

# How to Manage a Database and its Table(s) in PostgreSQL?

written by sysadmin | 29 October 2025

[The previous article](#) explained how to install a PostgreSQL database on Linux. This article will explain the basics of commands in a PostgreSQL database.

## Problem

How to manage a database and its table(s) in PostgreSQL?

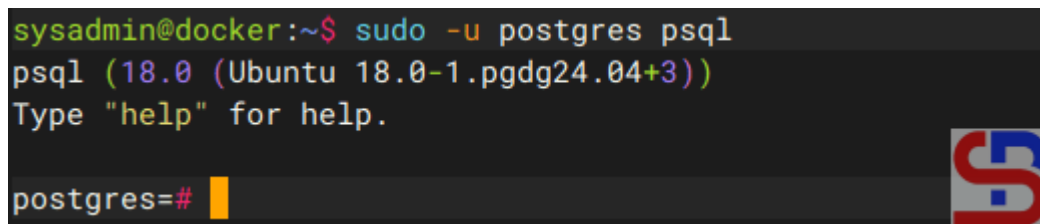
## Solution

Below are the basic commands of PostgreSQL to manage a database and its table(s):

### 1. Access to the PostgreSQL database

Use the command below to access PostgreSQL:

```
sudo -u postgres psql
```

A terminal window with a dark background. The prompt is 'sysadmin@docker:~\$'. The command 'sudo -u postgres psql' has been entered and executed. The output shows 'psql (18.0 (Ubuntu 18.0-1.pgdg24.04+3))' and 'Type "help" for help.'. Below this, the prompt has changed to 'postgres=#' with a yellow cursor bar. A small logo is visible in the bottom right corner of the terminal window.

```
sysadmin@docker:~$ sudo -u postgres psql
psql (18.0 (Ubuntu 18.0-1.pgdg24.04+3))
Type "help" for help.

postgres=#
```

Access to the PostgreSQL

## INFO

The use of capital letters in this article is only to distinguish between original commands from PostgreSQL and data from the user. You don't have to use the capital letters when running these commands, but you can use all lowercase letters.

## 2. List all databases

Use the command below to list all databases in PostgreSQL:

```
\l
```

```
postgres=# \l
```

| Name      | Owner    | Encoding | Locale Provider | Collate     | Ctype       | Locale | ICU Rules | Access privileges       |
|-----------|----------|----------|-----------------|-------------|-------------|--------|-----------|-------------------------|
| postgres  | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |        |           |                         |
| template0 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |        |           | =c/postgres +           |
| template1 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |        |           | postgres=CTc/postgres + |
| (3 rows)  |          |          |                 |             |             |        |           |                         |

```
postgres=#
```

List all databases

## 3. Creating a new database

Use the format below to create a new database:

```
CREATE DATABASE database_name;
```

You can see the options for this command [on this page](#). For example, if you want to create a new database called **mydb**, use the command below;

```
CREATE DATABASE mydb;
```

```
postgres=# CREATE DATABASE mydb;
CREATE DATABASE
postgres=#
```

Create a new database

## 4. Connect to the database

Use the format below to connect to the database:

```
\c db_name;
```

For example, if you want to connect to the mydb database, type the following command:

```
\c mydb;
```

```
postgres=# \c mydb;
You are now connected to database "mydb" as user "postgres".
mydb=#
```

Connect to the database

## 5. Create a table

Use the format below to create a new table:

```
CREATE TABLE table_name (name_of_column1 column_data_type1, name_of_column2
column_data_type2, ...);
```

You can see the options for this command [on this page](#). Type the command below to create an employee table:

```
CREATE TABLE employee (name varchar (100), age int);
```

```
mydb=# CREATE TABLE employee (name varchar (100), age int);
CREATE TABLE
mydb=#
```

Create a table

## 6. Display all tables

Use the command below to display all the tables:

```
\dt
```

```
mydb=# \dt
          List of tables
 Schema |   Name   | Type  | Owner
-----+-----+-----+-----
 public | employee | table | postgres
(1 row)
```

