Semantic Web handout including: lecture questions and practical sessions

In this document, you must provide your answers to the questions asked during the course **and** to the questions of the practical sessions; everything in one document.

The questions of the course have been repeated here; **do not delete the questions** but provide your answer to each question just below the question. You can use screenshots when appropriate as an answer to a question.

At the end, you must generate and submit only one final PDF file based on this template.

In questions where you are asked to create, invent or use your own data, make sure they are different from other student's.

First name: **SAFA**

Family name: **EL AZRAK**

Email: **safaelazrak1@gmail.com**

QUESTIONS FROM THE COURSES

Day 01: questions from the course.

Q1.1 Practice XML replace missing parts

```
<archi_book>
 <short_title >Architecture Now</short_title>
 <main_author>Jodidio, Philip</main_author >
 <ID isbn10="3822840912"/>
</archi_book>
```

Q1.2 Provide 10 first lines

Get 10 first lines of the five results for:

http://www.wikidata.org/entity/Q23014205
http://www.wikidata.org/entity/Q23014205.json
http://www.wikidata.org/entity/Q23014205.rdf
http://www.wikidata.org/entity/Q23014205.ttl
http://www.wikidata.org/entity/Q23014205.nt

http://www.wikidata.org/entity/Q23014205

Fabien Gandon (Q23014205)

computer science researcher edit

In more languages
Configure

Language	Label	Description	Also known as
English	Fabien Gandon	computer science researcher	
French	Fabien Gandon	chercheur en informatique	
Spanish	Fabien Gandon	investigador francés	
German	Fabien Gandon	No description defined	

All entered languages

Statements

instance of — human edit

1 reference

+ add value

http://www.wikidata.org/entity/Q23014205.json

{"entities":{"Q23014205":{"pageid":25028548,"ns":0,"title":"Q23014205","lastrevid":1346743353,"modified":"2021-01-23T14:23:53Z","type":"item","id":"Q23014205","labels":{"fr":{"language":"fr","value":"Fabien Gandon"},"en":{"language":"en","value":"Fabien Gandon"},"br":{"language":"br","value":"Fabien Gandon"},"de":{"language":"de","value":"Fabien Gandon"},"af":{"language":"af","value":"Fabien Gandon"},"an":{"language":"an","value":"Fabien Gandon"},"ast":{"language":"ast","value":"Fabien Gandon"},"bar":{"language":"bar","value":"Fabien Gandon"},"bm":{"language":"bm","value":"Fabien Gandon"},"ca":{"language":"ca","value":"Fabien Gandon"},"co":{"language":"co","value":"Fabien Gandon"},"cs":{"language":"cs","value":"Fabien Gandon"},"cy":{"language":"cy","value":"Fabien Gandon"},"da":{"language":"da","value":"Fabien Gandon"},"de-at":{"language":"de-at","value":"Fabien

http://www.wikidata.org/entity/Q23014205.rdf

<?xml version="1.0"?>

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:xsd="http://www.w3.org/2001/XMLSchema#" xmlns:ontolex="http://www.w3.org/ns/lemon/ontolex#" xmlns:dct="http://purl.org/dc/terms/" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:wikibase="http://wikiba.se/ontology#" xmlns:skos="http://www.w3.org/2004/02/skos/core#" xmlns:schema="http://schema.org/" xmlns:cc="http://creativecommons.org/ns#" xmlns:geo="http://www.opengis.net/ont/geosparql#" xmlns:prov="http://www.w3.org/ns/prov#" xmlns:wd="http://www.wikidata.org/entity/" xmlns:data="https://www.wikidata.org/wiki/Special:EntityData/" xmlns:s="http://www.wikidata.org/entity/statement/" xmlns:ref="http://www.wikidata.org/reference/" xmlns:v="http://www.wikidata.org/value/" xmlns:wdt="http://www.wikidata.org/prop/direct/" xmlns:wdtn="http://www.wikidata.org/prop/direct-normalized/" xmlns:p="http://www.wikidata.org/prop/"

http://www.wikidata.org/entity/Q23014205.ttl

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix ontolex: <http://www.w3.org/ns/lemon/ontolex#> .
@prefix dct: <http://purl.org/dc/terms/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix wikibase: <http://wikiba.se/ontology#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix schema: <http://schema.org/> .
@prefix cc: <http://creativecommons.org/ns#> .
@prefix geo: <http://www.opengis.net/ont/geosparql#> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix wd: <http://www.wikidata.org/entity/> .
@prefix data: <https://www.wikidata.org/wiki/Special:EntityData/> .
```

http://www.wikidata.org/entity/Q23014205.nt

<https://www.wikidata.org/wiki/Special:EntityData/Q23014205> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://schema.org/Dataset> .

<https://www.wikidata.org/wiki/Special:EntityData/Q23014205> <http://schema.org/about> <http://www.wikidata.org/entity/Q23014205> .

<https://www.wikidata.org/wiki/Special:EntityData/Q23014205> <http://creativecommons.org/ns#license> <http://creativecommons.org/publicdomain/zero/1.0/> .

<https://www.wikidata.org/wiki/Special:EntityData/Q23014205> <http://schema.org/softwareVersion> "1.0.0" .

<https://www.wikidata.org/wiki/Special:EntityData/Q23014205> <http://schema.org/version> "1346743353"^^<http://www.w3.org/2001/XMLSchema#integer> .

<https://www.wikidata.org/wiki/Special:EntityData/Q23014205> <http://schema.org/dateModified> "2021-01-23T14:23:53Z"^^<http://www.w3.org/2001/XMLSchema#dateTime> .

Q1.3 DBpedia

1. Find “London” on DBpedia.org; e.g. Google: "london site:dbpedia.org“
   make sure you are on the English chapter (dbpedia.org) as there are many others (fr.dbpedia.org, de.dbpedia.org)
2. Find dbp:populationDemonym and give its value
3. Find rdf:type and click on value yago:WikicatCapitalsInEurope
4. Find “Vienna” and get its URI
   (careful: with content negotiation and redirection, the URL of the page you are currently viewing may be different from the URI of the resource it describes)
5. Access to Vienna and find its native name?

1- https://dbpedia.org/page/London
2- Londoner
3- https://dbpedia.org/class/yago/WikicatCapitalsInEurope
4- http://dbpedia.org/resource/Vienna
5- Wien

Q1.4 WHO.IS?

1. contact for inria.fr
2. contact for fabien.info
3. contact for lemonde.fr

1- Institut National de Recherche en Informatique et en Automatique Florian DUFOUR
2- Private
3- REDACTED FOR PRIVACY (DT)

Q1.5 CURL

1. Ten first lines:

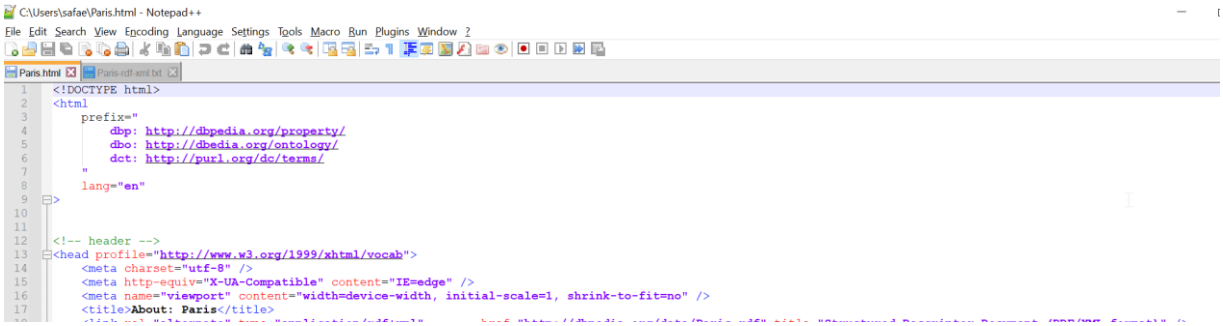
```
curl -o Paris.html -L -H "Accept: text/html" http://dbpedia.org/resource/Paris

curl -o Paris-rdf-xml.txt -L -H "Accept: application/rdf+xml" http://dbpedia.org/resource/Paris
```

2. Ten first lines for HTML and RDF http://ns.inria.fr/fabien.gandon#me
3. Ten first lines for HTML and RDF for ‘Vienna’ on Dbpedia

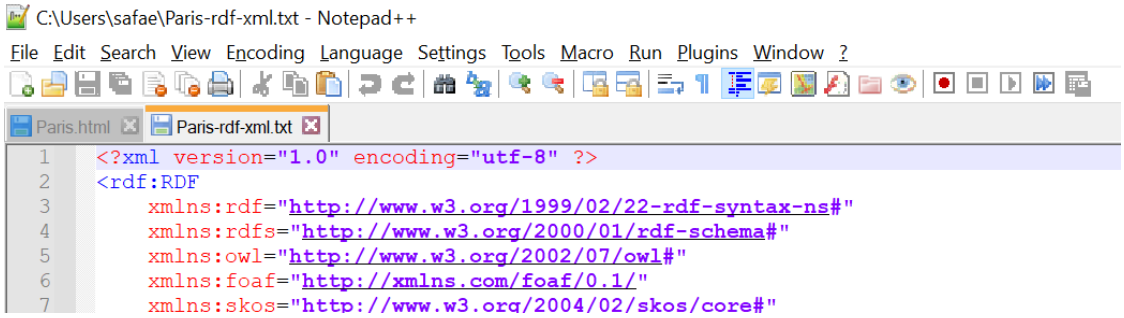
4. Ten first lines for the “URI of the name of Victor Hugo” in the Library of Congress:
   http://id.loc.gov/authorities/names/n79091479
5. Ten first lines for HTML and RDF
   http://purl.uniprot.org/uniprot/P43121
6. What is the topic and format of data obtained with
   curl -o json.txt -L -H "Accept: application/json" https://www.wikidata.org/wiki/Special:EntityData/Q551861
7. What is the topic and format of data obtained with
   curl -o turtle.txt -L -H "Accept: text/turtle" http://dx.doi.org/10.1007/3-540-45741-0_18

1- Paris

a. curl -o Paris.html -L -H "Accept: text/html" http://dbpedia.org/resource/Paris



b. curl -o Paris-rdf-xml.txt -L -H "Accept: application/rdf+xml" http://dbpedia.org/resource/Paris



2- Fabien Gandon

a. curl -o FabienGandon.html -L -H "Accept: text/html" http://ns.inria.fr/fabien.gandon#me

b. curl -o FabienGandon-rdf-xml.txt -L -H "Accept: application/rdf+xml" http://ns.inria.fr/fabien.gandon#me

FabienGandon-rdf-xml.txt

```
<?xml version='1.0' encoding='utf-8' ?>
<rdf:RDF
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
        xmlns:foaf="http://xmlns.com/foaf/0.1/"
        xml:base="http://ns.inria.fr/fabien.gandon">

        <foaf:PersonalProfileDocument rdf:about="">
```

3- Vienna

a. curl -o Vienna.html -L -H "Accept: text/html" http://dbpedia.org/resource/Vienna

Vienna.html

```
<!DOCTYPE html>
<html
    prefix="
        dbp: http://dbpedia.org/property/
        dbo: http://dbedia.org/ontology/
        dct: http://purl.org/dc/terms/
    "
    lang="en"
>


<!-- header -->
<head profile="http://www.w3.org/1999/xhtml/vocab">
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
    <title>About: Vienna</title>
```

b. curl -o Vienna-rdf-xml.txt -L -H "Accept: application/rdf+xml" http://dbpedia.org/resource/Vienna

Vienna-rdf-xml.txt

```
<?xml version="1.0" encoding="utf-8" ?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:foaf="http://xmlns.com/foaf/0.1/"
    xmlns:skos="http://www.w3.org/2004/02/skos/core#"
    xmlns:dbp="http://dbpedia.org/property/"
    xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
    xmlns:dbo="http://dbpedia.org/ontology/"
    xmlns:dct="http://purl.org/dc/terms/"
```

4- Name of Victor Hugo

a. curl -o VictorHugo.html -L -H "Accept: text/html" http://id.loc.gov/authorities/names/n79091479

VictorHugo.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN"
        "http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd">
<html version="XHTML+RDFa 1.0" xmlns="http://www.w3.org/1999/xhtml" xmlns:madsrdf="http://www.loc.gov/mads/rdf/v1#" xmlns:ri="http://id.loc.gov/ontologies
  <head>
    <title>Hugo, Victor, 1802-1885 - LC Linked Data Service: Authorities and Vocabularies | Library of Congress</title>
    <meta name="description" content=" The Linked Data Service provides access to commonly found standards and vocabularies promulgated by the Library of
    <link rel="schema.DC" href="http://purl.org/dc/elements/1.1/"/>
    <link rel="dc.relation.isPartOf" href="//www.loc.gov/" title="Library of Congress"/>
    <meta name="dc.title" content=" LC Linked Data Service: Authorities and Vocabularies (Library of Congress)"/>
    <meta name="dc.contributor" content="The Library of Congress"/>
```

b. curl -o VictorHugo-rdf-xml.txt -L -H "Accept: application/rdf+xml" http://id.loc.gov/authorities/names/n79091479

VictorHugo-rdf-xml.txt

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <madsrdf:PersonalName rdf:about="http://id.loc.gov/authorities/names/n79091479" xmlns:madsrdf="http://www.loc.gov/mads/rdf/v1#">
    <rdf:type rdf:resource="http://www.loc.gov/mads/rdf/v1#Authority"/>
    <madsrdf:authoritativeLabel>Hugo, Victor, 1802-1885</madsrdf:authoritativeLabel>
    <madsrdf:elementList rdf:parseType="Collection">
      <madsrdf:FullNameElement>
    <madsrdf:elementValue>Hugo, Victor,</madsrdf:elementValue>
      </madsrdf:FullNameElement>
      <madsrdf:DateNameElement>
```

5- UniProt

a. curl -o uniprot.html -L -H "Accept: text/html" http://purl.uniprot.org/uniprot/P43121

uniprot.html

```
<!DOCTYPE html SYSTEM "about:legacy-compat">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en"><head><title>MCAM - Cell surface glycoprotein MUC18 precursor - Homo sapiens (Human) -
        var BASE = '/';
    </script><script src="/js-compr.js2021_01" type="text/javascript"></script><script type="text/javascript">
        uniprot.isInternal = false;
        uniprot.namespace = 'uniprot';
        uniprot.releasedate = '2021_01';
    </script><script type="text/javascript">
    ;
    </script><link href="opensearch.xml" title="UniProtKB" type="application/opensearchdescription+xml" rel="search"/><link href="https://www.uniprot
        // variable to store annotation data
        var annotations = [];
        var entryId = 'P43121';
        var isObsolete = false || !true;
```

b. curl -o Uniprot-rdf-xml.txt -L -H "Accept: application/rdf+xml" http://purl.uniprot.org/uniprot/P43121

Uniprot-rdf-xml.txt

```
<?xml version='1.0' encoding='UTF-8'?>
<rdf:RDF xml:base="http://purl.uniprot.org/uniprot/" xmlns="http://purl.uniprot.org/core/" xmlns:ECO="ht
<owl:Ontology rdf:about="http://purl.uniprot.org/uniprot/">
<owl:imports rdf:resource="http://purl.uniprot.org/core/"/>
</owl:Ontology>
<rdf:Description rdf:about="http://purl.uniprot.org/uniprot/P43121">
<rdf:type rdf:resource="http://purl.uniprot.org/core/Protein"/>
```

6- It's a json file. The topic is Xavier Dolan

7- It's turtle file. The topic is Cooperative Information Agents VI

Q1.6 Recall five best practices of linked open data

1- On the web
2- Machine-readable data
3- Non-proprietary data
4- RDF standards
5- Linked RDF

Q1.7 Spotlight demo

Reproduce the demo:

1. Copy a text from Wikipedia (e.g. Muse Band page)
2. Find the DBpedia Spotlight service page
3. Paste the text and run the detection
4. Try with other texts and copy-paste one of the results you get.

3 • Muse are an English rock band from Teignmouth, Devon, formed in 1994. The band consists of Matt Bellamy (lead vocals, guitar, keyboards), Chris Wolstenholme (bass guitar, backing vocals), and Dominic Howard (drums).

Muse released their debut album, Showbiz, in 1999, showcasing Bellamy's falsetto and a melancholic alternative rock style. Their second album, Origin of Symmetry (2001), incorporated wider instrumentation and romantic classical influences, featured their acclaimed cover of "Feeling Good", and earned them a reputation for energetic live performances.[1] Absolution (2003) saw further classical influence, with strings on tracks such as "Butterflies and Hurricanes", and was the first of six consecutive UK number-one albums.

4 • Le Web sémantique, ou toile sémantique1, est une extension du Web standardisée par le World Wide Web Consortium (W3C)2. Ces standards encouragent l'utilisation de formats de données et de protocoles d'échange normés sur le Web, en s'appuyant sur le modèle Resource Description Framework (RDF).

Le Web sémantique est par certains qualifié de Web 3.0.

Here's another text :

Morocco,[a] officially the Kingdom of Morocco,[b] is the westernmost country in the Maghreb region of North Africa. It overlooks the Mediterranean Sea to the north and the Atlantic Ocean to the west, and has land borders with Algeria to the east, and the disputed territory of Western Sahara to the south. Morocco also claims the Spanish exclaves of Ceuta, Melilla and Peñón de Vélez de la Gomera, and several small Spanish-controlled islands off its coast.[14] Morocco spans an area of 710,850 km2 (274,460 sq mi), with a population of 37 million. Its predominant religion is Islam, and its official languages are Arabic and Berber. The Moroccan dialect of Arabic, and French are also widely spoken. Moroccan culture is a vibrant mix of Berber, Arab, and European cultures, and its capital is Rabat, while largest city is Casablanca.[15]

Day 02: questions from the course on RDF.

Q2.0 What is the mathematical structure built by the RDF triples?
(give the type of structure and its definition/explanation)

RDF is a triple model: every piece of knowledge is broken down into (subject , predicate , object)

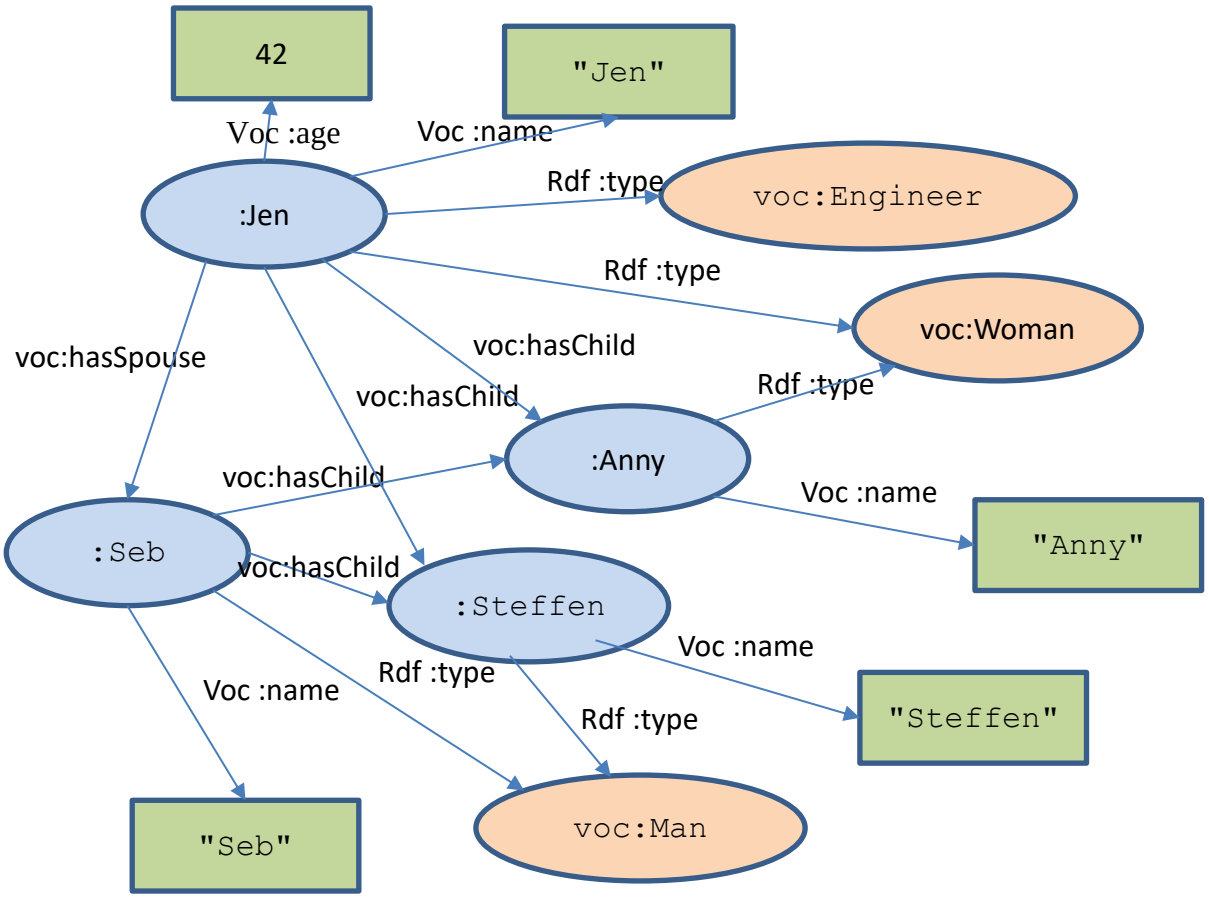
- Subject represent the resource to describe
- Object represent data resource : it is the value of the property
- Predicate represent a type of property applicable to this resource

Q2.1 Fill the blanks

“Jen is an engineer woman, 42-year old, married to Seb who is a man with whom she had two children: Anny who is a woman and Steffen who is a man". For each person we also explicitly specify the name.

To fill the blanks we use the values: `:Seb, :Steffen, voc:name, voc:hasChild, voc:age, voc:hasSpouse, rdf:type, voc:Engineer, voc:Man, "Jen", "Seb", "Anny", "Steffen"`

For each person we also explicitly specify the name



Q2.2 Fill the blanks (RDF/XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF [   <!ENTITY vocab "http://www.unice.fr/voc">
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#"> ]>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:voc="&vocab;#" xml:base="http://www.unice.fr/data">
 <voc:Woman rdf:about="#Jen">
    <voc:name>Jen</voc:name>
    <voc:age rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">42
</voc:age>
    <voc:hasSpouse rdf:resource="#Seb"></voc:hasSpouse>
    <voc:hasChild rdf:resource="#Steffen"></voc:hasChild>
    <voc:hasChild>
      <rdf:Description rdf:about="#Anny">
        <voc:name>Anny</voc:name>
        <rdf:type rdf:resource="&vocab;#Woman"></rdf:type>
      </rdf:Description>
    </voc:hasChild>
    <rdf:type rdf:resource="&vocab;#Engineer"></rdf:type>
 </voc:Woman>
 <voc:Man rdf:about="#Seb">
    <voc:name>Seb</voc:name>
    <voc:hasChild rdf:resource="#Steffen"></voc:hasChild>
    <voc:hasChild rdf:resource="#Anny"></voc:hasChild>
  </voc:Man>
  <voc:Man rdf:about="#Steffen">
    <voc:name>Steffen</voc:name>
  </voc:Man>
</rdf:RDF>
```

Q2.3 Fill the blanks (N3/Turtle)

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix voc: <http://www.unice.fr/voc#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://www.unice.fr/data#Jen> a voc:Engineer , voc:Woman ;
    voc:age "42"^^xsd:string ;
    voc:hasChild
<http://www.unice.fr/data#Anny>,<http://www.unice.fr/data#Steffen>;
    voc:hasSpouse <http://www.unice.fr/data#Seb> ;
    voc:name "Jen" .
<http://www.unice.fr/data#Seb> a voc:Man ;
    voc:hasChild <http://www.unice.fr/data#Anny>,
        <http://www.unice.fr/data#Steffen> ;
    voc:name "Seb" .
<http://www.unice.fr/data#Anny> a voc:Woman ;
    voc:name "Anny" .
<http://www.unice.fr/data#Steffen> a voc:Man ;
     voc:name "Steffen" .
```

Q2.4 Visit me please

Get the RDF data from: http://ns.inria.fr/fabien.gandon#me

1. Get the RDF data from: http://ns.inria.fr/fabien.gandon#me
2. What is the syntax used?
3. Validate it and see the graph:
   http://www.w3.org/RDF/Validator/
4. Translate into Turtle/N3:
   http://rdf-translator.appspot.com/
   http://www.easyrdf.org/converter
5. Visualize it also with:
   http://cltl.nl/visualrdf/
   http://www.easyrdf.org/converter (PNG, SVG)
6. Adapt to your data and do it again

RDF data

```
<?xml version='1.0' encoding='utf-8' ?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:foaf="http://xmlns.com/foaf/0.1/"
          xml:base="http://ns.inria.fr/fabien.gandon">

