Constructing an Enriched Clinical Knowledge Graph: Transforming EHR Data to OMOP and Modeling in Neo4j Graph Database

Thejas Bharadwaj, Eshna Sengupta
Author, Supervisor

Background

Electronic Health Records (EHR) data presents challenges due to its complex and diverse sources and lack of standardization. The OMOP Common Data Model (CDM) provides a solution by standardizing and harmonizing healthcare data, thus facilitating interoperability and cross-study analysis. This project explores how the capabilities of graph databases can be leveraged to enhance the use and understanding of EHR data, particularly when transformed into the OMOP CDM. By integrating the OMOP CDM into a Neo4j graph database, this project aims to improve data modeling and analytics in healthcare.

Graph databases are well-suited for modeling and querying complex, relationship-centric data. They provide flexibility in data modeling and can handle complex data traversals efficiently, making them ideal for healthcare applications where patient data, diagnoses, treatments, and outcomes are interrelated.

Methods

The project involved several steps to construct an enriched clinical knowledge graph:

1. **Data Source and Transformation**:
   - **MIMIC-III Dataset**: The MIMIC-III dataset, which includes comprehensive clinical data from ICU patients, was used as the primary data source.
   - **OMOP CDM**: MIMIC-III data was converted to the OMOP CDM to standardize the data and facilitate integration with other datasets.
   - The conversion process involved mapping MIMIC-III's relational data into the OMOP CDM structure.
2. **Graph Database Modeling**:
   - **Neo4j**: The transformed OMOP data was then modeled into a Neo4j graph database. This involved creating nodes and relationships that represent patients, observations, conditions, treatments, and visits.
   - **Graph Data Model**: The graph data model was designed to represent clinical events and relationships effectively, enabling complex queries and analyses that are difficult with traditional relational databases.
3. **Analytic Use Cases**:
   - **Patient Journey Mapping**: Mapping patient journeys through different healthcare events to visualize and analyze their medical history and treatments.
   - **Community Detection**: Using graph algorithms to identify clusters of patients with similar clinical journeys, which can help in understanding disease patterns and outcomes.
   - **Comorbidities Analysis**: Extracting and analyzing common comorbidities associated with specific conditions to support clinical research and decision-making.

Results

The Neo4j graph database successfully represented the OMOP CDM-transformed MIMIC-III data, providing a flexible and powerful platform for healthcare data analysis.

1. **Graph Statistics**:
   - **Nodes**: The database included various node types such as Condition Occurrence, Procedure Occurrence, Drug Exposure, Observation Period, Person, and Visit Occurrence, totaling **177,795 nodes.**
   - **Relationships**: The graph had a complex web of relationships, including ASSOCIATED_DURING_VISIT, HAS_PROCEDURE_OCCURRENCE, HAS_DRUG_EXPOSURE, and more, with a total of **9,765,849 relationships**.
2. **Patient Journey Mapping**:
   - The graph model allowed detailed mapping of patient journeys, including sequences of conditions, treatments, and observations. For example, the journey of a pancreatic cancer patient was mapped, highlighting interactions with healthcare services over time.

