Cognee and dlt
DataTalks.Club: LLM Zoomcamp 2025

dltHub

About me

- AI Engineer @ dltHub
- Previously:
  - *I do data, I do AI*
  - Data science and ML roles within startups and academia
- LinkedIn: linkedin.com/in/hiba-jamal/

dltHub

What we'll be doing

1. Going over some basic concepts.
    a. What is dlt?
    b. RAG systems: a refresher.
    c. What is cognee?
    d. What are cognee node sets?
2. An example with the NYC Taxi dataset.
3. An example with REST API source docs.
4. Key takeaways.

* If anyone wants to code together for part 2, please go on to the colab notebook and run all cells rn! It takes time to build a knowledge graph :)

dltHub

Basic Concept 1: What is dlt?

a.k.a data load tool

dltHub

Main idea:

dltHub

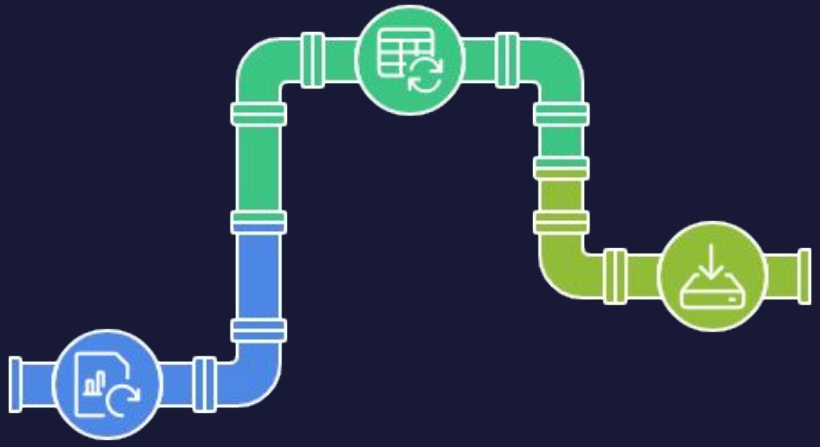
What is dlt?

Every data job is a data engineering job when you go through the ETL/ELT process.

1. dlt (data load tool) is an open-source Python library that lets you build modern ELT pipelines using just Python code.
2. It helps you:
    a. Extract data from APIs, databases, files, or custom sources
    b. Transform and normalize data
    c. Load data into destinations like BigQuery, DuckDB, Redshift, etc.
    d. Manage schemas, state, and incremental loading automatically

dltHub

What happens when you run `pipeline.run()`?



