

TP 3 : Privilèges d'accès à la base de données

1. Connectez-vous en tant que SYSTEM à AGRICOLPDB.

```
SQL> ALTER SESSION SET CONTAINER = AGRICOLPDB;
```

2. Créer un tablespace agricolTBS et un tablespace temporaire agricolTempTBS.

```
SQL> CREATE TABLESPACE agricolTBS DATAFILE  
'C:\APP\ORACLE21C\PRODUCT\21C\ORADATA\XE\AGRICOLPDB\agricolTBS.dbf' SIZE  
50M AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED;
```

```
SQL> CREATE TEMPORARY TABLESPACE agricolTempTBS TEMPFILE  
'C:\APP\ORACLE21C\PRODUCT\21C\ORADATA\XE\AGRICOLPDB\agricolTempTBS.dbf'  
SIZE 20M AUTOEXTEND ON;
```

```
SQL> ALTER SESSION SET CONTAINER = AGRICOLPDB;  
Session altered.  
  
SQL> CREATE TABLESPACE agricolTBS  
2 DATAFILE 'C:\APP\ORACLE21C\PRODUCT\21C\ORADATA\XE\AGRICOLPDB\agricol  
3 SIZE 50M  
4 AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED;  
  
Tablespace created.  
  
SQL>  
SQL> CREATE TEMPORARY TABLESPACE agricolTempTBS  
2 TEMPFILE 'C:\APP\ORACLE21C\PRODUCT\21C\ORADATA\XE\AGRICOLPDB\agricol  
3 SIZE 20M  
4 AUTOEXTEND ON;  
  
Tablespace created.
```

3. Créer un utilisateur DBAGRICOL et lui associer les tablespaces précédents. Donner lui tous les privilèges.

```
SQL> CREATE USER DBAGRICOL IDENTIFIED BY dbagricol
```

```
DEFAULT TABLESPACE agricolTBS  
TEMPORARY TABLESPACE agricolTempTBS  
QUOTA UNLIMITED ON agricolTBS;
```

```
SQL> GRANT CONNECT, RESOURCE, DBA TO DBAGRICOL;
```

```
SQL> CREATE USER DBAGRICOL IDENTIFIED BY dbagricol  
2 DEFAULT TABLESPACE agricolTBS  
3 TEMPORARY TABLESPACE agricolTempTBS  
4 QUOTA UNLIMITED ON agricolTBS;  
  
User created.  
  
SQL> GRANT CONNECT, RESOURCE, DBA TO DBAGRICOL;  
  
Grant succeeded.
```

4. Connectez-vous avec l'utilisateur DBAGRICOL@AGRICOLPDB :

```
SQL>CONNECT DBAGRICOL/dbagricol@localhost:1521/AGRICOLPDB
```

- localhost = nom de ta machine (ou IP du serveur).
- 1521 = port par défaut d'Oracle Listener.
- AGRICOLPDB = nom du service (SERVICE_NAME) de ta PDB.

a. Créer les tables du TP1 et y insérer les données (exécuter le fichier TP1.SQL).

```
SQL>@ 'C:\Users\MICROSOFT PRO DZ\Documents\M1TPABD\TP1_P2'
```

b. Créez un autre utilisateur : Admin en lui donnant les mêmes tablespace que DBAGRICOL.

```
SQL>CREATE USER Admin IDENTIFIED BY admin
      DEFAULT TABLESPACE agricoltBS
      TEMPORARY TABLESPACE agricoltTempTBS
      QUOTA UNLIMITED ON agricoltBS;
```

c. Utiliser le dictionnaire de données pour vérifier la création des tablespaces, des utilisateurs et des tables.

```
SQL>SELECT TABLESPACE_NAME, STATUS, CONTENTS FROM DBA_TABLESPACES
WHERE TABLESPACE_NAME IN ('AGRICOLTBS', 'AGRICOLTEMPTBS');
```

TABLESPACE_NAME	STATUS	CONTENTS
AGRICOLTBS	ONLINE	PERMANENT
AGRICOLTEMPTBS	ONLINE	TEMPORARY

```
SQL>SELECT USERNAME, DEFAULT_TABLESPACE, ACCOUNT_STATUS FROM
DBA_USERS WHERE USERNAME IN ('DBAGRICOL', 'ADMIN');
```

USERNAME	DEFAULT_TABLESPACE	ACCOUNT_STATUS
ADMIN	AGRICOLTBS	OPEN
DBAGRICOL	AGRICOLTBS	OPEN

```
SQL>SELECT TABLE_NAME FROM USER_TABLES;
```

TABLE_NAME
AGRICULTEUR
PRODUIT_ALIMENTAIRE
MARCHE
PRODUCTION
APPROVISIONNEMENT

5. Connectez-vous à l'aide de l'utilisateur Admin. Que remarquez-vous ?

```
SQL> connect admin/admin @localhost:1521/AGRICOLPDB
```

- Remarque : L'utilisateur Admin ne peut pas se connecter. Il reçoit l'erreur ORA-01045: user ADMIN lacks CREATE SESSION privilege; logon denied.

