Schema-Aware Retrieval for RAG Systems

Course Overview

Retrieval-Augmented Generation has become the standard pattern for building LLM-powered applications — but most RAG pipelines are built on a flawed assumption: that all data can be treated as flat text. The moment a user asks something like "show me all high-priority billing issues from last month," pure vector search fails. It cannot distinguish a categorical field from a free-text description, cannot apply a date range, and cannot reliably filter by structured attributes.

This course teaches you to build retrieval systems that understand the structure of your data before running any semantic search. You will learn to work with real-world schema types, design field-appropriate indexes, parse natural language queries into structured filters, and combine multiple retrieval strategies into a robust, production-ready pipeline.

By the end of the course, you will have built a complete query engine from scratch — one that handles both structured and unstructured queries accurately, degrades gracefully on edge cases, and is ready to solve the retrieval problems that actually show up in production.

Course Objectives

By completing this course, learners will be able to:

- Explain why naive vector search fails on structured queries and articulate the limitations of pure semantic search in production RAG systems
- Identify and extract schema information from tabular, JSON, relational, and document-store data sources
- Design and implement schema-aware chunking strategies that preserve structured metadata while producing high-quality embeddable text
- Build purpose-appropriate indexes — hash, inverted, range, and vector — and combine them into an efficient multi-index retrieval store
- Parse natural language queries into structured filters using both rule-based regex approaches and LLM-based tool calling
- Apply advanced retrieval strategies including filter-then-rank, multi-field retrieval, BM25 + vector hybrid search, and progressive fallback chains
- Evaluate retrieval system performance using standard metrics and compare schema-aware approaches against naive baselines

What You Will Learn

- **Why vector search breaks on structured data** — hands-on demonstration of naive search failure and the exact conditions that cause it
- **Schema types across real systems** — how to read and interpret tabular, JSON/XML, relational (SQL), and document-store schemas
- **Chunking strategies for structured data** — field-level, record-level, LLM-annotated, and hybrid chunking with practical decision guidance
- **Index design and construction** — building hash indexes for exact match, inverted indexes for full-text search, range indexes for dates and numbers, and vector indexes for semantic similarity
- **Natural language query parsing** — rule-based extraction with regex and LLM-based structured parsing using Claude API tool calling, including ambiguity detection
- **Retrieval strategy selection** — when to use filter-then-vector, multi-field retrieval, BM25 + vector hybrid with Reciprocal Rank Fusion, and fallback relaxation chains
- **End-to-end query engine assembly** — wiring schema understanding, indexing, query parsing, and retrieval into a unified pipeline
- **Retrieval evaluation and metrics** — measuring precision, recall, and system behavior to validate that your pipeline meets real-world quality standards

Prerequisites

- Solid working knowledge of Python
- Basic understanding of how text embeddings and cosine similarity work
- Familiarity with JSON and common data formats
- Some exposure to SQL or relational database concepts (helpful but not required)
- No prior experience with vector databases or retrieval frameworks is needed — all components are built from scratch

Who This Course Is For

- **RAG developers who have hit a ceiling** with basic vector search and need their systems to handle structured, filtered queries reliably
- **AI and ML engineers** working on applications that combine structured databases with LLM-generated responses
- **Backend developers** integrating language models into products that rely on relational databases, document stores, or complex JSON schemas
- **Data engineers and architects** who want to understand how retrieval pipelines interface with different storage systems and schema types

- **Students and early-career engineers** building a portfolio of production-quality AI engineering skills that go beyond tutorial-level RAG
- **Anyone preparing for applied AI engineering roles** where retrieval accuracy, structured query handling, and evaluation methodology are expected competencies

Course Syllabus

Chapter	Topic	Lab
1	**What Is Schema-Aware Retrieval** — Introduction to the core concept, the structured-first retrieval philosophy, and the problem it solves	Lab 1
2	**Why Schema Matters in Retrieval** — Live demonstration of naive vector search failure; building intuition for when and why pure semantic search falls short	Lab 2
3	**Types of Schemas in Real Systems** — Working with tabular (CSV/SQL), JSON/XML, relational, and document-store schemas; building a unified schema inspector	Lab 3
4	**Schema-Aware Chunking** — Field-level, record-level, LLM-annotated, and hybrid chunking strategies; the golden rule of chunk design	Lab 4
5	**Schema-Aware Indexing —** Hash indexes for exact match, inverted indexes for full-text search, range indexes for dates and numbers, vector indexes for semantic search; combining all four into a multi-index store	Lab 5
6	**Schema-Aware Query Understanding —** Rule-based query parsing with regex; LLM-based parsing with Claude API tool calling; handling ambiguous and underspecified queries	Lab 6
7	**Schema-Aware Retrieval Strategies —** Filter-then-vector, multi-field retrieval, BM25 + vector hybrid with Reciprocal Rank Fusion, and progressive fallback chains	Lab 7
8	**Naive vs. Schema-Aware Retrieval —** Side-by-side comparison of both approaches on identical query sets; quantifying the improvement	Lab 8
9	**Evaluation and Metrics —** Measuring retrieval quality with precision, recall, and domain-specific metrics; validating end-to-end pipeline performance	Lab 9

Skills You Will Gain

- **Schema analysis** — reading and interpreting data structure across tabular, JSON, relational, and document-store sources to inform retrieval design
- **Structured chunking** — designing chunking pipelines that embed structured metadata alongside semantic content without losing either
- **Multi-index construction** — building and combining hash, inverted, range, and vector indexes to serve different query types efficiently
- **LLM-powered query parsing** — using Claude API tool calling to extract structured filters, date ranges, and semantic components from natural language input
- **Hybrid retrieval** — implementing BM25 + vector search fusion and understanding when each signal contributes more effectively
- **Fallback and edge-case handling** — designing progressive query relaxation chains so retrieval degrades gracefully rather than returning empty results
- **Retrieval evaluation** — applying quantitative metrics to compare and validate retrieval pipeline performance against real-world query benchmarks