# **PostgreSQL – Copy Database**

<https://www.geeksforgeeks.org/postgresql-copy-database/>

In this article we will discuss the process of copying a PostgreSQL database on the same server or from a server to another.

**PostgreSQL copy database within the same server:**

If a situation arises where one needs to copy a PostgreSQL database within a database server for testing purposes. PostgreSQL makes it simple to do so using the CREATE DATABASE statement as follows:

**Syntax:**

CREATE DATABASE targetdb WITH TEMPLATE sourcedb;

This statement copies the sourcedb to the targetdb. For instance, to copy the dvdrental sample database which is described [here](https://www.geeksforgeeks.org/postgresql-loading-a-database/)and can be downloaded from [here](https://github.com/imkumaraju/dvdrenat-sample-databse), to the dvdrental\_test database, you use the following statement:

CREATE DATABASE dvdrental\_test

WITH TEMPLATE dvdrental;

It may take a while to complete copying depending upon the size of the original database.

CREATE DATABASE [db to create] WITH TEMPLATE [db to copy from] **OWNER** [Your username];

If the “db to copy from” is being accessed by a user, that connection will have to be terminated before copying the database. To do this, run the following command:

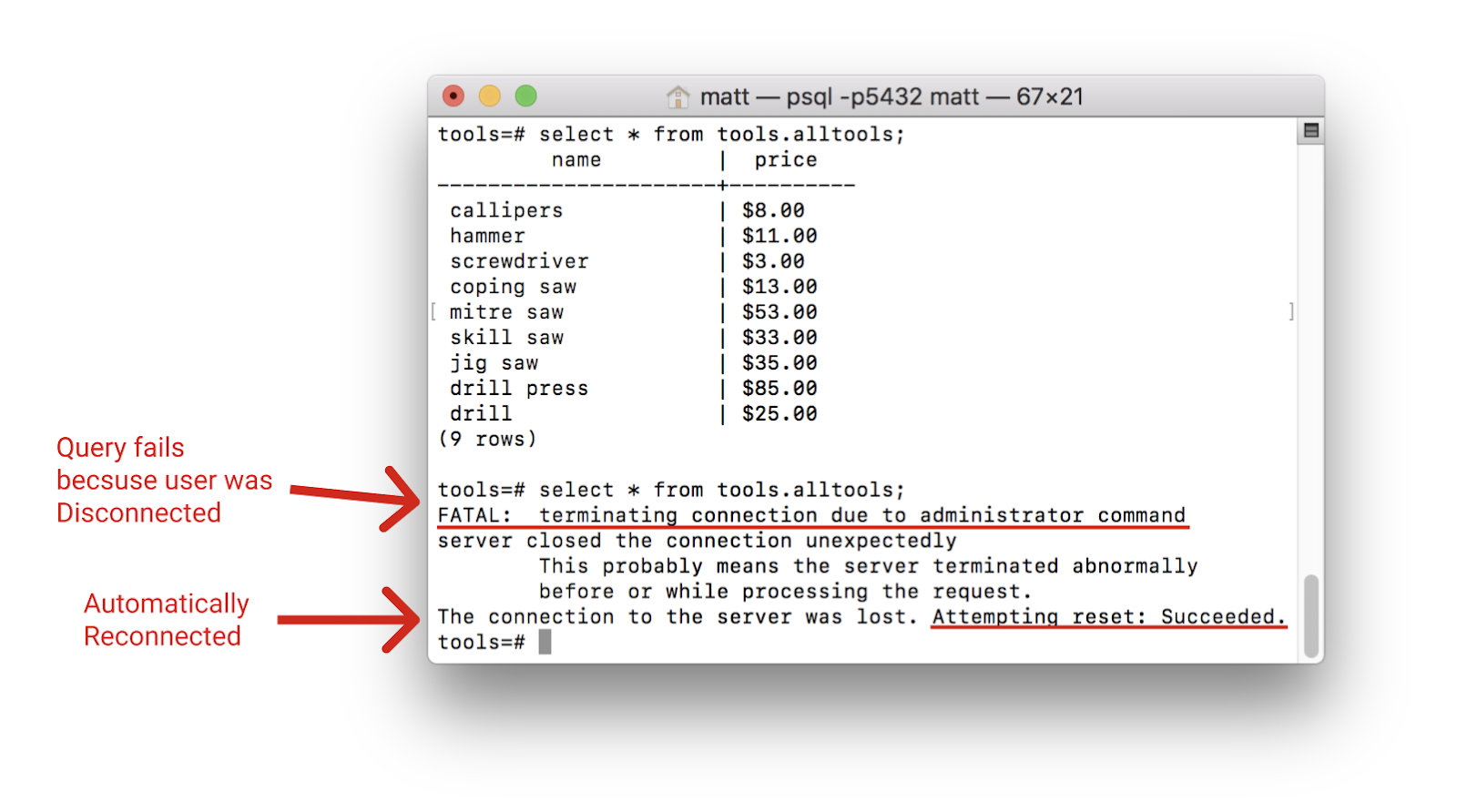
**SELECT** pg\_terminate\_backend(pg\_stat\_activity.pid)

