Πανεπιστήμιο Κρήτης

Τμήμα Επιστήμης Υπολογιστών

HY561 – Διαχείριση Δεδομένων στον Παγκόσμιο Ιστό

Εξάμηνο: Άνοιξη 2022

Γραπτή Αναφορά Έργου

Στοιχεία Φοιτητών

Μέλος	1ο
Ονοματωπώνυμο	Nikolaos Gounakis
AM	1254
Email	nicolaig@csd.uoc.gr

Πίνακας Περιεχομένων

1 Introduction

In this technical report we present the work done that represents the CS-561 Project. The general steps are the creation of an Ontology, based on an existing one, the development of python scripts in order to transform data of various sources in RDF and link them with the ontology, and lastly the Triple store and the queries. We tried to use the Facetize tool but we encountered errors when creating Facets or exporting the project. We used PascoLink in order to visualize the triplestore.

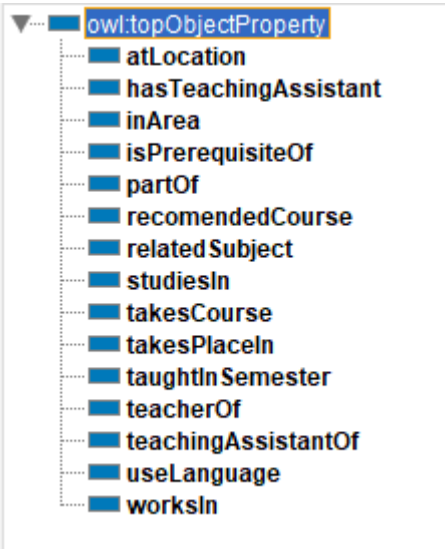
2 Query Requirements

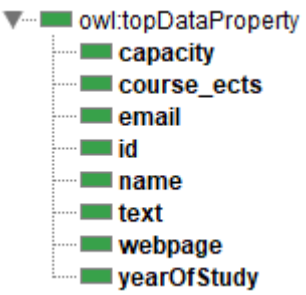
In this section we define the query requirements, in order to show a fraction of what information we can deduct from this operation.

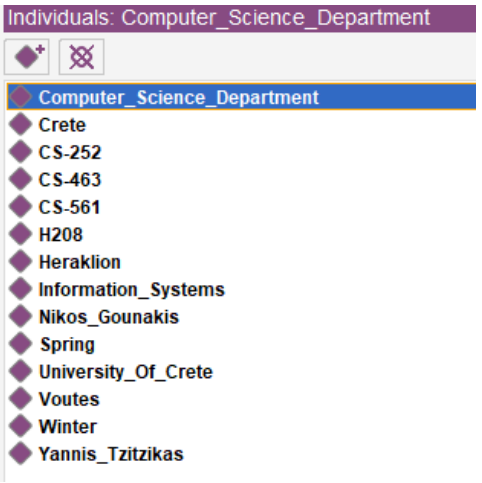
1. List of all courses (undergraduate and postgraduate) in the area "Information Systems" sorted by year of study showing the pre-requisites of each course.
2. All courses in which the "HY252" is a prerequisite.
3. What is the total capacity of student seats in CSD classrooms?
4. How many lessons each programming language is taught / used in total (in descending order)
5. The designers of programming languages of courses in the area of "Information Systems".
6. Artists that have created a song that matches a programming language name that is taught at the 4th year or less
7. Percentage of programming language coverage that csd provides (#csdProgrammingLannguages / #totalProgrammingLanguages)

