



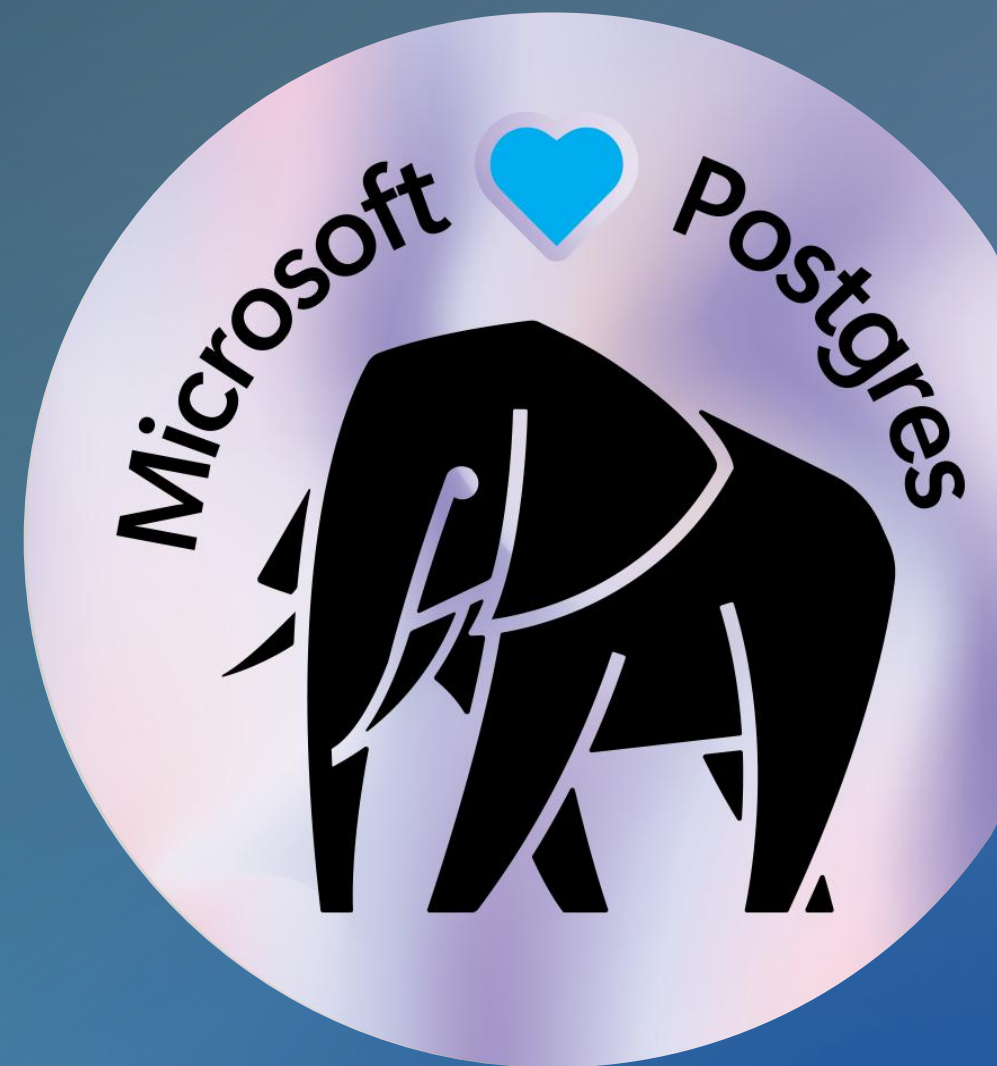
PostgreSQL at Microsoft: Open Source, AI, & Cloud

Claire Giordano

PostgreSQL Significant Contributor @ Microsoft
PGConf.EU 2026



@Microsoft for PostgreSQL



**WELCOME TO
VALENCIA**



SELECT 2026 FROM PGConf.EU;

Happy 30th
Birthday to the
Postgres project



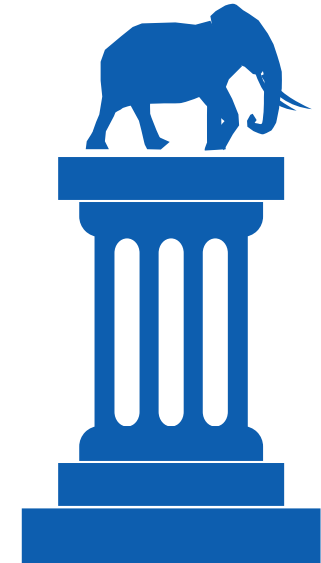
How many of you
have already picked
up your database
socks @ Microsoft
booth?



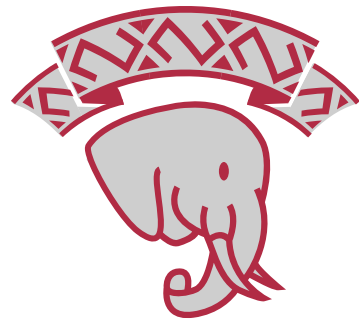
We've been giving away database socks since Milan!



PGConf.EU 2019



PGCONF.EU
2024



PGConf.EU 2025



PGConf.EU 2026

3

things you
might not
know (yet)

