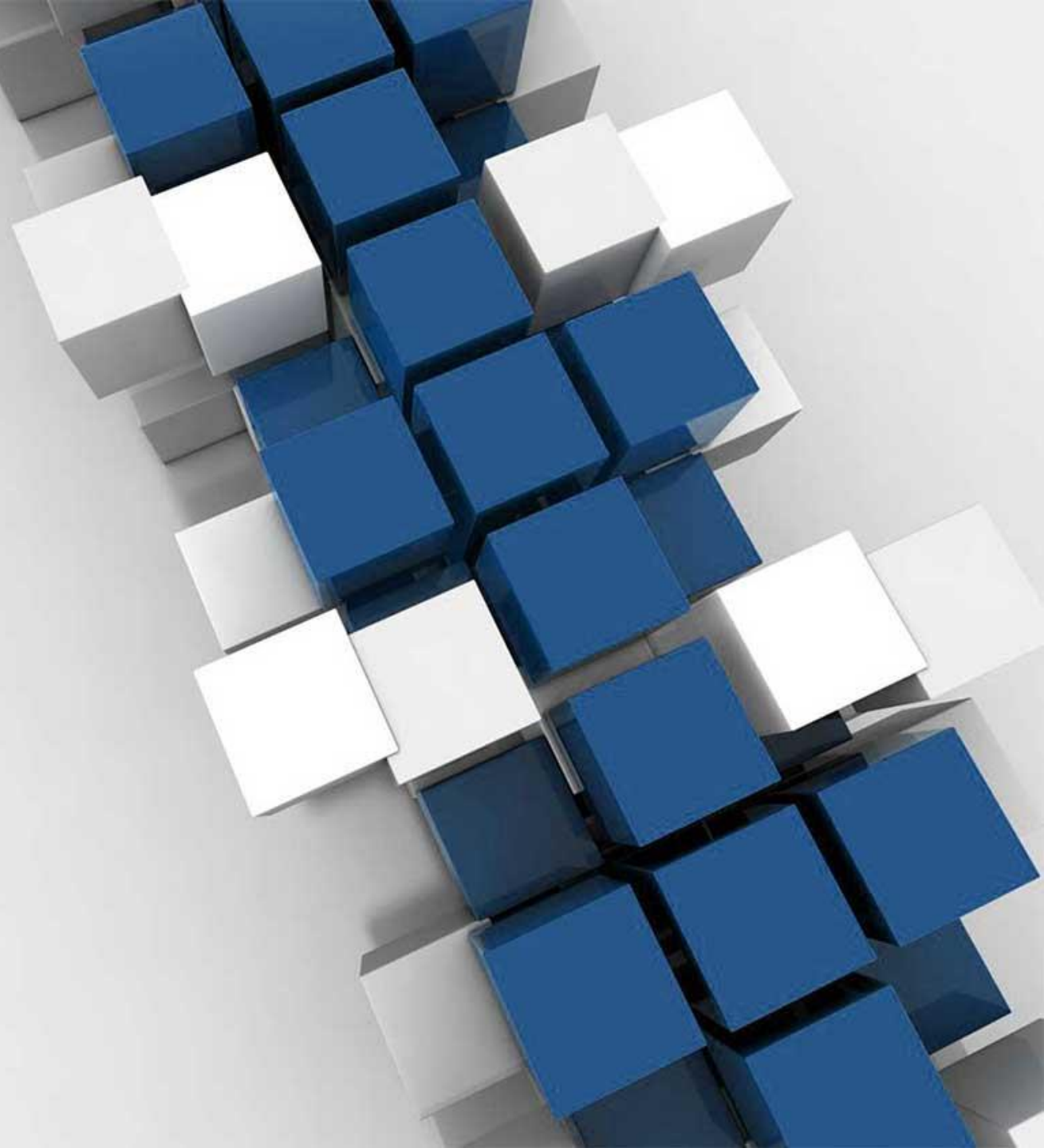
Administration de la Base de Données ORACLE – Partie 2

Pr. Amal KHTIRA

2022/2023

Plan du cours

1) **Gestion des utilisateurs**
2) Sauvegarde et Restauration de la BD
3) Introduction à RMAN
4) Catalogue de restauration RMAN
5) Sauvegarde avec RMAN
6) Restauration et récupération avec RMAN
7) Technologie Flashback



Chapitre 1

Gestion des Utilisateurs

- Gestion des comptes utilisateurs
- Gestion des privilèges
- Gestion des rôles
- Gestion des profils

Gestion des Comptes Utilisateurs

Gestion des Comptes Utilisateur

Définition

Un compte utilisateur de base de données constitue un moyen d'organiser **l'appartenance** des objets de base de données et **l'accès** à ces **objets**. Chaque compte utilisateur comporte les éléments suivants :

- **Nom utilisateur unique** : ne dépasse pas 30 octets, ne doit pas contenir de caractères spéciaux et doit commencer par une lettre.
- **Méthode d'authentification** : par mot de passe, global ou externe (exemple par biométrie, par certificat et par système tiers)
- **Tablespace par défaut** : l'emplacement dans lequel l'utilisateur crée des objets.
- **Tablespace temporaire** : l'emplacement dans lequel l'instance crée les objets temporaires (tris ou tables) pour le compte de l'utilisateur. Aucun quota n'est appliqué aux tablespaces temporaires.
- **Quota** : une allocation d'espace dans un tablespace donné.
- **Profil utilisateur** : Il s'agit d'un ensemble de restrictions de la base de données et les ressources des instances.
- **Statut de verrouillage** : Les utilisateurs n'ont accès qu'aux comptes "déverrouillés".

Gestion des Comptes Utilisateur

Comptes prédéfinis – SYS et SYSTEM

- Le compte **SYS** :
  - Reçoit le rôle d'administrateur de base de données (DBA)
  - Dispose de tous les privlièges associés à ADMIN OPTION
  - Est requis pour les opérations d'arrêt et de démarrage, ainsi que pour certaines commandes de maintenance
  - Est le propriétaire du dictonnaire de données
  - Est le propriétaire du référentiel AWR (Automatic Workload Repository)

- Le compte **SYSTEM** a le rôle d'administrateur de base de données (DBA)

- Ces comptes ne sont pas utilisés pour les opérations de routine.

Gestion des Comptes Utilisateur

Créer un utilisateur

```
CREATE USER user1
IDENTIFIED {BY password | EXTERNALLY | GLOBALLY AS}
[DEFAULT TABLESPACE tablespace]
[TEMPORARY TABLESPACE tablespace]
[QUOTA {integer [K|M] UNLIMITED} ON tablespace]
[PASSWORD EXPIRE]
[ACCOUNT {LOCK | UNLOCK}]
[PROFILE {profile | DEFAULT}]
```

IDENTIFIED

- **BY password** : indique que l'utilisateur est authentifié par la base de données et qu'il doit fournir un mot de passe pour se connecter
- **EXTERNALLY** : indique que l'utilisateur est authentifié par le système d'exploitation
- **GLOBALLY AS** : indique que l'utilisateur est authentifié de façon globale. Exemple par un LDAP

Gestion des Comptes Utilisateur

Créer un utilisateur

```
CREATE USER user1
IDENTIFIED {BY password | EXTERNALLY | GLOBALLY AS}
[DEFAULT TABLESPACE tablespace]
[TEMPORARY TABLESPACE tablespace]
[QUOTA {integer [K|M] UNLIMITED} ON tablespace]
[PASSWORD EXPIRE]
[ACCOUNT {LOCK | UNLOCK}]
[PROFILE {profile | DEFAULT}]
```

QUOTA

- Définit l'espace **maximum** alloué aux objets détenus par l'utilisateur dans le tablespace (le quota peut être défini par un entier représentant des octets ou des kilo-octets et des mégaoctets)
- Le mot-clé **UNLIMITED** permet d'indiquer que les objets détenus par l'utilisateur peuvent utiliser l'ensemble de l'espace disponible du tablespace
- Par défaut, aucun quota de tablespace n'est affecté aux utilisateurs

Gestion des Comptes Utilisateur

Créer un utilisateur

```
CREATE USER user1
IDENTIFIED {BY password | EXTERNALLY | GLOBALLY AS}
[DEFAULT TABLESPACE tablespace]
[TEMPORARY TABLESPACE tablespace]
[QUOTA {integer [K|M] UNLIMITED} ON tablespace]
[PASSWORD EXPIRE]
[ACCOUNT {LOCK | UNLOCK}]
[PROFILE {profile | DEFAULT}]
```

PASSWORD EXPIRE

- Force l'utilisateur à réinitialiser le mot de passe lorsqu'il se connecte à la base de données à l'aide de SQL*Plus
- Cette option n'est valide que si l'utilisateur est authentifié par la base de données

Gestion des Comptes Utilisateur

Créer un utilisateur

```
CREATE USER user1
IDENTIFIED {BY password | EXTERNALLY | GLOBALLY AS}
[DEFAULT TABLESPACE tablespace]
[TEMPORARY TABLESPACE tablespace]
[QUOTA {integer [K|M] UNLIMITED} ON tablespace]
[PASSWORD EXPIRE]
[ACCOUNT {LOCK | UNLOCK}]
[PROFILE {profile | DEFAULT}]
```

ACCOUNT

- Permet de définir un compte utilisateur comme verrouillé ou déverrouillé.

PROFILE

- Permet de contrôler l'utilisation des ressources et de définir le mécanisme de contrôle par mot de passe à appliquer à l'utilisateur

Gestion des Comptes Utilisateur

Créer un utilisateur – Authentification par la base de données

L'utilisateur est authentifié avec le mot de passe stocké dans la base de données.

