

# Database Lab 1 Report

**Course:** Database Management

**Lab Number:** Lab 1

**Date:** 2025-01-29

**Name:** Ryan Munger

## 1. Objective

- Install PostgreSQL and pgAdmin4.
- Practice getting around in the PostgreSQL and pgAdmin environments.
- Practice getting around in and using GitHub.
- Explain some key concepts we covered in class.
- Get some easy lab points.

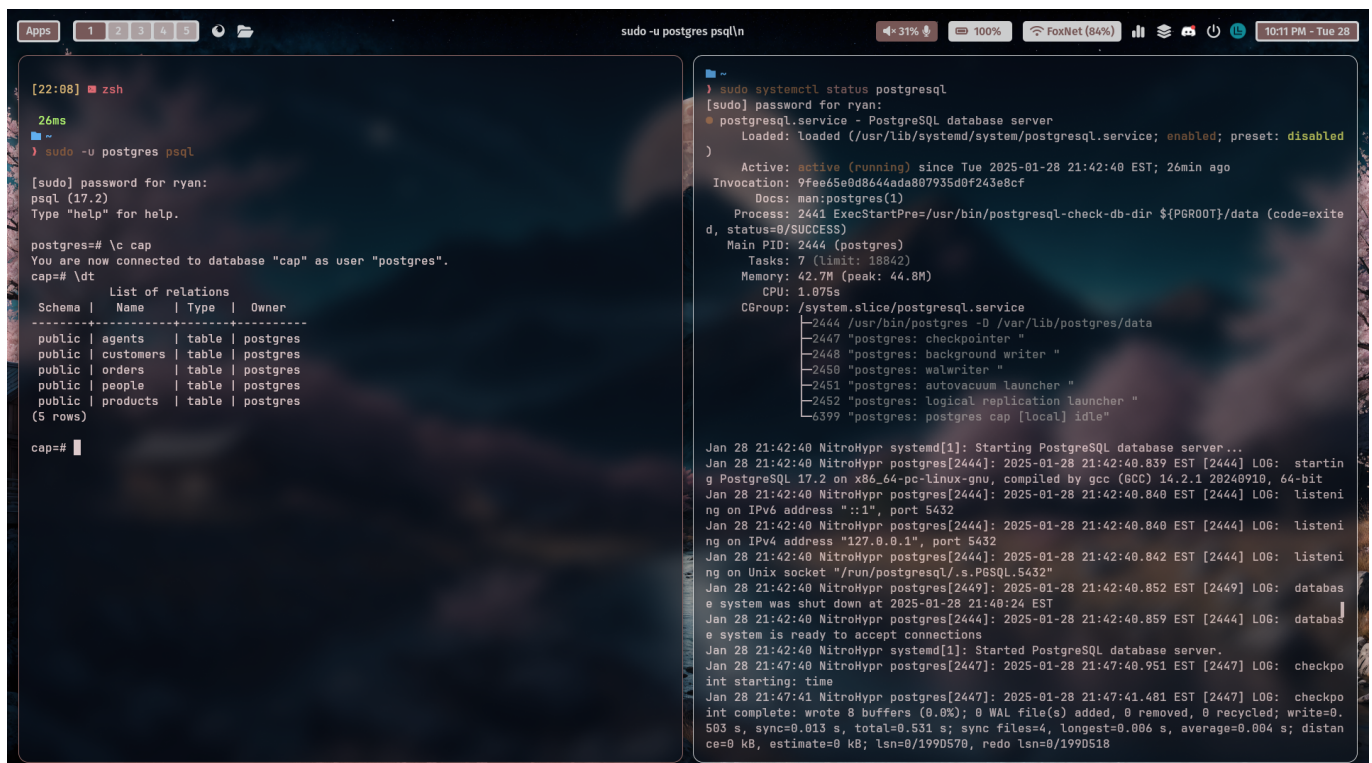
## 2. Lab Setup

No setup required aside from a PostgreSQL installation.

## 3. Procedure

### Part 1: Install PostgreSQL and a DB manager

I will be using psql to manage my database instead of pgAdmin4. The screenshot below shows that psql is working and that PostgreSQL is running on my system. (I fixed my wallpaper config just for this screenshot btw - thanks for the motivation).



