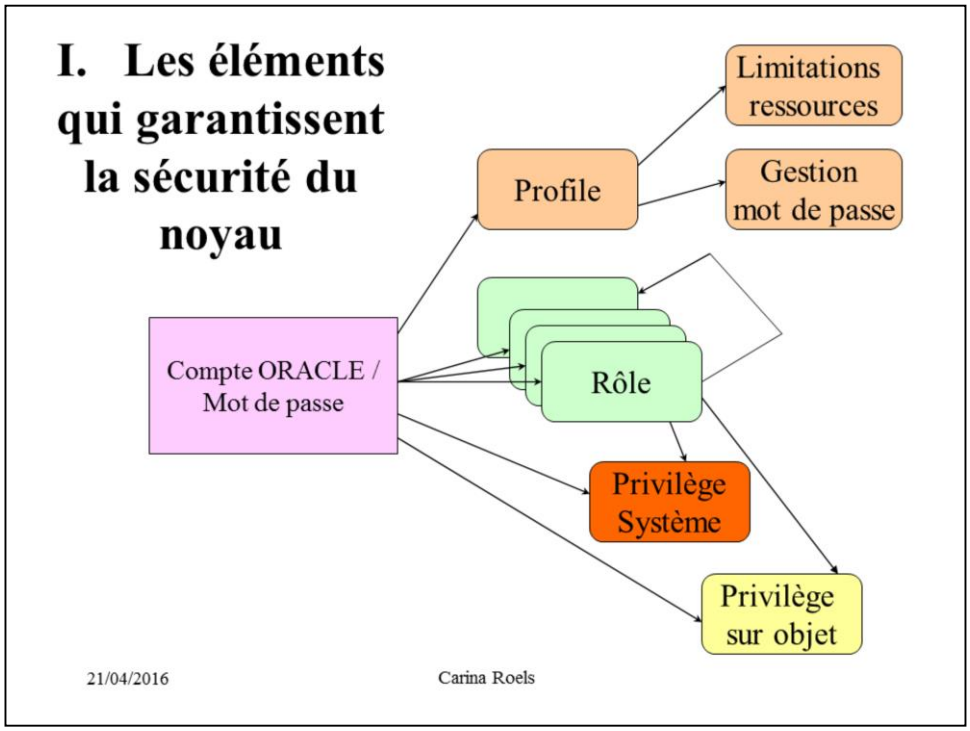
ORACLE
La sécurité du noyau

I. Les éléments qui garantissent la sécurité du noyau
II. Les comptes ORACLE
III. Les profils
IV. Les privilèges et rôles

21/04/2016 Carina Roels



- **Profile** : paramètres relatifs à la gestion de session
- **Limitations ressources** : contrôle de l'utilisation des ressources (temps de connexion, temps CPU, espace disque, etc.)
- **Gestion de mot de passe** : paramètres liés au mot de passe (délai d 'expiration, temps de blocage, etc.)

- **Rôle** : groupe de privilèges (système ou objet)
- **Privilège système** : droit d 'effectuer certaines opérations sur la base (créer des tables, modifier la structure de tables, etc.)
- **Privilège objet** : droit d 'accès à certains objets de la base (consulter une table, ajouter des lignes dans une table, etc.)

Un compte ORACLE se verra attribué :

1 profile

1 à n rôles

éventuellement des privilèges système et / ou des privilèges objet en direct

II. Les comptes ORACLE

- nom / mot de passe
- correspond au schéma de l'utilisateur
- Validation d'accès :
  - par le système d'exploitation
  - par la base de données

21/04/2016 Carina Roels

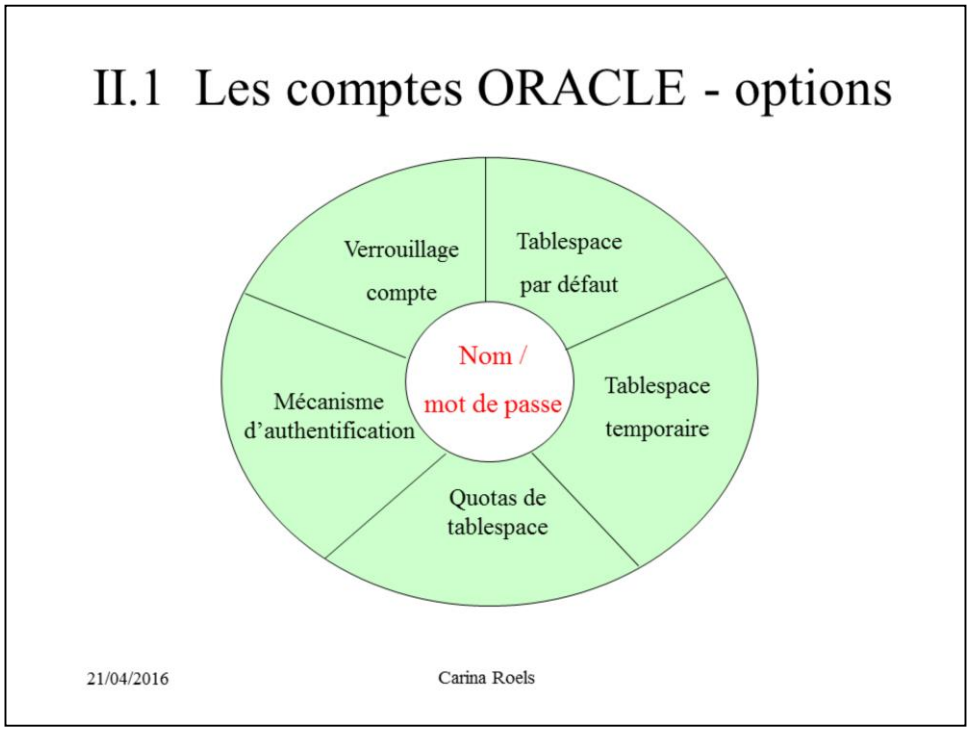
Un compte ORACLE est crée par un DBA. Les informations du compte sont stockées dans le dictionnaire des données.

Dès sa création, le compte correspond au schéma de l'utilisateur qui se verra attribué le compte. Tout objet crée par l 'utilisateur fera partie de son schéma (sauf en cas de précision contraire).

La validation d 'accès peut être gérée d 'une des 2 méthodes :

- **par le système d 'exploitation** : le nom est stocké dans le dictionnaire. Le mot de passe est géré par le système d 'exploitation.

- **Par la base de données** : le nom **et** le mot de passe sont stockés dans le dictionnaire.



- **Tablespace par défaut :** espace de stockage par défaut pour l'utilisateur. Tout objet crée par l'utilisateur, pour lequel n'aura pas été spécifié d'espace de stockage sera stocké dans le tablespace par défaut.

- **Tablespace temporaire** : espace de stockage où ORACLE réera des segments temporaires nécessaires pour effectuer des tris, des jointures ou des regroupements lors de requêtes SQL complexes.

- **Quotas de tablespace** :limitation en K° ou en M° de l 'espace qui pourra être utilisé par l 'utilisateur. Par défaut, un utilisateur n 'a aucun droit.
- **Par contre un DBA dispose de quotas illimités sur l'ensemble des tablespaces de type permanent de l'instance.**

- **Mécanisme d 'authentification** : par le système d 'exploitation ou par la base de données.

