PostgreSQL

The PostgreSQL cheat sheet provides you with the common PostgreSQL commands and statements.

Getting started

Getting started

Switch and connect

```
$ sudo -u postgres psql
```

List all databases

```
postgres=# \l
```

Connect to the database named postgres

```
postgres=# \c postgres
```

Disconnect

```
postgres=# \q
postgres=# \!
```

psql commands

Option	Example	Description
`[-d] <database>`	psql -d mydb	Connecting to database
`-U`	psql -U john mydb	Connecting as a specific user
`-h -p`	psql -h localhost -p 5432 mydb	Connecting to a host/port
`-U -h -p -d`	psql -U admin -h 192.168.1.5 -p 2506 -d mydb	Connect remote PostgreSQL
`-W`	psql -W mydb	Force password

Option	Example	Description
`-c`	psql -c '\c postgres' -c '\dt'	Execute a SQL query or command
`-H`	psql -c "\l+" -H postgres > database.html	Generate HTML report
`-l`	psql -l	List all databases
`-f`	psql mydb -f file.sql	Execute commands from a

Getting help

`\h`	Help on syntax of SQL commands
`\h` DELETE	DELETE SQL statement syntax
`\?`	List of PostgreSQL command

Run in PostgreSQL console

Working

Recon

Show version

```
SHOW SERVER_VERSION;
```

Show system status

```
\conninfo
```

Show environmental variables

```
SHOW ALL;
```

List users

```
SELECT rolname FROM pg_roles;
```

Show current user

```
SELECT current_user;
```

Show current user's permissions

```
\du
```

Show current database

```
SELECT current_database();
```

Show all tables in database

```
\dt
```

List functions

```
\df <schema>
```

Databases

List databases

```
\l
```

Connect to database

```
\c <database_name>
```

Show current database

```
SELECT current_database();
```

Create database

```
CREATE DATABASE <database_name> WITH OWNER <username>;
```

Drop database

```
DROP DATABASE IF EXISTS <database_name>;
```

Rename database

Rename database

```
ALTER DATABASE <old_name> RENAME TO <new_name>;
```

Tables

List tables, in current db

```
\dt

SELECT table_schema,table_name FROM information_schema.tables ORDER BY
table_schema,table_name;
```

List tables, globally

```
\dt *.*.

SELECT * FROM pg_catalog.pg_tables
```

List table schema

```
\d <table_name>
\d+ <table_name>

SELECT column_name, data_type, character_maximum_length
FROM INFORMATION_SCHEMA.COLUMNS
WHERE table_name = '<table_name>';
```

Create table

```
CREATE TABLE <table_name>(
  <column_name> <column_type>,
  <column_name> <column_type>
);
```

Create table, with an auto-incrementing primary key

```
CREATE TABLE <table_name> (
  <column_name> SERIAL PRIMARY KEY
);
```

Delete table

```
DROP TABLE IF EXISTS <table_name> CASCADE;
```

Permissions

Become the postgres user, if you have permission errors

```
sudo su - postgres
psql
```

Grant all permissions on database

```
GRANT ALL PRIVILEGES ON DATABASE <db_name> TO <user_name>;
```

Grant connection permissions on database

```
GRANT CONNECT ON DATABASE <db_name> TO <user_name>;
```

Grant permissions on schema

```
GRANT USAGE ON SCHEMA public TO <user_name>;
```

Grant permissions to functions

```
GRANT EXECUTE ON ALL FUNCTIONS IN SCHEMA public TO <user_name>;
```

Grant permissions to select, update, insert, delete, on a all tables

```
GRANT SELECT, UPDATE, INSERT ON ALL TABLES IN SCHEMA public TO <user_name>;
```

Grant permissions, on a table

```
GRANT SELECT, UPDATE, INSERT ON <table_name> TO <user_name>;
```

Grant permissions, to select, on a table

```
GRANT SELECT ON ALL TABLES IN SCHEMA public TO <user_name>;
```

Columns

Add column

```
ALTER TABLE <table_name> IF EXISTS
ADD <column_name> <data_type> [<constraints>];
```

Update column

Update column

```
ALTER TABLE <table_name> IF EXISTS
ALTER <column_name> TYPE <data_type> [<constraints>];
```

Delete column

```
ALTER TABLE <table_name> IF EXISTS
DROP <column_name>;
```

Update column to be an auto-incrementing primary key

```
ALTER TABLE <table_name>
ADD COLUMN <column_name> SERIAL PRIMARY KEY;
```

Insert into a table, with an auto-incrementing primary key

```
INSERT INTO <table_name>
VALUES (DEFAULT, <value1>);


INSERT INTO <table_name> (<column1_name>,<column2_name>)
VALUES ( <value1>,<value2> );
```

Data

[Select](http://www.postgresql.org/docs/current/static/sql-select.html] all data

```
SELECT * FROM <table_name>;
```

Read one row of data

```
SELECT * FROM <table_name> LIMIT 1;
```

Search for data

```
SELECT * FROM <table_name> WHERE <column_name> = <value>;
```

Insert data

```
INSERT INTO <table_name> VALUES( <value_1>, <value_2> );
```

Update data

```
UPDATE <table_name>
SET <column_1> = <value_1>, <column_2> = <value_2>
WHERE <column_1> = <value>;
```

Delete all data

```
DELETE FROM <table_name>;
```

Delete specific data

```
DELETE FROM <table_name>
WHERE <column_name> = <value>;
```

