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: Motoharu

Family name: DEI

Email: motoharu.dei@edu.dsti.institute

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


```
{"entities":{"Q23014205":{"pageid":25028548,"ns":0,"title":"Q23014205","lastrevid
":872123137,"modified":"2019-03-
02T12:03:45Z","type":"item","id":"Q23014205","labels":{"fr":{"language":"fr","val
ue":"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 Gandon"},"de-ch":{"language":"de-ch","value":"Fabien
Gandon"},"en-ca":{"language":"en-ca","value":"Fabien Gandon"},"en-
gb":{"language":"en-gb","value":"Fabien
Gandon"},"eo":{"language":"eo","value":"Fabien
Gandon"},"es":{"language":"es","value":"Fabien
Gandon"},"et":{"language":"et","value":"Fabien
Gandon"},"eu":{"language":"eu","value":"Fabien
Gandon"},"fi":{"language":"fi","value":"Fabien
```

```
Gandon"},"frc":{"language":"frc","value":"Fabien
Gandon"},"frp":{"language":"frp","value":"Fabien
Gandon"},"fur":{"language":"fur","value":"Fabien
Gandon"},"ga":{"language":"ga","value":"Fabien
Gandon"},"gd":{"language":"gd","value":"Fabien
Gandon"},"gl":{"language":"gl","value":"Fabien
Gandon"},"gsw":{"language":"gsw","value":"Fabien
Gandon"},"hr":{"language":"hr","value":"Fabien
Gandon"},"hu":{"language":"hu","value":"Fabien
Gandon"},"ia":{"language":"ia","value":"Fabien
Gandon"},"id":{"language":"id","value":"Fabien
Gandon"},"ie":{"language":"ie","value":"Fabien
Gandon"},"io":{"language":"io","value":"Fabien
Gandon"},"it":{"language":"it","value":"Fabien
Gandon"},"jam":{"language":"jam","value":"Fabien
Gandon"},"kab":{"language":"kab","value":"Fabien Gandon"},"kg":
```

```
{"entities":{"Q23014205":{"pageid":25028548,"ns":0,"title":"Q23014205","lastrevid
":872123137,"modified":"2019-03-
02T12:03:45Z","type":"item","id":"Q23014205","labels":{"fr":{"language":"fr","val
ue":"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 Gandon"},"de-ch":{"language":"de-ch","value":"Fabien
Gandon"},"en-ca":{"language":"en-ca","value":"Fabien Gandon"},"en-
gb":{"language":"en-gb","value":"Fabien
Gandon"},"eo":{"language":"eo","value":"Fabien
Gandon"},"es":{"language":"es","value":"Fabien
Gandon"},"et":{"language":"et","value":"Fabien
Gandon"},"eu":{"language":"eu","value":"Fabien
Gandon"},"fi":{"language":"fi","value":"Fabien
Gandon"},"frc":{"language":"frc","value":"Fabien
Gandon"},"frp":{"language":"frp","value":"Fabien
Gandon"},"fur":{"language":"fur","value":"Fabien
Gandon"},"ga":{"language":"ga","value":"Fabien
Gandon"},"gd":{"language":"gd","value":"Fabien
Gandon"},"gl":{"language":"gl","value":"Fabien
Gandon"},"gsw":{"language":"gsw","value":"Fabien
Gandon"},"hr":{"language":"hr","value":"Fabien
Gandon"},"hu":{"language":"hu","value":"Fabien
Gandon"},"ia":{"language":"ia","value":"Fabien
Gandon"},"id":{"language":"id","value":"Fabien
Gandon"},"ie":{"language":"ie","value":"Fabien Gandon"},"io":
```

```
<?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:v="http://www.wikidata.org/value/" 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:wdt="http://www.wikidata.org/prop/direct/" xmlns:wdtn="http://www.wikidata.org/prop/direct-
normalized/" xmlns:p="http://www.wikidata.org/prop/"
xmlns:ps="http://www.wikidata.org/prop/statement/"
xmlns:psv="http://www.wikidata.org/prop/statement/value/"
xmlns:psn="http://www.wikidata.org/prop/statement/value-normalized/"
xmlns:pq="http://www.wikidata.org/prop/qualifier/"
xmlns:pqv="http://www.wikidata.org/prop/qualifier/value/"
xmlns:pqn="http://www.wikidata.org/prop/qualifier/value-normalized/"
xmlns:pr="http://www.wikidata.org/prop/reference/"
xmlns:prv="http://www.wikidata.org/prop/reference/value/"
xmlns:prn="http://www.wikidata.org/prop/reference/value-normalized/"
xmlns:wdno="http://www.wikidata.org/prop/novalue/">
        <rdf:Description rdf:about="https://www.wikidata.org/wiki/Special:EntityData/Q23014205">
                <rdf:type rdf:resource="http://schema.org/Dataset"/>
                <schema:about rdf:resource="http://www.wikidata.org/entity/Q23014205"/>
                <cc:license rdf:resource="http://creativecommons.org/publicdomain/zero/1.0/"/>
                <schema:softwareVersion>1.0.0</schema:softwareVersion>
                <schema:version
rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">872123137</schema:version>
                <schema:dateModified
rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2019-03-
02T12:03:45Z</schema:dateModified>
                <wikibase:statements
rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">29</wikibase:statements>
```

```
@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#> .
```

```
<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>
"872123137"^^<http://www.w3.org/2001/XMLSchema#integer> .
<https://www.wikidata.org/wiki/Special:EntityData/Q23014205> <http://schema.org/dateModified> "2019-
03-02T12:03:45Z"^^<http://www.w3.org/2001/XMLSchema#dateTime> .
```

<https://www.wikidata.org/wiki/Special:EntityData/Q23014205> <http://wikiba.se/ontology#statements> "29"^^<http://www.w3.org/2001/XMLSchema#integer> .
<https://www.wikidata.org/wiki/Special:EntityData/Q23014205> <http://wikiba.se/ontology#identifiers> "10"^^<http://www.w3.org/2001/XMLSchema#integer> .
<https://www.wikidata.org/wiki/Special:EntityData/Q23014205> <http://wikiba.se/ontology#sitelinks> "0"^^<http://www.w3.org/2001/XMLSchema#integer> .
<http://www.wikidata.org/entity/Q23014205> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://wikiba.se/ontology#Item> .

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. Done
2. Londoner
3. Done
4. 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. Florian DUFOUR
2. REDACTED FOR PRIVACY
3. SOCIETE EDITRICE DU MONDE

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.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN" "http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
   xmlns:dbpprop="http://dbpedia.org/property/"
   xmlns:foaf="http://xmlns.com/foaf/0.1/"
   version="XHTML+RDFa 1.0"
   xml:lang="en"
>
```

```
<?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:dbo="http://dbpedia.org/ontology/"
        xmlns:dct="http://purl.org/dc/terms/"
        xmlns:foaf="http://xmlns.com/foaf/0.1/"
        xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
        xmlns:prov="http://www.w3.org/ns/prov#"
```

2.

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
 <head>
  <meta http-equiv="Content-Type" content="text/html;charset=utf-8" />
  <title>FOAF profile of Fabien GANDON</title>
 </head>
<body>
 <h1>FOAF profile of Fabien GANDON</h1>
 <p>You may have been redirected here by your browser.</p>
```

```
<?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"/>
```

3.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN" "http://www.w3.org/MarkUp/DTD/xhtml-
rdfa-1.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
   xmlns:dbpprop="http://dbpedia.org/property/"
   xmlns:foaf="http://xmlns.com/foaf/0.1/"
   version="XHTML+RDFa 1.0"
   xml:lang="en"
>
```

```
<?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:dbo="http://dbpedia.org/ontology/"
        xmlns:dbp="http://dbpedia.org/property/"
        xmlns:dct="http://purl.org/dc/terms/"
        xmlns:foaf="http://xmlns.com/foaf/0.1/"
        xmlns:skos="http://www.w3.org/2004/02/skos/core#"
```

4.

```
<!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/RecordInfo#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-
schema#" xmlns:skos="http://www.w3.org/2004/02/skos/core#"
xmlns:skosxl="http://www.w3.org/2008/05/skos-xl#" xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:cs="http://www.w3.org/2003/06/sw-vocab-status/ns#" xmlns:dcterms="http://purl.org/dc/terms/">
 <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 Congress. This includes data values and the controlled vocabularies that house them. Datasets available include LCSH, BIBFRAME, LC Name Authorities, LC Classification, MARC codes, PREMIS vocabularies, ISO language codes, and more."/>
<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"/>

<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 xml:lang="en">Hugo, Victor, 1802-1885</madsrdf:authoritativeLabel>
<madsrdf:elementList rdf:parseType="Collection">
<madsrdf:FullNameElement>
<madsrdf:elementValue xml:lang="en">Hugo, Victor,</madsrdf:elementValue>
</madsrdf:FullNameElement>
<madsrdf:DateNameElement>
<madsrdf:elementValue xml:lang="en">1802-1885</madsrdf:elementValue>

5.

<!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) - MCAM gene & protein</title><meta content="IE=edge" http-equiv="X-UA-Compatible"/><meta content="text/html; charset=UTF-8" http-equiv="Content-Type"/><meta content="width=device-width, initial-scale=1" name="viewport"/><link href="/" rel="home"/><link href="https://creativecommons.org/licenses/by/4.0/" rel="license"/><link type="image/vnd.microsoft.icon" href="/favicon.ico" rel="shortcut icon"/><link href="/uniprot.min.css2019_08" type="text/css" rel="stylesheet"/><script type="text/javascript">
var BASE = '/';
var ua = window.navigator.userAgent;
var directory = (~ua.indexOf('MSIE ') || ~ua.indexOf('Trident/')) === 0 ? "non-ie" : "ie";
</script><script src="/scripts/frontier/d3/d3.v3.min.js" type="text/javascript"></script><script src="/js-compr.js2019_08" type="text/javascript"></script><script type="text/javascript">
uniprot.namespace = 'uniprot';
uniprot.releasedate = '2019_08';
</script><script type="text/javascript">
;

<?xml version='1.0' encoding='UTF-8'?>
<rdf:RDF xml:base="http://purl.uniprot.org/uniprot/" xmlns="http://purl.uniprot.org/core/" xmlns:dcterms="http://purl.org/dc/terms/" 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:skos="http://www.w3.org/2004/02/skos/core#" xmlns:bibo="http://purl.org/ontology/bibo/"
xmlns:foaf="http://xmlns.com/foaf/0.1/" xmlns:void="http://rdfs.org/ns/void#"
xmlns:sd="http://www.w3.org/ns/sparql-service-description#"
xmlns:faldo="http://biohackathon.org/resource/faldo#">
<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"/>
<reviewed rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean">true</reviewed>
<created rdf:datatype="http://www.w3.org/2001/XMLSchema#date">1995-11-01</created>
<modified rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2019-09-18</modified>

6.

