

psql client options

-l	List all available databases
-d <code>__database__</code>	Connect to database
-U <code>__user__</code>	Connect with user
-W <code>__password</code>	Connect with password
-h <code>__hostname__</code>	Connect to host
-p <code>__port__</code>	Connect via port
-c ' <code>__command</code> '	Run command
-f	Run file
-L <code>__logfile__</code>	Output query output to log file and stdout
-o <code>__file__</code>	Output query output to log file
-t	Print tuples only (no headers/footers)

psql commands

Command overview	<code>\?</code>
Exit	<code>\q</code>
Connect to database	<code>\c __database__</code>
Show table definition	<code>\d __table__</code>
List tables	<code>\dt</code>
List databases	<code>\l</code>
List schemas	<code>\dn</code>
List functions	<code>\df</code>
List views	<code>\dv</code>
Show function SQL code	<code>\df+ __function__</code>
Pretty-format query results	<code>\x</code>

Query analysis

See the query plan for the given query	<code>EXPLAIN __query__</code>
See and execute the query plan for the given query	<code>EXPLAIN ANALYZE __query__</code>
Collect statistics	<code>ANALYZE [__table__]</code>
Get statistics for table	<code>SELECT * from pg_stat_user_tables where relname=__table__</code>
Get statistics for table in current transaction	<code>SELECT * from pg_stat_xact_user_tables where relname=__table__</code>

References/Credits

<https://gist.github.com/Kartones/dd3ff5ec5ea238d4c546>
<https://www.postgresql.org/docs/9.6/static/index.html>

DDL

Create database	<code>CREATE DATABASE <database_name> WITH OWNER <username>;</code>
Delete database	<code>DROP DATABASE IF EXISTS <database_name>;</code>
Rename database	<code>ALTER DATABASE <old_name> RENAME TO <new_name>;</code>
Create table	<code>CREATE TABLE <table_name> (<column_name> SERIAL PRIMARY KEY, <column_name> <column_type>);</code>
Delete table	<code>ROP TABLE IF EXISTS <table_name> CASCADE;</code>
Add column	<code>ALTER TABLE <table_name> IF EXISTS ADD <column_name> <data_type> [<constraints>;</code>
Alter column	<code>ALTER TABLE <table_name> IF EXISTS ALTER <column_name> TYPE <data_type> [<constraints>;</code>
Delete column	<code>ALTER TABLE <table_name> IF EXISTS DROP <column_name>;</code>

Constraints

Add primary key	<code>ALTER TABLE __table__ ADD CONSTRAINT __constraintname__ PRIMARY KEY(__col__, __col__);</code>
Add unique	<code>ALTER TABLE __table__ ADD CONSTRAINT __constraintname__ UNIQUE(__col__, __col__);</code>
Add foreign key	<code>ALTER TABLE __table__ ADD CONSTRAINT __constraintname__ FOREIGN KEY(__col__, __col__) REFERENCES __table__(__col__, __col__);</code>
Add check	<code>ALTER TABLE __table__ ADD CONSTRAINT __constraintname__ CHECK(__condition__);</code>
Drop	<code>ALTER TABLE __table__ DROP CONSTRAINT __constraint__;</code>



Handy queries

List procedure/function	<pre>SELECT * FROM pg_proc WHERE proname='__procedurename__'</pre>
Show DB table space in use	<pre>SELECT pg_size_pretty(pg_total_relation_size('__table_name__'));</pre>
Show DB space in use	<pre>SELECT pg_size_pretty(pg_database_size('__database_name__'));</pre>
Show current user's statement timeout	<pre>show statement_timeout;</pre>
Queries being executed	<pre>SELECT datname, application_name, pid, backend_start, query_start, state_change, state, query FROM pg_stat_activity WHERE datname='__database_name__';</pre>
Get all queries from all dbs waiting for data	<pre>SELECT * FROM pg_stat_activity WHERE waiting='t';</pre>
Currently running queries with process pid	<pre>SELECT pg_stat_get_backend_pid(s.backend_id) AS procpid, pg_stat_get_backend_activity(s.backendid) AS current_query FROM (SELECT pg_stat_get_backend_idset() AS backendid) AS s;</pre>

Users

Add user	<pre>ADD USER __user__;</pre>
Drop user	<pre>DROP USER __user__;</pre>
Add user with password	<pre>ADD USER __user__ WITH PASSWORD __password__;</pre>
Grant permissions for table	<pre>GRANT INSERT, UPDATE, DELETE ON TABLE __table__ TO __user__;</pre>
Grant permissions for all tables	<pre>GRANT INSERT, UPDATE, DELETE ON ALL TABLES TO __user__;</pre>
Grant permissions for sequence	<pre>GRANT USAGE ON SEQUENCE __sequence__ TO __user__;</pre>
Grant permissions for all sequences	<pre>GRANT USAGE ON ALL SEQUENCES TO __user__;</pre>
Grant permissions for function	<pre>GRANT EXECUTE ON FUNCTION __function__ TO __user__;</pre>
Grant permissions for all functions	<pre>GRANT EXECUTE ON ALL FUNCTIONS TO __user__;</pre>

Common Functions

Current date	<pre>current_date</pre>
Current time	<pre>current_time</pre>
Current timestamp	<pre>localtimestamp</pre>
Get date part of column	<pre>date(__col__)</pre>
Get time part of column	<pre>__col__::time</pre>
Add interval to timestamp	<pre>__col__ + interval '1 hour'</pre>
Return serial number	<pre>INSERT ... RETURNING LASTVAL();</pre>
Conditional expressions	<pre>CASE expression WHEN value THEN result [WHEN ...] [ELSE result] END</pre>
Null values	<pre>COALESCE(__value__, __default1__, __default2__)</pre>
Cast	<pre>__value__::__type__</pre>
Execute function returning value	<pre>SELECT __function__;</pre>
Execute function returning table	<pre>SELECT * FROM __function__;</pre>



By **hugomfcruz**

cheatography.com/hugomfcruz/

Not published yet.

Last updated 4th October, 2017.

Page 2 of 2.

Sponsored by **Readability-Score.com**

Measure your website readability!

<https://readability-score.com>