# Contents



# 1 Postgres Installation & Setup

Below we provide some basic information about how to install and setup Postgres.

# 2 Postgres Concepts

Postgres is a client-server database. A postgres server process manages database files on disk stored in a directory called the `cluster directory`. This directory stores the content of your databases as well as configuration files.

## 2.1 Terminology

- **Client/Server Architecture**
- **Postgres Cluster**
  - A `directory` on the machine running the server that stores data and configuration files
- **Postgres Server**
  - A postgres server handles the data of single cluster
  - Clients connect to the server via network (TCP/IP)
    - Send commands and receive results
    - per default the network port for postgres is `5432`

# 3 Installation & Administration

There are several ways of how to install postgres. Here are few options listed by operating system.

## 3.1 Installation

You can find installers for most OS here: `https://www.postgresql.org/download/`. We provide some more information about alternative methods.

### 3.1.1 Mac OS X

1. Homebrew If you use homebrew you can install postgres with:

   ```
   brew install postgresql@17
   ```

   - you can either start the server manually or use the `brew services`: `brew services start postgresql@17`

- cluster directory (on Apple Silicon): `/opt/homebrew/var/postgresql@17`

```
$ ls /opt/homebrew/var/postgresql@17
PG_VERSION           pg_commit_ts         pg_ident.conf
↪  pg_notify            pg_snapshots         pg_subtrans
↪  pg_wal               postgresql.conf
base                 pg_dynshmem          pg_logical
↪  pg_replslot          pg_stat              pg_tblspc
↪  pg_xact              postmaster.opts
global               pg_hba.conf          pg_multixact
↪  pg_serial            pg_stat_tmp          pg_twophase
↪  postgresql.auto.conf postmaster.pid
```

2. EDB Installer Download from `https://www.postgresql.org/download/macosx/`.

### 3.1.2 Windows

Download the installer from `https://www.postgresql.org/download/windows/`

### 3.1.3 Linux

1. Ubuntu Postgres is available in available in the Ubuntu package repositories: `https://www.postgresql.org/download/linux/ubuntu/`

   ```
   apt install postgresql
   ```

   If you prefer a newer version, you should add the Postgres repository:

   ```
   sudo apt install -y postgresql-common
   sudo
   ↪  /usr/share/postgresql-common/pgdg/apt.postgresql.org.sh
   sudo apt install postgresql-17
   ```

2. Other Distributions Most distributions provide a postgres package: `https://www.postgresql.org/download/linux/#generic`. Search for postgres or postgresql using your distribution's package manager.

### 3.1.4 Install from source

1. Download the source: `https://www.postgresql.org/ftp/source/`
2. Follow the instructions here: `https://www.postgresql.org/docs/current/installation.html`
3. you will have to install several dependencies
4. you will have to have a C compiler, make, and depending whether you install from git or downloaded the source from the webpage you may have to have `autotools`, `bison`, and `flex` installed.

## 3.2 Creating a Cluster Directory

- **Note: many installation methods will create a cluster for you!**

### 3.2.1 Find your cluster directory if it was created by the installer

If you are using a method that does create a cluster for you, you can determine where it is located by checking what options the postgres server was started with. For instance, on Mac OS X or Linux:

```
ps aux | grep postgres
lord_pretzel     39504   0.0  0.0 411107024   7376 s005  Ss+
↪  Thu09AM   0:25.94 postgres -p 5450 -D
↪  /Users/lord_pretzel/systems/postgres-docker/data_16_nix
```

### 3.2.2 Create your own cluster directory

1. Create a directory at a location of your choice

```
mkdir ~/mypostgres-data
```

2. Run the `initdb` command to initialize the cluster

```
$ initdb ~/mypostgres-data
The files belonging to this database system will be owned by
↪  user "lord_pretzel".
This user must also own the server process.

The database cluster will be initialized with locale
↪  "en_US.UTF-8".
```

```
The default database encoding has accordingly been set to
↪  "UTF8".
The default text search configuration will be set to
↪  "english".

Data page checksums are disabled.

fixing permissions on existing directory
↪  /Users/lord_pretzel/mypostgres-data ... ok
creating subdirectories ... ok
selecting dynamic shared memory implementation ... posix
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting default time zone ... America/Chicago
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok

initdb: warning: enabling "trust" authentication for local
↪  connections
initdb: hint: You can change this by editing pg_hba.conf or
↪  using the option -A, or --auth-local and --auth-host,
↪  the next time you run initdb.

Success. You can now start the database server using:

    pg_ctl -D /Users/lord_pretzel/mypostgres-data -l logfile
    ↪  start
```

## 3.3 Starting & Stopping the Postgres Server

- **Note: depending on your installation method, postgres may have been started for your already!**

### 3.3.1 Determine whether the postgres server is running

1. Mac OS X & Linux You can check whether the postgres server is running by running from the terminal:

   ```
   ps aux | grep postgres
   ```

Incidentally, this will also tell you where your cluster directory is located.

```
ps aux | grep postgres
lord_pretzel     39504   0.0  0.0 411107024   7376 s005
↪  Ss+  Thu09AM   0:25.94 postgres -p 5450 -D
↪  /Users/lord_pretzel/systems/postgres-docker/data_16_nix
```

For example, for this server the cluster directory is located at `/Users/lord_pretzel/systems/pos`

2. Windows Use the `Services console` to check whether the postgres service is running

   (a) Open the Run dialog box by pressing `Win + R`
   - Type `services.msc`
   - Press `Enter`
   - The Services console will open

   (b) Look for the `postgresql-[version]` service where `[version]` is the version of postgres you have installed.

