SQL

Ch 10 – DCL
Utilisateurs, privilèges, rôles

I. Introduction

Une fois les objets de la base de données créés par l'administrateur de la base de données (tables, etc), ce dernier va devoir définir

- dans un premier temps, des comptes d'**utilisateurs**
- dans un second temps, des **privilèges** associés à ces comptes d'utilisateurs, nécessaires à la réalisation de leurs tâches (créer de nouvelles tables, ajouter des lignes à une table, etc.),
  - soit de manière individuelle,
  - soit en définissant des rôles au sein de l'organisation, et en associant les utilisateurs à ces rôles.

Documentation Oracle : http://www.oracle.com/technology/documentation/index.html
Sur la version Oracle 10.2 : http://www.oracle.com/pls/db102/homepage

II. Les utilisateurs (« users »)

Les utilisateurs correspondent aux comptes (login et mot de passe) permettant de demander une connexion au SGBD (une personne ou une application).

A. Créer un utilisateur : CREATE USER

Syntaxe :

```
CREATE USER nomUtilisateur
    IDENTIFIED BY motDePasse;
```

(Pour créer un nouveau compte d'utilisateur, il faut avoir le privilège « CREATE USER »)

Exemple 1 : créer l'utilisateur « tim » avec le mot de passe « soleil »

```
CREATE USER tim IDENTIFIED BY soleil ;
```

Exemple 2 : créer l'utilisateur « lea » avec le mot de passe « secret » et définir des caractéristiques pour le stockage de ses données et l'utilisation du SGBD

```
CREATE USER lea IDENTIFIED BY secret
DEFAULT TABLESPACE donnees
TEMPORARY TABLESPACE temporaire
QUOTA 100Mo ON donnees
PROFILE profil_base;
```

(On associe à Lea un espace par défaut pour ses données (tables), un espace « temporaire » pour les tris et tables temporaires, un quota d'utilisation de 100Mo sur l'espace « données » ; le profil définit une configuration d'utilisation des ressources du système, « profil_base »).

B. Modifier un utilisateur : ALTER USER

Syntaxe :

```
ALTER USER nomUtilisateur
    IDENTIFIED BY motDePasse
    [ REPLACE ancientMotDePasse ];
```

Exemple : on change le mot de passe de « tim »

```
ALTER USER tim IDENTIFIED BY neptune REPLACE soleil ;
```

C. Supprimer un utilisateur : DROP USER

Syntaxe :

```
DROP USER nomUtilisateur
    [ CASCADE ]
;
```

Exemple : supprimer l'utilisateur « tim » et tous ses objets (tables, etc.) » :

```
DROP USER tim CASCADE;
```

III.Les Privilèges (« privileges »)

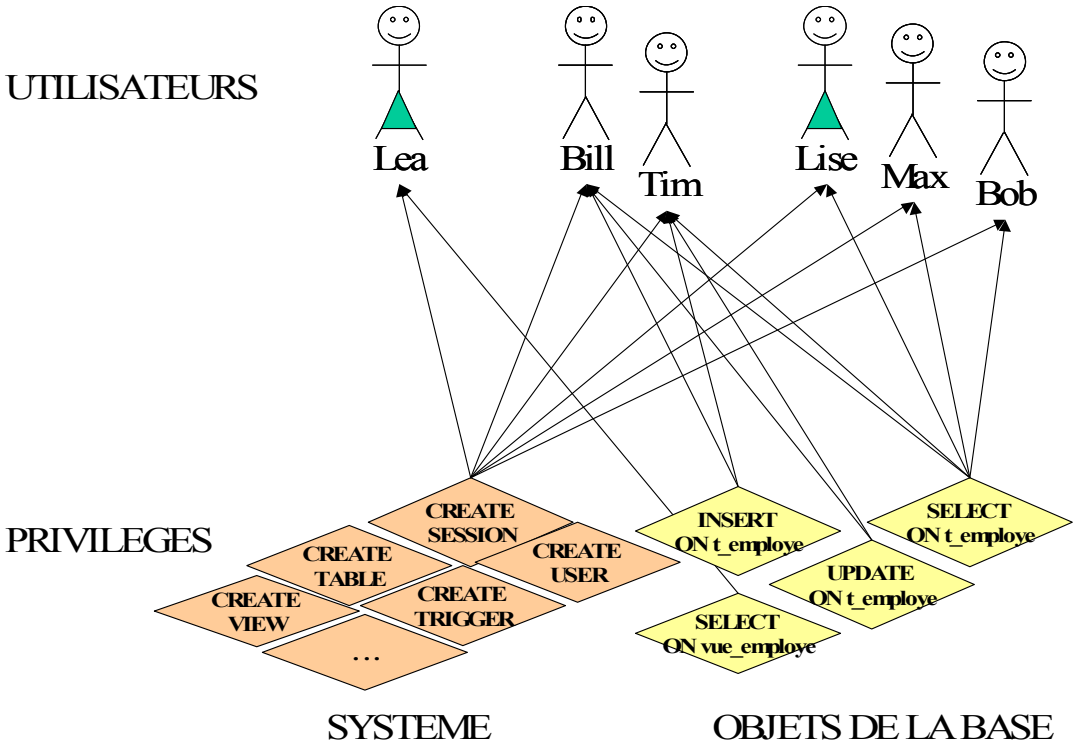
Un privilège est le droit d'exécuter un type d'instruction SQL. Quelques exemples de privilèges :

- le droit de se connecter à une base de données (= ouvrir une session) (instruction CONNECT),
- le droit de créer une table (instruction CREATE TABLE),
- le droit de sélectionner des lignes dans une table (instruction SELECT).

