SQL Backup document

Data Processing Project

Caterina Aresti  
Joey Krämer  
Nicanor Martinez  
Zsombor Hajzer

2024

# SQL Backup document

Table of content

[SQL Backup document 1](#_Toc155953607)

[Prerequisites 2](#_Toc155953608)

[Introduction and Backup strategy levels 2](#_Toc155953609)

[Daily Backup 2](#_Toc155953610)

[Weekly Backup 2](#_Toc155953611)

[Monthly Backup 3](#_Toc155953612)

[Notes on “pg\_dump” 3](#_Toc155953613)

## Prerequisites

Before starting with the backup of the postgreSQL database, make sure the following components are installed:

* “pgAdmin”: it is a comprehensive PostrgeSQL management tool,
* “pgAgent”: it is an extension for “pgAdmin” that facilitates the scheduling and execution of tasks, such as database backups.

## Introduction and Backup strategy levels

The backup strategy we chose for this project includes using an extension called “pgAgent” for our postgreSQL database management. It can serve as a job scheduler and executor for PostgreSQL, allowing the automation of various database-related tasks.

When pgAgent is installed and a relation is build as an extension, you can start creating jobs for all kind of operations, including backups.

We have designed our backup strategy creating three distinct levels: daily, weekly and monthly.

## Daily Backup

In the daily level, we focused on including only backups of the Netflix database, specifically its structure and its data. The command is this:

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

This command uses the "pg\_dump" utility with specific options to make sure a clean backup of the Netflix database is done, recording both the pre-data and data sections.

## Weekly Backup

In the weekly level, we concentrated on including backups of the Netflix database, along with some database objects such as triggers, views, functions, and roles. The command is this:

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

Here, the "--section" flag allows the selective inclusion of pre-data, data, and post-data sections in the backup, guaranteeing a more complete picture of the database.

## Monthly Backup

In the monthly level, we decided to cover a complete server backup. This includes backing up all databases present on the server, along with all the associated objects such as triggers, schemas, functions, constraints, and more. The command is this:

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

This command captures the entire “DATABASE” and provides a clean and organized backup file for archival purposes.

## Notes on “pg\_dump”

While there is a “pg\_dumpall” command available for backing up all databases simultaneously, using the “pg\_dump” in parallel provides faster backup time and divides all the databases into physically different .tar files. This division facilitates easier management, especially involving multiple databases.