- **Verrouillage du compte** : il est possible de créer un compte et de le verrouiller.

II.2 Démarche de création d'un compte ORACLE

- Choisir un nom d'utilisateur et un mécanisme d'authentification
- Choisir **tous les tablespaces** dans lesquels l'utilisateur pourra stocker ses objets
- Décider des quotas pour **chaque tablespace**
- Affecter un tablespace par défaut et un tablespace temporaire
- Créer l'utilisateur

21/04/2016 Carina Roels

Lorsqu'un utilisateur crée un objet dans la base de données :

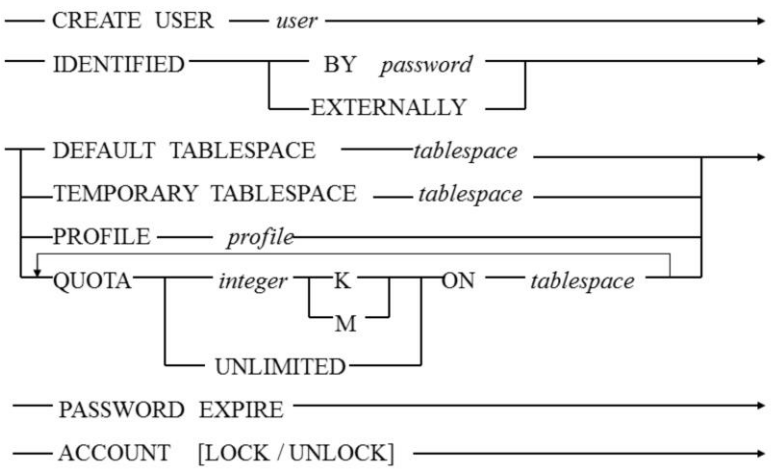
- **sans indication de tablespace dans l'instruction DDL :**
  l'objet est crée dans le tablespace par défaut de l'utilisateur

  Ex.: create table matab (col1 number(2), col2 varchar(25));

- **avec indication d 'un tablespace dans l'instruction DDL :**
  l'objet est crée dans le tablespace indiqué, à condition que l'utilisateur dispose d 'un quota sur le tablespace et que celui-ci ne soit pas épuisé

  Ex.: create table matab (col1 number(2), col2 varchar(25))
  tablespace espace_essai;

II.3 La création d'un compte



- ***Nom*** : nom à créer
- **IDENTIFIED** :

  BY *mot de passe* indique que la validation sera faite par ORACLE

  EXTERNALLY indique que la validation du mot de passe sera faite par le système d 'exploitation

- **DEFAULT TABLESPACE** : permet de spécifier l 'espace de stockage par défaut de l 'utilisateur
- **TEMPORARY TABLESPACE** : permet de spécifier l 'espace de stockage temporaire pour les requêtes de l 'utilisateur
- **PROFILE** : permet de rattacher un profil à l 'utilisateur. Sans spécification un profil par défaut lui sera attribué.
- **QUOTA** : permet d 'allouer un certain espace dans un tablespace. Un quota de 0 indique que l 'utilisateur n 'a pas le droit d'écriture dans le tablespace.
- **PASSWORD EXPIRE** : oblige l 'utilisateur de changer son mot de passe lors de la première connexion.
- **ACCOUNT LOCK/ UNLOCK** : permet de verrouiller ou de déverrouiller un compte. Sans spécification, le compte est non verrouillé.

Exemple

```
CREATE USER gestion
    IDENTIFIED BY gestintern
    DEFAULT TABLESPACE gest_trav
    TEMPORARY TABLESPACE gest_temp
    QUOTA 250 K ON data10
    QUOTA 500 K ON data11
    QUOTA UNLIMITED ON data12
    PASSWORD EXPIRE
```

21/04/2016 Carina Roels

Compte crée : gestion

Mot de passe associé : gestintern (géré par ORACLE)

Espace de stockage par défaut : gest_trav

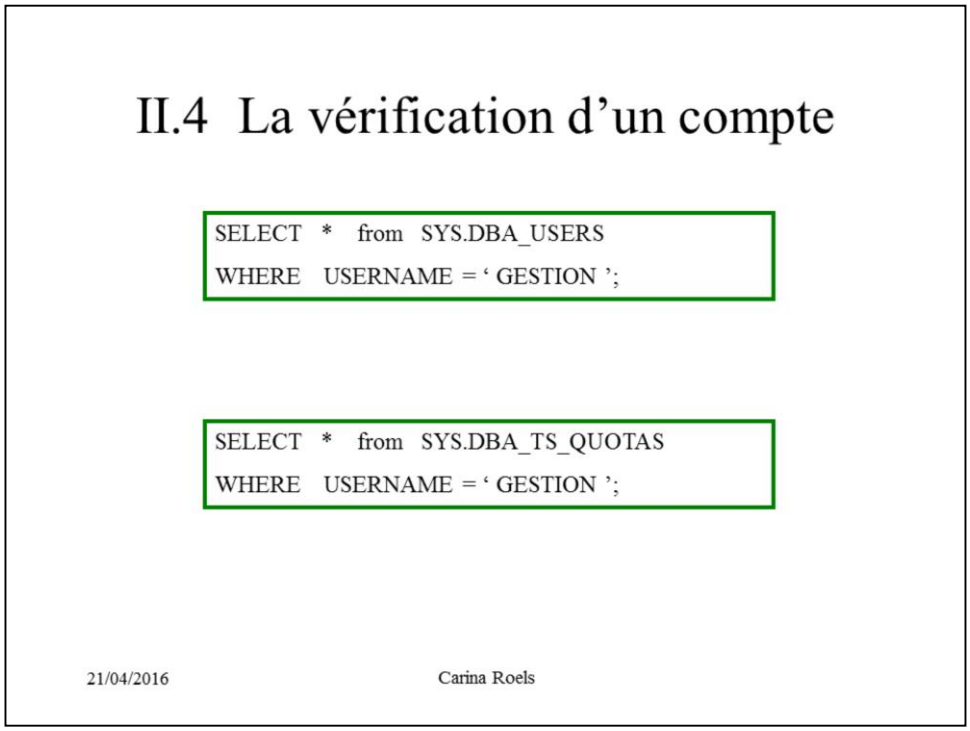
Espace de stockage temporaire : gest_temp

Dispose d ’un espace de stockage sur d ’autres tablespaces :

- 250 K° sur data10
- 500 K° sur data11
- illimité sur data12

Pas de profile spécifié, le profile DEFAULT lui sera automatiquement attribué.

Le mot de passe expire dès la première connexion. L ’utilisateur sera invité à le modifier.



