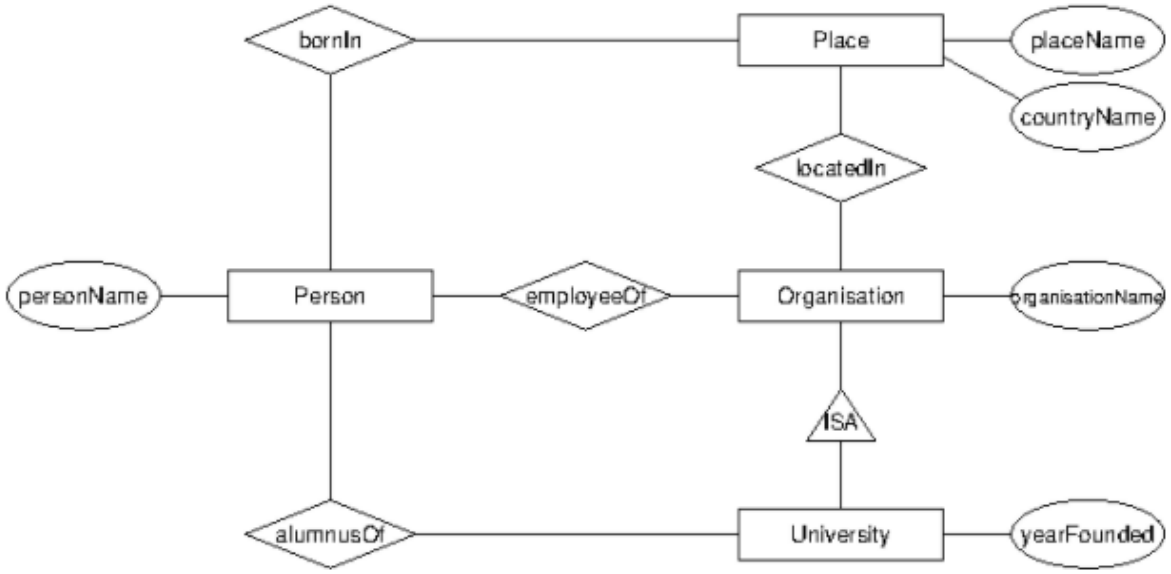
CHALMERS
UNIVERSITY OF TECHNOLOGY

Assignment 4

Techniques for large-scale data

Alejandro Corrochano, Yossra Gharbi
(Group 34)

Assignment 4| Exercises



Exercises

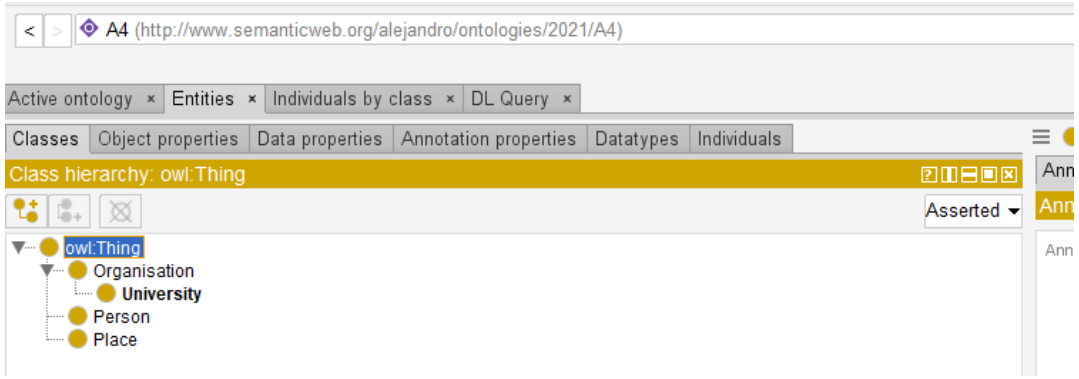
1. Create RDFS or OWL2 statements that correspond to the classes, relationships and attributes that are shown in the E-R diagram

