PostgreSQL Partitioning: An Overview of solutions with a deep dive into implementation details and use cases (Josef Machytka: Speaker portfolio)

Josef Machytka

1 min read · Feb 9, 2025

(See my bio, other talks in my portfolio and my speaker experience in the covering article.)

Duration: 45 minutes
Target Audience: App developers, system architects

Overview: In this presentation, we will discuss in detailes PostgreSQL's approach to inheritance and native partitioning. We will check database parameters that affect performance of these solutions, including rarely used enable_partitionwise_aggregate and enable_partitionwise_join parameters, will compare the results of performance tests between a single large table and partitioned tables on different datasets, look at efficiency of indexes, and discuss the application of multi-level partitioning for different use cases. Additionally, we will also share some tips and tricks — how to subsequently implement partitioning on existing table, or how to use inheritance for multiple interesting use cases. We will also discuss memory usage of connections which use multiple partitions in queries.

Key Takeaways:

- In-depth understanding of PostgreSQL implementation
- Practical tips and tricks for performance optimization and implementation of partitions

- Practical insights into memory usage related to partitioning

Slides: not available online

Presented at:

- NetApp internal workshop 2024.11.12

Picture created by the author using DeepDreamGenerator

Postgresql Partitioning Inheritance

Edit profile

Written by Josef Machytka

68 Followers · 25 Following

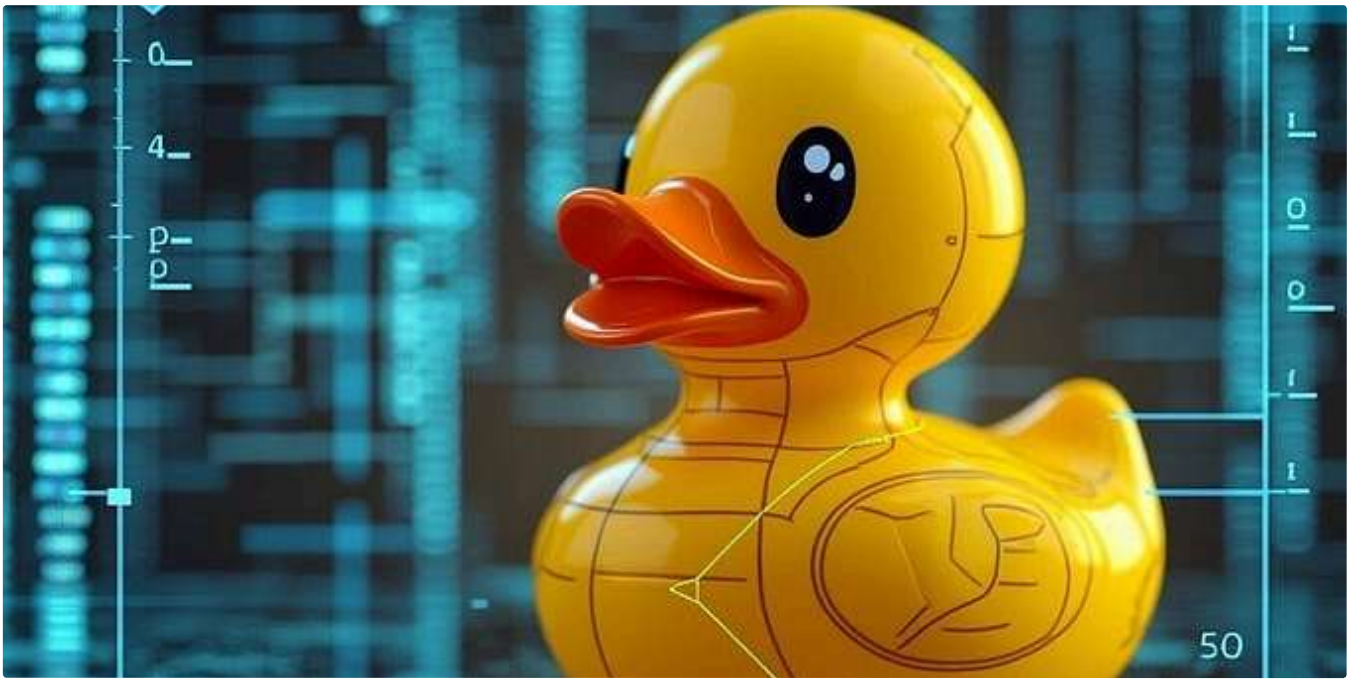
I work as PostgreSQL specialist & database reliability engineer at credativ GmbH.

No responses yet

Josef Machytka he/him

What are your thoughts?

