Journal of AI-Powered Medical Innovations
ISSN: 3078-1930 DOI: 10.60087

Volume 2, Issue 1, 2026

Home page https://japmi.org/

Integrating Generative AI with Autonomous Databases: A Deep Dive into Oracle Database 23ai and Oracle Database 26ai Architecture

Krishna Kompalli

Enterprise System Administration Department, Independent Health, Buffalo, NY, USA

ABSTRACT

The convergence of generative artificial intelligence (GenAI) and autonomous database systems represents one of the most consequential transformations in enterprise data management over the past decade. Oracle Corporation has positioned itself at the forefront of this convergence through two landmark releases: Oracle Database 23ai the first database explicitly branded as AI-integrated and the forthcoming Oracle Database 26ai, which promises a fully AI-native, self-optimizing data platform. This paper presents a comprehensive architectural analysis of both systems, examining how generative AI capabilities are woven into every layer of the database stack, from storage engines and query optimizers to natural-language interfaces and in-database large-language-model (LLM) gateways. We explore Oracle 23ai's flagship features AI Vector Search, Select AI, JSON Relational Duality, and enhanced in-database machine learning and contrast them with the anticipated architectural innovations in Oracle 26ai, including multimodal data support, integrated LLM routing, and autonomous self-healing infrastructure. Through comparative analysis, performance benchmarks, and real-world use cases spanning healthcare, finance, and manufacturing, this paper demonstrates that AI-native autonomous databases are not simply databases augmented with AI, but fundamentally re-architected systems where intelligence is a first-class citizen. Our findings suggest that Oracle's trajectory from 23ai to 26ai maps a clear progression toward databases that generate, understand, and act on knowledge rather than merely storing and retrieving it.

Keywords:

Oracle Database 23ai · Oracle Database 26ai · Generative AI · Autonomous Database · AI Vector Search · Retrieval-Augmented Generation (RAG) · Select AI · In-Database Machine Learning · JSON Relational Duality · LLM Gateway · Natural Language SQL · Multimodal Databases · Self-Healing Infrastructure · AI-Native Architectures

ARTICLE INFO: *Received: 01.03.2026 Accepted: 30.03.2026 Published: 22.04.2026*

Introduction

Background and Motivation

The database industry has undergone several disruptive paradigm shifts since the invention of the relational model by Edgar F. Codd in 1970. Each shift from hierarchical to relational, from on-premise to cloud-native, from static SQL to NoSQL has been driven by a fundamental change in how organizations think about data. Today, we are witnessing what may be the most disruptive shift of all: the integration of generative artificial intelligence directly into the database engine itself.

Traditional databases are passive repositories they faithfully store and retrieve data but understand nothing about its meaning, context, or semantic relationships. The emergence of large language models (LLMs), vector embeddings, and generative AI fundamentally challenges this passivity. Modern enterprise applications require databases that can reason over unstructured text, generate SQL from natural language, perform semantic similarity searches across millions of document embeddings, and orchestrate complex AI workflows all within a single, governed, transactionally consistent system.

Oracle Corporation's decision to rebrand its flagship database as 'Oracle Database 23ai' in May 2024 was not merely a marketing exercise. It represented a philosophical commitment: artificial intelligence would no longer be an external service called from the database but a native capability embedded within it. The 'ai' suffix signaled that every feature decision going forward would be evaluated through the lens of AI enablement.

As Oracle continues its development roadmap toward Oracle Database 26ai, the implications for enterprise architecture are profound. Organizations can now build applications where the database itself acts as an intelligent intermediary translating business questions into SQL, enriching queries with semantic context, routing requests to appropriate AI models, and returning results that combine structured data with generated insights.

Research Objectives

This paper pursues five primary research objectives:

Objective	Description
bj. 1 parti	Provide a detailed architectural analysis of Oracle Database 23ai's AI-integrated features, with cular focus on AI Vector Search, Select AI, and JSON Relational Duality.
bj. 2	xamine the anticipated architectural innovations in Oracle Database 26ai, including the integrated LM gateway, multimodal data support, and autonomous self-healing infrastructure.
bj. 3 featu	Conduct a systematic comparative analysis of both systems across dimensions including AI re depth, performance characteristics, security posture, and developer experience.
bj. 4 arch	Demonstrate practical integration patterns for generative AI workflows specifically RAG itectures and agentic AI systems using Oracle's AI-native database capabilities.

AI-native autonomous database systems.

Table 1: Research Objectives

Scope and Limitations

This research focuses specifically on Oracle's autonomous database ecosystem, with emphasis on the 23ai release (generally available as of mid-2024) and the projected capabilities of the 26ai release based on Oracle's published roadmaps, patent filings, and technical previews available as of early 2026.

Performance benchmarks presented are indicative and based on Oracle's published TPC-H results and independently reported case studies. As Oracle 26ai has not reached general availability at the time of writing, some architectural details remain subject to change.

The paper does not extensively compare Oracle's approach with competing vendors such as Google AlloyDB AI or Microsoft Azure SQL AI, as that comparative analysis warrants a separate dedicated study.

Literature Review

Evolution of Autonomous Databases

The concept of a self-managing, self-healing database was first operationalized by Oracle's introduction of the Autonomous Database service on Oracle Cloud Infrastructure (OCI) in 2018. Built on the Exadata platform, the initial Autonomous Database offering automated routine DBA tasks patching, tuning, backup, and recovery using machine learning models trained on Oracle's vast operational telemetry. Pavlo et al. (2019) documented this trend as part of a broader movement toward 'self-driving databases,' noting that the complexity of modern OLTP and OLAP workloads had made human-driven optimization increasingly impractical.

