Université de La Rochelle
IUT de La Rochelle
Département Informatique
BD2 - UE 21 - M2106
Responsable : J. Malki

TD-TP-07
Utilisateurs, Privilèges et Rôles
Confidentialité et sécurité des données
Application à la base de donnés « Port de plaisance »
Mise en oeuvre Oracle 12c

1 Les différents types d'utilisateurs dans un SGBD

Les types d'utilisateurs, leurs rôles et responsabilités dépendent du site de la base de données. Nous considérons trois types d'utilisateurs, que sont :

- l'administrateur de la base de données ;
- les développeurs d'applications accédant à la base de données ;
- les utilisateurs de la base de données.

Un site de base de données plus important peut nécessiter plusieurs administrateurs de base de données, et peut nécessiter d'autres types d'utilisateurs, comme :

- le responsable de sécurité ;
- le responsable de réseaux base de données.

2 Rôle des trois principaux types d'utilisateurs

- Administrateur de la base de données :
  - installation et mise à jour du serveur BD et les outils applicatifs ;
  - organisation de l'espace de stockage et planification des exigences des BD ;
  - création des unités de stockage primaires (tablespace) pour les développeurs ;
  - création et maintenance des différents objets d'un schéma BD (tables, vues, triggers, séquences, ...) selon les informations et exigences données par les développeurs ;
  - création et maintenance des utilisateurs BD ;
  - garantir la sécurité et la confidentialité des données ;
  - optimisation des performances de la base de données ;

  - planification des sauvegardes et récupération des données ;
  - maintenance des données archivées ;
  - contact du constructeur SGBD pour toute question relative au support ou licence ;
  - etc.

- Développeurs d'applications :
  - conception et développement de l'application BD ;
  - conception de la structure BD de l'application. Cette structure peut changer au cours du cycle de développement pour tenir compte des nouvelles exigences de l'application ou pour son optimisation ;
  - estimation de l'espace de stockage nécessaire à la BD de l'application ;
  - définition des mesures de sécurité d'accès aux données de l'application ;
  - collaboration avec le DBA de l'application en transmettant toutes les informations concernant la base de données de l'application ;
  - etc.

- Utilisateurs :
  - ils interagissent avec la base de données directement ou à travers des applications, ainsi ils peuvent saisir des données, les modifier ou les supprimer ;
  - ils peuvent générer des rapports à partir des données ;
  - etc.

3 Principes d'attribution des droits d'accès

- Dans le cas général, ne jamais donner le droit DBA ni à un utilisateur du logiciel, ni au développeur.
- Créer un rôle qui porte le nom du projet qui aura le droit de gérer les objets du projet et attribuer ce rôle à chacun des développeurs concernés.
- Créer les différents rôles correspondants aux acteurs des cas d'utilisation et attribuer chacun de ces rôles aux utilisateurs concernés.

4 Application au cas « Port de plaisance » : création et mise en oeuvre des rôles, des utilisateurs et leurs droits

4.1 Rôles

Dans le cas de la base de données « Port de plaisance », nous nous intéressons aux deux rôles :

- `OPER` : opérateur de la capitainerie ;
- `RESPORT` : responsable du port

Ces deux rôles peuvent être octroyés à des utilisateurs finaux de la base de données via l'application. Le tableau suivant donne les spécifications des deux rôles.

Rôle	**Description**
`OPER`	Les opérateurs enregistrent les sorties et les retours des bateaux résidents, les affectations d'emplacements et les départs des bateaux de passage.
`RESPORT`	Le responsable du port peut avoir le rôle d'un opérateur de la capitainerie. Le responsable du port gère les emplacements du port, leur type, les bateaux résidents et donc leur propriétaire, et les ports d'immatriculation. Le responsable du port attribue un emplacement à un bateau résident.

4.2 Mise en place du rôle : `OPER`

Créer le dossier de travail pour cette partie : `role_oper`

1. Dossier : `role_user_creation`

   Dans cette partie, on se connecte avec l'utilisateur `LOGIN`.

