



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

Gülçin Yıldırım Jelínek

Diva: Dive into AI, Istanbul

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



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's Playbook concept



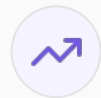
New patterns: Configuration as Data (YAML files),
Infrastructure as Code, Infrastructure as Data



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'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



joshobrien77
@joshobrien77 · Follow

Any tips on how to get to sleep when pager duty rips you out of bed seconds before you are sound asleep?

8:52 AM · Dec 17, 2019



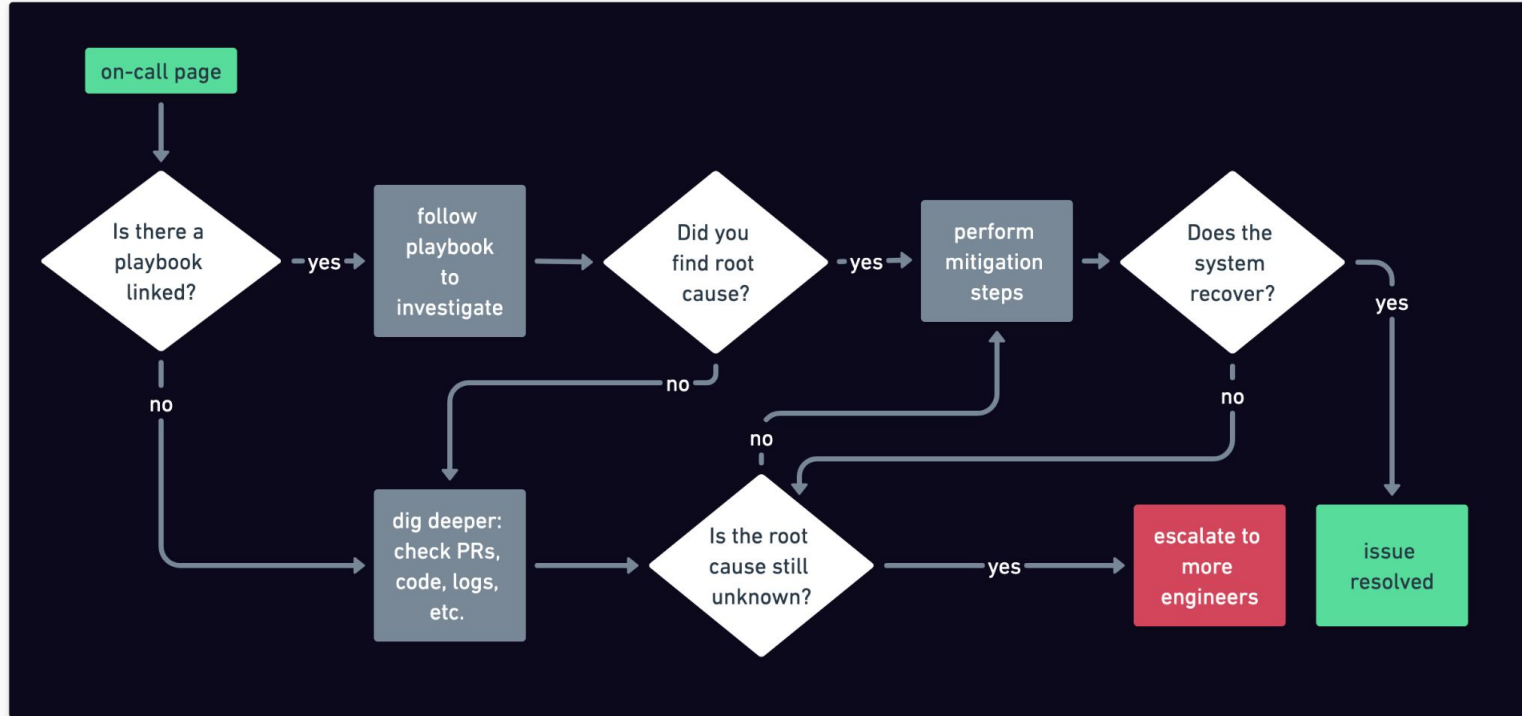
julian
@JulianVModesto · Follow

my greatest fear is getting a high priority page while on-call in a packed broadway theater and like pagerduty plays a progressively louder and louder quacking siren and alexander hamilton himself comes up to slap me and my phone out of my hand and walks me out of the theater