- Un utilisateur nouvellement créé n'a, par défaut, aucun privilège système, même pas celui d'ouvrir une session

6. Donnez le droit de création d'une session pour cet utilisateur (Create Session) et reconnectez-vous.

```
SQL> CONNECT DBAGRICOL/dbagricol@localhost:1521/AGRICOLPDB
```

```
SQL> GRANT CREATE SESSION TO Admin;
```

```
SQL> CONNECT DBAGRICOL/dbagricol@localhost:1521/AGRICOLPDB
Connected.
SQL> GRANT CREATE SESSION TO Admin;

Grant succeeded.
```

7. Donnez les privilèges suivants à Admin: créer des tables, des utilisateurs. Vérifier.

```
SQL> GRANT CREATE TABLE TO Admin;
```

```
SQL> GRANT CREATE USER TO Admin;
```

```
SQL> GRANT CREATE TABLE TO Admin;

Grant succeeded.

SQL> GRANT CREATE USER TO Admin;

Grant succeeded.
```

```
SQL> connect admin/admin @localhost:1521/AGRICOLPDB
```

```
SQL> SELECT * FROM SESSION_PRIVS;
```

```
PRIVILEGE
-----
CREATE TABLE
CREATE USER
CREATE SESSION
```

8. Exécutez la requête Q1 suivante : Select * from DBAGRICOL.AGRICULTEUR ; Que remarquez-vous ?

```
SQL> Select * from DBAGRICOL.AGRICULTEUR;
```

-- Remarque : La requête échoue avec l'erreur ORA-00942: table or view does not exist.

-- Même si Admin peut créer ses propres tables, il n'a pas reçu le privilège d'objet SELECT sur la table AGRICULTEUR qui appartient au schéma DBAGRICOL.

9. Donnez le droit de lecture à cet utilisateur sur la table AGRICULTEUR.

```
SQL> CONNECT DBAGRICOL/dbagricol@localhost:1521/AGRICOLPDB
```

```
SQL> GRANT SELECT ON AGRICULTEUR TO Admin;
SQL> Connect admin/admin @localhost:1521/AGRICOLPDB
SQL> SELECT * FROM DBAGRICOL.AGRICULTEUR;
```

AGRICULTEUR_ID

NOM

LOCALISATION

TAILLE_EXPLOITATION

CULTURE_PRINCIPALE

1
Benali Ahmed
Blida

10. On veut créer une vue AGRI_PROD qui sauvegarde pour chaque agriculteur ses produits.

Que faut-il faire ?

-- Pour permettre à Admin de créer une vue (AGRI_PROD) basée sur des tables appartenant à DBAGRICOL, il faut :

-- Donner le privilège système de créer une vue. (Exécuté par DBAGRICOL).

```
SQL> CONNECT DBAGRICOL/dbagricol@localhost:1521/AGRICOLPDB
```

```
SQL> GRANT CREATE VIEW TO Admin;
```

-- Donner le privilège d'objet SELECT sur toutes les tables sources (AGRICULTEUR, PRODUCTION, PRODUIT_ALIMENTAIRE). (Exécuté par DBAGRICOL).

```
SQL> GRANT SELECT ON PRODUCTION TO Admin;
```

```
SQL> GRANT SELECT ON PRODUIT_ALIMENTAIRE TO Admin;
```

