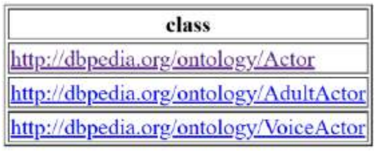
Part A. Exploring DBpedia

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

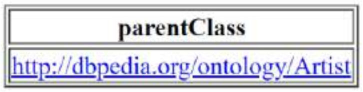
```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX dbpedia3: <http://dbpedia.org/ontology/>

SELECT ?class
WHERE
{
 ?class a owl:Class
 FILTER (contains(xsd:string(?class), "Actor"))
}
```



class
http://dbpedia.org/ontology/Actor
http://dbpedia.org/ontology/AdultActor
http://dbpedia.org/ontology/VoiceActor

2. Find the super class for the class Actor.

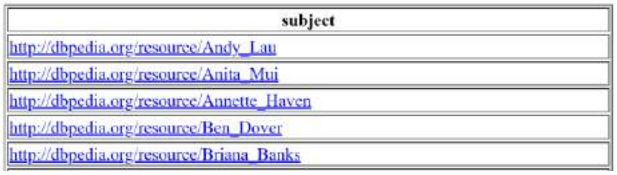
```
PREFIX dbpedia3: <http://dbpedia.org/ontology/>

SELECT * WHERE { dbpedia3:Actor rdfs:subClassOf ?parentClass }
```



parentClass
http://dbpedia.org/ontology/Artist

3. Find all the actors in the dataset.

```
PREFIX dbpedia3: <http://dbpedia.org/ontology/>

SELECT * WHERE { ?subject rdf:type dbpedia3:Actor}
```

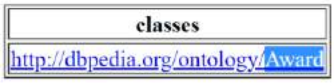


subject
http://dbpedia.org/resource/Andy_Lau
http://dbpedia.org/resource/Anita_Mui
http://dbpedia.org/resource/Annette_Haven
http://dbpedia.org/resource/Ben_Dover
http://dbpedia.org/resource/Briana_Banks

… more

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

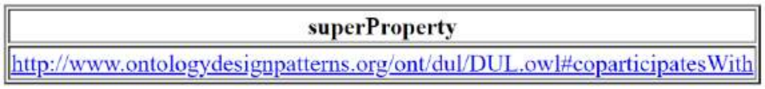
```
PREFIX dbpedia3: <http://dbpedia.org/ontology/>

SELECT distinct ?classes
WHERE
{
 ?s rdfs:domain dbpedia3:Actor.
 ?s rdfs:range ?classes
}
```



classes
http://dbpedia.org/ontology/Award

5. Find the super property of the goldenRaspberryAward property.

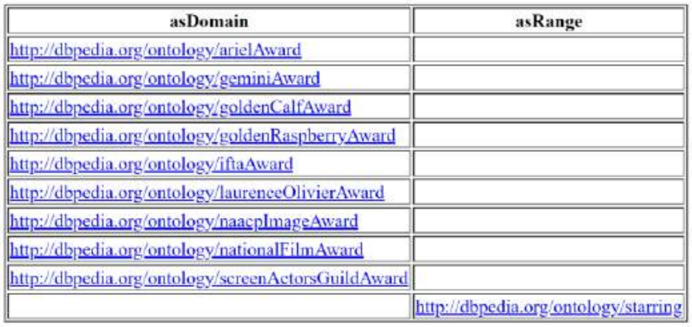
```
PREFIX dbpedia3: <http://dbpedia.org/ontology/>

SELECT *
WHERE { dbpedia3:goldenRaspberryAward rdfs:subPropertyOf ?superProperty }
```



superProperty
http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#coparticipatesWith

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

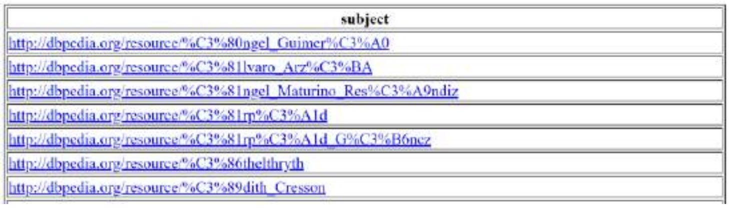
```
PREFIX dbpedia3: <http://dbpedia.org/ontology/>

SELECT * WHERE {
    {?asDomain rdfs:domain dbpedia3:Actor}
         UNION
    {?asRange rdfs:range dbpedia3:Actor}
}
```



asDomain	asRange
http://dbpedia.org/ontology/arielAward	
http://dbpedia.org/ontology/geminiAward	
http://dbpedia.org/ontology/goldenCalfAward	
http://dbpedia.org/ontology/goldenRaspberryAward	
http://dbpedia.org/ontology/iftaAward	
http://dbpedia.org/ontology/laurenceOlivierAward	
http://dbpedia.org/ontology/naacpImageAward	
http://dbpedia.org/ontology/nationalFilmAward	
http://dbpedia.org/ontology/screenActorsGuildAward	
	http://dbpedia.org/ontology/starring

7. Return all persons that are not actors.

```
PREFIX dbpedia3: <http://dbpedia.org/ontology/>

SELECT * WHERE {
   ?subject rdf:type dbpedia3:Person .
   MINUS {?subject rdf:type dbpedia3:Actor }
}
```



subject
http://dbpedia.org/resource/%C3%80ngel_Guimer%C3%A0
http://dbpedia.org/resource/%C3%81lvaro_Arz%C3%BA
http://dbpedia.org/resource/%C3%81ngel_Maturino_Res%C3%A9ndiz
http://dbpedia.org/resource/%C3%81rp%C3%A1d
http://dbpedia.org/resource/%C3%81rp%C3%A1d_G%C3%B6nez
http://dbpedia.org/resource/%C3%86thelthryth
http://dbpedia.org/resource/%C3%89dith_Cresson

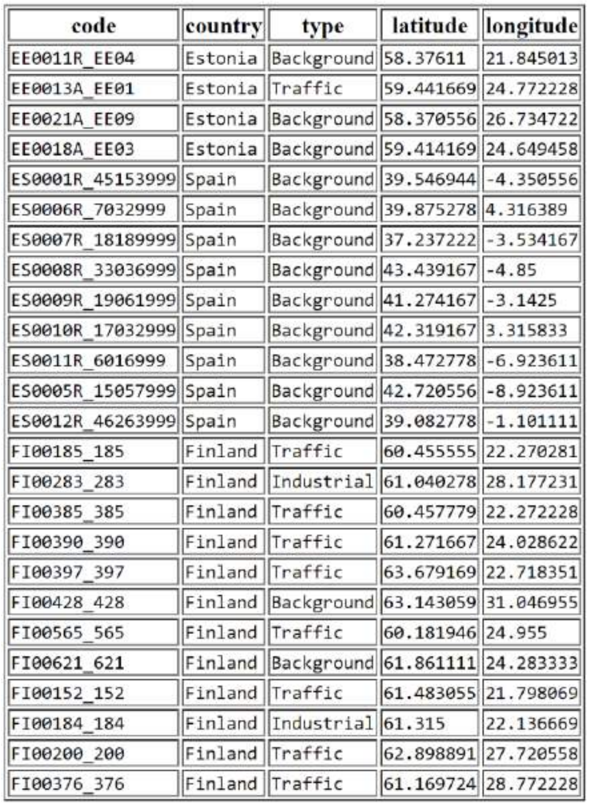
… more

Part 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.

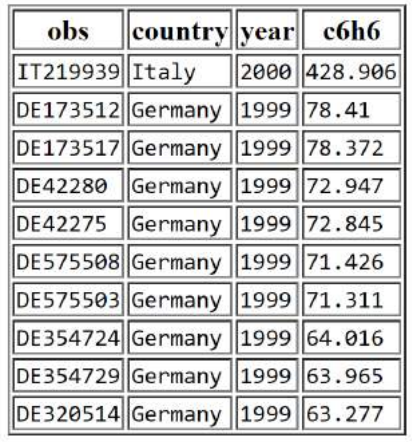
```
PREFIX airschema: <http://qweb.cs.aau.dk/airbase/schema/>
PREFIX airprop: <http://qweb.cs.aau.dk/airbase/property/>

SELECT DISTINCT concat(str(?stationEUname), '_', str(?stationLOCname)) as ?code
str(?country) as ?country str(?type) as ?type ?latitude ?longitude
WHERE {
[a <http://purl.oclc.org/NET/ssnx/ssn#Observation>] airschema:station ?station .
   ?station airprop:europeanCode ?stationEUname ;
            airprop:localCode ?stationLOCname ;
            airschema:inCountry ?countryIRI ;
            airprop:type ?type ;
            airprop:latitudeDegree ?latitude ;
            airprop:longitudeDegree ?longitude .
   ?countryIRI airprop:country ?country
}
LIMIT 25
```



