Plan du module

- **Partie 1- I. Introduction aux SGBDs**
  - Chapitre 1: Présentation des SGBDs
  - Chapitre 2: Rappel. Définition et Evolution des données
  - **Chapitre 3: Contrôle des données**
  - Chapitre 4: Gestion des objets utilisateurs
    - (Vues, séquences et Index)
- Partie 2- II. Langage procédural: PL/SQL
- Partie 3- III. Gestion des Transactions

ENICARTHAGE
المدرسة الوطنية للمهندسين بقرطاج
Ecole Nationale d'Ingénieurs de Carthage

I.3- Contrôle des données

2ème Ingénieurs info

Année Universitaire 2020-2021

Plan

I-Gestion des utilisateurs

1. Classification
2. Création
3. Modification
4. Suppression
5. Profil utilisateur

II- Gestion des droits

1. Politiques de contrôle d'accès
2. Les méthodes à base de privilèges (privilèges systèmes, privilèges Objet, Octroi, révocation, privilèges prédéfinis)
3. Les méthodes à base de rôles (Octroi, révocation, désactivation, réactivation, rôles prédéfinis)

III- Dictionnaire des données

I-Gestion des utilisateurs

1. Classification des utilisateurs

Les types d'utilisateurs, leurs fonctions et leur nombre peuvent varier d'une base à une autre.

- Les utilisateurs peuvent être classifiés pour chaque BD active de la manière suivante:

- Le DBA (*DataBase Administrator*) :Peut effectuer les tâches suivantes :
  – L'installation et la migration des bases;
  – La gestion du réseau, de l'espace disque et des utilisateurs
  – l'optimisation des performances ;
  – Les sauvegardes et les restaurations ;
- Les développeurs: qui interagissent avec les DBA. Ils conçoivent et mettent à jour la base. Ils peuvent aussi agir sur leurs objets (création et modification des tables, index, séquences, etc.).
- Les utilisateurs finaux qui doivent se connecter via les applications

Utilisateurs livrés par oracle

- Lors de l'installation vous avez dû noter la présence des utilisateurs SYS: le propriétaire des tables du dictionnaire de données.
- et SYSTEM L'utilisateur SYSTEM est le DBA qu'Oracle vous offre

2. Création

```
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 } ] ;
```

- *IDENTIFIED BY motdePasse permet d'affecter un mot de passe à un utilisateur local (cas le plus courant et le plus simple).*
- *EXTERNALLY récupère le password du compte utilisateur sur le système d'exploitation.*
- *IDENTIFIED BY GLOBALLY permet de se servir de l'authenticité d'un système d'annuaire.*
- *DEFAULT TABLESPACE associe un espace disque de travail par défaut à l'utilisateur |sinon des zones de travail temporaires seront implantées dans le tablespace SYSTEM*
- *QUOTA: en ko ou Mo permet de limiter l'espace des objets utilisateur dans son tablespace*
- *PROFILE caractéristiques d'utilisation du serveur | sinon le profil default est attribué*
- *PASSWORD EXPIRE Demander à l'utilisateur de changer son password*
- *ACCOUNT LOCK création mais compte encore verrouillé à déverrouiller avant utilisation*

2. Création

- Tablespaces déjà livrés par Oracle:
  - System et sysaux: contiennent notamment le DD, les procédures stockées et les déclencheurs de tout le monde
  - USERS proposé par défaut pour stoker vos données
  - TEMP qui peut agir en complément de la mémoire pour vos tris et jointures.
  - UNDOTBS1: permet la lecture consistante en mode transactionnel (rollback segment)

Exemple

- Identifier les différences entre les deux utilisateurs créés ci-dessous:

```
CREATE USER dev1
IDENTIFIED BY dev1
DEFAULT TABLESPACE USERS
QUOTA 10M ON USERS
QUOTA 1M ON ts_eyrolles
TEMPORARY TABLESPACE TEMP
QUOTA 5M ON TEMP
PASSWORD EXPIRE;
```

```
CREATE USER dev2
IDENTIFIED BY dev2
DEFAULT TABLESPACE USERS
ACCOUNT LOCK;
```

- *dev1 est déclaré « utilisateur »,qui pourrait stocker ses objets (pas plus de 11 mégaoctets) dans les espaces USERS et ts_eyrolles, 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*
- *dev2 est déclaré « utilisateur », ses objets seront stockés dans USERS, son espace temporaire est SYSTEM. Le compte est pour l'instant bloqué.*
- Rq: Si vous ne renseignez pas l'espace par défaut, le tablespace SYSTEM sera associé à l'utilisateur en tant qu'espace de travail et d'espace temporaire.
- Utilisez donc toujours explicitement soit des espaces prédefinis (USERS et TEMP) ou ceux que vous avez créés préalablement (exp ts_eyrolles).