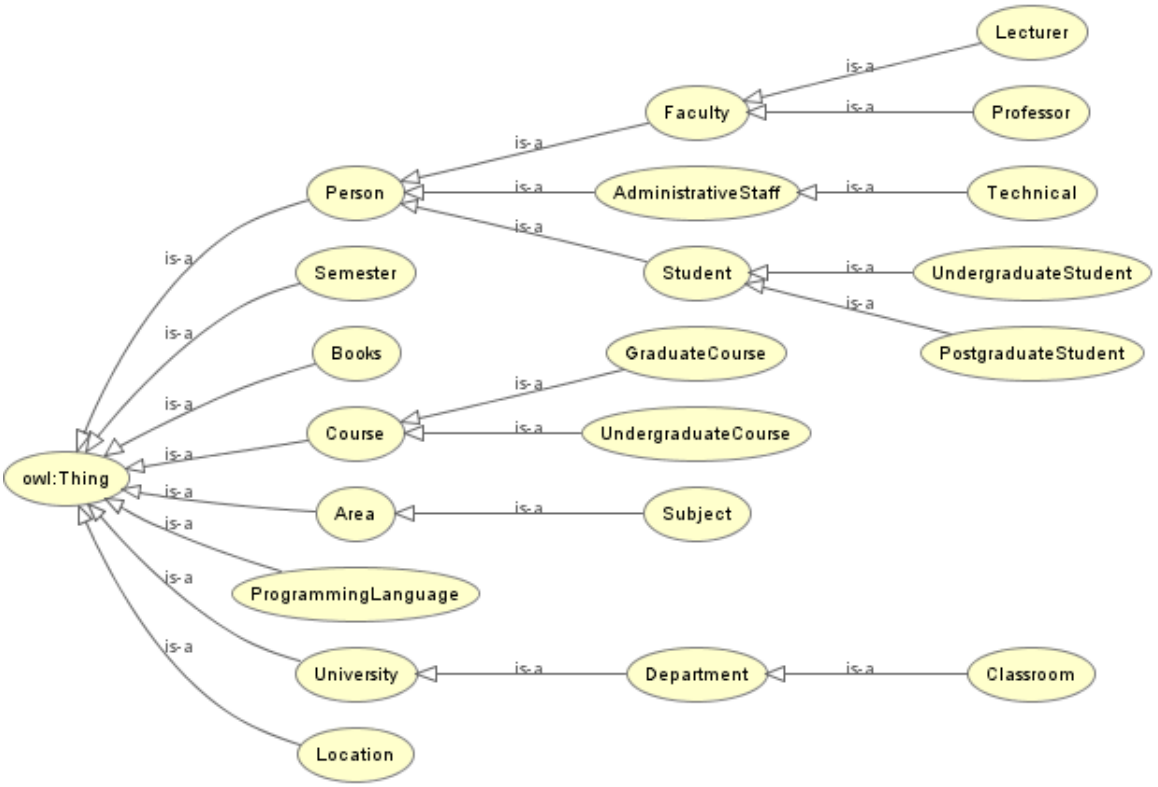
3 Ontology Definition

We consider the following ontology: SHOE. We used Protege for the creation of the ontology. We did not find any file or download link so we build the ontology from scratch based on the SHOE University Ontology. In the images below we can see the Class Hierarchy of the ontology. The ontology is described by the uri: http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#.
We refer to the ontology with the prefix **uni:**
Apart from the classes, we can see the relations and the data properties that we have created. Lastly, we created some individuals that the sources could not provide information. This was in order to show a more complete ontology. We must mention that the book class was not used as we did not find data associated with books that we can connect to a course.
.









4 Query Formulation

In this section we define the queries proposed in **Section 2** as SPARQL queries.

1. List of all courses (undergraduate and postgraduate) in the area "Information Systems" sorted by year of study showing the pre-requisites of each course.

```
DEFINE input:inference <http://localhost:8890/uni>
DEFINE input:same-as "yes"

SELECT DISTINCT ?courses ?yearStudy (GROUP_CONCAT(DISTINCT
?pre_names,',') as ?pre)
WHERE {
?courses rdf:type uni:Course ;
         uni:inArea uni:Information_Systems ;
         uni:yearOfStudy ?yearStudy .
?pre uni:isPrerequisiteOf ?courses ;
     uni:name ?pre_names .
}ORDER BY ?yearStudy
```

2. All courses in which the "HY252" is a prerequisite.

```
DEFINE input:inference <http://localhost:8890/uni>
DEFINE input:same-as "yes"

SELECT DISTINCT ?pre ?pre_names
WHERE {
```

```
uni:CS-252 uni:isPrerequisiteOf ?pre .
?pre uni:name ?pre_names .
}
```

3. What is the total capacity of student seats in CSD classrooms?

```
DEFINE input:inference <http://localhost:8890/uni>
DEFINE input:same-as "yes"

SELECT DISTINCT SUM(?cap) as ?total_capacity
WHERE {
?class rdf:type uni:Classroom ;
       uni:capacity ?cap .
}
```

4. How many lessons each programming language is taught / used in total (in descending order)

```
DEFINE input:inference <http://localhost:8890/uni>
DEFINE input:same-as "yes"

SELECT DISTINCT ?prog COUNT(?prog) as ?count
WHERE
{
?courses uni:useLanguage ?prog
} ORDER BY DESC(?count)
```

5. The designers of programming languages of courses in the area of “Information Systems”.

```
DEFINE input:inference <http://localhost:8890/uni>
DEFINE input:same-as "yes"

SELECT DISTINCT ?designer ?plang
WHERE
{
?courses rdf:type uni:Course ;
         uni:inArea uni:Information_Systems ;
         uni:useLanguage ?plang .
?plang dbo:designer ?designer .
}
```

6. Artists that have created a song that matches a programming language name that is taught at the 4th year or less

```
DEFINE input:inference <http://localhost:8890/uni>
DEFINE input:same-as "yes"

SELECT DISTINCT ?artist ?prog
WHERE
{
?prog rdf:type uni:ProgrammingLanguage ;
      uni:name ?name .
?course uni:useLanguage ?prog ;
        uni:yearOfStudy ?y .
FILTER(?y <= str(4))
SERVICE <http://dbtune.org/musicbrainz/sparql>
 {
  ?s rdfs:label ?prog ;
     foaf:maker ?maker .
  ?maker rdfs:label ?artist
 }
}
```

7. Percentage of programming language coverage that csd provides

```
DEFINE input:inference <http://localhost:8890/uni>
DEFINE input:same-as "yes"

SELECT DISTINCT (100*?csd/?prog) as ?coverage ?csd ?prog
WHERE
{
 {
  SELECT DISTINCT COUNT(?csdLang) as ?csd
  WHERE {
  ?course uni:useLanguage ?csdLang
  }
 }
 {
  SELECT DISTINCT COUNT(?progLang) as ?prog
  WHERE {
  ?progLang rdf:type uni:ProgrammingLanguage
 }
}
}
```

5 Source/Dataset Selection

Here we describe the sources we used in order to enrich the ontology with data (convert to RDF) and have some extra data in order to link them later.

