RE06: Unlocking Healthcare Insights:

Graph-Based Modeling & Analysis of Synthetic Patient Data

George Panagiotakis – Georgios.Panagiotakis@pfizer.com
Nik Kountouris – Nikolaos.Kountouris@pfizer.com
Maria Puri – Maria.Puri@pfizer.com

PHUSE EU CONNECT – NOVEMBER 2025

Who we are?

- **Nik Kountouris** joined Pfizer in June 2023 as a Senior Manager and has been leading the RWD & PRO teams.
- Nik holds a BSc in Applied Mathematics, a MSc in Financial Mathematics and a MSc in Big Data Analytics.
- Nik has gained significant experience in data analytics and risk management through his work in the banking, consulting and technology industries in the UK.
- Prior to Pfizer, Nik headed the Data Science & BI teams across EMEA at Citrix.

- **George Panagiotakis** joined Pfizer in June 2023 as a Senior Associate and has been supporting the RWE Internal Medicine team.
- George holds a BSc & MSc in Computer Science.
- He has experience with machine/deep learning techniques, software development and data analytics.

Agenda

1. Problem Statement
2. Introduction To Real-World Data (RWD)
3. Introduction to Synthetic Data
    - Using Synthea
    - Challenges of Synthetic Data
4. Introduction To Graphs & Graph Modeling
5. Practical Examples
6. Key Takeaways

Problem Statement

The Challenge: Capturing Complex Patient Journeys

- **Patient journeys are not linear** but rather a network of information.

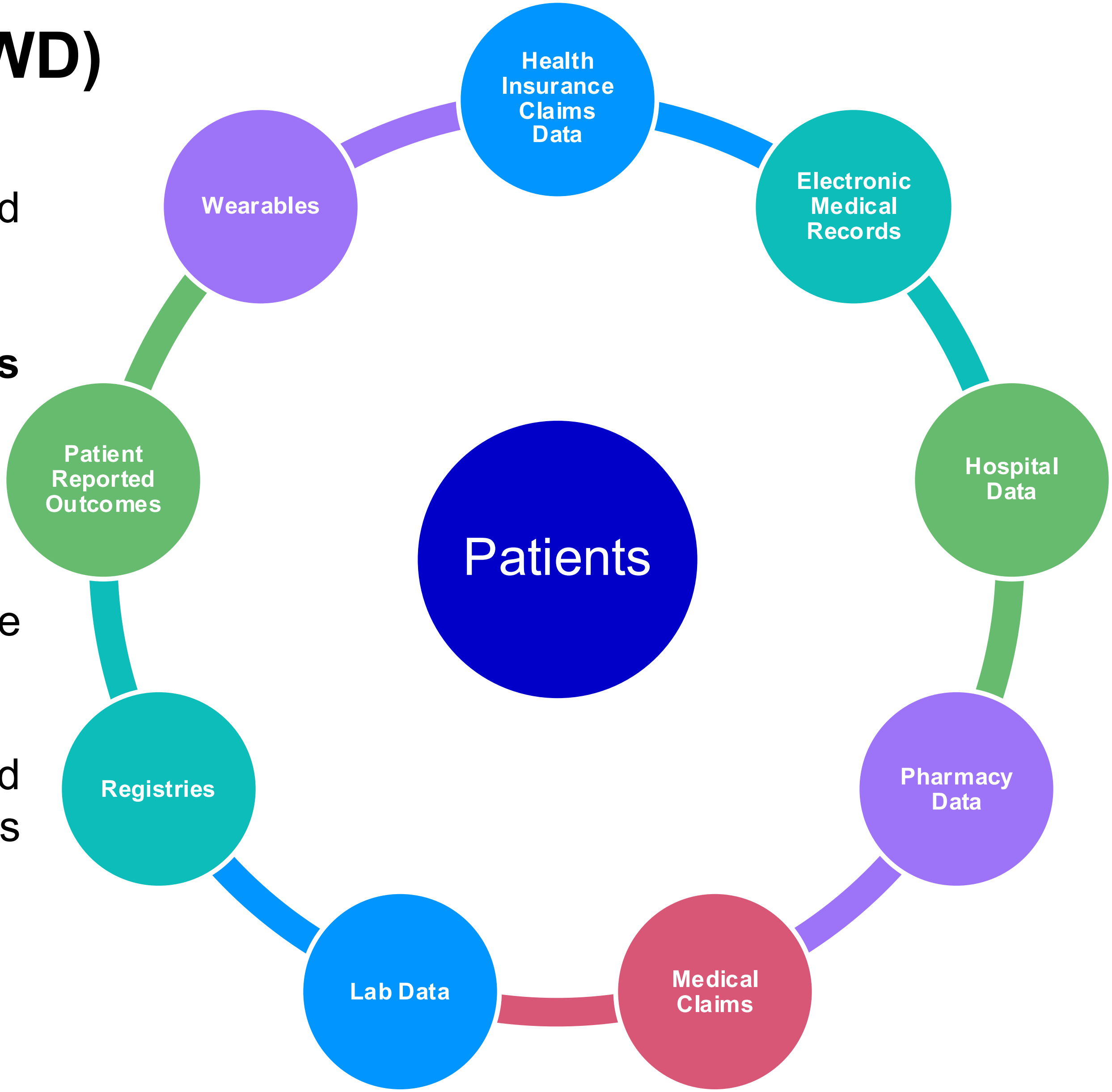
- **Relational Models** transforms this network into rigid tables losing meaning in the way.

- **Graph models**, **mirror the real structure** of healthcare data, preserving relationships and **unlocking deeper insights.**

Introduction to Real – World Data (RWD)

Introduction To Real-World Data (RWD)

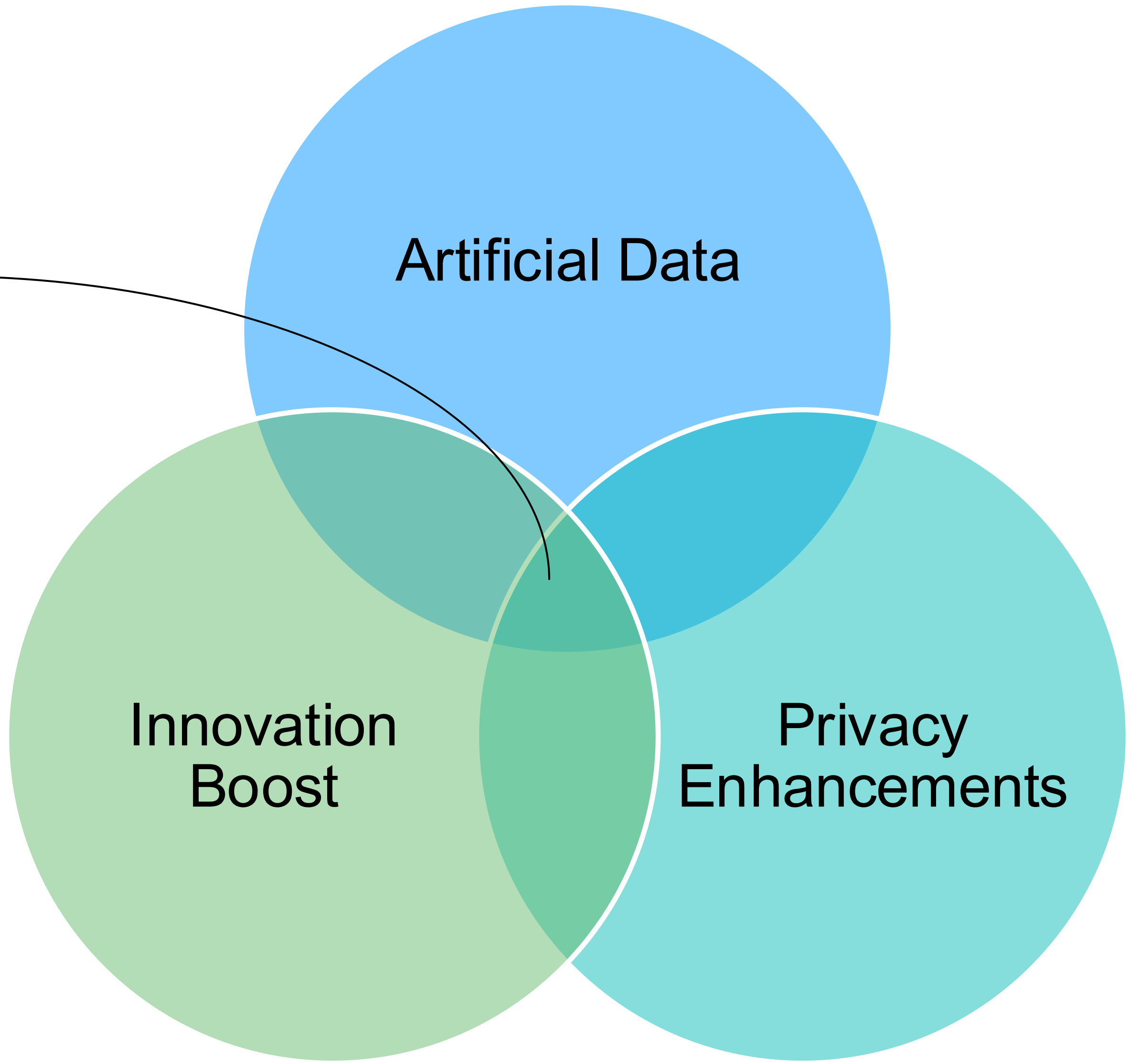
- **Real-World Data (RWD)** refers to health-related information gathered from different sources.
- Interest grew due to **Randomized Clinical Trials (RCT)** limitations along with tech advancements.
- Enables pragmatic insights on real-world settings.
- **Real-World Evidence (RWE)** is the clinical evidence derived from analyzing **RWD.**
- **Complements** clinical trial data with real-world **effectiveness**, **safety** and **utilization**, and enables pragmatic insights on real-world settings.



Introduction to Synthetic Data

Introduction To Synthetic Data

- **Synthetic data** refers to **artificially generated** information.
- Intended to mirror the features, correlations, and statistical patterns present in real-world datasets.
- Eliminated the **risk of compromising** real patient information.
- Encompasses clinical or biological information (e.g. Patient demographics, lab results, EEG etc).
- Synthetic data accelerates **innovation** & has empowered researchers & organizations to **explore new frontiers.**



Using Synthea

- **Synthea** is a rule-based synthetic data generator.
- Generates high-quality, clinically realistic but entirely **FAKE** patient data.
- Is **free** of cost and **privacy restrictions** to support research.
- It models conditions as modules, depicting a flowchart of clinical states and transitions.
- Supports multiple conditions such as atrial fibrillation, diabetes, breast cancer etc.
- Outputs in different formats such as **FHIR** and **CSV**.

