
PERCONA
DBA Roles in the Age of DBaaS: Evolution, Not Extinction
How Managed Databases Are Reshaping the Industry
Jericho Rivera - Percona University 2026, Bangkok, Thailand

The Landscape Has Shifted, But the Data Remains

Key Insight: The industry has moved decisively from on-premises hardware to Database-as-a-Service (DBaaS)

The Central Question: In a world of managed services, the C-suite is asking "Do we still need DBAs?"

Cloud vendors market 'self-driving' databases. While this promise is attractive, it often leads to a dangerous misconception: that 'Cloud' equals 'No Ops'.

The 'Self-Driving' Misconception

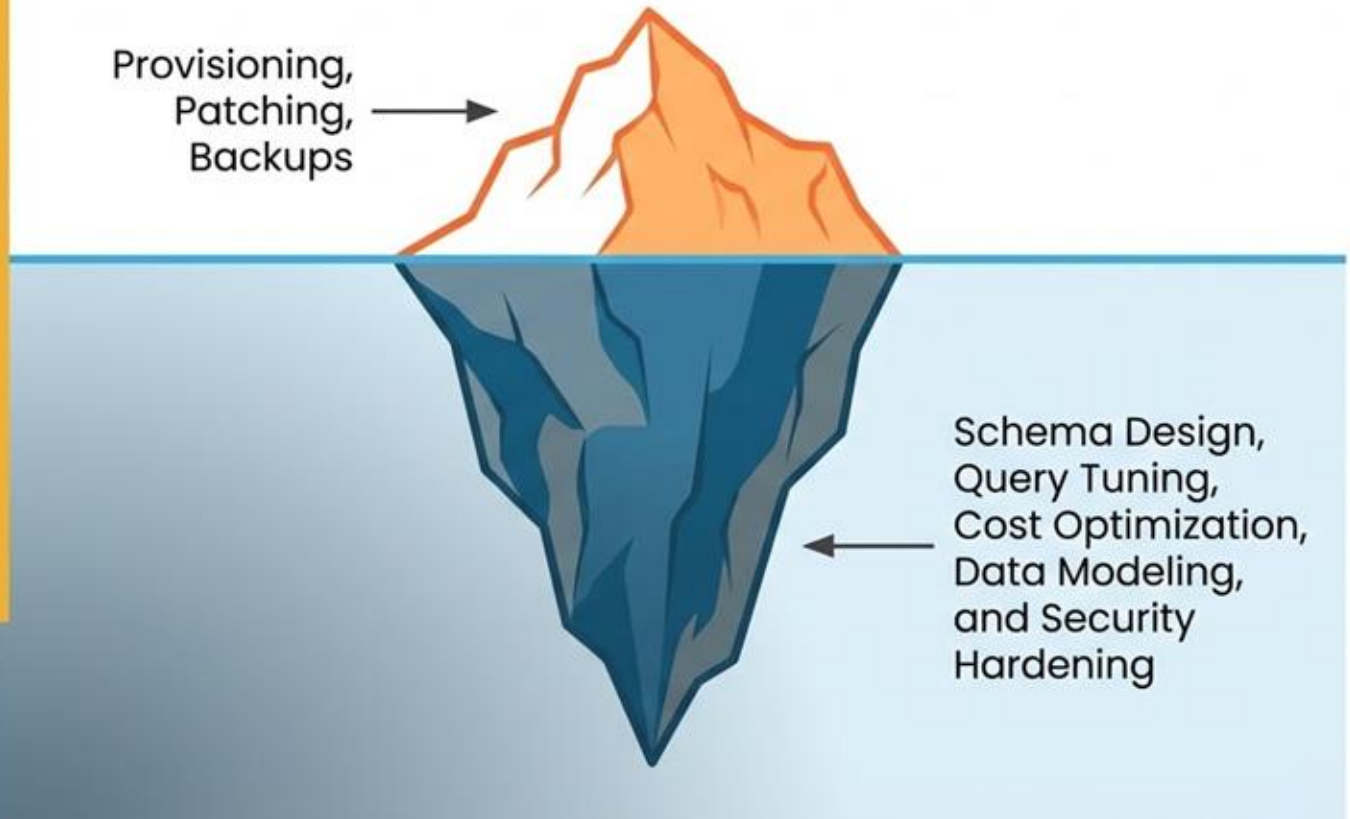
The Myth

- DBAs are no longer needed.
- DBaaS handles everything automatically.
- Developers can manage databases now.

