Advanced Databases (DAT410) Assignment 3

Luca Modica
Hugo Manuel Alves Henriques e Silva

May 6, 2024

1 Introduction

The goal of this report is to describe how we transformed data from a relational database to RDF triples, such that it conforms to the OWL/RDFS produced in the previous assignment. We will illustrate several results of querying the transformed data to answer specifically asked questions. This will be achieved by using SPARQL as a query language and GraphDB as a semantic graph database management system to store the triples.

2 Data preparation

As for the preparation of the data, we had to convert `.csv` files into `RDF triples`. We mostly followed the ontologies generated by WebProtégé, making changes where necessary:

- We changed column names on the `.csv` files to match our data properties.
- According to our ontology, the tables `Assigned_Hours` and `Reported_Hours` were merged into a single table called `Teacher_Hours`. The table contains a column to indicate the assigned hours and a column for the reported ones, for each data point.
- Regarding some object properties, we used the inverse relation (that is, switch the domain class with the range class). The reason is to make sure the domain class has the foreign keys that originally belonged to the range classes so that the RDF triples will represent the tables better.
- According to our ontologies, we added a boolean value attribute called `worksAsTA` to the `Student` class: if set to true, it will specify a student who is working as a teaching assistant. To add the column to the `Student.csv` file, we check for each teaching assistant if the related `teacherId` matches with any of the students: in that case, it means that the specific student is working as a teaching assistant.
- Some of the `.csv` files had float data types for integer values. These were converted back to integers so `GraphDB` could interpret properly the data type.
- In the `Course` table, a column with the id of the programme to which the course belongs to was missing. It was added in this step.
- Finally, we added department and division tables as we considered these to be classes as opposed to the initially presented relational schema where these were not considered as entities.

Most of these changes were directly applied through a Python script. However, some of them required manual altering of the `.csv` files.

3 SPARQL queries

Once the data were converted into RDF triples and loaded into GraphDB, the next task consisted of querying the data to answer the given questions in the assignment. Down below, for each question, we will report the related SPARQL query and the result obtained by running it.

- **Question 1: find the name, director and department of all programmes.**

  SPARQL query:

```
PREFIX ex: <http://assignment3.org/ontology#>

select ?programmeName ?teacherId ?teacherName ?departmentName
WHERE {
  ?programme a ex:Programme .
  ?programme ex:programmeName ?programmeName .
    ?programme ex:programmeDirectedBy ?director .
    ?programme ex:programmeBelongsTo ?department .

  ?director a ex:SeniorTeacher .
    ?director ex:teacherId ?teacherId .
    ?director ex:teacherName ?teacherName .

    ?department a ex:Department .
    ?department ex:departmentName ?departmentName
}
```

  Result:

programmeName	teacherId	teacherName	departmentName
P-01	19620522-0023	Teacher23	D1
P-11	19620424-0026	Teacher26	D2
P-12	19610623-0005	Teacher5	D2
P-13	19690408-0009	Teacher9	D2
P-14	19560812-0016	Teacher16	D2
P-21	19570615-0011	Teacher11	D3
P-31	19650303-0019	Teacher19	D4
P-32	19570826-0012	Teacher12	D4
P-33	19570828-0008	Teacher8	D4
P-34	19610918-0027	Teacher27	D4
P-41	19580218-0007	Teacher7	D5
P-42	19620831-0024	Teacher24	D5
P-51	19580515-0017	Teacher17	D6
P-52	19611219-0014	Teacher14	D6
P-53	19600905-0003	Teacher3	D6
P-54	19630126-0001	Teacher1	D6
P-61	19680712-0028	Teacher28	D7
P-71	19610620-0006	Teacher6	D8
P-72	19660630-0020	Teacher20	D8
P-73	19600814-0002	Teacher2	D8
P-74	19601021-0018	Teacher18	D8

- **Question 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.**

  SPARQL query:

```
PREFIX ex: <http://assignment3.org/ontology#>

select ?studentName
where {
    ?student a ex:Student .
    ?student ex:studentName ?studentName .
    ?student ex:workAsTA ?ta .
```

```
    # Find TeacherHours instances related to teaching assistants
    ?th a ex:TeacherHours .
    ?th ex:teacherHoursIn ?ta .
    ?th ex:courseHoursIn ?courseInstance .

  ?courseInstance a ex:CourseInstance .
    ?courseInstance ex:studyPeriod 2 .
    ?courseInstance ex:courseInstanceAcademicYear "2023-2024" .
    ?courseInstance ex:courseInstanceOf ?course .

    ?course a ex:Course .
    ?course ex:courseBelongsTo ?division .

  # find the division name
    ?division a ex:Division .
    ?division ex:divisionName "D3-2" .
}
```