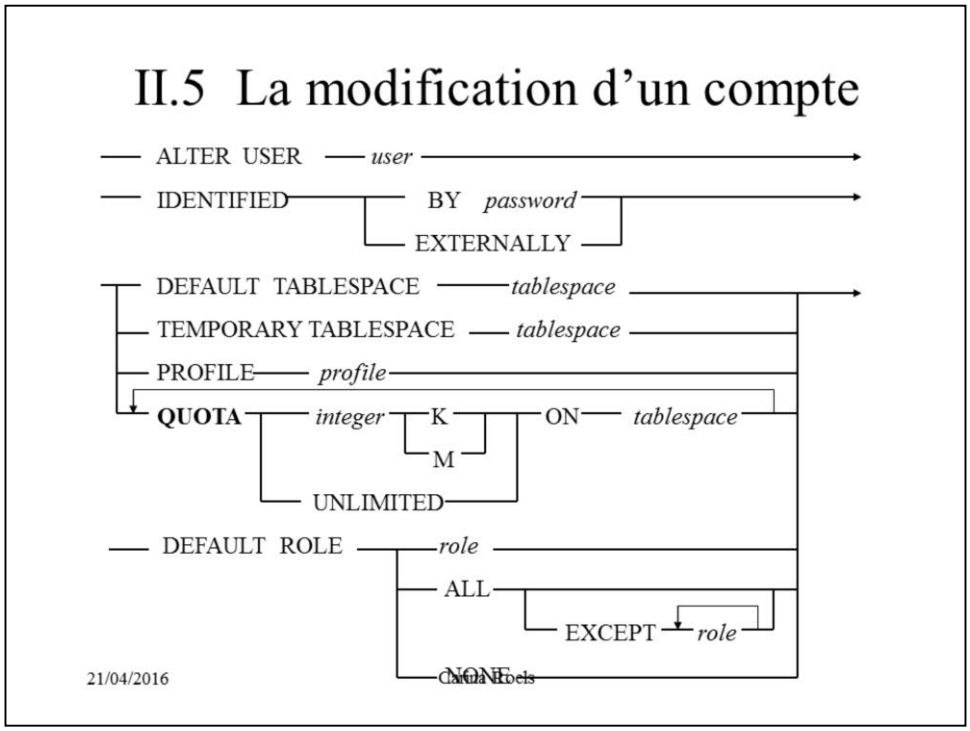
Les informations relatives aux comptes ORACLE sont stockées dans le dictionnaire des données.

Vérification des paramètres d 'un utilisateur :

```
SELECT * from SYS.DBA_USERS
WHERE USERNAME = ' GESTION ';
```

Vérification des quotas sur tablespaces :

```
SELECT * from SYS.DBA_TS_QUOTAS
WHERE USERNAME = ' GESTION ';
```



Les paramètres utilisables sont identiques à ceux utilisables lors de la création d 'un compte.

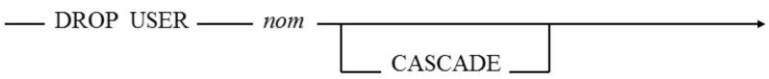
Y compris : PASSWORD EXPIRE
ACCOUNT [LOCK / UNLOCK]
(non présents dans le schéma)

+ **DEFAULT ROLE :**
permet d 'indiquer les rôles qui seront actifs lors de la connexion de l'utilisateur. Par défaut : tous les rôles rattachés à l 'utilisateur sont actifs lors de la connexion.

ATTENTION à la modification des quotas d 'un utilisateur !

Lorsqu 'un quota de 0 est affecté, les objets de l 'utilisateur restent dans le tablespace, mais il n 'est plus possible de leur allouer de l 'espace.

II.6 La suppression d'un compte

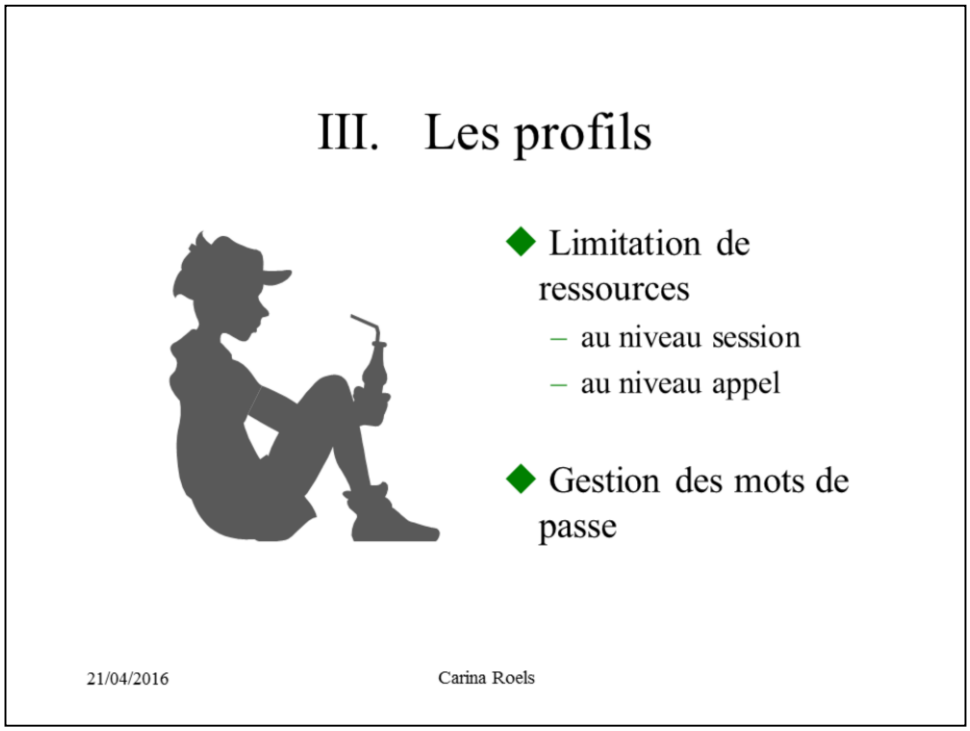


L'option CASCADE est nécessaire
si l'utilisateur possède des objets.

21/04/2016 Carina Roels

L 'option CASCADE supprime le compte ORACLE et tous les objets de l 'utilisateur.

Il n 'est pas possible de supprimer un utilisateur qui est connecté à la base de données.



Les limitations au niveau session :

Une session est crée à chaque connexion d'un utilisateur.

Lors d'un dépassement de ressources, un ROLLBACK est généré sur la transaction en cours et un message d'erreur est affiché. L'utilisateur peut se reconnecter afin de créer une nouvelle session pour continuer ses travaux.

Les limitations au niveau appel :

Un appel correspond à une instruction SQL. Les limitations du niveau appel permettent d'éviter qu'une instruction ne monopolise toutes les ressources.

Lors d'un dépassement de ces ressources, l 'instruction fautive est annulée.L'utilisateur peut continuer à travailler dans sa session.

