GENERATIVE AI · TAUGHT WITH QDRANT

Retrieval and Knowledge Systems

Chunking, embeddings, hybrid search, reranking, and citations that hold up. You also learn when a long context window beats retrieval outright, which is the part most courses skip.

Intermediate | 5 phases | 28 sessions | 42 contact hours

Syllabus

5 phases, 28 sessions. Each phase ends in something you have built.

1 Baseline and Ruler

Build the naive pipeline and the measurement that will later condemn it.

4 sessions
$72 alone

1.1 Naive RAG, Once

Fixed 512-token chunks, one embedding, top-k, answer: the baseline every later change must beat.

FREE PREVIEW

1.2 Building the Question Set

Mine real user questions and label the passages that answer them, instead of generating both.

1.3 Retrieval Metrics

recall@k, nDCG@10 and MRR, what each one hides, and why answer accuracy will not tell you.

1.4 Locating the Failure

Take one wrong answer and pin it on parsing, chunking, retrieval, ranking, or generation.

By the end: **A sixty-question labeled eval set and a naive top-k baseline with a recall@20 number to beat.**

2 Documents Before Vectors

Get text out of real files without destroying the structure that makes it findable.

6 sessions
$72 alone

2.1 Parsing Real PDFs

Docling and LlamaParse on scanned pages, two-column layouts, and tables that break across pages.

2.2 Chunking With Structure

Headings, sections, and parent-child chunks instead of a fixed window that cuts mid-sentence.

2.3 Contextual Retrieval

Prepend a generated one-line context to each chunk, then measure what it bought and what it cost.

2.4 Late Chunking

Embed the whole document first and slice after, then compare it against contextual retrieval.

2.5 Metadata You Filter On

Tenant, source, section path and effective date, designed at ingest rather than bolted on later.

2.6 Tables, Code, Images

The three content types plain text extraction quietly ruins, and what to store instead.

By the end: **A parsed corpus where every chunk carries its section path, page number, and source date.**

3 The Index

Serve millions of vectors with recall, latency and memory you can predict.

7 sessions
$72 alone

3.1	**Picking an Embedding Model**	Qwen3-Embedding, Cohere Embed v4 and Voyage 4 compared on your corpus, not on an MTEB average.
3.2	**Dimensions and Matryoshka**	Truncate 3072 down to 512, read the recall you lost, and price the storage you saved.
3.3	**HNSW, Tuned**	m, ef_construct and ef: the three numbers trading recall against latency and RAM.
3.4	**Quantization and Rescoring**	Scalar and binary quantization, and the float32 rescore that often costs more latency than it buys.
3.5	**Hybrid Search**	Sparse BM25 vectors and dense vectors in one Qdrant query, fused with reciprocal rank fusion.
3.6	**Filters and Tenants**	Payload filters, the filtered HNSW recall cliff, and testing recall under real access rules.
3.7	**Keeping the Index Fresh**	Incremental upserts, tombstones, and re-embedding into a versioned collection before swapping traffic.

By the end: **A Qdrant collection doing filtered hybrid search inside a stated p95 latency and RAM budget.**

4 Ranking and Grounding

Turn a hundred candidates into eight chunks and an answer a reader can check.

5 sessions
$72 alone

4.1	**The Reranking Pass**	Cohere Rerank 4 and Qwen3-Reranker over a hundred candidates, cut to eight, and the latency added.
4.2	**Late Interaction**	ColBERT-style multi-vector scoring, sitting between a bi-encoder and a full cross-encoder.
4.3	**Query Rewriting, Honestly**	HyDE, multi-query and decomposition, each kept only if it moves recall@20 on your own set.
4.4	**Assembling the Context**	Order, dedupe and budget the final chunks, because position still decides what the model reads.
4.5	**Citations That Survive**	Span-level attribution, and the deceptive-grounding case where real evidence names the wrong entity.

By the end: **An answer endpoint returning span-level citations, with reranking latency measured per query.**

5 What Retrieval Cannot Fix

Decide honestly whether retrieval is the right shape for the problem, and prove it.

6 sessions
$72 alone

5.1	**Groundedness Is Not Faithfulness**	Score one fabricated answer with Ragas and DeepEval, then work out why the two disagree.
5.2	**Long Context vs Retrieval**	Run both on the same eval: effective context lands near 50 to 65 percent of the advertised window.
5.3	**Caching and Cost**	Cost and latency per query for a cached million-token prompt against one retrieval call.
5.4	**GraphRAG, and When Not**	Microsoft GraphRAG, LightRAG and LazyGraphRAG: indexing cost weighed against real multi-hop gain.
5.5	**Agentic Retrieval**	Search, verify, refine, stop: capped loops, semantic caching, and a trace for every step.
5.6	**Shipping and Watching**	Evals in CI, drift on the live query log, and a retrieval config you can version and roll back.

By the end: **A written decision, backed by your own eval, on retrieval versus long context versus a graph index.**

Tools you will use

Qdrant 1.18 | Docling | Chonkie | FastEmbed | Sentence Transformers | Qwen3-Embedding-8B | Cohere Rerank 4 | Ragas | DeepEval

What you will build

1. **The Ruler**
   Sixty labeled questions and a naive baseline to beat
2. **Structured Corpus in Qdrant**
   Parsed, filtered, quantized, hybrid, measured at p95
3. **The Cited Answer**
   Reranked, attributed, benchmarked against long context

Registration

Phase 1: Baseline and Ruler	$72
Phase 2: Documents Before Vectors	$72
Phase 3: The Index	$72
Phase 4: Ranking and Grounding	$72
Phase 5: What Retrieval Cannot Fix	$72
Whole track, all 5 phases	**$300**

Buying every phase separately costs $360, so the whole track saves $60.

No payment is taken online. Registering creates an order and payment instructions are sent with it. Access opens once payment is confirmed.