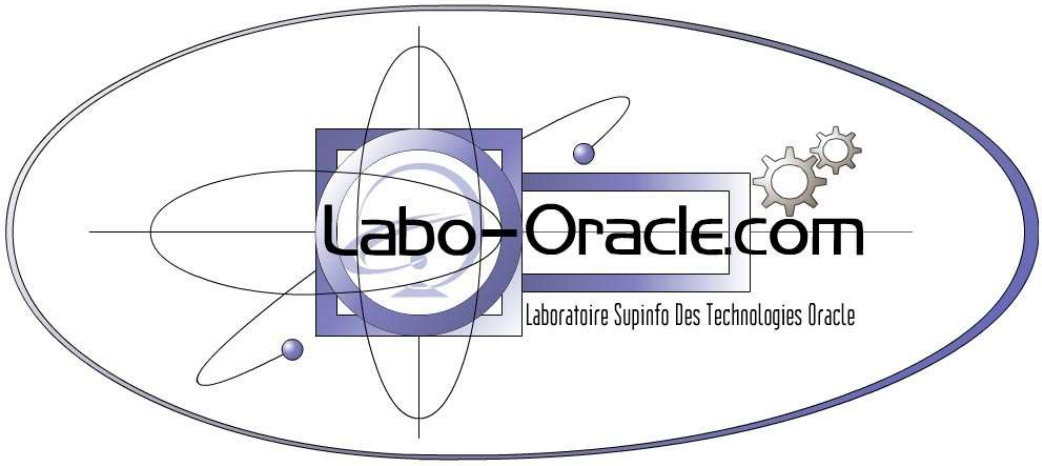


Module n°5

Gestion des utilisateurs

1Z0-031

Auteur : Davina Cité

Gestion des utilisateurs – d septembre yyyy

Nombre de pages : 40

Paris Academy of Computer Science
SUPINFO Paris
Ecole Supérieure d'Informatique

Ecole Supérieure d'Informatique de Paris
23. rue Château Landon 75010 – PARIS
www.supinfo.com

Table des matières

1.Gérer la sécurité des mot de passe et des ressources

1.1.Gestion des mot de passe et des profils

1.1.1.Les profils

Un profil est une configuration nommée des mots de passe et des limites de ressource suivants :

• Expiration de mot de passe
• Historique de mot de passe
• Vérification de la complexité du mot de passe
• Blocage du compte
• Temps de CPU
• Opération d'entrée /sortie
• Temps d'inactivité
• Temps de connexion
• Espace de mémoire (zone privée SQL uniquement pour MTS)
• Sessions concurrentes

Après la création du profil, l'administrateur peut l'attribuer à chaque utilisateur. Si les limites de ressource sont activées, le serveur Oracle limite l'usage et les ressources de la base de données au profil de l'utilisateur.

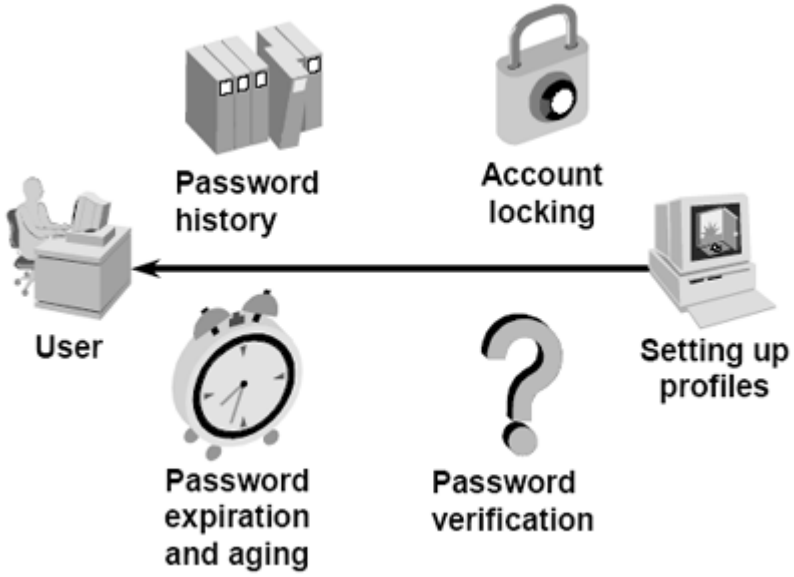
Le serveur Oracle crée automatiquement un profil par défaut quand la base de donnée est créée.
Les utilisateurs qui n'ont pas été explicitement assignés à un profil spécifique se conforment à toutes les limites du profil par défaut.
Le profil par défaut n'a pas de limites. Cependant, l'administrateur de base de données peut changer ses valeurs de sorte que des limites soient appliquées à tous les utilisateurs de ce profil.

L'utilisation du profil :

- Empêche les utilisateurs d'effectuer quelques opérations qui exigent de lourdes utilisations des ressources
- Assure que l'utilisateur se déconnecte de la base de données quand sa session est inactive pendant un certain temps
- Active les limites de ressource du groupe pour le même d'utilisateurs
- Assigne facilement les limites de ressource aux utilisateurs
- Gère l'utilisation de ressource dans des systèmes de base de données complexes à utilisateurs multiples
- Contrôle l'utilisation des mots de passe

Les attributions de profils n'affectent pas les sessions courantes. Un profil ne peut être assigné qu'à un utilisateur et non à des rôles ou à d'autre profils. Si vous n'assigné pas de profil à un utilisateur, le profil par défaut lui sera automatiquement attribué.

1.1.2.Gestion de mot de passe



Pour améliorer le contrôle de la sécurité de la base de données, la gestion de mot de passe d'Oracle est contrôlée par des administrateurs de base de données avec des profils.

Les dispositifs de gestion de mots de passe sont :

- Le blocage de compte : Active automatique le blocage d'un compte quand l'utilisateur échoue sa connexion un certain nombre de fois.
- L'expiration de mot de passe : détermine le temps d'utilisation du mot de passe avant qu'il expire.
- L'historique de mot de passe : vérifie si le nouveau mot de passe n'est pas réutilisé avant une période donnée.
- La vérification de la complexité du mot de passe : vérifie la complexité d'un mot de passe, contre les intrus qui pourraient essayer de pénétrer dans le système en devinant le mot de passe.

1.1.3.Mettre en place la gestion de mot de passe

Pour activer la gestion de mot de passe, il faut créer un profil pour limiter les configurations de mot de passe et attribuer le profil à l'utilisateur avec la commande CREATE USER ou ALTER USER.

Les configurations des limites du mot de passe dans les profils sont toujours imposées.

Lors de l'activation de la gestion de mot de passe, vous pouvez bloquer ou débloquer le compte d'un utilisateur avec la commande CREATE USER ou ALTER USER.

1.1.4.Verrouiller le mot de passe d'un compte

Le serveur Oracle bloque automatiquement un compte dès que la valeur de FAILED_LOGIN_ATTEMPTS est atteinte.
Le compte est automatique débloqué après le temps indiqué dans PASSWORD_LOCK_TIME ou doit être débloqué par l'administrateur avec la commande ALTER USER.

Le compte de base de données peut être bloqué explicitement avec la commande ALTER USER, mais ce compte ne pourra pas être automatique débloqué.

1.1.5.Expiration du mot de passe

Le paramètre PASSWORD_LIFE_TIME détermine le temps maximum avant le changement du mot de passe.
L'administrateur de la base de donnée peut spécifier une période de grâce avec PASSWORD_GRACE_TIME, qui commence dès la 1ère connexion à la base de données après que le mot de passe soit expiré. Un message d'alerte est généré à chaque fois que l'utilisateur essaie de se connecter pendant toute la période de grâce.
Si le mot de passe n'est toujours pas changé, le compte est bloqué et sont statut passe à EXPIRED par une configuration explicite du mot de passe.

1.1.6.Historique de mot de passe

L'historique de mot de passe assure que l'utilisateur ne peut pas réutiliser un mot de passe dans un intervalle de temps spécifié. Cette vérification peut être implémentée en utilisant un des paramètres suivants :

- PASSWORD_REUSE_TIME pour préciser que l'utilisateur ne peut pas réutiliser un mot de passe pour un nombre de jours donnés.
- PASSWORD_REUSE_MAX pour obliger l'utilisateur à définir, un certain nombre de fois, un mot de passe différent du premier.

Quand un paramètre est configuré à une valeur autre que DEFAULT ou UNLIMITED, les autres paramètres doivent être configuré à UNLIMITED.

1.1.7.Vérification du mot de passe

Avant d'assigner un nouveau mot de passe à l'utilisateur, la fonction PL/SQL PASSWORD_VERIFY_FUNCTION peut être appelée pour vérifier la validité du mot de passe.

Le serveur Oracle fournit une routine de vérification par défaut ou l'administrateur peut écrire une fonction PL/SQL.

1.2.Fonction du mot de passe

1.2.1.Fonction de mot de passe fournit par l'utilisateur

Pour ajouter une nouvelle fonction de vérification de mot de passe, l'administrateur de base de données doit considérer les règles suivantes :

- La fonction doit être créée dans le schéma SYS et doit avoir la syntaxe suivante:

```
function_name (   userid_parameter          IN VARCHAR2(30),
                  password_parameter        IN VARCHAR2(30),
                  old_password_parameter    IN VARCHAR2(30))
RETURN BOOLEAN
...
```

- La valeur de retour de la fonction est TRUE pour un succès et FALSE pour un échec.
- Si la fonction de mot de passe lève une exception, une erreur est retournée et la commande ALTER USER ou CREATE USER se termine.
- SYS doit être le propriétaire de la fonction de mot de passe
- Si la fonction de mot de passe est invalide, un message d'erreur est retourné et la commande ALTER USER ou CREATE USER se termine.

1.2.2.Fonction de vérification de mot de passe VERIFY_FUNCTION

Le serveur Oracle fournit une fonction de vérification complexe, dans le formulaire d'une fonction PL/SQL par défaut, appelé VERIFY_FUNCTION du script **utlpwdmg.sql**, qui doit s'exécuter dans le schéma SYS.

Pendant l'exécution du script utlpwdmg.sql, le serveur Oracle créé la fonction VERIFY_FUNCTION et change le profil par défaut avec la commande ALTER PROFIL suivante :

```
SQL> ALTER PROFIL DEFAULT LIMIT
  2  PASSWORD_LIFE_TIME 60
  3  PASSWORD_GRACE_TIME
  4  PASSWORD_REUSE_TIME 1800
  5  PASSWORD_REUSE_MAX UNLIMITED
  6  FAILED_LOGIN_ATTEMPTS 3
  7  PASSWORD_LOCK_TIME 1/1440
  8  PASSWORD_VERIFY_FUNCTION verify_function;
```

Avec VERIFY_FUNCTION, le mot de passe doit:
- avoir un minimum de 4 caractères
- ne doit pas être identique au login
- avoir au moins une lettre, un chiffre et un caractère spécial
- être différent de l'ancien mot de passe d'au moins 3 lettres

1.3.Gestion des profils

1.3.1.Créer un profil

On attribut un profil avec la commande CREATE USER ou ALTER USER. On ne peut attribué qu'un profil à la fois à un utilisateur.

