PL/SQL FOR ORACLE

Transactions & Curseurs

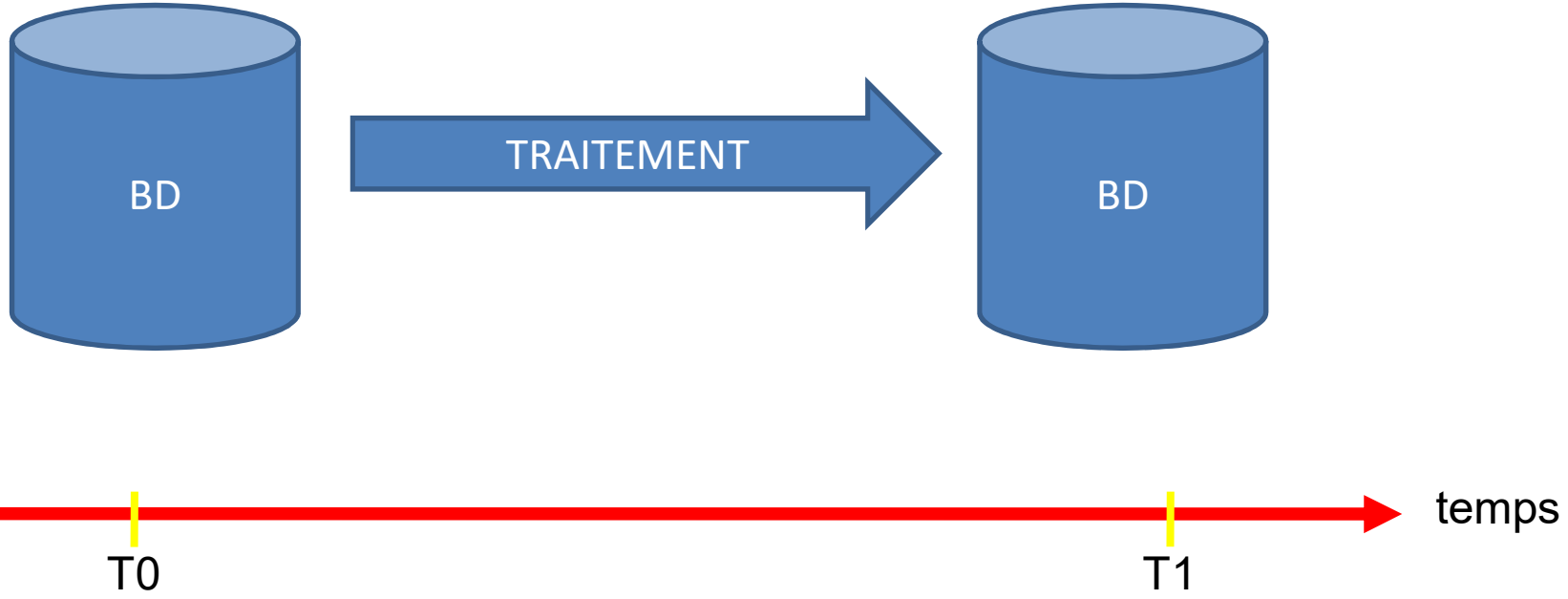
- **Equipe pédagogique**
- Prof. Ikram GHAZAL
- Prof. Mahmoud NASSAR
- Prof. Mohammed SALIHOUN
- Prof. Maria EL HAIBA

