

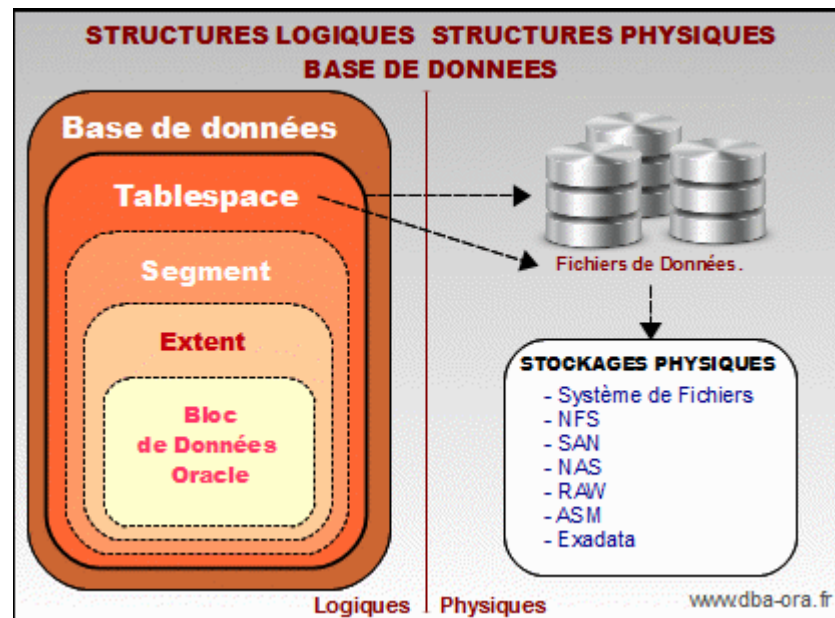


- ▶ Introduction
- ▶ Contenu d'un schéma
- ▶ Rôle de DBA
- ▶ Dictionnaire
- ▶ Gestion des utilisateurs
- ▶ Profil des utilisateurs
- ▶ Rôle et privilèges

- ▶ Correspond à un utilisateur : les deux portent le même nom
- ▶ Ensemble d'objets utilisateurs manipulable en SQL
- ▶ Exemples : Tables, index, triggers, procédures, ...
- ▶ Un objet correspond à :
 - Plusieurs extents : Un ensemble de blocs Oracle contigus
 - Un segment : une table, un index, une partition d'une table, Il est composé d'un ou plusieurs extents.
 - Un tablespace : Un lieu de stockage logique pour des données.
 - Un ou plusieurs fichiers de tablespace
 - Pas de correspondance entre schéma/tablespace

Structure logique de stockage Tablespace Oracle

- Un tablespace est une unité logique de stockage dans Oracle, elle est composée d'un ou plusieurs fichiers physiques appelés fichiers de données. Les niveaux de stockages suivant dans un tablespace Oracle sont les segments, composés d'extents, composés de bloc de données Oracle.



► Niveau logique de stockage Segment Oracle.

L'espace occupé par un objet dans une tablespace est appelé segment. Un segment est un ensemble d'extents alloués et appartient à un tablespace. Lorsque qu'un segment est créé, une ou plusieurs extensions lui sont attribuées.

- Segments de données ou Segments de tables : espace occupé par les tables, les données de table sont stockées dans les extents du segment de tables.
- Segments d'index : espace occupé par les index, les données des index sont stockées dans les extents du segment d'index.
- Segments d'annulation : espace temporaire utilisé pour stocker les données permettant d'annuler une transaction. Les segments d'annulation sont spécifiques au tablespace UNDO.
- Segments temporaire : espace temporaire créé par la base Oracle lorsque l'exécution d'une instruction SQL requiert une zone de travail temporaire, notamment lors d'un tri.

Il existe d'autres segments, les objets de schéma tels que les vues, triggers, packages qui sont stockés dans des segments de métadonnées du système.

► Niveau logique de stockage Extent Oracle

- Un **extent** ou extension est un ensemble de blocs contigus dans l'organisation logique d'une base de données Oracle.
- Un **extent** est composé d'un nombre de blocs de données.

► Niveau logique de stockage Bloc Oracle

- Le bloc de données Oracle est le niveau le plus fin.
- Le bloc de données correspond à un nombre d'octets spécifique d'espace physique sur le disque.
- La taille des blocs est définie lors de la création du tablespace. C'est depuis la version 9i d'Oracle, qu'il est maintenant possible d'utiliser plusieurs tailles de bloc dans une base de données.

