Semantic Data Management
Lab 3- Knowledge Graphs

Section A :Exploring DBpedia

For this section, we use Graphdb server local installation

1. Get the classes defined in the ontology.

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
SELECT ?s WHERE {
       ?s ?p owl:Class .
}
```

2. Get the datatype properties defined in the ontology.

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
SELECT ?s ?p WHERE {
?s ?p owl:DatatypeProperty.
}
```

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

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
SELECT ?s ?p WHERE {
?s ?p owl:ObjectProperty.
}
```

The difference between these two properties is that Datatype properties relate individuals to the literials(numbers, strings,etc) while object properties relate individuals to other individuals.

In other words, both domain and ranges of object property are individuals , whereas, in case of datatype property the domain is an individual and the range is a literal.

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

```
SELECT * WHERE {
?s ?p <http://www.w3.org/1999/02/22-rdf-syntax-ns#Property>.
}
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?label
WHERE {
{
?predicate a owl:DatatypeProperty;
             rdfs:label ?label.
}
UNION
{
?predicate a owl:ObjectProperty;
             rdfs:label ?label.
}}
```

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

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?class ?label
WHERE
{
?class rdf:type owl:Class;
        rdfs:label ?label
       FILTER(?label='actor'@en)
}
```

6. Find the super class for the class Actor.

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT DISTINCT ?superclass ?subclass
 WHERE
   {
       ?subclass  rdf:type  owl:Class.
       ?subclass  rdfs:subClassOf  ?superclass;
             rdfs:label ?label
       FILTER(?label="actor"@en)
```

```
   }
```

7. Find all the actors in the dataset.

```
PREFIX db: <http://dbpedia.org/ontology/>
SELECT (strAfter (str(?s), "http://dbpedia.org/resource/" ) as ?Actor) WHERE {
       ?s ?p db:Actor .}
```

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

```
PREFIX : <http://dbpedia.org/ontology/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT * WHERE {
        ?Class rdfs:domain :Actor
}
```

9. Find the super property of the goldenRaspberryAward property.

```
PREFIX : <http://dbpedia.org/ontology/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT * WHERE {
        :goldenRaspberryAward rdfs:subPropertyOf ?Super_property
}
```

An alternative

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT (?subcp as ?subclassproperty) (?supcp as ?superclassproperty)
WHERE {
?subcp  rdf:type rdf:Property.
?subcp  rdfs:subPropertyOf ?supcp.
?subcp  rdfs:label"Golden Raspberry Award"@en
  }
```

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

```
PREFIX dbo:<http://dbpedia.org/ontology/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT(?x as ?Domain)(?y as ?Range)
WHERE
{{
?x rdf:type rdf:Property;
   rdfs:domain dbo:Actor
       }
UNION
{
?y rdfs:range dbo:Actor
}
}
```

11. Return all persons that are not actors.

```
PREFIX db: <http://dbpedia.org/ontology/>
SELECT (strAfter (str(?s), "http://dbpedia.org/resource/" ) as ?Person_notActor)
WHERE {
       ?s ?p <http://dbpedia.org/ontology/Person>
       FILTER NOT EXISTS {?s ?p db:Actor}
}
```

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

```
PREFIX : <http://dbpedia.org/ontology/>
SELECT ?x ?p ?o WHERE {
  :Actor (:|!:)* ?x .
  ?x ?p ?o .
  ?o (:|!:)* :Person .
}
```

Section B:Analytical queries on top of QBAirbase

For this section we use graphDB server from **http://130.226.98.179:7200/**

1.List the country, station type, latitude, and longitude details of each station.

```
PREFIX Schema: <http://qweb.cs.aau.dk/airbase/schema/>
PREFIX Property: <http://qweb.cs.aau.dk/airbase/property/>
SELECT ?StationName ?StationType ?Country ?Latitude ?Longitude
WHERE{
?station Property:station ?StationName;
         Property:longitudeDegree ?Longitude;
         Property:latitudeDegree ?Latitude;
         Property:type ?StationType;
         Schema:inCountry ?country.
?country Property:country ?Country
} ORDER BY ASC (?Country)
```

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

```
PREFIX Observation: <http://purl.org/linked-data/cube#Observation>
PREFIX Station: <http://qweb.cs.aau.dk/airbase/schema/station>
PREFIX Schema: <http://qweb.cs.aau.dk/airbase/schema/>
PREFIX Property: <http://qweb.cs.aau.dk/airbase/property/>
select ?sensor_ID ?AverageEmission ?Country ?Year
where {?observation a Observation:.
       ?observation Schema:sensor ?sensor.
       ?sensor Property:europeanCode ?sensor_ID.
       ?sensor Property:statisticName "annual mean".
       ?observation Schema:C6H6 ?AverageEmission.
       ?observation Schema:year ?year.
       ?year Property:yearNum ?Year.
       ?observation Station: ?Station.
       ?Station Schema:inCountry ?country.
       ?country Property:country ?Country.
```

```
}order by desc(?AverageEmission) LIMIT 10
```

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

```
PREFIX Observation: <http://purl.org/linked-data/cube#Observation>
PREFIX Station: <http://qweb.cs.aau.dk/airbase/schema/station>
PREFIX Schema: <http://qweb.cs.aau.dk/airbase/schema/>
PREFIX Property: <http://qweb.cs.aau.dk/airbase/property/>
select  ?Year ?type ?CityName ?AverageNO2 ?AverageSO2 ?AveragePB ?AveragePM10
where {
       # Averge NO2
       ?obsno2 a Observation:.
       ?obsno2 Schema:NO2 ?AverageNO2.
       ?obsno2 Station: ?Station.
       ?obsno2 Schema:year ?year.
       #Aerage SO2
       ?obsso2 a Observation:.
       ?obsso2 Schema:SO2 ?AverageSO2.
       ?obsso2 Station: ?Station.
       ?obsso2 Schema:year ?year.
       # Average PB
       ?obspb a Observation:.
       ?obspb Schema:PB ?AveragePB.
       ?obspb Station: ?Station.
       ?obspb Schema:year ?year.
       # Average PM10
       ?obspm a Observation:.
       ?obspm Schema:PM10 ?AveragePM10.
       ?obspm Station: ?Station.
       ?obspm Schema:year ?year.

       ?sensorno2 Property:europeanCode ?sensor_ID.
       ?sensorso2 Property:europeanCode ?sensor_ID.
       ?sensorpb Property:europeanCode ?sensor_ID.
       ?sensorpm Property:europeanCode ?sensor_ID.


       ?obsno2 Schema:sensor ?sesensorno2.
       ?obsso2 Schema:sensor ?sesensorso2.
       ?obspb Schema:sensor ?sesensorpb.
       ?obspm Schema:sensor ?sesensorpm.

       ?year Property:yearNum ?Year.
       ?Station Property:type ?type.
       ?Station Schema:inCity ?city.
       ?city Property:city ?CityName.

       ?sensorno2 Property:statisticName "annual mean".
       ?sensorso2 Property:statisticName "annual mean".
```

```
      ?sensorpb Property:statisticName "annual mean".
      ?sensorpm Property:statisticName "annual mean".
}
  GROUP BY ?type ?CityName ?Year ?AverageNO2 ?AverageSO2 ?AveragePB ?AveragePM10
```

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) : Average Annual Nitrogen dioxide(NO2) and PM10 consumption by country, Industrial station and urban area.