SOMMAIRE GENERAL

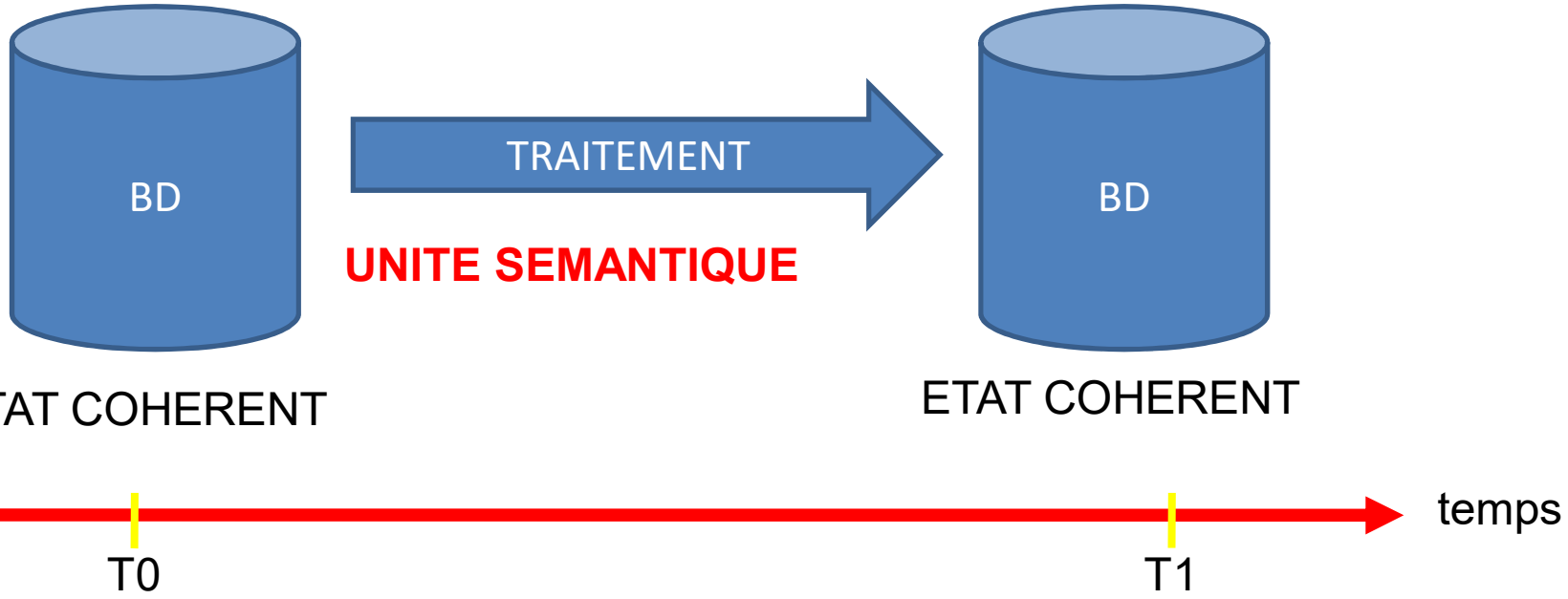
INSERT, UPDATE, DELETE DANS UNE BD

TRANSACTION

PROBLEME:



PROBLEME:



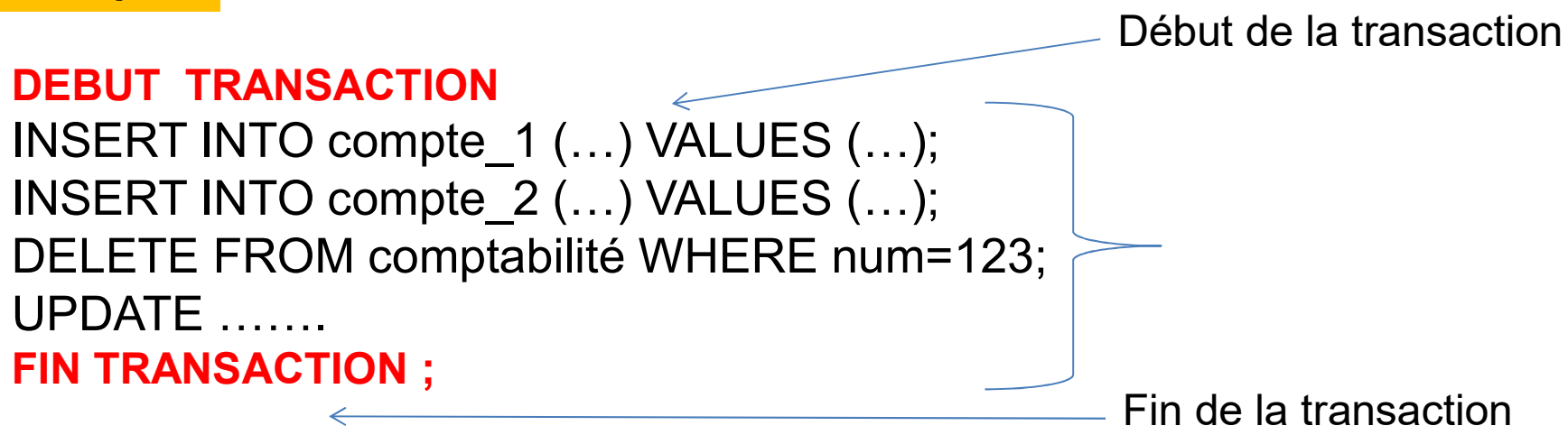
Exemple: Transfert d'argent entre 2 comptes:

```
UPDATE Compte
SETVal = Val -100
Where NumCompte=1;

UPDATE Compte
SETVal = Val + 100
Where NumCompte=2;
```

