Exploitation d'une base de données

Département Informatique

IUT2 de Grenoble

BUT1 - Ressource 2.06

Introduction à l'exploration et l'administration des BD

1. Révisions : BD relationnelle et SQL simple
2. SQL pour explorer et analyser
3. SQL augmenté : *triggers*
4. Administration de 1er niveau

4. Administration de 1er niveau

4.1. Introduction à l'administration d'une BD
- Administrer : créer et maintenir
- Authentification : à ne pas confondre avec les droits

4.2. Droits
- Rôle
- Droits sur une base
- Droits sur les tables et vues qu'elle contient

4.3. Vue et droits

Rôle

Définition

Rôle = ensemble de privilèges (droits) sur des objets

Rôle : Utilisateur ou Groupe

```
USER = ROLE WITH login
GROUP = ROLE WITH nologin
```

Créer un rôle-groupe

```
CREATE ROLE nom_groupe [WITH option];
```

Insérer dans un groupe

Comment ?

- à la création du rôle :
  CREATE ROLE *nom_role* **IN ROLE** *nom_groupe*;
- après la création du rôle :
  GRANT *nom_groupe* **TO** *nom_role*;
- avec transmission du privilège d'inclure d'autres rôles :
  GRANT *nom_groupe* **TO** *nom_role* **WITH ADMIN OPTION**;

Qui ?

- `superuser`
- `createrole`
- un membre inclus `WITH ADMIN OPTION`

Droits sur une base

Accorder

```
GRANT CONNECT ON DATABASE nom_base
    TO {role | PUBLIC} [,...];
```

Retirer

```
REVOKE CONNECT ON DATABASE nom_base
    FROM {role | PUBLIC} [,...];
```

⚠ par défaut le droit est accordé à **PUBLIC**.

Droits (privilèges) sur les tables et vues d'une base

Privilèges : interrogation et manipulation de données

SELECT, INSERT, UPDATE, DELETE

Accorder

```
GRANT privilège1, privilège2
ON ma_table, ma_vue
TO role1, role2;

GRANT ALL ON ma_table, ma_vue TO PUBLIC ;

GRANT privilège(attribut) ON ma_table TO role;
```

Retirer

```
REVOKE privilège ON ma_table FROM role;
```

⚠ par défaut **aucun** privilège n'est accordé à **PUBLIC**.

Gestion des droits

Qui peut gérer un droit d'utilisation sur une base/table/vue ? (CONNECT, SELECT, INSERT, UPDATE, DELETE)

- `superuser`.
- le propriétaire de la base/table/vue.
- un rôle à qui le droit a été accordé avec la permission de le transmettre :

```
GRANT privilège ON mon_objet
TO role WITH GRANT OPTION;
```

⚠ Droits de définition de données (ALTER, DROP)

- sont réservés à :
  - `superuser`.
  - propriétaire de la base/table/vue.
- **ne se transmettent pas** par **GRANT**.

Droits et rôles

Principe

Rôle = ensemble de privilèges sur des objets

Héritage

Un rôle hérite des privilèges accordés :

- à lui-même
- aux rôles auxquels il appartient
- à **`PUBLIC`**

Une procédure pour définir des droits

Utiliser les rôles-groupes

- Définir un groupe par rôle fonctionnel.
- Accorder les droits à ces groupes.
- Inclure un utilisateur dans les groupes selon ses rôles.

Fermé par défaut

- Définir les droits sur les objets dès leur création.
- **`REVOKE CONNECT ON`** *`ma_base`* **`FROM PUBLIC`**

Commandes `psql` utiles

Utilisateurs et rôles

- liste : **\du**
- utilisateurs et options spéciales

Droits (privilèges)

role=droits/user ≡ *user* accorde tels *droits* à tel *role*

- sur les bases : **\l**
- sur les tables/vues : **\dp**

Redéfinir le *prompt*

- **\set PROMPT1 '%n@%/=> '**
- à intégrer dans **~/.psqlrc**

4. Administration de 1er niveau

4.1. Introduction à l'administration d'une BD

- Administrer : créer et maintenir
- Authentification : à ne pas confondre avec les droits

4.2. Droits

- Rôle
- Droits sur une base
- Droits sur les tables et vues qu'elle contient

4.3. Vue et droits

Une vue pour définir des droits

Donner un droit sur une vue

- ne donne pas le droit sur les tables d'origine
- ni ne l'exige (les droits du créateur de la vue sont utilisés)

Pour un droit de modification sur une vue complexe

```
GRANT + TRIGGER +

CREATE FUNCTION f() RETURNS TRIGGER
AS $$ BEGIN
...
END; $$ LANGUAGE 'plpgsql' SECURITY DEFINER;
```

⚠ **current_user** = *DEFINER*