01

Extract Data

Data is retrieved from the source

02

Normalize Data

Data is transformed into a structured format

03

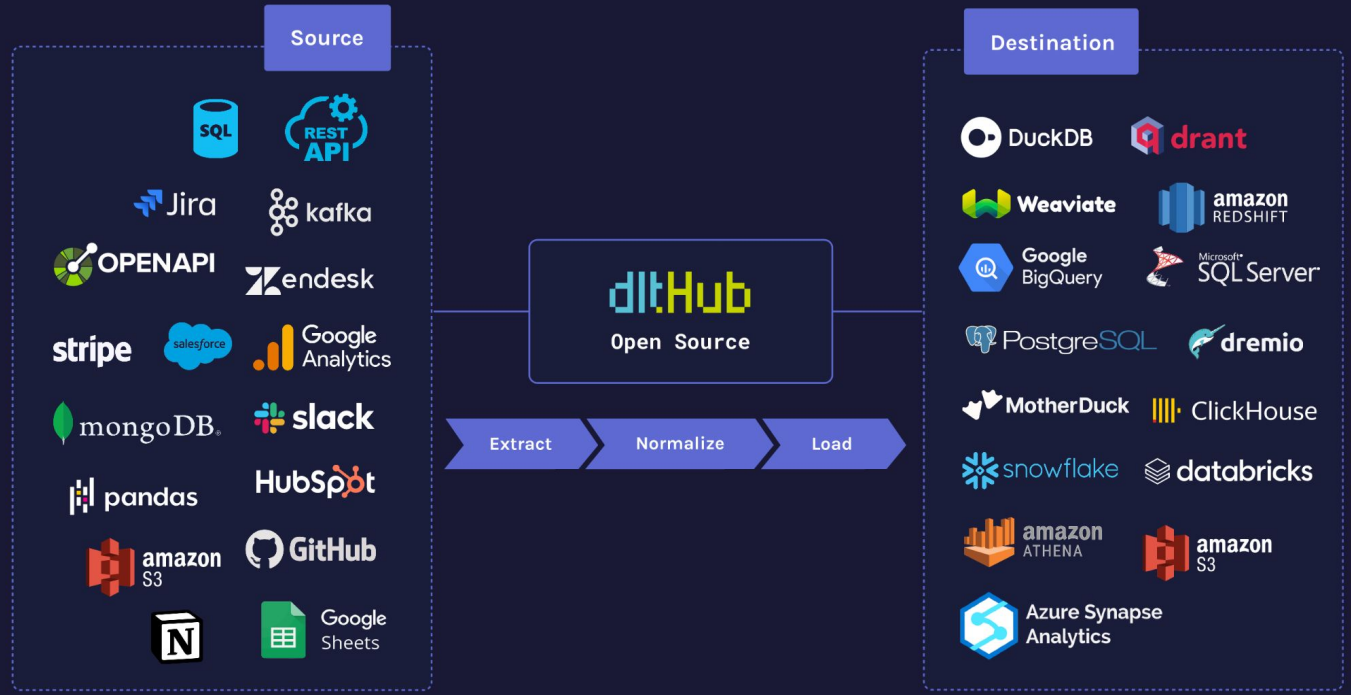
Load Data

Data is loaded into the destination

dltHub

What is dlt?

The dlt is an open-source Python library that simplifies data loading by automating complex tasks like schema creation, data normalization, incremental loading, etc. Much like dbt democratized the T(Transform) layer of ELT to SQL users, dlt democratizes the EL (Extract/Load) aspect of data handling with Python.



