NAND RESEARCH

Research Report:

Accelerating Enterprise AI with Oracle Database Vector Search

Steve McDowell, Chief Analyst

May 2024

This report was commissioned by Oracle Corporation

ACCELERATING ENTERPRISE AI WITH ORACLE DATABASE VECTOR SEARCH

Vector search is a powerful computing technique that performs searches based on the similarity of data points represented as vectors in a multidimensional space. Unlike traditional search methods that rely on keyword matching or exact data matches, vector search allows for identifying items based on contextual similarity. This approach is beneficial for handling unstructured data, such as text, images, audio, and video, enabling more nuanced and intelligent search capabilities.

Beyond its ability to expand the value of data with contextual search, vector search also lies at the heart of tailoring generative AI and large language models (LLMs) to the enterprise's specific needs.

Oracle introduced full support for vectors, including vector search, in its Oracle Database 23ai. Known as AI Vector Search, this capability represents a significant advancement in how databases can store, index, and search data semantically.

In addition to making this capability available in any Oracle Database 23ai instance, Oracle Exadata provides a scalable and optimized environment for running AI Vector Search and all other Oracle Database workloads. Built using high-performance AMD EPYC™ processors, Oracle Exadata is available in Oracle Cloud Infrastructure, with Oracle Database@Azure as the Exadata Cloud@Customer hybrid cloud platform and as a traditional on-premises database platform.

THE POWERFUL BUSINESS VALUE OF VECTOR SEARCH

Vector search brings a transformative approach to searching and analyzing large volumes of data. The technology changes how data is retrieved by examining its semantic content rather than relying solely on traditional keyword matches.

When combined with embedding models that create numerical representations (vectors) of underlying data, vector search helps identify the context and meaning behind queries and documents, providing search results that are semantically related to the query. This semantic understanding of the query's intent and the source information being searched goes beyond the limitations of keyword-based searches that often miss relevant results due to synonymy (different words with similar meanings) and polysemy (words with multiple meanings). This understanding significantly improves the accuracy of search results by returning content that is more aligned with the user's intent.

The technology isn't limited to text data. It applies to a wide range of data types, including images, audio, and video. Vector search opens possibilities for cross-modal searches, such as using an image to find related text documents or vice versa.

Using vector search can bring new business value to a range of applications:

- **Semantic Search**: Vector search excels in understanding the meaning behind queries and documents, enabling it to return contextually relevant results to the search intent, even if the exact words are not used.
- **Recommendation Systems**: By characterizing user behavior and item characteristics as vectors and then using similarity search and content-based filtering, vector search helps improve the quality and relevance of results provided by recommendation systems across industries, from retail to health care.
- **Image and Video Retrieval**: Vector search can find pictures or videos visually similar to a query image or video, supporting applications such as reverse image search and content-based media retrieval.
- **Natural Language Processing (NLP)**: In NLP tasks, vector search can help provide domain-specific context that LLMs trained on more general information lack. This can enhance information retrieval and knowledge discovery.
- **Retrieval-augmented generation** (RAG) combines the strengths of retrieval-based and generative AI models for natural language processing tasks. RAG relies heavily on vector search to identify relevant information quickly and accurately from files or database tables and augment the original prompts.

Vector search empowers organizations to harness their data's full potential, enhancing user experiences and driving increased operational efficiencies. Its ability to understand and process data semantically is a generational leap forward from traditional search methodologies, paving the way for more intelligent and adaptable information retrieval systems.

At the same time, fully realizing the benefits of vector search requires a database solution that supports the technology. This is where Oracle, which has enabled enterprises to recognize the value of their data for over forty years, enters the picture.

ENABLING GENERATIVE AI WITH VECTOR SEARCH

Large language models have quickly emerged as one of the most potentially transformative technologies in recent memory. LLMs promise to change how enterprises do business, from customer engagement to managing business processes.

One of the biggest challenges for enterprises adopting LLMs is what AI researchers call "hallucinations." The term refers to inaccuracies in responses that large language models can return, most often caused by the LLM not having the proper context or enterprise-specific data to return an accurate answer. This is where RAG and vector search come into the picture.