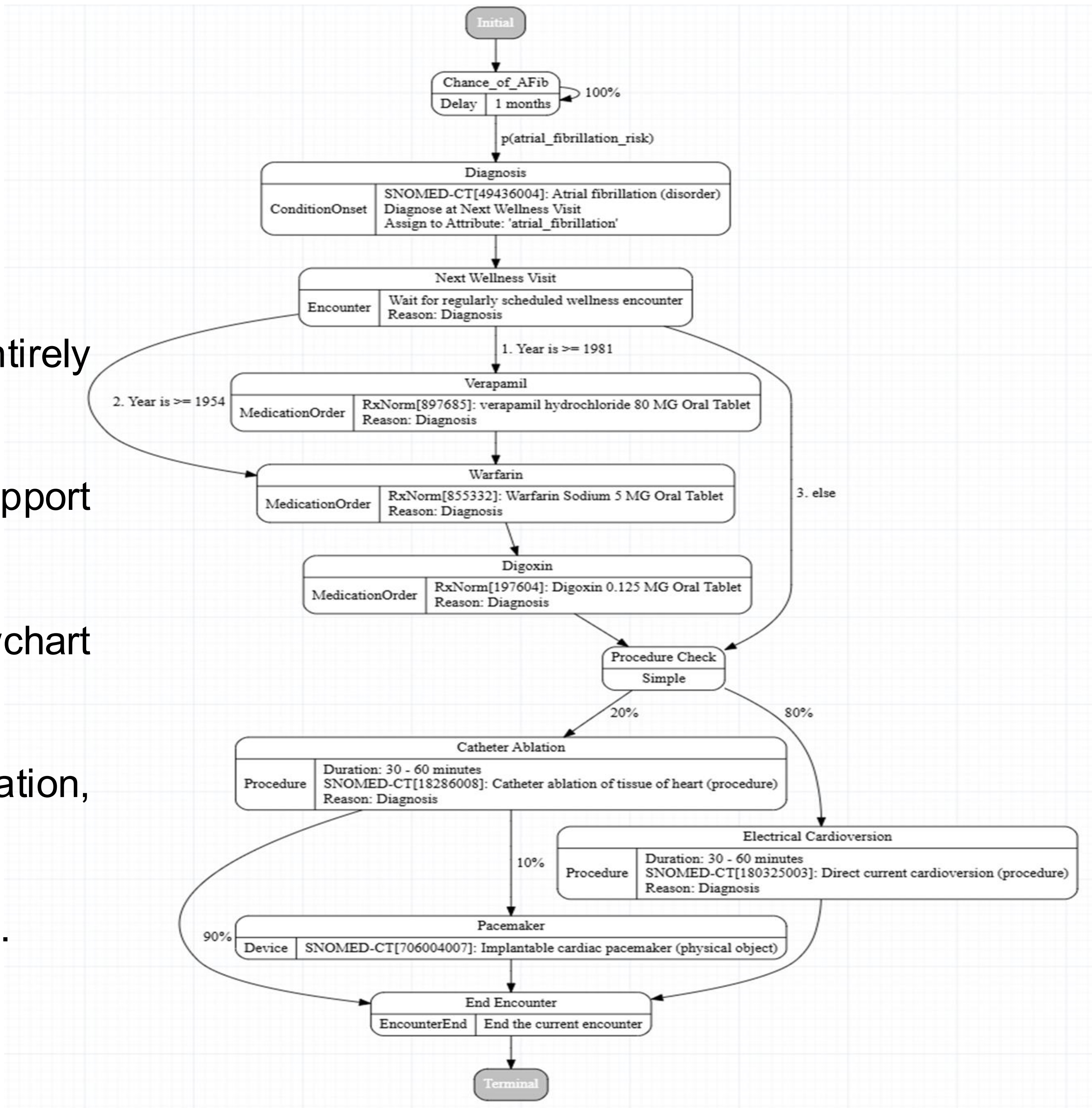


Figure 1: Overview of Atrial Fibrilation module from Synthea
Synthea Generic Module Builder accessed 10/1/2025

Challenges of Synthetic Data

Fairness & Bias

- Concerns on biases existence due to existence on original data (e.g. gender disparities).
- Such biases can be amplified in synthetic data, leading to distorted.

Data Utility & Privacy

- Concerns on re-identifying real patient information if closely resembling real data.

Fidelity

- **Fidelity** refers to how the synthetic data preserve statistical patterns of the real data.

Introduction to Graphs & Graph Modeling

Introduction To Graphs

What is a graph?

- Graphs and graph theory originated as a branch of discrete mathematics.
- Graphs are **schema-free** and **flexible**.
- They consist of nodes and edges reflecting entities and their relationships.
- Graphs can be directed, undirected, weighted, unweighted, heterogeneous, homogeneous, include loops and other.
- Graphs model many **real-world systems**, for instance, a healthcare network with patients, providers and their interactions.

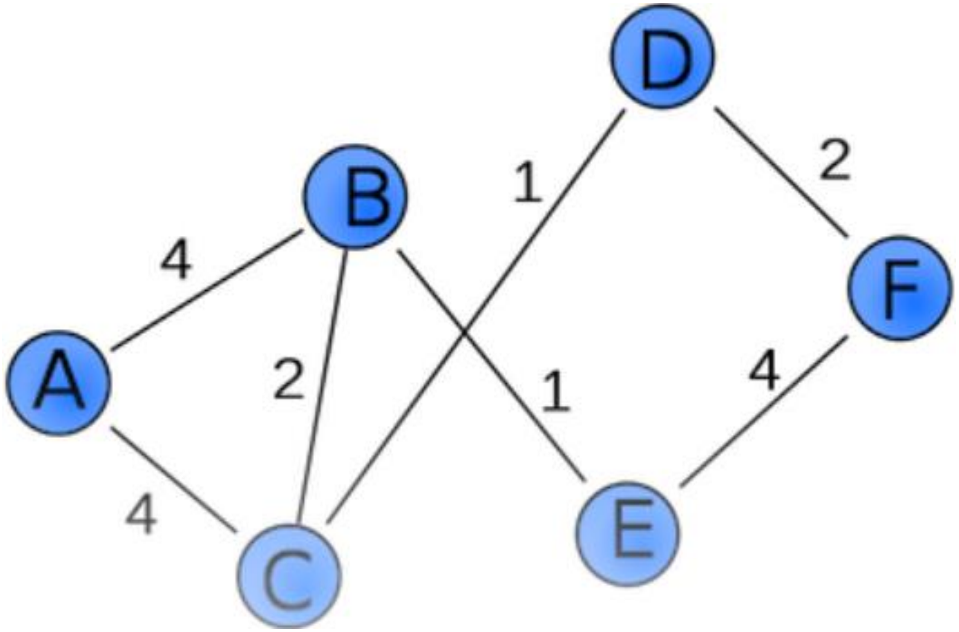


Figure 2: Illustration of a weighted graph, [Directed vs. Undirected Graphs | Overview, Examples & Algorithms - Lesson | Study.com] accessed 10/1/2025

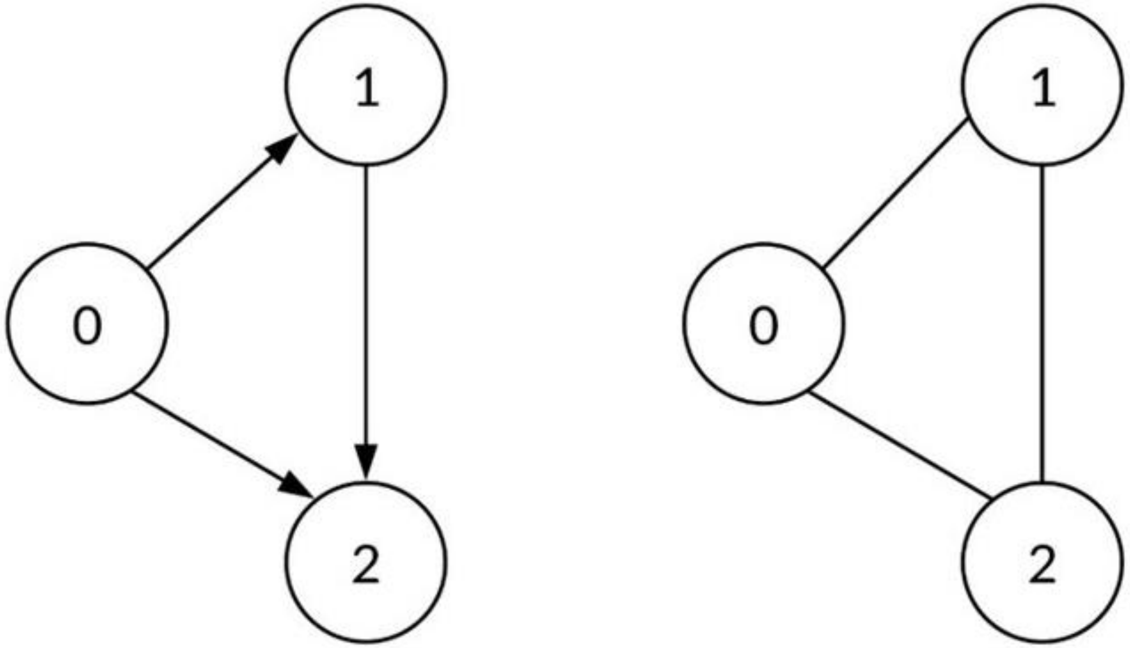


Figure 4: Overview of Directed and undirected graphs, [Graphs UMPIRE Cheat Sheet | CodePath Cliffnotes] accessed 10/1/2025

Introduction To Graph Data Science

What is Graph data science?

- ❑ Use of graph-based algorithms about analysis of data represented as graphs.
- ❑ Used for uncovering patterns, clustering and supporting decision-making.
- ❑ Different categories:
  - ❑ **Centrality Algorithms:** Measure importance in a network.
  - ❑ **Community Detection:** Finding groups or densely connected vertices.
  - ❑ **Path-Finding:** Searching for optimal paths in a graph.
  - ❑ **Similarity algorithms:** How similar nodes are based on their neighborhoods.