Transaction: définition

Une transaction est une séquence d'actions <a11, a12, ..., a1ni>

Exemple:



Transaction: Propriétés

Atomicité

Consistance

Isolation

Durabilité

Transaction: Propriétés

Atomicité

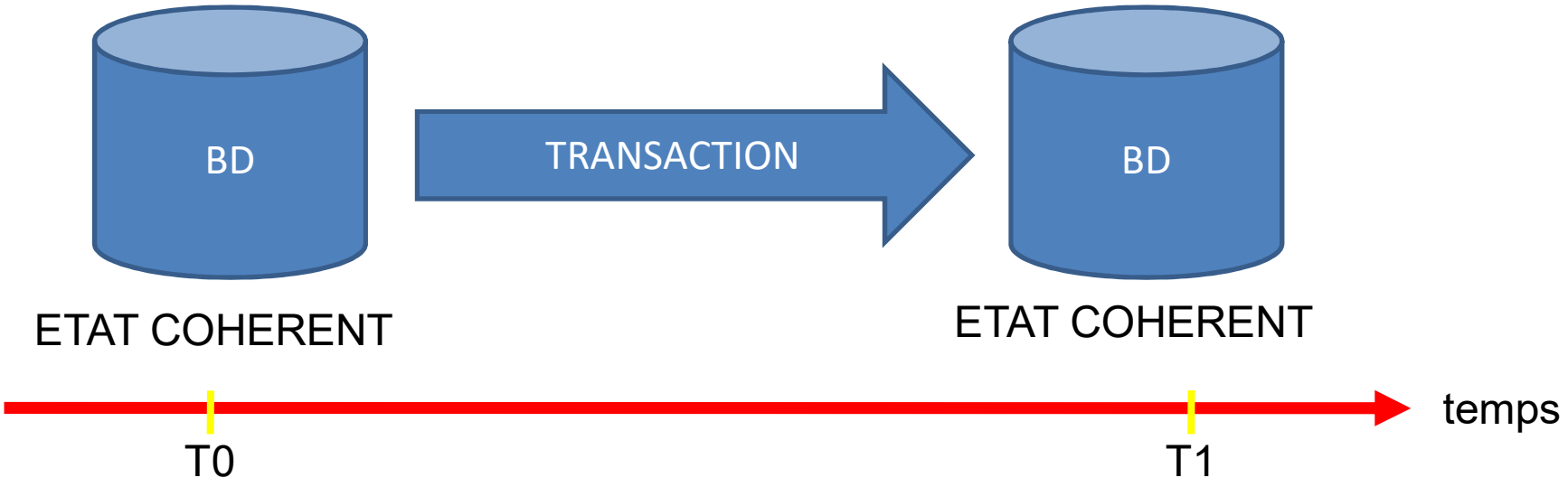
L'ensemble des opérations d'une transaction apparaît comme une seule opération atomique

Soit toutes les opérations sont validées ou toutes annulées (tout ou rien)

Transaction: Propriétés

Consistance

L'exécution de la transaction fait passer la base de données d'un état consistant à un autre état consistant



Transaction: Propriétés

Isolation

Chaque transaction est indépendante des autres transactions concurrentes.

