

# PK, FK y UNIQUE Psql

## Fuentes

```
create table usuarios(  
    id SERIAL,  
    nombre VARCHAR(50) NOT NULL,  
    password VARCHAR(40),  
  
    CONSTRAINT pk_usuarios PRIMARY KEY(id)  
);  
  
create table tablon_anuncios(  
    id SERIAL,  
    id_usuario INT NOT NULL,  
    fecha INT NOT NULL DEFAULT EXTRACT(epoch from now()),  
    titulo VARCHAR(255) NOT NULL,  
    contenido text NOT NULL,  
    CONSTRAINT pk_tablon_anuncios PRIMARY KEY(id_usuario),  
    CONSTRAINT fk1_tablon_anuncios FOREIGN KEY(id_usuario)  
        REFERENCES usuarios(id) MATCH SIMPLE  
        ON UPDATE CASCADE ON DELETE CASCADE  
);  
  
-- Crea la clave primaria de la tabla  
CONSTRAINT pk_tablon_anuncios PRIMARY KEY(id_usuario),  
  
-- Le dice, que la col id_usuario de la tabla tablon_anuncios  
-- es la foreign key, que usa como referencia la tabla usuarios  
-- col id  
CONSTRAINT fk1_tablon_anuncios FOREIGN KEY(id_usuario)  
    REFERENCES usuarios(id) MATCH SIMPLE  
    ON UPDATE CASCADE ON DELETE CASCADE  
  
-- unique  
CONSTRAINT uniq_tablon_anuncios UNIQUE(nombre_tabla)
```

## Comandos consola Postgres

### Nota

Cuando usa \$, significa que esta en el shell, cuando no tiene nada es que esta en psql

Login usuario postgres:

```
$ sudo su - postgres
```

Creación de un usuario:

```
CREATE USER nombre_usuario WITH password '123456'
```

Eliminar usuario:

```
DROP USER nombre_usuario
```

Crear base de datos:

```
CREATE DATABASE nombre_db WITH OWNER nombre_usuario;
```

Eliminar base de datos:

```
DROP DATABASE nombre_db
```

Acceder database con usuario x:

```
psql -U nombre_usuario nombre_db
```

Obtener ayuda:

```
\h
```

Quit

```
\q
```

Leer comandos desde un archivo:

```
\i input.sql
```

Dump db a un archivo:

```
$ pg_dump -U nombre_usuario nombre_db > db.out
```

Dump todas las bases de datos:

```
$ sudo su - postgres  
$ pg_dumpall > /var/lib/pgsql/backups/dumpall.sql
```

Restaurar db:

```
$ sudo su - postgres  
$ psql -f /var/lib/pgsql/backups/dumpall.sql mydb
```

También:

```
$ psql -U postgres nombredb < archivo_restauracion.sql
```

List databases:

```
\l
```

List tables in database:

```
\d
```

Describe table:

```
\d table_name
```

Describe table:

```
\d+ table_name
```

Use database\_name:

```
\c nombre_db
```

Show users:

```
select * from "pg_user";  
# también  
\du
```

Escribir las consultas en tu editor favorito:

```
\e
```

Activar/Desactivar ver el tiempo del query:

```
\timing
```

Reset a user password as admin:

```
ALTER USER usertochange WITH password 'new_passwd';
```

Select version

```
SELECT version();
```

Change Database Owner:

```
ALTER DATABASE database_name OWNER TO new_owner;
```

Create a superuser user:

```
ALTER USER mysuper WITH SUPERUSER;  
# or even better  
ALTER USER mysuper WITH SUPERUSER CREATEDB CREATEROLE INHERIT LOGIN REPLICATION
```

Saber el tamaño usado las tablas en una base de datos:

```
SELECT pg_size_pretty(pg_database_size('dbname'));
```

## Crear Database

Para crear una base de datos y poner como propietario a alguien:

### Nota

#### Crear User

```
sudo -u postgres psql postgres  
CREATE DATABASE practicas WITH OWNER snicoper;
```

## Crear User

Para crear un usuario:

### Nota

#### Crear Database

```
sudo -u postgres psql postgres  
CREATE USER snicoper WITH PASSWORD '123456' NOCREATEDB NOCREATEUSER;
```

## Duplicar tabla en postgresql

### Fuentes

```
CREATE TABLE 'nombre_nueva_tabla' AS SELECT 'tabla_a_copiar';
```

Si la tabla `tabla_a_copiar` esta poblada y no deseamos copiar los datos existentes, se puede poner un `WHERE` en la consulta.

```
CREATE TABLE 'nombre_nueva_tabla'  
AS SELECT 'tabla_a_copiar'  
WHERE 1 = 2;
```

Como 1 no es igual a 2, no se insertara ningún dato de la 1º tabla.

## Obtener ultimo ID insertado en la db

```
INSERT INTO persons (lastname,firstname) VALUES ('Smith', 'John') RETURNING id
```

Ejemplo con Python

```
with psycopg2.connect(conn_string) as conn:  
    with conn.cursor() as cur:  
        cur.execute(  
            "INSERT INTO persons(lastname, firstname) \  
            VALUES(%s, %s) RETURNING id", (firstname, lastname))  
        conn.commit()  
        print(cur.fetchone()[0])
```

## Reset Sequence ID

Ejecutar `pg_dump` primero para no meter la pata.

Se se han añadido y eliminado muchos campos y luego se quiere recuperar la secuencia del campo `id` en postgres.

```
SELECT setval('table_name_id_seq', (SELECT MAX(id) FROM table_name));
```

Si se ha vaciado toda una tabla y se quiere resetear

```
ALTER SEQUENCE table_name_id_seq RESTART;
```