What years of DBA experience taught me while building an LLM-based Postgres Agent

Gülçin Yıldırım Jelínek

Diva: Dive into AI, Istanbul

xata

Agenda

1. From Classic DBA to AI Agent
2. Designing the Xata Agent
3. How the Agent Helps
4. AI Trade-offs in Practice
5. Challenges & What's Next

Select * from me;

Current:

- Staff Engineer at Xata
- Postgres Contributor
- Co-founder of Prague PostgreSQL Meetup
- Co-founder & General Coordinator of Kadın Yazılımcı
- Co-founder & Chair of Diva Conference

Past:

- Board Member at PostgreSQL Europe
- Staff Engineer at EDB, 2ndQuadrant

DIVA 2025 & 2024
SILVER SPONSOR
Visit our booth!

Postgres at scale

xata.io

My journey with Postgres

- Started as a DBA in 2012
- Manual performance tuning, upgrades, backups
- Supported dev teams with query design and troubleshooting
- A lot has changed, but core instincts remain valuable

DBA responsibilities


Installation & Configuration


Maintenance & Optimization


Monitoring & Observability


Upgrades & Migrations

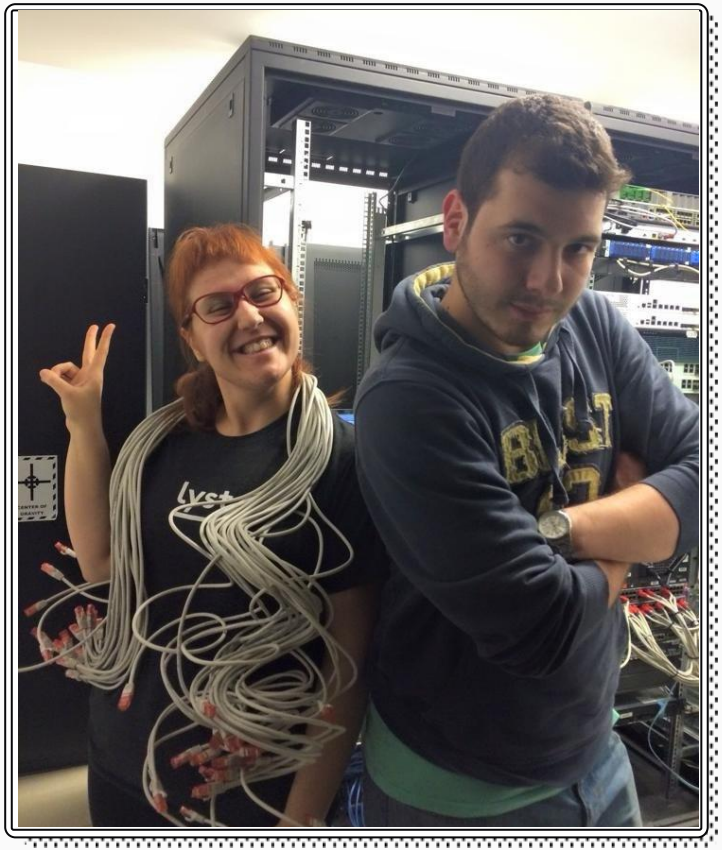

Backup & Recovery

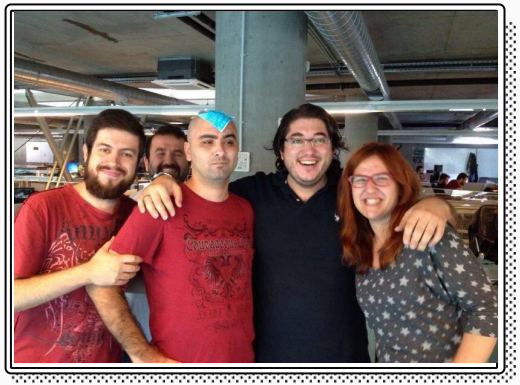

Lock management & Schema migrations


Performance & Query tuning


Automation & Tooling

What does a PostgreSQL DBA do? (Turkish, 2015)





A lost trade: DBA

What does a PostgreSQL DBA do? (Turkish, 2015)

Paradigms change, it is nothing new

“ Paradigm shifts occur when the dominant paradigm under which normal science operates is rendered incompatible with new phenomena. ”

Thomas S. Kuhn

The Structure of Scientific Revolutions
1962

DevOps, Site Reliability & Platform Engineering

IT automation & orchestration tools emerged (e.g Ansible, Chef, Puppet, Saltstack)

Provisioning & Configuration management

Application deployment & Continuous delivery

Security & Compliance

Ansible Loves PostgreSQL (English, 2016)

Ansible's Playbook concept

New patterns: Configuration as Data (YAML files), Infrastructure as Code, Infrastructure as Data

A Playbook contains Plays. Plays contain Tasks. Tasks call Modules.

