Direction Territoriale Nord Picardie
Département Risque et Développement des Territoires
Informations Géographiques et Statistiques

Jérôme GUILLAUME

Gestion des droits et utilisateurs sous PostgreSQL 9.3

Résumé ou contexte

Le passage des données à plat vers des données en format base possède beaucoup d'avantages. Outre le gain en volume, en fonctionnalités et en rapidité, il y a aussi un gain en possibilités d'administration et gestion des données. Au même titre que pour l'administration d'un réseau informatique avec les droits que vous avez déjà rencontrés. Il existe le même principe au niveau de PostgreSQL. Cette note va vous présenter les bases de la définition des droits dans PostgreSQL, ce qui vous permettra d'envisager le système que vous estimerez le plus adéquat selon votre situation.

les définitions au niveau de postgreSQL

PostgreSQL gère les connexions aux bases de données avec la notion de rôles. C'est un peu différent de ce que l'on connaît par ailleurs mais c'est finalement facile à comprendre.

Le manuel de PostgreSQL explique de manière très détaillée cette notion (cf manuel PostgreSQL 9.3)

Rôles

« **CREATE ROLE** » et « **DROP ROLE** » sont les deux fonctions qui permettent à un super-utilisateur de gérer les rôles (respectivement création et suppression).

La table *pg_roles* contient tous les rôles existants au niveau d'une base de donnée.

Droits des rôles

Les principaux droits des rôles sont :

Nom	Description	Syntaxe en code SQL
LOGIN	Permet au rôle de se connecter à une base de données	CREATE ROLE user LOGIN ;
SUPERUSER	Permet au rôle de tout faire au niveau de la base de données (à manipuler avec précaution)	CREATE ROLE postgres SUPERUSER ;
CREATEDB	Permet au rôle de créer des bases de données	CREATE ROLE user CREATEDB ;
CREATEROLE	Permet au rôle de créer d'autres rôles	CREATE ROLE user CREATEROLE ;
PASSWORD	Assigne un mot de passe à un rôle	CREATE ROLE user PASSWORD 'xxxxxx' ;
INHERIT ou NOINHERIT	Permet, ou pas, à un rôle d'hériter des attributs d'autres rôles	CREATE ROLE user INHERIT ;

Les droits existants

La très grande diversité des actions permises dans PostgreSQL ouvre la porte à un certain nombre de droits :

Nom	Description
SELECT	Permet la sélection sur tout ou partie des colonnes d'une table
INSERT	Permet d'insérer un enregistrement dans une table
UPDATE	Permet la mise à jour des champs d'une table (dépend du

Nom	Description
	SELECT)
DELETE	Permet d'effacer un enregistrement de la table
REFERENCES	Droit requis pour création de clé étrangère (où il faut faire référence à une table tierce)
TRUNCATE	Permet d'effacer une table ou un ensemble de tables (récupération immédiate de l'espace disque sans VACCUM) Truncate = Delete+Vacuum pour faire simple.
TRIGGER	Permet de créer de nouveaux déclencheurs associés à la table
CREATE	Permet la création de nouveaux schémas associés à une base de données. Pour les schémas, cela permet de créer de nouveaux objets (table, index, contraintes, triggers...)
CONNECT	Permet de se connecter à la base
TEMPORARY	Permet de créer des tables temporaires
EXECUTE	Permet d'exécuter une fonction (précise)
USAGE	Autorise l'accès aux objets contenus dans un schéma

N.B : il est inutile de données des droits sur des tables aux utilisateurs si vous ne donner pas le droit USAGE sur le schéma contenant lesdites tables aux utilisateurs.

Principe d'organisation déjà expérimenté

Le principe d'organisation se présente en plusieurs points synthétiques :

- un schéma par utilisateur pour stocker et patager le travail ;
- un schéma public non accessible en édition ;
- plusieurs schéma accueillant les données de références ou autres ;

ces deux derniers restant accessible en lecture.

Un super-utilisateur « postgres »

À l'installation de PostgreSQL, un super-utilisateur est créé qui possède tous les droits. Si vous suivez la méthode d'installation du site Géoinformations, il s'appelle « postgres ». Pour bien faire, cet utilisateur ne devrait être utilisé qu'en cas de nécessité. Le rôle d'administrateur devrait suffire à réaliser l'immense majorité des opérations d'administration et de maintenance des bases.

Un administrateur des bases et des utilisateurs

C'est le rôle qui possédera tous les droits d'administration et de maintenance des bases sans être pour autant un super-utilisateur. Il lui faudra pour ce faire les droits suivants :

- LOGIN
- CREATEDB
- CREATEROLE
- NOINHERIT

Requête de création de l'administrateur

```
CREATE ROLE admin NOSUPERUSER INHERIT CREATEDB CREATEROLE REPLICATION ;
```

Des utilisateurs

Ils sont regroupés sous un seul rôle « utilisateur ». Celui-ci permet, par héritage de droits, la consultation de toutes les tables de tous les schémas par tous les membres du « rôle-groupe » « utilisateur ».

Chaque utilisateur possède un schéma à son nom où lui seul peut créer des tables

et en supprimer ou éditer (un espace de travail).

Le schéma public est mis hors utilisation pour éviter tout basculement ou erreur.

Requête de création du groupe utilisateur

```
CREATE ROLE utilisateur NOSUPERUSER INHERIT NOCREATEDB
NOCREATEROLE NOREPLICATION;
```

Requête de création des rôles par utilisateurs et leur schéma de travail

```
--Creation du rôle nominatif et application des droits du groupe utilisateur
CREATE ROLE <login> WITH PASSWORD '<password_choisi>'
  NOSUPERUSER INHERIT NOCREATEDB NOCREATEROLE
NOREPLICATION LOGIN  ;
GRANT utilisateur TO <login>;
--Creation d'un schema pour chaque utilisateur
CREATE SCHEMA u_<login>
  AUTHORIZATION <login>;

--Chaque utilisateur donne acces en lecture a tout le monde aux tables de son schema
GRANT ALL ON SCHEMA u_<login> TO <login>;
GRANT USAGE ON SCHEMA u_<login> TO utilisateur;
ALTER DEFAULT PRIVILEGES IN SCHEMA u_<login>
    GRANT SELECT ON TABLES
    TO utilisateur;
```

Pour l'utilisation du plugin Cerema export

Pour que l'usage du plugin actuel se fasse sans heurt par tous les utilisateurs, il faut définir des droits spécifique sur la méta-table « geometry_columns ». Cela se fait de la manière suivante :

```
GRANT SELECT, TRUNCATE, DELETE, INSERT ON TABLE
geometry_columns TO utilisateur;
```

Restriction du schéma « public »

```
--Retrait des droits sur le schema public
--Acces en lecture au groupe utilisateur
REVOKE ALL ON SCHEMA public FROM public;
GRANT USAGE ON SCHEMA public TO utilisateur;
ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT SELECT
ON TABLES TO utilisateur;
```