

# TP 3 ORACLE : Gestion des Droits Utilisateurs et Transactions

## 1. Dictionnaire de données

---

Nous rappelons que le dictionnaire de données 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 ou sur lesquels il a des droits. L'administrateur a pour sa part accès à toutes les vues. Quelques vues et tables du dictionnaire de données :

- USER\_TABLES: tables et vues créées par l'utilisateur
- USER\_CATALOG (ou CAT): tables et vues sur lesquelles l'utilisateur a des droits à l'exception des tables et vues du dictionnaire de données
- USER\_TAB\_COLUMNS (ou COLS): colonne de chaque table ou vue créée par l'utilisateur courant
- USER\_CONSTRAINTS: définition des contraintes pour les tables des utilisateurs
- USER\_CONS\_COLUMNS: colonnes qui interviennent dans les définitions des contraintes
  
- Connectez vous (les tables du TP précédent étant créées)
- Donner toutes les informations sur les tables sur lesquelles vous avez des droits
- Donner la description des attributs de la table EMP
- Donner la liste des contraintes (avec leur statut) créées au cours du TP précédent
- Donner les informations sur les contraintes de type clé primaire que vous aviez créées au cours de ce TP
  
- Connectez vous (les tables du TP précédent étant créées)
- Donner toutes les informations sur les tables sur lesquelles vous avez des droits
- Donner la description des attributs de la table EMP
- Donner la liste des contraintes (avec leur statut) créées au cours du TP précédent
- Donner les informations sur les contraintes de type clé primaire que vous aviez créées au cours de ce TP

## 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 du master ont été créés par le DBA et sont autorisés à se connecter à la base Oracle master. 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 avez donc été en "concurrence" 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). 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 EMP et veut autoriser Y à lire cette table.

Il passe alors l'ordre **GRANT SELECT ON EMP TO Y;**

Les privilèges qui peuvent être donnés sont les suivants: **SELECT**: droit de lecture, **INSERT**: droit d'insertion de lignes, **UPDATE**: droit de modification de lignes, **DELETE**: droit de suppression de lignes, **ALTER**: droit de modification de la définition de la table, **INDEX**: droit de création d'index, **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" (X connecté sur la machine m1 et Y connecté sur la machine m2).

- X (Y) donne les droits de lecture de "sa" table concernant les employés à l'utilisateur Y (X)
- X (Y) donne les droits de modification de "sa" table DEPT à l'utilisateur Y (X)
- Vérifier que les privilèges ont été bien accordés
- Testez vos nouveaux droits (les objets que vous interrogerez et dont vous n'êtes pas propriétaire sont désignés par leur nom complet nompropriétaire.nomobjet).

### 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

### 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 modifier les 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 relâché)

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

- 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 nom et la date d'embauche des employés" sur la table EMP.
- Réessayez le même exercice mais avec des commandes provoquant des blocages  
Par exemple, X modifie la table EMP : "Modifiez le nom de l'employé BALIN en BALUN". Puis X et Y réalisent la même requête : "Donnez le nom et la date d'embauche des employés" sur la table EMP. Constatations.  
Tester avec `COMMIT` et `ROLLBACK`.

- Etreinte mortelle (DEADLOCK) :

- X fait un `UPDATE` sur le tuple i de la table EMP
- Y fait un `UPDATE` sur le tuple j de la table DEPT
- X fait un `UPDATE` sur le tuple j de la table DEPT
- Y fait un `UPDATE` sur le tuple i de la table EMP

Constatations. Quelle est la solution ? Quelles sont les opérations qui ont été effectivement effectuées sur les tables concernées ?