Global Academic Journal of Economics and Business

Available online at https://www.gajrc.com
DOI: https://doi.org/10.36348/gajeb.2026.v08i05.014

OPEN ACCESS

ISSN: 2706-9001 (P)
ISSN: 2707-2584 (O)

Original Research Article

AI-Enabled Autonomous Cloud Database Management for Improving Performance, Reliability and Operational Efficiency in Saudi Enterprises

Naga Srawan Kumar Pabba[1]*
[1]Independent Researcher

***Corresponding Author**
Naga Srawan Kumar Pabba
Independent Researcher

Article History
Received: 19.07.2026
Accepted: 12.09.2026
Published: 14.09.2026

Abstract: Cloud databases are now foundational to enterprise digital operations; however, their elasticity and distributed architecture increase the complexity of manual administration. This review investigates how artificial intelligence can facilitate bounded autonomy in cloud database management to enhance performance, reliability, and operational efficiency, with a specific focus on Saudi enterprises. A structured integrative review synthesizes peer-reviewed literature published primarily between 2020 and 2025, spanning database systems, cloud resource management, anomaly diagnosis, autonomous tuning, and Saudi cloud adoption. Learned query optimization and cardinality estimation improve execution decisions; reinforcement and Bayesian learning accelerate configuration tuning; workload-aware resource orchestration manages latency and utilization; and AI-supported monitoring reduces diagnosis and recovery cycles. Nevertheless, the literature reveals persistent limitations: model transfer across workloads is unreliable, optimization objectives are often narrower than enterprise service-level requirements, online exploration can introduce availability risks, and most evaluations are laboratory-based rather than longitudinal production studies. Saudi adoption studies further demonstrate that security, privacy, trust, and organizational readiness influence cloud decisions, indicating that technically advanced autonomy will not be adopted without auditable controls and clear accountability. Accordingly, this review proposes a closed-loop architecture in which telemetry, AI decision engines, policy constraints, database control actions, and human oversight function as a governed feedback system. The central conclusion is that autonomous cloud database management should be implemented as progressively delegated, observable, and reversible decision-making, rather than as unrestricted automation.

Keywords: Artificial Intelligence, Autonomous Databases, Cloud Databases, Self-Driving Database, Database Tuning, Query Optimization, Aiops, Saudi Arabia, Operational Efficiency, Reliability.

1. INTRODUCTION

Cloud database services increasingly support transaction processing, analytics, digital channels, enterprise resource planning, and AI workloads. Their primary advantages include elastic capacity, rapid provisioning, managed availability, and the transfer of routine infrastructure tasks from internal teams to programmable services. However,

this elasticity also introduces greater operational complexity. Workload intensity can fluctuate within minutes, resource bottlenecks shift among compute, memory, storage, and network layers, query plans become sensitive to changing data distributions, and high availability relies on coordinated replication, failover, backup, and recovery mechanisms. As a result, the traditional model—where database administrators manually monitor dashboards and tune a limited set of stable servers—becomes increasingly difficult to scale across heterogeneous cloud environments.

Artificial intelligence is being applied to this problem under labels such as AI for databases, self-driving databases, autonomous database systems, learned optimizers, and AIOps. The common principle is to replace selected heuristic or manual decisions with models that learn from data, workload behavior, prior tuning experience, documentation, or execution feedback. Zhou *et al*., (2022) describe this broader AI-for-database agenda as extending from cost estimation and query optimization to configuration, indexing, and other management functions. OpenGauss research demonstrates how multiple learned components can be integrated into an operational database rather than studied as isolated algorithms (Li *et al*., 2021). DBMind similarly combines self-monitoring, diagnosis, and optimization, illustrating that practical autonomy is inherently a closed-loop systems problem rather than a collection of disconnected predictions (Zhou *et al.,* 2021).

The performance case for such autonomy is strong but not uniform. Learned query optimizers can adapt to workload and data changes, while learned cardinality estimators can improve a cost-based optimizer's view of intermediate result sizes. Cloud resource controllers can forecast demand or learn scaling policies under service-level constraints. At the same time, these mechanisms introduce new operational risks: inaccurate models can recommend harmful actions, reinforcement learning may explore unsafe configurations, data drift can invalidate training assumptions, and black-box reasoning can make incidents harder to audit.

These tensions are especially relevant to Saudi enterprises. Studies of cloud adoption in the Kingdom report that security, privacy, confidence in providers, organizational factors, competitive pressure, and perceived benefit affect adoption decisions (Aldahwan and Ramzan, 2021; Hamdi *et al*., 2021; Alghaith, 2023). For banks, telecommunications operators, government-linked companies, energy firms, healthcare organizations, and large diversified groups, database performance cannot be optimized independently of service continuity, data governance, vendor assurance, and accountability. An autonomous control that reduces mean query latency but increases failover uncertainty, cost volatility, or compliance exposure is not an enterprise improvement.

Accordingly, this review conceptualizes AI-enabled autonomous cloud database management as a socio-technical control system. Rather than presuming that full autonomy is universally desirable, the analysis focuses on identifying which decisions can be safely delegated, how operational policy should constrain autonomy, and what evidence remains lacking for deployment in Saudi enterprises. The primary contribution is a critical integration of previously fragmented literature into a performance-reliability-efficiency framework, supported by a governed closed-loop architecture and a research agenda tailored to the Saudi context.

2. Aim of the Study

This study seeks to critically synthesize recent peer-reviewed evidence on AI-enabled autonomous management of cloud databases and to determine how learned optimization, self-tuning, intelligent resource control, automated diagnosis, and governed human-AI decision-making can collectively enhance enterprise performance, reliability, and operational efficiency in Saudi Arabia.