About Xavier Dolan; a Canadian film director, actor, voice actor and screenwriter; stored in JSON format.

7.

About a book titled “Distributed Artificial Intelligence for Distributed Corporate Knowledge Management”; stored in Terse RDF Triple Language (Turtle) format.

Q1.6 Recall five best practices of linked open data

☆ On the web

☆☆ Machine-readable data

☆☆☆ Non-proprietary format

☆☆☆☆ RDF standards

☆☆☆☆☆ 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.

1.

The Resource Description Framework (RDF) is a family of World Wide Web Consortium (W3C) specifications[1] originally designed as a metadata data model. It has come to be used as a general method for conceptual description or modeling of information that is implemented in web resources, using a variety of syntax notations and data serialization formats. It is also used in knowledge management applications. RDF was adopted as a W3C recommendation in 1999. The RDF 1.0 specification was published in 2004, the RDF 1.1 specification in 2014.

2. Done

3.

The Resource Description Framework (RDF) is a family of World Wide Web Consortium (W3C) specifications[1] originally designed as a metadata data model. It has come to be used as a general method for conceptual description or modeling of information that is implemented in web resources, using a variety of syntax notations and data serialization formats. It is also used in knowledge management applications. RDF was adopted as a W3C recommendation in 1999. The RDF 1.0 specification was published in 2004, the RDF 1.1 specification in 2014.

4.

The Shawshank Redemption is a 1994 American drama film written and directed by Frank Darabont, based on the 1982 Stephen King novella Rita Hayworth and Shawshank Redemption. It tells the story of banker Andy Dufresne (Tim Robbins), who is sentenced to life in Shawshank State Penitentiary for the murders of his wife and her lover, despite his claims of innocence. Over the following two decades, he befriends a fellow prisoner, contraband smuggler Ellis "Red" Redding (Morgan Freeman), and becomes instrumental in a money laundering operation led by the prison warden Samuel Norton (Bob Gunton). William Sadler, Clancy Brown, Gil Bellows, and James Whitmore appear in supporting roles.

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)

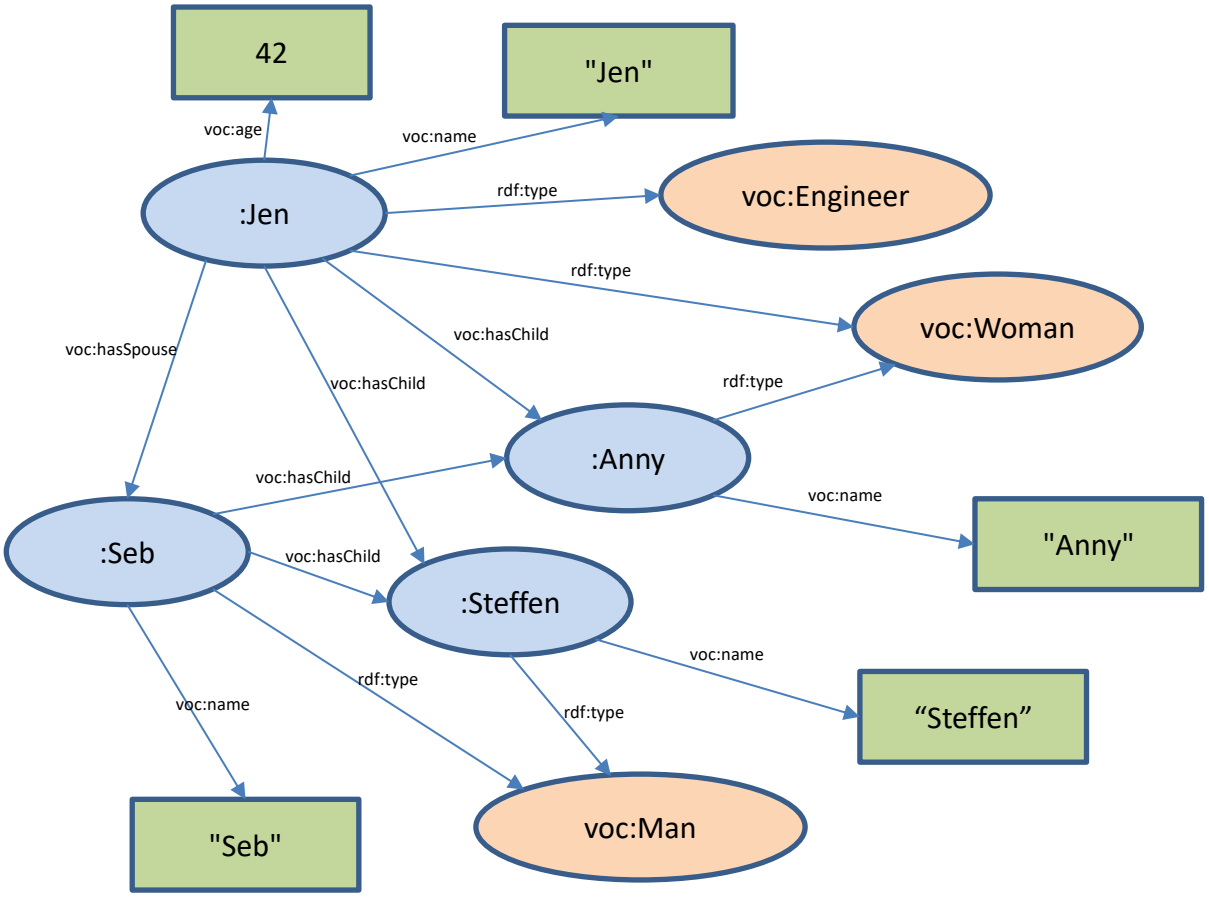
Directed labeled multigraphs: edges have direction starting from node to another node, every node and edge has to have labels, allowing multiple edges between nodes.

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, :Stefan, 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"?>
<?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)

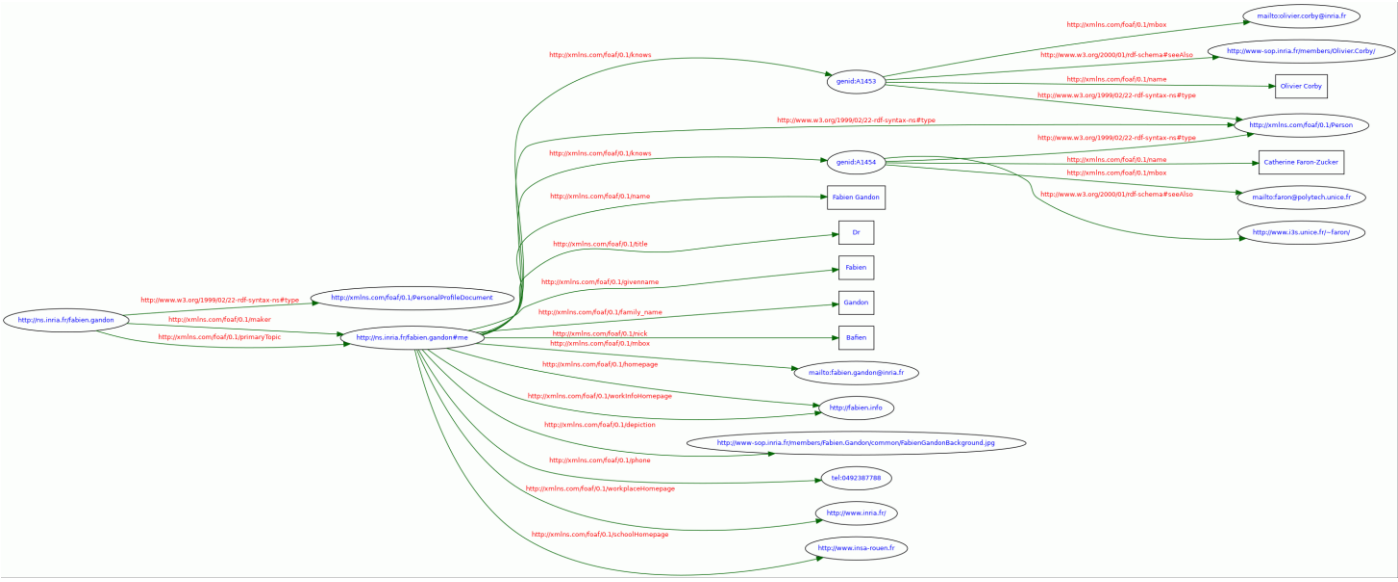
```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix voc: <http://www.unice.fr/voc#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://www.unice.fr/data#Jen> a voc:Engineer, voc:Woman;
    voc:age "42"^^xsd:string ;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

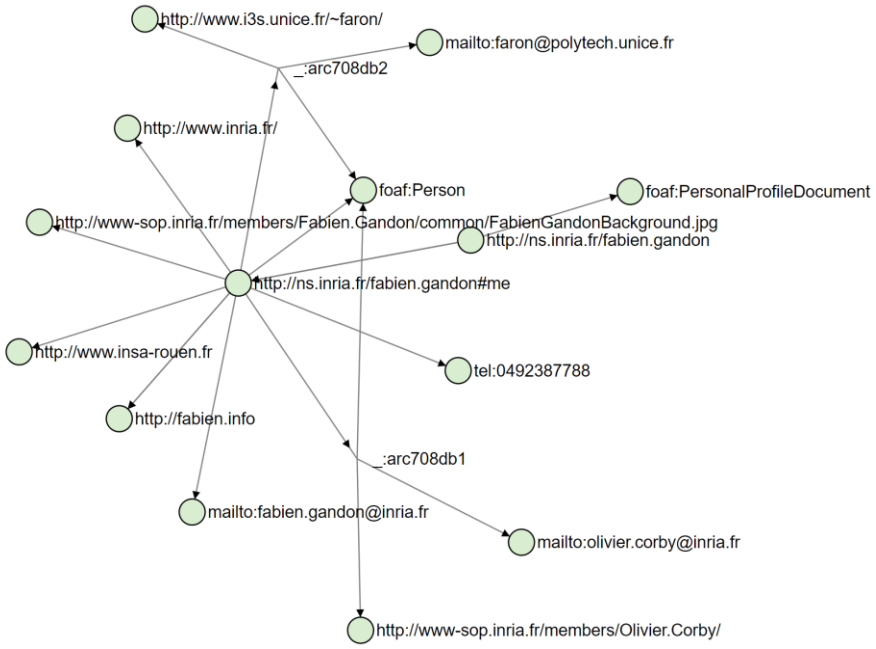
1. Done
2. Xml
3. 

4.

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<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/> .
```

5.