Les privilèges d'une base de données Oracle sont répartis en deux catégories:

- les **privilèges « système »** : concernent des actions globales sur le SGBD (se connecter, créer des tables, etc.)
- les **privilèges liés aux objets**. : relatifs aux objets de la base (table, vue, etc) qu'un utilisateur possède (en est le propriétaire, Owner).

Pour se connecter au SGBD, il faut avoir le privilège 'CREATE SESSION'



A. Attribuer un privilège : GRANT

C'est un utilisateur administrateur (DataBase Administrateur, le « superutilisateur ») qui attribue des privilèges « système » et c'est le possesseur des objets qui attribue les privilèges liés aux objets dont il est le propriétaire.
L'instruction GRANT est utilisée pour attribuer un privilège.

Syntaxe :

```
GRANT typeDePrivilege
    [ ON objetDeLaBase ]
    TO utilisateurs
    [WITH GRANT OPTION]
```

- `typeDePrivilege` :
  - système : CREATE SESSION, CREATE TABLE, DROP TABLE, etc..
  - objets : SELECT, INSERT, UPDATE, UPDATE (col1, col2), DELETE, ALTER, INDEX, ALL.
- `objetDeLaBase` : nom de la table sur laquelle porte le privilège
- `utilisateurs` :
  - nom du ou des utilisateurs auquel(s) sont accordés les privilèges
  - le mot clé PUBLIC peut être utilisé pour désigner tous les utilisateurs.
- `WITH GRANT OPTION` : on transmet le droit de donner ces privilèges

Exemple 1 : accorder le privilège de se connecter au SGBD

```
GRANT CREATE SESSION TO bob, tim, lise ;
```

Exemple 2 : accorder le privilège d'utiliser l'ordre SQL SELECT sur la table « t_employe »

```
GRANT SELECT ON t_employe TO bill, max, bob ;
```

Exemple 3 : accorder le privilège d'utiliser l'ordre SQL INSERT sur la table « t_employe » à Bill et Tim et on transmet à Bill et Tim le privilège d'accorder à leur tour ce privilège à d'autres :

```
GRANT INSERT ON t_employe TO bill, tim
      WITH GRANT OPTION;
```

L'utilisateur Administrateur de la base de données (DBA, DataBase Administrator) peut suivre l'attribution des privilèges en utilisant les tables systèmes suivantes :

- *DBA_TAB_GRANTS : Tous les droits sont accordés sur tous les objets de la base.*
- *DBA_USERS : Informations sur tous les utilisateurs de la base.*

B. Retirer un privilège – REVOKE

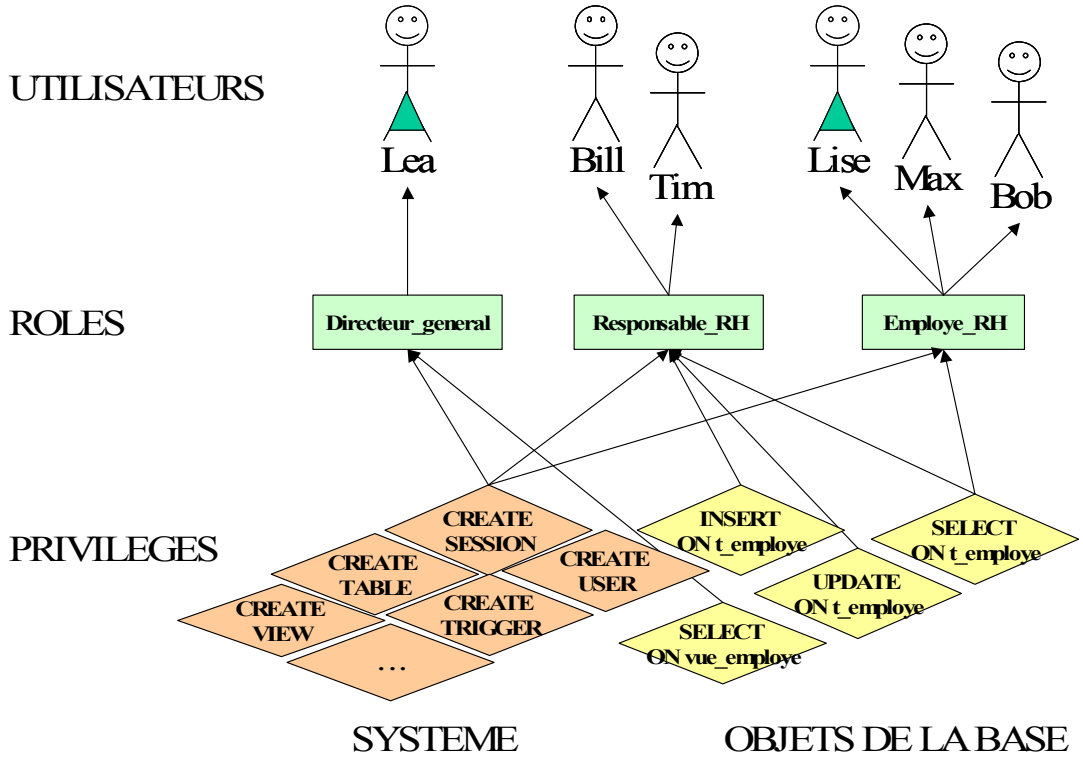
L'utilisateur ayant accordé un privilège à un autre utilisateur sur les objets qu'il possède peut les lui retirer en utilisant la commande REVOKE :