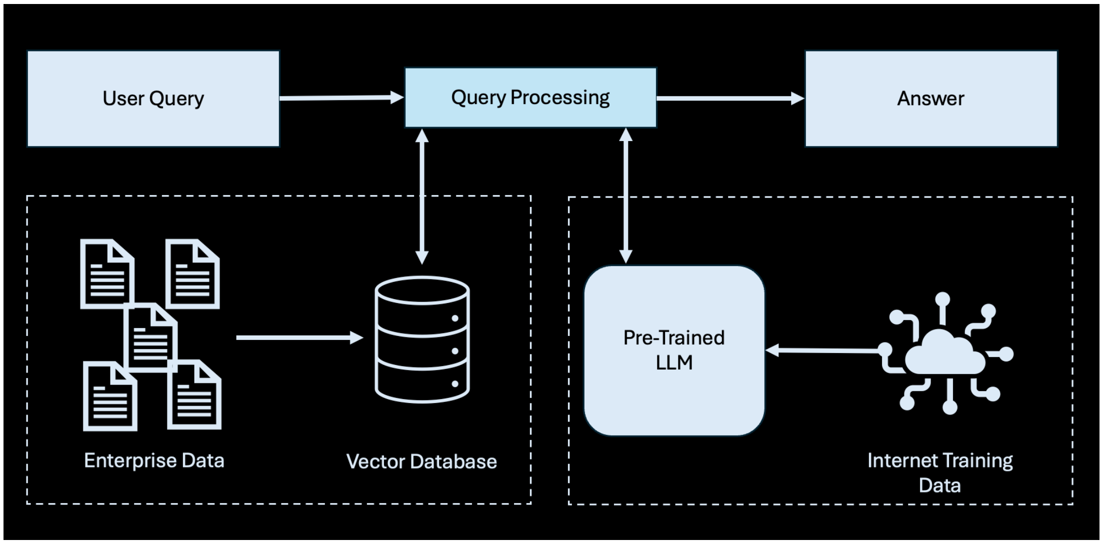


FIGURE 1: SIMPLIFIED RAG WORKFLOW

Combining the strengths of retrieval-based and generative models for natural language processing tasks, RAG relies on its ability to identify relevant information quickly and accurately from private enterprise databases. Vector search significantly enhances RAG by providing an efficient and effective method for the search and retrieval phase of the RAG process.

Vector search offers substantial benefits in tuning large language models with RAG:

- **Improved Relevance and Precision in Retrieval:** Vector search enables the RAG service to find the most contextually relevant information for any query. By converting text into high-dimensional vector spaces, vector search allows for comparing semantic similarities rather than just syntactic matches. This means the retrieved documents or data are more likely to be semantically related to the query, thus providing a better foundation for the generation step.
- **Enhanced Efficiency at Scale:** Vector search algorithms can handle massive datasets efficiently, especially those optimized for approximate nearest neighbor (ANN) searches. This scalability ensures that RAG services can analyze large bodies of information without significant performance degradation, which is crucial for real-time applications like chatbots, question-answering systems, e-commerce recommendation systems, and content generation.
- **Dynamic Contextualization:** The RAG service can dynamically contextualize queries by retrieving the most relevant information through vector search. This capability is particularly valuable when the context may shift rapidly, or the data is continually updated, such as news articles or trending social media content.
- **Richer Information Source for Generation:** Using vector search before prompts are sent to LLMs provides the generative component of RAG with a richer and more nuanced set of information. This enriched input enables the generation of more accurate, detailed, and contextually appropriate responses or content, leveraging the depth of semantic understanding captured in the vector representations.

- **Controllability and Customization:** Vector search allows for greater control over the retrieval process, enabling customization of the search space or applying filters based on specific needs or constraints. This flexibility lets the augmentation portion of the RAG process be finely tuned to support the desired results from the generative model, whether maintaining a particular tone or style or focusing on specific content types.

While this may seem complex, vector search simplifies the use of generative AI by allowing users to customize results while still using standard pre-trained models. Organizations no longer need data scientists on the payroll since traditional business units can implement AI-driven solutions. AI vector search democratizes generative AI.

VECTOR SEARCH BELONGS IN THE ENTERPRISE DATABASE

In the early days of generative AI, the industry looked towards specialized databases that focused only on supporting vector processing. While these point solutions provided a stop-gap solution for enterprises exploring the use of AI, extracting the full value of robust new AI-driven solutions requires a database that can handle the multi-data, multi-model converged needs of the enterprise. This is where Oracle steps in.

Oracle Database is already the world's most popular enterprise database[1], so adding vector data types, vector indexing, and vector search makes using these capabilities with existing enterprise data easier, more efficient, and more cost-effective.

In most cases, using a separate vector database to perform a similarity search is the most inefficient path since organizations must export business data, generate vector embeddings, and insert those vectors into a stand-alone database. Such a workflow means that the state of vector databases can be hours behind that of the business data driving it. Furthermore, IT resources must be expended on managing and securing separate database instances and infrastructure. It is hardly the efficiency level that most organizations are trying to achieve.