5.1 CSD-courses.json

It contains data about the courses of csd like the course name, code, text, ects and more.

5.2 CSD-coursesProgrammingLanguage.csv

It contains information about what programming language each course uses.

5.3 CSD-people

It contains information about professors of CSD. More specifically it contains names, text, emails, etc.

5.4 SubjectsForPerioxiD.txt

Contains course-relevant subjects in the area of "Information Systems"

5.5 Other

We are provided with acm_Classification.n3 , subjects.ttl and programmingLanguages.ttl from external sources (DBpedia etc.) that contain information about a hierarchy that contains "Information Systems" and subjects related to the area and programming languages.

6 Conversion to RDF

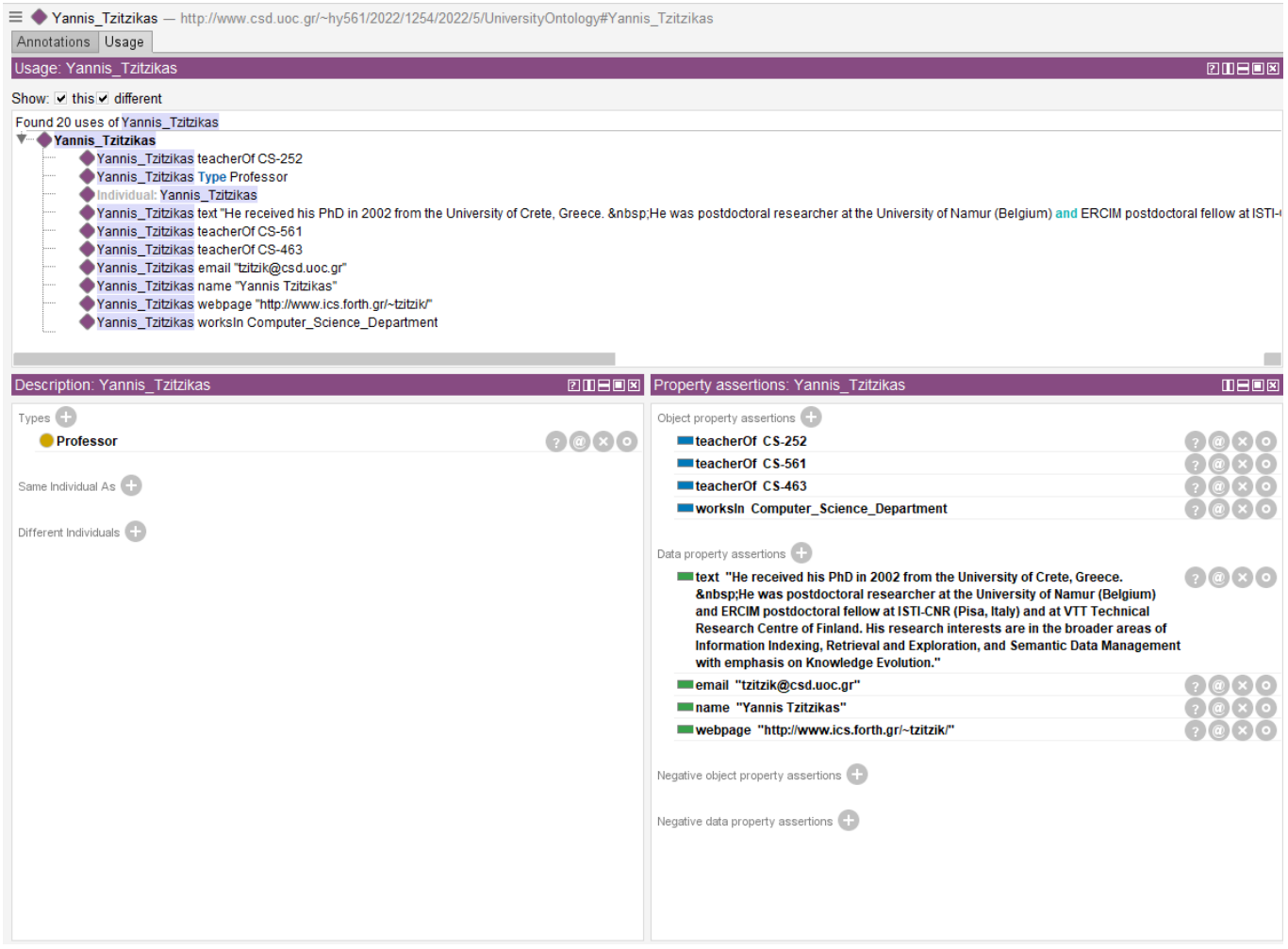
Here we describe the methodology we used in order to link the provided data about CSD.