code	country	type	latitude	longitude
EE0011R_EE04	Estonia	Background	58.37611	21.845013
EE0013A_EE01	Estonia	Traffic	59.441669	24.772228
EE0021A_EE09	Estonia	Background	58.370556	26.734722
EE0018A_EE03	Estonia	Background	59.414169	24.649458
ES0001R_45153999	Spain	Background	39.546944	-4.350556
ES0006R_7032999	Spain	Background	39.875278	4.316389
ES0007R_18189999	Spain	Background	37.237222	-3.534167
ES0008R_33036999	Spain	Background	43.439167	-4.85
ES0009R_19061999	Spain	Background	41.274167	-3.1425
ES0010R_17032999	Spain	Background	42.319167	3.315833
ES0011R_6016999	Spain	Background	38.472778	-6.923611
ES0005R_15057999	Spain	Background	42.720556	-8.923611
ES0012R_46263999	Spain	Background	39.082778	-1.101111
FI00185_185	Finland	Traffic	60.455555	22.270281
FI00283_283	Finland	Industrial	61.040278	28.177231
FI00385_385	Finland	Traffic	60.457779	22.272228
FI00390_390	Finland	Traffic	61.271667	24.028622
FI00397_397	Finland	Traffic	63.679169	22.718351
FI00428_428	Finland	Background	63.143059	31.046955
FI00565_565	Finland	Traffic	60.181946	24.955
FI00621_621	Finland	Background	61.861111	24.283333
FI00152_152	Finland	Traffic	61.483055	21.798069
FI00184_184	Finland	Industrial	61.315	22.136669
FI00200_200	Finland	Traffic	62.898891	27.720558
FI00376_376	Finland	Traffic	61.169724	28.772228

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

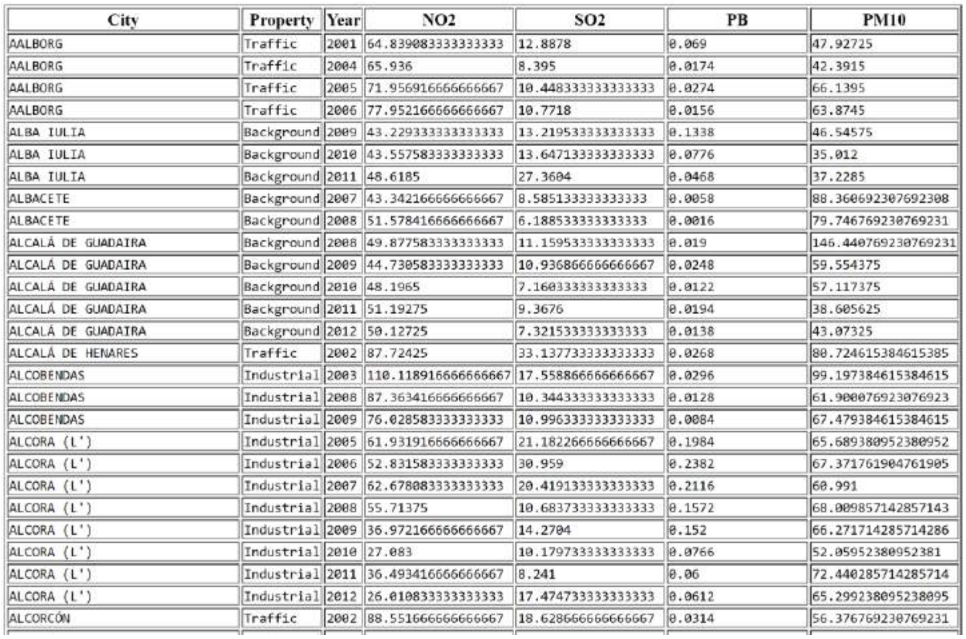
```
PREFIX schema: <http://qweb.cs.aau.dk/airbase/schema/>
PREFIX ob: <http://qweb.cs.aau.dk/airbase/data/observation/>
PREFIX property: <http://qweb.cs.aau.dk/airbase/property/>

SELECT distinct str(?obs) as ?obs str(?country) as ?country ?year ?c6h6
WHERE {
   ?obsIRI schema:C6H6 ?c6h6 .
   ?obsIRI schema:station ?station .
   ?station schema:inCountry ?countryIRI .
   ?countryIRI property:country ?country .
   ?obsIRI schema:year ?yearIRI .
   ?yearIRI property:yearNum ?year .
   ?obsIRI schema:sensor ?sensor .
   ?sensor property:statisticShortName "Mean"^^xsd:string
   bind(replace(str(?obsIRI),"http://qweb.cs.aau.dk/airbase/data/observation/","") as ?obs)
}
order by desc(?c6h6)
LIMIT 10
```



obs	country	year	c6h6
IT219939	Italy	2000	428.906
DE173512	Germany	1999	78.41
DE173517	Germany	1999	78.372
DE42280	Germany	1999	72.947
DE42275	Germany	1999	72.845
DE575508	Germany	1999	71.426
DE575503	Germany	1999	71.311
DE354724	Germany	1999	64.016
DE354729	Germany	1999	63.965
DE320514	Germany	1999	63.277

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

```
PREFIX airprop: <http://qweb.cs.aau.dk/airbase/property/>
PREFIX airschema: <http://qweb.cs.aau.dk/airbase/schema/>

SELECT str(?cityName) AS ?City str(?property) AS ?Property ?yearN AS ?Year ?NO2
?SO2 ?PB ?PM10
WHERE {
      {SELECT ?cityName ?property ?yearN avg(?no2) AS ?NO2
      WHERE {
             ?station airschema:inCity ?city .
             ?station airprop:type ?property .
             ?city airprop:city ?cityName .
             ?obs1 airschema:station ?station .
             ?obs1 airschema:NO2 ?no2 .
             ?obs1 airschema:year ?year .
             ?year airprop:yearNum ?yearN
      }
      GROUP BY ?cityName ?property ?yearN}

      {SELECT ?cityName ?property ?yearN avg(?so2) AS ?SO2
      WHERE {
             ?station airschema:inCity ?city .
             ?station airprop:type ?property .
             ?city airprop:city ?cityName .
             ?obs1 airschema:station ?station .
             ?obs1 airschema:SO2 ?so2 .
             ?obs1 airschema:year ?year .

             ?year airprop:yearNum ?yearN
      }
      GROUP BY ?cityName ?property ?yearN}
      {SELECT ?cityName ?property ?yearN avg(?pb) AS ?PB
      WHERE {
            ?station airschema:inCity ?city .
            ?station airprop:type ?property .
            ?city airprop:city ?cityName .
            ?obs1 airschema:station ?station .
            ?obs1 airschema:Pb ?pb .
            ?obs1 airschema:year ?year .
            ?year airprop:yearNum ?yearN
      }
      GROUP BY ?cityName ?property ?yearN}

      {SELECT ?cityName ?property ?yearN avg(?pm10) AS ?PM10
      WHERE {
            ?station airschema:inCity ?city .
            ?station airprop:type ?property .
            ?city airprop:city ?cityName .
            ?obs1 airschema:station ?station .
            ?obs1 airschema:PM10 ?pm10 .
            ?obs1 airschema:year ?year .

            ?year airprop:yearNum ?yearN
      }
      GROUP BY ?cityName ?property ?yearN}
}
ORDER BY ?cityName ?property ?yearN
```