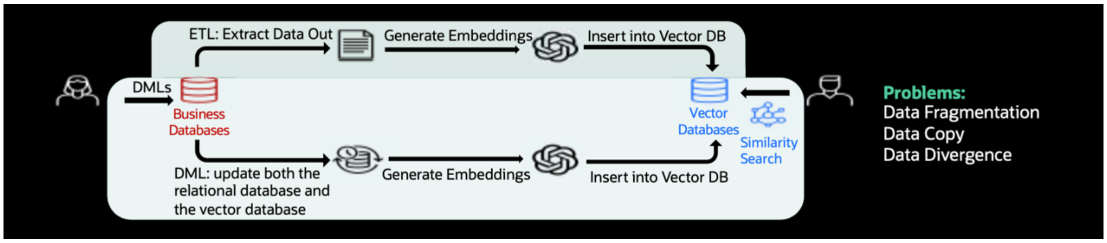
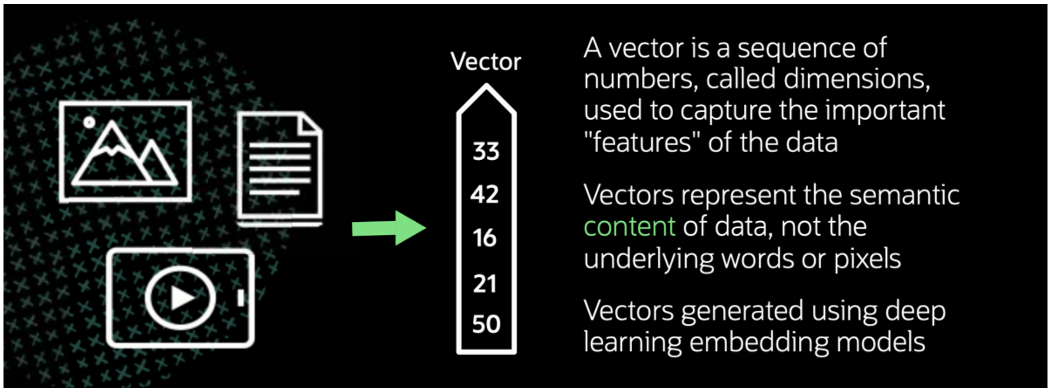


FIGURE 2: THE WORKFLOW REQUIRED TO UPDATE SEPARATE VECTOR DATABASES

[1] https://db-engines.com/en/ranking

UNDERSTANDING VECTOR SEARCH

Everything starts with storing semantic content as vectors, but what is a vector? Simply stated, a vector is a sequence of numbers representing a data point in a high-dimensional space. Each vector element (or dimension) captures some data feature, such as the relationship between words in a text document or the shape of various forms in an image. Machine learning models, especially neural networks, are often used to generate these vectors, transforming raw data into a format that captures its essential characteristics.



A vector is a sequence of numbers, called dimensions, used to capture the important "features" of the data

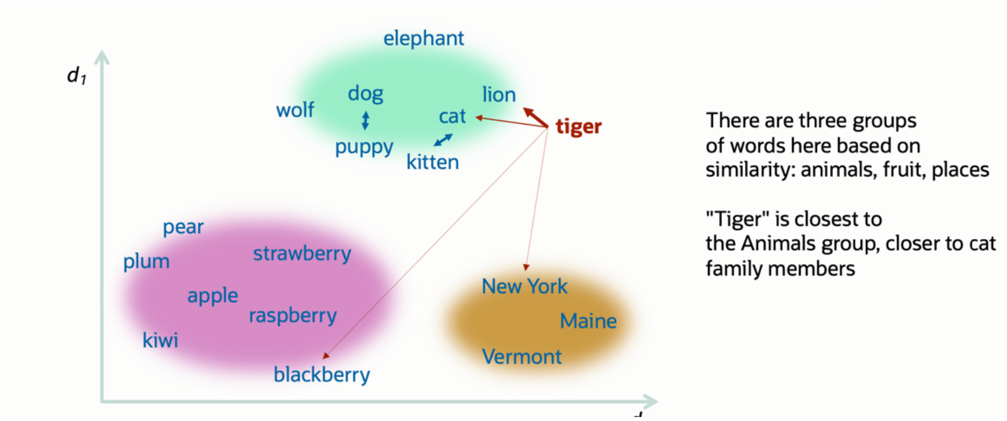
Vectors represent the semantic content of data, not the underlying words or pixels

Vectors generated using deep learning embedding models

Figure 3: What are vectors?

Vector search relies on mathematical measures of similarity (or distance) between vectors to determine how closely related two pieces of data are. Common measures include cosine similarity, Euclidean distance, and Manhattan distance. Vector search identifies and returns the most relevant results by calculating the similarity between a query vector and the vectors in the database.