01



PostgreSQL project

02



Cloud

03



AI & Developer tools

Microsoft's investment in the Postgres project starts with people



Alexander Kukushkin



Amit Langote



Andres Freund



Ashutosh Bapat



Ashutosh Sharma



Ayush Tiwari



Claire Giordano



Cornelia Biacsics



Daniel Gustafsson



David Rowley



Melanie Plageman



Nazir Bilal Yavuz



Nitin Jadhav



Noah Misch



Palak Chaturvedi



Richard Guo



Satya Narlapuram



Thomas Munro



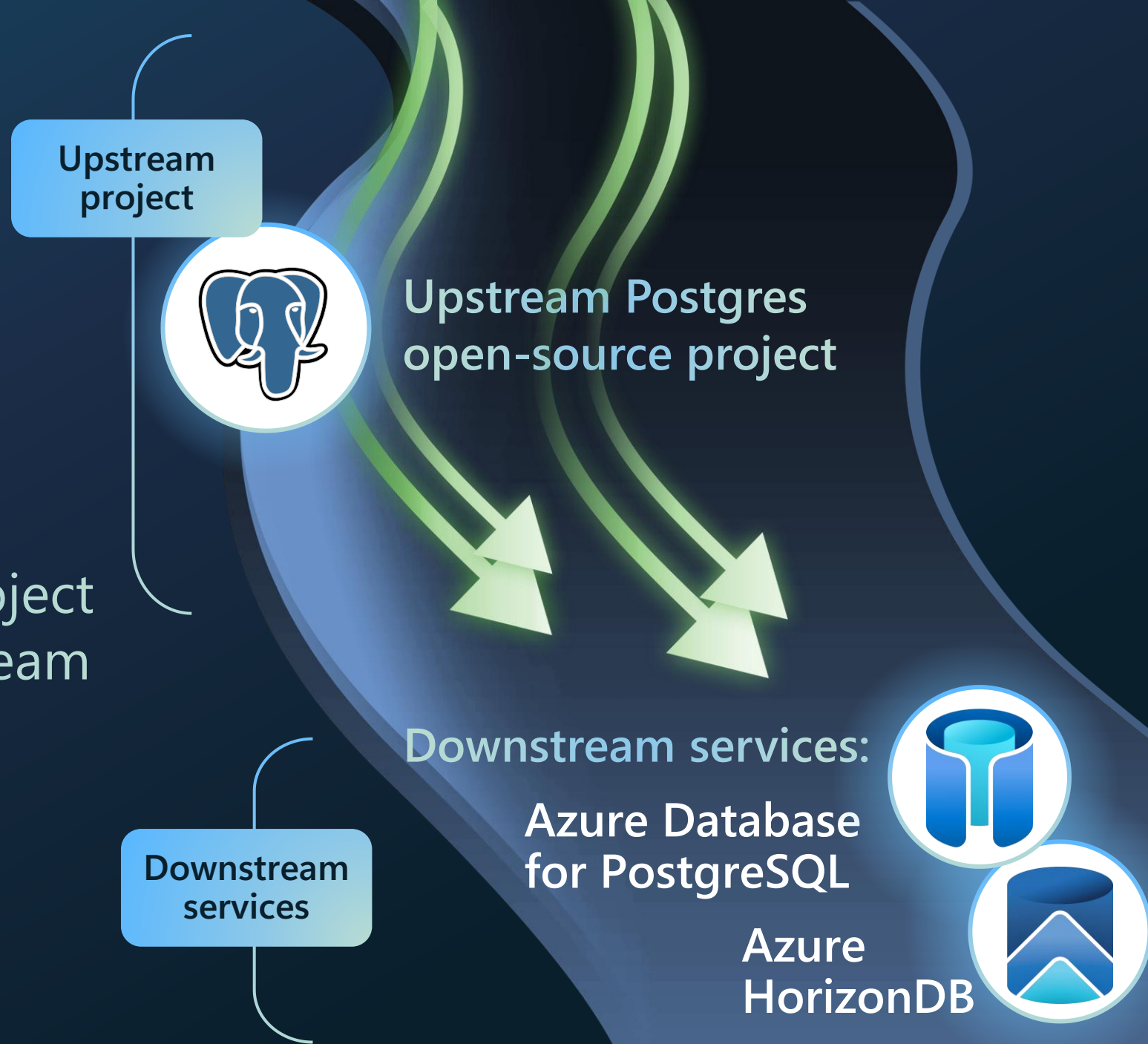
Tomas Vondra



Vinod Sridharan

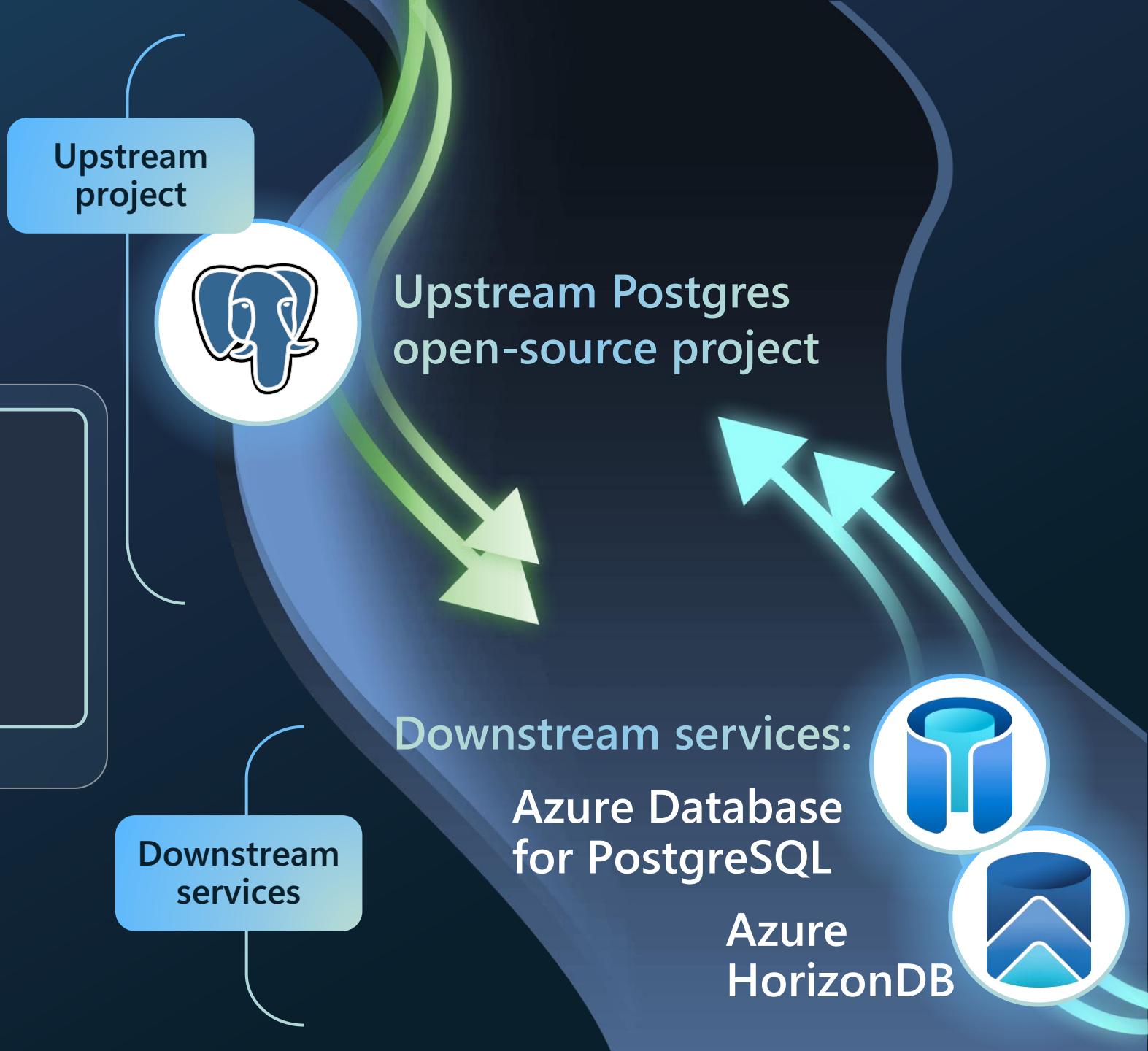
Contributing to the upstream Postgres project is necessary—

Contributions that benefit upstream open-source project ALSO benefit our downstream Azure customers



What might not be obvious—

Learnings from Azure customers also benefit the Postgres project



//

Commits are an imperfect
measure of value, impact,
& accomplishment."

Gratitude for all the amazing Postgres 19 work!

