Lab Assignment: Graph DB

Team:
RODAINA MOHAMED
JULIO CHRISTIANS CANDELA CACERES

A. Exploring DBpedia

1. Get the classes defined in the ontology.

```
SELECT ?classes
WHERE {
?classes rdf:type owl:Class
}
```

2. Get the datatype properties defined in the ontology.

```
Select ?property WHERE {
  {?property rdf:type owl:DatatypeProperty}
}
```

3. Get the object properties defined in the ontology. What is the difference between datatype and object properties?

```
SELECT ?properties
WHERE{
?properties rdf:type  owl:ObjectProperty
}
```

An **ObjectProperty** relates an instance to another instance while a **DatatypeProperty** relates an instance to a literal

4. Get the labels of all the properties (both datatype and object) defined in the ontology.

```
Select ?property ?label WHERE {
  {?property a owl:DatatypeProperty;
rdfs:label  ?label. }
UNION {
?property a owl:ObjectProperty;
rdfs:label  ?label. }
}
```

5. Find the class representing an Actor in the dataset (using filters).

```
SELECT DISTINCT ?classes
WHERE {
   ?classes rdf:type owl:Class.
   ?classes rdfs:label ?label.
   FILTER (str(?label) = "actor")
}
```

6. Find the super class for the class Actor.

```
PREFIX : <http://dbpedia.org/ontology/>
SELECT ?superclass
WHERE{
:Actor rdfs:subClassOf ?superclass .
}
```

7. Find all the actors in the dataset.

```
PREFIX : <http://dbpedia.org/ontology/>
SELECT ?actors
WHERE{
?actors rdf:type :Actor
}
```

8. Get different classes that are defined as range of the properties that have the class Actor dened as their domain.

```
PREFIX : <http://dbpedia.org/ontology/>
SELECT ?class
where{
    ?p rdfs:domain :Actor.
      ?p rdfs:range ?class
}
```

9. Find the super property of the goldenRaspberryAward property.

```
PREFIX : <http://dbpedia.org/ontology/>
SELECT ?superProperty
WHERE{
        :goldenRaspberryAward rdfs:subPropertyOf ?superProperty
```

```
}
```

10. Return all the properties that have the class Actor as either their range or domain.

```
PREFIX : <http://dbpedia.org/ontology/>
SELECT ?p AS ?Property
where{
{?p a rdf:Property.
    ?p rdfs:domain :Actor}
UNION
        {?p rdfs:range :Actor}
}
```

11. Return all persons that are not actors.

```
PREFIX : <http://dbpedia.org/ontology/>
SELECT ?person WHERE {
   {?person rdf:type :Person} MINUS
   {?person rdf:type :Actor}
}
```

12. Return the path (in properties and classes) between the Actor and Person classes.

```
PREFIX : <http://dbpedia.org/ontology/>
select ?p1  ?c ?p2
where {
        {:Actor ?p1 ?c .
        ?c ?p2 :Person}
        UNION
        {:Person ?p1 ?c .
        ?c ?p2 :Actor}
        }
```

B. Analytical queries on top of QBAirbase

1. List the country, station type, latitude, and longitude details of each station.
Note: Limit the query to 25 results, and extract only the string values of the required object and not the whole IRIs.

```
PREFIX schema: <http://qweb.cs.aau.dk/airbase/schema/>
PREFIX property: <http://qweb.cs.aau.dk/airbase/property/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?namestation ?countryname ?lat ?long ?type
WHERE {
?station schema:inCountry ?country.
?country property:country ?countryname.
?station property:station ?namestation.
?station property:latitudeDegree ?lat.
?station property:longitudeDegree ?long.
?station property:type ?type
}
LIMIT 25
```

