

DCL

Langage de Contrôle de Données

Création et suppression d'utilisateur

Les utilisateurs de la base de données et les utilisateurs de l'OS sont des principes totalement séparés.

Le nom d'utilisateur de base de données doit être unique et ne doit pas commencer par ``pg_``.

Le super utilisateur ``Postgres`` est créé par défaut lors de l'installation.

L'utilisateur ``Postgres`` a tous les privilèges avec l'option ``grant``.

Seulement les super utilisateurs et les utilisateurs avec le privilège ``create role`` peuvent créer un utilisateur.

Tous les utilisateurs peuvent se connecter à toutes les bases de données disponibles.

Création d'utilisateur

Il y a 2 façons de créer un utilisateur

1. Utilisation de la commande ``createuser``.
2. Dans ``psql`` avec la commande ``create user``.

Utilisation de la commande ``createuser``.

Voici un exemple de commande ``createuser``.

`createuser -U postgres -P -S jean`

-P: est utilisé pour assigner un mot de passe.

-S: est utilisé pour spécifier que l'utilisateur n'est pas un super utilisateur.

Au 3ème prompt le mot de passe qui est demandé pour la création de l'utilisateur est le mot de passe ``Postgres``.

Création d'utilisateur

Vous pouvez utiliser la commande suivante pour avoir de l'aide

`createuser --help`

Nous pouvons également créer un utilisateur de base de données en mode interactif avec la commande

`createuser -U postgres --interactive`

Dans `psql`, nous pouvons voir les utilisateurs disponibles à l'aide de la commande
`\du`

Pour se connecter à une base de données avec un utilisateur particulier, utilisez la syntaxe suivante

`psql -U <nom d'utilisateur> -d <nom de la base de données>`

Exemple: `psql -U jean -d postgres`

Création d'utilisateur

Création d'un super utilisateur dans `psql`.

Une fois connecté dans `psql`, voici la syntax à utiliser

```
create user <nom utilisateur> login superuser password '<mot de passe>';
```

Exemple:

```
create user arthur login superuser password 'pass123';
```

Dans `psql` j'ai la possibilité de restreindre l'accès à une base de données à d'autres utilisateurs.

Par exemple si je veux restreindre l'accès à la base de données `nano` aux autres utilisateurs, je peux utiliser la commande suivante

```
revoke connect on database nano from public;
```

Suppression d'utilisateur

En utilisant la commande `dropuser`, voici la syntaxe

`dropuser -U <nom utilisateur de la commande> <nom utilisateur à supprimer>`

Par exemple, si je veux supprimer l'utilisateur `jean`, voici la commande qu'il faut utiliser

`dropuser -U postgres jean`

Nous pouvons également supprimer un utilisateur à partir de `psql` avec la commande suivante

`drop user <nom d'utilisateur>;`

Exemple: `drop user henri;`

Pour supprimer l'utilisateur `henri`.

Les droits dans PostgreSQL

Un privilège est le droit d'exécuter un type de requête SQL ou un droit d'accès à des objets PostgreSQL.

Il y a 2 types de privilèges

1. Privilèges au niveau cluster.
2. Privilèges au niveau objet.

Les privilèges au niveau cluster ne peuvent être accordés que par des super utilisateurs.

Les privilèges au niveau objet peuvent être accordés par des super utilisateurs, les propriétaires des objets ou les utilisateur ayant un privilège `grant`.

Les droits dans PostgreSQL

Les privilèges au niveau cluster.

Par exemple lors de la suppression ou de l'attribution d'un rôle super utilisateur. Nous modifions des privilèges du niveau cluster. Pour pouvoir modifier ce privilèges, nous avons besoin d'un super utilisateur.

Par exemple, lors de la connexion en tant que `postgres` à `psql`, il est possible d'exécuter les commandes suivante

```
alter user arthur with nosuperuser;
```

Pour révoquer les droits super utilisateur.

```
alter user arthur with superuser;
```

Pour rétablir les droits super utilisateur.

Les droits dans PostgreSQL

Pour accorder des droits de création de table en base de données, il est possible d'exécuter la commande suivante dans `psql`.

```
alter user arthur createdb;
```

La gestion des rôles est gérée au niveau du cluster.

Les droits dans PostgreSQL

Ici nous allons voir comment se fait la gestion des privilèges au niveau de l'objet.

Exercice

Créer 2 utilisateurs `jean` et `clara` non super utilisateur. En se connectant via l'utilisateur `jean` sur la base `nano`, créer la table `arbre` contenant une colonne `type` de type `varchar(100)` et insérez un enregistrement.

Lors de la reconnexion sur la base `nano` avec l'utilisateur `clara`, nous remarquons que la requête suivante, génère une erreur.

```
select * from arbre;
```

ERROR: permission denied for table arbre

Les droits dans PostgreSQL

L'erreur précédente est obtenue car l'utilisateur `clara` n'a pas les droits `select` sur la table `arbre` créée par `jean`.

L'utilisateur `jean` peut accorder les droits select à `clara` en se connectant à son tour à la base de données via `psql` et en exécutant la commande suivante
`grant select on arbre to clara;`

Une fois les droits donnés, `clara` peut utiliser la commande select sur la table `arbre`. Mais par exemple une commande `insert` seront rejetés tant que les droits `insert` ne sont pas attribués.

Exemples de commandes Grant

Droits de connexion à une base de données

```
grant connect on database <nom_base_de_donnes> to <nom_utilisateur>;
```

Droits d'utilisation d'un schéma

```
grant usage on schema <nom_schéma> to <nom_utilisateur>;
```

Donner tous les droits DML (Data Manipulation Language)

```
grant select, insert, update, delete on all tables in schema <nom_schéma> to  
<nom_utilisateur>;
```

Donner tous les droits pour toutes les tables d'un schéma

```
grant all privileges on all tables in schema <nom_schéma> to <nom_utilisateur>;
```

Exemples de commandes permission

Donner les droits de création de table

```
ALTER USER <nom_utilisateur> CREATEDB;
```

Donner les droits super utilisateur à un utilisateur

```
ALTER USER <nom_utilisateur> WITH SUPERUSER;
```

Supprimer les droits super utilisateur

```
ALTER USER <nom_utilisateur> WITH NOSUPERUSER;
```

Donner les droits d'accès au niveau d'une colonne particulière

```
GRANT SELECT (col1), UPDATE (col1) ON <nom_table> TO <nom_utilisateur>;
```

Suppression des droits

Pour révoquer les droits d'accès à une table à un utilisateur, nous pouvons utiliser la commande suivante

```
revoke select on arbre from clara;
```

Voici d'autres exemples de suppression de droits

Suppression des droits select, update pour une table

```
REVOKE DELETE, UPDATE ON <nom_table> FROM <nom_utilisateur>;
```

Suppression de tous les droits pour une table pour un utilisateur

```
REVOKE ALL ON <nom_table> FROM <nom_utilisateur>;
```

Suppression d'un privilèges pour tous les utilisateurs publics.

```
REVOKE SELECT ON <nom_table> FROM PUBLIC;
```