# pgcli Smarter Postgres shell

**START HERE**

**`pgcli -h <host> -U <user> -d <db>`**
Connect

**`\ns <name> <query>`**
Save a named query

**`pgcli "postgresql://<user>@<host>/<db>"`**
Connect with a URI

**`\q`**
Quit

**`\dt`**
List tables

**`\d <table>`**
Describe a table

**`\x`**
Toggle vertical output

**`\e`**
Edit query in $EDITOR

**Connect** flags, URI, DSN alias → **Prompt** smart completion → **Result** table format, pager → **Config** ~/.config/pgcli/config

## CONCEPTS

pgcli is psql with context-aware completion and syntax colours. By default a query runs on Enter after the closing semicolon; F3 switches to multi-line mode.

Destructive statements (DROP, DELETE, UPDATE without WHERE ...) ask for confirmation; --yes skips it, --warn forces the check.

## CONNECT

**`pgcli -h <host> -p <port> -U <user> -d <db>`**
Connect with explicit options

**`pgcli <db>`**
Connect to a local database

**`pgcli "postgresql://<user>@<host>:<port>/<db>"`**
Connect with a URI
*Add ?sslmode=require for TLS.*

**`pgcli -W`**
Force a password prompt

**`pgcli -w`**
Never prompt for a password

**`pgcli -D <alias>`**
Connect via a DSN alias from the config
*Aliases live in the [alias_dsn] section.*

**`pgcli --list-dsn`**
List configured DSN aliases

## CONNECT (cont.)

**`pgcli -l`**
List databases and exit

**`pgcli --ping`**
Check connectivity and exit

**`pgcli --ssh-tunnel <user@host>`**
Connect through an SSH tunnel

**`pgcli --init-command "<sql>"`**
Run SQL right after connecting

**`\c <db>`**
Switch database

**`\conninfo`**
Show connection details

## INSPECT

**`\l`**
List databases

**`\dt`**
List tables

**`\d <table>`**
Columns, indexes and constraints

**`\d+ <table>`**
Same, with sizes and storage

**`\dn`**
List schemas

**`\dv`**
List views

**`\di`**
List indexes

## INSPECT (cont.)

**`\df`**
List functions

**`\sf <function>`**
Show a function's definition

**`\du`**
List roles

**`\dx`**
List extensions

## RUN SQL

**`pgcli -c "<sql>" <db>`**
Run a statement and exit

**`pgcli -f <file.sql> <db>`**
Run a script file and exit

**`\i <file>`**
Run a file inside pgcli

**`\e`**
Edit the query in your editor
*Uses $EDITOR; the saved text runs when you close it.*

**`\watch <sec>`**
Re-run the query every N seconds

**`\copy <table> from '<file>' csv header`**
Load a CSV file

**`\copy (<query>) to '<file>' csv header`**
Export a query to CSV

**`\h <statement>`**
SQL syntax help

**`\?`**
List all backslash commands

## OUTPUT

**`\x`**
Toggle vertical output

**`pgcli --auto-vertical-output`**
Vertical only when too wide

**`\timing`**
Toggle query timing

**`\T <format>`**
Change the table format
*Try psql, github, csv, vertical, plain, latex.*

**`\o <file>`**
Send results to a file

**`\pager <command>`**
Pipe results through a pager

**`pgcli -t -c "<sql>" <db>`**
Rows only, for scripts

## COMPLETION & KEYS

**F2**
Toggle smart completion

**F3**
Toggle multi-line mode

**F4**
Switch Vi and Emacs mode

**F5**
Toggle explain mode
*Prefixes each query with EXPLAIN.*

**Tab**
Force completion at the cursor

## COMPLETION & KEYS (cont.)

**`Ctrl+Space`**
Start or accept completion

**`Alt+Enter`**
Insert a line break

**`Ctrl+p / Ctrl+n`**
Previous or next history entry

**`Ctrl+r`**
Search history backwards

**`Ctrl+a / Ctrl+e`**
Start or end of line

**`\refresh`**
Reload completion metadata
*Run after DDL from another session.*

## NAMED QUERIES

**`\ns <name> <query>`**
Save a named query

**`\n`**
List named queries

**`\n <name> [args]`**
Run a named query
*Arguments fill $1, $2 ...; quote values with spaces.*

**`\nd <name>`**
Delete a named query

## CONFIG

**`~/.config/pgcli/config`**
Main config file
*Created on first run.*

## CONFIG (cont.)

**`smart_completion = True`**
Context-aware completion

**`multi_line = True`**
Queries need an explicit ; to end

**`table_format = psql`**
Default output format

**`auto_expand = True`**
Expanded only when wide

**`destructive_warning = drop, delete`**
Statements that ask first

**`row_limit = 1000`**
Warn above this many rows

## RECIPES

**Query a remote host with a DSN alias**

```
# config: [alias_dsn]  prod = postgresql://app@db.example.com/app
pgcli -D prod
```

**Parameterised report**

```
\ns by_user select * from orders where user_id = $1
\n by_user 42
```

**Save one result to a file**

```
\o out.txt
select * from <table>;
\o
```

**Tunnel through a bastion**

```
pgcli --ssh-tunnel me@bastion.example.com -h db.internal -U app app
```