# Retrieval-Augmented Generation (RAG) - Viva Q&A

**1. What is RAG?**

RAG (Retrieval-Augmented Generation) is a framework combining retrieval-based and generation-based models. It retrieves relevant data from an external knowledge base and uses that data to generate more accurate responses.

**2. Why do we need RAG?**

Because LLMs are trained on static data and lack access to new or specific information. RAG allows access to updated, relevant data, improving accuracy and reliability.

**3. What are the main components of RAG?**

Retriever, Generator (LLM), and Knowledge Base / Vector Database.

**4. What is the difference between RAG and a normal LLM?**

Normal LLMs use only internal knowledge. RAG retrieves external context before generating responses.

**5. What is the RAG pipeline?**

It includes data ingestion, embedding, vector storage, retrieval, and generation.

**6. What is an embedding?**

A numerical vector representation of text capturing its semantic meaning.

**7. What is a vector database?**

A specialized database (like FAISS or Pinecone) that stores embeddings and performs similarity searches.

**8. How does retrieval work?**

A query is embedded, compared to stored embeddings, and the most similar chunks are retrieved.

**9. What are 'chunks'?**

Small sections of text split from larger documents for efficient retrieval.

**10. What models/libraries are used in RAG?**

LangChain, LlamaIndex, SentenceTransformers, FAISS, Chroma, Pinecone, GPT, Llama, etc.

**11. Advantages of RAG?**

Up-to-date info, reduced hallucinations, no retraining needed, domain adaptability.

**12. Disadvantages of RAG?**

Can still hallucinate, needs maintenance, and has extra cost/complexity.

**13. What is hallucination?**

When LLMs generate factually incorrect but plausible information.

**14. How does RAG reduce hallucination?**

By grounding answers with real retrieved data.

**15. What happens if retrieval fails?**

Model may produce inaccurate answers without proper context.

**16. Difference between RAG and fine-tuning?**

RAG uses external data dynamically; fine-tuning modifies model weights.

**17. Retrieval types?**

Sparse (keyword-based) and Dense (embedding-based).

**18. Real-time applications of RAG?**

Chatbots, search systems, and enterprise assistants.

**19. RAG evaluation metrics?**

Precision/Recall, BLEU, ROUGE, and human evaluation.

**20. How to improve RAG performance?**

Use better embeddings, tune chunking, update knowledge, use re-ranking.