   (a) Écrire le programme SQL `attrib_privs_oper.sql` qui permet de :
   - créer le rôle `OPER` ;
   - attribuer au rôle les privilèges systèmes nécessaires ;
   - attribuer à ce rôle les privilèges objets nécessaires.

   (b) Écrire le programme SQL `affiche_privs_sys_oper.sql` qui permet d'afficher les privilèges systèmes accordés au rôle `OPER`.
   Indication : table `ROLE_SYS_PRIVS`

   (c) Écrire le programme SQL `affiche_privs_tab_oper.sql` qui permet d'afficher les privilèges objets accordés au rôle `OPER`.
   Indication : table `ROLE_TAB_PRIVS`

(a) Écrire le programme SQL `def_user_oper1.sql` qui permet de :
- créer l'utilisateur base de données `LOGIN_OPER1` ;
- attribuer le rôle `OPER` à l'utilisateur créé.

(b) Écrire le programme SQL permettant d'afficher les rôles attribués à l'utilisateur créé : `affiche_roles_user_oper1.sql`
Indication : table `DBA_ROLE_PRIVS`

(c) Écrire le programme SQL permettant d'afficher les privilèges systèmes attribués à l'utilisateur créé : `affiche_privs_sys_user_oper1.sql`
Indication : tables `DBA_SYS_PRIVS, DBA_ROLE_PRIVS`

(d) Écrire le programme SQL permettant d'afficher les privilèges objets attribués à l'utilisateur créé : `affiche_privs_tab_user_oper1.sql`
Indication : tables `DBA_TAB_PRIVS, DBA_ROLE_PRIVS`

2. Dossier : `user_oper_utils`

Dans cette partie, on se connecte avec l'utilisateur `LOGIN_OPER1`.

(a) Écrire le fichier de connexion de l'utilisateur créé :
`polarisql_connexion_{mac, linux, windows}_INFO1_login_oper1.sh`

(b) Tester la connexion avec l'utilisateur créé ;

(c) Écrire le programme SQL permettant d'afficher les rôles accordés à l'utilisateur créé : `affiche_role_user_oper1.sql`
Indication : table `SESSION_ROLES`

(d) Écrire le programme SQL permettant d'afficher les privilèges systèmes attribués à l'utilisateur créé : `affiche_privs_sys_user_oper1.sql`
Indication : table `ROLE_SYS_PRIVS`

(e) Écrire le programme SQL permettant d'afficher les privilèges objets attribués à l'utilisateur créé : `affiche_privs_tab_user_oper1.sql`
Indication : table `ROLE_TAB_PRIVS`

3. Dossier : `role_user_test`

Dans cette partie, on se connecte avec l'utilisateur `LOGIN_OPER1`.

L'objectif est de tester le fonctionnement des privilèges objets accordés au rôle `OPER` via l'utilisateur créé `LOGIN_OPER1`. Pour cela, on va mettre en place un ensemble de programmes de test à exécuter dans la session SQLPLUS de l'utilisateur `LOGIN_OPER1`.

Partie utilitaires

(a) fichier `afficheContenuBase_OPER.sql`

(b) fichier `prep_test_affect_empl_bat_passage.sql` : programme d'insertion des données initiales nécessaires à exécuter côté utilisateur `LOGIN` pour le déroulement du programme de test de « affecter emplacement à un bateau de passage »

(c) fichier `prep_test_sortie_bat_resident.sql` : programme d'insertion des données initiales nécessaires à exécuter côté utilisateur `LOGIN` pour le déroulement du programme de test de « sortie bateau résident »

Partie programmes de test

(a) fichier `test_valide_OPER_affect_empl_bat_passage.sql` : programme pour tester le privilège « affecter emplacement à un bateau de passage » domaine valide

(b) fichier `test_invalide_OPER_affect_empl_bat_passage.sql` : programme pour tester le privilège « affecter emplacement à un bateau de passage » domaine invalide

(c) fichier `test_valide_OPER_sortie_bat_resident.sql` : programme pour tester le privilège « sortie bateau résident » domaine valide

(d) fichier `test_invalide_OPER_sortie_bat_resident.sql` : programme pour tester le privilège « sortie bateau résident » domaine invalide

4.3 Mise en place du rôle : `RESPORT`

Créer le dossier de travail pour cette partie : `role_resport`

1. Dossier : `role_user_creation`

   Dans cette partie, on se connecte avec l'utilisateur `LOGIN`.