2.1 Research Objectives

The review addresses five objectives. First, it evaluates the technical maturity and practical limitations of AI methods for autonomous query optimization, cardinality estimation, indexing or view selection, and database configuration. Second, it examines the impact of AI-driven workload forecasting, resource allocation, and autoscaling on latency, throughput, utilization, and cost efficiency in cloud environments. Third, it analyzes the contributions of intelligent monitoring, anomaly detection, root-cause diagnosis, and recovery assistance to database reliability. Fourth, it identifies governance, safety, explainability, and organizational requirements for delegating database decisions to autonomous agents. Fifth, it formulates a Saudi enterprise research and implementation agenda that links technical autonomy with cloud trust, security, privacy, operational readiness, and service-level accountability.

2.2 Research Questions

The synthesis is structured around five research questions. RQ1 investigates which AI-enabled database management functions demonstrate the strongest operational benefits and where their transferability is limited. RQ2 explores how autonomous tuning and resource orchestration should balance objectives such as latency,

throughput, cost, availability, and change risk. RQ3 examines which monitoring and diagnosis mechanisms can support reliable closed-loop action beyond mere detection. RQ4 considers the necessary forms of human oversight, policy constraints, explainability, validation, and rollback for safe enterprise autonomy. RQ5 addresses how technical evidence should be adapted to the trust, security, privacy, and organizational conditions identified in Saudi cloud adoption research.

3. REVIEW METHODOLOGY

A structured integrative review was selected because the research problem crosses database systems, machine learning, cloud operations, software reliability, and organizational adoption. The review protocol followed the transparency principles proposed for independent literature reviews by Kraus *et al*., (2022). It used relevant elements of the PRISMA 2020 reporting logic to make identification, eligibility, and synthesis decisions explicit without presenting a fabricated quantitative flow (Page *et al*., 2021).

The primary search window was January 2020 through December 2025. Searches were designed for Scopus and Web of Science Core Collection and were cross-checked in IEEE Xplore, ACM Digital Library, ScienceDirect, SpringerLink, and the VLDB publication archive. Search strings combined four concept groups: database autonomy ("autonomous database", "self-driving database", "automatic database tuning", "AI for database"); cloud operation ("cloud database", "resource management", "autoscaling", "microservices"); AI mechanisms ("reinforcement learning", "machine learning", "learned query optimizer", "cardinality estimation", "large language model"); and outcomes ("latency", "throughput", "reliability", "anomaly detection", "root cause", "operational efficiency"). A contextual search added "Saudi Arabia", "cloud adoption", "banking", "trust", "security", and "privacy".

Eligible sources were peer-reviewed journal articles or full research papers in established conference proceedings that directly examined autonomous or learning-enabled database management, cloud resource control, database or system anomaly diagnosis, or Saudi cloud adoption. Studies had to provide a clear method, architecture, experiment, survey design, or review procedure and a substantive link to at least one review outcome. We excluded preprints without a peer-reviewed version, vendor marketing material, news items, unpublished theses, purely conceptual commentary, and studies focused on unrelated AI applications.

Screening occurred sequentially at the title, abstract, and full-text level. Duplicate records and alternate versions of the same study were consolidated in favor of the peer-reviewed version of record. No numerical screening count is reported because the review was constructed as an integrative evidence synthesis without a prospectively archived database-export log. Reporting invented identification or exclusion totals would create false reproducibility; a journal-specific update should report counts only after the search exports, deduplication file, and exclusion decisions are archived.

Methodological quality was assessed using a purpose-built matrix rather than a single score. Technical studies were examined for benchmark realism, baseline strength, workload diversity, sensitivity to drift, online safety, reproducibility, and whether end-to-end database outcomes were measured rather than proxy prediction accuracy alone. Organizational studies were examined for sampling clarity, construct validity, analytical transparency, and relevance to Saudi enterprise adoption. Reviews were assessed for search transparency and synthesis quality. Evidence was then coded into five themes: query intelligence; configuration and resource control; autonomous database design; observability and diagnosis; and governance and adoption.

The synthesis was comparative and explanatory. Results were not pooled statistically because performance metrics, database engines, workloads, hardware, cloud configurations, and evaluation protocols varied substantially. Instead, studies were compared on decision target, learning mechanism, feedback source, operational objective, safety mechanism, and external-validity limits. Methodological limitations remain: publication bias may favor successful prototypes, conference studies often use benchmark workloads, proprietary cloud data are underrepresented, and the Saudi-specific empirical literature addresses cloud adoption more often than autonomous database operations.

4. Thematic Literature Review and Critical Analysis

The most mature work optimizes bounded decisions such as plan selection, cardinality estimation, configuration tuning, materialized views, or resource allocation. Fewer studies demonstrate durable end-to-end autonomy under production drift, mixed objectives, and regulatory constraints. This distinction is important because an algorithm that wins a benchmark may still be unsuitable for unattended enterprise control.

4.1 From DBA-Centric Administration to Bounded Database Autonomy

The transition from manual administration to autonomy is best understood as progressive delegation. OpenGauss integrates learned components for optimization and validates models before deployment, while DBMind links monitoring, diagnosis, and optimization in one platform (Li *et al*., 2021; Zhou *et al*., 2021). These systems show why autonomy requires infrastructure around the model: telemetry collection, training-data management, model versioning, validation, and operational interfaces. A self-driving database is therefore not simply a reinforcement-learning agent attached to a DBMS; it is a managed control plane that decides when a learned recommendation is sufficiently trustworthy to influence production behavior.

The distinction between recommendation and action also matters. Many enterprise tools can suggest indexes, parameters, or diagnoses, but autonomous management implies authority to execute changes and assess their effects. That authority should be bounded. For example, a low-risk decision such as changing a query hint can be reversible at query granularity, whereas modifying durability settings or replica topology can alter recovery risk. Trummer's DB-BERT demonstrates a useful intermediate pattern: it combines natural-language knowledge from manuals with runtime feedback to accelerate tuning, reducing dependence on expert preselection while still optimizing measurable objectives (Trummer, 2024).