- La base de données doit être ouverte pour qu'un utilisateur puisse se connecter
- Mode par défaut

```
SQL > CREATE USER user1
IDENTIFIED BY pass1
DEFAULT TABLESPACE data_user
TEMPORARY TABLESPACE temp_user
QUOTA 15M ON data_user
ACCOUNT LOCK
PASSWORD EXPIRE;
```

```
> SQLPLUS user1/pass1
SQL >
```

Gestion des Comptes Utilisateur

Créer un utilisateur – Authentification par le système d'exploitation

- Ce mode permet à Oracle de se baser sur l'authentification de l'utilisateur par un système tierce ou par le système d'exploitation
- Avantage majeur : L'utilisateur n'a besoin de s'authentifier qu'une seule fois sur son système d'exploitation.
- Inconvénient : Faille de sécurité si l'utilisateur oublie de se déconnecter de sa machine.

```
SQL > CREATE USER ops$user1
IDENTIFIED BY EXTERNALLY
DEFAULT TABLESPACE data_user
TEMPORARY TABLESPACE temp_user;
```

```
> SQLPLUS /
SQL >
```

Gestion des Comptes Utilisateur

Créer un utilisateur – Authentification par le système d'exploitation

Le paramètre **REMOTE_OS_AUTHENT** indique si l'utilisateur peut être authentifié par un système d'exploitation distant.

SQL > show parameter REMOTE_OS_AUTHENT

```
SQL> show parameter remote_os_authent

NAME                                 TYPE        VALUE
------------------------------------ ----------- ----------------
remote_os_authent                    boolean     TRUE
```

Le paramètre **OS_AUTHENT_PREFIX** indique le format des noms utilisateur pour l'authentification par le système d'exploitation.

SQL > show parameter OS_AUTHENT_PREFIX

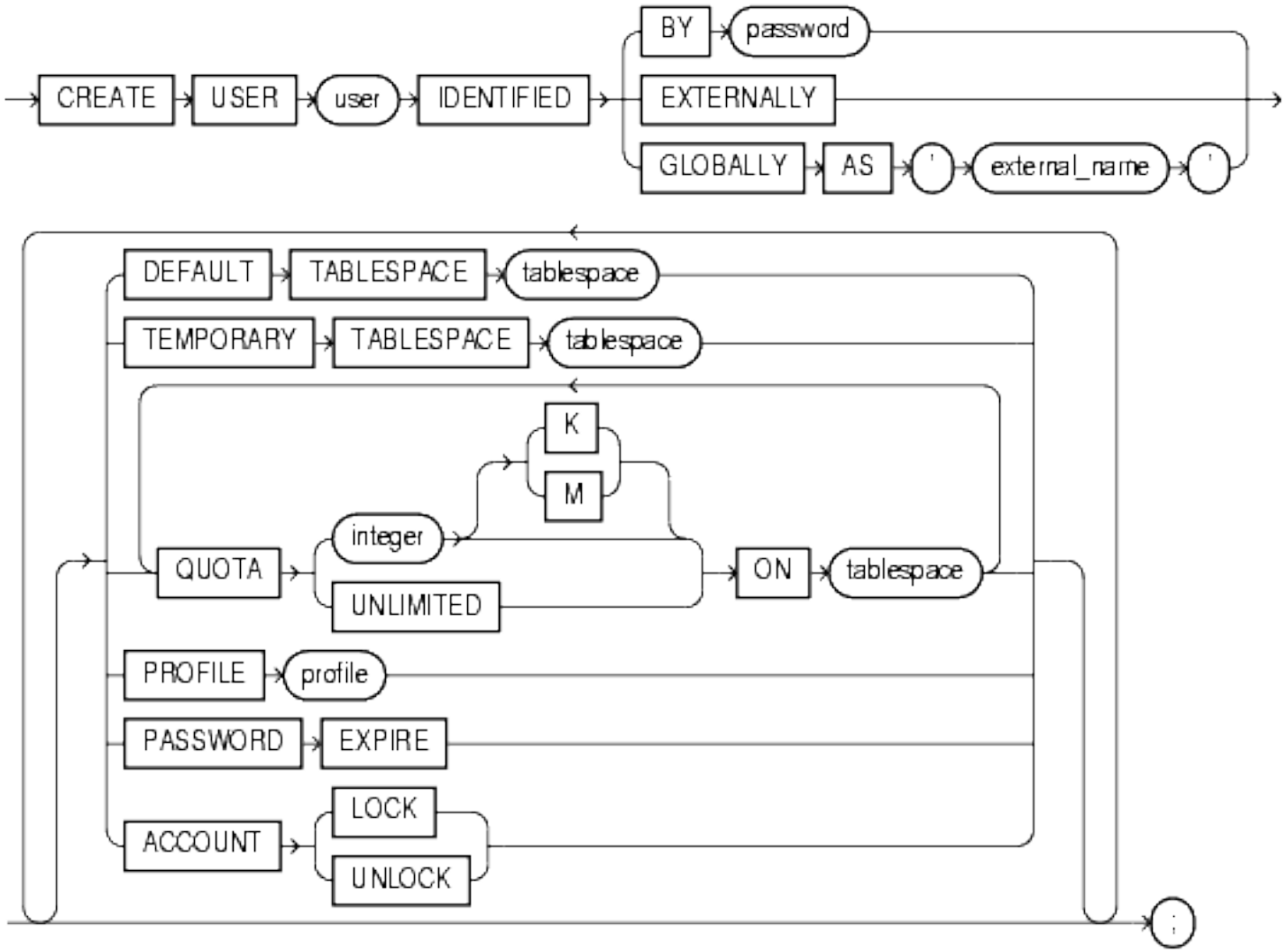
```
SQL> show parameter os_authent_prefix

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

Valeur de OS_AUTHENT_PREFIX	Login de l'utilisateur	Connexion distante possible
OS_	OS_USER1	Non
Pas de préfixe	USER1	Non
OPS$ (valeur par défaut) ou vide	OPS$USER1	Oui

Gestion des Comptes Utilisateur

Créer un utilisateur – Structure complète



Gestion des Comptes Utilisateur

Modifier un utilisateur – mot de passe

- Modification du mot de passe

```
SQL> ALTER USER user1 IDENTIFIED BY pass2 PASSWORD EXPIRE;
```

- Modification d'un ancien mot de passe avec un nouveau dans le cas où on utilise la fonction de **vérification de mots de passe** fournie par Oracle (script UTLPWDMG.SQL) ou si on spécifie cette fonction dans le paramètre **PASSWORD_VERIFY_FUNCTION** d'un profil assigné à l'utilisateur.

```
SQL> ALTER USER user1 IDENTIFIED BY new_password REPLACE old_password;
```

Gestion des Comptes Utilisateur

Modifier un utilisateur – tablespaces

- Modification d'un **tablespace par défaut**

```
SQL> ALTER USER user1 DEFAULT TABLESPACE data_user;
```

- Modification d'un **tablespace temporaire**

```
SQL> ALTER USER user1 TEMPORARY TABLESPACE temp_user;
```

Gestion des Comptes Utilisateur

Modifier un utilisateur – statut

- **Verrouillage** d'un compte

```
SQL> ALTER USER user1 ACCOUNT LOCK;
```

- **Activation** d'un compte

```
SQL> ALTER USER user1 ACCOUNT UNLOCK;
```

- Pour des raisons de sécurité, il peut parfois être utile de verrouiller le compte d'un utilisateur pour éviter que celui-ci soit utilisé de manière frauduleuse.
- Cette méthode peut aussi être utilisée pour organiser des objets dans un schéma sans que l'on puisse utiliser l'utilisateur en question.

Gestion des Comptes Utilisateur

Modifier un utilisateur - Quota

Le quota d'un tablespace d'un utilisateur est modifié dans les cas suivants :

- Lorsque la taille des tables appartenant à l'utilisateur augmente de manière imprévue
- Lorsqu'une application est étendue et nécessite des tables ou des index supplémentaires
- Lorsque les objets sont réorganisés et placés dans des tablespaces différents

- **QUOTA 0** : Les objets de l'utilisateur sont conservés dans le tablespace révoqué, mais aucun nouvel espace ne peut leur être alloué.

```
SQL> ALTER USER user1 QUOTA 0 ON data_user;
```

- **QUOTA 10M**

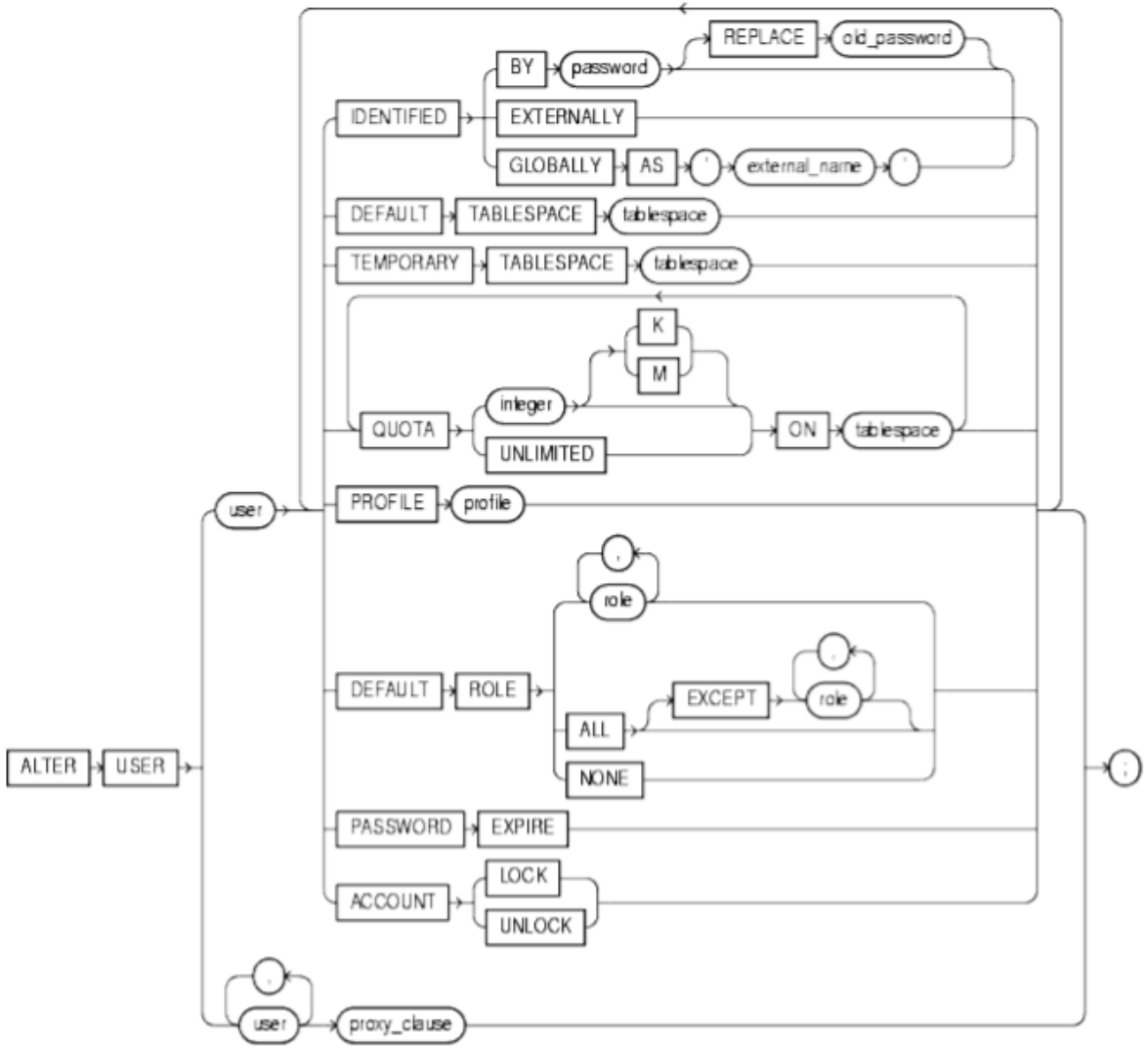
```
SQL> ALTER USER user1 QUOTA 10M ON data_user;
```

- **QUOTA UNLIMITED**