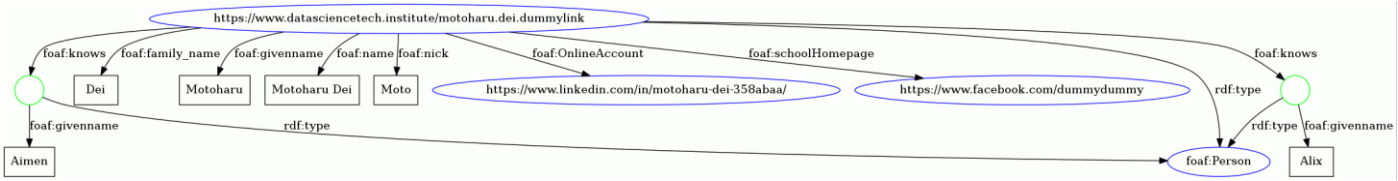


6.

In Turtle syntax:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .

<https://www.datasciencetech.institute/motoharu.dei.dummylink> a foaf:Person ;
        foaf:family_name "Dei";
        foaf:givenname "Motoharu";
        foaf:name "Motoharu Dei";
        foaf:nick "Moto";
        foaf:OnlineAccount <https://www.linkedin.com/in/motoharu-dei-358abaa/>;
        foaf:schoolHomepage <https://www.facebook.com/dummydummy>;
        foaf:knows [ a foaf:Person ;
                        foaf:givenname "Aimen"],
                    [ a foaf:Person ;
                     foaf:givenname "Alix"] .
```

Graph:



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>
```

http://example.org/doc.html is a report about music and dance with 73 pages.

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?

1. Done
2. Done
3. Xml
4. Done (too long to copy here)
5. Every language in citation-note is English.

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/topics/algebra>
{
  <http://inria.fr/rr/doc.html>
    dcterms:subject
      <http://data.bnf.fr/ark:/12148/cb121105993> .
}
```

A TriG representing that a triple (http://inria.fr/rr/doc.html, dcterms:subject, http://data.bnf.fr/ark:/12148/cb121105993) belongs to a graph http://inria.fr/topics/algebra.

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?

1. Done
2. Done
3. Xml
4. Done (too long to copy)
5. There are a lot of repetitions of statements, due to older version of xml.

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 . }
```

	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
```

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.wikidata.org/entity/Q486972>
:Paris	rdf:type	dbpedia:ontology/PopulatedPlace
:Paris	rdf:type	dbpedia:ontology/Settlement
:Paris	rdf:type	<http://www.w3.org/2003/01/geo/wgs84_pos#SpatialThing>
:Paris	rdf:type	<http://schema.org/Place>
:Paris	rdf:type	<http://umbel.org/umbel/rc/Location_Underspecified>
:Paris	rdf:type	<http://umbel.org/umbel/rc/PopulatedPlace>

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

wd:Q30 represents “United States of America”

p:P6 represents “head of government”

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. Find the age of resources whose name is ‘Fabien’

```
SELECT ?age WHERE {
        ?x c:name “Fabien” ;
           c:age ?age .
}
```

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

```
SELECT ?name WHERE {
        ?x c:name ?name ;
   c:age ?age .
        FILTER (?age < 50)
}
```

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

```
SELECT ?p ?y WHERE {
        ?x c:name “Fabien” ;
           c:age ?age ;
           ?p ?y .
        FILTER (?age < 50)
}
```

4. Find other names of resources whose name is ‘Fabien’

```
SELECT ?name WHERE {
        ?x c:name “Fabien” ;
           c:name ?name.
        FILTER (?name != “Fabien”)
}
```

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

```
SELECT * WHERE {
        ?x ?p1 ?y;
           ?p2 ?y .
        FILTER (?p1 != ?p2)
}
```

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

```
SELECT * WHERE {
        ?x ?p ?y1, ?y2 .
        FILTER (?y1 != ?y2)
}
```

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 }
}
```

No. There’s no common variable before and after minus.

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))
```

How many documents each person writes by descending order of the number of documents.

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

?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 }
}
```

The list of friends' names of each person, allowing filling the missing name with "John Doe".

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
```

The average age of the friends of each person.

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" -> ex:g1
- "graph ?g { ?y ?q ?z }" -> ex:g2

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

DELETE {?x foaf:name ?y}
INSERT {?x rdfs:label ?y}
WHERE {?x foaf:name ?y}

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)
  }
}
```

Converting age to integer if it is string.

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 {…} …

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

2nd, 3rd, 4th, 5th are true.

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

1st, 3rd, 4th are true.

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 .
```

It defines:

- Phone is a subclass of device.
- Computer is a subclass of device.
- Smartphone is a subclass of computer.
- Smartphone is a subclass of phone.

We can derive:

- Smartphone is a subclass of device. (Transitivity of subsumption)

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 .
```

It defines:

- EmployeeOf is a subproperty of proRelationWith
- hasControlOver is a subproperty of proRelationWith
- isShareholderOf is a subproperty of hasControlOver
- isCEOof is a subproperty of employeeOf and hasControlOver

We can derive:

- isCEOof is a subproperty of proRelationWith (Transitivity of subsumption)
- isShareholderOf is a subproperty of proRelationWith (Transitivity of subsumption)

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 .
```

isControling is a property from Human to Object. driverOf is a subproperty of isControling from unanimous type of resources to Car, but effective Car resource also has to be Object. pilotOf is a subproperty of isControling from Adult to Plane, but effective Adult also has to be Human and effective Plane also has to be Object.

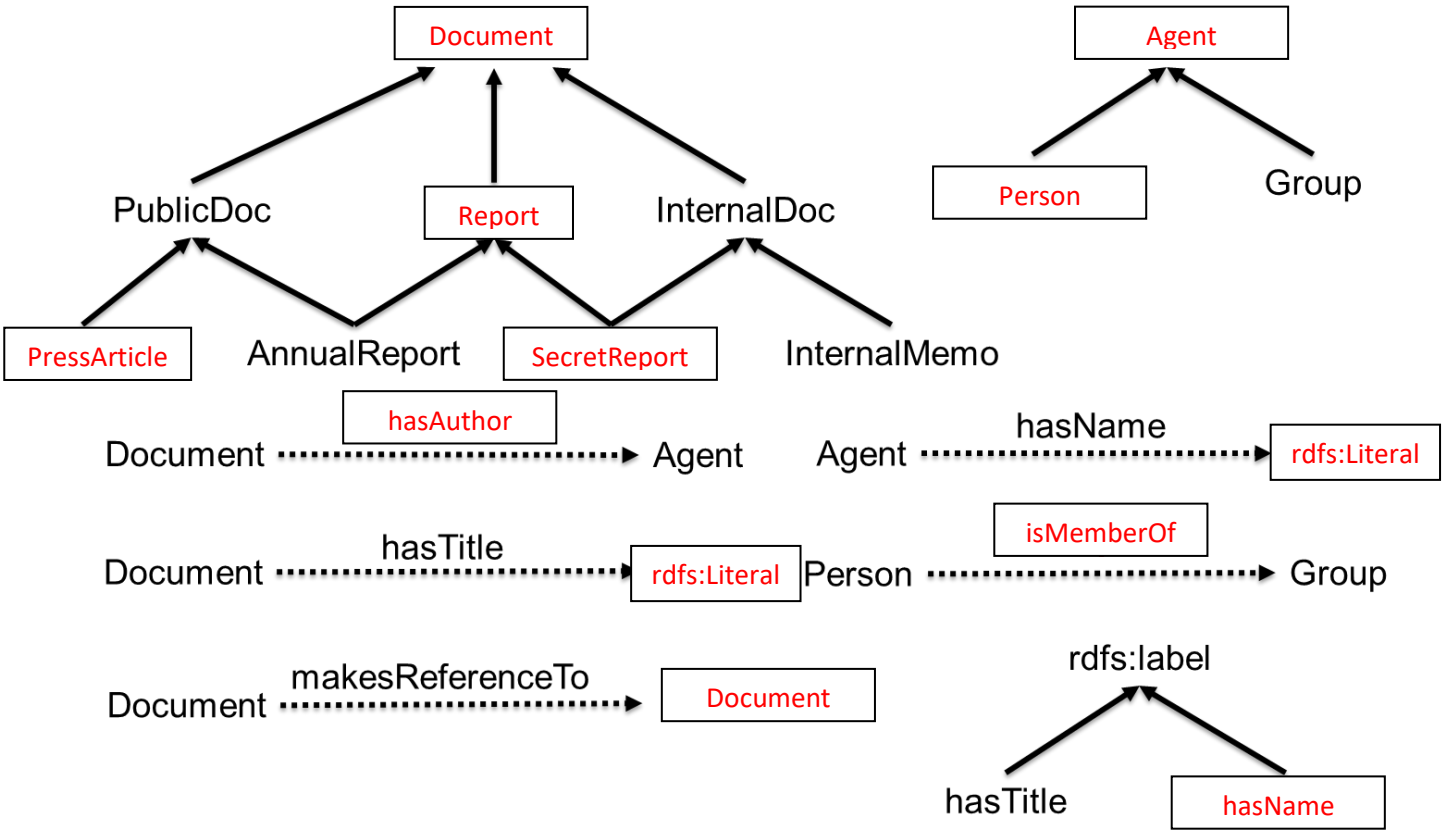
Q4.6. What could we add to this schema (several answers are possible)?

