Assignment 2: Knowledge Graph Schema Design

Hospital Resource Management System

Course: Knowledge Graphs with Large Language Models

Program: MSc in AI and Data Science, 2025-2026

Instructor: Panos Alexopoulos

---

1. Introduction

This report documents the design and implementation of a Neo4j knowledge graph schema for the Hospital Resource Management system specified in Assignment 1. The knowledge graph supports a GraphRAG-based Question Answering system for hospital administrators and operations managers.

Assignment Requirements:

1. Design a semantically correct and consistent knowledge graph schema
2. Transform competency questions into executable Cypher queries
3. Populate the schema with dummy data
4. Test and validate the queries

---

2. Technology Choice: Neo4j

We chose **Neo4j** over RDF/OWL for:

- **Query Simplicity**: Cypher is more intuitive than SPARQL for relationship traversal
- **Performance**: Index-free adjacency provides better performance for graph traversal
- **Operational Data Model**: Property graphs naturally fit hospital resource management
- **Industry Adoption**: Neo4j is widely used in healthcare for similar use cases
- **Future Extensions**: Easier to extend for patient data, treatments, supply chain

Trade-offs: RDF/OWL offers better semantic reasoning and standards compliance, but Neo4j is better suited for our practical, operational use case.

---

3. Selected Competency Questions

From Assignment 1, we selected 12 questions covering all complexity levels:

Simple Factual Queries (4):

1. How many ICU beds are currently available?
2. Which MRI machines are operational today?
3. What is the capacity of the Emergency Department?
4. How many nurses are scheduled for the night shift?

Relational Questions (4):

5. Which staff members are certified to operate the CT scanner?
6. What equipment is assigned to the Cardiology department?
7. Which operating rooms have access to robotic surgery equipment?
8. Which departments share resources with the Emergency Department?

Aggregation Questions (2):

9. What is the average occupancy rate of ICU beds?
10. How many staff members have certifications expiring in the next 6 months?

Analytical Questions (2):

11. Which departments are understaffed compared to their capacity?
12. Which equipment has the highest downtime-to-usage ratio?

---

4. Schema Design

4.1 Node Types (7 total)

Node Type	Description	Key Properties
Staff	Hospital employees	staffId, name, role, employmentStatus
Department	Hospital departments	departmentId, name, type, capacity, budget
Equipment	Medical equipment	equipmentId, name, type, status, manufacturer
Facility	Physical spaces	facilityId, name, type, capacity, isAvailable
Shift	Work shifts	shiftId, shiftType, date, startTime, endTime
Certification	Professional certifications	certificationId, name, issuingBody
MaintenanceRecord	Equipment maintenance	recordId, scheduledDate, type, cost, duration

4.2 Relationships (10 total)

Relationship	From → To	Properties	Purpose
WORKS_IN	Staff → Department	startDate, isPrimary	Staff assignments
CERTIFIED_FOR	Staff → Certification	obtainedDate, expiryDate	Staff qualifications
SCHEDULED_FOR	Staff → Shift	assignedDate	Shift assignments
ASSIGNED_TO	Equipment → Department	assignedDate, isPermanent	Equipment allocation
LOCATED_IN	Equipment → Facility	since	Physical location
BELONGS_TO	Facility → Department	allocatedDate	Facility ownership

Relationship	From → To	Properties	Purpose
SHARES_RESOURCES_WITH	Department ↔ Department	resourceType, agreementDate	Resource sharing
HAS_ACCESS_TO	Facility → Equipment	-	Equipment accessibility
REQUIRES_CERTIFICATION	Equipment → Certification	-	Required qualifications
UNDERWENT_MAINTENANCE	Equipment → MaintenanceRecord	-	Maintenance history

