ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 14, Issue 2, pp: (18-22), Month: April – June 2026, Available at: www.researchpublish.com

Evolution of Database Technology - Vector Database Technology for AI Systems

Emad Al-Mousa

Saudi Aramco, Dhahran, Upstream Digital Center, Saudi Arabia

DOI: https://doi.org/10.5281/zenodo.19592783

Published Date: 15-April-2026

Abstract: **Database technology has undergone a significant evolution, transitioning from traditional relational systems to NoSQL and Big Data architectures. In the current era of Artificial Intelligence (AI), vector technology has emerged as a transformative feature within database platforms. Vector databases are essential for modern AI applications, particularly for Large Language Models (LLMs), as they provide the necessary infrastructure to handle high-dimensional data at scale.**

Keywords: **Cloud, Vector, Database,Vector Database, AI, Large Language Model, Vector Embeddings, database systems and AI,RAG.**

I. INTRODUCTION

Vector database technology has become the essential backbone for Large Language Models (LLMs), primarily supporting semantic search and Retrieval-Augmented Generation (RAG). A vector database is a specialized system designed to store, manage, and index vector embeddings—high-dimensional numerical representations of unstructured data such as text, images, or audio (IBM, 2024). Unlike traditional databases that rely on exact keyword matches, vector embeddings facilitate semantic similarity searches. This allows systems to identify data points that are conceptually or contextually related, a capability that is critical for modern Artificial Intelligence (AI) and Machine Learning (ML) workflows.

II. BACKGROUND

The landscape of data management has shifted from traditional Relational Database Management Systems (RDBMS) to diverse data models necessitated by emerging technological demands. This includes NoSQL formats such as JSON (Document), Graph, and Time-Series. To meet the scalability requirements of "Big Data," platforms were introduced to manage the three Vs: velocity, volume, and variety (PingCAP, 2023).

Currently, vector database technology represents a "game-changer" in the AI adoption cycle. Historically, data mining focused on discovering patterns and correlations within structured datasets. However, data representation is shifting from purely relational to contextual. This transition is vital for generating meaningful insights from the vast amounts of unstructured data—estimated to comprise 80% of all organizational data—that were previously difficult to analyze (Elastic, 2024).

III. VECTOR DATA TYPE OVERVIEW

Vector data is a numeric representation of information, typically stored as an array of floating-point numbers. These arrays represent "embeddings" created by passing unstructured data through neural networks.

The primary objective of a vector database is similarity search: finding vectors that are mathematically "close" to a query vector in a multi-dimensional space. Key use cases include:

- Generative AI: Powering chatbots and virtual assistants.
- Semantic & Multi-Modal Search: Finding content across different media types.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 14, Issue 2, pp: (18-22), Month: April – June 2026, Available at: www.researchpublish.com

- Computer Vision: Image and video recognition.
- Security: Anomaly and fraud detection.

Pre-trained embedding models from providers like OpenAI, Google, and Meta, or open-source models like all-MiniLM-L6-v2, allow developers to map sentences into dense 384-dimensional vector spaces for clustering and retrieval tasks.

IV. TOP VECTOR DATABASES

The popularity of vector databases is driven by the performance requirements of Generative AI. Gartner predicts that by 2026, over 30% of enterprises will adopt vector databases to ground their foundation models with proprietary business data (fierce-network,2024).

The market is divided into three main categories:

- Purpose-Built Vector Databases: Engineered specifically for high-dimensional data (e.g., Milvus, Pinecone, Chroma) (Instaclustr,2023).
- NoSQL Extensions: Popular systems like MongoDB, Redis, and Cassandra now offer built-in vector search.
- Relational Integration: PostgreSQL utilizes the pgvector extension, while Oracle 23ai (rebranded to 26ai) and SQL Server 2025 have introduced native vector capabilities to support advanced analytical workloads (Oracle, 2024).