```
@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 .

3rd, 4th, 6th are true.

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.




```
<?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.somewhere.com/doctree/SecretReport">
    <rdfs:subClassOf rdf:resource="http://www.somewhere.com/doctree/InternalDoc"/>
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.somewhere.com/doctree/Report"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.somewhere.com/doctree/PublicDoc">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
```

```
    <rdfs:subClassOf rdf:resource="http://www.somewhere.com/doctree/Document"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.somewhere.com/doctree/hasTitle">
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain rdf:resource="http://www.somewhere.com/doctree/Document"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
    <rdfs:subPropertyOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#label"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.somewhere.com/doctree/hasName">
    <rdfs:subPropertyOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#label"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
    <rdfs:domain rdf:resource="http://www.somewhere.com/doctree/Agent"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.somewhere.com/doctree/Person">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.somewhere.com/doctree/Agent"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.somewhere.com/doctree/isMemberOf">
    <rdfs:range rdf:resource="http://www.somewhere.com/doctree/Person"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain rdf:resource="http://www.somewhere.com/doctree/Person"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.somewhere.com/doctree/Group">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.somewhere.com/doctree/Agent"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.somewhere.com/doctree/AnnualReport">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.somewhere.com/doctree/Report"/>
    <rdfs:subClassOf rdf:resource="http://www.somewhere.com/doctree/PublicDoc"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.somewhere.com/doctree/Agent">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.somewhere.com/doctree/InternalDoc">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.somewhere.com/doctree/Document"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.somewhere.com/doctree/makesReferenceTo">
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain rdf:resource="http://www.somewhere.com/doctree/Document"/>
    <rdfs:range rdf:resource="http://www.somewhere.com/doctree/Document"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.somewhere.com/doctree/Report">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.somewhere.com/doctree/Document"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.somewhere.com/doctree/hasAuthor">
    <rdfs:domain rdf:resource="http://www.somewhere.com/doctree/Document"/>
    <rdfs:range rdf:resource="http://www.somewhere.com/doctree/Agent"/>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.somewhere.com/doctree/Document">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.somewhere.com/doctree/PressArticle">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.somewhere.com/doctree/PublicDoc"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.somewhere.com/doctree/InternalMemo">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.somewhere.com/doctree/InternalDoc"/>
  </rdf:Description>
</rdf:RDF>
```


http://www.somewhere.com/doctree/PressArticle
http://www.somewhere.com/doctree/AnnualReport
http://www.somewhere.com/doctree/SecretReport
http://www.somewhere.com/doctree/InternalMemo
http://www.somewhere.com/doctree/PublicDoc
http://www.somewhere.com/doctree/Report
http://www.somewhere.com/doctree/InternalDoc
http://www.somewhere.com/doctree/MakesReferenceTo
http://www.somewhere.com/doctree/Document
http://www.somewhere.com/doctree/hasTitle
http://www.somewhere.com/doctree/hasAuthor
http://www.somewhere.com/doctree/hasName
http://www.somewhere.com/doctree/isMemberOf
http://www.somewhere.com/doctree/Group
http://www.somewhere.com/doctree/Person
http://www.somewhere.com/doctree/Agent
http://www.w3.org/2000/01/rdf-schema#Class
http://www.w3.org/2000/01/rdf-schema#Literal
http://www.w3.org/1999/02/22-rdf-syntax-ns#Property
http://www.w3.org/2000/01/rdf-schema#label
http://www.w3.org/2000/01/rdf-schema#subClassOf
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.w3.org/2000/01/rdf-schema#domain
http://www.w3.org/2000/01/rdf-schema#range
http://www.w3.org/2000/01/rdf-schema#subPropertyOf

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 .
```

Man is Male and Human.

Woman is Female is Human

Human is a union of Man ad Woman.

Jane is Human and Female, which means is Woman.

John is Man, which means is Male and Human.

James is 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 .
```

GrandFather is a subclass as intersection of Parent and Man.

Jim is Man and Parent, which DOES NOT mean is GrandFather.

Jack is GrandFather, which infers he is Man and Parent as well.

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. Jane hasAncestor Jim.

John hasSouse Jane.

Jim hasAncestor James. James hasChild Jim.

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 .
```

JimmyPage is the same as JamesPatrickPage, and Human and Person, but different from 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 ]
  )
] .
```

UnluckyPerson is a class which is equivalent to a unanimous class which are both Person and not 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.
```

We can assume James and Jane are defined as Human somewhere else, because due to the restriction in the Human class definition, parents of Human have to be Human if any.

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) .
```

First element of PersonList is Person. (a, b, c), (b, c), (c) is a PersonList. Its elements, a, b, and c are all 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 .
```

Human can have only one biological father and only one biological mother. Jane is a Human. James and Jhon are the same resource and a biological father of Jane.

Q5.9 What are we defining and inferring?

```
@prefix ex: <http://example.org/>
ex:Wealthy a owl:Class ;
     owl:equivalentClass  [
       a owl:Class ; owl:intersectionOf  (
         [ a owl:Restriction ;
           owl:onProperty ex:hasChild ;
           owl:allValuesFrom ex:Wealthy
         ] ,
         [ a owl:Restriction ;
           owl:onProperty ex:hasChild ;
           owl:someValuesFrom ex:Wealthy
         ]
     )] .
ex:Tom a ex:Wealthy ; ex:hasChild ex:Tim .
```

If a wealthy human has a child, they are wealthy too. Tom is wealth and has a child named Tim. Tim is wealthy too.

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 .
```

Having two preferred label in English is not allowed.

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

<rdfs:subPropertyOf rdf:resource="#mappingRelation"/>
<rdfs:subPropertyOf rdf:resource="#narrower"/>
<owl:inverseOf rdf:resource="#broadMatch"/>

Q6.3 practice:

1. Open the source file of Dublin Core Terms:
   http://dublincore.org/2012/06/14/dcterms.rdf
   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?

1. Look at the definition of the class FileFormat and find the class it inherits from.

<rdfs:subClassOf rdf:resource="http://purl.org/dc/terms/MediaType"/>

2. Choose your preferred book on Amazon, Fnac, etc. and describe it in an RDF annotation using as many DC primitives as necessary .

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

<https://www.amazon.com/gp/product/0399226907/ref=s9_acsd_ri_bw_c_x_1_w?pf_rd_m=ATVPDKIKX0DER&pf_rd_s=merchandised-search-12&pf_rd_r=9Y4T87KR4S1S1GHS4G44&pf_rd_t=101&pf_rd_p=b1b6e7c5-f122-45e8-9a8c-d025cdf29298&pf_rd_i=283155>
dc:creator < https://www.amazon.com/Eric-Carle/e/B00DVY7LO0/ref=dp_byline_cont_book_1> ;
dc:title "The Very Hungry Caterpillar" ;

dc:language "en" ;
dc:subject "Food" ;
dc:date "1994-03-23" ;
dc:format "paper book" ;
dc:type dcterms:Text .

3. Adding following lines in the previous rdf definition:

@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix cc: <http://creativecommons.org/ns#> .

<https://www.amazon.com/gp/product/0399226907/ref=s9_acsd_ri_bw_c_x_1_w?pf_rd_m=ATVPDKIKX0DER&pf_rd_s=merchandised-search-12&pf_rd_r=9Y4T87KR4S1S1GHS4G44&pf_rd_t=101&pf_rd_p=b1b6e7c5-f122-45e8-9a8c-d025cdf29298&pf_rd_i=283155>
dc:creator < https://www.amazon.com/Eric-Carle/e/B00DVY7LO0/ref=dp_byline_cont_book_1> ;
dc:title "The Very Hungry Caterpillar" ;
dc:language "en" ;
dc:subject "Food" ;
dc:date "1994-03-23" ;
dc:format "paper book" ;
dc:type dcterms:Text ;
cc:license cc:Sharing, cc:HighIncomeNationUse , cc:LesserCopyleft.

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?

3.

<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#InverseFunctionalProperty"/>

4.

<rdfs:subPropertyOf rdf:resource="http://xmlns.com/foaf/0.1/page"/>

5.

<rdfs:domain rdf:resource="http://xmlns.com/foaf/0.1/Agent"/>
<rdfs:range rdf:resource="http://xmlns.com/foaf/0.1/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 rel: <http://purl.org/vocab/relationship/> .

4.

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix rel: <http://purl.org/vocab/relationship/> .

<https://www.datasciencetech.institute/motoharu.dei.dummylink> a foaf:Person ;
        foaf:family_name "Dei";
        foaf:givenname "Motoharu";
        foaf:name "Motoharu Dei";
        foaf:nick "Moto";
        foaf:OnlineAccount <https://www.linkedin.com/in/motoharu-dei-358abaa/>;
        foaf:schoolHomepage <https://www.facebook.com/dummydummy>;
        foaf:knows [ a foaf:Person ;
                        foaf:givenname "Aimen"],
                    [ a foaf:Person ;
                     foaf:givenname "Alix"] ;
        rel:fridendOf [ a foaf:Person ; foaf:givenname "Alix"] ;
        rel:enemyOf [ a foaf:Person ; foaf:givenname "Aimen"] ;
        rel:parentOf [ a foaf:Person ; foaf:givenname "Koharu"] ;
        rel:hasMet [ a foaf:Person ; foaf:givenname "Catherine"] ;
        rel:lostContactWith <http://www.find-lost-ppl.com/#123> .
```

Q6.6 What does this mean?

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

BioRDF and DBLP are linked as Linkset BioRDF2DBLP with exactMatch and 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.

2. Giving a list of endpoints to Dataset.