Figure 1 conceptualizes this bounded closed loop. It separates sensing, AI inference, policy gating, execution, and outcome verification. The architecture makes two design principles explicit. First, learning models should not directly control production databases without a policy and safety layer. Second, every action should generate evidence for verification, rollback, and future learning. Autonomy becomes operationally credible when the loop is observable and reversible.

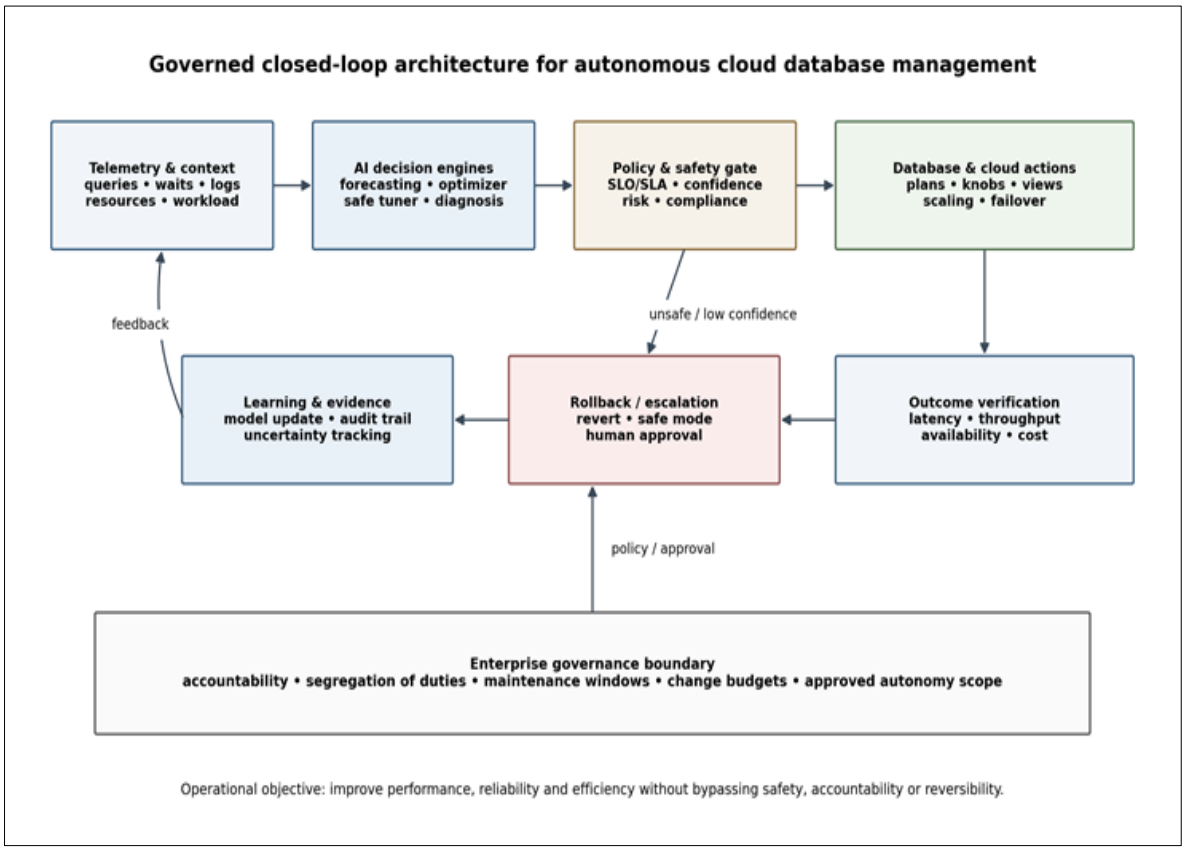

Figure 1: Governed closed-loop architecture for AI-enabled autonomous cloud database management

4.2 Learned Query Optimization and Cardinality Intelligence

Query optimization is a central target because poor plan selection can dominate database latency. Traditional cost-based optimizers depend on cardinality estimates, cost models, and plan enumeration, each of which introduces error (Lan *et al*., 2021). Recent systems do not uniformly replace the native optimizer. Bao augments an existing optimizer with learned hints and adapts through bandit feedback, reducing the cost of learning from scratch and improving robustness to changing workloads (Marcus *et al*., 2021). Balsa moves further toward learned planning by training without expert

demonstrations, but its learning process still depends on controlled execution and simulation-to-real transfer (Yang *et al*., 2022). Lero adopts a learning-to-rank strategy that compares plans rather than predicting exact latency, demonstrating a pragmatic way to reduce model difficulty while preserving the accumulated strengths of the underlying DBMS (Zhu *et al*., 2023).

These approaches differ in the degree of autonomy and risk. For enterprise clouds, this hybridity is attractive because query plans can be rolled back quickly, while replacing the entire optimizer creates a larger validation burden.

Cardinality estimation illustrates a similar trade-off between accuracy, inference cost, and generalization. DeepDB learns data distributions rather than query workloads, reducing dependence on executed training queries and supporting changing workloads (Hilprecht *et al*., 2020). NeuroCard models correlations across tables with a neural density estimator (Yang *et al*., 2020), while FLAT combines independent and conditional factorization to improve the balance among accuracy, model size, and inference speed (Zhu *et al*., 2021). Comparative analysis by Sun *et al*., (2022) shows that learned estimators have materially different strengths depending on data distributions, query templates, update patterns, and latency constraints; improvements in q-error do not automatically translate into better end-to-end plans. FactorJoin narrows this deployment gap by using single-table learned distributions and factor graphs for joins, emphasizing fast training and estimation (Wu *et al*., 2023). ASM further combines autoregressive models, sampling, and multidimensional statistics to address efficiency and query coverage simultaneously (Kim *et al.*, 2024).