The query helps us to know which countries are violating the standard set by European air quality agencies by consuming more annual NO2 and PM10 from the industry section and urban area.

```
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  (STR(?countryname) as ?CountrName) (avg(?no2) as ?avgno2consumption)
(avg(?pm10) as ?averagepm10) ?Year
WHERE {
       ?no2obs schema:NO2 ?no2.
       ?no2obs schema:station ?station.
       ?pm10obs schema:PM10 ?pm10.
       ?pm10obs schema:station ?station.
       ?station property:longitudeDegree ?long .
       ?station property:latitudeDegree ?lat .
       ?station property:type ?type.
       ?station property:areaType ?area.

       ?station schema:inCountry ?country.
       ?country property:country ?countryname.
       ?no2obs schema:year ?year.
       ?year property:yearNum ?Year.
       ?pm25obs schema:year ?year.
       ?year property:yearNum ?Year.

       ?pm10obs schema:year ?year.
       ?year property:yearNum ?Year.
    {
       ?station schema:inCity ?city .
       ?city schema:locatedIn ?country .
       } UNION {
       ?station schema:inCountry ?country.
       ?country property:country ?countryname.
```

```
       }
     ?country property:isoCode ?isocode .

       ?no2obs  schema:sensor ?sensorno2.
       ?pm10obs schema:sensor ?sensorpm10.
       ?sensorno2 property:statisticShortName "Mean"^^xsd:string.
       ?sensorpm10 property:statisticShortName "Mean"^^xsd:string
       FILTER(?Year>2008 && ?type="Industrial"^^xsd:string &&
?area="urban"^^xsd:string)
       FILTER(?no2>40 && ?pm10 >40)
}Group BY ?area ?type ?no2 ?pm10 ?countryname ?Year
ORDER BY DESC(?no2) (?pm10)
LIMIT 25
```

4-2) : Average annual no2 consumption of stations found in major cities.

This helps us to know about which stations found in a major cities consumes high annual no2 and how it is related with time. And through the concept of rollup to deduce information about cities and countries' consumption information. From the result we observed that stations found in major northern europe(stockholm, copenhagen and oslo) consume less no2 compared to Madrid, Rome, and London.

```
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 (STR(?station)as ?StationName) (avg(?no2) as ?AverageNO2)  (STR(?city) as
?CITY) ?Year WHERE {
       ?obsno2 Schema:NO2 ?no2 .
       ?obsno2 Schema:station ?stationOb .
       ?stationOb Property:station ?station.
       ?stationOb Schema:inCountry ?country.

       ?stationOb Schema:inCity ?cityObj.
       ?obs Schema:year ?year.

       ?year Property:yearNum ?Year.
       ?cityObj Property:city ?city.
       ?country Property:country ?countryName.
       ?obs Schema:sensor ?sensor .
       ?sensor Property:statisticShortName "Mean"^^xsd:string.
```

```
FILTER(?city="STOCKHOLM"^^xsd:string||?city = "BERLIN"^^xsd:string || ?city =
"COPENHAGEN"^^xsd:string ||
?city="OSLO"^^xsd:string||?city="BARCELONA"^^xsd:string
||?city="BARCELONA"^^xsd:string ||?city="AMSTERDAM"^^xsd:string
||?city="MADRID"^^xsd:string ||?city="LONDON"^^xsd:string
||?city="ROME"^^xsd:string)
FILTER(?Year>=2008)
 } GROUP BY ?station ?no2 ?Year ?countryName ?city HAVING(?no2>40)
ORDER BY desc(?no2)
```

4-3: Total number of cities and their yearly average consumption grouped by Country Name.

This query helps us to know the number of cities that violates air quality standards regulations and understand the correlation with time. The rank is based on the combination of the different air pollutants not an individual pollutant

```
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 (count(?city) as ?Numberofcities) ?Year (STR(?countryName) as
?CountrName)
WHERE {
{
       { ?obs Schema:PM10 ?pm10 .
       FILTER(?pm10 > 40) }
       UNION
       { ?obs Schema:NO2 ?no2.
       FILTER(?no2 > 40)  }
       UNION
       { ?obs Schema:Pb ?pb .
       FILTER(?pb > 0.5) }
       UNION
       { ?obs Schema:Cd ?cd .
       FILTER(?cd > 5) }
       UNION
       { ?obs Schema:Ni ?ni .
              FILTER(?ni > 20) }
       UNION
       { ?obs Schema:As ?as .
              FILTER(?as > 6) }
       UNION
       { ?obs Schema:C6H6 ?c6h6 .
```

```
            FILTER(?c6h6 > 6) }
      UNION
      { ?obs Schema:PM2.5 ?pm25 .
      FILTER(?pm25 > 25) } }
      ?obs Schema:station ?station.
      ?station Schema:inCountry ?country.
      ?country Property:country ?countryName.
      ?station Schema:inCity ?city .
      ?obs Schema:year ?year .
      ?year Property:yearNum ?Year .
      ?obs Schema:sensor ?sensor .
      ?sensor Property:statisticShortName "Mean"^^xsd:string.
      FILTER(?Year>=2008)
} GROUP BY ?Year ?countryName ?Name
```

Where the values in the filter are the maximum annual average allowed for different pollutants.

Section C: Ontology Creation

C.1 TBOX definition

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

To define the TBOX we use **Protege** ontology editor because it contains practical tutorials and well structured documentations. We based our design using the property graph of our first lab and also ,we extended the design to include concepts like organization, editions and reviews.