```
SQL> ALTER USER user1 QUOTA UNLIMITED ON data_user;
```

Gestion des Comptes Utilisateur

Modifier un utilisateur – Structure complète



Gestion des Comptes Utilisateur

Supprimer un utilisateur

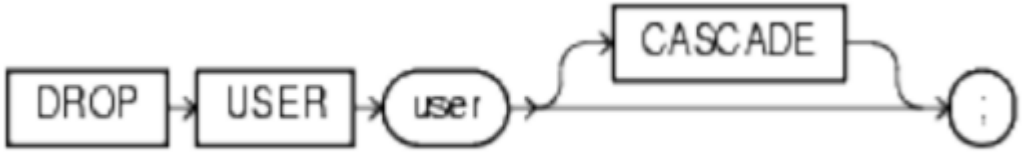
- Supprimer un utilisateur avec un **schéma vide** :

```
SQL > DROP USER user1;
```

- Supprimer un utilisateur avec **son schéma** :

```
SQL > DROP USER user1 CASCADE;
```

- L'option CASCADE supprime tous les objets du schéma avant de supprimer l'utilisateur.
- Oracle ne supprimera pas les rôles créés par l'utilisateur.



Gestion des Comptes Utilisateur

Récupérer les informations sur les utilisateurs

La vue **DBA_USERS** permet d'obtenir des informations sur tous les utilisateurs :

```
SQL> select username, password, account_status, default_tablespace, temporary_tablespace, profile from dba_users;

USERNAME                       PASSWORD   ACCOUNT_STATUS       DEFAULT_TABLESPACE   TEMPORARY_TABLESPACE PROFILE
------------------------------ ---------- -------------------- -------------------- -------------------- ----------
HR                                        OPEN                 USERS                TEMP                 DEFAULT
SCOTT                                     EXPIRED & LOCKED     USERS                TEMP                 DEFAULT
ORACLE_OCM                                EXPIRED & LOCKED     USERS                TEMP                 DEFAULT
XS$NULL                                   EXPIRED & LOCKED     USERS                TEMP                 DEFAULT
BI                                        EXPIRED & LOCKED     USERS                TEMP                 DEFAULT
PM                                        EXPIRED & LOCKED     USERS                TEMP                 DEFAULT
MDDATA                                    EXPIRED & LOCKED     USERS                TEMP                 DEFAULT
IX                                        EXPIRED & LOCKED     USERS                TEMP                 DEFAULT
SH                                        EXPIRED & LOCKED     USERS                TEMP                 DEFAULT
DIP                                       EXPIRED & LOCKED     USERS                TEMP                 DEFAULT
OE                                        EXPIRED & LOCKED     USERS                TEMP                 DEFAULT
APEX_PUBLIC_USER                          EXPIRED & LOCKED     USERS                TEMP                 DEFAULT
SPATIAL_CSW_ADMIN_USR                     EXPIRED & LOCKED     USERS                TEMP                 DEFAULT
```

Gestion des Comptes Utilisateur

Récupérer les informations sur les quotas

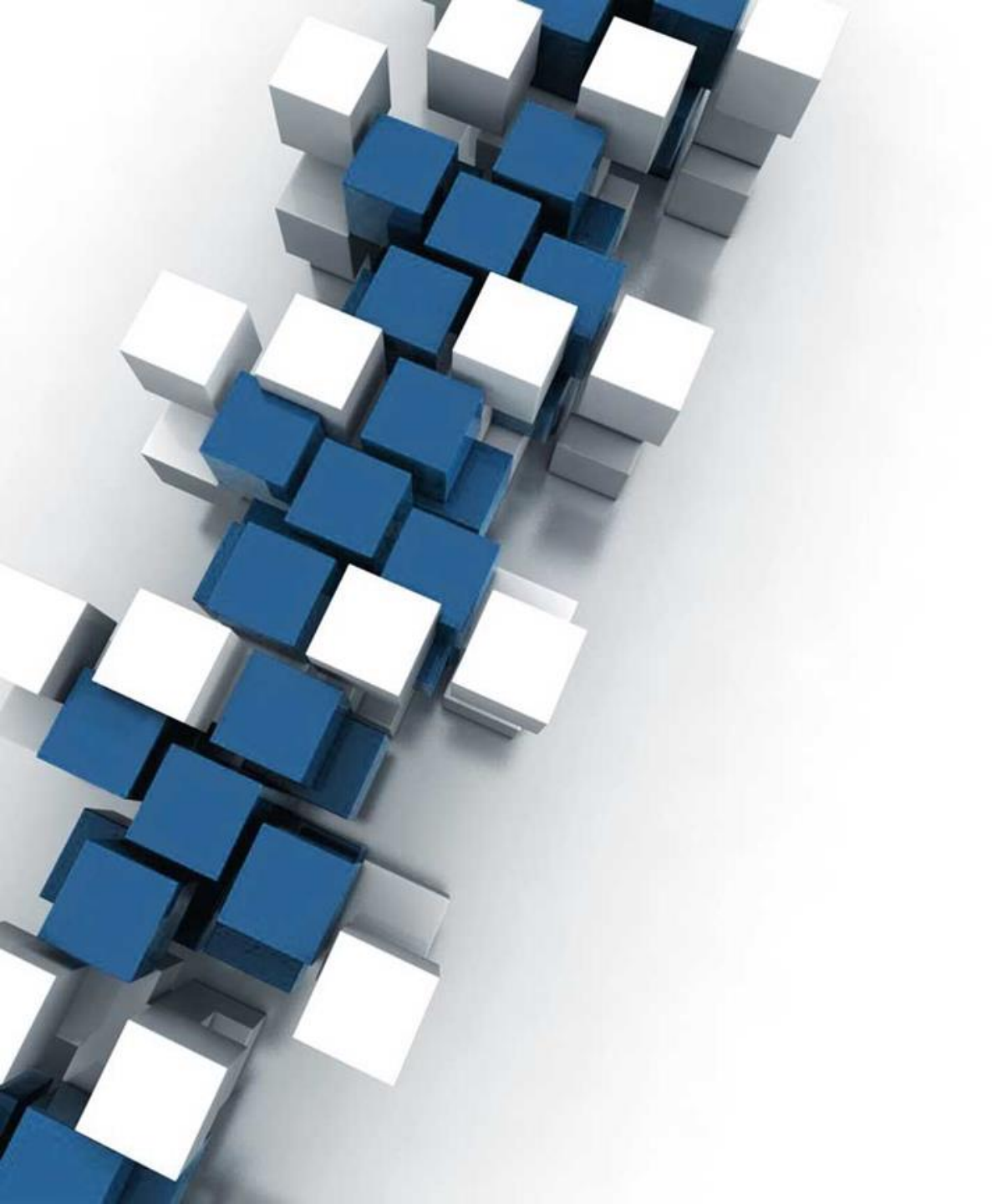
- La vue **DBA_TS_QUOTAS** permet d'obtenir des informations sur les quotas des tablespaces des utilisateurs :

```
SQL > select * from DBA_TS_QUOTAS;
```

```
SQL> select * from dba_ts_quotas;

