# Sécurité des données

## *TD1 : Contrôle d'accès*

Benjamin Nguyen

**Objectif :** Ce TP est une introduction aux méthodes de base du contrôle d'accès pour la gestion de données (sur Oracle XE). Dans une première partie nous abordons la création d'utilisateurs, de privilèges, de rôles, et de profiles, et de droits contextuels.

Pour lancer la version en ligne : ouvrez une fenêtre de commande puis tapez sqlplus
Vous entrerez ensuite le login / mdp de votre utilisateur admin ou système. Si vous perdez le mot de passe vous pouvez le réinitialiser en faisant :

```
sqlplus / as sysdba
alter user <username> identified by <password>;
```

## *Utilisateurs, privilèges, rôles, profiles.*

Q1 : **Création des utilisateurs, affectation des droits minimaux, vérification des droits**

1.1 Connectez vous avec un utilisateur ayant des droits administrateur sur la BD (ex: utilisateur «xdba / xdba» ou «system / manager»), et créez un utilisateurs « user_test » en lui donnant le mot de passe « test ».

La syntaxe de création d'utilisateurs est :

CREATE USER <username> IDENTIFIED BY <password> [options];

Vous pouvez préciser un ensemble d'options :

[options]: DEFAULT TABLESPACE <tablespace> -- espaces de travail par défaut (donnez le nom **users**)
TEMPORARY TABLESPACE <tablespace> (donnez le nom **temp**)
QUOTA int {K | M} ON <tablespace> -- quotas sur les TABLESPACES, mettre **temp**
ACCOUNT {LOCK|UNLOCK}

Note : On peut consulter les tablespaces attribués par défaut à l'utilisateur. Vous pourrez consulter les tables systèmes suivantes : **dba_tablespaces, dba_users** et **dba_ts_quotas**.

1.2 Essayez de vous connecter en tant que « user_test » pour constater que ça ne fonctionne pas et qu'il faut des privilèges système.

1.3 Donnez à « user_test » les privilèges systèmes nécessaires pour se connecter et pour pouvoir créer des objets de type tables et vues. Vous pouvez utiliser GRANT ALL PRIVILEGES TO user_test.

1.4 Consultez les tables systèmes adéquates (**dba_sys_privs, dba_tab_privs, dba_role_privs**). Notez les droits attribués à « user_test ». Vous vérifierez au moins 1) les droits systèmes, 2) les droits sur les objets, et 3) les rôles qui sont attribués à l'utilisateur.

Q2 : **Droits implicites et explicites sur des objets**

**Important :** pour pouvoir accéder à une table créée par un utilisateur « user » il faut préfixer le nom de la table par le nom de cet utilisateur. En effet, par défaut le nom du préfixe est celui de l'utilisateur qui est connecté. Ainsi si user2_test souhaite accéder à la table « test » créée par « user1 » il faut écrire : SELECT * FROM user1.test ;

2.1 Créez un deuxième utilisateur « user2_test » et donnez lui les mêmes droits qu'à « user_test ». L'utilisateur « user_test » crée une table nommée TEST_PUBLIC contenant 1 seule colonne DATA de type VARCHAR(100). Peut-il lui-même insérer dans cette table ?

2.2 L'utilisateur « user2_test », s'il connait le nom de cette table et le nom de l'utilisateur qui l'a créée (schéma), peut-il voir que cette table existe? Peut-il accéder à cette table ? Même question pour un administrateur (utilisateur « system »). Il est possible d'utiliser la table : ALL_TABLES voir :
http://docs.oracle.com/cd/B19306_01/server.102/b14237/statviews_2105.htm#REFRN20286

2.3 Faire en sorte que l'utilisateur « user_test » autorise « user2_test » à lire (SELECT) et écrire (INSERT, UPDATE) dans la table TEST_PUBLIC. L'utilisateur « user2_test » devra pouvoir accéder à la table TEST_PUBLIC en utilisant un nom d'objet utilisé comme "synonyme" pour cette table (instruction `CREATE PUBLIC SYNONYM`).

Q3 : **Administration des droits: notion de rôle et de profile utilisateur**

La notion de rôle permet à l'administrateur d'accorder plus simplement des privilèges aux utilisateurs. Une fois les privilèges associés à un rôle, chaque fois qu'un nouveau compte utilisateur sera créé, il suffira de l'associer à son (ou ses) rôle(s) pour lui accorder les droits.

3.1 Connectez-vous en tant qu'administrateur (utilisateur « system »), puis lancez le script VENDOR.sql. Des tables sont créées et peuplées, représentant des vendeurs, les commandes qu'ils ont passées pour les clients, les détails de ces commandes, les produits commandés et leurs fournisseurs.

3.2 Retirer les privilèges de type système (visibles dans la table `dba_sys_privs`) accordés à « user_test » et « user2_test ». Vous pourrez lancer la requête suivante depuis le compte administrateur pour vous aider à identifier les requêtes à exécuter:

```
SELECT 'REVOKE ' || PRIVILEGE || ' FROM ' || GRANTEE || ';' from
dba_sys_privs where grantee LIKE 'USER%';
```

3.3 Créez un rôle «VENDOR_ROLE » et affecter à ce rôle les privilèges "système" initialement accordés aux deux utilisateurs «user_test» et «user2_test». Associez ce rôle à « user_test » et « user2_test », puis vérifiez leurs privilèges. Permettez également à ce rôle de lire et écrire les tables VEN, COM, DET. Vérifiez que les utilisateurs ne peuvent pas accéder à ces tables.

3.4 Donnez le rôle VENDOR_ROLE à user_test. Vérifiez qu'il peut désormais accéder aux tables.

Q4 : **Autorisations basées sur le contenu.**

Modifier le rôle VENDOR_ROLE pour ne permettre à chaque vendeur que :
- de visualiser uniquement ses propres commandes/détails,
- d'insérer et de mettre à jour uniquement les commandes/détails le concernant,
- de visualiser la ligne de la table VEN le concernant, et de ne modifier que le Pays et le Pwd de cette ligne.

Vous serez amené à utiliser la table **sys_context** ou de créer de nouvelles tables.

Q5 : **Autorisations contextuelles.**
On souhaite que les vendeurs ne puissent pas passer de commande après 18h. Mettez en place ce contrôle et tester le.