namestation	countryname	lat	long	type
"Lahemaa"^^<http://www.w3.org/2001/XMLSchema#string>	"Estonia"^^<http://www.w3.org/2001/XMLSchema#string>	59.494446	25.930569	"Background"^^<http://www.w3.org
"Vilsandi"^^<http://www.w3.org/2001/XMLSchema#string>	"Estonia"^^<http://www.w3.org/2001/XMLSchema#string>	58.37611	21.845013	"Background"^^<http://www.w3.org
"Viru"^^<http://www.w3.org/2001/XMLSchema#string>	"Estonia"^^<http://www.w3.org/2001/XMLSchema#string>	59.441669	24.772228	"Traffic"^^<http://www.w3.org/20
"Majaka"^^<http://www.w3.org/2001/XMLSchema#string>	"Estonia"^^<http://www.w3.org/2001/XMLSchema#string>	59.458336	24.819448	"Background"^^<http://www.w3.org
"Rahu"^^<http://www.w3.org/2001/XMLSchema#string>	"Estonia"^^<http://www.w3.org/2001/XMLSchema#string>	59.456944	24.698624	"Industrial"^^<http://www.w3.org
"Saarejärve"^^<http://www.w3.org/2001/XMLSchema#string>	"Estonia"^^<http://www.w3.org/2001/XMLSchema#string>	58.702778	26.758898	"Background"^^<http://www.w3.org
"Õismäe"^^<http://www.w3.org/2001/XMLSchema#string>	"Estonia"^^<http://www.w3.org/2001/XMLSchema#string>	59.414169	24.649458	"Background"^^<http://www.w3.org
"Kohtla-Järve"^^<http://www.w3.org/2001/XMLSchema#string>	"Estonia"^^<http://www.w3.org/2001/XMLSchema#string>	59.409725	27.278622	"Industrial"^^<http://www.w3.org
"Liivalaia"^^<http://www.w3.org/2001/XMLSchema#string>	"Estonia"^^<http://www.w3.org/2001/XMLSchema#string>	59.431114	24.760565	"Traffic"^^<http://www.w3.org/20
"Tartu"^^<http://www.w3.org/2001/XMLSchema#string>	"Estonia"^^<http://www.w3.org/2001/XMLSchema#string>	58.370556	26.734722	"Background"^^<http://www.w3.org
"Narva"^^<http://www.w3.org/2001/XMLSchema#string>	"Estonia"^^<http://www.w3.org/2001/XMLSchema#string>	59.37616	28.179167	"Background"^^<http://www.w3.org
"6 Le Bouscat-Rue de la Liberation-Barriere Medoc"^^<http://www.w3.org/2001/XMLSchema#string>	"France"^^<http://www.w3.org/2001/XMLSchema#string>	44.855278	-0.592508	"Traffic"^^<http://www.w3.org/20
"2Station de ref-rue P. Montet"^^<http://www.w3.org/2001/XMLSchema#string>	"France"^^<http://www.w3.org/2001/XMLSchema#string>	48.579166	7.764181	"Traffic"^^<http://www.w3.org/20
"1F94"^^<http://www.w3.org/2001/XMLSchema#string>	"France"^^<http://www.w3.org/2001/XMLSchema#string>	48.795277	2.460842	"Background"^^<http://www.w3.org
"1F77"^^<http://www.w3.org/2001/XMLSchema#string>	"France"^^<http://www.w3.org/2001/XMLSchema#string>	48.855	2.596678	"Background"^^<http://www.w3.org
"4F75"^^<http://www.w3.org/2001/XMLSchema#string>	"France"^^<http://www.w3.org/2001/XMLSchema#string>	48.858055	2.349736	"Background"^^<http://www.w3.org
"2F94"^^<http://www.w3.org/2001/XMLSchema#string>	"France"^^<http://www.w3.org/2001/XMLSchema#string>	48.772221	2.421114	"Background"^^<http://www.w3.org
"7F78"^^<http://www.w3.org/2001/XMLSchema#string>	"France"^^<http://www.w3.org/2001/XMLSchema#string>	48.989166	1.809175	"Industrial"^^<http://www.w3.org
"5F78"^^<http://www.w3.org/2001/XMLSchema#string>	"France"^^<http://www.w3.org/2001/XMLSchema#string>	49.016388	1.566125	"Industrial"^^<http://www.w3.org
"6F78"^^<http://www.w3.org/2001/XMLSchema#string>	"France"^^<http://www.w3.org/2001/XMLSchema#string>	48.994999	1.744733	"Industrial"^^<http://www.w3.org
"2F91"^^<http://www.w3.org/2001/XMLSchema#string>	"France"^^<http://www.w3.org/2001/XMLSchema#string>	48.731388	2.291117	"Background"^^<http://www.w3.org
"8F75"^^<http://www.w3.org/2001/XMLSchema#string>	"France"^^<http://www.w3.org/2001/XMLSchema#string>	48.876942	2.311122	"Background"^^<http://www.w3.org
"6F93"^^<http://www.w3.org/2001/XMLSchema#string>	"France"^^<http://www.w3.org/2001/XMLSchema#string>	48.893059	2.519192	"Background"^^<http://www.w3.org
"15 Hotel de Ville-La Hadeleine"^^<http://www.w3.org/2001/XMLSchema#string>	"France"^^<http://www.w3.org/2001/XMLSchema#string>	50.651669	3.075564	"Unknown"^^<http://www.w3.org/20
"142 BVD STRASBOURG"^^<http://www.w3.org/2001/XMLSchema#string>	"France"^^<http://www.w3.org/2001/XMLSchema#string>	49.493057	0.114736	"Background"^^<http://www.w3.org

2. List the 10 highest averages of C6H6 emission and the country and the year on which they were recorded.

```
PREFIX schema: <http://qweb.cs.aau.dk/airbase/schema/>
PREFIX property: <http://qweb.cs.aau.dk/airbase/property/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT DISTINCT ?countryName ?yearNum (avg(?c6h6) AS
?avgC6H6emission)
where{
?obs schema:C6H6 ?c6h6.
```

```
?obs schema:sensor ?sensor.
?sensor property:statisticShortName "Mean"^^xsd:string .
?obs schema:station ?station.
?obs schema:year ?year.
?year property:yearNum ?yearNum.
?station schema:inCountry ?country.
?country property:country ?countryName.
}
Group by ?countryName ?yearNum
ORDER BY DESC(?avgC6H6emission)
LIMIT 10
```

	countryName	yearNum	avgC6H6emission
1	Germany	"1999"^^xsd:integer	"37.591632653061224489795918"^^xsd:decimal
2	Italy	"2000"^^xsd:integer	"17.138405405405405405405405"^^xsd:decimal
3	Belgium	"1997"^^xsd:integer	"10.483833333333333333333333"^^xsd:decimal
4	Romania	"2003"^^xsd:integer	"8.1995"^^xsd:decimal
5	Greece	"2004"^^xsd:integer	"7.5755"^^xsd:decimal
6	Switzerland	"1997"^^xsd:integer	"7.372"^^xsd:decimal
7	Italy	"2001"^^xsd:integer	"7.02884375"^^xsd:decimal
8	Greece	"2005"^^xsd:integer	"6.7235"^^xsd:decimal
9	Italy	"1999"^^xsd:integer	"6.695975"^^xsd:decimal
10	Italy	"2002"^^xsd:integer	"5.532095238095238095238095"^^xsd:decimal

Note: A sensor has a property (dened through the prex: <http://qweb.cs.aau.dk/airbase/property/>) stastisticShortName, and it can be Mean, Max, etc.

