Administration des bases de données

Gestion des utilisateurs

Les principales tâches d’un administrateur d’une base de données

- * Planification des ressources de mémorisation des données
- * Organisation des structures logiques et physiques des données
- * Création et gestion des utilisateurs et de leurs droits d’accès (privilèges)
- * Gestion de la sécurité du système : gestion des accès concurrents

La gestion des utilisateurs

- comment créer et configurer un utilisateur de base de données ou un compte avec lequel vous pourrez vous connecter et effectuer des actions sur la base de données en fonctions des droits qui vous serons alloués.
- Voici les différentes étapes qui seront nécessaire à la création d'un utilisateur Oracle :
  - Choisir un nom d'utilisateur
  - Choisir une méthode d'authentification
  - Choisir les TABLESPACEs que l'utilisateur pourra utiliser
  - Définir les quotas sur chaque TABLESPACEs
  - Définir les TABLESPACEs par défaut de l'utilisateur
  - Créer l'utilisateur
  - Assigner les rôles et privilèges à l'utilisateur

Rôles et privilèges

Introduction

- **Les Rôles et privilèges sont définis pour sécuriser l'accès aux données de la base .**

- Ces concepts sont mis en œuvre pour protéger les données en accordant (ou retirant) des privilèges a un utilisateur ou un groupe d'utilisateurs

- **Un rôle** est un **regroupement de privilèges.** Une fois créé il peut être assigné à **un utilisateur** ou à un **autre rôle**

Introduction

Les privilèges sont de deux types:

- **Les privilèges de niveau système:**

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

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

Introduction

- **Les privilèges de niveau objet:**

  Qui permettent les manipulations sur des objets spécifiques.

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

Création d’un utilisateur

- Ordre SQL simple pour créer un utilisateur:

```
CREATE  USER    login IDENTIFIED BY password;
```

- Cette instruction va créer un utilisateur dont le nom est **login**, et mot de passe et **password**.

Exemple

```
CREATE USER user1
        IDENTIFIED BY MDP
        DEFAULT TABLESPACE USERS
        QUOTA 10M ON USERS
        TEMPORARY TABLESPACE TEMP
```

Etapes de création

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

Exemple

- Cet ordre crée un utilisateur nommé ' **user1** ', et dont le mot de passe est ' *MDP* '.
- user1 créera par défaut ses objets dans la partition ‘**users**' (jusqu'à concurrence de '10Mo').
- Son espace de travail temporaire (utilisé en interne par Oracle) sera la partition '**temp**'.
- Lors de sa connexion, user1 bénéficiera d'une session dont les caractéristiques sont définies par le profil 'profil_ user1 '.

Les privilèges

Privilège

Nous venons de voir comment créer un compte utilisateur pour user1. 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.

En ce qui concerne notre utilisateur, malgré la création de son compte, il ne peut toujours pas se connecter à la base de données. **Il faut pour cela que le DBA lui accorde le privilège CREATE SESSION**

Privilège

Un privilège est le droit d'exécuter un type d'instruction SQL spécifique.
Quelques exemples de privilèges :
->le droit de se connecter à une base de données (autrement dit ouvrir une session),
->le droit de créer une table,
->le droit de sélectionner des lignes dans une table.

Les privilèges d'une base de données Oracle peuvent être répartis en deux catégories distinctes :
->les privilèges système,
->les privilèges objets.

Assigner des privilèges système à un utilisateur

- Lorsqu'un utilisateur est créé avec l'instruction CREATE USER, il ne dispose encore d'aucun droit car aucun privilège ne lui a encore été assigné.
- Il faut donc lui assigner les privilèges nécessaires .
- Il doit pouvoir se connecter, créer des tables, des vues, des séquences.