dltHub

How we'll be using dlt

1. Taxi dataset:
    a. Ingesting a dataset into a local DuckDB destination.
    b. Access local data as a dataframe.

2. REST API sources dataset:
    a. Ingesting data to a local filesystem.

dltHub

RAG Systems

A refresher

dltHub

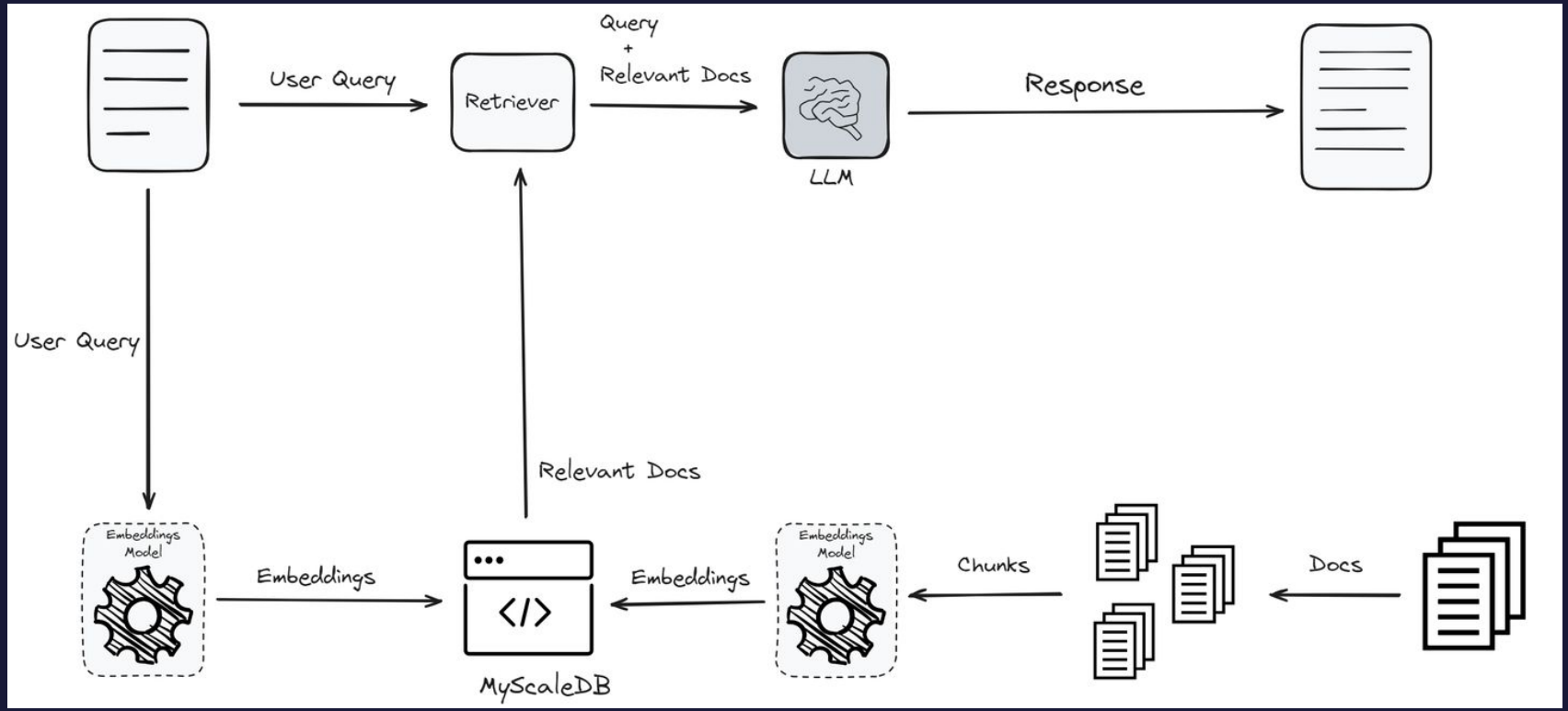


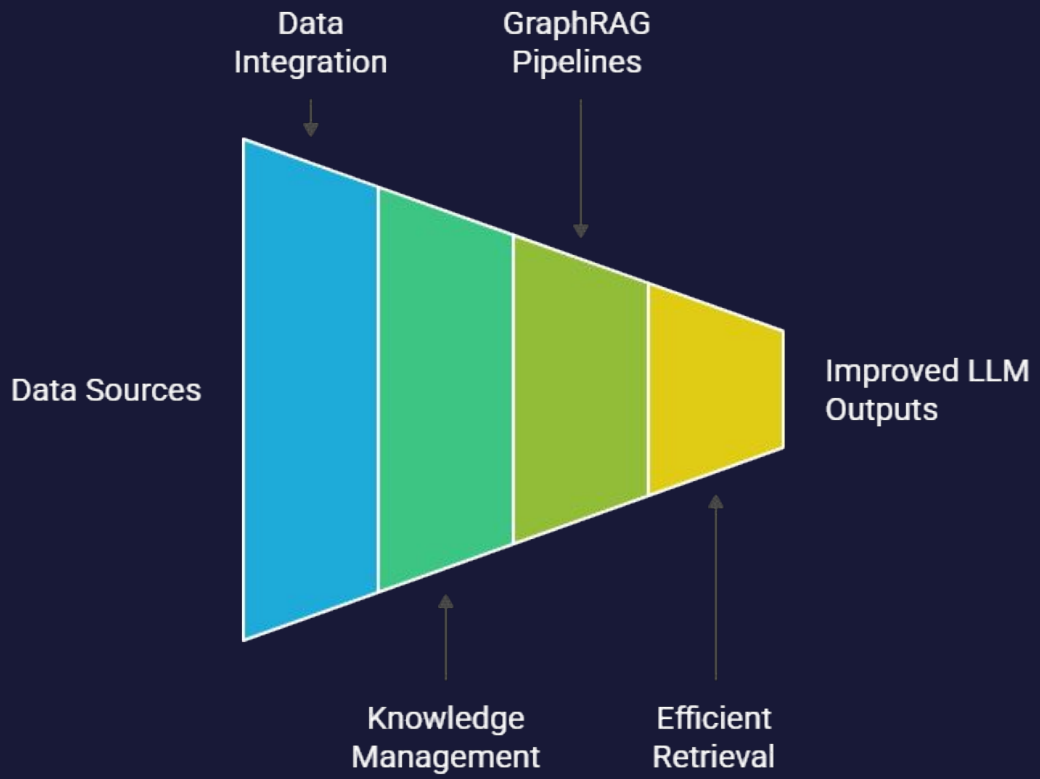
Image source: myscale.com

dltHub

What is cognee?