Stockage logique des objets dans les tablespaces.

```
SELECT tablespace_name, status, contents, extent_management  
FROM dba_tablespaces;
```

TABLESPACE_NAME	STATUS	CONTENTS	EXTENT_MAN
-----	-----	-----	-----
SYSTEM	ONLINE	PERMANENT	LOCAL
SYSAUX	ONLINE	PERMANENT	LOCAL
UNDOTBS1	ONLINE	UNDO	LOCAL
TEMP	ONLINE	TEMPORARY	LOCAL
USERS	ONLINE	PERMANENT	LOCAL
EXAMPLE	ONLINE	PERMANENT	LOCAL



Stockage physique des données dans les fichiers de données.

SELECT name FROM v\$datafile;

NAME

C:\APP\HADJRABIA-N\ORADATA**DB1**\SYSTEM01.DBF

C:\APP\HADJRABIA-N\ORADATA\DB1\SYSAUX01.DBF

C:\APP\HADJRABIA-N\ORADATA\DB1\UNDOTBS01.DBF

C:\APP\HADJRABIA-N\ORADATA\DB1\USERS01.DBF

C:\APP\HADJRABIA-N\ORADATA\DB1\EXAMPLE01.DBF

- ▶ Pour des tables de moins de 256 attributs
 - Si possible : un tuple est entièrement dans un bloc
 - Sinon : chaînage inter-blocs (augmente les E/S)
- ▶ Pour les tuples de plus de 256 attributs(compris)
 - Chaque partie de 255 tuples est chaînée
- ▶ Les tuples sont chaînés en utilisant les : **ROWID**
- ▶ A la création de la table, on peut :
 - Spécifier le tablespace
 - Spécifier la table du segment
 - Spécifier la quantité d'espace libre dans chaque bloc



- ▶ Accès au catalogue, gestion des utilisateurs, mémoire...
- ▶ Dès la création de la base, deux utilisateurs sont créés avec le rôle DBA
 - SYS
 - Possède toutes les tables système, dont le dictionnaire
 - SYSTEM
 - Pour créer des tables d'administration uniquement

Les privilèges SYSDBA et SYSOPER

- ▶ Permettent d'effectuer les opérations au-dessus de la base de données
 - Démarrer et arrêter le serveur
 - Créer ou supprimer les bases de données
 - Changer les modes d'archivage de la base de données
 -
- ▶ Leur identification est gérée soit :
 - SE
 - L'utilisateur doit appartenir au groupe dba ou oper(ora_dba ou ora_oper sous Windows)
 - connect / as sysdba ; connect / as sysoper.



- Par le fichier password d'Oracle.
 - Créer un fichier password avec l'outil ORAPWD
 - Accorder le privilege a partir de l'utilisateur SYS
 - connect user/user as sysdba;



Informations du dictionnaire

Le dictionnaire des données est organisé comme une base de données relationnelle de sorte que ses tables et vues sont accessibles en SQL, en lecture pour certains utilisateurs, en lecture et mise à jour par le noyau via l'utilisateur **SYS** ; ces vues sont réparties en quatre classes (la liste complète des tables, vues et synonymes s'obtient en consultant la vue **DICTIONARY**) :

- ▶ Ensemble de tables appartenant a l'utilisateur SYS
- ▶ Principales informations
 - La définition de tous les objets et leur espace alloué
 - Les valeurs par absence des attributs
 - Les contraintes d'intégrités
 - La liste des utilisateurs, leurs privilèges
 - Des informations d'audit sur les accès et modifications



Structure du dictionnaire

- ▶ Des tables de base, uniquement accessibles par Oracle et SYS
- ▶ Des vues mises à la disposition des utilisateurs (selon leurs

Droits)

- Vues utilisateurs : USER View. Pour les objets de l'utilisateur qui demande
- Vues d'ensemble : ALL Views. Pour une vue d'ensemble sur les objets auxquels on a accès.
- Vues d'administration : DBA Views. Sur l'ensemble de la

base, uniquement SYS.

- Vues dynamiques sur l'activité de la base : V_\$Views.



Gestion des utilisateurs

Un utilisateur (*user* ou *username*) est identifié au niveau de la base par son nom et peut se connecter puis accéder aux objets de la base sous réserve d'avoir reçu un certain nombre de privilèges.