TABLESPACE_NAME                USERNAME                       BYTES  MAX_BYTES     BLOCKS MAX_BLOCKS DRO
------------------------------ ---------------------- ---------- ---------- ---------- ---------- ---
SYSAUX                         SYSMAN                  73400320         -1       8960         -1 NO
SYSAUX                         OLAPSYS                  9240576         -1       1128         -1 NO
SYSAUX                         FLOWS_FILES                    0         -1          0         -1 NO
SYSAUX                         APPQOSSYS                      0         -1          0         -1 NO
USERDATA                       USER1                          0   20971520          0       2560 NO
```

- **MAX_BYTES** : Définit le quota d'un utilisateur.
  - S'il est égal à -1, alors il n'y a pas de limite sur le quota



Gestion des privilèges

Gestion des Privilèges

Définition

- Le privilège définit **le droit d'exécuter** un type particulier d'instruction SQL ou **d'accéder** à l'objet d'un autre utilisateur.
- La base de données Oracle permet de **contrôler les opérations** que les utilisateurs peuvent effectuer ou non au sein de la base.
- Il existe deux types de privilège utilisateur :
  - **Un privilège système** : Permet la création, modification, suppression, exécution de groupes d'objets.
    - Il existe plus de 170 privilèges système différents. Un grand nombre d'entre eux contiennent la clause ANY.
    - **Exemples** : Les privilèges CREATE TABLE, CREATE VIEW, CREATE SEQUENCE, etc.
  - **Un privilège objet** : Permet la manipulation des objets spécifiques (tables, vues, séquences, procédures, fonctions, packages)
    - **Exemples** : Les privilèges SELECT, INSERT, UPDATE, DELETE sur la table SCOTT.EMP.

Gestion des Privilèges

Accorder les privilèges – Privilèges Système

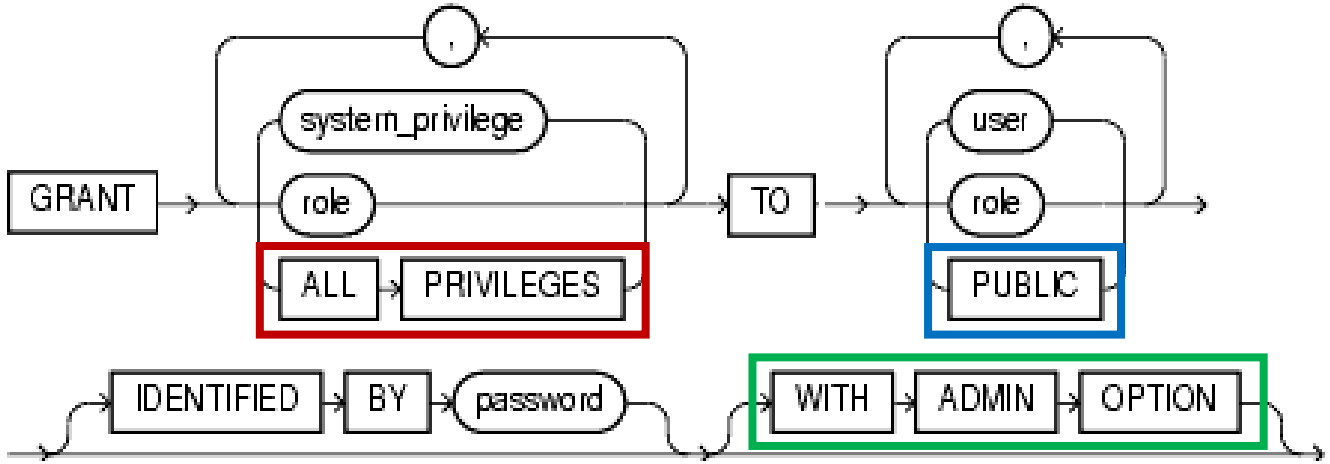
- Lorsqu'un utilisateur est créé avec l'instruction CREATE USER, il ne dispose encore d'aucun droit car aucun privilège ne lui a encore été assigné.
  - **Il ne peut même pas se connecter à la base !**
- Il faut donc lui assigner les privilèges nécessaires. Il doit pouvoir se connecter, créer des tables, des vues, des séquences.
- Pour lui assigner ces privilèges de niveau système, il faut utiliser l'instruction **GRANT**.

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

```
SQL > GRANT
CREATE SESSION ,
CREATE TABLE ,
CREATE VIEW
TO nom_user ;
```

Gestion des Privilèges

Accorder les privilèges – Privilèges Système



- **ALL PRIVILEGES** : représente tous les privilèges système (à l'exception de SELECT ANY DICTIONARY).
- **PUBLIC** : assigne le privilège à tous les utilisateurs.
- **WITH ADMIN OPTION** : assigne à l'utilisateur le droit d'assigner, de retirer, de modifier et de supprimer à son tour les privilèges du rôle reçus.

Gestion des Privilèges

Accorder les privilèges – Privilèges Objet

- Assigner à l'utilisateur le droit de sélectionner, insérer, modifier et supprimer des lignes dans la table EMP de l'utilisateur SCOTT.

```
SQL > GRANT
SELECT, INSERT, UPDATE, DELETE
ON SCOTT.EMP
TO user1 ;
```

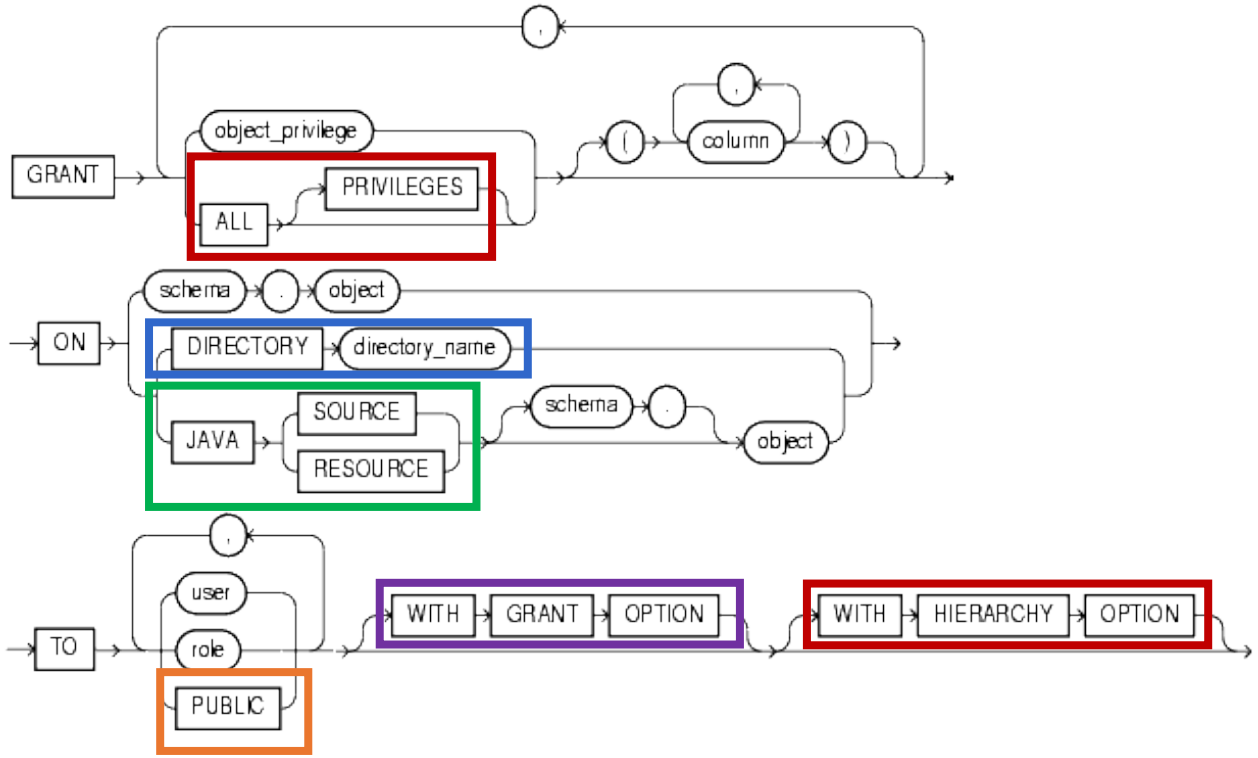
- Assigner à l'utilisateur le droit de modifier uniquement les colonnes JOB et MGR de la table SCOTT.EMP.

```
SQL > GRANT
UPDATE (JOB, MGR)
ON SCOTT.EMP
TO user1 ;
```

- Un utilisateur possède automatiquement tous les privilèges sur un objet qui lui appartient.
- Un utilisateur ne peut pas donner plus de privilèges qu'il n'en a reçu.
- S'il n'a pas reçu le privilège avec l'option WITH GRANT OPTION, un utilisateur ne peux pas assigner à son tour ce même privilège.

Gestion des Privilèges

Accorder les privilèges – Privilèges Objet



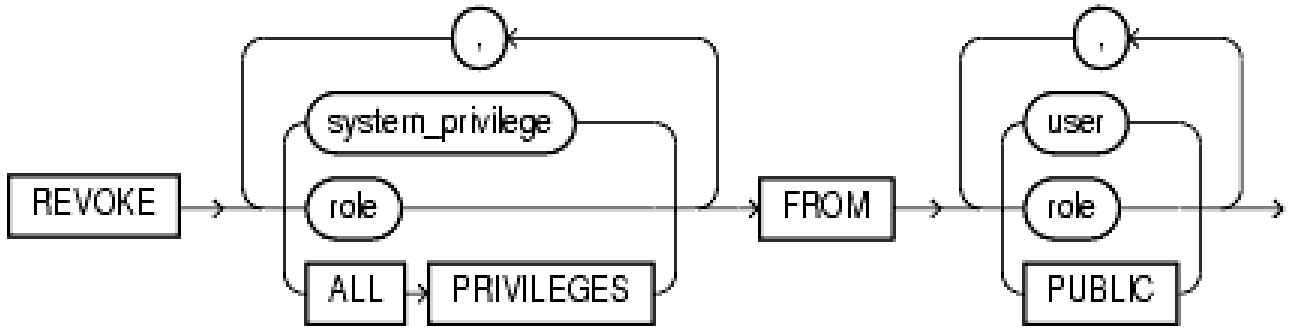
- **ALL PRIVILEGES** : représente tous les privilèges assignés à l'exécuteur de l'instruction.
- **DIRECTORY** : représente le nom d'un répertoire logique lié à un répertoire physique sur disque.
- **JAVA SOURCE/ RESOURCE** : représente le nom d'une source/ressource Java.
- **PUBLIC** : assigne le privilège à tous les utilisateurs.
- **WITH GRANT OPTION** : assigne à l'utilisateur le droit d'assigner à son tour le privilège reçu à un autre utilisateur (WITH GRANT OPTION s'applique à un utilisateur ou à PUBLIC, mais pas à un rôle).
- **WITH HIERARCHY OPTION** : assigne le privilège aux sous-objets.

Gestion des Privilèges

Retirer les privilèges – Privilèges Système

- Les privilèges système qui ont été octroyés directement à l'aide de la commande GRANT peuvent être révoqués via l'instruction SQL **REVOKE**.
- Les utilisateurs disposant de l'option **ADMIN OPTION** pour un privilège système peuvent révoquer ce privilège pour tout autre utilisateur de la base de données.
- La révocation peut être effectuée par un utilisateur différent de celui qui a initialement octroyé le privilège.
- Retirer des privilèges à un utilisateur ne supprime pas son schéma ni les objets qu'il contient.