2. Populate the GraphDB database with data extracted from Wikidata

3. Write some SPARQL queries that explore the data in GraphDB

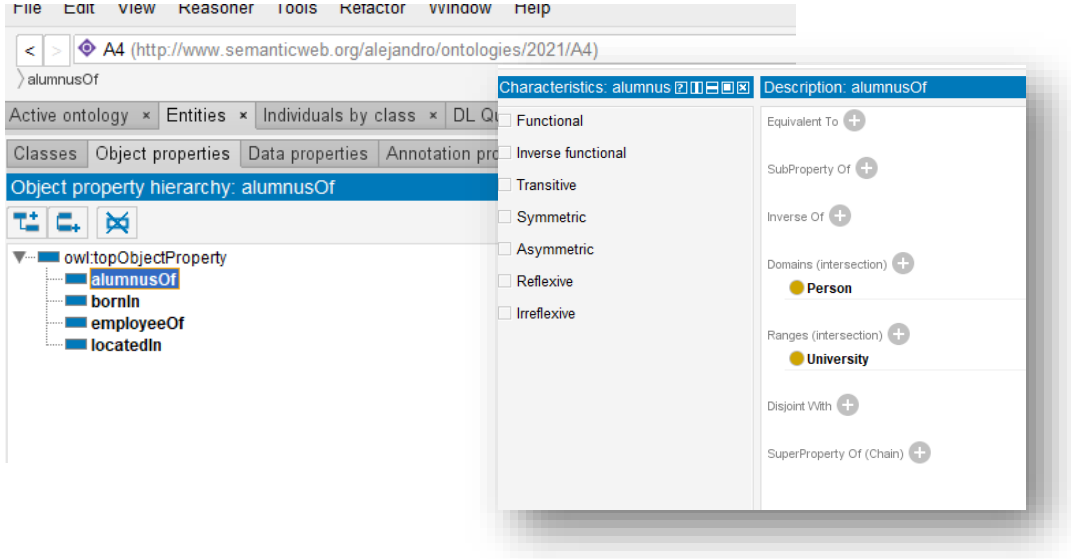
Exercise 1

For this exercise we made use of Protégé to avoid the cumbersome task of writing the OWL2 statements by hand. Below it can be seen the how we represented the E-R diagram.

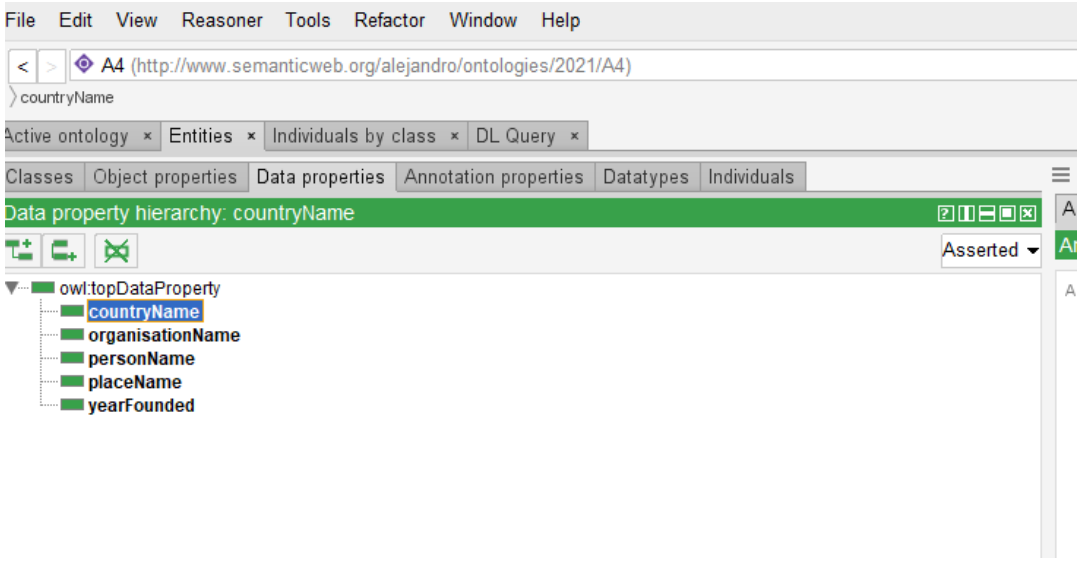


First, we created the classes, which are **Person**, **Place**, **University**, and **Organisation**. We specified **University as a subclass of Organisation**, following the representation in the diagram.

Next, we established the relationships between the classes. For instance, **alumnusOf** can be observed in the snippets, where we specify how the person is related to this type of organisation.



Exercise 1



Lastly, we created the attributes for each class.

- Person: **personName** – String
- University: **yearFounded** – Integer
- Organisation: **organisationName** – String
- Place: **placeName**, **countryName** - String

Exercise 2

Once we are done with the ontologies, we import the .owl file into a GraphDB repository created beforehand.

Immediately after uploading the file, this is the first Class hierarchy diagram we are able to see. Gray circle corresponds to the University class.