Syntaxe:

```
CREATE PROFIL       profil LIMIT
[FAILED_LOGIN_ATTEMPTS    max_value]
[PASSWORD_LIFE_TIME       max_value]
[ {PASSWORD_REUSE_TIME
|PASSWORD_REUSE_MAX}      max_value]
[PASSWORD_LOCK_TIME       max_value]
[PASSWORD_GRACE_TIME      max_value]
```

```
 [PASSWORD_VERIFY_FUNCTION          {function|NULL|
DEFAULT} ]
```

Avec:

profil	C'est le nom du profil à créer
FAILED_LOGIN_ATTEMPS	Définit le nombre d'échec de connexion permit avant que le compte de l'utilisateur se bloque
PASSWORD_LIFE_TIME	Limite le nombre de jours où le même mot de passe peut être utilisé pour une authentification. Le mot de passe expire si il n'est pas changé pendant cette période.
PASSWORD_REUSE_TIME	Définit le nombre de jours avant qu'un mot de passe puisse être réutilisé. Si vous configurer PASSWORD_REUSE_TIME à une valeur de nombre entier, alors vous devez placer PASSWORD_REUSE_MAX à UNLIMITED.
PASSWORD_REUSE_MAX	Définit le nombre de changement de mot de passe demandé avant de pouvoir réutiliser le mot de passe actuel. Si vous configurer PASSWORD_REUSE_MAX à une valeur de nombre entier, alors vous devez placer PASSWORD_REUSE_TIME à UNLIMITED.
PASSWORD_LOCK_TIME	Définit le nombre de jours où un compte sera bloqué après un nombre de connexion consécutive échouée.
PASSWORD_GRACE_TIME	Définit le nombre de jours de la période de grâce pendant laquelle une alerte s'effectuera à chaque connexion. Si le mot de passe n'est pas changé pendant cette période, il expire.
PASSWORD_VERIFY_FUNCTION	Permet au script PL/SQL de vérification complexe du mot de passe d'être passé comme argument à l'ordre CREATE PROFIL.

Exemple:

```
CREATE PROFIL       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;
```

→ Création du profil grace_5 avec la possibilité de se tromper 3 fois de mot de passe sans qu'il ne soit débloqué avant. Son mot de passe expire sous 35 jours (temps de grâce compris) si il n'est pas modifié avant et il est vérifié par la fonction de sécurité fournit par Oracle.

Créer un profil avec Oracle Entreprise Manager

Lancez Security Manager à partir de la Console.

- Lancez la Console
  %oemapp console
  Choisissez Launch standalone

Vous pouvez aussi lancer la Console du menu démarrer de Windows NT

- Déroulez votre base de données du dossier Database
- Déroulez le dossier de Security et sélectionner le dossier Profils
- Sélectionnez Create du menu Object
- Sélectionnez Profil dans la liste et cliquez sur Create
- Entrez le nom du Profil et complétez les autres champs ou acceptez les valeurs par défaut
- Sélectionnez le champ de mot de passe et entrez les paramètres de mot de passe de compte
- Cliquez sur Create.

1.3.2.Modifier un profil

La commande ALTER PROFIL est utilisée pour changer les limites de mot de passe attribuées à un profil.

Syntaxe :

```
ALTER PROFIL profil LIMIT
                [FAILED_LOGIN_ATTEMPTS     max_value]
          [PASSWORD_LIFE_TIME        max_value]
          [ {PASSWORD_REUSE_TIME
            |PASSWORD_REUSE_MAX}     max_value]
          [PASSWORD_LOCK_TIME        max_value]
          [PASSWORD_GRACE_TIME       max_value]
          [PASSWORD_VERIFY_FUNCTION
                {function|NULL|DEFAULT} ]
```

Si vous voulez configurer les paramètres de mot de passe à moins d'un jour :
1 heure => PASSWORD_LOCK_TIME = 1/24
10 minutes => PASSWORD_LOCK_TIME = 10/1400
5 minutes => PASSWORD_LOCK_TIME = 5/1440

Les changements n'affectent pas les sessions courantes, ils ne sont employés que sur les prochaines sessions.

1.3.3.Supprimer un profil

Supprimer un profil en utilisant la commande DROP PROFIL :

```
DROP PROFIL profil [CASCADE];
```

profil : est le nom du profil pour être supprimer
CASCADE : supprime le profil des utilisateurs à qui il est assigné (Le serveur Oracle attribut automatiquement le profil par défaut à des utilisateur. Spécifiez cette option pour supprimer un profil qui est actuellement attribué aux utilisateurs)

Le profil par défaut ne peut pas être supprimé.
Lors de la suppression d'un profil, ce changement ne s'applique qu'aux nouvelles sessions créées et non aux sessions actuelles.

1.4.Gestion des ressources

1.4.1.Gestion des ressources

Des limites de gestion de ressource peuvent être imposées au niveau de la session, au niveau de l'appel, ou aux deux niveaux.

L'activation des limites des ressources se fait avec le paramètre d'initialisation RESSOURCE_LIMIT et la commande ALTER SYSTEM.

Les étapes suivantes permettent de contrôler l'usage des ressources avec les profils :

1. Créez un profil avec la commande CREATE PROFILE pour définir les limites des ressources et du mot de passe.
2. Attribuez le profil avec la commande CREATE USER ou ALTER USER.
3. Imposez les limites de ressource avec la commande ALTER SYSTEM ou éditez le fichier de paramètre d'initialisation (et redémarrez l'instance).

Définir les limites de ressources n'est pas nécessaire pour l'activation de gestion de mot de passe Oracle.

1.4.2.Limiter l'utilisation des ressources

Activer ou désactiver l'application des limites des ressources en modifiant le paramètre d'initialisation RESSOURCE_LIMIT ou en utilisant la commande ALTER SYSTEM.

Le paramètre RESSOURCE_LIMIT :
- Se modifie dans le fichier d'initialisation.
- La valeur TRUE active l'application.
- La valeur FALSE désactive l'application (valeur par défaut).
- Utilisez ce paramètre quand la base de données est arrêtée.

La commande ALTER SYSTEM :
- La configuration faite avec la commande ALTER SYSTEM persiste jusqu'à la prochaine modification ou jusqu'à ce que la base de données soit arrêtée.
- L'utilisation de cette commande ne se fait pas quand la base de données est arrêtée.

1.4.3.Mise en place des limites de ressources

Les limites du profil peuvent être appliquées au niveau de la session, de l'appel ou des deux. Les limites au niveau de la session sont appliquées à chaque connexion.

Quand les limites de session sont dépassées :
- une erreur est retournée
  Par exemple : ORA-02391: exceeded simultaneous SESSION_PER_USER limit.
- le serveur Oracle déconnecte l'utilisateur

Ressource	Description
CPU_PER_SESSION	Temps de CPU total en centaine de secondes
SESSION_PER_USER	Nombre de sessions pour chaque utilisateur
CONNECT_TIME	Temps de connexion écoulé en minutes

IDLE_TIME	Période d'inactivité en minutes qui est calculée uniquement pour le processus serveur. Ne prend pas en compte l'application activée. N'est pas affectée par des requêtes d'exécution longues et d'autres opérations.
LOGICAL_READS_PER_SESSION	Nombre de blocs de données (lecture physique ou logique). C'est la limitation sur le nombre total de lectures mémoire ou disque. Ceci pourrait être fait pour s'assurer qu'aucun ordre intensif d'entrée/sortie ne peut jouer sur les performances.
PRIVATE_SGA	Espace privée dans la SGA en bits. S'applique seulement lors de l'exécution de l'architecture du serveur partagé et peut être définit par M ou K.

Les limites au niveau de l'appel sont imposées à chaque appel fait tout en exécutant un ordre SQL.
Quand une limite au niveau de l'appel est dépassée :

- l'exécution de l'ordre est arrêtée
- l'ordre est annulé
- Tous les ordres précédents demeurent intacts
- La session d'utilisateur reste connectée

Ressource	Description
CPU_PER_CALL	Temps de CPU par appel en centaine de secondes
LOGICAL_READS_PER_CALL	Nombre de blocs de données qui peuvent être lus par appel

1.4.4.Créer un profil et limiter les ressources

Syntaxe de la commande CREATE PROFIL :

```
CREATE PROFIL profil LIMIT
               [SESSIONS_PER_USER max_value]
               [CPU_PER_SESSION max_value]
               [CPU_PER_CALL max_value]
               [CONNECT_TIME max_value]
               [IDLE_TIME max_value]
               [LOGICAL_READS_PER_SESSION max_value]
               [LOGICAL_READS_PER_CALL max_value]
               [COMPOSITE_LIMIT max_value]
               [PRIVATE_SGA max_bytes]
```

Avec :

profil	Est le nom du profil
max_value	Est un nombre entier, UNLIMITED ou DEFAULT
max_bytes	Est un nombre entier optionnel suivit de K ou M, UNLIMITED ou DEFAULT
UNLIMITED	Indique que l'utilisateur du profil peut utiliser un nombre illimité de ressource
DEFAULT	Indique que le profil est sujet à de limites de ressources comme elles sont définit dans le profil par défaut.

COMPOSITE_LIMIT	Limite le coût total de ressource pour une session exprimer en unité de service.

Oracle calcule le coût des ressources avec la somme des paramètres suivants :
- CPU_PER_SESSION
- CONNECT_TIME
- LOGICAL_READS_PER_SESSION
- PRIVATE_SGA

La vue du dictionnaire de données RESOURCE_COST renseigne sur la limite de chaque ressource.

1.4.5.Gérer les ressources en utilisant Database Resource Manager

Le but de Database Resource Manager est de donner au serveur Oracle plus de contrôle sur les décisions de gestion de ressources, ainsi il évite des problèmes résultant de la gestion inefficace du système d'exploitation.

Database Resource Manager comporte différents éléments :

- Groupe de consommateurs de ressource (Resource consumer group): Des groupes d'utilisateurs ou de sessions groupés sur base de condition de traitement des ressources.

- Plan de ressource (Resource plan): Contient des directives qui indiquent comment les ressources sont allouées aux groupes de consommateurs de ressource.

- Méthode d'allocation de ressource (Resource allocation methode): La méthode ou police utilisée par Database Resource Manager lors de l'allocation de ressources particulières.

- Directive de plan de ressource (resource plan directive): Utilisé par les administrateurs pour associer des groupes de consommateurs de ressource avec des plans particuliers et pour allouer des ressources parmi des groupes de consommateurs de ressource.

Pour administrer Database Resource Manager, vous devez avoir le privilège système ADMINISTER_RESOURCE_MANAGER pour administrer le Database Resource Manager (DBMS_RESOURCE_MANAGER). Typiquement, les DBAs auront ce privilège avec l'option ADMIN entant qu'élément du rôle DBA.