          <foaf:PersonalProfileDocument rdf:about="">
           <foaf:maker rdf:resource="#me"/>
                <foaf:primaryTopic rdf:resource="#me"/>
          </foaf:PersonalProfileDocument>

          <foaf:Person rdf:ID="me">

           <foaf:name>Fabien Gandon</foaf:name>
                <foaf:title>Dr</foaf:title>
                <foaf:givenname>Fabien</foaf:givenname>
                <foaf:family_name>Gandon</foaf:family_name>
                <foaf:nick>Bafien</foaf:nick>

                <foaf:mbox rdf:resource="mailto:fabien.gandon@inria.fr"/>
                <foaf:homepage rdf:resource="http://fabien.info"/>
                <foaf:depiction rdf:resource="http://www-
sop.inria.fr/members/Fabien.Gandon/common/FabienGandonBackground.jpg"/>
                <foaf:phone rdf:resource="tel:0492387788"/>

                <foaf:workplaceHomepage rdf:resource="http://www.inria.fr/"/>
                <foaf:workInfoHomepage rdf:resource="http://fabien.info"/>
                <foaf:schoolHomepage rdf:resource="http://www.insa-rouen.fr"/>

                <foaf:knows>
                 <foaf:Person>
                  <foaf:name>Olivier Corby</foaf:name>
                        <foaf:mbox rdf:resource="mailto:olivier.corby@inria.fr"/>
                        <rdfs:seeAlso rdf:resource="http://www-sop.inria.fr/members/Olivier.Corby/"/>
                 </foaf:Person>
                </foaf:knows>

                <foaf:knows>
                 <foaf:Person>
```

```
                    <foaf:name>Catherine Faron-Zucker</foaf:name>
                        <foaf:mbox rdf:resource="mailto:faron@polytech.unice.fr"/>
                        <rdfs:seeAlso rdf:resource="http://www.i3s.unice.fr/~faron/"/>
                    </foaf:Person>
                </foaf:knows>

        </foaf:Person>

</rdf:RDF>
```

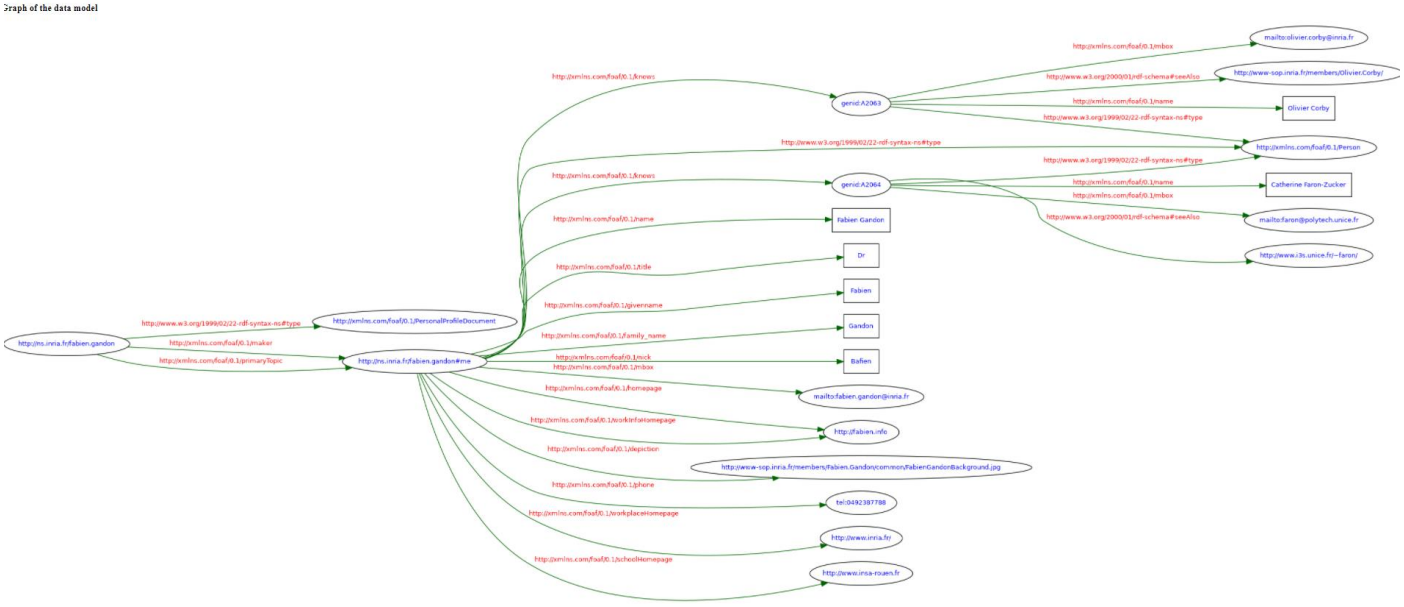
The syntax used is XML

Validate it :

Validation Results

Your RDF document validated successfully.

Get the graph :



Translate into Turtle/N3:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

<http://ns.inria.fr/fabien.gandon> a foaf:PersonalProfileDocument ;
   foaf:maker <http://ns.inria.fr/fabien.gandon#me> ;
   foaf:primaryTopic <http://ns.inria.fr/fabien.gandon#me> .

<http://ns.inria.fr/fabien.gandon#me> a foaf:Person ;
   foaf:depiction <http://www-sop.inria.fr/members/Fabien.Gandon/common/FabienGandonBackground.jpg> ;
   foaf:family_name "Gandon" ;
   foaf:givenname "Fabien" ;
   foaf:homepage <http://fabien.info> ;
   foaf:knows [ a foaf:Person ;
        rdfs:seeAlso <http://www.i3s.unice.fr/~faron/> ;
        foaf:mbox <mailto:faron@polytech.unice.fr> ;
```

```
      foaf:name "Catherine Faron-Zucker" ],
    [ a foaf:Person ;
      rdfs:seeAlso <http://www-sop.inria.fr/members/Olivier.Corby/> ;
      foaf:mbox <mailto:olivier.corby@inria.fr> ;
      foaf:name "Olivier Corby" ] ;
  foaf:mbox <mailto:fabien.gandon@inria.fr> ;
  foaf:name "Fabien Gandon" ;
  foaf:nick "Bafien" ;
  foaf:phone <http://ns.inria.fr/tel:0492387788> ;
  foaf:schoolHomepage <http://www.insa-rouen.fr> ;
  foaf:title "Dr" ;
  foaf:workInfoHomepage <http://fabien.info> ;
  foaf:workplaceHomepage <http://www.inria.fr/> .
```

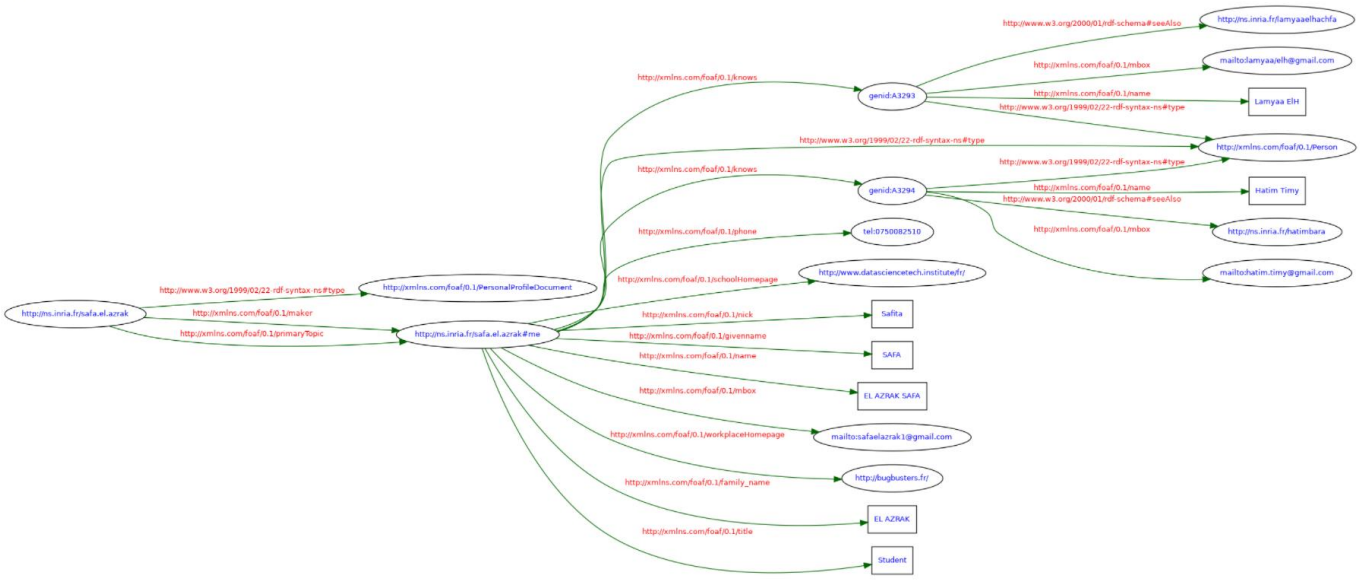
Visualise it :

Adapt to my data and do it again

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
   xmlns:foaf="http://xmlns.com/foaf/0.1/"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"

xml:base="http://ns.inria.fr/safa.el.azrak">

       <foaf:PersonalProfileDocument rdf:about="">
              <foaf:maker rdf:resource="#me"/>
              <foaf:primaryTopic rdf:resource="#me"/>
       </foaf:PersonalProfileDocument>

       <foaf:Person rdf:ID="me">

              <foaf:phone rdf:resource="tel:0750082510"/>
              <foaf:schoolHomepage rdf:resource="http://www.datasciencetech.institute/fr/"/>
              <foaf:nick>Safita</foaf:nick>
              <foaf:givenname>SAFA</foaf:givenname>
              <foaf:name>EL AZRAK SAFA</foaf:name>

              <foaf:mbox rdf:resource="mailto:safaelazrak1@gmail.com"/>
              <foaf:workplaceHomepage rdf:resource="http://bugbusters.fr/"/>
              <foaf:family_name>EL AZRAK</foaf:family_name>
              <foaf:title>Student</foaf:title>

              <foaf:knows>
                     <foaf:Person>
                     <foaf:name>Lamyaa ElH</foaf:name>
                     <rdfs:seeAlso rdf:resource="http://ns.inria.fr/lamyaaelhachfa"/>
```

```
                <foaf:mbox rdf:resource="mailto:lamyaa/elh@gmail.com"/>
                </foaf:Person>
            </foaf:knows>
            <foaf:knows>
                <foaf:Person>
                <foaf:name>Hatim Timy</foaf:name>
                <rdfs:seeAlso rdf:resource="http://ns.inria.fr/hatimbara"/>
                <foaf:mbox rdf:resource="mailto:hatim.timy@gmail.com"/>
                </foaf:Person>
                    </foaf:knows>

    </foaf:Person>
</rdf:RDF>
```

Graph of the data model



Converted to Turtle :

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .


<http://ns.inria.fr/safa.el.azrak> a foaf:PersonalProfileDocument ;

    foaf:maker <http://ns.inria.fr/safa.el.azrak#me> ;

    foaf:primaryTopic <http://ns.inria.fr/safa.el.azrak#me> .


<http://ns.inria.fr/safa.el.azrak#me> a foaf:Person ;

    foaf:family_name "EL AZRAK" ;

    foaf:givenname "SAFA" ;

    foaf:knows [ a foaf:Person ;

            rdfs:seeAlso <http://ns.inria.fr/hatimbara> ;

            foaf:mbox <mailto:hatim.timy@gmail.com> ;

            foaf:name "Hatim Timy" ],
```

```
        [ a foaf:Person ;
            rdfs:seeAlso <http://ns.inria.fr/lamyaaelhachfa> ;
            foaf:mbox <mailto:lamyaa/elh@gmail.com> ;
            foaf:name "Lamyaa ElH" ] ;
    foaf:mbox <mailto:safaelazrak1@gmail.com> ;
    foaf:name "EL AZRAK SAFA" ;
    foaf:nick "Safita" ;
    foaf:phone <http://ns.inria.fr/tel:0750082510> ;
    foaf:schoolHomepage <http://www.datasciencetech.institute/fr/> ;
    foaf:title "Student" ;
    foaf:workplaceHomepage <http://bugbusters.fr/> .
```

Q2.5 what is the meaning of this RDF? What is this description saying?

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:exs="http://example.org/schema#">
 <rdf:Description rdf:about="http://example.org/doc.html">
  <rdf:type rdf:resource="http://example.org/schema#Report"/>
  <exs:theme rdf:resource="http://example.org#Music"/>
  <exs:theme rdf:resource="http://example.org#Danse"/>
  <exs:nbPages
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">73</exs:nbPages>
 </rdf:Description>
</rdf:RDF>
```

I have an rdf file, I declared 2 namespace (rdf and exs)

The subject of my description is doc.html

The document is of type report, it has two topics : music and danse

It has 73 pages and 73 is an integer.

Q2.6 Visit to Victor Hugo

1. See HTML data from: http://id.loc.gov/authorities/names/n79091479.html
2. Get RDF data from: http://id.loc.gov/authorities/names/n79091479.rdf
3. What is the syntax?
4. Translate into Turtle/N3: http://rdf-translator.appspot.com/
5. Any remark about the values of the properties of Victor Hugo?

It is the uri that represents the name of victor hugo in the Library of Congress

The RDF is in XML format.

We find in it a number of translations, in a number of languages

If we look at the values, they don't specify the languages (for each of the literal).

Screenshot for an example :

```
    [ a madsrdf:PersonalName,
            madsrdf:Variant ;
        madsrdf:elementList ( [ a madsrdf:FullNameElement ;
                    madsrdf:elementValue "هوجو، فيكتور" ] [ a madsrdf:DateNameElement ;
                    madsrdf:elementValue "1802-1885" ] ) ;
        madsrdf:variantLabel "هوجو، فيكتور 1885-1802" ],
```

Q2.7 What is the syntax of the following RDF statement? What does it mean?

```
@prefix dcterms: <http://purl.org/dc/terms/>.
GRAPH  <http://inria.fr/data/algebra>
{
  <http://inria.fr/rr/doc.html>
    dcterms:subject
      <http://data.bnf.fr/ark:/12148/cb121105993> .
}
```

The syntax is TriG because we use "graph"

We define a prefix and then we say that we have a graph named http://inria.fr/data/algebra

Inside the graph, we have the triple : the document doc.html have the subject <http://data.bnf.fr/ark:/12148/cb121105993>

Q2.8 Visit Leukocyte surface antigen CD53

1. See HTML data from: http://www.uniprot.org/uniprot/Q61451
2. Get RDF data from: http://www.uniprot.org/uniprot/Q61451.rdf
3. What is the syntax?
4. Translate into Turtle/N3: http://rdf-translator.appspot.com/
5. Any remark about the structure of the data?

It's about Leukocyte surface antigen CD53; the syntax is RDF/ XML

They use rdf:Statement, rdf:Subject … each time
They do traceability on all the biological data, they need to check who made experiment :

Ex :

```
<http://purl.uniprot.org/uniprot/#_109368FB62889A2E_rdfs.comment_4E095ED51619D51E> a rdf:Statement ;
    up:attribution <http://purl.uniprot.org/uniprot/Q61451#attribution-609619A7DF3E53CE5E48EDEF8B8F9DD1
    rdf:object "Interacts with SCIMP." ;
    rdf:predicate rdfs:comment ;
    rdf:subject <http://purl.uniprot.org/uniprot/Q61451#SIP236C2DA8C2F13FDB> .
```

Day 03: questions from the course on SPARQL.

Q3.1 Test SPARQL online

Connect to: https://corese.inria.fr/srv/tutorial/sparql

Answers to the query:

```
prefix v: <http://www.inria.fr/2015/humans#>
select * where {  ?x a v:Person . }
```

We get the URIs of the resources of type Person :

	x
1	<http://www.inria.fr/2015/humans-instances#John>
2	<http://www.inria.fr/2015/humans-instances#Sophie>
3	<http://www.inria.fr/2015/humans-instances#Mark>
4	<http://www.inria.fr/2015/humans-instances#Eve>
5	<http://www.inria.fr/2015/humans-instances#David>
6	<http://www.inria.fr/2015/humans-instances#Laura>
7	<http://www.inria.fr/2015/humans-instances#William>
8	<http://www.inria.fr/2015/humans-instances#Karl>

Q3.2 Test SPARQL online

Connect to

http://dbpedia.org/snorql/ or

http://fr.dbpedia.org/sparql or …

http://wiki.dbpedia.org/Internationalization/Chapters

Answers to the query:

```
SELECT * WHERE {
  ?x rdfs:label "Paris"@fr .
  ?x ?p ?v .
}
LIMIT 10
```

Using http://dbpedia.org/snorql/

x	p	v
http://fr.dbpedia.org/resource/Catégorie:Paris	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2004/02/skos/core#Concept
http://fr.dbpedia.org/resource/Paris	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Thing
http://fr.dbpedia.org/resource/Paris	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://schema.org/Place
http://fr.dbpedia.org/resource/Paris	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://dbpedia.org/ontology/Place
http://fr.dbpedia.org/resource/Paris	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://dbpedia.org/ontology/PopulatedPlace
http://fr.dbpedia.org/resource/Paris	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://dbpedia.org/ontology/Settlement
http://fr.dbpedia.org/resource/Paris	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2003/01/geo/wgs84_pos#SpatialThing
http://fr.dbpedia.org/resource/Paris	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.wikidata.org/entity/Q486972
http://fr.dbpedia.org/resource/Paris	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://dbpedia.org/ontology/Location
http://fr.dbpedia.org/resource/Paris	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2004/02/skos/core#Concept

Using http://fr.dbpedia.org/sparql

SPARQL results:

x	p	v
:Paris	rdf:type	owl:Thing
:Paris	rdf:type	dbpedia:ontology/Place
:Paris	rdf:type	dbpedia:ontology/Location
:Paris	rdf:type	<http://www.w3.org/2003/01/geo/wgs84_pos#SpatialThing>
:Paris	rdf:type	dbpedia:class/yago/Object100002684
:Paris	rdf:type	dbpedia:class/yago/PhysicalEntity100001930
:Paris	rdf:type	dbpedia:class/yago/Prefecture108626947
:Paris	rdf:type	dbpedia:class/yago/Region108630985
:Paris	rdf:type	dbpedia:class/yago/Seat108647945
:Paris	rdf:type	dbpedia:class/yago/Site108651247

This query selects resources that has for label Paris in French and I also want to select its property and its value (I will select again the label because it's a property also). It will retrieve all the resources in the LIMIT of 10 results.

Q3.3 Test SPARQL online

Connect to:

https://query.wikidata.org/

What does this query retrieve?

```
SELECT distinct ?p ?n WHERE
{     wd:Q30 p:P6 [ ps:P6 ?p ].
      ?p rdfs:label ?n .
      FILTER (lang(?n)="en") }
```

Discover wd:Q30 using the namespace attached to wd:

PREFIX wd: <http://www.wikidata.org/entity/>

Discover p:P6 using the namespace attached to p:

PREFIX p: <http://www.wikidata.org/prop/>

Find q-name of the property "given name"

https://www.wikidata.org/wiki/Wikidata:List_of_properties

We are looking for the name p:P6 that's the value of the property Head of Government, attached to the USA (Q:30)

So we are looking at resources in USA, that are head of government, we are getting the names of theses resources, and we look for the name in English.

Given name : P735

Q3.4 SPARQL query to return 20 persons at most (use type foaf:Person)

```
Select *
WHERE {
?x a foaf:Person }
LIMIT 20
```

Q3.5 SPARQL query to return 20 persons (at most), after the 10th result i.e. from 11th to 30th

```
Select *
WHERE {
?x a foaf:Person }
LIMIT 20
OFFSET 10
```

Q3.6 You have two properties: c:name and c:age

1.Find the age of resources whose name is 'Fabien'

2.Find the name of resources whose age is less than 50

3.Find property values of resources whose name is 'Fabien' and whose age is less than 50

4.Find other names of resources whose name is 'Fabien'

5.Find resources which have two different properties with the same value

6.Find resources which have the same property with two different values

```
1- SELECT ?age WHERE {?x c:name "Fabien"; c:age ?age}
2- SELECT ?name  WHERE {?x c:name ?name; c:age ?age.  FILTER( ?age  < 50) }
3- SELECT ?p ?y WHERE {?x c:name 'Fabien'; c:age ?age; ?p ?y. FILTER (?age < 50)}
4- SELECT ?name WHERE {?x c:name "Fabien", ?name . FILTER (?name != 'Fabien') }
5- SELECT ?x WHERE {?x ?p1 ?v; ?p2 ?v. FILTER (?p1 != ?p2)}
6- SELECT ?x WHERE {?x ?p ?v1, ?v2. FILTER (?v1 != ?v2)}
```

Q3.7 Could this query return `ex:a c:memberOf ex:b` and why ?

```
select * where {
  ?x c:memberOf ?org .
  minus { ex:a c:memberOf ex:b }
}
```

The minus is ignored because there is no shared variable, so Yes this query could return it.

Q3.8 get the members of organizations (c:memberOf) but remove the resources author of a document (c:author) by using 'not exists'

Select ?x where { ?x c:memberOf ?org . Filter(not exists{?x c:author ?doc}}

Q3.9 what is retrieving this query ?

```
prefix ex: <http://example.org/>
select ?x (count(?doc) as ?c)
where {  ?x ex:author ?doc  }
group by ?x
order by desc(count(?doc))
```

count the number of document per author ordering from the biggest number of documents to the smallest (descending order)

Q3.10 What expression should we use to find the ?x related to ?y by paths composed of properties foaf:knows and/or rdfs: seeAlso?

- ?x (foaf:knows | rdfs:seeAlso)+ ?y
- ?x foaf:knows+ | rdfs:seeAlso+ ?y
- ?x (foaf:knows / rdfs:seeAlso)+ ?y

The first one : Repeat one or several times (either foaf:knows or rdfs:seeAlso)

?x (foaf:knows | rdfs:seeAlso)+ ?y

Q3.11 what is this query retrieving?

```
prefix foaf: <http://xmlns.com/foaf/0.1/>
select ?x (if (bound(?n), ?n, "John Doe") as ?m)
where {
   ?x foaf:knows ?y
   optional { ?y foaf:name ?n }
}
```

Select x and give either the name of the friend (if n is bound) or john doe if not known (n unbound)

Q3.12 what is this query retrieving?

```
prefix ex:  <http://example.org/>
select ?x (avg(?a) as ?b)
where {
  ?x ex:knows ?y .
  ?y ex:age ?a
}
group by ?x
```

For someone, we look at the person he knows, get the age , group by x and average the age of the friends :

⇨ display all persons and the average age of their friends

Q3.13 You have two properties: c:name and c:study and the resources c:Informatics and c:Mathematics

1. Find resources that study informatics or mathematics
2. In addition return the name of the resource if it has a name
3. In addition return the graph where the name is given

1- SELECT * where {?x c:study c:informatics } union {?x c:study c:mathematics}}
2- SELECT * where {{?x c:study c:informatics } union {?x c:study c:mathematics} optional {?x c:name ?name}}
3- SELECT * where {?x c:study c:informatics } union {?x c:study c:mathematics} optional {graph ?g {?x c:name ?name}} }

Q3.14 On which graph(s) is calculated ?x ?p ?y

On which graph(s) is calculated graph ?g { ?y ?q ?z }

```
prefix ex:  <http://example.org/>
select *
from ex:g1
from named ex:g2
where {
   ?x ?p ?y .
   graph ?g { ?y ?q ?z}  }
```

?x ?p ?y ➔ graph g1

?y ?q ?z ➔ graph g2

Q3.15 Write a query to change foaf:name into rdfs:label

DELETE {?x foaf:name ?n}

INSERT { ?x rdfs:label ?n}

WHERE {?x foaf:name ?n}

Q3.16 what is this query performing?

```
prefix ex: <http://example.org/>
delete { ?x ex:age ?a }
insert { ?x ex:age ?i }
where  {
  select ?x (xsd:integer(?a) as ?i)
  where {
    ?x ex:age ?a
    filter(datatype(?a) = xsd:string)
  }
}
```

Select the resources that have an age, check the datatype of age: where the datatype is in string and convert it to integer

Q3.17 Which clauses could you use to obtained results as RDF triples following a specific pattern?

- SELECT … WHERE {…} …
- CONSTRUCT { } WHERE {…} …
- DESCRIBE <…> DESCRIBE … {…}
- ASK {…}
- DELETE { … } INSERT { … } WHERE {…} …

CONSTRUCT { } WHERE {…} …

Q3.18 What is the difference between these two queries?

