

[Product](#) ▾[Solutions](#) ▾[Developers](#) ▾[Resources](#) ▾[Pricing](#) ▾[Company](#) ▾

☆ 5.1k

[Book a demo](#)[Try lakeFS](#)

# Cases

[Home](#) > [Blog](#) > What is a Vector Database? Top 12 Use Cases



[Idan Novogroder](#)

Last updated on November 10, 2025

Share this post

## Table of Contents

- What is a vector database?
- How does a vector database work?
- How does vector search work?
- How are vector databases used today?

Welcome to the lake, it's great to have you here!

[Product](#) ▾[Solutions](#) ▾[Developers](#) ▾[Resources](#) ▾[Pricing](#) ▾[Company](#) ▾

☆ 5.1k

[Book a demo](#)[Try lakeFS](#)

→ Wrap up

We're living in the age of AI. This technology has touched practically every industry, bringing about breakthroughs and, at the same time, introducing new challenges. Efficient data processing is very important for applications that involve AI/ML.

Today, they all rely on vector embeddings, which is a type of data representation that contains semantic data that AI engines need to develop knowledge and retain long-term memory used for performing complicated tasks.

AI models create embeddings that include a massive number of properties or features, making their representation difficult to manage. In the context of AI and machine learning, these characteristics represent the aspects of data that are critical for identifying patterns, correlations, and underlying structures.

That is why data practitioners require a special kind of database developed exclusively for dealing with this sort of data. This is where vector databases come in.

## What is a vector database?

Vector databases are [purpose-built to manage vector data](#) while also providing the performance, scalability, and flexibility teams need to get the most out of this type of

[Product](#) ▾[Solutions](#) ▾[Developers](#) ▾[Resources](#) ▾[Pricing](#) ▾[Company](#) ▾

☆ 5.1k

[Book a demo](#)[Try lakeFS](#)

sure that they're correctly calibrated to deliver excellent performance.

## Vector database vs. traditional database

Traditional databases, for example, relational databases, [store strings](#), integers, and other data in rows and columns. When you query such traditional databases, you look for rows that match your query. Vector databases, on the other hand, deal with vectors rather than strings and other such elements.

## How does a vector database work?

A vector database indexes vectors, among other things. To understand how a vector database works, let's take a look at the example of a large language model like GPT-4. This model contains a massive volume of data with a great deal of substance.

Here's a sequence of steps that happen when you interact with a GPT4-powered application:

1. You enter a query into the application as the user.
2. The query is sent into the embedding model, which generates vector embeddings depending on the material you want to index.
3. The vector embedding is stored in the vector database, along with the content from which it was created.
4. The vector database generates the output and returns it to the user as a query result.

[Product](#) ▾[Solutions](#) ▾[Developers](#) ▾[Resources](#) ▾[Pricing](#) ▾[Company](#) ▾

☆ 5.1k

[Book a demo](#)[Try lakeFS](#)

- **Indexation** – Once vector embeddings are in the vector database, a number of techniques are used to translate a given vector embedding to data structures for a quicker search.
- **Inquiry** – After completing its search, the vector database compares the query vector to the indexed vectors, using the similarity metric to determine its nearest neighbor.
- **Post-production** – Depending on the vector database, the final nearest neighbor will be post-processed to generate a final output for the query. In addition, the nearest neighbors may be re-ranked for future reference.

This embedding process is often carried out with the help of a neural network. Word embeddings, for example, turn words into vectors in such a way that words with similar meanings are closer together in the vector space.

## How does vector search work?

In a classic vector search use case, a query vector is passed to a vector database, and the vector database returns a customizable list of vectors with the smallest distance (“most similar”) to the query vector.

The following is a step-by-step workflow:

An engineer runs a dataset of documents, photos, or logs in R2 through a machine learning model made for that type of data to turn it into a set of vector embeddings,

[Product](#) ▾[Solutions](#) ▾[Developers](#) ▾[Resources](#) ▾[Pricing](#) ▾[Company](#) ▾

☆ 5.1k

[Book a demo](#)[Try lakeFS](#)

embeddings that are most comparable to the specified query.

In the absence of a vector database, you would have to provide your whole dataset alongside your query each time. This just isn't realistic because models have input size constraints. It's also not efficient, as it would use substantial resources and time. This use case alone shows why vector databases are such a welcome solution.

## How are vector databases used today?

Typically, vector DBs are used to power vector search scenarios such as visual, semantic, and multimodal search.