1.4.6.Resource Plan Directives

Database Resource Manager fournit plusieurs moyens d'allouer des ressources :

- La méthode CPU (CPU Method) : permet de définir comment les ressources CPU sont à allouer aux groupes de consommateur.

- Le pool de session active en file d'attente (Active session pool with Queuing) : vous pouvez contrôler un nombre maximum de sessions actives avec un groupe de consommateur. Une session est mise en file d'attente quand le pool de sessions est plein. Si une session active se termine, la 1ère session dans la file est programmée pour une exécution. Une période d'arrêt peut aussi être définit comme un travail en file d'attente qui s'arrête avec une erreur.

- Le degré de limite de parallélisme (Degree of Parallelism Limit): spécifie une limite de degré parallèle pour toutes les opérations se trouvant dans un groupe de consommateur.

- La commutation automatique de groupes de consommation (Automatic consumer group switching) : permet de contrôler les ressources par définition de critères. Si un critère n'est pas vu, cela provoquera une commutation automatique de sessions à un autre groupe de consommateur.
  Ces critères sont :
  Switch group : le groupe commuté.
  Switch time : temps de commutation en seconde.
  Switch estimate : estimation du temps que l'opération prendra pour s'accomplir, pour savoir si l'on commute une opération avant qu'elle ne commence.

- Le temps maximum d'exécution estimé (Maximum estimate execution time) : estime le temps d'exécution pour une opération proactive. Un DBA peut définir le temps d'exécution maximum estimé sur n'importe quelle opération en configurant le paramètre MAX_ESTIMATED_EXEC_TIME. Si le temps d'estimation de l'opération est plus important que celui définit dans MAX_ESTIMATED_EXEC_TIME, l'opération ne débutera pas, donc il y aura une élimination des grands travaux qui utilisent trop de ressources.
- Le pool Undo : Un pool Undo pour chaque groupe de consommateur peut être définit pour contrôler le nombre total d'Undo qui peut être généré par un groupe de consommation. Quand un groupe de consommateur dépasse ses limites, l'ordre DML courant générant le redo se terminera. Le pool Undo est définit par le paramètre UNDO_POOL.

1.4.7. Obtenir des mot de passe et des informations de ressources limite

Les informations sur le statut des comptes peuvent être obtenues avec la vue du dictionnaire de données DBA_USERS.

Exemple:

```
SQL> SELECT    username, password, account_status,
  2   FROM     dba_users;

USERNAME PASSWORD           ACCOUNT_STATUS
-------- ---------------- --------------
SYS      8A8F025737A9097A  OPEN
SYSTEM   D4DF7931AB130E37  OPEN
OUTLN    4A3BA55E08595C81  OPEN
DBSNMP   E066D214D5421CCC  OPEN
HR       BB69FBB77CFA6B9A  OPEN
OE       957C7EF29CC223FC  LOCKED
```

Pour afficher les informations sur le profil de mot de passe, il faut interroger la vue DBA_PROFILS :

Exemple:

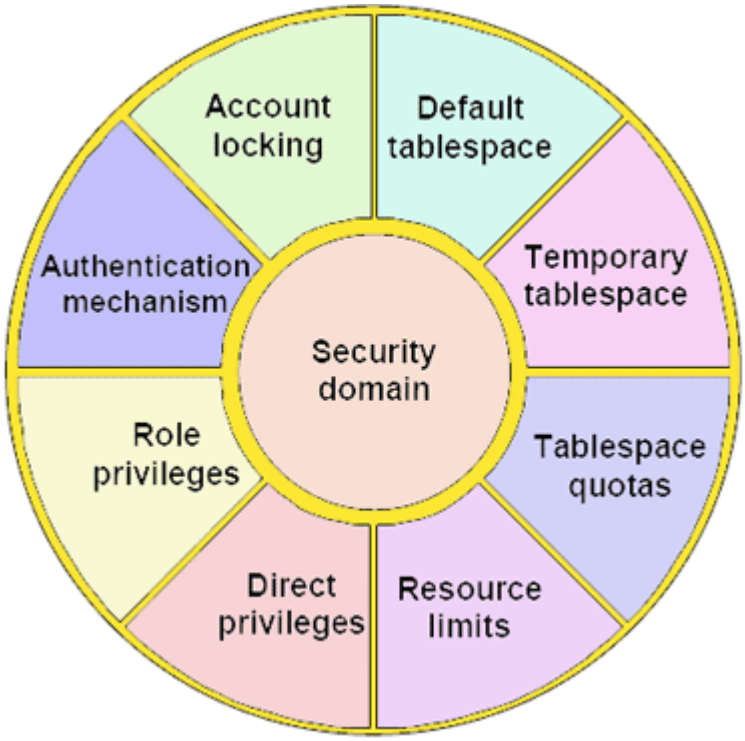
```
SQL> SELECT    *
  2   FROM     dba_profils
  3   WHERE    resource_type='PASSWORD'
  4  AND       profil='GRACE_5';

PROFIL RESOURCE_NAM                RESOURCE   LIMIT
--------------------               --------
    -------
```

```
 GRACE_5 FAILED_LOGIN_ATTEMPTS        PASSWORD   3
 GRACE_5 PASSWORD_LIFE_TIME           PASSWORD   30
 GRACE_5 PASSWORD_REUSE_TIME          PASSWORD   30
 GRACE_5 PASSWORD_REUSE_MAX           PASSWORD
UNLIMITED
 GRACE_5 PASSWORD_VERIFY_FUNCTION     PASSWORD
DEFAULT
 GRACE_5 PASSWORD_LOCK_TIME           PASSWORD
UNLIMITED
 GRACE_5 PASSWORD_GRACE_TIME          PASSWORD   5
```

2.Gestion des utilisateurs

2.1.Présentation de la gestion des utilisateurs et de la sécurité

2.1.1.Utilisateurs et sécurité



L'administrateur de base de données définit les logins des utilisateurs leur permettant d'accéder à la base. Un domaine de sécurité définit les configurations qui s'appliquent à l'utilisateur.

Avec le mécanisme d'authentification, un utilisateur qui a besoin de se connecter à la base de donnée peut être authentifié par :

- Le dictionnaire de données
- Le système d'exploitation
- Le réseau

Les moyens d'authentification sont spécifiés quand l'utilisateur est définit dans la base et ils peuvent être modifiés ultérieurement.

Les quotas de Tablespaces contrôle le nombre d'espace de stockage physique alloué à un utilisateur dans le tablespace de la base de données.

Le tablespace par défaut définit le lieu où sont stocké les segments créés par un utilisateur, si le tablespace n'est pas définit explicitement.

Le tablespace temporaire définit où les extents seront alloués par le serveur Oracle, si l'utilisateur réalise une opération avec d'une écriture de données sur le disque.

Des comptes peuvent être bloqués pour empêcher un utilisateur de se connecter à la base. Cela peut se faire automatiquent ou manuellement par l'administrateur.

Des limites peuvent être appliquées sur l'utilisation des ressources comme le temps de CPU, le nombre d'entrée/sortie et le nombre de session ouverte par utilisateur.

Les privilèges directs sont utilisés pour contrôler les actions d'un utilisateur sur la base de données.

Un utilisateur peut indirectement attribuer des privilèges aux utilisateurs avec les privilèges incluent dans les rôles.

2.1.2.Schéma dans la base de données

Un schéma est une collection nommée d'objets comme des table, vues, clusters, procédures et des packages associés à un utilisateur particulier.
Quand un utilisateur de base de données est créé, un schéma correspondant, avec le même nom, est créé pour cet utilisateur. Il ne peut avoir qu'un schéma par utilisateur, ainsi l'username et le schéma sont souvent interchangeables.

2.2.Créer et supprimer des utilisateurs

2.2.1.Créer un nouvel utilisateur : Authentification à la base de données

Syntaxe de création d'un nouvel utilisateur :

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

Avec:

user	Est le nom de l'utilisateur
BY password	Indique que l'utilisateur doit fournir un mot de passe pour se connecter à la base de données.
EXTERNALLY	Indique que l'authentification des utilisateurs se ferra par le système d'exploitation.
GLOBALLY AS	Indique que l'utilisateur doit s'authentifier globalement.
DEFAULT TEMPORARY TABLESPACE	Identifie le tablespace par défaut ou un tablespace temporaire à l'utilisateur si un tablespace temporaire n'a été attribué à aucun utilisateur.

QUOTA	Définit l'espace maximum permit pour les objets de l'utilisateur dans le tablespace *tablespace* (sa valeur peut être en un entier de bits, kilobits, mégabits ou UNLIMITED. Par défaut l'utilisateur n'a pas de quota sur les tablespaces)
PASSWORD EXPIRE	Force l'utilisateur à redéfinir sont mot de passe lors de sa connexion à la base en utilisant SQL Plus (cette option n'est valable que si l'utilisateur est authentifié par la base de données)
ACCOUNT LOCK/UNLOCK	Pour bloquer ou débloquer explicitement un compte d'utilisateur (UNLOCK est par défaut)
PROFIL	Est utilisé pour contrôler les ressources d'usage et pour définir le mécanisme de contrôle de mot de passe pour l'utilisateur

La méthode d'authentification par mot de passe est obligatoire. Si un mot de passe est spécifié, il est conservé par le serveur Oracle dans le dictionnaire de données. Les mécanismes de contrôle de mot de passe fournit par le serveur Oracle sont disponibles lorsque les utilisateurs s'authentifient sur le serveur.

Une fois que l'expiration de mot de passe est programmée, quand l'utilisateur se connecte sur SQL Plus, il reçoit le message suivant à l'ouverture et est invité à entrer un nouveau mot de passe :

```
ERROR:
ORA-28001: the account has expired
Changing password for PETER
Old password:
New password:
Retype new password:
Password changed
```

Créer un nouvel utilisateur avec Oracle Entreprise Manager

- Déroulez le dossier Security de votre base de données
- Sélectionnez le dossier User et sélectionnez Create dans le menu du clic droit
- Entrez les informations sur l'utilisateur dans la page principale de la fenêtre de propriétés
- Spécifier les quotas utilisés dans la page Quotas
- Cliquez sur Create

Vous pouvez aussi sélectionner un utilisateur et sélectionnez Object→Create Like dans la barre de menu pour créer un utilisateur avec les même quotas et privilèges que utilisateur existant dans la base.

Oracle Security Manager attribut automatiquement le rôle CONNECT à tous les utilisateurs qui sont créés à partir de cet outil.

2.2.2.Créer un nouvel utilisateur : Authentification par le système d'exploitation

Il faut utiliser la clause IDENTIFIED EXTERNALLY dans la commande CREATE USER pour spécifier que l'utilisateur doit s'authentifier par le système d'exploitation. Cette option est généralement utilisée quand l'utilisateur se connecte directement sur la machine où se trouve le serveur Oracle.

Exemple :

```
CREATE USER aaron
IDENTIFIED EXTERNALLY
DEFAULT TABLESPACE USERS
TEMPORARY TABLESPACE temp
QUOTA 15m ON data
PASSWORD EXPIRE;
```

→ Cette requête montre comment l'utilisateur aaron est définit dans la base. Ainsi l'utilisateur du système d'exploitation aaron pourra accéder à la base sans devoir passez une authentification du serveur Oracle.