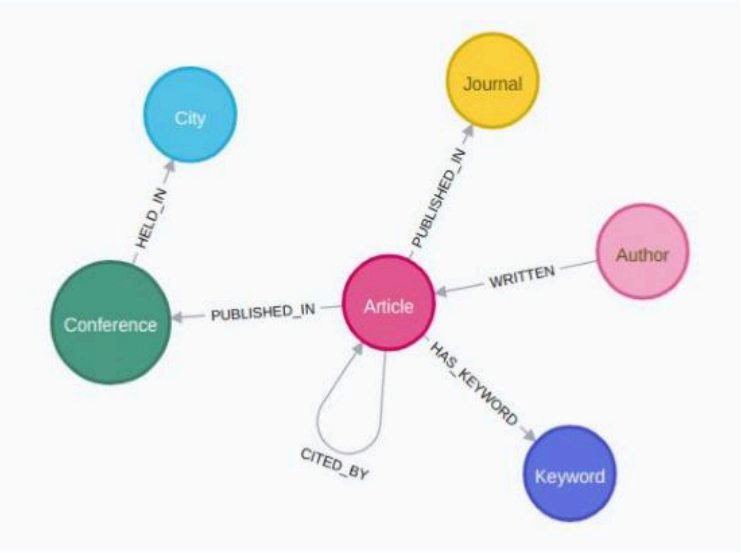


Figure 1 : Publication property graph

Therefore, the design of all the classes , objects, and datatypes properties are based on the property graph model shown above and the data we have, and the set of queries we need to apply on top of the ontology.

Therefore, the TBox is made of from three major constructs

1. **Classes/Concepts**
2. **Object properties/roles**
3. **Datatype properties**

Classes

We defined the following core classes with their constraints

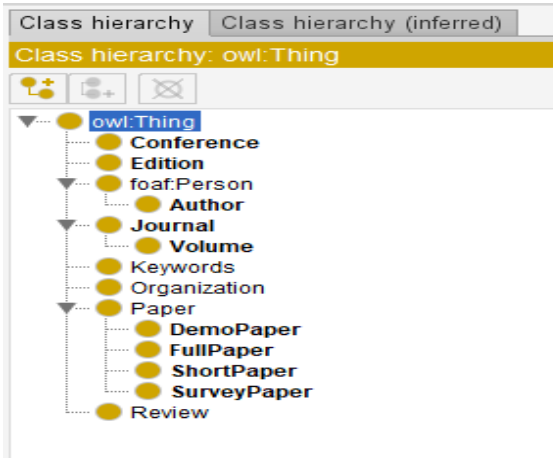


Figure 2: Classes

Except for the Author class where we defined it as a subclass of **foaf:Person** to extend its semantics with foaf vocabulary, we defined all classes as standalone classes.

- **Paper and its subclasses**: We defined the DemoPaper, FullPaper, ShortPaper, and Survey as a subclass of Paper class. To keep the semantics of we make the different types of papers disjoint to each other.

```
<!-- http://www.semanticweb.org/ontology/Paper -->

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

DemoPaper and Full Paper ontology

```
<!-- http://www.semanticweb.org/ontology/DemoPaper -->

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

<!-- http://www.semanticweb.org/ontology/FullPaper -->

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

All paper types are disjoint.

```
    -->

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

By default, OWL classes have some overlap and to avoid such misconception in the paper domain we make all the different papers disjoint.

- **Conference and Edition**: In our design we decided different conference types such as **database conference**, **semantic web conference**,etc to be recognized as a conference concept to make the design simple. And a given conference has multiple editions.

```
<!-- http://www.semanticweb.org/ontology//Conference -->

<owl:Class rdf:about="http://www.semanticweb.org/ontology//Conference">
    <owl:disjointWith rdf:resource="http://www.semanticweb.org/ontology//Journal"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontology//Edition -->

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

  Moreover, we make the conference and Journal classes disjoint to avoid the same paper not to be published twice.
- Keyword, Organization and Review
  The keyword, Organization and Review classes are defined as standalone classes.

```
<!-- http://www.semanticweb.org/ontology//Keywords -->

<owl:Class rdf:about="http://www.semanticweb.org/ontology//Keywords"/>
<!-- http://www.semanticweb.org/ontology//Organization -->

<owl:Class rdf:about="http://www.semanticweb.org/ontology//Organization"/>

<!-- http://www.semanticweb.org/ontology//Review -->

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

- Author:The author class is defined as a subclass of the foaf:Person class.

```
<!-- http://www.semanticweb.org/ontology/Author -->

<owl:Class rdf:about="http://www.semanticweb.org/ontology/Author">
    <rdfs:subClassOf rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
</owl:Class>
```

This helps us to pose a query against the **foaf:Person/Author** class especially when considering querying the ontology without assuming the local TBox.

- **Edition and Volume**- We make the edition and volume classes to be disjoint classes using the same reasoning of Journal and Conference classes.

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

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

We avoid putting many restrictions on the various classes we have to make it simple and easily understandable.

Object Properties

Object properties are roles/relationships between two individuals/objects. In our TBox we identified 10 object properties.

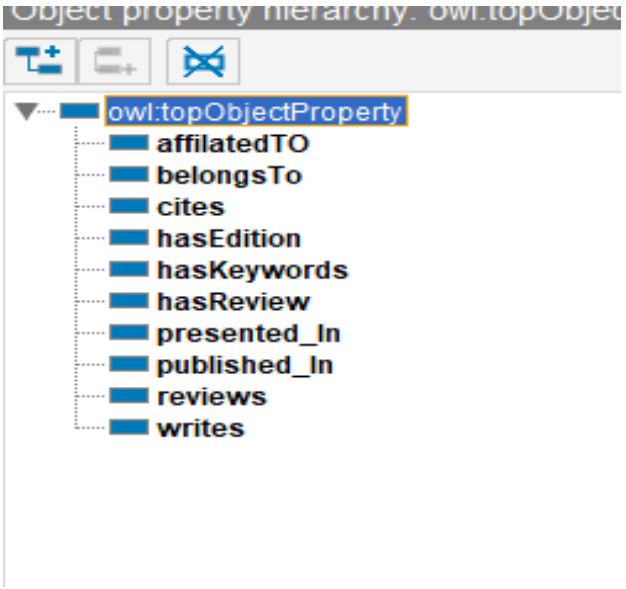


Figure 3: Object Properties

- **affilatedTO:** This property links authors to the company they are affiliated with. The domain and range are authors and organizations respectively. Moreover, an author is only affiliated to only one organization.

```
<!-- http://www.semanticweb.org/property/affilatedTO -->

<owl:ObjectProperty rdf:about="http://www.semanticweb.org/property/affilatedTO">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
    <rdfs:domain rdf:resource="http://www.semanticweb.org/ontology/Author"/>
    <rdfs:range rdf:resource="http://www.semanticweb.org/ontology//Organization"/>
    <rdfs:comment>&apos;An author affilated to an organization&apos;</rdfs:comment>
</owl:ObjectProperty>
```

- **belongsTo**: This property links volume to a Journal as volume belongs to a given Journal.