```
SQL > REVOKE CREATE TABLE FROM user1;
```

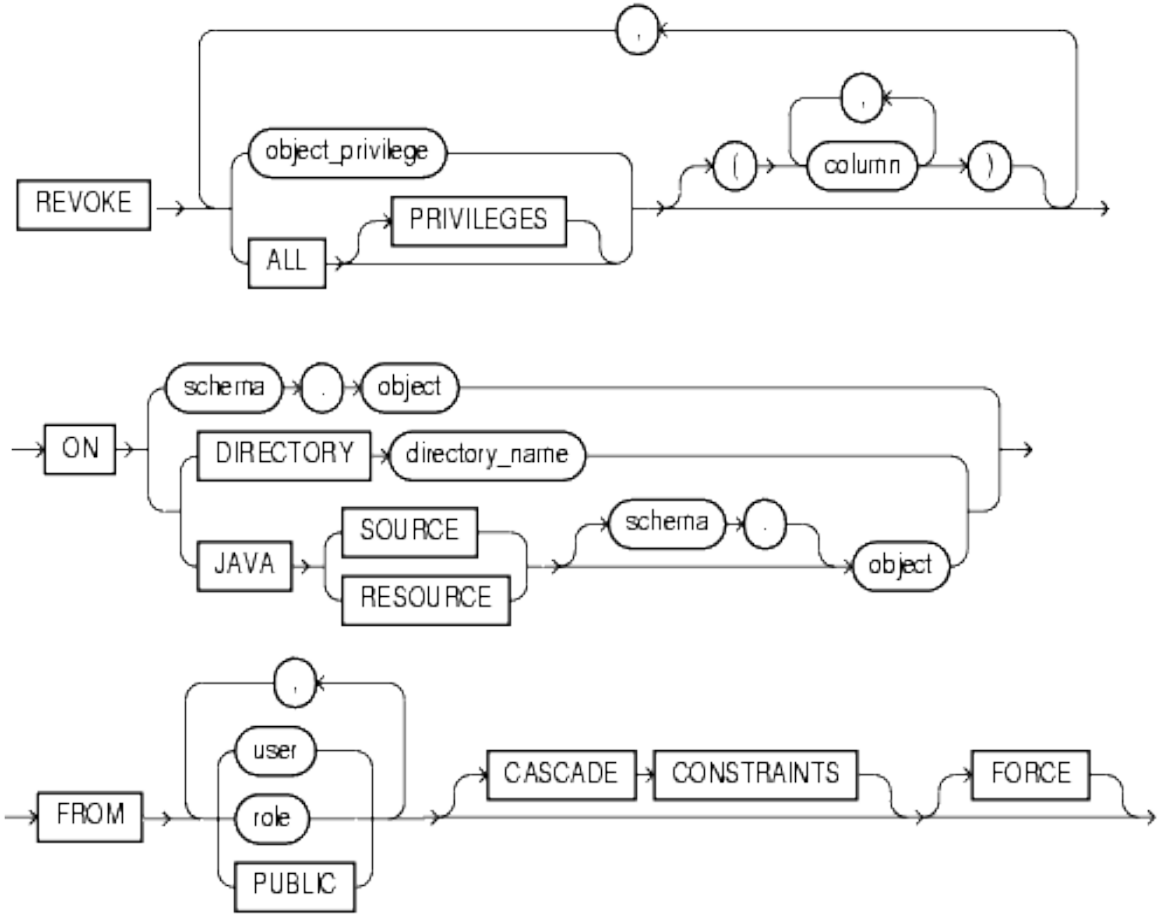


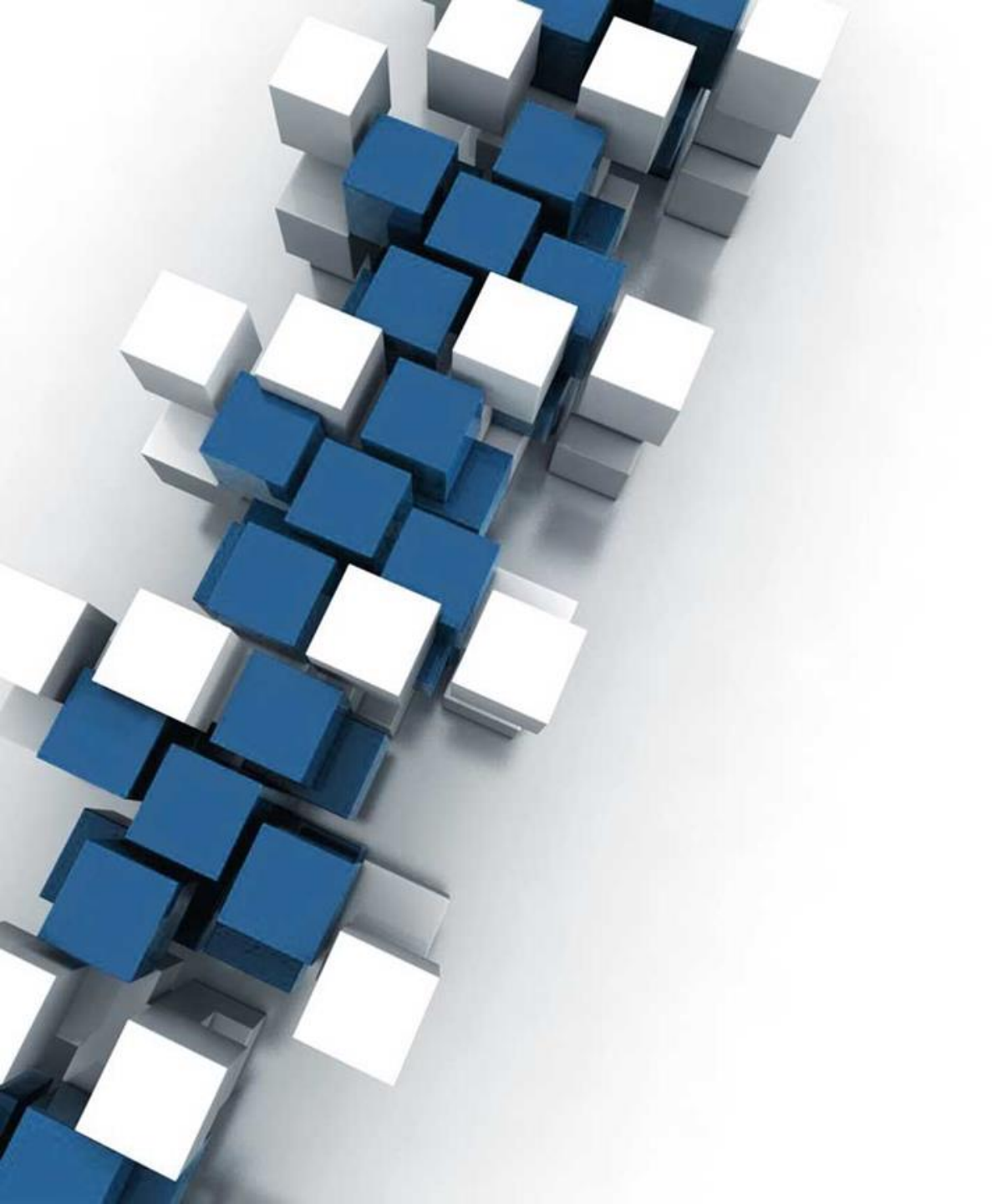
Gestion des Privilèges

Retirer les privilèges – Privilèges Objet

- Les privilèges objet qui ont été assignés à des utilisateurs ou à des rôles peuvent être retirés avec l'instruction **REVOKE**.
- La révocation de privilèges objet accordés avec l'option **GRANT OPTION** produit des effets en cascade.

SQL > **REVOKE** INSERT ON SCOTT.EMP **FROM** user1 ;





Gestion des Rôles

Gestion des Rôles

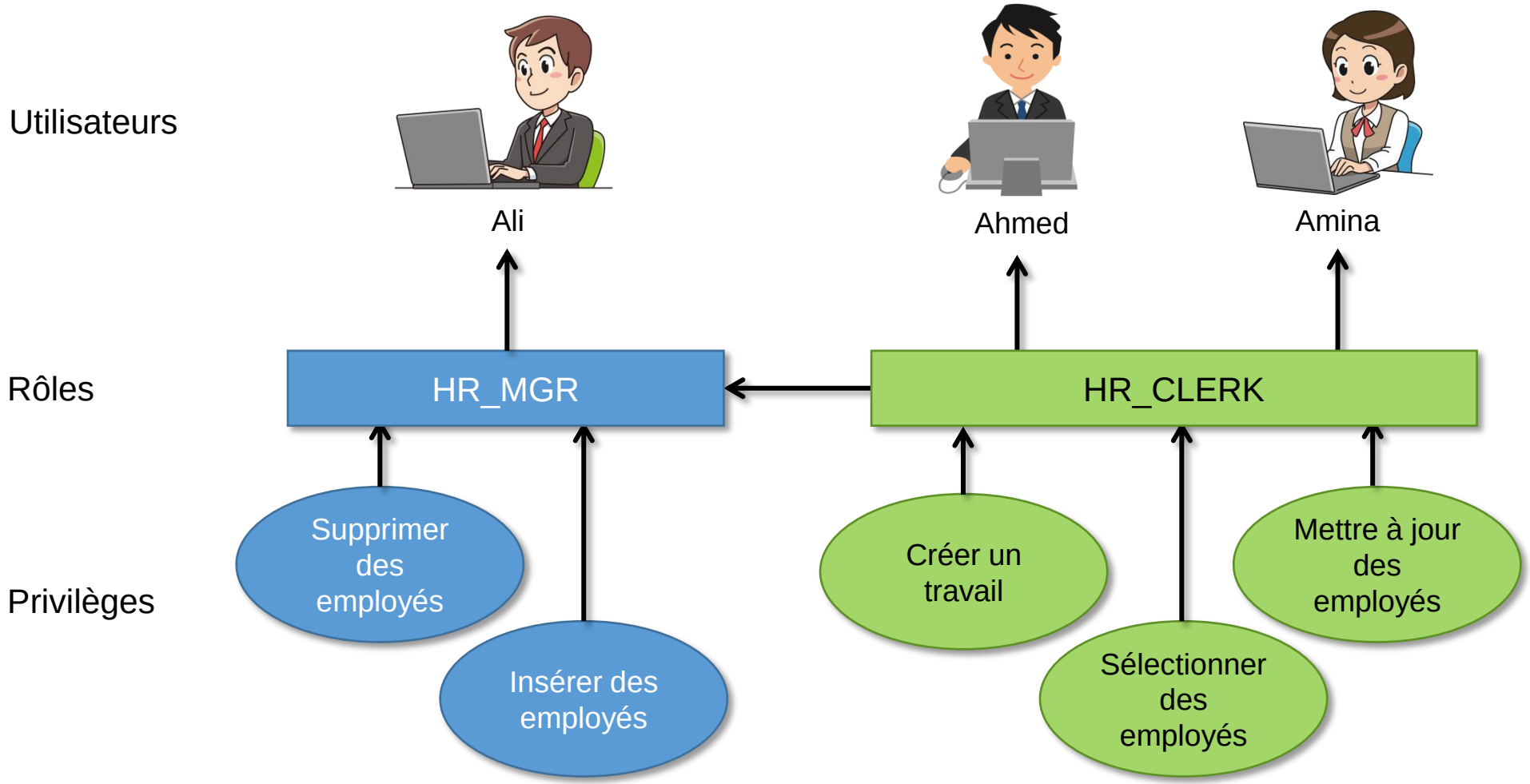
Définition d’un rôle

- Lorsque la liste des privilèges est importante, il est souhaitable de pouvoir **regrouper des privilèges** identiques dans un même ensemble, appelé **Rôle**.

- Les rôles sont des groupes de privilèges liés accordés aux utilisateurs ou à d'autres rôles. Ils permettent :
  - **La gestion simplifiée des privilèges :** Au lieu d'accorder le même ensemble de privilèges à plusieurs utilisateurs, vous pouvez accorder ces privilèges à un rôle, puis octroyer ce rôle à des utilisateurs individuels.
  - **La gestion dynamique des privilèges :** Si les privilèges associés à un rôle sont modifiés, tous les utilisateurs auxquels ce rôle est accordé bénéficient automatiquement et immédiatement des privilèges modifiés.
  - **La disponibilité sélective des privilèges :** Les rôles peuvent être activés et désactivés afin d'activer ou de désactiver temporairement les privilèges correspondants. Cela permet de contrôler les privilèges de l'utilisateur dans une situation donnée.

Gestion des Rôles

Définition d'un rôle



Gestion des Rôles

Caractéristiques d’un rôle

- Les privilèges sont accordés aux rôles (et révoqués) comme si le rôle était un utilisateur.
- Les rôles sont accordés aux utilisateurs ou à d'autres rôles (et révoqués de la même manière) comme s'il s'agissait de privilèges système.
- Un rôle peut être constitué de privilèges système et objet.
- Un rôle peut être activé ou désactivé pour chaque utilisateur auquel il est accordé.
- L'activation d'un rôle peut nécessiter un mot de passe.
- Les rôles n'appartiennent à personne et ne résident dans aucun schéma.

Gestion des Rôles

Créer et alimenter un rôle

- Créer un rôle

