouvertes sur la même base (connectez vous avec deux SQLPLUS avec le même compte)

1. créer la table "Transa" comme présenté ci-après.

```
CREATE TABLE Transa (  
ID NUMBER(5) PRIMARY KEY,  
NOM VARCHAR2(20));
```

2. Considérons les **ordres CREATE et DROP**. La création et la suppression d'une table sont-elles transactionnelles ?

Pour vérifier cela, avec vos deux connexions, tentez de créer la table "Transa" dans une transaction, et de vérifier dans l'autre session si vous la voyez .

3. Que constatez-vous ?

Avec desc Transa.

Nous avons la description de la table Transa dans la 2ème session.

4. Exécutez la même tentative avec un DROP.

5. Conclusion ?

Les commandes du langage de définition des données (LDD) : CREATE, ALTER, DROP sont transactionnelles, ces commandes valident implicitement une transaction : Transaction implicite.

```
Run SQL Command Line × + ▾
Grant succeeded.

SQL> disconnect
Disconnected from Oracle Database 11g Express Edition Release 11.2.0.2.0 - 6
4bit Production
SQL> connect khalil/ismail;
Connected.
SQL> desc Transa;
   Name                                         Null?    Type
-----
ID                                             NOT NULL NUMBER(5)
NOM                                           VARCHAR2(20)

SQL> drop table Transa;

Table dropped.

SQL>
```

```
Run SQL Command Line × + ▾
SQL*Plus: Release 11.2.0.2.0 Production on Mar. ĐÚc. 13 00:08:06 2022

Copyright (c) 1982, 2014, Oracle. All rights reserved.

SQL> connect khalil/ismail;
Connected.
SQL> CREATE TABLE Transa (
  2  ID NUMBER(5) PRIMARY KEY,
  3  NOM VARCHAR2(20)
  4 );

Table created.

SQL> desc Transa;
ERROR:
ORA-04043: object Transa does not exist

SQL> |
```

Partie 2 : Atomicité d'une transaction courante :

6. Insérez trois ou quatre lignes dans la table transa et les voir.

```
SQL> insert into Transa values (1,'ismail');
1 row created.

SQL> insert into Transa values (2,'khalil');
1 row created.

SQL> insert into Transa values (3,'yassine');
1 row created.

SQL> insert into Transa values (4,'yahya');
1 row created.

SQL> select * from Transa;

      ID NOM
-----
      1 ismail
      2 khalil
      3 yassine
      4 yahya

SQL> |
```

7. Modifiez une ligne, en supprimer une autre, enfin annuler les mises à jour venant d'être effectuées (en écrivant « ROLLBACK ; »).

```
SQL> update Transa set nom='yahya' where id=3;
1 row updated.

SQL> delete from Transa where id=4;
1 row deleted.

SQL> select * from Transa;

      ID NOM
-----
      1 ismail
      2 khalil
      3 yahya

SQL> ROLLBACK ;

Rollback complete.
```

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

Le contenu de la table est vide.

```
SQL> ROLLBACK ;  
  
Rollback complete.  
  
SQL> select * from Transa;  
  
no rows selected
```

9. Conclusion?

Avec " Rollback ; " Nous avons : L'annulation de la transaction.
et le défaut toutes les mises à jour de la transaction.

Les commandes langage de manipulation des données (LMD) :
SELECT, INSERT, UPDATE, DELETE, ne sont pas transactionnelles.