*As of Postgres 19 RC1 on 15 Oct 2026

3,723

Commits to date
in Postgres 19*

4,098

Author commits to
date in Postgres 19*

7,240

Patch reviews to
date in Postgres 19*

Postgres 19

Just some of the features the Microsoft team has authored (or co-authored)



Performance & Efficiency

Faster tuple deformation

Fast path for foreign key checks

Faster COPY using SIMD

Parallel TID range scan

On-access visibility map updates

Reduced vacuum WAL generation

Operability, security, & SQL

Automated resizing of AIO workers

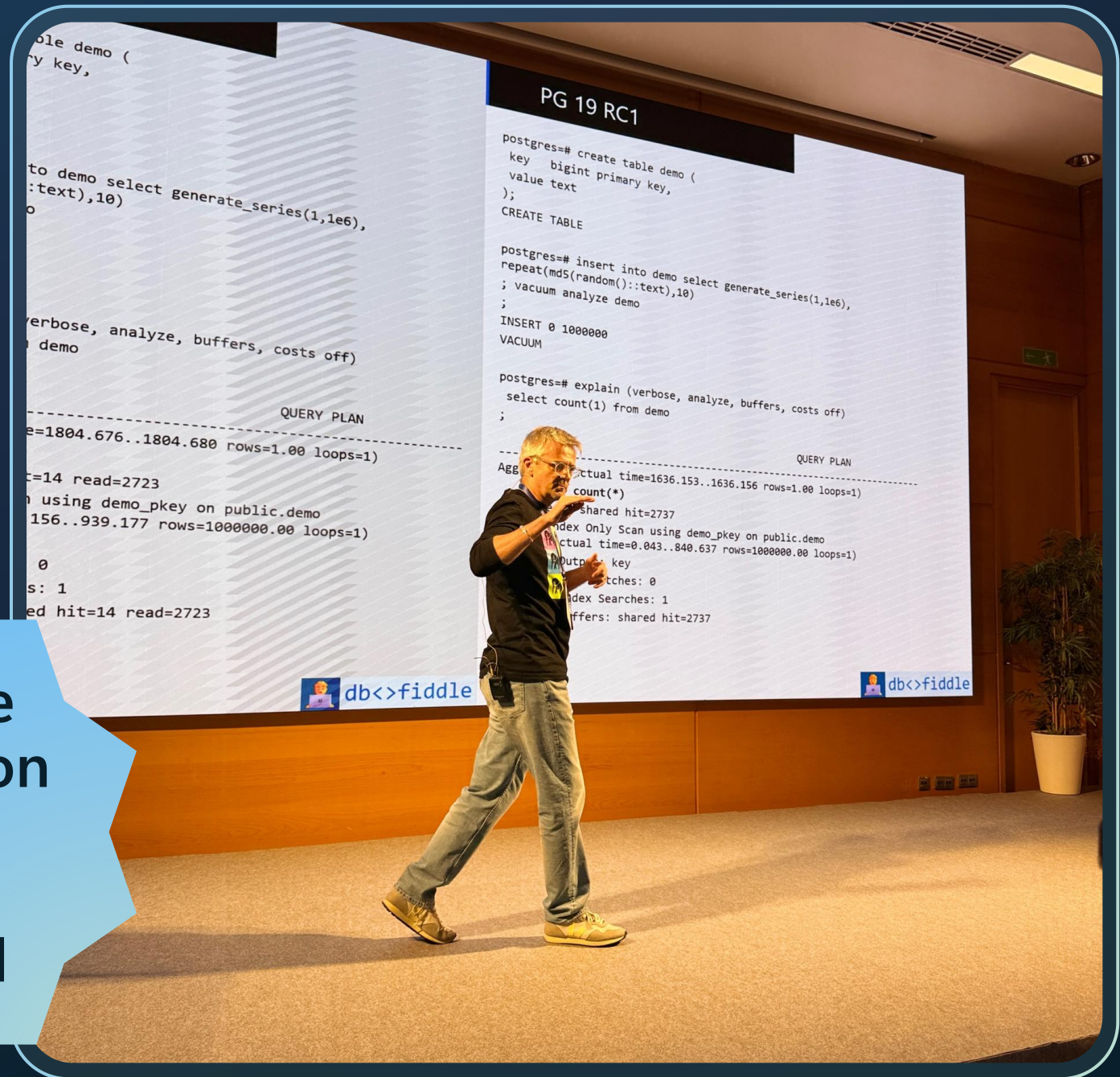
New EXPLAIN (IO) for better I/O visibility

Security CVEs (with whole PG community)

Server-side SNI support

Franck Pachot
of Microsoft
demo'd a few of
these PG19
features @
PGConfEU
yesterday

Watch the
recording on
YouTube,
once
published



Where our Microsoft team is focusing in Postgres 20

Storage, I/O, & concurrency

Index pre-fetching

Shared buffer resizing without restart

Async writes

Multi-threading preparatory work

Write combining: larger writes per I/O op

Planner & executor

Star join improvements

Improved large-join algorithms

Executor batching

Selective deformation

Improved locking with high # partitions

Supporting & Sponsoring Postgres Community Events



Financial Sponsorship—and much more!

Organizing
teams

Program
Committees

Financial
Sponsorship

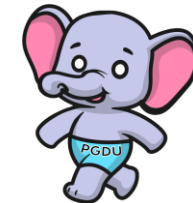
Speakers



PGConf.EU 2026



P2D2



Microsoft is a Benefactor Sponsor of the PostgreSQL project



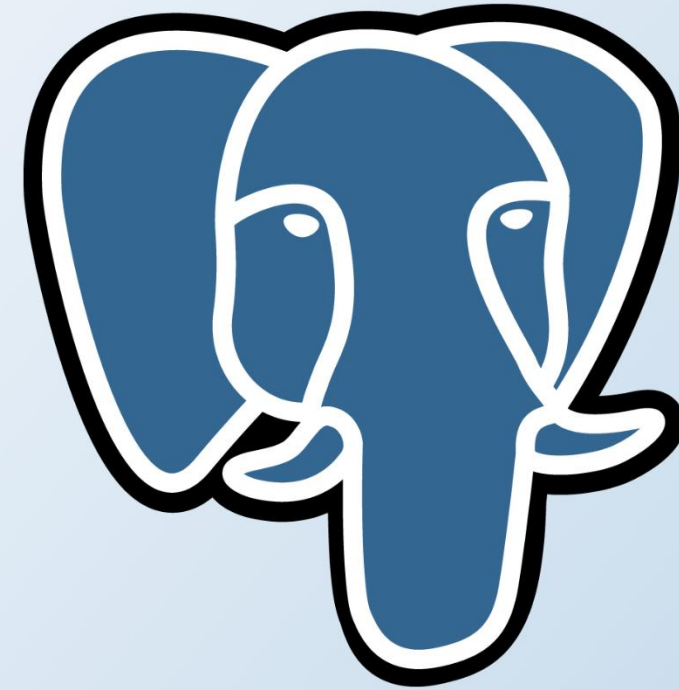
❤️ [Donate to the PGCA](#) ❤️ today to defend the PostgreSQL trademark & brand assets

Protecting the PostgreSQL project's trademarks & brand assets

The PostgreSQL Community Association (PGCA*) is an official non-profit organization chartered by the PostgreSQL Core Team to protect the PostgreSQL brand assets.

