Administration base de données ORACLE

Rôles et privilèges

Niveau:
DUT GI S3

Introduction

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

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 le mot de passe est **password**.

Les privilèges

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 (à l'exception de SELECT ANY DICTIONARY)

- 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 SCOTT.

```
GRANT
        SELECT ,
        INSERT ,
        UPDATE ,
        DELETE
        ON SCOTT.EMP
        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 (JOB, HIREDATE) **ON** SCOTT.EMP **TO** nom_utilisateur ;

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

Assigner des privilèges objet à un utilisateur

Attention:

- **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.
- S'il n'a pas reçu le privilège avec l'option **WITH GRANT OPTION**, un utilisateur ne peux pas assigner à son tour ce même privilège.

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

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

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

- **Trois rôles existent en standard:**

  - CONNECT

  - RESOURCE

  - DBA

- **Pour voir l**es 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]

Administration base de données ORACLE

Gestion des utilisateurs

Création des utilisateurs

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 chaques TABLESPACEs
- Définir les TABLESPACEs par défaut de l'utilisateur
- Créer l'utilisateur
- Assigner les rôles et les privilèges à l'utilisateur

Introduction

- Un **schéma** est une collection (ou un ensemble) nommé d'objets tels que des tables, vues, clusters, fonctions, procédures et packages associés à un utilisateur précis.
- Quand un utilisateur de base de données est crée; son schéma est automatiquement crée.
- Un utilisateur ne pourra alors être associé qu'à un seul schéma et réciproquement.

Introduction

- Un utilisateur de base de données va correspondre à un login qui aura reçu certains privilèges.
- Cet utilisateur sera stocké dans le dictionnaire de données et disposera d'un espace de stockage pour ses objets qui seront alors stockés dans son schéma.
- En Oracle on pourra assimiler un utilisateur avec son schéma.

Choix du nom de l'utilisateur

- La première chose à faire pour créer un nouvel utilisateur va être de définir un login.

- Afin d'éviter d'avoir trop de problèmes lors de l'ajout de nouveaux utilisateurs, il est fortement recommandé de mettre une stratégie de nommage en place.

- Par exemple tout les noms d'utilisateur devront être composé des 6 premières lettres de leur nom, d'un "_" et de la première lettre de leur prénom.

Choix du nom de l'utilisateur

Il convient ensuite de connaitre les limitations et règles de nommage à respecter:

- Taille maximale 30 caractères.
- Ne devra contenir que des lettres de [a-z] et des chiffres [0-9]. Tout les caractères accentués ou autres sont à éviter.
- Vous pourrez également utiliser les symboles #, $, _.
- Le login devra commencer par une lettre. Si vous désirez utiliser des logins composé uniquement de chiffres vous devrez alors entourer votre login entre des ".

Choisir la méthode d'authentification de l'utilisateur

Afin d'authentifier un utilisateur et de définir les actions que celui-ci sera en mesure d'effectuer sur la base de données, le serveur Oracle doit pouvoir vérifier les accès de l'utilisateur lorsque celui-ci se connecte.

Il existe différents type d'authentification :

- Authentification par la base de données.
- Authentification par le système d'exploitation.
- Authentification par le réseau.

Authentification par la base de données

- C'est le mode par défaut.

- Pour créer un utilisateur authentifié par la base de données, il faut utiliser la clause IDENTIFIED BY

CREATE USER scott IDENTIFIED **BY** tiger
DEFAULT tablespace USERS
quota unlimited on USERS
[PASSWORD EXPIRE];

Modification d'un utilisateur

- Pour changer le mot de passe d'un utilisateur:

  ALTER USER < login de l'utilisateur > IDENTIFIED **BY** < nouveau mot de passe >

- Pour modifier le statut d'un utilisateur:

```
-- Verrouillage du compte
ALTER USER scott ACCOUNT LOCK;
-- Activation du compte
ALTER USER scott ACCOUNT UNLOCK;
```

Suppression d'un utilisateur

- Pour supprimer un utilisateur:

DROP USER login [CASCADE];

- **CASCADE** pour supprimer tous les objets dans le schéma de l'utilisateur avant de supprimer l'utilisateur.

- Si le schéma de l'utilisateur ne contient aucun objet, on peut supprimer cet utilisateur sans utiliser **CASCADE**.