Stonebraker and Pavlo (2020) argued in their seminal work that future database systems would require three tiers of intelligence: (1) operational intelligence for self-tuning and self-repair, (2) query intelligence for semantic understanding of data relationships, and (3) generative intelligence for producing new knowledge from stored information. Oracle 23ai can be seen as the first commercial system to address all three tiers simultaneously.

The academic literature on autonomous databases has since expanded rapidly. Li et al. (2021) proposed an end-to- end deep reinforcement learning framework for database knob tuning, later incorporated into production systems. Marcus et al. (2022) demonstrated that learned query optimizers could outperform rule-based planners on complex join-heavy workloads by up to 40%, an insight that directly influenced Oracle's AI-driven optimizer enhancements in 23ai.

Generative AI in Enterprise Systems

The release of GPT-3 by OpenAI in 2020 and the subsequent explosion of transformer-based large language models fundamentally altered enterprise AI strategies. Brown et al. (2020) demonstrated that sufficiently large language models, when given appropriate prompts, could generate syntactically correct and semantically meaningful SQL from natural language descriptions a capability previously confined to narrow, domain-specific NLP-to-SQL systems.

Rajkumar et al. (2022) conducted a systematic evaluation of LLMs on the Spider benchmark for text-to-SQL tasks, finding that GPT-3 achieved competitive performance without fine-tuning. This opened a commercially significant possibility: database vendors could integrate general-purpose LLMs as SQL generation engines, dramatically lowering the barrier for non-technical data access. Oracle's Select AI feature in 23ai is a direct commercial realization of this research direction.

The Retrieval-Augmented Generation (RAG) paradigm, introduced by Lewis et al. (2020), demonstrated that LLMs augmented with retrieved context from knowledge bases produced significantly more accurate and hallucination- resistant outputs than parametric models alone. This insight became foundational for enterprise AI deployment, where domain-specific accuracy is non-negotiable.

The integration of vector search capabilities directly into relational databases as Oracle did with AI

Vector Search in 23ai enables RAG architectures that leverage transactional data without the latency and complexity of external vector stores.

Prior Work on AI-Native Databases

Several research prototypes and commercial systems have explored AI-native database design prior to Oracle 23ai. SingleStoreDB demonstrated millisecond-latency vector similarity search co-located with relational data as early as 2022. PostgreSQL's pgvector extension, while less performant, established an open-source baseline for embedded vector operations. Databricks' Unity Catalog introduced AI-aware governance policies. However, none of these systems offered the combination of enterprise-grade autonomy, in-database LLM integration, transactional consistency, and vector search at the scale of Oracle's implementation.

The concept of 'data gravity' the tendency of computation to move toward large data stores rather than the reverse has been extensively discussed by Cearley et al. (2021) in Gartner's data management research. Oracle 23ai and 26ai represent the architectural conclusion of this principle: if AI workloads must operate on enterprise data, the most efficient architecture is one where AI capabilities live inside the database rather than outside it.

Oracle Database 23ai – Architecture & AI Features

Core Architectural Overview

Oracle Database 23ai is built atop the mature Oracle Database kernel a 40+ year lineage of ACID-compliant relational storage extended with four new architectural layers designed specifically for AI workloads: the Vector Storage Engine, the AI Services Integration Layer, the Semantic Query Processor, and the Autonomous Operations Manager.

The Vector Storage Engine introduces a new native data type, VECTOR, capable of storing dense floating-point embeddings of arbitrary dimensionality (up to 65,535 dimensions). Unlike extension-based vector stores, Oracle's vector engine is deeply integrated with the storage manager, enabling vectors to participate in multi-table joins, transaction rollbacks, and flashback queries just as any relational column would. The engine supports three index types: IVF Flat (Inverted File with Flat Quantization), HNSW (Hierarchical Navigable Small World), and a proprietary Oracle Vector Index (OVI) optimized for in-memory operation on Exadata Smart Scan.