```
<!-- http://www.semanticweb.org/property/belongsTo -->

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

- **hasEdition**:This predicate represents that a given conference has many editions

```
<!-- http://www.semanticweb.org/property/hasEdition -->

<owl:ObjectProperty rdf:about="http://www.semanticweb.org/property/hasEdition">
    <rdfs:domain rdf:resource="http://www.semanticweb.org/ontology//Conference"/>
    <rdfs:range rdf:resource="http://www.semanticweb.org/ontology//Edition"/>
    <rdfs:comment>&apos;Conferenece has editions&apos;</rdfs:comment>
</owl:ObjectProperty>
```

- **cites**: A paper cites another paper. Paper is the domain and range

```
<!-- http://www.semanticweb.org/property/cites -->

<owl:ObjectProperty rdf:about="http://www.semanticweb.org/property/cites">
    <rdfs:domain rdf:resource="http://www.semanticweb.org/ontology/Paper"/>
    <rdfs:range rdf:resource="http://www.semanticweb.org/ontology/Paper"/>
    <rdfs:comment>&apos; Paper cities another paper&apos;</rdfs:comment>
</owl:ObjectProperty>
```

- **hasKeywords**: A given paper has set of keywords

```
<!-- http://www.semanticweb.org/property/cites -->

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

- **hasReview**: Both **Journal** and **Conference** have **reviews** and hasReview object creates the link in the graph.

```
<!-- http://www.semanticweb.org/property/hasReview -->

<owl:ObjectProperty rdf:about="http://www.semanticweb.org/property/hasReview">
    <rdfs:domain rdf:resource="http://www.semanticweb.org/ontology//Conference"/>
    <rdfs:domain rdf:resource="http://www.semanticweb.org/ontology//Journal"/>
    <rdfs:range rdf:resource="http://www.semanticweb.org/ontology//Review"/>
    <rdfs:comment>&apos;Jounrals and conferences have reviews&apos;</rdfs:comment>
</owl:ObjectProperty>
```

- **presented_In and published_In::** A paper is presented in a conference and published in a given Journal.

  Note: both presented_In and published_In refer to the similar concept with subtle differences. A paper first needs to be presented in a conference before getting published in a specific edition of a conference whereas in case of Journal only review is enough to get published.

To avoid the same paper to get published in both journal and conference editions we make **published_In** and **presented_In** disjoint. For example, If paper1 is published in **IEEE(Journal name)** then it is less likely to be published in a Database conference of **edition 1**.

```
<!-- http://www.semanticweb.org/property/presented_In -->

<owl:ObjectProperty rdf:about="http://www.semanticweb.org/property/presented_In">
    <rdfs:domain rdf:resource="http://www.semanticweb.org/ontology/Paper"/>
    <rdfs:range rdf:resource="http://www.semanticweb.org/ontology//Conference"/>
    <owl:propertyDisjointWith rdf:resource="http://www.semanticweb.org/property/published_In"/>
</owl:ObjectProperty>

<!-- http://www.semanticweb.org/property/published_In -->

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

- **reviews:** an author gives review to paper. Before a paper is published either in a given edition of **conference** and journal the paper needs to be reviewed by a set of reviewers which are authors of another paper.

```
<!-- http://www.semanticweb.org/property/reviews -->

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

- **writes**: An author writes a paper which can be any of the paper types (Demo, Short, Survey, and Full paper)

```
<!-- http://www.semanticweb.org/property/writes -->

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

DataType Properties

These properties relate concepts(individuals) to literals. We identified the following core properties.

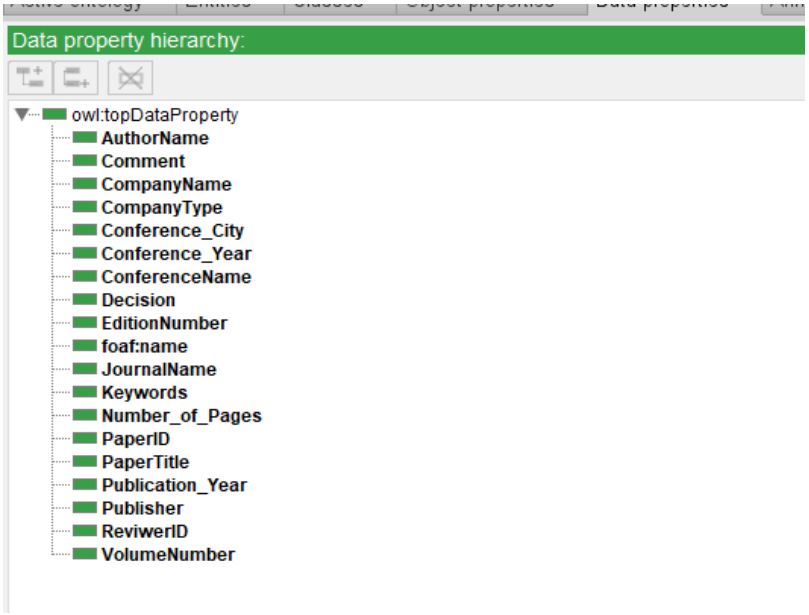


Figure 4: DataType Properties

Note: All the properties have a class as domain and a literal as a range in the form of xsd:datatype.

- **AuthorName: An identifier that** represents the name of a given author.

```
<!-- http://www.semanticweb.org/property/AuthorName -->

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

- **Comment**: An identifier that stores comments of a given review.

```
<!-- http://www.semanticweb.org/property/Comment -->

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

- **Decision**: an identifier that stores the final remark of the review for a given paper made by a a reviewer

```
<!-- http://www.semanticweb.org/property/Decision -->

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

- **CompanyName** and **CompanyType**: These two columns store the type of organization(i.e University or any other organization) and specific name of an organization(i.e UPC, Facebook).

```
<!-- http://www.semanticweb.org/property/CompanyName -->

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

<!-- http://www.semanticweb.org/property/CompanyType -->

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

- **ConferenceName ,Conference_City, and Conference_Year**:These columns store the name of the conference, city where an edition of conference is held, and year when the conference is held.

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

<!-- http://www.semanticweb.org/property/Conference_City -->

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

<!-- http://www.semanticweb.org/property/Conference_Year -->

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

- **EditionNumber**: This property represents an edition number of a conference because, conference has multiple editions.

```
<!-- http://www.semanticweb.org/property/EditionNumber -->

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

- **JournalName**: Name of the given journal where a given paper is published.

```
<!-- http://www.semanticweb.org/property/JournalName -->

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

- **Keywords**: set of indexes(keywords) that a given paper can contain.

```
<!-- http://www.semanticweb.org/property/Keywords -->

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

- **PaperID**: A property that uniquely identifies a paper. This is not equivalent to ISBN of a given book, rather this is a custom Identifier used to identify a paper from the rest in the dataset we have.