Alors pour se connecter avec SQL Plus sous un système UNIX, l'utilisateur aaron doit rentrer la commande système suivante :

```
$ sqlplus/
```

Le paramètre d'initialisation OS_AUTHENT_PREFIX définit le format des logins pour une authentification du système d'exploitation. Cette valeur sera par défaut à OPS$ pour permettre une rétro compatibilité avec les version antérieur du serveur Oracle.
Pour configurer le préfixe à NULL, il faut initialiser le paramètre comme tel :
OS_AUTHENT_PREFIX = ""

OS_AUTHENT_PREFIX=OPS$ donne la flexibilité d'avoir un authentification par le système d'exploitation ou par le serveur Oracle. Dans ce cas le DBA peut créé l'utilisateur en entrant une commande :

```
CREATE USER ops$user
IDENTIFIED password...
```

Un utilisateur qui se connecte directement sur une machine avec un serveur Oracle n'a pas besoin de fournir un mot de passe. Si l'utilisateur se connecte avec une machine cliente distante, il devra fournir un mot de passe.

Le paramètre d'initialisation, REMOTE_OS_AUTHENT=TRUE indique qu'un utilisateur peut s'authentifier à distante. La valeur FALSE ne permet qu'une connexion en local sur le serveur. Attention à l'utilisation de ce paramètre qui peut provoquer des problèmes de sécurité.

2.2.3.Supprimer un utilisateur

La syntaxe de suppression d'un utilisateur est :

```
DROP USER user [CASCADE];
```

L'option **CASCADE** supprime tous les objets dans le schéma avant de supprimer l'utilisateur. Cela doit être spécifié si le schéma contient des objets.
Un utilisateur qui est connecté à la base ne peut pas être supprimé.

2.3.Surveillance des utilisateurs

2.3.1.Changer le quota d'un utilisateur pour un tablespace

Si une table d'un utilisateur montre une croissance imprévue, si une application à besoin d'un table ou d'un indexe supplémentaire ou si des objets sont réorganisés et placés dans différents tablespaces ; vous aurez besoin de modifier les quotas de tablespace avec la commande suivante :

```
ALTER USER user
[ DEFAULT TABLESPACE tablespace]
[ TEMPORARY TABLESPACE tablespace]
[ QUOTA {integer [K | M] | UNLIMITED } ON tablespace
[ QUOTA {integer [K | M] | UNLIMITED } ON
tablespace ]
...]
```

Une fois qu'un quota de 0 est assigné, les objets de l'utilisateur sont supprimés du tablespace et ne peuvent être alloués à de nouveaux espaces.

Par exemple, si une table de 10MB se trouve dans le tablespace USERS, et que vous exécutez cette commande :

```
ALTER USER aaron
QUOTA 0 ON USERS;
```

Aucun nouvel extent ne pourra être assigné à cette table.
Toutes les options inchangées demeurent sans changement.

2.3.2.Récupérer les informations utilisateurs

Pour afficher le tablespace par défaut pour tous les utilisateurs, on exécute la commande suivante:

```
 SQL> SELECT   username, default_tablespace
   2  FROM     dba_users;

USERNAME      DEFAULT_TABLESPACE
-----------   ------------------
SYS           SYSTEM
SYSTEM        SYSTEM
OUTLN         SYSTEM
DBSNMP        SYSTEM
HR            EXAMPLE
OE            EXAMPLE
```

3.Gestion des privileges

3.1.Présentation des privilèges

3.1.1.Gestion des privilèges

Un privilège est un droit d'exécuter un type particulier d'ordre SQL ou d'accéder à un objet d'un autre utilisateur. Cela inclut le droit de :

- se connecter à une base
- créer une table
- sélectionner les lignes d'un autre utilisateur
- exécuter la procédure stockée d'un autre utilisateur

Chaque privilège système permet à un utilisateur d'exécuter une opération de base de données particulière ou une classe d'opération de base de données. Par exemple, le privilège de créer un tablespace est un privilège système.

Chaque privilège objet permet à l'utilisateur d'exécuter une action particulière sur un objet spécifique, tel qu'une table, une vue, une séquence, une procédure, une fonction ou un package.

Le contrôle des privilèges d'un DBA permet de:

- Fournir à un utilisateur les droits d'exécuter un type d'opération.
- Attribuer ou retirer le droit d'exécuter des fonctions systèmes
- Donner des privilèges à un utilisateur ou à un rôle
- Attribuer des privilèges à tous les utilisateurs (PUBLIC)

3.2.Privilèges système

3.2.1.Privilèges système

Les privilèges peuvent être classés de la manière suivante :

- Des privilèges permettant de opérations système ; par exemple, CREATE SESSION, CREATE TABLESPACE.
- Des privilèges permettant la gestion des objets dans in le schéma d'utilisateur ; par exemple, CREATE TABLE.
- Des privilèges permettant la gestion des objets dans tous les schémas ; par exemple, CREATE ANY TABLE.

Les privilèges peuvent être contrôlés avec les commandes GRANT et REVOKE, qui ajoute et supprime des privilèges à un utilisateur ou à un rôle.

3.2.2.Exemples de privilèges

Catégorie	Exemple
INDEX	**CREATE ANY INDEX** **ALTER ANY INDEX** **DROP ANY INDEX**
TABLE	**CREATE TABLE** **CREATE ANY TABLE** **ALTER ANY TABLE** **DROP ANY TABLE** **SELECT ANY TABLE** **UPDATE ANY TABLE** **DELETE ANY TABLE**
SESSION	**CREATE SESSION** **ALTER SESSION** **RESTRICTED SESSION**
TABLESPACE	**CREATE TABLESPACE** **ALTER TABLESPACE** **DROP TABLESPACE** **UNLIMITED TABLESPACE**

Il n'y a pas de privilège CREATE INDEX.

CREATE TABLE inclut les commandes CREATE INDEX et ANALYZE. L'utilisateur doit avoir un quota pour le tablespace ou doit avoir le privilège UNLIMITED TABLESPACE.

UNLIMITED TABLESPACE ne peut pas être attribué à un rôle.

Pour supprimer une table d'un autre schéma, il faut avoir le privilège DROP ANY TABLE.

3.2.3.Accorder des privilèges système

On utilise l'ordre SQL, GRANT pour accorder des privilèges système aux utilisateurs.

L'option ADMIN permet d'attribuer des privilèges à d'autres utilisateurs. Ce privilège est habituellement réservé par sécurité à l'administrateur et rarement aux autres utilisateurs.

Syntaxe :

```
GRANT {system_privilege|role}
    [, {system_privilege|role} ]...
TO {user|role|PUBLIC}
    [, {user|role|PUBLIC} ]...
[WITH ADMIN OPTION]
```

system_privilege : spécifie le privilège système à accorder à un utilisateur ou un rôle.
PUBLIC : Attribut le système privilège à tous les utilisateurs
WITH ADMIN OPTION : permet à l'utilisateur d'attribuer à son tour des privilèges à d'autres utilisateurs ou rôles.

Accorder des privilèges systèmes avec Oracle Entreprise Manager

- Après un clic droit sur votre base de données, cliquez sur Connect.
- Indiquez le login, le mot de passe et le nom pour votre base de données et cliquez sur OK.
- Dérouler le dossier Security
- Déroulez le dossier Users et sélectionnez l'utilisateur de votre choix
- Cliquez sur System Privileges dans la partie détail de la console.
- Sélectionner le privilège système que vous voulez accorder. Eventuellement, vérifié la box Admin Option et cliquez sur Apply.

3.2.4.SYSDBA et SYSOPER privilèges

Seul l'administrateur de base de données devrait avoir la possibilité de se connecter à la base avec des privilèges d'administrateurs.
Se connecté comme un SYSDBA donne à l'utilisateur tous les privilèges pour exécuter toutes les opérations sur une base de données ou les objets dans la base.

Catégorie	exemple
SYSOPER	**ALTER DATABASE ARCHIVELOG** **RECOVER DATABASE** **ALTER DATABASE BACKUP CONTROLFILE TO** **ALTER DATABASE OPEN | MOUNT** **SHUTDOWN** **STARTUP**
SYSDBA	**SYSOPER PRIVILEGES WITH ADMIN OPTION** **CREATE DATABASE** **ALTER DATABASE BEGIN/END BACKUP** **RESTRICTED SESSION** **RECOVER DATABASE UNTIL**

3.2.5.Restrictions des privilèges systèmes

Le mécanisme de protection du dictionnaire dans Oracle9i empêche les utilisateurs non autorisé d'accéder aux objets du dictionnaire.

L'accès aux objets du dictionnaire est restreint au rôle SYSDBA et SYSOPER. Les privilèges de système, permettant d'accéder aux objets dans d'autres schémas, ne donnent pas l'accès aux objets de dictionnaire. Par exemple, le privilège SELECT ANY TABLE vous permet d'accéder aux vues et tables des autres schémas, mais vous ne pouvez pas sélectionner les objets du dictionnaire (les table, les vues, les packages et les synonymes).

Si le paramètre est configuré à TRUE, l'accès aux objets dans le schéma SYS est permis. Si ce paramètre est configuré à FALSE, les privilèges SYSTEM qui permettent aux objets se trouvant dans les autres schémas, ne permettent pas l'accès aux objets dans le dictionnaire de schéma.

Par exemple, si O7_DICTIONARY_ACCESSIBILITY=FALSE, alors l'ordre SELECT ANY TABLE permettra d'accéder aux vues ou tables dans tous les schémas sauf le schéma SYS. Le privilège système, EXECUTE ANY PROCEDURE permet l'accès aux procédures sur tous les autres schémas sauf le schéma SYS.

3.2.6.Supprimer des privilèges système

Le privilège système peut être supprimé avec la commande REVOKE. Tous les utilisateurs avec ADMIN OPTION dans leur privilège système peuvent supprimer le privilège des autres utilisateurs de la base de données. Pour supprimer un privilège, il n'est pas obligatoire de l'avoir accordé.