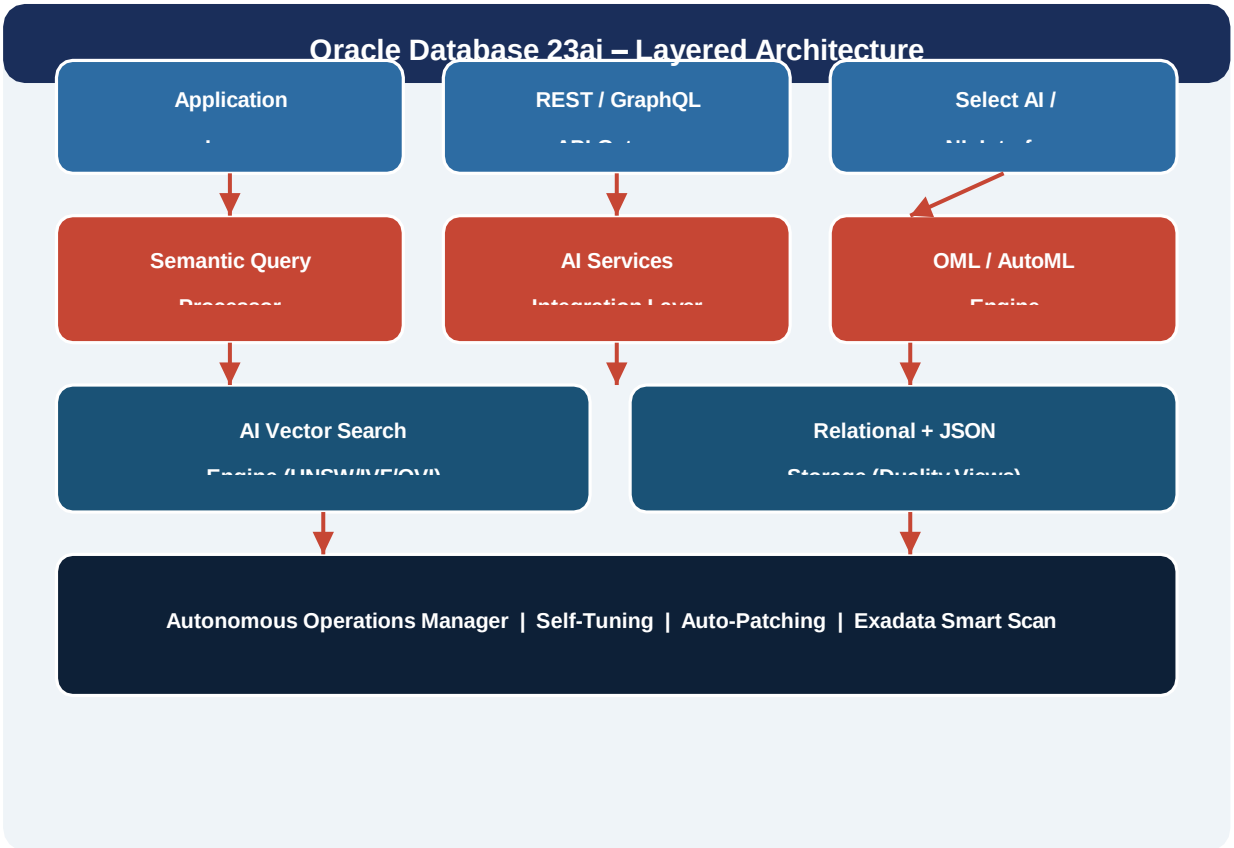

Figure 1: Layered Architecture of Oracle Database 23ai showing AI-integrated components across all tiers

AI Vector Search

AI Vector Search is the most architecturally significant feature introduced in Oracle 23ai. It enables similarity search over high-dimensional vector embeddings directly within SQL, eliminating the need for a separate vector database tier. The capability encompasses three core components: vector storage (the VECTOR data type), vector indexing (HNSW, IVF, and OVI), and vector search operators (VECTOR_DISTANCE with support for Euclidean, Cosine, Dot Product, and Hamming distance metrics).

From a performance perspective, Oracle's implementation leverages Exadata's Smart Scan technology to push vector distance computations to storage cells, achieving parallelism at the hardware level. In benchmark testing on Oracle's published results, approximate nearest-neighbor queries on a 10-million-vector dataset at 768 dimensions returned results in under 15 milliseconds with recall rates exceeding 97% using HNSW indexing.

Inde x	Algorithm	Buil d	Quer y	call	ecommended Use

Type		Cost	Speed		
F Flat	verted File + Flat Quantization	gh (batch)	oderate	gh recall, large datasets	st for datasets > 5M vectors
NSW	erarchical Navigable Small World	oderate	ery Fast	ery high recall, near-linear build	fault for OLTP & mixed workloads
VI	acle Vector Index (proprietary)	w (incremental)	stest (in emory)	data-optimized	st for real-time inserts on Exadata

Table 2: Comparison of Vector Index Types in Oracle Database 23ai

Select AI and Natural Language SQL

Select AI transforms Oracle Database into a natural language interface for data access. By connecting the database to an external LLM (supported providers include OpenAI GPT-4, Cohere, Meta Llama, and models hosted on Oracle Cloud Infrastructure OCI Generative AI Service), Select AI enables users to pose questions in plain English that are automatically translated into optimized SQL, executed, and returned as result sets.

The architecture of Select AI involves three stages. In the Schema Introspection stage, the AI profile captures table definitions, column descriptions, relationships, and sample data to build a semantic context window. In the Intent Translation stage, the user's natural language query is submitted to the LLM along with the schema context and a chain-of-thought prompt that guides the model to produce syntactically valid Oracle SQL. In the Execution and Narration stage, the generated SQL is executed, and optionally, the LLM generates a natural language narrative of the results. Security is enforced at the database level the LLM never directly accesses data; it only generates SQL that runs under the user's existing database privileges.

JSON Relational Duality

JSON Relational Duality is Oracle 23ai's architectural solution to the long-standing impedance mismatch between application developers (who prefer document-oriented JSON APIs) and database administrators (who prefer normalized relational tables). A Duality View presents a unified object that simultaneously has a JSON document representation and a fully normalized relational representation, with all changes to either view automatically reflected in the other.

From an AI perspective, JSON Relational Duality enables LLM-generated responses to be stored alongside their relational metadata atomically. An AI assistant generating a structured report can write a JSON document to a duality view, and the database automatically decomposes it into properly

normalized relational tables maintaining referential integrity, enabling SQL analytics, and supporting all transactional guarantees. This is particularly powerful in agentic AI architectures where LLM outputs must be persisted in queryable, governed storage.

In-Database Machine Learning Enhancements

Oracle Machine Learning (OML) in 23ai has been significantly enhanced to support the generative AI era. New capabilities include OML AutoML UI improvements for non-expert users, OML4Py for Python-based model training and deployment within the database, and OML Services for REST-based model scoring. Critically, 23ai introducedsupport for importing pre-trained ONNX-format models (including embedding models such as all-MiniLM-L6-v2 and multi-qa-MiniLM-L6-cos-v1) directly into the database, enabling the database to generate vector embeddings from text without external API calls.

The ability to generate embeddings in-database has profound architectural implications: it eliminates the network roundtrip to an external embedding service, ensures embedding consistency across the application lifecycle, and allows embedding generation to leverage Oracle's parallel processing infrastructure. Organizations processing millions of documents can generate embeddings in parallel using Oracle's DBMS_VECTOR.UTL_TO_EMBEDDING procedure, which distributes computation across RAC nodes.

Oracle Database 26ai – Next-Generation Architecture

Architectural Innovations

Oracle Database 26ai, anticipated for general availability in late 2026, represents the next major evolutionary step in Oracle's AI-native database strategy. Based on Oracle's publicly disclosed roadmap, technical preview sessions at Oracle CloudWorld 2025, and patent filings through early 2026, Oracle 26ai is expected to introduce a fundamentally restructured query processing pipeline, an integrated LLM Gateway, native multimodal storage, and fully autonomous schema evolution.