```
<!-- http://www.semanticweb.org/property/PaperID -->

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

- **PaperTitle**: Property that represents a human readable title of a paper

```
<!-- http://www.semanticweb.org/property/PaperTitle -->

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

- **Number_of_Pages**: property that shows the number of pages of a given paper.

  Note: The domain is paper and the range is **xsd:string** because in our dataset we use a range of page numbers(for example 350-360) as a total number of pages in the paper.

```
<!-- http://www.semanticweb.org/property/Number_of_Pages -->

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

- **Publication_Year**: This property refers to the year when a given paper is published.

```
<!-- http://www.semanticweb.org/property/Publication_Year -->

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

- **Publisher**: This property represents the name of the **publisher.**

```
<!-- http://www.semanticweb.org/property/Publisher -->

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

- **ReviewerID**: This property refers to an ID assigned to a given **reviewer**.

```
<!-- http://www.semanticweb.org/property/ReviwerID -->

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

- **VolumeNumber**: This property shows the volume number of a paper published in a given Journal.

```
<!-- http://www.semanticweb.org/property/VolumeNumber -->

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

C. 2. Visual Representation of TBOX

To visualize the TBox we use **VOWL** which is a Protege plugin to represent visual notations of ontology graphs. In the diagram classes are represented with circles, data type properties with rectangles and predicates(object properties) with arrows.

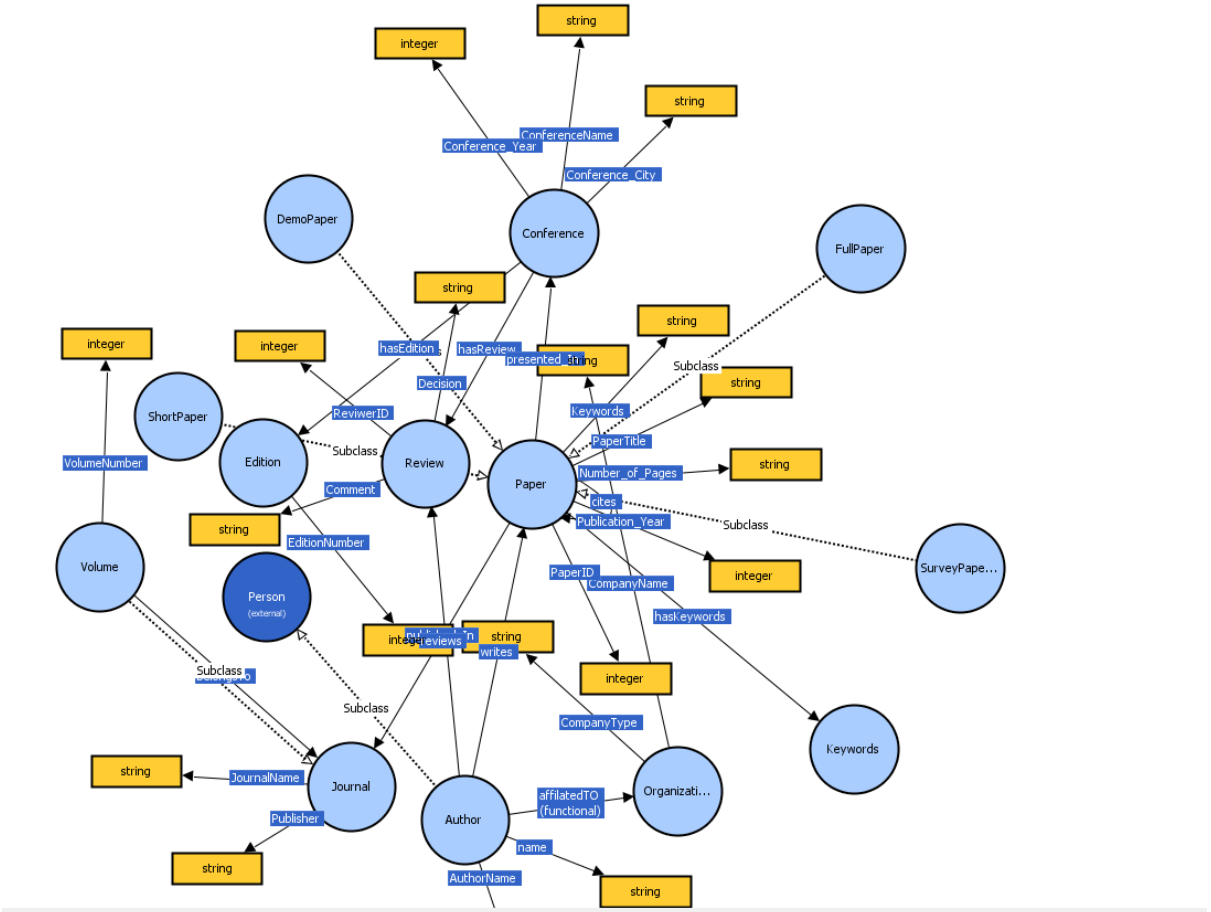


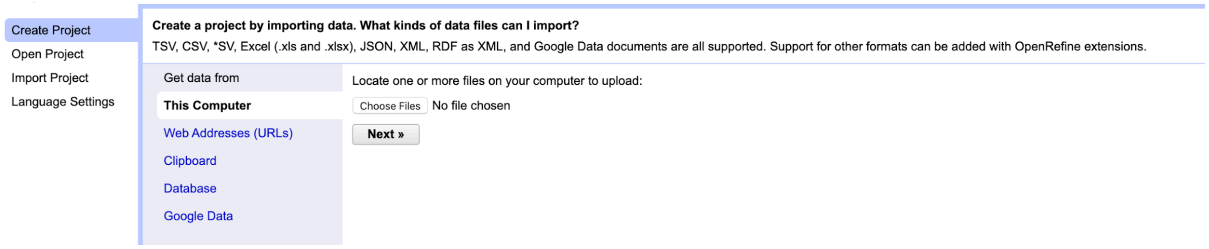
Figure 5: Visual representation of TBox

C.2 ABOX definition

Explain the method used to define the ABOX.

The method we used to convert csv files and define the ABOX is by using OpenRefine along with two extension VIB-BITS and RDF extension.

- Step 1: Choose the csv file from our computer then choose create project. Then we can remove the columns that we don't need. We do it repeatedly until all the csv files we need are imported.