Users

List roles

```
SELECT rolname FROM pg_roles;
```

Create user

```
CREATE USER <user_name> WITH PASSWORD '<password>';
```

Drop user

```
DROP USER IF EXISTS <user_name>;
```

Alter user password

```
ALTER ROLE <user_name> WITH PASSWORD '<password>';
```

Schema

List schemas

```
\dn

SELECT schema_name FROM information_schema.schemata;

SELECT nspname FROM pg_catalog.pg_namespace;
```

Create schema

```
CREATE SCHEMA IF NOT EXISTS <schema_name>;
```

```
CREATE SCHEMA IF NOT EXISTS <schema_name>;
```

Drop schema

```
DROP SCHEMA IF EXISTS <schema_name> CASCADE;
```

Commands

Tables

`\d <table>`	Describe table
`\d+ <table>`	Describe table with details
`\dt`	List tables from current schema
`\dt *.*`	List tables from all schemas
`\dt <schema>.*`	List tables for a schema
`\dp`	List table access privileges
`\det[+]`	List foreign tables

Query buffer

`\e [FILE]`	Edit the query buffer (or file)
`\ef [FUNC]`	Edit function definition
`\p`	Show the contents
`\r`	Reset (clear) the query buffer
`\s [FILE]`	Display history or save it to file
`\w FILE`	Write query buffer to file

Informational

`\l[+]`	List all databases
`\dn[S+]`	List schemas
`\di[S+]`	List indexes
`\du[+]`	List roles

`\ds[S+]`	List sequences
`\df[antw][S+]`	List functions
`\deu[+]`	List user mappings
`\dv[S+]`	List views
`\dl`	List large objects
`\dT[S+]`	List data types
`\da[S]`	List aggregates
`\db[+]`	List tablespaces
`\dc[S+]`	List conversions
`\dC[+]`	List casts
`\ddp`	List default privileges
`\dd[S]`	Show object descriptions
`\dD[S+]`	List domains
`\des[+]`	List foreign servers
`\dew[+]`	List foreign-data wrappers
`\dF[+]`	List text search configurations
`\dFd[+]`	List text search dictionaries
`\dFp[+]`	List text search parsers
`\dFt[+]`	List text search templates
`\dL[S+]`	List procedural languages
`\do[S]`	List operators
`\dO[S+]`	List collations
`\drds`	List per-database role settings
`\dx[+]`	List extensions

S: show system objects, +: additional detail

Connection

`\c [DBNAME]`	Connect to new database
`\encoding [ENCODING]`	Show or set client encoding
`\password [USER]`	Change the password
`\conninfo`	Display information

Formatting

`\a`	Toggle between unaligned and aligned
`\C [STRING]`	Set table title, or unset if none
`\f [STRING]`	Show or set field separator for unaligned
`\H`	Toggle HTML output mode
`\t [on|off]`	Show only rows
`\T [STRING]`	Set or unset HTML <table> tag attributes
`\x [on|off]`	Toggle expanded output

Input/Output

`\copy ...`	Import/export table See also: copy
`\echo [STRING]`	Print string
`\i FILE`	Execute file
`\o [FILE]`	Export all results to file
`\qecho [STRING]`	String to output stream

Variables

`\prompt [TEXT] NAME`	Set variable
`\set [NAME [VALUE]]`	Set variable (or list all if no parameters)
`\unset NAME`	Delete variable

Misc

`\cd [DIR]`	Change the directory
`\timing [on|off]`	Toggle timing

`\! [COMMAND]`	Execute in shell
`\! ls -l`	List all in shell

Large Objects

- `\lo_export LOBOID FILE`
- `\lo_import FILE [COMMENT]`
- `\lo_list`
- `\lo_unlink LOBOID`

Miscellaneous

Backup

Use pg_dumpall to backup all databases

```
$ pg_dumpall -U postgres > all.sql
```

Use pg_dump to backup a database

```
$ pg_dump -d mydb -f mydb_backup.sql
```

`-a`	Dump only the data, not the schema
`-s`	Dump only the schema, no data
`-c`	Drop database before recreating
`-C`	Create database before restoring
`-t`	Dump the named table(s) only
`-F`	Format (`c`: custom, `d`: directory, `t`: tar)

Use `pg_dump -?` to get the full list of options

Restore

Restore a database with psql

```
$ psql -U user mydb < mydb_backup.sql
```

Restore a database with pg_restore

```
$ pg_restore -d mydb mydb_backup.sql -c
```

`-U`	Specify a database user
`-c`	Drop database before recreating
`-C`	Create database before restoring
`-e`	Exit if an error has encountered
`-F`	Format (`c`: custom, `d`: directory, `t`: tar, `p`: plain text sql(default))

Use `pg_restore -?` to get the full list of options

Remote access

Get location of postgresql.conf

```
$ psql -U postgres -c 'SHOW config_file'
```

Append to postgresql.conf

```
listen_addresses = '*'
```

Append to pg_hba.conf (Same location as postgresql.conf)

```
host  all  all  0.0.0.0/0  md5
host  all  all  ::/0       md5
```

Restart PostgreSQL server

```
$ sudo systemctl resatart postgresql
```

Import/Export CSV

Export table into CSV file

```
\copy table TO '<path>' CSV
\copy table(col1,col1) TO '<path>' CSV
\copy (SELECT...) TO '<path>' CSV
```

Import CSV file into table

```
\copy table FROM '<path>' CSV
\copy table(col1,col1) FROM '<path>' CSV
```

See also: Copy

See Also

Posgres-cheatsheet (gist.github.com)

POPULAR

Bash scripting
Vim
RegEX
QuickRef

RECENT

C++
Java
Tmux
Lsof

WEBSITE

Privacy Policy
About