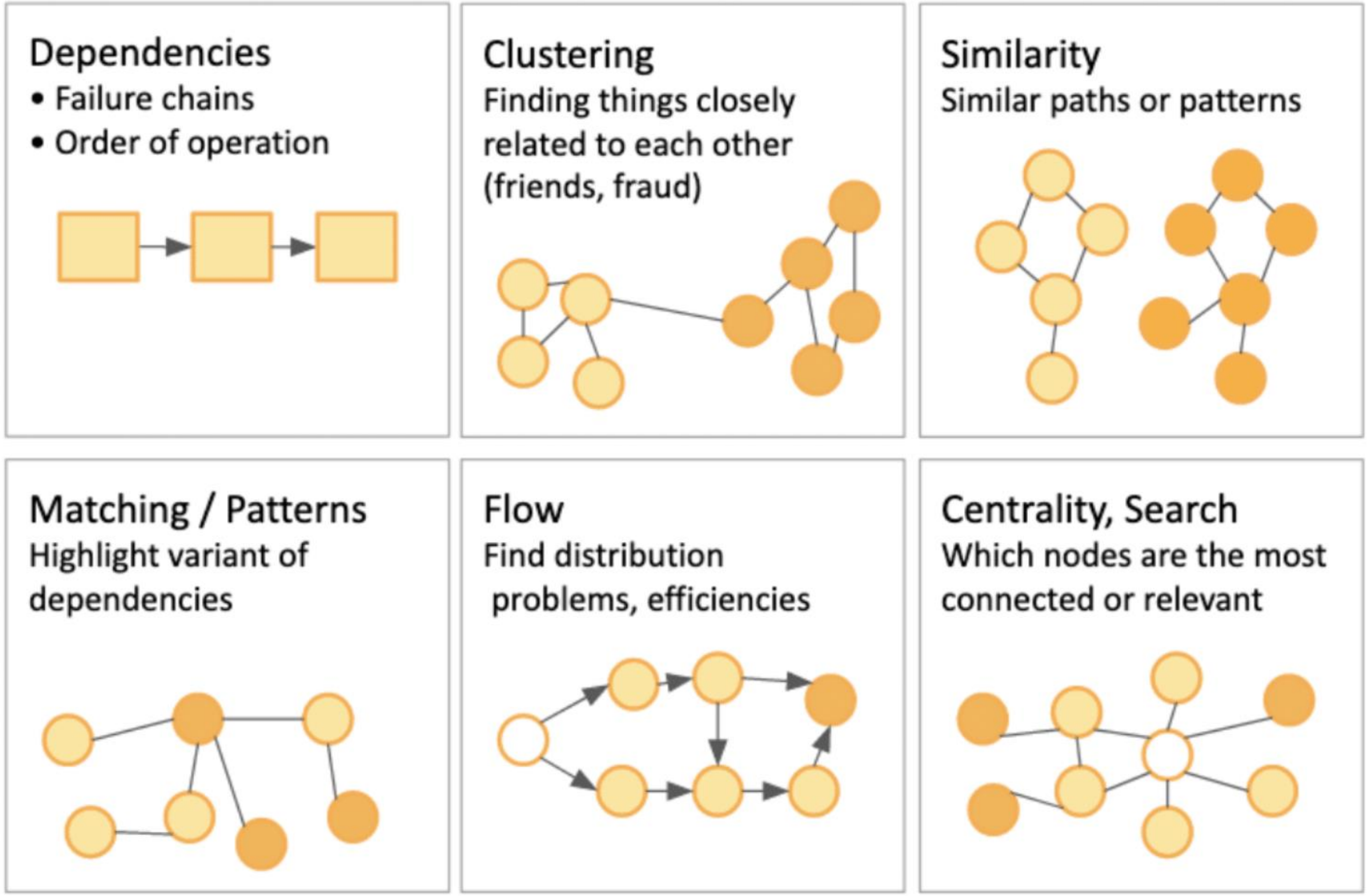


Figure 4: Different categories of graph algorithms, [Data Science with Graphs - using knowledge graphs on the data before it reaches the ML phase - deepsense.ai] accessed 10/1/2025

Introduction to Graph Databases

What is a graph database?

- ❑ Graph databases store information in graph structures (such as Neo4j, Amazon Neptune) .
- ❑ Represent data as a **property graph** model or **RDF** models.
- ❑ Treat relationships as first-class citizens, eliminating the need of costly **join** operations.
- ❑ Support specialized query languages for navigating and pattern-matching within graphs, such as **Cypher** for **Neo4j**.

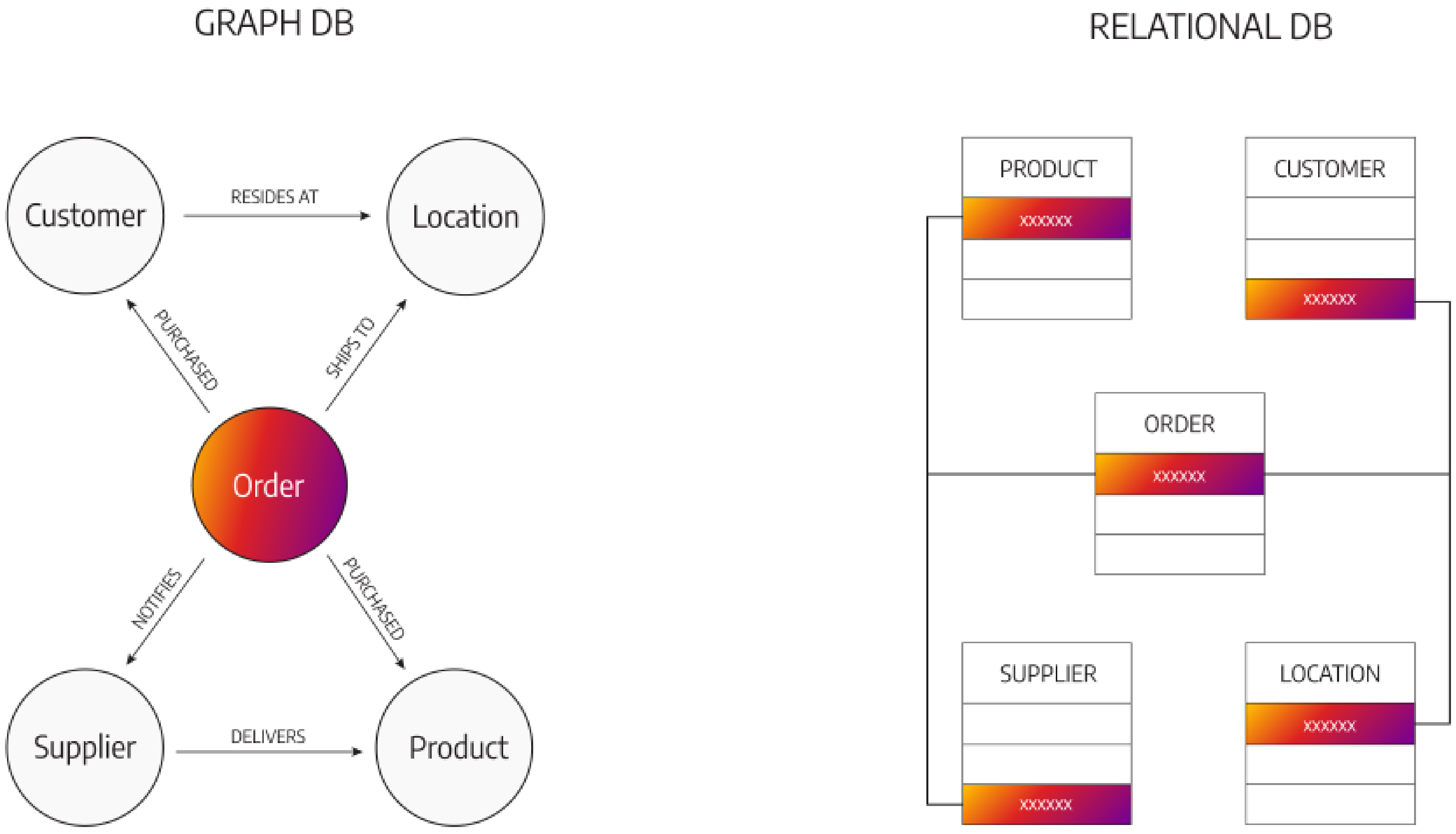


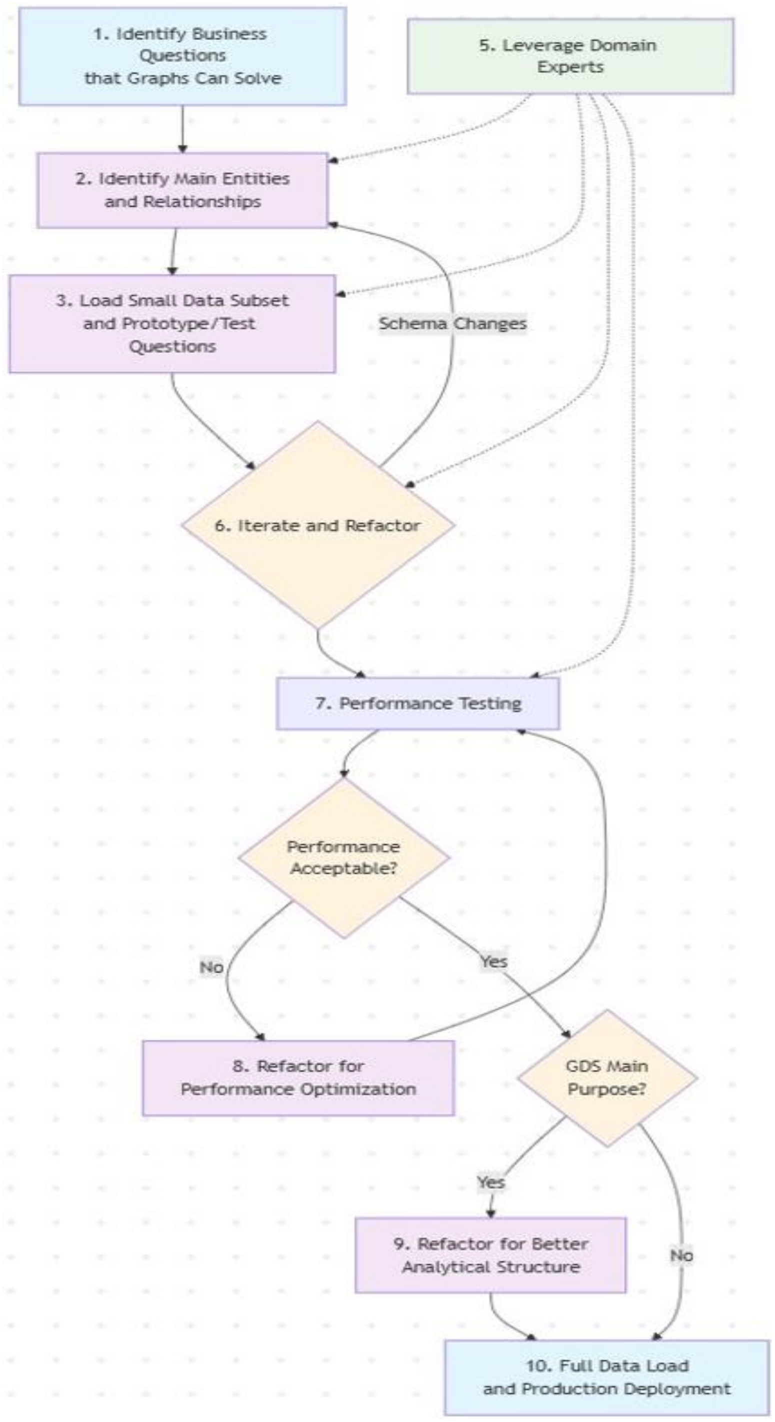
Figure 5: Graph Database overview, [What is a Graph Database?](#) accessed 10/1/2025

Introduction to Graph Modelling

Graph Modeling: A Blend of Art and Science

Key steps:

1. Identify business questions **best addressed by graphs** and let them define the schema. Examples include:
   - ✓ *Which providers serve as connectors/bridges?*
   - ✓ *What patient clusters emerge naturally?*
2. Identify **main entities** and **relationships that represent business logic.**
3. Load **small sample** of the data and **test the questions.**
4. **Iterate and optimize** the schema and its performance **as more data is added.**
5. Engage with **domain experts** for insights and validation from early steps.
6. If **focusing on Graph Data Science**, **restructure** the schema to enhance efficiency and usability.