```
SELECT DISTINCT ?x  ?p WHERE {
        ?x rdfs:label "DBpedia-fr" ;
           ?p ?y .
}
```

➔ There are 19 properties.

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 used stats1998, bar-chart was generated by that plot, and stats1998 is in distribution of CSV format with label “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#> .
@base : <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 distribution in URL <http://wimmics.inria.fr/docs/employ 2014.sql>, titled “SQL Dump of the employees”, with spatial at <http://www.geonames.org/6640252>, issued “2015-01-12”, temporal at <http://reference.data.gov.uk/id/year/2014>, having publisher <http://inria.fr>, in mediaType and format “SQL”, in language <http://id.loc.gov/vocabulary/iso639-1/fr>, and with 38729 triples.

R2RTransform12 is an algorithm used by db-employ, and also by R2R-employ-mapping <http://xmlns.com/foaf/0.1/>, which is used to generate FoaFDump mentioned below.

FoaFDump is a dataset, having feature <http://www.w3.org/ns/formats/RDF_XML>, having dataDump <http://wimmics.inria.fr/docs/employ-2014.rdf>, having exampleResource <http://ns.inria.fr/fabien.gandon#me>, having vocabulary <http://xmlns.com/foaf/0.1/>, having 12875 triples, having title "RDF Dump of the employees",

generated by R2RTransform12 as mentioned above, generated at "2015-01-14T11:38:27", and with derived from db-employ.

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)

mo:MusicArticle

Version: v2.15

There are 10 incoming links to mo.

54 classes and 153 properties.

It is in RDF, RDFS, and 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/rfc2616.txt>, dc :title, “Hypertext Transfer Protocol -- HTTP/1.1”}

{< https://www.ietf.org/rfc/rfc2396.txt>, 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>
```

{#cathy, a, Person}, {#cathy, foaf:name, “Catherine Faron”}, {#cathy, foaf:mbox, “faron@i3s.unice.fr”}, {#cathy, foaf:knows, < http://ns.inria.fr/fabien.gandon#me>}

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

3.

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

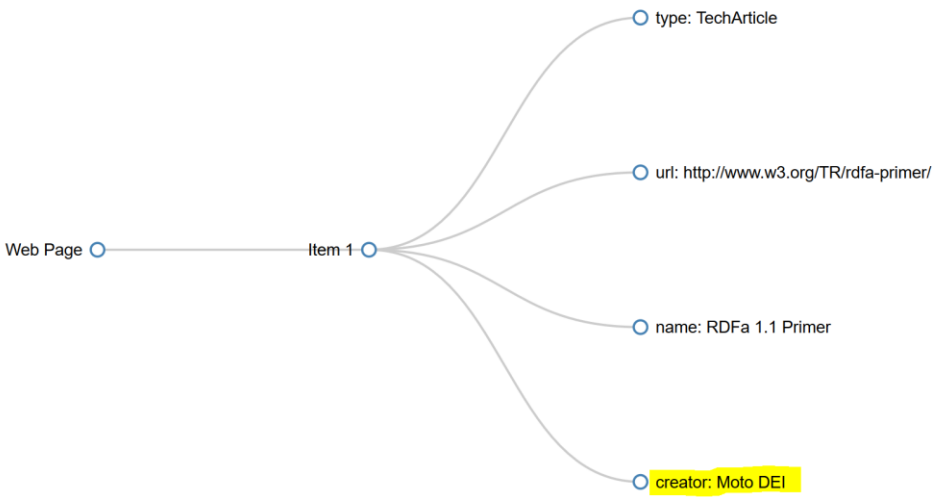
4.

A huge ontology written in HTML.

5.

About some authors, their books, and mutual relationship.

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



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/

Distiller:

```
@prefix fb: <http://www.facebook.com/2008/fbml> .
@prefix ns1: <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/tt0111161/?ref_=nv_sr_1?ref_=nv_sr_1> og:description
over a number of years, finding solace and eventual redemption through acts of co
    og:image "https://m.media-amazon.com/images/M/MV5BMDFkYTc0MGEtZmNhMC00ZDIzLWF
    og:site_name "IMDb" ;
    og:title "The Shawshank Redemption (1994) - IMDb" ;
    og:type "video.movie" ;
    og:url "http://www.imdb.com/title/tt0111161/" ;
    ns1:fbmlapp_id "115109575169727" .
```

Translator:

```
@prefix fb: <http://www.facebook.com/2008/fbml> .
@prefix ns1: <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/tt0111161/?ref_=nv_sr_1?ref_=nv_sr_1> og:description
    og:image "https://m.media-amazon.com/images/M/MV5BMDFkYTc0MGEtZmNhMC00ZDIzLWF
    og:site_name "IMDb" ;
    og:title "The Shawshank Redemption (1994) - IMDb" ;
    og:type "video.movie" ;
    og:url "http://www.imdb.com/title/tt0111161/" ;
    ns1:fbmlapp_id "115109575169727" .
```

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/

Example of 'Compacted':

```
{
```

```
"@graph": [
 {
  "@id": "_:ub12bL17C14",
  "@type": "http://xmlns.com/foaf/0.1/Person",
  "http://xmlns.com/foaf/0.1/givenname": "Aimen"
 },
 {
  "@id": "https://www.datasciencetech.institute/motoharu.dei.dummylink",
  "@type": "http://xmlns.com/foaf/0.1/Person",
  "http://purl.org/vocab/relationship/enemyOf": {
   "@id": "_:ub12bL17C14"
  },
  "http://purl.org/vocab/relationship/fridendOf": {
   "@id": "_:ub12bL16C16"
  },
  "http://purl.org/vocab/relationship/hasMet": {
   "@id": "_:ub12bL19C13"
  },
  "http://purl.org/vocab/relationship/lostContactWith": {
   "@id": "http://www.find-lost-ppl.com/#123"
  },
  "http://purl.org/vocab/relationship/parentOf": {
   "@id": "_:ub12bL18C15"
  },
  "http://xmlns.com/foaf/0.1/OnlineAccount": {
   "@id": "https://www.linkedin.com/in/motoharu-dei-358abaa/"
  },
  "http://xmlns.com/foaf/0.1/family_name": "Dei",
  "http://xmlns.com/foaf/0.1/givenname": "Motoharu",
  "http://xmlns.com/foaf/0.1/knows": [
   {
    "@id": "_:ub12bL14C7"
   },
   {
    "@id": "_:ub12bL12C13"
   }
  ],
  "http://xmlns.com/foaf/0.1/name": "Motoharu Dei",
  "http://xmlns.com/foaf/0.1/nick": "Moto",
  "http://xmlns.com/foaf/0.1/schoolHomepage": {
   "@id": "https://www.facebook.com/dummydummy"
  }
 },
 {
  "@id": "_:ub12bL16C16",
  "@type": "http://xmlns.com/foaf/0.1/Person",
  "http://xmlns.com/foaf/0.1/givenname": "Alix"
 },
 {
  "@id": "_:ub12bL12C13",
  "@type": "http://xmlns.com/foaf/0.1/Person",
  "http://xmlns.com/foaf/0.1/givenname": "Aimen"
 },
 {
  "@id": "_:ub12bL18C15",
  "@type": "http://xmlns.com/foaf/0.1/Person",
  "http://xmlns.com/foaf/0.1/givenname": "Koharu"
```

```
    },
    {
     "@id": "_:ub12bL19C13",
     "@type": "http://xmlns.com/foaf/0.1/Person",
     "http://xmlns.com/foaf/0.1/givenname": "Catherine"
    },
    {
     "@id": "_:ub12bL14C7",
     "@type": "http://xmlns.com/foaf/0.1/Person",
     "http://xmlns.com/foaf/0.1/givenname": "Alix"
    }
   ]
}
```

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.

2nd, 4th are correct.

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

2nd, 4th, 7th, 8th, and 9th are correct.

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: http://wimmics.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 voc: <http://www.unice.fr/voc#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://www.Jen.ne> a voc:Engineer, voc:Woman;
        voc:name "Jen";
        voc:age "42"^^xsd:int;
        voc:shoeSize "36"^^xsd:int;
        voc:trouserSize "38"^^xsd:int;
        voc:hasSpouse <http://www.Seb.ne> ;
        voc:hasChild <http://www.Ammy.ne>, <http://www.Steffen.ne>;
        voc:hasColleague <http://www.Catherine.ne>, <http://www.Fabien.ne>;
        voc:hasFather <http://www.Thomas.ne> .

<http://www.Seb.ne> a voc:Man;
```

```
        voc:hasChild <http://www.Ammy.ne>, <http://www.Steffen.ne> .

<http://www.Ammy.ne> a voc:Woman .

<http://www.Steffen.ne> a voc:Man .

<http://www.Thomas.ne> a voc:Man;
        voc:name "Thomas" .

<http://www.Catherine.ne> voc:name "Catherine" .