Sérialisation des transactions.

Les résultats d'une transaction ne sont visibles aux autres transactions qu'une fois la transaction validée.

Les concurrences sont parfaitement contrôlées

Transaction: Propriétés

Durabilité

C'est la persistance des mises à jour d'une transaction validée.

Les effets d'une transaction validée sont durables et permanents, quelques soient les problèmes logiciels ou matériels, notamment après la fin de la transaction.

Transaction: primitives de gestion

BEGIN TRANSACTION
SET TRANSACTION
OUVERTURE DE SESSION

DEBUT_DE_TRANSACTION

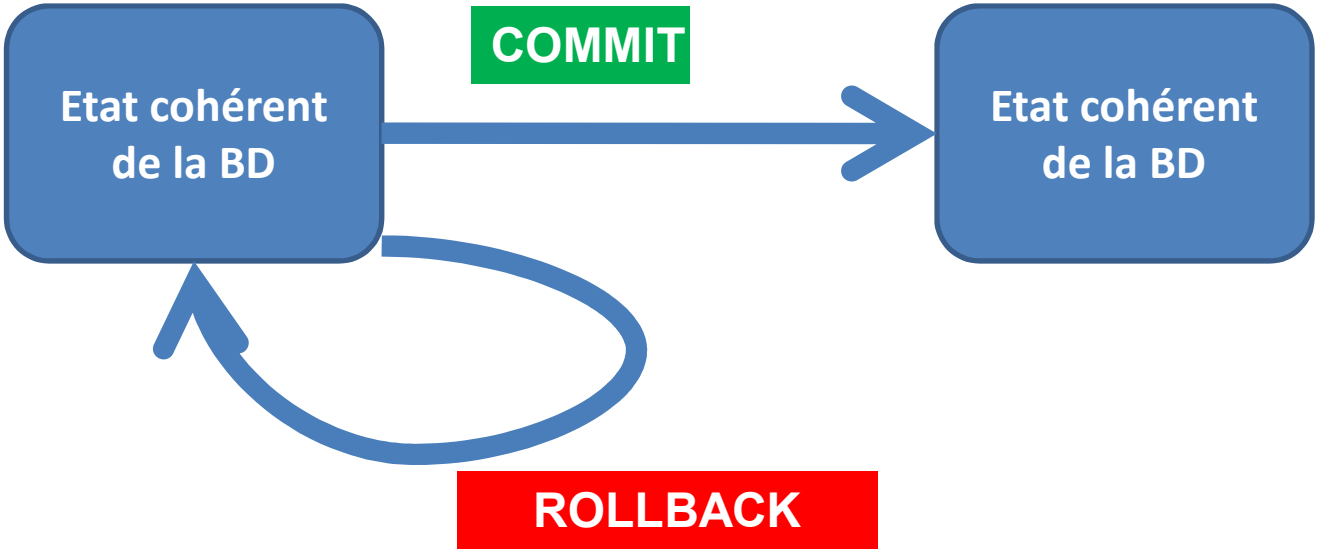
INSERT INTO compte_1 (…) VALUES (…);
……
INSERT INTO compte_2 (…) VALUES (…);
…….
DELETE FROM comptabilité WHERE num=123;
……
FIN_DE_TRANSACTION ;

COMMIT

- Validation de la transaction
- Rend effectives toutes les mises à jour de la transaction

ROLBACK, ABORT

- Annulation de la transaction
- Défait toutes les mises à jour de la transaction


Etat cohérent de la BD
COMMIT
Etat cohérent de la BD
ROLLBACK

TRANSACTION SOUS ORACLE

SET TRANSACTION READ [ONLY | WRITE]
INSERT INTO compte_1 (…) VALUES (…);
INSERT INTO compte_2 (…) VALUES (…);
SAVEPOINT <NOM>
DELETE FROM comptabilité WHERE num=123;
UPDATE …….
SAVEPOINT <NOM>
DELETE FROM
[COMMIT | ROLLBACK] ;

Une transaction commence soit à la connexion ou en début de session, soit à la fin d'une transaction précédente annulée ou validée.