```
SQL > CREATE ROLE comptabilite ;
```

- Accorder des privilèges à un rôle

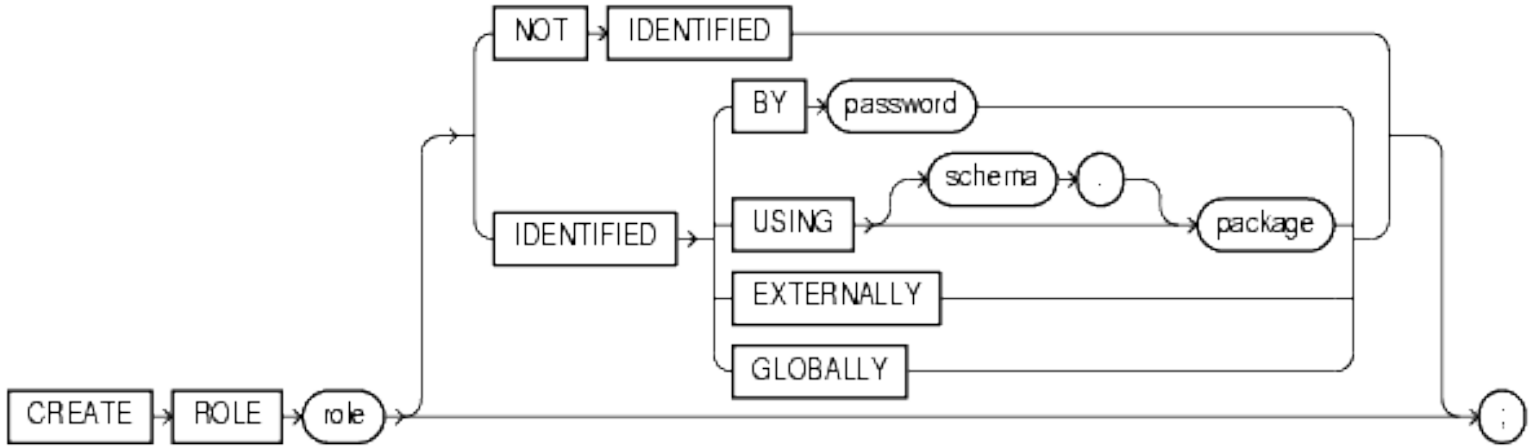
```
GRANT SELECT, INSERT, UPDATE, DELETE ON CPT.FACTURE TO comptabilite ;
GRANT SELECT, INSERT, UPDATE, DELETE ON CPT.LIG_FAC TO comptabilite ;
GRANT SELECT, INSERT, UPDATE, DELETE ON CPT.JOURNAL TO comptabilite ;
```

- Assigner un rôle à un utilisateur

```
SQL > GRANT comptabilite TO user1 ;
```

Gestion des Rôles

Créer et alimenter un rôle



- **NOT IDENTIFIED** (défaut) indique qu'aucun mot de passe n'est nécessaire pour activer le rôle.
- **IDENTIFIED BY password** indique qu'un mot de passe est nécessaire pour activer le rôle.
- **IDENTIFIED USING package** indique qu'un package va être utilisé pour fixer les droits de l'utilisateur.
- **IDENTIFIED EXTERNALLY** indique que l'autorisation provient d'une source externe (S.E.).
- **IDENTIFIED GLOBALLY** pour un user GLOBAL géré par exemple par Enterprise Directory Service.

Gestion des Rôles

Les rôles prédéfinis

Rôle	Privilèges
CONNECT	Ce rôle permet l'ouverture (CREATE SESSION) et la modification (ALTER SESSION) d'une session, la création de tables, vues, clusters, séquences, synonymes et liens de bases de données.
RESOURCE	Ce rôle permet de créer des types, tables clusters, opérateurs, séquences, index et procédures. Le rôle RESOURCE accorde un privilège UNLIMITED QUOTA à l'utilisateur est n'est donc à assigner qu'en connaissance de cause.
DBA	La liste des privilèges assignés au rôle DBA est beaucoup plus longue du fait que ce rôle est octroyé aux utilisateurs ayant des droits d'administration de la base. D'une façon générale, il est fortement déconseillé d'utiliser ces rôles standards car ils accordent trop de droits aux utilisateurs.
SCHEDULER_ADMIN	CREATE ANY JOB, CREATE EXTERNAL JOB, CREATE JOB, EXECUTE ANY CLASS, EXECUTE ANY PROGRAM, MANAGE SCHEDULER.
SELECT_CATALOG_ROLE	Aucun privilège système, mais le rôle HS_ADMIN_ROLE et plus de 1700 privilèges objet sur le dictionnaire de données.

Gestion des Rôles

Supprimer un rôle

- Le rôle spécifié ainsi que tous les privilèges qui lui sont associés sont supprimés de la base et également retirés à tous les utilisateurs qui en bénéficiaient.

```
SQL >  DROP ROLE nom_role ;
```

Gestion des Rôles

Récupérer les informations sur les rôles

- La liste des rôles définis est visible depuis la vue **DBA_ROLES**

```
SQL >  select * from DBA_ROLES;
```

```
SQL> select * from dba_roles;

ROLE                           PASSWORD AUTHENTICAT
------------------------------ -------- -----------
CONNECT                        NO       NONE
RESOURCE                       NO       NONE
DBA                            NO       NONE
SELECT_CATALOG_ROLE            NO       NONE
EXECUTE_CATALOG_ROLE           NO       NONE
DELETE_CATALOG_ROLE            NO       NONE
EXP_FULL_DATABASE              NO       NONE
IMP_FULL_DATABASE              NO       NONE
LOGSTDBY_ADMINISTRATOR         NO       NONE
DBFS_ROLE                      NO       NONE
AQ_ADMINISTRATOR_ROLE          NO       NONE
```

Gestion des Rôles

Récupérer les informations sur les rôles

- La liste des privilèges système assignés à un rôle s'obtient en interrogeant les vues **DBA_SYS_PRIVS** et **USER_SYS_PRIVS**

```
SQL >  select * from DBA_SYS_PRIVS where grantee= 'CONNECT';
```

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

GRANTEE                        PRIVILEGE                                ADM
------------------------------ ---------------------------------------- ---
CONNECT                        CREATE SESSION                           NO
```

Gestion des Rôles

Récupérer les informations sur les rôles

- La liste des rôles assignés à un utilisateur s'obtient via les vues **DBA_ROLE_PRIVS** et **USER_ROLE_PRIVS**

```
SQL >  select * from DBA_ROLE_PRIVS where grantee = 'HR';
```

