Evolution of PostgreSQL Job Schedulers

Rajesh Kandasamy

Technical Consultant
Fujitsu

PG Down Under 2025

Agenda

- Introduction to Job schedulers
- Need for Job scheduler
- pgAgent
  - Overview of pgAgent
  - Setup pgAgent
  - Limitations
- pg_cron
  - Overview of pg_cron
  - Setup pg_cron
  - Limitations
- pg_timetable
  - Overview of pg_timetable
  - Setup pg_timetable
  - Limitations
- Use-cases and Comparison
- Q&A

Introduction to Job schedulers

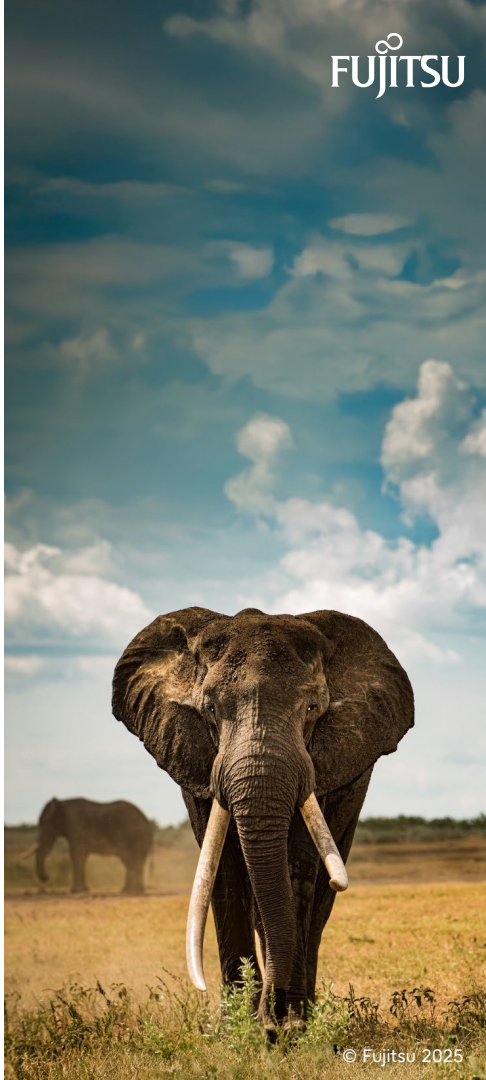
PostgreSQL is a powerful relational database system, but it does not include a native job scheduler

But often database administrators and developers need to run recurring tasks, housekeeping tasks and automate maintenance jobs

Scheduling is usually handled at the **OS level** (cron, systemd, Windows Task Scheduler, etc.).

For in-database scheduling, you need DB **extensions**

Some **cloud providers** (like AWS RDS or Azure Database for PostgreSQL) also offer managed scheduling options.



Why we need Job schedulers?

FUJITSU

Routine maintenance

Automate housekeeping tasks such as vacuuming, reindexing, or statistics updates.

Data import and export

Schedule regular loading of incoming files or exporting data for downstream systems

Backup/Restore operations

Ensure backups run on time and restore tests happen periodically

Analytical processing

Trigger reporting queries, aggregations, or ETL pipelines at scheduled intervals

Monitoring and alerts

Run health checks and raise alerts if thresholds are crossed

External Integrations

Invoke scripts, APIs, or other system processes from within or alongside the database

pgAgent

FUJITSU

pgAgent

Introduced as part of pgAdmin 3 project

- The job scheduling agent runs as a separate service or daemon
- Written in C++
- Key features
  - Multi-step jobs (SQL + shell commands)
  - Runs across multiple servers
  - GUI integration with pgAdmin
- pgAdmin act as an interface for
  - Defining pgAgent job
  - Scheduling job and
  - Monitoring the job
- Supports PostgreSQL **versions 9.1 or newer**

pgAgent ● Setup

FUJITSU

- Install pgAgent from pgdg Red Hat repository
- Create database extension pgAgent
- CREATE LANGUAGE plpgsql;
- Run pgAgent as a daemon
- Create pgAgent job from pgAdmin interface
  - Define the steps, including connection type, code or scripts
  - Add a schedule for the job to run
  - Monitor the database job
  - Modify the job

