# **Marvin’s PostgreSQL cheat sheet**

> **psql** -U postgres -d dbname (or **psql** --userame postgres --dbname test, **psql**  -d testdb)

**SELECT** \* FROM users WHERE email = ‘xxxx’;

**SELECT** \* FROM users JOIN login ON users.name = login.name;

**SELECT** \* FROM users ORDER BY score ASC;

**INSERT** INTO table (column1, column2, …) VALUES (value1, value2, …);

**UPDATE** table\_name SET column1 = value1, column2 = value2...., columnN = valueN WHERE [condition];

**DELETE FROM users where name='jenny';**

**\d** display tables in the database

**\d table\_name** or **\d+ table\_name** to find the information on columns of a **table**.

**\du, \du+** - display users

**\l** - check the available database list

**\c database\_name** - connect to database

**CREATE TABLE** users (name text, age smallint, birthday date);

**CREATE TABLE** users (

id uuid UNIQUE DEFAULT uuid\_generate\_v4 (),

email VARCHAR(128) NOT NULL UNIQUE,

password VARCHAR(128) NOT NULL,

registered BIGINT,

token VARCHAR(128) UNIQUE,

createdtime BIGINT,

emailVerified BOOLEAN,

tokenusedbefore BOOLEAN,

PRIMARY KEY (email)

);

**ALTER TABLE** users

ADD reset\_password\_token VARCHAR(128) UNIQUE,

ADD reset\_password\_expires BIGINT,

ADD reset\_password\_token\_used BOOLEAN;

**DROP TABLE login;**

**DROP DATABASE** [ IF EXISTS ] db\_name

**CREATE USER** user\_name WITH PASSWORD 'password';

**GRANT ALL PRIVILEGES** ON DATABASE database\_name to user\_name;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\* <https://www.tutorialspoint.com/postgresql/postgresql_quick_guide.htm>

\*\*\*\*

(1) Installing PostgreSQL: http://www.postgresqltutorial.com/install-postgresql/

Installation Directory: C:\Program Files\PostgreSQL\11

Server Installation Directory: C:\Program Files\PostgreSQL\11

(2)Creating a Database Cluster

(3) initdb -D C:\Program Files\PostgreSQL\11\data

\*\* this database system will be owned by user "Marvin"

\*\* max\_connections...100

\*\* default shared buffer...128MB

\*\* Enabling "trust" authentication for local connections.

- you can change this bu editing pg\_hba.conf or

using the option -A or --auth-local and --auth-host, the next time you run initdb

The PostgreSQL database directory, it can be something like C:\Program Files\PostgreSQL\11\data.

Then open Command Prompt and execute this command:

(a) To **start the server**

pg\_ctl -D "C:\Program Files\PostgreSQL\11\data" start

(b) To **stop the server**

pg\_ctl -D "C:\Program Files\PostgreSQL\11\data" stop

(c) To **restart the server**:

pg\_ctl -D "C:\Program Files\PostgreSQL\11\data" restart

(d) **Check the status** for errors

pg\_ctl -D "C:\Program Files\PostgreSQL\11\data" status

Another way:

1. Open **Run Window** by Winkey + R, then type services.msc
2. Search Postgres service based on version installed. Click stop, start or restart the service option.

**CREATE USER** your-user-name-here WITH SUPERUSER; , and we can verify that he was created with \du . Now we can exit by typing \q and then exit , and enter our database just like Andrei does, with psql 'test' .

Lastly, with pgAdmin4 we need **to create a connection** with the server the first time we use it:

* Right-clicking 'Servers' on the left pane, and choosing 'Create' > 'Server'.
* Give our server a name, and in the 'Connection' tab we type in 'localhost' as the host, and press 'Save'.

## **Part 1 - SQL Shell (psql)**

If you're new to Postgres, it's a good idea to get familiar with the psql shell as you may find yourself needing it from time to time. Start the shell from the start menu and hit enter when asked for Server, Database, Port, Username and enter the password you chose when you first installed postgres.

You are now connected to your server through the shell using the default username and password (postgres).

### Create a user

A good first step after installing PostgreSQL is to create a new user and leave the default 'postgres' user alone. Enter the following SQL to create a new user:

CREATE USER user\_name WITH PASSWORD 'password';

If successful, you should see the message 'CREATE ROLE' and if you type:

\du

you will see a list of all the users on your postgreSQL server (including your new one)

### Create a database

Let's create a database for that awesome project your working on and give your new user privileges to access it. Enter the following to create a new database:

CREATE DATABASE database\_name;

Now grant your user privileges to use this database:

GRANT ALL PRIVILEGES ON DATABASE database\_name to user\_name;

You should see the following if everything went well:

### Test the new user and database

Exit out of the shell using:

\q

and reopen the shell from the start menu. This time login using your new user and database:

You can see in the image above that I am logged in and connected to the simpleapp database. Type the following to see if there are any tables in this database:

\dt

And of course you will get a message telling you there is nothing in this database because you just created it:

### Create a Table

Let's add a table to our database. Enter the following, replacing datatype with one appropriate for the data you expect to keep in that column (See: [PostgreSQL datatypes](https://www.postgresql.org/docs/current/datatype.html)):

CREATE TABLE table\_name(column\_one datatype, column\_two datatype);

and type the following again to see that your table exists:

\dt

So far the shell has been useful for running some basic SQL and to get an idea of how our database is structured. But you can imagine if it contained millions of rows of data across multiple tables how hard it would be to manipulate it all in a shell based environment. That's where pgAdmin comes in.

## Part 2 - pgAdmin

pgAdmin is a browser based GUI that connects to your PostgreSQL server. It allows you to better visualise what is going on in your databases and manage them in almost any way you need. Launch pgAdmin now from the windows **Start Menu.**

### Connect to a Server

On the left hand side of the window, right click on your postgreSQL server, select properties and then the connection tab on the pop up window. Enter the database and username you setup through the shell earlier. Click **Save.**

Back in the main pgAdmin window, this time left click on your server name and enter your users password in the following popup. Click **OK.**

You should now see the pgAdmin dashboard showing some stats for your server. On the left hand side should be 3 new drop downs. Click on 'Databases' to see a list of databases on your server. An 'x' over a database name simply means you aren't connected to that database right now.

Click on your database to connect to it and select **Schemas > public > Tables.** You should see the table you created through the shell earlier. Click on the table and select **Columns** to see those columns you set up at the same time. Pretty cool huh? \

### Use the Query Tool

Now to finish off let's add some data to our table using the pgAdmin query tool. At the top of pgAdmin, right under the main menu on the left click the **lightning bolt icon.** This will bring up the query tool where you can run the same SQL you ran through the shell (and any SQL for that matter). Enter the following to add some data to your table:

INSERT INTO table\_name

(column\_one, column\_two)

VALUES

('value 1', 'value 2'),

('value 3', 'value 4');

And click the lightning bolt at the top of the query editor to execute it.

Now clear out the query editor and execute the following:

SELECT \* FROM table\_name;

to see the values that you just entered in your table