Syntaxe:

```
REVOKE {system_privilege|role}
    [, {system_privilege|role} ]...
FROM {user|role|PUBLIC}
    [, {user|role|PUBLIC} ]...
```

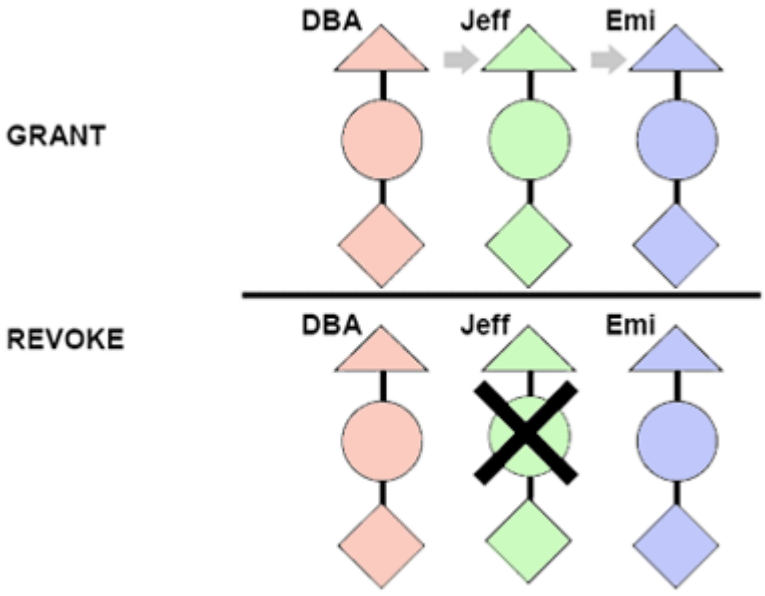
La commande REVOKE ne peut supprimer que des privilèges qui ont été directement accordé avec une commande GRANT.
La suppression de privilèges système a un effet sur quelques objets dépendants.
Par exemple, si SELECT ANY TABLE est attribué à un utilisateur et si l'utilisateur a créé des procédures ou des vues qui utilisent des tables se trouvant dans d'autres schémas, la suppression de ce privilège rend invalide les procédures ou les vues.

Exemple:

```
REVOKE    CREATE TABLE
FROM      emi;
```

→ Supprime le privilège CREATE TABLE de l'utilisateur Emi

3.2.7.Supprimer des privilèges système WITH ADMIN OPTION



Il n'y a pas d'effets en cascade lors de la suppression d'un privilège système, même si il a été donné avec WITH ADMIN OPTION.

Le scénario suivant explique le schéma ci-dessus :

1. le DBA accorde le privilège système CREATE TABLE à Jeff avec ADMIN OPTION
2. Jeff créé une table
3. Jeff attribue le privilège système CREATE TABLE à Emi
4. Emi créé un table
5. Le DBA supprime le privilège système CREATE TABLE à Jeff.

Il en résulte que :
- La table de Jeff existe toujours, mais aucune table ne peut être créée
- La table d'Emi existe encor et elle a toujours le privilège système CREATE TABLE.

3.3.Privilèges sur les objets

3.3.1.Privilèges sur les objets

Object priv.	Table	View	Sequence	Procedure
ALTER	√		√	√
DELETE	√	√		
EXECUTE				√
INDEX	√	√		
INSERT	√	√		
REFERENCES	√			
SELECT	√	√	√	
UPDATE	√	√		

Un privilège objet est un privilège ou un droit pour exécuter une action particulière sur une table, une vue, une séquence, une procédure, une fonction ou un package spécifique.
Chaque objet a une configuration particulière d'attribution de privilège. La table a des listes de privilèges pour des objets variés.
Notez que les seuls privilèges qui s'appliquent à un ordre sont SELECT et ALTER.
UPDATE, REFERENCES et INSERT peuvent être limités en indiquant un sous-ensemble de colonnes à traité. Un SELECT peut être limité par la création d'une vue avec un sous-ensemble de colonnes et une attribution du privilège SELECT sur cette vue. Une attribution de privilège sur un synonyme est convertie à une attribution de privilège sur la table de base référencée par le synonyme.

3.3.2.Accorder des privilèges sur des objets

On utilise l'ordre GRANT pour attribuer des privilèges objets.
Pour accorder des privilèges, l'objet doit être dans votre schéma ou vous devez avoir le privilège WITH GRANT OPTION.
Par défaut, si vous êtes propriétaire d'un objet, tous les privilèges sur cet objet sont automatiquement acquis.
Pour des raisons de sécurité, il faut faire attention aux privilèges que vous accordez aux autres utilisateurs.

Syntaxe:

```
GRANT     { object_privilege [(column_list)]
          [, object_privilege [(column_list)] ]...
          |ALL [PRIVILEGES]}
ON        [schema.]object
TO        {user|role|PUBLIC}
          [, {user|role|PUBLIC} ]...
[WITH GRANT OPTION]
```

Avec:

object_privilege : qui spécifie le privilège objet à attribuer

column_list : définit une colonne d'une table ou d'une vue (Cela ne peut être définit uniquement lors d' l'attribution des privilèges INSERT, REFERENCE ou UPDATE)

ALL : Accorde tous les privilèges pour l'objet qui a été accordé avec WITH ADMIN OPTION

ON *object* : identifie l'objet sur lequel les privilèges ont été attribués

WITH GRANT OPTION : donne la possibilité d'attribué des privilèges objets à d'autre utilisateurs ou rôle.

3.3.3.Supprimer des privilèges sur des objets

L'ordre REVOKE est utilisé pour supprimer des privilèges objets. Pour supprimer un privilège objet, l'utilisateur doit être celui qui l'a accordé.

Syntaxe:

```
REVOKE    { object_privilege [(column_list)]
          [, object_privilege [(column_list)] ]...
          |ALL [PRIVILEGES]}
ON        [schema.]object
TO        {user|role|PUBLIC}
          [, {user|role|PUBLIC} ]...
    [CASCADE CONSTRAINTS]
```

Avec :

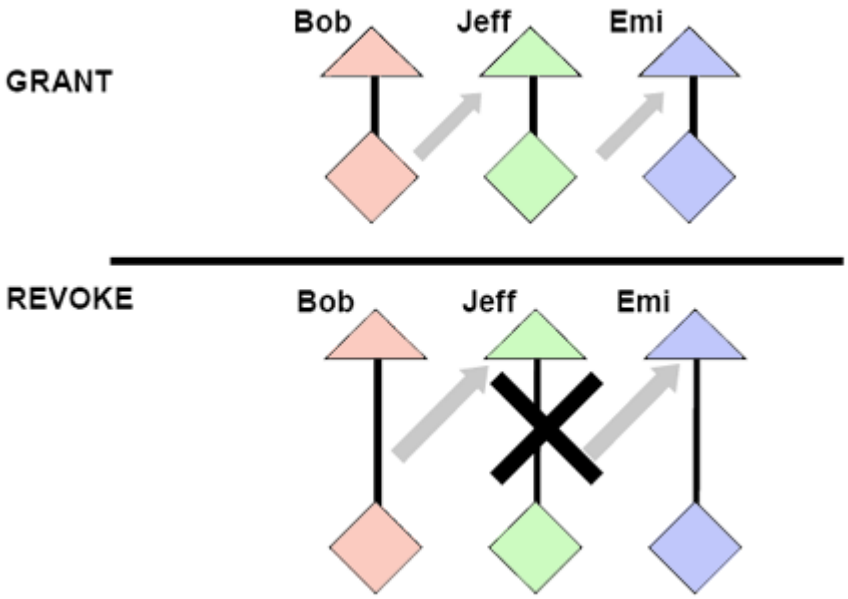
ALL : Supprime tous les privilèges objet qui ont été accordé à l'utilisateur

ON : Identifie l'objet sur lequel les privilèges objet sont supprimés

FROM : Identifie des utilisateurs ou des rôle qui ont leurs privilèges objets supprimés

CASCADE CONSTRAINTS : Supprime toutes les contraintes d'intégrité référentielle que la suppression a définit en utilisant les privilèges REFERENCES ou ALL

3.3.4.Supprimer des privilèges sur des objets WITH GRANT OPTION



Des effets en cascade peuvent être observé lors de la suppression d'un privilège système relié à une opération de DML. Par exemple, si SELECT ANY TABLE est attribué à un utilisateur et que ce dernier a créé des procédures utilisant la table, toutes les procédures se trouvant dans le schéma de l'utilisateur doivent être recompilées avant qu'elles ne puissent être réutilisées.
La suppression d'objet privilège aura aussi un effet en cascade quand WITH GRANT OPTION est donné.

Le scénario suivant explique le schéma ci-dessus :

- Jeff a le privilège objet SELECT sur la table EMPLOYEES avec GRANT OPTION.
- Jeff donne le privilège SELECT sur la table EMPLOYEES à Emi.
- Plus tard, Jeff perd le privilège SELECT. Cette suppression de privilège à un effet en cascade sur Emi.

3.3.5.Récupérer des informations sur les privilèges

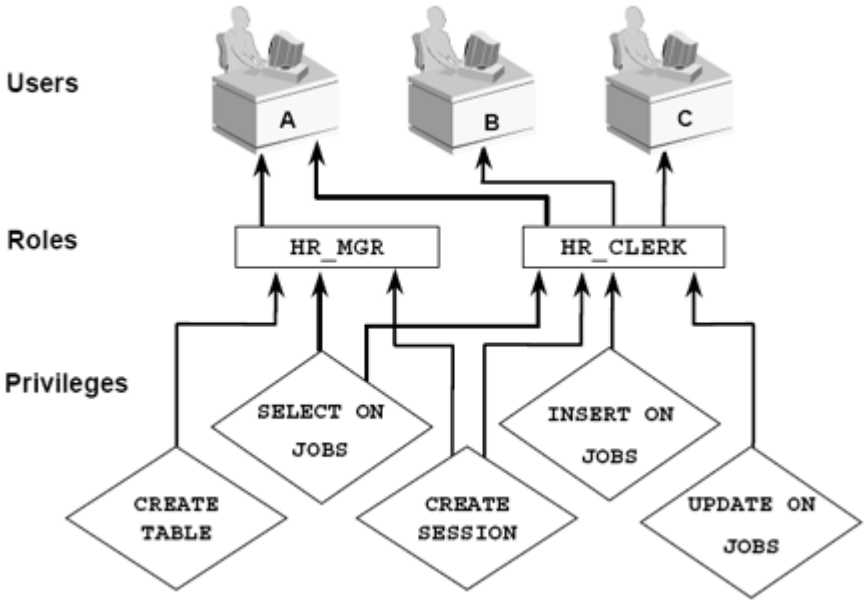
Pour récupérer des informations sur les privilèges, nous avons quatre vues du dictionnaire de données :

BDA_SYS_PRIVS	liste les privilèges système attribués aux utilisateurs et aux rôles.
SESSION_PRIVS	liste les privilèges dont les utilisateurs disposent actuellement.
DBA_TAB_PRIVS	liste toutes les attributions de privilèges sur tous les objets de la base de données.
DBA_COL_PRIVS	décrit toutes les attributions de privilèges sur les colonnes dans la base de données.

4.Gestion des rôles

4.1.Présentation des rôles

4.1.1.Présentation des Rôles



Oracle fournit une facilité de contrôle avec la gestion des privilèges par des rôles.
Les rôles sont des groupes de privilèges qui sont accordé à des utilisateurs ou d'autres rôles. Ils sont conçus pour faciliter l'administration des privilèges dans la base de données.