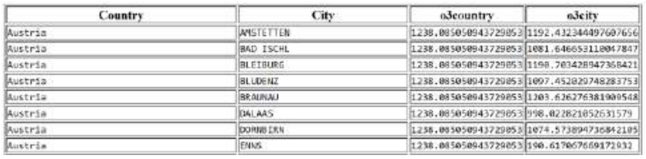
City	Property	Year	NO2	SO2	PB	PM10
AALBORG	Traffic	2001	64.839083333333333	12.8878	0.069	47.92725
AALBORG	Traffic	2004	65.936	8.395	0.0174	42.3915
AALBORG	Traffic	2005	71.956916666666667	10.448333333333333	0.0274	66.1395
AALBORG	Traffic	2006	77.952166666666667	10.7718	0.0156	63.8745
ALBA IULIA	Background	2009	43.229333333333333	13.219533333333333	0.1338	46.54575
ALBA IULIA	Background	2010	43.557583333333333	13.647133333333333	0.0776	35.012
ALBA IULIA	Background	2011	48.6185	27.3604	0.0468	37.2285
ALBACETE	Background	2007	43.342166666666667	8.585133333333333	0.0058	88.360692307692308
ALBACETE	Background	2008	51.578416666666667	6.188533333333333	0.0016	79.746769230769231
ALCALÁ DE GUADAIRA	Background	2008	49.877583333333333	11.159533333333333	0.019	146.440769230769231
ALCALÁ DE GUADAIRA	Background	2009	44.730583333333333	10.936866666666667	0.0248	59.554375
ALCALÁ DE GUADAIRA	Background	2010	48.1965	7.160333333333333	0.0122	57.117375
ALCALÁ DE GUADAIRA	Background	2011	51.19275	9.3676	0.0194	38.605625
ALCALÁ DE GUADAIRA	Background	2012	50.12725	7.321533333333333	0.0138	43.07325
ALCALÁ DE HENARES	Traffic	2002	87.72425	33.137733333333333	0.0268	80.724615384615385
ALCOBENDAS	Industrial	2003	110.118916666666667	17.558866666666667	0.0296	99.197384615384615
ALCOBENDAS	Industrial	2008	87.363416666666667	10.344333333333333	0.0128	61.900076923076923
ALCOBENDAS	Industrial	2009	76.028583333333333	10.996333333333333	0.0084	67.479384615384615
ALCORA (L')	Industrial	2005	61.931916666666667	21.182266666666667	0.1984	65.689380952380952
ALCORA (L')	Industrial	2006	52.831583333333333	30.959	0.2382	67.371761904761905
ALCORA (L')	Industrial	2007	62.678083333333333	20.419133333333333	0.2116	60.991
ALCORA (L')	Industrial	2008	55.71375	10.683733333333333	0.1572	68.009857142857143
ALCORA (L')	Industrial	2009	36.972166666666667	14.2704	0.152	66.271714285714286
ALCORA (L')	Industrial	2010	27.083	10.179733333333333	0.0766	52.05952380952381
ALCORA (L')	Industrial	2011	36.493416666666667	8.241	0.06	72.440285714285714
ALCORA (L')	Industrial	2012	26.010833333333333	17.474733333333333	0.0612	65.299238095238095
ALCORCÓN	Traffic	2002	88.551666666666667	18.628666666666667	0.0314	56.376769230769231

… more

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

4.1 Return the cities with lower ozone level than the average of the country.

```
PREFIX airprop: <http://qweb.cs.aau.dk/airbase/property/>
PREFIX airschema: <http://qweb.cs.aau.dk/airbase/schema/>

SELECT str(?countryName) AS ?Country str(?cityName) AS ?City ?o3country  ?o3city

WHERE{
             ?country airprop:country ?countryName .
             ?city airprop:city ?cityName .
             ?city airschema:locatedIn ?country

             {SELECT ?city avg(?o3) AS ?o3city
             WHERE {
                    ?station airschema:inCity ?city .

                    ?obs1 airschema:station ?station .
                    ?obs1 airschema:O3 ?o3
             }
             GROUP BY ?city}

             {SELECT ?country avg(?o3) AS ?o3country
             WHERE {
                    ?city airschema:locatedIn ?country .
                    ?station airschema:inCity ?city .
                    ?obs1 airschema:station ?station .
                    ?obs1 airschema:O3 ?o3
             }
             GROUP BY ?country}

             FILTER(?o3country > ?o3city)
}
ORDER BY ?countryName ?cityName ?o3country ?o3city
```



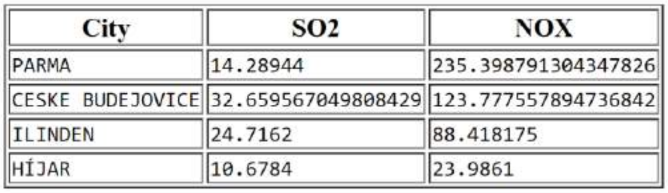
Country	City	o3country	o3city
Austria	AMSTETTEN	1238.085050943729053	1192.432344497607656
Austria	BAD ISCHL	1238.085050943729053	1081.646653110047847
Austria	BLEIBURG	1238.085050943729053	1198.703428947368421
Austria	BLUDENZ	1238.085050943729053	1097.452029748283753
Austria	BRAUNAU	1238.085050943729053	1203.626276381909548
Austria	DALAAS	1238.085050943729053	998.022821052631579
Austria	DORNBIRN	1238.085050943729053	1074.573894736842105
Austria	ENNS	1238.085050943729053	190.617067669172932

… more

Interpretation: This query returns all the cities with a lower level of O3 than the average of the country. For example, all the previous cities are a sample of the cities with a lower level of O3 than the average of Austria (1238).

4.2 Find cities with high acidification (excess of both SO2 and NOx) in average over the years, considering top 100 countries with more excess in one of each element.

```
PREFIX airprop: <http://qweb.cs.aau.dk/airbase/property/>
PREFIX airschema: <http://qweb.cs.aau.dk/airbase/schema/>

SELECT str(?cityName) AS ?City ?SO2 ?NOX
WHERE {
        {SELECT ?cityName avg(?so2) AS ?SO2
        WHERE {
                ?station airschema:inCity ?city .
                ?city airprop:city ?cityName .

                ?obs1 airschema:station ?station .
                ?obs1 airschema:SO2 ?so2
        }
        GROUP BY ?cityName
        LIMIT 100}

        {SELECT ?cityName avg(?nox) AS ?NOX
        WHERE {
                ?station airschema:inCity ?city .
                ?city airprop:city ?cityName .

                ?obs1 airschema:station ?station .
                ?obs1 airschema:NOX ?nox
        }
        GROUP BY ?cityName
        LIMIT 100}
}
ORDER BY DESC(?NOX) (?SO2)
```

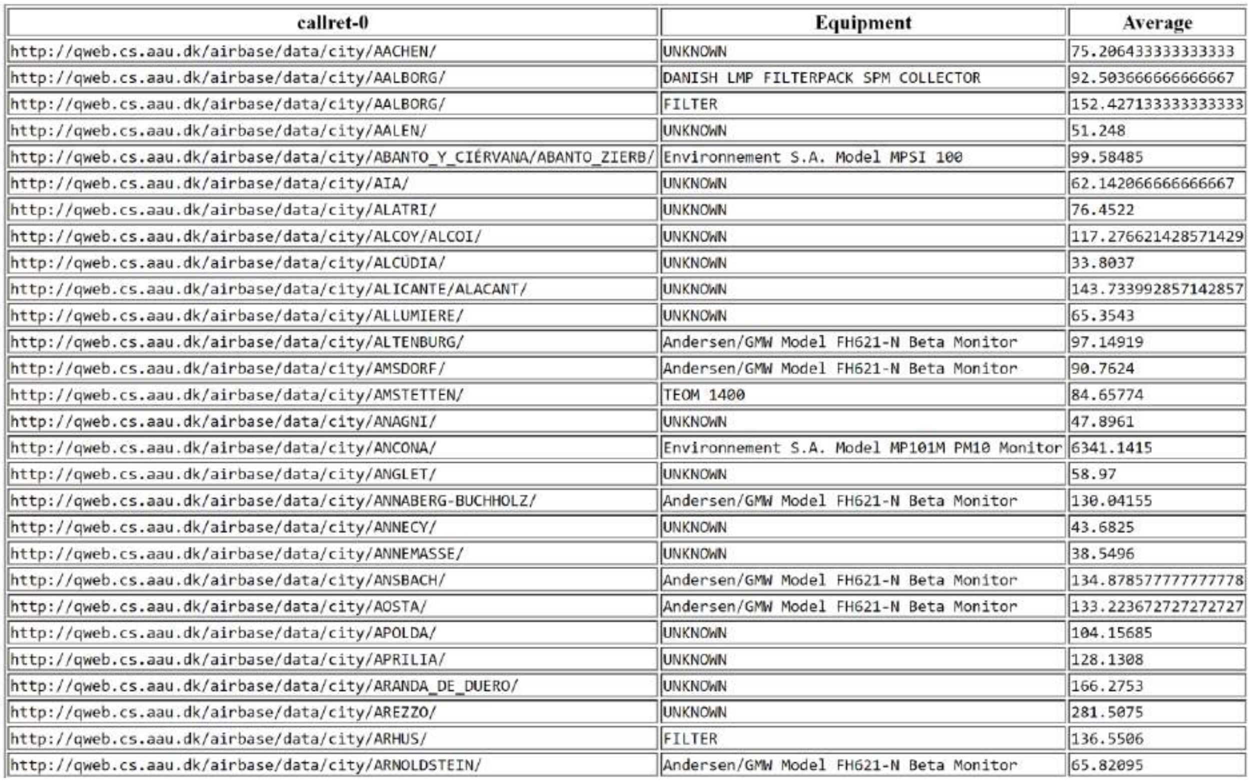


City	SO2	NOX
PARMA	14.28944	235.398791304347826
CESKE BUDEJOVICE	32.659567049808429	123.777557894736842
ILINDEN	24.7162	88.418175
HÍJAR	10.6784	23.9861

Interpretation: This query returns all the cities that shares both top 100 level in SO2 and NOX, what can lead to acidification of water and ground. From the output, only 4 cities share high levels in both elements, in comparison to the rest of the data.