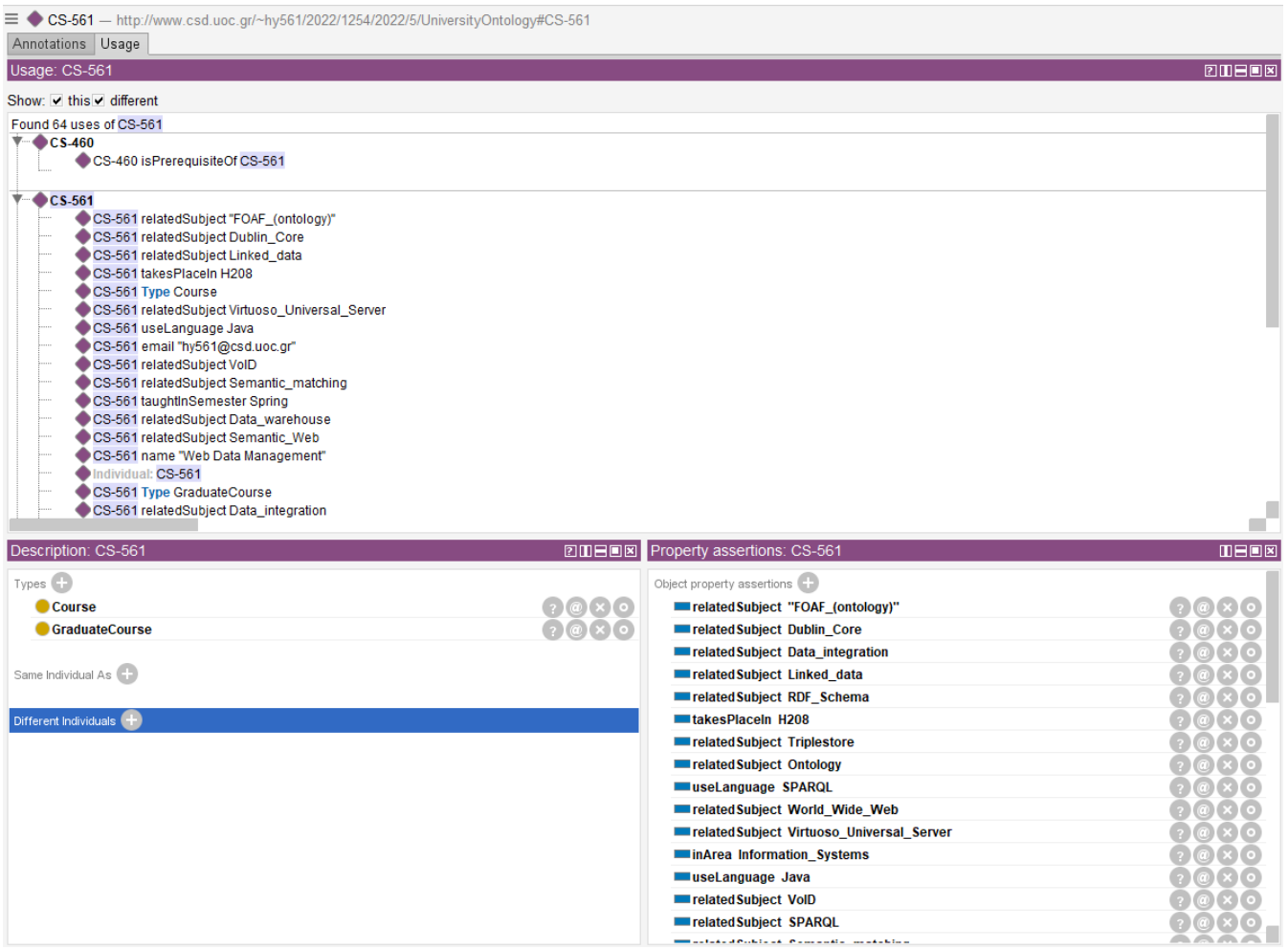
We write python scripts that uses the RDFlib. We parse the owl file we created that contains the ontology which is in RDF format. Then we read each desired dataset and according to the data, we add triples and relations to the graph and export the "enhanced" version of the ontology as we call it. So in the end we can open the **enhanced.owl** file and see all the added data (individuals, relations, data properties) in protege.

important: we must add the following line in order to open it in protege.

```
<owl:Ontology
rdf:about="http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology"/>
```

Here we can see 2 examples of the final file, one with **Yiannis Tzitzikas** and one with the course **CS-561.**