Un schéma est une collection nommée (du nom de l'utilisateur qui en est propriétaire) d'objets
(tables, vues, séquences, index, procédures, etc.).



Types d'utilisateurs

- ▶ Un compte utilisateur = Un schema de BD
 - C'est un ensemble de d'objets : tables, vues, index,...
 - L'utilisateur crée, modifie, ses objets
- ▶ On peut créer des utilisateurs "simples"
 - Droits de requêtes sur un schéma précis
 - Pas de droits de créations



Gestion des utilisateurs

Etapes de création

- ▶ Choisir un nom et un mode d'identification
 - identification Oracle ou SE (EXTERNALLY)
- ▶ Identifier les tablespaces : trois en general
 - Données = tablespace par défaut de l'utilisateur
 - Tris = tablespace temporaire
 - Index = tablespace dédié aux index en général
- ▶ Décider les quotas pour chaque tablespace
 - Créer l'utilisateur
 - Accorder les rôles et privilèges



Gestion des utilisateurs

La syntaxe SQL de création d'un utilisateur est la suivante :

CREATE USER *utilisateur* IDENTIFIED

{ BY *motdePasse* | EXTERNALLY | GLOBALLY AS
'*nomExterne*' }

[DEFAULT TABLESPACE *nomTablespace*

[QUOTA { *entier* [K | M] | UNLIMITED } ON
nomTablespace]]

[TEMPORARY TABLESPACE *nomTablespace*

[QUOTA { *entier* [K | M] | UNLIMITED } ON
nomTablespace].]

[PROFILE *nomProfil*] [PASSWORD EXPIRE]

[ACCOUNT { LOCK |

UNLOCK }] ;



Gestion des utilisateurs

- ▶ IDENTIFIED BY *motdePasse* permet d'affecter un mot de passe à un utilisateur
- ▶ IDENTIFIED BY EXTERNALLY permet de se servir de l'authenticité du système d'exploitation pour s'identifier à Oracle (cas des compte OPS\$ pour Unix).
- ▶ IDENTIFIED BY GLOBALLY permet de se servir de l'authenticité d'un système d'annuaire.
- ▶ DEFAULT TABLESPACE *nomTablespace* associe un espace disque de travail (appelé *tablespace*) à l'utilisateur.
- ▶ TEMPORARY TABLESPACE *nomTablespace* associe un espace disque temporaire

(dans lequel certaines opérations se dérouleront) à l'utilisateur.



Gestion des utilisateurs

- ▶ QUOTA permet de limiter ou pas chaque espace alloué.
- ▶ PROFILE *nomProfil* affecte un profil (caractéristiques système relatives au CPU et aux connexions) à l'utilisateur.
- ▶ PASSWORD EXPIRE pour obliger l'utilisateur à changer son mot de passe à la première connexion (par défaut il est libre). Le DBA peut aussi changer ce mot de passe.
- ▶ ACCOUNT pour verrouiller ou libérer l'accès à la base (par défaut UNLOCK).

*Pour pouvoir créer un compte utilisateur, il faut disposer du privilège CREATE USER. Il est préférable que cette commande ne soit accessible qu'au **DBA**.*



Création d'un utilisateur

► Exemple :

```
CREATE USER ilc01a IDENTIFIED BY bd2014  
DEFAULT TABLESPACE USERS QUOTA 10M ON USERS  
TEMPORARY TABLESPACE TEMP QUOTA 5M ON TEMP  
PASSWORD EXPIRE;
```

Ilc01a est déclaré « utilisateur », ses objets (pas plus de 10 mégaoctets) seront stockés dans USERS, certaines de ses opérations nécessiteront de ranger des données dans TEMP (pas plus de 5 mégaoctets). Il devra changer son mot de passe à la première connexion.



► Exemple 2 :

```
CREATE USER i1c01a IDENTIFIED BY bd2014  
DEFAULT TABLESPACE USERS  
ACCOUNT LOCK;
```

I1c01a est déclaré « utilisateur », ses objets seront stockés dans USERS, son espace temporaire est SYSTEM. Le compte est pour l'instant bloqué.



