GESTION DES UTILISATEURS

Comptes d'utilisateur de base de données

Chaque compte utilisateur de base de données comporte :

- Un nom utilisateur unique
- Une méthode d'authentification
- Un tablespace par défaut
- Un tablespace temporaire
- Un profil utilisateur

Comptes d'administration prédéfinis

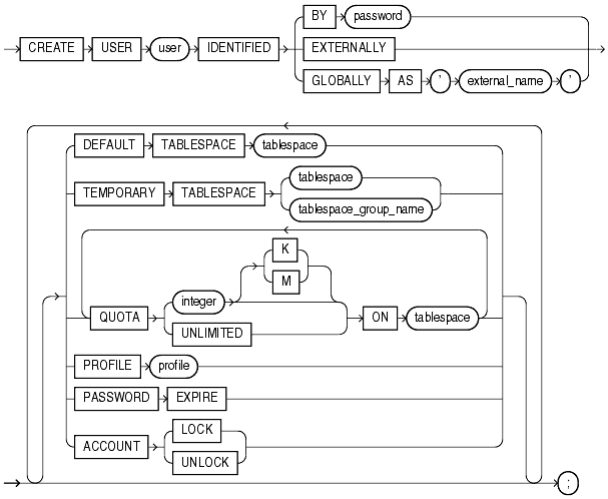
- Deux administrateurs par défaut de la BD : SYS et SYSTEM
  - SYS : Administration de la BD et l'instance
  - SYSTEM : Administration de la BD
- Deux rôles utilisés par les administrateur SYSDBA et SYSOPER (Hérite de SYSDBA sauf CREATE DATABASE)

Créer un utilisateur

Commande :

```
CREATE USER user
IDENTIFIED {BY password |EXTERNALLY|GLOBALLY}
[ DEFAULT TABLESPACE tablespace ]
[ TEMPORARY TABLESPACE tablespace ]
[ QUOTA {integer [K | M ] | UNLIMITED } ON
tablespace
[ PASSWORD EXPIRE ]
[ ACCOUNT { LOCK | UNLOCK }]
[ PROFILE { profile | DEFAULT }]
```

Créer un utilisateur



Source : https://docs.oracle.com/cd/B13789_01/server.101/b10759/statements_8003.htm

Authentification Externe

Le paramètre OS_AUTHENT_PREFIX spécifie un préfixe qu'Oracle utilise pour authentifier les utilisateurs qui tentent de se connecter au serveur.
Oracle concatène la valeur de ce paramètre au début du nom de compte et du mot de passe du système d'exploitation de l'utilisateur.

Sa valeur par défaut est `OPS$`.

```
SQL> show parameter OS_AUTHENT_PREFIX

NAME                         TYPE        VALUE
---------------------------- ----------- -------------
os_authent_prefix             string      OPS$
```

Modification d'un utilisateur

- Modifier le mot de passe

```
Alter User <Login_user> identified by
<nouveau mot de passe>
```

- Modifier les quotas

```
Alter User <Login_user>
Quota 10M on tabs1
Quota 12M on tabs2;
```

Modification d'un utilisateur

- Modifier le tablespace par défaut

```
Alter User <Login_user>
DEFAULT tablespace tbs2
Temporary tablespace temp2
```

- Verrouiller / déverrouiller un compte

```
Alter User <Login_user>
Account lock | unlock
```

Modification d'un utilisateur

- Supprimer un utilisateur avec un schéma vide

```
Drop User <Login_user>
```

- Supprimer un utilisateur avec son schéma

```
Drop User <Login_user> CASCADE
```

Informations sur les utilisateurs

Les vues

- DBA_USERS
- DBA_TS_QUOTAS

Rôles & Privilèges

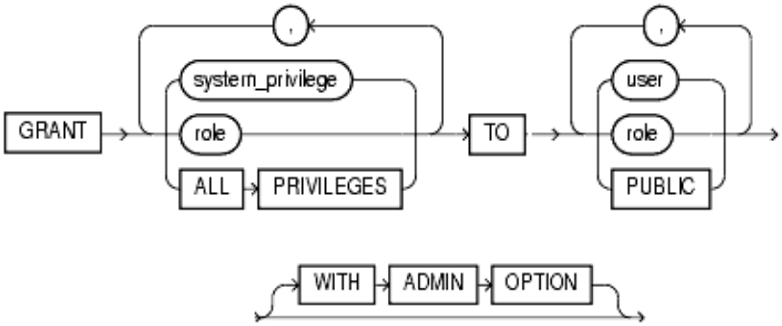
Les rôles et les privilèges sont définis pour sécuriser l'accès aux BDD

Privilèges

Il existe deux types de privilège utilisateur :

