# PosgreSQL database

## Installation

#### Change root password

Ubuntu current user can connect to the database without a password if PostgreSql was installed correctly. Just type the command **psql** in the terminal. If you can't log in, then follow this procedure on terminal:

1. Type command **whoami**. It will report your login name. Let's say that your login is "**JohnTrevolta**".
2. Run the psql command line interface in full admin mode.

sudo -u postgres psql

1. Create database and postgreSQL user for the JohnTrevolta. You will need to press **Enter** after every command:

CREATE DATABASE JohnTrevolta;

CREATE USER JohnTrevolta;

GRANT ALL PRIVILEGES ON DATABASE JohnTrevolta TO JohnTrevolta;

1. Verify and correct permissions of the current user entering command **\du**. JohnTrevolta would gave the same permissions as **postgres** user (**Superuser**, **Create role**, **Replication**, **Bypass RLS**).

ALTER USER JohnTrevolta [ WITH ] CREATEROLE CREATEDB;

It is possible to create user with enhanced permissionsbut standard security rules do not recommend it:

ALTER USER JohnTrevolta [ WITH ] SUPERUSER CREATEROLE

CREATEDB REPLICATION;

There are two concepts in standard SQL: **role** and **user**. **User** is **role** with **LOGIN** attribute in the PostgreSQL database. **CREATE USER** command creates a role with the **LOGIN** attribute. You need to add the **LOGIN** attribute yourself in the **CREATE ROLE** command. Roles without the **LOGIN** attribute are used for creating groups. Group attributes may be inherited by users belonging to that group.

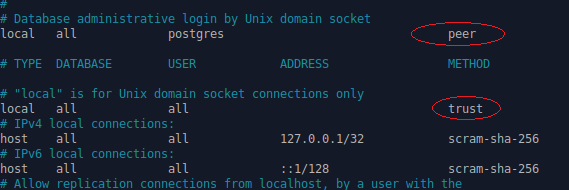
1. Set password if you are going to use the same login from remote computer:

ALTER USER JohnTrevolta WITH ENCRYPTED PASSWORD 'YourPassword';

1. Close psql entering **\q**.
2. Open file /*etc*/postgresql/14/main/pg\_hba.conf in nano or any other text editor:

sudo nano /etc/postgresql/14/main/pg\_hba.conf

1. Verify settings fo **local all**. They must have **peer** and **trust** values (look at picture below):



1. Correct values if they are different, save the file and restart postgreSQL server:

sudo systemctl restart postgresql

1. Now you will be able to connect to postgreSQL by typing the **psql** command on local computer. You would enter command

psql -h host\_name -d database\_name -U JohnTrevolta -W

for connecting from remote computer.

PostgreSql has many GUI tools for a management of the database. You can install [pgAdmin](https://www.pgadmin.org/) if you have a powerful enough computer. Here I will describe **psql**, which works in the terminal and does not require a lot of resources from the computer.

## PSQL

**psql** is a terminal-based front-end to PostgreSQL. It enables you to type in queries interactively, issue them to PostgreSQL, and see the query results. Alternatively, input can be from a file or from command line arguments. In addition, psql provides a number of meta-commands and various shell-like features to facilitate writing scripts and automating a wide variety of tasks.

### Get help on psql commands

1. To know all available psql commands, you use the **\?** command.
2. \?

### Connect to PostgreSQL database

psql

This command will connect to the same computer and install the database under the name you connected to the computer. Read paragraph **Change root password** if login fails.

The following command connects to a database under a specific user. After pressing Enter PostgreSQL will ask for the password of the user.

psql -d database -U user -W

If you want to connect to a database that resides on another host, you add the -h option as follows:

psql -h host -d database -U user -W

This command will only work if psql is installed on your computer. If you don't have **psql**, use **ssh** and connect your terminal to the remote computer where the **PostgreSql** database is installed.

ssh remote\_username@remote\_server\_ip

ssh remote\_username@remote\_server\_name

In case you want to use SSL mode for the connection, just specify it as shown in the following command:

psql -U user -h host "dbname=db sslmode=require"

This command also requires psql installed on your computer.