```
prefix ex: <http://example.org/>
insert { ?x a ex:Parent }
where  { ?x ex:hasChild  ?y }


prefix ex: <http://example.org/>
construct { ?x a ex:Parent }
where  { ?x ex:hasChild  ?y }
```

One inserts data into existing RDF, one gives a new RDF as result

Day 04: questions from the course on RDFS.

Q4.1 Choose among the following assertions one or more you consider to be true:

- ☐ an ontology is necessarily formalized in first-order logic
- ☐ an ontology may allow inferences on data that uses it
- ☐ conceptual graphs can represent an ontology
- ☐ a shared ontology promotes interoperability
- ☐ description logics can represent an ontology

➔All the answers except the first one

Q4.2 RDFS contains primitives to (several answers possible)…

- ☐ describe classes of resources
- ☐ describe formulas of calculation for values of properties
- ☐ describe types of properties of resources
- ☐ document definitions in natural language
- ☐ sign and authenticate the authors of the definitions of classes and properties

➔first, third and fourth answers are correct

Q4.3. What is defined and derived from these definitions?

```
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#>
@prefix : <http://inria.fr/devices#>
 :Phone rdfs:subClassOf  :Device .
 :Computer rdfs:subClassOf  :Device .
 :Smartphone rdfs:subClassOf :Computer .
 :Smartphone rdfs:subClassOf :Phone .
```

➔We defined rdfs prefix and :

- Class Device has for subclass Phone and Computer
- Computer has for subclass Smartphone
- So Smartphone is also a subclass of Device (by transitivity).

Q4.4. What is defined and derived from these definitions?

```
@prefix rdfs:  < http://www.w3.org/2000/01/rdf-schema# >
@prefix : <http://inria.fr/member#>
:employeeOf rdfs:subPropertyOf  :proRelationWith .
:hasControlOver rdfs:subPropertyOf :proRelationWith .
:isShareholderOf rdfs:subPropertyOf :hasControlOver .
:isCEOof rdfs:subPropertyOf :employeeOf, :hasControlOver .
```

➔we define the rdfs prefix and :

- proRelation is a superproperty of employeeof and hascontrolover
- hascontrolover has for subprop isshareholderof and isCEOof
- isCEOof is a subprop of employeeof

Q4.5. What can be said about the types of the resources that will be linked by the properties defined below?

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
@prefix : <http://inria.fr/humans#>
:driverOf  rdfs:subPropertyOf  :isControling .
:piloteOf  rdfs:subPropertyOf  :isControling .
:isControling  rdfs:domain  :Human ;  rdfs:range :Object .
:driverOf rdfs:range  :Car .
:piloteOf rdfs:domain  :Adult ; rdfs:range  :Plane .
```

➔we define the rdfs prefix and :

- driverOf and piloteOf are subproperties of isControlling
- Is controlling : goes from a resource of type human to a resource of type object
- DriverOf goes to a car (it has for range Car)
- PiloteOf : goes from an adult to a plane (it has for domain Adult and for range Plane)

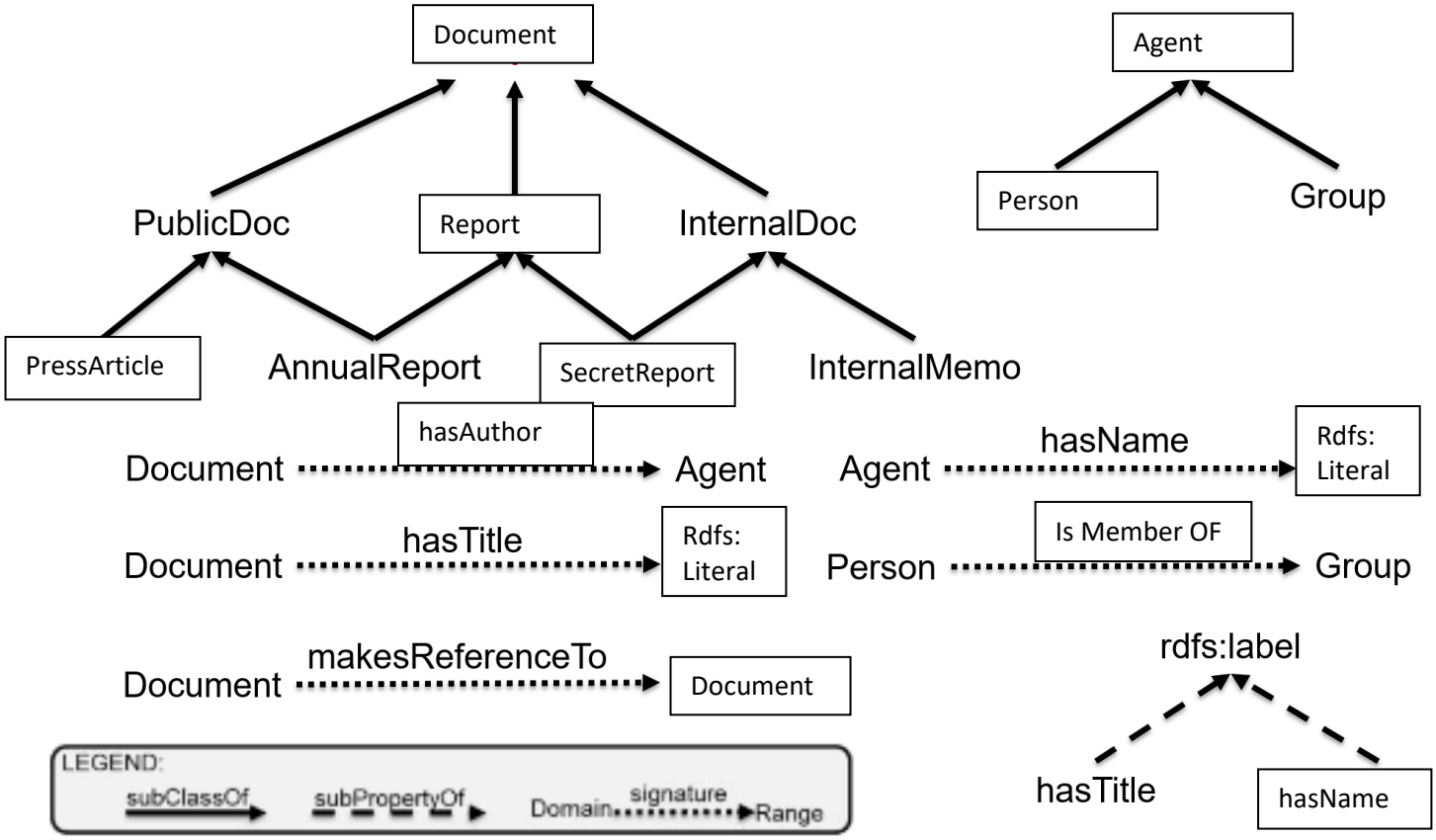
Q4.6. What could we add to this schema (several answers are possible)?

```
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#>
@base  <http://inria.fr/2005/humans.rdfs>
<p1>  a rdf:Property ;  rdfs:label  "age"@fr .
<c1>  a rdfs:Class; rdfs:comment "un être humain"@fr .
```

- ☐ <p1> rdfs:label "prénom"@fr .
- ☐ <c1> rdfs:comment "a human being"@fr .
- ☐ <c1> rdfs:label "personne"@fr .
- ☐ <p1> rdfs:label "age"@en .
- ☐ <c1> rdfs:label "woman"@en .
- ☐ <c1> rdfs:label "persona"@es .

Q4.7. (a) Fill the blanks with: `Document, PublicDoc, PressArticle, Report, AnnualReport, InternalDoc, SecretReport, InternalMemo, Agent, Person, Group, hasTitle, hasAuthor, makesReferenceTo, hasName, isMemberOf` + rdf / rdfs primitives.

(b) Write it in RDFS and validate the RDF.



Day 04: questions from the course on OWL.

Q5.1 What can we deduce?

```
ex:Man owl:intersectionOf (ex:Male ex:Human) .
ex:Woman owl:intersectionOf (ex:Female ex:Human) .
ex:Human owl:unionOf (ex:Man ex:Woman) .
ex:Jane a ex:Human .
ex:John a ex:Man .
ex:James a ex:Male .
ex:Jane a ex:Female .
```

John is a Man, he is also Male and human

Jane is Female and Human so she is a Woman

James is a Male

Q5.2 What are we defining and inferring?

```
@prefix ex: <http://example.org/>

ex:GrandFather rdfs:subClassOf [
   a owl:Class ;
   owl:intersectionOf ( ex:Parent ex:Man )
] .

ex:Jim a ex:Man, ex:Parent .
ex:Jack a ex:GrandFather .
```

Jack is a grandfather so he is a Man and a Parent

Jim is a Man and a Parent

Q5.3 What can we deduce?

```
ex:hasSpouse a owl:SymmetricProperty .
ex:hasChild  owl:inverseOf ex:hasParent .
ex:hasParent rdfs:subPropertyOf ex:hasAncestor .
ex:hasAncestor a owl:TransitiveProperty .
ex:Jim  ex:hasChild  ex:Jane .
ex:Jane ex:hasSpouse ex:John .
ex:Jim  ex:hasParent ex:James .
```

Jane hasParent Jim
John hasSpouse Jane
James hasChild Jim
Jim has Ancestor James
Jane has Ancestor Jim and James

Q5.4 What can we deduce?

```
ex:Human owl:equivalentClass foaf:Person .
foaf:name owl:equivalentProperty ex:name .
ex:JimmyPage a ex:Human ;
             owl:sameAs ex:JamesPatrickPage .
ex:JimmyHendrix owl:differentFrom ex:JimmyPage .
```

Foaf:Person is equivalentClass as ex:Human
Ex:name is also equivalentProperty as foaf:name
Ex:JimmyPage is a human also person (foaf:person) and it's the same as ex:JamesPatrickPage and different from ex:JimmyHendrix

Q5.5 What are we defining and inferring?

```
ex:UnluckyPerson owl:equivalentClass [
  a owl:Class ;
  owl:intersectionOf (
    ex:Person
    [ a owl:Class ; owl:complementOf ex:Lucky ]
  )
] .
```

An unlucky person is the intersection of person and complement of lucky.

Q5.6 What can we deduce?

```
ex:Human rdfs:subClassOf
  [ a owl:Restriction ;
    owl:onProperty ex:hasParent ;
    owl:allValuesFrom ex:Human ] .
ex:Tom a ex:Human .
ex:Tom ex:hasParent ex:James, ex:Jane.
```

Tom is human. James and Jane are parents so they become human

Q5.7 What are we defining and inferring?

```
@prefix ex: <http://example.org/>
ex:PersonList rdfs:subClassOf
  [
    a owl:Restriction ;
    owl:onProperty rdf:first ;
    owl:allValuesFrom ex:Person
  ] ,  [
    a owl:Restriction ;
    owl:onProperty rdf:rest ;
    owl:allValuesFrom ex:PersonList
  ] .

ex:value rdfs:range ex:PersonList .
ex:abc ex:value (ex:a ex:b ex:c) .
```

OnProperty first must be a person
onProperty of rest must be personlist
(a, b, c) become a personlist
A becomes a person, (b c) become a person list
B becomes a person, (c) becomes a person list. And C becomes a person

Q5.8 What are we defining and inferring?

```
@prefix ex: <http://example.org/>
ex:Human rdfs:subClassOf [
  owl:intersectionOf (
    [
      a owl:Restriction ;
        owl:onProperty ex:hasBiologicalFather ;
        owl:maxCardinality 1
    ] , [
      a owl:Restriction ;
        owl:onProperty ex:hasBiologicalMother ;
        owl:maxCardinality 1
    ] )
] .
ex:Jane a ex:Human ;
        ex:hasBiologicalFather ex:James , ex:Jhon .
```

A human can only one biological father, and one biological mother
The system would deduce that James and John are the same person

Day 05: questions from the course on Vocabularies.

Q6.1 What do you think of the annotation?

```
@prefix skos: <http://www.w3.org/2004/02/skos/core#>.
<#B-A-Ba> a skos:Concept ;
  skos:prefLabel    "B.A.-BA"@en , "b.a.-ba"@en ;
  skos:altLabel     "B-A-BA"@en ,  "b-a-ba"@en ;
  skos:hiddenLabel "BABA"@en , "baba"@en .
```

Someone gave 2 preferred labels in English and that's not possible .

Q6.2 practice:

1. Using the site prefix.cc find back the namespace usually associated to the SKOS prefix
2. Access the URL of the namespace and find the RDF source file defining the SKOS vocabulary
3. Find the definition of the property `narrowMatch` and give all the relations it has with other properties

http://www.w3.org/2004/02/skos/core#

Definition : mapping properties

Super-properties: skos:mappingRelation, skos:narrower

Inverse of: skos:broadMatch

skos:narrowMatch	
URI:	http://www.w3.org/2004/02/skos/core#narrowMatch
Definition:	Section 10. Mapping Properties
Label:	*has narrower match*
Super-properties:	skos:mappingRelation skos:narrower
Inverse of:	skos:broadMatch

Q6.3 practice:

1. Find and open the source file of Dublin Core Terms: https://dublincore.org/schemas/rdfs/
   Look at the definition of the class `FileFormat` and find the class it inherits from.
2. Choose your preferred book on Amazon, Fnac, etc. and describe it in an RDF annotation using as many DC primitives as necessary.
3. Add the most restrictive CC license to your preferred book ; is this license appropriate?

⇨ It's a SubclassOf MediaType

@prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>.

@prefix dc:<http://purl.org/dc/elements/1.1/>.

```
@prefix dcterms:<http://purl.org/dc/terms/>.
@prefix cc:< https://creativecommons.org/licenses/by-nc-nd/4.0/ >.

<https://www.amazon.fr/Rich-Dad-Poor-Teach-Middle/dp/1612680194>

dc:creator <https://www.richdad.com/about/robert-t-kiyosaki>;
dc:title "Rich Dad Poor Dad" ;
cc:license [a cc:License;
cc:permits cc:DerivativeWorks, cc:Distribution;
cc:requires cc:Attribution, cc:Notice, cc:ShareAlike];
dc:language "en";
dc:subject "Finance, Budget";
dc:date "2017 05 01" ;
dc:publisher <https://www.hachettebookgroup.com/>;
dc:format "text /html";
dc:type dcterms:Text.
```

Q6.4 practice:

1. Get the source of the FoaF schema: http://xmlns.com/foaf/spec/index.rdf
2. Find the property `weblog`
3. What are the types of this property?
4. Does it inherit from other properties?
5. What is its signature?

- InverseFunctionalProperty in owl ontology (means if 2 resources have the same result, they become the same resource)
- It's an Object property
- (subPropertyOf) Inherit from home page
- Signature ➔ Domain: Agent, Range: Document

Q6.5 practice:

1. Find the FOAF-a-Matic web page
2. Use this tool to generate your FOAF profile in RDF/XML
3. Translate it into Turtle, save and give the result in your answers.
4. Add five specific relationships to your FOAF file using RELATIONSHIPS: http://purl.org/vocab/relationship/

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

@prefix rel: <http://vocab.org/relationship/> .

<http://ns.inria.fr/safa.el.azrak> a foaf:PersonalProfileDocument ;

    foaf:maker <http://ns.inria.fr/safa.el.azrak#me> ;

    foaf:primaryTopic <http://ns.inria.fr/safa.el.azrak#me> .

<http://ns.inria.fr/safa.el.azrak#me> a foaf:Person ;

    foaf:family_name "EL AZRAK" ;

    foaf:givenname "SAFA" ;

    foaf:knows [ a foaf:Person ;

            rdfs:seeAlso <http://ns.inria.fr/hatimbara> ;

            foaf:mbox <mailto:hatim.timy@gmail.com> ;

            foaf:name "Hatim Timy" ],

        [ a foaf:Person ;

            rdfs:seeAlso <http://ns.inria.fr/lamyaaelhachfa> ;

            foaf:mbox <mailto:lamyaa/elh@gmail.com> ;

            foaf:name "Lamyaa ElH" ],

        [ a foaf:Person ;

            rel:childOf <http://ns.inria.fr/soumiachahid> ;

            rel:closeFriendOf <http://ns.inria.fr/hatimbara> ;

            rel:colleagueOf <http://ns.inria.fr/loicpequinot> ] ;

    foaf:mbox <mailto:safaelazrak1@gmail.com> ;

    foaf:name "EL AZRAK SAFA" ;

    foaf:nick "Safita" ;

    foaf:phone <http://ns.inria.fr/tel:0750082510> ;

    foaf:schoolHomepage <http://www.datasciencetech.institute/fr/> ;

    foaf:title "Student" ;

    foaf:workplaceHomepage <http://bugbusters.fr/> .
```

Q6.6 What does this mean?

```
:BioRDF2DBLP a void:Linkset;
               void:target :BioRDF;
               void:target :DBLP;
               void:linkPredicate skos:exactMatch;
               void:triples 8936 .
```

:BioRDF2DBLP is a link between dataset (BioRDF, DBLP). The predicate for the link is skos:exactMatch.

It contains 8936 triples.

Q6.7 practice:

1. Connect to the Void Store SPARQL endpoint: http://void.rkbexplorer.com/sparql/
2. What is the meaning of the default SPARQL query in the interface, run it and look at the results.
3. Write a SPARQL query to find the dataset that has for label "DBpedia-fr" and all its properties.

Default SPARQL query ➔ Find all dataset that have SPARQL endpoints

```
PREFIX rdf:      <http://www.w3.org/1999/02/22-rdf-
syntax-ns#>
PREFIX rdfs:     <http://www.w3.org/2000/01/rdf-
schema#>
PREFIX owl:      <http://www.w3.org/2002/07/owl#>
PREFIX xsd:      <http://www.w3.org/2001/XMLSchema#>
PREFIX foaf:     <http://xmlns.com/foaf/0.1/>
PREFIX dcterms:  <http://purl.org/dc/terms/>
PREFIX scovo:    <http://purl.org/NET/scovo#>
PREFIX void:     <http://rdfs.org/ns/void#>
PREFIX akt:
<http://www.aktors.org/ontology/portal#>


SELECT DISTINCT ?endpoint WHERE { ?ds a void:Dataset
. ?ds void:sparqlEndpoint ?endpoint }
```

Query to find the dataset that has for label “DB-Pedia-fr”:

Select * where{?x rdfs:label "DBpedia-fr".?x ?y ?z}

Q6.8 What does this mean?

```
ex:plot prov:used ex:stats1998 .
ex:bar-chart prov:wasGeneratedBy ex:plot .
ex:stats1998 a dcat:Distribution ;
             dcat:format [ rdfs:label "CSV" ] ;
             dcat:mediaType "text/csv" .
```

Plot is using data stats1998. Bar-chart is generated by plot. Stats1998 is a distribution in format csv

Q6.9 What does this mean?

```
@prefix dcat: <http://www.w3.org/ns/dcat#> .
@prefix void: <http://rdfs.org/ns/void#> .
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix dct:  <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix : <http://inria.fr/data#> .

:db-employ
  a dcat:Distribution ;
  dcat:downloadURL <http://wimmics.inria.fr/docs/employ-2014.sql> ;
  dct:title "SQL Dump of the employees" ;
  dct:spatial <http://www.geonames.org/6640252> ;
  dct:issued "2015-01-12"^^xsd:date ;
  dct:temporal <http://reference.data.gov.uk/id/year/2014> ;
  dct:publisher <http://inria.fr> ;
  dcat:mediaType "application/sql" ;
  dcat:format [ rdfs:label "SQL" ] ;
  dct:language <http://id.loc.gov/vocabulary/iso639-1/fr> ;
  dcat:byteSize "38729"^^xsd:decimal .

:R2RTransform12 prov:used :db-employ ;
                prov:used :R2R-employ-mapping ;
                prov:used <http://xmlns.com/foaf/0.1/> .

:FoaFDump a void:Dataset;
          void:feature <http://www.w3.org/ns/formats/RDF_XML>;
          void:dataDump <http://wimmics.inria.fr/docs/employ-2014.rdf>;
          void:exampleResource <http://ns.inria.fr/fabien.gandon#me> ;
          void:vocabulary <http://xmlns.com/foaf/0.1/>;
          void:triples 12875;
          dct:title "RDF Dump of the employees" ;
          prov:wasGeneratedBy :R2RTransform12 ;
          prov:generatedAtTime "2015-01-14T11:38:27"^^xsd:dateTime ;
          prov:wasDerivedFrom :db-employ .
```

:db-employ is a a dcat:distribution can be found http://wimmics.inria.fr/docs/employ-2014.sql. It has a title, a Spatial location, issue date, temporal, publisher, SQL format, language is in fr, and the byteSize of it.

FoafDump is a XML dataset, it can be found <http://ns.inria.fr/fabien.gandon#me> use the vocabulary foaf, number of triples is 12875, we have the title. It was generated by R2RTransform12 at 2015-01-14T11:38:27. It was derived from db-employ.

Which we know that :R2RTransform12 is using db:employ, R2R-employee-mapping and http://xmlns.com/foaf/0.1/ as input.

Q6.10 practice:

1. Connect to the LOV directory: https://lov.linkeddata.es/
2. Search for schemas talking about “music artist”.
3. What is the top ontology you find?
4. What is its version number?
5. Is it reused by other ontologies?
6. How many classes and properties does it have?
7. What expressivity does it use? (RDFS, OWL)

The music ontology is used http://purl.org/ontology/mo/

The version number is : Revision: 2.1.5

It is reused by other ontologies as there are many outgoing links (playlist ontology, theater ontology, open voal ontology)

It has 54 Classes and 153 properties

It uses RDF, RDFS, OWL

Day 05: questions from the course on other data formats.

Q7.1 What are the triples produced with this mapping and this table?

```
:My_Table rdf:type rr:TriplesMap ;
  rr:subjectMap [  rr:template
"https://www.ietf.org/rfc/rfc{NUM}.txt"; ];
  rr:predicateObjectMap [
    rr:predicateMap [  rr:predicate dc:title  ];
    rr:objectMap [     rr:column "ttl"     ]
  ].
```

ID	NUM	ttl
87	2616	Hypertext Transfer Protocol -- HTTP/1.1
88	2396	Uniform Resource Identifiers (URI): Generic Syntax

https://www.ietf.org/rfc/rfc{2616} dc:title “Hypertext Transfer Protocol -- HTTP/1.1”

https://www.ietf.org/rfc/rfc{2396} dc:title “Uniform Resource Identifiers (URI): Generic Syntax”

Q7.2 What are the triples encoded in this HTML?

```
<div vocab="http://xmlns.com/foaf/0.1/" resource="#cathy"
typeof="Person">
 <p> <span property="name">Catherine Faron</span>
  (mail: <span property="mbox">faron@i3s.unice.fr</span>) is a
friend of
  <span property="knows"
resource="http://ns.inria.fr/fabien.gandon#me">Fabien Gandon</span>
 </p>
</div>
```

@prefix n2: < http://xmlns.com/foaf/0.1/">

<#cathy> a rdf:Person;

<#cathy> n2:name “Catherine Faron”;

<#cathy> n2:mbox < faron@i3s.unice.fr >;

<#cathy> n2:knows http://ns.inria.fr/fabien.gandon#me.

4 triples: it uses the foaf vocabulary, Catherine foaf:type Person, Catherine foaf:name Catherine Faron, Catherine foaf:knows Fabien Gandon.

Q7.3 practice:

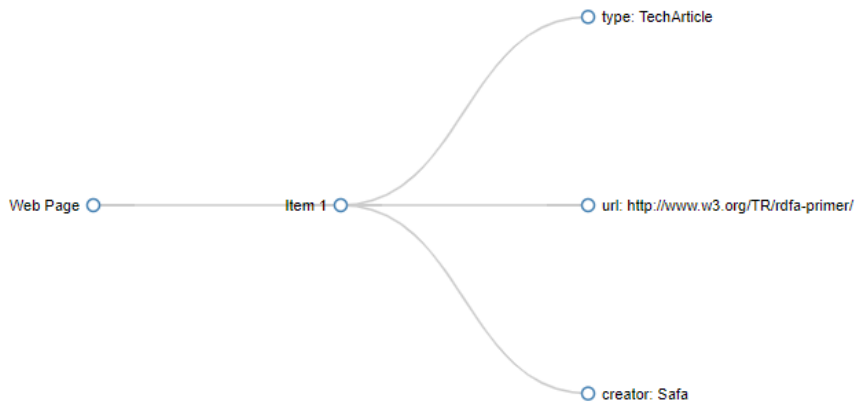
1. Look at the Web Page
   https://www.w3.org/TR/xhtml-rdfa-scenarios/scenario-2.html
2. Call the translator on this Web page to get Turtle:
   http://rdf-translator.appspot.com/
3. What does the extracted triple say?
4. Do the same with:
   http://schema.org/docs/schema_org_rdfa.html
   What kind of data is represented in that page?
5. Again, what are the different subjects described in RDFa in this page:
   http://iricelino.org/rdfa/sample-annotated-page.html

For the 1st one : the triple says the creator is Paul

```
@prefix dc: <http://purl.org/dc/terms/> .