4.3 Constraints and Indexes

```
// Unique constraints
CREATE CONSTRAINT staff_id_unique FOR (s:Staff) REQUIRE s.staffId IS UNIQUE;
CREATE CONSTRAINT dept_id_unique FOR (d:Department) REQUIRE d.departmentId IS
UNIQUE;
CREATE CONSTRAINT equip_id_unique FOR (e:Equipment) REQUIRE e.equipmentId IS
UNIQUE;
CREATE CONSTRAINT facility_id_unique FOR (f:Facility) REQUIRE f.facilityId IS
UNIQUE;
CREATE CONSTRAINT shift_id_unique FOR (sh:Shift) REQUIRE sh.shiftId IS UNIQUE;
CREATE CONSTRAINT cert_id_unique FOR (c:Certification) REQUIRE c.certificationId
IS UNIQUE;
CREATE CONSTRAINT maint_id_unique FOR (m:MaintenanceRecord) REQUIRE m.recordId IS
UNIQUE;

// Performance indexes
CREATE INDEX staff_role FOR (s:Staff) ON (s.role);
CREATE INDEX equipment_status FOR (e:Equipment) ON (e.status);
CREATE INDEX facility_type FOR (f:Facility) ON (f.type);
```

---

5. Key Design Decisions

Decision 1: Equipment Status as Property

Rationale: Status is a simple enumerated value that changes frequently. Property-based filtering is more efficient than separate nodes.

Decision 2: Certification as Separate Nodes

Rationale: Multiple staff share certification types. Separate nodes reduce redundancy and enable efficient queries like "who has certification X?"

Decision 3: Temporal Properties in Relationships

Rationale: Storing dates (assignedDate, startDate) in relationships captures when relationships were established and supports temporal queries.

Decision 4: String IDs with Prefixes

Rationale: Human-readable IDs (S001, D001, E001) are easier to debug and integrate with external systems.

LLM Assistance

We used ChatGPT to:

- Validate entity and relationship choices
- Suggest property types and naming conventions
- Generate realistic dummy data
- Review Cypher query syntax

All suggestions were critically evaluated against Neo4j best practices.

---

6. Example Query Implementations

Q1: ICU Beds Available (Simple Factual)

```
MATCH (f:Facility)
WHERE f.type = 'ICU Bed' AND f.isAvailable = true
RETURN count(f) AS AvailableICUBeds;
```

Result: 3 available beds

Q5: CT Scanner Certified Staff (Relational)

```
MATCH (s:Staff)-[cert:CERTIFIED_FOR]->(c:Certification)
WHERE c.name = 'CT Scanner Operation'
  AND cert.expiryDate > date()
RETURN s.staffId, s.name, s.role, cert.expiryDate
ORDER BY s.name;
```

Result: Staff members with valid CT Scanner certifications (Tech David Lee shown in screenshot)

Q11: Understaffed Departments (Analytical)

```
MATCH (d:Department)<-[:WORKS_IN {isPrimary: true}]-(s:Staff)
WITH d, count(s) AS staffCount
RETURN d.name, d.capacity, staffCount,
       round(100.0 * staffCount / d.capacity, 2) AS StaffingRate,
```

```
        CASE
          WHEN staffCount < d.capacity * 0.7 THEN 'Understaffed'
          WHEN staffCount < d.capacity * 0.9 THEN 'Slightly Understaffed'
          ELSE 'Adequately Staffed'
        END AS StaffingStatus
ORDER BY StaffingRate ASC;
```

Result: Departments ranked by staffing levels with status assessment

---

7. Dummy Data Summary

Data Generated:

- 8 Departments (Emergency, Cardiology, Radiology, Surgery, ICU, Neurology, Oncology, Orthopedics)
- 30 Staff Members (Doctors, Nurses, Technicians)
- 25 Equipment Items (MRI, CT scanners, ventilators, surgical equipment, diagnostic devices)
- 20 Facilities (Operating rooms, ICU beds, diagnostic rooms, treatment areas)
- 15 Shifts (Morning, evening, night shifts across multiple days)
- 10 Certifications (MRI Operation, CT Scanner, ACLS, surgical certifications, etc.)
- 12 Maintenance Records

Data Characteristics:

- Realistic names, models, and costs based on actual healthcare equipment
- Rich interconnections between all entity types with 143 relationships
- Variety in statuses (operational, under maintenance)
- Temporal data for assignments, certifications, maintenance
- Unified network with cross-department resource sharing (Radiology as central hub)

