Operational Ontology using Semantic Knowledge Graph in Snowflake

A Solution for Pharmaceutical R&D

Executive Summary

We have built a complete pharmaceutical knowledge graph solution entirely within Snowflake that enables scientists to ask natural language questions and receive answers that combine experimental data with biological knowledge. The system ingests standard biomedical ontologies (OWL/RDF), links them to experimental data products, and provides intelligent query capabilities through an AI agent.

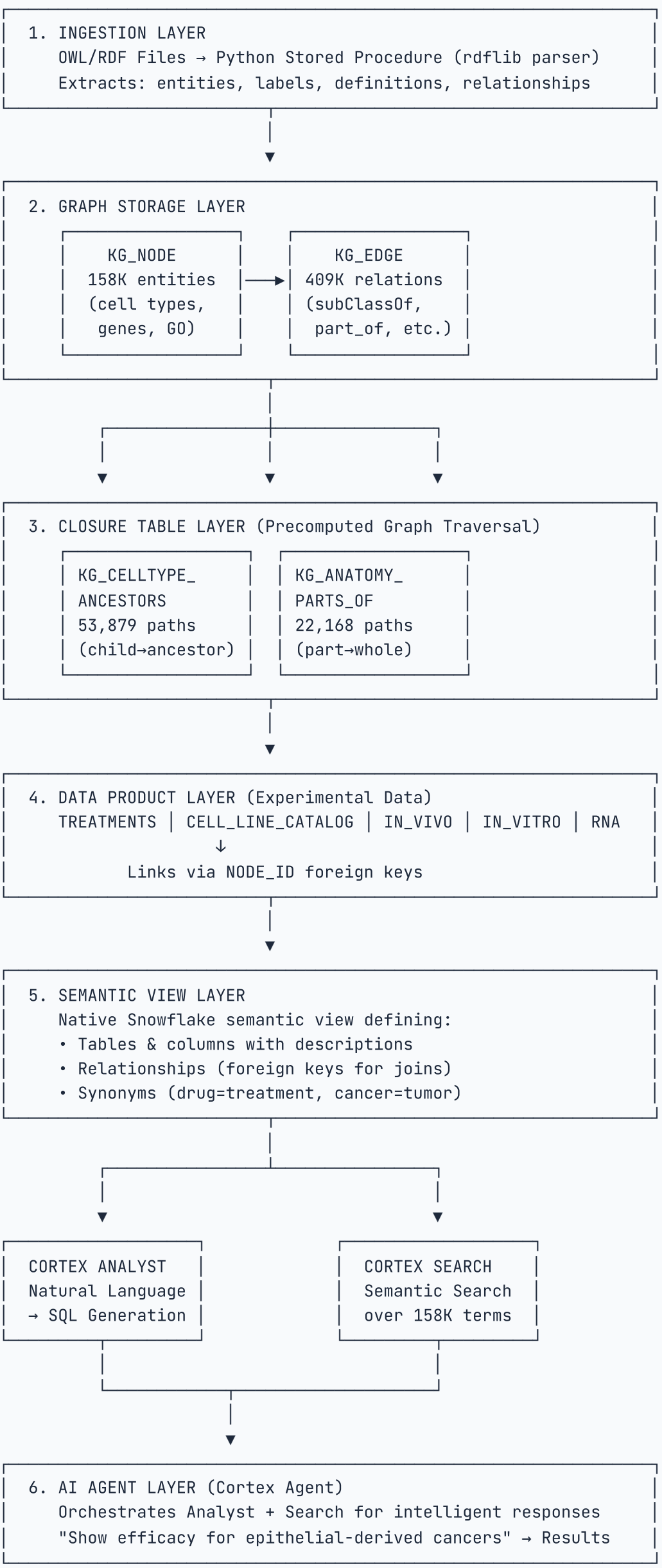
What We Built:

- Automated ontology ingestion from OWL/RDF files into a queryable graph structure
- Linkage between experimental data (drug efficacy, cell lines, gene expression) and ontology entities
- Precomputed graph traversal for real-time lineage-based queries
- Semantic search over 158,000+ ontology terms
- An AI agent that combines structured queries with semantic search to answer complex scientific questions

> **Key Result:** Scientists can now ask questions like *"Show drug efficacy for all epithelial-derived cancers"* and the system automatically discovers that hepatocytes, kidney cells, and pancreatic cells are all epithelial-derived—knowledge that exists only in the ontology, not in the experimental data.