<https://www.w3.org/TR/xhtml-rdfa-scenarios/scenario-2.html> dc:creator "Paul"@en .
```

Q7.4 Use the online tool to play with RDFa adding for instance a “creator” property
https://rdfa.info/play/

```
<span vocab="http://schema.org/" typeof="TechArticle">
   <a property="url" href="http://www.w3.org/TR/rdfa-primer/">
      <span property= "creator">Safa</span></a>.
</span>
```



Q7.5 IMDB uses RDFa – OGP for the I like button

1. Choose a movie on IMDB http://www.imdb.com
2. Copy the URL of the page of the movie
3. Go to the RDFa 1.0 RDFa Distiller and Parser: https://www.w3.org/2007/08/pyRdfa/
4. Open the URI option, past the URL of the movie page and configure and perform the extraction to get Turtle
5. Try also the transformation on the translator: http://rdf-translator.appspot.com/

I chose King Kong 😊

URI : https://www.imdb.com/title/tt0360717/?ref_=hm_stp_pvs_piv_tt_i_2

Using : https://www.w3.org/2007/08/pyRdfa/

```
@prefix fb: <http://www.facebook.com/2008/fbml> .
@prefix ns1: <http://www.w3.org/1999/xhtml/vocab#> .
@prefix ns2: <http://www.facebook.com/2008/> .
@prefix og: <http://ogp.me/ns#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xlink: <http://www.w3.org/1999/xlink> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<https://www.imdb.com/title/tt0360717/?ref_=hm_stp_pvs_piv_tt_i_2> og:description "Directed by Peter Jackson.  With Naomi \
A greedy film producer assembles a team of moviemakers and sets out for the infamous Skull Island, where they find more tha
    og:image "https://m.media-amazon.com/images/M/MV5BMjYxYmRlZWYtMzAwNC00MDA1LWJjNTItOTBjMzlhNGMzYzk3XkEyXkFqcGdeQXVyMTQxM
    og:site_name "IMDb" ;
    og:title "King Kong (2005) - IMDb" ;
    og:type "video.movie" ;
    og:url "http://www.imdb.com/title/tt0360717/" ;
    ns2:fbmlapp_id "115109575169727" .

<https://www.imdb.com/title/tt0360717/?ref_=hm_stp_pvs_piv_tt_i_2#imdbHeader-navDrawerOpen> ns1:role ns1:button .

<https://www.imdb.com/title/tt0360717/?ref_=hm_stp_pvs_piv_tt_i_2#imdbHeader-navDrawerOpen--desktop> ns1:role ns1:button .

<https://www.imdb.com/title/tt0360717/?ref_=hm_stp_pvs_piv_tt_i_2#imdbHeader-searchClose> ns1:role ns1:button .

<https://www.imdb.com/title/tt0360717/?ref_=hm_stp_pvs_piv_tt_i_2#imdbHeader-searchOpen> ns1:role ns1:button .

<https://www.imdb.com/title/tt0360717/?ref_=hm_stp_pvs_piv_tt_i_2#nav-search-form> ns1:role ns1:search .

[] ns1:role ns1:presentation .
```

Using : http://rdf-translator.appspot.com/

```
@prefix fb: <http://www.facebook.com/2008/fbml> .
@prefix og: <http://ogp.me/ns#> .
@prefix xhv: <http://www.w3.org/1999/xhtml/vocab#> .

<https://www.imdb.com/title/tt0360717/?ref_=hm_stp_pvs_piv_tt_i_2> og:description "Directed by Peter Jackson.  With Naomi Wat
    og:image "https://m.media-amazon.com/images/M/MV5BMjYxYmRlZWYtMzAwNC00MDA1LWJjNTItOTBjMzlhNGMzYzk3XkEyXkFqcGdeQXVyMTQxNzI
    og:site_name "IMDb" ;
    og:title "King Kong (2005) - IMDb" ;
    og:type "video.movie" ;
    og:url "http://www.imdb.com/title/tt0360717/" ;
    fb:app_id "115109575169727" .

<https://www.imdb.com/title/tt0360717/?ref_=hm_stp_pvs_piv_tt_i_2#imdbHeader-navDrawerOpen> xhv:role xhv:button .

<https://www.imdb.com/title/tt0360717/?ref_=hm_stp_pvs_piv_tt_i_2#imdbHeader-navDrawerOpen--desktop> xhv:role xhv:button .

<https://www.imdb.com/title/tt0360717/?ref_=hm_stp_pvs_piv_tt_i_2#imdbHeader-searchClose> xhv:role xhv:button .

<https://www.imdb.com/title/tt0360717/?ref_=hm_stp_pvs_piv_tt_i_2#imdbHeader-searchOpen> xhv:role xhv:button .

<https://www.imdb.com/title/tt0360717/?ref_=hm_stp_pvs_piv_tt_i_2#nav-search-form> xhv:role xhv:search .

[] xhv:role xhv:presentation .
```

Q7.6 Test JSON-LD online

1. Transform your FOAF profile in JSON-LD with the translator: http://rdf-translator.appspot.com/
2. Use the following online tool to generate different variations of JSON-LD of your profile (expanded, collapsed, flattened, etc.) http://json-ld.org/playground/

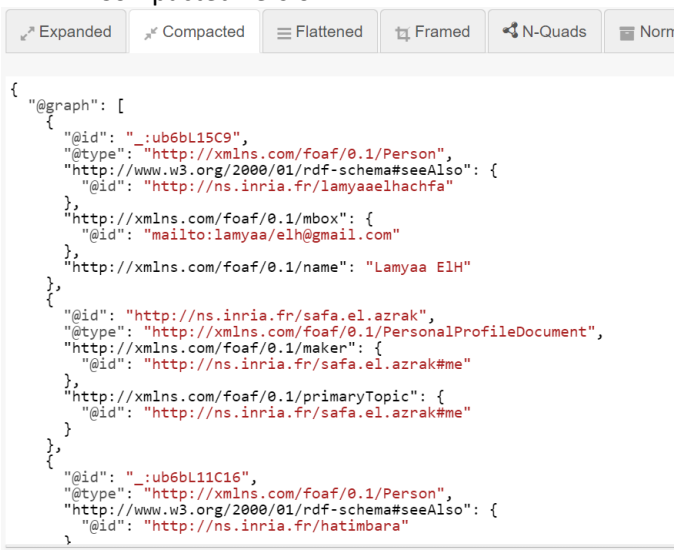
Translation to Json-LD :


```
[
 {
  "@id": "_:ub6bL15C9",
  "@type": [
   "http://xmlns.com/foaf/0.1/Person"
  ],
  "http://www.w3.org/2000/01/rdf-schema#seeAlso": [
   {
```

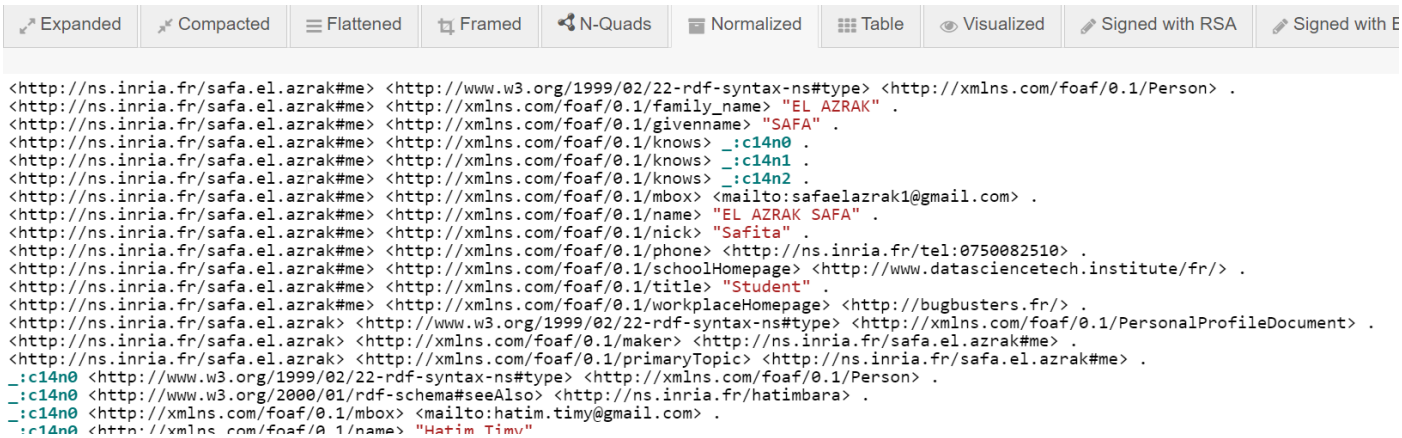
```
    "@id": "http://ns.inria.fr/lamyaaelhachfa"
   }
  ],
  "http://xmlns.com/foaf/0.1/mbox": [
   {
    "@id": "mailto:lamyaa/elh@gmail.com"
   }
  ],
  "http://xmlns.com/foaf/0.1/name": [
   {
    "@value": "Lamyaa ElH"
   }
  ]
 },
 {
  "@id": "http://ns.inria.fr/safa.el.azrak",
  "@type": [
   "http://xmlns.com/foaf/0.1/PersonalProfileDocument"
  ],
  "http://xmlns.com/foaf/0.1/maker": [
   {
    "@id": "http://ns.inria.fr/safa.el.azrak#me"
   }
  ],
  "http://xmlns.com/foaf/0.1/primaryTopic": [
   {
    "@id": "http://ns.inria.fr/safa.el.azrak#me"
   }
  ]
```


Using the online tool for different variations :

- Compacted version



- Normalized version :



Q7.7 To provide the metadata of a CSV file I can...

- ☐ include them in a special column of the CSV.
- ☐ put them in a file with the same name plus “-metadata.json”.
- ☐ put them in the first line of my CSV file.
- ☐ put them in a file called “csv-metadata.json” in the same directory.
- ☐ add the URL of the metadata file to the content of my CSV file.

Q7.8 TV Catalog : Imagine we submit the following call to an LDP platform

```
GET /catalog/tv/ HTTP/1.1
Host: example.org
Accept: text/turtle; charset=UTF-8
```

and we receive the following answer:

```
HTTP/1.1 200 OK
Content-Type: text/turtle; charset=UTF-8
Link: <http://www.w3.org/ns/ldp#Resource>; rel="type",
<http://www.w3.org/ns/ldp#DirectContainer>; rel="type"
Allow: OPTIONS,HEAD,GET,POST,PUT
Accept-Post: text/turtle, application/ld+json
Content-Length: 232
ETag: W/"90231678"
@prefix ldp: <http://www.w3.org/ns/ldp#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix cat: <http://example.org/vocab/catalog#> .
<> a ldp:DirectContainer;   ldp:membershipResource <#cat>;
ldp:hasMemberRelation cat:hasProduct;
  dcterms:title "Container of the TV descriptions";
  ldp:contains <tv1>, <tv2> .
<#cat> a cat:Catalog;   dcterms:title "Catalog of TVs";   cat:hasProduct <tv1>,
<tv2> .
```

Which ones of the following statements are true?

- ☐ the container is just a basic container.
- ☐ the container is a direct container.
- ☐ the container is an indirect container.
- ☐ the platform accepts the GET calls.
- ☐ the platform accepts the PATCH calls.
- ☐ the platform accepts RDF/XML format.
- ☐ the platform accepts RDF Turtle.
- ☐ the platform accepts RDF JSON-LD.
- ☐ a link `hasProduct` is automatically created between the resource #cat and the resources of this container

PRACTICAL SESSIONS

Day 02: Answers to the practical session on RDF.

Software requirements

- A real text editor (e.g. Notepad++, Gedit, Sublime Text, Emacs, etc.)
- The RDF XML online validation service by W3C: https://www.w3.org/RDF/Validator/
- The RDF online translator: http://rdf-translator.appspot.com/
- The SPARQL Corese engine: https://project.inria.fr/corese/

Create RDF

Read carefully the following statements:

> "Jen is a 42-year old woman and she has a shoe size of 36 and trouser size of 38. She is, married to Seb who is a man with whom she had two children: Anny who is a woman and Steffen who is a man. Jen is also an engineer and Catherine and Fabien are her colleagues. Jen's father is a man named Thomas"

1. Use your text editor and write the above statements in RDF in N3 syntax inventing your own vocabulary. Save you file as "Jen.ttl"
2. Use your favorite text or XML editor and write the above statements in RDF in XML syntax reusing the same vocabulary "Jen.rdf"
3. Use the RDF XML online validation service to validate your XML and see the triples https://www.w3.org/RDF/Validator/
4. In the validator use the option to visualize the graph
5. Use the RDF online translator to validate your N3 and translate it into RDF/XML: http://rdf-translator.appspot.com/
6. Compare your RDF/XML with the result of the N3 translation
7. Translate in other formats to see the results.

Code of validated RDF in N3 syntax:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix voc: <http://www.unice.fr/voc#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://www.unice.fr/data#Jen> a voc:Engineer , voc:Woman ;
   voc:age "42"^^xsd:string ;
  voc:shoesize "36"^^xsd:string;
  voc:trousersize "38"^^xsd:string;
   voc:hasChild <http://www.unice.fr/data#Anny>,<http://www.unice.fr/data#Steffen>;
   voc:hasSpouse <http://www.unice.fr/data#Seb> ;
  voc:hasfather <http://www.unice.fr/data#Thomas>;
  voc:hascolleague <http://www.unice.fr/data#Catherine>,<http://www.unice.fr/data#Fabien>;
   voc:name "Jen" .
<http://www.unice.fr/data#Seb> a voc:Man ;
```

```
   voc:hasChild <http://www.unice.fr/data#Anny>, <http://www.unice.fr/data#Steffen> ;
   voc:name "Seb" .
<http://www.unice.fr/data#Anny> a voc:Woman ;
   voc:name "Anny" .
<http://www.unice.fr/data#Steffen> a voc:Man ;
   voc:name "Steffen" .
<http://www.unice.fr/data#Thomas> a voc:Man;
  voc:name "Thomas".
<http://www.unice.fr/data#Catherine> a voc:Woman;
  voc:name "Catherine".
<http://www.unice.fr/data#Fabien> a voc:Man;
  voc:name "Fabien".
```

Code of validated RDF in XML syntax:

```
<?xml version="1.0" encoding="UTF-8"?>

<rdf:RDF

   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

   xmlns:voc="http://www.unice.fr/voc#"

>

  <rdf:Description rdf:about="http://www.unice.fr/data#Jen">

    <voc:hasChild rdf:resource="http://www.unice.fr/data#Steffen"/>

    <voc:hascolleague rdf:resource="http://www.unice.fr/data#Catherine"/>

    <voc:trousersize rdf:datatype="http://www.w3.org/2001/XMLSchema#string">38</voc:trousersize>

    <voc:age rdf:datatype="http://www.w3.org/2001/XMLSchema#string">42</voc:age>

    <voc:hascolleague rdf:resource="http://www.unice.fr/data#Fabien"/>

    <voc:hasChild rdf:resource="http://www.unice.fr/data#Anny"/>

    <voc:shoesize rdf:datatype="http://www.w3.org/2001/XMLSchema#string">36</voc:shoesize>

    <rdf:type rdf:resource="http://www.unice.fr/voc#Engineer"/>

    <voc:hasfather rdf:resource="http://www.unice.fr/data#Thomas"/>

    <voc:hasSpouse rdf:resource="http://www.unice.fr/data#Seb"/>

    <rdf:type rdf:resource="http://www.unice.fr/voc#Woman"/>

    <voc:name>Jen</voc:name>

  </rdf:Description>

  <rdf:Description rdf:about="http://www.unice.fr/data#Thomas">

    <rdf:type rdf:resource="http://www.unice.fr/voc#Man"/>

    <voc:name>Thomas</voc:name>

  </rdf:Description>

  <rdf:Description rdf:about="http://www.unice.fr/data#Seb">

    <voc:hasChild rdf:resource="http://www.unice.fr/data#Steffen"/>

    <voc:hasChild rdf:resource="http://www.unice.fr/data#Anny"/>
```

```
    <rdf:type rdf:resource="http://www.unice.fr/voc#Man"/>
    <voc:name>Seb</voc:name>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.unice.fr/data#Steffen">
    <rdf:type rdf:resource="http://www.unice.fr/voc#Man"/>
    <voc:name>Steffen</voc:name>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.unice.fr/data#Anny">
    <voc:name>Anny</voc:name>
    <rdf:type rdf:resource="http://www.unice.fr/voc#Woman"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.unice.fr/data#Fabien">
    <voc:name>Fabien</voc:name>
    <rdf:type rdf:resource="http://www.unice.fr/voc#Man"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.unice.fr/data#Catherine">
    <voc:name>Catherine</voc:name>
    <rdf:type rdf:resource="http://www.unice.fr/voc#Woman"/>
  </rdf:Description>
</rdf:RDF>
```

Triples :

Your RDF document validated successfully.

Triples of the Data Model

Number	Subject	Predicate	Object
1	http://www.unice.fr/data#Jen	http://www.unice.fr/voc#hasChild	http://www.unice.fr/data#Steffen
2	http://www.unice.fr/data#Jen	http://www.unice.fr/voc#hascolleague	http://www.unice.fr/data#Catherine
3	http://www.unice.fr/data#Jen	http://www.unice.fr/voc#trousersize	"38"^^http://www.w3.org/2001/XMLSchema#string
4	http://www.unice.fr/data#Jen	http://www.unice.fr/voc#age	"42"^^http://www.w3.org/2001/XMLSchema#string
5	http://www.unice.fr/data#Jen	http://www.unice.fr/voc#hascolleague	http://www.unice.fr/data#Fabien
6	http://www.unice.fr/data#Jen	http://www.unice.fr/voc#hasChild	http://www.unice.fr/data#Anny
7	http://www.unice.fr/data#Jen	http://www.unice.fr/voc#shoesize	"36"^^http://www.w3.org/2001/XMLSchema#string
8	http://www.unice.fr/data#Jen	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.unice.fr/voc#Engineer
9	http://www.unice.fr/data#Jen	http://www.unice.fr/voc#hasfather	http://www.unice.fr/data#Thomas
10	http://www.unice.fr/data#Jen	http://www.unice.fr/voc#hasSpouse	http://www.unice.fr/data#Seb
11	http://www.unice.fr/data#Jen	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.unice.fr/voc#Woman
12	http://www.unice.fr/data#Jen	http://www.unice.fr/voc#name	"Jen"
13	http://www.unice.fr/data#Thomas	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.unice.fr/voc#Man
14	http://www.unice.fr/data#Thomas	http://www.unice.fr/voc#name	"Thomas"
15	http://www.unice.fr/data#Seb	http://www.unice.fr/voc#hasChild	http://www.unice.fr/data#Steffen
16	http://www.unice.fr/data#Seb	http://www.unice.fr/voc#hasChild	http://www.unice.fr/data#Anny
17	http://www.unice.fr/data#Seb	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.unice.fr/voc#Man
18	http://www.unice.fr/data#Seb	http://www.unice.fr/voc#name	"Seb"
19	http://www.unice.fr/data#Steffen	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.unice.fr/voc#Man
20	http://www.unice.fr/data#Steffen	http://www.unice.fr/voc#name	"Steffen"
21	http://www.unice.fr/data#Anny	http://www.unice.fr/voc#name	"Anny"
22	http://www.unice.fr/data#Anny	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.unice.fr/voc#Woman
23	http://www.unice.fr/data#Fabien	http://www.unice.fr/voc#name	"Fabien"
24	http://www.unice.fr/data#Fabien	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.unice.fr/voc#Man
25	http://www.unice.fr/data#Catherine	http://www.unice.fr/voc#name	"Catherine"
26	http://www.unice.fr/data#Catherine	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.unice.fr/voc#Woman

Graph :

Code translated to RDFa :

```
<div xmlns="http://www.w3.org/1999/xhtml"
  prefix="
    rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
    voc: http://www.unice.fr/voc#
    xsd: http://www.w3.org/2001/XMLSchema#
    rdfs: http://www.w3.org/2000/01/rdf-schema#"
  >
  <div typeof="voc:Engineer" about="http://www.unice.fr/data#Jen">
    <div rel="voc:hasSpouse">
      <div typeof="voc:Man" about="http://www.unice.fr/data#Seb">
        <div rel="voc:hasChild" resource="http://www.unice.fr/data#Anny"></div>
        <div rel="voc:hasChild" resource="http://www.unice.fr/data#Steffen"></div>
        <div property="voc:name" content="Seb"></div>
      </div>
    </div>
    <div rel="voc:hasfather">
      <div typeof="voc:Man" about="http://www.unice.fr/data#Thomas">
        <div property="voc:name" content="Thomas"></div>
      </div>
    </div>
```

```
    <div property="voc:name" content="Jen"></div>
    <div property="voc:trousersize" datatype="xsd:string" content="38"></div>
    <div property="voc:age" datatype="xsd:string" content="42"></div>
    <div rel="voc:hasChild">
      <div typeof="voc:Woman" about="http://www.unice.fr/data#Anny">
        <div property="voc:name" content="Anny"></div>
      </div>
    </div>
    <div rel="voc:hascolleague">
      <div typeof="voc:Woman" about="http://www.unice.fr/data#Catherine">
        <div property="voc:name" content="Catherine"></div>
      </div>
    </div>
    <div rel="voc:hasChild">
      <div typeof="voc:Man" about="http://www.unice.fr/data#Steffen">
        <div property="voc:name" content="Steffen"></div>
      </div>
    </div>
    <div rel="voc:hascolleague">
      <div typeof="voc:Man" about="http://www.unice.fr/data#Fabien">
        <div property="voc:name" content="Fabien"></div>
      </div>
    </div>
    <div rel="rdf:type" resource="http://www.unice.fr/voc#Woman"></div>
    <div property="voc:shoesize" datatype="xsd:string" content="36"></div>
  </div>
</div>
```

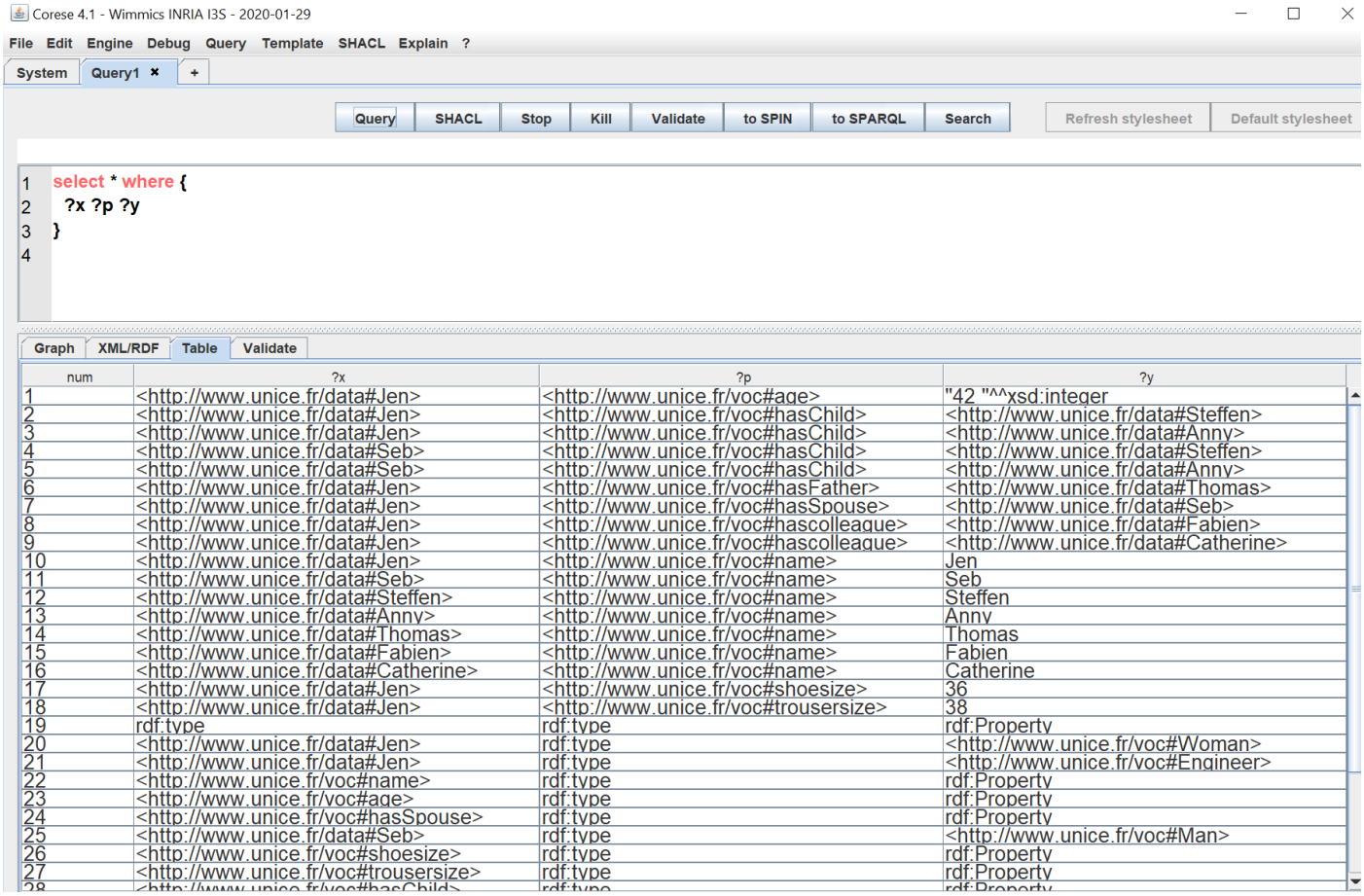
Query your data

Download the Corese.jar library and start it as a standalone application: On Window double-click the file ".jar". If it does not work or on other platforms, run the command " java -jar -Dfile.encoding=UTF8 " followed by the name of the ".jar" archive. Notice that you need java on your machine and proper path configuration.