Since no Person or Place instances have been yet created, no figures representing them are shown.

Exercise 2 | Observations

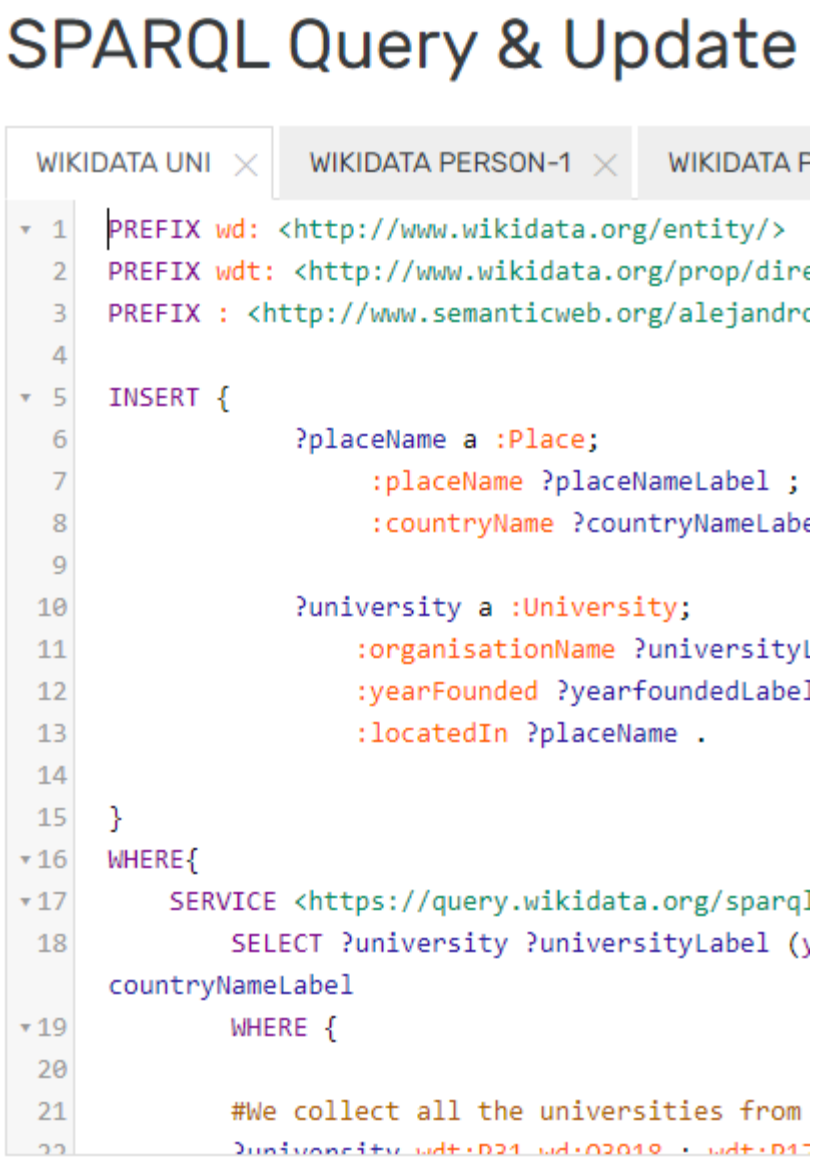
CHALMERS
UNIVERSITY OF TECHNOLOGY

In order to fill our repository, we move into the SPARQL queries section, where we will connect with the Wikidata service to retrieve all the information related to alumni of Swedish universities and their respective employers.

Something we noticed during the queries development and check of the results, was that if we specified many relationships among the different classes at once, the instances who missed any of the properties we were looking for would not appear, hence the total number of instances would be much lower.

For example, if we had tried to retrieve all the alumni of Swedish universities that were born at any location, everything within the same query, we would miss those that were alumni but had no information regarding its birthplace.

To solve this issue we executed many queries to collect all the desired information little by little.