- Système : permet aux utilisateurs d'effectuer des actions particulières dans la base de données
  - Spécifique : Create TABLE
  - Générique : CREATE ANY TABLE
- Objet : permet aux utilisateurs d'accéder à un objet spécifique et de le manipuler
- Vue SESSION_PRIVS : liste les privilèges dont les utilisateurs disposent actuellement

Privilèges



Privilèges systèmes

- Commande :

```
GRANT [Priv_SYSTEM|ROLE|ALL PRIVILEGES]
To [USER|ROLE|PUBLIC]
[WITH ADMIN OPTION]
```

- PUBLIC :affecté à tous les utilisateurs
- WITH ADMIN OPTIONS : autorise celui qui a reçu le privilège de l'octroyer à un autre utilisateurs (Si l'utilisateur qui l'a octroyé est supprimé, les autres utilisateurs continueront le travail avec le privilège)
- Vue DBA_SYS_PRIVS : liste les privilèges systèmes dont les utilisateurs disposent actuellement

Privilèges Objet

- Commande :

```
GRANT [Priv_OBJET|[ALL] PRIVILEGES] ON
  object
To [USER|ROLE|PUBLIC]
[WITH GRANT OPTION]
```

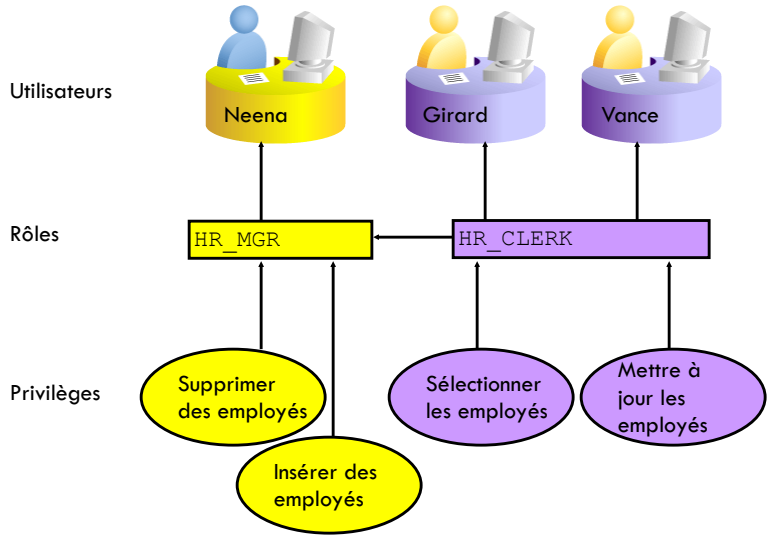
- WITH GRANT OPTIONS : autorise celui qui a reçu le privilège de l'octroyer à un autre utilisateur (Si l'utilisateur qui l'a octroyé est supprimé, les autres utilisateurs n'auront pas le privilège)
- DBA_TAB_PRIVS : liste les privilèges objets dont les utilisateurs disposent actuellement

Privilèges

- Exemple

```
Grant Create session to user1
    With Admim options
Grant Create Table to user 2
    With grant options
```

Rôles



Rôles

```
CREATE ROLE role [NOT IDENTIFIED |
IDENTIFIED
{BY password | EXTERNALLY | GLOBALLY |
USING package}]
```

- Non identifié :

```
CREATE ROLE oe_clerk;
```

- Identifié par mot de passe :

```
CREATE ROLE hr_clerk
IDENTIFIED BY bonus;
```

- Identifié de manière externe :

```
CREATE ROLE hr_manager
IDENTIFIED EXTERNALLY;
```

Rôles

- Exemple

```
CREATE ROLE gestion1;
GRANT SELECT, INSERT, DELETE, UPDATE ON produits TO gestion1
GRANT SELECT, INSERT, UPDATE ON clients TO gestion1

GRANT gestion1 TO user1
```

Les rôles standards

- Trois rôles :
- Connect
- Ressource
- DBA → GRANT DBA TO user1

→ La vue DBA_SYS_PRIVS

```
SQL> select * from dba_sys_privs where grantee='DBA';
```

Les rôles & privilèges assignés à un utilisateur

Liste de rôles assignés à un utilisateur :

Les vues :

- DBA_ROLE_PRIVS
- USER_ROLE_PRIVS

```
SQL> select * from dba_role_privs where grantee='user';
```

- Liste des privilèges objets :
- Les vues : DBA_TAB_PRIVS, ALL_TAB_PRIVS, USER_TAB_PRIVS

```
select * from dba_tab_privs where grantee='SYS';
```