   (c) You can stop / restart postgres from here

### 3.3.2 Starting postgres manually (Mac OS X & Linux)

To start postgres, you can either run the server directly or use `pg_ctl`. Assuming your cluster directory is located at `~/mypostgres-data` (note that this runs postgres in the foreground):

```
postgres -D ~/mypostgres-data/
```

Using `pg_ctl`:

```
pg_ctl start -D ~/mypostgres-data/
```

### 3.3.3 Stopping postgres manually (Mac OS X & Linux)

You can either kill the postgres process or use `pg_ctl`. You need to know where your cluster directory is located (see above). Determine the process id `(PID)` and then kill this postgres.

```
ps aux | grep postgres | grep -e '-D'
lord_pretzel     39504   0.0  0.0 411107024   7376 s005  Ss+
↪  Thu09AM   0:26.02 postgres -p 5450 -D
↪  /Users/lord_pretzel/systems/postgres-docker/data_16_nix

kill 39504
```

Using `pg_ctl`:

```
pg_ctl stop -D
↪  /Users/lord_pretzel/systems/postgres-docker/data_16_nix
```

# 4 Configuration

By default postgres configuration files are stored in the cluster directory, but some installations put the configuration into a separate folder. There are two main configuration files that are of interest:

- `postgresql.conf` - this is the main configuration file for a postgres clusters.
- `pg_hba.conf` - in this file you setup rules that determine which IP addresses can access the server and what authentication methods they are allowed to use
- **Note: you have to restart postgres to force it to reload the configuration if you have updated it.**

## 4.1 Network access

The default setting in most installations is that only connections from the local machine are allowed. To change that, you have to edite `postgresql.conf` and `pg_hba.conf`.

1. Allow connections from any machine (`*`) in `postgresql.conf` (or change it to a comma-separated list of IP addresses that should be allowed to access the postgres server.

```
listen_addresses = '*'
```

2. /Optional: / change port, e.g., use `5555`. Depending on how you start the server, you can also pass the port as a commandline option.

```
port = 5555
```

3. Change the access control rules in `pg_hba.conf`.

Each line in this file represents a rule. When a client tries to connect to the postgres server, it is matched against the rules in this file based on the type of connection (`local` meaning a unix socket, `host` meaning over TCP/IP), the database the client is trying to connect to (`all` matches any databases), the IP address of the client for which postgres uses a pattern of the form `xxx.xxx.xxx.xxx/y` where `xxx.xxx.xxx.xxx` is an IP address and `y` is the number of bits that should be matched, e.g., `127.0.0.1/32` means only allow connections from an IP that matches `127.0.0.1` on all 32 bits of the IP address and `192.168.0.0/16` means match all IP addresses that start with `192.168` and `0.0.0.0/0` matches any IP address. The first rule that matches the client will be used to determine what methods the client is allowed to use to identify themselves. Some example methods are: - `trust`: do not authenticate the user and allow them to connect (**never use this in a real deployment**) - `reject`: deny access - `password`: send password as clear text (**never use this in a real deployment**) - `md5`: send secure hashed password

4. Test access

Use any client of your choice, e.g., the build-in `psql` CLI client or `pgAdmin` to check that you can connect. Assuming that you are connecting as a user `postgres` (the default user for most installations) to a database `postgres` (the default database created by most installations) for a server running on port `5450`:

```
$ psql -h 127.0.0.1 -p 5450 -U postgres -d postgres
psql (16.4, server 16.6)
Type "help" for help.

postgres=#
```

To exit `psql` enter `\q`

# 5 SQL Clients

To run SQL commands on the postgres server you need a client application that connects to the server and allows you to write SQL code that is send to the server.

## 5.1 Starting psql (postgres commandline client)

**psql** is the standard commandline client of Postgres. If you have created a database `cs480` and loaded the university schema into this databases and have a user `postgres`.

```
psql -U postgres -d cs480
```

This will start the **psql** client and connect to a database `cs480` (storing the textbook university schema) using user `postgres`. Within `psql` you can now run SQL commands or control commands (starting with \). For instance `\q` quits the client. See below for an example session.

```
$psql -U postgres -d cs480
psql (9.6.3)
Type "help" for help.

cs480=# SELECT * FROM department;
 dept_name  | building |  budget
------------+----------+-----------
 Biology    | Watson   |  90000.00
 Comp. Sci. | Taylor   | 100000.00
 Elec. Eng. | Taylor   |  85000.00
 Finance    | Painter  | 120000.00
 History    | Painter  |  50000.00
 Music      | Packard  |  80000.00
 Physics    | Watson   |  70000.00
(7 rows)

cs480=# \q
$
```

## 5.2 pgAdmin

A GUI client for administrating a postgres database. Also useful for editing SQL code:

- `https://www.pgadmin.org/download/`

### 5.3 Third party clients

For instance, **SQuirreL SQL** (`http://squirrel-sql.sourceforge.net/`) is a popular 3rd party client written in Java that supports multiple DBMS, **pgcli** (`https://www.pgcli.com/`) is an enhanced version of *psql* written in python.

## 6 Postgres Links and Information

- Postgres official webpage: `https://www.postgresql.org/`

### 6.1 Documentation

- Online Postgres documentation: `https://www.postgresql.org/docs/17/index.html`
- Postgres docu as a PDF: `https://www.postgresql.org/files/documentation/pdf/17/postgresql-17-A4.pdf`

### 6.2 Programming Language APIs

How to connect to postgres programmatically from various languages. There is a list of interfaces at `https://www.postgresql.org/download/products/2-drivers-and-interfaces/`. Below are the direct links for common languages.

- **Java**: `https://jdbc.postgresql.org/`
- **C**: **libpq** is part of Postgres
- **C++**: `http://pqxx.org/development/libpqxx/`
- **ODBC**: `https://www.postgresql.org/ftp/odbc/versions/`
- **Python**: `http://www.psycopg.org/psycopg/`