Exercise 2 | Universities

CHALMERS UNIVERSITY OF TECHNOLOGY

```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX : <http://www.semanticweb.org/alejandro/ontologies/2021/A4#>

INSERT {
            ?placeName a :Place;
                 :placeName ?placeNameLabel ;
                 :countryName ?countryNameLabel .

            ?university a :University;
                :organisationName ?universityLabel ;
                :yearFounded ?yearfoundedLabel ;
                :locatedIn ?placeName .

}
WHERE{
    SERVICE <https://query.wikidata.org/sparql>{
        SELECT ?university ?universityLabel (year(?yearfounded) as ?yearfoundedLabel) ?placeName ?placeNameLabel ?
countryNameLabel
        WHERE {
        #We collect all the universities from sweden
        ?university wdt:P31 wd:Q3918 ; wdt:P17 wd:Q34; wdt:P571 ?yearfounded .
        ?university wdt:P276 ?placeName .
        ?placeName wdt:P17 ?countryName .
        ?university <http://www.w3.org/2000/01/rdf-schema#label> ?universityLabel .
        ?placeName <http://www.w3.org/2000/01/rdf-schema#label> ?placeNameLabel .
        ?countryName <http://www.w3.org/2000/01/rdf-schema#label> ?countryNameLabel .

        filter(lang(?universityLabel) = "en")
        filter(lang(?placeNameLabel) = "en")
        filter(lang(?countryNameLabel) = "en")
```

Prefixes are repeated in every query (We will not show this part in the following slides)

We first insert the place where the university is located in, as we will use it in the next Insert statement.

We add every university that has been found with its name, year of foundation, and location.

We search for all the instances that are universities (wdt:P31 wd:Q3918), which also are Swedish (wdt:P17 wd:Q34)

Once we have all the universities, we just need to collect the desired data.
Year of foundation (wdt:P571)
Location (wdt:P276)
Country (wdt:P17)

Exercise 2 | Alumni (alumnusOf)

CHALMERS UNIVERSITY OF TECHNOLOGY

```
INSERT {
            ?inst a :University;
                :organisationName ?instLabel ;
                :yearFounded ?instYearFoundedLabel.

            ?person a :Person;
                :personName ?personLabel ;
                :alumnusOf ?inst .
}
WHERE{
    SERVICE <https://query.wikidata.org/sparql>{
        SELECT ?university ?person ?personLabel ?inst ?instLabel
        (year(?instYearFounded) as ?instYearFoundedLabel)
        WHERE {

        #We collect all the universities from sweden
        ?university wdt:P31 wd:Q3918 ; wdt:P17 wd:Q34.
        #Alumni of swedish universities
        ?person wdt:P69 ?university .
        #Institutions in which alumnis were educated at
        ?person wdt:P69 ?inst .
        #We might find universities where they studied apart from the swedish ones,
        # so the name and year founded is required as well

       ?person <http://www.w3.org/2000/01/rdf-schema#label> ?personLabel .
       ?inst <http://www.w3.org/2000/01/rdf-schema#label> ?instLabel .

       filter(lang(?personLabel) = "en")
       filter(lang(?instLabel) = "en")
       }
    }
}
```

We insert those universities that are not Swedish but in which the students were also alumni.

We create the person object and its relationship with the university. The location of both university and person (birthplace) will be added later, as some of them might miss that information.

Inst refers to “institution”. First, we get all the alumni of Swedish universities, and then we retrieve any other university outside the country.

Exercise 2 | Alumni (bornIn)

CHALMERS UNIVERSITY OF TECHNOLOGY

```
INSERT {
            ?locationPerson a :Place;
                :placeName ?locationPersonLabel ;
                :countryName ?locationCountryLabel.

            ?person a :Person;
                :bornIn ?locationPerson .
}
WHERE{
    SERVICE <https://query.wikidata.org/sparql>{
        SELECT ?university ?person ?inst ?locationPerson ?locationPersonLabel
        ?locationCountry ?locationCountryLabel
        WHERE {

        #We collect all the universities from sweden
        ?university wdt:P31 wd:Q3918 ; wdt:P17 wd:Q34.
        #Alumni of swedish universities
        ?person wdt:P69 ?university .
        #We get the birthplace and the country of the location
        ?person wdt:P19 ?locationPerson .
        ?locationPerson wdt:P17 ?locationCountry .

        ?locationPerson <http://www.w3.org/2000/01/rdf-schema#label> ?locationPersonLabel .
        ?locationCountry <http://www.w3.org/2000/01/rdf-schema#label> ?locationCountryLabel .

        filter(lang(?locationPersonLabel) = "en")
        filter(lang(?locationCountryLabel) = "en")
        }
    }
}
```

We now create the Place instances for those students who have it.

And we relate them.