*Also known as PostgreSQL Community Association of Canada

[Donate to PGCA →](#)



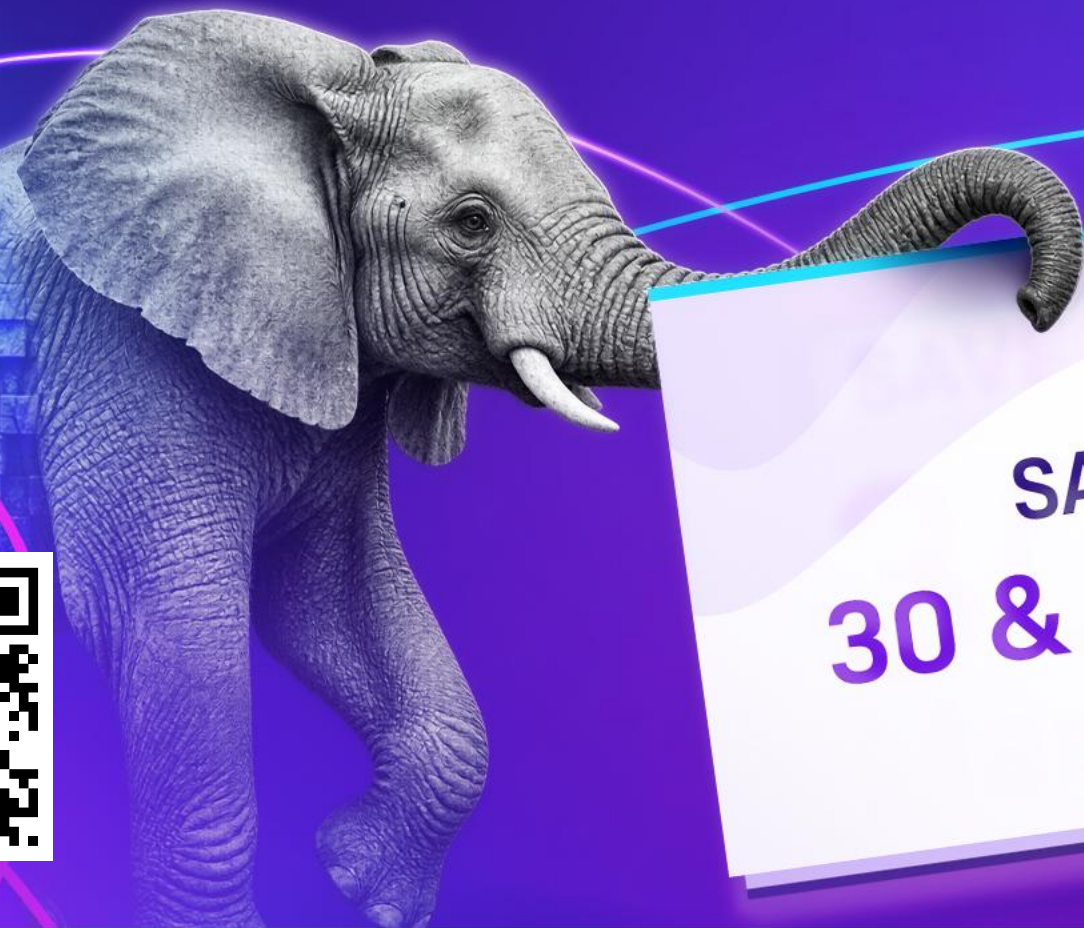
50 amazing guests on TalkingPostgres.com