This interface provides two tabs: (1) one to load input files and see traces of execution, and (2) the default tab to start loading or writing queries and see their result. Load the annotations contained in the file "Jen.rdf" you created and validated before. The interface contains a default SPARQL query:

```
Select ?x ?t where { ?x rdf:type ?t}
```

The SPARQL language will be presented in the next course. Just know that this query can find all of the resources referred to in the data you loaded and their types. Launch the query and check the results.



num	?x	?p	?y
1	<http://www.unice.fr/data#Jen>	<http://www.unice.fr/voc#age>	"42 "^^xsd:integer
2	<http://www.unice.fr/data#Jen>	<http://www.unice.fr/voc#hasChild>	<http://www.unice.fr/data#Steffen>
3	<http://www.unice.fr/data#Jen>	<http://www.unice.fr/voc#hasChild>	<http://www.unice.fr/data#Anny>
4	<http://www.unice.fr/data#Seb>	<http://www.unice.fr/voc#hasChild>	<http://www.unice.fr/data#Steffen>
5	<http://www.unice.fr/data#Seb>	<http://www.unice.fr/voc#hasChild>	<http://www.unice.fr/data#Anny>
6	<http://www.unice.fr/data#Jen>	<http://www.unice.fr/voc#hasFather>	<http://www.unice.fr/data#Thomas>
7	<http://www.unice.fr/data#Jen>	<http://www.unice.fr/voc#hasSpouse>	<http://www.unice.fr/data#Seb>
8	<http://www.unice.fr/data#Jen>	<http://www.unice.fr/voc#hascolleague>	<http://www.unice.fr/data#Fabien>
9	<http://www.unice.fr/data#Jen>	<http://www.unice.fr/voc#hascolleague>	<http://www.unice.fr/data#Catherine>
10	<http://www.unice.fr/data#Jen>	<http://www.unice.fr/voc#name>	Jen
11	<http://www.unice.fr/data#Seb>	<http://www.unice.fr/voc#name>	Seb
12	<http://www.unice.fr/data#Steffen>	<http://www.unice.fr/voc#name>	Steffen
13	<http://www.unice.fr/data#Anny>	<http://www.unice.fr/voc#name>	Anny
14	<http://www.unice.fr/data#Thomas>	<http://www.unice.fr/voc#name>	Thomas
15	<http://www.unice.fr/data#Fabien>	<http://www.unice.fr/voc#name>	Fabien
16	<http://www.unice.fr/data#Catherine>	<http://www.unice.fr/voc#name>	Catherine
17	<http://www.unice.fr/data#Jen>	<http://www.unice.fr/voc#shoesize>	36
18	<http://www.unice.fr/data#Jen>	<http://www.unice.fr/voc#trousersize>	38
19	rdf:type	rdf:type	rdf:Property
20	<http://www.unice.fr/data#Jen>	rdf:type	<http://www.unice.fr/voc#Woman>
21	<http://www.unice.fr/data#Jen>	rdf:type	<http://www.unice.fr/voc#Engineer>
22	<http://www.unice.fr/voc#name>	rdf:type	rdf:Property
23	<http://www.unice.fr/voc#age>	rdf:type	rdf:Property
24	<http://www.unice.fr/voc#hasSpouse>	rdf:type	rdf:Property
25	<http://www.unice.fr/data#Seb>	rdf:type	<http://www.unice.fr/voc#Man>
26	<http://www.unice.fr/voc#shoesize>	rdf:type	rdf:Property
27	<http://www.unice.fr/voc#trousersize>	rdf:type	rdf:Property

Understand existing data

1, Get the RDF/XML about http://ns.inria.fr/fabien.gandon#me and translate the RDF/XML into Turtle/N3

Code of validated RDF in N3 syntax:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

<http://ns.inria.fr/fabien.gandon> a foaf:PersonalProfileDocument ;
   foaf:maker <http://ns.inria.fr/fabien.gandon#me> ;
   foaf:primaryTopic <http://ns.inria.fr/fabien.gandon#me> .

<http://ns.inria.fr/fabien.gandon#me> a foaf:Person ;
   foaf:depiction <http://www-sop.inria.fr/members/Fabien.Gandon/common/FabienGandonBackground.jpg> ;
   foaf:family_name "Gandon" ;
   foaf:givenname "Fabien" ;
   foaf:homepage <http://fabien.info> ;
   foaf:knows [ a foaf:Person ;
         rdfs:seeAlso <http://www.i3s.unice.fr/~faron/> ;
         foaf:mbox <mailto:faron@polytech.unice.fr> ;
         foaf:name "Catherine Faron-Zucker" ],
      [ a foaf:Person ;
         rdfs:seeAlso <http://www-sop.inria.fr/members/Olivier.Corby/> ;
         foaf:mbox <mailto:olivier.corby@inria.fr> ;
         foaf:name "Olivier Corby" ] ;
   foaf:mbox <mailto:fabien.gandon@inria.fr> ;
```

foaf:name "Fabien Gandon" ;
foaf:nick "Bafien" ;
foaf:phone <http://ns.inria.fr/tel:0492387788> ;
foaf:schoolHomepage <http://www.insa-rouen.fr> ;
foaf:title "Dr" ;
foaf:workInfoHomepage <http://fabien.info> ;
foaf:workplaceHomepage <http://www.inria.fr/> .

Can you guess the link between http://ns.inria.fr/fabien.gandon and http://ns.inria.fr/fabien.gandon#me

This URI <http://ns.inria/fr/fabien.gandon> is of type Personal Profile Document

The URI <http://ns.inria.fr/fabien.gandon#me> is Fabien, it's the person

The type of the link is maker, it's between the document and the person : the maker of that document is this resource (Fabien)

2, Get the Turtle data of Paris on DBpedia.org then in the file find the triple that declares it as a capital in Europe.

The triple is:

```
Subject : <http://dbpedia.org/resource/Paris>
Predicate: <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
Object : <http://dbpedia.org/class/yago/WikicatCapitalsInEurope>
```

3, If you don't have the human dataset file yet, at the following address you will find an RDF file containing several annotations:

http://wimmics.inria.fr/doc/tutorial/human_2013.rdf

Download the file and use the RDF XML online validation service to validate the XML and see the triples and the graph.

1. **What is the namespace used for instances / resources created in this file?**

<Because we use ID, the namespace is : `xml:base="http://www.inria.fr/2007/09/11/humans.rdfs-instances">`

2. **By which mechanism is the association between instances and namespace done i.e. how was the instance namespace specified?**

<The entity mechanism :

<!DOCTYPE rdf:RDF [

<!ENTITY humans "http://www.inria.fr/2007/09/11/humans.rdfs">

<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>

>

We declare the base of the URI. Then we have to declare id after the namespace every time we create a new element

3. **What is the namespace of the vocabulary used to describe the resources in the dataset and how is it associated with the tags?**

The namespace is “humans” that we declared in the entity declaration, to called it we use “&humans”

4. **Explain the code `xmlns="&humans;#"`**

It’s a shortcut that refers to http://www.inria.fr/2007/09/11/humans.rdfs

It is a default namespace, using the uri of entity humans + the hashtag (#)

5. **Find *everything* about information on John in this file.**
   all the information:

John is a human of type person, he is 37 years old, name “John”.
His shirt size is 12, shoe size is 14 and trouser size is 44.
He has a parent, Sophie.
He has for child Mark
He has for Spouse Jennifer
HE has for friend Alice.
He has for father Harry

6. **Translate the file in turtle and save it as human_2013.ttl**
   10 first lines:

```
@prefix : <http://www.inria.fr/2007/09/11/humans.rdfs#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve> a :Lecturer,
      :Person ;
   :hasFriend <http://www.inria.fr/2007/09/11/humans.rdfs-instances#Alice> ;
   :hasSpouse <http://www.inria.fr/2007/09/11/humans.rdfs-instances#David> ;
   :name "Eve" .

<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Flora> a :Woman ;
   :age 95 ;
   :hasChild <http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre> ;
   :hasSpouse <http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston> ;
   :name "Flora" .
```

7. **In the turtle version find *everything* about Laura.**
   all the information:

Laura is a human of type person with name “Laura”.
She is a researcher and a lecturer, she has a friend named Alice.
She has child “Catherine”, has for Spouse “William”

Day 02: Answers to the practical session on SHACL.

Software requirements

- A real text editor (e.g. Notepad++, Gedit, Sublime Text, Emacs, etc.)
- The RDF XML online validation service by W3C: https://www.w3.org/RDF/Validator/
- The SPARQL Corese engine: https://project.inria.fr/corese/
- The human dataset file and the human shape file from the archive

What is that shape

With you text editor open the file human_2013_shape.ttl and look at the content

What is the qualified name of the main shape being defined:

<PersonShape>

What is the type of that shape:

<NodeShape/>

What is the target of that shape:

<Person/>

Explain in English the constraint it places on the focus node:

<the constraint is that the person must have a name in English or it's of severity level "violation". It should have at least 1 name>

What is the severity level of that constraint?

<Violation>

In Corese load the dataset human_2013_dataset_rdf.ttl (menu "load RDF") and this shape (menu "load SHACL") and run the validation in a query tab (button "SHACL"). Explain in English what the report is saying:

The constraint is valid. The report conforms false. It means that not all the Person nodes have at least one name : here "Karl" has no name.

Add your constraints

Extend the shape to add a constraint of severity level "Warning" enforcing that a Person should have an age:

```
:PersonShape a sh:NodeShape ;
        sh:targetClass :Person;
        sh:property [
    sh:message "a Person must have a name"@en;
    sh:severity sh:Violation;
    sh:path :name ;
```

```
    sh:minCount 1
  ] ;
  sh:property [
    sh:message "a Person must have an age"@en;
    sh:severity sh:Warning;
    sh:path :age ;
    sh:minCount 1
  ] .
```

In Corese load the human dataset (menu “load RDF”) and this shape (menu “load SHACL”) and run the validation in a query tab (button “SHACL”). Explain in English shat the report is saying:

The report is not valid (conforms false), there are some Person nodes that don’t respect the constraints :

- Karl ➔no name
- Laura ➔ no age
- David ➔ no age
- Eve ➔ no age

Extend the shape to add a constraint of severity level “Info” enforcing that a person’s name should be in English:

<ANSWER HERE/>

In Corese load the human dataset (menu “load RDF”) and this shape (menu “load SHACL”) and run the validation in a query tab (button “SHACL”). Explain in English shat the report is saying:

<ANSWER HERE/>

Day 03: Answers to the practical session on SPARQL.

Software requirements

- The RDF XML online validation service by W3C: https://www.w3.org/RDF/Validator/
- The RDF online translator: http://rdf-translator.appspot.com/
- The SPARQL Corese engine: https://project.inria.fr/corese/

Basic query on RDF human.rdf

If you haven't done it yet download the SPARQL Corese engine.

On Window double-click the file ".jar". If it does not work or on other platforms, run the command " java -jar -Dfile.encoding=UTF8 " followed by the name of the ".jar" archive. Notice that you need java on your machine and proper path configuration

This interface provides two tabs: (1) one to load input files and see traces of execution, and (2) the default tab to start loading or writing queries and see their result.

If you don't have the human dataset file yet download the following file of annotations and save it as "human.rdf":

http://wimmics.inria.fr/doc/tutorial/human_2013.rdf

Load the file human.rdf as RDF data in corese.

Question 1:

Create a new tab to enter the following query and explain what it does and the results you get. This is a good way to familiarize yourself with the data.

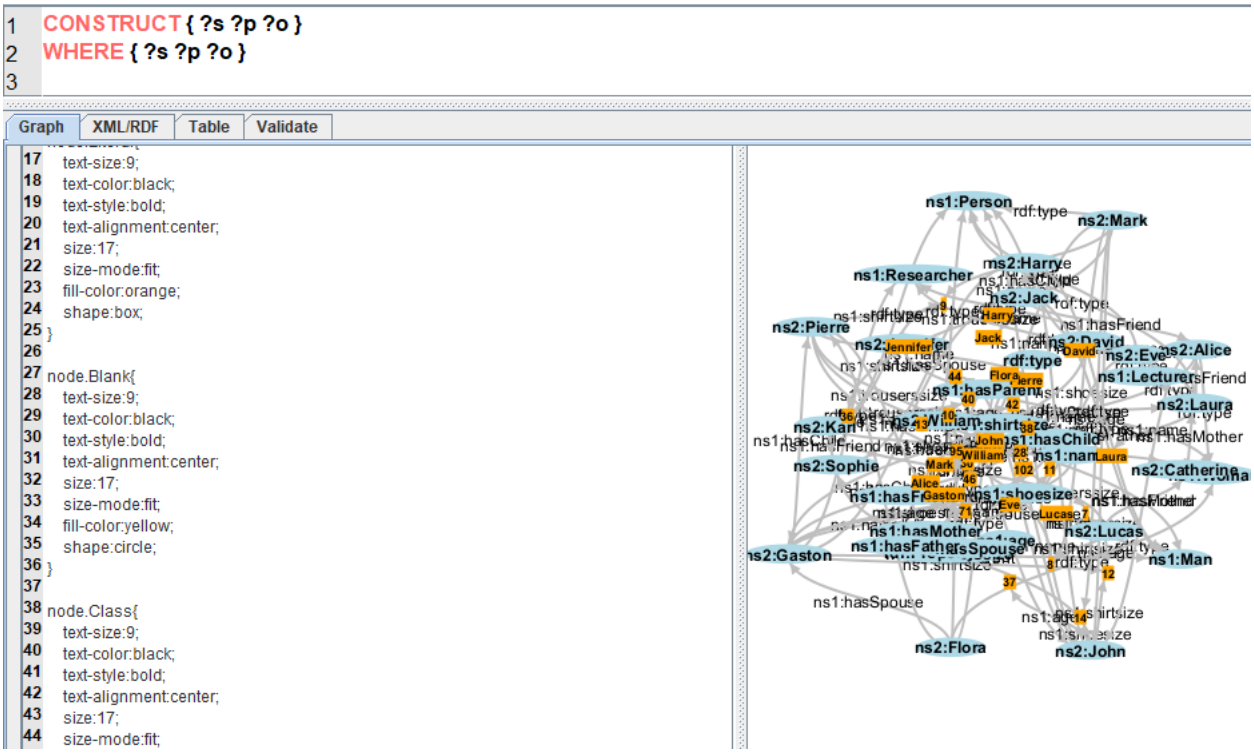
```
CONSTRUCT { ?s ?p ?o } WHERE { ?s ?p ?o }
```

Explanation:

This query constructs the graph with the data from the file loaded.

It's a triple pattern, we have only variables, so it will match all the triples ➔ construct is generating a triple whenever it finds a triple in the data ➔ so the result is all the database

Screenshot:



Question 2:

Create a new tab to enter the following query:

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>

select * where { ?x a ?t . filter(strstarts(?t, h:)) }
```

Translate this query in plain English.

This query retrieves all the types of the resources x Select all the instances/resource and type start with the namespace of h

Run this query. How many answers do you get?

We get 21 answers

Find John and his types in the answers.

John's types:

John is of type Person : <http://www.inria.fr/2007/09/11/humans.rdfs#Person>

Question 3:

In the previous answer, locate the URI of John.

1. formulate a SELECT query to find all the properties of John, using his URI
   Query

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
prefix c: <http://www.inria.fr/2007/09/11/humans.rdfs-instances#>
select *  where {c:John ?p ?v}
```

Results:

Graph | XML/RDF | Table | Validate

num	?p	?v
1	<http://www.inria.fr/2007/09/11/humans.rdfs#age>	37
2	<http://www.inria.fr/2007/09/11/humans.rdfs#hasParent>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Sophie>
3	<http://www.inria.fr/2007/09/11/humans.rdfs#name>	John
4	<http://www.inria.fr/2007/09/11/humans.rdfs#shirtsize>	12
5	<http://www.inria.fr/2007/09/11/humans.rdfs#shoesize>	14
6	<http://www.inria.fr/2007/09/11/humans.rdfs#trouserssize>	44
7	rdf:type	<http://www.inria.fr/2007/09/11/humans.rdfs#Person>

2. request a description of John using the SPARQL clause for this.

Query

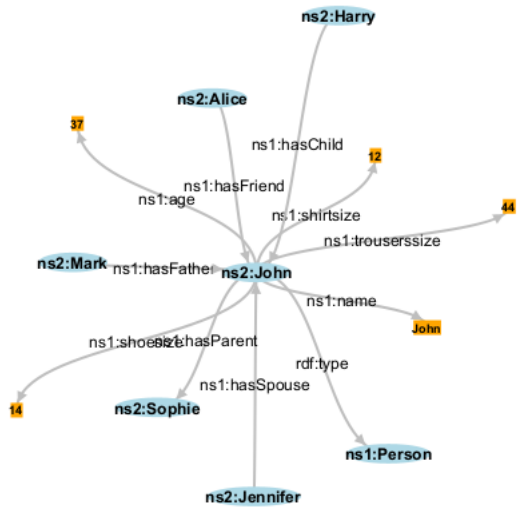
```
DESCRIBE http://www.inria.fr/2007/09/11/humans.rdfs-instances#John
```

Results:

```
1 DESCRIBE <http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>
2
```

Graph | XML/RDF | Table | Validate

```
13    shape:circle;
14 }
15
16 node.Literal{
17    text-size:9;
18    text-color:black;
19    text-style:bold;
20    text-alignment:center;
21    size:17;
22    size-mode:fit;
23    fill-color:orange;
24    shape:box;
25 }
26
27 node.Blank{
28    text-size:9;
29    text-color:black;
30    text-style:bold;
31    text-alignment:center;
32    size:17;
33    size-mode:fit;
34    fill-color:yellow;
35    shape:circle;
36 }
37
38 node.Class{
39    text-size:9;
40    text-color:black;
41    text-style:bold;
42    text-alignment:center;
43    size:17;
44    size-mode:fit;
45    fill-color:blue;
46    shape:circle;
47 }
48
49 edge{
```



Question 4

Create a new tab to enter the following query:

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>

select * where { ?x h:hasSpouse ?y }
```

Translate this query in plain English.

This query returns all the resources x that have the property “hasSpouse” with a resource y

Run this query. How many answers do you get?

We get 6 answers

Graph | XML/RDF | Table | Validate

num	?x	?y
1	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Harry>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Sophie>
2	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#David>
3	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Flora>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>
4	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jennifer>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>
5	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#William>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>
6	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Catherine>

Question 5:

In the RDF file, find the name of the property that is used to give the shoe size of a person.

1. Deduce a query to extract all the persons (h:Person) with their shoe size.

Query:

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
select ?x ?s
where{ ?x a h:Person ;
h:shoesize ?s}
```

Result:

We get : John, Mark, William and Karl

Graph | XML/RDF | Table | Validate

num	?x	?s
1	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	14
2	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	8
3	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#William>	10
4	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>	7

2. Change this query to retrieve all the persons and, if available, their shoe size.

Query:

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
select ?x ?s
where{ ?x a h:Person
optional {?x h:shoesize ?s}}
```

Result:

We get all the resources of type Person, having shoesize or not :

Graph | XML/RDF | Table | Validate

num	?x	
1	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	14
2	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	8
3	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve>	
4	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#David>	
5	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	
6	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#William>	10
7	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>	7

3. Change this query to retrieve all the persons whose shoe size is greater than 8 or whose shirt size is greater than 12.

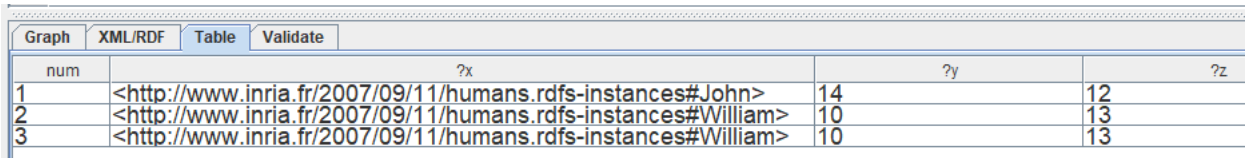
Query:

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
prefix c: <http://www.inria.fr/2007/09/11/humans.rdfs-instances#>
select ?x ?y ?z
where {
?x a h:Person ;
{ ?x h:shoesize ?y .
```

```
FILTER (?y > 8)
OPTIONAL {?x h:shirtsize ?z }
}
UNION
{ ?x h:shirtsize ?z .
FILTER (?z > 12)
OPTIONAL {?x h:shoesize ?y }
}}
```

Result:

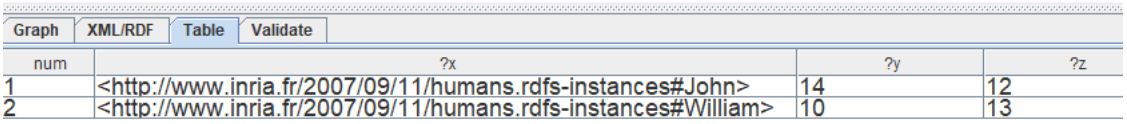
We get 3 answers : John and William (2 times)



Graph | XML/RDF | Table | Validate

num	?x	?y	?z
1	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	14	12
2	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#William>	10	13
3	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#William>	10	13

We can get rid of the duplicate using DISTINCT



Graph | XML/RDF | Table | Validate

num	?x	?y	?z
1	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	14	12
2	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#William>	10	13

Question 6:

In the RDF file, find the name of the property that is used to indicate the children of a person.

1. Formulate a query to find the parents who have at least one child.

Query:

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
select ?x ?s
where{ ?x h:hasChild ?s}
```

How many answers do you get? How many duplicates do you identify in these responses?

We get 5 results and there is a duplicate for Gaston because he has 2 childs.

2. Find a way to avoid duplicates.

Query:

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>

select distinct ?x

where{ ?x h:hasChild ?s }
```

How many answers do you get then?

Now, we get 4 answers.

3. Rewrite a query to find the Persons who have no child.

Query:

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
select *
where{ {?x a h:Person}
minus {?x h:hasChild ?s} }
```

Question 7

In the RDF file, find the name of the property that is used to give the age of a person.

1. Formulate a query to find people with their age.

   Query:

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
select ?x ?a
where{ ?x h:age ?a}
```

   Result:

We get 8 answers :

Graph | XML/RDF | Table | Validate

num	?x	?a
1	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	37
2	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	14
3	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>	102
4	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Flora>	95
5	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>	71
6	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>	12
7	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#William>	42
8	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>	36

2. Formulate a query to find people who are not adults.

   Query:

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
select ?x ?a
where{ ?x h:age ?a.
Filter (?a <=21)}
```

   How many answers do you get?

IF we say that being adult is having above 21 years old : we have 2 answers : Mark and Lucas

Graph | XML/RDF | Table | Validate

num	?x	?a
1	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	14
2	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>	12

3. Use the appropriate query clause to check if Mark is an adult; use the proper clause statement for this type of query to get a true or false answer.

   Query:

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>

prefix c: <http://www.inria.fr/2007/09/11/humans.rdfs-instances#>

ASK {c:Mark h:age ?age

FILTER (?age >21)}
```

➔We get the answer : FALSE

```
<?xml version="1.0" ?>
<sparql xmlns='http://www.w3.org/2005/sparql-results#'>
<head>
</head>
<boolean>false</boolean>
</sparql>
```

4. Write a query that indicates for each person if her age is even (true or false).

Query:

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

PREFIX h: <http://www.inria.fr/2007/09/11/humans.rdfs#>

SELECT ?x ?age ?even

WHERE { ?x a h:Person; h:age ?age BIND( xsd:integer(?age/2)*2=?age AS ?even) }
```

Question 8

1. **Construct** the symmetric of all hasFriend relations using the good SPARQL statement (ex. When finding `Thomas hasFriend Fabien`, your query should construct `Fabien hasFriend Thomas`)

Query:

```
PREFIX h: <http://www.inria.fr/2007/09/11/humans.rdfs#>

CONSTRUCT  {?x h:hasFriend ?y} where {?y h:hasFriend ?x}
```

2. **Insert** the symmetric of all hasFriend relations using the adequate SPARQL statement but check the results with a select query before and after.