-- Création de la vue (Par Admin)

```
SQL> connect admin/admin @localhost:1521/AGRICOLPDB
```

```
SQL> CREATE VIEW AGRI_PROD AS
```

```
SELECT A.nom AS Agriculteur_Nom, P.quantite_produite, PA.nom AS Produit_Nom
FROM DBAGRICOL.Agriculteur A
```

```
JOIN DBAGRICOL.Production P ON A.agriculteur_id = P.agriculteur_id
```

```
JOIN DBAGRICOL.Produit_Alimentaire PA ON P.produit_id = PA.produit_id;
```

```

SQL> CONNECT DBAGRICOL/dbagricol@localhost:1521/AGRICOLPDB
Connected.
SQL> GRANT CREATE VIEW TO Admin;

Grant succeeded.

SQL> GRANT SELECT ON PRODUCTION TO Admin;

Grant succeeded.

SQL> GRANT SELECT ON PRODUIT_ALIMENTAIRE TO Admin;

Grant succeeded.

SQL> connect admin/admin @localhost:1521/AGRICOLPDB
Connected.
SQL> CREATE VIEW AGRI_PROD AS
  2 SELECT A.nom AS Agriculteur_Nom, P.quantite_produite, PA.nom AS Pr
oduit_Nom
  3 FROM DBAGRICOL.Agriculteur A
  4 JOIN DBAGRICOL.Production P ON A.agriculteur_id = P.agriculteur_id

  5 JOIN DBAGRICOL.Produit_Alimentaire PA ON P.produit_id = PA.produit
_id;

View created.

```

11. Créez un index NAMEPROD_IX sur l'attribut NOM de la table PRODUIT. Que remarquez-vous ?

-- Connecté en tant que Admin

```

SQL> CREATE INDEX NAMEPROD_IX ON
DBAGRICOL.PRODUIT_ALIMENTAIRE (NOM);

```

-- Remarque : La commande échoue avec ORA-01031: insufficient privileges.

-- Admin n'a pas le droit de créer un index sur une table qu'il ne possède pas, même s'il a le droit de lecture.

-- Il manque le privilège d'objet INDEX.

12. Donnez le droit de création d'index à Admin sur la table PRODUIT, ensuite réessayez de créer l'index. Que se passe-t-il ?

```

SQL> GRANT INDEX ON PRODUIT_ALIMENTAIRE TO Admin;

```

```

SQL> CREATE INDEX NAMEPROD_IX ON DBAGRICOL.PRODUIT_ALIMENTAIRE
(NOM);

```

13. Afficher tous les privilèges attribués à Admin.

```

SQL> SELECT * FROM SESSION_PRIVS;

```

```
SQL> show user
USER is "ADMIN"
SQL> SELECT * FROM SESSION_PRIVS;
```

```
PRIVILEGE
-----
CREATE VIEW
CREATE TABLE
CREATE USER
CREATE SESSION
```

```
SQL>SELECT * FROM USER_TAB_PRIVS_RECD;
```

```
SQL> SELECT * FROM USER_TAB_PRIVS_RECD;

OWNER
-----
TABLE_NAME
-----
GRANTOR
-----
PRIVILEGE                                GRA HIE COM TYPE
-----
INH
---
DBAGRICOL
PRODUIT_ALIMENTAIRE
DBAGRICOL

OWNER
-----
TABLE_NAME
-----
GRANTOR
-----
PRIVILEGE                                GRA HIE COM TYPE
-----
INH
---
DELETE                                NO NO NO TABLE
NO
```

14. Enlevez les privilèges précédemment accordés.

```
-- Privilèges objet (par DBAGRICOL)
```

```
SQL> REVOKE SELECT ON AGRICULTEUR FROM Admin;
```

```
SQL> REVOKE SELECT ON PRODUCTION FROM Admin;
```

```
SQL> REVOKE SELECT ON PRODUIT_ALIMENTAIRE FROM Admin;
```

```
SQL> REVOKE INDEX ON PRODUIT_ALIMENTAIRE FROM Admin;
```

```
-- Privilèges système (par DBAGRICOL/SYS)
```

```
SQL> REVOKE CREATE SESSION FROM Admin;
```

```
SQL> REVOKE CREATE TABLE FROM Admin;
```

```
SQL> REVOKE CREATE USER FROM Admin;
```

```
SQL> REVOKE CREATE VIEW FROM Admin;
```

```

SQL> -- Privilèges objet (par DBAGRICOL)
SQL> REVOKE SELECT ON AGRICULTEUR FROM Admin;

Revoke succeeded.

SQL> REVOKE SELECT ON PRODUCTION FROM Admin;

Revoke succeeded.

SQL> REVOKE SELECT ON PRODUIT_ALIMENTAIRE FROM Admin;

Revoke succeeded.

SQL> REVOKE INDEX ON PRODUIT_ALIMENTAIRE FROM Admin;

Revoke succeeded.

SQL>
SQL> -- Privilèges système (par DBAGRICOL/SYS)
SQL> REVOKE CREATE SESSION FROM Admin;

Revoke succeeded.

SQL> REVOKE CREATE TABLE FROM Admin;

Revoke succeeded.

SQL> REVOKE CREATE USER FROM Admin;

Revoke succeeded.

SQL> REVOKE CREATE VIEW FROM Admin;|