dltHub

Main idea:



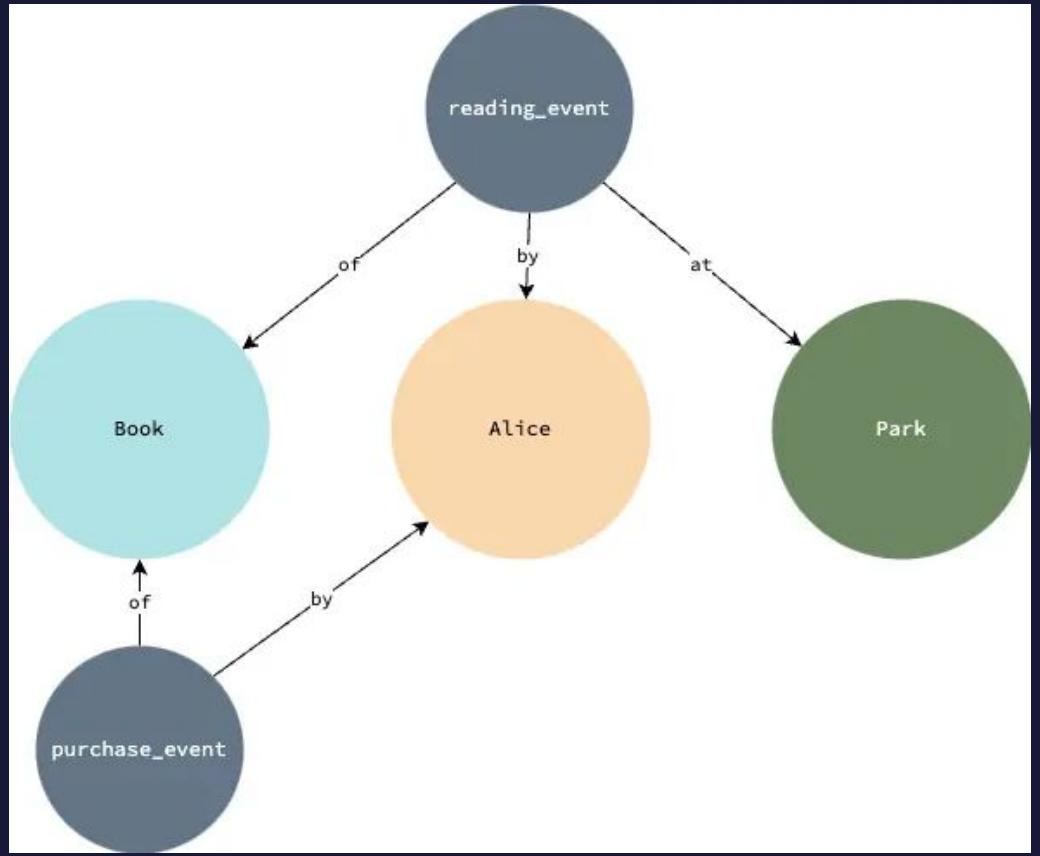
dltHub

What is cognee?

1. Cognee turns your data into a queryable memory and knowledge graph.
2. It lets you:
    a. Add structured or unstructured data (DataFrames, documents, tables)
    b. Automatically build a knowledge graph from it
    c. Ask natural language questions and get grounded, context-rich answers