Here's an illustrative example:



There are three groups of words here based on similarity: animals, fruit, places

"Tiger" is closest to the Animals group, closer to cat family members

Figure 4: Word relatedness in two dimensions

Neural Net driven vector embeddings create representations with much higher levels of dimensionality, but the concept remains the same.

VECTOR SEARCH IN ORACLE DATABASE 23AI

In its Oracle Database 23ai release, Oracle added several new capabilities that work together to support a broad range of AI functions, including support for vector search:

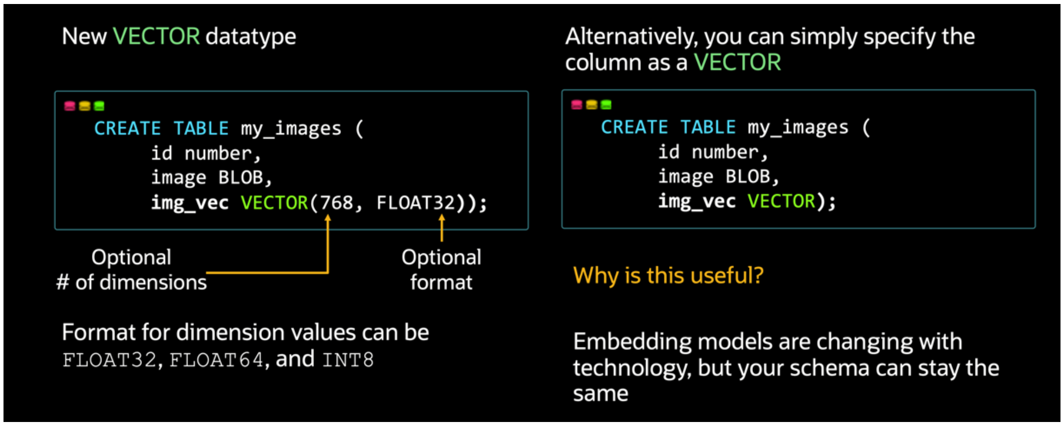
- **VECTOR data type**
- **SQL operators and functions**
- **Vector Indexes**

Oracle's new VECTOR datatype allows semantic content to be stored as vector encodings in the database, by itself in a separate database or alongside the data the vector represents in an existing database. New SQL syntax and functions allow users and applications to manage vector data and perform similarity searches easily. The vector indexes allow balancing performance and accuracy to meet the application's needs.

An important consideration for many customers is the complexity of the data flows in their overall system. As shown earlier in Figure 1, data flows for updating single-purpose vector databases can be complex and time-consuming. The AI Vector Search capability in Oracle Database 23ai simplifies these workflows by enabling everything from the creation of vector embeddings to the storage of vector data, searching it for similarities, and ultimately retrieving the data the vector points to take place inside the database.

NEW VECTOR DATATYPE

Anyone used to working with SQL will be familiar with using the vector datatype. This example shows how one could create a table with images stored as BLOBs and their vector representations stored along with them.



New VECTOR datatype

```
CREATE TABLE my_images (
      id number,
      image BLOB,
      img_vec VECTOR(768, FLOAT32));
```

Optional
# of dimensions

Optional
format

Format for dimension values can be FLOAT32, FLOAT64, and INT8

Alternatively, you can simply specify the column as a VECTOR

```
CREATE TABLE my_images (
      id number,
      image BLOB,
      img_vec VECTOR);
```

Why is this useful?

Embedding models are changing with technology, but your schema can stay the same

FIGURE 5: NEW VECTOR OBJECT CREATION IN ORACLE DATABASE 23AI

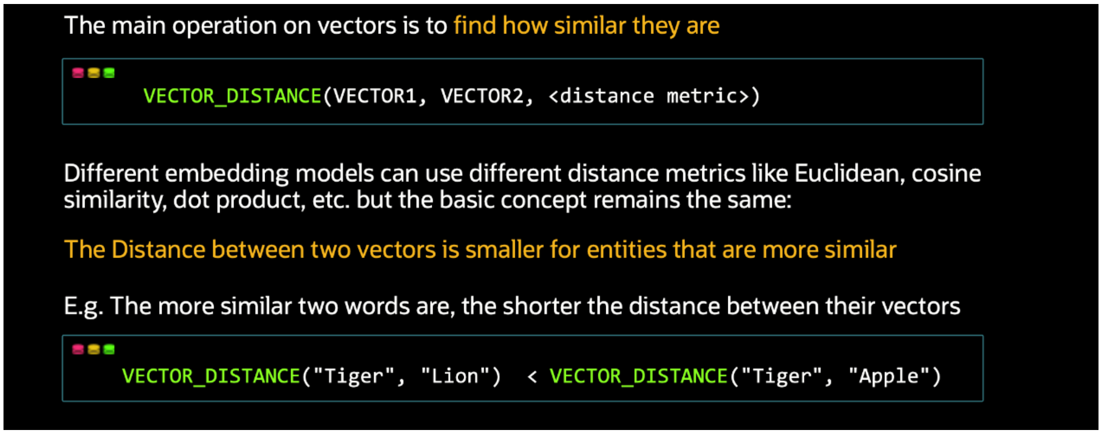
Developers should note how the simplified form on the right allows them to create their vector representations without having to know the number of dimensions it has or the data type for each entry. Using vector embeddings and vector search technologies is relatively new, and models constantly change. Oracle Database can recognize the data format and adjust how it deals with it so developers don't have to change their applications as their data and embeddings evolve.