In Ansible, Playbooks organize Tasks. Roles organize Playbooks.

Suddenly you could manage lots of resources

Ansible Loves PostgreSQL (English, 2016)

Ansible's Playbook concept

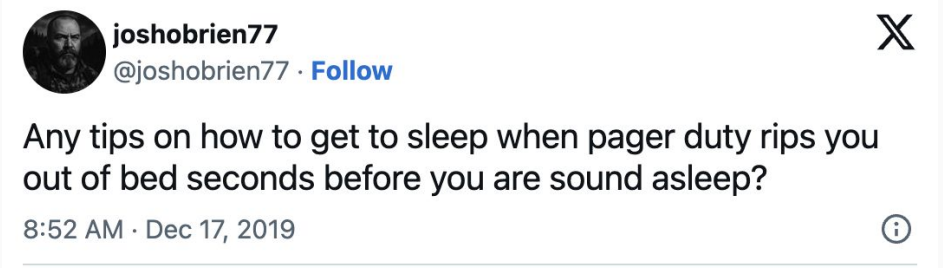
```
- block:
  - name: Install pg_hba.conf
    template:
      src: "{{ hba }}"
      dest: "{{ postgres_conf_dir }}/pg_hba.conf"
      owner: "{{ postgres_user }}"
      group: "{{ postgres_group }}"
      mode: "0644"
    when: _postgres_create_hba_file
    notify:
      - Note Postgres reload required

  - name: Set a fact for contents of pg_hba.conf
    set_fact:
      postgres_pg_hba_config: "{{
          lookup('ansible.builtin.template', hba)
          |split('\n')
          |reject('match', '#')
          |list
        }}"
  when:
    hba != ''
  vars:
    tmpl: pg_hba.conf.j2
    hba: "{{ postgres_hba_template|default(tmpl) }}"
```

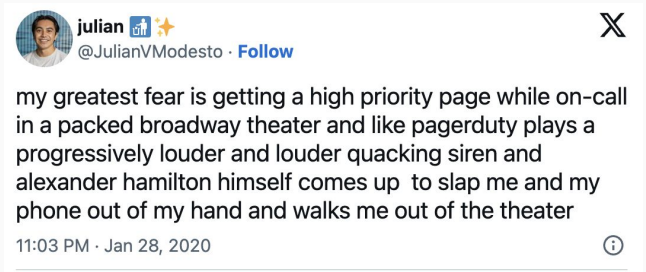
Example from open source TPA project (past contributor)

The old way of ensuring uptime

- **Collect telemetry:** metrics, logs, traces (observability pillars)
- **Set alerts:** thresholds + ML-based anomaly detection
- **24/7 on-call rotations** for incident response
- **Dashboards** to correlate and explore data quickly
- **Use playbooks**, chaos testing, incident tools to support on-call engineers
- **Define and track SLOs/SLAs** to align alerts with business goals