Aperçu sur la création des tablespaces

```
CREATE [BIGFILE|SMALLFILE] TABLESPACE nom_espace
   DATAFILE 'nom_fichier' SIZE taille_initiale K/M/G/T/P/E [REUSE]
   AUTOEXTEND {OFF | ON  NEXT valeur_extension K/M/G/T/P/E}
   MAXSIZE {UNLIMITED |  taille_maxi K/M/G/T/P/E}
```

- *Exemple*

```
CREATE BIGFILE TABLESPACE bigtbs_01 DATAFILE
'c:\App\oradata\ORCL\tablespc_f1.data' SIZE 20M
AUTOEXTEND ON;
```

3. Modification

```
ALTER USER utilisateur
  [ IDENTIFIED { BY password [ REPLACE old_password ] |
               EXTERNALLY | GLOBALLY AS 'external_name' } ]
  [ DEFAULT TABLESPACE nomTablespace
     [QUOTA { entier [ K | M ] | UNLIMITED } ON nomTablespace ] ]
  [ TEMPORARY TABLESPACE nomTablespace
     [QUOTA { entier [ K | M ] | UNLIMITED } ON nomTablespace ].]
  [ PROFILE nomProfil ]
  [ DEFAULT ROLE { rôle1 [,rôle2]… | ALL [EXCEPT rôle1 [,rôle2]…]
  | NONE }
  [ PASSWORD EXPIRE ]   [ ACCOUNT { LOCK | UNLOCK } ] ;
```

- **Exemple: Modification de mdp :**

```
ALTER USER1 IDENTIFIED BY mypass2;
```


3.Modification

- *Exemples (suite)*
  - Changer le MDP de dev1, et affecter lui un quota illimité pour son espace ts_eyrolles. Il ne devra plus changer son mot de passe à la première connexion.
    - **ALTER USER** dev1
      IDENTIFIED BY Mdp_dev1
      QUOTA UNLIMITED ON ts_eyrolles;
  - L'espace de travail de *dev2* doit être maintenant limité à 2 mégaoctets dans USERS. Débloquer le compte.
    - **ALTER USER** dev2
      QUOTA 2M ON USERS
      ACCOUNT UNLOCK;

Pour rappel:

```
CREATE USER dev1
IDENTIFIED BY dev1
DEFAULT TABLESPACE USERS
QUOTA 10M ON USERS
QUOTA 1M ON ts_eyrolles
TEMPORARY TABLESPACE TEMP
QUOTA 5M ON TEMP
PASSWORD EXPIRE;
```

```
CREATE USER dev2
IDENTIFIED BY dev2
DEFAULT TABLESPACE USERS
ACCOUNT LOCK;
```

4. Suppression

Pour pouvoir supprimer un utilisateur vous devez posséder le privilège **DROP USER**

Syntaxe:

```
DROP USER nom_utilisateur [CASCADE];
```

*CASCADE: force la suppression des objets de l'utilisateur (*tables, séquences, index, déclencheurs, etc.) , *nécessite sinon de les supprimer avant)*

- *Rqs:*
  - Si l'utilisateur possède des objets et l'option CASCADE n'est pas présente : Erreur
  - Un utilisateur connecté ne peut pas être supprimé ➔ fermer la session
  - Pour forcer cette suppression, ➔ Arrêter ses sessions par la commande ALTER SYSTEM et l'option KILL SESSION.
  - Pour effacer juste l'utilisateur en tant qu'entrée dans la base sans supprimer ses objets, préférez le retrait par REVOKE du privilège CREATE SESSION.

5. Profil utilisateur

- Un profil est un ensemble nommé de limitations de ressources ou de mot de passe qui peut être attribué à un utilisateur
- Les ressources qui peuvent être limitées:
  - Temps CPU par appel et/ou par session
  - Nombre de lectures logiques par appel et/ou par session (le nombre maximal de blocs à transférer)
  - Nombre de sessions ouvertes simultanément
  - Temps d'inactivité par session
  - Durée totale de la session
- Limitations sur le mots de passe
- Pour pouvoir créer un profil vous devez posséder le privilège CREATE PROFILE.