Query:

1st Select Query :

```
PREFIX h: <http://www.inria.fr/2007/09/11/humans.rdfs#>

select ?y ?x where {?x h:hasFriend ?y}
```

INSERT Query :

```
PREFIX h: <http://www.inria.fr/2007/09/11/humans.rdfs#>

INSERT {?x h:hasFriend ?y} where {?y h:hasFriend ?x}
```

2nd Select Query :

```
PREFIX h: <http://www.inria.fr/2007/09/11/humans.rdfs#>

select ?y ?x where {?y h:hasFriend ?x}
```

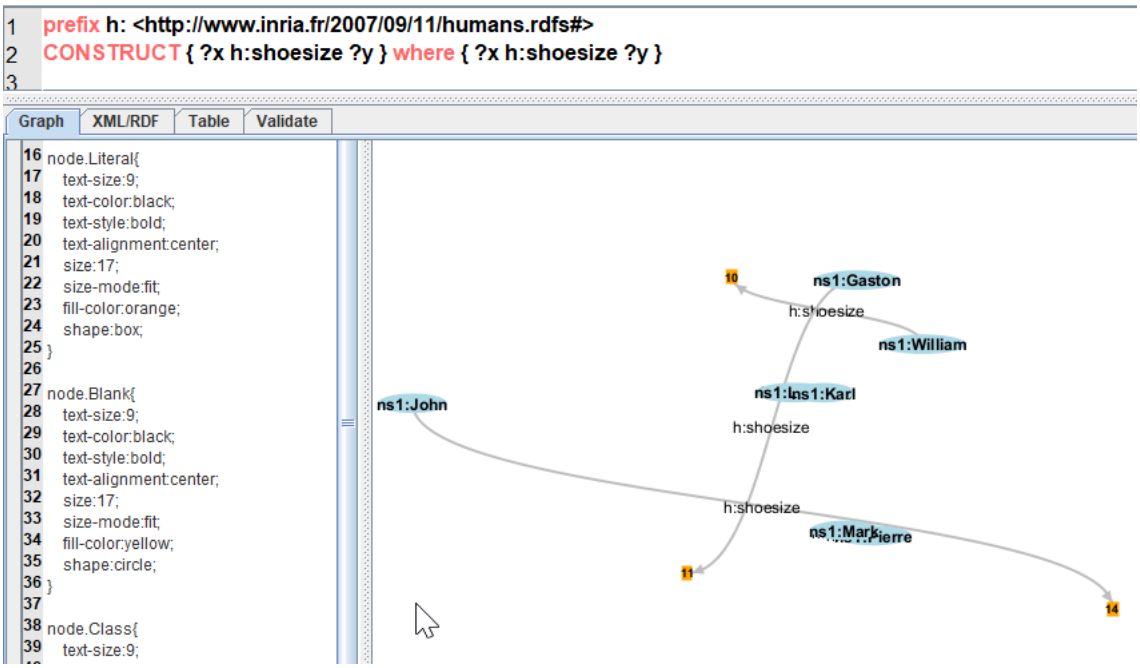
Question 9

Choose and edit one of the SELECT WHERE queries previously written to transform them into a CONSTRUCT WHERE query (retaining the same WHERE clause) in order to visualize the results as a graph.

Query:

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>

CONSTRUCT { ?x h:shoesize ?y } where { ?x h:shoesize ?y }
```

Result:



Question 10

Edit the file to add your own annotation (about you) to the RDF file reusing the properties of the file. Build queries to verify and visualize the annotations you added.

screenshots:

Information added on the file :

```
<Person rdf:ID="Safa">
    <shoesize rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">12</shoesize>
    <age rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">27</age>
    <trouserssize rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">38</trouserssize>
</Person>

<Engineer rdf:about="#Safa">
  <name>Safa</name>
</Engineer>
```

Queries :

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
prefix c: <http://www.inria.fr/2007/09/11/humans.rdfs-instances#>
select * where {c:Safa ?p ?v}
```

Graph | XML/RDF | Table | Validate

num	?p	?v
1	<http://www.inria.fr/2007/09/11/humans.rdfs#age>	27
2	<http://www.inria.fr/2007/09/11/humans.rdfs#name>	Safa
3	<http://www.inria.fr/2007/09/11/humans.rdfs#shoesize>	12
4	<http://www.inria.fr/2007/09/11/humans.rdfs#trouserssize>	38
5	rdf:type	<http://www.inria.fr/2007/09/11/humans.rdfs#Person>
6	rdf:type	<http://www.inria.fr/2007/09/11/humans.rdfs#Engineer>

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
select ?x ?s
where{ ?x a h:Person ;
h:shoesize ?s}
```

aph | XML/RDF | Table | Validate

num	?x	
	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	14
	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	8
	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#William>	10
	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>	7
	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Safa>	12

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
select *
where{ {?x a h:Person}
minus {?x h:hasChild ?s} }
```

aph | XML/RDF | Table | Validate

num	
	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>
	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>
	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve>
	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#David>
	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>
	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#William>
	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>
	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Safa>

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
prefix c: <http://www.inria.fr/2007/09/11/humans.rdfs-instances#>
select ?x ?y ?z
where {
?x a h:Person ;
{ ?x h:shoesize ?y .
FILTER (?y > 8)
OPTIONAL {?x h:shirtsize ?z }
}
UNION
{ ?x h:shirtsize ?z .
FILTER (?z > 12)
OPTIONAL {?x h:shoesize ?y }
}}
```

aph | XML/RDF | Table | Validate

um	?x	?y	
	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	14	12
	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#William>	10	13
	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Safa>	12	
	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#William>	10	13

Question 11

1. Formulate a query to find the persons who share the same shirt size.

Query:

```
PREFIX h: <http://www.inria.fr/2007/09/11/humans.rdfs#>

SELECT DISTINCT * where {?x1 h:shirtsize ?y . ?x2 h:shirtsize ?y FILTER (?x1!=?x2 && ?x1<?x2)}
```

2. Find the persons who have the same size shirt and construct a seeAlso relationship between them.

Query:

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>

construct {?x h:seeAlso ?y} where{?y h:shirtsize ?s. ?x h:shirtsize ?s. filter(?x!=?y)}
```

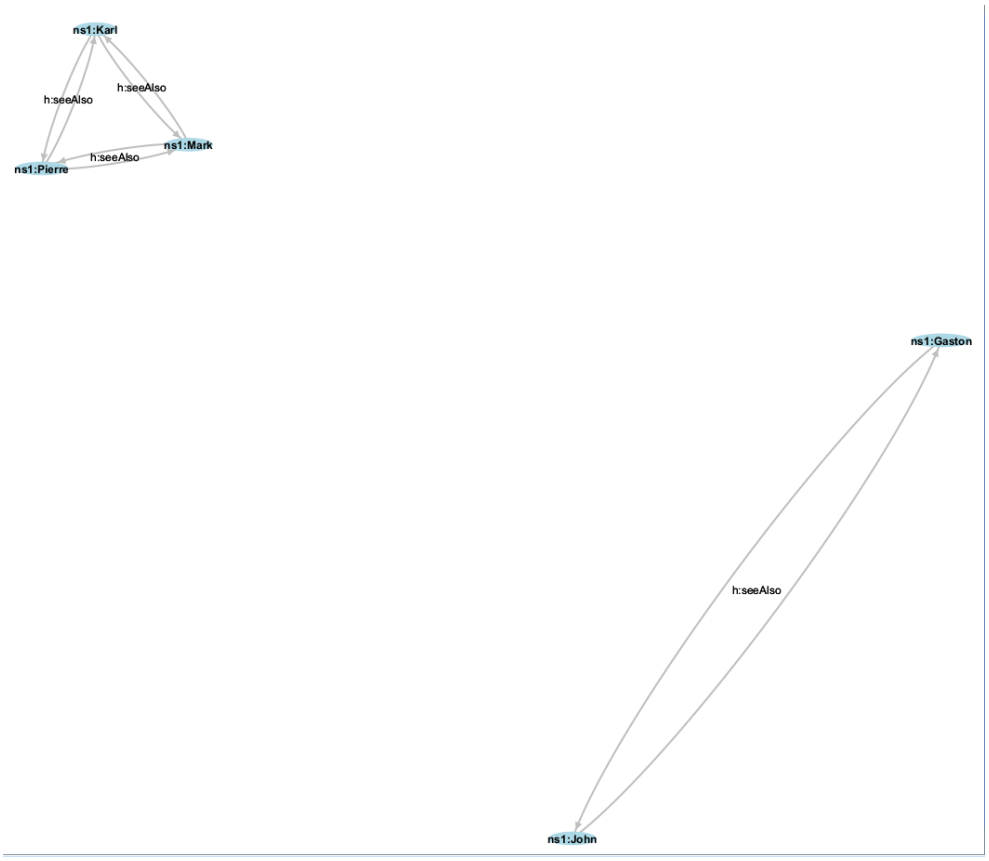
Graph | XML/RDF | Table | Validate

num	?y	?s	?x
1	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	12	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>
2	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	9	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>
3	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	9	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>
4	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>	12	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>
5	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>	9	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>
6	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>	9	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>
7	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>	9	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>
8	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>	9	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>

3. Change the query into an insert.
4. Visualize the resources connected by seeAlso (use the CONSTRUCT clause).

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
construct {?x h:seeAlso ?y}
where {?x h:seeAlso ?y}
```

screenshot:



5. Adapt the first query to find persons who have the same shoe size and insert a seeAlso relationship between them.

Query:

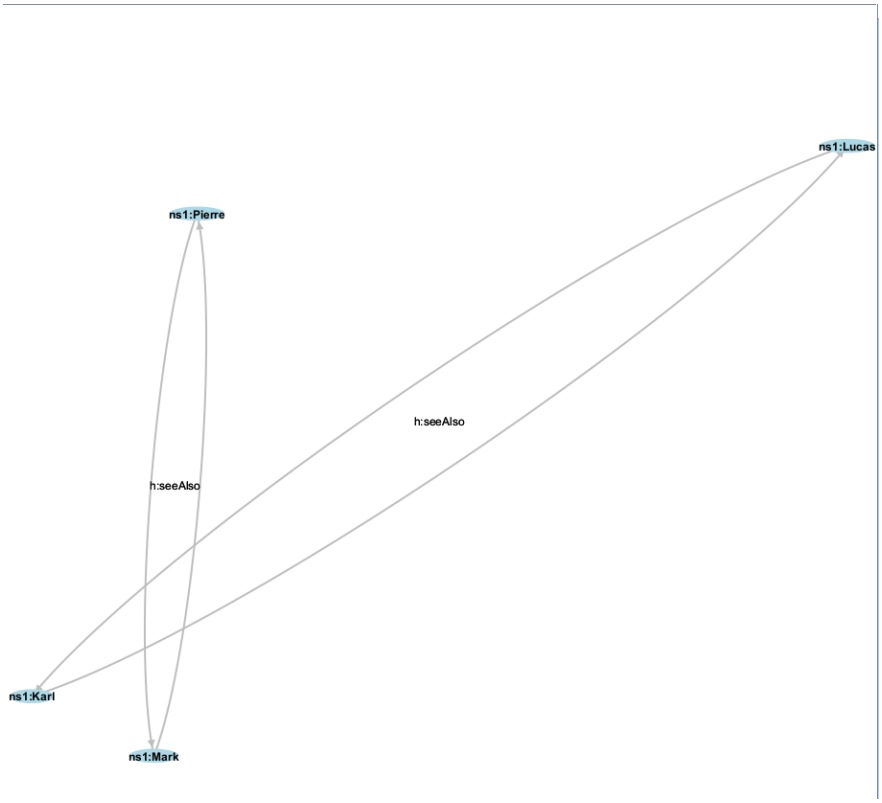
```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>

insert {?x h:seeAlso ?y}

where{?y h:shoesize ?s.

?x h:shoesize ?s.

Filter(?x!=?y)}
```

XML/RDF | Table | Validate

	?y	?s	?x
	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	8	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>
	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>	8	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>
	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>	7	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>
	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>	7	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>

6. Visualize the resources connected by seeAlso (use the CONSTRUCT clause)

screenshot:



7. Change the query to find the resources connected by a path consisting of one or several seeAlso relationships.

Query:

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>

select distinct ?x where {?x h:seeAlso ?y}
```

?x
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>

8. Reload the engine (option reload in the menu) and rerun the last visualization query.

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
select distinct ?x where {?x h:seeAlso ?y}
```

Graph | XML/RDF | Table | Validate

num	?x

Question 12

1. Find the largest shoe size

Query:

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
select (max(?shoe) as ?max) where {?x h:shoesize ?shoe}
```

The max shoe size is 14.

2. Find people who have the biggest size of shoe (subquery + aggregate)

Query:

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>

select ?name{

{ select (max(?shoe) as ?max) where {?x h:shoesize ?shoe}}

?x h:shoesize ?max

?x h:name ?name}
```

John has the biggest size.

3. Calculate the average shoe size using the appropriate aggregation operator

Query:

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>

select (avg(?shoe) as ?avg) where {?x h:shoesize ?shoe}
```

The average shoesize is 9.625

4. Check the average with your own calculation using `sum()` and `count()`

Query:

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>

select (sum(?shoe)/count(?shoe) as ?avg) where {?x h:shoesize ?shoe}
```

Question 13

Find couples without children

Query:

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>

select * where

{{?x h:hasSpouse ?couple} minus{?x h:hasChild ?child}}
```

We have 4 couples without children :

?x	?couple
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#David>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jennifer>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Catherine>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#William>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>

Question 14

Using INSERT DATA, create a new person with its properties. Then, check that it has been created.

Insert:

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>

prefix instance: <http://www.inria.fr/2007/09/11/humans.rdfs-instances#>

Insert DATA{instance:Hatim h:name "Hatim"; h:age 29}
```

Screenshot result:

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
select ?x ?y where {<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Hatim> ?x ?y}
```

Graph | XML/RDF | Table | Validate

num	?x	
	<http://www.inria.fr/2007/09/11/humans.rdfs#age>	29
	<http://www.inria.fr/2007/09/11/humans.rdfs#name>	Hatim

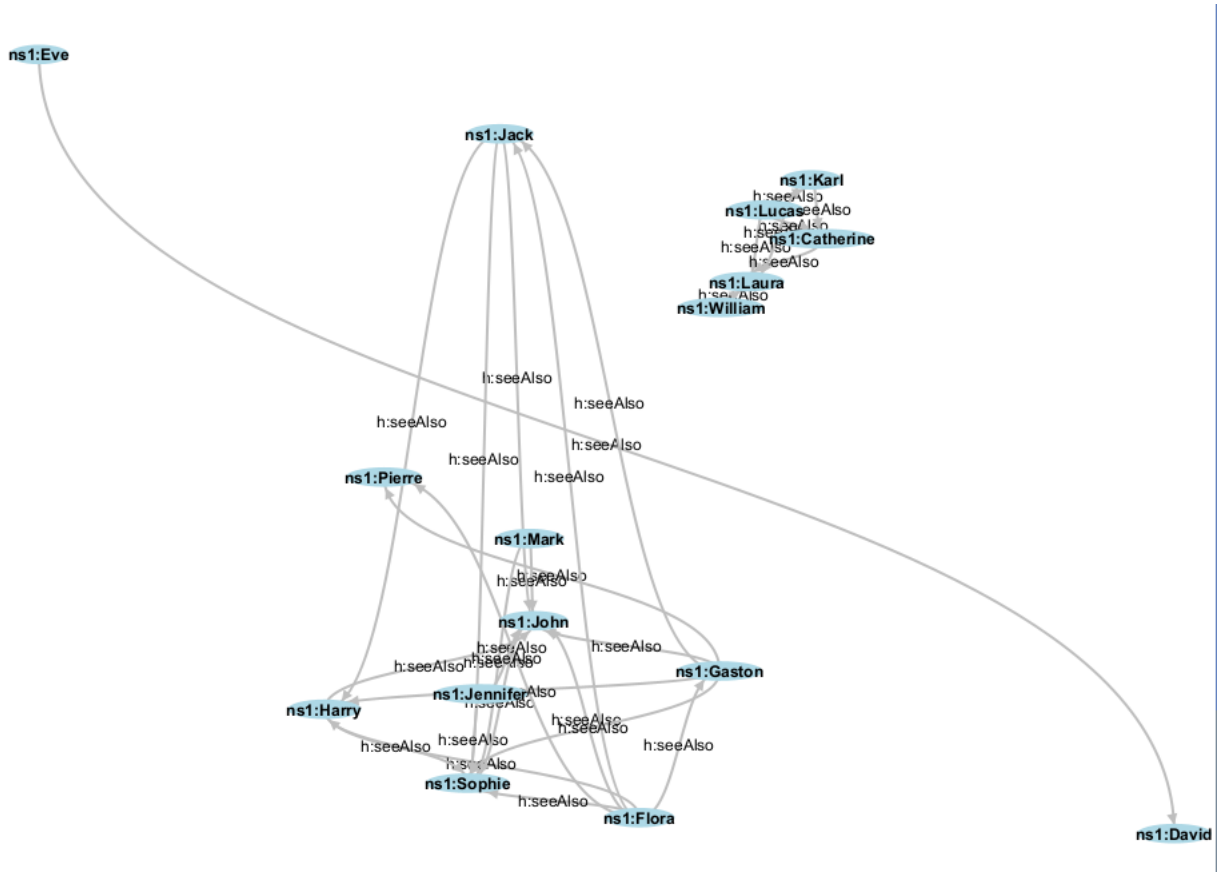
Question 15

Find the people connected by paths of any family links. Construct an arc seeAlso between them to visualize the result.

query:

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>

Construct {?x h:seeAlso ?y}  where {?x ?y ?z . Filter (strstarts(?y,h:has)) . minus{?x h:hasFriend ?f}}
```

screenshot:



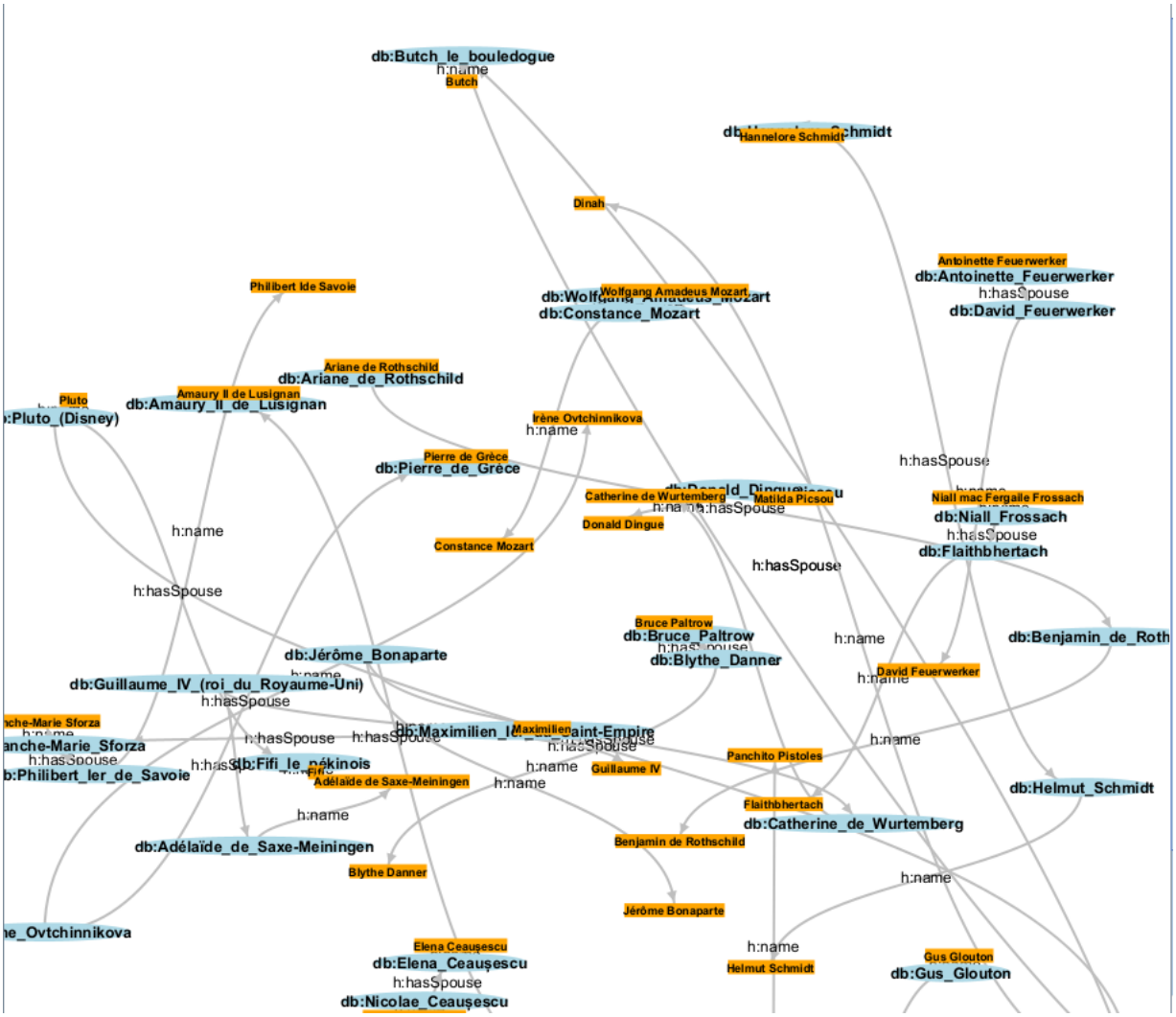
Question 16

Run the following query:

```
prefix db: <http://dbpedia.org/ontology/>
prefix foaf: <http://xmlns.com/foaf/0.1/>
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
construct  { ?x h:name ?nx . ?y h:name ?ny . ?x h:hasSpouse ?y }
where {
  service <http://fr.dbpedia.org/sparql/> {
     select * where {
       ?x db:spouse ?y .
       ?x foaf:name ?nx .
       ?y foaf:name ?ny .
     }
     limit 20
  }
}
```

Explain what it does

This query connects to the french Dbpedia sparql. It queries inside Dbpedia, retrieving the 20 first data, to construct the graph of ?x h:name ?nx . ?y h:name ?ny . ?x h:hasSpouse ?y }. That 20 first data are the 20 first triples from couples with their names.



modify it to insert new persons in the base and check the results.

query:

```
prefix db: <http://dbpedia.org/ontology/>

prefix foaf: <http://xmlns.com/foaf/0.1/>

prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>

Insert { ?x h:name ?nx . ?y h:name ?ny . ?x h:hasSpouse ?y }

where {

service <http://fr.dbpedia.org/sparql/> {

select * where {

?x db:spouse ?y .

?x foaf:name ?nx .

?y foaf:name ?ny .

} limit 20}}
```

?x	?y	?nx	?ny
db:Guillaume_IV_(roi_du_Royaume-Uni)	db:Adélaïde_de_Saxe-Meiningen	"Guillaume IV"@fr	"Adélaïde de Saxe-Meiningen"@fr
db:Isabelle_Ire_de_Jérusalem	db:Amaury_II_de_Lusignan	"Isabellede Jérusalem"@fr	"Amaury II de Lusignan"@fr
db:Ariane_de_Rothschild	db:Benjamin_de_Rothschild	"Ariane de Rothschild"@fr	"Benjamin de Rothschild"@fr
db:Maximilien_Ier_du_Saint-Empire	db:Blanche-Marie_Sforza	"Maximilien"@fr	"Blanche-Marie Sforza"@fr
db:Philibert_Ier_de_Savoie	db:Blanche-Marie_Sforza	"Philibert Ide Savoie"@fr	"Blanche-Marie Sforza"@fr
db:Bruce_Paltrow	db:Blythe_Danner	"Bruce Paltrow"@fr	"Blythe Danner"@fr
db:Dinah_le_teckel	db:Butch_le_bouledogue	"Dinah"@fr	"Butch"@fr
db:Jérôme_Bonaparte	db:Catherine_de_Wurtemberg	"Jérôme Bonaparte"@fr	"Catherine de Wurtemberg"@fr
db:Gus_Glouton	db:Clara_Cluck	"Gus Glouton"@fr	"Clara Cluck"@fr
db:Panchito_Pistoles	db:Clara_Cluck	"Panchito Pistoles"@fr	"Clara Cluck"@fr
db:Wolfgang_Amadeus_Mozart	db:Constance_Mozart	"Wolfgang Amadeus Mozart"@fr	"Constance Mozart"@fr
db:Antoinette_Feuerwerker	db:David_Feuerwerker	"Antoinette Feuerwerker"@fr	"David Feuerwerker"@fr
db:Butch_le_bouledogue	db:Dinah_le_teckel	"Butch"@fr	"Dinah"@fr
db:Pluto_(Disney)	db:Dinah_le_teckel	"Pluto"@fr	"Dinah"@fr
db:Matilda_Picsou	db:Donald_Dingue	"Matilda Picsou"@fr	"Donald Dingue"@fr
db:Nicolae_Ceausescu	db:Elena_Ceausescu	"Nicolae Ceausescu"@fr	"Elena Ceausescu"@fr
db:Pluto_(Disney)	db:Fifi_le_pékinois	"Pluto"@fr	"Fifi"@fr
db:Niall_Frossach	db:Flaithbhertach	"Niall mac Fergaile Frossach"@fr	"Flaithbhertach"@fr
db:Hannelore_Schmidt	db:Helmut_Schmidt	"Hannelore Schmidt"@fr	"Helmut Schmidt"@fr
db:Irène_Ovtchinnikova	db:Pierre_de_Grèce	"Irène Ovtchinnikova"@fr	"Pierre de Grèce"@fr

Day 04: Answers to the practical session on RDFS.

Software requirements

- The RDF XML online validation service by W3C: https://www.w3.org/RDF/Validator/
- The RDF online translator: http://rdf-translator.appspot.com/
- The SPARQL Corese engine: https://project.inria.fr/corese/

Create your own schema Family.rdfs

- Write the the RDF schema that you used in the description of Jen in a RDF/XML (or in turtle and then translate it) and save the RDF/XML in a file called “Family.rdfs”. Of course, this assumes that the URIs for the classes and properties declared/used must match in both files. You mays have to update the files Jen.rdf and Jen.ttl to use your ontology.

Your schema:

In Turtle:

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix voc: <http://www.unice.fr/voc#> .
@base <http://www.unice.fr/voc> .

#class
voc:Man a rdfs:Class;
rdfs:subClassOf voc:Person ;
rdfs:seeAlso voc:Women ;
rdfs:comment "a male adult"@en.

voc:Women a rdfs:Class;
rdfs:subClassOf voc:Person ;
rdfs:seeAlso voc:Man ;
rdfs:comment "a female adult"@en.

voc:Enginneer a rdfs:Class;
rdfs:subClassOf voc:Women, voc:Man.

#Property
voc:hasParent a rdf:Property.
voc:hasMother a rdf:Property;
rdfs:subProertyOf voc:hasParent;
rdfs:domain voc:Women ;
rdfs:range voc:Person .
voc:hasFather a rdf:Property;
rdfs:subProertyOf voc:hasParent;
rdfs:domain voc:Man ;
rdfs:range voc:Person.
voc:hasChild a rdf:Property.
voc:hasSpouse a rdf:Property.
voc:hasColleague a rdf:Property.

#label
voc:name a rdf:Property;
```

rdfs:label "nom"@fr,"name"@en.
voc:age a rdf:Property.
voc:shoesSize a rdf:Property.

voc:trouserSize a rdf:Property.

