Assignment 4
Group 52

Nicolas Johansson
Aapo Haavisto

Since the labeled property graphs allow for relations to have properties, the TeacherAssignment and Registrations entities (and corresponding dual dummy relations) in our schema could be simplified to single relations.

However, we chose to not implement this, and the only modification made was replacing the missing values (some grades in Registrations) with *-1.0*. This allowed us to import our RDF data from assignment 3 directly into Neo4j with the following three commands:

```
> CALL n10s.graphconfig.init();

> CREATE CONSTRAINT n10s_unique_uri FOR (r:Resource) REQUIRE r.uri IS
UNIQUE

> CALL n10s.rdf.import.fetch("<path to raw data>", "RDF/XML");
```

This turned out to work fine for this assignment, although some queries could be simplified should the schema be improved as described above. Please find the queries and corresponding results in the following pages.

1:

```
MATCH (programme:Programme)-[:OfDepartment]->(department:Department),
(programme:Programme)- [:HasDirector]-> (director:SeniorTeacher),
(director:SeniorTeacher)-[:subClassOf]->(teacher:Teacher)
RETURN programme.programmeName as programme_name, department.dept as
department, teacher.name as name
```

result:

	programme_name	department	name
1	"P-74"	"D8"	"Teacher 18"
2	"P-73"	"D8"	"Teacher 2"
3	"P-72"	"D8"	"Teacher 20"
4	"P-71"	"D8"	"Teacher 6"
5	"P-61"	"D7"	"Teacher 28"
6	"P-54"	"D6"	"Teacher 1"
7			

2:

```
MATCH
(division:ns0__Division)<-[:ns0__OfDivision]-(course:ns0__Course)<-[:ns0__I
nstanceOf]-(instance:ns0__CourseInstance)<-[:ns0__AssignedOn]-(assignment:n
s0__TeacherAssignment)-[:ns0__AssignedTo]->(teacher:ns0__Teacher),
(student:ns0__Student) WHERE teacher.ns0__teacherID =
student.ns0__studentID and instance.ns0__studyPeriod = 4.0 and
instance.ns0__academicYear = "2023-2024" and division.ns0__div = "D3-2"
RETURN student.ns0__name as student_name LIMIT 25
```

result:

	student_name
1	"TA 84"
2	"TA 111"
3	"TA 90"
4	"TA 35"
5	"TA 39"
6	"TA 42"

3:

```
MATCH (ci:CourseInstance)-[:InstanceOf]->(c:Course),
(ci:CourseInstance)<-[:AssignedOn]-(ta:TeacherAssignment)-[:AssignedTo]->(t
:Teacher)
WHERE c.courseCode = 1015 and ci.studyPeriod = 1.0 and ci.academicYear =
"2018-2019" and ta.assignedHours > 120
RETURN t.name as Name
```

result:

	Name
1	"TA 38"
2	"Teacher 7"
3	"Teacher 20"
4	"TA 59"
5	"Teacher 19"

4:

```
MATCH
(student:ns0__Student)<-[:ns0__StudentRegistered]-(registration:ns0__Regist
ration)-[:ns0__CourseRegistered]->(courseInstance:ns0__CourseInstance)
WHERE courseInstance.ns0__instanceID = "I-910" AND NOT EXISTS
{(student)<-[:ns0__StudentRegistered]-(registration2:ns0__Registration)-[:n
s0__CourseRegistered]->(courseInstance2{ns0__instanceID: "I-911"})}
RETURN student.ns0__studentID as student
```

result:

	student
1	"19921201-0094"

5:

```
MATCH (p:Programme) RETURN p.programmeCode ,COUNT{ MATCH
(p:Programme)<-[:ProgrammeIn]-(pc:ProgrammeCourse) RETURN pc.academicYear}
as Num_of_courses
```

result:

	p.programmeCode	Num_of_courses
1	10074	211
2	10073	318
3	10072	318
4	10071	319
5	10061	207
6	10054	319

6a:

```
MATCH (teacher:ns0__SeniorTeacher)
RETURN COUNT(teacher) as num_senior_teachers
```

result: 30

6b:

```
RETURN SIZE(COLLECT
{MATCH (teacher:ns0__Teacher)
RETURN teacher.ns0__teacherID AS id
UNION
MATCH (student:ns0__Student)
RETURN student.ns0__studentID AS id})
```

result: 440