Activation des limitations spécifiées dans les profiles :

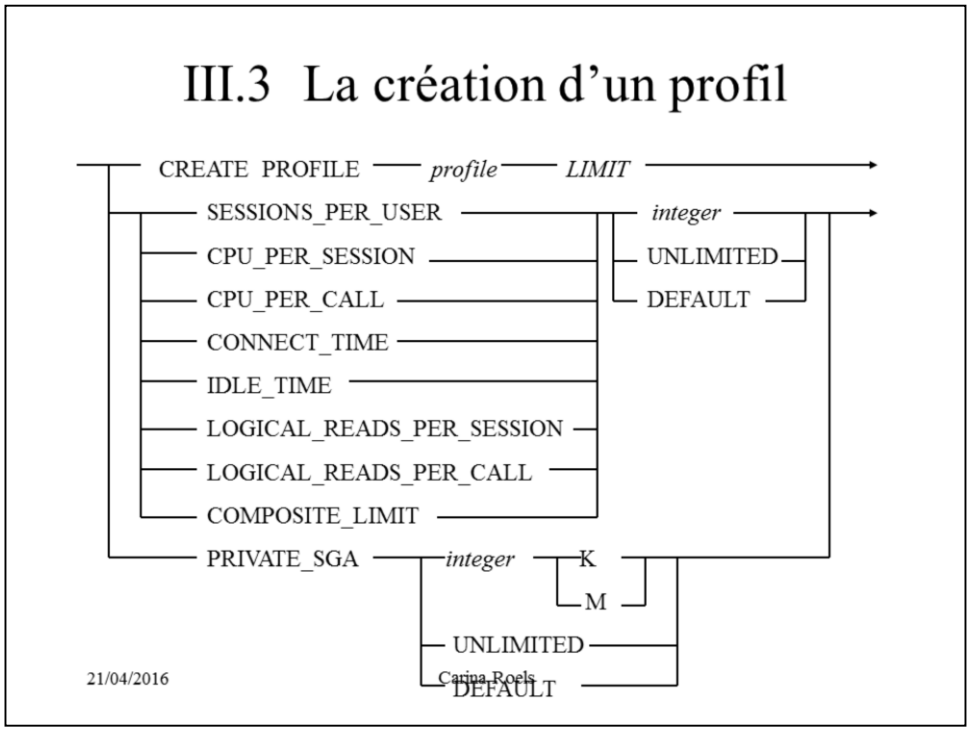
INITsid.ORA : RESOURCE_LIMIT = TRUE

ou : ALTER SYSTEM set resource_limit = true;

La gestion des mots de passe :

Les limites des mots de passe sont toujours actives, même si

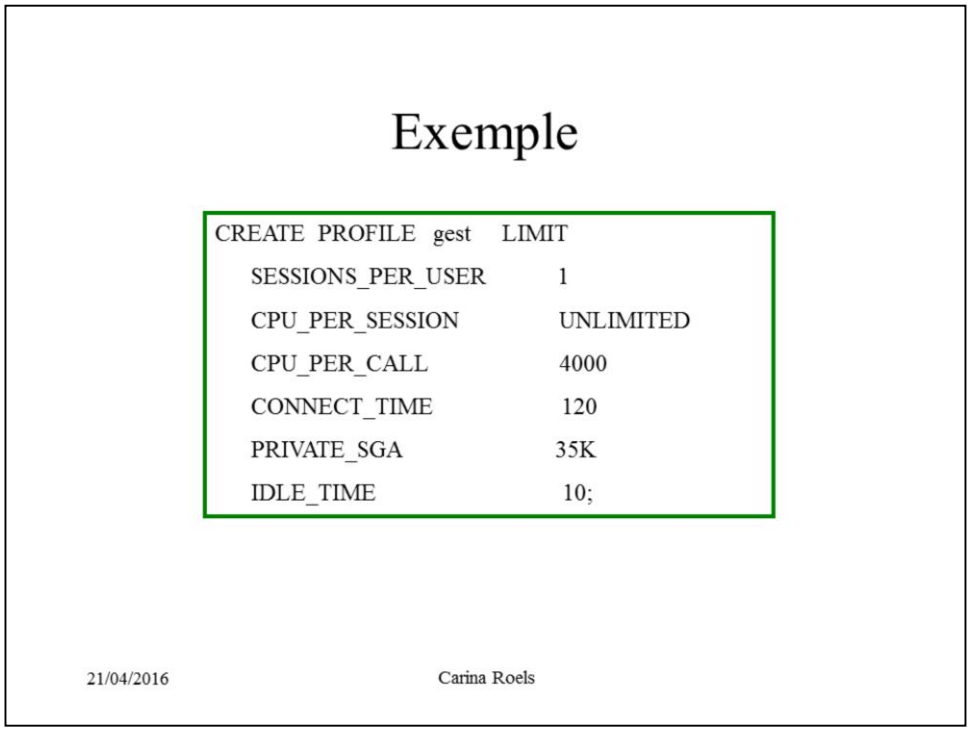
RESOURCE_LIMIT = FALSE !



- ***Profile*** : nom du profil à créer
- **SESSIONS_PER_USER** : permet de limiter le nombre de connexions
- **CPU_PER_SESSION** : limite le temps de CPU pour une session (en 100ème de secondes)
- **CPU_PER_CALL** : limite le temps de CPU par ordre SQL (en 100ème de secondes)
- **CONNECT_TIME** : limite la durée totale de connexion pour une session (en minutes)
- **IDLE_TIME** : limite le temps d'inactivité pour une session (en minutes). Passé ce délai, la connexion est annulée mais l'utilisateur reste dans son environnement. Il se rendra compte de l'annulation à la prochaine instruction.
- **LOGICAL_READS_PER_SESSION** : limite le nombre de blocs de données pouvant être lus dans une session.
- **LOGICAL_READS_PER_CALL** : limite le nombre de blocs de données pouvant être lus par instruction SQL.
- **PRIVATE_SGA** : précise l 'espace mémoire privé qu 'un utilisateur pourra occuper dans la zone SQL partagée (partie du SGA).
- **COMPOSITE_LIMIT** : correspond à un poids total de ressources à ne pas dépasser. Des poids sont attribués aux ressources PRIVATE_SGA, CPU_PER_SESSION, CONNECT_TIME, LOGICAL_READS_PER_SESSION. (la vue RESOURCE_COST indique les coefficients de pondération des différents

paramètres).

- La modification d’un profil existant est réalisable par l’instruction ALTER PROFILE . Les paramètres sont identiques à ceux du CREATE PROFILE

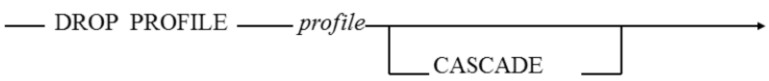


Profile crée : gest

Limitations de ressource :

- 1 seule session autorisé par utilisateur (pour ceux qui ont ce profil)
- pas de limitation de consommation CPU
- maximum 40 secondes de temps CPU par instruction SQL
- maximum 2 h de connexion lors d ’une session
- taille mémoire privée par utilisateur = 35 K°
- Au bout de 10 minutes d ’inactivité : annulation de la session.

III.6 La suppression d'un profil



L'option CASCADE est nécessaire pour retirer le profil des utilisateurs auxquels il avait été attribué.

21/04/2016 Carina Roels

L'option CASCADE supprime le profil et le retire de tous les utilisateur qui l'avaient reçu. Ils se voient alors attribué le profil DEFAULT.

DEFAULT est le seul profil ne pouvant pas être supprimé !

Lorsqu'un profil est supprimé, cette action ne s'appliquera qu'aux nouvelles sessions.

III.7 La gestion des mots de passe

- Paramètres supplémentaires dans CREATE ou ALTER PROFILE
  - FAILED_LOGIN_ATTEMPTS
  - PASSWORD_LOCK_TIME
  - PASSWORD_LIFE_TIME
  - PASSWORD_GRACE_TIME
  - PASSWORD_REUSE_TIME
  - PASSWORD_REUSE_MAX
  - PASSWORD_VERIFY_FUNCTION

21/04/2016 Carina Roels

FAILED_LOGIN_ATTEMPTS Nombre de tentatives de connexion avant verrouillage du compte

PASSWORD_LOCK_TIME Nombre de jours pendant lesquels le compte reste verrouillé après expiration du mot de passe

PASSWORD_LIFE_TIME Durée de vie du mot de passe avant expiration.