5.1 Création de profil

```
CREATE PROFILE nomProfil LIMIT
{ ParamètreRessource | ParamètreMotdePasse }
[ ParamètreRessource | ParamètreMotdePasse ]…;

ParamètreRessource :
{ { SESSIONS_PER_USER | CPU_PER_SESSION | CPU_PER_CALL
  | CONNECT_TIME | IDLE_TIME | LOGICAL_READS_PER_SESSION
  | LOGICAL_READS_PER_CALL | COMPOSITE_LIMIT } { entier | UNLIMITED
  | DEFAULT }
  | PRIVATE_SGA {entier[K|M] | UNLIMITED | DEFAULT}}

ParamètreMotdePasse :
{ FAILED_LOGIN_ATTEMPTS | PASSWORD_LIFE_TIME | PASSWORD_REUSE_TIME
  | PASSWORD_REUSE_MAX | PASSWORD_LOCK_TIME | PASSWORD_GRACE_TIME }
  { expression | UNLIMITED | DEFAULT } }
```

Les options principales sont les suivantes :

- SESSIONS_PER_USER : nombre de sessions concurrentes autorisées.
- CPU_PER_SESSION : temps CPU maximal pour une session en centièmes de secondes.
- CPU_PER_CALL : temps CPU autorisé pour un appel noyau en centièmes de secondes.
- CONNECT_TIME : temps total autorisé pour une session en minutes.
- COMPOSIT LIMIT: le coût total des limitations autorisée pour une session. le coût total de toute les ressources à partir du poids attribué aux paramètres CPU_PER_SESSION, CONNECT_TIME, LOGICAL_READS_PER_SESSION, et PRIVATE_SGA.

5.1 Création de profil

- IDLE_TIME : temps d'inactivité autorisé, en minutes, au sein d'une même session (pour les étudiants qui ne clôturent jamais leurs sessions).
- PRIVATE_SGA : espace mémoire privé alloué dans la SGA (System Global Area).
- FAILED_LOGIN_ATTEMPTS : nombre de tentatives de connexion avant de bloquer l'utilisateur (pour la carte bleue, c'est trois).
- PASSWORD_LIFE_TIME : nombre de jours de validité du mot de passe (il expire s'il n'est pas changé au cours de cette période).
- PASSWORD_REUSE_TIME : nombre de jours avant que le mot de passe puisse être utilisé à nouveau. Si ce paramètre est initialisé à un entier, le paramètre PASSWORD_REUSE_MAX doit être passé à UNLIMITED.
- PASSWORD_REUSE_MAX : nombre de modifications de mot de passe avant de pouvoir réutiliser le mot de passe courant. Si ce paramètre est initialisé à un entier, le paramètre PASSWORD_REUSE_TIME doit être passé à UNLIMITED.
- PASSWORD_LOCK_TIME : nombre de jours d'interdiction d'accès à un compte après que le nombre de tentatives de connexions a été atteint (pour la carte bleue, …).
- PASSWORD_GRACE_TIME : nombre de jours d'une période de grâce qui prolonge l'utilisation du mot de passe avant son changement (un message d'avertissement s'affiche lors des connexions). Après cette période le mot de passe expire.

5.1 Création de profil

Exemple 1: Créer le profil profil-Etudiant suivant:

- 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.

```
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
  PASSWORD_GRACE_TIME        10;
```

Affecter le profil créé à l'utilisateur Paul:

```
ALTER USER Paul PROFILE profil_Etudiants;
```


5.2 Modification d'un profil

- Pré-requis: posséder le privilège ALTER PROFILE.

```
ALTER PROFILE nomProfil LIMIT
{ ParamètreRessource | ParamètreMotdePasse }
[ ParamètreRessource | ParamètreMotdePasse ]…;
```

- Exemple: dans le profil_etudiants, on envisage changer le nombre de tentatives de connexion avant le blocage du compte à 3 au lieu de 5 et réduire le temps de blocage du compte après avoir atteint les trois tentatives à 1 heure
  - **ALTER PROFILE profil_etudiants LIMIT**
    FAILED_LOGIN_ATTEMPTS 5
    PASSWORD_LOCK_TIME 1/24;

5.3 Suppression d'un profil

- **DROP PROFILE** *nomprofil* [CASCADE]

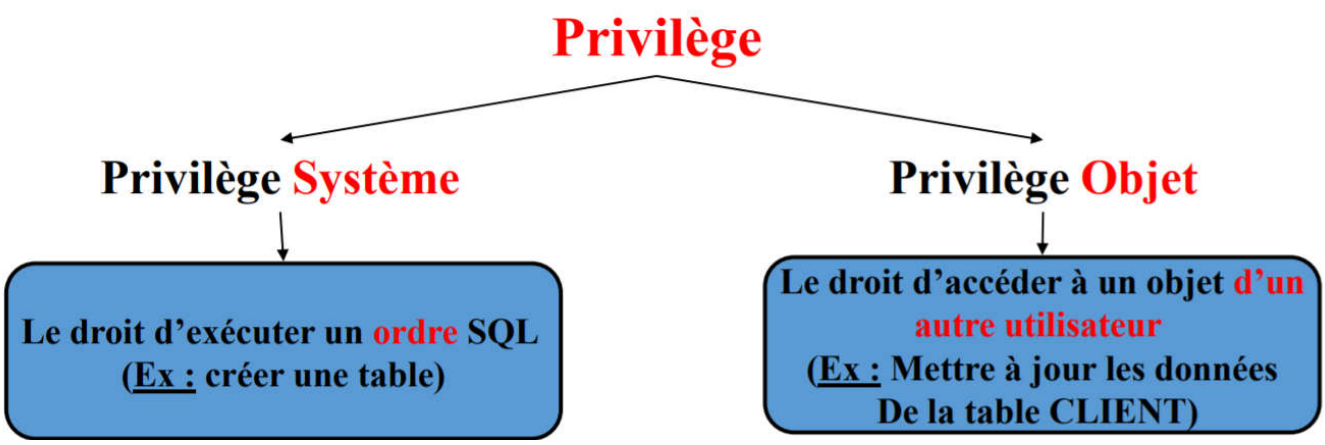
- Si le profile est attribué à des utilisateurs, l'option CASCADE doit être présente.
- Un profil default sera alors affecté aux utilisateurs (toutes les limitations sont enlevées: unlimited)