4.3 Find, for each city, the sensors they use to measure SPM and the average measure for each sensor

```
PREFIX airprop: <http://qweb.cs.aau.dk/airbase/property/>
PREFIX airschema: <http://qweb.cs.aau.dk/airbase/schema/>

SELECT DISTINCT str(?city) str(?eq) AS ?Equipment avg(?spm) AS ?Average
WHERE{
	?obs airschema:sensor ?sensor .
	?sensor airprop:equipment ?eq .
	?sensor airschema:measures ?component .
	?component airprop:caption ?unit .
	?obs airschema:SPM ?spm .
	?obs airschema:station ?station .
	?station airschema:inCity ?city
} GROUP BY ?city ?unit ?eq
ORDER BY ?city ?Average ?eq
```



callret-0	Equipment	Average
http://qweb.cs.aau.dk/airbase/data/city/AACHEN/	UNKNOWN	75.206433333333333
http://qweb.cs.aau.dk/airbase/data/city/AALBORG/	DANISH LMP FILTERPACK SPM COLLECTOR	92.503666666666667
http://qweb.cs.aau.dk/airbase/data/city/AALBORG/	FILTER	152.427133333333333
http://qweb.cs.aau.dk/airbase/data/city/AALEN/	UNKNOWN	51.248
http://qweb.cs.aau.dk/airbase/data/city/ABANTO_Y_CIÉRVANA/ABANTO_ZIERB/	Environnement S.A. Model MPSI 100	99.58485
http://qweb.cs.aau.dk/airbase/data/city/AIA/	UNKNOWN	62.142066666666667
http://qweb.cs.aau.dk/airbase/data/city/ALATRI/	UNKNOWN	76.4522
http://qweb.cs.aau.dk/airbase/data/city/ALCOY/ALCOI/	UNKNOWN	117.276621428571429
http://qweb.cs.aau.dk/airbase/data/city/ALCÚDIA/	UNKNOWN	33.8037
http://qweb.cs.aau.dk/airbase/data/city/ALICANTE/ALACANT/	UNKNOWN	143.733992857142857
http://qweb.cs.aau.dk/airbase/data/city/ALLUMIERE/	UNKNOWN	65.3543
http://qweb.cs.aau.dk/airbase/data/city/ALTENBURG/	Andersen/GMW Model FH621-N Beta Monitor	97.14919
http://qweb.cs.aau.dk/airbase/data/city/AMSDORF/	Andersen/GMW Model FH621-N Beta Monitor	90.7624
http://qweb.cs.aau.dk/airbase/data/city/AMSTETTEN/	TEOM 1400	84.65774
http://qweb.cs.aau.dk/airbase/data/city/ANAGNI/	UNKNOWN	47.8961
http://qweb.cs.aau.dk/airbase/data/city/ANCONA/	Environnement S.A. Model MP101M PM10 Monitor	6341.1415
http://qweb.cs.aau.dk/airbase/data/city/ANGLET/	UNKNOWN	58.97
http://qweb.cs.aau.dk/airbase/data/city/ANNABERG-BUCHHOLZ/	Andersen/GMW Model FH621-N Beta Monitor	130.04155
http://qweb.cs.aau.dk/airbase/data/city/ANNECY/	UNKNOWN	43.6825
http://qweb.cs.aau.dk/airbase/data/city/ANNEMASSE/	UNKNOWN	38.5496
http://qweb.cs.aau.dk/airbase/data/city/ANSBACH/	Andersen/GMW Model FH621-N Beta Monitor	134.878577777777778
http://qweb.cs.aau.dk/airbase/data/city/AOSTA/	Andersen/GMW Model FH621-N Beta Monitor	133.223672727272727
http://qweb.cs.aau.dk/airbase/data/city/APOLDA/	UNKNOWN	104.15685
http://qweb.cs.aau.dk/airbase/data/city/APRILIA/	UNKNOWN	128.1308
http://qweb.cs.aau.dk/airbase/data/city/ARANDA_DE_DUERO/	UNKNOWN	166.2753
http://qweb.cs.aau.dk/airbase/data/city/AREZZO/	UNKNOWN	281.5075
http://qweb.cs.aau.dk/airbase/data/city/ARHUS/	FILTER	136.5506
http://qweb.cs.aau.dk/airbase/data/city/ARNOLDSTEIN/	Andersen/GMW Model FH621-N Beta Monitor	65.82095

… more

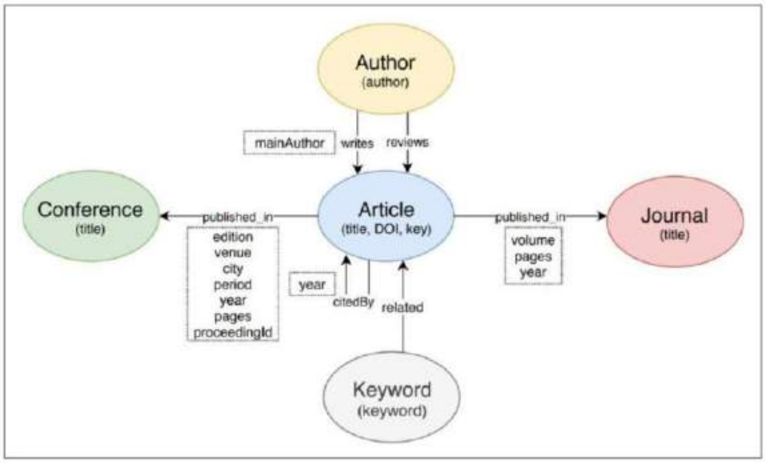
Interpretation: This query gets the different equipments used to measure SO2, along with the average output, which can lead to a comparison of results and with more details, a comparison of precision.

Part C. Ontology creation

C1. TBOX Definition

1. Describe the methodology/method you used and the output generated in a graphical form.

First, we based ourselves in the following schema submitted for the first assignment of the course:



Trying to constraint our schema as much as possible, we evolved it and defined the following logical description rules:

Related to class **Paper**:

- Union(Short paper, Demo paper, Survey paper, Full paper) is equivalent to Paper
- Short paper, Demo paper, Survey paper, Full paper are disjoint
- Paper has only one(1) main author
- Paper has a title
- Paper written by / reviewed Author
- Paper can be cited by other Papers
- Paper can't be cited by his/her self
- Reviewer can't review his/her own paper(s)
- Paper has 1 to many Keyword(s)

Related to class **Author**:

- Author ⊆ Writer
- Reviewer ⊆ Author
- Reviewers only review Short or Full papers

Related to class **Journal / Volume**:

- ➔ Journal ⊆ Media
- ➔ Journal has a title
- ➔ Journal has 1 to many Volumes
- ➔ Union(Non-Open-Access_Journal, Open-Access_Journal) is equivalent to Journal
- ➔ Non-Open-Access_Journal, Open-Access_Journal are disjoint

Related to class **Conference / Edition**:

- ➔ Conference has a title
- ➔ Conference has 1 to many Editions
- ➔ Union(AI_Conference, BigData_Conference, BPM_Conference, Database_Conference) is equivalent to Conference
- ➔ AI_Conference, BigData_Conference, BPM_Conference, Database_Conference are disjoint
- ➔ Edition has a unique venue, city, year

Related to **relationships**:

- ➔ The relationship published_in (Paper, Edition or Volume) is unique for a given domain
- ➔ The relationship belongs_to (Edition, Conference) is unique for a given domain
- ➔ Full_Paper or Short_Paper are the only classes of Paper that can cite and be cited
- ➔ The relationship part_of (Volume, Journal) is unique for a given domain
- ➔ The relationship main_author (Paper, Author) is unique for a given domain
- ➔ Paper has to be published either to a Journal's Volume or a Conference's Edition, but not both
- ➔ A reviewer can only review a Full_Paper or a Short_Paper
- ➔ The relationship published_in has a unique set of pages
- ➔ The set of pages between two papers can not overlap
- ➔ A paper can have 1..many keywords
- ➔ A keyword has a keywordName

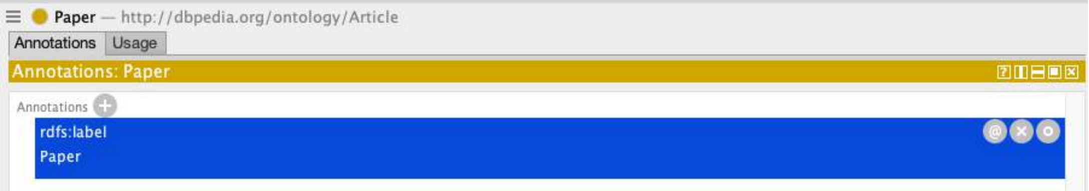
Given these assumptions, we proceeded to create the TBOX. To do so, we used Protégé as we found it to be more friendly for its interface. We downloaded the environment and created our classes, object & data properties, along with dependencies and restrictions, and in connection with dbpedia whenever possible.

