

POSTGRESQL - DROP DATABASE

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

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

Be careful before using this operation because by 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 as below:

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

Parameters

Table below lists the parameters with their descriptions.

Parameter	Description
IF EXISTS	Do not throw an error if the database does not exist. A notice is issued in this case.
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

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 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 superuser or the owner of

the database.

Syntax

The syntax for *createdb* is as shown below:

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

Parameters

Table below lists the parameters with their descriptions.

Parameter	Description
dbname	The name of a database to be deleted.
option	command-line arguments, which dropdb accepts.

Options

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

Option	Description
-e	Shows the commands being sent to the server.
-i	Issues a verification prompt before doing anything destructive.
-V	Print the dropdb version and exit.
--if-exists	Do not throw an error if the database does not exist. A notice is issued in this case.
--help	Show help about dropdb command-line arguments, and exit.
-h host	Specifies the host name of the machine on which the server is running.
-p port	Specifies the TCP port or the local UNIX domain socket file extension on which the server is listening for connections.
-U username	User name to connect as.
-w	Never issue a password prompt.
-W	Force dropdb to prompt for a password before connecting to a database.
--maintenance-db=dbname	Specifies the name of the database to connect to in order to drop the target database.

Example

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 database **testdb**. Here, I've used the **postgres** found under the `template1` username to drop the database.