II. Gestion des droits

1.Politiques de contrôle d'accès

- Un privilège est le droit d'exécuter un ordre SQL spécifique.
- Après la création, l'utilisateur ne dispose d'aucun privilège.
  - Le DBA peut alors lui en accorder.
  - Ces privilèges déterminent ce que l'utilisateur peut faire au niveau de la base de données.
- **Deux familles de contrôle d'accès:**
  - **Politique de contrôle discrétionnaire (DAC)** basée sur la notion de privilège
  - **Politique de contrôle à base de rôles (RBAC)** basée sur le regroupement de privilèges sous un rôle à octroyer à l'utilisateur

2.Politiques à base de privilèges (DAC)



2.1. Privilège système

Attribution de privilèges système:

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

- privilègeSystème: 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.
- IDENl'IFIED 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)).
- **Exemple:** `GRANT CREATE SESSION, CREATE TABLE, CREATE SEQUENCE, CREATE VIEW TO USER1;`

➔ Grant succeeded.

2.1. Privilège système

Révocation de privilèges système

```
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 reçuce(s) privilège(s) par l'option PUBLIC.

Exemples

- On présente quelques exemples de privilèges système
  - Exemple: **GRANT CREATE INDEX ou**
  - **GRANT ALTER|CREATE |DROP ANY INDEX**

Privilège	ALTER	CREATE	DROP	Autre
INDEX		×		QUERY REWRITE (index basés sur des fonctions)
ANY INDEX	×	×	×	
TABLE		×		
ANY TABLE	×	×	×	BACKUP, INSERT, DELETE, SELECT, UPDATE
USER	×	×	×	BECOME (pour des importations de bases)
PROFILE	×	×	×	
SEQUENCE		×		
ANY SEQUENCE	×	×	×	SELECT (pour utiliser toute séquence)
ANY OBJECT PRIVILEGE				pour manipuler tout objet

Exemples

Administrateur

```
1 GRANT CREATE SESSION,
      CREATE SEQUENCE TO Paul;

2 GRANT CREATE TABLE TO Paul
      WITH ADMIN OPTION;

3 GRANT CREATE SESSION,
      CREATE ANY TABLE,
      DROP ANY TABLE TO Paul2;

4 REVOKE
    CREATE SESSION
    FROM Paul, Paul2;

5 REVOKE ALL PRIVILEGES FROM Paul2;
```

① L'admin attribut à Paul les droits: de se connecter (créer une session) et de créer des séquences

② Paul également acquiert le droit de création des tables dans son schéma avec la possibilité de retransmettre ce privilège à un tiers.

③ Paul 2 peut se connecter, créer et détruire des table dans tout schéma (ANY)

④ Les utilisateurs Paul et Paul2 ne peuvent plus se connecter à la base. Ils conservent tous les autres droits. Un tiers qui a acquis le droit de création de tables dans 2 peut toujours le faire.

⑤ La commande revoke retourne une erreur puisque Paul2 n'a pas reçu tous les droits.

2.2. Privilèges objet

- Est le droit d'accéder à un objet d'un autre utilisateur
- Par défaut, seul le propriétaire d'un objet a le droit d'y accéder
- Pour qu'un autre utilisateur puisse accéder à l'objet, le propriétaire de l'objet doit lui donner un privilège objet
- Les principaux privilèges objets sont les suivants :

Privilège	Table	Vue	Séquence	Programme PL/SQL
ALTER			×	
DELETE	×	×		
EXECUTE				×
INDEX	×			
INSERT	×	×		
REFERENCES	×			
SELECT	×	×	×	
UPDATE	×	×		

2.2. Privilège objet

Attribution de privilèges objets

```
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] ;
```

- *privilègeObjet : 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(typeAvion) pour n'autoriser que la modification de la colonne typeAvion).*
- *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

- Soit les schémas suivants appartenant chacun à un utilisateur:

olivier_teste

```
--Table Pilote
BREVET NOM        AGE ADRESSE
------ -------- ----- ---------------
P1     Laroche     39 Montauban
P2     Lamothe     34 Ramonville
P3     Albaric     34 Vieille-Toulouse
P4     Labat       33 Pau
```

christian_soutou

```
--Table Qualif
TYPEQUALIF PIL
---------- ------
PPL        P1
FI/A       P1
PPL        P4
CPL        P4
IFR        P3
```

1. Affectation des privilèges de lecture de la table Pilote, de modification des colonnes nom et age et de référence à la clé primaire brevet à l'utilisateur *christian_soutou*.