POSETTE: An Event for Postgres

2027



SAVE THE DATE
30 & 31 March 2027



Brought to you by



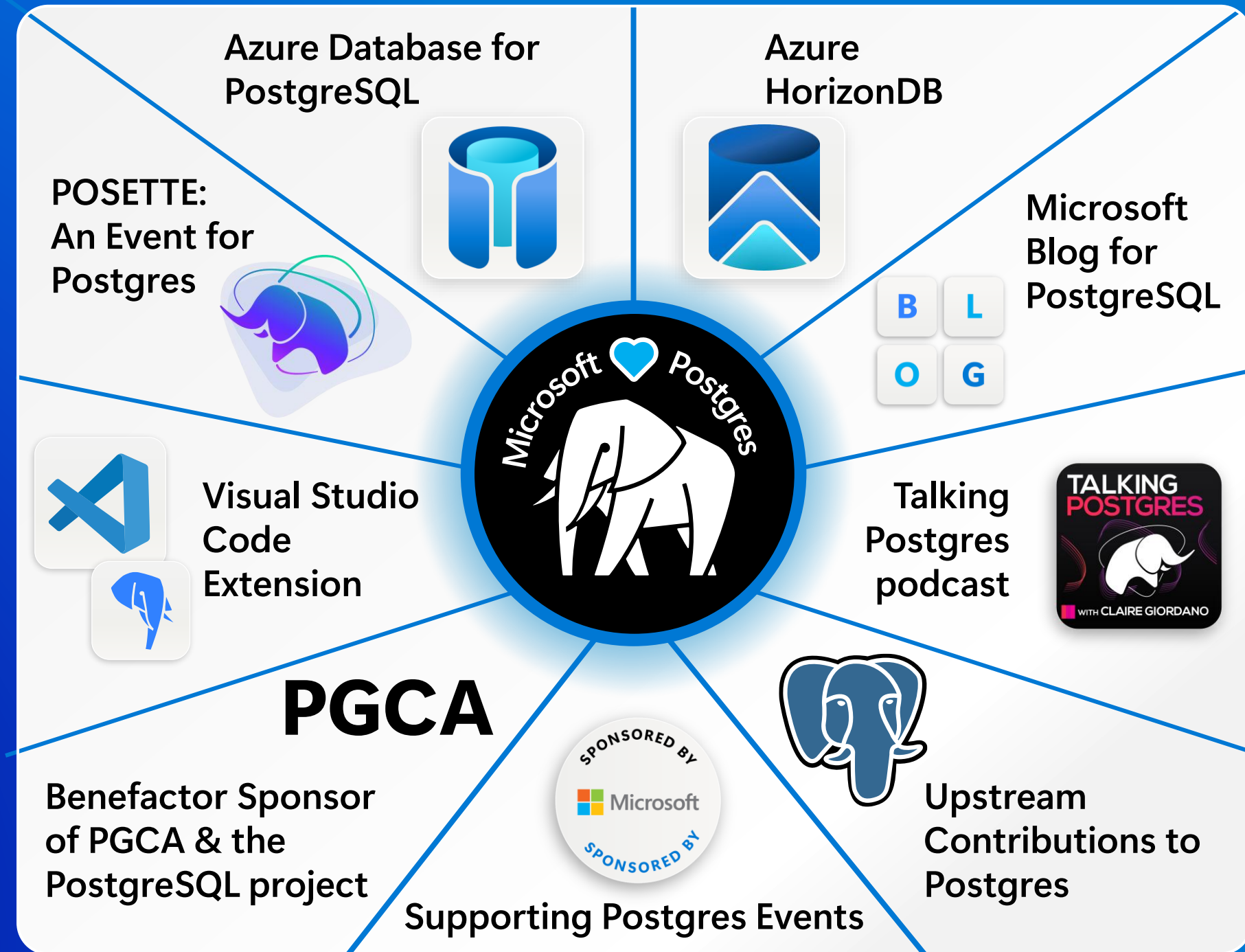
posetteconf.com

The Toothbrush Test



Nine of the ways Microsoft contributes to Postgres

Postgres, PostgreSQL and the Slonik Logo are trademarks or registered trademarks of the PostgreSQL Community Association of Canada, and used with their permission.



Cloud



General Availability

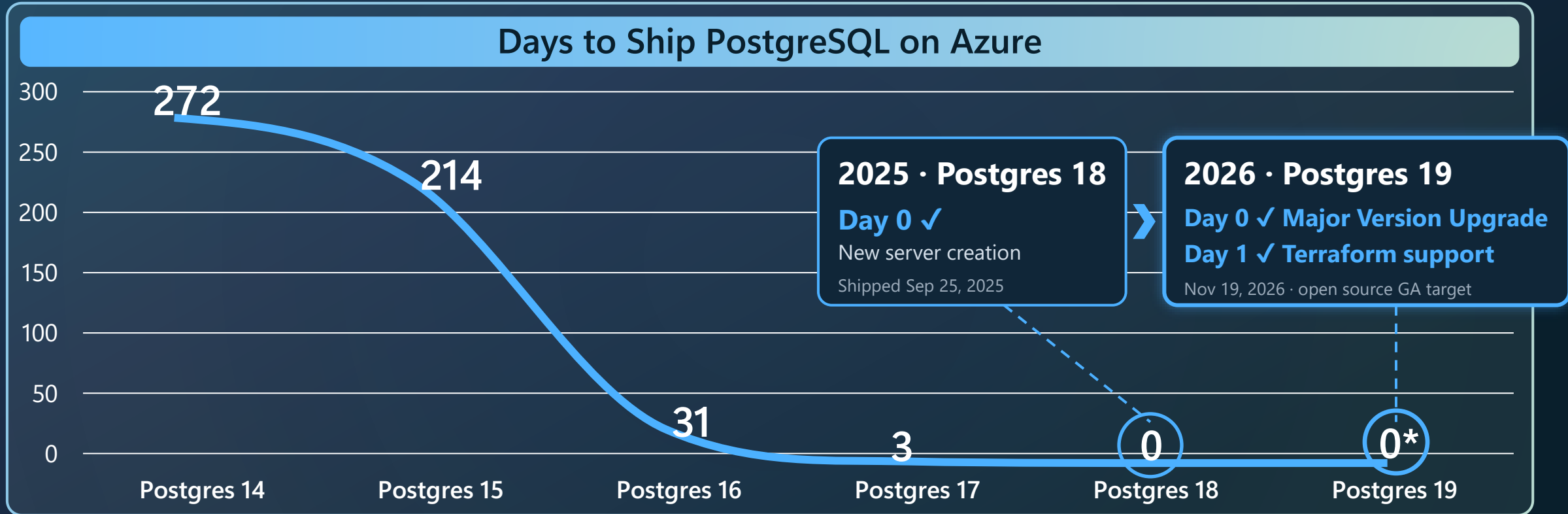


Azure Database for PostgreSQL flexible server

Managed service for PostgreSQL

Day 0 PostgreSQL Availability in Azure

No Cloud Lag. With PG19, No Upgrade Lag.



Driving Azure Database for PostgreSQL forward



The collage displays a series of blog posts from the 'Tech Community' section, dated from January to September 2026. The posts cover a range of topics including:

- January:** REST API, Terraform, Ansible, and Achievements.
- February:** Features, Terraform, and Dashboards.
- March:** SSDV, Premium efficiency, and Cascading.
- April:** POSE, Features, and Entra-.
- June:** POSE, Features, and Chaos.
- July 2026:** PgBouncer, Documentation, and Maintenance.
- August:** Adaptive, Pre-up, and support.
- September 2026 Recap:** Azure Database for PostgreSQL. This post includes a 'Table of Contents' with links to:
 - Adaptive
 - Pre-upgrade
 - New PgBouncer
 - August 2026
 - Elastic Cloud
 - New and
 - Azure Adv
 - Terraform
 - Entra-ID t
 - Extended

The September 2026 recap post also features a section titled 'Cross-Tenant Customer Managed Key (CMK) now Generally Available' and another titled 'Database Hub in Microsoft Fabric: A Single View of Your PostgreSQL Estate (Preview)'. At the bottom right of the collage, there is a screenshot of the 'Databases Overview' dashboard, which shows various metrics and charts for the database estate.

What's new with Postgres at Microsoft, 2026 edition



aka.ms/blog-pg-at-microsoft



Public Preview



Azure HorizonDB

Built for Business. Engineered for Builders.