PASSWORD_GRACE_TIME Période de grâce (en jours) pendant laquelle l'utilisateur peut encore changer son mot de passe après expiration.

PASSWORD_REUSE_TIME Nombre de jours avant qu 'un mot de passe puisse être réutilisé.

PASSWORD_REUSE_MAX Nombre de fois qu 'un mot de passe puisse être réutilisé.

PASSWORD_VERIFY_FUNCTION Fonction PL/SQL qui effectue un contrôle de complexité du mot de passe.

III.7bis La fonction de vérification du mot de passe

- La fonction PL/SQL livrée : VERIFY_FUNCTION

  (script utlpwdmg.sql dans $ORACLE_HOME/rdbms/admin)

- La création d'une fonction PL/SQL spécifique

21/04/2016 Carina Roels

La fonction livrée VERIFY_FUNCTION.

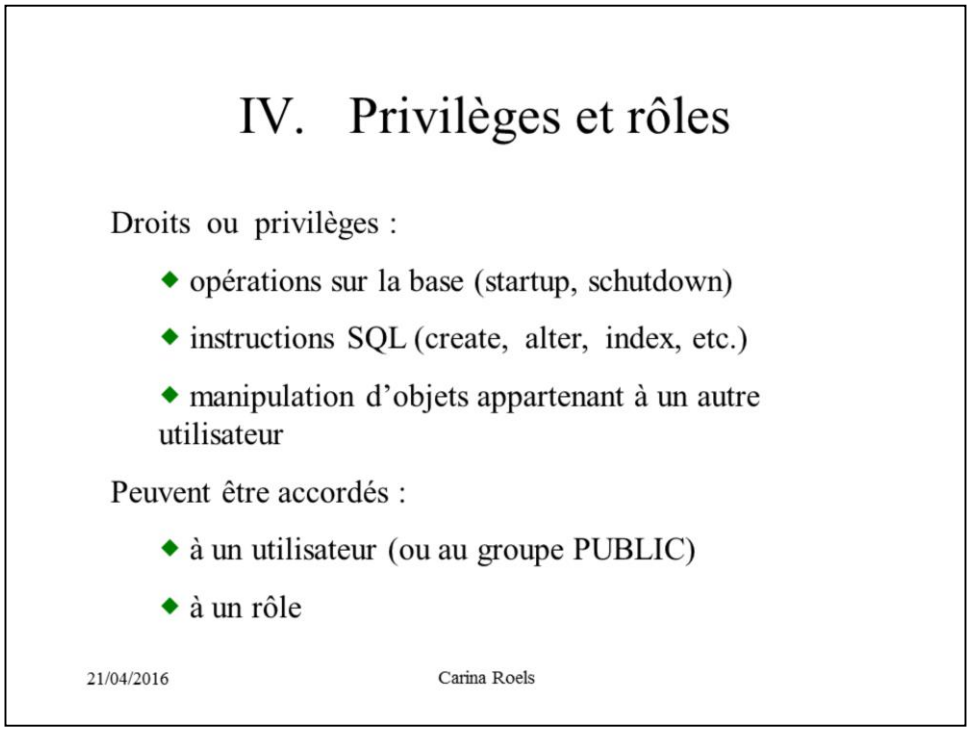
Elle vérifie que :

- la longueur minimale du mot de passe soit de 4 caractères
- le mot de passe ne soit pas identique au nom
- le mot de passe comporte au moins 1 caractère alphabétique, 1 chiffre et 1 caractère spécial.
- Le mot de passe soit différent du précédent par au moins 3 caractères.

La création d 'une fonction PL/SQL spécifique

```
function_name (     userid_parameter      IN  varchar2(30),
                    password_parameter    IN  varchar2(30),
                    old_password          IN  carchar2(30)
              ) RETURN BOOLEAN;
```

- Doit appartenir à l'utilisateur SYS
- Doit retourner TRUE si OK, FALSE en cas d 'échec.



Tout objet crée sous ORACLE est contrôlé par ORACLE.

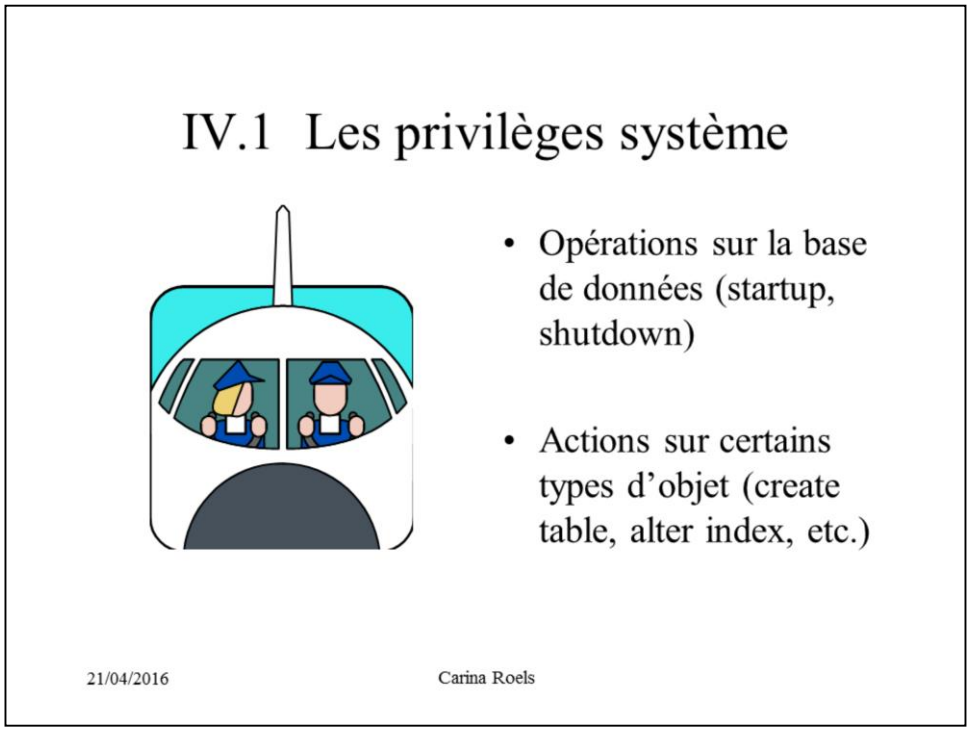
Pour qu 'un utilisateur ait le droit d 'exécuter une tâche sous ORACLE, il doit en avoir reçu le droit.

Les privilèges sont accordés et retirés par des instructions DCL :

GRANT et REVOKE

Pour pouvoir accorder un privilège, il faut :

- l'avoir reçu avec indication WITH GRANT OPTION
- avoir reçu le privilège GRANT ANY PRIVILEGE



Il existe environ 100 privilèges système. La liste de tous les privilèges système est consultable grâce à la vue SYSTEM_PRIVILEGE_MAP.

Le mot clé ANY utilisé lors de l'attribution d'un privilège système signifie que l'on possède le privilège dans tous les schémas de la base de données.