Practical Examples

Before we continue

In the following slides, we will demonstrate a series of practical tasks designed to illustrate the application of graph-based modeling in healthcare data.

1. **Task 1: Model** a set of synthetic data **(~100 patients**) into a **graph structure**.

2. **Task 2: Define** a graph schema and **compare** performance between **GDB** and **RDB** on a larger dataset **(~80,000 patients**).

3. **Task 3: Showcase** some **Graph Data Science (GDS)** algorithms for extracting meaningful insights.

Practical Examples – Task 1

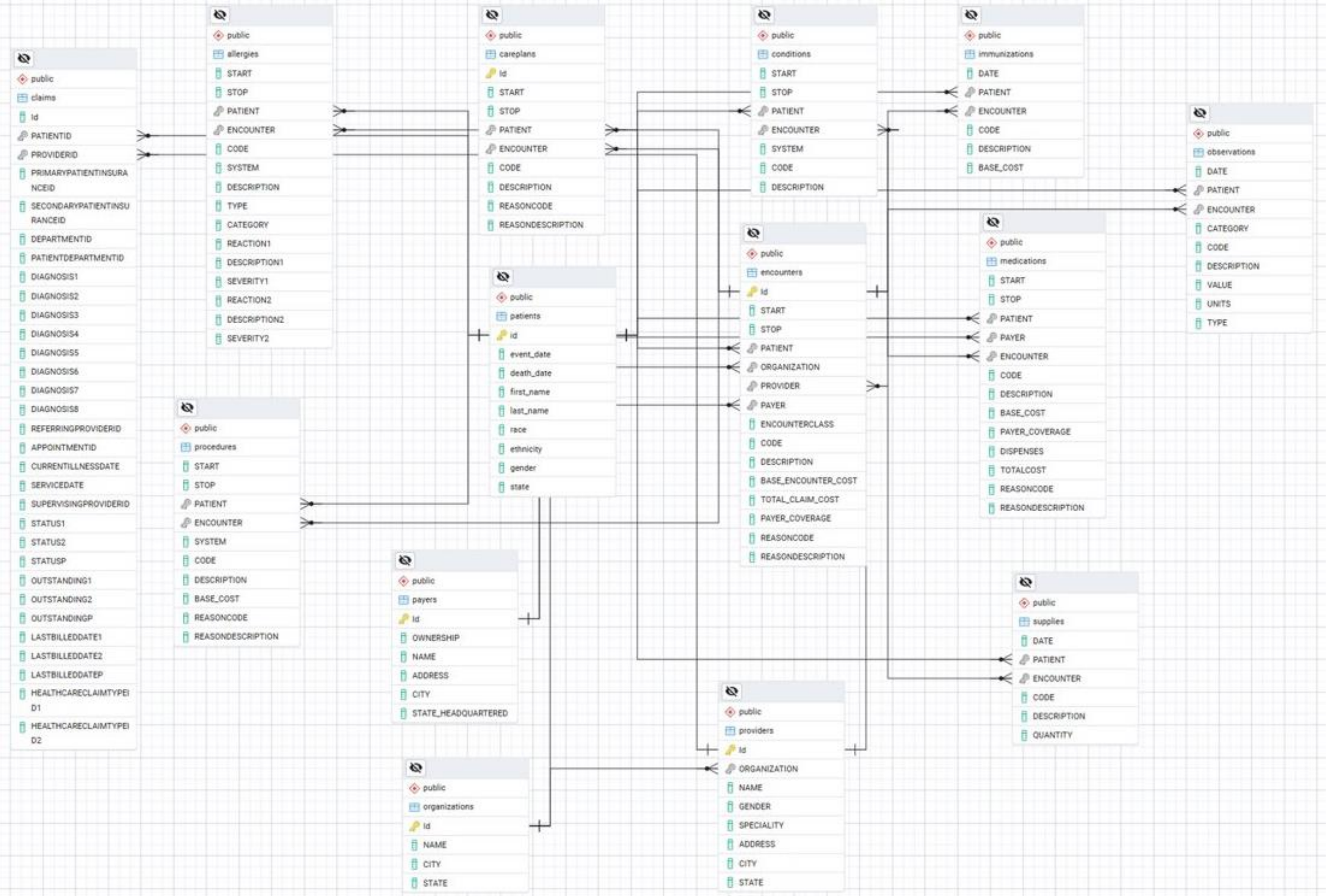
Data generation

- Generated data for ~100 patients using **Synthea**
- Output is in **CSV** format.
- The generated data resemble real-world healthcare databases.

```
Patients Table Shape ->  (127, 10)
Payers Table Shape ->  (10, 22)
Providers Table Shape ->  (289, 13)
Organizations Table Shape ->  (289, 11)
Encounters Table Shape ->  (11083, 15)
Careplans Table Shape ->  (549, 9)
Conditions Table Shape ->  (6374, 7)
Immunizations Table Shape ->  (2644, 6)
Medications Table Shape ->  (8539, 13)
Observations Table Shape ->  (144724, 9)
Procedures Table Shape ->  (29819, 10)
Claims Table Shape ->  (19622, 31)
Allergies Table Shape ->  (143, 15)
Symptoms Table Shape ->  (31179, 9)
Supplies Table Shape ->  (4849, 6)
```

```
'allergies.csv',
'careplans.csv',
'claims.csv',
'claims_transactions.csv'
'conditions.csv',
'devices.csv',
'encounters.csv',
'imaging_studies.csv',
'immunizations.csv',
'medications.csv',
'observations.csv',
'organizations.csv',
'patients.csv',
'payers.csv',
'payer_transitions.csv',
'procedures.csv',
'providers.csv',
'supplies.csv']
```

Overview of the relational Schema



Step 1: Load data as initial graph

When transforming from relational into a graph format:

1) **De-normalize** the relational database to a lower normal form.

2) Represent **table names as labels.**

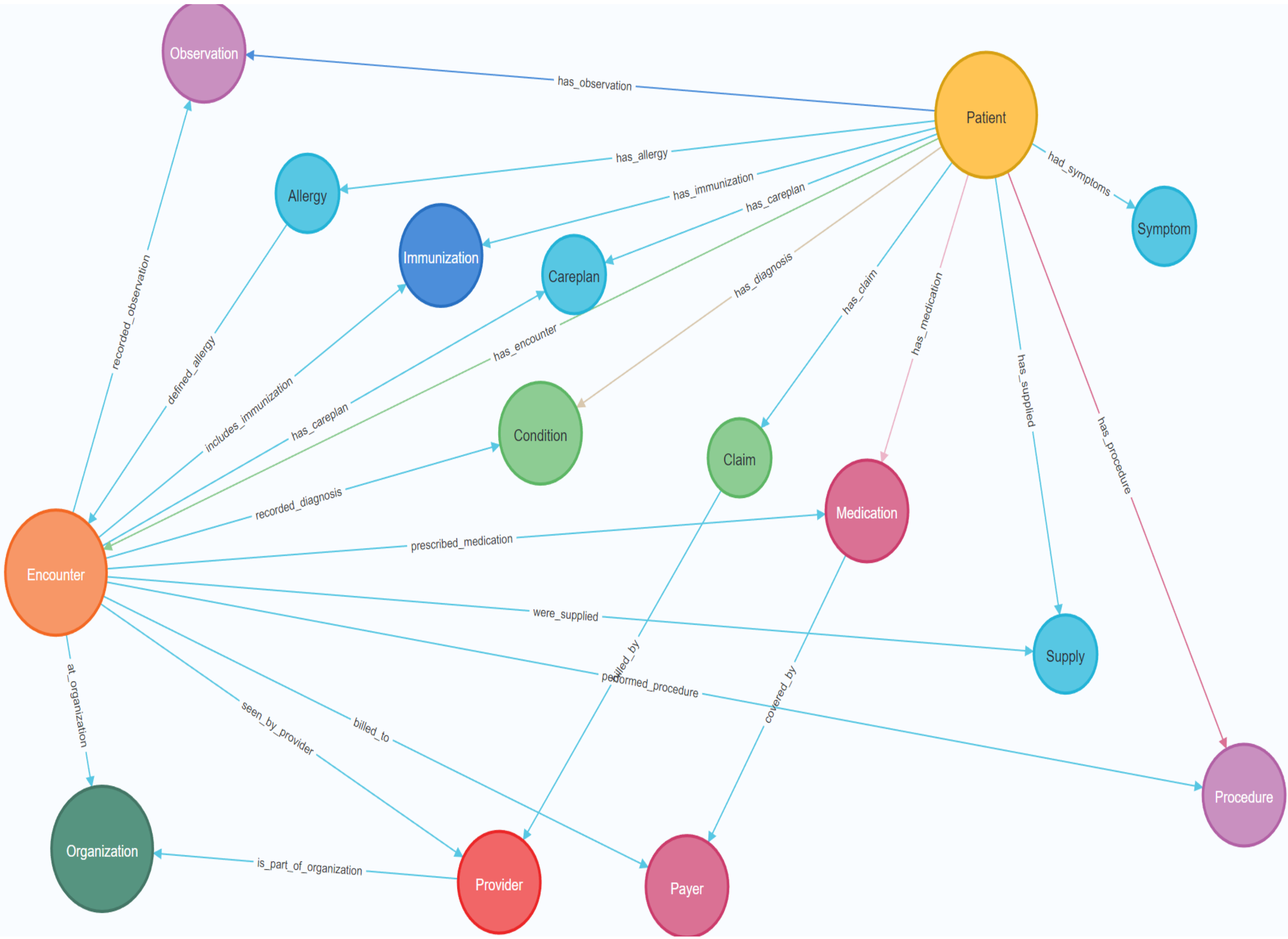
3) Convert each **row into a unique node.**

4) Translate **joins between tables into relationships**.