V. VECTOR SIMILARITY SEARCH OVERVIEW

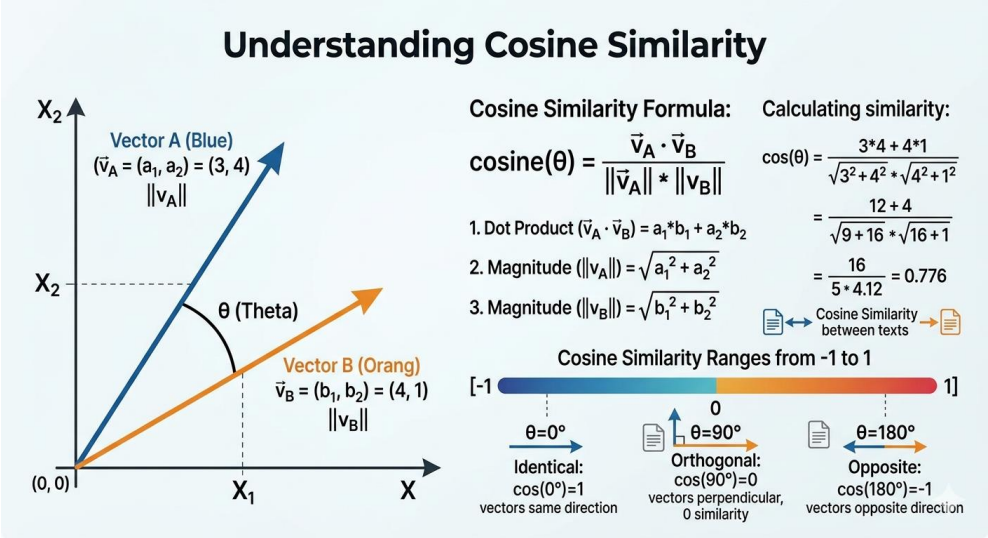
Similarity search uses distance metrics to identify how closely related two data points are. In a database like Oracle 26ai, the VECTOR_DISTANCE function compares vectors based on a chosen mathematical model.

The Vector Pipeline:

- Embedding: Data is converted into high-dimensional vectors.
- Indexing: Specialized algorithms (like HNSW or IVF) speed up the search process.
- Searching: A "Nearest Neighbor" search is performed using a distance algorithm.
- Retrieval: The most relevant vectors are returned based on the query input.

Common Algorithms:

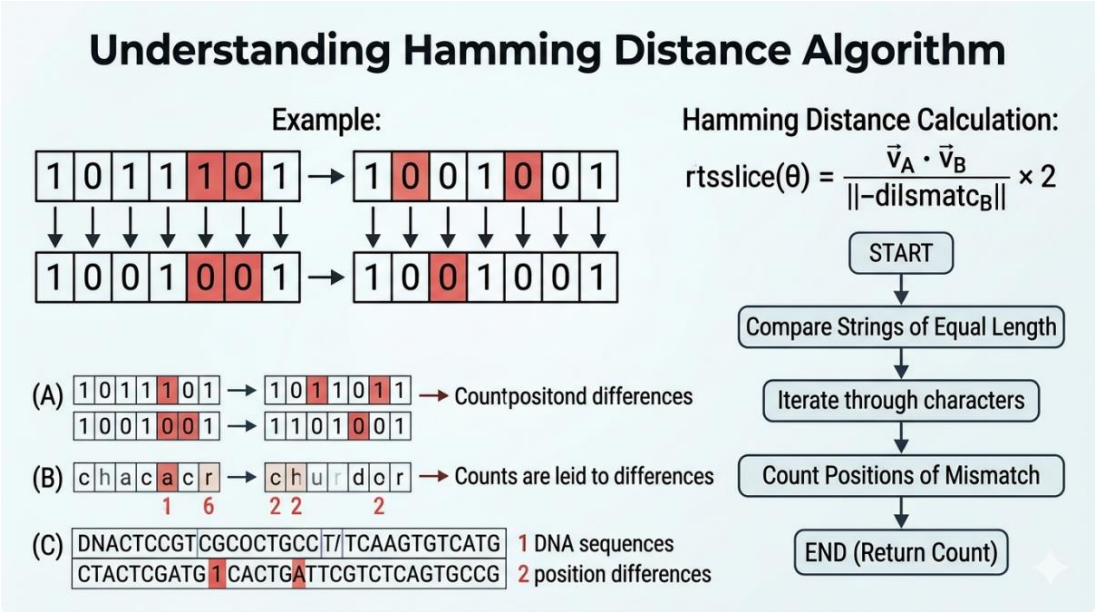
- Cosine Similarity: Measures the cosine of the angle between two vectors. It is widely used in NLP because it focuses on the orientation (context) rather than the magnitude of the vectors.



