

Backup – Netflix Database

**Data Processing INT 2024-2025**

Group members:

Ofelia Hantoiu - 5280737

Mihaela Covali - 5292891

Erika Nicolau - 5326745

Prerequisites

Before you start backing up your PostgreSQL database, make sure you have these tools ready:

* pgAdmin: A handy tool to manage PostgreSQL databases.
* pgAgent: An add-on for pgAdmin that helps you schedule and automate tasks like backups.

Backup Plan

We’ve set up a backup plan using pgAgent to make managing and automating database tasks easy. Once it’s installed and set up, you can start creating jobs to handle things like backups without much hassle.

Our backup plan is split into three levels: **daily**, **weekly**, and **monthly** backups. Each level is designed to cover everything you might need.

Daily Backup

Daily backups focus on saving the Netflix database’s structure and data. Here’s the command you’ll use:

pg\_dump --username=postgres --dbname=Netflix --clean --file=/backup/daily/`date +%Y-%m-%d`/netflix.sql pre-data data

This command uses pg\_dump to grab a clean copy of the database, making sure the schema and data are included.

Weekly Backup

Weekly backups cover more ground by including not just the database, but also things like triggers, views, functions, and roles. Use this command:

pg\_dump --username=postgres --dbname=Netflix --clean --file=/backup/weekly/`date +%Y-%m-%d`/netflix.sql --section=pre-data,data,post-data

The --section option lets you include pre-data (schema), data, and post-data (extras like triggers and roles) for a more complete backup.

Monthly Backup

Monthly backups go all out. They include everything on the server—all databases and their objects, like triggers, schemas, and constraints. Here’s the command:

pg\_dump --username=postgres --dbname=DATABASE --clean --file=/backup/monthly/`date +%Y-%m-%d`/DATABASE.sql

This gives you a full snapshot of the entire server, neatly organized for easy storage.

Observations

There is a reason why we have choose not to use pg\_dumpall, which can back up everything at once. While that’s an option, using pg\_dump in parallel is quicker and breaks the databases into separate .tar files. This makes managing multiple databases much simpler.