For autonomous management, the lesson is that prediction accuracy is an intermediate metric. A cardinality model should be judged by plan quality, tail latency, retraining cost, update tolerance, and the consequences of error. In Saudi enterprises with high-volume transactional services, a model that is marginally more accurate but expensive to retrain or difficult to explain may be less valuable than a slightly less accurate estimator that behaves predictably during workload shifts. Learned components should therefore be selected against service objectives, not leaderboard accuracy alone.

4.3 Configuration Tuning, Resource Orchestration and Cost Efficiency

CDBTune+ applies deep reinforcement learning to cloud database tuning and uses performance feedback to search continuous configuration spaces, demonstrating adaptation across workload and hardware changes (Zhang *et al*., 2021a). ResTune adds meta-learning and resource-oriented objectives, so experience from prior tuning tasks can accelerate new cloud configurations (Zhang *et al.,* 2021b). LlamaTune addresses sample efficiency through search-space transformations, an important operational issue because each tuning experiment consumes time and resources and may disturb production (Kanellis *et al*., 2022). Zhao *et al*., (2023) later confirm that high dimensionality, expensive evaluation, scenario diversity, and transferability remain fundamental challenges.

The strongest operational advance is the move from offline tuning toward safe online adaptation. OnlineTune explicitly addresses dynamic workloads and uses safety knowledge to constrain exploration while adjusting configurations in production-like settings (Zhang *et al*., 2022). This is conceptually more important for enterprises than improving average benchmark throughput. Autonomous databases need to adapt without violating availability or latency objectives during learning. Safe exploration, context awareness, and rollback therefore become first-class design requirements.

Resource management broadens the control surface beyond DBMS knobs. Sinan uses machine learning for QoS-aware resource management of microservices and demonstrates that tail latency depends on interactions across services, not simply average CPU utilization (Zhang *et al*., 2021c). Abdel Khaleq and Ra (2021) similarly use reinforcement learning to identify autoscaling thresholds under response-time constraints. For cloud databases embedded in microservice architectures, database tuning and infrastructure scaling cannot be optimized independently: changing buffer allocation may alter CPU and I/O demand, while autoscaling can change cache warmth, connection distribution, and replica load.

Operational efficiency should also be defined more broadly than infrastructure cost. Repeated tuning experiments consume engineering time; overprovisioning wastes budget; slow diagnosis extends incidents; and conservative static configurations leave performance unused. A controller that minimizes compute cost while increasing incident frequency is not efficient. Multi-objective policies should combine latency, throughput, utilization, cloud spend, change frequency, availability risk, and recovery objectives, with hard constraints for non-negotiable service levels.

4.4 Autonomous Database Design and Workload Adaptation

Autonomy also extends to physical database design. Materialized views can reduce repeated

computation but create storage and maintenance costs, making selection a workload-sensitive optimization problem. AutoView uses learned representations to manage materialized views automatically and shows how changing query demand can drive database design rather than periodic manual review (Han *et al*., 2023).

However, automatic design introduces feedback interactions. Adding a view changes optimizer choices, cache behavior, storage consumption, write amplification, and maintenance work. An autonomous controller should therefore evaluate interventions at the system level rather than treating each design task independently. The same principle applies to indexes, partitioning, caching, and replica placement. When multiple autonomous modules act concurrently, locally beneficial actions can conflict. A tuner may increase memory for caching while a resource controller scales down the instance; a view manager may increase storage while a cost optimizer attempts to reduce it. Coordination mechanisms, priorities, and change budgets are required to prevent control oscillation.

This systems perspective also changes how learning should be evaluated. Offline benchmarks normally isolate one decision variable, but enterprise operation involves sequences of interacting changes. Future autonomous platforms need orchestration policies that decide not only what change is beneficial but also whether now is the right time to make it, what other automated actions are active, and how to attribute subsequent performance changes.

4.5 Reliability: Observability, Anomaly Detection, Diagnosis and Self-Healing

Performance autonomy is incomplete without reliability autonomy. Cloud databases produce high-volume telemetry across queries, locks, caches, storage, replication, operating systems, containers, and application dependencies. AI can help prioritize anomalies and connect symptoms that are difficult to interpret manually. Han *et al*., (2021) show that robust feature extraction and online learning can improve log-based anomaly detection as system behavior evolves, while LogBERT uses self-supervised transformer representations to learn normal log sequences without requiring exhaustive anomaly labels (Guo *et al*., 2021). These methods are relevant to databases because abnormal behavior often appears first in logs and metrics rather than in explicit failure signals.

Detection, however, is only the beginning of operational value. DBMind couples anomaly identification with root-cause diagnosis and optimization, while D-Bot uses large language models, retrieved diagnostic knowledge, tool use, and structured search to produce database diagnoses for complex anomalies (Zhou *et al*., 2021; Zhou *et al*., 2024).

The reliability risk is automation bias. A plausible diagnosis can still be wrong, and a wrong automated remediation can turn a degraded service into an outage. Self-healing should therefore be stratified by action risk. Low-risk actions, such as collecting additional diagnostics, canceling a clearly pathological query, or reverting a recently applied plan hint, can be automated with relatively narrow safeguards. Higher-risk actions, including failover, topology change, durability modification, or broad parameter resets, require stronger evidence, quorum or policy checks, and sometimes human approval. Reliability autonomy should optimize expected service continuity, not merely response speed.

Table 1 synthesizes the major autonomous capabilities and their operational trade-offs. Across themes, the recurring pattern is that AI produces the greatest enterprise value when it operates inside a constrained feedback loop with explicit service objectives, uncertainty awareness, and reversible actions.