Le début d'une transaction dans Oracle peut être IMPLICITE (pas besoin de SET TRANSACTION ….)

COMMIT	**VALIDE entièrement la transaction**
ROLLBACK	**ANNULE entièrement la transaction**

Les savepoint sont des points de contrôle utilisés dans les transactions pour annuler partiellement l'une d'elles.
Il est possible de définir des Savepoint qui sont des points de contrôle utilisés dans une transaction ainsi on a la possibilité de faire des annulations partielles de transaction.

EXEMPLE 1 : COMMIT et ROOLLBACK

```
SQL> create table trans (
  2  id  number(5) primary key,
  3  nom varchar2(20)
  4  );

Table créée.

SQL>
SQL>
SQL> desc trans;
 Nom                                       NULL ?   Type
 ----------------------------------------- -------- ----------------------------
 ID                                        NOT NULL NUMBER(5)
 NOM                                                VARCHAR2(20)
```

```
SQL> --transaction implicite dans oracle
SQL> insert into trans values (1, ‘Rachid');

1 ligne créée.

SQL>
SQL>
SQL> select * from trans;

        ID NOM
---------- --------------------
         1 Rachid

SQL>
SQL>
SQL> rollback;

Annulation (rollback) effectuée.

SQL>
SQL> select * from trans;

aucune ligne sélectionnée
```

```
SQL>
SQL> insert into trans values (1, ‘Rachid');

1 ligne créée.

SQL> commit;

Validation effectuée.

SQL>
SQL>
SQL> select * from trans;

        ID NOM
---------- --------------------
         1 Rachid

SQL>
```

```
SQL> set transaction read only;

Transaction définie.

SQL>
SQL>
SQL> --test d'insertion
SQL>
SQL> insert into trans values (4, ‘Said');
insert into trans values (4, ‘Said')
            *
ERREUR à la ligne 1 :
ORA-01456: impossible d'exécuter l'opération insérer/supprimer/modifier dans
une transaction READ ONLY



SQL>
SQL> update trans set nom=‘Filali’ where id=1;
update trans set nom=‘Filali' where id=1
       *
ERREUR à la ligne 1 :
ORA-01456: impossible d'exécuter l'opération insérer/supprimer/modifier dans
une transaction READ ONLY
```

```
SQL> set transaction read write;

Transaction définie.

SQL>
SQL> select * from trans;

        ID NOM
---------- --------------------
         1 Rachid

SQL> update trans set nom='Mohmmed' where id=1;

1 ligne mise à jour.

SQL> commit;

Validation effectuée.

SQL> select * from trans;

        ID NOM
---------- --------------------
         1 Mohammed
```

EXEMPLE 2 : SAVEPOINT et ROLLBACK TO

```
SQL> --test savepoint
SQL>
SQL> select * from trans;

aucune ligne sélectionnée

SQL>
SQL> insert into trans values (1, ‘Rachid');

1 ligne créée.

SQL>
SQL> select * from trans;

        ID NOM
---------- --------------------
         1 Rachid

SQL>
SQL> savepoint P1;

Savepoint créé.
```

```
SQL>
SQL> insert into trans values (2, ‘Said');

1 ligne créée.

SQL>
SQL> select * from trans;

        ID NOM
---------- --------------------
         1 Rachid
         2 Said

SQL>
SQL> savepoint P2;

Savepoint créé.

SQL>
```

```
SQL>
SQL> insert into trans values (3, ‘Mohammed');

1 ligne créée.

SQL>
SQL> select * from trans;

        ID NOM
---------- --------------------
         1 Rachid
         2 Said
         3 Mohammed

SQL>
SQL> savepoint P3;

Savepoint créé.
```

```
SQL>
SQL> rollback to P3;

Annulation (rollback) effectuée.

SQL>
SQL> select * from trans;

        ID NOM
---------- --------------------
         1 Rachid
         2 Said
         3 Mohammed
```

```
SQL> rollback to P2;

Annulation (rollback) effectuée.

SQL>
SQL> select * from trans;

        ID NOM
---------- --------------------
         1 Rachid
         2 Said
```

```
SQL> rollback to P1;

Annulation (rollback) effectuée.

SQL>
SQL> select * from trans;

        ID NOM
---------- --------------------
         1 Rachid

SQL> rollback;

Annulation (rollback) effectuée.

SQL>
SQL>
SQL> select * from trans;

aucune ligne sélectionnée
```