Assigner des privilèges système à un utilisateur

- Pour lui assigner ces privilèges de niveau système il faut utiliser l'instruction **GRANT** dont voici la syntaxe:

GRANT [systeme_privilege||rôle||ALL PRIVILEGES] to [user ||PUBLIC||rôle]

- **systeme_privilege** représente un privilège système.
- **rôle** représente un rôle préalablement créé.
- **ALL PRIVILEGES représente tous les privilèges système**
- **user** représente le nom de l'utilisateur qui doit bénéficier du privilège.
- **PUBLIC** assigne le privilège à tous les utilisateurs.

Assigner des privilèges système à un utilisateur

- Pour que l'utilisateur puisse simplement se connecter à la base, il doit bénéficier du privilège système **CREATE SESSION**

  GRANT CREATE SESSION TO nom_utilisateur ;

- Ensuite il faut lui assigner des droits de création de table:

  GRANT CREATE TABLE TO nom_utilisateur ;

Assigner des privilèges système à un utilisateur

- Puis les droits de création de vues:

  GRANT CREATE VIEW TO nom_utilisateur ;

- L'ensemble de ces privilèges peuvent être assignés au sein d'une même commande

  GRANT CREATE SESSION ,
  CREATE TABLE ,
  CREATE VIEW TO nom_utilisateur ;

Assigner des privilèges objet à un utilisateur

- On utilise aussi la commande GRANT:

GRANT [object_privilege||ALL PRIVILEGES] (column) ON schema . Object TO [user ||role ||PUBLIC]

- **object_privilege** représente un privilège objet.
- **role** représente un rôle préalablement créé.
- **ALL PRIVILEGES** représente tous les privilèges assignés à l'exécuteur de l'instruction.
- **column** représente le nom de colonne d'une table.
- **schema** représente le nom d'un schéma.
- **object** représente le nom d'un objet du schéma.

Assigner des privilèges objet à un utilisateur

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

```
GRANT
    SELECT ,
    INSERT ,
    UPDATE ,
    DELETE
    ON SYS.EMP(ou une autre table)
    TO nom_utilisateur ;
```

Assigner des privilèges objet à un utilisateur

- Une liste de colonnes peut être indiquée dans l'instruction afin de restreindre davantage les droits sur une table.

  GRANT UPDATE (colonnes1, colonne2) **ON** SYS.EMP **TO** nom_utilisateur ;

- L'utilisateur peut modifier la table SYSTEM.EMP mais uniquement les colonnes JOB et HIREDATE.

Assigner des privilèges objet à un utilisateur

- Pour pouvoir mettre à jour ou supprimer des lignes d'une table, les privilèges **UPDATE ET DELETE** ne suffisent pas. Le privilège **SELECT** est nécessaire.

- Un utilisateur munis des droits **DBA** ne pourra pas accorder de privilèges sur un objet qui ne lui appartient pas

Assigner des privilèges objet à un utilisateur

Principes généraux appliqués aux privilèges:

- Un utilisateur possède automatiquement tous les privilèges sur un objet qui lui appartient.

- Un utilisateur ne peut pas donner plus de privilèges qu'il n'en a reçu.

