# RAG in Typical Gen-AI application

A diagram of a software application

Description automatically generated

## Vector DB:

Let’s say a user ask query to your LLM,

Eg: What is Asynchronous programming in java?

Now you have documents or video transcript related to Asynchronous programming, Completable future and Web flux. But how to know these are related documents for the Query. ***There comes Vector DB.***

**Vector databases** are designed to store and search these vectors efficiently, supporting tasks like **semantic search**, **recommendations**, and **AI/ML applications**.

Typically, data like text, images, or other objects are transformed into **high-dimensional vectors** (via embeddings) using machine learning models, like **Word2Vec**, **BERT** etc.

Why Vector notation?

Vector is nothing but mathematical representation of an object (a piece of text, a sentence, a document) in a **high-dimensional space**.

These vectors are created for capturing the **underlying relationships** and **features** of the data.

How can vectors capture relationships?

**Vectors in a vector database** are stored as a collection of **points** (vectors) in this multi-dimensional space.

While you are uploading a document like completable future or Webflux to the vector DB, a model is used to create a vector for it.

Suppose the document on **CompletableFuture** is transformed into a vector with, say, **768 dimensions (This 768 is** those from BERT, OpenAI, or Nomic typically create high-dimensional vectors). Each of the **768 dimensions** will **capture different aspects** of the document’s content. Each dimension represents each characteristic of the document

Eg:

Consider 3 dimension’s X, Y and Z. Each dimension represents each characteristic. So, while uploading a document a data point will be added in that multiple dimensions for that document.

So next time if you add a similar document, a datapoint will be added near the previously added one.

That’s how when you query a question it’s giving you relevant data considering the data points in n-dimensions

### Why This Matters

* Traditional keyword search fails to capture semantic relationships
* Vector search understands context and meaning, not just exact words
* Enables more intelligent, context-aware retrieval

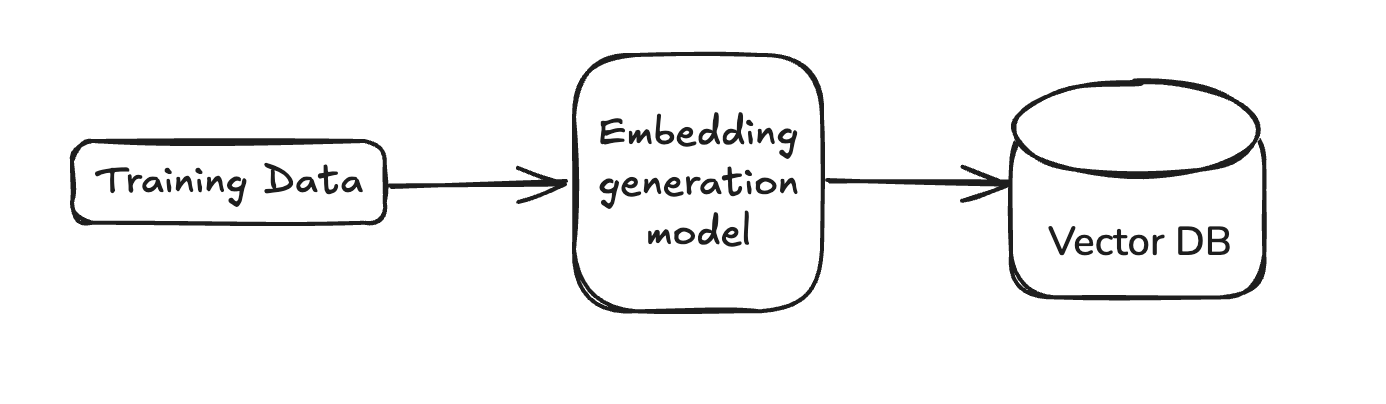
## Semantic search layer:

It’s nothing but retrieval model present in RAG. The semantic search layer is responsible for transforming both the **query** and the **documents** into **vector embeddings** using a technique like **Dense Embeddings and stored in Vector DB.** Then documents that are closest in semantic space to the query are retrieved.