Syntaxe générale :

```
REVOKE typeDePrivilege
    ON objetDeLaBase
    FROM utilisateur;
```

IV. Les rôles (« roles »)

Les rôles permettent le regroupement d'un certain nombre de privilèges communs à un groupe d'utilisateurs. Ils facilitent ainsi l'administration des privilèges.



A. Créer un rôle : CREATE ROLE

Syntaxe :

```
CREATE ROLE nomRole;
```

Exemple : création d'un rôle nommé « responsable_RH» :

```
CREATE ROLE responsable_RH;
```

Exemple : création d'un rôle nommé « employe_RH» :

```
CREATE ROLE employe_RH;
```

(Pour créer un rôle, l'utilisateur doit avoir le privilège « CREATE ROLE »).

B. Attribuer et retirer des privilèges à un rôle : GRANT et REVOKE

De la même manière qu'on a attribué ou rétiré des privilèges à un utilisateur, on pourra faire de même pour un rôle.

Exemple : attribution des droits pour un employé des RH (Ressources Humaines, service du personnel):

```
GRANT CREATE SESSION TO employe_RH ;
GRANT SELECT ON t_employe TO employe_RH ;
```

Exemple : attribution des droits pour un responsable des RH (Ressources Humaines, service du personnel):

```
GRANT CREATE SESSION TO responsable_RH ;
GRANT SELECT ON t_employe TO responsable _RH ;
GRANT INSERT ON t_employe TO responsable _RH ;
GRANT UPDATE ON t_employe TO responsable _RH ;
```

Exemple : supprimer un privilège (tous les utilisateurs associés à ce rôle voient leurs privilèges modifiés):

```
REVOKE UPDATE ON t_employe FROM responsable _RH ;
```

C. Associer des utilisateurs à un rôle : GRANT et REVOKE

De la même manière qu'on a attribué ou retiré des privilèges à un utilisateur, on pourra faire de même pour un rôle.

Exemple : attribuer aux utilisateurs bill et tim les droits relatifs au rôle 'responsable_RH' :

```
GRANT responsable_RH TO bill, tim;
```

Exemple : retirer aux utilisateurs bill et tim les droits relatifs au rôle 'responsable_RH' :

```
REVOKE responsable_RH FROM bill, tim;
```

D. Supprimer un rôle

La suppression d'un rôle entraîne la suppression des droits qui lui avait été attribués ; cette suppression de privilèges s'applique en cascade aux utilisateurs qui y étaient liés.

Exemple : Supprimer le rôle 'responsable_RH' :

```
DROP ROLE responsable_RH;
```

E. Rôles prédéfinis

Certains rôles sont pré-définis dans Oracle :

- **CONNECT** : Autorise la connexion à une base Oracle ainsi qu'un certain nombre d'autres actions (privilèges associés CREATE TABLE, CREATE VIEW, CREATE SESSION, etc.)
- **RESOURCE** : Permet, en plus de la création de table et de vue, l'utilisation de trigger et de procédure
- **DBA** : Ce rôle regroupe tous les privilèges système pour la gestion des utilisateurs et de leurs tables.

Chacun de vos comptes ont été créés de la manière suivante :

```
create user votreLogin identified by votreMotDePasse;
grant connect, resource to votreLogin;
```

On aurait pu utiliser la syntaxe suivante :

```
create role etudiantAnnee1;
grant connect, resource to etudiantAnnee1;

create user votreLogin identified by votreMotDePasse;
grant etudiantAnnee1 to votreLogin ;
```

De cette manière toute modification 'extension ou restriction appliquée au rôle « etudiantAnnee1 » s'appliquerait automatiquement à tous les étudiants bénéficiant de ce rôle.