<http://www.Fabien.ne> 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.Jen.ne">
  <rdf:type rdf:resource="http://www.unice.fr/voc#Engineer"/>
  <voc:trouserSize rdf:datatype="http://www.w3.org/2001/XMLSchema#int">38</voc:trouserSize>
  <voc:hasChild rdf:resource="http://www.Ammy.ne"/>
  <voc:name>Jen</voc:name>
  <voc:shoeSize rdf:datatype="http://www.w3.org/2001/XMLSchema#int">36</voc:shoeSize>
  <voc:age rdf:datatype="http://www.w3.org/2001/XMLSchema#int">42</voc:age>
  <voc:hasSpouse rdf:resource="http://www.Seb.ne"/>
  <rdf:type rdf:resource="http://www.unice.fr/voc#Woman"/>
  <voc:hasChild rdf:resource="http://www.Steffen.ne"/>
  <voc:hasColleague rdf:resource="http://www.Fabien.ne"/>
  <voc:hasFather rdf:resource="http://www.Thomas.ne"/>
  <voc:hasColleague rdf:resource="http://www.Catherine.ne"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://www.Steffen.ne">
  <rdf:type rdf:resource="http://www.unice.fr/voc#Man"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://www.Seb.ne">
  <voc:hasChild rdf:resource="http://www.Steffen.ne"/>
  <voc:hasChild rdf:resource="http://www.Ammy.ne"/>
  <rdf:type rdf:resource="http://www.unice.fr/voc#Man"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://www.Fabien.ne">
  <voc:name>Fabien</voc:name>
 </rdf:Description>
 <rdf:Description rdf:about="http://www.Thomas.ne">
  <voc:name>Thomas</voc:name>
  <rdf:type rdf:resource="http://www.unice.fr/voc#Man"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://www.Ammy.ne">
  <rdf:type rdf:resource="http://www.unice.fr/voc#Woman"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://www.Catherine.ne">
  <voc:name>Catherine</voc:name>
 </rdf:Description>
```

</rdf:RDF>

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.

-> Done.

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 rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<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

My guess is that http://ns.inria.fr/fabien.gandon is an abstract container for the resource of fabien.gandon, while the http://ns.inria.fr/fabien.gandon#me is the actual implementation of it. The former one can potentially have links to resources which the latter one does not fit; possible less 'physical' attributes

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:

(dbr:Paris, rdf:type, 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?

"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?

xml:base="http://www.inria.fr/2007/09/11/humans.rdfs-instances"

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

xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" and used it by rdf:ID= or rdf:resource=.

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

From this code, we can assume there is a predefined entity "human" as such <!DOCTYPE rdf:RDF [ENTITY human "http://www.something">]>. Here, `xmlns="&humans;#"` represents the same as `xmlns="http://www.something#"`

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

John is:

- Child of “#Harry”
- A type Person
- Name “John”
- Shoesize 14
- Age 37
- Father of “#Mark”
- Friend of “#Alice”
- Spouse of “#Jennifer”
- Shirtsize 12
- Trouserssize 44
- Child of “#Sophie”

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 rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@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" .
```

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

Laura is:

- A type Lecturer, Person, and Researcher
- Friend of <http://www.inria.fr/2007/09/11/humans.rdfs-instances#Alice>
- Name “Laura”
- Spouse of <http://www.inria.fr/2007/09/11/humans.rdfs-instances#William>
- Mother of <http://www.inria.fr/2007/09/11/humans.rdfs-instances#Catherine>

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: http://wimmics.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:

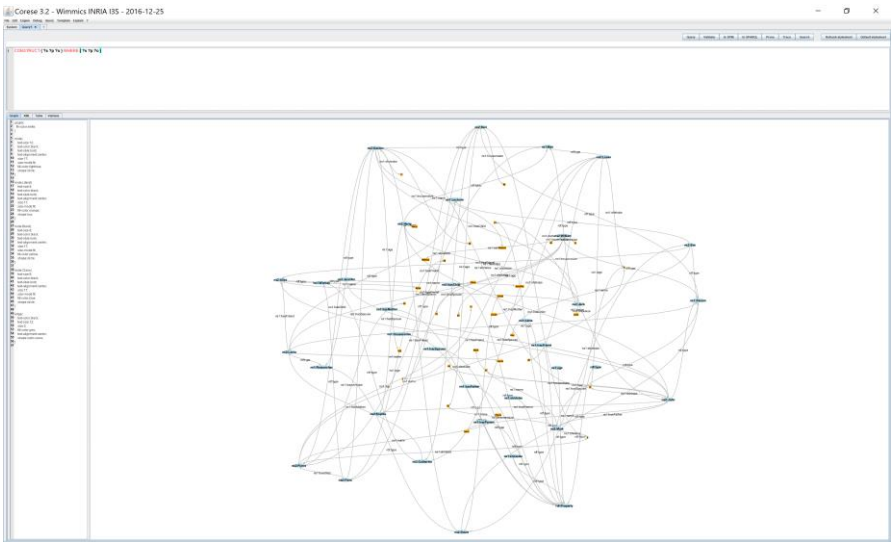
Create a new tab to enter the following query and explain what it does and the results you get. This is a good way to familiarize yourself with the data.

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

Explanation:

> In general, this query will pull all triples exist in human_2013.rdf and construct RDF triples from the result of them. Corese also displayed the graph as attached below.

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.

> List subjects and objects of triples with rdf:type as property and with object existing in <http://www.inria.fr/2007/09/11/humans.rdfs#>.

Run this query. How many answers do you get?

> 21

Find John and his types in the answers.

John's types:

> http://www.inria.fr/2007/09/11/humans.rdfs#Person

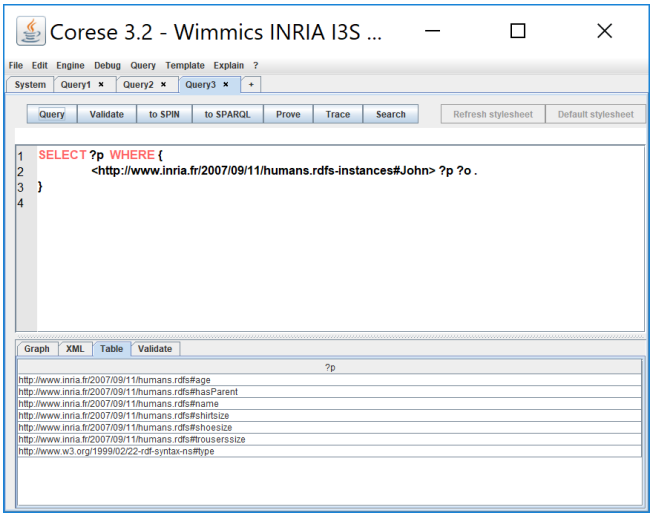
Question 3:

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

   ```
   SELECT ?p WHERE {
           <http://www.inria.fr/2007/09/11/humans.rdfs-instances#John> ?p ?o .
   }
   ```

   Results:

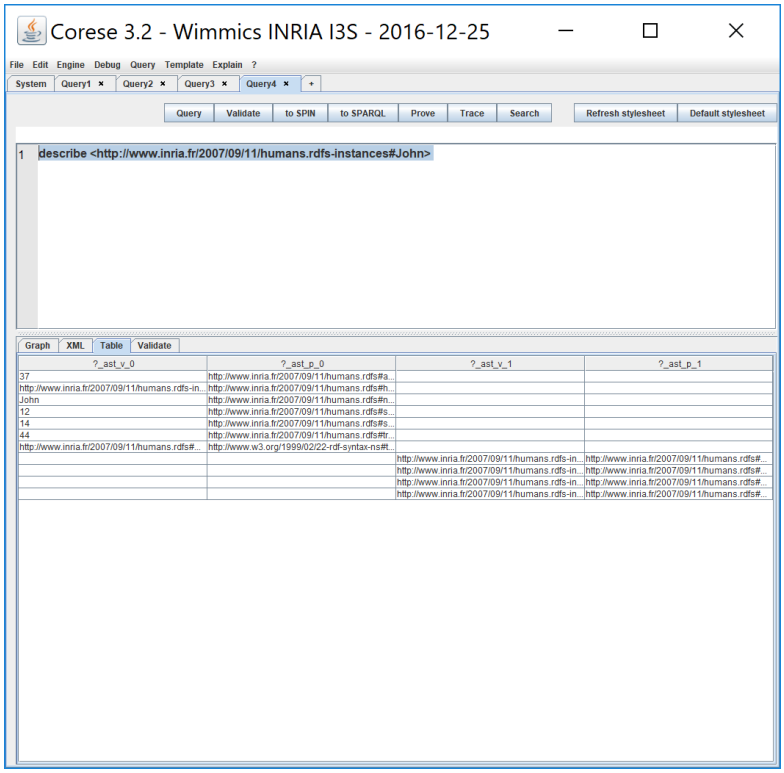


2. request a description of John using the SPARQL clause for this.

Query

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

Results:



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.

Selecting all sets of subject and object where their property is http://www.inria.fr/2007/09/11/humans.rdfs#hasSpouse. (meaning getting all the couples)

Run this query. How many answers do you get?

6

Question 5:

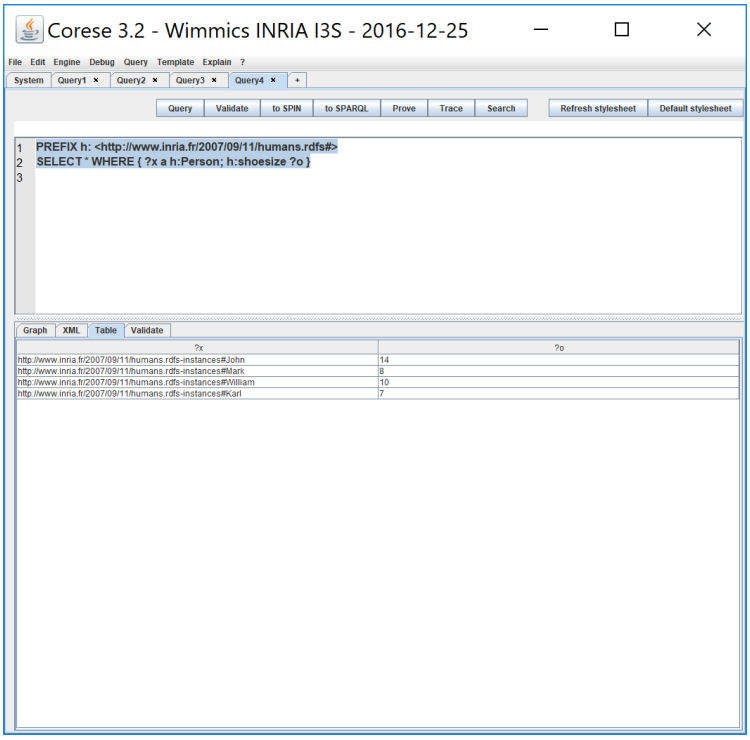
In the RDF file, find the name of the property that is used to give the shoe size of a person.