More recently, they're often combined with generative artificial intelligence (AI) text models to develop intelligent agents capable of providing conversational search experiences. They can also keep generative AI models from hallucinating, which can lead to bots providing nonsensical but plausible replies.

Vector databases are designed for engineers looking to create experiences that use vector search. To build embeddings and hydrate a vector database, an application developer can leverage open-source models, automated machine learning (ML) tools, and fundamental model services. This requires only a basic understanding of machine learning.

A team of data scientists and engineers can create highly adjusted embeddings and make them operational via a vector DB. This will allow them to deploy artificial

[Product](#) ▾[Solutions](#) ▾[Developers](#) ▾[Resources](#) ▾[Pricing](#) ▾[Company](#) ▾

☆ 5.1k

[Book a demo](#)[Try lakeFS](#)

indexing and retrieval of textual material encoded as word embeddings or sentence vectors.

A more nuanced understanding of language is possible thanks to vector databases' ability to reflect intricate semantic links that traditional data models are unable to capture. These linkages make it possible to analyze tone and context in Natural language processing applications, which enhances the efficiency of jobs like sentiment analysis and translation.

## 2. Anomaly and fraud detection

A vector database can be used to detect abnormalities in a variety of fields, including network traffic analysis, fraud detection, and cybersecurity. Teams can use it to compare data points to normal behavior patterns to identify anomalies based on distance from the typical vectors.

## 3. Improving machine learning models

Vector DBs can store and retrieve model embeddings that teams can use to enhance machine learning models and generative AI.

## 4. Similarity matching in recommendation systems

Vector databases can deliver customized suggestions based on user preferences, item attributes, or content similarity.

[Product](#) ▾[Solutions](#) ▾[Developers](#) ▾[Resources](#) ▾[Pricing](#) ▾[Company](#) ▾

☆ 5.1k

[Book a demo](#)[Try lakeFS](#)

## 7. Clustering and classification

Classification and clustering are supported by vector DBs, as they enable quick similarity-based grouping of data points.

## 8. Graph analytics

This is another use case of vector databases and includes community recognition, connection prediction, and graph similarity matching. They provide efficient graph embedding storage and retrieval for improved results.

## 9. Healthcare

Patient similarity analysis is one of the many applications of vector databases in the healthcare sector. Patient information, such as symptoms, medical history, and genetics, can be turned into vectors in a hospital context.

The vector database can be used to find prior patients with similar profiles when a doctor is treating a patient with an uncommon set of symptoms. This allows the clinician to evaluate therapies that have worked in the past or discover potential risk factors.

## 10. Autonomous vehicles

Vector databases are essential for processing sensor data in autonomous cars so that the vehicle can comprehend and navigate its surroundings. They transform input into

[Product](#) ▾[Solutions](#) ▾[Developers](#) ▾[Resources](#) ▾[Pricing](#) ▾[Company](#) ▾

☆ 5.1k

[Book a demo](#)[Try lakeFS](#)

## 12. MUSIC AND MULTIMEDIA STREAMING SERVICES

A vector representation of every song on a music streaming service like Spotify can be created using attributes like genre, rhythm, melody, and instrumentals. This is another powerful similarity search use case.

Such software can make use of the vector database to recommend additional tracks that have a similar feel to a user who is listening to jazz music with a specific tempo and mood, thus improving the user experience.

## How vector databases power Retrieval Augmented Generation (RAG)

The retrieval augmented generation (RAG) method is used to give an LLM (Large Language Model) more information about the context it is given. It's used in generative AI applications like chatbots and general question-answer apps. A vector database comes in handy here to supplement the query supplied to the LLM with additional context.

In the RAG technique, instead of providing the prompt straight to the LLM, engineers can create vector embeddings from an existing dataset or corpus – for example, one they wish to use to provide context to the LLM's response. Product paperwork, research data, technical specs, or your product catalog and descriptions can all be included here. Output embeddings are saved in the vector database index.

[Product](#) ▾[Solutions](#) ▾[Developers](#) ▾[Resources](#) ▾[Pricing](#) ▾[Company](#) ▾

☆ 5.1k

[Book a demo](#)[Try lakeFS](#)

particularly when handling and retrieving the high-dimensional data they generate.

This is where vector databases come into play.