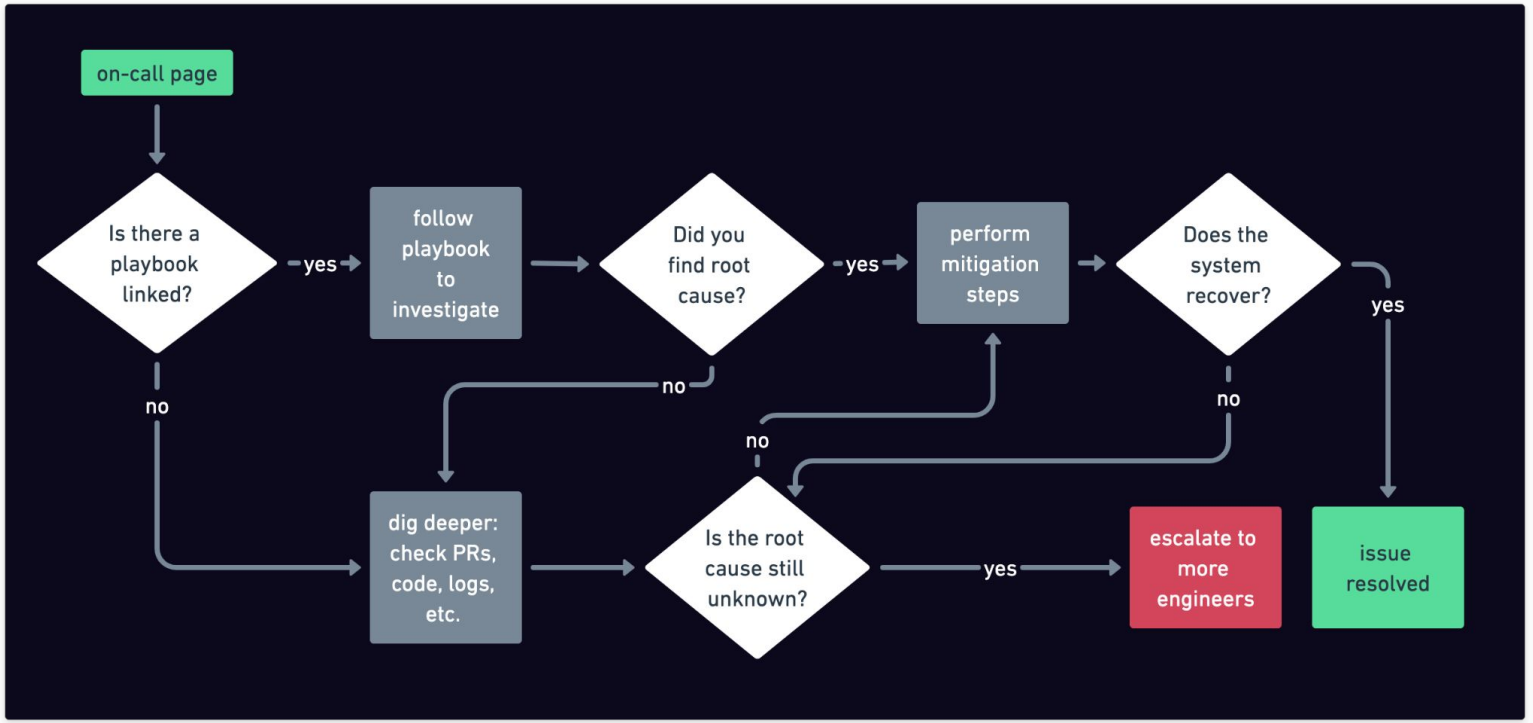
Let’s talk about on-call





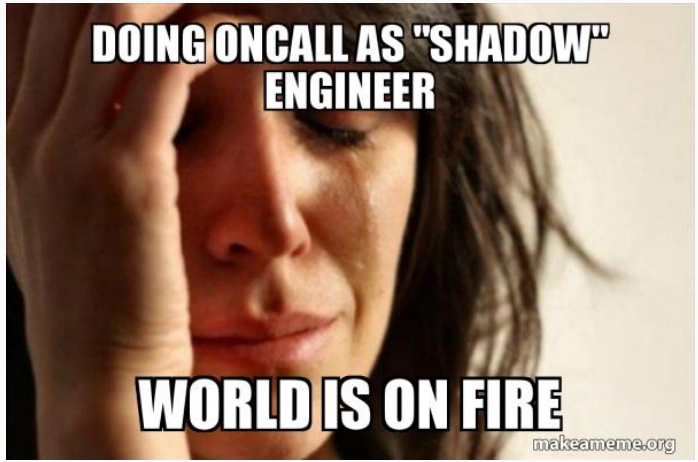
xata

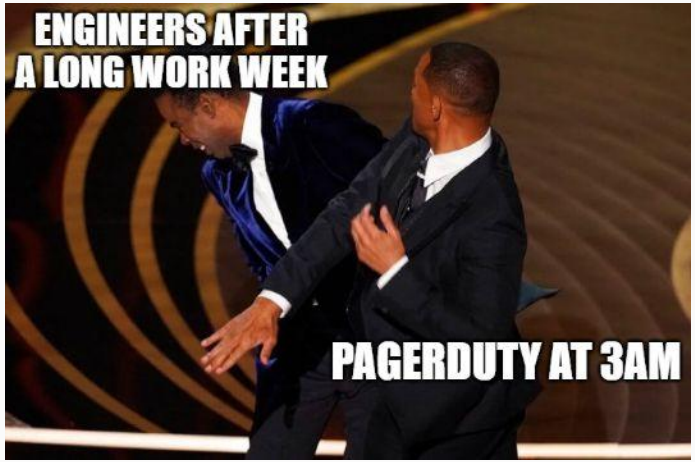
Typical on-call experience



Are AI agents the future of observability?