Display all table

## 7. Display the table structure

Use the format below to display the table structure:

```
\d table_name;
```

For example, if you want to display the employee table structure, run the command below:

```
\d employee;
```

```
mydb=# \d employee
```

| Column | Type                   | Collation | Nullable | Default |
|--------|------------------------|-----------|----------|---------|
| name   | character varying(100) |           |          |         |
| age    | integer                |           |          |         |

```
mydb=#
```

Describe a table

If you want to display more information about the table, type the command below:

```
\d+ employee;
```

```
mydb=# \d+ employee
```

| Column | Type                   | Collation | Nullable | Default | Storage  | Compression | Stats target | Description |
|--------|------------------------|-----------|----------|---------|----------|-------------|--------------|-------------|
| name   | character varying(100) |           |          |         | extended |             |              |             |
| age    | integer                |           |          |         | plain    |             |              |             |

Access method: heap

```
mydb=#
```

Display more information in a table

## 8. Add the column

Use the format below to make a column in the table:

```
ALTER TABLE table_name ADD column column_name type(nnn);
```

You can see the options for this command [on this page](#). For example, if you want to add to the city column in the employee table, use the command below:

```
ALTER TABLE employee ADD column city varchar(100);
```

```
mydb=# \d employee
               Table "public.employee"
  Column |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
 name   | character varying(100) |           |          |
 age    | integer                |           |          |
mydb=# ALTER TABLE employee ADD column city varchar(100);
ALTER TABLE
mydb=# \d employee
               Table "public.employee"
  Column |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
 name   | character varying(100) |           |          |
 age    | integer                |           |          |
 city   | character varying(100) |           |          |
mydb=#
```

Add a column

## 9. Insert data into the table

Use the format below to enter data in a table:

```
INSERT INTO table_name (Column1, Column2,..., ColumnN) VALUES (Value1,...ValueN),
(Value1,...ValueN);
```

You can see the options for this command [on this page](#). Type the command below if you want to insert 2 data to the employee table:

```
INSERT INTO employee (name,age,city) VALUES ('bob',21,'New York'),
('John',22,'Chicago');
```

```
mydb=# INSERT INTO employee (name,age,city) VALUES ('bob',21,'New York'), ('John',22,'Chicago');
INSERT 0 2
mydb=#
```

Insert data

### INFO

If you want to insert a value in the form of a number, the number does not have to be flanked with an apostrof ('...') sign, whereas if it is a character or a combination of characters and numbers, it must be flanked with an

apostrof ('...').

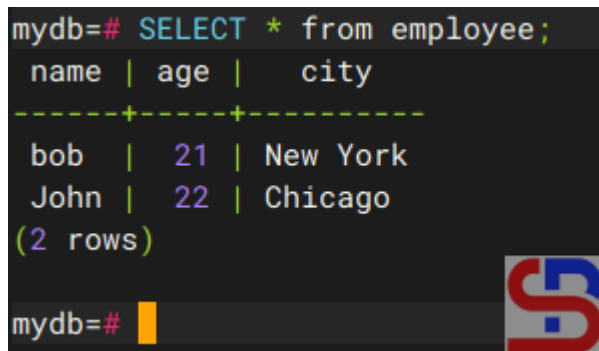
## 10. Displays data in a table

Use the format below to display all data in a table:

```
SELECT option1 FROM table_name option2;
```

You can see the options for this command [on this page](#). For example, use the command below to display all data in the employee table:

```
SELECT * from employee;
```



```
mydb=# SELECT * from employee;
name | age | city
-----+-----+-----
bob   | 21  | New York
John  | 22  | Chicago
(2 rows)
```

Display the data

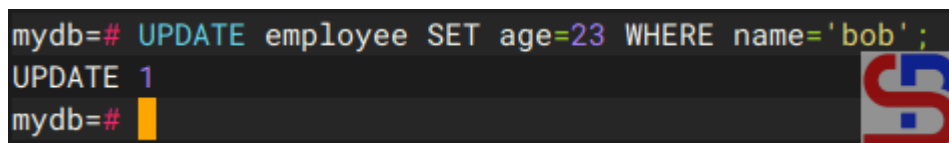
## 11. Update data

Use the format below to update data in a table:

```
UPDATE table_name SET columnX=valueX WHERE columnY=valueY;
```