Modification d'un utilisateur

```
ALTER USER ilc01a IDENTIFIED BY X_Men  
TEMPORARY TABLESPACE TEMP QUOTA UNLIMITED  
ON TEMP;
```

ilc01a a changé de mot de passe, son espace temporaire est illimité dans TEMP. Il ne devra plus changer son mot de passe à la première connexion.

```
ALTER USER ilc01a DEFAULT TABLESPACE USERS  
QUOTA 10M ON USERS ACCOUNT UNLOCK;
```

L'espace de travail de *ilc01a* est limité à 10 mégaoctets dans USERS. Le compte est débloquent.



Suppression d'un utilisateur

DROP USER *utilisateur* [CASCADE];

Oracle ne supprime pas par défaut un utilisateur s'il possède des objets (tables, séquences, index, déclencheurs, etc.). L'option CASCADE force la suppression et détruit tous les objets du schéma de l'utilisateur.



Profil utilisateurs

Un profil regroupe des caractéristiques système (ressources) qu'il est possible d'affecter à un ou plusieurs utilisateurs.

```
CREATE PROFILE profil_Etudiants LIMIT  
SESSIONS_PER_USER 3 CPU_PER_CALL 3000  
CONNECT_TIME 45 LOGICAL_READS_PER_CALL 1000  
PRIVATE_SGA 15K IDLE_TIME 40  
FAILED_LOGIN_ATTEMPTS 5  
PASSWORD_LIFE_TIME 70  
PASSWORD_REUSE_TIME 60  
PASSWORD_REUSE_MAX UNLIMITED  
PASSWORD_LOCK_TIME 1/24  
PASSWORD_GRACE_TIME 10;
```




- ▶ 3 sessions simultanées autorisées.
- ▶ Un appel système ne peut pas consommer plus de 30 secondes de CPU.
- ▶ Chaque session ne peut excéder 45 minutes.
- ▶ Un appel système ne peut lire plus de 1 000 blocs de données en mémoire et sur le disque.
- ▶ Chaque session ne peut allouer plus de 15 ko de mémoire en SGA.
- ▶ Pour chaque session, 40 minutes d'inactivité maximum sont autorisées.
- ▶ 5 tentatives de connexion avant blocage du compte.
- ▶ Le mot de passe est valable pendant 70 jours et il faudra attendre 60 jours avant qu'il puisse être utilisé à nouveau.
- ▶ 1 seul jour d'interdiction d'accès après que les 5 tentatives de connexion ont été atteintes.
- ▶ La période de grâce qui prolonge l'utilisation du mot de passe avant son changement est de 10 jours.



Profil utilisateurs

L'affectation de ce profil à l'utilisateur `ilc01a` est réalisée via l'instruction `ALTER USER` suivante :

```
ALTER USER ilc01a PROFILE profil_Etudiants;
```

```
ALTER PROFILE nomProfil LIMIT
```

```
{ ParamètreRessource | ParamètreMotdePasse }
```

```
[ ParamètreRessource | ParamètreMotdePasse ]...;
```

Il est plus prudent de restreindre certaines valeurs du profil `DEFAULT` à l'aide de cette commande (`ALTER PROFILE DEFAULT LIMIT...`).



Rôles et Privilèges

► Les privilèges de niveau système

Qui permettent la création, modification, suppression, exécution de groupes d'objets

les privilèges CREATE TABLE, CREATE VIEW, CREATE SEQUENCE par exemple permettent à l'utilisateur qui les a reçu de créer des tables, des vues et des séquences

► Les privilèges de niveau objet

Qui permettent les manipulations sur des objets spécifiques

les privilèges SELECT, INSERT, UPDATE, DELETE sur la table i1c01a.EMPLOYEE par exemple permettent à l'utilisateur qui les a reçu de sélectionner, ajouter, modifier et supprimer des lignes dans la table EMPLOYEE appartenant à l'utilisateur i1c01a



Privilèges système

► Il existe une centaines de privilèges système :

la création d'utilisateurs (`CREATE USER`), la création et la suppression de tables (`CREATE/DROP TABLE`), la création d'espaces (`CREATE TABLESPACE`), la sauvegarde des tables (`BACKUP ANY TABLE`), etc.