It's also important to note that vector embedding models can be run within the Oracle Database itself instead of exporting data and running the embedding models on separate systems.

VECTOR SEARCH SQL OPERATORS

Vector search SQL operators are specialized SQL commands developed to leverage the power of vector indexes and perform similarity searches. These operators enable the database to execute queries that find items similar to a given input, not just exact matches.

For instance, in response to a query about a specific topic, the database can return documents, images, or other data types that are semantically related to the topic, even if they don't share specific keywords. This capability dramatically enhances the relevance and usefulness of search results.



The main operation on vectors is to find how similar they are

```
VECTOR_DISTANCE(VECTOR1, VECTOR2, <distance metric>)
```

Different embedding models can use different distance metrics like Euclidean, cosine similarity, dot product, etc. but the basic concept remains the same:

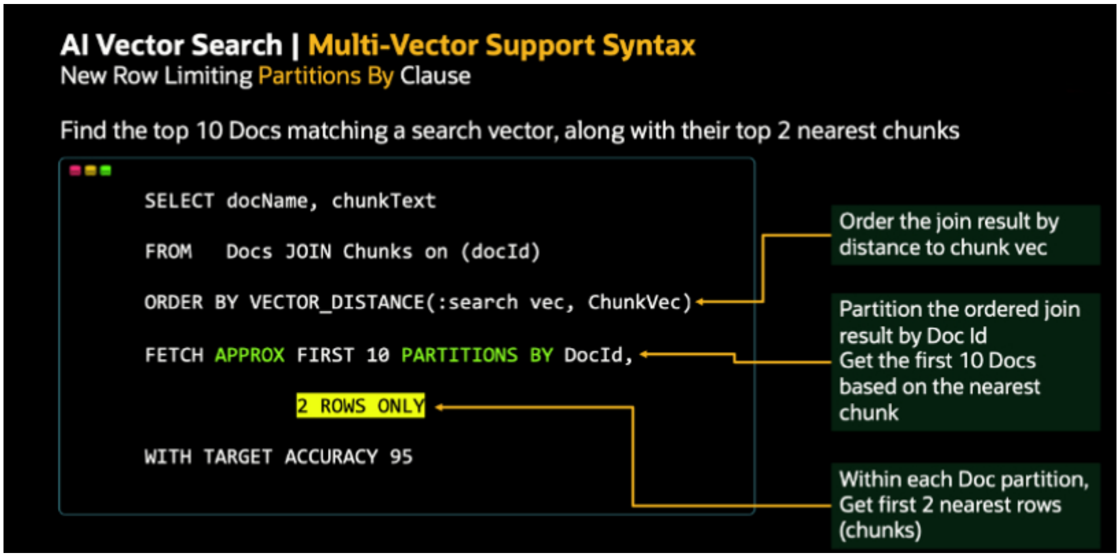
The Distance between two vectors is smaller for entities that are more similar

E.g. The more similar two words are, the shorter the distance between their vectors

```
VECTOR_DISTANCE("Tiger", "Lion")  < VECTOR_DISTANCE("Tiger", "Apple")
```

FIGURE 6: SQL VECTOR OPERATOR EXAMPLES IN ORACLE DATABASE 23AI

Oracle also supports multi-vector search for queries where multiple vectors may relate to a single entity; for example, a document can be broken up, or "chunked," into paragraphs, sentences, and words. Each chunk can be embedded into its vector, allowing users to perform a vector search with increased granularity.