How It Works

The Architecture Pipeline



How Each Component Satisfies the Use Case

Component	What It Does	Why It Matters
KG_NODE / KG_EDGE	Stores ontology entities and relationships	Enables biological reasoning (hepatocyte *is a* epithelial cell)
Closure Tables	Precomputes all ancestor/descendant paths	Makes graph traversal instant (no recursion at query time)
Data Products	Stores experimental results with NODE_ID links	Connects lab data to biological knowledge
Semantic View	Native Snowflake object with tables, relationships, synonyms	Enables natural language → SQL translation with proper joins
Cortex Search	Indexes ontology for semantic similarity	Finds relevant terms even with imprecise language
Cortex Agent	Orchestrates across all tools	Provides intelligent, context-aware responses

---

The Query Flow

When a scientist asks: *"Show drug efficacy for all epithelial-derived cancers"*

1. **Agent receives question** → Parses intent
2. **Agent may use Search** → Finds "epithelial cell" in ontology (semantic match)
3. **Agent uses Analyst** → Generates SQL that:
   - Queries closure table to find all epithelial descendants
   - Joins to cell line catalog via NODE_ID
   - Retrieves efficacy data for matching cell lines
4. **Results returned** → Includes liver (hepatocyte), kidney, pancreas, breast, lung models

> **The Key Insight:** Without the ontology, "Liver" and "Breast" would be unrelated categories. With the ontology, the system knows they share epithelial lineage and can analyze them together.

Validation Summary

We validated the architecture through:

1. **Ontology Ingestion** — Successfully loaded Cell Ontology and Gene Ontology (158K nodes, 409K edges)
2. **Graph Traversal** — Verified closure tables correctly capture transitive relationships
3. **Data Linkage** — Confirmed experimental data correctly references ontology nodes
4. **Query Accuracy** — Tested lineage-based queries return biologically correct results
5. **Natural Language** — Validated Cortex Analyst translates questions to correct SQL
6. **Semantic Search** — Confirmed ontology terms are discoverable by meaning

---

Scaling Considerations

Current PoC Scale

Asset	Count
Ontology Nodes	158,395
Ontology Edges	409,132
Cell Type Closure	53,879 rows
Anatomy Closure	22,168 rows
Data Products	~1,000 records

Production Scale Projections

The customer will likely have:

- **10-100x more ontologies** (Disease Ontology, ChEBI, UniProt, etc.)
- **1M+ experimental records** across all data products
- **10-50 concurrent users** querying the system

Scaling Strategy

Challenge	Solution	Snowflake Feature
Large Ontologies	Partition closure tables by NODE_TYPE	Clustering, Partitioning
Closure Table Size	Filter to relevant subsets; incremental updates	Materialized Views, Tasks
Query Performance	Pre-aggregate common lineage queries	Result Caching, Clustering
Search Latency	Cortex Search auto-scales with data	Managed Service
Concurrent Users	Scale warehouse independently	Multi-cluster Warehouses
Data Freshness	Scheduled refresh of closures	Snowflake Tasks

Performance Expectations at Scale

Scale	Closure Build Time	Query Latency	Notes
100K nodes	~10 min	< 1 sec	Current PoC
1M nodes	~2 hours	1-2 sec	With filtering
10M nodes	~8 hours	2-5 sec	Requires partitioning
100M nodes	Incremental only	5-10 sec	Full rebuild impractical

> **Key Insight:** Closure tables should be built incrementally and filtered to relevant subsets. Full transitive closure of 100M nodes would produce trillions of rows—unnecessary if only a fraction links to data products.

