

Administrar usuarios y bases de datos

Administrar usuarios y bases de datos	1
¿Qué aprenderás?	2
Introducción	2
Algunas características de PostgreSQL	3
Administración de usuarios en PostgreSQL	4
Crear usuarios	4
Eliminar usuarios	5
Permisos para los usuarios	6
Cambiar de usuario	6
Crear una base de datos	7
Eliminar una base de datos	8
Operaciones comunes a nivel de consola	8



¡Comencemos!

¿Qué aprenderás?

- Crear usuarios y asignar permisos para administrar una base de datos.
- Crear bases de datos para almacenar información de forma organizada.

Introducción

Si piensas en el funcionamiento de una biblioteca, para que una persona acceda a un libro debe contar con algún tipo de credencial de socio. Para poder reservar ese libro y tenerlo en préstamo, no sólo requerirá de la credencial, sino también ser del grupo de usuarios que pueden pedir libros y además no contar con ninguna deuda.

Ese método de control es útil para resguardar la seguridad de los libros de la biblioteca y tener un control de quién accede a qué servicios.

De la misma manera, una de las primeras acciones que debemos realizar sobre nuestra base de datos es definir los permisos, es decir, qué usuario puede acceder a ella y qué acciones puede realizar.

A continuación veremos cómo administrar usuarios dentro de una base de datos, viendo qué permisos pueden tener al momento de enfrentarse a una base de datos.

Algunas características de PostgreSQL

En primer lugar, debemos dejar claro algunos puntos con respecto a la sintaxis de PostgreSQL:

- Las instrucciones deben ser terminadas con el símbolo ";" (punto y coma).
- Hay que tener en cuenta que existen palabras reservadas para variables, ya que estos suelen ser comandos y pasa a ser fácil de confundir.

Algunos comandos son:

ALIAS	AND	AS
CREATE	CREATEDB	CREATEUSER
DATABASE	FROM	INNER
JOIN	LARGE	PASSWORD
WHERE	INSERT	UPDATE

Tabla 2. Palabras reservadas.
Fuente: Desafío Latam.

Existen varios más aparte de estos, los que puedes encontrar en la documentación de PostgreSQL.

- No es sensible a las letras mayúsculas/minúsculas, por lo que se pueden escribir los comandos de cualquiera de estas formas y PostgreSQL lo reconocerá de igual manera.

Administración de usuarios en PostgreSQL

A la hora de trabajar, más de una persona ocupa la información de una misma base de datos. PostgreSQL da la posibilidad de crear usuarios mediante la definición del rol de administrador de base de datos, quien tenga este rol será el encargado de asignar roles, los que van a definir el rango de acciones posibles para usuarios ordinarios; así como la creación/modificación/eliminación de tablas, revocación de roles/usuarios por nombrar algunas de las funcionalidades y solo los usuarios registrados van a poder acceder a la base de datos.

Crear usuarios

Primero, hay que entrar mediante la terminal a la base de datos con el comando:

```
psql
```

Para poder crear un usuario, se debe aplicar el siguiente comando:

```
CREATE USER nombre_usuario;
```

Para ver los usuarios existentes en la base de datos, se usa el comando:

```
\du
```

El cual, lista todos los usuarios existentes en la base de datos, con sus respectivos permisos.

Para poner algún límite a un usuario, se debe hacer lo siguiente:

```
CREATE USER nombre_usuario WITH comando_opcional;
```

Donde algunos de los comandos posibles son:

- **PASSWORD**: Asigna una contraseña al usuario creado.
- **ENCRYPTED PASSWORD**: Le asigna una contraseña encriptada al usuario creado.
- **UNENCRYPTED PASSWORD**: Le asigna una contraseña no encriptada al usuario creado.
- **VALID UNTIL**: La cuenta expirará en la fecha indicada.
- **CREATEDB**: Permite al usuario crear bases de datos.
- **NOCREATEDB**: No permite al usuario crear bases de datos.
- **SUPERUSER**: Puede crear otros usuarios (volveremos a ver esto más adelante).
- **NOSUPERUSER**: No puede crear otros usuarios.