3. For each city and property type, give the yearly average emission for NO2, SO2, PB, and PM10.

Note: the sensors gather different statistical measures (Mean, percentiles, max, etc) being the property:statisticShortName "Mean" the value to take into account

```
PREFIX schema: <http://qweb.cs.aau.dk/airbase/schema/>
PREFIX property: <http://qweb.cs.aau.dk/airbase/property/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT DISTINCT
?propertytype
?cityname
?yearnum
```

```
(avg(?no2) as ?avgno2)
(avg(?so2) as ?avgso2)
(avg(?pb) as ?avgpb)
(avg(?pm10) as ?avgpm10)
WHERE {
?obs schema:station ?station.
?station property:type ?propertytype.
?sensor property:statisticShortName "Mean"^^xsd:string .
?obs schema:year ?year.
?obs schema:sensor ?sensor.
?year property:yearNum ?yearnum.
?station schema:inCity ?city.
?city property:city ?cityname.
   {
      {?obs schema:NO2 ?no2} UNION
      {?obs schema:SO2 ?so2} UNION
      {?obs schema:Pb ?pb} UNION
      {?obs schema:PM10 ?pm10}
   }

}
GROUP BY ?propertytype ?cityname ?yearnum
```

propertytype	cityname	yearnum	avgno2	avgso2	avgpb	avgpm10
"Background"^^<http://www.w3.org/2001/XMLSchema#string>	"BAMBERG"^^<http://www.w3.org/2001/XMLSchema#string>	2002	28.635	4.2345		24.671
"Background"^^<http://www.w3.org/2001/XMLSchema#string>	"LATINA"^^<http://www.w3.org/2001/XMLSchema#string>	2011	37.94425			30.6555
"Background"^^<http://www.w3.org/2001/XMLSchema#string>	"BARRIOS (LOS)"^^<http://www.w3.org/2001/XMLSchema#string>	2012				17.433
"Industrial"^^<http://www.w3.org/2001/XMLSchema#string>	"UTRILLAS"^^<http://www.w3.org/2001/XMLSchema#string>	2005	14.681	10.073		24.3065
"Background"^^<http://www.w3.org/2001/XMLSchema#string>	"JOACHIMSTAL"^^<http://www.w3.org/2001/XMLSchema#string>	2000	7.9695	4.0535		15.922
"Traffic"^^<http://www.w3.org/2001/XMLSchema#string>	"LAPPEENRANTA"^^<http://www.w3.org/2001/XMLSchema#string>	1999	17.535			23.7985
"Industrial"^^<http://www.w3.org/2001/XMLSchema#string>	"ESCH-SUR-ALZETTE"^^<http://www.w3.org/2001/XMLSchema#string>	2008	28.261	1.7695		26.849
"Background"^^<http://www.w3.org/2001/XMLSchema#string>	"KAISERSLAUTERN"^^<http://www.w3.org/2001/XMLSchema#string>	2011	24.1445	1.9665		
"Background"^^<http://www.w3.org/2001/XMLSchema#string>	"LÜNEN"^^<http://www.w3.org/2001/XMLSchema#string>	2008	29.2275			23.36
"Background"^^<http://www.w3.org/2001/XMLSchema#string>	"CORROY LE GRAND"^^<http://www.w3.org/2001/XMLSchema#string>	1996		14.2975		
"Background"^^<http://www.w3.org/2001/XMLSchema#string>	"NEUHAUSEN"^^<http://www.w3.org/2001/XMLSchema#string>	2011	10.9785	9.571		17.837
"Background"^^<http://www.w3.org/2001/XMLSchema#string>	"BOIS-HERPIN"^^<http://www.w3.org/2001/XMLSchema#string>	2011				21.8505
"Background"^^<http://www.w3.org/2001/XMLSchema#string>	"ODENSE"^^<http://www.w3.org/2001/XMLSchema#string>	1996	23.805			
"Traffic"^^<http://www.w3.org/2001/XMLSchema#string>	"SEGOVIA"^^<http://www.w3.org/2001/XMLSchema#string>	2010	16.049	2.8635		14.221
"Traffic"^^<http://www.w3.org/2001/XMLSchema#string>	"LIEPAJA"^^<http://www.w3.org/2001/XMLSchema#string>	2010	27.5735	2.461		30.565
"Traffic"^^<http://www.w3.org/2001/XMLSchema#string>	"LEIDEN"^^<http://www.w3.org/2001/XMLSchema#string>	2010				30.894
"Background"^^<http://www.w3.org/2001/XMLSchema#string>	"BRESCIA"^^<http://www.w3.org/2001/XMLSchema#string>	2011		6.561		42.846
"Traffic"^^<http://www.w3.org/2001/XMLSchema#string>	"YORK"^^<http://www.w3.org/2001/XMLSchema#string>	2009	36.519			18.9025
"Background"^^<http://www.w3.org/2001/XMLSchema#string>	"FRYDEK-MISTEK"^^<http://www.w3.org/2001/XMLSchema#string>	2001	22.4	12.274		45.231
"Background"^^<http://www.w3.org/2001/XMLSchema#string>	"NEUSTADT"^^<http://www.w3.org/2001/XMLSchema#string>	2002	15.373	2.1		14.7725
"Traffic"^^<http://www.w3.org/2001/XMLSchema#string>	"WÜRZBURG"^^<http://www.w3.org/2001/XMLSchema#string>	1994		9.14175		
"Background"^^<http://www.w3.org/2001/XMLSchema#string>	"ILMENAU"^^<http://www.w3.org/2001/XMLSchema#string>	2003	19.4075			24.3495
"Background"^^<http://www.w3.org/2001/XMLSchema#string>	"PRATO"^^<http://www.w3.org/2001/XMLSchema#string>	2006	33.979	2.9095		39.145
"Traffic"^^<http://www.w3.org/2001/XMLSchema#string>	"ANTIBES"^^<http://www.w3.org/2001/XMLSchema#string>	2001	54.6685	1.053		30.281
"Background"^^<http://www.w3.org/2001/XMLSchema#string>	"POLKOWICE"^^<http://www.w3.org/2001/XMLSchema#string>	2012			0.026	29.228

4. Dene 3 additional SPARQL queries (and their corresponding interpretation) that you think could be interesting for the domain of analyzing air quality/pollution.

a) **The worst record of SO2 pollution in Cities in Spain Annually.**

This query returns the worst SO2 pollution in all cities in spain. This can be used to annually examine the effectiveness of the measures taken to fight this pollutant or whether new measures should be imposed to reduce SO2 in the air as it is one of the most dangerous air pollutants.