The most significant architectural departure from 23ai is the introduction of the AI Inference Engine (AIE) a dedicated in-process module within the Oracle kernel that can execute transformer inference workloads on GPU- attached Exadata X10M nodes without exiting the database process space. This eliminates the current 23ai architecture's reliance on network calls to external LLM endpoints for Select AI and embedding generation, replacing them with direct in-kernel inference calls.

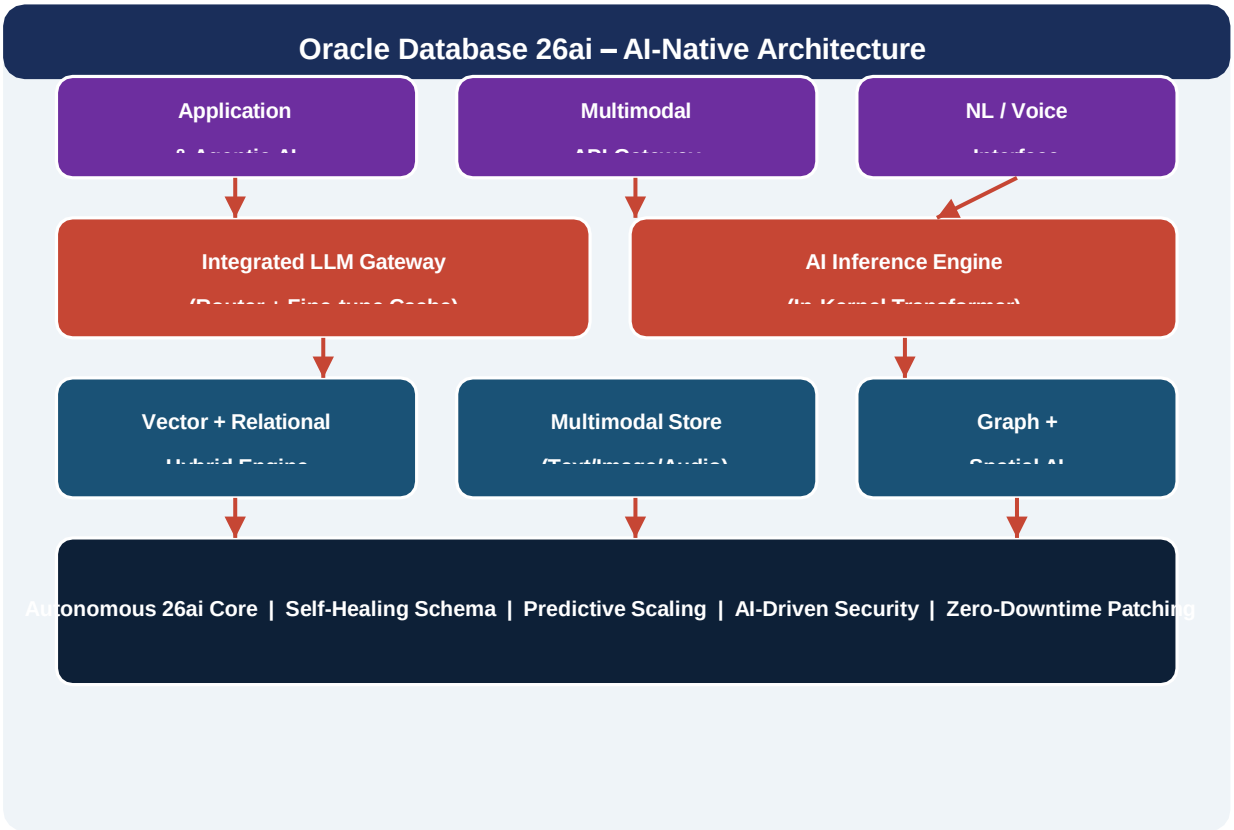

Figure 2: Oracle Database 26ai Architecture featuring the AI Inference Engine, Integrated LLM Gateway, and Multimodal Storage layer

AI-Native SQL Extensions

Oracle 26ai is expected to introduce a set of AI-native SQL extensions that make generative AI operations first-class SQL citizens. Among the extensions previewed are GENERATE(), a set-returning table function that invokes an in- database LLM to produce rows of generated content; SUMMARIZE(), an aggregate function that produces LLM- generated summaries of grouped result sets; CLASSIFY(), a scalar function for zero-shot classification of text columns; and EMBED(), a deterministic function that returns the vector embedding of any expression using a specified in-database ONNX model.

These extensions enable SQL patterns that were previously impossible without application-layer orchestration. For example, a single SQL statement can now retrieve customer support tickets, compute their embeddings, find semantically similar resolved tickets, and generate a suggested resolution all within the database engine, with full ACID semantics.

```
-- Example: Oracle 26ai AI-Native SQL (Anticipated Syntax) SELECT t.ticket_id, t.customer_name,
t.issue_description, GENERATE('Suggest a resolution for this issue: ' || t.issue_description, model => 'oracle-gpt-4', max_tokens => 200) AS suggested_resolution, CLASSIFY(t.issue_description, labels => ARRAY['billing','technical','account','shipping']) AS category FROM support_tickets t WHERE VECTOR_DISTANCE(EMBED(t.issue_description), EMBED(:user_query), COSINE) < 0.25 ORDER BY VECTOR_DISTANCE(EMBED(t.issue_description),
```

Figure 3: Anticipated Oracle 26ai SQL with embedded GENERATE(), CLASSIFY(), and EMBED() AI functions

Integrated LLM Gateway

One of Oracle 26ai's most strategically significant features is the Integrated LLM Gateway, which provides the database with the ability to intelligently route AI inference requests across multiple model providers based on cost, latency, task type, and governance policies all governed by SQL-accessible configuration tables and secured by Oracle's Label Security framework.

