

PostgreSQL - DROP Database

In this chapter, we will discuss how to delete the database in PostgreSQL. There are two options to delete a database –

- Using DROP DATABASE, an SQL command.
- Using *dropdb* a command-line executable.

Be careful before using this operation because deleting an existing database would result in loss of complete information stored in the database.

Using DROP DATABASE

This command drops a database. It removes the catalog entries for the database and deletes the directory containing the data. It can only be executed by the database owner. This command cannot be executed while you or anyone else is connected to the target database (connect to postgres or any other database to issue this command).

Syntax

The syntax for DROP DATABASE is given below –

```
DROP DATABASE [ IF EXISTS ] name
```

Parameters

The table lists the parameters with their descriptions.

S. No.	Parameter & Description
1	IF EXISTS Do not throw an error if the database does not exist. A notice is issued in this case.
2	name The name of the database to remove.

We cannot drop a database that has any open connections, including our own connection from *psql* or *pgAdmin III*. We must switch to another database or *template1* if we want to delete the database we are currently connected to. Thus, it might be more convenient to use the program *dropdb* instead, which is a wrapper around this command.

Example

The following is a simple example, which will delete **testdb** from your PostgreSQL schema –

```
postgres=# DROP DATABASE testdb;  
postgres=#
```

Using dropdb Command

PostgreSQL command line executable **dropdb** is a command-line wrapper around the SQL command *DROP DATABASE*. There is no effective difference between dropping databases via this utility and via other methods for accessing the server. *dropdb* destroys an existing PostgreSQL database. The user, who executes this command must be a database super user or the owner of the database.

Syntax

The syntax for *dropdb* is as shown below –

```
dropdb [option...] dbname
```

Parameters

The following table lists the parameters with their descriptions

S. No.	Parameter & Description
1	dbname The name of a database to be deleted.
2	option command-line arguments, which dropdb accepts.

Options

The following table lists the command-line arguments dropdb accepts –

S. No.	Option & Description
1	-e Shows the commands being sent to the server.
2	-i Issues a verification prompt before doing anything destructive.
3	-V Print the dropdb version and exit.
4	--if-exists Do not throw an error if the database does not exist. A notice is issued in this case.
5	--help Show help about dropdb command-line arguments, and exit.
6	-h host Specifies the host name of the machine on which the server is running.
7	-p port Specifies the TCP port or the local UNIX domain socket file extension on which the server is listening for connections.
8	-U username User name to connect as.
9	-w Never issue a password prompt.
10	-W Force dropdb to prompt for a password before connecting to a database.
11	--maintenance-db=dbname

Specifies the name of the database to connect to in order to drop the target database.

Example

The following example demonstrates deleting a database from OS command prompt –

```
dropdb -h localhost -p 5432 -U postgres testdb  
Password for user postgres: ****
```

The above command drops the database **testdb**. Here, I have used the **postgres** (found under the pg_roles of template1) username to drop the database.