More from Josef Machytka

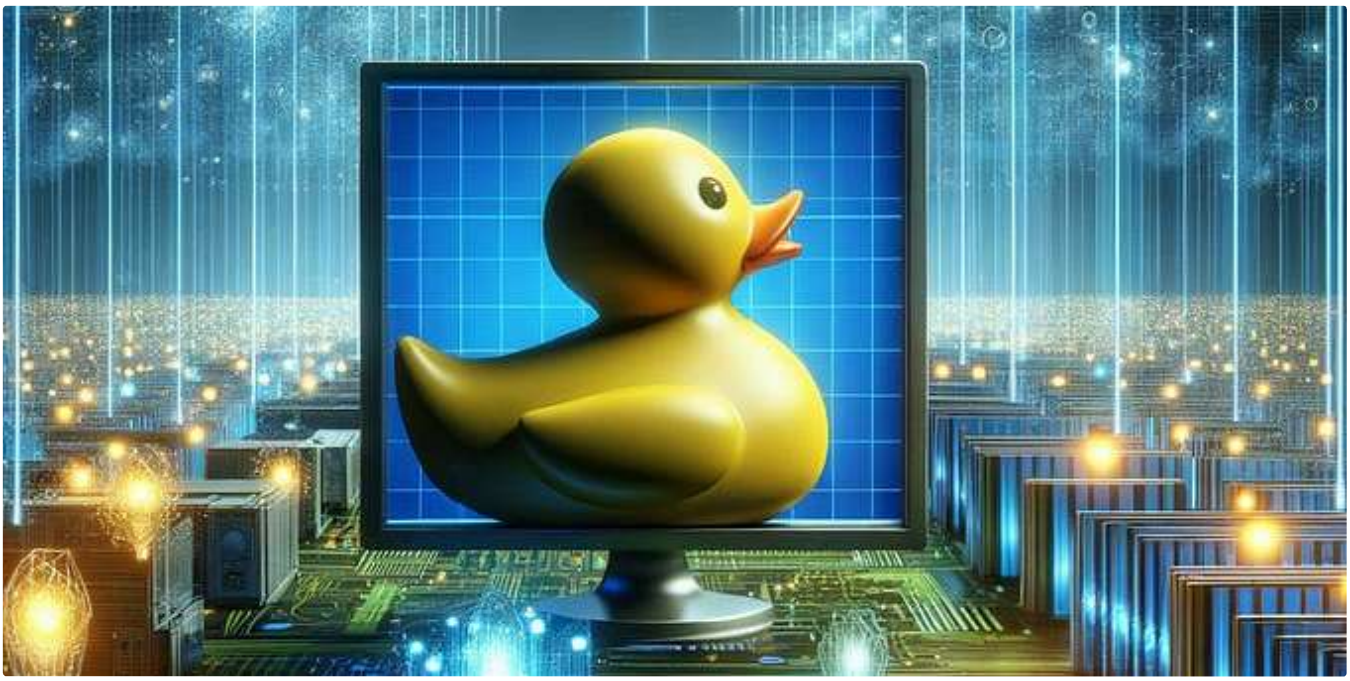


Josef Machytka

DuckDB Database File as a New Standard for Sharing Data?

This is not my original idea; I came across it in an excellent article titled “DuckDB Beyond the Hype” by Alireza Sadeghi. However, it...

Dec 30, 2024 24 3



Josef Machytka

Quick and Easy Statistics and Histograms with DuckDB

DuckDB is an exceptional tool that demonstrates how tasks requiring sometimes considerable manual effort in other tools can be accomplished...

Dec 16, 2024 26

Josef Machytka

PostgreSQL JSONB Operator Classes of GIN Indexes and Their Usage

Throughout 2024, I worked on an internal project exploring the use of JSONB data in PostgreSQL and its various indexing options. During...

Jan 8 5

Bob	2100.0	600.0	
Charlie	2300.0	1500.0	1100.0

```
D pivot pg.sales on (product,year) using sum(sales_amount) group by salesperson order by salesperson;
```

salesperson varchar	(Laptop, 2022) double	(Laptop, 2023) double	(Phone, 2022) double	(Phone, 2023) double	(Tablet, 2022) double	(Tablet, 2023) double
Alice	1200.0	1400.0	800.0	900.0	300.0	400.0
Bob	1000.0	1100.0	600.0			
Charlie	1100.0	1200.0	700.0	800.0	500.0	600.0

```
D pivot pg.sales on (year,product) using sum(sales_amount) group by salesperson order by salesperson;
```

salesperson varchar	(2022, Laptop) double	(2022, Phone) double	(2022, Tablet) double	(2023, Laptop) double	(2023, Phone) double	(2023, Tablet) double
Alice	1200.0	800.0	300.0	1400.0	900.0	400.0
Bob	1000.0	600.0		1100.0		
Charlie	1100.0	700.0	500.0	1200.0	800.0	600.0

```
D pivot pg.sales on (year) using sum(sales_amount) group by salesperson order by salesperson;
```

Josef Machytka

Easy and Intelligent Pivot Tables with DuckDB

After exploring the various capabilities of DuckDB in my earlier articles, I want to focus more on its powerful data analytical...

