Gestion des utilisateurs et de leurs privilèges

I.1 Gestion des utilisateurs

- Pour accéder à un SGBD, il faut disposer d'un compte donc d'un mot de passe et un ensemble
de droits identifiés par le SGBD.
- Seul un administrateur peut créer des utilisateurs.

CONNECT, RESSOURCE et DBA défissent des classes d'utilisateurs (appelées rôles) avec des droits
prédéfinis :
a) **CONNECT** : droits de connexion et d'exécution d'ordre LMD sur des données sur lesquelles
une autorisation a été attribuée au préalable. Droits aussi à la création de vues et synonymes sur
les objets autorisés. Par contre ne donne pas les droits de créer des tables, index ou d'autres types d'objets.
b) **RESOURCE** : (ie. création de ressources) Donne les droits de CONNECT plus les droits d'exécution des ordres LDD. Autrement dit, les droits :
a. de 'CONNECT',
b. de création de tables, index, et de regroupement (clusters),
c. d'attribution et restitution de privilèges sur les tables, index et regroupement à d'autres utilisateurs.
c) **DBA** : (DataBase Administrator) donne tous les droits. Autrement dit, les droits :
a. de 'CONNECT',
b. de 'RESOURCE',
c. d'accès à toutes les données de tous les utilisateurs,
d. de création et de suppression d'utilisateurs et de droits,
e. d'exécuter toutes les opérations d'administration sur les partitions, de sauvegarde, de restauration, etc .

Gestion des droits des utilisateurs sur les objets

- Toute table n'est initialement accessible que par l'utilisateur qui la créée.
- Le propriétaire d'un objet telle qu'une table peut accordé des droits sur cet objet (table) à d'autres utilisateurs.

La syntaxe de la commande est la suivante :

```
GRANT <droit1>, <droit2>, …
ON <Objet>
TO <user1>, <user2>, …
[WITH GRANT OPTION]; ← Octroi le droit d'accorder ce droit à d'autres utilisateurs
```

La commande inverse de l'attribution des droits a la syntaxe suivante :

```
REVOKE <droit1>, <droit2>, …
ON <Objet>
FROM <user1>, <user2>, …
```

Un droit ne peut être retiré que par l'utilisateur qui l'a accordé ou par le DBA.
Remarque : on peut utiliser les options
- ALL pour désigner tous les droits,
- PUBLIC pour désigner tous les utilisateurs.

I.2.1 Utilisation des rôles

Définition : 'Rôle'

Un rôle est un objet auquel on peut donner ou retirer des privilèges. On peut le voir comme un nom de
groupe décrivant un ensemble de privilèges.
Caractéristiques :
- On peut attribuer ou retirer des privilèges à un rôle comme pour un utilisateur
- Le rôle est attribué à un utilisateur comme s'il s'agissait d'un privilège élémentaire. Cet utilisateur bénéficiera des privilèges contenus dans le rôle.
But d'un rôle :
- Réduire la complexité de l'administration des privilèges
- Gestion dynamique des privilèges
- Sélection des privilèges actifs.
La syntaxe de création de rôles est la suivante :
CREATE ROLE <rôle> ; *Il faut avoir le privilège create role*

II Manipulation des données à travers des vues

Définition :
- C'est une perception particulière des données d'une ou de plusieurs tables,
- Elle est stockée sous forme de requête de sélection,
La vue permet de :
- manipuler les données autrement qu'à travers les tables définies dans la base.
- restreindre l'accès à certaines colonnes ou certaines lignes d'une table en autorisant un ou plusieurs utilisateurs à manipuler cette table qu'à travers une vue
- simplifier la tâche de l'utilisateur en le déchargeant de la formulation de requêtes complexes.
Remarque : On manipule une vue comme on manipule une table (voir plus loin)

. *II.2 Suppression d'une vue*

La syntaxe est la suivante :
DROP VIEW <Nom_vue> ;
Exemple 11 :
DROP VIEW CLIENTS_ALGER ;
Note : Supprime uniquement sa définition du dictionnaire de données de la base.

II.3 Mise à jour de vue

Les commandes INSERT, DELETE et UPDATE ne s'appliques pas à une vue si :
1. la vue est définie à partir de plus d'une table,
2. une expression ou une fonction agrégat sont utilisées dans la définition de la vue (exp. AVG)

II.4 Consultation d'une table

La consultation d'une vue se fait de la même façon qu'une table.