Assigner des privilèges à un utilisateur

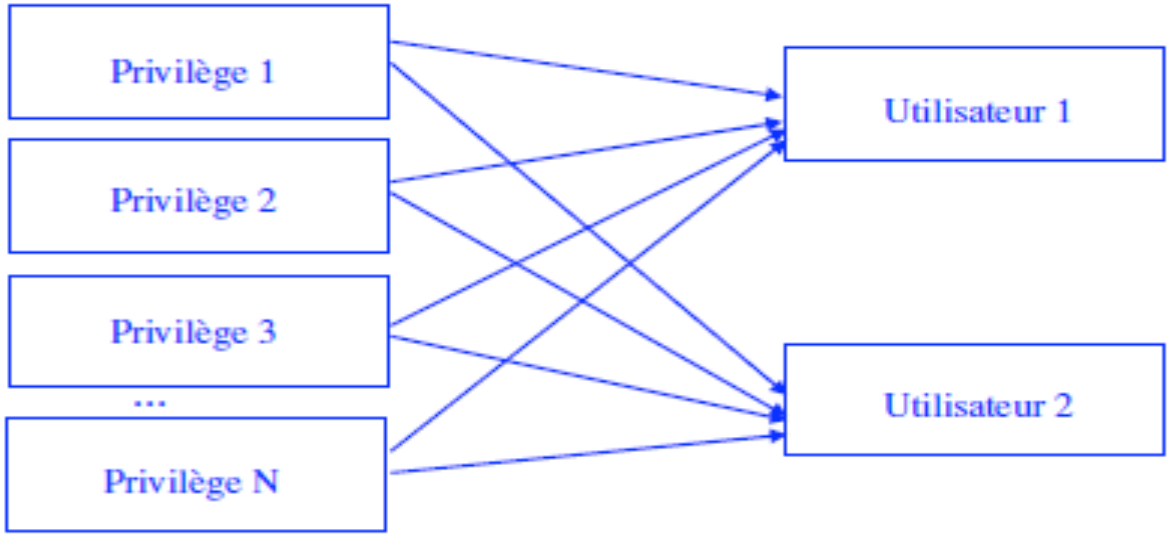
- L'instruction **GRANT** permet d'assigner un ou plusieurs privilèges système ou objet.
- Cependant, lorsque la liste des privilèges est importante, cette manière de procéder s'avère rapidement fastidieuse et répétitive.
- C'est pourquoi il est souhaitable de pouvoir regrouper des privilèges identiques dans un même ensemble

Les rôles

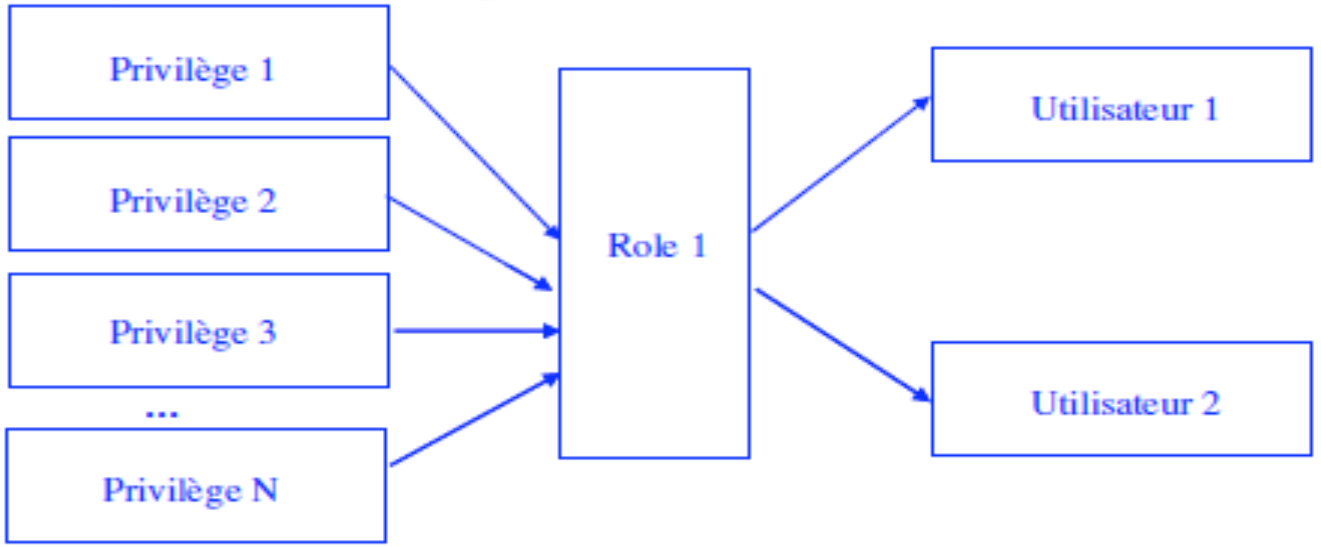
Les Rôles

- Regroupement de privilèges pour des familles d'utilisateurs
- Facilitent la gestion des autorisations des privilèges objet en évitant les ordres GRANT
- Un rôle par défaut est donné à un utilisateur
- Un utilisateur peut posséder plusieurs rôles mais n'est connecté qu'avec un seul à la fois
- On peut donner un mot de passe pour certains rôles
- Un rôle facilite la gestion des privilèges
- pour des raisons de sécurité, un **mot de passe peut être** assigné à un rôle

