Vector databases (VDBs) have emerged to manage high-dimensional data that exceed the capabilities of traditional database management systems, and are now tightly integrated with large language models as well as widely applied in modern artificial intelligence systems. Although relatively few studies describe existing or introduce new vector database architectures, the core technologies underlying VDBs, such as approximate nearest neighbor search, have been extensively studied and are well documented in the literature. In this work, we present a comprehensive review of the relevant algorithms to provide a general understanding of this booming research area. Specifically, we first provide a review of storage and retrieval techniques in VDBs, with detailed design principles and technological evolution. Then, we conduct an in-depth comparison of several advanced VDB solutions with their strengths, limitations, and typical application scenarios. Finally, we also outline emerging opportunities for coupling VDBs with large language models, including open research problems and trends, such as novel indexing strategies. This survey aims to serve as a practical resource, enabling readers to quickly gain an overall understanding of the current knowledge landscape in this rapidly developing area.

A vector database (VDB) PDF is not a type of PDF, but rather a PDF document that describes or explains the concept of a vector database. A vector database itself is a specialized database system designed to store, search, and manage high-dimensional vectors, which are numerical representations of complex data like text, images, and audio. These databases enable efficient similarity searches, which are crucial for AI applications such as recommendation systems, semantic search, and generative AI.
What is a Vector Database?
Stores data as vectors:
Instead of traditional structured data, vector databases store information as high-dimensional vectors, also known as vector embeddings.
Captures semantic meaning:
These vector embeddings are numerical representations of data (like documents, images, or audio) that capture their meaning, characteristics, and relationships.
Enables similarity searches:
They excel at performing similarity searches, allowing you to find items that are semantically similar to a given query, even if they don't share keywords.
Why are Vector Databases Important?
Powering AI applications:
They are essential for large language models (LLMs) and other AI applications to understand context, provide personalized recommendations, and process complex, unstructured data.
Bridging the gap for AI:
Traditional databases struggle with the nuances of unstructured data, but vector databases allow AI models to "understand" and find relationships within vast datasets that can't be easily categorized.
How do they work?
Embedding generation: Data is first converted into vectors (embeddings) using an AI model.
Storage and indexing: These embeddings are then stored in the vector database and indexed for fast retrieval.
Similarity search: When a query is made, it's also converted into a vector, and the database finds the nearest vectors (most similar items) to it.
Examples:
Popular vector database solutions:
Pinecone, Milvus (mentioned in the initial results), and ChromaDB are some examples of vector databases used in AI applications.