PostgreSQL Cheat Sheet (Basics)

#psql is the interactive terminal used to work with PostgreSQL

#all the commands below can be executed inside de psql terminal

#Get the current version of PostgreSQL

SELECT version();

#or:

\g

#Get all the available psql commands

\?

#Get all the available SQL commands

\h

#Quit psql

\q

#Create a new database

#A database contains one or more schemas

create database staff;

#Select/connect to a database

\c staff;

#Delete a database

drop database staff;

#Create a new user with an encrypted password

create user mihai with encrypted password 'python';

#Grant all priveleges on a database to a user

grant all privileges on database staff to mihai;

#Create a new schema

#A schema contains one or more tables

#The default schema of any database is called 'public'

#Every new table is created within the 'public' schema of the database, unless specified otherwise

#Create a non-default schema

create schema mystaff;

#Create a new schema with a different owner

create schema mystaff authorization john;

#Access tables in a schema (different than 'public')

mystaff.employees

#or, generally, using the database name:

staff.mystaff.employees

#Delete a schema

drop schema mystaff;

#Create a table

#A table contains one or more columns

#Creating a table in the 'public' schema

create table salary(columnA datatype [constraints] primary key,

columnB datatype [constraints],

columnC datatype [constraints]);

#Creating a table in a certain schema

create table mystaff.salary(columnA datatype [constraints] primary key,

columnB datatype [constraints],

columnC datatype [constraints]);

#Delete a table and all of its objects

drop table salary cascade;

#Rename a table

alter table salary rename to salaries;

#Add a new column to a table

alter table salary add column raise varchar(30);

#Delete a column from a table

alter table salary drop column raise;

#Rename a column in a table

alter table salary rename raise to next_raise;

#Insert a new row in the table

insert into salary(columnA, columnB, columnC) values (valueA, valueB, valueC);

#Insert multiple rows in a table

insert into salary(columnA, columnB, columnC) values (valueA, valueB, valueC), (valueA, valueB, valueC), (valueA, valueB, valueC), (valueA, valueB, valueC), (valueA, valueB, valueC);

#Update data in all rows

update salary set columnA = valueA, columnB = valueB;

#Update data for certain rows, based on a condition

update salary set columnA = valueA where condition;

#Delete a certain row from a table, based on a condition

delete from salary where condition;

#Delete all the rows inside a table

delete from salary;

#Query all the data within a table

select * from salary;

#Query the data within a table and return only unique rows

select distinct department from salary;

#Return the number of rows in a certain table

select count (*) from salary;

#Query all the data within a table, based on a condition

select * from salary where condition;

#Query all the data within a table, based on a condition, using LIKE

select * from salary where name like '%Mike%';

#Query all the data within a table, based on a condition, using IN

select * from salary where name in (valueA, valueB, valueC);

#Query all the data within a table, based on a condition, using BETWEEN

select * from salary where raise between valueA and valueB;

#Fetch - retrieve rows from a query using a cursor

FETCH [direction [FROM | IN]] cursor_name

where direction can be empty or one of:

NEXT

PRIOR

FIRST

LAST

ABSOLUTE count

RELATIVE count

count

ALL

FORWARD

FORWARD count

FORWARD ALL

BACKWARD

BACKWARD count

BACKWARD ALL

(source: https://www.postgresql.org/docs/10/static/sql-fetch.html)

#Commit - commiting a transaction

commit;

#Rollback - aborting a transaction

rollback;

#Show users

\du

#Show users (more information)

\du+

#Show databases *(backslash lowercase L)*

\l

#or:

select datname from pg_database;

#Show schemas

\dn

#Show schemas (more information)

\dn+

#Show tables in a certain schema

\dt mystaff.*

#Show tables in a certain schema (more information)

\dt+

#Show tables in all schemas

\dt *.*

#Documentation with all the SQL commands explained in detail

https://www.postgresql.org/docs/10/static/sql-commands.html