AI Vector Search | Multi-Vector Support Syntax

New Row Limiting Partitions By Clause

Find the top 10 Docs matching a search vector, along with their top 2 nearest chunks

```
SELECT docName, chunkText

FROM   Docs JOIN Chunks on (docId)

ORDER BY VECTOR_DISTANCE(:search vec, ChunkVec)

FETCH APPROX FIRST 10 PARTITIONS BY DocId,

                 2 ROWS ONLY

WITH TARGET ACCURACY 95
```

Order the join result by distance to chunk vec

Partition the ordered join result by Doc Id
Get the first 10 Docs based on the nearest chunk

Within each Doc partition,
Get first 2 nearest rows (chunks)

FIGURE 7: ORACLE DATABASE AI VECTOR SEARCH MULTI-VECTOR SUPPORT – NEW ROW LIMITING PARTITIONS BY CLAUSE

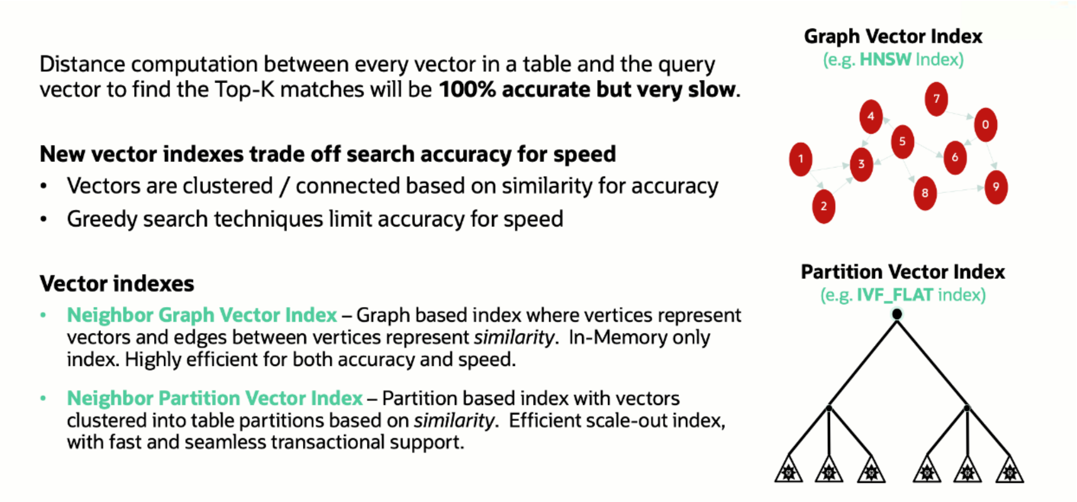
NEW VECTOR INDEXES

The VECTOR datatype and new SQL capabilities allow for vector data to be stored and managed. Still, the real power and performance of Oracle's implementation of vector search lies in its new vector index capabilities. Computing the distance between every vector in a table and the query vector to find matches is highly accurate but computationally demanding and slow.

Oracle addresses this with its new vector indexes, which give the user the flexibility to balance accuracy and speed. Just as traditional indexes help speed up queries by providing quick paths to data based on key values, vector indexes facilitate the rapid retrieval of semantically similar content by organizing and accessing data to reflect its semantic closeness or similarity.

Oracle Database 23ai offers two approaches for vector indexing, each providing different benefits depending on the application:

- **Neighbor Graph Vector Index**: A graph-based in-memory index that's highly efficient for accuracy and speed.
- **Neighbor Partition Vector Index**: This is a partition-based index with vectors clustered into table partitions based on similarity. It is an efficient scale-out index with fast and seamless transactional support.



Distance computation between every vector in a table and the query vector to find the Top-K matches will be **100% accurate but very slow**.

New vector indexes trade off search accuracy for speed

- Vectors are clustered / connected based on similarity for accuracy
- Greedy search techniques limit accuracy for speed

Vector indexes

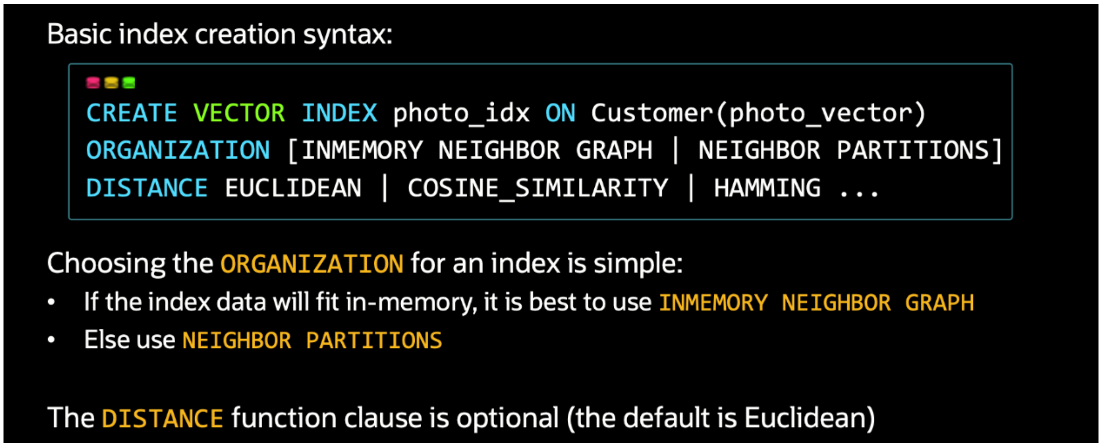
- **Neighbor Graph Vector Index** – Graph based index where vertices represent vectors and edges between vertices represent *similarity*. In-Memory only index. Highly efficient for both accuracy and speed.
- **Neighbor Partition Vector Index** – Partition based index with vectors clustered into table partitions based on *similarity*. Efficient scale-out index, with fast and seamless transactional support.