Un rôle est attribué ou supprimé à des utilisateurs avec les même commandes utilisées pour accorder ou enlever des privilèges systèmes.
Il peut être attribué à n'importe quel utilisateur ou rôle. Cependant, un rôle ne peut pas être attribué ni à lui-même et ni circulairement.
Il peut se composer de privilège système et objet.
Il peut être activé ou désactivé pour chaque utilisateur qui ont ce rôle.
Il peut demander un mot de passe pour être activé.
Son nom doit être unique parmi les logins et les autres noms de rôle.
Il n'appartient à personne et ne se trouve dans aucun schéma.
La description du rôle se trouve dans le dictionnaire de données.

4.1.2.Les avantages de rôles

L'utilisation de rôles simplifie la gestion des privilèges. Au lieu d'accorder le même type de privilège à un ensemble d'utilisateur, vous pouvez accorder ces privilèges à un rôle et ainsi attribuer le rôle à chaque utilisateur.
Si les privilèges associés à un rôle sont modifiés, tous les utilisateurs, qui ont ce rôle, auront automatiquement et immédiatement leurs privilèges modifiés, ce qui fournit une gestion dynamique des privilèges.
Les rôles peuvent être activés ou désactivés temporairement. L'activation d'un rôle peut aussi être utilisé pour vérifier qu'un utilisateur a rôle.

On peut assigner des rôles aux utilisateurs de la base de données en utilisant les commandes ou les outils du système d'exploitation.
L'utilisation de rôle réduit les nombre d'accords de privilèges dans le dictionnaire de données.

4.2.Implémentation des rôles

4.2.1.Créer des rôles

L'ordre CREATE ROLE permet de créer un rôle. Pour cela il faut avoir le privilège système CREATE ROLE. Quand vous créer un rôle qui est NOT IDENTIFIED, INDENTIFIED EXTERNALLY ou By password, Oracle attribut le rôle avec ADMIN OPTION.

Syntaxe:

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

Avec:

role : est le nom du rôle.

NOT IDENTIFIED : indique qu'aucune authentification n'est requise pour activer le rôle.

INDENTIFIED : indique qu'une authentification est requise pour activer le rôle.

BY password : fournit le mot de passe que l'utilisateur doit spécifier quand il active le rôle.

USING package : créé un rôle d'application, c'est est un rôle qui peut être activé uniquement par des applications utilisant un package autorisé.

ESTERNALLY : indique qu'un utilisateur doit être autorisé par un service externe (tel que le système d'exploitation ou le service 3-tiers) avant d'activer le rôle.

GLOBALLY : indique qu'un utilisateur doit être autorisé à employer le rôle par le service d'annuaire d'entreprise avant que le rôle soit activé avec l'ordre de SET ROLE, ou à l'ouverture.

4.2.2.Rôles prédéfinis

Nom du rôle	Description
CONNECT, RESOURCE, DBA	Ces rôles sont fournient pour une compatibilité avec les anciennes versions
EXP_FULL_DATABASE	Des Privilèges pour exporter la base de données
IMP_FULL_DATABASE	Des privilèges pour importer la base de données Database
DELETE_CATALOG_ROLE	Le privilège DELETE sur les tables du dictionnaire de données

EXECUTE_CATALOG_ROLE	Le privilège EXECUTE sur les packages du dictionnaire de données
SELECT_CATALOG_ROLE	Le privilège SELECT sur les tables du dictionnaire de données

Les rôles listés sont définit automatiquement par les bases de données quand vous exécutez les scripts de création de base de données.
Les rôles CONNECT, RESSOURCE et DBA sont fournis pour assurer la compatibilité avec les anciennes versions d'Oracle.
Les rôles EXP_FULL_DATABASE et IMP_FULL_DATABASE sont fournis pour utiliser les utilités d'import et d'export.
Les rôles DELETE_ CATALOG_ROLE DELETE, EXECUTE_CATALOG_ROLE EXECUTE et SELECT_CATALOG_ROLE SELECT sont fournis pour accéder aux vues et packages du dictionnaire de données. Ces rôles peuvent être accordés aux utilisateurs qui n'ont pas le rôle DBA mais qui ont besoin d'accéder aux vues et aux packages du dictionnaire de données.

Le serveur Oracle crée aussi d'autres rôles qui vous autorisent à administrer la base de données. Sur beaucoup de systèmes d'exploitation, ces rôles sont appelés OSOPER et OSDBA, cela dépend du système d'exploitation que vous avez.

D'autres rôles sont définis par des scripts SQL fournis avec la base de données.
Par exemple, AQ_ADMINISTRATOR_ROLE donne des privilèges pour administrer Advanced Queuing. AQ_USER_ROLE est obsolète mais il est conservé pour être compatible avec la version 8.0.

4.2.3.Modifier des rôles

On ne peut modifier que la méthode d'authentification d'un rôle. Vous devez avoir un rôle avec une option ADMIN ou avoir le privilège système ALTER ANY ROLE.

Syntaxe:

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

Exemple:

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

4.2.4.Assigner des rôles

Pour attribuer un rôle à un utilisateur, la syntaxe est la même que pour l'attribution d'un privilège système à un utilisateur :

```
 GRANT     role [, role ]...
 TO        {user|role|PUBLIC}
          [, {user|role|PUBLIC} ]...
 [WITH ADMIN OPTION]
```

Avec:

role	: est un rôle attribué ou un rôle qui reçoit le rôle.
user	: est l'utilisateur qui reçoit le rôle.
PUBLIC	: Assigne le rôle à tous les utilisateurs.
WITH ADMIN OPTION	: permet à l'utilisateur d'assigner le rôle à d'autre utilisateurs ou rôles.

L'utilisateur qui crée un rôle est implicitement assigné à ce rôle avec ADMIN OPTION. Un utilisateur qui n'a pas assigné de rôle avec ADMIN OPTION doit avoir le privilège système GRANT ANY ROLE pour attribué ou retiré un rôle à un utilisateur.

Le nombre maximum de rôles de base de données que des utilisateurs peuvent avoir est configuré par le paramètre d'initialisation MAX_ENABLED_ROLES.

4.3.Gestion des rôles

4.3.1.Mettre en place des rôles par défaut

Un utilisateur peut avoir beaucoup de rôles. Un rôle par défaut est un sous-ensemble de ces rôles qui sont automatiquement activés lors de la connexion de l'utilisateur. Par défaut, tous les rôles assignés à un utilisateur sont activés à la connexion sans avoir besoin de mot de passe. On limite les rôles par défaut pour un utilisateur avec la commande ALTER USER.

La clause DEFAULT ROLE s'applique seulement aux rôles qui ont été attribué directement à l'utilisateur avec l'ordre GRANT. Cette clause ne peut pas être utilisé pour activer :

- des rôles non attribués à l'utilisateur
- des rôles attribués par d'autres rôles
- des rôles gérés par d'autres services externes (comme le système d'exploitation)

Syntaxe:

```
ALTER USER       user
DEFAULT ROLE     {role [,role]...
                 | ALL [EXCEPT role [,role]... ] | NONE}
```

Avec:

user : c'est le nom de l'utilisateur

role : est le rôle qui doit être le rôle par défaut pour l'utilisateur

ALL : tous les rôles sont attribués comme rôle par défaut sauf ceux qui sont dans la liste de EXCEPT

EXCEPT : indique que les rôles suivants ne sont pas inclut dans les rôles par défaut.

NONE : Aucun rôle ne sera attribué comme rôle par défaut.

Vous ne pouvez pas définir les rôles par défaut avec la commande CREATE USER.

4.3.2.Les rôles d'application

La clause USING suivit d'un nom de package, dans l'ordre CREATE ROLE créé un rôle d'application. Un rôle d'application peut être activé uniquement par des applications utilisant un package PL/SQL autorisé.
Les développeurs d'applications n'ont pas besoin de sécuriser un rôle en incluant des mots de passe dans les applications. Au lieu de cela, ils peuvent créer un rôle d'application et spécifier quel package PL/SQL est autorisé à activer le rôle.

Exemple:

```
CREATE ROLE          admin_role
IDENTIFIED USING     hr.employee;
```

→ Dans cet exemple, admin_role est un rôle d'application et le rôle peut être activé seulement par des modules définis dans le package PL/SQL hr.employee.

4.3.3.Activer et désactiver des rôles

Activer ou désactiver des rôles pour activer ou désactivé temporairement les privilèges associés au rôles. Pour activer un rôle, il doit être d'abord attribué à un utilisateur.

Quand un rôle est activé, l'utilisateur peut utiliser les privilèges attribués à ce rôle. Si un rôle est désactivé, l'utilisateur ne peut pas utiliser les privilèges associés à ce rôle sauf si un autre rôle ou ces privilèges sont aussi attribués directement à l'utilisateur. Des rôles sont activés pour une session. A la session suivante, les rôles actifs de l'utilisateur seront retournés comme rôle par défaut.

La commande SET ROLE et la procédure DBMS_SESSION.SET_ROLE activent tous les rôles inclus dans la commande et désactivent les autres rôles. Les rôles peuvent être activés à partir de n'importe quels outils ou programmes qui permettent des commandes PL/SQL ; cependant, un rôle ne peut être activé dans une procédure stockée.

Vous pouvez utiliser la commande ALTER USER … DEFAULT ROLE pour indiquer quels rôles seront activés pour l'utilisateur à sa connexion. Tous les autres rôles sont désactivés.

Un mot de passe peut être demandé pour activer un rôle. Le mot de passe doit être inclut dans la commande SET ROLE pour activer le rôle. Les rôles par défaut assignés à l'utilisateur n'ont pas besoin de mot de passe ; ils sont activés à la connexion, comme des rôles sans mot de passe.

Un rôle ne peut pas être activé à partir d'une procédure stockée, car son action peut changer la sécurité du domaine (la configuration des privilèges) qui permet à la procédure d'être appelé en premier lieu. Ainsi, en PL/SQL, des rôles peuvent être activés ou désactivés dans des blocs anonymes et des procédures d'application (par exemple, Oracle Form Procedure), mais pas dans des procédures stockées.
Si une procédure stockée contient la commande SET ROLE, l'erreur suivante est générée lors de son exécution :

```
ORA-06565: cannot execute SET ROLE from within stored
           procedure
```

Syntaxe:

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

Avec:

Role : le nom du rôle.

IDENTIFIED BY *password* : fournit le mot de passe pour activer le rôle.

ALL : active tous les rôles attribués à l'utilisateur actuel, sauf ceux lister après la clause EXCEPT (vous ne pouvez pas utilisé cette option pour activer des rôles avec mot de passe).

EXCEPT *role* : ne doit pas activer ces rôles.

NONE : désactive tous les rôles de la session actuelle (seul les privilèges assignés directement sont actifs).

L'option ALL sans la clause EXCEPT fonctionne seulement si tous les rôles, qui sont activés, n'ont pas de mot de passe.

4.3.4.Enlever des rôles à des utilisateurs

L'ordre SQL REVOKE enlève un rôle à un utilisateur. Tout utilisateur avec l'option ADMIN pour un rôle peut enlever ce rôle à tous les autres utilisateurs ou rôles de la base de données. Il y a aussi les utilisateurs avec GRANT ANY ROLE qui peuvent enlever tous les rôles aux utilisateurs.