The Reality Gap

Vendors promise seamless automation, leading organizations to believe that **database complexity** disappears in the cloud. This belief fuels the decision to **reduce specialized staff**, often prematurely.

DBaaS Reduces Toil, Not Responsibility.



Key Takeaway

Database complexity doesn't disappear when you move to the cloud - it **shifts**.

The cloud environment introduces new challenges that require **more sophisticated oversight**, not less.

The Automation Audit: Where the Vendor Stops and You Begin

What DBaaS Actually Automates (Commoditized Tasks)

- **Provisioning** and instance setup
- **Backups** and snapshots
- Basic **failover**
- Minor version **patching**
- Storage **autoscaling**
- **Monitoring** dashboards

What DBaaS Does NOT Automate (High-Value Engineering)

- **Query tuning** and **performance optimization**
- **Schema** and **index design**
- **Capacity planning** beyond storage
- **Cost optimization** (FinOps)
- **High-availability** architecture & Cross-region DR strategy
- **Security Hardening** and compliance
- **Understanding engine-specific** behavior

PERCONA

The High-Cost of the "No DBA" Strategy

Cost Explosion – Overruns from inefficient queries and "missing index" spikes

Opaque Internals – "Black box" behavior leads to troubleshooting dead ends.

Vendor Lock-in – Deep dependency on proprietary features makes migration impossible

Security & Network Complexity – Misconfigured VPCs and cross-region gaps due to shared responsibility confusion

PERCONA

When ‘Self-Managing’ Fails: Read-World Scenarios

Availability Incident

Failover delays due to replica lag.

Automated failover isn’t instant if the data isn’t synced

Maintenance Incident

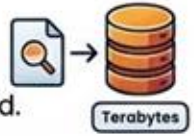
Maintenance windows causing downtime.

Forced updates occurring at peak times because no one configured the maintenance window.

Cost Incident

A missing index causing a massive cost spike.

A single bad query scanning terabytes of data because the schema wasn’t optimized.



The Lesson:
Teams that removed DBAs often find themselves **reinstating them after a major incident.**

Enter the DBRE: The Evolution of the Role

Database Administrator (DBA)

- Operations
- Maintenance
- Keeping the lights on
- Gatekeeper

Database Reliability Engineer (DBRE)

- Engineering
- Strategy
- Architecture
- Enabler

Alternative Titles: Data Platform Engineer, Cloud Database Architect

Core Shift: The focus moves from **'protecting the server'** to **'enabling the platform'** through code and design.

The Modern Mandate: Engineering Over Administration

Strategy: Multi-cloud and hybrid execution

Infrastructure as Code: Managing provisioning via code

Advisory: Advising developers on data modeling

Observability: Moving beyond dashboards to defining SLOs

Cost Governance: Actively managing cloud spend

Performance Engineering: Designing for scale and throughput

The ROI of Deep Database Expertise

Why You Still Need Humans

Performance: Cloud doesn't eliminate deadlocks or slow queries.

Architecture: Bad schema design is expensive to fix later.

Reliability: Someone must understand engine behavior under load.

Organization Impact: Teams without DBAs experience higher bills and slower troubleshooting.

Tooling Up for the Cloud-Native Stack

FinOps & Governance

Cloud cost optimization strategies

Observability

Prometheus, Grafana, OpenTelemetry, PMM

Automation/IaC

Terraform, Pulumi, Ansible, Scripting

Architecture & Engine

AWS/GCP Nuances, Distributed Systems, Storage Engines

Career Trajectories in a Cloud-First World

Summary: The Rules of Engagement Have Changed

Operational & Automation (The "What") >> **Strategic & Engineering (The "How" & "Why")**

1. DBaaS changes the DBA role - it does not eliminate it.
2. The work is now strategic (engineering) rather than operational (toil).
3. Companies still need deep expertise to manage costs, schema, and performance.
4. DBAs who evolve into DBREs and Architects will be more valuable than ever.
5. Automation handles the "what" (provisioning); Humans handle the "how" and "why" (architecture).

Building the Reliable, Hybrid Future

The future of data infrastructure is hybrid, automated, and cloud-native. However, automation is a tool, not a strategy.

DBAs are essential to making the future reliable, performance, and cost-efficient. The role is evolving – and that evolution creates the opportunity for a more resilient architecture.

ขอบคุณมาก
Questions?