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX schema: <http://qweb.cs.aau.dk/airbase/schema/>
PREFIX property: <http://qweb.cs.aau.dk/airbase/property/>
PREFIX country: <http://qweb.cs.aau.dk/airbase/data/country/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

 SELECT ?city, ?yearNum as ?Year,(max(?so2) as ?avgso2) WHERE {
    ?obs schema:SO2 ?so2 .
    ?obs schema:station ?station .
    ?obs schema:sensor ?sensor.
    ?sensor property:statisticShortName "Max"^^xsd:string .
    ?station schema:inCity ?city .
    ?obs schema:year ?year .
    ?year property:yearNum ?yearNum.

    ?city schema:locatedIn <http://qweb.cs.aau.dk/airbase/data/country/Spain/> .
    ?city owl:sameAs ?cityName .
   }
GROUP BY ?city ?yearNum
```

b) **The affected population by SO2 in cities in Europein 2010 .**
This query can be used to examine the population of the cities affected by the pollution of by one of the most dangerous pollutants such as SO2.

```
PREFIX yago: <http://yago-knowledge.org/resource/>
PREFIX schema: <http://qweb.cs.aau.dk/airbase/schema/>
PREFIX property: <http://qweb.cs.aau.dk/airbase/property/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

 SELECT ?city,?p as ?Population, (max(?so2) as ?avgso2) WHERE {
    ?obs schema:SO2 ?so2 .
    ?obs schema:station ?station .
    ?obs schema:sensor ?sensor.
    ?sensor property:statisticShortName "Mean"^^xsd:string .
    ?station schema:inCity ?city .
    ?obs schema:year ?year .
    ?year property:yearNum "2010"^^xsd:decimal.

    SERVICE <https://linkeddata1.calcul.u-psud.fr/sparql> {
    ?city yago:hasNumberOfPeople ?p . FILTER (?p > 0)
```

```
    }
}
GROUP BY ?city ?p
LIMIT 20
```

c) **The yearly number of the stations that has sensors for NO2, SO2, PB, and PM10 by country .**
This can be useful to know these numbers are increasing or decreasing each year by country and maybe make decisions about how many stations are needed.

```
PREFIX schema: <http://qweb.cs.aau.dk/airbase/schema/>
PREFIX property: <http://qweb.cs.aau.dk/airbase/property/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT DISTINCT
?countryName
?year
(count(?obs) as ?stations)
WHERE {
?obs schema:station ?station.
?obs schema:year ?year.
?obs schema:sensor ?sensor.
?year property:yearNum ?yearnum.
?station schema:inCountry ?country.
?country property:country ?countryName.

  {
    {?obs schema:NO2 ?no2} UNION
    {?obs schema:SO2 ?so2} UNION
    {?obs schema:Pb ?pb} UNION
    {?obs schema:PM10 ?pm10}
  }

}
GROUP BY  ?countryName ?year
```

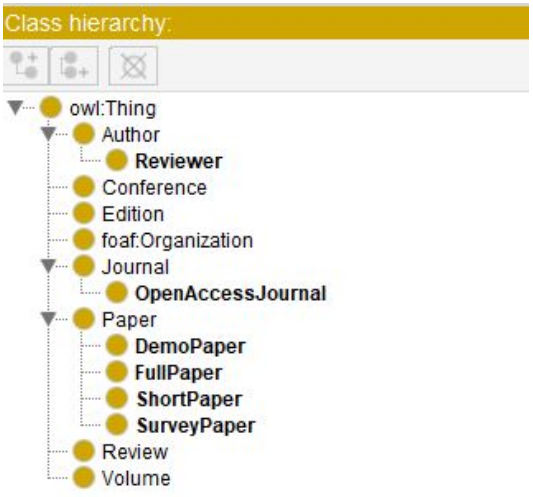
C. Ontology creation

C.1 TBOX denition

1. Depending on how you created the TBOX, you need to provide either the SPARQL queries you used for creating the TBOX

The TBOX was created using “Protege”, a OWL ontology development environment. This tool provides a visual interface to create the rdfs logic of our graph and saves the code in a “.owl” file.

a. **Classes**



- **Edition:** edition of the conference

```
<owl:Class rdf:about="http://www.semanticweb.org/Edition"/>
```

- **Volume:** volume of the journal

```
<owl:Class rdf:about="http://www.semanticweb.org/Volume"/>
```

- **Author:** author of papers

```
<owl:Class rdf:about="http://www.semanticweb.org/Author"/>
```

- **Conference:** renowned/main conference

```
<owl:Class rdf:about="http://www.semanticweb.org/Conference"/>
```

- **DemoPaper:** disjoint subclass of paper

```
<owl:Class rdf:about="http://www.semanticweb.org/DemoPaper">
   <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/Paper"/>
</owl:Class>
```

- **FullPaper:** disjoint subclass of paper

```
<owl:Class rdf:about="http://www.semanticweb.org/FullPaper">
   <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/Paper"/>
</owl:Class>
```

- **Journal:** renowned/main journal

```
<owl:Class rdf:about="http://www.semanticweb.org/Journal"/>
```

- **OpenAccessJournal:** subclass of journal

```
<owl:Class rdf:about="http://www.semanticweb.org/OpenAccessJournal">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/Journal"/>
</owl:Class>
```

- **Paper:** renowned/main journal

```
<owl:Class rdf:about="http://www.semanticweb.org/Paper"/>
```

- **Review:** main review which relates author to paper

```
<owl:Class rdf:about="http://www.semanticweb.org/Review"/>
```

- **Reviewer:** subclass of author