Assignation de privilèges aux utilisateurs : SANS ROLES



Assignation de privilèges aux utilisateurs : VIA UN ROLE



Création d'un rôle

- **Création d'un rôle**
- A sa création, un rôle ne contient aucun privilège
- **Exemple**
  - sql> CREATE ROLE rl_etudiant ;
  - sql> CREATE ROLE rl_admin_backup;
  - sql> CREATE ROLE rl_admin_secu IDENTIFIED BY secu_pass ;

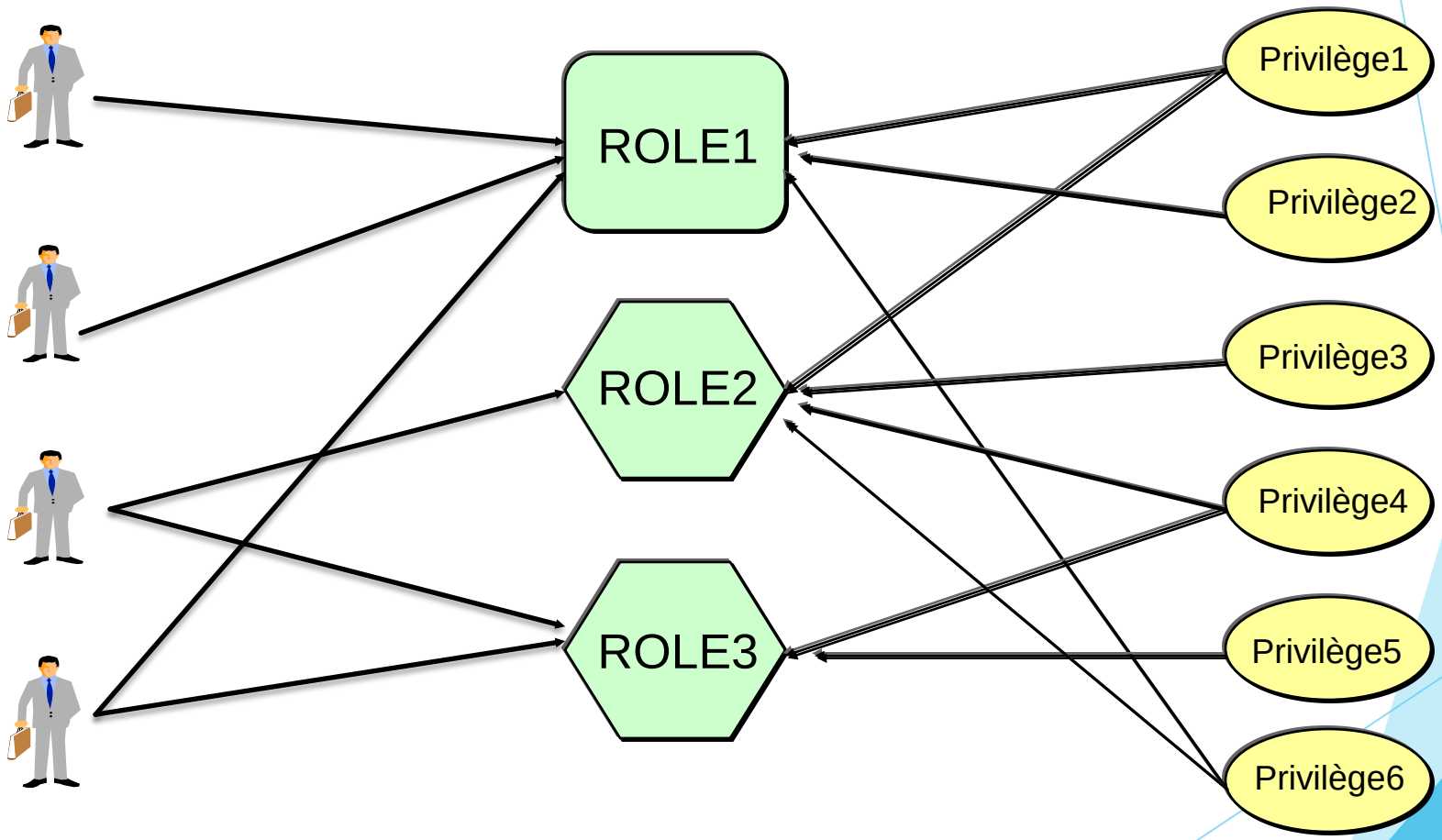
Création des rôles

- L’ ensemble qui regroupe plusieurs privilèges s’appelle un rôle, et se crée avec l’instruction CREATE ROLE

  CREATE ROLE role [**NOT IDENTIFIED**| | **IDENTIFIED BY password**];

- **role** représente le nom du rôle
- **NOT IDENTIFIED** (défaut) indique qu'aucun mot de passe n'est nécessaire pour activer le rôle
- **IDENTIFIED BY password** indique qu'un mot de passe est nécessaire pour activer le rôle

Les Rôles :


ROLE1
ROLE2
ROLE3
Privilège1
Privilège2
Privilège3
Privilège4
Privilège5
Privilège6

Manipulation des rôles : Exemples

CREATE ROLE secretariat_info ;

GRANT SELECT,UPDATE (adr,tel)
ON ens_info TO secretariat_info;
GRANT SELECT,INSERT,UPDATE
ON etud_info TO secretariat_info;
GRANT SELECT,INSERT
ON cours_info TO secretariat_info;

GRANT secretariat_info TO
laurent, thomas, corinne ;

Assigner des privilèges à un rôle

- Lorsque le rôle est créé, il ne contient rien et il faut l'alimenter à l'aide d'instructions GRANT.

  CREATE ROLE comptabilite ;

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