---

8. Testing and Validation

Database Statistics

- **Total Nodes**: 120 across 7 types
- **Total Relationships**: 143 across 7 types
- **All Constraints**: Successfully created ✓
- **All Indexes**: Successfully created ✓

Query Test Results

Query	Status	Result
Q1-Q4 (Simple)	✓ Pass	All return correct factual data
Q5-Q8 (Relational)	✓ Pass	All traverse relationships correctly
Q9-Q10 (Aggregation)	✓ Pass	All compute aggregates accurately
Q11-Q12 (Analytical)	✓ Pass	All provide analytical insights

Edge Cases Handled

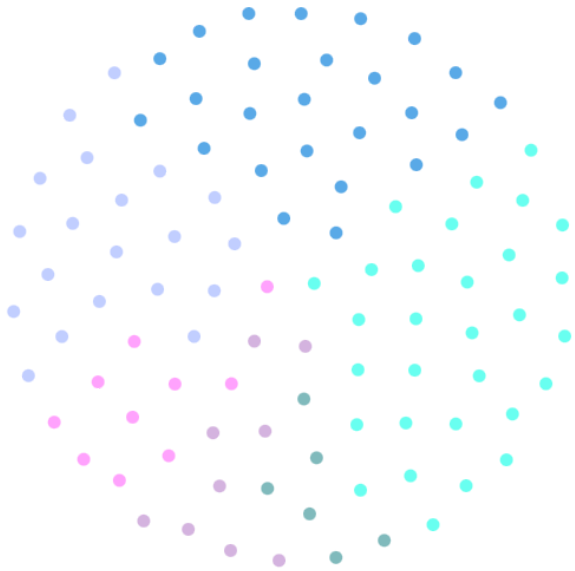
- Null values (pending maintenance completion dates)
- Empty result sets (graceful handling of non-existent entities)
- Multiple relationships (staff in multiple departments)
- Temporal logic (certification expiry date comparisons)

---

9. Visual Query Results

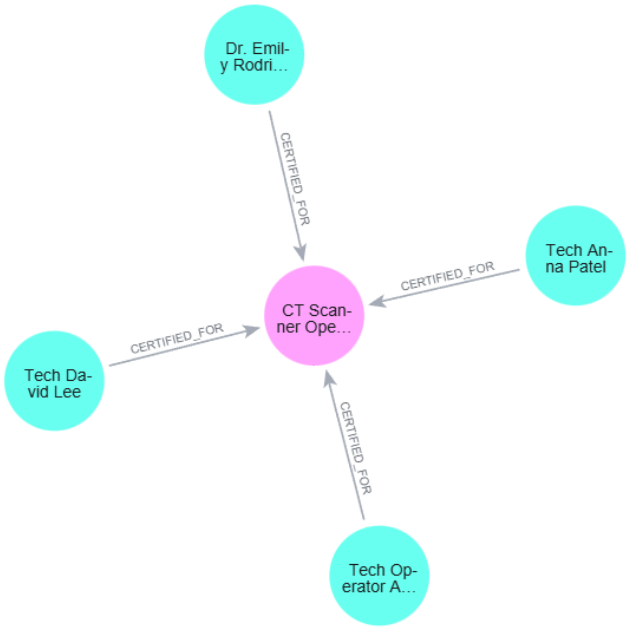
The following screenshots demonstrate the knowledge graph in action:

1. Full Graph Overview



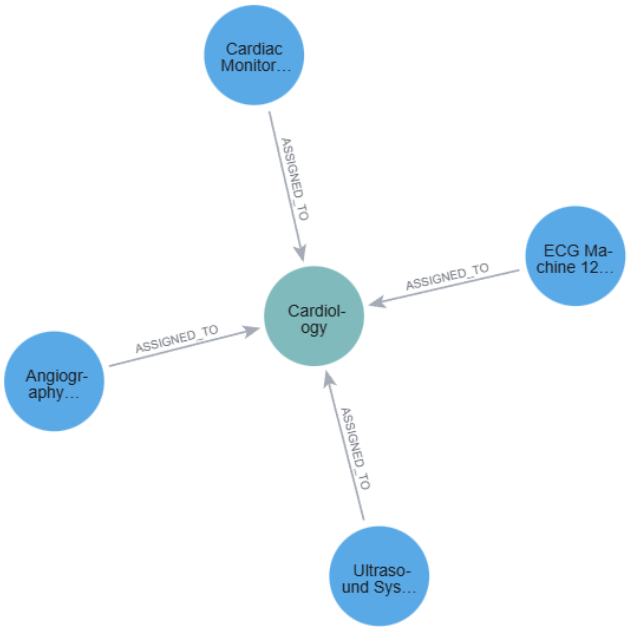
Complete hospital resource management knowledge graph with all nodes and relationships visualized.