Exemples :

GRANT create session to util1;

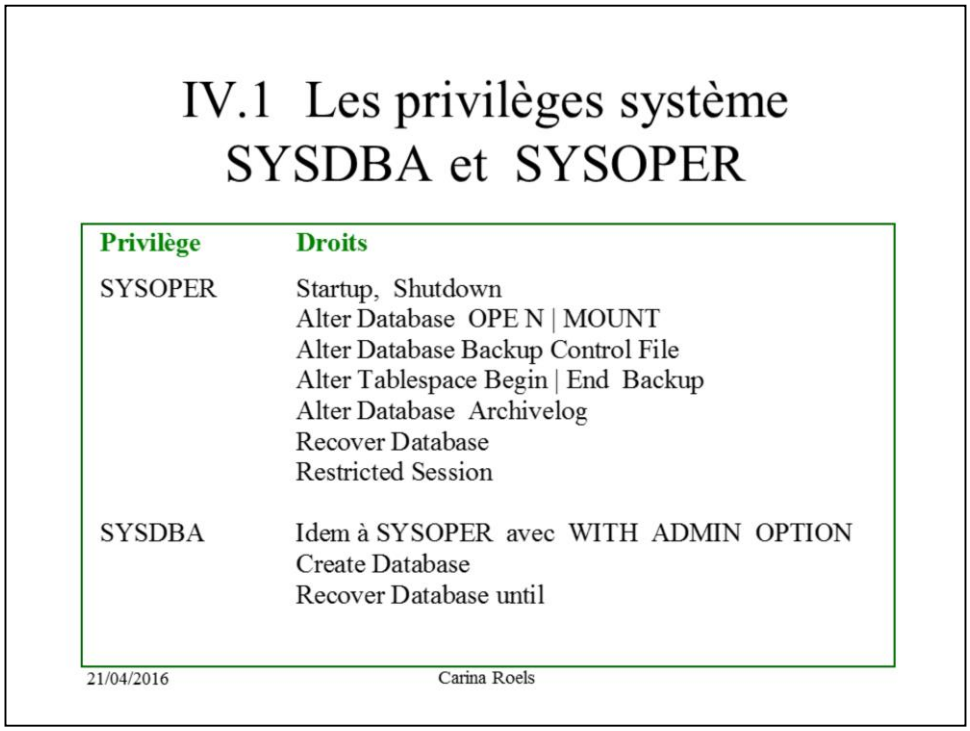
GRANT alter system, alter database, alter tablespace to util1;

GRANT create any table, alter any table, create any index to util1;

GRANT create table to util2;

Remarques :

- CREATE INDEX n'existe pas, CREATE ANY INDEX existe.
- CREATE TABLE comprend CREATE INDEX et ANALYZE, mais l'utilisateur doit avoir un quota sur le tablespace.



Les privilèges système SYSDBA et SYSOPER permettent un accès à une instance de base même lorsque cette base n'est pas ouverte.

Ces privilèges permettent par exemple le démarrage ou l'arrêt de l'instance ou encore les sauvegardes – restauration de base.

Le contrôle de ces privilèges est complètement extérieur à la base de données. (Appartenance à des groupes OS, par exemple ora_dba sous Windows)

L'utilisateur SYS reçoit automatiquement le privilège SYSDBA lors de l'installation.

L'utilisateur SYS doit obligatoirement se connecter en tant que sysdba sous EMDC

IV.2 Les privilèges objet

- Actions (select, insert, etc.) sur certains objets de la base de données crées par d'autres utilisateurs

21/04/2016 Carina Roels

Exemples :

L'utilisateur USERBD possède 3 tables : produit, client, réservation

- Il a tous les privilèges sur les objets de son propre schéma (privilèges implicites).

- Il peut attribuer des privilèges sur ses propres tables :

  GRANT select on produit, client, réservation to util1;

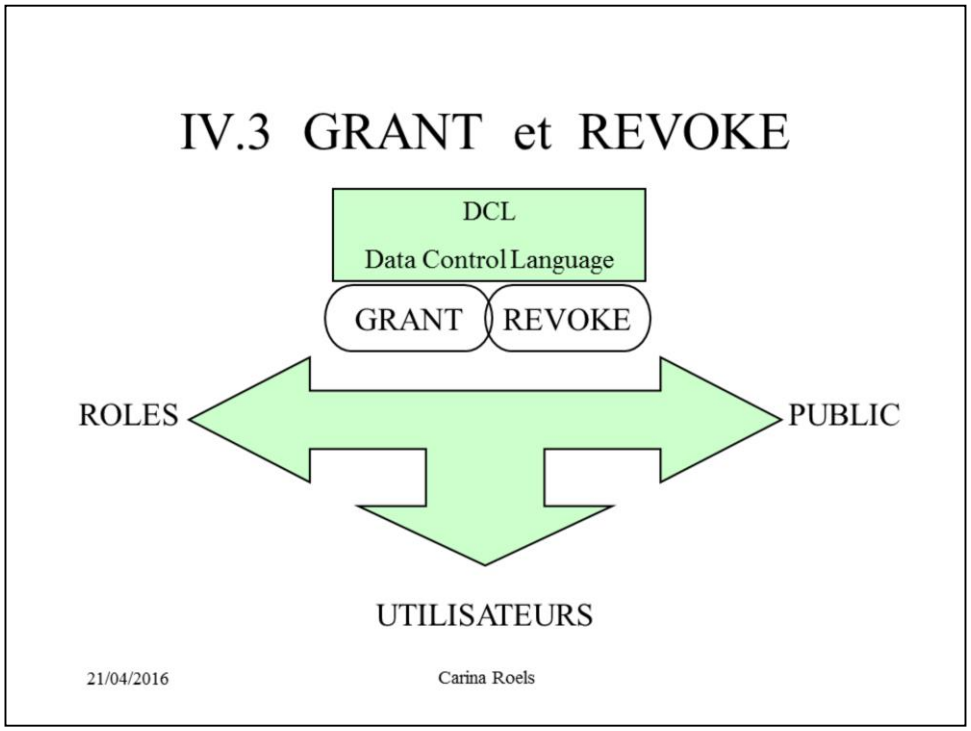
  GRANT select, insert, delete on réservation to util2;

  GRANT select, update on produit, client to util3 **with grant option;**

- L'utilisateur UTIL3 ayant reçu des privilèges avec GRANT OPTION peut transmettre les privilèges reçus à d 'autres utilisateurs :

  GRANT update on **USERBD.**produit to util1;

Un utilisateur ayant le rôle DBA ou ayant reçu le privilège CREATE ANY SCHEMA peut créer des objets dans le schéma de l 'utilisateur USERBD.