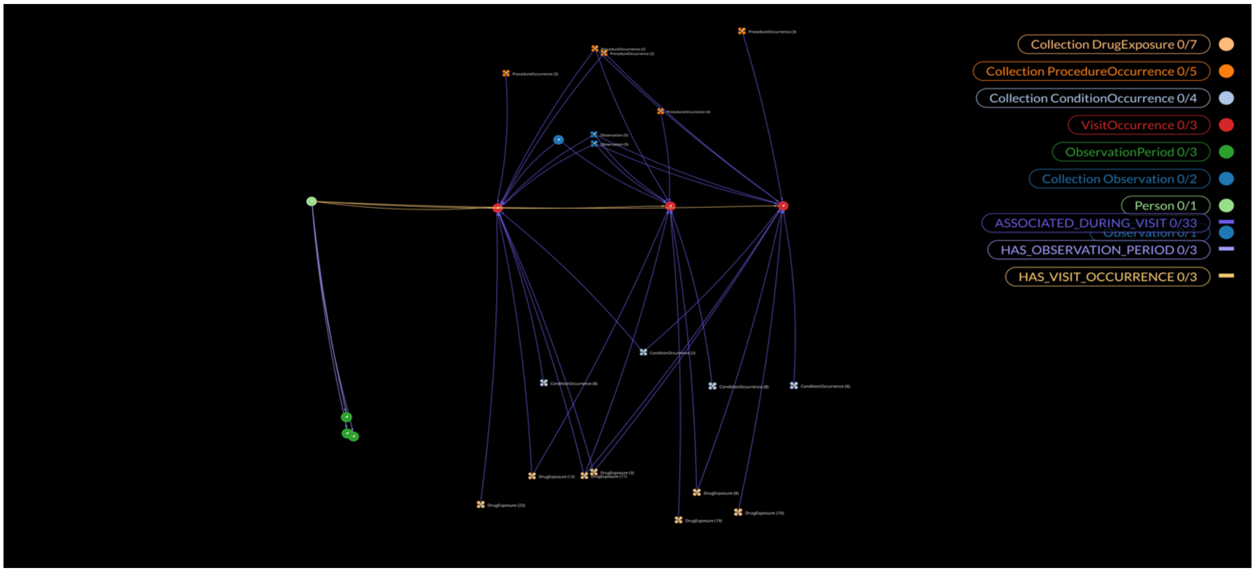


Figure 1. Patient journey who had the pancreatic cancer ICD code (ICD 9: 157.*, ICD 10: C25.*) through observations, drugs and visits

3. **Community Detection**:
   - Using algorithms like label propagation, communities of patients with similar health journeys were identified, providing insights into disease progression and potential areas for targeted interventions.

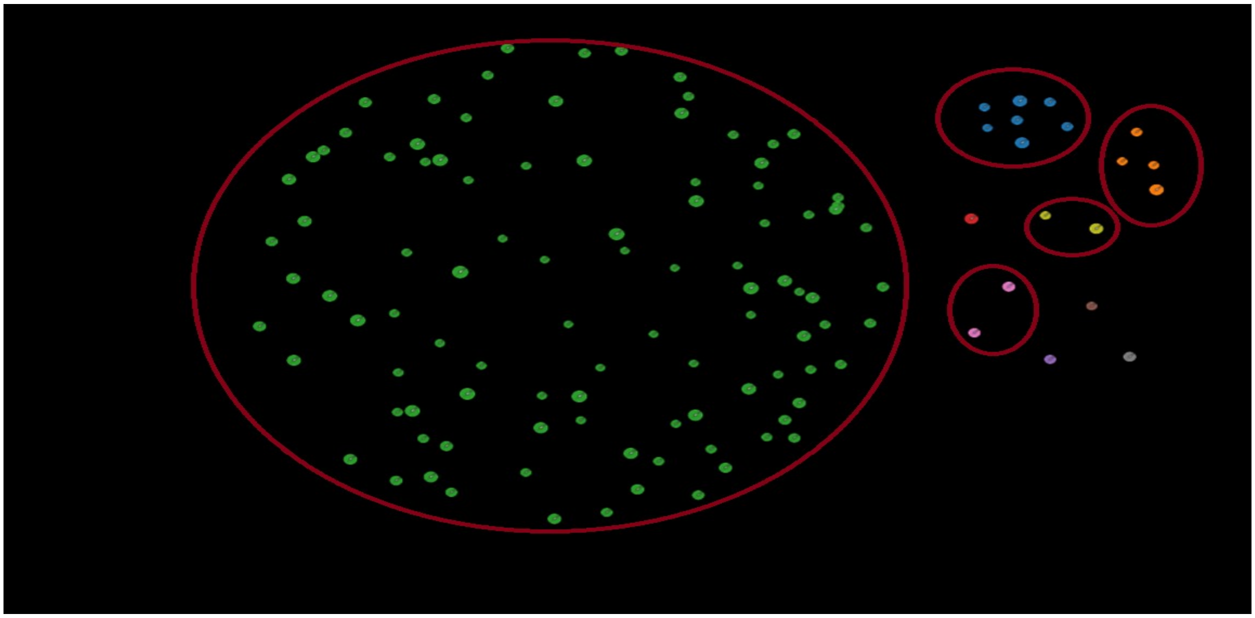
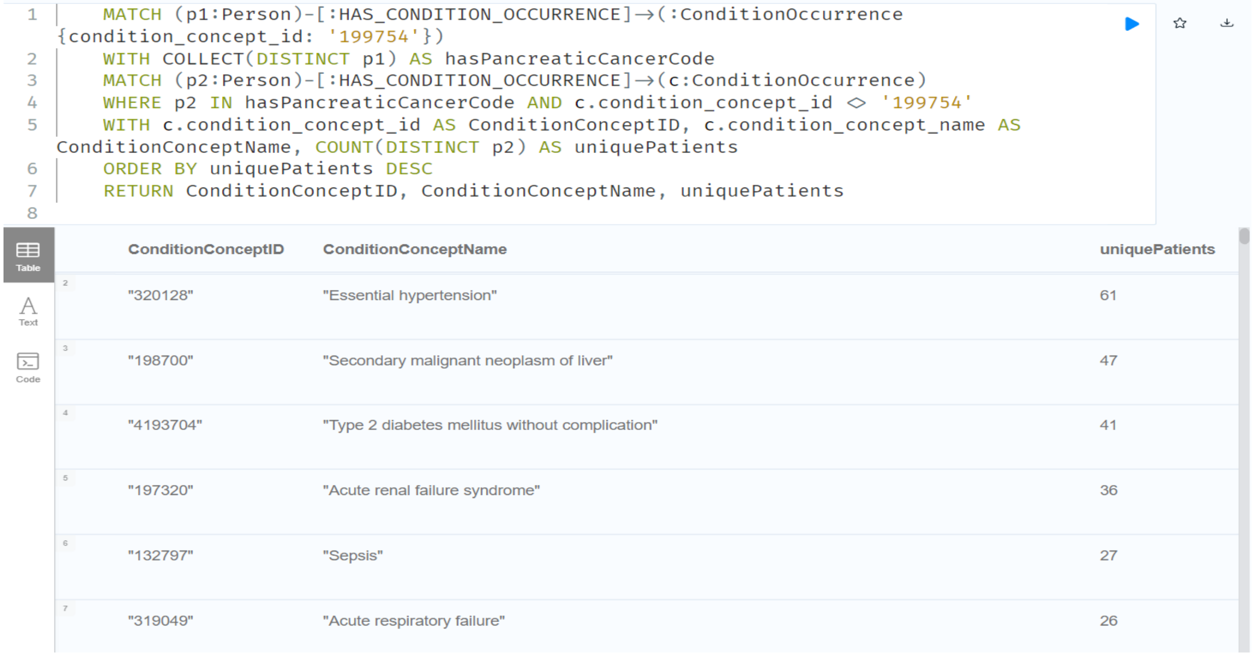