Graph Vector Index
(e.g. **HNSW** Index)

Partition Vector Index
(e.g. **IVF_FLAT** index)

FIGURE 8: VECTOR INDEX TYPES IN ORACLE DATABASE 23AI

The vector indexes are created and managed using new SQL syntax introduced in Oracle Database 23ai:



Basic index creation syntax:

```
CREATE VECTOR INDEX photo_idx ON Customer(photo_vector)
ORGANIZATION [INMEMORY NEIGHBOR GRAPH | NEIGHBOR PARTITIONS]
DISTANCE EUCLIDEAN | COSINE_SIMILARITY | HAMMING ...
```

Choosing the `ORGANIZATION` for an index is simple:

- If the index data will fit in-memory, it is best to use `INMEMORY NEIGHBOR GRAPH`
- Else use `NEIGHBOR PARTITIONS`

The `DISTANCE` function clause is optional (the default is Euclidean)

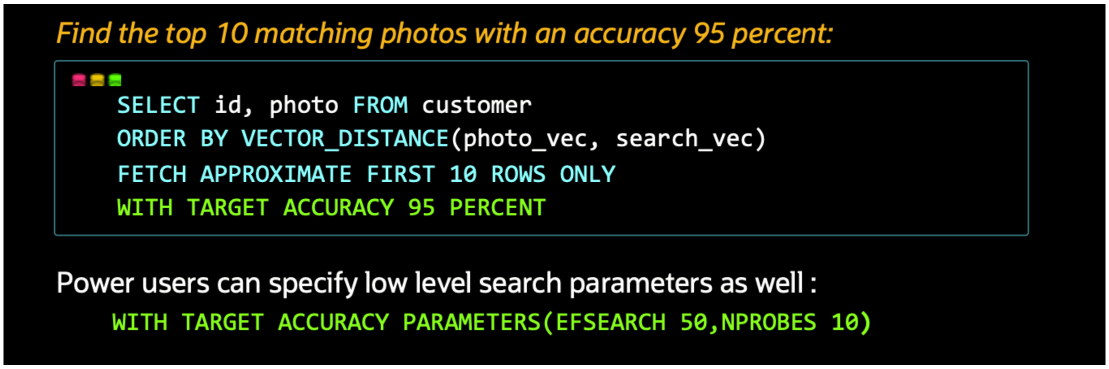
FIGURE 9: NEW SQL SYNTAX FOR CREATING VECTOR INDEXES IN ORACLE DATABASE 23AI

Vector indexes are generated and maintained by the Oracle Database. Index updates are performed automatically when database changes occur, and full updates can be performed in the background and without user intervention. As described below, the Oracle Database keeps tabs on the accuracy of ongoing searches and will automatically take action to improve the accuracy of customer searches when required. Oracle has a long history in optimizing indexes for all types of data and has built on decades of experience in this area to make using vector indexes fast and accurate.

SPECIFYING VECTOR SEARCH ACCURACY

Different applications require different levels of search accuracy and performance. For example, end-users expect near-instant responses to similarity searches for online shopping but may not require the highest level of accuracy. Conversely, applications such as those used in law enforcement or the financial sector may demand extremely high accuracy regardless of the impact on performance.

Oracle Database allows users to choose between accuracy and speed and is one of, if not the only, database with vector search capabilities that makes that choice available. Oracle Database 23ai allows users to target accuracy in vector index creation and similarity search queries, where the user can override the default indexing scheme.



Find the top 10 matching photos with an accuracy 95 percent:

```
SELECT id, photo FROM customer
ORDER BY VECTOR_DISTANCE(photo_vec, search_vec)
FETCH APPROXIMATE FIRST 10 ROWS ONLY
WITH TARGET ACCURACY 95 PERCENT
```

Power users can specify low level search parameters as well :

```
WITH TARGET ACCURACY PARAMETERS(EFSEARCH 50,NPROBES 10)
```

FIGURE 10: NEW SQL SYNTAX FOR SEARCHING VECTORS WITH USER-DEFINED ACCURACY

This example shows how a user can specify accuracy as part of a query. This code says that the query vector search query should quickly return the ten best results with 95% accuracy, so the user should expect that there should be one result that is semantically relevant but isn't in the top 10 for every two queries.