Exercise 2 | Alumni (employeeOf)

CHALMERS
UNIVERSITY OF TECHNOLOGY

```
INSERT {
            ?employer a :Organisation;
                :organisationName ?employerLabel .

            ?person a :Person;
                :employeeOf ?employer .
}
WHERE{
    SERVICE <https://query.wikidata.org/sparql>{
        SELECT ?university ?person ?employer ?employerLabel
        WHERE {

        #We collect all the universities from sweden
        ?university wdt:P31 wd:Q3918 ; wdt:P17 wd:Q34.
        #Alumni of swedish universities
        ?person wdt:P69 ?university .
        #Employers of alumni
        ?person wdt:P108 ?employer .

        ?employer <http://www.w3.org/2000/01/rdf-schema#label> ?employerLabel .

        filter(lang(?employerLabel) = "en")
        }
    `
```

The employer is an organization with its respective name.

Relationship between student and employer is established.

Exercise 2 | Organisation (locatedIn)

CHALMERS UNIVERSITY OF TECHNOLOGY

```
INSERT {
            ?locationEmployer a :Place;
                :placeName ?locationEmployerLabel ;
                :countryName ?countryEmployerLabel .

            ?employer a :Organisation;
                :locatedIn ?locationEmployer.
}
WHERE{
    SERVICE <https://query.wikidata.org/sparql>{
        SELECT ?university ?person ?employer ?employerLabel ?locationEmployer
        ?locationEmployerLabel ?countryEmployer ?countryEmployerLabel
        WHERE {

        #We collect all the universities from sweden
        ?university wdt:P31 wd:Q3918 ; wdt:P17 wd:Q34.
        #Alumni of swedish universities
        ?person wdt:P69 ?university .
        #Employers of universities alumni
        ?person wdt:P108 ?employer .
        #We retrieve the location of each employer
        ?employer wdt:P276 ?locationEmployer .
        #We now collect the countries of the locations
        ?locationEmployer wdt:P17 ?countryEmployer .

        ?employer <http://www.w3.org/2000/01/rdf-schema#label> ?employerLabel .
        ?locationEmployer <http://www.w3.org/2000/01/rdf-schema#label> ?locationEmployerLabel .
        ?countryEmployer <http://www.w3.org/2000/01/rdf-schema#label> ?countryEmployerLabel .

        filter(lang(?employerLabel) = "en")
        filter(lang(?locationEmployerLabel) = "en")
        filter(lang(?countryEmployerLabel) = "en")
        }
    }
```

We create the place instance in those organisations who have it.

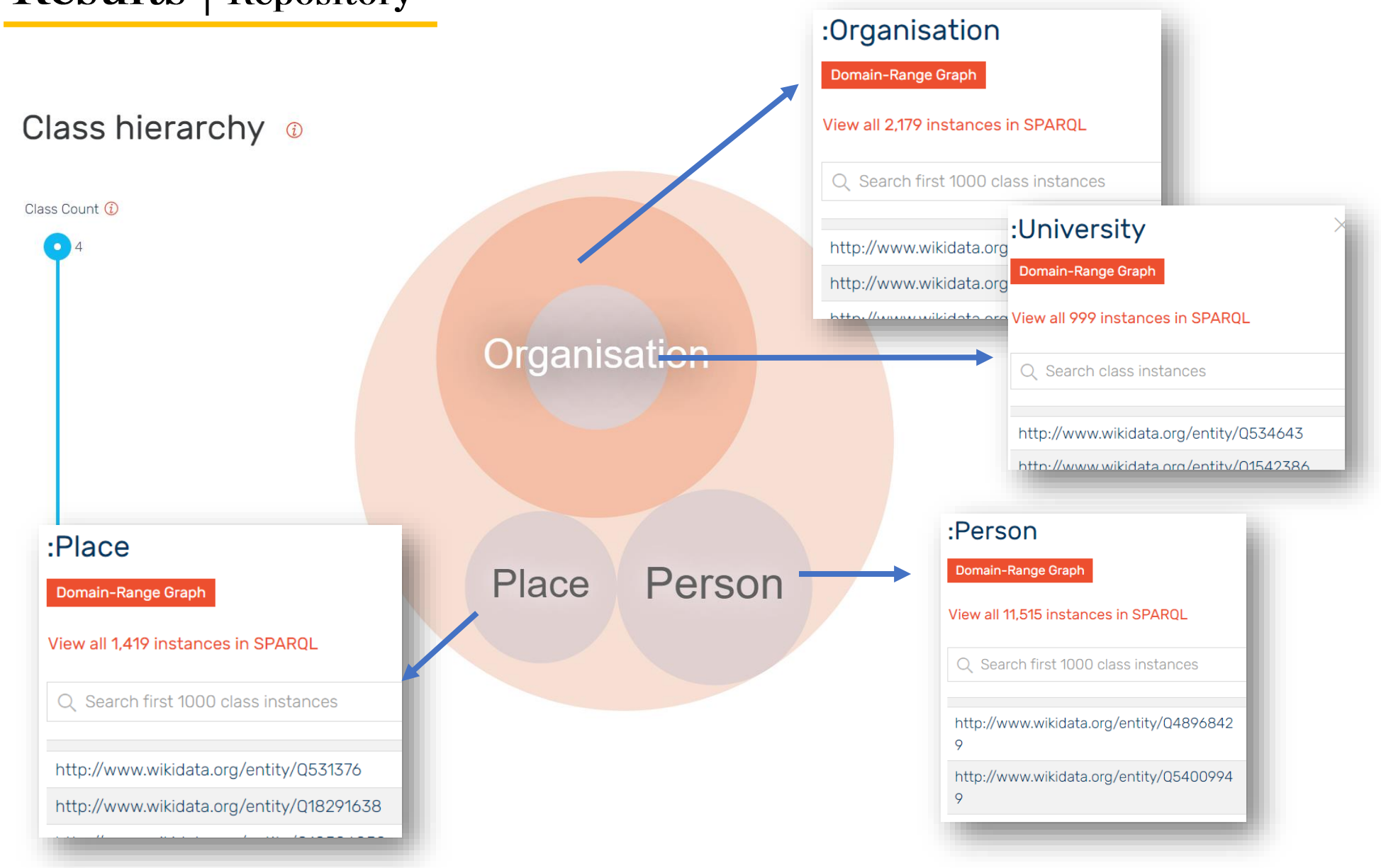
Exercise 2 | Universities (locatedIn)