Result:

studentName
TA36
TA138
TA38
TA74
TA60

- **Question 3: find all teachers who are assigned more than 120 hours in course 1015 in study period 1 in academic year 2018/2019.**

  SPARQL query:

```
PREFIX ex: <http://assignment3.org/ontology#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT DISTINCT ?teacherId ?teacherName (xsd:float(?assignedHourss) as
    ?assignedHours)
WHERE {
    ?teacher rdf:type ?teacherType .
  ?teacherType rdfs:subClassOf* ex:Teacher .
    ?teacher ex:teacherId ?teacherId .
    ?teacher ex:teacherName ?teacherName .

    # Find TeacherHours instances related to teaching assistants
    ?th a ex:TeacherHours .
    ?th ex:teacherHoursIn ?teacher .
    ?th ex:assignedHours ?assignedHourss .
    ?th ex:courseHoursIn ?courseInstance .

  ?courseInstance a ex:CourseInstance .
    ?courseInstance ex:studyPeriod 1 .
    ?courseInstance ex:courseInstanceAcademicYear "2018-2019" .
    ?courseInstance ex:courseInstanceOf ?course .

    ?course a ex:Course .
    ?course ex:courseCode 1015 .

  FILTER (?assignedHourss > 120)
```

```
}
```

Result:

teacherId	teacherName	assignedHours
19580218-0007	Teacher7	280.0
19790702-0038	TA38	140.0
19660630-0020	Teacher20	240.0
19650303-0019	Teacher19	240.0
19750102-0059	TA59	140.0

- **Question 4: find all students registered for course instance I-910 that were not registered for course instance I-911.**

  SPARQL query:

```
PREFIX ex: <http://assignment3.org/ontology#>

select ?studentId ?studentName
where {
    ?registration a ex:Registration .
    ?registration ex:studentRegistered ?student .
    ?registration ex:registrationContainsInstance ?courseInstance .

    ?courseInstance a ex:CourseInstance .
    ?courseInstance ex:instanceId "I-910" .

    ?student a ex:Student .
    ?student ex:studentId ?studentId .
    ?student ex:studentName ?studentName .

    minus {
        ?registration a ex:Registration .
        ?registration ex:studentRegistered ?student .
        ?registration ex:registrationContainsInstance ?courseInstance .

        ?courseInstance a ex:CourseInstance .
        ?courseInstance ex:instanceId "I-911" .

        ?student a ex:Student .
        ?student ex:studentId ?studentId .
        ?student ex:studentName ?studentName .
  }
}
```

Result:

studentId	studentName
19921201-0094	TA94

- **Question 5: find all programmes along with the total number of owned courses. List the results in descending order of number of owned courses.**

  SPARQL query:

```
PREFIX ex: <http://assignment3.org/ontology#>

SELECT ?programmeCode ?programmeName ?numberOfCourses
WHERE {
```

```
    ?programme a ex:Programme .
    ?programme ex:programmeCode ?programmeCode .
    ?programme ex:programmeName ?programmeName .

    {select ?programme (COUNT(?course) AS ?numberOfCourses)
        where {
            ?course a ex:Course .
            ?course ex:courseOwnedBy ?programme .
        }
        group by ?programme
    }

}
ORDER BY DESC(?numberOfCourses)
```

Result:

programmeCode	programmeName	numberOfCourses
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
10033	P-33	12
10012	P-12	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

- **Question 6a: find the number of: senior teachers**

  SPARQL query:

```
PREFIX ex: <http://assignment3.org/ontology#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

select (COUNT(DISTINCT ?teacher) AS ?numberOfTeacher)
where {
    ?teacher a ex:SeniorTeacher
}
```

Result:

numberOfTeacher
30

- **Question 6b: find the number of: all people**

  SPARQL query:

```
PREFIX ex: <http://assignment3.org/ontology#>

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT (COUNT(DISTINCT ?id) AS ?numberOfPeople)
WHERE {
    {
        ?teacher rdf:type/rdfs:subClassOf* ex:Teacher .
        ?teacher ex:teacherId ?id .
    } UNION {
        ?student rdf:type ex:Student .
        ?student ex:studentId ?id .
    }
}
```

  Result:

numberOfPeople
440