Vector databases, with their capacity to manage high-dimensional data and execute quick similarity searches, are well-suited to support the operations of LLMs. They give a structured way to store and get back the vector embeddings that these models create, which lets you do quick searches for similarity in space with many dimensions.

## Why are vector databases important?

Data practitioners can index vectors made by embedding into a vector database. This allows them to locate comparable assets by searching for surrounding vectors.

This is how vector DBs allow embedding models to be operationalized. Database characteristics such as resource management, security controls, scalability, fault tolerance, and rapid information retrieval via complex query languages make the development process (and [data lifecycle management](#)) more productive.

Vector databases also enable developers to build one-of-a-kind application experiences. For example, your users may use their smartphones to take photos and search for comparable images.

Developers may use different types of machine learning models to automate the metadata extraction process from data such as scanned documents and photos. They can index information with vectors, allowing for hybrid searches that include both

[Product](#) ▾[Solutions](#) ▾[Developers](#) ▾[Resources](#) ▾[Pricing](#) ▾[Company](#) ▾

☆ 5.1k

[Book a demo](#)[Try lakeFS](#)

information. Vector databases can help solve this. Data practitioners can use them to supplement generative AI models with an external knowledge base to make sure they offer reliable information.

## Advantages of vector databases

Advantage	Description
<b>Data type</b>	Unlike traditional relational databases like PostgreSQL, which stores tabular data in rows and columns, or NoSQL databases which store data in JSON documents, vector databases are only designed to handle one form of data: vector embeddings.
<b>Scalability</b>	Vector DBs are built to manage massive amounts of data. They're great for large-scale machine-learning applications because they can store and explore billions of high-dimensional vectors.
<b>High-speed search performance</b>	Vector DBs use advanced indexing algorithms to enable the quick retrieval of related vectors in the vector space, even in large-scale datasets.
<b>Similarity searches</b>	Vector DBs may do similarity searches to determine the greatest match between a user's prompt and a certain vector embedding. This feature is very beneficial in the deployment of Large Language

[Product](#) ▾[Solutions](#) ▾[Developers](#) ▾[Resources](#) ▾[Pricing](#) ▾[Company](#) ▾

☆ 5.1k

[Book a demo](#)[Try lakeFS](#)

## Dimensional data

dimensional vectors into lower-dimensional spaces while retaining important information. As a result, they are efficient in terms of storage and computing.

## Key challenges of vector databases

Vector DBs share many of the issues that other database systems also face. The push to increase scalability, approximation accuracy, latency performance, and economics affects them as well.

As a relatively new technology, many vector databases must advance in key database skills such as security, robustness, operational support, and workload diversification. As artificial intelligence (AI) applications advance, they will demand more than just vector search.

Here are the most significant challenges of vector databases:

Challenge	Description
<b>Data management</b>	Vector DBs don't have strong transactional support and data management features, which makes it harder to guarantee data consistency, integrity, and scalability.

[Product](#) [Solutions](#)[Developers](#) [Resources](#)[Pricing](#) [Company](#)

☆ 5.1k

[Book a demo](#)[Try lakeFS](#)

<b>High dimensionality</b>	Debugging or fine-tuning AI and machine learning models that rely on vector representations can be tough when dealing with high-dimensional vectors since they can be hard to see and understand.
<b>Data structure</b>	High-dimensional vector data is the main use case for vector databases. This means that data structures that don't fit well into a vector format, including data with a large number of categorical variables or missing values, might not be a good fit.
<b>Data sparsity</b>	The majority of the vector components in high-dimensional vectors are often zero. Sparsity can cause inefficiencies in querying and indexing processes, necessitating the use of specific techniques to solve the problem.
<b>Latency</b>	When working with huge datasets or complicated queries, vector databases may have more latency than conventional databases.
<b>Problematic updates and deletions</b>	Changing or removing data in vector databases can be difficult since doing so requires preserving index consistency and controlling how the changes affect the data's vector representations.
<b>Flexibility</b>	Compared to standard databases, vector databases could be less capable in terms of data administration and querying. They

[Product](#) ▾[Solutions](#) ▾[Developers](#) ▾[Resources](#) ▾[Pricing](#) ▾[Company](#) ▾

☆ 5.1k

[Book a demo](#)[Try lakeFS](#)

## How to choose the right vector database

How do you pick a vector database that meets your needs and helps you achieve your data management and analytical objectives?

Keep scalability, data model, and integration capabilities in mind as you set out to [choose the best vector database](#) for your specific needs.