Table 1: Autonomous cloud database capabilities, benefits and control requirements

Capability	Representative evidence / approach	Enterprise benefit	Principal control requirement
Query and cardinality intelligence	Bao, Balsa, Lero; DeepDB, NeuroCard, FLAT, FactorJoin, ASM	Lower query latency; more adaptive plan selection	Tail-risk testing, native-optimizer fallback, drift monitoring
Configuration tuning	CDBTune+, ResTune, LlamaTune, OnlineTune, DB-BERT	Higher throughput; less manual search; faster adaptation	Safe exploration, bounded knob ranges, rollback and change windows
Cloud resource orchestration	Sinan; reinforcement-learning autoscaling	QoS-aware capacity; lower overprovisioning	Hard SLO constraints, scaling-rate limits, cross-layer coordination
Physical design adaptation	AutoView and workload-sensitive materialized-view management	Reduced repeated computation; less periodic DBA design work	Storage/maintenance budgets, attribution of effects, reversible deployment
Reliability and diagnosis	DBMind, robust log learning, LogBERT, D-Bot	Earlier detection; faster root-cause analysis; shorter recovery cycles	Evidence-backed diagnosis, action-risk tiers, escalation for high-impact remediation

Capability	Representative evidence / approach	Enterprise benefit	Principal control requirement
Governed autonomy	Policy gate around telemetry, model decisions and actions	Consistent operations with auditable delegation	Human override, accountability, confidence thresholds and full audit trail

4.6 Saudi Enterprise Context: Adoption, Trust and Governance

Saudi cloud adoption research does not yet provide direct longitudinal evidence on autonomous database control, but it identifies conditions that will shape deployment. Aldahwan and Ramzan (2021) report that security, confidentiality, integrity, availability, trust, and economic considerations affect organizational acceptance of community cloud. In the Saudi banking sector, Hamdi *et al*., (2021) find that security and privacy concerns can discourage adoption while perceived benefits and competitive pressure encourage it. Alghaith (2023) further highlights the role of provider trust, reputation, and service-level agreement verification in cloud adoption intentions.

These findings imply that autonomy must be made governable. A Saudi enterprise cannot treat an autonomous database as a black box whose decisions are delegated entirely to a provider. Procurement and architecture teams need evidence about where telemetry and model artifacts are processed, which actions the service may perform automatically, how service-level commitments are verified, how changes are logged, and how humans can intervene.

Governance should therefore be encoded in the control plane. Policy constraints can define maximum scaling rates, prohibited parameter ranges, approved maintenance windows, data-location requirements, rollback thresholds, and actions requiring dual authorization. Model confidence should affect delegation level. When a controller encounters an unfamiliar workload or a diagnosis with weak evidence, the correct autonomous action may be to reduce its own authority and escalate. Such graceful degradation signals mature autonomy, not failure.

This tiered model strengthens accountability without sacrificing adaptive performance.

Figure 2 presents a Saudi-oriented adoption model in which organizational readiness and governance mediate technical autonomy. The model rejects a purely technological maturity ladder. An enterprise becomes ready for bounded autonomy when observability, data quality, staff capability, change governance, cloud trust, and service-level accountability develop alongside the AI components.

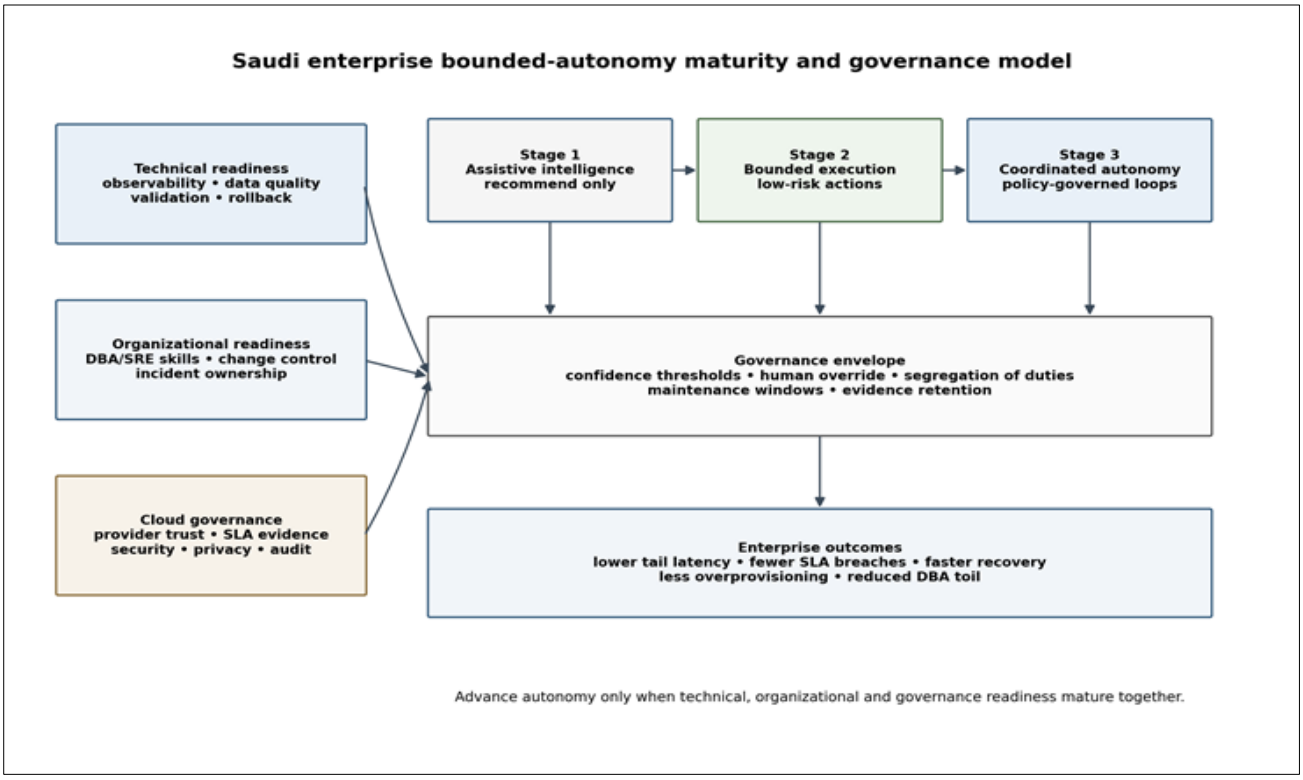

Figure 2: Saudi enterprise bounded-autonomy maturity and governance model

5. DISCUSSION

Query optimization, cardinality estimation, configuration tuning, resource scaling, physical design, anomaly detection, and diagnosis have all demonstrated credible advances, but they operate at different levels of maturity. Online knob tuning, failover, and infrastructure reconfiguration are higher-risk because learning mistakes can affect availability, durability, or cost. Treating all of these functions as a single category of "autonomous database" obscures the engineering controls required for safe delegation.