Figure 2. Community detection for patients with similar journeys

4. **Comorbidities Analysis**:
    - The graph model facilitated the extraction of commonly associated comorbidities, helping to understand the relationships between various health conditions and aiding in clinical decision support.



```
    MATCH (p1:Person)-[:HAS_CONDITION_OCCURRENCE]->(:ConditionOccurrence
{condition_concept_id: '199754'})
    WITH COLLECT(DISTINCT p1) AS hasPancreaticCancerCode
    MATCH (p2:Person)-[:HAS_CONDITION_OCCURRENCE]->(c:ConditionOccurrence)
    WHERE p2 IN hasPancreaticCancerCode AND c.condition_concept_id <> '199754'
    WITH c.condition_concept_id AS ConditionConceptID, c.condition_concept_name AS
ConditionConceptName, COUNT(DISTINCT p2) AS uniquePatients
    ORDER BY uniquePatients DESC
    RETURN ConditionConceptID, ConditionConceptName, uniquePatients
```

ConditionConceptID	ConditionConceptName	uniquePatients
"320128"	"Essential hypertension"	61
"198700"	"Secondary malignant neoplasm of liver"	47
"4193704"	"Type 2 diabetes mellitus without complication"	41
"197320"	"Acute renal failure syndrome"	36
"132797"	"Sepsis"	27
"319049"	"Acute respiratory failure"	26

Figure 3. Comorbidities analysis of Diabetic patients

Conclusion

This project demonstrated the effectiveness of using a graph database to enhance the representation and analysis of healthcare data transformed into the OMOP CDM. The Neo4j graph model provided a robust platform for complex data queries and analyses, supporting various healthcare use cases such as patient journey mapping, community detection, and comorbidities analysis.

Future work includes expanding the graph model to incorporate additional data sources and relationships, enhancing NLP capabilities, and integrating with advanced analytics platforms like VertexAI.

References

1. **Johnson, Alistair E. W., Tom J. Pollard, Lu Shen, Leo W. H. Lehman, Matthieu Feng, Marzyeh Ghassemi, et al.** "MIMIC-III, a freely accessible critical care database." *Scientific Data*, vol. 3, 2016, pp. 1-9. Available from: https://doi.org/10.1038/sdata.2016.35.
2. **NUSCRIPT.** "OMOP to Graph." *GitHub Repository*, NUSCRIPT, Accessed 14 June 2024. Available from: https://github.com/NUSCRIPT/OMOP_to_Graph.