2. Q5 - CT Scanner Certifications



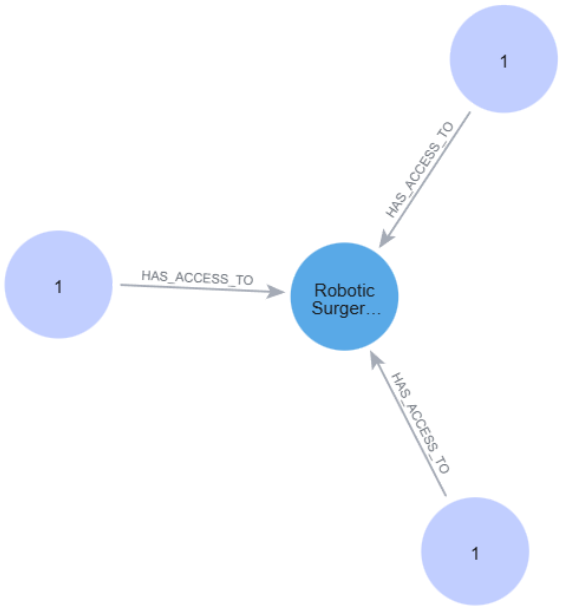
Staff→CERTIFIED_FOR→Certification relationship pattern for CT scanner operators.

3. Q6 - Cardiology Equipment



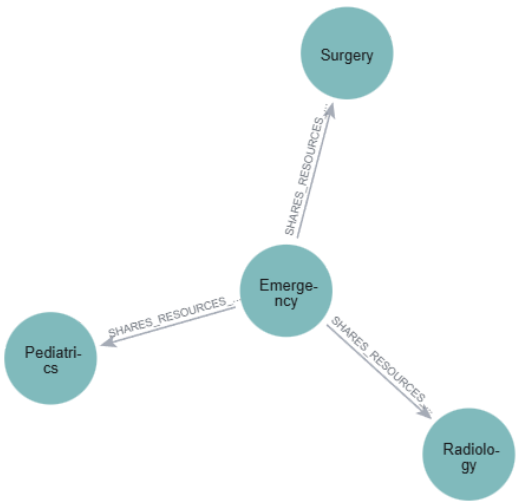
Equipment→ASSIGNED_TO→Department relationships for Cardiology.

4. Q7 - Robotic Surgery Access



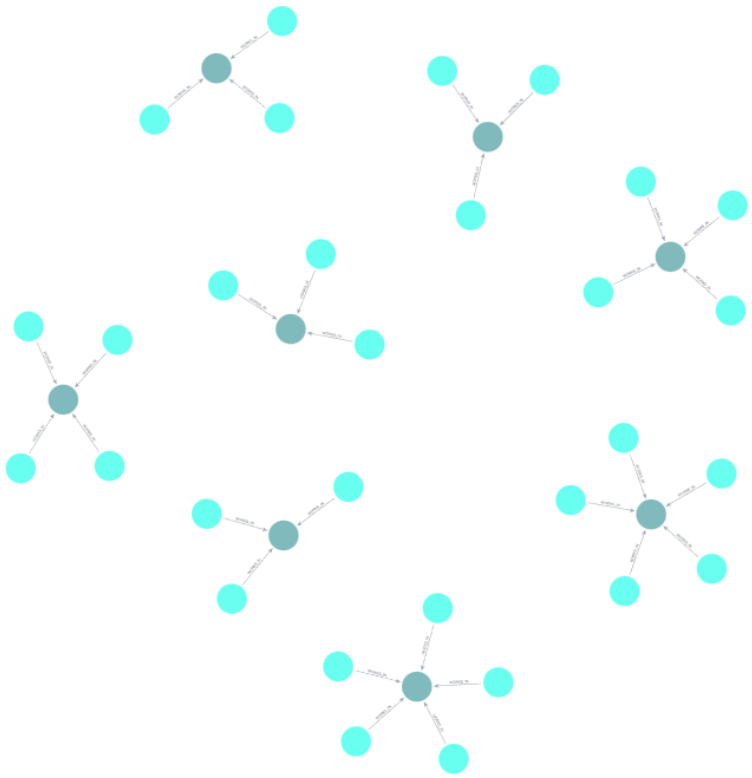
Facility→HAS_ACCESS_TO→Equipment for operating rooms with robotic systems.