Here are some key points to consider when assessing different vector databases for your project:

### Scalability and performance

Check scalability in terms of the amount of data and the number of dimensions the database can successfully manage. Consider its performance metrics, including query response time and throughput, to make sure it meets your workload requirements.

### Data model and indexing methods

Explore the data model and indexing methods offered by the vector database. For example, check if it supports flexible schema designs.

Examine the database's indexing mechanisms to ensure efficient similarity search and retrieval operations. Tree-based structures, locality-sensitive hashing (LSH), and

[Product](#) ▾[Solutions](#) ▾[Developers](#) ▾[Resources](#) ▾[Pricing](#) ▾[Company](#) ▾

☆ 5.1k

[Book a demo](#)[Try lakeFS](#)

Check the vector database's integration with your existing systems, tools, and programming languages. See if the vector database has APIs, connectors, or SDKs to help with integration.

Compatibility with common frameworks and data processing tools will guarantee a good experience.

## Community and support

A vibrant community often acts as a source of useful information, discussion forums, and access to professional advice. Consider the level of support supplied by the database's developers, such as tutorials, documentation, and prompt customer care.

## License cost

Take into account any licensing or subscription fees involved with using the vector DB. Compare the pricing structure to your budget and the advantages provided by the database to check how it fits your financial goals.

## The future of vector database

As the need for managing machine learning vector data at scale expands, vector DBs are expected to become more important. After all, they deliver the performance, scalability, and flexibility required by AI applications across sectors.

[Product](#) ▾[Solutions](#) ▾[Developers](#) ▾[Resources](#) ▾[Pricing](#) ▾[Company](#) ▾

☆ 5.1k

[Book a demo](#)[Try lakeFS](#)

database with a vector-first design that no other technology offers at the moment.

## Wrap up

The world of search and information retrieval is rapidly evolving, and vector databases will [play an important role](#) in every software sector for the foreseeable future. As we see greater use of ChatGPT – particularly as an embedded component within applications – vector databases are expected to become increasingly common. They're expected to be used in every major application in the next few years, from search engines to accounting systems to meme generators.

Due to their outstanding capabilities for processing highly dimensional data and enabling sophisticated analysis, vector databases are game changers in data management. Benefits such as efficient similarity search capabilities and matching and query efficiency are invaluable to organizations across many industries.

There are already a number of great solutions on the market, each with its own set of advantages and limitations. Stay tuned for the second part of this post, where we will review the key vector databases for 2023.

Product ▾ Solutions ▾

Developers ▾ Resources ▾

Pricing ▾ Company ▾

☆ 5.1k

[Book a demo](#)[Try lakeFS](#)

Idan Novogroder

Author



Idan has an extensive background in software and DevOps engineering. He is passionate about tackling real-life coding and system design challenges. As a key contributor, Idan played a significant role in launching, maintaining, and shaping lakeFS Cloud, which is a fully-managed solution offered by lakeFS. In his free time, Idan enjoys playing basketball, hiking in beautiful nature reserves, and scuba diving in coral reefs.

## Related articles

### 2023 Top 10 Defining Product Milestones

[BEST PRACTICES](#)   [DATA ENGINEERING](#)   [MACHINE LEARNING](#)

## lakeFS Top 10 Defining Product Milestones in 2025

Oz Katz   January 14, 2026

Product ▾ Solutions ▾

Developers ▾ Resources ▾

Pricing ▾ Company ▾

☆ 5.1k

[Book a demo](#)[Try lakeFS](#)

Einat Orr, PhD January 4, 2026



DATA ENGINEERING MACHINE LEARNING

## What is Iceberg Versioning and How It Improves Data Reliability

Itai Gilo November 10, 2025

Try lakeFS open source data version control. Watch how it works

[Product](#) ▾[Solutions](#) ▾[Developers](#) ▾[Resources](#) ▾[Pricing](#) ▾[Company](#) ▾

☆ 5.1k

[Book a demo](#)[Try lakeFS](#)

Need help getting started?

[Talk to a lakeFS engineer](#)[Cloud](#)[Docs](#)[Blog](#)[Community](#)[Careers](#)[About Us](#)[Newsroom](#)[Contact Us](#)

[Product ▾](#)[Solutions ▾](#)[Developers ▾](#)[Resources ▾](#)[Pricing ▾](#)[Company ▾](#)

5.1k

[Book a demo](#)[Try lakeFS](#)