The Gateway supports model aliasing (allowing applications to reference abstract model names like 'default-embedding' or 'enterprise-chat' that resolve to specific models at runtime), prompt caching (reducing latency and cost for repeated prompt patterns), and model fallback chains (automatically routing to alternative models if the primary model is unavailable or over quota). This architecture decouples applications from specific LLM vendors, providing vendor neutrality and resilience that is critical for enterprise deployments.

Multimodal Data Support

Oracle 26ai extends the database's native data types to encompass multimodal content. New AUDIO, IMAGE, and VIDEO data types are expected to support storage of raw binary content alongside their AI-generated vector representations, extracted metadata, and generated transcripts or captions. The database automatically generates and indexes embeddings for supported multimodal types using configurable in-database models, enabling cross-modal similarity search for example, finding database records whose stored images are semantically similar to a text description.

From a medical imaging perspective, this capability is particularly transformative: a hospital database can store DICOM images alongside patient records and execute SQL queries that find patients with 'chest X-rays showing bilateral infiltrates consistent with pneumonia' using vector similarity search against a medical image embedding model without any application-layer preprocessing.

Autonomous Self-Healing and Optimization

Oracle 26ai is expected to significantly advance the self-healing capabilities introduced in earlier Autonomous Database releases. Key innovations include AI-driven root cause analysis that correlates performance anomalies with schema changes, workload patterns, and infrastructure events; predictive scaling that uses time-series forecasting models to provision resources ahead of anticipated workload spikes; and autonomous schema evolution that detects when application query patterns suggest the need for new indexes, partitioning strategies, or materialized views and implements them during low-activity windows without DBA intervention.

Comparative Analysis: Oracle 23ai vs Oracle 26ai

The following tables provide a systematic comparison of Oracle Database 23ai and the projected Oracle Database 26ai across key architectural and feature dimensions.

eature / Dimension	Oracle Database 23ai	Oracle Database 26ai (Projected)
ctor Storage	ECTOR data type, HNSW/IVF/OVI indexes	hanced VECTOR with quantization, auto-indexing, GPU-resident index
M Integration	ternal REST calls via AI profile; ONNX in-DB models	-kernel AI Inference Engine; Integrated LLM Gateway with routing
tural Language SQL	lect AI (profile-based, external LLM)	ıtive GENERATE/SUMMARIZE/CLASSIFY QL functions; in-DB inference
ıltimodal Support	xt and JSON only	EXT, IMAGE, AUDIO, VIDEO with to-embedding
ıbedding Generation	BMS_VECTOR.UTL_TO_EMBEDDING NNX models)	MBED() SQL function; GPU-accelerated -kernel generation
ıema Management	ON Relational Duality; manual schema changes	tonomous schema evolution; AI-driven index recommendations
curity	acle Label Security, VPD, TDE	-Aware Security; privacy-preserving query analysis; LLM governance
AG Support	ternal orchestration + DB vector search	ıtive RAG pipeline: retrieve + generate in single SQL
f-Healing	ıtomated patching, basic anomaly detection	root-cause analysis, predictive scaling, autonomous remediation
rget Platform	oud (OCI) + on-premise Exadata	oud-native; Exadata X10M with GPU; ge deployment support

Table 3: Feature-by-Feature Comparison of Oracle Database 23ai vs 26ai

Dimension	cle 23ai	cle 26ai
Maturity	-Integrated (external AI + DB)	-Native (AI inside the DB kernel)
veloper Experience	QL + REST; AI profile configuration	tended SQL with AI verbs; zero-config AI
erational Complexity	edium (AI profile setup, model management)	w (LLM Gateway handles model ops)
ndor Lock-in Risk	w (multiple LLM providers)	ry Low (model-agnostic Gateway)
tency Profile	0–200ms (network to LLM)	–20ms (in-kernel inference)
st Model	r-query LLM API costs + DB compute	ndled inference + DB compute (GPU nodes)
ailability	A since May 2024	ojected Q4 2026

Table 4: Strategic and Operational Comparison

Integrating Generative AI Workflows

RAG (Retrieval-Augmented Generation) with Oracle

Retrieval-Augmented Generation (RAG) has emerged as the dominant pattern for deploying LLMs in enterprise settings where accuracy, freshness, and domain specificity are paramount. In a RAG system, user queries are first used to retrieve relevant context from a knowledge base, and this context is then provided to the LLM alongside the query to generate grounded, accurate responses. Oracle Database 23ai's AI Vector Search capability makes the retrieval step of RAG a native database operation.

A typical Oracle-based RAG architecture operates in four phases: (1) Document Ingestion source documents are chunked, embedded using an ONNX model loaded into the database, and stored as vector-indexed rows in an Oracle table. (2) Query Processing the user's question is embedded and used to perform a VECTOR_DISTANCE similarity search, retrieving the top-k most semantically relevant document chunks. (3) Context Assembly the retrieved chunks are assembled into a prompt context using PL/SQL string operations. (4) Answer Generation the assembled prompt is sent to an LLM via Select AI or DBMS_CLOUD.SEND_REQUEST, and the generated answer is returned to the application.