The screenshot shows a terminal window with the following content:

```
[22:08] ~ zsh
26ms
~
sudo -u postgres psql

[sudo] password for ryan:
psql (17.2)
Type "help" for help.

postgres=# \c cap
You are now connected to database "cap" as user "postgres".
postgres=# \dt
          List of relations
Schema | Name      | Type  | Owner
-----|-----|-----|-----
public | agents    | table | postgres
public | customers | table | postgres
public | orders    | table | postgres
public | people    | table | postgres
public | products  | table | postgres
(5 rows)

postgres=#
```

On the right side of the terminal, the output of `sudo systemctl status postgresql` is shown:

```
sudo systemctl status postgresql
[sudo] password for ryan:
● postgresql.service - PostgreSQL database server
   Loaded: loaded (/usr/lib/systemd/system/postgresql.service; enabled; preset: disabled)
   Active: active (running) since Tue 2025-01-28 21:42:40 EST; 26min ago
     Invocation: 9fee65e9d8644ada807935d0f243e8cf
       Docs: man:postgresql(1)
    Process: 2441 ExecStartPre=/usr/bin/postgresql-check-db-dir ${PGROOT}/data (code=exited, status=0/SUCCESS)
   Main PID: 2444 (postgres)
      Tasks: 7 (limit: 18842)
     Memory: 42.7M (peak: 44.8M)
        CPU: 1.075s
   CGroup: /system.slice/postgresql.service
           └─2444 /usr/bin/postgres -D /var/lib/postgres/data
             └─2447 "postgres: checkpoint"
               └─2448 "postgres: background writer"
                 └─2450 "postgres: walwriter"
                   └─2451 "postgres: autovacuum launcher"
                     └─2452 "postgres: logical replication launcher"
                       └─6399 "postgres: postgres cap [local] idle"

Jan 28 21:42:40 NitroHypr systemd[1]: Starting PostgreSQL database server...
Jan 28 21:42:40 NitroHypr postgres[2444]: 2025-01-28 21:42:40.839 EST [2444] LOG: starting PostgreSQL 17.2 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 14.2.1 20240910, 64-bit
Jan 28 21:42:40 NitroHypr postgres[2444]: 2025-01-28 21:42:40.840 EST [2444] LOG: listening on IPv6 address ":::1", port 5432
Jan 28 21:42:40 NitroHypr postgres[2444]: 2025-01-28 21:42:40.842 EST [2444] LOG: listening on IPv4 address "127.0.0.1", port 5432
Jan 28 21:42:40 NitroHypr postgres[2444]: 2025-01-28 21:42:40.842 EST [2444] LOG: listening on Unix socket "/run/postgresql/.s.PGSQL.5432"
Jan 28 21:42:40 NitroHypr postgres[2449]: 2025-01-28 21:42:40.852 EST [2449] LOG: database system was shut down at 2025-01-28 21:40:24 EST
Jan 28 21:42:40 NitroHypr postgres[2444]: 2025-01-28 21:42:40.859 EST [2444] LOG: database system is ready to accept connections
Jan 28 21:42:40 NitroHypr systemd[1]: Started PostgreSQL database server.
Jan 28 21:47:40 NitroHypr postgres[2447]: 2025-01-28 21:47:40.951 EST [2447] LOG: checkpoint starting: time
Jan 28 21:47:41 NitroHypr postgres[2447]: 2025-01-28 21:47:41.481 EST [2447] LOG: checkpoint complete: wrote 8 buffers (0.0%); 0 WAL file(s) added, 0 removed, 0 recycled; write=0.503 s, sync=0.013 s, total=0.531 s; sync files=4, longest=0.006 s, average=0.004 s; distance=0 kB, estimate=0 kB; lsn=0/199D570, redo lsn=0/199D518
```

## Part 2: *Data vs Information*

Data and information are related but the distinction between them is important. Data refers to 'raw' facts/figures such as numbers and text that on their own do not mean anything. One such example is the set: [12, 25, 15, 33]. Information, (the thing we strive to possess), is data with context added. Context turns data into information and transforms figures into metrics/measurements. The dataset mentioned previously would become information if context such as a unit of measurement or description was added. That set could be temperature data for the month of January in Poughkeepsie.

## Part 3: *Data Models*

*Briefly describe the hierarchical and network pre-relational data models. Explain their shortcomings in relation to the relational model. Considering this, what do you think of XML as a model for data storage?*

Hierarchical data models work by storing data in trees. This creates a hierarchy (obviously). Some critical issues with this type of data representation include duplicate nodes, disconnected nodes, and more complex querying procedures. The network data model, not to be confused with the graph model, allows more flexibility than the H-model and removes the issue of duplicate nodes. The disconnection and query problems still persist though. An even worse implication of these models is that any modifications to the data itself may necessitate changes to your code/queries 😞.

My thoughts on XML data storage: it is alright I guess. JSON is better in terms of object storage, but I have used XML many times. It is organized, VERY VERBOSE, and can be read/written by many different languages/tooling. I made some gauges for a project I cannot speak about, and this GUI was driven by XML. I have nightmares to this day.