

# Psql

## What is psql

Psql is the interactive terminal for working with Postgres. There's an abundance of flags available for use when working with psql, but let's focus on some of the most important ones, then how to connect:

- -h the host to connect to
- -U the user to connect with
- -p the port to connect to (default is 5432)

```
psql -h localhost -U username databasename
```

The other option is to use a full string and let psql parse it:

```
psql "dbname=dbhere host=hosthere user=userhere password=pwhere port=5432 sslmode=requ
```

Once you've connected you can begin querying immediately. In addition to basic queries you can also use certain commands. Running `\?` will give you a list of all available commands, though a few key ones are called out below.

## Commonly used commands

### Turn query timing on

By default the timing of query results will not be available, but we can turn it on by using the following command.

```
# \timing
Timing is on.
```

This will show query timing in milliseconds.

### List tables in database

```
# \d
      List of relations
 Schema | Name      | Type  | Owner
-----+-----+-----+-----
 public | employees | table | craig
(1 row)
```

### Describe a table

```
# \d employees
      Table "public.employees"
  Column |          Type          | Modifiers
-----+-----+-----
  id     | integer                |
  last_name | character varying(50) |
  salary  | integer                |
Indexes:
  "idx_emps" btree (salary)
```

## List all tables in database along with some additional information

```
# \d+
      List of relations
 Schema | Name  | Type  | Owner  | Size  | Description
-----+-----+-----+-----+-----+-----
 public | users | table | jarvis | 401 MB |
(1 row)
```

## Describe a table with additional information

```
#\d+ users
      Table "public.users"
  Column | Type  | Modifiers | Storage | Stats target | Description
-----+-----+-----+-----+-----+-----
  userid | bigint | not null  | plain   |               |
  fullname | text  | not null  | extended |               |
  email   | text  | not null  | extended |               |
  phone   | text  | not null  | extended |               |
  credits | money | default 0.0 | plain   |               |
  parked  | boolean | default false | plain   |               |
  terminated | boolean | default false | plain   |               |
Indexes:
  "users_pkey" PRIMARY KEY, btree (userid)
```

## List all databases

```
# \l
      List of databases
 Name  | Owner  | Encoding | Collate | Ctype | Access privileges
-----+-----+-----+-----+-----+-----
 learning | jarvis | UTF8     | C       | UTF-8 |
```

## List all databases with additional information

```
# \l+
      List of databases
  Name  | Owner  | Encoding | Collate | Ctype | Access privileges | Si
-----+-----+-----+-----+-----+-----+-----
learning | jarvis | UTF8     | C       | UTF-8 |                    | 492
```

## List all schemas

```
# \dn
List of schemas
  Name | Owner
-----+-----
public | jarvis
(1 row)
```

## List all schemas with additional information

```
# \dn+
List of schemas
  Name  | Owner  | Access privileges | Description
-----+-----+-----+-----
public | jarvis | jarvis=UC/jarvis +| standard public schema
      |      | =UC/jarvis        |
(1 row)
```

## List all functions

```
#\df
```

## List all functions with additional information

```
#\df+
```

## Connect to another database

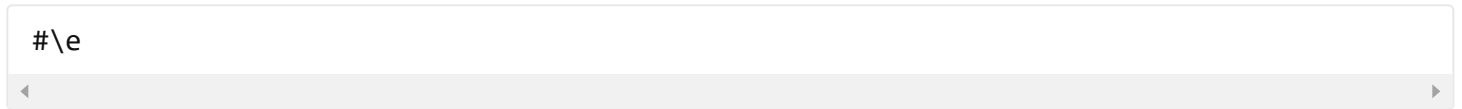
```
#\c dbname
```

## Quit from postgres shell

```
#\q
```

## Text editor inside psql

```
#\e
```

A screenshot of a text editor window. The window has a white background and a light gray border. Inside the window, the text "#\e" is displayed in a monospaced font. Below the text, there is a horizontal scrollbar with a light gray track and a white slider.

This opens your default text editor inside psql shell. Pretty handy for query modifications.

The commands can be given a regex for eg. `\df *_to_array*` lists all functions that contain `_array` in its name.

Of course there are lot many other commands, as said above `\?` will list all of them, and from there we can pick up whatever we want. Psql is a powerful tool once we master it, and since it is command line, we can use it across environments.