```
GRANT REFERENCES(brevet),
      UPDATE(nom, age), SELECT
 ON Pilote
 TO christian_soutou;
```

L'option REFERENCES permet d'implanter une contrainte d'intégrité entre deux tables de schémas distincts. Ici, l'ajout d'une qualification n'est permise que si le pilote est référencé dans la table Pilote du schéma *olivier_*

Exemples

2. Modification des colonnes nom et Age de la table Pilote de *olivier_teste*.

```
UPDATE olivier_teste.Pilote
 SET   nom = 'Boutrand', age = age+1
 WHERE nom = 'Labat';
```

3. Lecture de la table Pilote de *olivier_teste*.

```
SELECT * FROM olivier_teste.Pilote
       WHERE nom = 'Boutrand';
BREVET NOM           AGE ADRESSE
------ -------- ------- -----------
P4     Boutrand       34 Pau
```

4. Déclaration d'une clé étrangère vers la table Pilote de *olivier_teste*.

```
ALTER TABLE Qualifications
 ADD CONSTRAINT dans_Pilote_olivier_teste
 FOREIGN KEY(pil)
 REFERENCES olivier_teste.Pilote(brevet);
```

2.2. Privilège objet

Révocation des privilèges

```
REVOKE { privilègeObjet | ALL PRIVILEGES } [(colonne1 [,colonne2]…)]
   [, { privilègeObjet | ALL PRIVILEGES }] [(colonne1 [,colonne2]…)]…
ON { [schéma.]nomObjet | { DIRECTORY nomRépertoire
   | JAVA { SOURCE | RESOURCE } [schéma.]nomObjet } }
FROM { utilisateur | nomRôle | PUBLIC } [,{ utilisateur | nomRôle |
PUBLIC } ]…
[CASCADE CONSTRAINTS] [FORCE];
```

- CASCADE CONSTRAINTS concerne les privilèges REFERENCES ou ALL PRIVILEGES. Cette option permet de supprimer la contrainte référentielle entre deux tables de schémas distincts.
- FORCE : concerne les privilèges EXECUTE sur les types (extensions SQL3). En ce cas, tous les objets dépendants (types, tables ou vues) sont marqués INVALID et les index sont notés UNUSABLE.

Exemples

`REVOKE UPDATE, SELECT ON Pilote FROM christian_soutou;`	*christian_soutou* ne peut plus modifier ni lire la table `Pilote` de *olivier_teste*.
`REVOKE REFERENCES ON Pilote FROM christian_soutou;`	Commande incorrecte car l'option `CASCADE CONSTRAINT` doit être utilisée.
`REVOKE REFERENCES ON Pilote FROM christian_soutou CASCADE CONSTRAINTS;`	*christian_soutou* ne peut plus bénéficier de la table `Pilote` pour programmer une contrainte référentielle via une clé étrangère.

2.3. Privilèges prédéfinis

- Oracle propose des privilèges prédéfinis pour faciliter la gestion des droits.
  - Le tableau suivant en décrit quelques-uns :

Nom	Privilèges
GRANT ANY PRIVILEGE	Autorisation de donner tout privilège système.
GRANT ANY OBJECT PRIVILEGE	Autorisation de donner tout privilège objet.
COMMENT ANY TABLE	Commenter une table, vue ou colonne de tout schéma.
SELECT ANY DICTIONARY	Interroger les objets du dictionnaire des données (schéma SYS).
SYSDBA	ALTER DATABASE OPEN | MOUNT | BACKUP, CREATE DATABASE, ARCHIVELOG, RECOVERY, CREATE SPFILE, RESTRICTED SESSION
SYSOPER	Idem sauf CREATE DATABASE

privilège spécialement adapté aux tâches opérationnelles (démarrage & arrêt de la base) mais sans donner la possibilité de visualiser le contenu des tables (ni les vues DBA_xxx), hormis celles qui sont accessibles à PUBLIC.

3. Politique à base de rôles

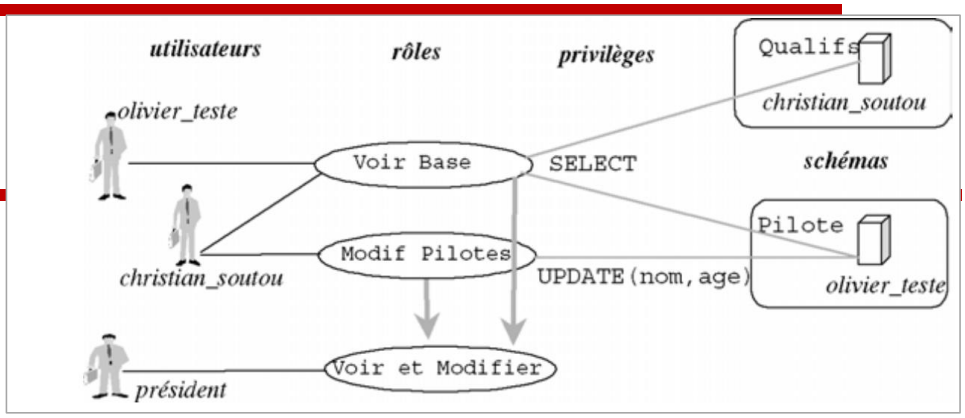
- Un rôle est un groupe nommé de privilèges qui peuvent être assignés à un utilisateur.
- Cette méthode facilite la gestion des privilèges.
- Un utilisateur peut avoir accès à plusieurs rôles et plusieurs utilisateurs peuvent recevoir le même rôle.
- Les rôles sont spécialement créés pour une application de la base de données.

3.1. Création