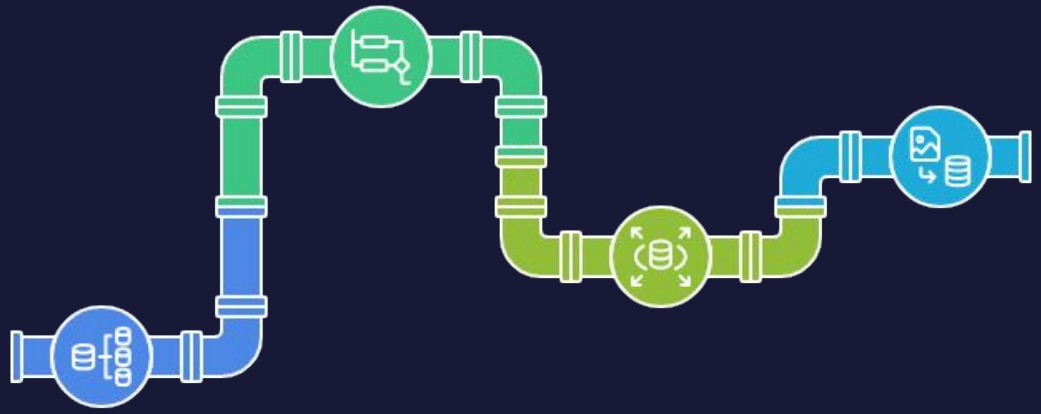
dltHub

What is a knowledge graph?



dltHub

What happens when you run cognee.cognify()?



01

Parse Data

Reads and extracts data from DataFrame or document

02

Build Knowledge Graph

Identifies and connects entities and relationships

03

Enrich with Metadata

Adds contextual information to the data

04

Save to Cognee Memory

Stores the enriched data for future queries

dltHub

Challenges of RAG Systems

dltHub

Challenges of RAG systems

RAG Challenge	Description	Cognee's Solution
Retrieval Quality & Relevance	RAG may retrieve irrelevant, incomplete, or outdated information, leading to poor or misleading outputs.	Graph-based retrieval: Uses a semantic graph to identify and connect relevant facts, improving precision and contextual relevance.

dltHub

Challenges of RAG systems

RAG Challenge	Description	Cognee's Solution
Ranking & Consolidation	Retrieved chunks are not always ranked by usefulness, leading to suboptimal generation.	Graph traversal: The graph structure helps prioritize and consolidate the most relevant information for the query.

dltHub

Challenges of RAG systems

RAG Challenge	Description	Cognee's Solution
Pipeline Scalability & Data Ingestion	Scaling ingestion and retrieval pipelines for large datasets is complex and resource-intensive.	LanceDB & **dlt** integration: Simplifies local, scalable vector storage and reduces complexity in ETL and metadata management.

dltHub

Example of how these challenges can look like:

Leo likes German cars, advanced tech and high performance sport features

Which exact car model should I buy?

LLM

Leo should buy an electric sports car such as Tesla Roadster or Porsche Taycan.

cognee

Leo should buy Porsche Taycan, produced between 2019 and 2025, with 79.2 kWh or 93.4 kWh battery and a range of 333 km to 463 km, produced in Germany based on his pref based on his preferences for electric cars that emphasize sustainability and advanced technology.

dltHub

What are Cognee Node Sets?

dltHub

What are cognee node sets?

1. They are lists of tags (strings) attached to content in Cognee's knowledge graph for organizing, filtering, and categorizing data.
2. Advantages:
   a. Topic Isolation: Agents can write/retrieve data by topic, keeping domains separate.
   b. Better Organization: Maintains structure as your knowledge base grows.
   c. Improved Retrieval: Enables precise, tag-based filtering for searches.
   d. Topic-Based Analysis: Makes it easy to analyze patterns within specific domains.
   e. Scalable Knowledge Management: Reduces complexity as content increases.
   f. Node sets are supported with Kuzu and Neo4j graph databases.

dltHub

Let’s get into the first demo!

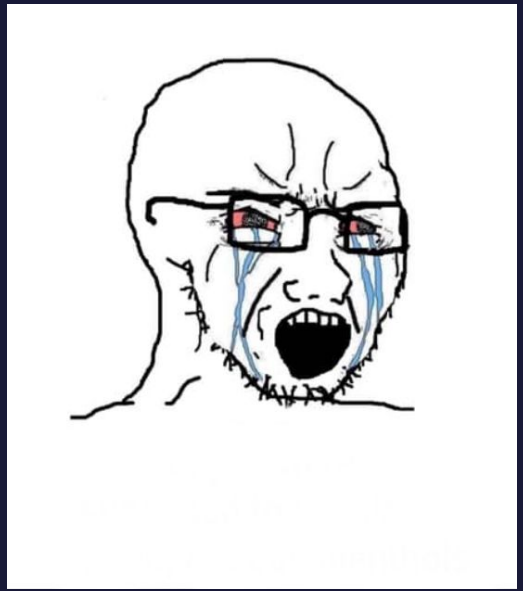
dltHub

Demo 2:
Intro to a real-life dlt project 👀

dltHub

Understanding API sources is important to us

1. We were building some RAGs to be able to understand API sources and docs.
2. We figure it is an important element to help you write data pipelines in today's AI accelerated world!