   (a) Écrire le programme SQL `attrib_privs_resport.sql` qui permet de :

   - créer le rôle `RESPORT` ;
   - attribuer au rôle les privilèges systèmes nécessaires ;
   - attribuer à ce rôle les privilèges objets nécessaires.

   (b) Écrire le programme SQL `affiche_privs_sys_resport.sql` qui permet d'afficher les privilèges systèmes accordés au rôle `OPER`.

   (c) Écrire le programme SQL `affiche_privs_tab_resport.sql` qui permet d'afficher les privilèges objets accordés au rôle `OPER`.

   (a) Écrire le programme SQL `def_user_resport1.sql` qui permet de :

   - créer l'utilisateur base de données `LOGIN_RESPORT1` ;
   - attribuer le rôle `RESPORT` à l'utilisateur créé.

   (b) Écrire le programme SQL permettant d'afficher les rôles attribués à l'utilisateur créé : `affiche_roles_user_resport1.sql`

   (c) Écrire le programme SQL permettant d'afficher les privilèges systèmes attribués à l'utilisateur créé : `affiche_privs_sys_user_resport1.sql`

   (d) Écrire le programme SQL permettant d'afficher les privilèges objets attribués à l'utilisateur créé : `affiche_privs_tab_user_resport1.sql`

2. Dossier : `user_resport_utils`

   Dans cette partie, on se connecte avec l'utilisateur `LOGIN_RESPORT1`.

(a) Écrire le fichier de connexion de l'utilisateur créé : `polarisql_connexion_{mac, linux, windows}_INFO1_login_resport1.sh`

(b) Tester la connexion avec l'utilisateur créé ;

(c) Écrire le programme SQL permettant d'afficher les rôles accordés à l'utilisateur créé : `affiche_role_user_resport1.sql`

(d) Écrire le programme SQL permettant d'afficher les privilèges systèmes attribués à l'utilisateur créé : `affiche_privs_sys_user_resport1.sql`

(e) Écrire le programme SQL permettant d'afficher les privilèges objets attribués à l'utilisateur créé : `affiche_privs_tab_user_resport1.sql`

3. Dossier : `role_user_test`

   Dans cette partie, on se connecte avec l'utilisateur `LOGIN_RESPORT1`.

   L'objectif est de tester le fonctionnement des privilèges objets accordés au rôle `RESPORT` via l'utilisateur créé `LOGIN_RESPORT1`. Pour cela, on va mettre en place un ensemble de programmes de test à exécuter dans la session SQLPLUS de l'utilisateur `LOGIN_RESPORT1`.

   Partie utilitaires

   (a) fichier `afficheContenuBase_RESPORT.sql`

   (b) fichier `prep_test_gerer_bat_resident.sql` : programme d'insertion des données initiales nécessaires à exécuter côté utilisateur `LOGIN` pour le déroulement du programme de test de « gérer bateau résident »

   Partie programmes de test

   (a) fichier `test_valide_RESPORT_select_bat_resident.sql` : programme pour tester le privilège « gérer un bateau résident » domaine valide ;

   (b) fichier `test_valide_RESPORT_insert_bat_resident.sql` : programme pour tester le privilège « gérer un bateau résident » domaine valide ;

   (c) fichier `test_valide_RESPORT_update_bat_resident.sql` : programme pour tester le privilège « gérer un bateau résident » domaine valide ;

   (d) fichier `test_valide_RESPORT_delete_bat_resident.sql` : programme pour tester le privilège « gérer un bateau résident » domaine valide ;

5 Livrables du TD-TP-07

- Une archive contenant tous les programmes SQL pour la mise en oeuvre du role `OPER` : `role_oper.zip`
- Une archive contenant tous les programmes SQL pour la mise en oeuvre du role `RESPORT` : `role_resport.zip`