Syntaxe :

```
REVOKE    role [, role ]...
 FROM     {user|role|PUBLIC}
          [, {user|role|PUBLIC} ]...
```

Avec:

role : est le rôle qui doit être enlevé ou le rôle à partir duquel sera enlevé le rôle.

user : est l'utilisateur à partir duquel sera enlevé le rôle.

PUBLIC : enlève le privilège ou le rôle à tous les utilisateurs.

Exemple:

```
REVOKE    oe_clerk
FROM      scott;
```

→ enlève à Scott le rôle oe_clerck

4.3.5.Supprimer des rôles

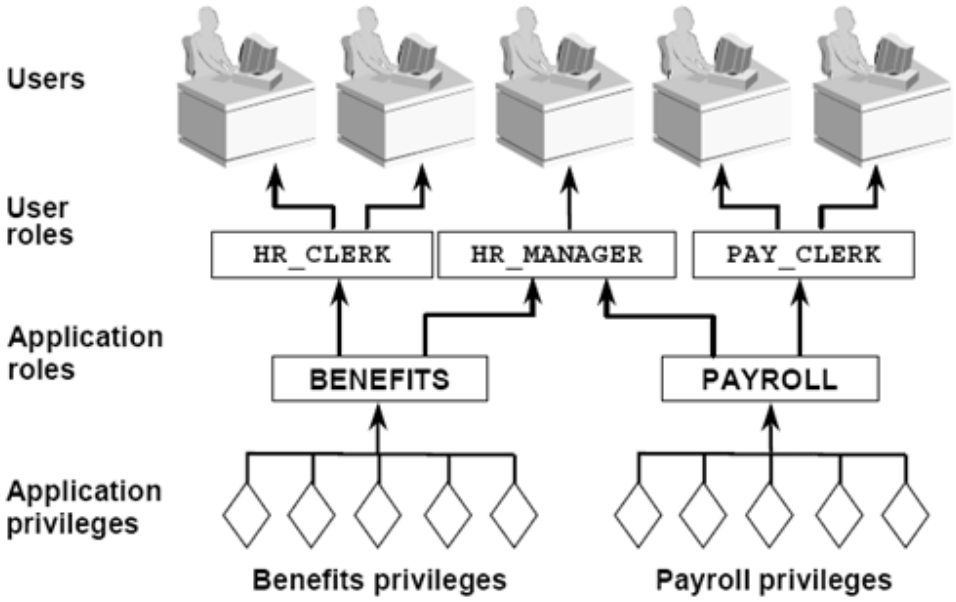
Pour supprimer rôle dans la base de donnée, il faut utiliser la syntaxe suivante :

```
DROP ROLE role;
```

Quand vous supprimer un rôle, le serveur Oracle l'enlève à tous les utilisateurs et rôles qui l'avait et il le supprime de la base de données.
Vous devez avoir le rôle avec l'option ADMIN ou avoir le privilège système DROP ANY ROLE, pour le supprimer.

4.4.Guide d'utilisation des rôles

4.4.1.Guide de création de rôle

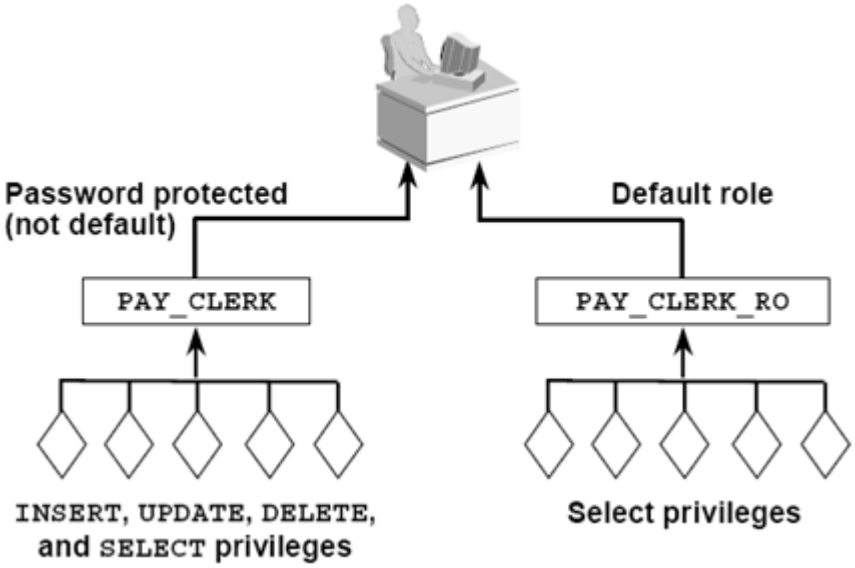


Parce qu'un rôle contient des privilèges nécessaires à l'exécution d'une tache, le nom du rôle est habituellement composer du nom de la tache d'application ou de l'emploi. L'exemple suivant utilise le nom de la tache de l'application et l'emploi concerné pour construire le nom du rôle.

1. Créez un rôle pour chaque tache d'application. Le nom du rôle d'application correspond à une tâche dans l'application, c'est-à-dire PAYROLL.
2. Assignez les privilèges nécessaires pour exécuter la tache du rôle d'application.
3. Créez un rôle pour chaque type d'utilisateur. Le nom du rôle de l'utilisateur correspond à son emploi, c'est-à-dire PAY_CLERCK.
4. Assignez les rôles d'application aux rôles de l'utilisateur.
5. Attribuez les rôles d'utilisateurs aux utilisateurs.

Si une modification de l'application demande que de nouveaux privilèges doivent exécuter la tache payroll, alors le DBA à simplement besoin d'assigner les nouveaux privilèges au rôle d'application, PAYROLL. Ainsi, tous les utilisateurs qui utilisent cette tache, recevront ces nouveaux privilèges.

4.4.2.Guide d'utilisation des mots de passe et des rôles par défaut



Les mots de passe fournissent un niveau de sécurité supplémentaire lors de l'activation du rôle. Par exemple, l'application pourrait demander à l'utilisateur d'entrer un mot de passe pour activer le rôle PAY_CLERK, parce que ce rôle peut être utilisé pour publier des informations confidentielles.

Le mot de passe permet d'activer un rôle uniquement par une application. Cette méthode est illustrée par les points suivants :

1. Le DBA a attribué à l'utilisateur deux rôles, PAY_CLERK et PAY_CLERK_RO.
2. PAY_CLERK a tous les privilèges nécessaires pour exécuter la fonction payroll_clerk.
3. PAY_CLERK_RO (RO pour Read Only) a seulement le privilège SELECT sur les tables qui sont demandés à l'exécution de la fonction payroll_clerk.
4. L'utilisateur peut se connecter à SQL*Plus pour exécuter des requêtes, mais il ne peut pas modifier les données, car PAY_CLERK n'est pas un rôle par défaut et l'utilisateur ne connaît pas son mot de passe.
5. Quand l'utilisateur se connecte à l'application de payroll, cela active PAY_CLERK en fournissant le mot de passe. Il est codé dans le programme et illisible par l'utilisateur.

4.4.3.Afficher les informations d'un rôle

Plusieurs vues du dictionnaire de données qui contiennent des informations sur des privilèges des utilisateurs, contiennent également des informations sur les privilèges des rôles.

Vue	Description
DBA_ROLES	Tous les rôles qui existent dans la base de données
DBA_ROLE_PRIVS	Les rôles assignés aux utilisateurs et aux rôles
ROLE_ROLE_PRIVS	Les rôles qui sont attribués aux rôles
DBA_SYS_PRIVS	Les privilèges systèmes attribués aux rôles et aux utilisateurs
ROLE_SYS_PRIVS	Les privilèges systèmes assignés aux rôles
ROLE_TAB_PRIVS	Les privilèges objets donnés aux rôles
SESSION_ROLES	Les rôles que l'utilisateur actuel a activés

Exemple :

```
SQL> SELECT     role, password_required
  2  FROM       dba_roles;


ROLE                          PASSWORD
----------------------        -----------
CONNECT                       NO
RESOURCE                      NO
DBA                           NO
.
.
.
SELECT_CATALOG_ROLE           NO
EXECUTE_CATALOG_ROLE          NO
DELETE_CATALOG_ROLE           NO
IMP_FULL_DATABASE             NO
EXP_FULL_DATABASE             YES
SALES_CLERK HR_CLERK          EXTERNAL
```

5.Audit

5.1.Catégories d'audit

5.1.1.Audit

Si un utilisateur non autorisé supprime des données, le DBA pourrait décider d'auditer toutes les connexions à la base de données et toutes les suppressions réussite ou non dans les tables de la base de données. Le DBA peut, par exemple, avoir des statistiques sur la mise à jour des tables, l'exécution des entrée/sortie logiques ou le nombre d'utilisateur connectés aux heures de pointes.

5.1.2.Guide d'utilisation d'audit

Il faut limiter l'audit en identifiant d'abord les conditions d'audit et en réglant les options minimales d'audit qui complèteront les conditions. L'audit d'objet doit être employé, si possible, pour réduire le nombre d'entrée générer. Si l'audit de requête et de privilège est nécessaire, la configuration suivant permet de réduire au minimum la génération d'audit :

- spécifiez les utilisateurs à auditer.
- Auditez par session et non par accès.
- Auditez les succès ou les échec, mais pas les deux.
- Les enregistrements d'audit peuvent être écrit sur SYS.AUD$ ou sur l'audit du système d'exploitation, mais cela dépend du système d'exploitation.

Si l'audit atteint sa taille maximale, aucun évènement ne peut être enregistré et l'audit des requêtes ne s'exécutera pas correctement. Des erreurs sont retournées à tous les utilisateurs qui exécutent des requêtes audités. Vous devez alors libérer de l'espace dans le fichier d'audit avant l'exécution de nouvelles requêtes.

Pour être sur que le fichier d'audit ne grossit pas trop rapidement, il faut :

- activer l'audit que si nécessaire.
- Etre sélectif sur les spécifications des options de l'audit
- Contrôler étroitement l'audit des objets de schéma. Les utilisateurs peuvent déclencher un audit sur leurs objets.
- Le privilège AUDIT ANY permet à l'utilisateur de déclencher un audit.

Il faut supprimer périodiquement les enregistrements d'audit du fichier d'audit avec les commandes DELETE et TRUNCATE. Il se trouve dans le dossier $ORACLE_HOME/rdbms/audit directory.

Vous pouvez protéger le fichier d'audit de sorte que les informations d'audit ne puissent pas être ajoutées, modifiées ou supprimées.

Exemple:

```
AUDIT delete ON sys.aud$ BY ACCESS;
```

→ Pour protéger le fichier d'audit de suppressions non autorisées, seul le DBA devrait avoir le rôle DELETE_CATALOG_ROLE.

Comme les nouveaux enregistrements sont insérés dans le fichier d'audit de la base de données, la table AUD$ peut se développer sans limite. Bien que vous ne devriez pas supprimer la table AUD$,

vous pouvez supprimer ou tronquer ses données car elles ne sont là que pour information et n'influencent pas le bon fonctionnement de l'instance Oracle.
Puisque la table AUD$ a une évolution aléatoire (elle grossit, puis rétrécit), elle devrait être stockée hors du tablespace du système.