More specifically, let us take the class **Paper** as an example, as the rest of classes are defined the same way:

- ➔ IRI: http://dbpedia.org/ontology/Article
  For the IRIs that we could, we reused the dbpedia existing vocabulary. In the cases we didn't find a related concept, we created them based on our semanticweb URI (e.g. for

the subclass Demo_Paper
http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#Demo_Paper)

- Label: rdfs:label “Paper”



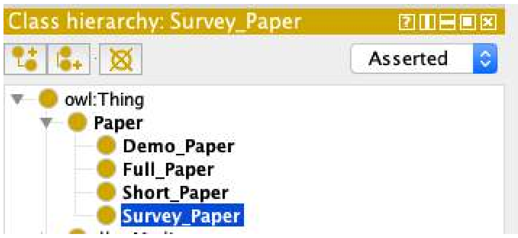
Paper — http://dbpedia.org/ontology/Article

Annotations: Paper

Annotations

rdfs:label
Paper

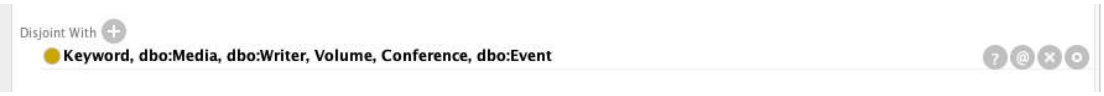
- Subclasses: in this case we have 4 subclasses added



Class hierarchy: Survey_Paper

- owl:Thing
  - Paper
    - Demo_Paper
    - Full_Paper
    - Short_Paper
    - Survey_Paper

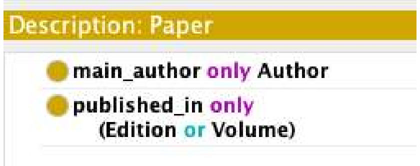
- Disjoint: We added the restriction that Paper was disjoint with every other class, to be more restrictive



Disjoint With

Keyword, dbo:Media, dbo:Writer, Volume, Conference, dbo:Event

- Outgoing relationships:

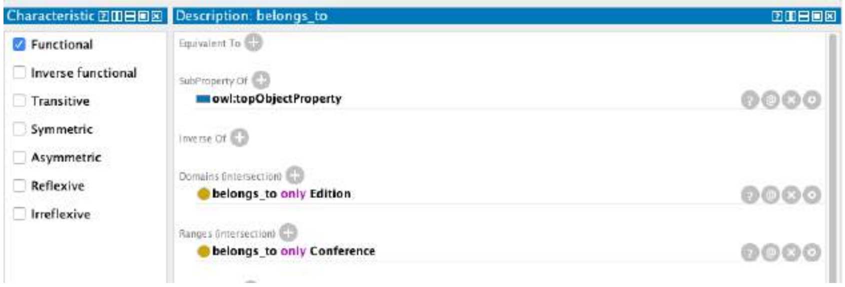


Description: Paper

main_author only Author
published_in only (Edition or Volume)

  - Main_author: Is restricted so that a Paper has only a unique instance of this relationship
  - Published_in: Is restricted so that a Paper has only a unique instance of this relationship that can go either to an Edition or a Volume

When creating relationships (named ObjectProperties), we take as an example **belongs_to:**



Characteristic

☑ Functional
☐ Inverse functional
☐ Transitive
☐ Symmetric
☐ Asymmetric
☐ Reflexive
☐ Irreflexive

Description: belongs_to

Equivalent To

SubProperty Of
owl:topObjectProperty

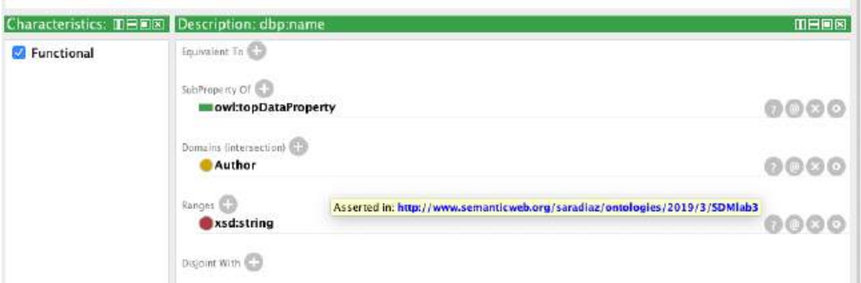
Inverse Of

Domains (intersection)
belongs_to only Edition

Ranges (intersection)
belongs_to only Conference

- ➔ **IRI:** here the IRI can be reused from dbpedia (like we used it on the relationship cites) or it can be created on the semanticweb
- ➔ **Domain:** only class Edition
- ➔ **Range:** only class Conference
- ➔ **Functional:** guarantee that the an Edition belongs_to only one Conference

Finally, we added attributes (DataProperty) for the given classes. In this case we take as an example **name** an attribute of the class **Author:**


Characteristics:
Description: dbp:name

☑ Functional

Equivalent To

SubProperty Of
owl:topDataProperty

Domains (intersection)
Author

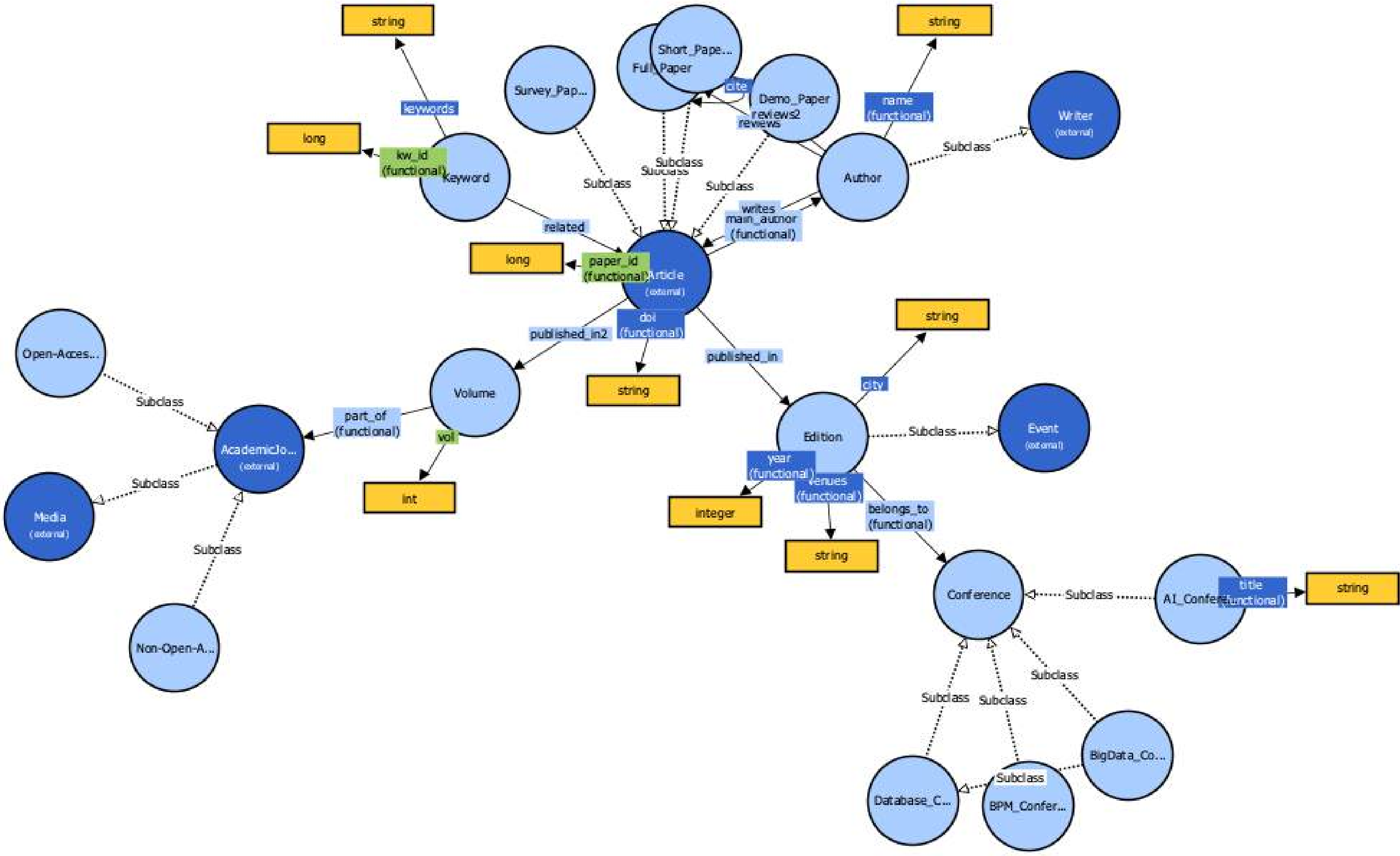
Ranges
xsd:string
Asserted in: http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3

