

Isham HANEEFA
Mohamed Amine MZOUGH
Gr1
TDA

R310 - Gestion d'un système de bases de données

Tp2 : Administration sur PostgreSQL

1. Initialisation et lancement

On installe PostgreSQL, à l'aide de la commande "**apt-get install postgresql**".

```
q20303: root /home/user# apt-get install postgresql
```

Puis on ouvrira un terminal pour chaque utilisateur.

2. En tant que Postgres

Tout d'abord, on crée un utilisateur "**util**", il pourra juste se connecter à la base "**dbutil**".

```
user@q20303:~$ sudo su - postgres
[sudo] Mot de passe de user :
postgres@q20303:~$ psql
psql (9.6.24)
Saisissez « help » pour l'aide.

postgres=# CREATE USER util;
CREATE ROLE
postgres=#
```

Pour lui donner les droits d'accès à la base "**dbutil**", on doit d'abord créer la base de données "**dbutil**".

```
postgres@q20303:~$ createdb dbUtil
postgres@q20303:~$ psql -l
```

Liste des bases de données					
Nom	Propriétaire	Encodage	Collationnement	Type caract.	Droits d'accès
dbUtil	postgres	UTF8	fr_FR.UTF-8	fr_FR.UTF-8	
postgres	postgres	UTF8	fr_FR.UTF-8	fr_FR.UTF-8	
template0	postgres	UTF8	fr_FR.UTF-8	fr_FR.UTF-8	=c/postgres +
template1	postgres	UTF8	fr_FR.UTF-8	fr_FR.UTF-8	postgres=CTc/postgres +
					=c/postgres +
					postgres=CTc/postgres

(4 lignes)

Puis, on fait la commande pour qu'il ait accès à cette base.

```
postgres=# GRANT CONNECT ON DATABASE dbutil to util;
GRANT
```

On voit ainsi qu'en créant l'utilisateur sur UNIX, on peut se connecter à la base.

```
q20303: root /home/user# useradd util
q20303: root /home/user# su - util
No directory, logging in with HOME=/
$ psql dbUtil
psql (9.6.24)
Saisissez « help » pour l'aide.
```

Ensuite, sur un autre terminal, on crée un autre utilisateur “**admin**”, qui aura un mot de passe encrypté pour créer des bases de données.

```
user@q20303:~$ sudo su - postgres
[sudo] Mot de passe de user :
postgres@q20303:~$ psql
psql (9.6.24)
Saisissez « help » pour l'aide.

postgres=# CREATE ROLE admin WITH ENCRYPTED PASSWORD 'mdpadmin' createdb;
CREATE ROLE
postgres=#
```

puis

```
postgres=# ALTER ROLE admin WITH login encrypted password 'mdpadmin';
ALTER ROLE
```

Pour lui donner les droits d'accès à la base “**dbutil**”, on fait la commande ci-dessous.

```
postgres=# GRANT CONNECT ON DATABASE dbutil to admin;
GRANT
```

On voit ainsi qu'en créant l'utilisateur sur UNIX, on peut se connecter à la base.

```
q20303: root /home/user# su - admin
No directory, logging in with HOME=/
$ psql dbUtil
psql (9.6.24)
Saisissez « help » pour l'aide.

dbUtil=>
```

On modifie le mot de passe de “**util**”, à l'aide de la commande suivante.

```
postgres=# ALTER ROLE util PASSWORD 'newMDPutil';
ALTER ROLE
```

On crée les groupes “**grpA**” pour “**admin**” et “**grpU**” pour “**util**”.

```
postgres=# CREATE ROLE grpA WITH ROLE admin;
CREATE ROLE
postgres=# CREATE ROLE grpU WITH ROLE util;
CREATE ROLE
```

On peut vérifier l'attribution d'accès aux groupes à l'aide de la commande suivante.

```
postgres=# \du
```

Nom du rôle	Liste des rôles Attributs	Membre de
admin	Créer une base	{grpA}
grpA	Ne peut pas se connecter	{}
grpU	Ne peut pas se connecter	{}
postgres	Superutilisateur, Créer un rôle, Créer une base, Réplication, Contournement RLS	{}
util		{grpU}

3. En tant qu'*util*

On essaye de se connecter à la base template1, on voit qu'on peut se connecter.

```
q20303: root /home/user# su - util
No directory, logging in with HOME=/
$ psql template1
psql (9.6.24)
Saisissez « help » pour l'aide.

template1=> █
```

On essaye de créer des utilisateurs et des tables. Nous n'avons pas les droits pour créer un utilisateur mais nous pouvons créer des tables.

```
template1=> CREATE ROLE test;
ERREUR:  droit refusé pour créer un rôle
template1=> CREATE TABLE test();
CREATE TABLE
```

4. En tant que *Postgres*

On crée une base de données nommée *dbinfo*.

```
postgres@q20303:~$ createdb dbinfo
```

Puis on crée les tables suivantes.

```
dbinfo=# CREATE TABLE Reference(
dbinfo=# no_ref serial PRIMARY KEY,
dbinfo=# marque VARCHAR(50) NOT NULL,
dbinfo=# model VARCHAR(50) NOT NULL,
dbinfo=# prix INT NOT NULL);
CREATE TABLE
dbinfo=#
```

```
dbinfo=# CREATE TABLE ORDINATEUR
(no_ordi serial PRIMARY KEY,
nom VARCHAR(50) NOT NULL,
no_ref integer REFERENCES reference(no_ref)
);
CREATE TABLE
dbinfo=# █
```

On insère 2 références puis 4 ordinateurs en faisant les commandes suivantes.

```
CREATE TABLE  
dbinfo=# INSERT INTO Reference VALUES (1,'HP','4GEN',500);  
INSERT 0 1  
dbinfo=# INSERT INTO Reference VALUES (2,'DELL','5GEN',1200);  
INSERT 0 1  
dbinfo=#
```

```
dbinfo=# INSERT INTO ORDINATEUR VALUES (1,'DELL',2);  
INSERT 0 1
```