How an AI agent can help





Picture this:

An agent that's **always on-call**. No sleep. No burnout.
No missed alerts. It watches your systems 24/7, reacts instantly and only wakes a human when it really matters.

How an AI agent can help

- **24/7 on-call**, escalating to humans only when needed
- **Investigates root cause** using:
  - Pre-written playbooks
  - Past incidents and internal knowledge base
  - GitHub code, recent commits/PRs
  - External sources (status pages, docs, web)
- **Self-generates and executes playbooks** if none exist
- **Recognizes self-healing cases** and monitors recovery
- **Executes low-risk fixes** (e.g., scaling) autonomously
- **Escalates risky or uncertain cases** with a full summary
- **Writes and stores incident reports** for future reference

The big idea

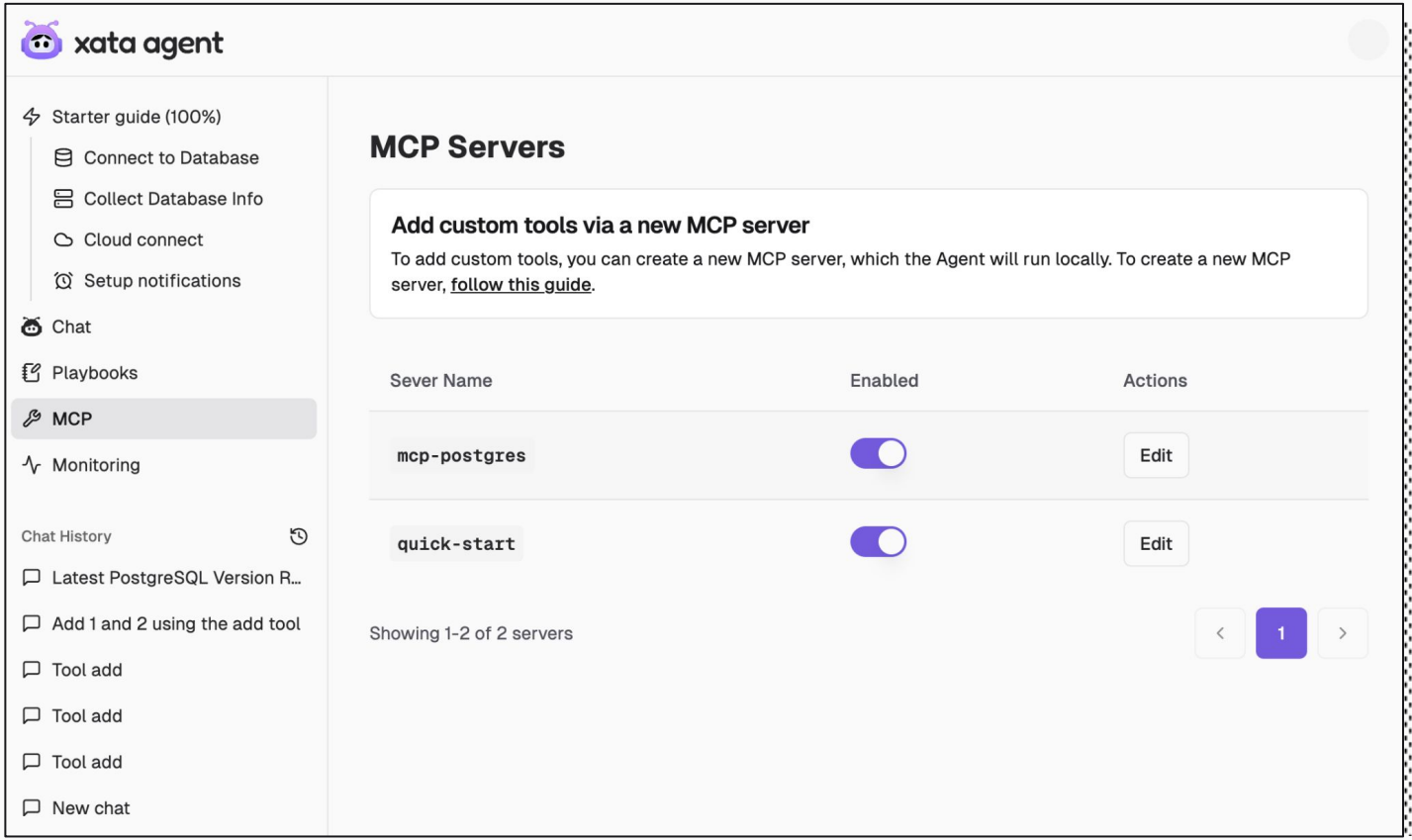
What if we could turn years of DBA/SRE instincts into a smart, LLM-powered Postgres agent?