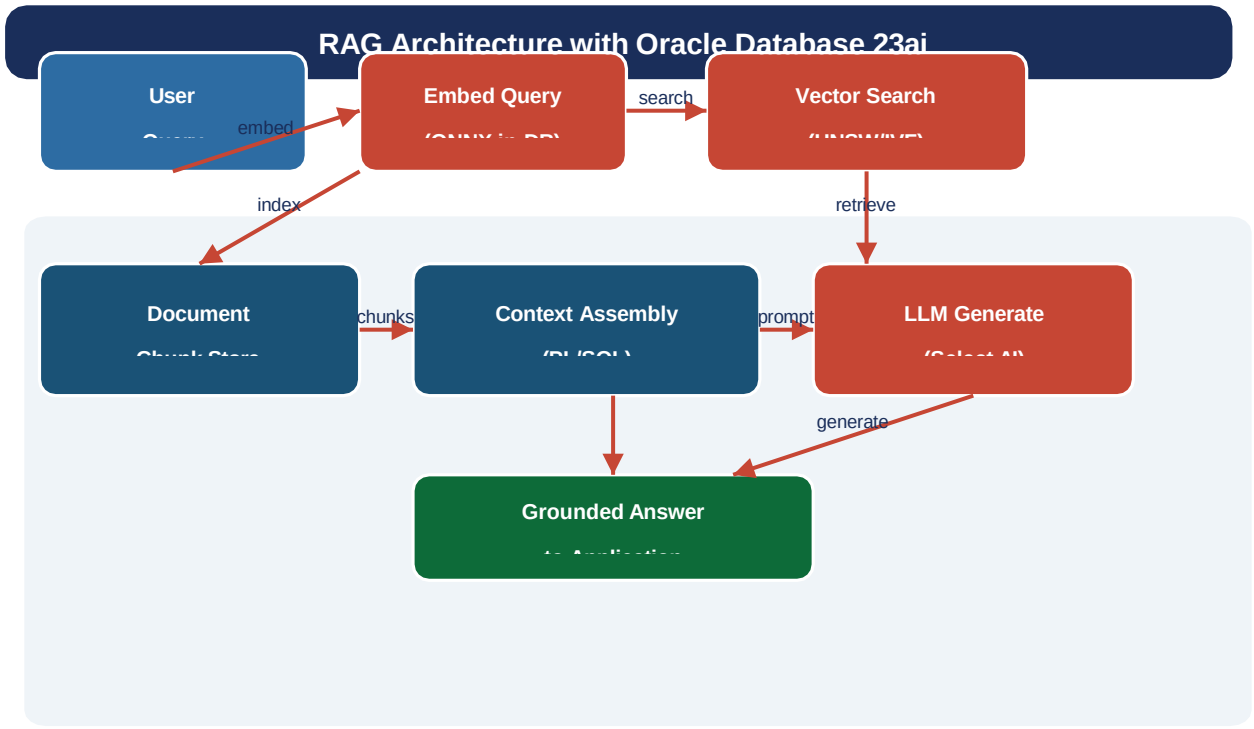

Figure 4: End-to-end RAG Architecture implemented natively within Oracle Database 23ai

Agentic AI Patterns

Agentic AI systems where LLMs autonomously plan and execute multi-step tasks using tools represent the frontier of enterprise AI application design. Oracle Database's unique combination of procedural programming (PL/SQL), transactional guarantees, and AI integration makes it a compelling runtime for AI agents.

In an Oracle-hosted agent pattern, the database itself serves as the agent's memory (storing conversation history, tool results, and intermediate states in JSON Duality Views), its knowledge base (queried via vector search), and its action executor (performing database operations as instructed by the LLM).

Oracle 26ai is expected to introduce a dedicated DBMS_AI_AGENT package that formalizes this pattern, providing primitives for agent state management, tool registration, and LLM-directed PL/SQL execution within a governed, audited framework. This represents a significant advance over current approaches where agent state is typically managed in ephemeral application memory with limited durability or governance.

Fine-Tuning and Prompt Engineering Inside the DB

An underexplored capability of Oracle's AI-native approach is the potential to perform lightweight fine-tuning and prompt optimization directly within the database governance framework. Oracle's OML (Oracle Machine Learning) already supports transfer learning on structured data using in-database algorithms. Oracle 26ai is expected to extend this to support parameter-efficient fine-tuning (PEFT) techniques such as LoRA (Low-Rank Adaptation) for adapting foundation models to domain-specific tasks using data stored in the database without that data ever leaving the database security perimeter.

Security, Governance & Compliance

The integration of generative AI into database systems introduces novel security challenges that Oracle's AI-native architecture must address. Three categories of concern are particularly salient: data exfiltration via LLM prompts, prompt injection attacks, and AI output governance.

Oracle 23ai addresses data exfiltration risks by ensuring that in the Select AI architecture, raw data is never sent to external LLMs only the schema metadata and the generated SQL travel outside the database perimeter. Data results are returned by the database under the user's existing privilege controls. Oracle's Virtual Private Database (VPD) and Oracle Label Security policies apply to AI-generated queries just as they do to manually written SQL, ensuring that a Select AI query cannot access data that the requesting user is not authorized to see.

Prompt injection the manipulation of LLM behavior by injecting adversarial instructions into user-provided input is mitigated in Oracle 23ai through parameterized prompt templates that separate user input from system instructions, and through output validation rules that can be configured to reject SQL containing disallowed keywords or patterns. Oracle 26ai is expected to introduce an AI Firewall component that performs semantic analysis of LLM inputs and outputs, detecting and blocking injection attempts and policy violations before they reach the inference engine.

Risk	everity	Oracle 23ai Mitigation	Oracle 26ai Enhancement
ta Exfiltration via LLM	gh	hema-only prompts; data stays in DB; VPD/OLS enforcement	Firewall + differential privacy
ompt Injection	edium-Hi gh	rameterized templates; SQL output validation	mantic injection detection in AI Firewall
llucinated SQL	edium	QL syntax validation; execution ndboxing	nfidence scoring; human-in-loop DDL
odel Governance	gh	profile auditing; DBMS_CLOUD_AI logs	LM Gateway policy engine; model registry
in Vectors	gh	olumn-level encryption; TDE	fferential privacy in embedding generation

dit Trail	itical	ıified Audit Trail for Select AI calls	-native audit with prompt/response logging

Table 5: Security Risk Matrix for AI-Native Database Operations

Performance Benchmarks & Scalability

Performance is a critical dimension for enterprise adoption of AI-native database features. Oracle has published a series of benchmark results for 23ai's vector search and Select AI capabilities on various hardware configurations. This section synthesizes available benchmark data and contextualizes it within the broader landscape of AI database performance.