Yannis_Tzitzikas — http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#Yannis_Tzitzikas
Annotations
Usage
Usage: Yannis_Tzitzikas
Show: this different
Found 20 uses of Yannis_Tzitzikas
Yannis_Tzitzikas
Yannis_Tzitzikas teacherOf CS-252
Yannis_Tzitzikas Type Professor
Individual: Yannis_Tzitzikas
Yannis_Tzitzikas text "He received his PhD in 2002 from the University of Crete, Greece. He was postdoctoral researcher at the University of Namur (Belgium) and ERCIM postdoctoral fellow at ISTI-
Yannis_Tzitzikas teacherOf CS-561
Yannis_Tzitzikas teacherOf CS-463
Yannis_Tzitzikas email "tzitzik@csd.uoc.gr"
Yannis_Tzitzikas name "Yannis Tzitzikas"
Yannis_Tzitzikas webpage "http://www.ics.forth.gr/~tzitzik/"
Yannis_Tzitzikas worksIn Computer_Science_Department
Description: Yannis_Tzitzikas
Types
Professor
Same Individual As
Different Individuals
Property assertions: Yannis_Tzitzikas
Object property assertions
teacherOf CS-252
teacherOf CS-561
teacherOf CS-463
worksIn Computer_Science_Department
Data property assertions
text "He received his PhD in 2002 from the University of Crete, Greece. He was postdoctoral researcher at the University of Namur (Belgium) and ERCIM postdoctoral fellow at ISTI-CNR (Pisa, Italy) and at VTT Technical Research Centre of Finland. His research interests are in the broader areas of Information Indexing, Retrieval and Exploration, and Semantic Data Management with emphasis on Knowledge Evolution."
email "tzitzik@csd.uoc.gr"
name "Yannis Tzitzikas"
webpage "http://www.ics.forth.gr/~tzitzik/"
Negative object property assertions
Negative data property assertions


CS-561 — http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#CS-561
Annotations
Usage
Usage: CS-561
Show: this different
Found 64 uses of CS-561
CS-460
CS-460 isPrerequisiteOf CS-561
CS-561
CS-561 relatedSubject "FOAF_(ontology)"
CS-561 relatedSubject Dublin_Core
CS-561 relatedSubject Linked_data
CS-561 takesPlaceIn H208
CS-561 Type Course
CS-561 relatedSubject Virtuoso_Universal_Server
CS-561 useLanguage Java
CS-561 email "hy561@csd.uoc.gr"
CS-561 relatedSubject VoID
CS-561 relatedSubject Semantic_matching
CS-561 taughtInSemester Spring
CS-561 relatedSubject Data_warehouse
CS-561 relatedSubject Semantic_Web
CS-561 name "Web Data Management"
Individual: CS-561
CS-561 Type GraduateCourse
CS-561 relatedSubject Data_integration
Description: CS-561
Types
Course
GraduateCourse
Same Individual As
Different Individuals
Property assertions: CS-561
Object property assertions
relatedSubject "FOAF_(ontology)"
relatedSubject Dublin_Core
relatedSubject Data_integration
relatedSubject Linked_data
relatedSubject RDF_Schema
takesPlaceIn H208
relatedSubject Triplestore
relatedSubject Ontology
useLanguage SPARQL
relatedSubject World_Wide_Web
relatedSubject Virtuoso_Universal_Server
inArea Information_Systems
useLanguage Java
relatedSubject VoID
relatedSubject SPARQL

The other data are already in RDF so they are not involved in the process.

7 Schema Mappings

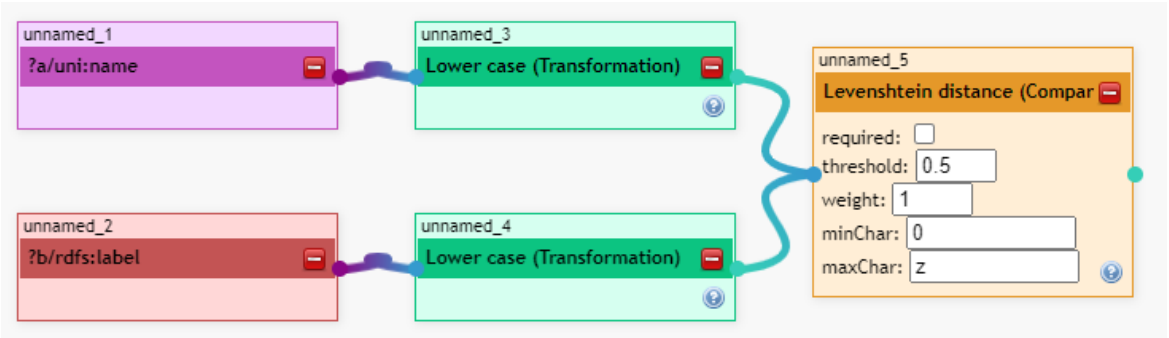
The final file from protege already follows the rdf schema so the only thing that we add is 3 triples.

```
<http://dbpedia.org/ontology/ProgrammingLanguage>
<http://www.w3.org/2002/07/owl#sameAs>
<http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#Pro
grammingLanguage> .
<http://dbpedia.org/class/yago/Abstraction100002137>
<http://www.w3.org/2002/07/owl#sameAs>
<http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#Sub
ject> .
<http://www.w3.org/2000/01/rdf-schema#label>
<http://www.w3.org/2002/07/owl#sameAs>
<http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#nam
e> .
```

Briefly we connect the DBpedia ProgrammingLanguage class with our own, we connect the Abstraction100002137 with our class Subject and finally we connect our data property named “name” to the rdfs:label.