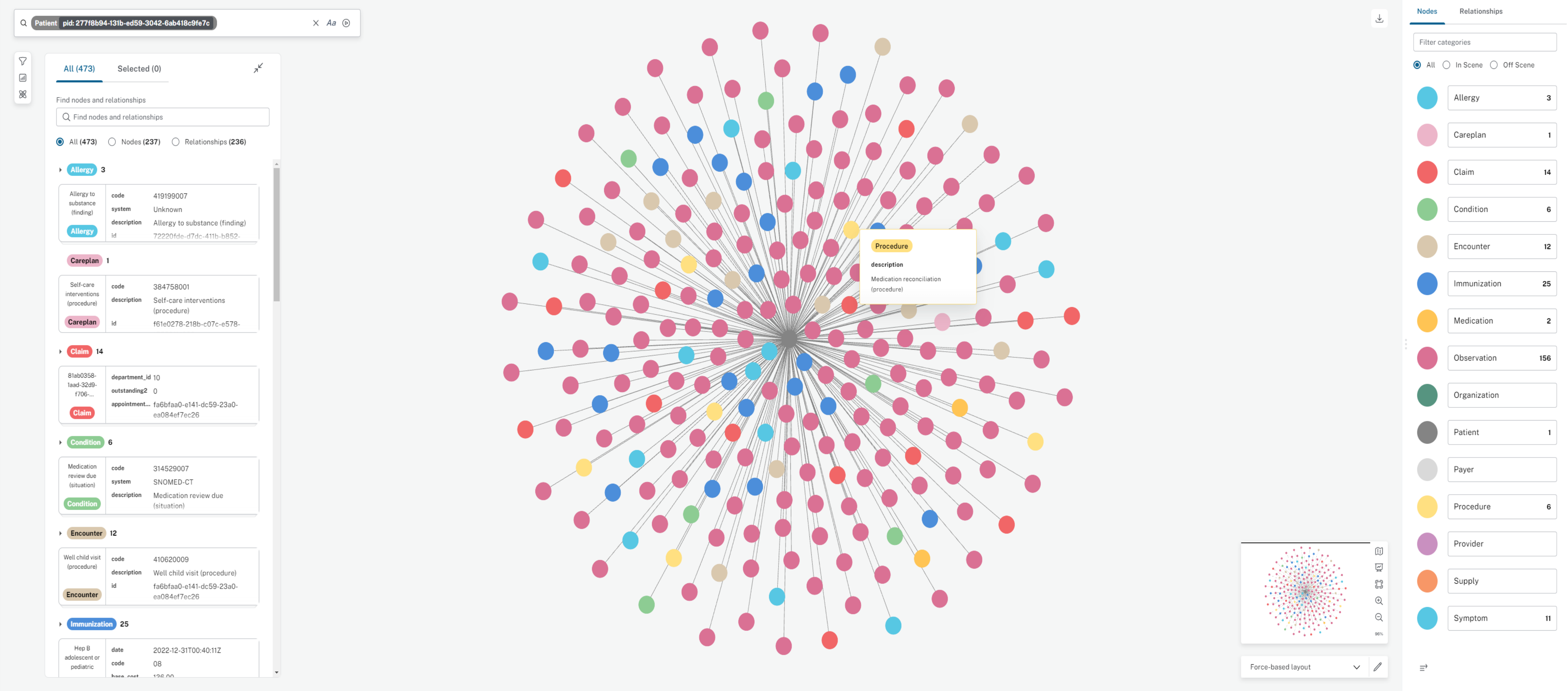
Meta-graph consists of:

- **Entities:** ~260,240
- **Relationships:** ~505,000

Note: Denormalization is commonly used in analytical workflows for relational systems as it minimizes the number of join operations required during query execution.



Visualizing a single Patient



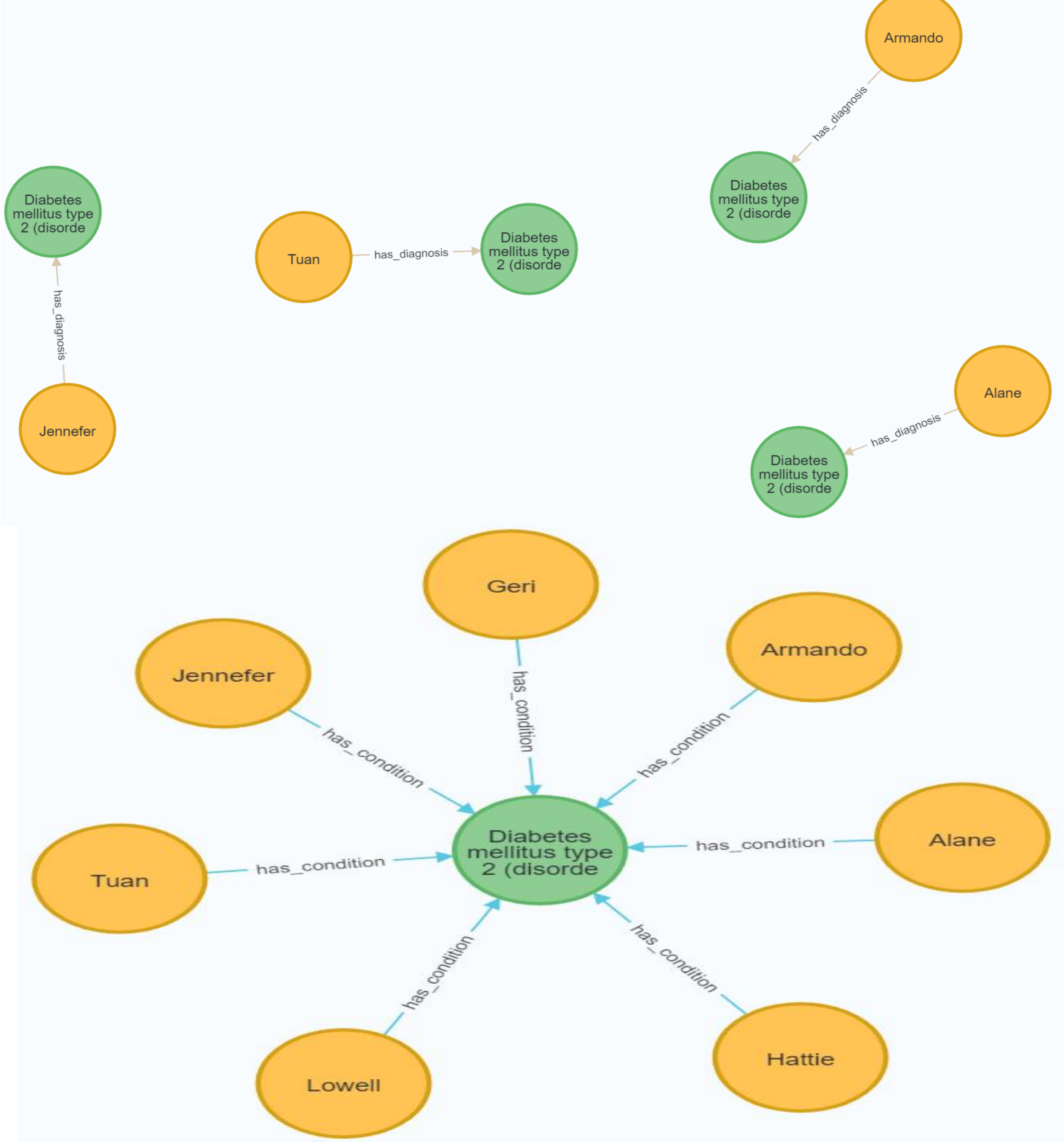
Step 2: Normalizing nodes

Graphs are particularly effective at interconnecting information.

- **Objective: i**dentify shared elements across patients.
- **Initial graph structure:** Each patient was linked exclusively to their own data.
- **Approach:** Applying **entity normalization** by deduplicating concept.
- All event-level relationships were **preserved**.

Resulting meta-graph remains the same but:

- **Entities:** ~64,179 (approximately 75% fewer than the original)
- **Relationships:** ~505,000



Step 3: Eliminate Redundancy

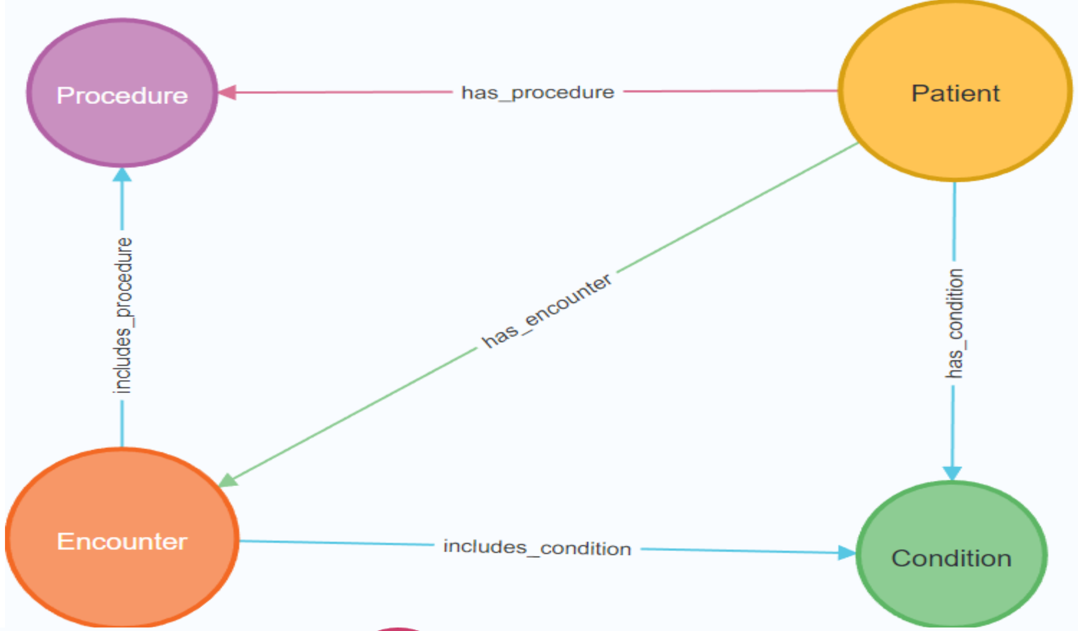
Multiple connection paths linking patients, encounters to clinical entities (e.g., conditions).

