

AI-04012 Postgres 18 Lab Notes

Owner	Ⓒ Rigel Arcayan
Tags	Project
Created time	July 10, 2026 7:29 AM

I'm seeing 6 db now! ;-)) Thank you very much. In this conversation, there were a lot of things to learn. Create a summary of the instructions that worked so that I can retrace back for later practice of starting over and studying how the db creation worked. I'm new to postgres and I want to become an expert by not just following your scripts and copy and pasting. Looking forward, I sensed that db knowledge is a must for cloud admin.

You should be incredibly proud! Getting a multi-database, replicated PostgreSQL environment up and running in Docker—and troubleshooting real-world pathing and script errors along the way—is a massive milestone.

You are completely right: **Database mastery is an absolute superpower for a Cloud or DevOps Administrator.** Everything in the cloud eventually relies on persistent data.

Here is your master architectural and command blueprint summarizing how we solved these hurdles, why things worked, and how to reset your lab for practice later.

The Architectural Lessons (The "Why")

1. Docker Networking vs. Host Networking

- **Inside Docker:** When pgAdmin or your helper container connects to Postgres 18, they talk *container-to-container* over the shared `dockerlab-network`. Inside this network, the database always listens on its **internal port (5432)** using its container name (`pg18book`) as the domain name.
- **Outside Docker:** Your desktop machine can't see into that virtual network directly, which is why your compose file opened a window to the outside

world using `5433:5432` . External tools must use port `5433` on `localhost` .

2. Standard SQL vs. `psql` Meta-Commands

- When you saw the syntax error near `\set ON_ERROR_STOP on` , you learned that lines starting with a backslash (`\`) are **native `psql` command-line utility features**, not core SQL.
- Graphic tools like pgAdmin's Query Tool only understand standard ANSI SQL, which is why we had to stream the scripts through a native `psql` client terminal.

3. Context & Working Directories

- Script files like `master_setup.sql` often use relative lines like `\i replication/file.sql` .
- For those paths to work inside Docker, we used `w /scripts` to force the `psql` engine to set its working directory directly where those files live on your drive.

The "Reset & Re-Run" Practice Playbook

When you want to practice building this from scratch, open your terminal inside `~/dockerlab/infrastructure/postgres18` and use these commands:

Step 1: Wipe the Slate Clean (Teardown)

This executes the book's rollback scripts, drops the custom user roles, and completely safely deletes the 5 book databases.

Bash

```
docker run --rm -i --network dockerlab-network \  
-v $(pwd)/psql_scripts:/scripts \  
-w /scripts \  
postgres:18-alpine \  
psql -h pg18book -U postgres -d postgres -f master_teardown.sql
```

Step 2: Build it Fresh (Setup)

This launches a transient (temporary) container that reads your locally modified `master_setup.sql` file and pipes the instructions straight over the virtual network to the `pg18book` database.

Bash

```
docker run --rm -i --network dockerlab-network \
-v $(pwd)/psql_scripts:/scripts \
-w /scripts \
postgres:18-alpine \
psql -h pg18book -U postgres -d postgres -f master_setup.sql
```

Next Self-Study Steps to Become an Expert

Instead of just letting the scripts run blindly, use your newly opened pgAdmin environment to inspect *how* the book built this layout. Open a **Query Tool** on your databases and look at these files:

- **Study how the databases were created:** Look inside your local folder at `psql_scripts/database_definitions/create_databases.sql`.
- **Study how tables and relations are built:** Open `psql_scripts/database_definitions/ecommerce_reference_data.sql` to study `CREATE TABLE` structures, primary keys, and indices.
- **Explore the schemas visually:** In pgAdmin, expand the `aidb` or `central_analytics` databases, drill down into **Schemas**, and inspect the differences between the `public` schema and custom schemas the book built for you.

You've built an incredible sandbox environment here. Happy hacking and studying!