```
CREATE ROLE nomRôle
[ NOT IDENTIFIED | IDENTIFIED
   { BY motdePasse | USING [schéma.]paquetage | EXTERNALLY |
   GLOBALLY } ] ;
```

- *NOT IDENTIFIED indique que l'utilisation de ce rôle est autorisée sans mot de passe.*
- *IDENTIFIED signale que l'utilisateur doit être autorisé par une méthode (locale par un mot de passe, applicative par un paquetage, externe à Oracle et globale par un service d'annuaire) avant que le rôle soit activé par SET ROLE (voir plus loin).*
- *Il n'est pas possible de donner le privilège REFERENCES à un rôle.*

Exemple



- Création des rôles
  - **CREATE ROLE** *Voir_Base* NOT IDENTIFIED;
    CREATE ROLE *Modif_Pilotes* NOT IDENTIFIED;
    CREATE ROLE *Voir_et_Modifier* **NOT IDENTIFIED;**
- Attribution des privileges aux rôles
  - **GRANT** SELECT **ON** olivier_teste.Pilote **TO** Voir_Base ;
    GRANT SELECT **ON** christian_soutou.Qualifications **TO** Voir_Base ;
    GRANT UPDATE (nom,age) **ON** olivier_teste.Pilote **TO** *Modif_Pilotes*;
- Alimentation des rôles par d'autres rôles
  - **GRANT** Voir_Base, Modif_Pilotes **TO** Voir_et_Modifier ;
- Affectation des rôles aux utilisateurs
  - **GRANT** Modif_Pilotes **TO** christian_soutou;
    GRANT Voir_Base **TO** christian_soutou, olivier_teste;
    GRANT Voir_et_Modifier **TO** président;

Exemple

- Créer un rôle intitulé manger
  - CREATE ROLE manager;
  - ⇒ Role created.
- Assigner des privilèges systèmes de création de tables et de vues au rôle manger
  - GRANT create table, create view TO manager;
  - ⇒ Grant succeeded.
- Assigner le rôle aux utilisateurs MTIR et MILADI
  - GRANT manager TO MTIR, MILADI;
  - ⇒ Grant succeeded.

3.2. Activation/ désactivation d'un rôle

- Quand un utilisateur se connecte, il détient par défaut tous les privilèges qui lui ont été attribués soit directement soit via des rôles.
- Les rôles, une fois créés et alimentés, sont donc actifs par défaut.
- Durant la session (SQL*Plus ou programme), des rôles peuvent être désactivés puis réactivés par la commande SET ROLE.
- Le nombre de rôles qui peuvent être actifs en même temps est limité par le paramètre d'initialisation MAX_ENABLED_ROLES.