**FROM** pg\_stat\_activity

**WHERE** pg\_stat\_activity.datname **=** '[Database to copy]'

**AND** pid **<>** pg\_backend\_pid();

This query will terminate any open connections to the “Database to copy”, and will cause brief interruptions to anyone accessing the “Database to copy”. It will disconnect users from the database, however psql will automatically reconnect a user whenever they run their next query as shown below:



Once they reconnect they can then run queries again against the database.

Note: They will not be able to reconnect until the database is completely copied.

Once you terminate the connections, create the copy using the first command to CREATE a copy of the specified database. Due to the fact that people can reconnect between the time you terminate and the time you copy, you may want to structure your commands like so:

**SELECT** pg\_terminate\_backend(pg\_stat\_activity.pid)

**FROM** pg\_stat\_activity

**WHERE** pg\_stat\_activity.datname **=** '[Database to copy]'

**AND** pid **<>** pg\_backend\_pid();

**CREATE** **DATABASE** [**Database** **to** **create**]

**WITH** **TEMPLATE** [**Database** **to** **copy**]

**OWNER** [Your username];

When structured and run like this, the CREATE DATABASE command will run immediately after terminating connections. This will help ensure no connections form between terminating connections and copying the database.

**PostgreSQL copy database from a server to another:**

There are many ways to copy a database between various PostgreSQL database servers. If the source database is large and the connection between servers is relatively slower, you can dump the source database to a file, copy the file to the remote server, and restore it. To do so the following commands need to be followed:

* Step 1: Dump the source database to a file.

pg\_dump -U postgres -O sourcedb sourcedb.sql

* Step 2: Copy the dump file to the remote server.
* Step 3: Create a new database in the remote server:

CREATE DATABASE targetdb;

* Step 4: Restore the dump file on the remote server:

psql -U postgres -d targetdb -f sourcedb.sql

**Example:**  
Here we will copy the **winn\_la\_1** database from the local server to the remote server.

(1) we will dump the **winn\_la\_1** database into a dump file e.g., **winn\_la\_1\_2022NOV19**.sql:

* sudo su – postgres
* pg\_dump -U postgres -f winn\_la\_1\_2022nov19.sql winn\_la\_1

(2) we will copy the dump file to a remote server.

(3) we will create the winn\_la\_1 database on the remote server (if not done already):

CREATE DATABASE winn\_la\_1 OWNER winter;

(4) we will restore the winn\_la\_1\_2022nov19.sql dump file in the remote server:

psql -U postgres -d dvdrental -f dvdrental.sql

\*\*\* steps to remove subscription (copied from the source db):

* $sudo su -u postgres
* $sql
* postgres=#SELECT \* FROM pg\_catalog.pg\_publication; /\* to check any existing publication \*/
* postgres=#SELECT \* FROM pg\_catalog.pg\_subscription; /\* to check if any existing subscriptions \*/
* Before you can drop a subscription, you have to disable it first
* postgres=#\d winn\_la\_1;
* winn\_la\_1=# ALTER SUBSCRIPTION pi\_subscription DISABLE;
* winn\_la\_1=# ALTER SUBSCRIPTION pi\_subscription SET (slot\_name=NONE);
* winn\_la\_1=# DROP SUBSCRIPTION pi\_subscription;

If the connection between servers is fast and the size of the database is not big, you can use the following command:

pg\_dump -C -h local -U localuser sourcedb | psql -h remote -U remoteuser targetdbpg\_dump

If one desires to copy the dvdrental database from the localhost server to the remote server, you do it as follows:

pg\_dump -C -h localhost -U postgres dvdrental | psql -h remote -U postgres dvdrental