

# TP Transactions et gestion des droits utilisateurs

## 1. Dictionnaire de données

---

Nous rappelons que le dictionnaire de données (ou méta-schéma) est un ensemble de tables dans lesquelles sont stockées les descriptions des objets de la base.

Les tables de ce dictionnaire peuvent être consultées au moyen du langage SQL. Des vues de ces tables permettent à l'utilisateur de voir les objets qui lui appartiennent (tables préfixées par USER) ou sur lesquels il a des droits (tables préfixées par ALL). L'administrateur a pour sa part accès à toutes les vues (les tables précédentes ainsi que les tables préfixées par DBA). Nous nous concentrerons dans ce TP sur les vues permettant de consulter les privilèges accordés aux usagers.

Quelques vues et tables du dictionnaire de données :

- USER\_TAB\_PRIVS : droits attribués et/ou reçus par l'utilisateur
- USER\_SYS\_PRIVS : privilèges donnés à l'utilisateur de manière générale ;
- USER\_TAB\_PRIVS\_MADE : droits attribués par l'utilisateur ;
- USER\_TAB\_PRIVS\_RECD : droits reçus par l'utilisateur ;

Répondez aux requêtes (portant sur le méta-schéma) suivantes :

1. Avez vous reçu des droits sur des objets provenant d'autres schémas utilisateurs ?
2. Avez vous donné des droits sur des objets provenant de votre schéma utilisateur ?

## 2. Privilèges d'accès à la base de données

---

Oracle permet à plusieurs utilisateurs de travailler sur la même base de données en toute sécurité. Deux commandes sont à ce titre particulièrement importantes : **GRANT** et **REVOKE** et permettent de définir les droits de chaque utilisateur sur les objets de la base.

Tout utilisateur accède à la base à l'aide de son nom utilisateur et de son mot de passe. C'est le nom utilisateur qui permet de déterminer les droits d'accès aux objets de la base de données.

Les utilisateurs ont été créés par le DBA et sont autorisés à se connecter à la base Oracle LICENCE. Ils ont aussi les privilèges de créer des objets de schéma de base de données utilisateur (tables, vues, contraintes etc).

Au cours des TP précédents vous avez travaillé sous le nom d'un utilisateur et vous n'avez donc été en concurrence qu'avec vous-même.

Nous allons vérifier que le SGBD gère la concurrence d'accès à des objets de la base entre plusieurs utilisateurs différents.

Tout utilisateur qui crée des objets est propriétaire de ces objets (**table\_name**, **owner** de **USER\_tables** dans le dictionnaire des données). A ce titre, tout objet dont vous êtes propriétaire est préfixé par votre nom utilisateur. Le créateur d'un objet peut décider de donner (ou de supprimer) certains droits d'accès à cet objet à tout autre utilisateur de sa connaissance.

## 2.1 L'ordre GRANT

`GRANT privilege ON table/vue TO utilisateur [WITH GRANT OPTION]`

Cet ordre permet de donner le privilège concerné sur la table ou la vue à l'utilisateur.

Exemple : X a créé la table EXEMPLAIRE et veut autoriser Y à lire cette table.

Il passe alors l'ordre `GRANT SELECT ON EXEMPLAIRE TO Y;`

Les privilèges qui peuvent être donnés sont les suivants :

1. `SELECT` : droit de lecture ;
2. `INSERT` : droit d'insertion de lignes ;
3. `UPDATE` : droit de modification de lignes ;
4. `DELETE` : droit de suppression de lignes ;
5. `ALTER` : droit de modification de la définition de la table ;
6. `INDEX` : droit de création d'index ;
7. `ALL` : tous les droits ci-dessus.

Un utilisateur ayant reçu un privilège avec la mention facultative `WITH GRANT OPTION` peut les transmettre à son tour à un autre utilisateur.

Pour la suite du TP vous allez donc fonctionner par paire d'étudiants (X connecté sur la machine m1 et Y connecté sur la machine m2).

- X (Y) donne les droits de lecture de sa table EXEMPLAIRE à l'utilisateur Y (X)
- X (Y) donne les droits de modification de sa table ABONNE à l'utilisateur Y (X)
- Vérifier que les privilèges ont été bien accordés en interrogeant les tables du méta-schéma appropriées.
- Testez vos nouveaux droits (les objets que vous interrogerez et dont vous n'êtes pas propriétaire sont désignés par leur nom complet `nompropriétaire.nomobjet`).
- Créez une vue sur une table sur laquelle X (Y) vous a donné des droits en lecture.

## 2.2 L'ordre REVOKE

Un utilisateur ayant accordé un privilège peut le reprendre à tout moment à l'aide de l'ordre `REVOKE`.

`REVOKE privilege ON table/vue FROM utilisateur`

- Enlever les privilèges précédemment accordés
- Vérifier que les privilèges ont bien été supprimés, notamment en interrogeant les tables sur lesquelles vous n'avez plus de droits et en consultant également les tables du méta-schéma comme `USER_TAB_PRIVS`.
- Que constatez vous par rapport à la vue précédemment créée ?
- Rétablissez les droits précédents (utiles à la suite du TP)

## 3. Gestion des accès concurrents

---

Une transaction (ensemble d'ordres SQL) est atomique c'est-à-dire qu'elle ne peut se terminer que par un succès (elle est alors validée) ou un échec (tous ses effets sont alors détruits).

En conséquence, en contexte multi-utilisateurs, les modifications effectuées par une transaction réalisée par un utilisateur ne sont connues des autres utilisateurs que lorsque la transaction a été confirmée

par un COMMIT.

Oracle gère automatiquement les accès concurrents. Si une transaction est en train de modifier les lignes d'une table, les autres transactions peuvent accéder aux données telles qu'elles étaient avant ces dernières modifications (pas de temps d'attente pour la lecture).

Pour rester "simple" nous dirons que toute transaction pose des verrous sur les objets qu'elle manipule et que deux grands types de verrous existent :

- en lecture (verrou passant plusieurs lectures simultanées peuvent avoir lieu)
- en écriture (verrou bloquant la première écriture bloque les autres jusqu'à ce que le verrou soit relaché)

Commandes qui provoquent un blocage implicite sur les tables et les lignes impliquées : DELETE, INSERT, UPDATE, ...

- Faites des sélections sur les mêmes lignes des mêmes tables avec deux noms utilisateurs différents  
Par exemple, X et Y réalisent la même requête : Donnez le numéro et l'état des exemplaires de la table EXEMPLAIRE de X.
- Réessayez le même exercice mais avec des commandes provoquant des blocages  
Par exemple, X modifie la table EXEMPLAIRE : Modifiez l'état de l'exemplaire 5003 de 'BON' à 'ABIME'. Puis X et Y réalisent la même requête : Donnez le numero et l'état des exemplaires sur la table EXEMPLAIRE. Constatations.  
Tester avec COMMIT et ROLLBACK.
- Etreinte mortelle (DEADLOCK) :
  - X fait un UPDATE sur un attribut du tuple 5004 de la table EXEMPLAIRE
  - Y fait un UPDATE sur le tuple de ALBERT RENARD de la table ABONNE
  - X fait un UPDATE sur le tuple de ALBERT RENARD de la table ABONNE
  - Y fait un UPDATE sur le tuple 5004 de la table EXEMPLAIREConstatations. Quelle est la solution ? Quelles sont les opérations qui ont été effectivement effectuées sur les tables concernées ?