```
SET ROLE
{ nomRôle [IDENTIFIED BY motdePasse] [,nomRôle [IDENTIFIED BY
motdePasse]]…

   | ALL [EXCEPT nomRôle [,nomRôle]…]
   | NONE } ;
```

- ALL active tous les rôles (non identifiés) accordés à l'utilisateur qui exécute la commande. Cette activation n'est valable que dans la session courante.
- La clause EXCEPT permet d'exclure des rôles accordés à l'utilisateur (mais pas via d'autres rôles) de l'activation globale.
- NONE désactive tous les rôles dans la session courante (rôle DEFAULT inclus).

3.3. Modification d'un rôle

```
ALTER ROLE nomRôle
[ NOT IDENTIFIED | IDENTIFIED
   { BY motdePasse | USING
        [schéma.]paquetage | EXTERNALLY | GLOBALLY } ] ;
```

- Lacommande ALTER ROLE permet de changer le mode d'identification d'un rôle.
- Vous devez être propriétaire du rôle ou l'avoir reçu avec l'option WITH ADMIN OPTION, ou détenir le privilège ALTER *ANY* ROLE.
- La modification du contenu d'un rôle (ajout ou retrait de privilèges) se programme à l'aide des commandes GRANT (pour ajouter un privilège) et REVOKE (pour enlever un privilège).

3.4. Rôles prédéfinis

Nom du rôle	Commentaires
CONNECT	Se connecter (CREATE SESSION), créer des tables, vues et séquences.
RESOURCE	Créer des procédures, déclencheurs, tables et types.
DBA	Détenir tous les privilèges système avec la possibilité de les retransmettre.
EXP_FULL_DATABASE et DATAPUMP_EXP_FULL_DATABASE	Réaliser des exportations.
IMP_FULL_DATABASE et DATAPUMP_IMP_FULL_DATABASE	Réaliser des importations.
EM_EXPRESS_BASIC et EM_EXPRESS_ALL	Utiliser la console d'administration (version 12*c* Express).
OEM_ADVISOR	Régler des requêtes
SELECT_CATALOG_ROLE	Accéder à tous les objets de tout schéma (en consultation).
XDBADMIN	Accéder à XML DB Repository (voir le chapitre 11).

III- Dictionnaire des données

1.Interrogation du dictionnaire

- La démarche à suivre afin d'interroger correctement le dictionnaire des données à propos d'un objet est la suivante :
  - Trouver le nom de la vue ou des vues qui sont pertinentes à partir de la vue DICTIONARY situé au niveau le plus haut de la hiérarchie ;
  - Choisir les colonnes de la vue à sélectionner en affichant la structure de la vue (par la commande DESC) ;
  - Interroger la vue en exécutant une requête SELECT contenant les colonnes intéressantes.

La première étape peut être omise si on connaît déjà le nom de la vue (ce sera le cas pour les vues usuelles que vous aurez déjà utilisées à plusieurs reprises).

1.Interrogation du dictionnaire

- Pour savoir les privilèges système affectés à un rôle:
- select PRIVILEGE from DBA_SYS_PRIVS where grantee=nom_role' ;
  - Exemple:
  - select PRIVILEGE from DBA_SYS_PRIVS where grantee='RESOURCE' ;