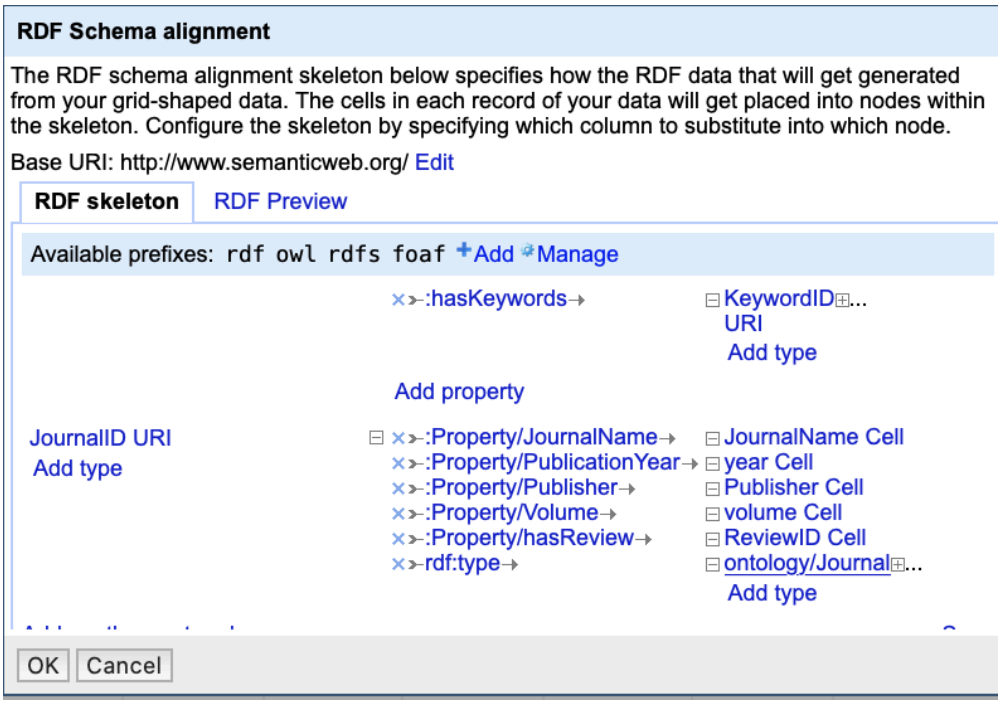


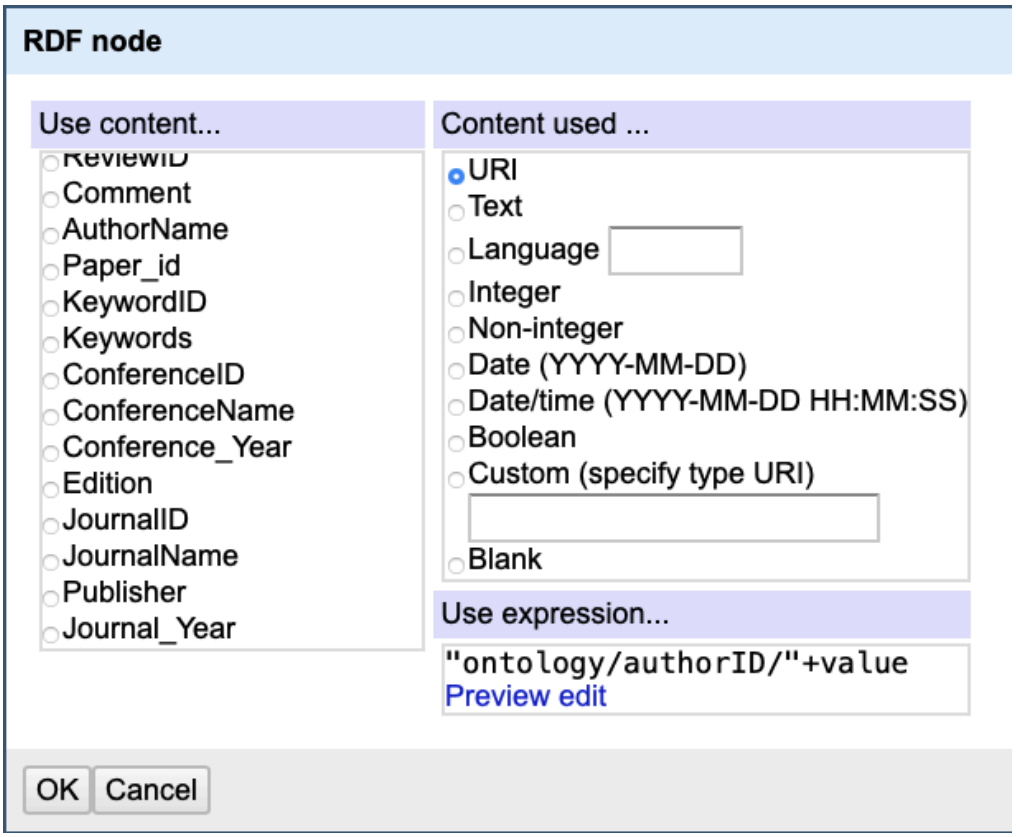
- Step2: Add the related columns from other csv files based on the SELECTed column. We used ID (AuthorID, PaperID, ReviewID, ConferenceID, JournalID, etc...) as a key to do this step. We repeatedly doing this step until all the columns we need are added to the current project we are working on.