A second finding is that the performance, reliability, and efficiency objectives are coupled. Faster queries can increase resource consumption; aggressive consolidation can reduce resilience headroom; frequent tuning can create operational instability; and anomaly detectors can reduce response time while increasing alert noise. Future systems should extend this logic to enterprise multi-objective policies in which some metrics are optimized, and others are hard constraints.

Third, the evidence favors hybrid intelligence over unrestricted model replacement. Bao and Lero exploit native optimizer knowledge, DB-BERT combines textual expert knowledge with feedback, and openGauss surrounds learned components with validation and model management. These designs preserve institutional knowledge and create fallback paths. They are also easier to explain to risk, audit, and operations teams than an opaque agent with broad production authority.

For Saudi enterprises, the implications are clear. Cloud trust and SLA verification are not peripheral adoption issues; they define whether autonomous actions will be accepted. The enterprise architecture should make the autonomy boundary contractible and auditable: which telemetry feeds models, which actions are permitted, what evidence precedes high-impact changes, and how responsibility is divided between enterprise teams and cloud providers. The strategic opportunity is therefore not "lights-out" database administration, but a measurable reduction in repetitive human toil while specialists retain authority over exceptional, ambiguous, and high-consequence decisions.

6. Research Gaps and Future Research

Five gaps deserve priority. First, Saudi-specific production evidence is almost absent. Future studies should evaluate autonomous tuning and diagnosis on representative banking, telecom, government-linked, energy, retail, and healthcare workloads while respecting confidentiality through controlled testbeds or privacy-preserving telemetry. Second, cross-workload and cross-engine transfer remains weak. Research should test whether models trained on one database version, instance family, or workload can estimate uncertainty reliably when moved to another environment, rather than reporting only average transfer performance.

Third, multi-objective safe control is underdeveloped. Future work should model latency, throughput, cost, carbon or energy use, availability, recovery risk, and change frequency simultaneously, with explicit hard constraints and risk budgets. Fourth, autonomous modules need coordination. Benchmarks should evaluate sequences of interacting actions across query optimization, tuning, scaling, physical design, and incident remediation to detect oscillation and conflicting recommendations. Fifth, explainability must move from post-hoc visualization to operational evidence: controllers should state why an action is proposed, which telemetry supports it, expected benefit, uncertainty, rollback condition, and which policy authorized execution.

The research agenda in Table 2 translates these gaps into testable Saudi-enterprise questions and measurable outcomes. Longitudinal field studies are especially important because reliability and organizational trust cannot be inferred from short benchmark runs. Future evaluations should therefore include sustained operation, incident drills, model update cycles, audit review, and human override behaviour.

Table 2: Saudi-oriented research gaps and future research agenda

Research gap	Current evidence limitation	Saudi-oriented research design	Suggested outcomes
Localized production benchmarks	Most database-autonomy evaluations use public or controlled workloads	Controlled testbeds and privacy-preserving field studies across banking, telecom, energy, retail, healthcare and government-linked enterprises	Tail latency, throughput, SLO breaches, availability, tuning time
Safe multi-objective control	Studies often optimize one or two performance metrics	Constrained policies combining latency, cost, capacity, recovery risk and change frequency	Constraint violations, cloud spend, recovery time, rollback rate

Research gap	Current evidence limitation	Saudi-oriented research design	Suggested outcomes
Transfer and drift	Generalization across engines, versions and instance families is uncertain	Longitudinal transfer tests with explicit uncertainty calibration and model update cycles	Performance decay, calibration error, retraining cost, drift detection delay
Coordinated autonomy	Query, tuning, scaling and physical-design modules are usually evaluated separately	Multi-controller experiments that expose conflicting actions and oscillation	Control stability, action conflicts, cumulative service impact
Explainability and human oversight	Post-hoc explanations rarely support operational authorization	Human-in-the-loop studies requiring rationale, evidence, confidence and rollback condition before execution	Decision acceptance, override rate, incident quality, audit completeness
Provider accountability and SLA evidence	Saudi studies emphasize trust, security and SLA verification but not autonomous database action	Contract and governance studies defining provider/customer responsibility for automated changes	Traceability, SLA verifiability, segregation of duties, time to evidence

7. Practical, Managerial and Policy Implications

Practically, enterprises should begin with high-observability, reversible use cases. Learned diagnosis, query-plan advice, index or view recommendations, and bounded autoscaling can demonstrate value before granting broader action authority. Each autonomous function should have an owner, service objective, action scope, confidence threshold, audit log, rollback path, and fallback mode. Shadow deployment is valuable: models can recommend actions without executing them until their performance is compared against DBA decisions across representative workloads.

For managers, the business case should measure more than headcount reduction. Relevant benefits include lower tail latency, fewer SLA breaches, faster incident triage, reduced overprovisioning, fewer emergency changes, and increased DBA capacity for architecture and data engineering. Investment is also required in observability, telemetry quality, model operations, and skills. Poor telemetry makes autonomous control unreliable, so monitoring infrastructure is part of the AI investment rather than a separate prerequisite.