```
PRIVILEGE
----------------------------------------
CREATE TRIGGER
CREATE SEQUENCE
CREATE TYPE
CREATE PROCEDURE
CREATE CLUSTER
CREATE OPERATOR
CREATE INDEXTYPE
CREATE TABLE
```


2. Informations sur les utilisateurs à partir du dictionnaire oracle

- ALL_USERS
- USER_USERS

→

USERNAME
USER_ID
CREATED

- DBA_USERS

→

USERNAME
USER_ID
PASSWORD
ACCOUNT_STATUS
LOCK_DATE
EXPIRY_DATE
DEFAULT_TABLESPACE
TEMPORARY_TABLESPACE
CREATED
PROFILE
…

•Exp:

- Select *username* from all_users;
- •Select* from dba_users where username like 'A%';

•Rq: il existe d'autres tables systèmes qui peuvent nous renseigner sur les infos user

3. Informations sur les privilèges à partir du dictionnaire Oracle

Dictionnaire Oracle

- ALL_TAB_PRIVS
- USER_TAB_PRIVS

GRANTOR
GRANTEE
TABLE_SCHEMA
TABLE_NAME
COLUMN_NAME
PRIVILEGE
GRANTABLE
HIERARCHY

- ALL_COL_PRIVS
- USER_COL_PRIVS

- ALL_ROLE_PRIVS

4. Informations sur les rôles à partir du dictionnaire Oracle

- ROLE_TAB_PRIVS

ROLE
OWNER
TABLE_NAME
COLUMN_NAME
PRIVILEGE
GRANTABLE

- ROLE_SYS_PRIVS
- DBA_ROLES
- DBA_ROLE_PRIVS
- USER_ROLE_PRIVS
- ROLE_ROLE_PRIVS

USERNAME
GRANTED_ROLE
ADMIN_OPTION
DEFAULT_ROLE
OS_GRANTED

Exemples

Commande SQL	Résultat
DESC **DICTIONARY**	Nom NULL ? Type ---------- ------- ---------- TABLE_NAME VARCHAR2(30) COMMENTS VARCHAR2(4000)
SELECT * FROM **DICTIONARY** WHERE table_name LIKE '%SEQUENCE%';	TABLE_NAME COMMENTS ---------- ---------- ALL_SEQUENCES Description of SEQUENCEs accessible to the user DBA_SEQUENCES Description of all SEQUENCEs in the database USER_SEQUENCES Description of the user's own SEQUENCEs

Principales vues du dictionnaire

Nature de l'objet	Vues
Objets (au sens général)	USER_OBJECTS : objets appartenant à l'utilisateur (synonyme OBJ). USER_ERRORS-: erreurs après compilation des objets PL/SQL stockés (procédures, fonctions, paquetages, déclencheurs). USER_STORED_SETTINGS-: paramètres des objets PL/SQL stockés. USER_SOURCE : source des objets PL/SQL stockés.
Tables	USER_TABLES : description des tables relationnelles de l'utilisateur (synonyme TABS). USER_ALL_TABLES : description des tables relationnelles et objets de l'utilisateur.
Colonnes	USER_TAB_COLUMNS : colonnes des tables et vues (synonyme COLS). USER_UNUSED_COL_TABS : colonnes éliminées des tables.
Contraintes	• **USER_CONSTRAINTS** : définition des contraintes de tables. • **USER_CONS_COLUMNS** : composition des contraintes (colonnes).

Principales vues du dictionnaire

Nature de l'objet	Vues
Utilisateurs	`USER_USERS` : caractéristiques de l'utilisateur courant. `DBA_USERS` et `ALL_USERS` : caractéristiques de tous les utilisateurs.
Privilèges	`USER_TAB_GRANTS` : liste des autorisations sur les tables et les vues pour lesquelles l'utilisateur est le propriétaire, ou ayant donné ou reçu l'autorisation. `USER_TAB_GRANTS_MADE` : liste des autorisations sur les objets appartenant à l'utilisateur. `USER_COL_GRANTS` : colonnes autorisées à l'accès `USER_COL_GRANTS_MADE` : liste des autorisations sur les colonnes des tables ou des vues appartenant à l'utilisateur. `USER_COL_PRIVS_MADE` : informations sur les colonnes pour lesquelles l'utilisateur est propriétaire ou bénéficiaire. `USER_TAB_GRANTS_RECD` : liste des objets pour lesquels l'utilisateur a reçu une autorisation. `USER_COL_PRIVS_RECD` : informations sur les colonnes pour lesquelles l'utilisateur a reçu une autorisation.
Rôles	`DBA_ROLES` : tous les rôles existants. `DBA_ROLE_PRIVS` : rôles donnés aux utilisateurs et aux autres rôles. `USER_ROLE_PRIVS` : rôles donnés à l'utilisateur. `ROLE_ROLE_PRIVS` : rôles donnés aux autres rôles. `ROLE_SYS_PRIVS` : privilèges système donnés aux rôles. `ROLE_TAB_PRIVS` : privilèges sur les tables donnés aux rôles. `SESSION_ROLES` : rôles actifs à un instant *t*.

Exemples

- Exemple de consultation du dictionnaire de données

```
SELECT OBJECT_NAME, OBJECT_TYPE, CREATED FROM USER_OBJECTS;

OBJECT_NAME                    OBJECT_TYPE         CREATED
------------------------------ ------------------- --------
ACCES_SECURISE                 PACKAGE             03/09/03
ACCES_SECURISE                 PACKAGE BODY        03/09/03
AFFICHEAVIONS                  PROCEDURE           03/09/03
Compagnies                     JAVA CLASS          17/08/03
EFFECTIFSHEURE                 FUNCTION            16/09/03
ESPIONCONNEXION                TRIGGER             16/09/03
PILOTE                         TABLE               18/09/03
PK_PILOTE                      INDEX               18/09/03
VUEMULTICOMPPIL                VIEW                14/09/03
```

Références

Livres :

1. G. Gardarin, Bases de données , Eyrolles, 2003.
2. C.SOUTOU, SQL pour Oracle, Edition3

Supports de cours :

1. J.Y. Antoine, Administration des bases de données
2. F. FELHI, Administrer la sécurité utilisateur: Gestion des utilisateurs et de leurs droits.

Autres ressources et Liens utiles:

- **Oracle Database 10*g* : Administration Workshop I**
- https://www.supinfo.com/cours/1ORC/chapitres/01-introduction-au-sql

Plan du module

- **Partie 1- I. Introduction aux SGBDs**
  - Chapitre 1: Présentation des SGBDs
  - Chapitre 2: Définition et Evolution des données
  - Chapitre 3: Contrôle des données
  - **Chapitre 4: Gestion des objets utilisateurs**
    - **(Vues, séquences et Index)**
- **Partie 2- II. Langage procédural: PL/SQL**
- **Partie 3- III. Gestion des Transactions**