Disjoint With

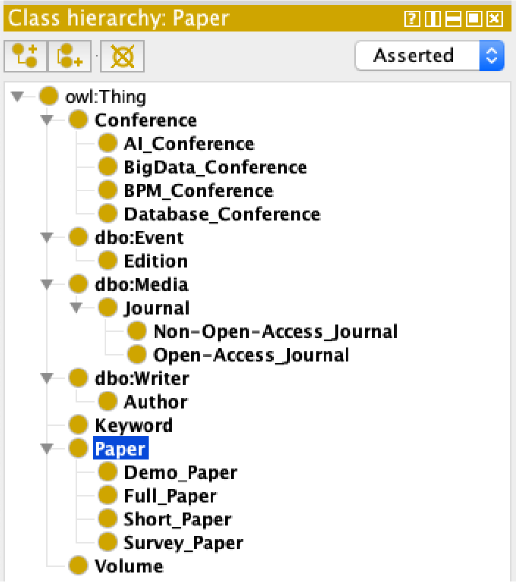
- ➔ **IRI:** we reused the dbpedia URI: http://dbpedia.org/property/name
- ➔ **Domain:** Author
- ➔ **Range:** since its of text value, xsd:string

2. Provide a visual representation of the TBOX

Note that external URIs are marked with dark blue color.



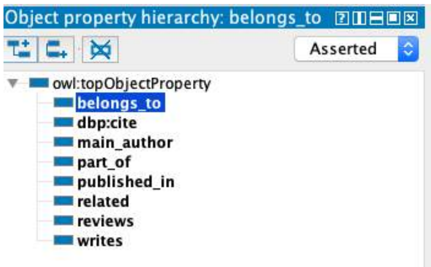
The classes mapped:



Class hierarchy: Paper

- owl:Thing
  - Conference
    - AI_Conference
    - BigData_Conference
    - BPM_Conference
    - Database_Conference
  - dbo:Event
    - Edition
  - dbo:Media
    - Journal
      - Non-Open-Access_Journal
      - Open-Access_Journal
  - dbo:Writer
    - Author
  - Keyword
  - Paper
    - Demo_Paper
    - Full_Paper
    - Short_Paper
    - Survey_Paper
  - Volume

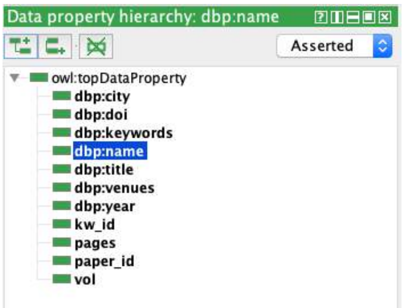
The relationships mapped:



Object property hierarchy: belongs_to

- owl:topObjectProperty
  - belongs_to
  - dbp:cite
  - main_author
  - part_of
  - published_in
  - related
  - reviews
  - writes

The attributes mapped are:



Data property hierarchy: dbp:name

- owl:topDataProperty
  - dbp:city
  - dbp:doi
  - dbp:keywords
  - dbp:name
  - dbp:title
  - dbp:venues
  - dbp:year
  - kw_id
  - pages
  - paper_id
  - vol

We re-used the following dbpedia vocabulary:

- http://dbpedia.org/ontology/Event
- http://dbpedia.org/ontology/Media
- http://dbpedia.org/ontology/Writer
- http://dbpedia.org/ontology/Article
- http://dbpedia.org/property/cite
- http://dbpedia.org/property/city
- http://dbpedia.org/property/doi
- http://dbpedia.org/property/keywords
- http://dbpedia.org/property/name
- http://dbpedia.org/property/title
- http://dbpedia.org/property/venues
- http://dbpedia.org/property/year

C2. ABOX Definition

1. Explain the method used to define the ABOX.

We have used Jena in order to define the ABOX and the Jena-Virtuoso driver in order to import the ABOX (along with TBOX) into Virtuoso. More specifically, we created a model within Jena, imported the TBOX we created on section C.1 into that model and converted all JSON files we had into RDF graph.

In the below code, we perform the connection with Virtuoso and create an empty model which we will later fill in with our CSV files.

```
InputStream in = FileManager.get().open(inputFileName); //locate input OWL file
base = ModelFactory.createOntologyModel(OntModelSpec.OWL_MEM_MICRO_RULE_INF); //create the model
base.read(in, "RDF/XML"); //read owl file of RDF/XML type

virtGraph = new
VirtGraph("http://localhost:8890/research","jdbc:virtuoso://jynx.fib.upc.es:1111", "dba",
"dba");
virtGraph.clear();
virtModel = new VirtModel(virtGraph);
```

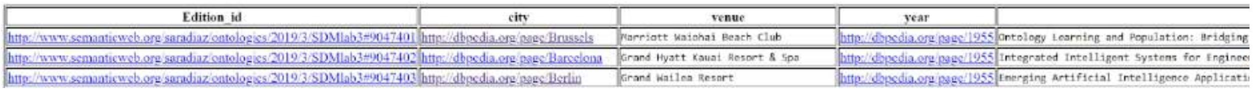
Example of class

Let us take the example of Edition as a class. We read the relevant CSV file and import its details into relevant triplets. Note that for “city” and “year” properties, we search for them in dbpedia and insert the relevant resource URI from there.

```
private static void processEditions() throws IOException {
       BufferedReader br = new BufferedReader(new FileReader("input/edition.csv"));

       // [...] code omitted: remove header, read line by line and split on “;”
      // DBpediaPage = ‘http://dbpedia.org/page/’

       // create Edition properties
       Individual edition = base.getOntClass(NS + "Edition").createIndividual(NS + edition_id);

       DatatypeProperty has_title = base.getDatatypeProperty(DBpediaProp + "title");
       Literal title_value = base.createTypedLiteral(title, XSDDatatype.XSDstring);
       base.add(edition, has_title, title_value);

       DatatypeProperty has_venue = base.getDatatypeProperty(DBpediaProp + "venues");
       Literal venue_value = base.createTypedLiteral(venue, XSDDatatype.XSDstring);
       base.add(edition, has_venue, venue_value);

       DatatypeProperty has_city = base.getDatatypeProperty(DBpediaProp + "city");
       Resource city_value;
```

```
        try {
            city_value = base.createResource(DBpediaPage + city); //trying to take it from dbpedia
        }catch (Exception e){
            city_value = base.createTypedLiteral(city, XSDDatatype.XSDstring).asResource();
        }
        base.add(edition, has_city, city_value);

        DatatypeProperty has_year = base.getDatatypeProperty(DBpediaProp + "year");
        Resource year_value;
        try {
            year_value = base.createResource(DBpediaPage + year); //trying to take it from dbpedia
        }catch (Exception e){
            year_value = base.createTypedLiteral(year, XSDDatatype.XSDint).asResource();
        }
        base.add(edition, has_year, year_value);
    }
}
```

Here’s the corresponding details loaded and queried into Virtuoso.



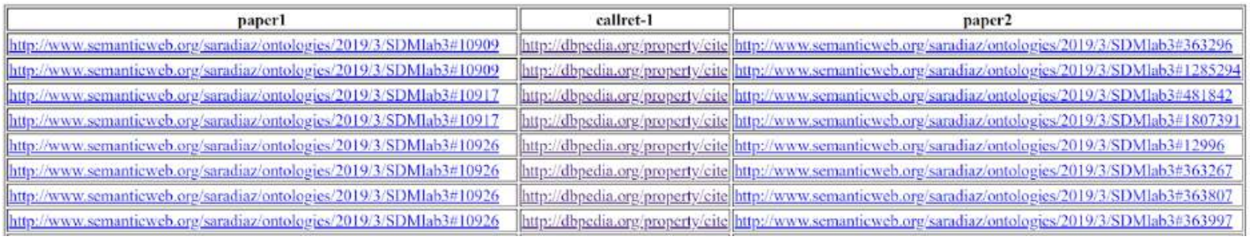
Edition_id	city	venue	year	
http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#9047401	http://dbpedia.org/page/Brussels	Marriott Waiohai Beach Club	http://dbpedia.org/page/1955	Ontology Learning and Population: Bridging
http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#9047402	http://dbpedia.org/page/Barcelona	Grand Hyatt Kauai Resort & Spa	http://dbpedia.org/page/1955	Integrated Intelligent Systems for Enginee
http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#9047403	http://dbpedia.org/page/Berlin	Grand Wailea Resort	http://dbpedia.org/page/1955	Emerging Artificial Intelligence Applicati