Attribution de privilèges Système

- ▶ La commande *GRANT* permet d'attribuer un ou plusieurs privilèges à un ou plusieurs bénéficiaires.
- ▶ L'utilisateur qui exécute cette commande doit avoir reçu lui-même le droit de transmettre ces privilèges.
- ▶ Les utilisateurs *SYS* et *SYSTEM* ont le droit d'utiliser la commande *GRANT*.

```
GRANT { privilègeSystème | nomRôle | ALL PRIVILEGES }  
[ , { privilègeSystème | nomRôle | ALL PRIVILEGES } ]...  
TO { utilisateur | nomRôle | PUBLIC }  
[ , { utilisateur | nomRôle | PUBLIC } ]...  
[ IDENTIFIED BY motdePasse ]  
[ WITH ADMIN OPTION ] ;
```



- ▶ *Privilège Système* : description du privilège système (exemple CREATE TABLE, CREATE SESSION, etc.).
- ▶ ALL PRIVILEGES : tous les privilèges système.
- ▶ PUBLIC : pour attribuer le(s) privilège(s) à tous les utilisateurs.
- ▶ IDENTIFIED BY désigne un utilisateur encore inexistant dans la base. Cette option n'est pas valide si le bénéficiaire est un rôle ou est PUBLIC.
- ▶ WITH ADMIN OPTION : permet d'attribuer aux bénéficiaires le droit de retransmettre le(s) privilège(s) reçu(s) à une tierce personne (utilisateur(s) ou rôle(s)).



- Une fois l'utilisateur créer, il faut que le DBA lui donne le droit de se connecter :

```
GRANT CREATE SESSION TO i1c01a
```

- De créer des tables, des vues, etc...

```
GRANT CONNECT TO i1c01a
```

- De créer des triggers, des procédures, etc...

```
GRANT RESSOURCE TO TO i1c01a
```



Révocation de privilège système

► REVOKE

REVOKE

{ *privilègeSystème* | *nomRôle* | ALL PRIVILEGES }

[, { *privilègeSystème* | *nomRôle* }]...

FROM { *utilisateur* | *nomRôle* | PUBLIC } [, { *utilisateur* | *nomRôle* }]... ;

- ALL PRIVILEGES (valable si l'utilisateur ou le rôle ont tous les privilèges système).
- PUBLIC pour annuler le(s) privilège(s) à chaque utilisateur ayant reçu ce(s) privilège(s) par l'option PUBLIC.



Révocation de privilèges Système

```
REVOKE CREATE SESSION FROM i1c01a;
```

```
REVOKE ALL PRIVILEGES FROM i1c01a;
```



Privilèges objets

Les privilèges objets sont relatifs aux données de la base et aux actions sur les objets (table, vue, séquence, procédure).

GRANT { *privilègeObjet* | *nomRôle* | ALL PRIVILEGES }
[(*colonne1* [,*colonne2*]...)]

[, { *privilègeObjet* | *nomRôle* | ALL PRIVILEGES }]

[(*colonne1* [,*colonne2*]...)]...

ON { [*schéma.*]*nomObjet* | { DIRECTORY *nomRépertoire*
| JAVA { SOURCE | RESOURCE } [*schéma.*]*nomObjet* } }

TO { *utilisateur* | *nomRôle* | PUBLIC } [, { *utilisateur* | *nomRôle* |
PUBLIC }]...

[WITH GRANT OPTION] ;



- ▶ *Privilège Objet* : description du privilège objet (ex : SELECT, DELETE, etc.).
- ▶ *colonne* précise la ou les colonnes sur lesquelles se porte le privilège INSERT, REFERENCES, ou UPDATE (exemple : UPDATE(salaire) pour n'autoriser que la modification de la colonne salaire).
- ▶ ALL PRIVILEGES donne tous les privilèges avec l'option GRANT OPTION) l'objet en question.
- ▶ PUBLIC : pour attribuer le(s) privilège(s) à tous les utilisateurs.
- ▶ WITH GRANT OPTION : permet de donner aux bénéficiaires le droit de retransmettre les privilèges reçus à une tierce personne (utilisateur(s) ou rôle(s)).



Exemples

► Soient deux utilisateurs `ilc01a` et `ilc01b`

- Sur le compte de `ilc01a` :

`Pilote(nump, nom, age, adresse, compagnie, nbHvol)`

- Sur le compte de `ilc01b`

`Qualification(typeQualifn, numpil)`

► `Ilc01a` :

```
GRANT REFERENCES ( nump ) , UPDATE ( nom , age ) , SELECT  
ON Pilote TO ilc01b;
```

Affectation des privilèges de lecture de la table `Pilote`, de modification des colonnes `nom` et `age` et de référence à la clé primaire `nump` à l'utilisateur `ilc01b`.