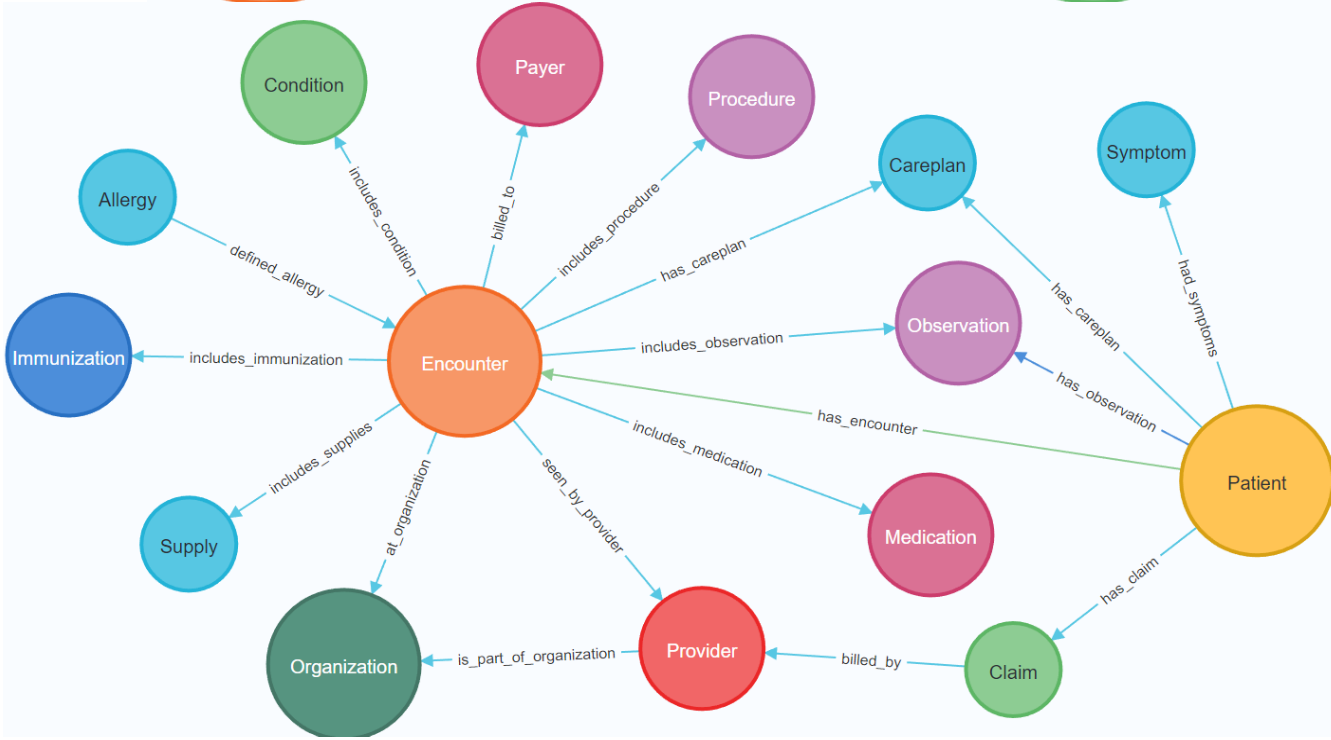
- **Similar patterns exist such as** :
  1. Patient → **has_condition** → Condition
  2. Patient → **has_encounter** → Encounter -> **includes_condition** → Condition

- **Rationale for retaining only one pattern:**
  - **Reason 1:** Improve **path explosion** problem**.**
  - **Reason 2:** GDBs **excel** at traversal.
  - **Reason 3:** Improve **memory efficiency**.

Impact:

•Relationships dropped from **~505,000 to ~316,000.**





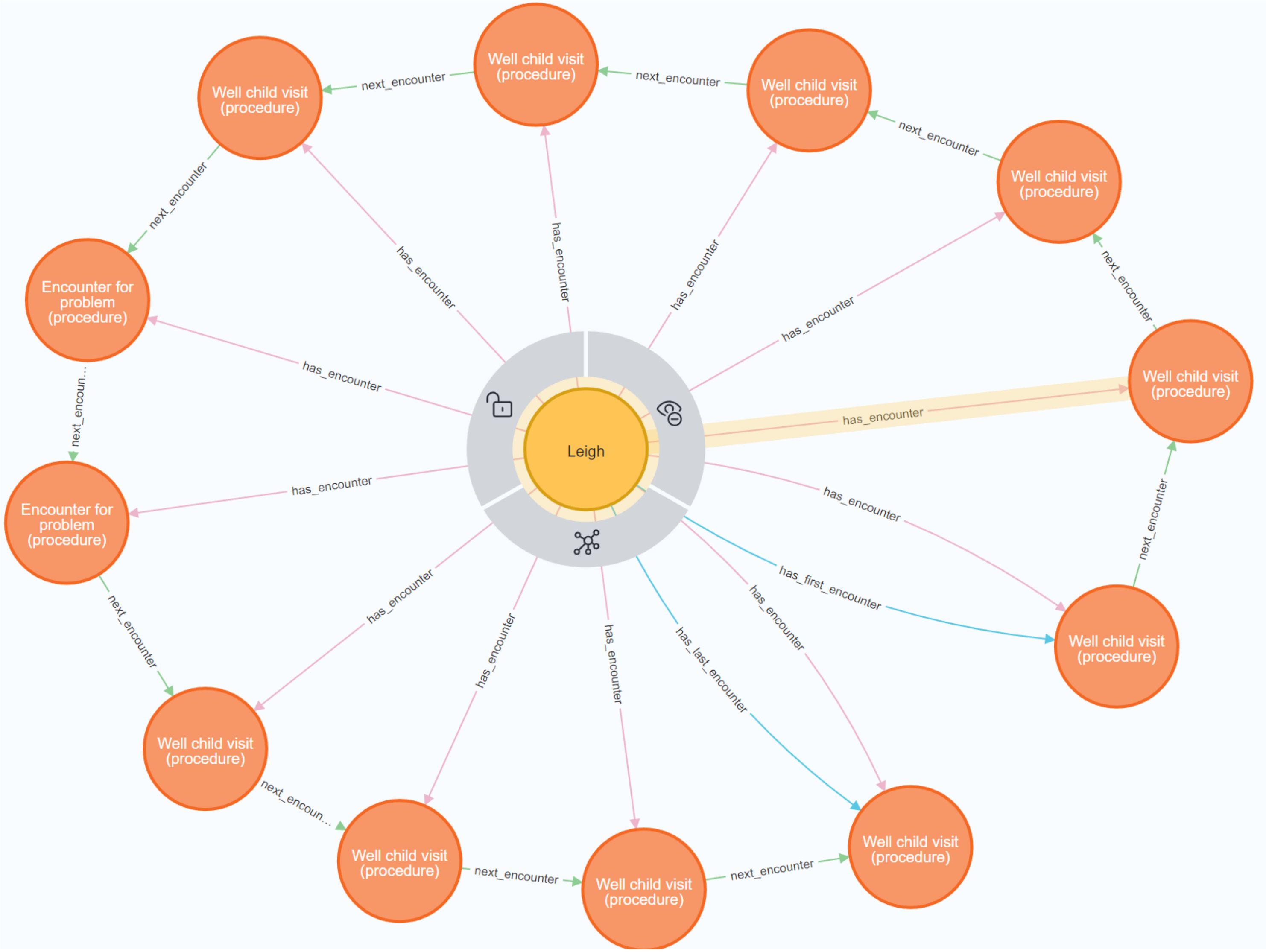
Step 4: Introduce Hierarchical Relationships

Introduced hierarchical & temporal relationships:

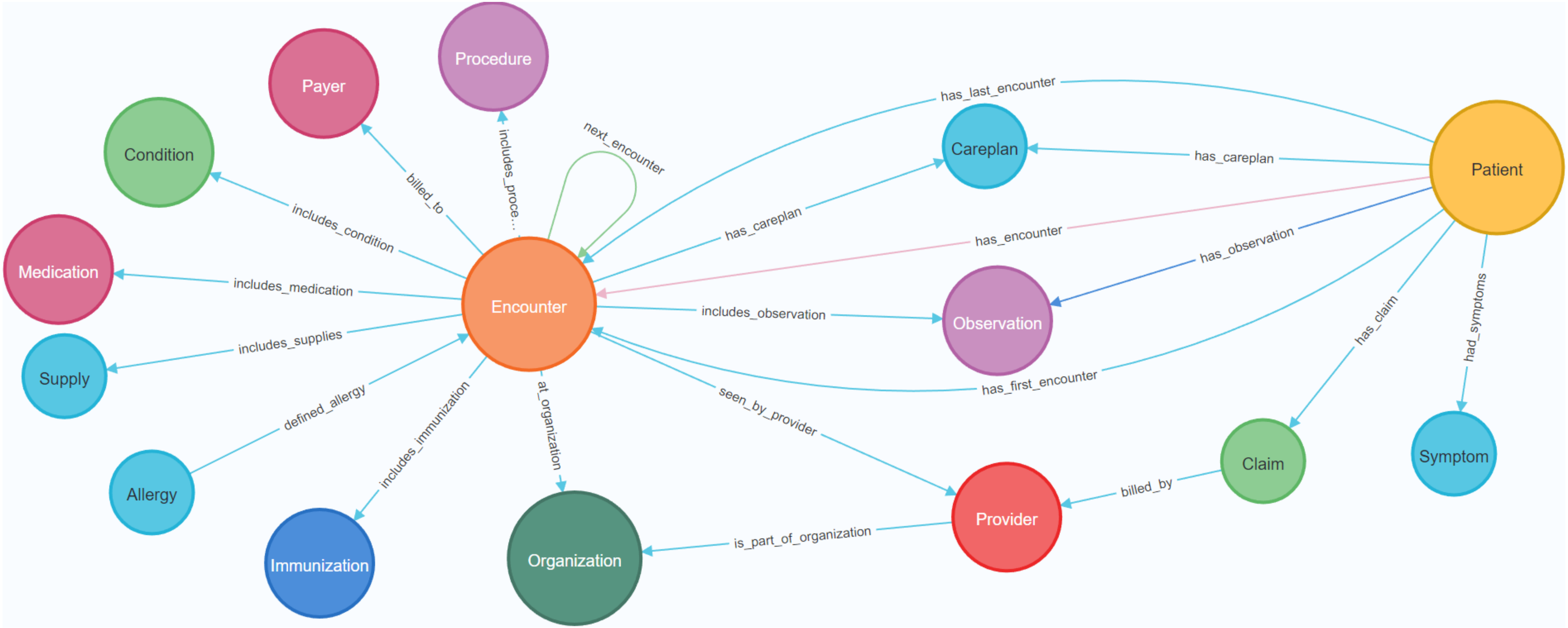
- Encounter → **next_encounter** → Encounter
- Patient → **has_first_encounter** → Encounter
- Patient → **has_last_encounter** → Encounter

Impact after applying these steps:

- **Entities:** ~64,179
- **Relationships:** ~330,818



Task 1: Final Meta-Graph



Practical Examples – Task 2

Task 2: Performance Comparison

- Conducted a **comparative performance evaluation** of **relational** and **graph** database systems using **identical query scenarios**.

- Employed **PostgreSQL (RDB)** and **Neo4j (GDB)**.