8 Instance Matching - Silk

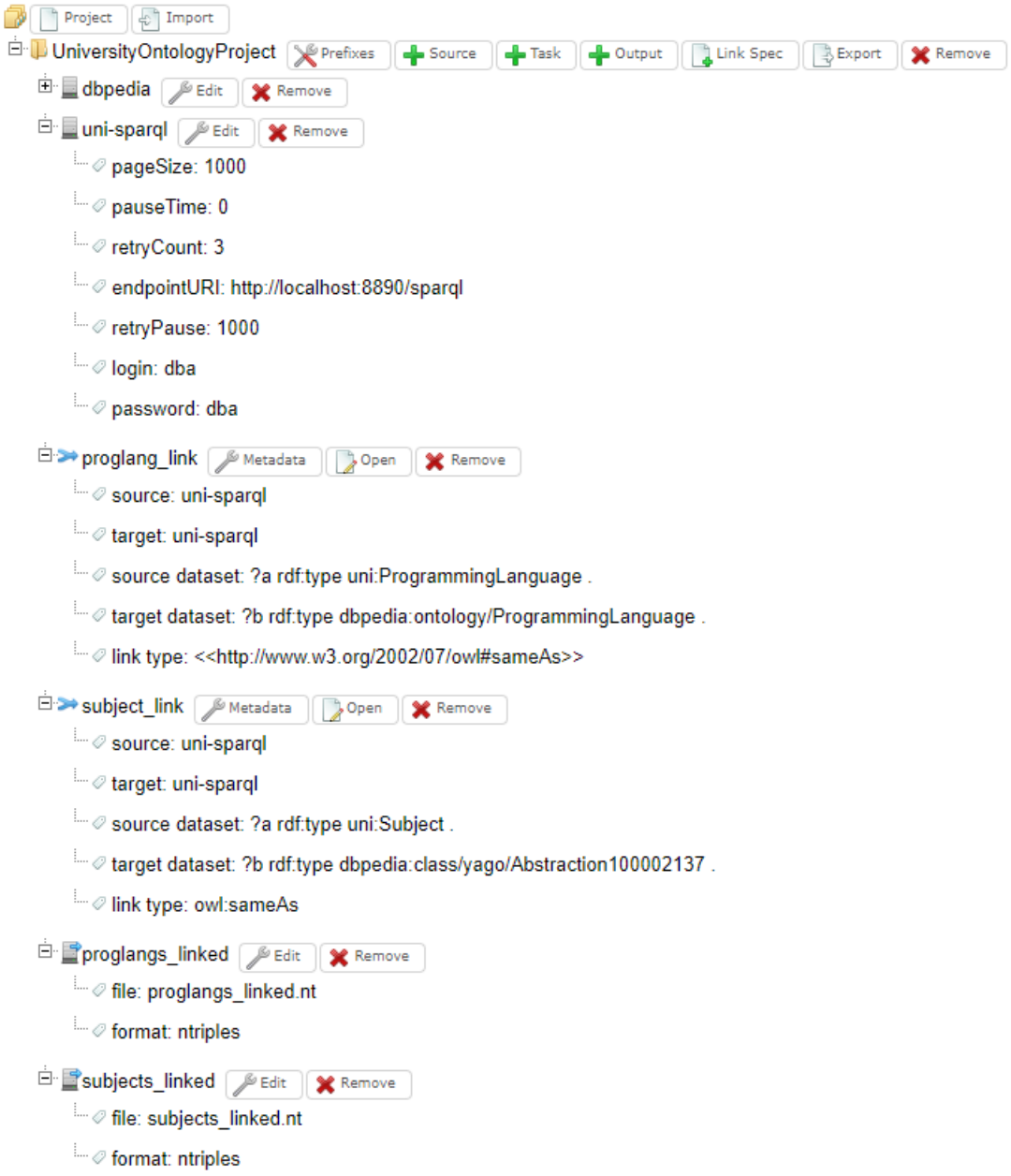
Using silk our main goal is to link the subjects we extracted from the csd-dataset to the DBpedia provided dataset and do the same also for the programming languages.
We used the same rules. we fetch the rdfs:label and the uni:name of each source, we used the lowercase function, and then we use Levenstein distance.



we get results like this:

Source: uni-sparql	Target: uni-sparql	Score	Correct?
uni:OCaml	dbpedia:resource/OCaml	100.0%	
uni:Javascript	dbpedia:resource/JavaScript	100.0%	
uni:Java	dbpedia:resource/Java_(programming_language)	100.0%	

Now we display the rules of each task we created and the project configurations:



9 Triplestore

After all that we have the enhanced ontology file, the rdf data that was provided from the start, and the new files that contains the same as triples for subjects and programming languages. We used the Virtuoso Conductor as we set it up in a previous exercise of the course. We upload each file in the graph name **uni**. After uploading all the rdf files we have a total of 91559 triples.