10. Insérer à nouveau trois ou quatre lignes, les modifier et les détruire partiellement,
puis valider (en écrivant « COMMIT ; ») ces mises à jour,

```
1 row created.  
  
SQL> insert into Transa values (2,'khalil');  
  
1 row created.  
  
SQL> insert into Transa values (3,'yassine');  
  
1 row created.  
  
SQL> insert into Transa values (4,'yahya');  
  
1 row created.  
  
SQL> update Transa set nom='yahya' where id=3;  
  
1 row updated.  
  
SQL> delete from Transa where id=4;  
  
1 row deleted.  
  
SQL> commit;  
  
Commit complete.
```

11. Faites maintenant un ROLLBACK. Que s'est-il passé ?

Les lignes insérer sont affichée parce qu'elles sont validées avec commit.

```
SQL> rollback;

Rollback complete.

SQL> select* from Transa ;

      ID NOM
-----
      1 ismail
      2 khalil
      3 yahya
```

13. Insérer à nouveau dans votre table vide trois ou quatre lignes et clore la transaction par un EXIT .

```
SQL> delete from Transa where id in (1,2,3);

3 rows deleted.

SQL> insert into Transa values (1,'ismail');

1 row created.

SQL> insert into Transa values (2,'khalil');

1 row created.

SQL> insert into Transa values (3,'yassine');

1 row created.

SQL> EXIT;|
```

14. Reconnectez-vous à SQLPLUS. Que s'est-il passé ? Expliquez

Le contenu de la table est affiché lorsqu'on relance le bloc de SQL*Plus après clore de la transaction par un EXIT (quit) et on se reconnecte, on voit bien que le tuple est désormais stocké dans la table.

Donc EXIT effectue une transaction implicite.

```
Run SQL Command Line
SQL> connect
Enter user-name: khalil
Enter password:
Connected.
SQL> select * from Transa ;

      ID NOM
-----
       1 ismail
       2 khalil
       3 yassine
```

15. Dans votre table, insérez à nouveau deux ou trois lignes dans la table et fermez brutalement votre session.

```
SQL> insert into Transa values(4,'Laila');

1 row created.

SQL> insert into Transa values(5,'Hiba');

1 row created.
```

16. Reconnectez-vous à SQLPLUS. Les données saisies ont-elles été préservées ? Expliquez !

Les nouvelles lignes ne sont pas affichées car la fermeture brutale sans commit ne préserve pas les modifications effectuées par les commandes LMD.

```
SQL> connect khalil/ismail;
Connected.
SQL> select * from Transa;

      ID NOM
-----
       1 ismail
       2 khalil
       3 yassine

SQL> |
```

17. Insérer à nouveau deux ou trois lignes dans la table, puis ajouter une nouvelle colonne à la table et essayer d'annuler les dernières insertions puis faites un DESC de la table.

Conclusion : Les lignes ajoutées sont validées implicitement avec la commande ALTER, qui est une commande du LDD.

```
SQL> insert into Transa values (4,'Laila');
```

```
1 row created.
```

```
SQL> insert into Transa values (5,'Hiba');
```

```
1 row created.
```

```
SQL> alter table Transa add Age number(4);
```

```
Table altered.
```

```
SQL> select * from Transa;
```

ID	NOM	AGE
1	ismail	
2	khalil	
3	yassine	
4	Laila	
5	Hiba	

18. Videz votre table par un delete.

19. Insérez trois ou quatre lignes dans la table transa et les voir.

```
SQL> insert into Transa values (1,'ismail',20);
```

```
1 row created.
```

```
SQL> insert into Transa values (2,'khalil',21);
```

```
1 row created.
```

```
SQL> insert into Transa values (3,'yahya',22);
```

```
1 row created.
```

```
SQL> select * from Transa;
```

ID	NOM	AGE
1	ismail	20
2	khalil	21
3	yahya	22

20. Insérez deux ou trois lignes puis faites une sauvegarde partielle de la transaction (SAVEPOINT)

21. Insérez deux ou trois lignes puis faites une sauvegarde partielle de la transaction (SAVEPOINT)

```
SQL> insert into Transa values (4,'yassine',23);  
1 row created.  
  
SQL> insert into Transa values (5,'amine',24);  
1 row created.  
  
SQL> savepoint p1;  
Savepoint created.  
  
SQL> insert into Transa values (6,'Hiba',24);  
1 row created.  
  
SQL> insert into Transa values (7,'adam',24);  
1 row created.  
  
SQL> savepoint p2;  
Savepoint created.
```