Scale to 3,072 vCores
and 128TB storage

Up to 3x faster transactions
and vector search

<1 millisecond multi-zone
commit latency

Azure HorizonDB Architecture

Cloud Native PostgreSQL

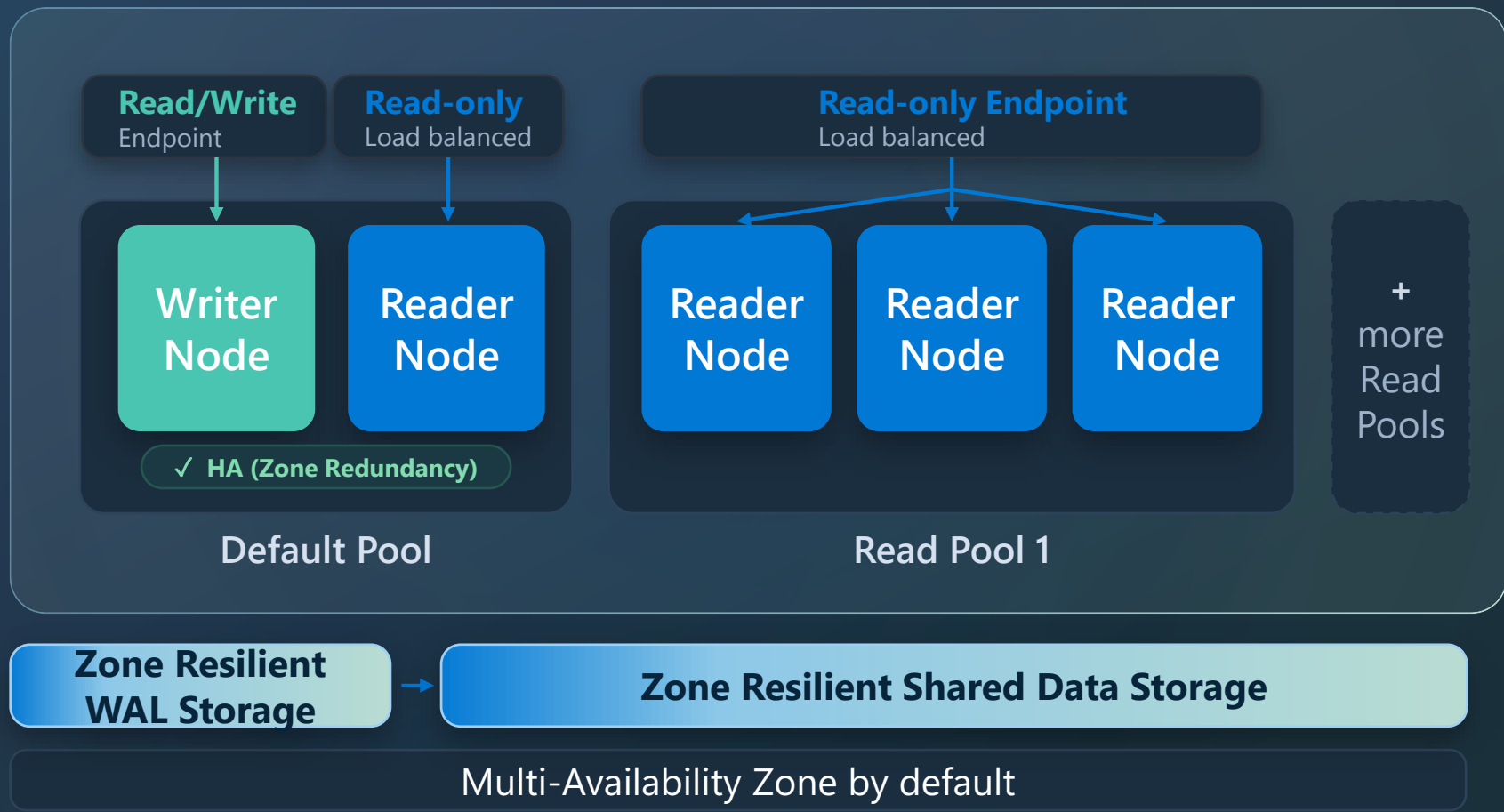
Max Node Size **192 vCores**

Commit Latency **< 1 ms**

Faster Failovers **~ 5 sec**

Pay for storage used **Up to 128 TB**

Read Nodes **Up to 15**



CMUDB talk by Adam Prout about Azure HorizonDB



aka.ms/talk-HorizonDB-cmudb

Gifts in the open-source world



Have you heard about?

pg_durable



A screenshot of the GitHub repository page for 'pg_durable'. The page shows the repository name, public status, and statistics: 4 watchers, 80 forks, and 2.8k stars. The 'main' branch is selected. A commit by Thom Chiovoloni (thomcc-work) is shown, dated 5 days ago. The commit message is 'Add failure-isolat...'. The commit includes a list of files: .agents/skills/pg-..., .cargo, .devcontainer, .github, benchmarks, and docker-entrypoin... Each file has a description and a date. On the right, the 'About' section describes the project as 'PostgreSQL in-database durable execution' and provides a link to 'microsoft.github.io/pg_dur...'. Below the link are tags: ai-pipelines, ai-workflows, durable-execution, durable-functions, durable-workflows, and postgresql. At the bottom, there are links for 'Readme' and 'License'.

Stay tuned for more
Azure HorizonDB news
at Microsoft Ignite!



AI & Developer tools



// I think we can all agree
that AI has changed
things in our day to day."

Where AI meets PostgreSQL on Azure

AI helps **you** work on Postgres

While you write code

VS Code + GitHub Copilot

PostgreSQL MCP Server

PostgreSQL Skills

While you run the database

Performance Troubleshooting

Query Optimization

Database Agent

Postgres does AI work for **your app**

Search by meaning

DiskANN Vector Search

Full-text + Hybrid Search

Semantic Re-ranking

Graph

Call models from SQL

AI Model Management

AI Functions

AI Pipelines

AI Pipelines

Your AI steps, written in SQL
Run inside the database
Survive crashes, retry failures
In Azure HorizonDB

Source: Microsoft Learn: Implement AI pipelines in Azure HorizonDB

Step	Purpose
<code>ai.chunk()</code>	Split a long text column into overlapping chunks.
<code>ai.embed()</code>	Generate vector embeddings for a column.
<code>ai.extract()</code>	Extract structured fields from text using an LLM.
<code>ai.generate()</code>	Generate text from a prompt template using an LLM.
<code>ai.rank()</code>	Score documents against a required <code>query</code> .

Database Agent

From problem
to proposed fix

Active 1 Resolved 0 Closed 0

⌵ ⚠ Repeated customer-history lookups on `large_transactions` likely need a composite index

PENDING 26d ago

issue-20260821091954-8395941c · Created Aug 21, 02:20 AM PDT · Updated Sep 9, 01:28 AM PDT

DETECTION ✨

Summary:

Reads for a customer's newest transactions are probably doing more work than necessary, which can slow those requests down even though the server is healthy overall. The database is not overloaded right now, but this access pattern is a good candidate for an index improvement.

> Technical details

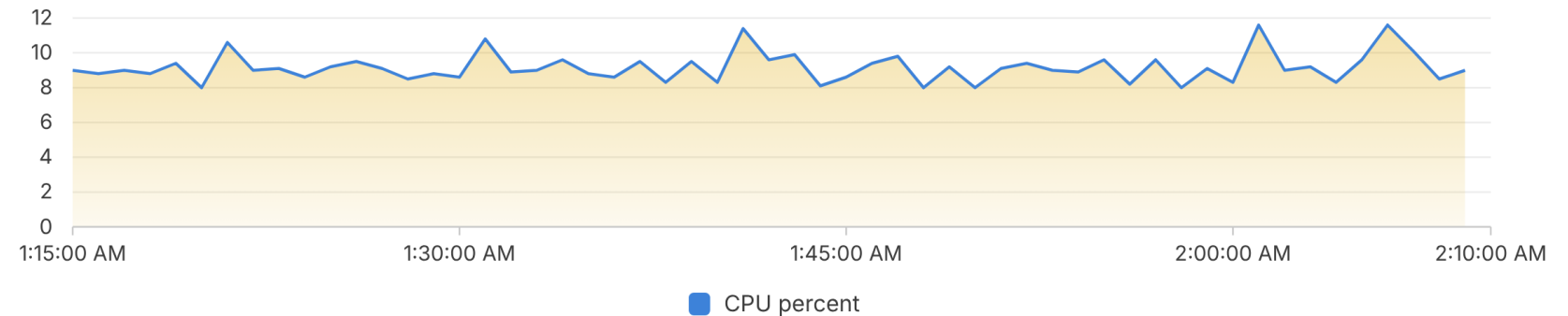
TIMELINE ✨

DETECTED Aug 21, 02:20 AM PDT

Detected during investigation: Repeated customer-history lookups on `large_transactions` likely need a composite index

METRIC ANALYSIS ✨

CPU percent



● Issue detected 02:17 AM PDT DETECTED

RECOMMENDED ACTION ✨

Why this index

The evidence points to a common lookup pattern on `dbagent_scn_929c9ea9.large_transactions`: fetch a single customer's newest rows.