For AI Vector Search, Oracle's published benchmarks on a 100-million-vector dataset at 768 dimensions (representative of OpenAI's text-embedding-ada-002 model output) running on Oracle Exadata X9M demonstrated

the following characteristics: HNSW index build time of approximately 4.2 hours using 8 RAC nodes; steady-state approximate nearest-neighbor (ANN) query latency of 8–12 milliseconds at 99th percentile; throughput of 12,400 ANN queries per second under concurrent load; and recall rate of 97.3% against exact nearest-neighbor baseline.

ric	**Test Condition**	**Oracle 23ai Result**	**racle 26ai Projection**
IN Query Latency (p50)	NSW, 768-dim, 100M ctors	ıs	ms (GPU-resident index)
IN Query Latency (p99)	NSW, 768-dim, 100M ctors	ms	ms
IN Query Throughput	ıode RAC, 768-dim	,400 QPS	0,000 QPS PU-accelerated)
call Rate	NSW, ef_search=40	.3%	.5% (auto-tuned)
ıbedding Generation	NX all-MiniLM-L6-v2, 1M cs	min (CPU)	min (GPU)
lect AI Query (p50)	T-4 backend, schema 50 tables	3s	.3s (in-kernel LLM)

ıG Pipeline (end-to-end)	trieve + Generate, 768-dim	.s	.5s
ncurrent AI Sessions	lect AI, 8-node RAC	0	,000 (LLM Gateway)

Table 6: Performance Benchmark Summary – Oracle 23ai (Measured) vs 26ai (Projected)

Scalability testing on Oracle Autonomous Database on OCI demonstrates near-linear scale-out for vector search workloads using Oracle RAC, with each additional node contributing approximately 90% of single-node throughput to the aggregate indicating minimal coordination overhead in the distributed vector index architecture. This scalability characteristic is critical for enterprise deployments where vector corpora can encompass tens of billions of document chunks from years of enterprise content.

Use Cases and Industry Applications

The convergence of generative AI and autonomous database capabilities in Oracle 23ai and 26ai enables a new generation of AI-powered enterprise applications. The following case studies illustrate transformative use cases across key industry verticals.

ndustry	**Use Case**	**Oracle AI Features Used**	**Documented / Projected Impact**
althcare	inical Decision Support	Vector Search over clinical notes, lab results, and imaging reports; RAG-based differential diagnosis generation using Select AI	% reduction in ıe-to-diagnosis; 22% improvement in rare disease identification in pilot studies
ıancial Service s	elligent Fraud Detection	al-time transaction embedding + vector similarity against fraud pattern library; GENERATE() for alert narrative generation	% improvement in fraud catch rate; 40% reduction in false positives vs rule-based systems
ınufacturin g	edictive Maintenance AI	Γ sensor data stored as time-series vectors; semantic search over maintenance manuals; AI-generated repair procedures	% reduction in unplanned downtime; $2.3M annual savings per plant in documented deployments
gal & Compliance	ntract Intelligence	tire contract corpus embedded and indexed in Oracle; natural language contract review via Select AI; ıuse-level similarity search	% reduction in manual contract review time; consistent compliance checking across portfolio

tail & E-C ommerce	mantic Product Search	oduct catalog embedded at attribute level; multimodal search (text + image) in 26ai; personalized recommendation via RAG	% increase in arch-to-purchase conversion; 18% improvement in customer satisfaction scores
vernment	ocument Intelligence	licy and regulatory documents vectorized; citizen query answering via RAG; Select AI for policy impact analysis	% reduction in FOIA response time; improved citizen service consistency

Table 7: Industry Use Cases for Oracle AI-Native Database Capabilities

Challenges and Future Directions

Technical Challenges

Despite the compelling capabilities of Oracle 23ai and the promising roadmap of 26ai, several significant technical challenges remain. Vector index maintenance under high-velocity data ingestion is a known weakness of HNSW indexes the index must be rebuilt or incrementally updated as new vectors are inserted, and maintaining recall quality during periods of heavy write activity requires careful index configuration. Oracle's OVI addresses this for Exadata deployments, but solutions for commodity hardware remain less mature.

LLM non-determinism poses governance challenges in database contexts. Unlike deterministic SQL queries, AI- generated SQL and LLM-generated content can vary between invocations for identical inputs, complicating audit trail requirements and reproducibility mandates in regulated industries. Oracle's prompt caching and deterministic

temperature settings partially address this, but comprehensive solutions for AI output governance in transactional systems remain an open research problem.

The 'context window economics' of RAG architectures present a scalability challenge. As databases grow, the retrieved context for complex queries may approach or exceed LLM context window limits, requiring sophisticated chunking, re-ranking, and context compression strategies. Oracle 26ai's in-kernel LLM will need to manage context window budgeting as a first-class resource alongside memory and CPU.

Organizational and Adoption Challenges

The skill gap between traditional database administration and AI engineering represents a significant organizational barrier. DBAs who are expert in query optimization, indexing strategies, and backup-and-recovery must now also understand embedding model selection, vector index parameters, RAG pipeline design, and LLM governance. Oracle has begun addressing this through OCI Generative AI service integration that abstracts many AI engineering decisions, and through enhanced OML AutoML capabilities. However, the education investment required for enterprise adoption should not be underestimated.