10 Queries

In this section we test the queries we mention in **Section 4.**

10.1 Normal Queries

1. This query displays each course and its prerequisites along with the year of study.

courses	yearStudy	pre
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#CS-359	3	Object-Oriented Programming
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#CS-358	3	Data Structures
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#CS-463	4	Data Structures
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#CS-460	4	Files and Databases
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#CS-482	4	Algorithms and Complexity,Linear Algebra,Probability
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#CS-561	5	Database Management Systems
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#CS-567	5	Algorithms and Complexity
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#CS-553	5	Computer Graphics
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#CS-558	5	Operating Systems
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#CS-583	5	Algorithms and Complexity,Data Structures
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#CS-540	5	Languages and Compilers
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#CS-565	5	Files and Databases
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#CS-577	5	Algorithms and Complexity,Probability,Programming
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#CS-543	5	Files and Databases,Object-Oriented Programming
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#CS-562	5	Files and Databases

2. This query returns the courses that CS252 is a prerequisite

pre	pre_names
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#CS-351	Information System Analysis and Design
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#CS-342	Parallel Programming
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#CS-352	Software Engineering
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#CS-446	Managed Runtime Systems
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#CS-543	Software Systems and Technologies for Big Data Applications
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#CS-359	Web Programming

3. Here we can see the total capacity of classrooms in CSD

total_capacity
669

4. In this query, we see how many courses each language is used. E.g. Java is used in 17 courses

prog	count
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#Java	17
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#C++	17
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#C	17
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#Matlab	14
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#SQL	4
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#Javascript	3
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#Ajax	2
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#Verilog	2
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#R	2
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#XSLT	2
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#XPATH	2
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#XQUERY	2
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#Scala	2
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#CSharp	1
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#Matab	1
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#SPARQL	1
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#PHP	1
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#Assembly	1
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#JDL(JobDescriptionLanguage)	1
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#OCaml	1
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#Prolog	1
http://www.csd.uoc.gr/~hy561/2022/1254/2022/5/UniversityOntology#Python	1

5. Here we can see the designers of each programming language in the area of ”Information Systems”

designer	plang
http://dbpedia.org/resource/James_Gosling	http://dbpedia.org/resource/Java_(programming_language)
http://dbpedia.org/resource/Bjarne_Stroustrup	http://dbpedia.org/resource/C++
http://dbpedia.org/resource/Raymond_F._Boyce	http://dbpedia.org/resource/SQL
http://dbpedia.org/resource/Donald_D._Chamberlin	http://dbpedia.org/resource/SQL
http://dbpedia.org/resource/Martin_Odersky	http://dbpedia.org/resource/Scala_(programming_language)
http://dbpedia.org/resource/Alain_Colmerauer	http://dbpedia.org/resource/Prolog
http://dbpedia.org/resource/Brendan_Eich	http://dbpedia.org/resource/JavaScript
http://dbpedia.org/resource/Jesse_James_Garrett	http://dbpedia.org/resource/Ajax_(programming)
http://dbpedia.org/resource/World_Wide_Web_Consortium	http://dbpedia.org/resource/XQuery

6. We found that the programming language coverage for CSD is 8%. That means the CSD covers 8% of all the programming languages we have in our triplestore.

10.2 Federated Queries

As we encountered a problem and could not query another SPARQL endpoint from ours, this query is more theoretical. We want to fetch artists from MusicBrainz who have created a song that matches the rdfs:label of a programming language that is taught in the 4th year or less. We have formulated a query but unfortunately, the SERVICE keyword seems not to work as we expected. So we query the MusicBrainz in a separate query to show what the results would look like. In the following image, we see all artists that have created a song named “Java”.So if instead of the fixed word “Java” we have a variable as in the query we would get results for all languages that matches the condition that they are taught in the 4th year.

```
SELECT DISTINCT ?artist WHERE {
  ?s rdfs:label "Java" ;
     foaf:maker ?maker .
  ?maker rdfs:label ?artist
}
```

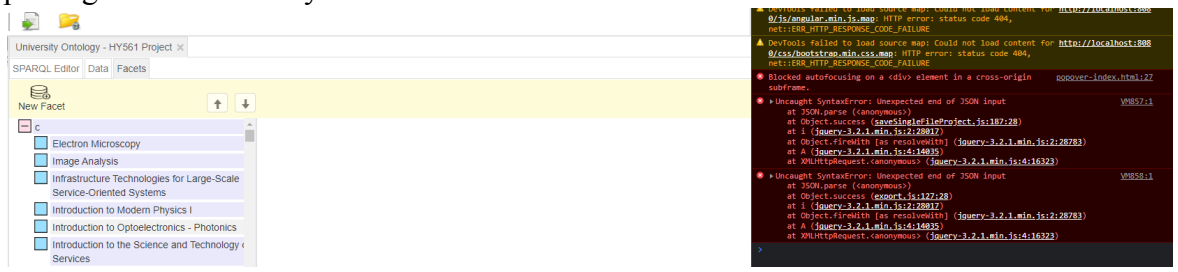
Results: Browse Go! Reset

SPARQL results:

