

APUNTES EXÁMEN BASE DE DATOS

Creación de usuarios y conexión a la base de datos.

Creación de una **base de datos**:

Solo usuarios administradores: **Postgres / marcosfa**

create database nombreBaseDatos

Ej: **create database Dam1**

Comprobar con que **usuario** estamos **conectados**:

select user;

Crear un usuario nuevo : **create user nombreUsuario with password 'contraseña'**

Ej: **create user marcos with password 123**

Borrar un usuario: **drop user nombreUsuario;**

drop user marcos; En ocasiones no deja borrar un usuario ya que este tiene objetos, para borrar estos objetos: **drop owned by usuario cascade**:

drop owned by marcos cascade.

Para **conectarnos** a una **base de datos** con este **usuario** debemos usar:

\c nombreBaseDatos nombreUsuario

Ej: **\c Dam1 marcos** → aquí nos conectamos a la **base de datos Dam1** con el **Usuario marcos**

Para crear un esquema y poner de propietario a un usuario de ese esquema

Create schema nombreEsquema

Alter schema nombreEsquema owner to nombreUsuario

Comandos para mostrar información:

Mostrar información de los usuarios creados en Postgres además de los roles:

\du

Mostrar los esquemas de la base de datos a la que estamos conectados:

\dn

Mostrar el listado de tablas (solo podremos ver aquellas de las cuales tenemos permiso)

\d → por defecto va a esquema public, si ponemos **\d *.*** nos muestra todas las tablas de todos los esquemas

En un esquema pueden existir tablas de diferentes usuarios

\d nombreEsquema. *

Mostrar los privilegios de acceso.

\dp

Mostrar los privilegios de forma más visual de las tablas (por defecto si no cambiamos nada no se ve nada, por ello damos un permiso y después lo quitamos

\z nombreEsquema.*

- **a** insertar (append)
- **r** select (read)
- **w** update
- **d** delete
- **D** truncate
- **x** ejecutar
- **t** trigger

ej:

Schema	Name	Type	Access privileges	Column privileges	p
clinica	trabajadores	table	postgres=arwdDxt/postgres+ u2=a/postgres		

Mostrar todas las tablas dentro de un esquema

\dt esquema.*

Mostrar el listado de las bases de datos creadas, dueños, permisos, etc.

\l → ele

List of databases							
Name	Owner	Encoding	Collate	Ctype	ICU Locale	Locale Provider	Access privileges
basedatos	marcosfa	UTF8	en_US.UTF-8	en_US.UTF-8	en-US	icu	
basex	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	en-US	icu	=T/postgres + postgres=CTc/postgres+ ua=c/postgres + readonly=c/postgres

En los permisos la **C** mayúscula indica que alguien le permitió crear esquemas en la base de datos

Mostrar privilegios de un esquema

\dn+

List of schemas			
Name	Owner	Access privileges	Description
esquemaprueba	marcosfa		
public	pg_database_owner	pg_database_owner=UC/pg_database_owner+ =U/pg_database_owner	standard public schema
schemauser1	user1	user1=UC/user1	

La **C** mayúscula significa crear tablas. La **U** mayúscula significa usar tablas

¡PREGUNTA DE EXAMEN!! Access privileges

Postgres = CTc/postgres +

CTc → Son los distintos permisos que se le dio al usuario. **c** en este caso es conexión

A partir de la **/** indica quien da esos permisos.

Usuario=CTc/postgres+

=c/postgres+

Cuando antes del = no aparece nada → **Public** (todo el mundo puede hacer la próxima acción después del igual) **=c todo el mundo se puede conectar** =Cc todo el mundo puede crear y conectarse. A postgres no se le pueden quitar los permisos

Quitar permisos conexión:

Revoke connect on database dam1 from public

Dar permisos de conexión:

Grant connect on database dam1 to public

Para comprobar los permisos sería bueno hacer esto, ya que por defecto no se actualiza, y al crear una nueva base de datos no aparecen.

Podemos comprobar lo anterior conectándonos **\c dam1 marcos** **¡¡Siempre comprobar los cambios realizados!!**

Crear un esquema:

Create schema nombreDelEsquema

create shcema esquema3

Cambiar el propietario de un esquema:

Alter shcema nombreEsquema owner to usuario

Crear una tabla: **create table nombreTabla(código integer);**

Si queremos crear una tabla dentro de un esquema

Create table esquema3.nombreTabla (código Integer)

Insertar valores:

Insert into esquema.tabla values(x)

Comprobamos: **select * from esquema.tabla;**

Si queremos borrar una tabla de public

Drop table nombreTabla

si queremos eliminar dentro de un esquema

drop table esquema.nombreTabla

Eliminar solo el contenido de la tabla sin eliminar la tabla

Delete from nombreTabla;

Si queremos poner un esquema de búsqueda por defecto para que un usuario vaya directamente a este esquema y no a public (esto se debe hacer desde postgres)

Alter user nombreUsuario in database nombreDatabase set search_path to nombreEsquema

Tambien:

Altet user nombreusuario set search_path to nombreEsquema

Se comprueba con `show search_path >` se debe reiniciar la terminal para que se aplique

Comprobamos con `\dn` si no aparece nos conectaremos con este usuario a la base de datos, y después, haremos **show search_path**

Si este cambio no lo hacemos desde **postgres** el cambio se hace de manera temporal

Borrar una base de datos:

drop database nombreBaseDeDatos

crear un esquema como postgres pero poner como propietario a otro usuario;

create schema nombreEsquema authorization udam2;

crear una **nueva columna** en una tabla que ya existía y ya tenía valores

alter table esquema.tabla add column NOMBRE varchar(20);

actualizar un campo que ya existe:

update esquema.tabla set campo='cambio' where campoClaveprimaria =x

update e1.tabla set nombre='marcos' where coda=1;

insertar un dato en una columna determinada

insert into esquema.tabla (columna) values ('loquesea');

OJO!!