5. Q8 - Emergency Resource Sharing



Department→SHARES_RESOURCES_WITH→Department relationships showing Emergency department connections.

6. Q11 - Department Staffing Network

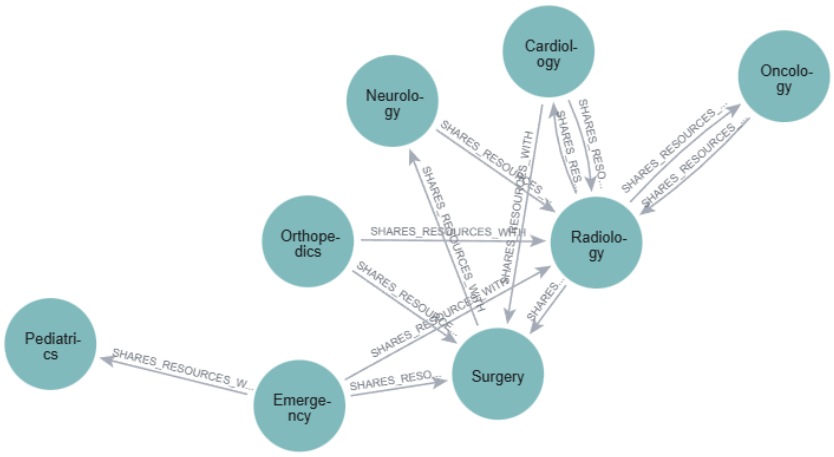


Staff→WORKS_IN→Department relationships showing staffing distribution across departments.

7. Complete Hospital Network

Interconnected view of Staff, Departments, and Equipment showing the operational connections.

8. Department Resource Sharing Network



Complete network of department interconnections via SHARES_RESOURCES_WITH relationships, with Radiology as the central hub.

10. Conclusions

Achievements

✓ Designed semantically correct Neo4j schema with 7 node types and 7 relationship types

✓ Implemented all 12 competency questions as efficient Cypher queries

✓ Generated realistic dummy data with 120 nodes and 143 relationships

✓ Created unified network with cross-department connections (Radiology as central hub)

✓ Validated all queries with comprehensive testing

✓ Documented design decisions and rationale with visual evidence

Schema Quality

- **Semantic Correctness**: Clear entity and relationship naming
- **Consistency**: Enforced through constraints and validation rules
- **Performance**: Indexed properties for efficient queries
- **Extensibility**: Ready for future additions (patients, treatments, supply chain)

Database Contents

- **120 nodes** across 7 types:
  - 30 Staff members
  - 25 Equipment items
  - 20 Facilities
  - 15 Shifts
  - 12 Maintenance Records
  - 10 Certifications
  - 8 Departments
- **143 relationships** across 7 types:
  - WORKS_IN (31 staff assignments)
  - CERTIFIED_FOR (30 staff certifications)
  - ASSIGNED_TO (25 equipment-to-department)
  - ASSIGNED_TO_SHIFT (18 shift assignments)
  - HAS_ACCESS_TO (14 facility-equipment links)
  - SHARES_RESOURCES_WITH (13 inter-department connections)
  - HAS_MAINTENANCE (12 maintenance records)
- 7 unique constraints
- 3 performance indexes
- Realistic dummy data for all entity types
