

Préambule

Configuration du prompt

Dans votre *homedir*, créer le fichier `~/.psqlrc` contenant la ligne suivante :

```
\set PROMPT1 '%n%/=> '
```

Cette ligne sera exécutée automatiquement à chacune de vos connexions PostgreSQL, transformant le prompt par défaut

```
postgres=>
```

en le prompt suivant

```
login@base=>
```

où **login** est votre login postgresql et **base** est le nom de la base à laquelle vous êtes connecté.

Fichier de réponses

Pour conserver trace de votre travail et pouvoir discuter de vos solutions avec votre enseignant, vous écrirez les énoncés des questions (en commentaires entre `/*` et `*/`) suivis de vos solutions dans un fichier `~/M2106/tp4.sql`. Nous vous recommandons d'éditer ce fichier puis d'effectuer des copier-coller dans les terminaux où s'exécute une connexion postgresql.

Organisation du bureau

Vous allez accorder des droits à différents utilisateurs, puis vous les testerez en vous connectant avec le login de ces utilisateurs. Nous vous recommandons l'organisation suivante de votre bureau :

- Fenêtre 1 : sujet de votre tp, à gauche sur la moitié supérieure de l'écran ;
- Fenêtre 2 : éditeur de texte pour la mise à jour du fichier `tp4.sql`, à droite sur la moitié supérieure de l'écran ;
- Fenêtre 3: terminal où s'exécute votre connexion postgresql à votre base, à droite sur la moitié inférieure de l'écran ;
- Fenêtre 4: terminal où s'exécute la connexion postgresql d'un utilisateur à votre base, à gauche sur la moitié inférieure de l'écran.

Vérifier grâce aux prompts que vous êtes connectés avec le bon login à la bonne base dans chacun des deux terminaux.

TP 4 – Utilisateurs et droits

Lors du dernier entretien, le président M. Aflau s'est inquiété de la sécurité des informations. En effet, il souhaite répartir les différentes tâches de maintien de la base de données entre les personnes du bureau et les skippers.

Nous rappelons qu'un rôle peut être de deux natures :

- s'il peut se connecter, on parle alors d'utilisateur ;
- sinon, il permet de gérer de façon groupée les droits des différents utilisateurs qui appartiennent à ce rôle.

Des rôles suivants, sans droit de connexion (pour gestion de groupes d'utilisateurs) ont été créés :
president, secretaire, tresorier, bureau, skipper, adherent

ainsi que les utilisateurs suivants (dont le mot de passe est identique au login) :
aflau, guy, michal, merlu, morue, limande

aflau	a reçu les droits des rôles	president, skipper et adherent
guy	a reçu les droits des rôles	tresorier, skipper et adherent
michal	a reçu les droits des rôles	secretaire et adherent
merlu	a reçu les droits des rôles	skipper et adherent
morue et limande	ont reçu les droits du rôle	adherent
president, tresorier et secretaire	ont reçu les droits du rôle	bureau

A. Droits sur la base

1. Commencer par sécuriser l'accès à votre base de données. Pour cela, modifier votre mot de passe, supprimer à tous les utilisateurs le droit de se connecter à votre base puis redonner ce droit uniquement aux rôles **utilisateurs** cités ci-dessus.

B. Droits sur les objets qu'elle contient

Définissez en SQL, les droits accordés à ces différents rôles. A l'issue de chaque question, connectez-vous en tant que l'utilisateur concerné et essayez quelques opérations autorisées ou non.

2. Le président doit avoir tous les droits sur les relations de cette base. Pour cela, accordez au rôle **president** tous les droits d'interrogation et de manipulation de données sur les relations qui ont été créées par le fichier *create.sql*. Connectez-vous en tant que **aflau**, et testez quelques commandes sur ces relations. Quels autres droits particuliers avez-vous sur ces relations lorsque vous êtes connecté avec votre login postgresql (par exemple users2d01), que vous n'avez pas lorsque vous êtes connecté en tant que **aflau** ?
3. Le bureau (président, secrétaire et trésorier) doit pouvoir prendre des décisions, donc a des droits de consultation sur tout ce qui n'est pas confidentiel. Seules les cotisations sont confidentielles.
4. Le trésorier gère les cotisations. Il a donc tous les droits sur la relation *cotisation* et le droit de consultation sur les vues concernant les cotisations (en l'occurrence **vadherentsdebit** créée lors du TP précédent).
5. Intermède : inspection des droits d'un utilisateur
 - Tapez la commande **\du guy**. A quels groupes appartient l'utilisateur **guy** ?
 - Tapez la commande **\dp**. Quels sont les droits de **guy** sur la relation **bateau** ?
 - Tapez la commande : **Revoke select on bateau from guy**; Quels sont les nouveaux droits de **guy** sur la relation **bateau** ?
6. Le secrétaire gère (insertion, mise à jour et consultation mais pas suppression) les adhérents, les bateaux, les agences, les propriétaires et les loueurs de bateaux. Suite aux décisions prises en assemblée générale, le secrétaire saisit les rallyes programmés. Pour cela, créer une vue **vrallye** donnant uniquement les activités de type rallye. Le secrétaire a tous les droits sur cette vue. Il peut insérer et supprimer les régates mais ne peut mettre à jour que la force du vent. Il a tous les droits sur les résultats.
7. La vue **vrallye** étant simple, une fois les droits accordés, insertion et mise-à-jour sont possibles sur celle-ci, sauf s'ils entraînent des actions sur d'autres tables auxquelles le secrétaire n'a pas accès. De plus, rien n'assure que les nouveaux n-uplets correspondent à la vue, et donc soient visibles par celle-ci après impression. Pour cela il faut créer un **trigger** et sa **procédure**. Enfin, pour n'accorder ces droits que sur la vue et non sur la relation **activite**, la procédure doit être définie avec la propriété **SECURITY DEFINER**.