11:03 PM · Jan 28, 2020

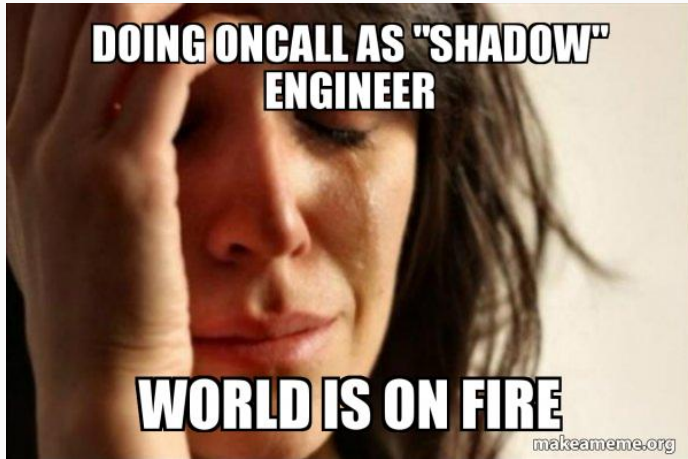


Typical on-call experience



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.

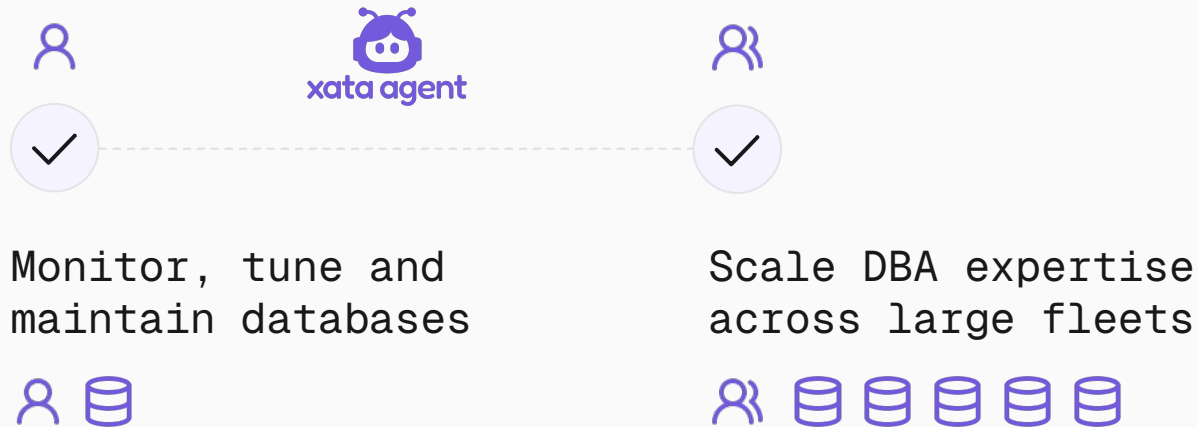
How an AI agent can help

- **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



Designing the Xata Agent


xata agent

⚡ Starter guide (100%)

📄 Connect to Database

📄 Collect Database Info

☁️ Cloud connect

🔔 Setup notifications

💬 Chat

📄 Playbooks

🔗 MCP

📈 Monitoring

Chat History

📄 Latest PostgreSQL Version R...

📄 Add 1 and 2 using the add tool

📄 Tool add

📄 Tool add

📄 Tool add

📄 New chat

🔗 MCP Servers

➕ Add custom tools via a new MCP server

To add custom tools, you can create a new MCP server, which the Agent will run locally. To create a new MCP server, [follow this guide](#).

Sever Name	Enabled	Actions
mcp-postgres	<input checked="" type="checkbox"/>	Edit
quick-start	<input checked="" type="checkbox"/>	Edit

Showing 1-2 of 2 servers

<

1

>

19

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



U

⚡ Starter guide (75%)

- 📖 Connect to Database
- 📖 Collect Database Info
- ☁️ Cloud connect
- 🔔 Setup notifications

💬 Chat

📝 Playbooks

📊 Monitoring

Chat History



📄 New chat

📄 Common Database Errors

📄 Database Performance Issue...

📄 Optimizing Database Queries

📄 SQL Query for Top 10 Tracks ...

☐ Collapse menu

Edit Playbook

Name

dailySummary

Description

Creates a daily summary of the agent

Playbook Content

✎ Generate Content

Objective:

Provide a summary of the agent and PostgreSQL database for last 24 hours. Include the number of times each playbook ran, a health status for the databases monitored and any noteworthy events that occurred with links back to the chats from the agent.