From DBA to DB Agent

Monitor, tune and maintain databases

Scale DBA expertise across large fleets

Designing the Xata Agent



xata

Designing the Xata Agent

Internal tool for managing many Postgres clusters

Built on lessons from classic DBA work

Combines:

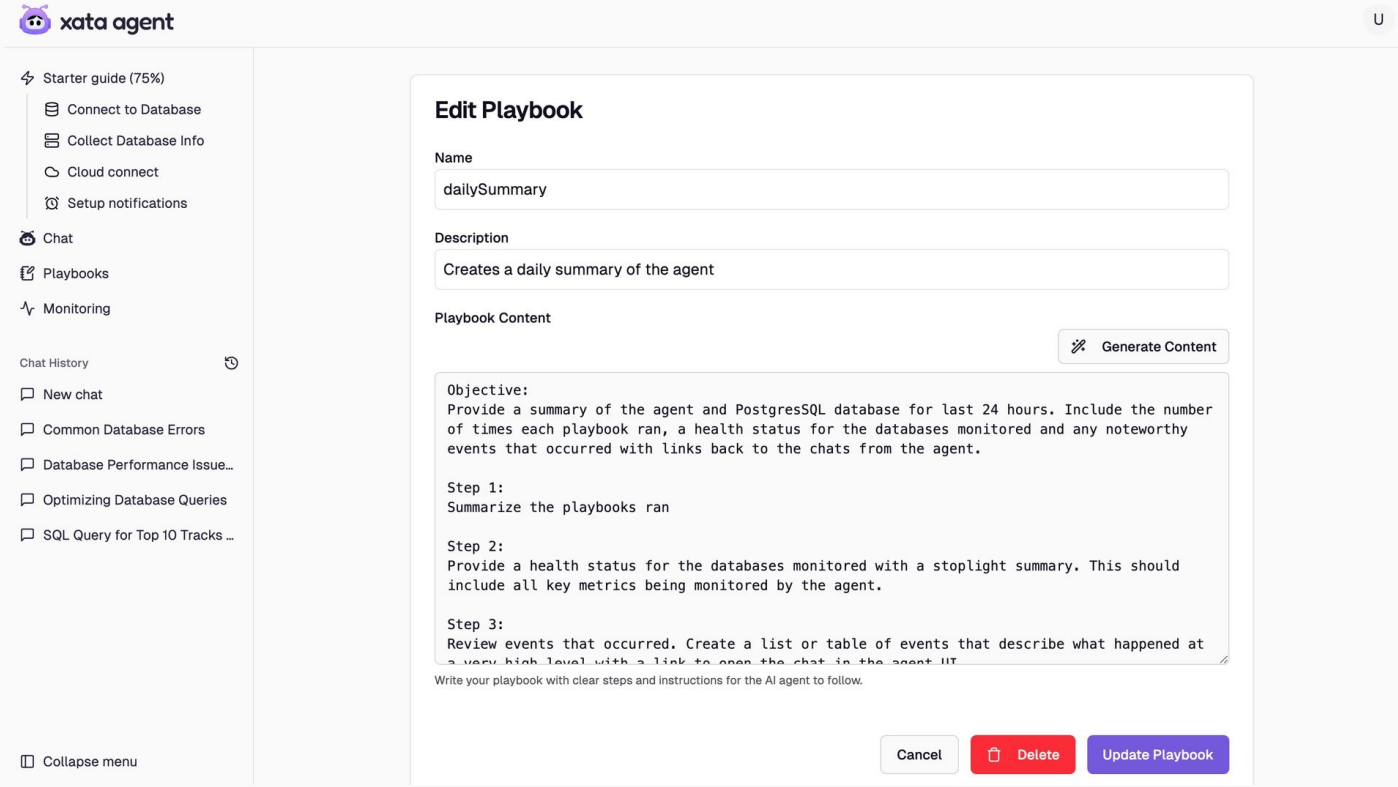
- SQL analysis
- Logs and metrics
- Prompted reasoning from LLMs