Modification des colonnes nom et age de la table Pilote de *i1c01a*.

```
UPDATE i1c01a.Pilote SET nom = 'Boutrand',  
        age=age+1 WHERE nom='toto';
```

Lecture de la table Pilote de *i1c01a*.

```
SELECT * FROM i1c01a.Pilote WHERE nom= 'Boutrand';
```

ALTER TABLE Qualification

ADD CONSTRAINT dans_Pilote_i1c01a

```
FOREIGN KEY(numpil) REFERENCES i1c01a.Pilote(nump);
```



Packages et procédures

► Procédure

- i1c01a

GRANT execute on creer_pilote to
i1c01b;

- i1c01b

execute i1c01a.creer_pilote(__,__,__);

► Package

- i1c01a

GRANT execute on P_Creation to i1c01b;

- i1c01b

execute i1c01a.P_creation.creer_pilote(__,__,__);



```
CREATE PACKAGE BODY UPDATE_EMPLOYE AS  
PROCEDURE change_salaire(...) IS BEGIN ... END;  
PROCEDURE change_bonus(...) IS BEGIN ... END;  
PROCEDURE creer_employe(...) IS BEGIN ... END;  
PROCEDURE supprimer_employe(...) IS BEGIN ... END;  
END;  
  
CREATE PROCEDURE P1  
BEGIN update_employe.creer_employe(...)  
END;  
  
CREATE PROCEDURE P2  
BEGIN  
  
    update_employe.supprimer_employe(...)  
  
END ;
```



```
PACKAGE Update_bonus IS
```

```
PROCEDURE P3(...) AS
```

```
BEGIN
```

```
    update_employe.change_salaire(...)
```

```
END give_raise;
```

```
PROCEDURE P4(...)
```

```
BEGIN
```

```
    update_employe.change_bonus(...)
```

```
END;
```

```
GRANT EXECUTE ON P1, P2 TO i1c02a;
```

```
GRANT EXECUTE ON Update_bonus TO i1c02b;
```




Vues

► i1c01a

CREATE or REPLACE View PilotesAF as

Select * from pilote where compagnie='AF'

CREATE or REPLACE View Etat_Civil as

Select nom, adresse, nbHvol, compagnie from Pilote

GRANT SELECT on Pilote_AF, Etat_Civil to i1c01b;

► i1c01b

SELECT SUM(nbHVol) FROM **i1c01a.PilotesAF**;

SELECT count(*) from **i1c01a.ETAT_Civil**;

Révocation de privilèges objets

► I1c01a

- **REVOKE** UPDATE, SELECT **ON** Pilote **FROM** i1c01b;
- **REVOKE** REFERENCES **ON** Pilote **FROM** i1c01b;



Privilèges prédéfinis

- ▶ GRANT GRANT ANY OBJECT PRIVILEGE,
GRANT ANY PRIVILEGE to ilc01a;
- ▶ REVOKE GRANT ANY OBJECT PRIVILEGE,
GRANT ANY PRIVILEGE to ilc01a;

Grant Any Object Privilege : Autorisation de donner tout privilège objet.