Example of object property

Let us take the example of *Cites* as object property. We get the object property from dbpedia URI, read and load the relevant CSV file, populate relevant resources and insert the triplets. Note that for the “year” data property, it is fetched from the relevant dbpedia resource URI. Finally, as we allow only Full_Paper and Short_Paper to cite and be cited, we filter these cases to make sure that correct data will end up on our graph, to also agree with the restriction we had setup during TBOX creation.

```
private static void processCites() throws IOException {
   BufferedReader br = new BufferedReader(new FileReader("input/cited_by_year.csv"));
   ObjectProperty cite = base.getObjectProperty(DBpediaProp + "cite");

   String line = br.readLine(); //remove header: Author_ID - Paper_ID
   while ((line = br.readLine()) != null) {
       String[] tokens = line.split(";");
       long paper1_id = Long.parseLong(tokens[0]);
       long paper2_id = Long.parseLong(tokens[1]);
       Integer year = Integer.parseInt(tokens[2]);

       DatatypeProperty has_year = base.getDatatypeProperty(DBpediaProp + "year");
       Resource year_value;
       try {
           year_value = base.createResource(DBpediaPage + year); //trying to take it from dbpedia
       }catch (Exception e){
           year_value = base.createTypedLiteral(year, XSDDatatype.XSDint).asResource();
       }
       cite.addProperty(has_year,year_value);

       OntResource paper1 = base.getOntResource(NS + paper1_id);
       OntResource paper2 = base.getOntResource(NS + paper2_id);
       String paper_type1 = paper1.getRDFType().getLocalName();
       String paper_type2 = paper2.getRDFType().getLocalName();

       if (("Full_Paper".equals(paper_type1) || "Short_Paper".equals(paper_type1))
               && ("Full_Paper".equals(paper_type2) || "Short_Paper".equals(paper_type2))) {
           base.add(paper2, cite, paper1);
       }
   }
}
```

Here's the corresponding details loaded and queried into Virtuoso



paper1	callret-1	paper2
http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#10909	http://dbpedia.org/property/cite	http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#363296
http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#10909	http://dbpedia.org/property/cite	http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#1285294
http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#10917	http://dbpedia.org/property/cite	http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#481842
http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#10917	http://dbpedia.org/property/cite	http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#1807391
http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#10926	http://dbpedia.org/property/cite	http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#12996
http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#10926	http://dbpedia.org/property/cite	http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#363267
http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#10926	http://dbpedia.org/property/cite	http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#363807
http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#10926	http://dbpedia.org/property/cite	http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#363997

Note: All code can be found under the following link: https://github.com/spapadop/Jena

C3. Linking ABOX to TBOX

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

As mentioned, we have used Jena to load TBOX + ABOX together into Virtuoso and during ABOX creation we used TBOX model, to make sure that the links happen. For completeness, we wrote some Construct & Insert into graph statements.

Insert triples: <paper instance> rdf:type <Article>. As all paper instances are of rdf:type Full_Paper, Short_Paper, Demo_Paper and Survey_Paper, and all of these are subclasses of Article, we perform the below insert.

```
PREFIX schema: <http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#>
prefix dbpediaOnt: <http://dbpedia.org/ontology/>
```

```
insert
{ graph <http://localhost:8890/research> {?s rdf:type dbpediaOnt:Article}}
where
{
   ?s rdf:type ?type .
   ?type rdfs:subClassOf dbpediaOnt:Article .
}
```

Insert triples: <journal instance> rdf:type <AcademicJournal>. As all journal instances are of rdf:type Open-access-Journal and Closed-Access-Journal, and these are subclasses of Journal, we perform the below insert.

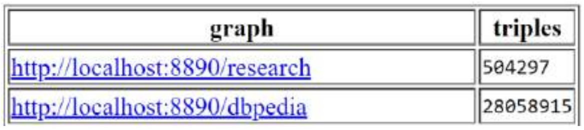
```
PREFIX schema: <http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#>
prefix dbpediaOnt: <http://dbpedia.org/ontology/>

insert
{ graph <http://localhost:8890/research> {?s rdf:type dbpediaOnt:AcademicJournal}}
where
{
   ?s rdf:type ?type .
   ?type rdfs:subClassOf dbpediaOnt:AcademicJournal .
}
```

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.

Summary table

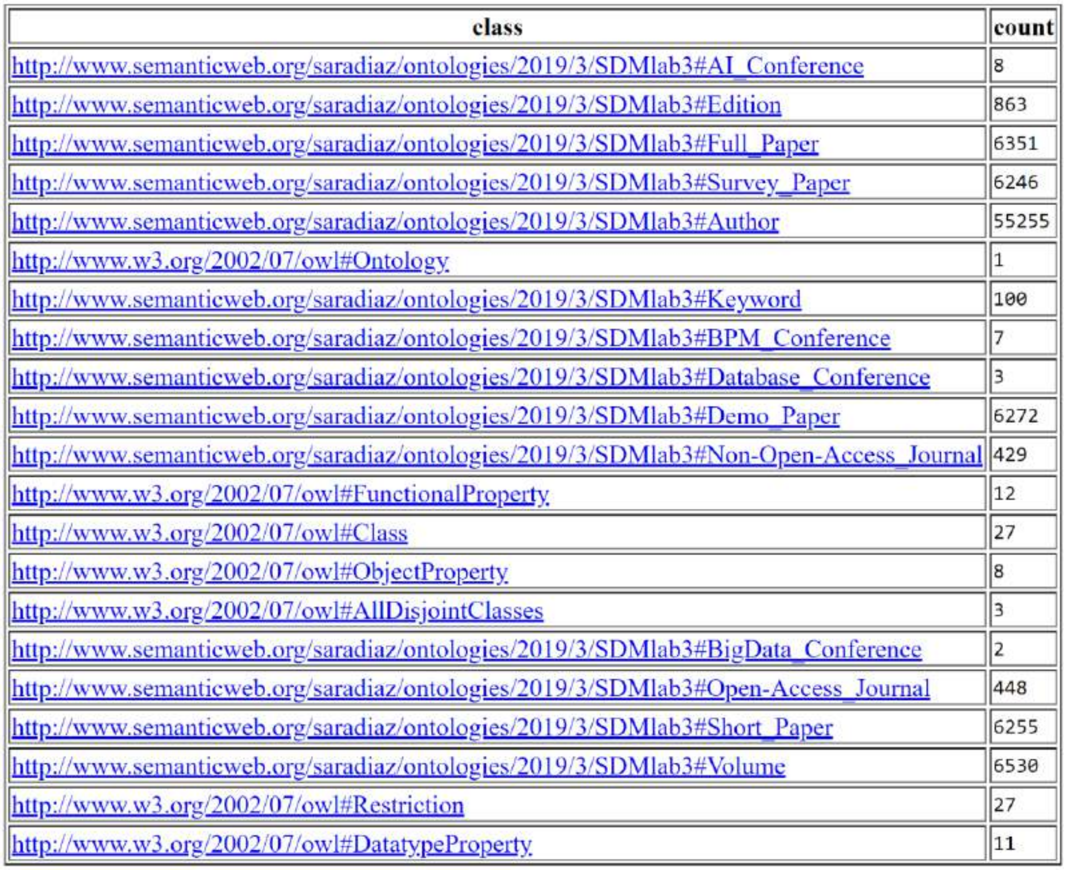
# classes	20
# object properties	8
# data properties	11
# class instances	88,858
# triples	504,297

Total number of triples per graph



graph	triples
http://localhost:8890/research	504297
http://localhost:8890/dbpedia	28058915

Number of instances per class.



class	count
http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#AI_Conference	8
http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#Edition	863
http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#Full_Paper	6351
http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#Survey_Paper	6246
http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#Author	55255
http://www.w3.org/2002/07/owl#Ontology	1
http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#Keyword	100
http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#BPM_Conference	7
http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#Database_Conference	3
http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#Demo_Paper	6272
http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#Non-Open-Access_Journal	429
http://www.w3.org/2002/07/owl#FunctionalProperty	12
http://www.w3.org/2002/07/owl#Class	27
http://www.w3.org/2002/07/owl#ObjectProperty	8
http://www.w3.org/2002/07/owl#AllDisjointClasses	3
http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#BigData_Conference	2
http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#Open-Access_Journal	448
http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#Short_Paper	6255
http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#Volume	6530
http://www.w3.org/2002/07/owl#Restriction	27
http://www.w3.org/2002/07/owl#DatatypeProperty	11

C4. Queries on top of the Ontology

Write two versions for each of the following queries (one exploiting the TBOX, and another assuming the TBOX does not exist). Please explicitly state any assumptions you make.