```
<owl:Class rdf:about="http://www.semanticweb.org/Reviewer">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/Author"/>
</owl:Class>
```

- **ShortPaper:** disjoint subclass of paper

```
<owl:Class rdf:about="http://www.semanticweb.org/ShortPaper">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/Paper"/>
</owl:Class>
```

- **SurveyPaper:** disjoint subclass of paper

```
<owl:Class rdf:about="http://www.semanticweb.org/SurveyPaper">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/Paper"/>
</owl:Class>
```

- **Organization:** disjoint subclass of paper

```
<owl:Class rdf:about="http://xmlns.com/foaf/0.1/Organization"/>
```

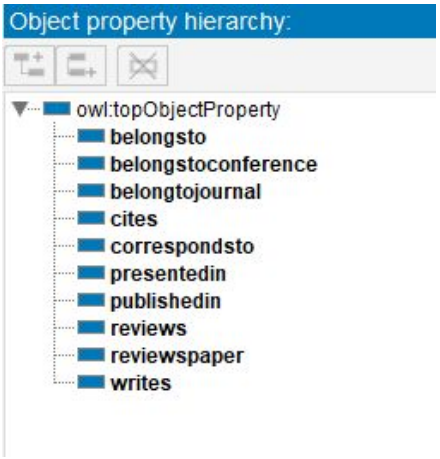
b. General axioms

- This axiom prevents that different subclasses of papers overlap.

```
<rdf:Description>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#AllDisjointClasses"/>
  <owl:members rdf:parseType="Collection">
    <rdf:Description rdf:about="http://www.semanticweb.org/DemoPaper"/>
```

```
      <rdf:Description rdf:about="http://www.semanticweb.org/FullPaper"/>
      <rdf:Description rdf:about="http://www.semanticweb.org/ShortPaper"/>
      <rdf:Description rdf:about="http://www.semanticweb.org/SurveyPaper"/>
   </owl:members>
</rdf:Description>
```

c. Object Properties



- **belongsto:** An author may belong to some organizations

```
<owl:ObjectProperty rdf:about="http://www.semanticweb.org/property/belongsto">
   <rdfs:domain rdf:resource="http://www.semanticweb.org/#Author"/>
   <rdfs:range rdf:resource="http://xmlns.com/foaf/0.1/Organization"/>
</owl:ObjectProperty>
```

- **belongstoconference:** An edition belongs to a renowned/main conference

```
<owl:ObjectProperty rdf:about="http://www.semanticweb.org/property/belongstoconference">
   <rdfs:domain rdf:resource="http://www.semanticweb.org/Edition"/>
   <rdfs:range rdf:resource="http://www.semanticweb.org/#Conference"/>
</owl:ObjectProperty>
```

- **belongstojournal:** A volume belongs to a renowned journal

```
<owl:ObjectProperty rdf:about="http://www.semanticweb.org/property/belongtojournal">
   <rdfs:domain rdf:resource="http://www.semanticweb.org/Volume"/>
   <rdfs:range rdf:resource="http://www.semanticweb.org/#Journal"/>
</owl:ObjectProperty>
```

- **cites:** A paper can cite to other papers

```
<owl:ObjectProperty rdf:about="http://www.semanticweb.org/property/cites">
   <rdfs:domain rdf:resource="http://www.semanticweb.org/Paper"/>
   <rdfs:range rdf:resource="http://www.semanticweb.org/Paper"/>
</owl:ObjectProperty>
```

- **correspondsto:** A paper corresponds to one author

```
<owl:ObjectProperty rdf:about="http://www.semanticweb.org/property/correspondsto">
   <rdfs:domain rdf:resource="http://www.semanticweb.org/Paper"/>
   <rdfs:range rdf:resource="http://www.semanticweb.org/Author"/>
</owl:ObjectProperty>
```

- **presentedin:** A paper was presented in an edition of a conference

```
<owl:ObjectProperty rdf:about="http://www.semanticweb.org/property/presentedin">
   <rdfs:domain rdf:resource="http://www.semanticweb.org/#Paper"/>
   <rdfs:range rdf:resource="http://www.semanticweb.org/Edition"/>
</owl:ObjectProperty>
```

- **publishedin:** A paper was published in a volume of a journal

```
<owl:ObjectProperty rdf:about="http://www.semanticweb.org/property/publishedin">
   <rdfs:domain rdf:resource="http://www.semanticweb.org/#Paper"/>
   <rdfs:range rdf:resource="http://www.semanticweb.org/Volume"/>
</owl:ObjectProperty>
```

- **reviews:** A reviewer reviews a paper

```
<owl:ObjectProperty rdf:about="http://www.semanticweb.org/property/reviews">
   <rdfs:domain rdf:resource="http://www.semanticweb.org/Author"/>
   <rdfs:range rdf:resource="http://www.semanticweb.org/Review"/>
</owl:ObjectProperty>
```

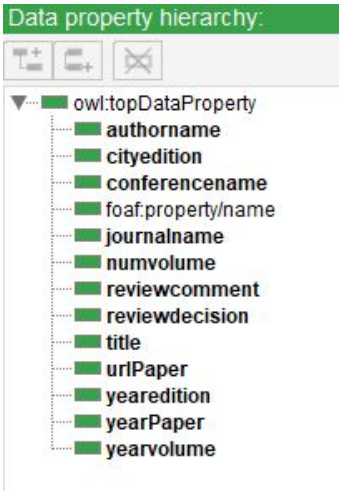
- **reviewspaper:** A review is related to a paper

```
<owl:ObjectProperty rdf:about="http://www.semanticweb.org/property/reviewspaper">
   <rdfs:domain rdf:resource="http://www.semanticweb.org/#Review"/>
   <rdfs:range rdf:resource="http://www.semanticweb.org/#Paper"/>
</owl:ObjectProperty>
```

- **writes:** An author writes a paper