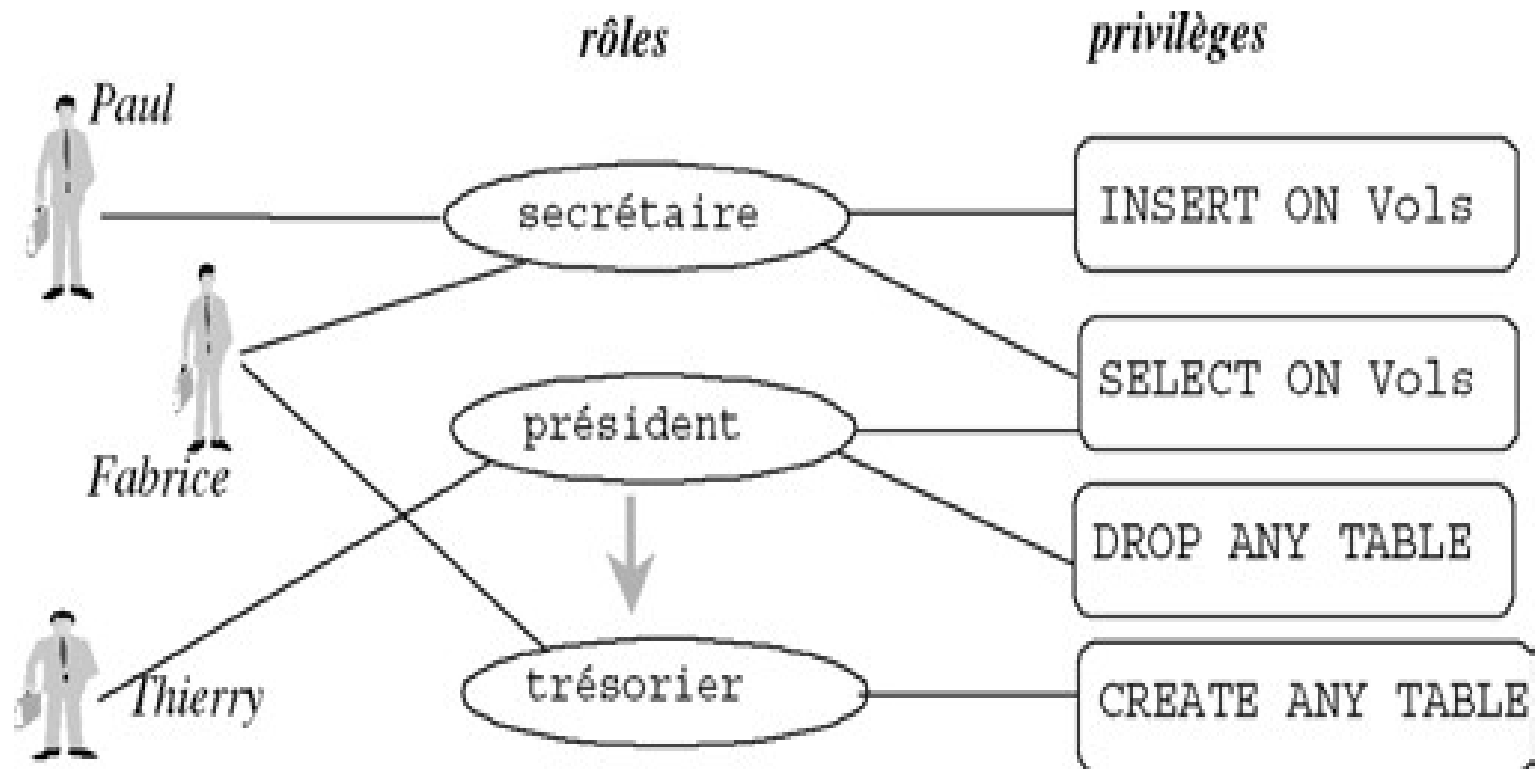
GRANT ANY PRIVILEGE : Autorisation de donner tout privilège système.

Les privilèges système SYSDBA et SYSOPER sont nécessaires pour qu'un utilisateur puisse démarrer (*startup*) ou arrêter (*shutdown*) la base de données