Si queremos cambiar todos los campos de una columna haremos lo siguiente

Update esquema.tabla set nombreCampo = 'valor';

Hacer que un campo de la table sea clave priamria

Alter table el.tabla add primary key (codigo);

Si esa columna tiene un valor nulo, va a dar error, por lo que primero debemos darle un valor

Update esquema.tabla set código=1 where name='juan';

GRANT -DAR PERMISOS -REVOKE

Dar permiso de **select** a un determinado usuario en un esquema (solo si el ususario que da el permiso es postgres o propietario del esquema)

Grant select on nombreEsquema.NOMBRETABLA to nombreUser;

OJO PIOJO:

Si hacemos lo anterior es posible que no podamos hacer select, ya que debemos darle permisos de uso del esquema donde esta dicha tabla, pro lo tanto primero haremos

Grant usage on schema nombreEsquema to nombreuser;

Permiso de conexión a una base de datos:

Grant connect on database nombreBaseDatos

Permiso de uso de un esquema **grant usage**

Permiso para poder hacer **SELECT** o **INSERT** en una tabla

Grant select on nombreTabla to nombreUser;

Grant insert on nombreTabla to nombreusuario;

Dar permisos de select en solo un campo de una tabla

Grant select campo on esquema.tabla to nombreUser

Dar permisos de select a un usuario, para que después este mismo pueda administrar permisos

grant select on esquema.tabla to usuario with grant option;

Ahora este usuario va a poder dar permisos de select a otros usuarios;

OJO!! Primero hay que darles el permiso de usage, para el esquema.

COPIAS DE SEGURIDAD Y ARCHIVOS COMPRIMIDOS:

Primero salimos de la base de datos **\q**

Existen dos maneras de crear un copia de seguridad, de texto plano y en comprimido

pg_dump -U postgres -Fp nombreBase > nombreCopia.dump → crear una copia de seguridad con un **fichero plano**. Para lanzar esta copia tenemos que crear una base de datos y lanzar el **\i**

-U indica el usuario tiene que ser admin
-Fp indica Fichero plano es decir sin comprimir

Te pedirá la contraseña del usuario de -U

Para borrar una base de datos usaremos: **dropdb -U postgres futbol2**

Para recuperar la base de datos, al ser un fichero plano podemos hacerlo desde dentro de postgres o desde fuera.

createdb -U postgres -T template0 nombreBaseACrear → para crear la base de datos

template0 se pone porque necesitas referenciar una base de datos para que cree la nueva base de datos, tomando referencias de esa template, de esta manera será más rápido.

Una vez creada la nueva base de datos, nos conectaremos a postgres y haremos un **li** y arrastraremos el archivo .dump de la copia de seguridad

Al hacer esto, **perdemos el search_path** y vuelve a estar en public.

pg_dump -U postgres -Fc nombreBase > nombreCopia.dump → crear una copia de seguridad con un fichero **comprimido**.

Con un fichero comprimido, no hace falta conectarse a la base de datos para recuperarla con **li**

Al restaurar se puede hacer de varias maneras.

- Restaurarla con el mismo nombre
- Con otro nombre
- Solo parte de la información (ej solo una tabla)

pg_restore -U postgres -C -d nombreBaseQueEsteCreada nombreCopiaComprimida.dump → crea la base y recupera su contenido -C es para que cree la base

nombreBaseQueEsteCreada. → Puedes usar postgres

al restaurarla se pierde el search path.

Crear una nueva base de datos y cargar en ella lo que hemos guardado en el archivo comprimido.

createdb -U postgres -T template0 futbol3;

para cargar los datos de la copia de seguridad

pg_restore -U postgres -d nombreBaseQueEsteCreada nombreCopiaComprimida.dump

A la hora de hacer una copia de seguridad de solo tablas de un determinado esquema y que esas tablas sean las que empiezan por x, debe ser en formato texto plano

pg_dump -U postgres -t 'esquema.x*' -Fp nombreBase > nombreCopia.dump

-T mayúscula excluye

-t minúscula incluye

Se pueden poner varios -t

RECUPERAR ELEMENTOS EN ESPECIFICO, COMO UNA TABLA O EL CONTEIDO DE UNA TABLA

pg_restore -l copiaComprimida > nombreFichero.txt → hace un listado legible de la copia comprimida, gracias a esto podemos borrar cosas para que solo lancemos tablas que se borraron o datos de las tablas etc.

pg_restore -U usuarioAdmin -d nombreBaseCreada -L nombreFichero.txt copiaComprimida.dump (Para recuperar parte de la información)

EJEMPLO EJERCICIO:

```
oracle@oracle-VirtualBox:~$ pg_restore -l fup2.dump > ficheroFup2.txt
oracle@oracle-VirtualBox:~$ pg_restore -U postgres -d futbol2 -L ficheroFup2.txt fup2.dump
```