CHALMERS
UNIVERSITY OF TECHNOLOGY

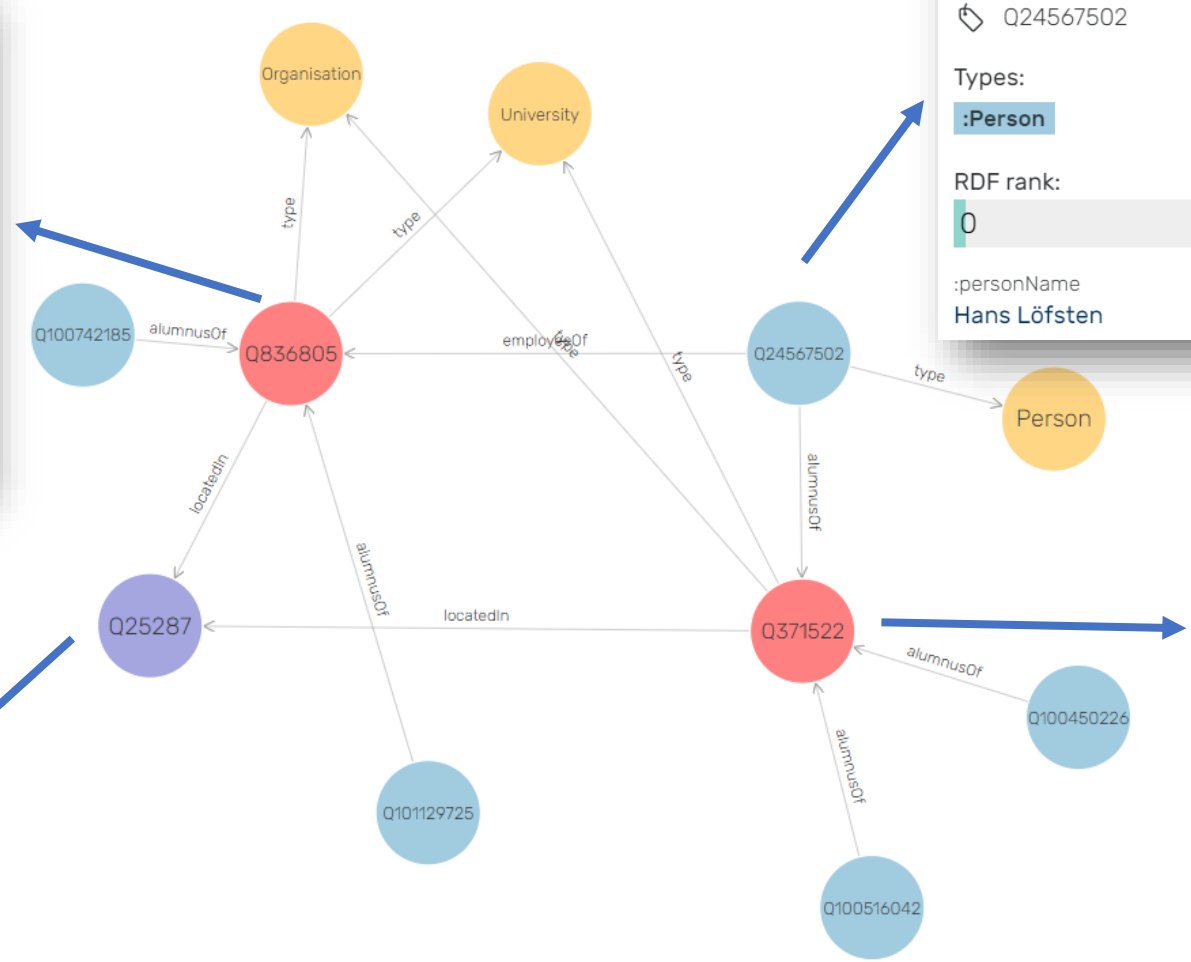
```
INSERT {
            ?locationInst a :Place;
                :placeName ?locationInstLabel;
                :countryName ?countryInstLabel.
            ?inst a :University;
                :locatedIn ?locationInst.
}
WHERE{
    SERVICE <https://query.wikidata.org/sparql>{
        SELECT ?university ?person ?inst ?locationInst ?locationInstLabel ?countryInst ?countryInstLabel
        WHERE {

        #We collect all the universities from sweden
        ?university wdt:P31 wd:Q3918 ; wdt:P17 wd:Q34.
        #Alumni of swedish universities
        ?person wdt:P69 ?university .
        #Institutions in which alumnis were educated at
        ?person wdt:P69 ?inst .
            #Location of remaining universities (not swedish)
        ?inst wdt:P276 ?locationInst .
        ?locationInst wdt:P17 ?countryInst .

        ?locationInst <http://www.w3.org/2000/01/rdf-schema#label> ?locationInstLabel .
        ?countryInst <http://www.w3.org/2000/01/rdf-schema#label> ?countryInstLabel .

        filter(lang(?locationInstLabel) = "en")
        filter(lang(?countryInstLabel) = "en")
        }
    }
}
```

