

# HAI502I - TRAVAUX PRATIQUES 4

## Transactions et gestion des droits utilisateurs

### Objectifs du TP :

L'objet de ce TP est de se familiariser avec les notions de transaction et de gestion de droits utilisateurs.

A partir de l'exercice 2 de ce TP doit, vous devrez travailler à 2 afin de voir comment se comportent les transactions et tester les droits de plusieurs utilisateurs sur une même base de données.

Pour ce TP, vous utiliserez les tables de la BD Bibliothèque créées aux TP précédents.

### Exercice 1 : Dictionnaire de données

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).

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 ;
- 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 ;
- ALL\_CATALOG : liste de tous les objets accessibles par l'utilisateur ;
- ALL\_TABLES Description des tables accessibles par l'utilisateur.

Connectez vous à ORACLE et répondez aux requêtes (portant sur le méta-schéma) suivantes :

1. Donner toutes les informations sur les tables sur lesquelles vous avez des droits
2. Donner le nombre d'attributs de la table ABONNE
3. Donner la liste des contraintes (avec leur statut) créées au cours des TP précédents
4. Donner les informations sur les contraintes de type clé primaire que vous aviez créées au cours des TP précédents
5. Utilisez à bon escient la table USER\_TAB\_COLUMNS de manière à avoir un maximum d'informations sur une table donnée (partageant des similarités avec la commande desc de SQL\*Plus).

## Exercice 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 numéro 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 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 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 ABONNE et veut autoriser Y à lire cette table.

Il passe alors l'ordre : `GRANT SELECT ON ABONNE 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 d'étudiants (X connecté sur la machine m1 et Y connecté sur la machine m2).

Effectuez les instructions suivantes :

1. X (Y) donne les droits de lecture de sa table ABONNE à l'utilisateur Y (X)

2. X (Y) donne les droits de modification de sa table LIVRE à l'utilisateur Y (X)
3. Vérifier que les privilèges ont été bien accordés en interrogeant les tables du méta-schéma appropriées.
4. Testez vos nouveaux droits (les objets que vous interrogerez et dont vous n'êtes pas propriétaire sont désignés par leur nom complet numpropriétaire.nomobjet ; le numéro propriétaire correspond au numéro disponible sur la page web de Sapiens : E2021....)
5. Créez une vue sur une table sur laquelle X (Y) vous a donné des droits.

## 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
```

Effectuez les instructions suivantes :

1. Enlever les privilèges précédemment accordés
2. 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.
3. Que constatez vous par rapport à la vue précédemment créée ?

<b>Exercice 3 : Gestion des accès concurrents</b>
---

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 relâché).

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

Effectuez les instructions suivantes :

1. 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 le prénom des abonnés sur la table ABONNE.
2. Réessayez le même exercice mais avec des commandes provoquant des blocages  
Par exemple, X modifie la table ABONNE : "Modifiez le nom de l'abonné DUPONT en DUPOND".  
Puis X et Y réalisent la même requête : "Donnez le nom et la prénom des abonnés" sur la table ABONNE.  
Constatations.  
Tester avec COMMIT et ROLLBACK.
3. Étreinte mortelle (DEADLOCK) :
  - X fait un UPDATE sur le tuple i de la table ABONNE
  - Y fait un UPDATE sur le tuple j de la table LIVRE
  - X fait un UPDATE sur le tuple j de la table LIVRE
  - Y fait un UPDATE sur le tuple i de la table ABONNEConstatations.  
Quelle est la solution ?  
Quelles sont les opérations qui ont été effectivement effectuées sur les tables concernées ?