22. Faites un ROLLBACK puis vérifiez le contenu votre table. Conclusion ?

Les mises à jour de la transaction sont annulées avec ROLLBACK et on affiche la table de la question 17.

```
SQL> rollback;  
Rollback complete.  
  
SQL> select * from Transa;  
  
-----  
ID NOM  
-----  
1 ismail  
2 khalil  
3 yassine  
4 Laila  
5 Hiba
```

23. Insérez deux ou trois lignes puis faites une sauvegarde partielle de la transaction (SAVEPOINT)
24. Insérez deux ou trois lignes puis faites une sauvegarde partielle de la transaction (SAVEPOINT)

```
SQL> insert into Transa values (6,'yahya',22);
1 row created.

SQL> savepoint p1;
Savepoint created.

SQL> insert into Transa values (7,'Mohammed',22);
1 row created.

SQL> savepoint p2;
Savepoint created.
```

25. Faites un COMMIT puis vérifiez le contenu votre table. Conclusion?

COMMIT valide toute les mises à jour de la transaction.

```
SQL> COMMIT;
Commit complete.

SQL> select * from Transa;

   ID NOM                AGE
-----
   6 yahya                22
   7 Mohammed            22
   1 ismail
   2 khalil
   3 yassine
   4 Laila
   5 Hiba

7 rows selected.
```

26. Videz votre table par un delete.

27. Insérez deux ou trois lignes puis faites une sauvegarde partielle de la transaction (SAVEPOINT)

28. Insérez deux ou trois lignes puis faites une sauvegarde partielle de la transaction (SAVEPOINT)

```
SQL> insert into Transa values (1,'ismail',22);
1 row created.

SQL> insert into Transa values (2,'khalil',23);
1 row created.

SQL> savepoint p1;
Savepoint created.

SQL> insert into Transa values (3,'Laila',24);
1 row created.

SQL> insert into Transa values (4,'Hiba',25);
1 row created.

SQL> savepoint p2;
Savepoint created.
```

29. Faites une annulation partielle de la transaction (ROLLBACK TO ...)

30. Vérifier le contenu de la table par un SELECT.

```
SQL> rollback to p1;
Rollback complete.

SQL> select *from Transa;

  ID NOM                AGE
-----
  1 ismail                22
  2 khalil                23
```

31. Faites un COMMIT. Et vérifiez le contenu de la table. Conclusion ?

Puisqu'on a validé avec COMMIT après ROLLBACK to P1 :
Les modifications enregistrer sont celles de la 1^{er} sauvegarde partielle de la transaction.

```
SQL> commit;

Commit complete.

SQL> select *from Transa;
```

ID	NOM	AGE
1	ismail	22
2	khalil	23

Partie 3 : Transaction et ordre DDL :

32. Videz votre table par un delete.
33. Insérez deux ou trois lignes.
34. Faites un ALTER TABLE (par exemple, modification du schéma logique en général).
35. Faites un DESC de la table
36. Vérifiez le contenu de la table. Que constatez-vous? Conclusion??

```
SQL> insert into Transa values (2,'khalil',24);

1 row created.
```

```
SQL> insert into Transa values (1,'ismail',23);

1 row created.
```

```
SQL> ALTER TABLE Transa add Tel number(15) ;

Table altered.
```

```
SQL> desc Transa;
```

Name	Null?	Type
ID	NOT NULL	NUMBER(5)
NOM		VARCHAR2(20)
AGE		NUMBER(4)
TEL		NUMBER(15)

```
SQL> select * from Transa ;
```

ID	NOM	AGE	TEL
2	khalil	24	
1	ismail	23	

Les lignes ajoutées sont affichées.

La commande ALTER est une commande du LDD.

Les LDD sont transactionnelles : Transaction implicite.