At the policy and governance level, Saudi enterprises should require transparent allocation of responsibility between cloud providers and customers. Contracts and internal standards should specify how automated changes are recorded, how service-level evidence is produced, how sensitive operational data are handled, and which decisions remain human-controlled. High-impact autonomy should be treated as a controlled production change mechanism subject to testing, segregation of duties, and periodic assurance. This approach aligns innovation with the trust and security concerns documented in Saudi cloud adoption research rather than framing governance as an obstacle to automation.

8. Limitations

This review is limited by heterogeneity in benchmarks, engines, workloads, and outcome measures, which prevents meaningful meta-analysis. Publication bias may overrepresent successful learning-based prototypes, and several influential database papers evaluate controlled benchmarks rather than long-running production systems. Saudi-specific literature remains concentrated on cloud adoption rather than autonomous database operations, so contextual implications are analytical transfers rather than direct causal evidence. The review also excludes non-peer-reviewed vendor evidence even though commercial autonomous databases may contain operational practices not described in academic publications. Finally, fast-moving large-language-model and agentic database research means that some capabilities will evolve faster than conventional peer-review cycles.

9. CONCLUSION

AI-enabled autonomous cloud database management is becoming technically credible, but its enterprise value depends on how autonomy is bounded and governed. The strongest recent evidence shows that learned query optimization, cardinality estimation, configuration tuning, resource orchestration, physical design, anomaly detection, and diagnostic reasoning can outperform static or manual approaches in selected tasks. Yet no single technique provides reliable end-to-end autonomy across changing workloads, engines, infrastructure, and organizational constraints. The practical unit of design is therefore the closed feedback loop: observe system state, infer or predict, evaluate policy and safety constraints, act within an approved scope, verify outcomes, and roll back or escalate when confidence falls.

For Saudi enterprises, this architecture is particularly appropriate because cloud adoption is

shaped by security, privacy, trust, provider reputation, and verifiable service commitments. These conditions do not argue against autonomous database management; they define the control requirements that make it acceptable. Enterprises should move from recommendation to bounded execution in stages, beginning with reversible actions and expanding authority only when telemetry quality, validation evidence, governance, and staff readiness mature.

The review contributes a synthesis that links performance engineering with reliability and operational efficiency rather than treating autonomous tuning as an isolated machine-learning problem. It also identifies a central research need: production evidence demonstrating safe multi-objective autonomy under Saudi organizational and regulatory conditions. Progress should be judged not by whether AI can replace a database administrator, but by whether a governed system can reduce repetitive toil, make faster and more consistent decisions, preserve service continuity, and expose enough evidence for humans to remain accountable. Under that standard, the most promising future is not unbounded self-driving infrastructure, but trustworthy database autonomy that knows both how to act and when not to act.

REFERENCES

- Abdel Khaleq, A., & Ra, I. (2021). Intelligent Autoscaling of Microservices in the Cloud for Real-Time Applications. IEEE Access, 9, 35464–35476. https://doi.org/10.1109/ACCESS.2021.3061890
- Aldahwan, N. S., & Ramzan, M. S. (2021). Factors Affecting the Organizational Adoption of Secure Community Cloud in KSA. Security and Communication Networks, 2021, Article 8134739. https://doi.org/10.1155/2021/8134739
- Alghaith, W. (2023). The Role of Trust in Cloud Service Providers on the Adoption of Cloud Computing in Saudi Arabia: An Empirical Investigation. Journal of Computer Science, 19(9), 1107–1123. https://doi.org/10.3844/jcssp.2023.1107.1123
- Guo, H., Yuan, S., & Wu, X. (2021). LogBERT: Log Anomaly Detection via BERT. In 2021 International Joint Conference on Neural Networks (IJCNN), 1–8. https://doi.org/10.1109/IJCNN52387.2021.9534113
- Hamdi, M., Olayah, F., Al-Awady, A. A., Shamsan, A. F., & Ghilan, M. M. (2021). Attitude Towards Adopting Cloud Computing in the Saudi Banking Sector. Intelligent Automation & Soft Computing, 29(2), 605–617. https://doi.org/10.32604/iasc.2021.018170
- Han, S., Wu, Q., Zhang, H., Qin, B., Hu, J., Shi, X., Liu, L., & Yin, X. (2021). Log-Based Anomaly Detection With Robust Feature Extraction and Online Learning. IEEE Transactions on Information Forensics and Security, 16, 2300–2311. https://doi.org/10.1109/TIFS.2021.3053371
- Han, Y., Li, G., Yuan, H., & Sun, J. (2023). AutoView: An Autonomous Materialized View Management System With Encoder-Reducer. IEEE Transactions on Knowledge and Data Engineering, 35(6), 5626–5639. https://doi.org/10.1109/TKDE.2022.3163195
- Hilprecht, B., Schmidt, A., Kulessa, M., Molina, A., Kersting, K., & Binnig, C. (2020). DeepDB: Learn from Data, not from Queries! Proceedings of the VLDB Endowment, 13(7), 992–1005. https://doi.org/10.14778/3384345.3384349
- Kanellis, K., Ding, C., Kroth, B., Müller, A. C., Curino, C., & Venkataraman, S. (2022). LlamaTune: Sample-Efficient DBMS Configuration Tuning. Proceedings of the VLDB Endowment, 15(11), 2953–2965. https://doi.org/10.14778/3551793.3551844
- Kim, K., Lee, S., Kim, I., & Han, W.-S. (2024). ASM: Harmonizing Autoregressive Model, Sampling, and Multi-dimensional Statistics Merging for Cardinality Estimation. Proceedings of the ACM on Management of Data, 2(1), Article 45, 1–27. https://doi.org/10.1145/3639300
- Kraus, S., Breier, M., Lim, W. M., Dabić, M., Kumar, S., Kanbach, D., Mukherjee, D., Corvello, V., Piñeiro-Chousa, J., Liguori, E., Palacios-Marqués, D., Schiavone, F., Ferraris, A., Fernandes, C., & Ferreira, J. J. (2022). Literature reviews as independent studies: guidelines for academic practice. Review of Managerial Science, 16, 2577–2595. https://doi.org/10.1007/s11846-022-00588-8
- Lan, H., Bao, Z., & Peng, Y. (2021). A Survey on Advancing the DBMS Query Optimizer: Cardinality Estimation, Cost Model, and Plan Enumeration. Data Science and Engineering, 6, 86–101. https://doi.org/10.1007/s41019-020-00149-7
- Li, G., Zhou, X., Sun, J., Yu, X., Han, Y., Jin, L., Li, W., Wang, T., & Li, S. (2021). openGauss: An Autonomous Database System. Proceedings of the VLDB Endowment, 14(12), 3028–3041. https://doi.org/10.14778/3476311.3476380
- Marcus, R., Negi, P., Mao, H., Tatbul, N., Alizadeh, M., & Kraska, T. (2021). Bao: Making Learned Query Optimization Practical. In Proceedings of the 2021 International Conference on Management of Data (SIGMOD '21), 1275–1288. https://doi.org/10.1145/3448016.3452838
- Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A.,

Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., McGuinness, L. A., Stewart, L. A., Thomas, J., Tricco, A. C., Welch, V. A., Whiting, P., & Moher, D. (2021). The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. BMJ, 372, n71. https://doi.org/10.1136/bmj.n71

- Sun, J., Zhang, J., Sun, Z., Li, G., & Tang, N. (2022). Learned Cardinality Estimation: A Design Space Exploration and A Comparative Evaluation. Proceedings of the VLDB Endowment, 15(1), 85–97. https://doi.org/10.14778/3485450.3485459
- Trummer, I. (2024). DB-BERT: making database tuning tools 'read' the manual. The VLDB Journal, 33, 1085–1104. https://doi.org/10.1007/s00778-023-00831-y
- Wu, Z., Negi, P., Alizadeh, M., Kraska, T., & Madden, S. (2023). FactorJoin: A New Cardinality Estimation Framework for Join Queries. Proceedings of the ACM on Management of Data, 1(1), Article 41, 1–27. https://doi.org/10.1145/3588721
- Yang, Z., Chiang, W.-L., Luan, S., Mittal, G., Luo, M., & Stoica, I. (2022). Balsa: Learning a Query Optimizer Without Expert Demonstrations. In Proceedings of the 2022 International Conference on Management of Data (SIGMOD '22), 931–944. https://doi.org/10.1145/3514221.3517885
- Yang, Z., Kamsetty, A., Luan, S., Liang, E., Duan, Y., Chen, X., & Stoica, I. (2020). NeuroCard: One Cardinality Estimator for All Tables. Proceedings of the VLDB Endowment, 14(1), 61–73. https://doi.org/10.14778/3421424.3421432
- Zhang, J., Zhou, K., Li, G., Liu, Y., Xie, M., Cheng, B., & Xing, J. (2021a). CDBTune+: An efficient deep reinforcement learning-based automatic cloud database tuning system. The VLDB Journal, 30, 959–987. https://doi.org/10.1007/s00778-021-00670-9
- Zhang, X., Wu, H., Chang, Z., Jin, S., Tan, J., Li, F., Zhang, T., & Cui, B. (2021b). ResTune: Resource Oriented Tuning Boosted by Meta-Learning for Cloud Databases. In Proceedings of the 2021 International Conference on Management of Data (SIGMOD '21), 2102–2114. https://doi.org/10.1145/3448016.3457291
- Zhang, X., Wu, H., Li, Y., Tan, J., Li, F., & Cui, B. (2022). Towards Dynamic and Safe Configuration Tuning for Cloud Databases. In Proceedings of the 2022 International Conference on Management of Data (SIGMOD '22), 631–645. https://doi.org/10.1145/3514221.3526176
- Zhang, Y., Hua, W., Zhou, Z., Suh, G. E., & Delimitrou, C. (2021c). Sinan: ML-Based and QoS-Aware Resource Management for Cloud Microservices. In Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '21), 167–181. https://doi.org/10.1145/3445814.3446693
- Zhao, X., Zhou, X., & Li, G. (2023). Automatic Database Knob Tuning: A Survey. IEEE Transactions on Knowledge and Data Engineering, 35(12), 12470–12490. https://doi.org/10.1109/TKDE.2023.3266893
- Zhou, X., Chai, C., Li, G., & Sun, J. (2022). Database Meets Artificial Intelligence: A Survey. IEEE Transactions on Knowledge and Data Engineering, 34(3), 1096–1116. https://doi.org/10.1109/TKDE.2020.2994641
- Zhou, X., Jin, L., Sun, J., Zhao, X., Yu, X., Feng, J., Li, S., Wang, T., Li, K., & Liu, L. (2021). DBMind: A Self-Driving Platform in openGauss. Proceedings of the VLDB Endowment, 14(12), 2743–2746. https://doi.org/10.14778/3476311.3476334
- Zhou, X., Li, G., Sun, Z., Liu, Z., Chen, W., Wu, J., Liu, J., Feng, R., & Zeng, G. (2024). D-Bot: Database Diagnosis System using Large Language Models. Proceedings of the VLDB Endowment, 17(10), 2514–2527. https://doi.org/10.14778/3675034.3675043
- Zhu, R., Chen, W., Ding, B., Chen, X., Pfadler, A., Wu, Z., & Zhou, J. (2023). Lero: A Learning-to-Rank Query Optimizer. Proceedings of the VLDB Endowment, 16(6), 1466–1479. https://doi.org/10.14778/3583140.3583160
- Zhu, R., Wu, Z., Han, Y., Zeng, K., Pfadler, A., Qian, Z., Zhou, J., & Cui, B. (2021). FLAT: Fast, Lightweight and Accurate Method for Cardinality Estimation. Proceedings of the VLDB Endowment, 14(9), 1489–1502. https://doi.org/10.14778/3461535.3461539