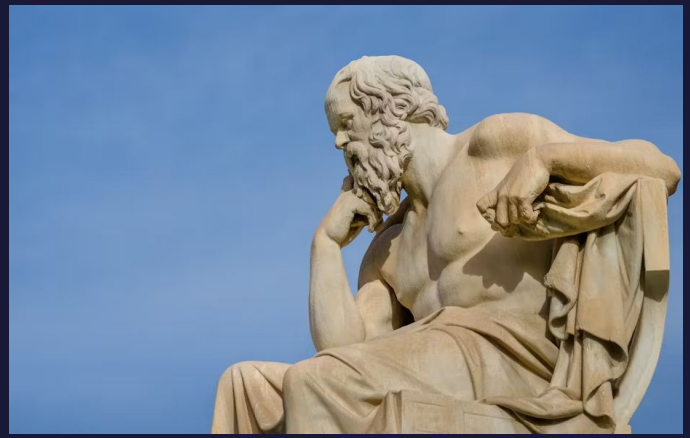
- API docs are huge
- We build pipelines on some predetermined rules.
- API docs are defined quite differently.
- API docs are huge.

<insert hiba's rant>

dltHub

Using node sets with ontology

dltHub



What is ontology?

dltHub

What is ontology?

1. Ontology is a formal, machine-readable specification that defines:
    a. Entities (things, concepts, or objects)
    b. Classes (types or categories of entities)
    c. Relationships (how entities and classes are connected)
    d. Properties (attributes or characteristics of entities)

Ontologies are typically written in standards like OWL (Web Ontology Language) or RDF/XML.

dltHub

What ontologies do (in any system)

1. Provide a shared vocabulary: Everyone (and every system) refers to things the same way, reducing ambiguity.
2. Structure data: They organize information into clear categories and relationships, making it easier to search, analyze, and reason over the data.
3. Enable reasoning: Systems can infer new facts or connections based on the rules and relationships defined in the ontology.
4. Support interoperability: Data from different sources can be integrated and understood consistently if they use the same ontology.

dltHub

Ontology use case

1. In business analytics, an ontology might define concepts like “Customer,” “Order,” and “Product,” and specify how they relate, so different departments and tools can collaborate and analyze data accurately.
2. In API docs, an ontology defines concepts like “Endpoint”, “Base URL”, “Pagination”, “Authentication”, and how they relate, so the data pipeline and data engineer can understand how to use them.

dltHub

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
>
  <rdf:Description rdf:about="http://example.org/apispec#Resource">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://example.org/apispec#api_types_available">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <rdfs:domain rdf:resource="http://example.org/apispec#SourceMetadata"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://example.org/apispec#cursor_param">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <rdfs:domain rdf:resource="http://example.org/apispec#Paginator"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://example.org/apispec#error_message">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <rdfs:domain rdf:resource="http://example.org/apispec#ErrorPattern"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
```

dltHub

Let’s get into the second demo!

dltHub

Helpful queries

1. Show Everything:

   MATCH (n)-[r]->(m) RETURN n, r, m

2. Focus on NodeSets

   // Get all NodeSet nodes MATCH (s:NodeSet) WITH s // Step 1: Sample 4 entities per NodeSet CALL { WITH s MATCH (n)-[r:belongs_to_set]->(s) RETURN n, r, s AS originalSet LIMIT 4 } // Step 2: For each sampled entity, fetch all belongs_to_set relationships (across all NodeSets) OPTIONAL MATCH (n)-[r2:belongs_to_set]->(otherSet:NodeSet) // Return everything RETURN DISTINCT originalSet, n, r, otherSet, r2

dltHub

How did cognee help us?

1. Specific search into the knowledge graph. We can draw some parallels with normal RAG system.
2. We had a central definition to inform our knowledge graph about our definitions of API docs.
3. Retrieval was very targeted. Otherwise I suffer with context windows in my other RAG pipelines.
4. Semantic definitions enrichment!

dltHub

Questions?

dltHub