Rôles

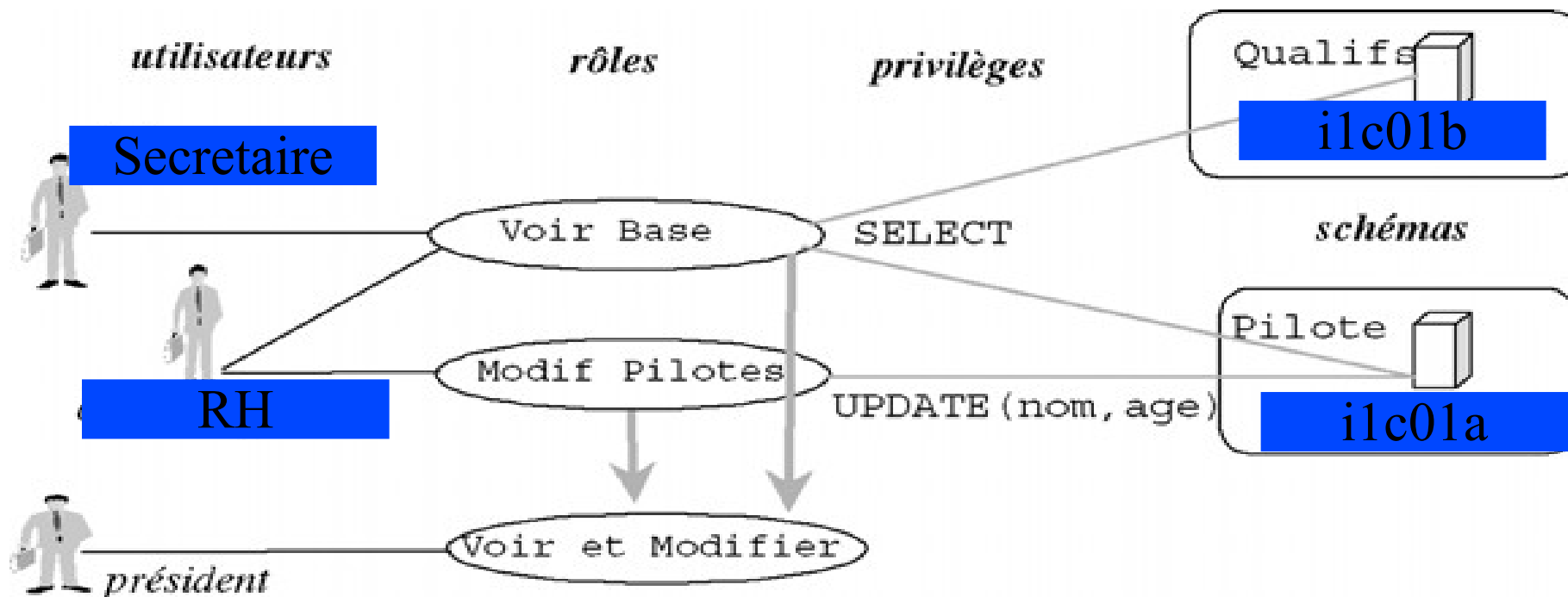
Un rôle (*role*) est un ensemble nommé de privilèges (système ou objets). Un rôle est accordé à un ou plusieurs utilisateurs, voire à tous (utilisation de PUBLIC). Ce mécanisme facilite la gestion des privilèges.





La chronologie des actions à entreprendre pour travailler avec des rôles est la suivante :

- ▶ créer le rôle (CREATE ROLE) ;
- ▶ l'alimenter de privilèges système ou objets par GRANT ;
- ▶ l'attribuer par GRANT à des utilisateurs (voire à tous avec PUBLIC), ou à d'autres rôles ;
- ▶ lui ajouter éventuellement de nouveaux privilèges système ou objets par GRANT.





► Création de 3 rôles

- CREATE ROLE VOIR_BASE;
- CREATE ROLE MODIF_PILOTES;
- CREATE ROLE VOIR_ET_MODIFIER;

► Alimentation des rôles par les privilèges.

- GRANT Select on ilc01a.Pilote to VOIR_BASE;
- GRANT Select on ilc01b.Qualification to VOIR_BASE;
- GRANT UPDATE on ilc01a.Pilote(nom,age) TO MODIF_PILOTES;

► Alimentation d'un rôle par deux autres rôles

- GRANT VOIR_BASE, MODIF_PILOTES TO VOIR_ET_MODIFIER;



- ▶ Affectation des 3 rôles à des utilisateurs.
 - GRANT MODIF_PILOTES TO **RH**;
 - GRANT VOIR_BASE TO **RH, secretaire**;
 - GRANT VOIR_ET_MODIFIER TO **president**;



Révocation d'un rôle

- ▶ Révocation d'un privilège d'un rôle :

```
REVOKE SELECT ON i2c01b.Qualifications  
FROM Voir_Base;
```

- ▶ Révocation d'un rôle d'un utilisateur :

```
REVOKE Voir_base FROM RH
```

- ▶ Révocation d'un rôle d'un rôle. :

```
REVOKE Voir_base FROM VOIR_ET_MODIFIER
```



Suppression d'un rôle

Pour pouvoir supprimer un rôle vous devez en être propriétaire ou en bénéficier via l'option WITH ADMIN OPTION. Le privilège DROP ANY ROLE vous donne le droit de supprimer un rôle dans tout schéma.

La commande DROP ROLE supprime le rôle et le désaffecte en cascade aux bénéficiaires

```
DROP ROLE NOM_ROLE
```



Bibliographie :

- ▶ Christian Soutou : SQL pour Oracle
Edition EYROLLES – 3ième édition.