VS Code extension works with *any* Postgres database



aka.ms/pgsql-vscode



PostgreSQL

Microsoft

microsoft.com 

↓ 771K

Develop, query, and manage PostgreSQL (Postgres) databases with IntelliSense,...



FREE

VS Code extension for PostgreSQL

Schema Visualization

PostgreSQL

Connections

Servers

cf-flex17

Databases

azure_maintenance

azure_sys

postgres

Schemas

System

cron

public

Tables

customers

inventory

product

region

transactions

Views

Materialized Views

Functions

Procedures

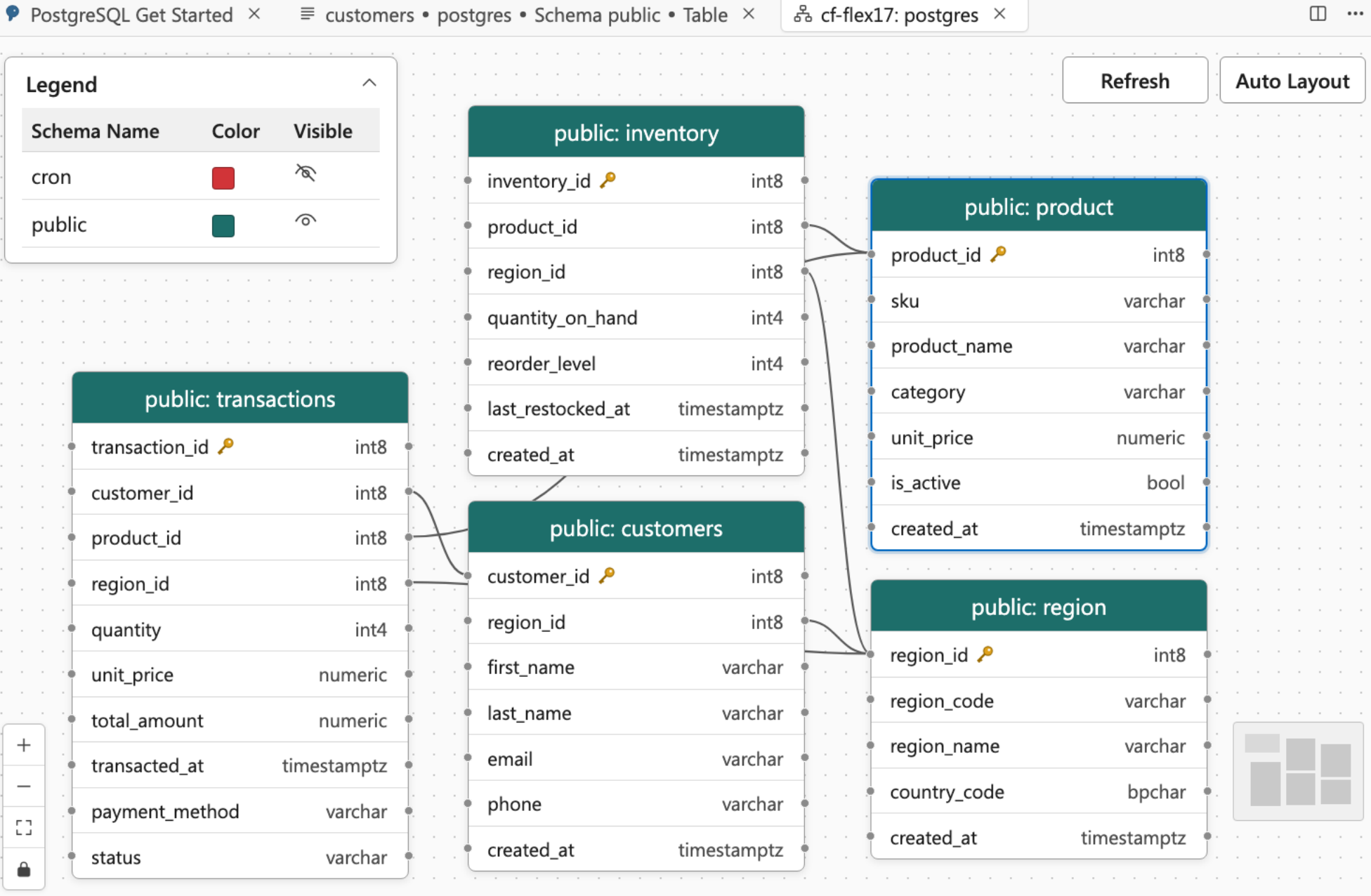
Trigger Functions

Collations

Query History

Azure Deployments

Migrations



+

-

VS Code extension for PostgreSQL

Execution Plans

PostgreSQL

Connections

- Servers
 - cf-flex17
 - Databases
 - azure_maintenance
 - azure_sys
 - postgres
 - Schemas
 - System
 - cron
 - public
 - Tables
 - customers
 - inventory
 - product
 - region
 - transactions
 - Views
 - Materialized Views
 - Functions
 - Procedures
 - Trigger Functions
 - Collations