We create the Place instances we are missing for some universities located outside Sweden.

Results | Repository



Results | Visual graph example



Q836805

Q836805

Types:

:University

RDF rank:

0

:organisationName

Chalmers University of Technology

:yearFounded

1829

Q24567502

Q24567502

Types:

:Person

RDF rank:

0

:personName

Hans Löfsten

Q25287

Q25287

Types:

:Place

RDF rank:

0

:countryName

Sweden

:placeName

Gothenburg

Q371522

Q371522

Types:

:University

RDF rank:

0

:organisationName

University of Gothenburg

:yearFounded

1954

Exercise 3 | Queries

CHALMERS
UNIVERSITY OF TECHNOLOGY

First query will look for the people who studied and worked at the same organization, hence it has to be a university

```
SELECT ?name ?university ?organisation
WHERE{
    ?university rdf:type :University.
    ?organisation rdf:type :Organisation.
    ?person rdf:type :Person;
            :personName ?name;
            :alumnusOf ?university;
            :employeeOf ?organisation.
    FILTER (?university = ?organisation)
}
```

Filter query results

Showing results from 1 to 1,000 of 2,754. Query took 0.1s, minutes

	name	university	organisation
1	"Carl Curman"@en	http://www.wikidata.org/entity/Q167920	http://www.wikidata.org/entity/Q167920
2	"Hinrich Bitter-Suermann"@en	http://www.wikidata.org/entity/Q371522	http://www.wikidata.org/entity/Q371522
3	"Arne Tiselius"@en	http://www.wikidata.org/entity/Q185246	http://www.wikidata.org/entity/Q185246
4	"Eva Gothlin"@en	http://www.wikidata.org/entity/Q371522	http://www.wikidata.org/entity/Q371522
5	"Carl Linnaeus"@en	http://www.wikidata.org/entity/Q185246	http://www.wikidata.org/entity/Q185246
6	"Anita Hansbo"@en	http://www.wikidata.org/entity/Q371522	http://www.wikidata.org/entity/Q371522
7	"Gunhild Kyle"@en	http://www.wikidata.org/entity/Q371522	http://www.wikidata.org/entity/Q371522
8	"Lotta Lotass"@en	http://www.wikidata.org/entity/Q371522	http://www.wikidata.org/entity/Q371522
9	"Cecilia Malmström"@en	http://www.wikidata.org/entity/Q371522	http://www.wikidata.org/entity/Q371522
10	"Gunnar D Hansson"@en	http://www.wikidata.org/entity/Q371522	http://www.wikidata.org/entity/Q371522
11	"Jan-Eric Gustafsson"@en	http://www.wikidata.org/entity/Q371522	http://www.wikidata.org/entity/Q371522
12	"Jan Lötvall"@en	http://www.wikidata.org/entity/Q371522	http://www.wikidata.org/entity/Q371522
13	"Osborne Bartley"@en	http://www.wikidata.org/entity/Q371522	http://www.wikidata.org/entity/Q371522