Step 1:

Summarize the playbooks ran

Step 2:

Provide a health status for the databases monitored with a stoplight summary. This should include all key metrics being monitored by the agent.

Step 3:

Review events that occurred. Create a list or table of events that describe what happened at a very high level with a link to open the chat in the agent UI.

Write your playbook with clear steps and instructions for the AI agent to follow.

Cancel

🗑 Delete

Update Playbook

Agent concepts: Playbooks

U

⚡ Starter guide (75%)

- 📄 Connect to Database
- 📄 Collect Database Info
- ☁️ Cloud connect
- 🔔 Setup notifications

💬 Chat

📖 Playbooks

📈 Monitoring

Chat History 🕒

- 📄 New chat
- 📄 Common Database Errors
- 📄 Database Performance Issue...
- 📄 Optimizing Database Queries
- 📄 SQL Query for Top 10 Tracks ...

🔒 Collapse menu

Playbooks

Create Custom Playbook

Name	Type	Description	Actions
<code>generalMonitoring</code>	Built-in	General monitoring of the database, checking logs, slow queries, main metrics, etc.	▶️ ⋮
<code>investigateSlowQueries</code>	Built-in	Investigate slow queries using pg_stat_statements and EXPLAIN calls.	<div> 📄 View Details ⌚ Schedule 📄 Copy Playbook </div>
<code>investigateHighCpuUsage</code>	Built-in	Investigate high CPU usage. This playbook should be execute while the CPU usage is elevated.	▶️ ⋮
<code>investigateLowMemory</code>	Built-in	Investigate low freeable memory. This playbook should be execute while the freeable memory is low.	▶️ ⋮
<code>investigateHighConnectionCount</code>	Built-in	Investigate high connection count. This playbook should be execute while the connection count is elevated.	▶️ ⋮
<code>tuneSettings</code>	Built-in	Tune configuration settings for the database, based on the instance type, the database schema.	▶️ ⋮
<code>dailySummary</code>	Custom	Creates a daily summary of the agent	▶️ ⋮

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.

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.

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

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.

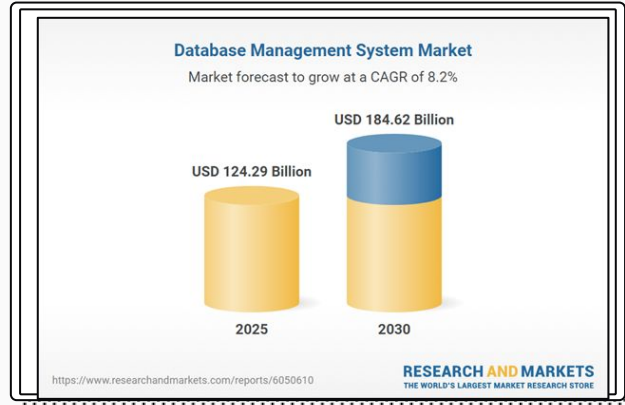


Supabase: Offers built-in Supabase AI Assistant v2 inside the dashboard with schema design, SQL generation, error debugging, RLS policies. Supports vector embeddings.

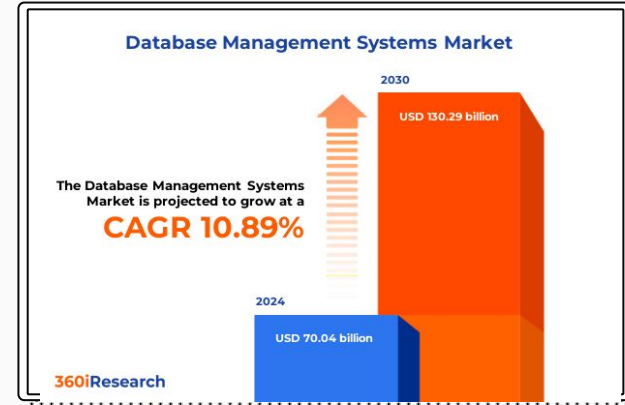


ClickHouse: ClickHouse.ai for analytics agents, AgentHouse for conversational data interaction, internal Dwaine LLM for company-wide insights.

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.

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.

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

Where AI shines



Explaining slow queries



Suggesting high-level next steps



Supporting devs with SQL guidance

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

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.



Postgres at scale

Thank you!

 gulcin@xata.io

 xata.io