```
<owl:ObjectProperty rdf:about="http://www.semanticweb.org/property/writes">
   <rdfs:domain rdf:resource="http://www.semanticweb.org/#Author"/>
   <rdfs:range rdf:resource="http://www.semanticweb.org/#Paper"/>
</owl:ObjectProperty>
```

d. Data Properties



- **authorname:** name of the author

```
<owl:DatatypeProperty rdf:about="http://www.semanticweb.org/property/authorname">
   <rdfs:domain rdf:resource="http://www.semanticweb.org/#Author"/>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
```

- **cityedition:** city in which the conference edition was celebrated

```
<owl:DatatypeProperty rdf:about="http://www.semanticweb.org/property/cityedition">
   <rdfs:domain rdf:resource="http://www.semanticweb.org/Edition"/>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
```

- **conferencename:** name of the renowned/main conference

```
<owl:DatatypeProperty rdf:about="http://www.semanticweb.org/property/conferencename">
   <rdfs:domain rdf:resource="http://www.semanticweb.org/#Conference"/>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
```

- **journalname:** name of the renowned/main journal

```
<owl:DatatypeProperty rdf:about="http://www.semanticweb.org/property/journalname">
   <rdfs:domain rdf:resource="http://www.semanticweb.org/#Journal"/>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
```

- **numvolume:** number of volume of the journal publication

```
<owl:DatatypeProperty rdf:about="http://www.semanticweb.org/property/numvolume">
   <rdfs:domain rdf:resource="http://www.semanticweb.org/Volume"/>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
</owl:DatatypeProperty>
```

- **reviewcomment:** comments from the author about a specific book.

```
<owl:DatatypeProperty rdf:about="http://www.semanticweb.org/property/reviewcomment">
   <rdfs:domain rdf:resource="http://www.semanticweb.org/#Review"/>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
```

- **reviewdecision:** final decision about the review

```
<owl:DatatypeProperty rdf:about="http://www.semanticweb.org/property/reviewdecision">
   <rdfs:domain rdf:resource="http://www.semanticweb.org/#Review"/>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
```

- **title:** title of the paper

```
<owl:DatatypeProperty rdf:about="http://www.semanticweb.org/property/title">
   <rdfs:domain rdf:resource="http://www.semanticweb.org/#Paper"/>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
```

- **urlPaper:** urls where you can find the paper.

```
<owl:DatatypeProperty rdf:about="http://www.semanticweb.org/property/urlPaper">
   <rdfs:domain rdf:resource="http://www.semanticweb.org/#Paper"/>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
```

- **yearPaper:** year of publication of the paper

```
<owl:DatatypeProperty rdf:about="http://www.semanticweb.org/property/yearPaper">
   <rdfs:domain rdf:resource="http://www.semanticweb.org/#Paper"/>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
</owl:DatatypeProperty>
```

- **yearedition:** year of the edition of the conference

```
<owl:DatatypeProperty rdf:about="http://www.semanticweb.org/property/yearedition">
   <rdfs:domain rdf:resource="http://www.semanticweb.org/Edition"/>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
</owl:DatatypeProperty>
```

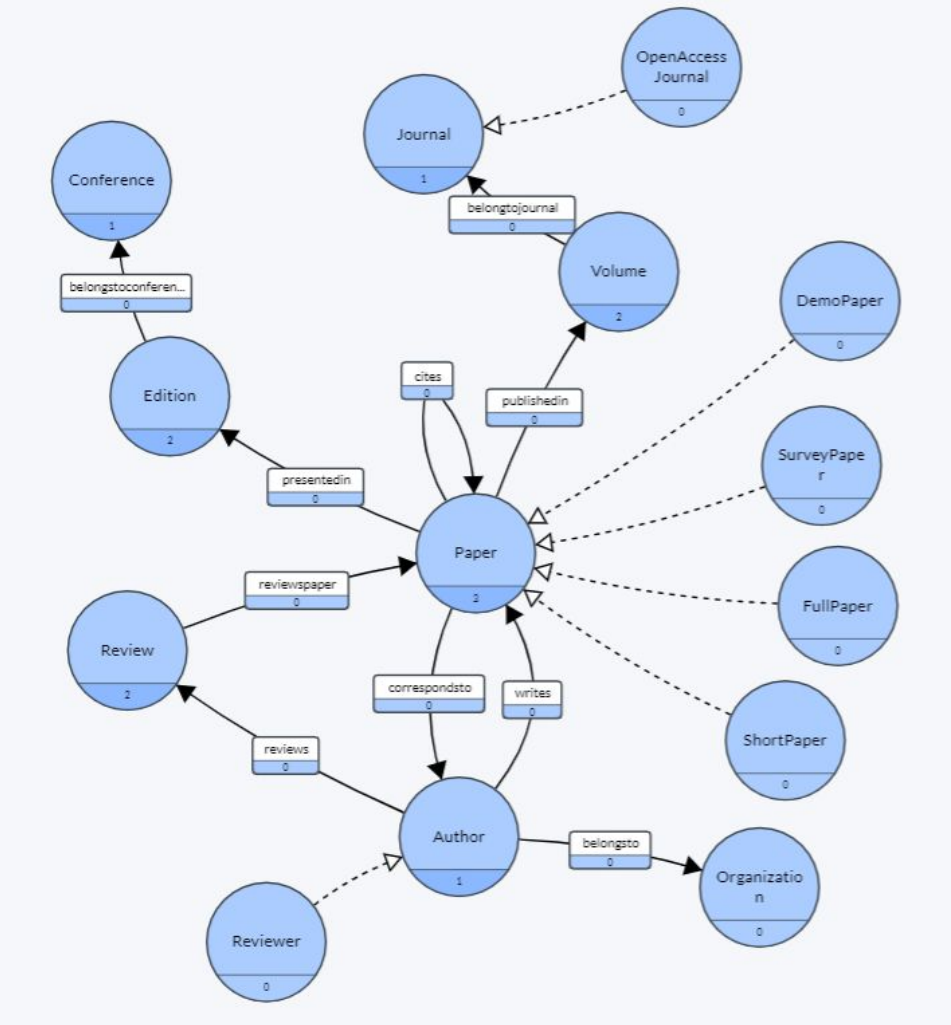
- **yearvolume:** year of publication of the journal's volume