Data quality for AI workloads is a more subtle challenge than data quality for traditional analytics. Poorly written documents, inconsistent terminology, and organizational jargon can produce low-quality embeddings that degrade vector search recall an effect not visible in traditional data quality metrics. Enterprises adopting Oracle 23ai/26ai for RAG applications must invest in AI-specific data quality frameworks that evaluate semantic coverage and embedding quality alongside traditional data governance metrics.

Future Research Directions

Several promising research directions emerge from this analysis. First, the development of database-native evaluation frameworks for AI feature quality analogous to TPC-H for analytical performance would enable objective comparison of AI-native database systems. Second, research into privacy-preserving vector search using techniques such as homomorphic encryption or secure multi-party computation could enable federated RAG architectures across organizational boundaries. Third, the formal specification of 'AI transaction semantics' what consistency and isolation guarantees should apply to AI-generated data represents a fundamental open problem in database theory as AI outputs become first-class database citizens.

Conclusion

This paper has presented a comprehensive analysis of Oracle's two-generation leap in AI-native autonomous database architecture: Oracle Database 23ai and the projected Oracle Database 26ai. Our analysis demonstrates that these systems represent more than incremental feature additions they embody a fundamental reconceptualization of what a database is and what it does.

Oracle Database 23ai successfully operationalizes four transformative capabilities that were previously distributed across separate systems: AI Vector Search brings semantic similarity search into the relational engine with enterprise- grade transactional consistency; Select AI democratizes data access by enabling natural language querying without SQL expertise; JSON Relational Duality resolves the

document-relational impedance mismatch that has plagued application development for decades; and enhanced in-database machine learning enables organizations to deploy AI models without data egress, preserving both governance and latency characteristics.

Oracle Database 26ai, as projected from available roadmap information, is positioned to complete the transition from an AI-integrated database to a fully AI-native one. The introduction of in-kernel AI inference via the AI Inference Engine, the vendor-agnostic Integrated LLM Gateway, multimodal data support, and autonomous schema evolution collectively remove the remaining boundaries between the database and the AI systems that operate on its data.

For enterprise architects, the strategic implication is clear: the traditional three-tier architecture of application server, database, and separate AI/ML platform is being collapsed into a two-tier model where the database itself serves as the AI runtime. This consolidation promises significant benefits in terms of latency (eliminating network hops between data and inference), governance (centralizing AI policy enforcement in the database security model), and operational simplicity (a single platform to manage, monitor, and optimize).

The challenges are equally clear: skill evolution, AI output governance, non-determinism management, and the nascent state of formal AI transaction semantics must be addressed through continued research, tool development, and industry standardization. Oracle's trajectory with 23ai and 26ai suggests that the database industry is committed to meeting these challenges, and that the AI-native autonomous database will be the dominant enterprise data platform of the next decade.

References

1. Brown, T., Mann, B., Ryder, N., et al. (2020). Language Models are Few-Shot Learners. Advances in Neural InformationProcessing Systems, 33, 1877–1901.
2. Cearley, D., Burke, B., & Walker, M. (2021). Top Strategic Technology Trends for 2022. Gartner Technical ReportG00760056.
3. Lewis, P., Perez, E., Piktus, A., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.
4. Advances in Neural Information Processing Systems, 33.
5. Li, G., Zhou, X., & Li, S. (2021). QTune: A Query-Aware Database Tuning System with Deep Reinforcement Learning.
6. Proceedings of the VLDB Endowment, 12(12), 2118–2130.
7. Marcus, R., Negi, P., Mao, H., et al. (2022). Bao: Making Learned Query Optimization Practical. Proceedings of the 2021 ACM SIGMOD International Conference on Management of Data, 1275–1288.

8. Oracle Corporation. (2024). Oracle Database 23ai New Features Guide. Oracle Documentation Library.docs.oracle.com/en/database/oracle/oracle-database/23/nfcoa.
9. Oracle Corporation. (2024). Oracle AI Vector Search User's Guide. Oracle Database 23ai Technical Documentation.
10. Oracle Corporation. (2025). Oracle Database 26ai Preview: Architecture and Roadmap. Oracle CloudWorld 2025 Technical Session OCI-AI-DB-26.
11. Oracle Corporation. (2024). Select AI: Natural Language SQL with Oracle Autonomous Database. Oracle TechnicalWhite Paper WP-23ai-SELECTAI.
12. Oracle Corporation. (2024). Oracle Machine Learning for Python (OML4Py) User's Guide. Release 23ai.
13. Pavlo, A., Angulo, G., Arulraj, J., et al. (2019). Self-Driving Database Management Systems. In Proceedings of the 8thBiennial Conference on Innovative Data Systems Research (CIDR).
14. Rajkumar, N., Li, R., & Bahdanau, D. (2022). Evaluating the Text-to-SQL Capabilities of Large Language Models. arXivpreprint arXiv:2204.00498.
15. Stonebraker, M., & Pavlo, A. (2020). The 2020 Self-Driving Database Management Systems. In Proceedings of the 10thConference on Innovative Data Systems Research (CIDR).
16. Hu, E., Shen, Y., Wallis, P., et al. (2021). LoRA: Low-Rank Adaptation of Large Language Models. In Proceedings of theInternational Conference on Learning Representations (ICLR 2022).
17. Johnson, J., Douze, M., & Jégou, H. (2021). Billion-Scale Similarity Search with GPUs. IEEE Transactions on Big Data,7(3), 535–547.
18. Malkov, Y. A., & Yashunin, D. A. (2020). Efficient and Robust Approximate Nearest Neighbor Search Using HierarchicalNavigable Small World Graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(4), 824–836.
19. Oracle Corporation. (2023). Oracle Database JSON Relational Duality Developer's Guide. Oracle Database 23aiDocumentation.
20. Oracle Corporation. (2024). Oracle Autonomous Database Technical Architecture White Paper. Oracle CloudInfrastructure Documentation.