Advanced databases - Assignment 4

José Cutileiro - Group 70

May 2024

1 Introduction

In this new assignment, the objective is quite similar to the previous one. However, instead of using GraphDB and SPARQL queries, we will make use of the Neo4J Sandbox and Cypher queries.

2 My approach

Initially, I intended to adopt a similar approach to the previous one. However, instead of generating an extensive TTL file, I would opt to create a query that imports the data from my CSV files into the database.

I did go through this process for some classes; however, before proceeding with all of them, I discovered a way to transfer the data in TTL format, which I had already generated for the previous assignment, to Neo4J.

To **initialize the database**, simply execute these three commands in Neo4J. I have placed the data in a public GitHub repository to make access easier.

```
CREATE CONSTRAINT n10s_unique_uri FOR (r:Resource) REQUIRE r.uri IS UNIQUE;
CALL n10s.graphconfig.init({ handleVocabUris: "IGNORE" });
CALL n10s.rdf.import.fetch("https://raw.githubusercontent.com
            /JoseCutileiro/ImageLinks/master/ADB/out3.ttl", "Turtle");
```

After running this commands, all the data is inserted into the database without any further complication. Therefore, now all that remains is to create the queries and obtain the desired results accurately.

3 Decisions and assumptions

I completed this assignment before receiving feedback on assignment 3, so I assumed that assignment 3 was correct. However, if that's not the case, I'll need to make the necessary adjustments to the solution.

4 Queries and results

4.1 Find the name, director and department of all programmes.

```
=== QUERY ===
MATCH (programme:Programme)-[:programmeDept]->(department)
MATCH (director:Senior_Teacher)-[:oversees]->(programme)
RETURN programme.programmeName, department.deptName, director.teacherId AS director_id;

=== RESULTS ===
programme.programmeName,department.deptName,director_id
"P-74","""D8""","""19601021-0018"""
"P-71","""D8""","""19610620-0006"""
"P-73","""D8""","""19600814-0002"""
"P-72","""D8""","""19660630-0020"""
"P-61","""D7""","""19680712-0028"""
"P-11","""D2""","""19620424-0026"""
"P-13","""D2""","""19690408-0009"""
"P-12","""D2""","""19610623-0005"""
"P-14","""D2""","""19560812-0016"""
"P-01","""D1""","""19620522-0023"""
"P-31","""D4""","""19650303-0019"""
"P-33","""D4""","""19570828-0008"""
"P-32","""D4""","""19570826-0012"""
"P-34","""D4""","""19610918-0027"""
"P-21","""D3""","""19570615-0011"""
"P-53","""D6""","""19600905-0003"""
"P-52","""D6""","""19611219-0014"""
"P-54","""D6""","""19630126-0001"""
"P-51","""D6""","""19580515-0017"""
"P-42","""D5""","""19620831-0024"""
"P-41","""D5""","""19580218-0007"""
```

4.2 Find the names of all students who worked as teaching assistants in courses given by the D3-2 division in study period 2 in academic year 2023/2024.

```
=== QUERY ===
MATCH (ta:Teacher)
MATCH (ta:Student)
MATCH (ci:Course_Instance)
MATCH (ta)-[:worksIn]->(ci)
MATCH (course:Course)
```

```
MATCH (course)-[:hasExecution]->(ci)
MATCH (course)-[:courseDivision]->(division)
WHERE (ci.studyPeriod = "2.0")
AND (ci.academicYearCourse = "2023-2024")
AND (division.divisionName = "D3-2")
RETURN ta.teacherName, ci.academicYearCourse, ci.studyPeriod;

=== RESULTS ===
ta.teacherName,ci.academicYearCourse,ci.studyPeriod
"TA 138","""2023-2024""","""2.0"""
"TA 60","""2023-2024""","""2.0"""
"TA 36","""2023-2024""","""2.0"""
"TA 38","""2023-2024""","""2.0"""
"TA 74","""2023-2024""","""2.0"""
```

4.3 Find all teachers who are assigned more than 120 hours in course 1015 in study period 1 in academic year 2018/2019.

```
=== QUERY ===
MATCH (hours:Hour_Information)
MATCH (teacher:Teacher)
MATCH (course:Course)
MATCH (course)-[:hasExecution]->(ci:Course_Instance)
MATCH (hours)-[:associatedWith]->(teacher)
MATCH (hours)-[:given]->(ci)
WHERE (hours.assignedHours > "120")
AND (course.courseCode = "1015")
AND (ci.studyPeriod = "1.0")
AND (ci.academicYearCourse = "2018-2019")
RETURN teacher.teacherName,teacher.teacherId;

=== RESULTS ===
teacher.teacherName,teacher.teacherId
"TA 59","""19750102-0059"""
"Teacher 19","""19650303-0019"""
"Teacher 7","""19580218-0007"""
"TA 38","""19790702-0038"""
"TA 57","""19780424-0057"""
"Teacher 20","""19660630-0020"""
```

4.4 Find all students registered for course instance I-910 that were not registered for course instance I-911.

```
=== QUERY ===
MATCH (student:Student)
MATCH (ci:Course_Instance)
MATCH (student)-[:register]-(regis:Registration)
MATCH(ci)-[:hasRegistration]-(regis:Registration)
WHERE (ci.instanceId = "I-910")
RETURN student.studentId,student.studentName,student.graduated;

=== RESULTS ===
student.studentId,student.studentName,student.graduated
"19921201-0094","""TA 94""","""False"""
```

4.5 Find all programmes along with the total number of owned courses. List the results in descending order of number of owned courses.

```
=== QUERY ===
MATCH (programme:Programme)-[:owns]->(course:Course)
RETURN programme.programmeCode AS code,
     programme.programmeName AS name, COUNT(course) AS courseCount
ORDER BY courseCount DESC;

=== RESULTS ===
code,name,courseCount
"10061","""P-61""",45
"10021","""P-21""",33
"10001","""P-01""",32
"10041","""P-41""",21
"10042","""P-42""",20
"10071","""P-71""",14
"10032","""P-32""",14
"10052","""P-52""",13
"10012","""P-12""",12
"10033","""P-33""",12
"10054","""P-54""",11
"10051","""P-51""",11
"10072","""P-72""",10
"10011","""P-11""",10
"10074","""P-74""",9
"10013","""P-13""",8
"10014","""P-14""",8
```

```
"10073","""P-73""",6
"10034","""P-34""",6
"10053","""P-53""",3
"10031","""P-31""",2
```

4.6 Find the number of: senior teachers

```
=== QUERY ===
MATCH (senior_teacher:Senior_Teacher)
RETURN COUNT(senior_teacher) AS count;

=== RESULTS ===
count
30
```

4.7 Find the number of: people

```
=== QUERY ===
MATCH (senior_teacher:Senior_Teacher)
WITH COUNT(senior_teacher) AS senior_teacher_count
MATCH (student:Student)
WITH senior_teacher_count, COUNT(student) AS student_count
RETURN (senior_teacher_count + student_count) AS total_count;

=== RESULTS ===
total_count
440
```