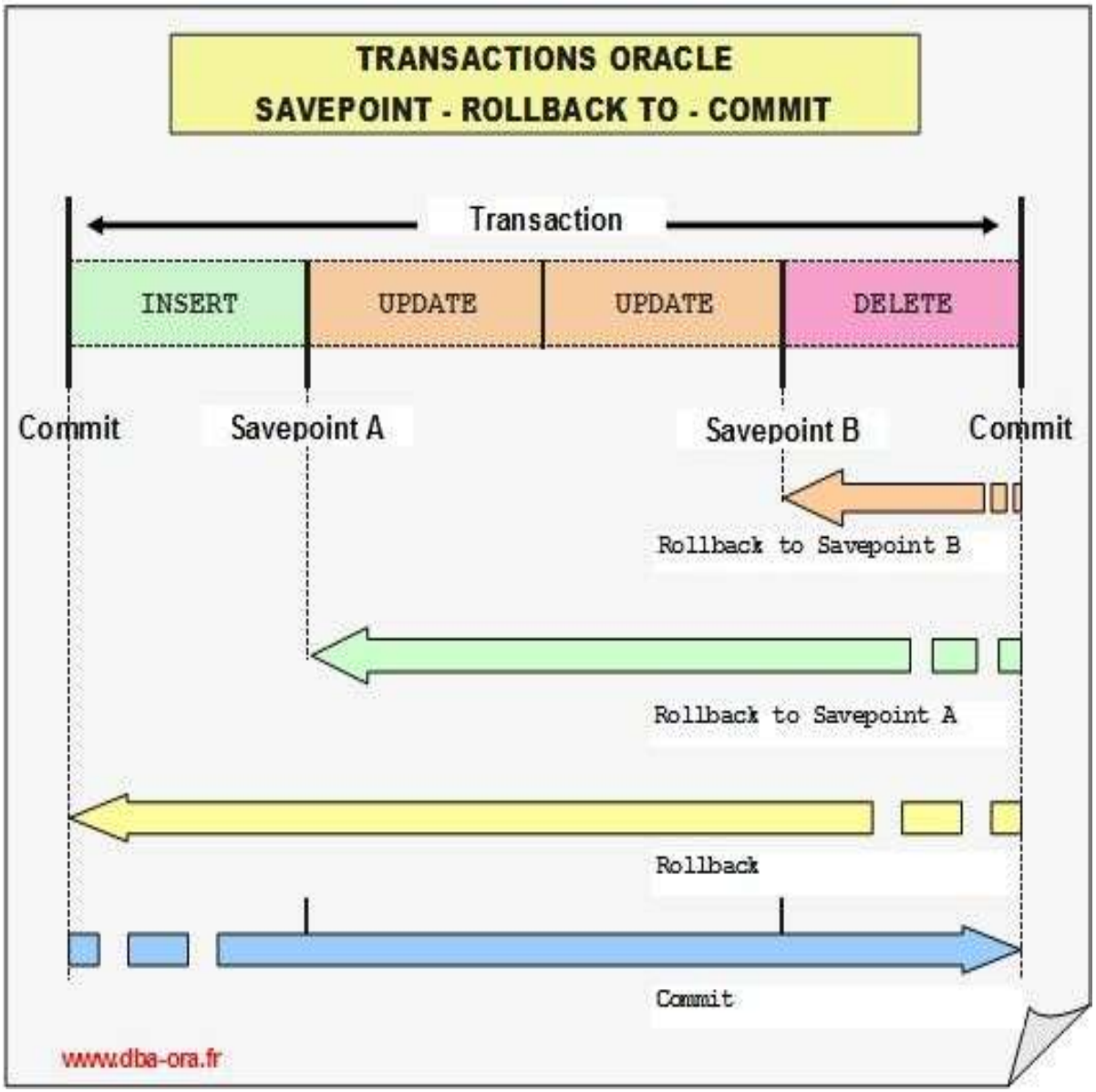
EXEMPLE 3 : SAVEPOINT et ROLLBACK GLOBAL

```
SQL> --test savepoint et rollback et commit;
SQL> insert into trans values (1, ‘Rachid');
1 ligne créée.

SQL> savepoint P1;
Savepoint créé.

SQL> insert into trans values (2, ‘Said');
1 ligne créée.

SQL> savepoint P2;
Savepoint créé.

SQL> select * from trans;

        ID NOM
---------- --------------------
         1 Rachid
         2 Said

SQL> rollback;
Annulation (rollback) effectuée.

SQL> select * from trans;
aucune ligne sélectionnée
```