Estos comando pueden ser usados simultáneamente como por ejemplo:

```
CREATE USER Bastian WITH PASSWORD 'contraseña_secreta' VALID UNTIL  
'2019-12-31';
```

En este caso, el usuario Bastian tendrá de contraseña 'contraseña_secreta', su cuenta expirará el 31 de Diciembre del 2019 y no tiene permiso para crear bases de datos (en caso de que no se especifique, se considera NOCREATEDB por defecto)

Eliminar usuarios

Para eliminar usuarios sin tener que esperar la fecha de expiración, se puede utilizar el comando DROP USER:

```
DROP USER nombre_usuario;
```

Habrán casos en que se necesite saber cuales son todos los usuarios que tienen acceso a la base de datos, para ello, se hace lo siguiente:

```
SELECT nombre_usuario FROM pg_user;
```

A veces es necesario crear otro usuario que tenga todos los permisos, al igual que el administrador de la base de datos, lo que entonces se busca es crear un superusuario o superuser y para esto se debe aplicar lo siguiente:

```
CREATE USER nombre_usuario WITH SUPERUSER;
```

Permisos para los usuarios

Una vez creados los usuarios, se pueden cambiar los permisos con los siguientes comandos:

- Para dar acceso a todos los privilegios de una base de datos:

```
GRANT ALL PRIVILEGES ON DATABASE database_name TO nombre_usuario;
```

- Para dar permiso de creación de una base de datos se usa:

```
ALTER USER nombre_usuario CREATEDB;
```

- Para el caso de querer transformar al usuario en superuser:

```
ALTER USER myuser WITH SUPERUSER;
```

- Remover el superusuario:

```
ALTER USER username WITH NOSUPERUSER;
```

Cambiar de usuario

Ahora que se han creado los usuarios, es normal preguntarse cómo se puede acceder a la base de datos con alguno de ellos, entonces ahora corresponde aprender cómo cambiar de usuario.

Primeramente, hay que salir de PostgreSQL con el comando `\q` e ingresar usando lo siguiente:

```
psql -U nombre_usuario
```

Pero al hacer esto, la consola arrojará un error indicando que la base de datos no existe:

```
psql: FATAL: database "nombre_usuario" does not exist
```

Ya que toma por defecto al usuario que estamos ingresando nombre_usuario como el nombre de la base de datos a la que nos queremos conectar.
Para solucionar esto, escribiremos:

```
psql -U nombre_usuario -d nombre_base_de_datos
```

Ahora, para revisar que hemos ingresado de manera correcta, usaremos:

```
SELECT * FROM user;
```

y verán que aparece el nombre de usuario con el que han ingresado a la base de datos:

```
user
-----
nombre_usuario
(1 row)
```

Crear una base de datos

Dentro de un motor de base de datos PostgreSQL, podemos tener varias bases de datos. Para crear una base de datos, usaremos el siguiente comando, el que nos permitirá alojar todas las tablas que se generen posteriormente:

```
CREATE DATABASE nombre_base_de_datos;
```

Pero podemos ver que al crearla no estamos en dicha base de datos, ya que a la izquierda donde escribiremos los comandos nos aparece el nombre de la base de datos a la que estamos conectados.

Para cambiarnos de base de datos, usaremos el comando `\c` de la siguiente forma:

```
\c nombre_base_de_datos
```

Eliminar una base de datos

Tal y como vimos, podemos crear bases de datos pero también eliminarlas, hay que ser consciente y tener cuidado, ya que si los datos de la base de datos no están respaldados, se perderá toda la información.

Para eliminar la base de datos utilizamos el comando:

```
DROP DATABASE nombre_base_de_datos;
```

Operaciones comunes a nivel de consola

En la consola de PostgreSQL existen una serie de comandos que nos van a permitir conectarnos a bases de datos, listar tablas y también listar usuarios. A continuación se presenta una tabla con sus comando más utilizados

Comando	Acción
\c nombre_base	Conectarse a una base de datos específica
\l	Listar todas las bases de datos existentes
\du	Listar todos los usuarios en el motor
\d	Listar todas las relaciones (o tablas) existentes en una base de datos específica
\q	Salir de la consola de PostgreSQL
\h	Mostrar la lista de comandos

Tabla 3. Comandos básicos de PostgreSQL.

Fuente: Desafío Latam.