III Encore plus loin dans le contrôle de données : Commande 'AUDIT'

Certains SGBD fournissent des moyens pour vérifier la manière dont est utilisée la BD. Autrement dit,
si on veut vérifier si la gestion des droits qu'on a mis en place correspond à ce qui est attendu, ORACLE
par exemple fournit la commande « AUDIT » de contrôle d'accès.
Dans cette partie, nous montrons les possibilités de contrôle offertes par une telle commande.
L'opération d' « AUDIT » permet de connaitre par exemple :
- le nombre d'utilisateurs qui se sont connectés durant une période de la journée,

- les tables que les utilisateurs ont manipulés, ainsi que le type d'utilisation des tables (lecture, écriture)
- les terminaux à partir desquels il y a eu des tentatives de connexions non réussies.

Ces informations permettront ensuite au DBA :
- d'identifier les traitements consommateurs de charge : ce qui lui permettra d'opérer les optimisations nécessaires ainsi qu'un bon 'tuning' de la BD,
- renforcer la sécurité en cas d'utilisation douteuse de la BD,
- établir des statistiques sur l'utilisation des objets de la BD, pour éventuellement les reconfigurer.

Attention :

Cette commande d'audit insère une ligne (tuple) dans une vue du système pour chaque action 'auditée'.

Cela implique une surcharge de travail pour le SGBD et donc une possible dégradation des performances globale du système d'information.

Note :

Pour pouvoir utiliser l'audit, il faudrait positionner le paramètre 'AUDIT_TRAL' à 'TRUE' dans le
fichier de paramétrage INIT.ORA lors du démarrage de l'instance de la BD.

Il existe deux types d'AUDIT. Ceux effectués par le DBA et ceux effectués par les propriétaires des
objets.

III.1 Audits système

Nous décrivons brièvement les différentes options de cette commande d'audit.

Commande 1 : AUDIT CONNECT ;
- Permet de connaitre les connexions et déconnexions sur la BD
- Le résultat de l'audit est insérer dans la vue système : DBA_AUDIT_CONNECT.

Commande 2 : **AUDIT DBA** ;
- Permet de connaitre les GRANT, REVOKE, AUDIT, NOAUDIT, etc.
- Le résultat se trouve dans la vue : DBA_AUDIT_DBA

Commande 3 : **AUDIT NOT EXISTS ;**
- Permet de connaitre les références des objets inexistants de la BD,
- Le résultat se trouve dans la vue : DBA_AUDIT_EXISTS.

Commande 4 : **AUDIT RESOURCE ;**
- Permet de connaître les ordres de création, de suppression, de modification sur les tables, les tablespaces, les rollbaccksegment, etc.
- Le résultat se trouve dans la vue : DBA_AUDIT_RESOURCE ;

7

Commande 5 : **NOAUDIT ALL ;**
- Cet ordre permet d'arrêter toutes les traces système. C'est-à-dire d'arrêter les audits.

III.2 Les audits des propriétaires d'objets

- Tout utilisateur peut auditer (tracer) toutes les actions effectuées sur ces tables.
- Les résultats de l'audit est dans la vue USER_AUDIT_TRAIL.
- Pour commencer, il doit déclencher l'audit en choisissant les actions à auditer grâce à la commande sql AUDIT.

La syntaxe de la commande AUDIT est la suivante :

AUDIT <action1>, <action2>, … ON <table1>, <table2>, … ;

Où chacune des actions est dans la liste suivante :

ALTER, INSERT, SELECT, UPDATE, RENAME,
DELETE, AUDIT, COMMENT, GRANT, INDEX

Exemple 1 :

AUDIT SELECT, INSERT
ON T1
BY ACCESS ;
Permet de tracer tous les ordres SELECT et INSERT sur la table T1.
Exemple 2 :
AUDIT ALL
ON DEFAULT
BY ACCESS ;
Trace toutes les actions sur les tables qui seront créées.
Exemple 3 :
NOAUDIT ALL
ON T1, T2 ;
Permet d'arrêter les audits sur les tables T1 et T2.