Note. Image generated by Gemini 3 Flash, April 13, 2026, from the prompt: "Cosine Distance algorithm." https://gemini.google.com/

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 14, Issue 2, pp: (18-22), Month: April – June 2026, Available at: www.researchpublish.com

- Hamming Distance: Measures the number of positions at which the corresponding symbols are different. This is often used for binary vectors and error detection.



Note. Image generated by Gemini 3 Flash, April 13, 2026, from the prompt: "Diagram of Hamming Distance algorithm." https://gemini.google.com/

VI. SIMILARITY SEARCH USE-CASE

In the presented case I am using a simulation of a "movie review" of multiple Tom Hanks movies. This simulation was implemented against Oracle 26ai release database system.

First, I will create table and will call it "MOVIE_REVIEWS_FOR_TOM_HANKS" and insert a dummy data for multiple reviewers (end-users) using SQL Language , and using Oracle SQL Developer Tool:

SQL> create table MOVIE_REVIEWS_FOR_TOM_HANKS (reviewer varchar2(20), toy_story int, forrest_gump int, saving_private_ryaan int , v vector);

Then, I will insert data to populate the table:

SQL> INSERT INTO movie_reviews_for_tom_hanks VALUES ('Emad',8,9,7,TO_VECTOR('[8.0, 9.0, 7.0]'));

SQL> INSERT INTO movie_reviews_for_tom_hanks VALUES ('Ali',8,8,6, TO_VECTOR('[8.0, 8.0, 6.0]'));

SQL> INSERT INTO movie_reviews_for_tom_hanks VALUES ('Sami',6,7,9, TO_VECTOR('[6.0, 7.0, 9.0]'));

Checking the table content by executing a full SQL select statement against the table:

```
select * from movie_reviews_for_tom_hanks;
```

Query Result

SQL | All Rows Fetched: 3 in 0.384 seconds

	REVIEWER	TOY_STORY	FORREST_GUMP	SAVING_PRIVATE_RYAN	V
1	Emad	8	9	7	[8.0E+000,9.0E+000,7.0E+000]
2	Ali	8	8	6	[8.0E+000,8.0E+000,6.0E+000]
3	Sami	6	7	9	[6.0E+000,7.0E+000,9.0E+000]

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 14, Issue 2, pp: (18-22), Month: April – June 2026, Available at: www.researchpublish.com

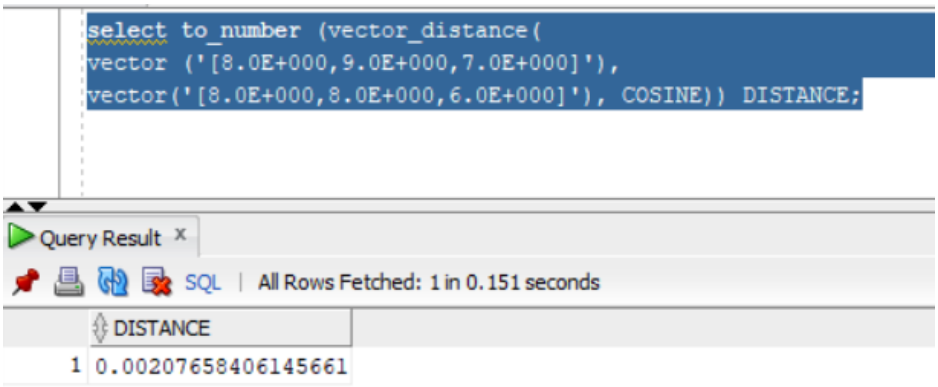
Let us now use "cosine" similarity search to compare Ali and Sami preferences against Emad and checking which one of them is the closest to Emad's preference:

Compare Emad Ratings With Ali :

select to_number (vector_distance(

vector ('[8.0E+000,9.0E+000,7.0E+000]'),

vector('[8.0E+000,8.0E+000,6.0E+000]'), COSINE)) DISTANCE;

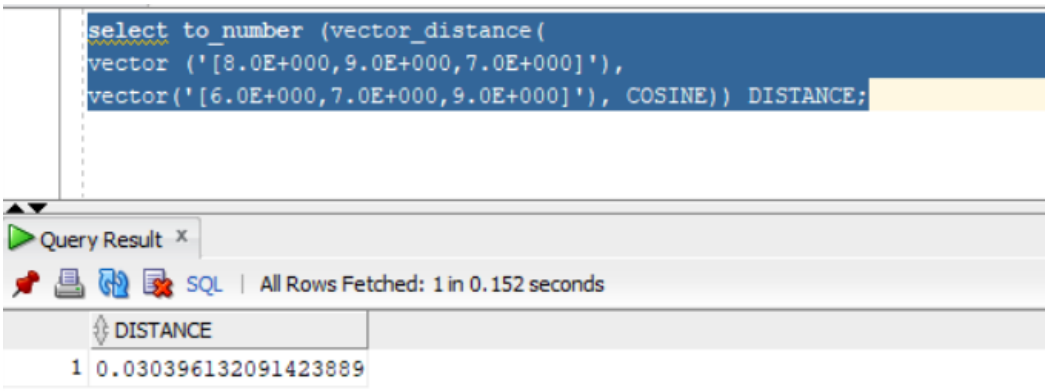


Compare Emad Ratings With Sami:

select to_number (vector_distance(

vector ('[8.0E+000,9.0E+000,7.0E+000]'),

vector('[6.0E+000,7.0E+000,9.0E+000]'), COSINE)) DISTANCE;



Results: The calculation yielded a lower distance score for the Emad-Ali comparison, confirming that Ali's movie preferences are semantically closer to Emad's.

VII. CONCLUSION

In the rapidly evolving landscape of Generative AI, vector database technology has transitioned from a specialized tool to a foundational enterprise requirement. By enabling efficient similarity searches and high-dimensional indexing, these databases address critical challenges regarding data latency and factual accuracy.

Key Pillars of Vector Integration:

- Semantic Precision: Enables systems to understand intent rather than just keywords.
- Scalable Memory: Serves as "long-term memory" for AI, allowing retrieval from massive datasets.
- Reduced Hallucination: Provides the structure for RAG, ensuring AI outputs are grounded in verified organizational data.

REFERENCES

[1] Elastic. (2024). What is a Vector Database? Retrieved from https://www.elastic.co/what-is-vector-database

[2] IBM. (2024). Vector Databases Explained. Retrieved from https://www.ibm.com/think/topics/vector-database

[3] Instaclustr.(2023). Best open source vector database software: Top 8 in 2026. Retrieved from https://www.instaclustr.com/education/vector-database/best-open-source-vector-database-software-top-8-in-2026/

[4] Oracle.(2024). AI Vector Search User's Guide. Retrieved from https://docs.oracle.com/en/database/oracle/oracle-database/23/vecse/overview-ai-vector-search.html

[5] PingCAP. (2023). Vector Stores vs. Traditional Databases. Retrieved from https://www.pingcap.com/article/vector-stores-vs-traditional-databases/

[6] fierce-network.(2024). Here's why vector databases are a big deal for AI. Retrieved from https://www.fierce-network.com/cloud/heres-why-vector-databases-are-big-deal-ai#:~:text=But%20there%20are%20some%20interesting,on%20capability,%E2%80%9D%20Chandrasekaran%20said.