```
<owl:DatatypeProperty rdf:about="http://www.semanticweb.org/property/yearvolume">
   <rdfs:domain rdf:resource="http://www.semanticweb.org/Volume"/>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
</owl:DatatypeProperty>
```

- **name:** name of the organization

```
<owl:DatatypeProperty rdf:about="http://xmlns.com/foaf/0.1/property/name"/>
```

2. Provide a visual representation of the TBOX



C.2. ABOX definition

1. Explain the method used to dene the ABOX.

In the first lab (Property Graphs), we created several input files in csv format regarding a research article dataset. For the purpose of this lab, we are going to reuse them and transform them into an ABOX. The Apache Jena API lets us convert our input files into RDF files suitable for GraphDB importing. As an example we show the function to create the RDF file of Authors and we will later explain step by step the method used:

```
public static void createAuthor() throws IOException {
  Model model = ModelFactory.createDefaultModel();
  BufferedReader csvReader = new BufferedReader(new FileReader(CSV_AUTHOR));
  String line;
  String[] organizations = {"1","2","3"};
  line = csvReader.readLine();//First Line doesn't matter (columns names)
  while ((line = csvReader.readLine()) != null) {
     try {
        String[] row_data = line.split(";");
        String id = row_data[0];
        String name = row_data[1];
        String writes = row_data[2];
        String reviews = row_data[3];
        Random random = new Random();
        String org = organizations[random.nextInt(3)];
        String organizationUri = RESOURCE_URL+"org_" + org.replace(" ","_");
        String authorUri = RESOURCE_URL + "author_" + id;
        Resource currentAuthor = model.createResource(authorUri)
                    .addProperty(model.createProperty(PROPERTY_URL + "authorname"), name)
.addProperty(model.createProperty(PROPERTY_URL+"belongsto"),model.createResource(organizati
onUri));
        if(!writes.equals("")){
           String[] paperlist = writes.split("[|]");
           for(String paper:paperlist){
              currentAuthor.addProperty(model.createProperty(PROPERTY_URL+"writes"),
                    model.createResource(RESOURCE_URL+ "paper_" + paper
                          ));
           }
        }
        if(!reviews.equals("")){
           String[] reviewlist = reviews.split("[|]");
           for(String review:reviewlist){
              currentAuthor.addProperty(model.createProperty(PROPERTY_URL+"reviews"),
                    model.createResource(RESOURCE_URL+ "review_" + review
                       ));
           }
        }
     }
     catch(Exception e){
     }
  }
  csvReader.close();
  model.write(new PrintStream(
        new BufferedOutputStream(
              new FileOutputStream(OUTPUT+"author.nt")), true), "NT");
```

}

Input files and output files

Input Files	Output Files
author_node.csv	author.nt
conference_node.csv	conference.nt
edition_node.csv	edition.nt
journal_node.csv	journal.nt
org_node.csv	organization.nt
paper_node.csv	paper.nt
review_node.csv	review.nt
volume_node.csv	volume.nt

Steps

1. Set up the environment (Input/Output folders, main URLs)
2. Create and Initialize the Jena Model

```
Model model = ModelFactory.createDefaultModel();
```

3. Read the csv file (line by line)
4. Split into columns by the separator in order to get the columns to add.
5. Create the URI of the instance based on the ID (id of the author in the example code)

```
authorUri = RESOURCE_URL + "author_" + id;
```

6. Create the resource for the respective instance in the Jena model.

```
 Resource currentAuthor = model.createResource(authorUri)
```

7. Add properties to the resource created:
    a. DataProperties: relates to literals

```
.addProperty(model.createProperty(PROPERTY_URL + "authorname"), name)
```

    b. ObjectProperties: relates to other objects (we have to create a new resource with the URI of the object)

```
.addProperty(model.createProperty(PROPERTY_URL+"belongsto"),model.createResource(or
ganizationUri));
```

c. Many ObjectProperties: The graph also allows you to relates a property from an object to many objects. For example, an author can write many books.

```
String[] paperlist = writes.split("[|]");
            for(String paper:paperlist){
                currentAuthor.addProperty(model.createProperty(PROPERTY_URL+"writes"),
                        model.createResource(RESOURCE_URL+ "paper_" + paper
                                ));
            }
```

8. Save the model related to the class. This .nt file will be imported in our GraphDB database

```
model.write(new PrintStream(
        new BufferedOutputStream(
                new FileOutputStream(OUTPUT+"author.nt")), true), "NT");
```

At the end, we create 8 rdf files related to our 8 nodes of the class which will be finally imported:

```
public static void main(String[] args) throws Exception {

  createPaper();
  createAuthor();
  createConference();
  createEdition();
  createJournal();
  createVolume();
  createReview();
  createOrganization();
}
```

C.2. Linking ABOX to TBOX

1. Provide the SPARQL queries required to create the link between the ABOX and TBOX.

In order to link the ABOX to the TBOX we need to import the rdf files to our GraphDB database:

a) Create a new repository according to the steps followed in the lab instructions
   i) Set RuleSet RDF-Plus (Optimized)
b) Upload and Import ontologies.owl which contains the TBOX
c) Upload and Import the 8 files which contains the ABOX.

d) Optional: Import from URL: http://xmlns.com/foaf/0.1/ (external information regarding to the Organization)

The next step is to each object with their respective class in the TBOX since our instance “author_id” isn’t type of class Author yet. For this purpose, we execute the following script:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX base: <http://www.semanticweb.org/>
PREFIX proper: <http://www.semanticweb.org/property/>
PREFIX res: <http://www.semanticweb.org/resource/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

- Author