Query History

Azure Deployments

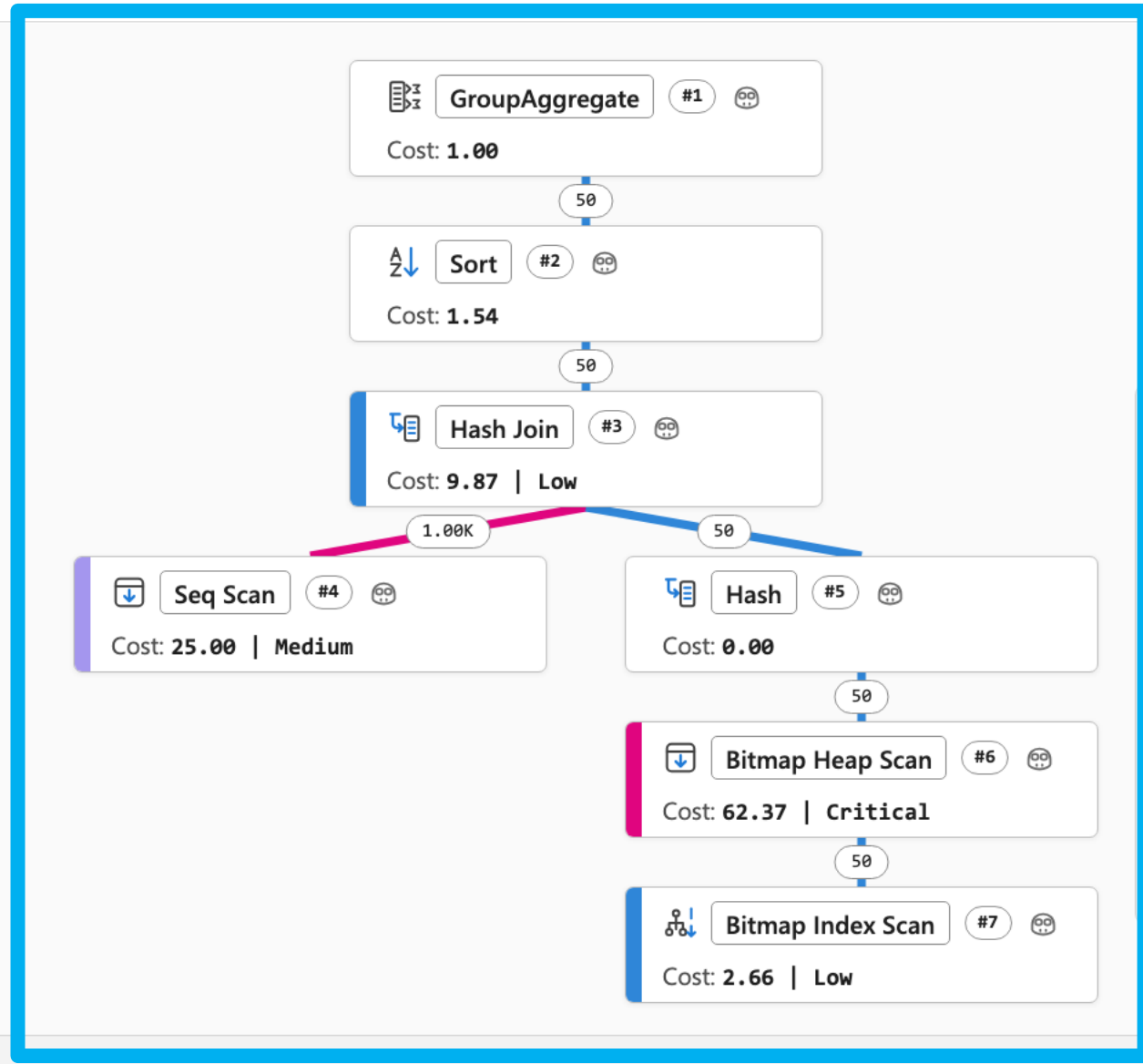
Migrations

SELECT Untitled-1

Query Plan: SELECT c.customer_id, c.first_...

Tree View | Icicle View | Table View | Source View

Analyze with Copilot | Copy Plan



Graph Layout

Color By: Cost

Color Scope: Self Total

Node Color: Low Critical

Edge Color: Estimated Rows Low Critical

Options

VS Code extension for PostgreSQL

AI-Assisted Debugging

PostgreSQL

Connections

Servers

cf-flex17

Databases

azure_maintenance

azure_sys

postgres

Schemas

System

cron

public

Tables

customers

inventory

product

region

transactions

Views

Materialized Views

Functions

Procedures

Trigger Functions

Collations

Query History

Azure Deployments

Migrations

-- Dashboard dataset: customers ranked bUntitled-1 • Query n...rd dataset: customer... ×

Tree ViewIcicle ViewTable ViewSource ViewAnalyzeCopy Plan

Search

Cost	Rows	Node ID	Node Operation
0.39	17 (est)	#1	Sort
0.34	17 (est)	#2	GroupAggregate
0.39	17 (est)	#3	Sort
5.38	17 (est)	#4	Hash Join
25.00	1.0K (est)	#5	Seq Scan on customers as c
0.00	17 (est)	#6	Hash
26.83	17 (est)	#7	Bitmap Heap Scan on transactions as t
0.26	17 (est)	#8	BitmapAnd
2.66	50 (est)	#9	Bitmap Index Scan using idx_transacti
35.29	3.3K (est)	#10	Bitmap Index Scan using idx_transacti

Cost

Rows

Options

Chat

PostgreSQL query plan analysis

Analyze PostgreSQL Query Plan

I need help analyzing the following PostgreSQL query execution plan for performance

2. Refresh planner statistics

To address the **[Medium] row-estimate error**, run during an appropriate maintenance window:

```
ANALYZE public.customers;
ANALYZE public.transactionor
```

Then regenerate `EXPLAIN (ANALYZ`

`BUFFERS)`.

Connect to database an...

-- Dashboard dataset: customers

Describe what to build

Agent

LocalDefault permissions



Introducing pghacker-mcp



github.com/Azure-Samples/pghacker-mcp

README Code of conduct Contributing MIT license Security

pghacker-mcp

Claude Code Codex Copilot CLI Copilot app OpenCode VS Code Cursor

pghacker-mcp answers questions about PostgreSQL's design history and source code, with citations. Ask why a feature works as it does, or what happened to SQL/PgQ, and an agent finds the pgsql-hackers threads, commits and code that answer it, and quotes them.

The project is powered by PostgreSQL itself. The mail, the graph and PostgreSQL's full commit history, live in [Azure HorizonDB](#) (PostgreSQL when run locally), and PostgreSQL indexes do the searching: trigram for code and text, vector for semantic search, BM25 for full-text search, and B-trees over the history's commit ranges, with Apache AGE for the graph. Typical searches across the whole commit history return in milliseconds.

It is a [PilotSwarm](#) 0.8.2 app in three parts:

- **pghacker-mcp**: a public, read-only [MCP](#) (Model Context Protocol) server, in Rust, that serves the mail and the code to agents.
- **pghacker-indexer**: an internal Rust service that retrieves the mail and PostgreSQL's commit history, indexes them and projects them into a graph. Its MCP tools serve the crawler agents.
- **Crawler agents**: PilotSwarm sessions that read the threads and record the features, decisions and authors' positions in them, with verbatim quotes.

pghacker-mcp.dev



• Public MCP server • no sign-in

PostgreSQL's design history and code, **with citations**, for your AI agent

pghacker-mcp is a public MCP server for AI agents. It indexes the pgsql-hackers mailing list and PostgreSQL's source with its full commit history, so your agent finds the evidence in seconds instead of searching the archive blindly, and answers with citations you can check.



MCP server

• Available

✓ Streamable HTTP ✓ No sign-in

1

Copy the endpoint

`https://pghacker-mcp.dev/mcp`

Copy URL

2

Add it to your agent

Claude Code

Codex

Copilot CLI

Copilot app

OpenCode

VS Code

Cursor

Other

3

Ask, and check the citations

"Use pghacker to find out what happened to SQL/PgQ in PostgreSQL 19."



Powered by **HorizonDB**



Thank you, &
Come say hello
at the Microsoft
booth!