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 * WHERE { ?x a h:Person; h:shoesize ?o }
```

Result:

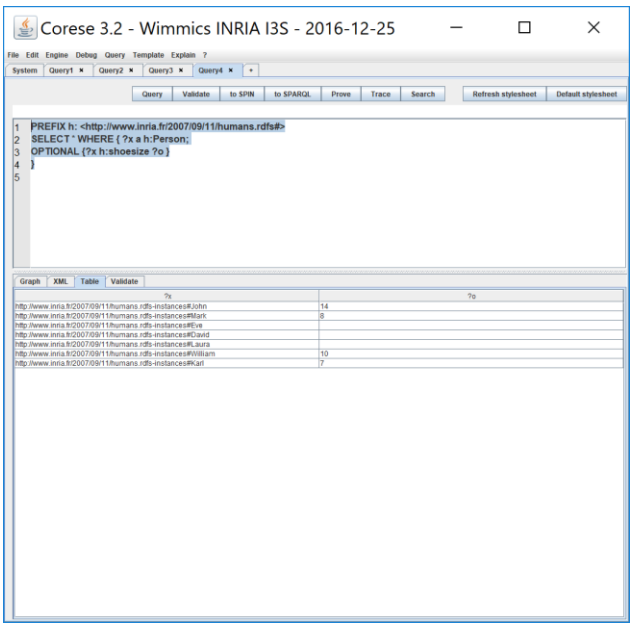


2. Change this query to retrieve all the persons and, if available, their shoe size.

Query:

```
PREFIX h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
SELECT * WHERE { ?x a h:Person .
OPTIONAL {?x h:shoesize ?o }
}
```

Result:

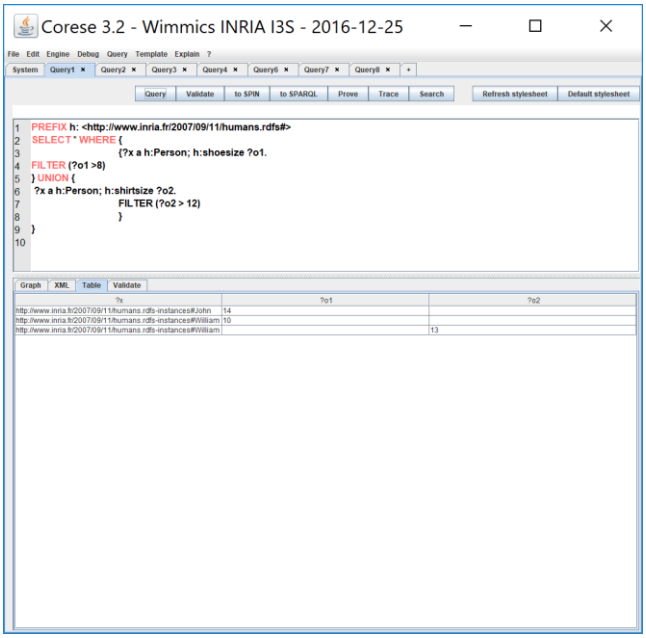


3. Change this query to retrieve all the persons whose shoe size is greater than 8 or whose shirt size is greater than 12.

Query:

```
PREFIX h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
SELECT * WHERE {
                {?x a h:Person; h:shoesize ?o1.
FILTER (?o1 >8)
} UNION {
 ?x a h:Person; h:shirtsize ?o2.
                FILTER (?o2 > 12)
                }
}
```

Result:



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 ?parent WHERE { ?parent h:hasChild ?child}
```

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

5 records. Gaston has two duplicates.

2. Find a way to avoid duplicates.

Query:

```
PREFIX h: <http://www.inria.fr/2007/09/11/humans.rdfs#> .
SELECT DISTINCT ?parent WHERE { ?parent h:hasChild ?child}
```

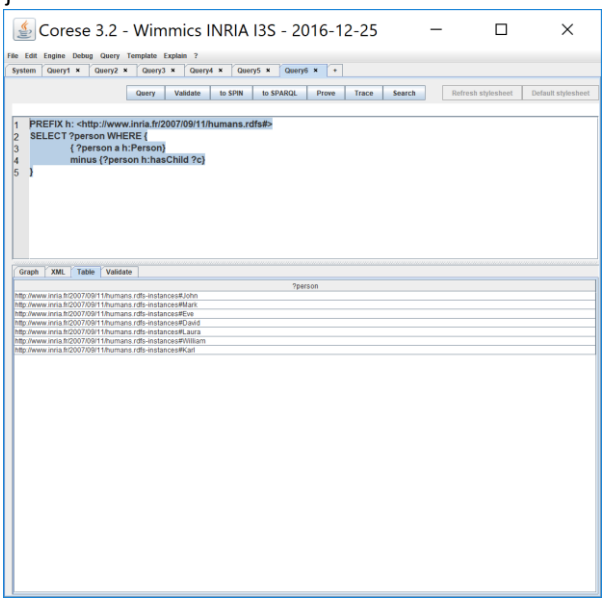
How many answers do you get then?

4 records.

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 ?person WHERE {
        { ?person a h:Person}
        minus {?person h:hasChild ?c}
}
```



Question 7

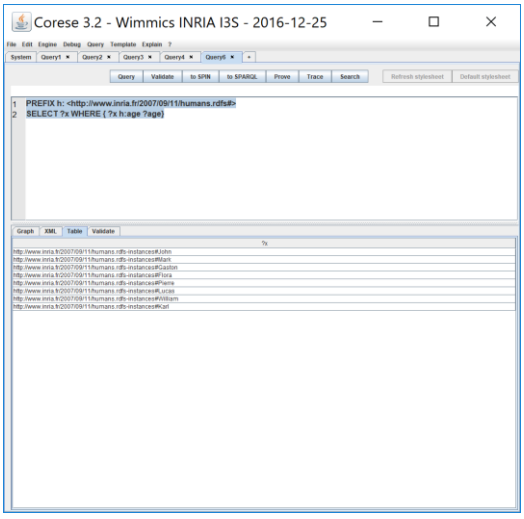
In the RDF file, find the name of the property that is used to give the age of a person.

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

Query:

```
PREFIX h: <http://www.inria.fr/2007/09/11/humans.rdfs#> .
SELECT ?x WHERE { ?x h:age ?age}
```

Result:



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

Query:

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

SELECT ?x WHERE { ?x h:age ?age FILTER (xsd:integer(?age) <= 17) }

How many answers do you get?

2 records.

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 xsd: <http://www.w3.org/2001/XMLSchema#> .
ASK {?x h:name "Mark"; h:age ?age FILTER (xsd:interger(?age) >=18)}
```

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

Query:

```
PREFIX h: <http://www.inria.fr/2007/09/11/humans.rdfs#> .
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> .
SELECT ?x ?even_age WHERE {
        ?x h:age ?age
        BIND( IF(xsd:float(?age) = ROUND(xsd:float(?age)/2)*2, xsd:boolean(1), xsd:boolean(0)) as
?even_age) .
}
```

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 {?y h:hasFriend ?x .}
WHERE {?x  h:hasFriend ?y .}
```

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:

[before insert -> **6 records**]

```
PREFIX h: <http://www.inria.fr/2007/09/11/humans.rdfs#> .
SELECT * WHERE {?x h:hasFriend ?y .}
```

[Insert]

```
PREFIX h: <http://www.inria.fr/2007/09/11/humans.rdfs#> .
INSERT {?y h:hasFriend ?x .}
WHERE {?x h:hasFriend ?y .}
```

[after insert -> **12 records**]

```
PREFIX h: <http://www.inria.fr/2007/09/11/humans.rdfs#> .
SELECT * WHERE {?x h:hasFriend ?y .}
```

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:hasFriend ?y; h:age ?age}
WHERE {?x  ?p ?y .}
```

Result:

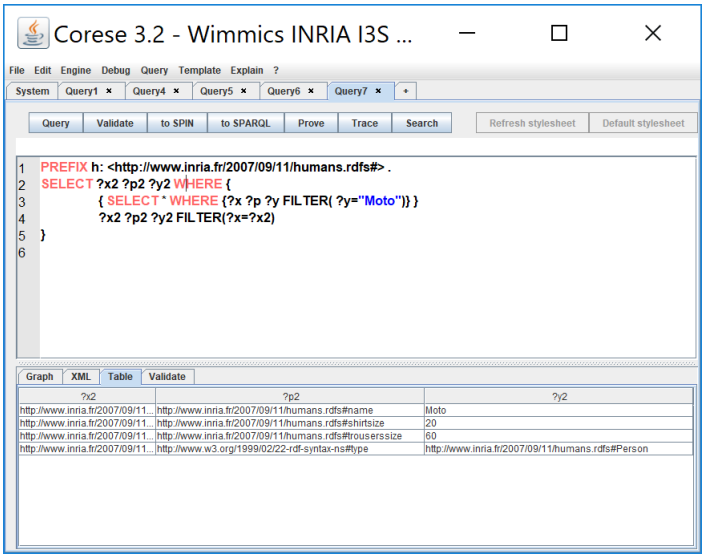
Question 10

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.