```

15. Vérifiez que les privilèges ont bien été supprimés.

```

SQL> connect admin/admin @localhost:1521/AGRICOLPDB
Connected.
SQL> SELECT * FROM SESSION_PRIVS;

PRIVILEGE
-----
CREATE SESSION

```

16. Créez un profil « AGRICOL_Profil » qui est caractérisé par : (3 sessions simultanés autorisés, Un appel système ne peut pas consommer plus de 35 secondes de CPU, Chaque session ne peut excéder 90 minutes, Un appel système ne peut lire plus de 1200 blocs de données en mémoire et sur le disque, Chaque session ne peut allouer plus de 25 ko de mémoire en SGA, Pour chaque session, 30 minutes d'inactivité maximum sont autorisés, 5 tentatives de connexion avant blocage du compte, Le mot de passe est valable pendant 50 jours et il faudra attendre 40 jours avant qu'il puisse être utilisé à nouveau, 1 seul jour d'interdiction d'accès après que les 5 tentatives de connexion ont été atteintes, La période de grâce qui prolonge l'utilisation du mot de passe avant son changement est de 5 jours).

```

SQL>CREATE PROFILE AGRICOL_Profil LIMIT
      SESSIONS_PER_USER 3                -- 3 sessions simultanés

```

CPU_PER_CALL 3500	-- 35 secondes de CPU (3500 centièmes)
CONNECT_TIME 90	-- 90 minutes de session
LOGICAL_READS_PER_CALL 1200	-- 1200 blocs lus par appel
PRIVATE_SGA 25K	-- 25 ko de SGA
IDLE_TIME 30	-- 30 minutes d'inactivité
FAILED_LOGIN_ATTEMPTS 5	-- 5 tentatives avant blocage
PASSWORD_LIFE_TIME 50	-- 50 jours de validité du mot de passe
PASSWORD_REUSE_TIME 40	-- 40 jours avant réutilisation
PASSWORD_LOCK_TIME 1	-- 1 jour d'interdiction d'accès
PASSWORD_GRACE_TIME 5;	-- Période de grâce de 5 jours

```
SQL> CONNECT DBAGRICOL/dbagricol@localhost:1521/AGRICOLPDB
Connected.
SQL> CREATE PROFILE AGRICOL_Profil LIMIT
 2     SESSIONS_PER_USER 3
 3     CPU_PER_CALL 3500
 4     CONNECT_TIME 90
 5     LOGICAL_READS_PER_CALL 1200
 6     PRIVATE_SGA 25K
 7     IDLE_TIME 30
 8     FAILED_LOGIN_ATTEMPTS 5
 9     PASSWORD_LIFE_TIME 50
10     PASSWORD_REUSE_TIME 40
11     PASSWORD_LOCK_TIME 1
12     PASSWORD_GRACE_TIME 5;

Profile created.
```

17. Affectez ce profil à l'utilisateur Admin.

```
SQL> ALTER USER Admin PROFILE AGRICOL_Profil;
```

18. Créez le rôle : « MARCHE_MANAGER » qui peut voir les tables AGRICULTEUR, PRODUCTION et peut modifier les lignes de la table MARCHE.

```
SQL> CREATE ROLE MARCHE_MANAGER;
```

```
SQL> ALTER USER Admin PROFILE AGRICOL_Profil;

User altered.

SQL> CREATE ROLE MARCHE_MANAGER;

Role created.
```

19. Assignez ce rôle à Admin. Vérifier que les autorisations assignées au rôle MARCHE_MANAGER, ont été bien transférées à l'utilisateur à Admin.

```
SQL> GRANT MARCHE_MANAGER TO Admin;
```

--verifier

```
SQL> GRANT SELECT ON AGRICULTEUR TO MARCHE_MANAGER;
```

```
SQL> GRANT SELECT ON PRODUCTION TO MARCHE_MANAGER;
```

```
SQL> GRANT UPDATE ON MARCHE TO MARCHE_MANAGER;
```



```

SQL> SELECT * FROM USER_ROLE_PRIVS;

USERNAME
-----
GRANTED_ROLE
-----
ADM DEL DEF OS_ COM INH
--- --- --- --- ---
ADMIN
MARCHE_MANAGER
NO NO YES NO NO NO

SQL> SELECT * FROM ROLE_SYS_PRIVS WHERE ROLE = 'MARCHE_MANAGER';

no rows selected

SQL> SELECT * FROM ROLE_TAB_PRIVS WHERE ROLE = 'MARCHE_MANAGER';

no rows selected

SQL> SELECT * FROM SESSION_ROLES;

ROLE
-----
MARCHE_MANAGER

```