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

GRANTEE                        GRANTED_ROLE                   ADM DEF
------------------------------ ------------------------------ --- ---
HR                             RESOURCE                       NO  YES
```

Gestion des Rôles

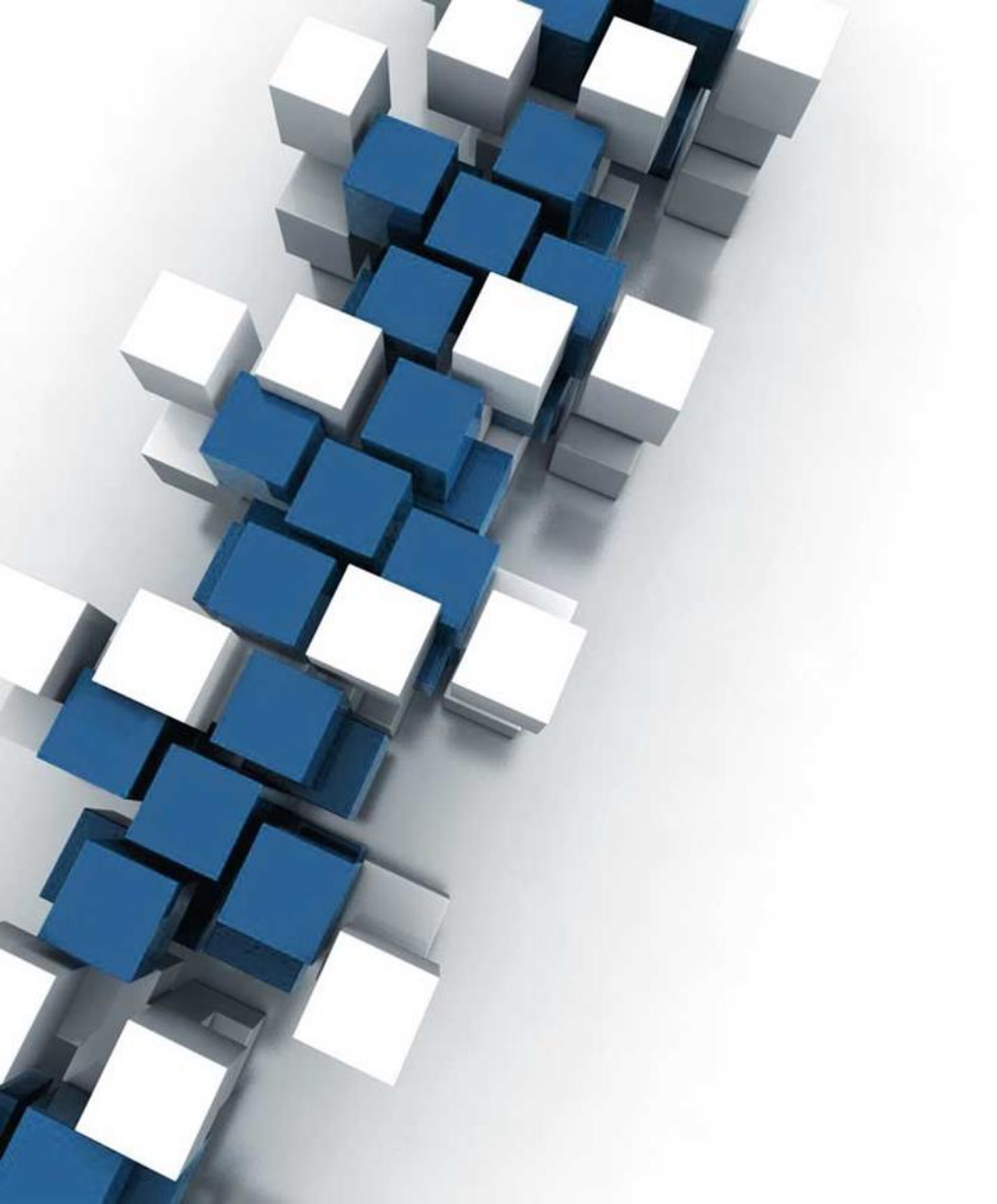
Récupérer les informations sur les rôles

- La liste des privilèges objet assignés à un utilisateur s'obtient en interrogeant les vues **DBA_TAB_PRIVS**, **ALL_TAB_PRIVS** et **USER_TAB_PRIVS**

```
SQL >  select * from DBA_TAB_PRIVS where grantee= 'HR';
```

```
SQL> select * from dba_tab_privs where grantee='HR';

GRANTEE    OWNER      TABLE_NAME GRANTOR    PRIVILEGE  GRA HIE
---------- ---------- ---------- ---------- ---------- --- ---
HR         SYS        DBMS_STATS SYS        EXECUTE    NO  NO
```



Gestion des Profils

Gestion des profils

Définition

- Les profils imposent un ensemble de **limites concernant les ressources système** allouées à un utilisateur afin d'éviter une surcharge inutile du serveur.

- Les profils permettent une **gestion des mots de passe** (comme le nombre maximal de tentatives de connexion à la base, le temps de verrouillage d'un compte, etc).

Gestion des profils

Caractéristiques

- Chaque utilisateur ne peut appartenir qu'à un **seul profil à la fois**.
- Si des utilisateurs sont connectés au moment de la modification de leur profil, le **changement** ne prend effet que lors de leur **prochaine connexion**.
- Les profils ne peuvent imposer des limitations de ressources aux utilisateurs que si le paramètre d'initialisation **RESOURCE_LIMIT=TRUE**. Sinon, les limitations de ressources liées aux profils sont ignorées.
- Il y a **deux types de limitations** :
  - Les limitations des mots de passe
  - Les limitations des ressources système

Gestion des profils

Les limitations des mots de passe (1/3)

Option	Description
PASSWORD_REUSE_TIME	Défini en nombre de **jours**, permet de définir **le délai entre deux utilisations du même mot de passe**. Par exemple, si celui-ci vaut 30 et que votre mot de passe actuel est toto. Il vous faudra attendre 30 jours à compter de la date de votre changement de mot de passe avant de pouvoir à nouveau utiliser toto comme mot de passe.
PASSWORD_REUSE_MAX	Permet de définir **le nombre de réutilisations du même mot de passe** (consécutive ou non).

Attention

- Si le paramètre PASSWORD_REUSE_TIME a été initialisé avec une valeur numérique, alors le paramètre PASSWORD_REUSE_MAX devra être à UNLIMITED et inversement.
- Si les deux paramètres possèdent la valeur UNLIMITED, alors Oracle n'utilisera aucune de ces deux limitations de mot de passe.
- Si le paramètre PASSWORD_REUSE_MAX possède la valeur DEFAULT et PASSWORD_REUSE_TIME est à UNLIMITED, alors Oracle utilisera PASSWORD_REUSE_MAX avec la valeur définie dans le profil par défaut.
- Si le paramètre PASSWORD_REUSE_TIME est à DEFAULT et PASSWORD_REUSE_MAX est à UNLIMITED, alors Oracle utilisera le paramètre PASSWORD_REUSE_TIME avec la valeur définie dans le profil DEFAULT.
- Si les deux paramètres sont à DEFAULT, alors Oracle utilisera les valeurs définies dans le profil DEFAULT.

Gestion des profils

Les limitations des mots de passe (2/3)

Option	Description
FAILED_LOGIN_ATTEMPTS	Permet de définir **le nombre maximal de tentatives** de connexion. Si le nombre de connexion donné par ce paramètre est atteint, le compte sera alors verrouillé pendant une période donnée par le paramètre PASSWORD_LOCK_TIME.
PASSWORD_LOCK_TIME	Permet de définir **la durée de verrouillage** du compte utilisateur après avoir bloqué le compte avec le paramètre FAILED_LOGIN_ATTEMPTS. Le compte sera alors automatiquement déverrouillé lorsque le temps défini par ce paramètre sera atteint. Il y a deux façons pour définir ce paramètre : ▪ Il peut être défini **en jours** (vous pouvez aussi spécifier un nombre de minutes ou heure, par exemple 30 minutes donnera 30/1440) ▪ Il peut avoir la valeur **UNLIMITED** pour un verrouillage définitif et donc une action d'un administrateur pour débloquer le compte.

Gestion des profils

Les limitations des mots de passe (3/3)

Option	Description
PASSWORD_LIFE_TIME	Permet de définir **la durée d'utilisation du même mot de passe**. Ce paramètre devra être défini **en jours**. Une fois la date limite d'utilisation arrive, Oracle demandera alors automatiquement à l'utilisateur de bien vouloir changer son mot de passe.
PASSWORD_GRACE_TIME	Permet de définir **en jours le temps de grâce** qui vous sera alloué **pour changer votre mot de passe**. Par exemple, vous avez défini le paramètre PASSWORD_LIFE_TIME (l'utilisateur devra changer son mot de passe tous les 30 jours). Alors, Oracle bloquera son compte automatiquement au bout de 3 demandes. L'intérêt de ce paramètre est d'ajouter une période de grâce pendant laquelle l'utilisateur sera en mesure de ne pas changer son mot de passe. Cela revient à donner un délai supplémentaire à l'utilisateur pour changer son mot de passe.
PASSWORD_VERIFY_FUNCTION	Ce paramètre contient le nom d'une **fonction PL/SQL** qui servira à **vérifier les mots de passe saisi**. Il est possible d'utiliser celle fournie par Oracle (script utlpwdmg.sql). La fonction fournie en argument devra avoir cette définition : (username varchar2, password varchar2, old_password varchar2) RETURN boolean. Si vous ne souhaitez pas utiliser de fonction de vérification utiliser la valeur NULL.

Gestion des profils

Les limitations des ressources système (1/2)

Option	Description
SESSIONS_PER_USER	Permet de définir **le nombre de session maximum** qu'un utilisateur pourra ouvrir.
CPU_PER_SESSION	Permet de définir **le temps de processeur maximum** en centièmes de secondes qu'une **session** pourra utiliser.
CPU_PER_CALL	Permet de définir **le temps de processeur maximum** en centièmes de secondes qu'un **"appel serveur**" pourra utiliser. On appelle "appel serveur" un passage de requête, une exécution de requête ou la récupération d'une requête (FETCH).
CONNECT_TIME	Permet de définir le temps en minutes pour **la durée de connexion maximale d'une session**. A la fin du temps imparti la session sera automatiquement déconnectée.
IDLE_TIME	Permet de définir le temps en minutes pour **la durée d'inactivité maximale d'une session**. A la fin du temps imparti la session sera automatiquement déconnectée.

Gestion des profils

Les limitations des ressources système (2/2)

Option	Description
LOGICAL_READS_PER_SESSION	Permet de définir **le nombre maximal de blocs** lus durant une **session**. On parle ici des blocs lus sur le disque et dans la mémoire.
LOGICAL_READS_PER_CALL	Permet de définir **le nombre maximal de blocs** lus durant un "**appel serveur**". On parle ici des blocs lus sur le disque et dans la mémoire.
COMPOSITE LIMIT	Permet de définir **le coût total des limitations autorisées** pour une **session**. Oracle calcule le coût total de toutes les ressources à partir du poids attribué aux paramètres CPU_PER_SESSION, CONNECT_TIME, LOGICAL_READS_PER_SESSION, et PRIVATE_SGA. Il est possible de changer le poids associé à chaque limitation système avec la commande ALTER RESOURCE COST.
PRIVATE_SGA	Permet de définir **la taille en Kbytes ou MBytes** que pourra utiliser une **session**.

Gestion des profils

Mettre en place un profil

- Pour créer un profil, il faut disposer du privilège système **CREATE PROFILE**.
- Les étapes à suivre sont :
  - Préciser les limitations de mot de passe et les limitations système
  - Créer le profil
  - Attribuer le profil aux utilisateurs qui devront être limités

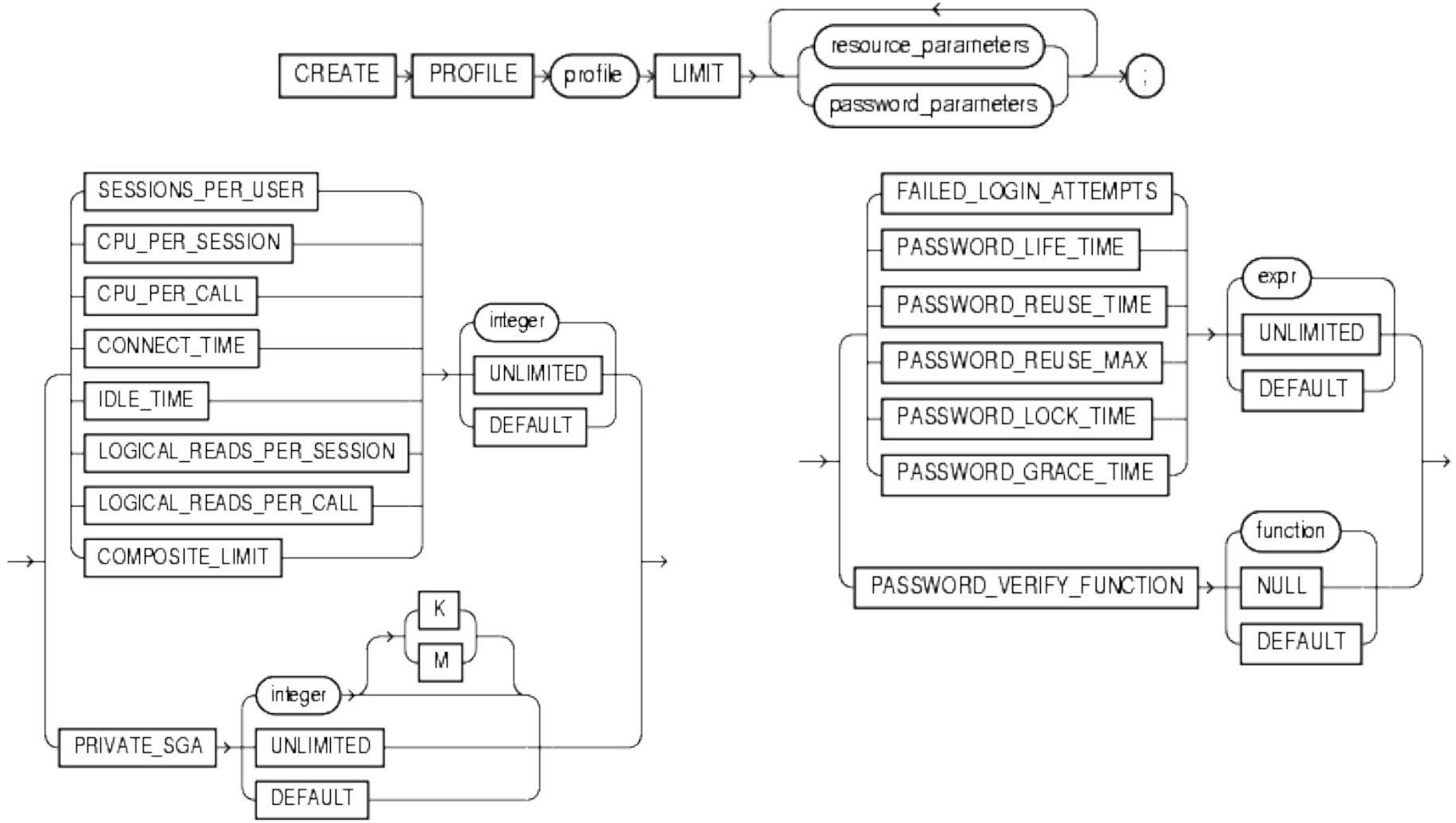
```
SQL > CREATE PROFILE profil_user
LIMIT
SESSIONS_PER_USER                UNLIMITED
CPU_PER_SESSION                  UNLIMITED
CPU_PER_CALL                     3000
CONNECT_TIME                     45
LOGICAL_READS_PER_SESSION        DEFAULT
LOGICAL_READS_PER_CALL           1000
PRIVATE_SGA                      15K
COMPOSITE_LIMIT                  5000000;
```

- Par défaut, un utilisateur se voit assigner le profil DEFAULT lors de sa création.
- Si vous souhaitez lui assigner un nouveau profil, cela sera possible soit lors de la création (voir page 10) soit avec la commande ALTER USER :

