Vector Databases – Interview & Revision Notes

1. Definition
A Vector Database stores and indexes embeddings (numerical vectors) that represent the semantic meaning of data like text, images, or audio.

2. Core Concepts
• Embedding – numeric vector representing data meaning
• Similarity Search – find nearest vectors
• Distance Metrics – cosine, Euclidean, dot product
• Indexing – HNSW, IVF, PQ for fast lookup
• ANN – Approximate Nearest Neighbor search
• Metadata Filtering – combine semantic + structured filters

3. Workflow
1. Input → embedding model → vector
2. Store in vector DB
3. Query → vector
4. Retrieve nearest vectors

4. Comparison: Vector vs Traditional DB
Aspect	Traditional DB	Vector DB
Data Type	Structured (text, numbers)	Embeddings (float arrays)
Query	Exact match	Similarity search
Index	B-tree, Hash	HNSW, IVF, PQ
Use	CRUD, transactions	AI search, RAG

5. Common Index Types
• HNSW – fast, graph-based
• IVF – partitions for scalability
• PQ – compresses vectors
• Flat – exact search

6. Popular Databases
• Pinecone – managed cloud
• FAISS – Meta open-source
• Milvus – distributed & scalable
• Weaviate – hybrid semantic queries
• Chroma – local, lightweight
• Qdrant – Rust-based

7. Query Examples
Weaviate (GraphQL)
```
{ Get { Document( nearText: { concepts: ["artificial intelligence"] } limit: 3 )
{ title summary _additional { distance } } } }
```

Pinecone (Python)
```
query_embedding = embed_model.embed_text("machine learning basics") results =
index.query(vector=query_embedding, top_k=3, include_metadata=True)
```

FAISS
```
import faiss, numpy as np index = faiss.IndexFlatL2(768) index.add(embeddings)
D, I = index.search(query_vector, k=5)
```

Milvus (PyMilvus)
```
from pymilvus import connections, Collection connections.connect("default",
host="localhost", port="19530") collection = Collection("docs")
collection.search(query_vectors, "embedding", param={"metric_type": "L2"},
limit=3)
```

8. Use Cases
• RAG (Retrieval-Augmented Generation)
• Semantic search
• Chatbots with document retrieval
• Recommendation systems

9. Advantages
✓ Semantic understanding
✓ High-dimensional search
✓ Scalable
✓ Ideal for AI data

10. Limitations
✗ High storage cost
✗ Embedding generation required
✗ Approximate results

11. Interview Quick Answers
Q. Why vector DB? → Semantic search via embeddings
Q. ANN? → Approximate Nearest Neighbor
Q. Metric? → Cosine similarity
Q. Common DBs? → Pinecone, Milvus, FAISS

Summary:
A Vector Database stores embeddings and performs similarity search — enabling AI systems to find semantically related data, unlike traditional databases that rely on exact matches.