Master RAG Study Guide

Retrieval-Augmented Generation — Theory, Architecture, Industry Practice, and Interviews

1. What is Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) is a system design pattern where an LLM is augmented with external knowledge retrieved at runtime. Instead of relying only on model parameters, RAG systems retrieve relevant data and inject it into the prompt before generation.

Baseline flow: Query → Retrieve → Generate

2. Complete RAG Taxonomy

- Semantic RAG (Vector RAG): Embedding-based similarity search over unstructured text.
- Keyword / Lexical RAG: BM25 or keyword-based retrieval for exact matches.
- Hybrid RAG: Combination of semantic and lexical retrieval.
- Structured / Deterministic RAG: SQL, Pandas, APIs for exact answers.
- Agentic RAG: LLM agent decides when and how to retrieve.
- Hybrid Agentic RAG: Agent controls both semantic and deterministic tools.
- Adaptive RAG: Retrieval depth adapts to query complexity.
- Corrective RAG: Retrieval is repeated if answer quality is low.
- Self-RAG: Model evaluates its own answer and re-retrieves if needed.
- Multi-hop RAG: Multiple chained retrieval steps.
- Graph RAG: Knowledge stored as nodes and edges.
- Rerank-based RAG: Retrieve many candidates then rerank.

3. LangGraph Mapping of RAG Types

LangGraph represents RAG systems as explicit graphs. Each node is a step, and edges define control flow.

- Simple RAG: Query → Retrieve → Generate
- Hybrid RAG: Query → {Keyword Retrieve, Vector Retrieve} → Rerank → Generate
- Agentic RAG: Query → Agent → Tool Nodes → Synthesis
- Adaptive RAG: Query → Router → (No Retrieve | Shallow Retrieve | Deep Retrieve) → Generate
- Hybrid Agentic RAG: Query → Agent → {SQL Node, Vector Node, API Node} → Final Answer

4. Models Used in Industry (2025)

Embedding Models

- BAAI/bge-small-en-v1.5 – strong general-purpose
- bge-large-en – high accuracy
- E5-large – enterprise-grade
- MiniLM-L6-v2 – cost-efficient

LLMs

- GPT-4.1 / Claude 3 Opus – agentic reasoning
- Claude 3 Sonnet – hybrid RAG
- Gemini 1.5 Flash – fast semantic RAG
- Groq + LLaMA 3 – cost-optimized agentic systems

5. Interview Q&A; Playbook

- Q: What is RAG? A: Retrieval + generation to ground LLMs in external knowledge.
- Q: Difference between RAG and fine-tuning? A: RAG is dynamic and cheaper.
- Q: What is Agentic RAG? A: An agent decides how to retrieve before answering.
- Q: Why Structured RAG? A: For exact, non-hallucinatory answers.
- Q: When to use Adaptive RAG? A: Mixed query complexity and cost constraints.

6. Recommended GitHub Study Repository Structure

- rag-study/
- docs/ (all READMEs & PDFs)
- simple_rag/
- agentic_rag/
- hybrid_agentic_rag/
- langgraph_examples/
- interview_notes/
- README.md

Final Takeaway: Modern RAG systems are layered architectures combining truth, meaning, and control.