```
SQL > ALTER USER user1 PROFILE profil_user;
```

Gestion des profils

Mettre en place un profil – Syntaxe complète

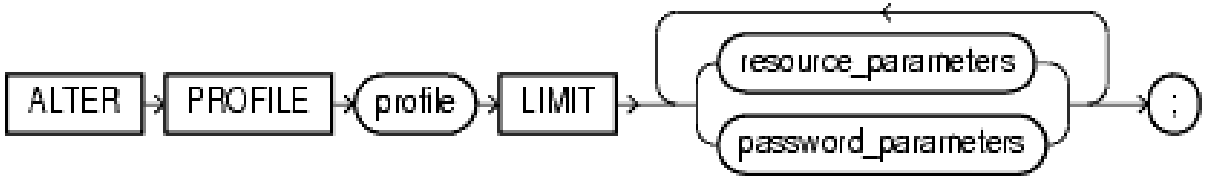


Gestion des profils

Modifier un profil

- Pour pouvoir modifier des limitations de ressources système, il faut disposer du privilège système **ALTER PROFILE**.
- Pour modifier des limitations de mot de passe, il faut disposer des privilèges **ALTER PROFILE** et **ALTER USER**.
- Si une limitation est modifiée, elle ne sera prise en compte que dans les nouvelles sessions.
- **Attention :** Vous ne pouvez pas retirer une limitation du profil DEFAULT, vous pourrez juste la faire passer à la valeur UNLIMITED.

```
SQL > ALTER PROFILE profil_user
LIMIT
FAILED_LOGIN_ATTEMPTS          5
PASSWORD_LOCK_TIME             1;
```



Gestion des profils

Supprimer un profil

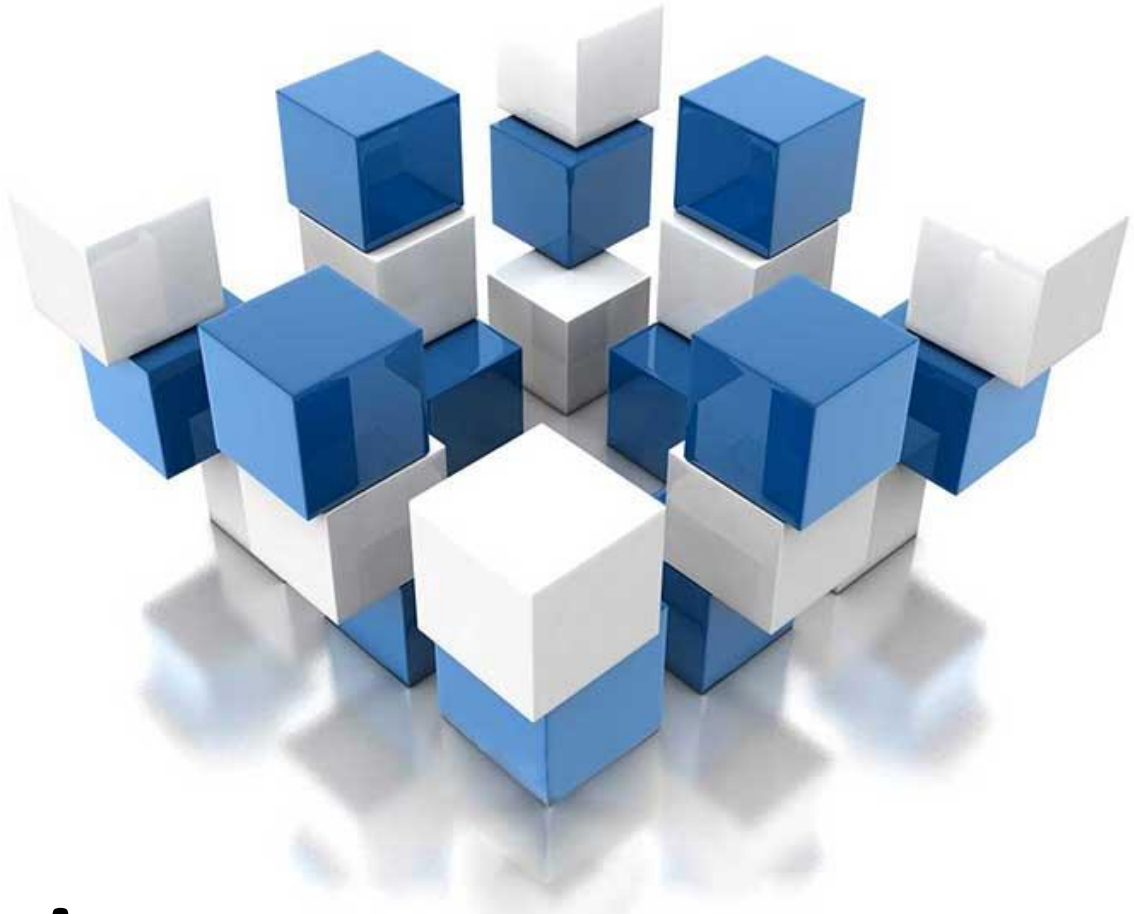
- Pour supprimer un profil, il existe deux cas de figure possibles :
  - Supprimer un profil qui n'a été **assigné à aucun utilisateur**

```
SQL > DROP PROFILE profil_user ;
```

  - Supprimer un profil qui a été **assigné à un utilisateur**

```
SQL > DROP PROFILE profil_user  CASCADE;
```

- CASCADE supprime le profil et assigne le profil DEFAULT à tous les utilisateurs qui possèdent le profil qui vient d'être supprimé



Merci pour votre attention