Les rôles

- **Pour voir** les privilèges système assignés au rôle:

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

Les rôles

- La liste des rôles assignés à l'utilisateur au cours de sa session est visible via la vue **SESSION_ROLES:**

```
select *  from SESSION_ROLES ;
```

- La liste des privilèges assignés à l'utilisateur au cours de sa session est visible via la vue **SESSION_PRIVS:**

```
select *  from SESSION_PRIVS ;
```

Les rôles

- Un rôle peut être supprimé en utilisant l'instruction **DROP ROLE**
- **DROP ROLE** nom_role ;
- Le rôle spécifié ainsi que tous les privilèges qui lui sont associés sont supprimés de la base et également retiré à tous les utilisateurs qui en bénéficiaient

Retirer des privilèges

Retirer des privilèges système

- Les privilèges système qui ont été assignés à des utilisateurs ou à des rôles peuvent être retirés avec l'instruction **REVOKE:**

  REVOKE [**systeme_privilege** || **rôle** || **ALL PRIVILEGES**] FROM [**user** || **PUBLIC**||**rôle**].

- Les arguments sont identiques à ceux décrits pour l'instruction GRANT.

- **Retirer des privilèges à un utilisateur ne supprime pas son schéma ni les objets qu'il contient**

Retirer des privilèges objet

- Les privilèges objet qui ont été assignés à des utilisateurs ou à des rôles peuvent être retirés avec l'instruction **REVOKE:**

REVOKE [object_privilege||ALL PRIVILEGES] (column) ON schema . Object FROM[user ||role ||PUBLIC]