1. Find all the Authors.

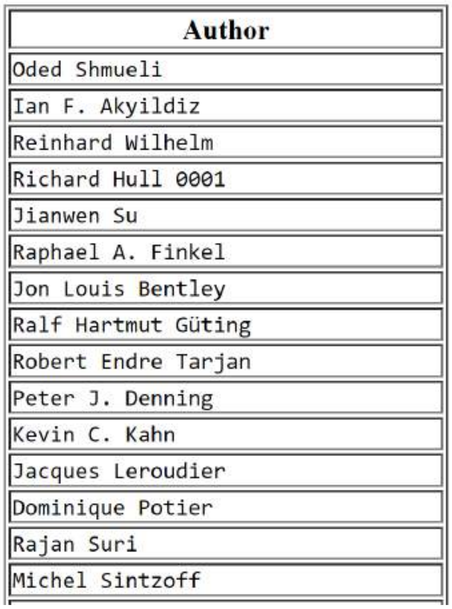
A. **Exploiting TBOX**

```
prefix schema: <http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#>
prefix dbpediaOnt: <http://dbpedia.org/ontology/>
prefix dbpediaProp: <http://dbpedia.org/property/>

select str(?Author) as ?Author
Where
{
      ?s a schema:Author.
      ?s dbpediaProp:name ?Author
}
```

B. Assuming TBOX does not exist

Note: *mainAuthor* object property (Author, Paper) has *writes* object property as replicated (meaning that whenever *mainAuthor* triplet is created, *writes* relation is also created for the same (Author, Paper). As a result, there is no need for *mainAuthor* to be included into the query below.

```
prefix schema: <http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#>
prefix dbpediaOnt: <http://dbpedia.org/ontology/>
prefix dbpediaProp: <http://dbpedia.org/property/>

select distinct str(?author) as ?author
where
{
   {?a schema:writes ?x}
   UNION
   {?a schema:reviews ?x}

   ?a dbpediaProp:name ?author
}
```



Author
Oded Shmueli
Ian F. Akyildiz
Reinhard Wilhelm
Richard Hull 0001
Jianwen Su
Raphael A. Finkel
Jon Louis Bentley
Ralf Hartmut Güting
Robert Endre Tarjan
Peter J. Denning
Kevin C. Kahn
Jacques Leroudier
Dominique Potier
Rajan Suri
Michel Sintzoff

… more

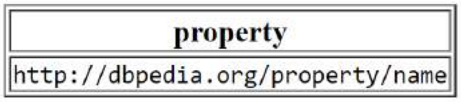
2. Find all the properties whose domain is Author.

A. Exploiting TBOX

```
PREFIX schema: <http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#>

select str(?prop) as ?property
where { ?prop rdfs:domain schema:Author }
```

B. Assuming TBOX does not exist

```
prefix schema: <http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#>
prefix dbpediaOnt: <http://dbpedia.org/ontology/>
prefix dbpediaProp: <http://dbpedia.org/property/>

select distinct str(?property) as ?property where
{
   {?a schema:writes ?x}
   UNION
   {?a schema:reviews ?x}

   ?a ?property ?value
}
```



property
http://dbpedia.org/property/name

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

A. Exploiting TBOX

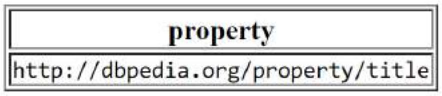
Note: both Conference and Journal have the same property "dbpediaProp:title". That is why with the use of *distinct* the property appears only once.

```
PREFIX schema: <http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#>
prefix dbpediaOnt: <http://dbpedia.org/ontology/>
prefix dbpediaProp: <http://dbpedia.org/property/>

select distinct str(?prop) as ?property
where {
   {?prop rdfs:domain schema:Conference }
   UNION
   {?prop rdfs:domain dbpediaOnt:AcademicJournal }
}
```

B. Assuming TBOX does not exist

Note: *schema:part_of* only has Journal as range and schema:belongs_to has only Conference as range, thus we use those in order to get journals and conferences respectively, and afterwards extract their properties.

```
prefix schema: <http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#>
prefix dbpediaOnt: <http://dbpedia.org/ontology/>
prefix dbpediaProp: <http://dbpedia.org/property/>

select distinct str(?property) as ?property
where
{
   {
    select distinct ?conf_journal
      where
      {
        {?a schema:part_of ?conf_journal}
        UNION
        {?x schema:belongs_to ?conf_journal}
      }
   }

   ?conf_journal ?property ?value
}
```



property
http://dbpedia.org/property/title

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

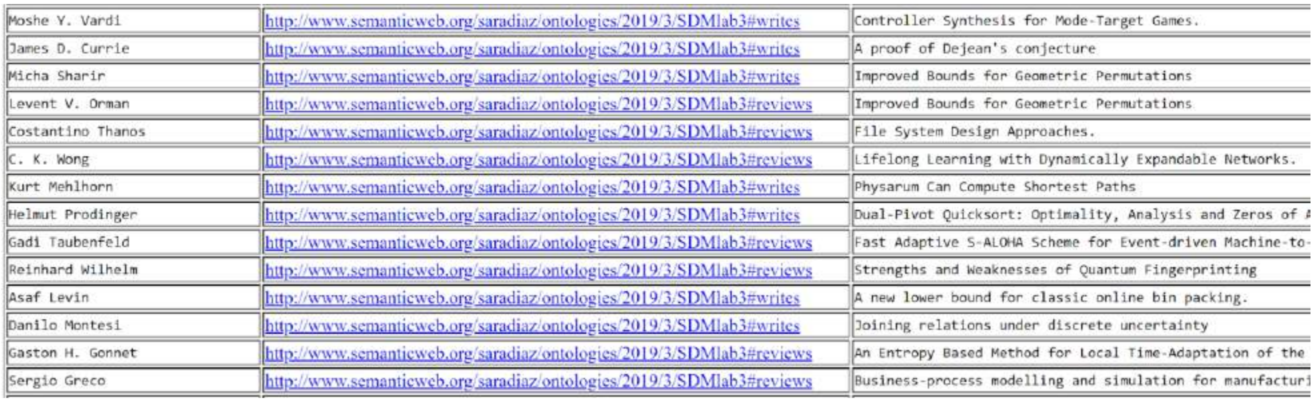
A. Exploiting TBOX

Note: Both reviews and writes refer to Papers that have an attribute title.

```
PREFIX schema: <http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#>
prefix dbpediaOnt: <http://dbpedia.org/ontology/>
prefix dbpediaProp: <http://dbpedia.org/property/>

select str(?author) as ?author ?pr str(?title) as ?title
where
{
   ?s rdf:type schema:Author .
   ?s dbpediaProp:name ?author .
   ?s ?pr ?thing .
   ?thing dbpediaProp:title ?title
}
```

B. Assuming TBOX does not exist

```
PREFIX schema: <http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#>
prefix dbpediaOnt: <http://dbpedia.org/ontology/>
prefix dbpediaProp: <http://dbpedia.org/property/>

select str(?name) as ?name ?created str(?title) as ?title
where
{
   {
    select distinct ?a
      where
      {
        {?a schema:writes ?x}
        UNION
        {?a schema:reviews ?x}
      }
   }

   ?a ?created ?things.
   ?things dbpediaProp:title ?title .
   ?a dbpediaProp:name ?name
}
```



Moshe Y. Vardi	http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#writes	Controller Synthesis for Mode-Target Games.
James D. Currie	http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#writes	A proof of Dejean's conjecture
Micha Sharir	http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#writes	Improved Bounds for Geometric Permutations
Levent V. Orman	http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#reviews	Improved Bounds for Geometric Permutations
Costantino Thanos	http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#reviews	File System Design Approaches.
C. K. Wong	http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#reviews	Lifelong Learning with Dynamically Expandable Networks.
Kurt Mehlhorn	http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#writes	Physarum Can Compute Shortest Paths
Helmut Prodinger	http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#writes	Dual-Pivot Quicksort: Optimality, Analysis and Zeros of
Gadi Taubenfeld	http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#reviews	Fast Adaptive S-ALOHA Scheme for Event-driven Machine-to-
Reinhard Wilhelm	http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#reviews	Strengths and Weaknesses of Quantum Fingerprinting
Asaf Levin	http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#writes	A new lower bound for classic online bin packing.
Danilo Montesi	http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#writes	Joining relations under discrete uncertainty
Gaston H. Gonnet	http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#reviews	An Entropy Based Method for Local Time-Adaptation of the
Sergio Greco	http://www.semanticweb.org/saradiaz/ontologies/2019/3/SDMlab3#reviews	Business-process modelling and simulation for manufacturi

… more