Dec 4, 2024 7 1

See all from Josef Machytka

Recommended from Medium

Adarsh Rajesh

Understanding Collations in PostgreSQL

Sorting, ordering and comparisons are fundamental to any database. However, are they always deterministic and consistent across all...

5d ago 8

Ramin Orujov

PostgreSQL Story: How to optimize the query by refactoring data model

Summary

Feb 17 4

Lists

Staff picks

819 stories · 1637 saves

Stories to Help You Level-Up at Work

19 stories · 944 saves

Self-Improvement 101

20 stories · 3324 saves

Productivity 101

20 stories · 2796 saves

Query Optimization Method	Execution Time	Performance Improvement (vs. No Index)
No Index	42,049 ms (≈42 sec)	Baseline
With B-tree Index	9,684 ms (≈9.7 sec)	77% faster
With Chunk-Skipping Index + Columnstore	304 ms (0.3 sec)	99.28% faster

In Timescale by Team Timescale

Handling Billions of Rows in PostgreSQL

Here's how to scale PostgreSQL to handle billions of rows using Timescale compression and chunk-skipping indexes.

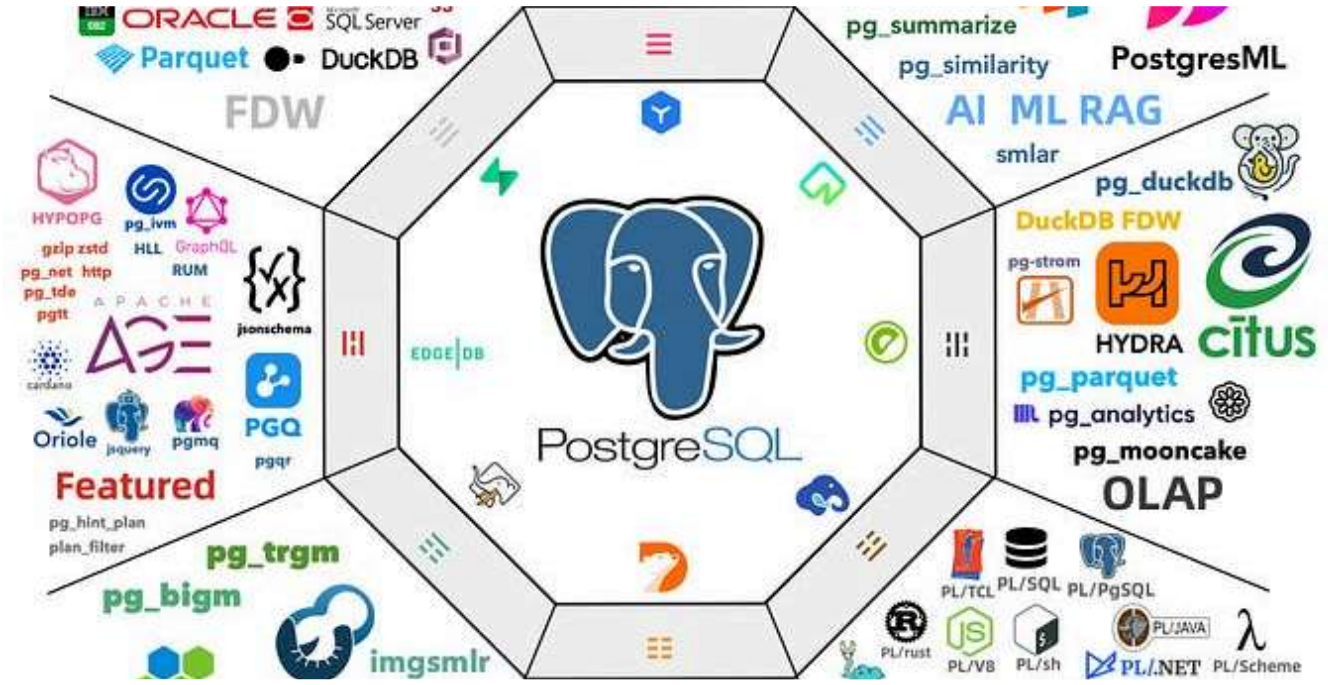
Jan 17 72 1

Oz

Data Masking in PostgreSQL

In today's digital age, securing sensitive data is paramount for businesses to maintain trust and comply with regulations. One effective...

Sep 6, 2024 38



Vasanth Bharathi

Can PostgreSQL Handle All Database Needs?

Credits To: Pigsty

Dec 16, 2024 6

Shailesh Kumar Mishra

How a Simple Query Brought Down Performance: A PostgreSQL Partition Pruning Mystery

The Curious Case of the Missing Pruning

4d ago 4