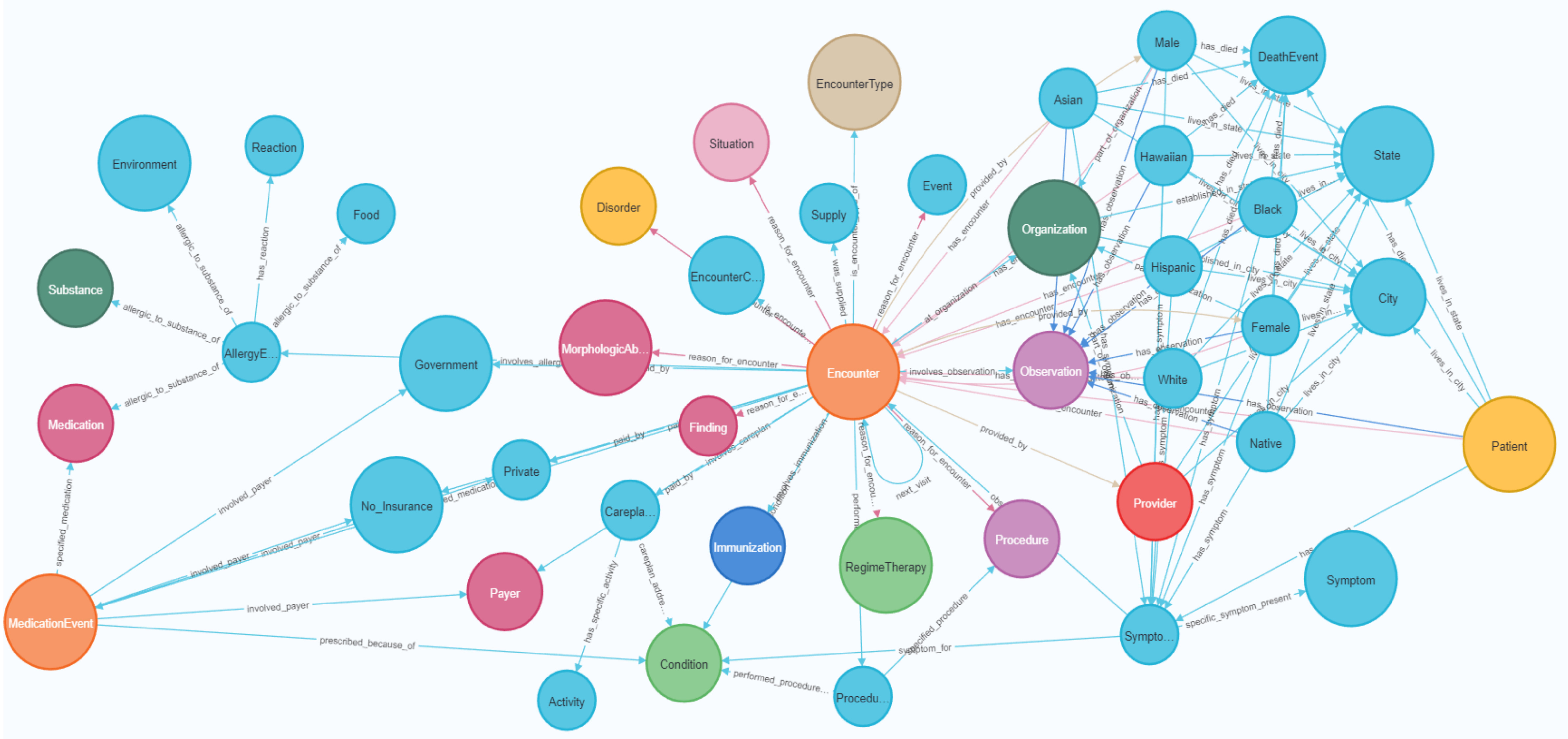
- **Both** databases were **used** on the **same machine and under similar configurations.**

- Generated a dataset of **~80.000 patients** (Total Size: **~27 GB**)

- **Same** model schema as **task 1** was used with extra modifications:
  - **Turned some Properties -> Labels**
  - **Introduced Intermediate nodes**

- Resulting graph characteristics:
  - **Entities: ~35,000,000**
  - **Relationships: ~235,000,000**

Task 2: Performance Comparison



Task 2: Results

Use case	RDB (Postgresql)	GDB (Neo4j)
Data ingestion time	**22 mins**	~ 2.1 hours
Memory utilization	36 GB	**32 GB**
Question 1. Identify patients who experienced a **fever within 14 days after an immunization** and subsequently received an **antipyretic within 7 days**.	38.7 sec	**16.25 sec**
Question 2. Among patients with a **death event**, determine the most frequent **care pathways in the 60 days preceding death**.	17.5 sec	**10.5 sec**
Question 3. Identify **pairs of providers** who tend to **co-treat the same patients for the same symptom within a ±30-day window**.	2.2 mins	**15.5 sec**

- Ingestion time in **GDBs** seems to be ~5.9 **slower** than **RDBs**
- The queries (after having defined the graph structure) perform **~x4 times faster** on **GDB** compared to **RDB.**
- Despite the above, there are studies that support that as the number of **data increases**, performance difference is more **noticeable.**

Practical Examples – Task 3

Task 3: Applied Graph Data Science

The final task demonstrates how Graph Data Science (GDS) algorithms can be applied to enhance knowledge discovery within healthcare networks.

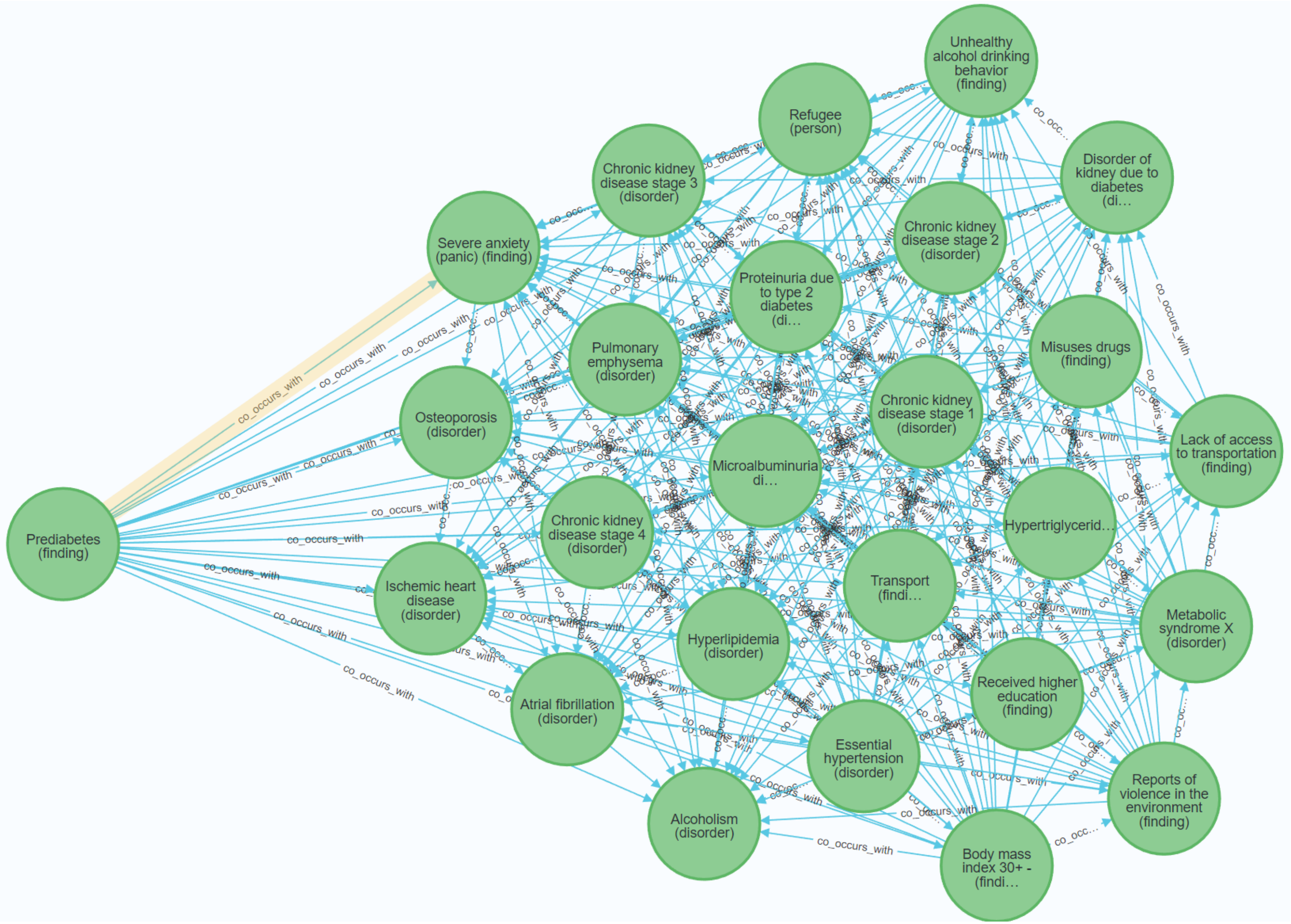
Two analytical questions were addressed:

- **What clusters of conditions tend to co-occur across patients?**
  - Created a **co-occur** relationship with weighting to be frequency on pairs on conditions.
  - Applied **Louvain** community detection to create groups of conditions.