In RDF :

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
>
  <rdf:Description rdf:about="http://www.unice.fr/voc#hasFather">
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:subProertyOf rdf:resource="http://www.unice.fr/voc#hasParent"/>
    <rdfs:range rdf:resource="http://www.unice.fr/voc#Person"/>
    <rdfs:domain rdf:resource="http://www.unice.fr/voc#Man"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.unice.fr/voc#Man">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.unice.fr/voc#Person"/>
    <rdfs:comment xml:lang="en">a male adult</rdfs:comment>
    <rdfs:seeAlso rdf:resource="http://www.unice.fr/voc#Women"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.unice.fr/voc#hasColleague">
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.unice.fr/voc#name">
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:label xml:lang="en">name</rdfs:label>
    <rdfs:label xml:lang="fr">nom</rdfs:label>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.unice.fr/voc#Enginneer">
    <rdfs:subClassOf rdf:resource="http://www.unice.fr/voc#Women"/>
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.unice.fr/voc#Man"/>
  </rdf:Description>
```

```
  <rdf:Description rdf:about="http://www.unice.fr/voc#Women">
    <rdfs:comment xml:lang="en">a female adult</rdfs:comment>
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.unice.fr/voc#Person"/>
    <rdfs:seeAlso rdf:resource="http://www.unice.fr/voc#Man"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.unice.fr/voc#hasChild">
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.unice.fr/voc#age">
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.unice.fr/voc#hasMother">
    <rdfs:domain rdf:resource="http://www.unice.fr/voc#Women"/>
    <rdfs:subProertyOf rdf:resource="http://www.unice.fr/voc#hasParent"/>
    <rdfs:range rdf:resource="http://www.unice.fr/voc#Person"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.unice.fr/voc#shoesSize">
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.unice.fr/voc#hasParent">
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.unice.fr/voc#hasSpouse">
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.unice.fr/voc#trouserSize">
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
</rdf:RDF>
```

Check that your RDF schema and RDF files are valid using the W3C's RDF validation service.

Launch the standalone interface of Corese and load your files Family.rdfs and Jen.rdf

The interface contains a default SPARQL query:
`Select ?x ?t where {?x rdf:type ?t}`
Launch the query and look at the results.

Screenshot:

```
1  Select ?x ?t where {?x rdf:type ?t}
```

Graph | XML/RDF | Table | Validate

num	?x	?t
1	rdf:type	rdf:Property
2	<http://www.unice.fr/data#Jen>	<http://www.unice.fr/voc#Engineer>
3	<http://www.unice.fr/data#Jen>	<http://www.unice.fr/voc#Woman>
4	<http://www.unice.fr/voc#hasChild>	rdf:Property
5	<http://www.unice.fr/data#Steffen>	<http://www.unice.fr/voc#Man>
6	<http://www.unice.fr/voc#hascolleague>	rdf:Property
7	<http://www.unice.fr/data#Catherine>	<http://www.unice.fr/voc#Woman>
8	<http://www.unice.fr/voc#trousersize>	rdf:Property
9	<http://www.unice.fr/voc#age>	rdf:Property
10	<http://www.unice.fr/data#Fabien>	<http://www.unice.fr/voc#Man>
11	<http://www.unice.fr/data#Anny>	<http://www.unice.fr/voc#Woman>
12	<http://www.unice.fr/voc#shoesize>	rdf:Property
13	<http://www.unice.fr/voc#hasfather>	rdf:Property
14	<http://www.unice.fr/data#Thomas>	<http://www.unice.fr/voc#Man>
15	<http://www.unice.fr/voc#hasSpouse>	rdf:Property
16	<http://www.unice.fr/data#Seb>	<http://www.unice.fr/voc#Man>
17	<http://www.unice.fr/voc#name>	rdf:Property
18	<http://www.unice.fr/voc#hasMother>	rdf:Property

- Modify your ontology to declare the classes of Man and Woman as sub classes of Human (don't change the data), reload the schemas and data and search for the humans to see the results

Screenshot:

```
1  prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
2  prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
3  prefix voc: <http://www.unice.fr/voc#> .
4
5  select *  where {?x a voc:Human}
6
```

Graph | XML/RDF | Table | Validate

num	
1	<http://www.unice.fr/data#Jen>
2	<http://www.unice.fr/data#Steffen>
3	<http://www.unice.fr/data#Catherine>
4	<http://www.unice.fr/data#Fabien>
5	<http://www.unice.fr/data#Anny>
6	<http://www.unice.fr/data#Thomas>
7	<http://www.unice.fr/data#Seb>

Explanation:

Since all the persons are either Woman or Man, and we defined Human as super Class of these 2 classes so they are Humans.

- Modify your ontology to declare the properties hasChild and hasSpouse as sub properties of familyLink (don't change the data), reload the schemas and data and search for the family links to see the results.

Screenshot:

```
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
prefix voc: <http://www.unice.fr/voc#> .

select *  where {?x  voc:familyLink  ?y}
```

Graph | XML/RDF | Table | Validate

num	?x	
1	<http://www.unice.fr/data#Jen>	<http://www.unice.fr/data#Steffen>
2	<http://www.unice.fr/data#Jen>	<http://www.unice.fr/data#Anny>
3	<http://www.unice.fr/data#Jen>	<http://www.unice.fr/data#Seb>
4	<http://www.unice.fr/data#Seb>	<http://www.unice.fr/data#Steffen>
5	<http://www.unice.fr/data#Seb>	<http://www.unice.fr/data#Anny>

Explanation:

It gives everyone that has a property of has child or has spouse because these two are subproperties of family link.

- Modify your ontology to declare the class FamilyMember and use it to specify the signature of the property familyLink (don't change the data) then reload the schemas and data and search for the family members.

Screenshot:

```
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
prefix voc: <http://www.unice.fr/voc#> .

select *  where {?x  a voc:FamilyMember }
```

Graph | XML/RDF | Table | Validate

num	
1	<http://www.unice.fr/data#Jen>
2	<http://www.unice.fr/data#Steffen>
3	<http://www.unice.fr/data#Anny>
4	<http://www.unice.fr/data#Seb>

Explanation:

We defined the class FamilyMember and definedits domain and range using Family link.

The result of the query : we got all the family members who was linked with Family link.

About the human.rdfs schema

1. If you don't have the human schema file yet, download the RDF schema available at this address and save it as "human.rdfs":
   http://wimmics.inria.fr/doc/tutorial/human_2013.rdfs

2. What is the namespace associated with this ontology? How was it associated?

Namespace : http://www.inria.fr/2007/09/11/humans.rdfs

The namespace is associated with the base using : xml:base="http://www.inria.fr/2007/09/11/humans.rdfs"

3. Look at the XML structure of this file and locate different syntactic properties: the different possible uses of the markup (ex: opening tag and closing, single tag), the use of namespaces for qualified names, the use of entities, etc.

< and > for declaration is using the namespace. Every time we declare a new class it would be located to the base namespaces.
It would transfer to http://www.inria.fr/2007/09/11/humans.rdfs +# +"ID" where ID would be replaced by the defined name

4. Locate the use of the terms of the RDF (S) language: Class, Property, label, comment, range, domain, subClassOf, subPropertyOf, etc. To what namespaces are they associated?

- **Property** ➔ namespace of xml:base=http://www.inria.fr/2007/09/11/humans.rdfs
- **Class, Label, comment, range, domain, subClassOF, subPropertyOf** ➔ namespace of xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"

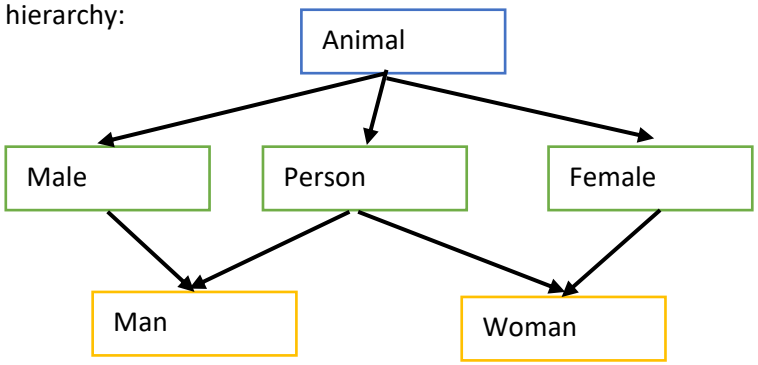
5. What are the classes of resources that can have the age property? Explain

Age property doesn't have signature (domain, range), any resource can use it

Person can use it : Man, Woman, Lecturer, Researcher.

6. Look at the beginning of the file and draw the subgraph of the hierarchy containing the classes `Animal`, `Man` and `Woman`.

Drawing of hierarchy:



CLASS : Animal

SubClass of Animal

Male, Person, Female

SubClass of Person

Man and Woman

Query the schema itself

Reset or relaunch the standalone Corese search engine interface and load the file `human.rdfs` (and only this one).

1. Write a query to find all the classes of the ontology.

query:

```
select distinct ?x where {?x a rdfs:Class}
```

?x
<http://www.inria.fr/2007/09/11/humans.rdfs#Animal>
<http://www.inria.fr/2007/09/11/humans.rdfs#Male>
<http://www.inria.fr/2007/09/11/humans.rdfs#Female>
<http://www.inria.fr/2007/09/11/humans.rdfs#Man>
<http://www.inria.fr/2007/09/11/humans.rdfs#Person>
<http://www.inria.fr/2007/09/11/humans.rdfs#Lecturer>
<http://www.inria.fr/2007/09/11/humans.rdfs#Researcher>
<http://www.inria.fr/2007/09/11/humans.rdfs#Woman>

2. Write a query to find all the links subClassOf in the ontology.

query:

```
select   ?x   ?y where {?x rdfs:subClassOf ?y}
```

?x	?y
<http://www.inria.fr/2007/09/11/humans.rdfs#Male>	<http://www.inria.fr/2007/09/11/humans.rdfs#Animal>
<http://www.inria.fr/2007/09/11/humans.rdfs#Female>	<http://www.inria.fr/2007/09/11/humans.rdfs#Animal>
<http://www.inria.fr/2007/09/11/humans.rdfs#Man>	<http://www.inria.fr/2007/09/11/humans.rdfs#Male>
<http://www.inria.fr/2007/09/11/humans.rdfs#Man>	<http://www.inria.fr/2007/09/11/humans.rdfs#Person>
<http://www.inria.fr/2007/09/11/humans.rdfs#Person>	<http://www.inria.fr/2007/09/11/humans.rdfs#Animal>
<http://www.inria.fr/2007/09/11/humans.rdfs#Lecturer>	<http://www.inria.fr/2007/09/11/humans.rdfs#Person>
<http://www.inria.fr/2007/09/11/humans.rdfs#Researcher>	<http://www.inria.fr/2007/09/11/humans.rdfs#Person>
<http://www.inria.fr/2007/09/11/humans.rdfs#Woman>	<http://www.inria.fr/2007/09/11/humans.rdfs#Female>
<http://www.inria.fr/2007/09/11/humans.rdfs#Woman>	<http://www.inria.fr/2007/09/11/humans.rdfs#Person>

3. Write a query to find the definitions and translations of "shoe size" (*other* labels and comments in different languages for the resource labeled "shoe size").

query:

```
select * where {<http://www.inria.fr/2007/09/11/humans.rdfs#shoesize> rdfs:label ?y;

rdfs:comment ?z. Filter(?y!="shoe size"@en && (lang(?y)!=lang(?z))).}
```

answers:

```
1  select * where {<http://www.inria.fr/2007/09/11/humans.rdfs#shoesize> rdfs:label ?y;
2  rdfs:comment ?z. Filter(?y!="shoe size"@en && (lang(?y)!=lang(?z))).}
3
```

Graph | XML/RDF | Table | Validate

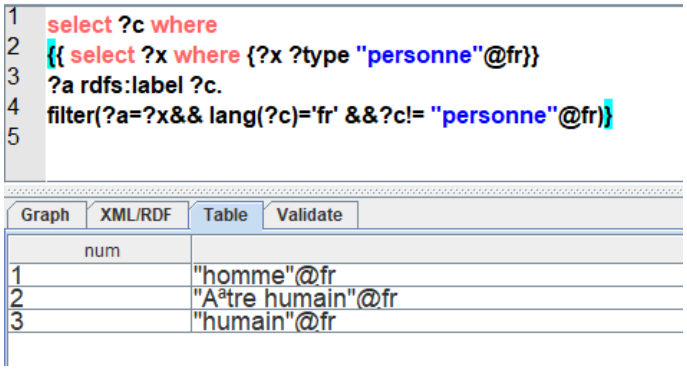
num	?y	?z
1	"size"@en	"taille, exprimA©e en points, des chaussures d'une personne."@fr
2	"pointure"@fr	"express in some way the approximate length of the shoes for a person."@en

4. Write a query to find the synonyms in French of the word 'personne' in French (*other* labels in the same language for the same resource/class/property). What are the answers?

query:

```
select ?c where{

{ select ?x where {?x ?type "personne"@fr}}

?a rdfs:label ?c.

Filter(?a=?x && lang(?c)='fr' && ?c!= "personne"@fr)}
```

answers:



```
1  select ?c where
2  {{ select ?x where {?x ?type "personne"@fr}}
3  ?a rdfs:label ?c.
4  filter(?a=?x&& lang(?c)='fr' &&?c!= "personne"@fr)}
5
```

Graph | XML/RDF | Table | Validate

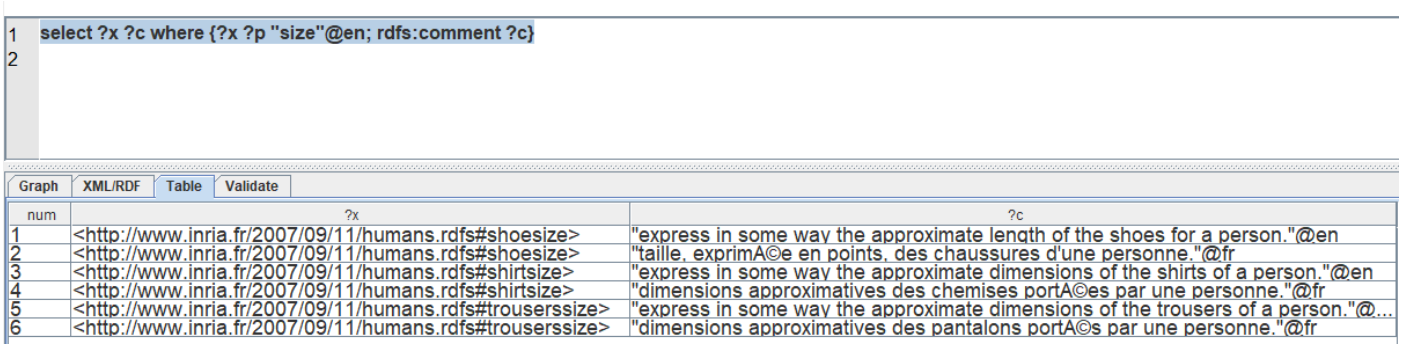
num	
1	"homme"@fr
2	"Aªtre humain"@fr
3	"humain"@fr

5. Write a query to find the different meaning of the term "size" (disambiguation using the different comments attached to different resources/classes/properties having the label "size"). What are the answers?

query:

```
select ?x ?c where {?x ?p "size"@en; rdfs:comment ?c}
```

answers:



```
1  select ?x ?c where {?x ?p "size"@en; rdfs:comment ?c}
2
```

Graph | XML/RDF | Table | Validate

num	?x	?c
1	<http://www.inria.fr/2007/09/11/humans.rdfs#shoesize>	"express in some way the approximate length of the shoes for a person."@en
2	<http://www.inria.fr/2007/09/11/humans.rdfs#shoesize>	"taille, exprimA©e en points, des chaussures d'une personne."@fr
3	<http://www.inria.fr/2007/09/11/humans.rdfs#shirtsize>	"express in some way the approximate dimensions of the shirts of a person."@en
4	<http://www.inria.fr/2007/09/11/humans.rdfs#shirtsize>	"dimensions approximatives des chemises portA©es par une personne."@fr
5	<http://www.inria.fr/2007/09/11/humans.rdfs#trouserssize>	"express in some way the approximate dimensions of the trousers of a person."@...
6	<http://www.inria.fr/2007/09/11/humans.rdfs#trouserssize>	"dimensions approximatives des pantalons portA©s par une personne."@fr

6. Write a query to find the properties that use the class Person in their signatures?

query:

```
prefix : <http://www.inria.fr/2007/09/11/humans.rdfs#>

select distinct ?x where {?x ?p :Person; rdf:type rdf:Property.}
```

```
1  prefix : <http://www.inria.fr/2007/09/11/humans.rdfs#>
2  select distinct ?x where {?x ?p :Person; rdf:type rdf:Property.}
3
```

Graph | XML/RDF | Table | Validate

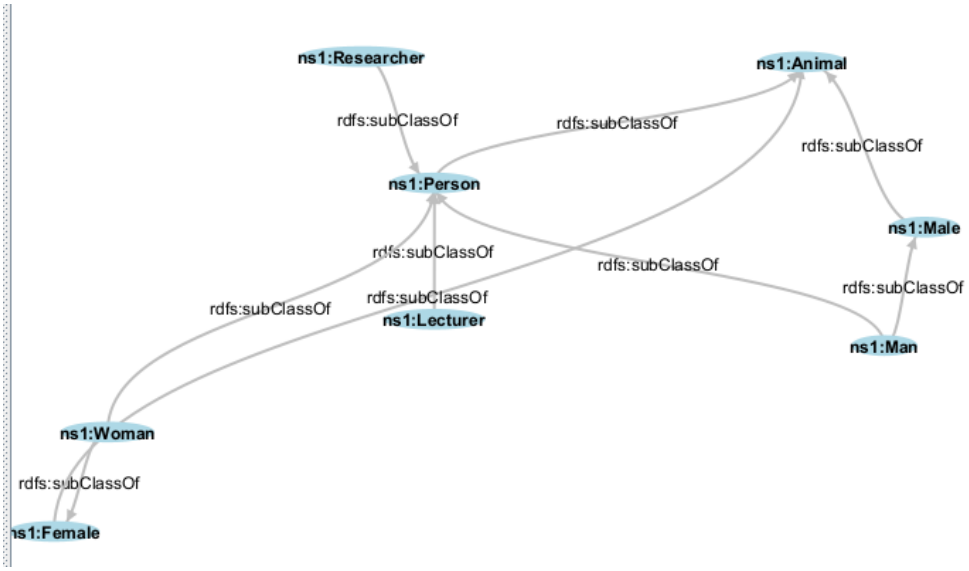
num	?x
1	<http://www.inria.fr/2007/09/11/humans.rdfs#hasSpouse>
2	<http://www.inria.fr/2007/09/11/humans.rdfs#shoesize>
3	<http://www.inria.fr/2007/09/11/humans.rdfs#shirtsize>
4	<http://www.inria.fr/2007/09/11/humans.rdfs#trouserssize>
5	<http://www.inria.fr/2007/09/11/humans.rdfs#hasFriend>

7. Make CORESE draw the graph of the hierarchy of Classes using a CONSTRUCT query considering only the classes in the humans.rdfs schema

query:

```
construct {?x rdfs:subClassOf ?y}

where {?x rdfs:subClassOf ?y}
```

screenshot:

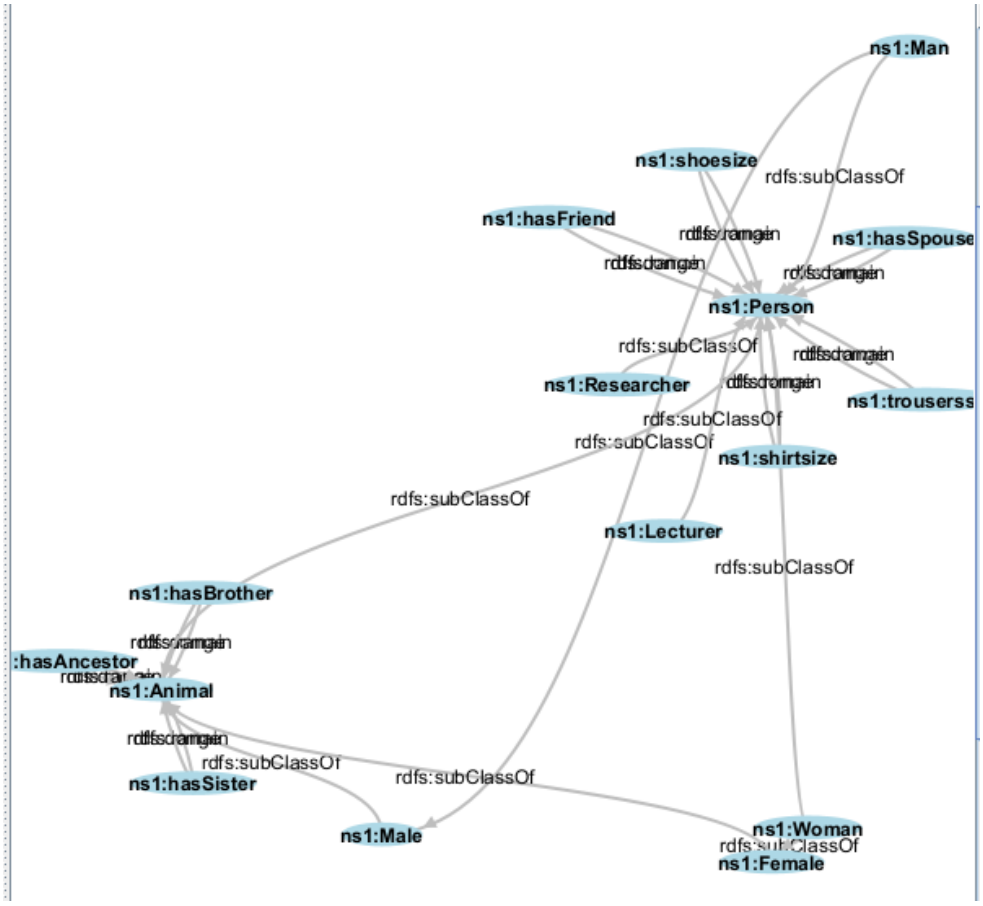


8. To the previous CONSTRUCT add the signatures of the relations.

query:

```
construct {?x rdfs:subClassOf ?y.
?p rdfs:domain ?c1.
?p rdfs:range ?c1.
}
where{?x rdfs:subClassOf ?y.
?p rdfs:domain ?c1.
?p rdfs:range ?c1.
}
```

screenshot:



You now know how to query schemas on the semantic Web!

Query data augmented by an RDFS schema

Question 1

1. Reset the Corese engine and load only the annotations (.rdf)
2. Write a query to find the Persons.

Query:

```
prefix : <http://www.inria.fr/2007/09/11/humans.rdfs#> .
select * where {?x a :Person}
```

Number of results before:

Number of results before = 8

3. Load the schema (.rdfs)
4. Rerun the query to find the Persons and explain the result.

New number of results after and your explanation:

Number of Persons after : 18.

Some persons were under class Lecturer, Man, Woman, Researcher and were not recognized as Person

When we load our predefined Schema in Corese, it automatically add in the class person to those objects who are under the class of Person.

The loading of the schema enabled the inferencing

Question 2

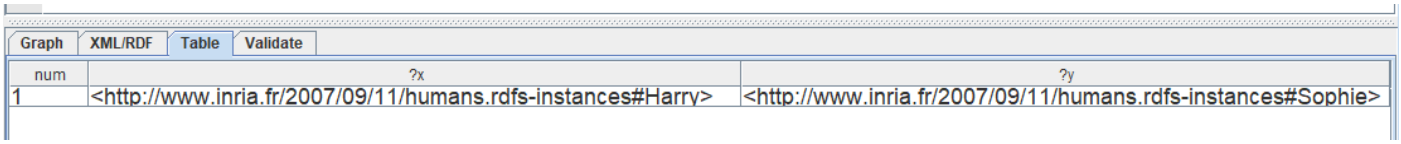
1. Write a query to find Males and their wives. How many answers do you get? Explain this result.

Query:

```
prefix : <http://www.inria.fr/2007/09/11/humans.rdfs#> .
select * where {?x a :Male; :hasSpouse ?y.}
```

Number of results and explanation:

We find only one result : Harry because he is a man, that makes him a peron and a male and his hasSpuse Sophie.



Graph | XML/RDF | Table | Validate

num	?x	?y
1	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Harry>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Sophie>

2. In the data declare that Lucas has to father Karl. Reset Corese, reload the ontology and the data, and then rerun the query to find Males and their wives. Explain the new result.

Line added in RDF:

```
<hasFather rdf:resource="#Karl"/>
```

Number of results before and after and explanation:

Now we have 2 results : Karl is added to the result of before with Harry.

He was a Person, we add him as Father of Lucas, so he is Parent : he become a Man and due to the definition of Man he is also a Male. And he has Spouse Catherine So he is shown in the results.

Question 3

1. Write a query to find the Lecturers and their types. How many answers do you get? See how this typing is declared in the data and explain the result.

Query:

```
prefix : <http://www.inria.fr/2007/09/11/humans.rdfs#> .

select * where {?x a :Lecturer; a ?y. filter(?y!=:Lecturer)}
```

Number of results and your explanation:

By removing the class Lecturer with the filter in the query, we get 5 results.

Laura is a Lecturer, so she is a Person and Person is a subclass of Animal. She is also Researcher.

But for EVE, I have a weird result : she is a Lecturer, so she is a Person but my result don't show the class Animal : even if Person is a subclass of Animal

```
prefix : <http://www.inria.fr/2007/09/11/humans.rdfs#> .
select * where {?x a :Lecturer; a ?y. filter(?y!=:Lecturer)}
```

Graph | XML/RDF | Table | Validate

num	?x	?y
1	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve>	<http://www.inria.fr/2007/09/11/humans.rdfs#Person>
2	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#Person>
3	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#Researcher>
4	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#Animal>
5	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#Female>

2. Write a query to find common instances of the classes Person and Male. See how this typing is declared in the data and explain the presence of Jack.

Query:

```
prefix : <http://www.inria.fr/2007/09/11/humans.rdfs#> .
select * where {?x a :Person; a :Male}
```

Your explanation of the result:

Graph | XML/RDF | Table | Validate

num	?x
1	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Harry>
2	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>
3	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>
4	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>
5	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>
6	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>
7	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>

In the data Jack was shown as a Man, while man is subclass of both person and male it also inherit those classes so he appears in the results

Question 4

Write a query to find the hasAncestor relations. Explain the result after checking where this property is used in the data.

Query:

```
prefix : <http://www.inria.fr/2007/09/11/humans.rdfs#> .

select * where {?x :hasAncestor ?y}
```

Your explanation of the result:

Graph | XML/RDF | Table | Validate

num	?x	?y
1	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Sophie>
2	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>
3	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Catherine>
4	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>
5	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Catherine>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>

The property hasParent is a sub Property of hasAncestor, and HasFather and hasMother are sub Properties of hasParent. When we have the relationship has parent the relationship has ancestor would be inherit

Question 5

1. Write a query to find the family cores (couples and their children) using a SELECT

Query:

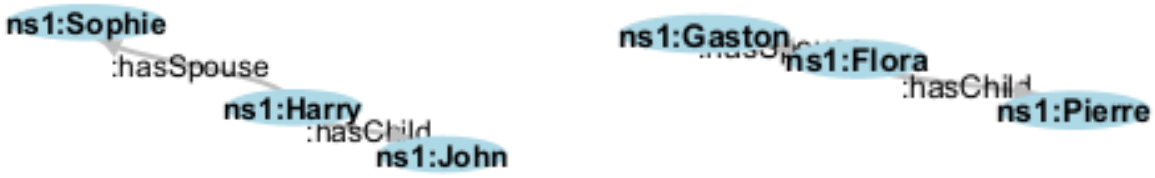
```
prefix : <http://www.inria.fr/2007/09/11/humans.rdfs#> .

select * where {?x :hasSpouse ?y; :hasChild ?z}
```

?x	?y	?z
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Harry>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Sophie>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Flora>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>

2. Modify it to display the result with a CONSTRUCT query

Query:

```
prefix : <http://www.inria.fr/2007/09/11/humans.rdfs#> .

construct {?x :hasSpouse ?y; :hasChild ?z}

where {?x :hasSpouse ?y; :hasChild ?z}
```



Question 6

1. Declare the olderThan relationship in the schema to indicate between two people which is eldest and construct the arcs between peoples with a SPARQL query

Addition to schema:

```
<rdf:Property rdf:ID="olderThan">

<domain rdf:resource="#Person"/>

<range rdf:resource="#Person"/>

</rdf:Property>
```

Query:

```
prefix : <http://www.inria.fr/2007/09/11/humans.rdfs#> .
Construct {?x :olderThan ?y} where {?x :age ?a1. ?y:age ?a2. filter(?a1>?a2)}
```

?x	?a1	?y	
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	37	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	14
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	37	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>	12
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	37	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>	36
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	37	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Safa>	27
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	14	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>	12
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>	102	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	37
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>	102	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	14
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>	102	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Flora>	95
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>	102	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>	71
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>	102	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>	12
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>	102	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>	36
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>	102	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#William>	42
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>	102	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Safa>	27
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Flora>	95	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	37
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Flora>	95	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	14
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Flora>	95	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>	71
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Flora>	95	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>	12
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Flora>	95	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>	36
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Flora>	95	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#William>	42
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Flora>	95	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Safa>	27
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>	71	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	37
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>	71	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	14
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>	71	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>	12
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>	71	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>	36
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>	71	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#William>	42
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>	71	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Safa>	27
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>	36	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	14
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>	36	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>	12
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>	36	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Safa>	27
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#William>	42	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	37
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#William>	42	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	14
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#William>	42	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>	12
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#William>	42	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>	36
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#William>	42	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Safa>	27
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Safa>	27	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	14
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Safa>	27	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>	12

2. Find a query that generates only the minimum number of links without redundancy with olderThan transitivity.

Query:

```
prefix : <http://www.inria.fr/2007/09/11/humans.rdfs#> .

select ?x ?y (min(?aged) as ?min) where { ?x :olderThan ?y; :age ?a1. ?y :age ?a2.

bind((?a1-?a2) as ?aged)}

group by ?x
```

?x	?y	?min
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Flora>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	24
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	7
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	1
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	9
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>	2
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	29
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Safa>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	13
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#William>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	5

Question 7

Write a query to find for John the properties which label contains the string "size" and the value of these properties.

Query:

```
select * where {<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>

?p ?o. ?p rdfs:label ?val.

filter (contains(?val, "size")) }
```

um	?p	?o	?val
	<http://www.inria.fr/2007/09/11/humans.rdfs#shirtsize>	12	"size"@en
	<http://www.inria.fr/2007/09/11/humans.rdfs#shirtsize>	12	"shirt size"@en
	<http://www.inria.fr/2007/09/11/humans.rdfs#shoesize>	14	"shoe size"@en
	<http://www.inria.fr/2007/09/11/humans.rdfs#shoesize>	14	"size"@en
	<http://www.inria.fr/2007/09/11/humans.rdfs#trouserssize>	44	"size"@en
	<http://www.inria.fr/2007/09/11/humans.rdfs#trouserssize>	44	"trousers size"@en

Question 8

Use the ontology to document your answers in natural language: write a query to find the types and properties of Laura in French.

Query:

```
prefix : <http://www.inria.fr/2007/09/11/humans.rdfs#> .
select ?y where {?x :name "Laura";?property ?type.
{{?property rdfs:label ?y.}
UNION
{?type rdfs:label ?y.}}
filter(lang(?y)="fr").}
```

Graph | XML/RDF | Table | Validate

num	
1	"a pour ami"@fr
2	"nom"@fr
3	"homme"@fr
4	"personne"@fr
5	"Aªtre humain"@fr
6	"humain"@fr
7	"professeur"@fr
8	"chercheur"@fr
9	"scientifique"@fr
10	"animal"@fr
11	"femelle"@fr

Day 04: Answers to the practical session on OWL.

Software requirements

- The RDF XML online validation service by W3C: https://www.w3.org/RDF/Validator/
- The RDF online translator: http://rdf-translator.appspot.com/
- The SPARQL Corese engine: https://project.inria.fr/corese/

A, Query data augmented by an OWL schema

Make a copy of the human.rdfs file, name it humans.owl and use it for the rest of the session. For each of the following statements, specify a SPARQL query that shows that the difference before and after running the OWL inferences: you will find that answers to these queries are different depending on whether you load the ontology humans.rdfs or the humans.owl you modified.

1. Declare that `hasSpouse` is a symmetrical property and do the same for and `hasFriend` .

Code added to the schema:

First, we needed to declare the namespace : "http://www.w3.org/2002/07/owl#"

```
<owl:SymmetricProperty rdf:ID="hasSpouse" />

<owl:SymmetricProperty rdf:ID="hasFriend" />
```

Query:

```
prefix : <http://www.inria.fr/2007/09/11/humans.rdfs#> .

select * where {?x ?p ?y. filter(?p= :hasFriend || ?p= :hasSpouse)}
```

Result before addition to the schema:

12 results : 6 for hasFriend and 6 for hasSpouse

Graph | XML/RDF | Table | Validate

num	?x	?p	?y
1	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasFriend>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Alice>
2	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#David>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasFriend>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>
3	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Alice>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasFriend>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>
4	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasFriend>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Alice>
5	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasFriend>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Alice>
6	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasFriend>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Sophie>
7	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Harry>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasSpouse>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Sophie>
8	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasSpouse>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#David>
9	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Flora>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasSpouse>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>
10	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jennifer>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasSpouse>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>
11	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasSpouse>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Catherine>
12	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#William>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasSpouse>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>

Result after addition to the schema:

24results : 12 for hasFriend and 12 for hasSpouse

Graph | XML/RDF | Table | Validate

num	?x	?p	?y
1	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasFriend>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Alice>
2	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Sophie>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasFriend>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>
3	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasFriend>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Alice>
4	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#David>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasFriend>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>
5	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Alice>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasFriend>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>
6	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Alice>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasFriend>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve>
7	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Alice>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasFriend>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>
8	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Alice>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasFriend>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>
9	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasFriend>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#David>
10	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasFriend>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Alice>
11	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasFriend>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Alice>
12	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasFriend>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Sophie>
13	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Harry>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasSpouse>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Sophie>
14	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasSpouse>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jennifer>
15	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Sophie>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasSpouse>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Harry>
16	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasSpouse>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#David>
17	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#David>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasSpouse>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve>
18	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasSpouse>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Flora>
19	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Flora>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasSpouse>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>
20	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasSpouse>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#William>
21	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jennifer>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasSpouse>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>
22	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Catherine>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasSpouse>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>
23	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasSpouse>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Catherine>
24	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#William>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasSpouse>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>

Explanation:

The number of results doubled because we made the relationship hasFriend and hasSpouse symmetric. For example, Eve hasFriend Alice : before the modification, Alice didn't have friend Eve but after the modification, the relationship is on both ways. It is the same for all the others: for each relation, we created the relation in the inverse way

2. Declare that `hasChild` is the inverse property of the `hasParent` property.

Code added to the schema:

```
<rdf:Property rdf:ID="hasChild">

<owl:inverseOf rdf:resource="#hasParent" />

</rdf:Property>
```

Query:

```
prefix : <http://www.inria.fr/2007/09/11/humans.rdfs#> .

select * where { ?x ?p ?y. filter(?p= :hasChild || ?p= :hasParent )}
```

Result before addition to the schema:

able | Validate

?x	?p	?y
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Harry>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasChild>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasChild>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasChild>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasChild>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Harry>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Flora>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasChild>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasParent>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Sophie>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasParent>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasParent>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Catheri...
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasParent>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Catheri...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasParent>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>

Result after addition to the schema:

F | Table | Validate

?x	?p	?y
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Harry>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasChild>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasChild>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Sophie>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasChild>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasChild>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasChild>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasChild>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Harry>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Flora>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasChild>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasChild>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Catherine>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Catherine>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasChild>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasChild>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Harry>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasParent>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasParent>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Harry>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasParent>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Sophie>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasParent>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasParent>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasParent>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasParent>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Flora>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasParent>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Catherine>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasParent>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Catherine>	<http://www.inria.fr/2007/09/11/humans.rdfs#hasParent>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>

Explanation:

The modification we did means that if we only have has child, the other relationship has parent would be created. Likewise if we only have has Parent relation then has child inverse would be setup.

Example Harry :

- He has child John
- And John has parent Sophie.
- Then the new query John would have parent Harry
- And Sophie would has child John.

3. Declare `hasAncestor` as transitive property.

Code added to the schema:

```
<owl:TransitiveProperty rdf:ID="hasAncestor">

</owl:TransitiveProperty>
```

Query:

```
prefix : <http://www.inria.fr/2007/09/11/humans.rdfs#> .

select * where {?x :hasAncestor ?y}
```

Result before addition to the schema:

?x	?y
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Sophie>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Catherine>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Catherine>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>

Result after addition to the schema:

ate

?x	?y
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Harry>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Harry>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Sophie>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Flora>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Catherine>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Catherine>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>

Explanation:

Transitive property is that the parents of our parent would be our ancestor too even if we didn't specify it.

Mark has ancestor John and John has ancestor Sophie. Now Mark has ancestor Sophie too (also Harry because from the last exercise we get that harry is John's father)

4. Declare the disjunction between `Male` and `Female`. Violate the constraint in the data, check the results and then remove the violation you created.

Code added to the schema:

```
<rdf:Class rdf:ID="Male">

<owl:disjointWith rdf:resource="#Female" />

</rdf:Class>
```

Query:

```
prefix : <http://www.inria.fr/2007/09/11/humans.rdfs#> .
SELECT ?z ?y
WHERE {?x ?z ?y. ?y :name "Jack"}
```

Result before addition to the schema:

?z	?y
<http://www.inria.fr/2007/09/11/humans.rdfs#hasAncestor>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>
<http://www.inria.fr/2007/09/11/humans.rdfs#hasChild>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>
<http://www.inria.fr/2007/09/11/humans.rdfs#hasFriend>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>
<http://www.inria.fr/2007/09/11/humans.rdfs#hasParent>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>

Result after addition to the schema:

?z	?y
sp:violationRoot	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>
<http://www.inria.fr/2007/09/11/humans.rdfs#hasAncestor>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>
<http://www.inria.fr/2007/09/11/humans.rdfs#hasAncestor>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>
<http://www.inria.fr/2007/09/11/humans.rdfs#hasAncestor>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>
<http://www.inria.fr/2007/09/11/humans.rdfs#hasChild>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>
<http://www.inria.fr/2007/09/11/humans.rdfs#hasFriend>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>
<http://www.inria.fr/2007/09/11/humans.rdfs#hasParent>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>

Explanation:

```
<Man rdf:ID="Jack">
 <rdf:type rdf:resource="&humans;#Female"/>
     <hasFriend rdf:resource="#Alice"/>
     <hasChild rdf:resource="#Harry"/>
     <name>Jack</name>
</Man>
```

I have added Female to Jack, who was a Man (that would auto inherit Person, Male and Animal class). Which violate the rule we set up about the disjointWith in the schema. We can query and see the sp:violationRoot result.

5. Declare that the class `Professor` is the intersection of the class `Lecturer` and `Researcher` class.

Code added to the schema:

```
<Class rdf:ID="Professor">

<owl:intersectionOf rdf:parseType="Collection">

<Class rdf:about="#Researcher"/>

<Class rdf:abou="#Lecturer"/>

</owl:intersectionOf>

</Class>
```

Query:

```
prefix : <http://www.inria.fr/2007/09/11/humans.rdfs#> .

select * where {?x a :Lecturer; a ?y}
```

Result before addition to the schema:

?x	?y
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve>	<http://www.inria.fr/2007/09/11/humans.rdfs#Person>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve>	<http://www.inria.fr/2007/09/11/humans.rdfs#Lecturer>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#Person>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#Lecturer>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#Researcher>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#Animal>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#Female>

Result after addition to the schema:

?x	?y
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve>	<http://www.inria.fr/2007/09/11/humans.rdfs#Person>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve>	<http://www.inria.fr/2007/09/11/humans.rdfs#Lecturer>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve>	<http://www.inria.fr/2007/09/11/humans.rdfs#Animal>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#Person>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#Lecturer>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#Researcher>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#Animal>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#Female>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#Professor>

```
prefix : <http://www.inria.fr/2007/09/11/humans.rdfs#> .
select * where {?x a :Professor}
```

raph | XML/RDF | Table | Validate

num	?x
	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>

Explanation:

We defined Professor as the intersection of Lecturer AND researcher: so the person must have them both to be a Professor (not the case of Eve). Here, it is the case of Laura.

6. Declare that the `Academic` class is the union of classes `Lecturer` and `Researcher`.

Code added to the schema:

```
<Class rdf:ID="Academic">

<owl:unionOf rdf:parseType="Collection">

<Class rdf:about="#Lecturer"/>

<Class rdf:about="#Researcher"/>

</owl:unionOf>

</Class>
```

Query:

```
prefix : <http://www.inria.fr/2007/09/11/humans.rdfs#> .
select * where {?x a ?y. filter (?y = :Lecturer || ?y = :Researcher)}
```

Result before addition to the schema:

?x	?y
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve>	<http://www.inria.fr/2007/09/11/humans.rdfs#Lecturer>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#David>	<http://www.inria.fr/2007/09/11/humans.rdfs#Researcher>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>	<http://www.inria.fr/2007/09/11/humans.rdfs#Researcher>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#Lecturer>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#Researcher>

Result after addition to the schema:

```
prefix : <http://www.inria.fr/2007/09/11/humans.rdfs#> .
select * where {?x a :Academic}
```

aph | XML/RDF | Table | Validate

num	?x
	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve>
	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#David>
	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>
	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>

Explanation:

Academic (union of Lecturer and Researcher) was added to the relationship but we don't see unless we look for it

7. Create a class `Organization` and its sub class `University`. Create a new property `mainEmployer`, with domain `Person` and range `Organization`. Use a restriction to declare that any `Professor` has for main employer a `University`.

Code added to the schema (new property, new classes and new restriction):

```
<Class rdf:ID="Professor">
<owl:intersectionOf rdf:parseType="Collection">
<Class rdf:about="#Lecturer"/>
<Class rdf:about="#Researcher"/>
</owl:intersectionOf>
<subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#mainEmployer" />
<owl:allValuesFrom rdf:resource="#University" />
</owl:Restriction>
</subClassOf>
</Class>

<Class rdf:ID="Organization">
</Class>

<Class rdf:ID="University">
<subClassOf rdf:resource="#Organization"/>
</Class>

<rdf:Property rdf:ID="mainEmployer">
<domain rdf:resource="#Person"/>
<range rdf:resource="#Organization"/>
</rdf:Property>
```

Code added to the data (just declare the main employer of a Professor):

```
<Professor rdf:ID="#Laura">
<mainEmployer rdf:resource="#DSTI"/>
</Professor>
```

Query:

```
prefix : <http://www.inria.fr/2007/09/11/humans.rdfs#> .

select * where {?x :mainEmployer ?y. optional {?y a ?z}}
```

Result before addition to the schema:

?x	?y	?z
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#DSTI>	

Result after addition to the schema:

XML/RDF | Table | Validate

?x	?y	?z
<http://www.inria.fr/2007/09/11/humans.rdfs-instances##Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#DSTI>	<http://www.inria.fr/2007/09/11/humans.rdfs#University>
<http://www.inria.fr/2007/09/11/humans.rdfs-instances##Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#DSTI>	<http://www.inria.fr/2007/09/11/humans.rdfs#Organization>

Explanation:

Laura is a researcher and lecturer so she is a professor. By adding the restriction we can tell that all the mainEmployer of the Laura should be a class university and Organization.

8. Use a restriction to declare that any person must have a parent who is a woman. For this last statement, you need to run the rule engine after loading the ontology and data.

Code added to the schema:

```
<Class rdf:ID="Person">

<subClassOf>

<owl:Restriction>

<owl:onProperty rdf:resource="#hasParent" />

<owl:someValuesFrom rdf:resource="#Woman" />

</owl:Restriction>

</subClassOf>

</Class>
```

Query:

```
prefix : <http://www.inria.fr/2007/09/11/humans.rdfs#> .

select ?y where {?x a ?y; :name "Sarah". }
```

Result before addition to the schema:

I added one resource to the data for this example : “Sarah” as a Female, has Mother Catherine that is a Woman

```
<Woman rdf:ID="Catherine">
  <hasMother rdf:resource="#Laura"/>
</Woman>

<Female rdf:ID="Sarah">
  <name>Sarah</name>
  <hasMother rdf:resource="#Catherine"/>
</Female>
```

Results :

```
1  prefix : <http://www.inria.fr/2007/09/11/humans.rdfs#> .
2  select ?y where {?x a ?y; :name "Sarah". }
3
```

Graph | XML/RDF | Table | Validate

num	?y
1	<http://www.inria.fr/2007/09/11/humans.rdfs#Female>
2	<http://www.inria.fr/2007/09/11/humans.rdfs#Animal>

Result after addition to the schema:

num	?y
	<http://www.inria.fr/2007/09/11/humans.rdfs#Female>
	<http://www.inria.fr/2007/09/11/humans.rdfs#Person>
	<http://www.inria.fr/2007/09/11/humans.rdfs#Animal>

Explanation:

Before adding the restriction to the schema, Sarah was just a Female, that's a subClassOf Animal but after adding the restriction, she become also a Person.

B, Make your own OWL models:

For each one of the following OWL primitives imagine a definition that could use it and provide that definition in OWL using your preferred syntax (RDF/XML or N3/Turtle). For instance a possible definition using owl:TransitiveProperty would be a definition of the Ancestor property. For each primitive in the following list you imagine the definition of a class or property that was not given in the course and you give that definition in English and in OWL.

1. owl:oneOf <YOUR EXAMPLE HERE/>
2. owl:unionOf <YOUR EXAMPLE HERE/>
3. owl:intersectionOf <YOUR EXAMPLE HERE/>
4. owl:complementOf <YOUR EXAMPLE HERE/>
5. owl:disjointWith <YOUR EXAMPLE HERE/>
   or owl:AllDisjointClasses
   or owl:disjointUnionOf
6. owl:ObjectProperty <YOUR EXAMPLE HERE/>
7. owl:DatatypeProperty <YOUR EXAMPLE HERE/>
8. owl:SymmetricProperty <YOUR EXAMPLE HERE/>
   or owl:AsymmetricProperty
9. owl:inverseOf <YOUR EXAMPLE HERE/>
10. owl:TransitiveProperty <YOUR EXAMPLE HERE/>
11. owl:propertyDisjointWith <YOUR EXAMPLE HERE/>
12. owl:ReflexiveProperty <YOUR EXAMPLE HERE/>
    or owl:IrreflexiveProperty
13. owl:propertyChainAxiom <YOUR EXAMPLE HERE/>
14. owl:FunctionalProperty <YOUR EXAMPLE HERE/>
15. owl:InverseFunctionalProperty <YOUR EXAMPLE HERE/>
16. owl:hasKey <YOUR EXAMPLE HERE/>
17. owl:allValuesFrom <YOUR EXAMPLE HERE/>
18. owl:someValuesFrom <YOUR EXAMPLE HERE/>
19. owl:hasValue <YOUR EXAMPLE HERE/>
20. owl:maxCardinality <YOUR EXAMPLE HERE/>
    or owl:minCardinality
21. owl:qualifiedCardinality <YOUR EXAMPLE HERE/>