Agent concepts: Playbooks

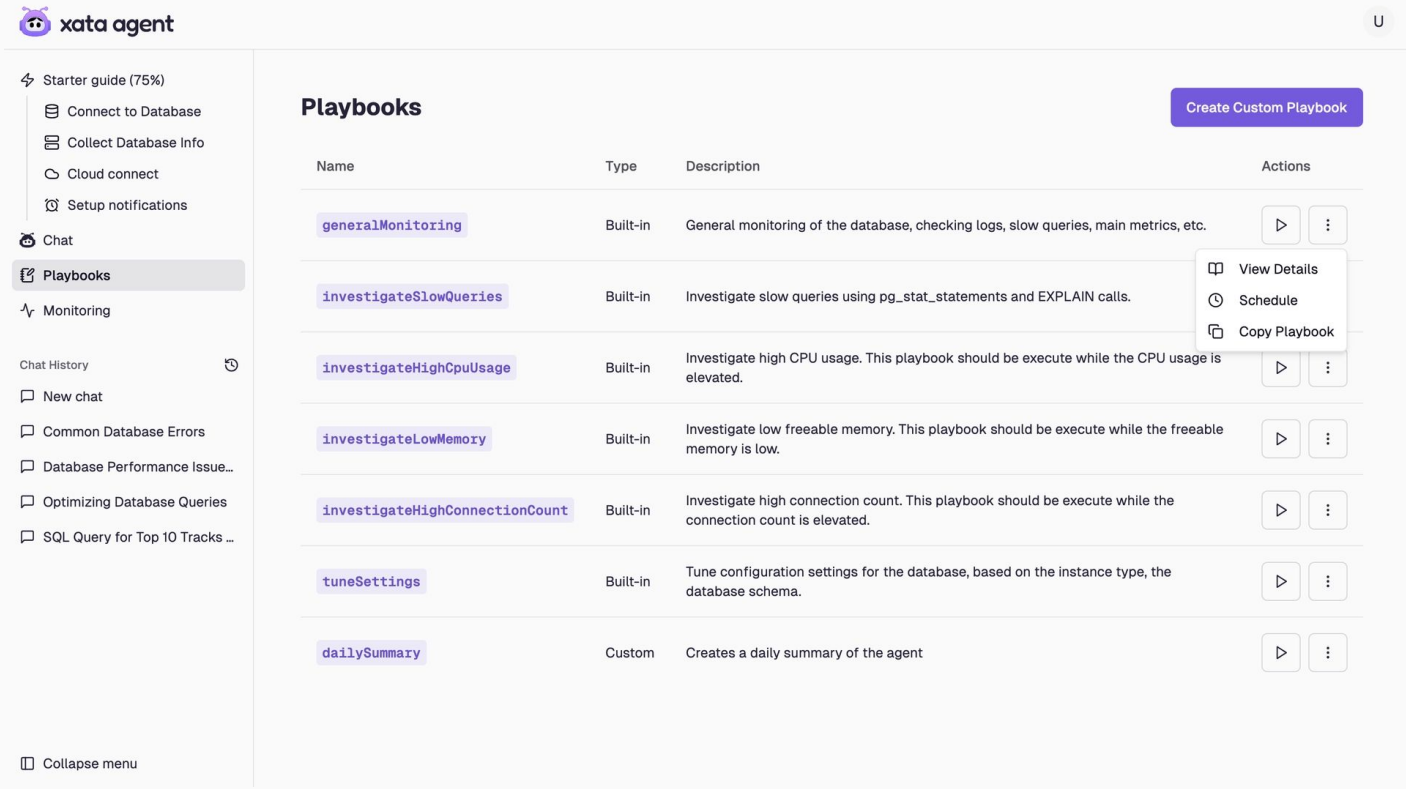
Imagine a DBA/SRE's playbook, then remember the Ansible playbooks. We inherited the same logic.

Playbooks are a sequence of steps that the Agent can follow to detect, diagnose, and fix issues in the Postgres database. Playbooks are written in English and you can easily create your own, or modify the pre-built ones.

Agent concepts: Playbooks



Agent concepts: Playbooks



xata

Agent concepts: Schedules

Traditional DBA/SysAdmin/Platform roles relied on cron expressions to schedule maintenance and administration jobs and agent has the same ability.