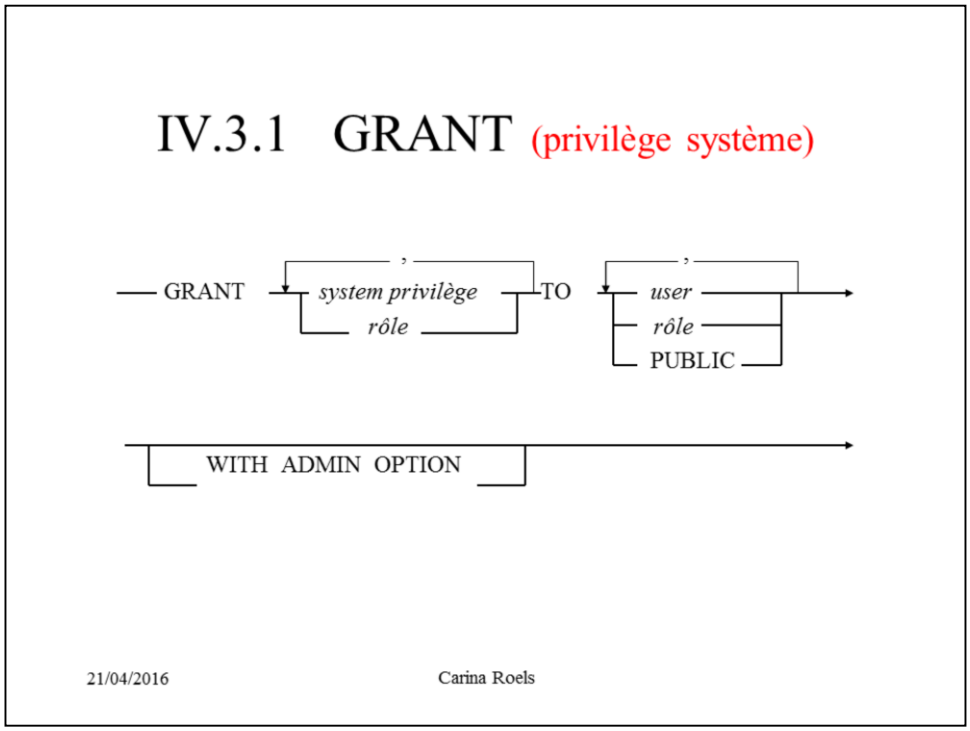


GRANT : attribuer des privilèges
REVOKE : retirer des privilèges

soit :

- **au groupe PUBLIC** : à tous les utilisateurs de la base de données
  Tout utilisateur est automatiquement membre de ce groupe.
  Tout utilisateur peut accorder des privilèges ou des rôles à ce groupe.
  On ne peut pas attribuer de quotas sur des tablespace au groupe PUBLIC.
  Aucun objet ne peut être crée par PUBLIC (ce n'est pas un compte utilisateur).
- **à des utilisateurs**

- **à des rôles**



WITH ADMIN OPTION

Le privilège système reçu peut être accordé à un autre utilisateur par le bénéficiaire.

Si un rôle a été accordé avec cette clause, le bénéficiaire peut modifier et / ou détruire ce rôle !

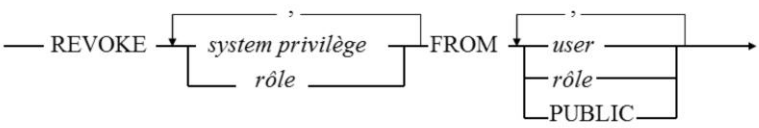
Vérifier les privilèges système accordés :

```
Select * from SYS.DBA_SYS_PRIVS
```

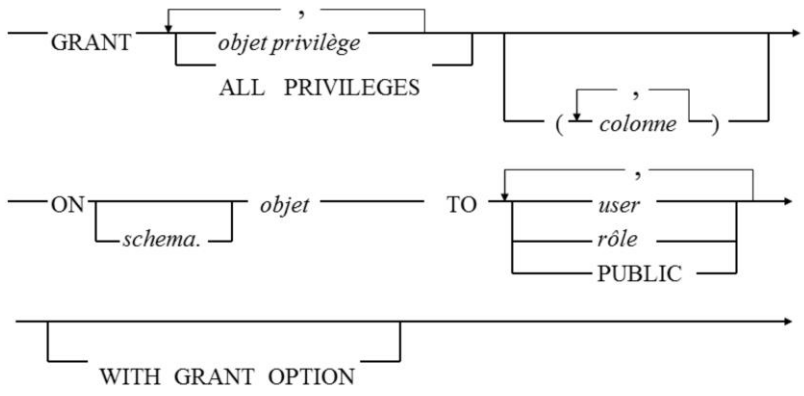
Vérification des privilèges en cours d 'utilisation lors d 'une session :

```
Select * from Session_privs;
```

IV.3.2 REVOKE (privilège système)



IV.3.3 GRANT (privilège objet)



21/04/2016 Carina Roels

Privilège	Table	Vue	Séquence	Procédure
ALTER	✓		✓	
DELETE	✓	✓		
EXECUTE				✓
INDEX	✓			
INSERT	✓	✓		
REFERENCES	✓			
SELECT	✓	✓	✓	
UPDATE	✓	✓		

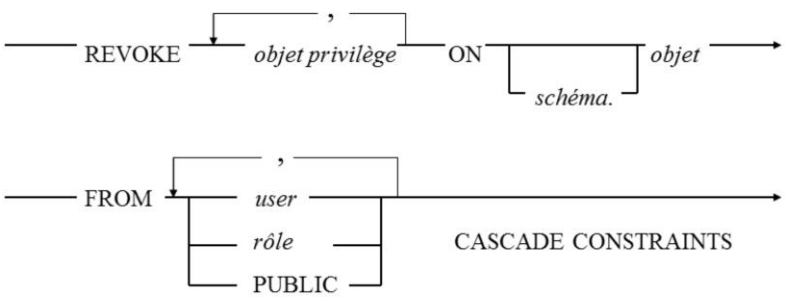
WITH GRANT OPTION

Le privilège objet reçu peut être accordé à un autre utilisateur par le bénéficiaire.

Vérifier les privilèges objet accordés :

```
Select * from SYS.DBA_TAB_PRIVS;
Select * from SYS.DBA_COL_PRIVS;
```

IV.3.4 REVOKE (privilège objet)