Les rôles & privilèges assignés à un utilisateur

- Liste des rôles assignés à l'utilisateur au cous de sa session

La vue : SESSION_ROLES

- Liste des privilèges assignés à un utilisateur au cours de sa session

La vue : SESSION_PRIVS

Modifier les rôles

- Utilisez `ALTER ROLE` pour modifier la méthode d'authentification.
- Cette commande requiert l'option `ADMIN` ou le privilège `ALTER ANY ROLE`.

```
ALTER ROLE oe_clerk
IDENTIFIED BY order;
```

```
ALTER ROLE hr_clerk
IDENTIFIED EXTERNALLY;
```

```
ALTER ROLE hr_manager
NOT IDENTIFIED;
```

Activer et désactiver les rôles

Commande :

```
SET ROLE {role [ IDENTIFIED BY password ]
         [, role [ IDENTIFIED BY password
           ]]...
         | ALL [ EXCEPT role [, role ]
           ...]
         | NONE }
```

- `SET ROLE hr_clerk;`

```
SET ROLE oe_clerk IDENTIFIED BY order;
```

```
SET ROLE ALL EXCEPT oe_clerk;
```

Supprimer des rôles

- Lorsque vous supprimez un rôle :
  - il est retiré à tous les utilisateurs et rôles auxquels il était accordé,
  - il est supprimé de la base de données.
- La suppression d'un rôle requiert l'option `ADMIN OPTION` ou le privilège `DROP ANY ROLE`.
- Pour supprimer un rôle, utilisez la syntaxe suivante :

```
DROP ROLE hr_manager;
```

Profils

- Un profil est un ensemble nommé contenant les limites relatives aux mots de passe et aux ressources.
- La commande CREATE USER ou ALTER USER permet d'affecter des profils aux utilisateurs.
- Les profils peuvent être activés ou désactivés.
- Par défaut, affectation du profil DEFAULT.

Profils : Gestion des mots de passe

Profils : Gestion des mots de passe

La gestion des mots de passe :

- **Verrouillage d'un compte** : active le verrouillage automatique d'un compte lorsque l'utilisateur ne parvient pas à se connecter au système après un nombre défini de tentatives
- **Durée de vie et expiration des mots de passe** : affecte au mot de passe une durée de vie après laquelle il expire et doit être changé
- **Historique des mots de passe** : vérifie les nouveaux mots de passe pour garantir qu'un mot de passe ne sera pas réutilisé avant un certain temps ou avant un certain nombre de changements de mot de passe
- **Vérification de la complexité des mots de passe** : vérifie qu'un mot de passe est suffisamment complexe pour garantir une protection contre les intrus qui tenteraient de forcer l'accès au système

Profils

Paramètre	Description
FAILED_LOGIN_ATTEMPTS	Nombre d'échecs de connexion avant verrouillage du compte
PASSWORD_LOCK_TIME	Durée, en jours, de verrouillage du compte après le nombre d'échecs de connexion défini
PASSWORD_LIFE_TIME	Durée de vie, en jours, du mot de passe avant expiration
PASSWORD_GRACE_TIME	Période de grâce, en jours, pendant laquelle l'utilisateur peut changer de mot de passe après la première connexion établie une fois le mot de passe expiré

Profils

Paramètre	Description
PASSWORD_REUSE_TIME	Période, en jours, pendant laquelle un mot de passe ne peut pas être réutilisé
PASSWORD_REUSE_MAX	Nombre maximum de réutilisations d'un mot de passe
PASSWORD_VERIFY_FUNCTION	Fonction PL/SQL qui vérifie la complexité d'un mot de passe avant que celui-ci ne soit affecté

Profils

EXEMPLE :

```
CREATE PROFILE grace_5 LIMIT
 FAILED_LOGIN_ATTEMPTS 3
 PASSWORD_LOCK_TIME UNLIMITED
 PASSWORD_LIFE_TIME 30
 PASSWORD_REUSE_TIME 30
 PASSWORD_VERIFY_FUNCTION verify_function
 PASSWORD_GRACE_TIME 5;
```

Cas de l'architecture Multitenant

- Un utilisateur peut être Local ou commun
- Un utilisateur local est le même que ce qui est défini dans le cours
- Un utilisateur commun doit être défini au niveau de la CDB et son nom commence par c##
- Pour octroyer ou révoquer un privilège la syntaxe diffère légerement
- Commande

{GRANT | REVOKE} …….CONTAINER={CURRENT | ALL}

Cas de l'architecture Multitenant

- CDB_USERS donne les informations sur les utilisateurs de la BD(champ common pour voir est ce que c'est un utilisateur commun ou non)
- Cdb_sys_privs donne les informations sur les privilèges détenus par les utilisateurs communs de la BD