ENTERPRISE-GRADE CAPABILITIES

Oracle Database is found in environments where applications can't fail, performance is critical, and seamless scalability is paramount. Oracle's new vector search capabilities inherit all the attributes that define the Oracle Database.

Oracle’s premier platform for delivering these capabilities is Oracle Exadata, which is available in Oracle Cloud Infrastructure, as a foundational element of Oracle Database@Azure, as a hybrid cloud platform in customer data centers with Exadata Cloud@Customer, and as a traditional non-cloud platform with Oracle Exadata Database Machine.

Some of the key capabilities delivered by Oracle Exadata include:

- **Transparent scale-out of database operations**, with database operations distributed across multiple database servers via Oracle Real Application Clusters (RAC), which also provides high availability across component or server failures.
- **Single-system scaling** of database compute and storage. Oracle Database transparently scaled vector processing across computers in an RAC cluster with full data consistency. With AMD EPYC processors, Exadata systems can scale to have more than 4,000 cores of database compute and 3 petabytes of database storage.

- **Sharding** for further scale-out and geographic distribution of data.
- **SQL offload to storage servers** for processing core data in parallel on storage servers instead of requiring everything to be loaded into database servers for processing. On Exadata, storage-based Neighbor Partition Vector Indexes is accelerated by running distributed across multiple smart storage servers.

While each of these capabilities are critical differentiators for Oracle, the ability to run resource-intensive generative AI workloads on Exadata stands out.

Exadata is a pre-configured combination of hardware and software, including servers, storage, and networking. Exadata simplifies IT infrastructure by providing a single system optimized for running Oracle Database.

Oracle Exadata platforms are powered by AMD EPYC processors, which provide it with more processor cores to process more concurrent requests, more and faster memory to hold larger in-memory HNSW indexes, and more storage server processing cores to increase the rate at which data is processed. AMD EPYC processors help the database rapidly create vector embeddings inside the database, create and update vector indexes across multiple smart storage servers, and search through terabytes of vector index data. The high throughput of AMD EPYC processors and their ability to effectively encrypt and decrypt database data in real-time also helps customers maintain the security of their vector data.

KEY TAKEAWAYS

In the early days of generative AI, the industry looked towards specialized databases that focused only on supporting vector processing. While these point solutions provided a stop-gap solution for enterprises exploring the use of similarity search and AI, extracting the total value of robust new AI-driven solutions requires a database that can handle the fully converged needs of the enterprise. This is where Oracle steps in.

Choosing a converged database like Oracle Database 23 that integrates vector search capabilities has several advantages:

- **Versatility**: It can handle multiple data types and workloads, offering flexibility for current and future applications. You can add vector capabilities to any type of data – document, graph, spatial, text, images, etc.
- **Simplicity**: It reduces the complexity of managing multiple databases by serving as a corporate standard for various data-management needs. Add vectors to your existing data; don't add complexity to your IT architecture.
- **Integration of structured and unstructured data**: It allows combining business data with vector data, simplifying data management and enhancing consistency.

- **Utilization of existing skills**: Many developers and DBAs are already familiar with Oracle Database, which facilitates the adoption of vector search capabilities. AI Vector Search capabilities are available through familiar SQL APIs, APIs, and SDKs.

- **Enterprise-class features**: Oracle AI Vector Search benefits from Oracle's enterprise capabilities, including the performance, scalability, security, and reliability provided by Oracle Exadata platforms powered by AMD EPYC processors. This combination helps customers meet business expectations with their vector-enabled applications.

Oracle Database 23ai is a powerful platform for all business-critical applications, including AI-driven workloads. Oracle has enriched its database with multiple new AI building blocks, such as an LLM-based natural language interface and vector search capabilities supporting generative AI. Oracle takes a comprehensive approach to managing and leveraging vector data within AI-driven applications.

Oracle Database's AI Vector Search represents a leap forward in database technology, enabling more intelligent, efficient, and relevant searches. It opens new possibilities for managing and extracting value from unstructured data, making it a pivotal tool for businesses looking to harness the power of AI and semantic understanding in their operations.

The information contained within this document was believed by NAND Research to be reliable and is provided for informational purposes only. The content may contain technical inaccuracies, omissions, or typographical errors. This document reflects the opinions of NAND Research, which is subject to change. NAND Research does not warranty or otherwise guarantee the accuracy of the information contained within.

NAND Research is a technology-focused industry analyst firm providing research, customer content, market and competitive intelligence, and custom deliverables to technology vendors, investors, and end-customer IT organizations.

Contact NAND Research via email at info@nand-research.com or visit our website at nand-research.com.