

PostgreSQL Cheatsheet

By Dejan Panovski • Updated on Apr 16, 2026 • [Download PDF](#)

Quick reference for connecting to PostgreSQL, managing databases and roles, granting privileges, and running backup commands.

PostgreSQL handles everything from local development databases to production application backends. This cheatsheet covers common `psql` connection patterns, database and role management, privilege commands, introspection queries, and backup basics.

Basic Syntax

Core PostgreSQL command forms.

<code>psql</code>	Open an interactive PostgreSQL shell using local defaults
<code>psql -U user -d dbname</code>	Connect as a specific user to a specific database
<code>psql -h host -p 5432 -U user -d dbname</code>	Connect to a remote PostgreSQL server
<code>psql -c "SQL_STATEMENT"</code>	Run one SQL command and exit
<code>sudo -u postgres psql</code>	Open <code>psql</code> as the local <code>postgres</code> superuser

Connect and Switch

Common ways to connect and move between databases.

<code>sudo -u postgres psql</code>	Connect locally as the <code>postgres</code> system user
<code>psql -U app_user -d app_db</code>	Connect to <code>app_db</code> as <code>app_user</code>
<code>psql "host=localhost port=5432 dbname=app_db user=app_user"</code>	Connect with a connection string
<code>\c app_db</code>	Switch to another database inside <code>psql</code>
<code>\conninfo</code>	Show the current connection details

Databases

Create, list, rename, and remove databases.

<code>CREATE DATABASE app_db;</code>	Create a new database
<code>CREATE DATABASE app_db OWNER app_user;</code>	Create a database owned by a specific role
<code>\l</code>	List databases
<code>ALTER DATABASE app_db RENAME TO app_prod;</code>	Rename a database
<code>DROP DATABASE app_db;</code>	Delete a database

Roles and Users

Create login roles and inspect existing roles.

<code>CREATE ROLE app_user;</code>	Create a role without login
<code>CREATE ROLE app_user WITH LOGIN PASSWORD 'strong_password';</code>	Create a login role
<code>CREATE USER app_user WITH PASSWORD 'strong_password';</code>	Shortcut for a login role
<code>ALTER ROLE app_user WITH PASSWORD 'new_password';</code>	Change a role password
<code>\du</code>	List roles and attributes

Grant and Revoke Privileges

Give or remove access at the database, schema, and table levels.

<code>GRANT CONNECT ON DATABASE app_db TO app_user;</code>	Allow a role to connect to a database
<code>GRANT USAGE, CREATE ON SCHEMA public TO app_user;</code>	Allow schema access and object creation
<code>GRANT SELECT, INSERT, UPDATE, DELETE ON TABLE orders TO app_user;</code>	Grant table privileges
<code>REVOKE INSERT, UPDATE ON TABLE orders FROM app_user;</code>	Remove selected table privileges
<code>ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT SELECT ON TABLES TO app_user;</code>	Grant defaults for future tables

Table and Schema Introspection

Inspect schemas, tables, columns, and query results.

<code>\dn</code>	List schemas
<code>\dt</code>	List tables in the current search path
<code>\dt public.*</code>	List tables in the public schema
<code>\d orders</code>	Describe a table, view, or sequence
<code>SELECT * FROM orders LIMIT 10;</code>	Preview rows from a table

psql Meta-Commands

Useful built-in `psql` commands for daily administration.

<code>\?</code>	Show <code>psql</code> meta-command help
<code>\h CREATE ROLE</code>	Show SQL help for one statement
<code>\x</code>	Toggle expanded output for wide rows
<code>\timing</code>	Toggle query timing display
<code>\q</code>	Quit <code>psql</code>

Backup and Restore

Common logical backup and restore commands.

<code>pg_dump -U app_user -d app_db > app_db.sql</code>	Export a database as plain SQL
<code>pg_dump -Fc -U app_user -d app_db -f app_db.dump</code>	Create a custom-format backup
<code>psql -U app_user -d app_db < app_db.sql</code>	Restore a plain SQL dump
<code>pg_restore -U app_user -d app_db app_db.dump</code>	Restore a custom-format dump
<code>pg_dumpall > cluster.sql</code>	Back up all databases and global objects

Version and Service Checks

Quick checks for server version and service status.

<code>SELECT version();</code>	Show the PostgreSQL server version
<code>psql --version</code>	Show the client version
<code>SHOW server_version;</code>	Show the server version only
<code>sudo systemctl status postgresql</code>	Check the PostgreSQL service state
<code>sudo systemctl restart postgresql</code>	Restart the PostgreSQL service

Related Guides

Use these guides for full PostgreSQL walkthroughs.

[PostgreSQL User Management: Create Users and Grant Privileges](#)

Full guide to roles, passwords, and grants

[How to Check the PostgreSQL Version](#)

Find the installed and running PostgreSQL version

[How to Install PostgreSQL on Ubuntu 20.04](#)

Install PostgreSQL on Ubuntu

[How to Install PostgreSQL on Debian 10](#)

Install PostgreSQL on Debian

[How to Install PostgreSQL on CentOS 8](#)

Install PostgreSQL on CentOS