Pour déplacer la table AUD$ dans le tablespace AUDIT_TAB, il faut :

- s'assurer que l'audit est désactivé
- exécuter la commande suivante :

```
ALTER TABLE          aud$
MOVE TABLESPACE      AUDIT_TAB;
```

- puis exécuter la commande suivante :

```
CREATE INDEX    i_aud1
ON              aud$(sessionid, ses$tid)
TABLESPACE      AUDIT_IDX;
```

- enfin, activer l'audit

5.1.3.Catégories d'audit

Indépendamment de l'audit de la base de données, Oracle enregistre toujours quelques opérations de base de données dans le fichier d'audit du système d'exploitation, qui sont :

- Le lancement de l'instance : L'audit détaille le démarrage de l'instance par l'utilisateur du système d'exploitation ; l'identifiant de l'utilisateur, la date, le time stamp et l'état de l'audit (si il est activé ou pas).
- L'arrêt de l'instance : les détails au niveau du système d'exploitation lorsque l'utilisateur arrête l'instance ; l'identifiant de l'utilisateur, la date et le time stamp.
- Les privilèges d'administration : les détails au niveau du système d'exploitation quand l'utilisateur se connecte à Oracle avec des privilèges d'administrateur.

L'audit de la base de données surveille et enregistre une sélection d'action des utilisateurs de la base. Les informations sur les évènements sont stockées dans la table d'audit.

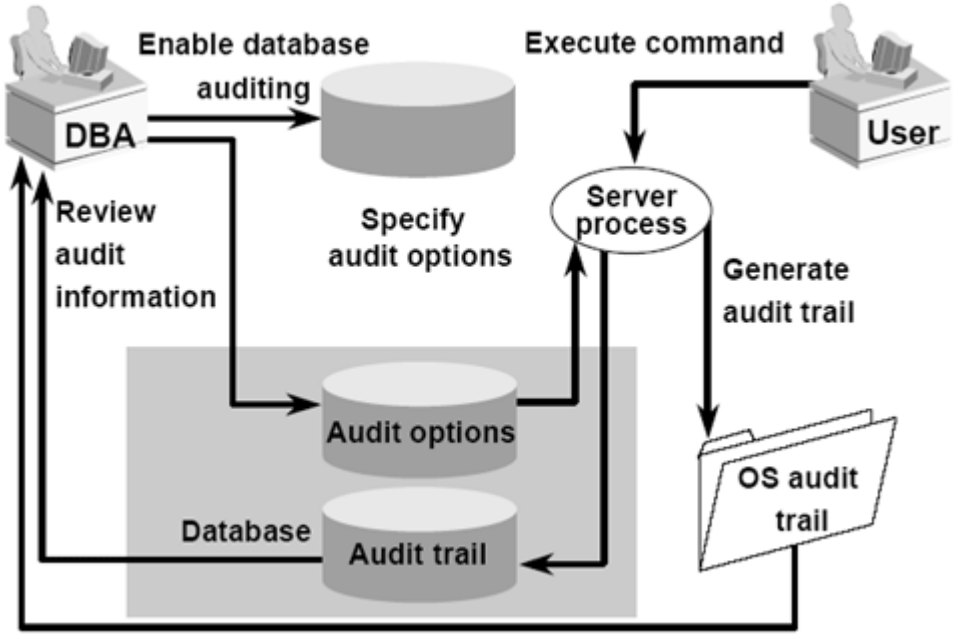
La table d'audit peut être employée pour étudier des activités suspectes. Par exemple, si un utilisateur non autorisé supprime les données d'une table, l'administrateur de la base peut décider de faire un audit sur toutes les connexions à la base de données en rapport avec un succès ou un échec de suppression de ligne dans la table de la base.

L'audit peut également être employé pour surveiller ou recueillir des données sur des activités spécifiques de la base de données. Par exemple, l'administrateur de la base peut récupérer des statistiques sur des tables qui ont été mises à jour, le nombre d'entrées/sorties logiques qui ont été exécuté et le nombre d'utilisateur qui se sont connectés à l'heure de pointe.

L'audit de base de données ne peut pas enregistré les valeurs des colonnes. Si les changements des colonnes de la base de données ont besoins d'être surveillés et que les valeurs de colonnes doivent être stockées à chaque modification, il faut utiliser une application d'audit.
L'application d'audit peut être faite par un code client, des procédures stockées ou des trigger de base de données.

5.2.Audit sur la base de données

5.2.1.Audit sur la base de données



Une fois que vous avez décidez d'auditer, il faut configurer le paramètre d'initialisation AUDIT_TRAIL pour activer l'audit de l'instance. Ce paramètre indique si les informations d'audit sont écrites dans une table de la base de donnée ou dans le fichier d'audit du système d'exploitation.

AUDIT_TRAIL = *value*

Où *value* peut prendre une des valeurs suivante :

DB active l'audit et dirige tous les enregistrement d'audit dans la table d'audit de la base de données (SYS.AUD$)
OS active l'audit et dirige tous les enregistrements d'audit dans le fichier d'audit du système d'exploitation (si cela est possible sur le système d'exploitation)
NONE désactive l'audit (c'est la valeur par défaut)

Les enregistrements d'audit ne sont pas écrits dans le fichier d'audit si le DBA configure le paramètre AUDIT_TRAIL à DB ou OS. Bien que les ordres SQL, AUDIT et NOAUDIT peuvent être employé à tout moment, les informations d'audit ne seront écrites dans le fichier d'audit que si le DBA configure le paramètre AUDIT_TRAIL dans le fichier d'initialisation.

Ensuite, il faut configurer les options d'audit avec la commande AUDIT. Avec la commande AUDIT, on définit quelles commandes, utilisateurs, objets ou privilèges à auditer. Vous pouvez aussi définir si un enregistrement d'audit peut être généré à chaque occurrence ou seulement un fois par session. Si une option d'audit n'est pas nécessaire, vous pouvez interrompre l'option avec la commande NOAUDIT.

Quand les utilisateurs exécutent des requêtes PL/SQL ou SQL, le processus serveur examine les options audit pour déterminer si la requête exécutée génère un enregistrement d'audit. Si nécessaire, les ordres SQL dans des unités de programme PL/SQL sont individuellement audités, quand le programme est exécuté. Comme les vues et les procédures peuvent référencer d'autres objets de base

de données, quelques enregistrements d'audit peuvent être générés comme le résultat d'une exécution d'ordre simple.

La génération et l'insertion des informations dans les tables d'audit sont indépendantes de la transaction de l'utilisateur ; donc, si une transaction de l'utilisateur est annulée, la table d'audit reste intacte. Puisque l'enregistrement d'audit est généré pendant la phase d'exécution, une erreur de syntaxe qui se produit pendant la phase de parse, ne générera pas un enregistrement d'audit.

Pour examiner les informations générer pendant l'audition, il faut interroger les vues du dictionnaire de données de la table d'audit ou utiliser les outils du système d'exploitation pour voir son fichier d'audit. Ces informations sont employées pour déceler des activités suspectes ou pour monitorer les activités de la base de données.

5.2.2.Options d'audit

Nous avons quelques options d'audit :

L'audit des requêtes : C'est un audit sélectif sur les ordres SQL et non sur les objets de schéma sur lesquels ils fonctionnent. Vous pouvez configurer l'audit des requêtes pour auditer des utilisateurs spécifiques ou tous les utilisateur de la base de données.

Exemple :

```
AUDIT TABLE ;
```

→ Traque les ordres DDL indépendamment de la table sur laquelle ils sont publiés.

L'audit sur les privilèges : C'est un audit sélectif sur les privilèges systèmes pour exécuter des actions correspondantes. Il est possible d'utiliser cet audit sur un ou plusieurs utilisateur.

Exemple :

```
AUDIT CREATE ANY TRIGGER;
```

→ Traque le privilège CREATE ANY TRIGGER

L'audit sur les objets du schéma: C'est un audit sélectif d'ordre spécifique sur des objets particulier d'un schéma. L'audit s'applique toujours à tous les utilisateurs de la base de données.

Exemple :

```
AUDIT SELECT ON emi.orders;
```

→ Traque tous les ordres SELECT fait sur la table ORDERS du schéma Emi

Vous pouvez définir n'importe quelle option d'audit et spécifier les conditions suivantes :

- WHENEVER SUCCESSFUL / WHENEVER NOT SUCCESSFUL
- BY SESSION / BY ACCESS

Pour des utilisateurs spécifiques ou pour tous les utilisateurs de la base de données (uniquement pour l'audit des requêtes et des privilèges).

L'audit de précision : Cela fournit le monitoring de l'accès au données basé sur le contenu. Un package PL/SQL DBMS_FGA administre des politiques d'audit. En utilisant DBMS_FGA, le DBA

créé une police d'audit sur la table cible. Si une des lignes retournées par la requête correspond aux conditions d'audit, un évènement d'audit est généré, incluant le nom de l'utilisateur, le texte SQL, la variable demandé, le nom de police, l'id de session, le timestamp et d'autres attributs sont insérés dans la table d'audit.

L'ordre NOAUDIT est employé pour arrêter un audit définit par la commande AUDIT.
L'ordre NOAUDIT inverse les effets de l'ordre AUDIT, mais il a la même syntaxe. Par conséquent, si un ordre AUDIT (l'ordre A) active un audit sur un utilisateur spécifique, et qu'un deuxième ordre (l'ordre B) active un audit sur tous les utilisateurs, puis un ordre NOAUDIT, pour désactiver l'audit sur tous les utilisateurs, inverse l'ordre B mais laisse ordre A actif auditer de l'utilisateur que l'ordre A à définit.

5.2.3.Vues sur les options d'audit

Pour voir les options d'audit, il existe quelques vues du dictionnaire de données :

Vues du dictionnaire de données	Description
ALL_DEF_AUDIT_OPTS	Les options d'audit par défaut
DBA_STMT_AUDIT_OPTS	Les options d'audit des requêtes
DBA_PRIV_AUDIT_OPTS	Les options d'audit des privilèges
DBA_OBJ_AUDIT_OPTS	Les options d'audit des objets d'un schéma

5.2.4.Obtenir des enregistrements d'audit

La table d'audit de la base de données (SYS.AUD$) est une table simple dans chaque dictionnaire de la base de données Oracle. Il y a quelques vues disponibles :

Vues du dictionnaire de données	Description
DBA_AUDIT_TRAIL	Toutes les listes d'audit rentré
DBA_AUDIT_EXISTS	Les enregistrements pour AUDIT EXISTS/NOT EXISTS
DBA_AUDIT_OBJECT	Les enregistrements concernant les objets de schéma
DBA_AUDIT_SESSION	Tous les évènements de connexion ou de déconnexion
DBA_AUDIT_STATEMENT	Les enregistrements de l'audit des requêtes