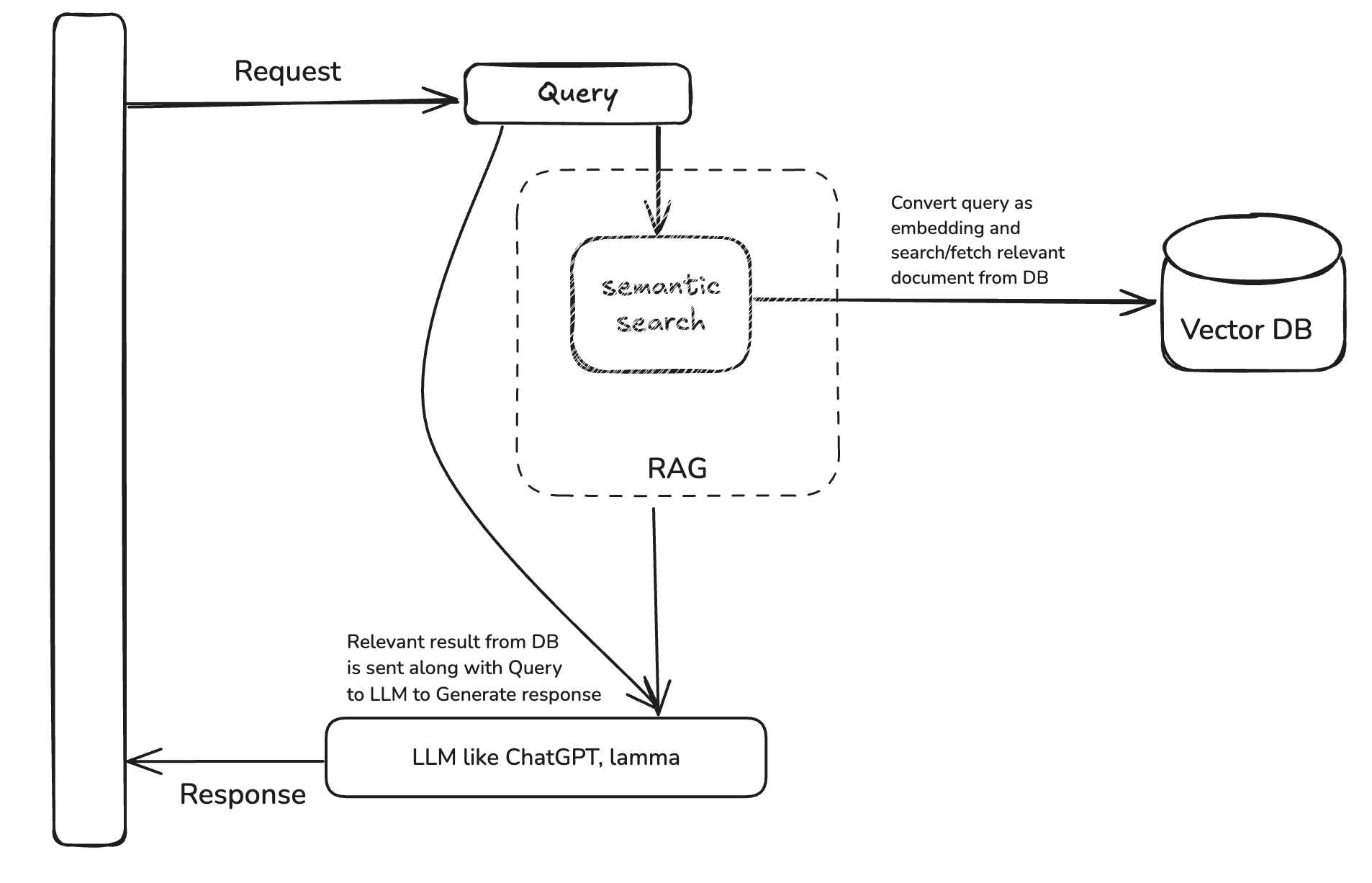
|  |  |
| --- | --- |
| Data source | Search method |
| Document management systems (Google Drive, Sharepoint, etc.) | Keyword search, custom query string |
| Relational databases (Postgres, MySQL, etc.) | SQL query |
| Vector databases | Semantic search query |

Now let’s look the whole flow about working of RAG:

Pre-Requisites thing -> Training data added in Vector DB



When a user searches a query to our LLM below will happen



So why RAG?

Simply put. LLMs can't access your company's information directly. RAG lets you share relevant documents with the LLM, so it can answer your questions using your own data.