What Makes This Architecture Work

1. Unified Platform

Everything runs in Snowflake—no external graph database, no separate search engine, no custom ML infrastructure. This reduces complexity, cost, and data movement.

2. Precomputed Traversal

By materializing the transitive closure, we avoid expensive recursion at query time. The closure tables act as a "graph index."

3. Semantic + Structured

The combination of Cortex Search (finds concepts by meaning) and Cortex Analyst (queries data by structure) handles both fuzzy scientific language and precise data analysis.

4. Foreign Key Linkage

Experimental data links to ontology via simple foreign keys (NODE_ID). This is the bridge between lab results and biological knowledge.

5. AI Orchestration

The Cortex Agent decides when to search vs. query, how to combine results, and how to present answers—mimicking how a data scientist would approach the problem.

Conclusion

This proof-of-concept demonstrates that Snowflake can host a production-grade pharmaceutical knowledge graph that:

- ✓ **Ingests standard ontologies** (OWL/RDF) into a queryable structure
- ✓ **Links experimental data** to biological knowledge via foreign keys
- ✓ **Enables graph traversal** through precomputed closure tables
- ✓ **Supports natural language access** via Cortex Analyst semantic model
- ✓ **Provides semantic search** over ontology terms via Cortex Search
- ✓ **Orchestrates intelligently** via Cortex Agent
- ✓ **Scales to production** with established patterns (filtering, partitioning, incremental updates)

The architecture satisfies the Pfizer-style use case requirements and provides a foundation for enterprise-scale pharmaceutical R&D analytics.

Appendix: Assets Created

Ontology Layer

Asset	Type	Purpose
KG_NODE	Table	158,395 ontology entities (cell types, genes, anatomy)
KG_EDGE	Table	409,132 ontology relationships (subClassOf, part_of, etc.)
KG_CELLTYPE_ANCESTORS	Table	Precomputed cell type closure for instant traversal
KG_ANATOMY_PARTS_OF	Table	Precomputed anatomy closure for part_of queries

Data Product Layer

Asset	Type	Purpose
TREATMENTS	Table	Drug registry with gene targets and mechanisms
CELL_LINE_CATALOG	Table	Cell line/model catalog with ontology links
IN_VIVO_STUDIES	Table	In vivo efficacy studies (TGI, survival)
IN_VITRO_STUDIES	Table	In vitro efficacy studies (concentration, response)
RNA_SEQ_RESULTS	Table	Gene expression data

AI Services Layer

Asset	Type	Purpose
PHARMA_KG_SEMANTIC_VIEW	Semantic View	Native Snowflake semantic view for Cortex Analyst
ONTOLOGY_SEARCH_SERVICE	Cortex Search	Semantic search over 158K ontology terms
PHARMA_KG_AGENT	Cortex Agent	Orchestrates Analyst and Search for intelligent responses

---

Sample Questions That Demonstrate Graph Traversal

Question	What It Tests
"Show drug efficacy for all epithelial-derived cancers"	Cell lineage traversal + efficacy data
"Which drugs have the best TGI in PDX models?"	Data product filtering
"Find all cell lines derived from leukocytes"	Ancestry query via closure table
"What is a hepatocyte?"	Ontology search
"Compare ADC efficacy to kinase inhibitors"	Cross-drug class analysis
"What genes are targeted by our breast cancer treatments?"	Gene-drug-disease linking

---

Document generated from validated Snowflake environment: RDF_ONTOLOGY_POC

Semantic View: PHARMA_KG_SEMANTIC_VIEW | Search Service: ONTOLOGY_SEARCH_SERVICE | Agent: PHARMA_KG_AGENT