pgAgent ● Limitations

FUJITSU

- pgAgent requires a separate external agent (daemon) running on the OS
- More complex to configure and maintain — especially cross-platform
- Limited to PostgreSQL jobs, but executed from outside the DB
- One job step runs at a time; concurrency is limited
- Not HA-aware; job execution is tied to the pgAgent host

pg_cron

pg_cron

A lightweight SQL Job scheduler for PostgreSQL

- Developed by **CitusData** (now part of Microsoft)
- A PostgreSQL extension that enables scheduling SQL commands using **OS crontab-like syntax**
- Well-suited for **simple SQL maintenance tasks**: VACUUM, ANALYZE updating stats, generating reports, etc
- Very **low overhead** since it runs inside PostgreSQL as a background worker
- Jobs are defined inside PostgreSQL and executed by a background worker process
- Supports PostgreSQL **version 10 or newer**

pg_cron ● Setup

FUJITSU

pg_cron can be installed in few simple steps

- Install pgAgent from pgdg for APT or Red Hat repository
- Add **`pg_cron`** to the **`shared_preload_libraries`** parameter
- Restart the PostgreSQL cluster
- Create the database extension pg_cron
- Examples:
  - `SELECT cron.schedule('reindex_tbl5','* 22 * * *','REINDEX TABLE tbl5');`
  - `SELECT cron.schedule('analyze-tbl5','0 2 * * *','VACUUM ANALYZE tbl5’);`
  - `SELECT cron.schedule_in_database(‘vac', '0 4 * * 0', 'VACUUM', ‘appdb');`

pg_cron ● Limitations

FUJITSU

- Cannot run external programs or shell scripts (**SQL-only**)
- Not suitable to run complex workflows that require dependencies or combining multiple tasks
- Dependent on the PostgreSQL instance — jobs do not run if the instance is down, and missed jobs are not replayed later
- Limited error handling and logging compared to advanced schedulers

pg_timetable

FUJITSU

pg_timetable

FUJITSU

A standalone binary (written in Go) that connects to PostgreSQL

- Provides an **advanced job scheduler** with support for task chains, retries, and conditional logic
- Runs as an **external daemon** separate from PostgreSQL
- Supports execution of both **SQL jobs** and **external shell scripts/programs**
- Can **retry missed runs** after downtime
- Supports PostgreSQL **version 11 or newer**

pg_timetable • Setup

FUJITSU

- Install **Golang** (required for building from source)
- Clone the **pg_timetable** GitHub repository
- Create a **dedicated database user** and schema for job management (optional)
- Build the **binary from source** (or download pre-built binaries/Docker image)
- Start the **pg_timetable daemon**, which connects to PostgreSQL
- Examples:
  - `SELECT timetable.add_job('run_vacuum','30 22 * * *','VACUUM');`

pg_timetable ● Limitations

FUJITSU

- Requires **separate installation** and setup
- Depends on a **standalone external daemon** running on the OS
- Setup and configuration are **more complex** compared to pg_cron or pgAgent
- **Steeper learning curve** due to advanced features and flexibility

Use cases

Use cases - pgAgent

- For multi-step job scheduling - pgAgent
  - Data import and export
  - Data migration consisting of multiple steps
  - For PostgreSQL legacy versions using pgAdmin

Use cases – pg_cron

- Simple database maintenance tasks
  - Vacuum
  - Reindex
  - Analyze
  - Copy TO / Copy FROM
  - Refresh Materialized views
- Can be run from container-based environment where Postgres is installed.
- In the DBaaS or PaaS, where OS level access is not available.

Use cases – pg_timetable

- Advanced job scheduler for complex tasks – pg_timetable
  - Supports job chains or dependent tasks
  - Complex ETL jobs
  - Automation pipelines
  - External integration with ELK or Prometheus

Comparison

FUJITSU

Feature	pgAgent	pg_cron	pg_timetable
SQL jobs	✓	✓	✓
Shell scripts	✓	✗	✓
GUI support	✓ pgAdmin	✗	✗
Setup complexity	Medium	Low	Medium/High
Advanced workflows	Limited	✗	✓
Best for	GUI users, legacy	Simple tasks	Complex pipelines

Thank you

Rajesh Kandasamy

Technical Consultant
Fujitsu