

Database Administration

Lab 03: Performance Monitoring and Diagnosis.

Andrés Oswaldo Calderón Romero, Ph.D.

February 16, 2026

1 Introduction

In this lab, we will focus on an essential aspect of database administration: monitoring and diagnosing the performance of our PostgreSQL systems. Beyond simply running queries, an effective DBA must be able to observe how the database behaves in real time, identify bottlenecks, and detect trends that could impact performance.

To achieve this, we will simulate a continuous stream of sensor data being inserted into a PostgreSQL table using a single Python or Bash script. This live feed will serve as our testing ground for implementing performance monitoring techniques. We will then integrate PostgreSQL with Grafana to build an interactive dashboard, allowing us to visualize metrics, track trends, and gain actionable insights into query execution and system health.

By the end of this lab, you will not only have a functioning monitoring setup but also a deeper understanding of how to analyze ongoing database activity, skills directly applicable to production environments.

2 Additional Sections in the Textbook

As you have already seen, Chapter 8 of “*PostgreSQL 16 Administration Cookbook*” (Ciolli et al., 2023) is quite extensive. In class, we examined only four of the most relevant recipes. You will now read and summarize three additional ones:

- Monitoring the progress of commands (page 352).
- Understanding why queries slow down (page 368).
- Tracking important metrics over time (page 374).

Prepare concise summaries of these three recipes and include them in your report. These summaries must be **handwritten** and submitted as a **snapshot** to be considered valid.

3 Grafana + PostgreSQL

Grafana is an open-source analytics and visualization platform that enables you to query, explore, and display data through interactive dashboards. It supports multiple data sources, including PostgreSQL.

Watch the following videos to learn how to connect Grafana to a PostgreSQL table containing sensor data: [video 1](#) and [video 2](#).

The SQL code to create the table is provided below:

```
CREATE TABLE IF NOT EXISTS public.sensor
(
    oid bigint NOT NULL,
    lat real NOT NULL,
    lon real NOT NULL,
    tim timestamp with time zone NOT NULL,
    CONSTRAINT sensor_pkey PRIMARY KEY (tim)
)
```

You can also download the code used to generate the data from [here](#). Reproduce the steps shown in the videos to gain a better understanding of how to display and monitor trends in your database. There is no deliverable required for this section; however, you will perform a similar task using alternative tools in the autonomous work.

4 Autonomous Work

As a practice, in addition to preparing the summaries for section 2, you will review the following [document](#). This resource contains information about various relational DBMSs—both commercial and open source and, towards the end, presents alternatives to Grafana for visualization and monitoring.

Your task is to select one new RDBMS and one visualization tool from the document, and create a tutorial similar to the one you previously followed in Section 3.

We expect you to submit a well-structured report in **PDF** format by **March 2, 2026** before **2:00pm**.

Happy Hacking 😎!