Search Execution Flow: From Query to Result

You've got a vector DB filled with millions of high-dimensional embeddings. A user sends a query.

What exactly happens?

Step 1: Embed the Query

The query (text, image, etc.) is first encoded using the same model that was used to embed the documents.

- The result: a high-dimensional vector (e.g., 384D, 768D, etc.)
- This is the query vector. It lives in the same semantic space as the document vectors.

Step 2: Search the Index

VectorDBs handle N dimensions at once.

So, unlike relational DBs, you cannot use binary search on a sorted column. Because every row in that column has N values of its own.

Instead, we use ANN (Approximate Nearest Neighbor) indexes like IVF and HNSW.

Step 3: Score & Rank

For each candidate vector from the index, compute a similarity score using either:

1. *L2 distance (Euclidean)*
2. *Cosine similarity*
3. *Dot product (less common)*

Then return the top-K closest vectors. Cosine similarity is usually preferred for textual embeddings since it's scale-invariant.

Step 4: Post-processing & Filtering

Now you apply filters if needed:

- Language = English
- Published after 2023
- Category = Product Manual

Metadata filters are applied *after* vector scoring.

Summary

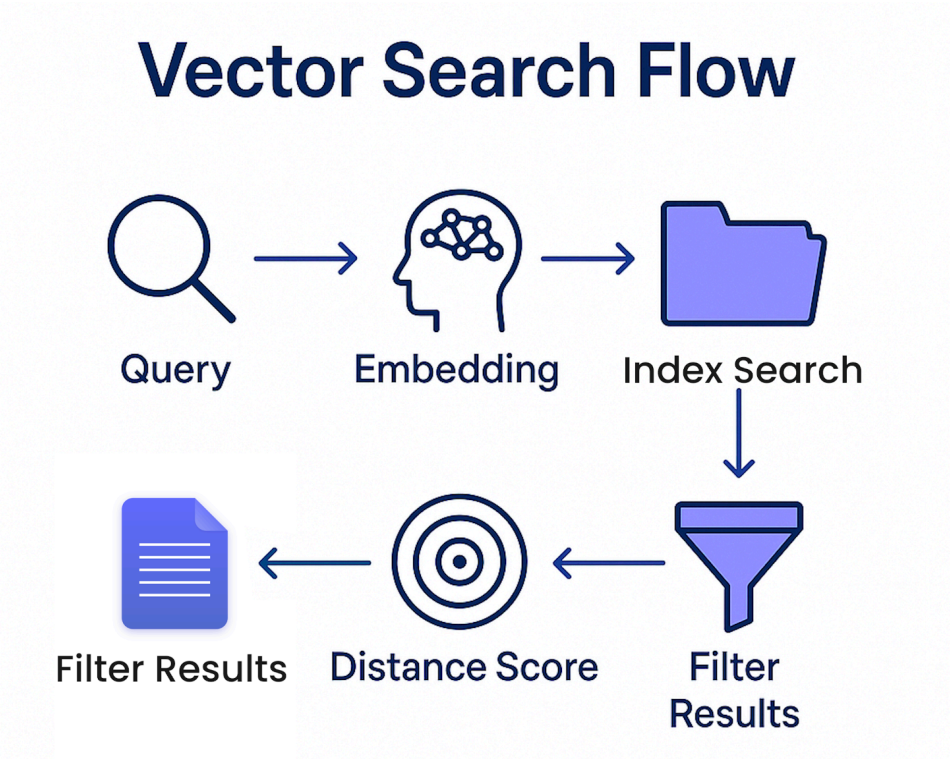

***Figure**: The vector search flow is*
Query → Embedding → Index Search → Distance Score → Top-K → Filter Results