```
INSERT {
?a rdf:type base:Author
}
WHERE {
?a proper:authorname ?b
}
```

- Reviewer

```
INSERT {
?a rdf:type base:Reviewer
}
WHERE {
?a proper:reviews ?b
}
```

- Paper

```
INSERT {
?a rdf:type base:Paper
}
WHERE {
?a proper:title ?b
}
```

- Edition

```
INSERT {
?a rdf:type base:Edition
}
WHERE {
```

```
?a proper:cityedition ?b
}
```

- Conference

```
INSERT {
?a rdf:type base:Conference
}
WHERE {
?a proper:conferencename ?b
}
```

- Volume

```
INSERT {
?a rdf:type base:Volume
}
WHERE {
?a proper:numvolume ?b
}
```

- Journal

```
INSERT {
?a rdf:type base:Journal
}
WHERE {
?a proper:journalname ?b
}
```

- Review

```
INSERT {
?a rdf:type base:Review
}
WHERE {
?a proper:reviewdecision ?b
}
```

- Organization

```
INSERT {
?a rdf:type foaf:Organization
}
WHERE {
```

```
?a proper:name ?b
}
```

- SurveyPaper

```
INSERT {
?a rdf:type base:SurveyPaper
}
WHERE {
   ?a rdf:type base:Paper.
   ?a proper:title ?b.
   FILTER REGEX (?b , "survey","i")
}
```

- DemoPaper

```
INSERT {
?a rdf:type base:DemoPaper
}
WHERE {
   ?a rdf:type base:Paper.
   ?a proper:title ?b.
   FILTER REGEX (?b , "demo ","i")
}
```

- ShortPaper

```
INSERT {
?a rdf:type base:ShortPaper
}
WHERE {
   {?a rdf:type base:Paper.
   ?a proper:title ?b.
      FILTER REGEX (?b , "short ","i").}
   MINUS
   {{?a rdf:type base:SurveyPaper} UNION
   {?a rdf:type base:DemoPaper}
   }
}
```

- FullPaper

```
INSERT {
?a rdf:type base:FullPaper
}
WHERE {
   {?a rdf:type base:Paper.}
```

```
    MINUS
    {{?a rdf:type base:SurveyPaper} UNION
    {?a rdf:type base:DemoPaper} UNION
    {?a rdf:type base:ShortPaper}
    }
}
```

- OpenAccessJournal

```
INSERT {
?a rdf:type base:OpenAccessJournal
}
WHERE {
   {?a rdf:type base:Journal.
   ?a proper:journalname ?b
      FILTER REGEX (?b , "open" ,"i")}
}
```

2. Provide a summary table with simple statistics about the RDF graph obtained, e.g., the number of classes, the number of properties, the number of instances, etc.

The following table shows the general statistics:

Number of Classes	14
Number of ObjectProperties	10
Number of DatatypeProperties	13
Number of Instances	622,582
Number of Triples	4,187,942

The following table show the number of instances linked to each class (linking ABOX to TBOX):

Classes	SubClasses	# objects
:Author	Total	105,224
	:Reviewer	105,224
:Review	Total	270,089
foaf:Organization	Total	3
:Edition	Total	42,423
:Conference	Total	11,410
:Volume	Total	142,139
:Journal	Total	1,790
	:OpenAccessJournal	3
:Paper	Total	49,504
	:DemoPaper	2
	:FullPaper	49,100
	:ShortPaper	76
	:SurveyPaper	326
Total		622,582

C.4. Queries on top of the Ontology

1. Find all the Authors.

(with TBOX)

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX base: <http://www.semanticweb.org/>
PREFIX pro: <http://www.semanticweb.org/property/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?Author ?Name
WHERE{
?Author rdf:type base:Author.
?Author pro:authorname ?Name.
}
```

(without TBOX)

```
PREFIX pro: <http://www.semanticweb.org/property/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT DISTINCT ?Author ?Name
WHERE{
?Author pro:writes ?x.
?Author pro:authorname ?Name.
}
```

2. Find all the properties whose domain is Author.

(with TBOX)

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX base: <http://www.semanticweb.org/>
SELECT *
WHERE{
?Author rdfs:domain base:Author
}
```

(without TBOX)

```
PREFIX pro: <http://www.semanticweb.org/property/>
SELECT DISTINCT ?property
WHERE{
?Author pro:writes ?x.
?Author ?property ?y .
filter ( ?property not in ( rdf:type ) )
}
```

3. Find all the properties whose domain is either Conference or Journal.

(with TBOX)

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX base: <http://www.semanticweb.org/>
SELECT DISTINCT ?property
WHERE{
   {?property rdfs:domain base:Journal}
   Union
   {?property rdfs:domain base:Conference}
}
```

(without TBOX)

```
PREFIX pro: <http://www.semanticweb.org/property/>
SELECT DISTINCT ?property
WHERE{
   {
   ?journal pro:journalname ?j.
```

```
    ?journal ?property ?x
    }
       Union
    {
    ?conference pro:conferencename ?c .
    ?conference ?property ?x
    }
filter ( ?property not in ( rdf:type ) )
}
```

4. Find all the things that Authors have created (either Reviews or Papers).(with TBOX)

(with TBOX)

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX base: <http://www.semanticweb.org/>
SELECT DISTINCT?thingCreated
WHERE{
        ?a rdfs:domain base:Author.
   ?author ?a ?thingCreated.
      {{?thingCreated rdf:type base:Paper.} UNION
        {?thingCreated rdf:type base:Review.}}
   }
```

(without TBOX)

```
PREFIX pro: <http://www.semanticweb.org/property/>
SELECT DISTINCT ?thingCreated
WHERE{
   {
      ?Author pro:writes ?thingCreated.
   }
      Union
   {
   ?Author pro:reviews ?thingCreated
   }
}
```