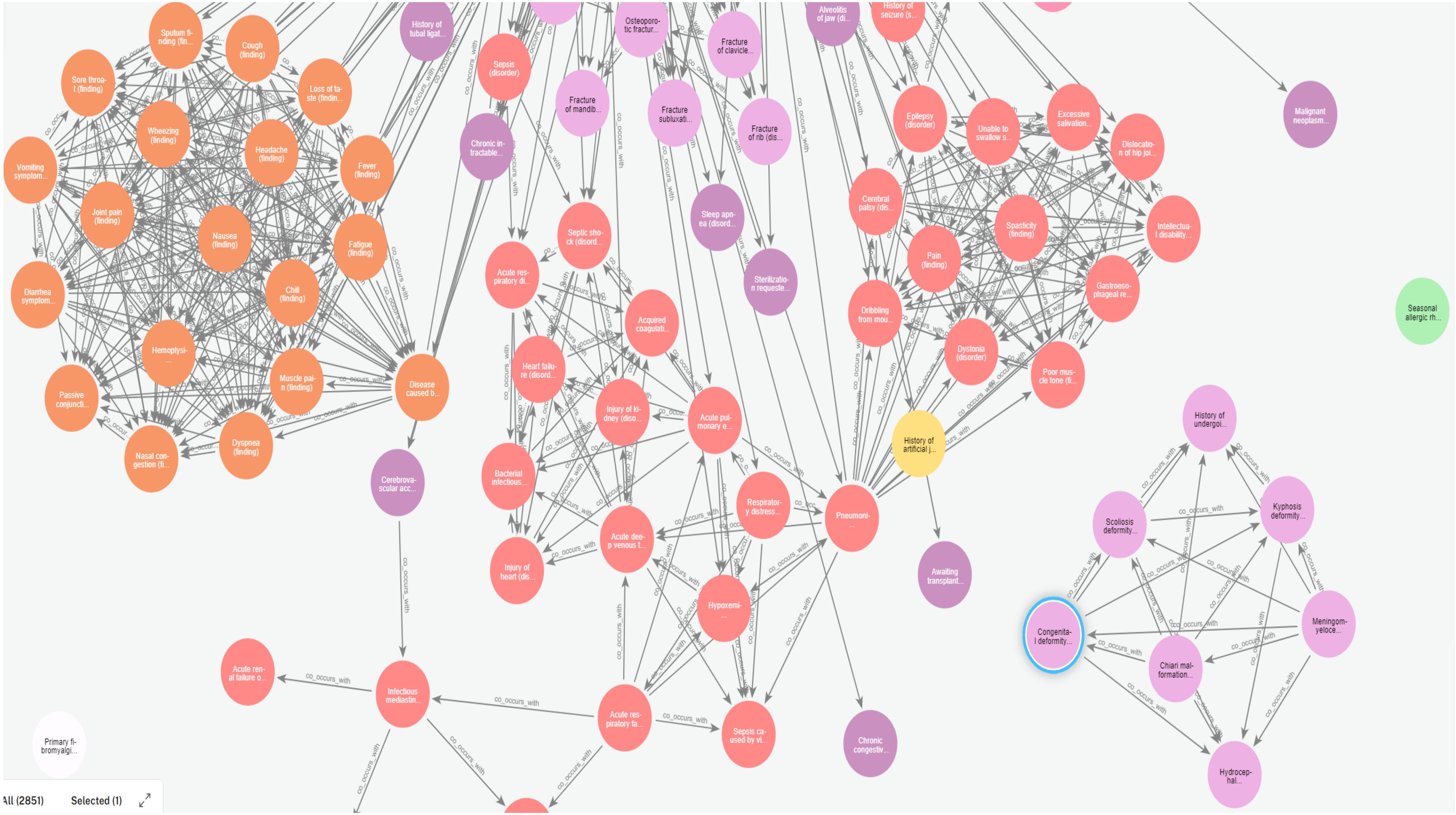
- **Which providers act as chokepoints in referral pathways?**
  - Created a **referral** relationship between providers.
    - By selecting patients that within a short window changed providers.
  - Applied **Betweeness** centrality to find important nodes acted as bridges.

GDS – Use case 1

Use case 1: Create Co-occurred conditions

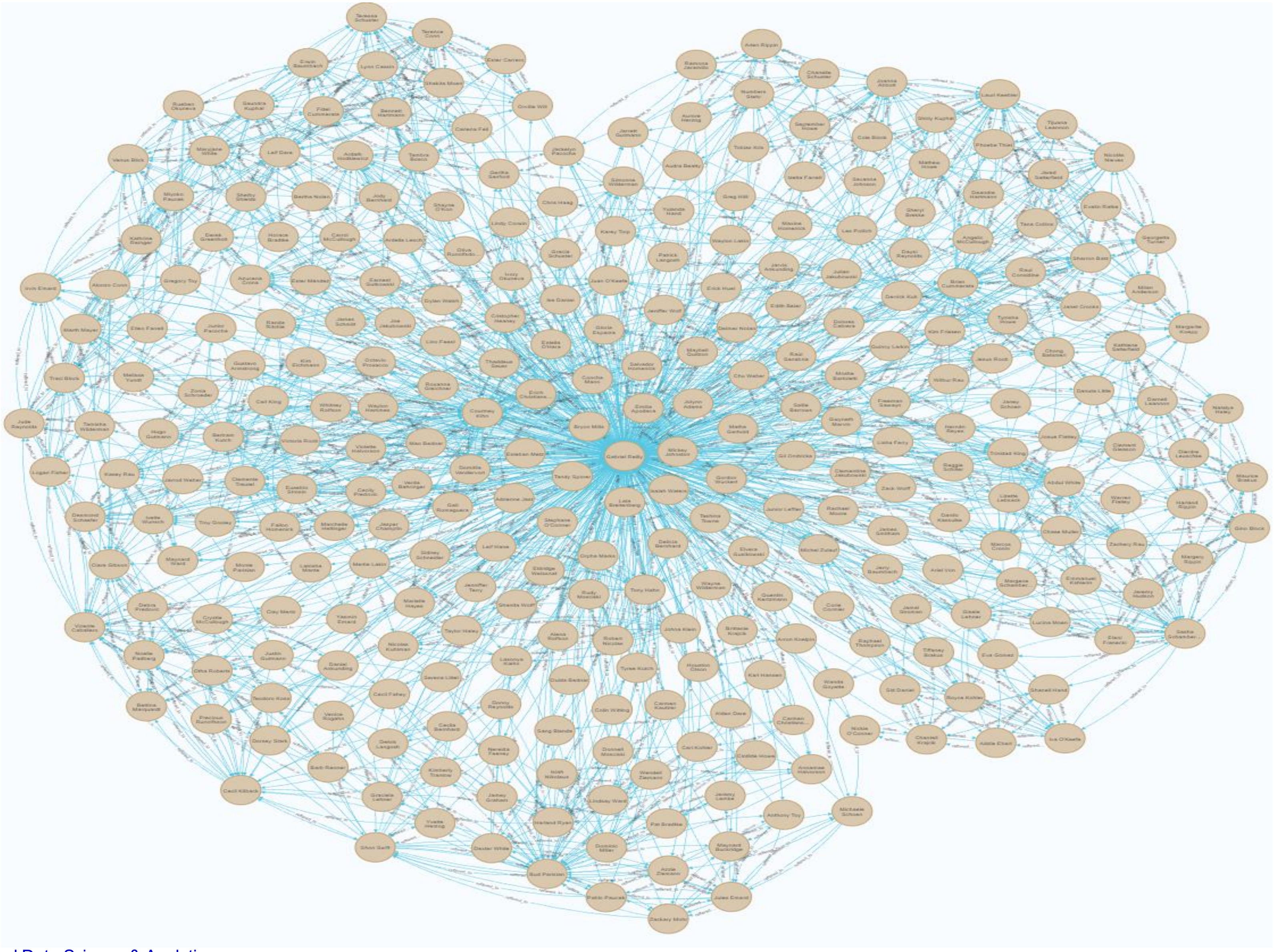


Use case 1: Visualizing groups of conditions

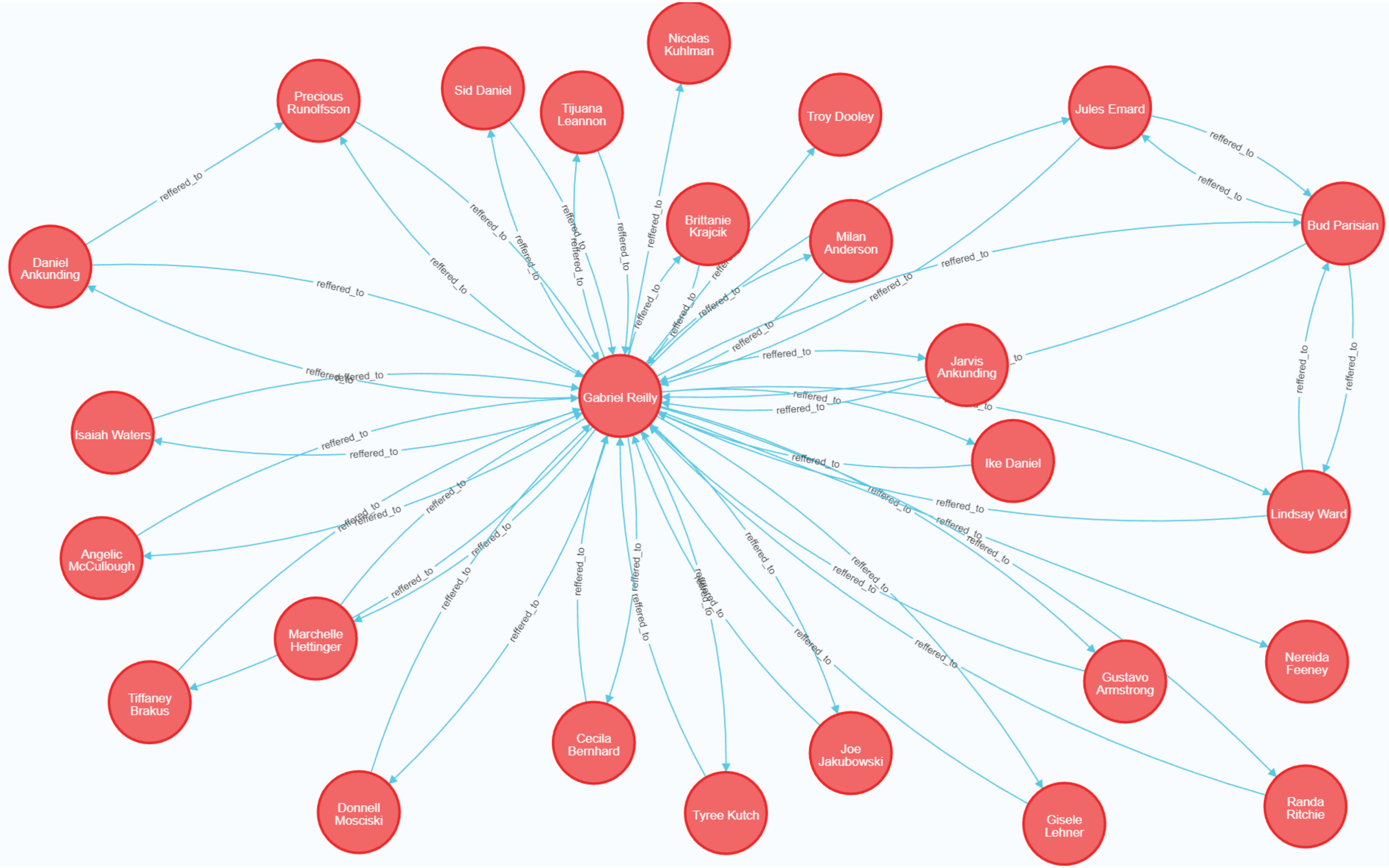


GDS – Use case 2

Use case 2: Providers network



Use case 2: Providers network



Use case 2: Providers network

provider	score
"Gabriel Reilly"	1272233.81822999
"Courtney Kihn"	29378.421248573144
"Bud Parisian"	25932.835528117175
"Cortez Price"	24256.21033923793
"Ana Luisa Gallardo"	17282.83680187145
"Terrie Ratke"	16852.69970083501

Key Takeaways

- **Synthetic data** enables research and innovation **without compromising patient privacy** and with **free of cost**.
- Graph modeling **preserves complex** healthcare relationships, unlocking deeper insights than traditional relational models.
- **Performance trade-off: Graphs** deliver ~4x faster query performance vs **RDBs**, thought ingestion is slower due to **upfront transformation**.
- **Graph Data Science Algorithms** can **reveal** hidden patterns, paths and clusters, and important nodes.
- **Future potential:** Integration with ontologies and multi-modal data (-omics, phenotypes and other) for richer insights.

Thank You