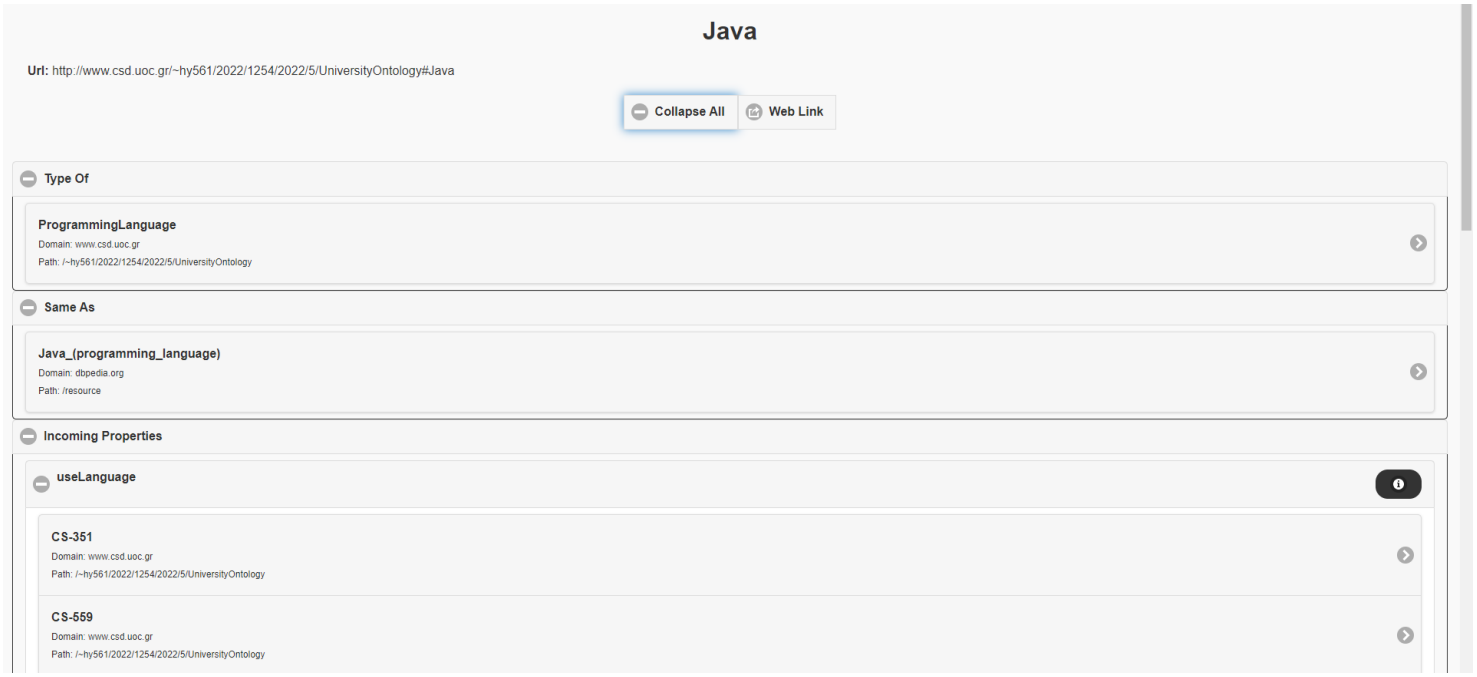
artist
"Al Hirt"
"Q Connection"
"Les Parisiennes"
"Jumping Jewels"
"Kenny Larkin"
"Dingle"
"Augustus Pablo"
"Franz Lambert"
"Cusco"
"Bob Crosby"
"André Popp"
"Jean Wiéner"
"Les Croquants"
"Floyd Cramer"
"The Halibuts"
"Bill Black's Combo"
"James Last"
"The Beautiful South"

11 Visualization and Data exploration

In this section we talk about the visualization and the exploration of the triplestore. We tried to use Facetize in order to create facets for courses and programming languages and explore the in Hippalus but we could not export the file or create groups due to parsing errors that the system could not handle.



So we used PascoLink in order to have visualization and a keyword search functionality. After a keyword search, we found the Programming Language Java:



Here we can see information for the CS359 course:

12 Conclusion

We provide an implementation of the University Ontology from SHOE. The ontology is not perfect thus it can be further extended in order to support more complicated situations. We created a triple store that links programming languages and subjects from DBpedia and CSD datasets. The users have not very much flexibility with the views we provide but

an engineer could use this triple store and deduct information about the department its people, courses, and students that are not visible through the chaotic representation and the numerous data formats. We wanted to give a faceted exploration of courses structured in graduate and undergratuate groups and a department group for courses from other departments, but we faced difficulties with the Facetise tool. We believe that our work is a good base for further enrichment.