21/04/2016 Carina Roels

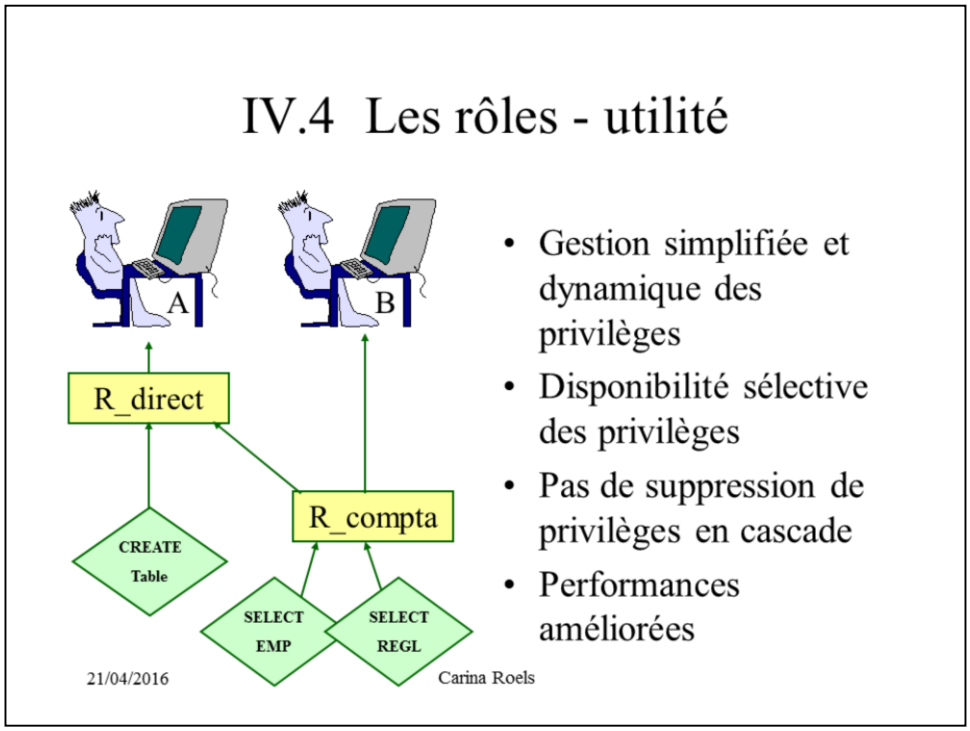
CASCADE CONSTRAINTS

Supprime toute contrainte d 'intégrité référentielle que l 'utilisateur aurait pu créer (grâce au privilège REFERENCES ou ALL).

ATTENTION aux privilèges accordés
avec GRANT ou ADMIN OPTION

Privilèges Objet accordés avec **WITH GRANT OPTION** :
REVOKE retire les privilèges **en cascade** !

Privilèges Système accordés avec **WITH ADMIN OPTION** :
REVOKE ne retire **pas** les privilèges **en cascade** !



Gestion simplifié et dynamique des privilèges

• Au lieu d ’attribuer un ensemble de privilèges à plusieurs utilisateurs :

- o attribution des privilèges à un rôle
- o attribution du rôle aux utilisateurs

• Possibilité d ’imbriquer plusieurs niveaux de rôles

• Si les privilèges d ’un rôle sont modifiés, ces modifications sont automatiquement valables pour tous les utilisateurs ayant reçu le rôle.

Disponibilité sélective des privilèges

Un utilisateur peut posséder plusieurs rôles. Un rôle attribué à un utilisateur peut être temporairement activé ou désactivé.

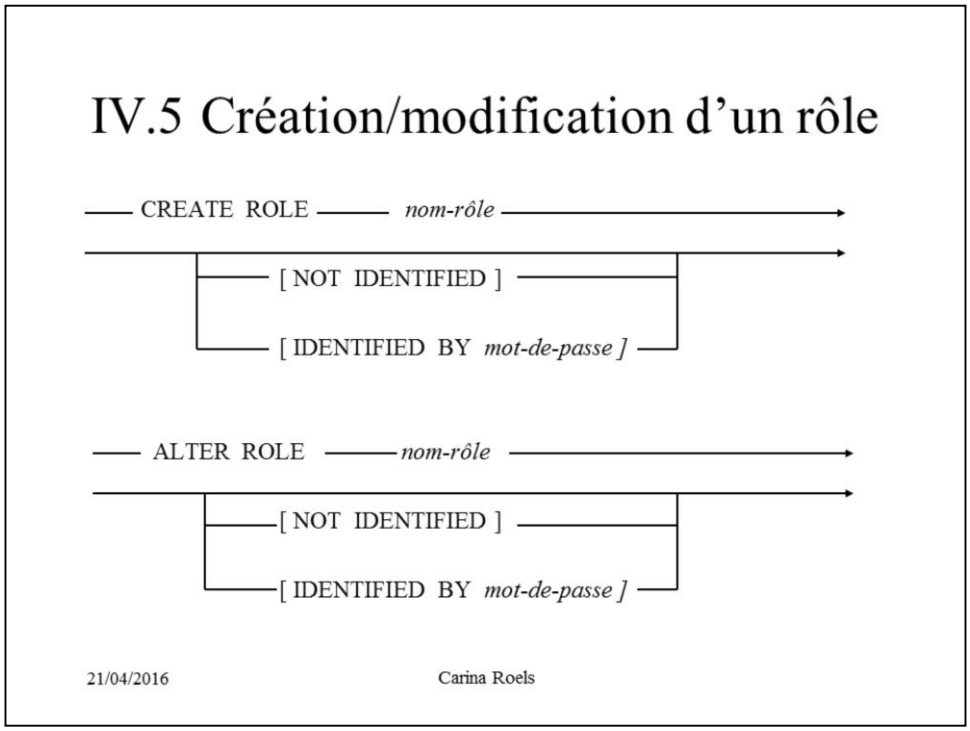
Pas de suppression de privilèges en cascade

Les privilèges peuvent être supprimés d ’un rôle sans entraîner une suppression en cascade.

Performances améliorés

L ’utilisation de rôles diminue le nombre d ’autorisations stockées dans le

dictionnaire des données.



NOT IDENTIFIED / IDENTIFIED BY

indique si une vérification est nécessaire lors de l 'activation du rôle.

Après création d 'un rôle, il faut lui attribuer des privilèges ou éventuellement d 'autres rôles.

Exemples :

Create role r_compta;
Create role r_directeur;
Create role r_gestionnaire

Grant select on reglement to r_compta;
grant select on acompte to r_compta;

grant r_compta to r_directeur, r_gestionnaire;

IV.6 Attribution des rôles

① GRANT nom_rôle TO nom_utilisateur;

② ALTER USER nom_utilisateur DEFAULT ROLE ALL [EXCEPT nom_rôle, nom_rôle, …];

③ ALTER USER nom_utilisateur DEFAULT ROLE nom_rôle [, nom_rôle, …];

④ ALTER USER nom_utilisateur DEFAULT ROLE NONE;

21/04/2016 Carina Roels

① Attribution d ’un rôle à un utilisateur.

Par défaut tous les rôles attribués à un utilisateur sont activés dès la connexion à la base. Il est possible d ’indiquer les rôles (qu ’il a reçu précédemment) qui doivent être activés lors des connexions :

② Attribution des rôles par défaut à un utilisateur :
Ici, tous les rôles existants sauf ceux spécifiés.

③ Attribution d ’un ou plusieurs rôles par défaut à un utilisateur.

④ Indication qu ’un utilisateur n ’a aucun rôle par défaut.

Il est également possible d ’attribuer un rôle pour une session :

SET ROLE nom_rôle [IDENTIFIED BY mot_de_passe];
SET ROLE ALL [EXCEPT nom_rôle, nom_rôle, …];
SET ROLE NONE;

IV.7 Exemple : Attribution d'un rôle applicatif

```
GRANT role_app TO nom_utilisateur;
ALTER USER nom_utilisateur DEFAULT ROLE
  ALL EXCEPT role_app ;

/ * Dans l'application, juste après la connexion:
SET ROLE role_app ;
/* Exécution de l'application
/*Avant de sortir de l'application juste avant la déconnexion
SET ROLE ALL EXCEPT role_app;
```

21/04/2016 Carina Roels

Dans cet exemple, on souhaite attribuer des droits sur des tables mais uniquement dans un contexte applicatif donné.

Lorsque l'utilisateur n'est pas dans le contexte de l'application, il ne joue pas le rôle et donc ne possède aucun droit sur les tables accédées par l'application

Si le rôle a été crée avec un mot de passe associé, il est possible d'utiliser l'application pour demander le mot de passe associé au rôle à l'utilisateur