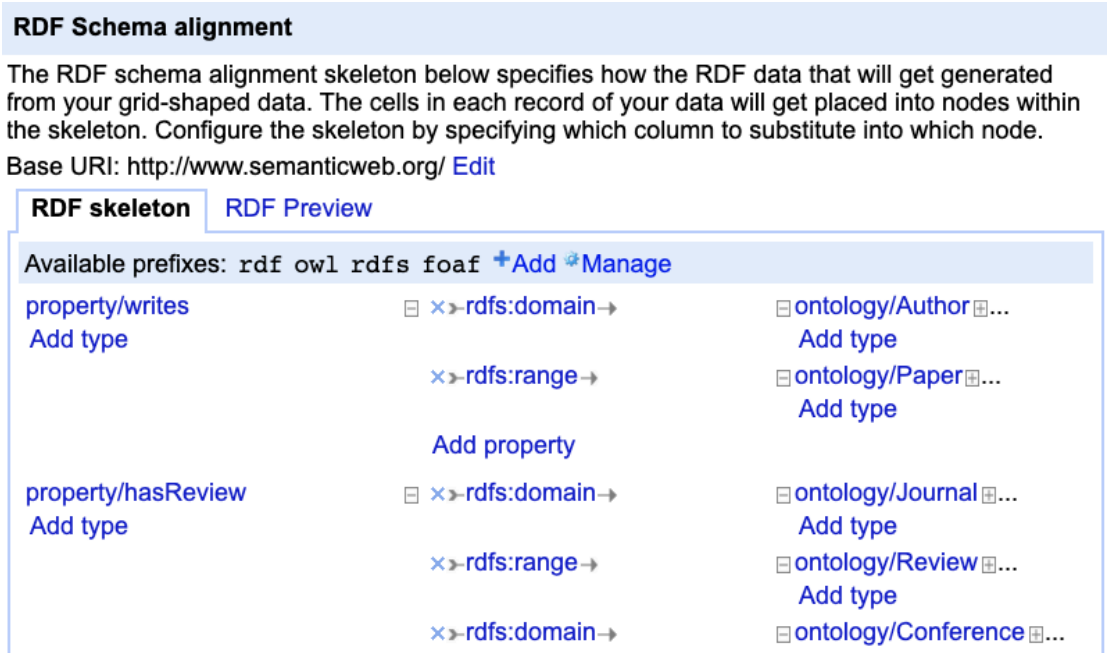
Author_id	ReviewID	Comment	AuthorNam	Paper_id	Conferenc	ConferenceName	Year	Edition	JournalID	JournalNan	ISSN	Publisher	year	volume	PaperTitle	pages
nt		asdkjfhaksjfh kkjhaskj	Garcia-Molina			Database Management									identification of robot sensing data based on nested infinite GMMs.	3167
7433575	3	dsafbasdm afjashkjhkjhsdjfh akj kahsdfhkashdk	Gregor Erbach	98	4	Statistical and Scientific Database Management	2016	4							A Dynamic Periodic Distributing Scheme for Authentication data based on EAP-AKA in Heterogeneous Interworking Networks.	
7433573	1	sdafbsadfb asjdfjaskjdf kh asdkjfhaksjfh kkjhaskj	Héctor Garcia-Molina	196	4	Statistical and Scientific Database Management	2016	2							A simulation of a minicomputer-based data base transaction system.	543-553
7433575	3	dsafbasdm afjashkjhkjhsdjfh akj kahsdfhkashdk	Gregor Erbach	198	4	Statistical and Scientific Database Management	2016	4							Utility of data base management to analyze the output from complex simulations.	702-709
7433573	1	sdafbsadfb asjdfjaskjdf kh asdkjfhaksjfh kkjhaskj	Héctor Garcia-Molina	296					4	Expert Systems with Applications	0957-4174	Elsevier	2019	1	Optimizing distributed data bases: a framework for research.	473-478
7433575	3	dsafbasdm afjashkjhkjhsdjfh akj kahsdfhkashdk	Gregor Erbach	298					4	Expert Systems with Applications	0957-4174	Elsevier	2020	1	User/system interface within the context of an integrated corporate data base.	27-33
7433573	1	sdafbsadfb asjdfjaskjdf kh asdkjfhaksjfh kkjhaskj	Héctor Garcia-Molina	396					4	Expert Systems with Applications	0957-4174	Elsevier	2019	1	Mapping FPAR in China with modis time-series data based on the Wide Dynamic Range Vegetation Index.	2790-2793
7433575	3	dsafbasdm afjashkjhkjhsdjfh akj kahsdfhkashdk	Gregor Erbach	398					4	Expert Systems with Applications	0957-4174	Elsevier	2020	1	Correlation analysis between forest canopy density and LANDSAT TM data based on sub-compartment objects.	1717-1720
7433573	1	sdafbsadfb asjdfjaskjdf kh asdkjfhaksjfh kkjhaskj	Héctor Garcia-Molina	496					4	Expert Systems with Applications	0957-4174	Elsevier	2019	1	Machine learning in computer forensics (and the lessons learned from machine learning in computer security).	99-104
7433575	3	dsafbasdm afjashkjhkjhsdjfh akj kahsdfhkashdk	Gregor Erbach	498					4	Expert Systems with Applications	0957-4174	Elsevier	2020	1	Off the beaten path: machine learning for offensive security.	1-2
7433573	1	sdafbsadfb asjdfjaskjdf kh asdkjfhaksjfh kkjhaskj	Héctor Garcia-Molina	70	4	Statistical and Scientific Database Management	2016	1							SIGBDP 3 Performance measurement and	49-50

- Step3: Edit RDF skeleton according the TBOX we defined, after we finish apply our TBox model in the skeleton, we can export the project to rdf form(rdf/xml or turtle)

  Example:







Example of output rdf file

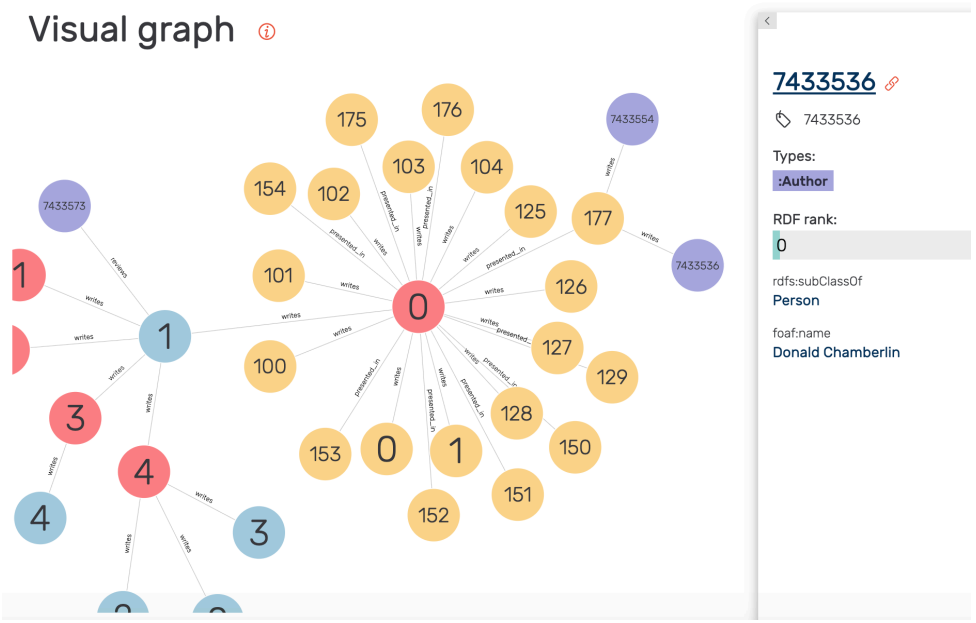
```
<rdf:Description rdf:about="http://www.semanticweb.org/ontology/paperID/0">
    <PaperTitle xmlns="http://www.semanticweb.org/property/">MBase: Representing mathematical knowledge in a relational data base.</PaperTitle>
    <rdf:type rdf:resource="http://www.semanticweb.org/ontology/Paper"/>
    <Number_of_Pages xmlns="http://www.semanticweb.org/property">451-468</Number_of_Pages>
</rdf:Description>

<rdf:Description rdf:about="http://www.semanticweb.org/ontology/paperID/0">
    <hasKeywords xmlns="http://www.semanticweb.org/property" rdf:resource="http://www.semanticweb.org/ontology/keywordID/5"/>
    <presented_in xmlns="http://www.semanticweb.org/property/" rdf:resource="http://www.semanticweb.org/ontology/conferenceID/0"/>
</rdf:Description>

<rdf:Description rdf:about="http://www.semanticweb.org/Ontology/conferenceID/0">
    <Edition xmlns="http://www.semanticweb.org/property/">1</Edition>
    <ConferenceYear xmlns="http://www.semanticweb.org/property/">2016</ConferenceYear>
    <ConferenceName xmlns="http://www.semanticweb.org/property/">VLDB</ConferenceName>
    <rdf:type rdf:resource="http://www.semanticweb.org/ontology/Conference"/>
</rdf:Description>

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

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

C.3 Linking ABOX to TBOX

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

   We use Openrefine to define our ABox according to our TBOX model. To confirm the link exists, we wrote some Construct & Insert into graph statements.
   By using the CONSTRUCT query in GraphDB, we can see the visualization of our dataset. The visual graph below shows the ID of each node. For example, we can see the purple node is Author that which is the subclass of Person and name Donald Chamberlin