EXEMPLE 4 : SAVEPOINT et COMMIT GLOBAL

```
SQL> insert into trans values (1, ‘Rachid');
1 ligne créée.

SQL> savepoint P1;
Savepoint créé.

SQL> insert into trans values (2, ‘Said');
1 ligne créée.

SQL> savepoint P2;
Savepoint créé.

SQL> insert into trans values (3, ‘Mohammed');
1 ligne créée.

SQL> commit;
Validation effectuée.

SQL> select * from trans;

        ID NOM
---------- --------------------
         1 Rachid
         2 Said
         3 Mohammed
```

SYNTHESE


TRANSACTIONS ORACLE
SAVEPOINT - ROLLBACK TO - COMMIT
Transaction
INSERT
UPDATE
UPDATE
DELETE
Commit
Savepoint A
Savepoint B
Commit
Rollback to Savepoint B
Rollback to Savepoint A
Rollback
Commit
www.dba-ora.fr

TRAVAUX PRATIQUES

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-aprés.

```
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 ?

4. Exécutez la même tentative avec un DROP.

5. Conclusion ?

Partie 2 : atomicité d'une transaction courante

6. Insérez trois ou quatre lignes dans la table transa et les voir,;
7. Modifiez une ligne, en supprimer une autre, enfin annuler les mises à jour venant d'être effectuées (en écrivant « ROLLBACK ; »).
8. Vérifier le contenu de le contenu de la table et sa structure.
9. Conclusion?
10. Insérer à nouveau trois ou quatre lignes, les modifier et les détruire partiellement, puis valider (en écrivant « COMMIT ; ») ces mises à jour,
11. Faites maintenant un ROLLBACK. Que s'est-il passé ?
12. Maintenant détruire les données de votre table et valider.
13. Insérer à nouveau dans votre table vide trois ou quatre lignes et clore la transaction par un EXIT .
14. Reconnectez vous à SQLPLUS. Que s'est-il passé ? Expliquez
15. Dans votre table, insérez à nouveau deux ou trois lignes dans la table et fermez brutalement votre session.
16. Reconnectez vous à SQLPLUS. Les données saisies ont-elles été préservées ? Expliquez!
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.
18. Videz votre table par un delete.

Partie 2 : atomicité d'une transaction courante

19. Insérez trois ou quatre lignes dans la table transa et les voir,;
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)
22. Faites un ROLLBACK puis vérifiez le contenu votre table. Conclusion?
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)
25. Faites un COMMIT puis vérifiez le contenu votre table. Conclusion?
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)
29. Faites une annulation partielle de la transaction (ROLLBACK TO …)
30. Vérifier le contenu de la table par un SELECT.
31. Faites un COMMIT. Et vérifiez le contenu de la table. Conclusion?

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??