Exercise 3 | Queries

CHALMERS UNIVERSITY OF TECHNOLOGY

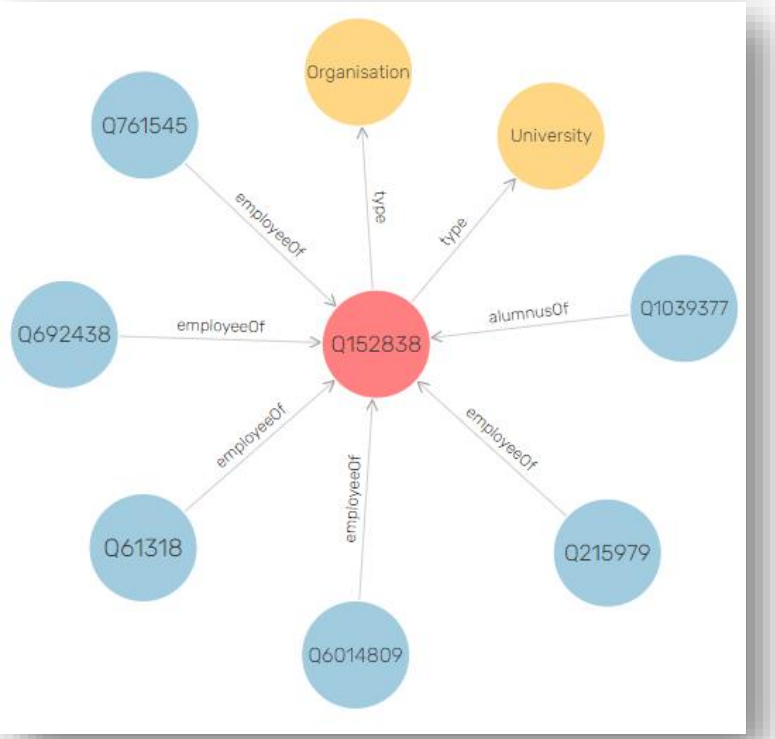
Second query finds how many alumni are employed by each organization

```
SELECT ?organisation (COUNT(?name) AS ?var)
WHERE{
    ?university rdf:type :University.
    ?organisation rdf:type :Organisation.
    ?person rdf:type :Person;
            :personName ?name;
            :alumnusOf ?university;
            :employeeOf ?organisation.
}
GROUP BY ?organisation
```

Filter query results

Showing results from 1 to 1,000 of 1,608

	organisation	var
1	http://www.wikidata.org/entity/Q167920	"13"^^xsd:integer
2	http://www.wikidata.org/entity/Q371522	"334"^^xsd:integer
3	http://www.wikidata.org/entity/Q151510	"8"^^xsd:integer
4	http://www.wikidata.org/entity/Q579321	"9"^^xsd:integer
5	http://www.wikidata.org/entity/Q333886	"7"^^xsd:integer
6	http://www.wikidata.org/entity/Q152838	"5"^^xsd:integer



We can observe that the number of "employeeOf" relationships indicated is the correct one, as the next relationship shown is different.

Exercise 3 | Submission

Part A

- A4Group34.owl : .OWL file

Part B

- Queries_B : Contains all the queries together in the order that needs to be copied, pasted, and executed to populate GraphDB.
- Queries_B_sep: The folder contains the queries in separated files with the order of execution specified in their respective names, e.g., query_1, query_2…

Part C

- Queries_c: Contains both queries together. The order of execution do not matter.

Assignment4Group34.pdf: The report itself.


CHALMERS
UNIVERSITY OF TECHNOLOGY
UNIVERSITAS GOTHOBURGENSIS
1891