**Engineer**

# 30 days of electricity data in 180 ms

Engineered hourly electricity-consumption aggregation variants in Python, SQL and Bash + Jq, with the Python benchmark processing one 30-day JSONL dataset in 180ms.

2025

## 180 ms

to aggregate a month of meter readings

**Situation.** The company processes large volumes of heterogeneous electricity data coming from multiple data sources, each with its own reporting frequency — ranging from hourly intervals to 15-minute intervals, and in some cases, irregular timestamps. This variability poses a challenge when trying to create a coherent, comparable dataset. To support accurate energy analytics, this data needs to be normalized into consistent hourly consumption values, aggregated per zone.

**Task.** The objective was to take raw event data from a historical dataset (in JSONL format) and convert it into hourly-aligned electricity consumption figures, structured as one row per hour and per zone. Specifically, the task involved:

- Aligning timestamped data to strict hourly intervals,
- Aggregating total electricity production within each interval by summing mix values,
- Incorporating cross-border exchanges by adding imports and subtracting exports,
- Outputting the final values in a structured, scalable format suitable for further analysis.
- The solution also needed to be efficient enough to scale for large time windows (30+ days) and across multiple countries/zones.

**Action.** To approach this task, three different variants of the solution were implemented using Python, JQ (for command-line JSON processing), and SQL, each optimized for different contexts:

Python was chosen for its flexibility, ease of data manipulation, and ability to handle in-memory transformations efficiently. Using pure Python functions, a pipeline was built that:

- Parsed the JSONL files into data frames
- Resampled the time series data to hourly intervals
- Aggregated production and calculated net electricity consumption (production + imports – exports)
- Exported the results as CSV or loaded them into a lightweight SQLite database for inspection.

JQ was used to create a quick, minimal-dependency solution for command-line environments, built as a JQ filter chain.

In PostgreSQL the JSONL data was imported, normalized tables created, and a series of SQL queries written.

To evaluate performance, each solution was benchmarked with both 1-day and 30-day historical datasets.

**Result.** The Python implementation emerged as the fastest and most scalable solution, completing:

- 1-day data processing in just 34ms
- 30-day dataset in 180ms

This confirmed its suitability for handling larger timeframes while maintaining sub-second performance. It also offered clear, maintainable code that could be easily extended to multi-zone processing or integrated into an ETL pipeline.

Overall, the multi-tool approach demonstrated flexibility in tool usage, strong performance optimization, and robust handling of time-alignment and aggregation challenges in real-world electricity data pipelines.

### Bring this experience to your project

Use “30 days of electricity data in 180 ms” as a starting point for a similar challenge in your organisation. The services behind this work are available for your project. Finding the queries and the missing indexes behind a system that has got heavier every year, and fixing the ones that matter.

**Explore our services** Get in touch →

### Look behind the result

Read the situation, decisions and result in this example. We can discuss which parts transfer to your constraints and which need another approach. 30 days of electricity data in 180 ms — 180 ms to aggregate a month of meter readings

**Explore our services** Get in touch →

## Need fast, efficient data pipelines?

Talk through your project with the engineer responsible for delivery. We agree the scope, price basis and handover in writing before work begins.

Role context: Platform and Data Engineer. It draws on Databases, PostgreSQL, SQL, Data Engineering, Data Pipelines (ETL/ELT), Data Analytics, Automation & CI/CD, Performance Tuning and Python. And while you are here, we are rather good at Database Performance Tuning, Data Pipeline Development (ETL/ELT) and Data Analytics & BI Dashboards.

**Get in touch** Explore our services

## One piece of **a longer record** — and it is all on this site.

Every entry is written the same way: the situation, the task, what we did and what changed.

**See the whole record**

### Define the improvement you need

Bring a representative dataset, the questions it must answer and the current bottleneck. We can agree how to measure quality, throughput and recovery.

**See what it costs** Get in touch →

# Need fast, efficient data pipelines?

Ask about the decisions behind “30 days of electricity data in 180 ms”. You will speak with the people who can explain the work and assess your own project.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://engineer.company/portfolio/a-month-of-meter-readings-in-180-milliseconds/