### List available databases

To list all databases in the current PostgreSQL database server, you use **\l** command:

\l

### Switch connection to a new database

Once you are connected to a database, you can switch the connection to a new database under a user-specified by user. The previous connection will be closed. If you omit the user parameter, the current user is assumed.

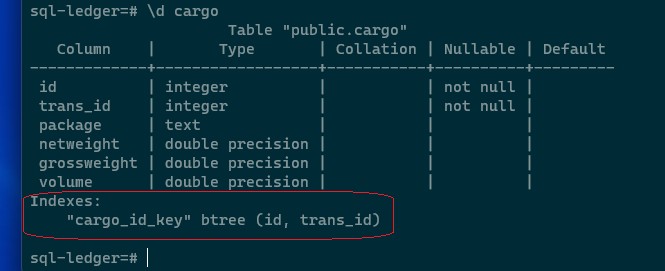
\c dbname username

### Describe a table

To describe a table such as a column, type, or modifiers of columns, you use the following command:

\d table\_name

This command displays information about [indexes](https://www.postgresql.org/docs/current/indexes.html) in addition to columns:



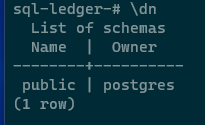
### List commands

#### Available schemas

A database contains one or more named **schemas**, which in turn contain tables. Schemas also contain other kinds of named objects, including data types, functions, and operators. The same object name can be used in different schemas without conflict; for example, both schema1 and myschema can contain tables named mytable. Unlike databases, schemas are not rigidly separated: a user can access objects in any of the schemas in the database they are connected to, if they have privileges to do so. **Schemas** are analogous to directories at the operating system level, except that schemas cannot be nested.

**psql** can only to display a list of schemas. **Use** SQL for all other operations with schemas. To list all schemas of the currently connected database, you use the **\dn** command.

\dn



#### Available tables

1. To list all tables in the current database, you use the \dt command:
2. \dt
3. \dt+
4. The second command displays extended table.

#### Available functions

To list available functions in the current database, you use the **\df** command.

\df

#### Available views

To list available views in the current database, you use the **\dv** command.

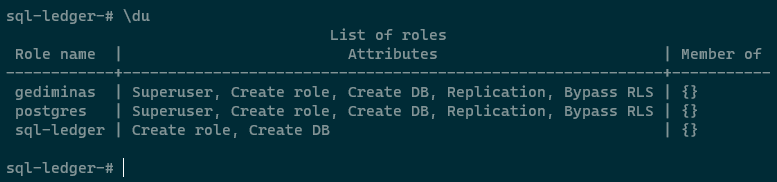
\dv

#### Users and their roles

To list all users and their assigned roles, you use **\du** command:

\du

\du+



### History

#### Execute the previous command

The **\q** command allows you to repeat a previously executed SQL statement.

\q

If you want to save the command history to a file, you need to specify the file name followed the \s command as follows:

#### Command history

To display command history, you use the \s command.

\s

If you want to save the command history to a file, you need to specify the file name followed the \s command as follows:

\s filename

Working with SSH it is important to remember that the file will be saved on the remote computer (the one you connected to with SSH). Working in a Windows environment, you can connect to the remote host with WinScp and copy the file to your computer. Use scp command working from Linux environment (see SSH topics in 01\_OperatingSystem.docx).

[17 Practical psql Commands That You Don't Want To Miss (postgresqltutorial.com)](https://www.postgresqltutorial.com/postgresql-administration/psql-commands/)

[Run PostgreSQL and pgAdmin in docker for local development using docker compose - Blogs, Ideas, Train of Thoughts (belowthemalt.com)](https://belowthemalt.com/2021/06/09/run-postgresql-and-pgadmin-in-docker-for-local-development-using-docker-compose/)

https://medium.com/coding-blocks/creating-user-database-and-adding-access-on-postgresql-8bfcd2f4a91e

## Backup a Single PostgreSQL Database

## SqlLedger admin page

Connect to the site entering URL

http://localhost/sql-ledger/admin.pl

The initial password is "pass123"/ I chnged it to „rublis“.