```
PREFIX Schema:<http://www.semanticweb.org/property/>
CONSTRUCT {
    ?node Schema:writes ?newNodeLTR .
    ?newNodeRTL ?edge ?node .
} WHERE {
    {
        ?node ?edge ?newNodeLTR .
        FILTER(isIRI(?newNodeLTR))
    } UNION {
        ?newNodeRTL ?edge ?node .
        FILTER(isIRI(?newNodeRTL))
    }
} ORDER BY ?edge
```



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.

Number of Classes	**15**
Number of Object Properties	**10**
Number of data Properties	**19**
Number of Instances	**35680**

Table 1: RDF Graph Statistics

C.4 Queries on top of the Ontology

1. Find all the Authors.

(1) Exploit the TBOX

```
PREFIX ont:<http://www.semanticweb.org/ontology/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
select ?AuthorName where {
      ?s rdf:type ont:Author .
    ?s foaf:name ?AuthorName
}
```

(2) Assume no TBOX

```
PREFIX ont:<http://www.semanticweb.org/ontology/>
PREFIX pro:<http://www.semanticweb.org/property/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT distinct ?AuthorName WHERE {
    {?s pro:writes ?o.}
    UNION
    {?s pro:reviews ?o.}
    ?s foaf:name ?AuthorName
}
```

	AuthorName
1	Esteban Zimanyi
2	Daniel Genkin
3	Daniel Gruss
4	Werner Haas
5	Mike Hamburg

2. Find all the properties whose domain is Author.

(1) Exploit the TBOX

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ont:<http://www.semanticweb.org/ontology/>
select * where {
    ?property rdfs:domain ont:Author}
```

(2) Assume no TBOX

```
PREFIX pro:<http://www.semanticweb.org/property/>
PREFIX ont:<http://www.semanticweb.org/ontology/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
select distinct ?property where {
    ?s ?property ?o.
    ?s rdf:type ont:Author.
      filter( strstarts( str(?property), str(pro:) ) )
}
```

	property
1	:property/writes
2	:property/reviews

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

(1) Exploit the TBOX

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ont:<http://www.semanticweb.org/ontology/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT distinct ?Property WHERE {
    { ?Property rdfs:domain ont:Conference.}
    UNION
    { ?Property rdfs:domain ont:Journal.}
} limit 100
```

(2) Assume no TBOX

```
PREFIX pro:<http://www.semanticweb.org/property/>
PREFIX ont:<http://www.semanticweb.org/ontology/>
select distinct ?Property where {
    {?Property ?p ont:Journal.}
        UNION
    {?Property ?p ont:Conference.}
      filter( strstarts( str(?Property), str(pro:) ) )
}
```

	Property
1	:property/hasReview
2	:property/hasEdition

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

(1) Exploit the TBOX

```
PREFIX ont:<http://www.semanticweb.org/ontology/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
select distinct ?AuthorName ?property ?content where {
    ?property rdfs:domain ont:Author.
    ?authorid ?property ?reviewid.
    ?authorid foaf:name ?AuthorName.
    ?reviewid ?p ?content
    filter( !strstarts( str(?content), str(ont:) ) )
}ORDER BY asc(?AuthorName)
```

Ingrid Renz	:property/writes	Analysis of neural activity from EEG data based on EMD frequency bands.
Ingrid Renz	:property/writes	Insuring individual's privacy from statistical data base users.
Ingrid Renz	:property/writes	Is it my body? Body extraction from uninterpreted sensory data based on the invariance of multiple se nsory attributes.
Ingrid Renz	:property/writes	Skin sheriff: a machine learning solution for detecting explicit images.
Ingrid Renz	:property/reviews	xyzaskhksdh asdfkjhkdsahkhsdka hksadkhkj s
Ipke Wachsmuth	:property/writes	Audit considerations of data bases.

(2) Assume no TBOX

```
PREFIX ont:<http://www.semanticweb.org/ontology/>
PREFIX pro:<http://www.semanticweb.org/property/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
select distinct ?AuthorName ?Creation where {
    ?s rdf:type ont:Author.
     ?s foaf:name ?AuthorName.
    {
     ?s pro:writes ?o.
     ?o foaf:title ?Creation}
        UNION{
      ?s pro:reviews ?o.
    ?o pro:Comment ?Creation.
```

```
    }
}ORDER BY asc(?AuthorName)
```

Ingrid Renz	Analysis of neural activity from EEG data based on EMD frequency bands.
Ingrid Renz	Insuring individual's privacy from statistical data base users.
Ingrid Renz	Is it my body? Body extraction from uninterpreted sensory data based on the invariance of multiple sensory attribute s.
Ingrid Renz	Skin sheriff: a machine learning solution for detecting explicit images.
Ingrid Renz	xyzaskhksdh asdfkjhkdsahkhsdka hksadkhkj s
Ipke Wachsmuth	Audit considerations of data bases.

*Note: The review content here is not really meaningful as we used to randomly generate comments from the last lab section for the review csv.

References

1. https://protegewiki.stanford.edu/wiki/Main_Page
2. http://openrefine.org/download.html
3. http://graphdb.ontotext.com/documentation/free/devhub/custom-graph-views.html
4. https://ec.europa.eu/environment/air/quality/standards.htm