Schedules are used to run playbooks at specified intervals. This way the Agent can detect issues 24/7, find the root cause, and fix them before they impact the users. Schedules are defined as cron expressions, or you can let the Agent decide the best time to run the playbook.

xata

Agent concepts: Tools

Traditional DBA/SysAdmin/Platform engineers relied on certain tools to monitor, backup, configure, upgrade the systems they managed. Agent provides a large library of pre-built tools for Postgres.

Tools are functions that can be called by the Agent to get information about the Postgres database, the instance/clusters on which it is running, and metrics and logs. Custom tools can be provided by integrations and by MCP servers.

xata

What the Agent does

Finds root causes: Pinpoint slow queries, deadlocks and performance bottlenecks and regressions

Gets actionable fixes: AI-powered suggestions from query optimization to suggesting indexes to infrastructure upgrades

Ensures uptime: Proactive monitoring to keep your database healthy. It monitors backups, upgrades, disk and more metrics.

Clarke's three laws

“ Any sufficiently advanced technology is indistinguishable from magic.”

Arthur C. Clarke

Profiles of the Future
1973 revision

Future of Xata Agent

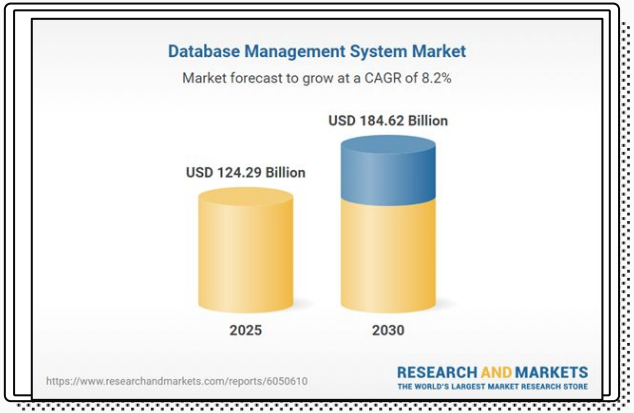
- Building workflows for a **self-optimizing** database.
- Approval workflows through Github to review changes recommended by the Agent
- More extensible and customizable observability workflows inside the Xata Agent via MCP servers

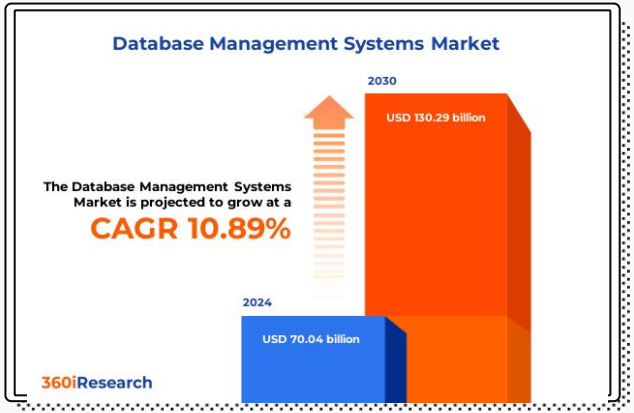
xata

Database agents are having a moment

- Neon: More than 80% of Neon databases are auto-created by AI agents.They also provide an MCP server.
- Supabase: Offers built-in Supabase AI Assistant v2 inside the dashboard with schema design, SQL generation, error debugging, RLS policies. Supports vector embeddings.
- ClickHouse: ClickHouse.ai for analytics agents, AgentHouse for conversational data interaction, internal Dwaine LLM for company-wide insights.

Why now? The growing market





The DBMS market is projected to grow at a **CAGR of 8.24%** from **US$124.291 billion** in 2025 to **US$184.623 billion** by the end of 2030.

The DBMS market size was estimated at **USD 70.04 billion** in 2024 and expected to reach **USD 77.49 billion** in 2025, at a **CAGR 10.89%** to reach **USD 130.29 billion** by 2030.

xata

Observations on the landscape

Most agents focus on development or analytics

DBA-style tools are still rare

Postgres is at the center of many AI-native workflows

The dev landscape is wild with fast iteration

Challenges & Next steps

Privacy: Agent may include metadata (e.g., table names, logs) in LLM prompts sent to external APIs. Self-hosted LLMs are a path forward for sensitive environments.

Cost: AI models aren't free—but so far but cheaper than traditional observability tools, even with premium LLMs.

Safety: Risk of hallucinations or destructive actions.

- Agent is sandboxed: no arbitrary SQL execution without human approval.
- Future support for riskier actions will require gated workflows.