You can see the options for this command [on this page](#). For example, if you want to update the age of the employee named Bob, use the command below:

```
UPDATE employee SET age=23 WHERE name='bob';
```



```
mydb=# UPDATE employee SET age=23 WHERE name='bob';
UPDATE 1
mydb=#
```

Update the data

## 12. Delete data

Use the format below to delete one or more rows of a table:

```
DELETE FROM table_name WHERE column=value;
```

You can see the options for this command on [this page](#). For example, if you want to delete the data where the user is in Chicago, use the command below:

```
DELETE FROM employee WHERE city='Chicago';
```

```
mydb=# SELECT * from employee;
name | age | city
-----+-----+-----
John  |  22 | Chicago
bob   |  23 | New York
(2 rows)

mydb=# DELETE FROM employee WHERE city='Chicago';
DELETE 1
mydb=# SELECT * from employee;
name | age | city
-----+-----+-----
bob   |  23 | New York
(1 row)

mydb=#
```

Delete the data

## 13. Delete the column

Use the format below to make changes in the table:

```
ALTER TABLE db_name DROP COLUMN column_name;
```

If you want to delete the column, for example, city, you can use the following command:

```
ALTER TABLE employee DROP COLUMN city;
```

```
mydb=# \d employee
          Table "public.employee"
  Column |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
 name   | character varying(100) |           |          |
 age    | integer                |           |          |
 city   | character varying(100) |           |          |

mydb=# ALTER TABLE employee DROP column city;
ALTER TABLE
mydb=# \d employee
          Table "public.employee"
  Column |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
 name   | character varying(100) |           |          |
 age    | integer                |           |          |
```

Delete the column

## 14. Empty the table

You can delete all data in a table with the format as below:

```
TRUNCATE table_name;
```

You can see the options for this command on [this page](#). For example, if you want to delete all the data in the employee table, type the command below:

```
TRUNCATE employee;
```

```

mydb=# SELECT * from employee;
 name | age
-----+-----
  bob  |  23
(1 row)

mydb=# TRUNCATE employee;
TRUNCATE TABLE
mydb=# SELECT * from employee;
 name | age
-----+-----
(0 rows)

mydb=#

```



Truncate the table

## 15. Change a table name

You can change the name of the table using the format below:

```
ALTER TABLE old_table_name RENAME TO new_table_name;
```

You can see the options for this command [on this page](#). Use the command below if, for example, you want to change the name of the employee table to employees:

```
ALTER TABLE employee RENAME TO employees;
```

```

mydb=# \dt
 public | employee | table | postgres

mydb=# ALTER TABLE employee RENAME TO employees;
ALTER TABLE
mydb=# \dt
 public | employees | table | postgres

mydb=#

```



Rename the table

## 16. Delete a table

Use the format below to delete a table:

```
DROP TABLE table_name;
```

You can see the options for this command on [this page](#). For example, if you want to delete the employee table, use the command below:

```
DROP TABLE employees;
```

```
mydb=# \dt
public | employees | table | postgres

mydb=# DROP TABLE employees;
DROP TABLE
mydb=# \dt
Did not find any tables.
mydb=#
```

Delete the table

## 17. Delete a database

To delete a database, use the format below:

```
DROP DATABASE database_name;
```

You can see the options for this command on [this page](#). For example, if you want to delete the mydb database, type the command below:

```
DROP DATABASE mydb;
```

```
mydb=# DROP DATABASE mydb;
ERROR: cannot drop the currently open database
mydb=# \c postgres
You are now connected to database "postgres" as user "postgres".
postgres=# DROP DATABASE mydb;
DROP DATABASE
postgres=#
```

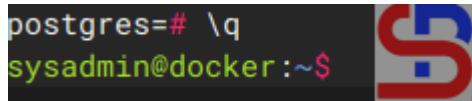
Delete the database

## 17. Quit from the database

To quit from the database, run the command below:

```
\q
```

```
postgres=# \q
sysadmin@docker:~$
```



Quit from the database

## Note

Actually, the basic commands for PostgreSQL and [MariaDB](#) are almost the same, with only there is a slight difference between the two databases.

## References

[hasura.io](https://hasura.io)  
[postgresql.org](https://postgresql.org)  
[w3schools.com](https://w3schools.com)  
[geeksforgeeks.org](https://geeksforgeeks.org)