Testing: I'll go into more detail on this next.

Evals & trade-offs

Testing an AI agent is not like testing normal code

We need to evaluate if the agent still works when prompts change or LLMs switch.

xata

What's an Eval?

An Eval is like an integration/system test tailored to LLM behavior.

LLMs are black boxes

Prompt/model changes can drastically change outputs

We need to trace the reasoning and outputs step-by-step

xata

Where AI shines

Explaining slow queries

Suggesting high-level next steps

Supporting devs with SQL guidance

xata

Where AI struggles

Stateful understanding like multi-step schema changes

Predictable cost control

Consistent behaviour

“ Anything that is in the world when you’re born is normal and ordinary and is just a natural part of the way the world works. Anything invented after you’re thirty-five is against the natural order of things ”

Douglas Adams

The Salmon of Doubt: Hitchhiking the Galaxy One Last Time, 2002

xata

The shift to autonomous agents

Can you trust an agent with your most critical infrastructure?

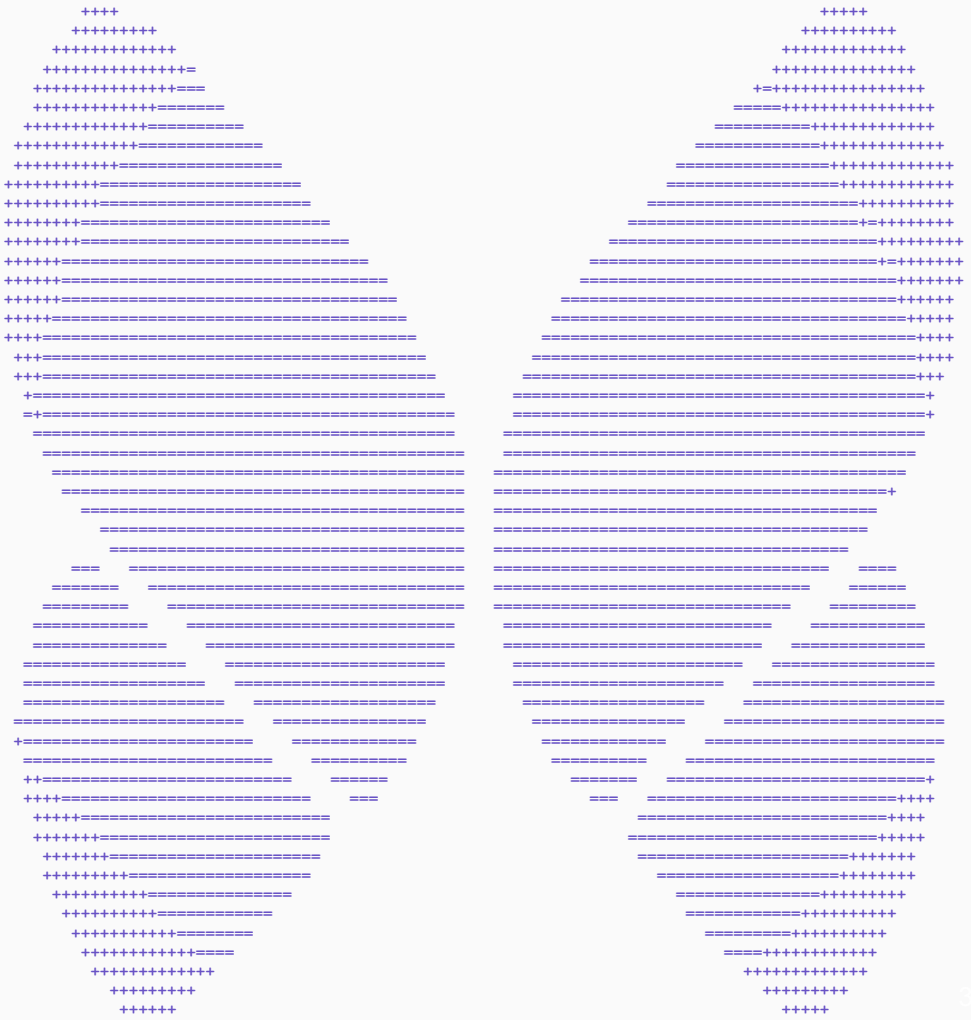
Just 2 years ago: “No way!”

Now? More and more teams are saying yes.

xata

Postgres at scale

Thank you!



gulcin@xata.io

xata.io