Query:

```
PREFIX h: <http://www.inria.fr/2007/09/11/humans.rdfs#> .
SELECT ?x2 ?p2 ?y2 WHERE {
        { SELECT * WHERE {?x ?p ?y FILTER( ?y="Moto")} }
        ?x2 ?p2 ?y2 FILTER(?x=?x2)
}
```

screenshots:



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 ?x1 ?x2 WHERE {
        ?x1 h:shirtsize ?size .
        ?x2 h:shirtsize ?size .
        FILTER (?x1 != ?x2)
}
```

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

Query:

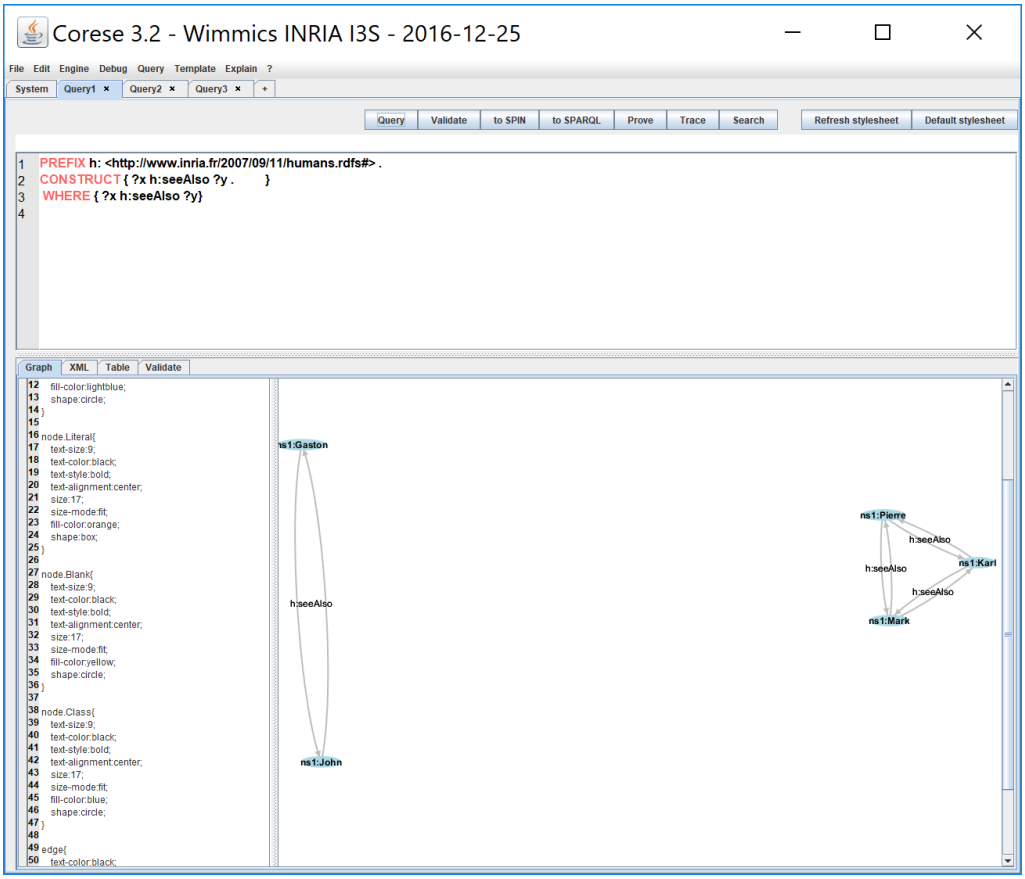
```
PREFIX h: <http://www.inria.fr/2007/09/11/humans.rdfs#> .
CONSTRUCT {
        ?x1 h:seeAlso ?x2 .
        ?x2 h:seeAlso ?x1 .
}
WHERE {
        ?x1 h:shirtsize ?size .
        ?x2 h:shirtsize ?size .
        FILTER (?x1 != ?x2)
}
```

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

Query:

```
PREFIX h: <http://www.inria.fr/2007/09/11/humans.rdfs#> .
CONSTRUCT { ?x h:seeAlso ?y .  }
 WHERE { ?x h:seeAlso ?y}
```

screenshot:



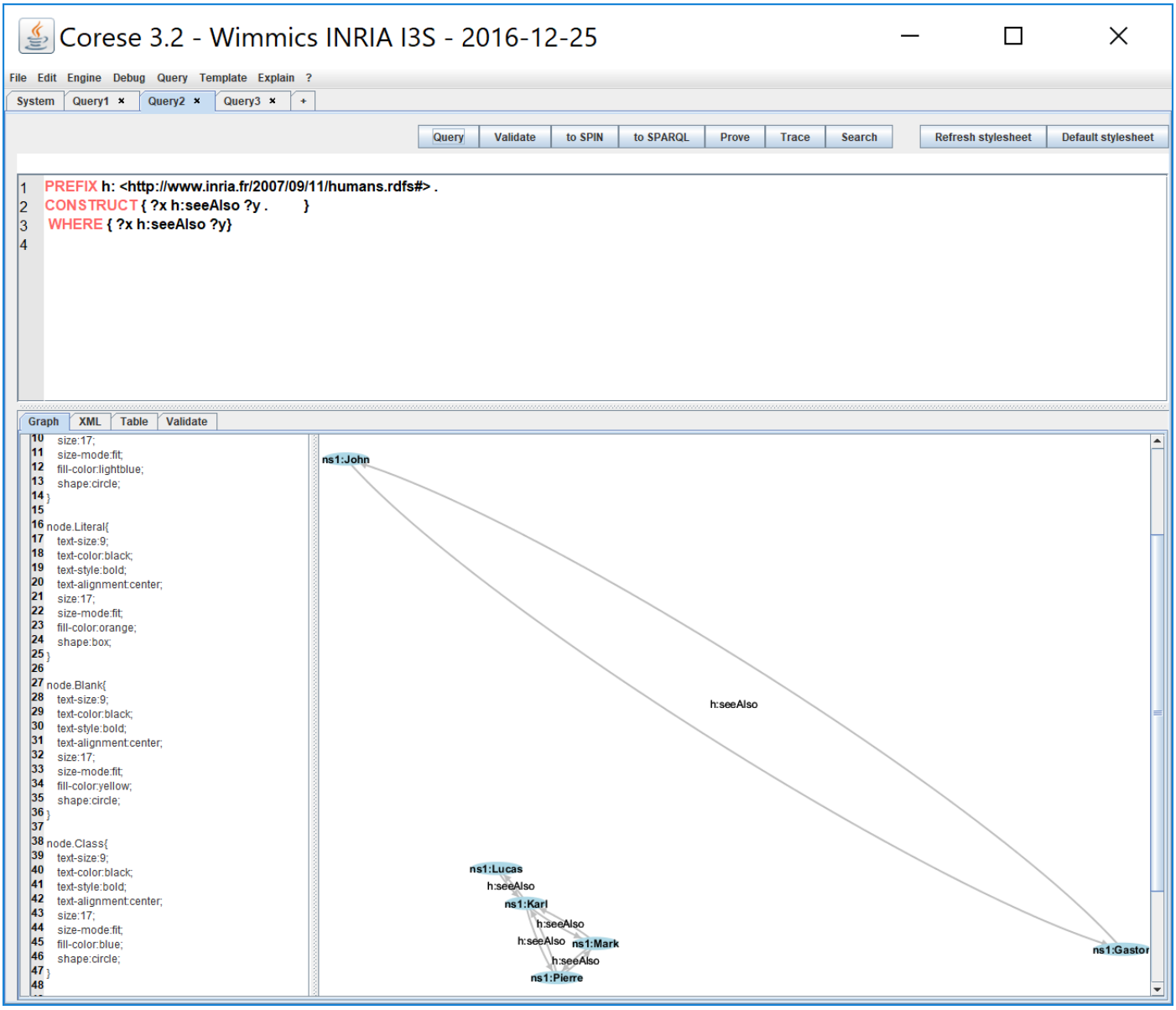
5. Adapt the first query to find persons who have the same shoe size and insert a seeAlso relationship between them.

Query:

```
PREFIX h: <http://www.inria.fr/2007/09/11/humans.rdfs#> .
INSERT {
        ?x1 h:seeAlso ?x2 .
        ?x2 h:seeAlso ?x1 .
}
WHERE {
        ?x1 h:shoesize ?size .
        ?x2 h:shoesize ?size .
        FILTER (?x1 != ?x2)
}
```

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#> .
CONSTRUCT { ?x h:seeAlso ?y . }
WHERE { ?x (h:seeAlso)+ ?y}
```

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

   No linked appeared.

Question 12

1. Find the largest shoe size

Query:

```
PREFIX h: <http://www.inria.fr/2007/09/11/humans.rdfs#> .
SELECT * WHERE {
        {SELECT (max(?ss) as ?max_shoesize) WHERE { ?x h:shoesize ?ss}}
}
```

➔ Result: 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 ?x WHERE {
        {SELECT (max(?ss) as ?max_shoesize) WHERE { ?x h:shoesize ?ss}}
        ?x h:shoesize ?max_shoesize
}
```

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

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

Query:

```
PREFIX h: <http://www.inria.fr/2007/09/11/humans.rdfs#> .
SELECT * WHERE {
        {SELECT (avg(?ss) as ?avg_shoesize) WHERE { ?x h:shoesize ?ss}}
}
```

➔ Result: 9.285714285714286

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 * WHERE {
        {SELECT (sum(?ss)/count(?ss) as ?avg_test) WHERE { ?x h:shoesize ?ss}}
}
```

➔ Result: 9.285714285714286 (Same as the prior result.)

Question 13

Find couples without children

Query:

```
PREFIX h: <http://www.inria.fr/2007/09/11/humans.rdfs#> .
SELECT ?x ?y WHERE {
        {?x h:hasSpouse ?y}
        Minus { {?x h:hasChild ?c1} UNION {?y h:hasChild ?c2} }
}
```

Question 14

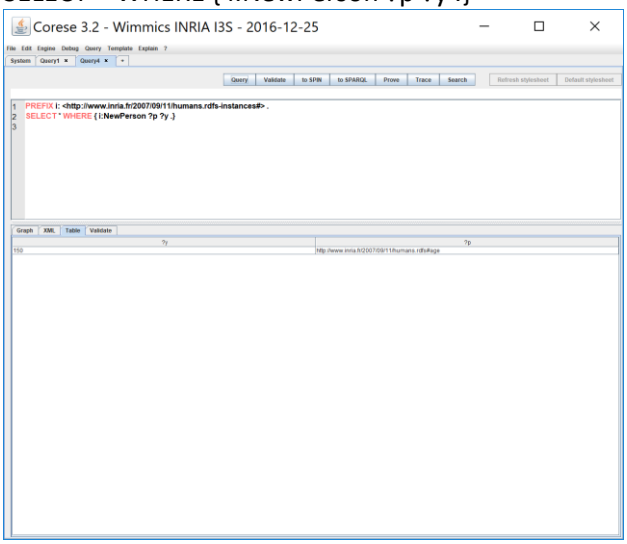
Using INSERT DATA, create a new person with its properties. Then, check that it has been created.

Insert:

```
PREFIX h: <http://www.inria.fr/2007/09/11/humans.rdfs#> .
PREFIX i: <http://www.inria.fr/2007/09/11/humans.rdfs-instances#> .
INSERT DATA {i:NewPerson h:age 150 .}
```

Screenshot result:

```
PREFIX i: <http://www.inria.fr/2007/09/11/humans.rdfs-instances#> .
SELECT * WHERE { i:NewPerson ?p ?y .}
```



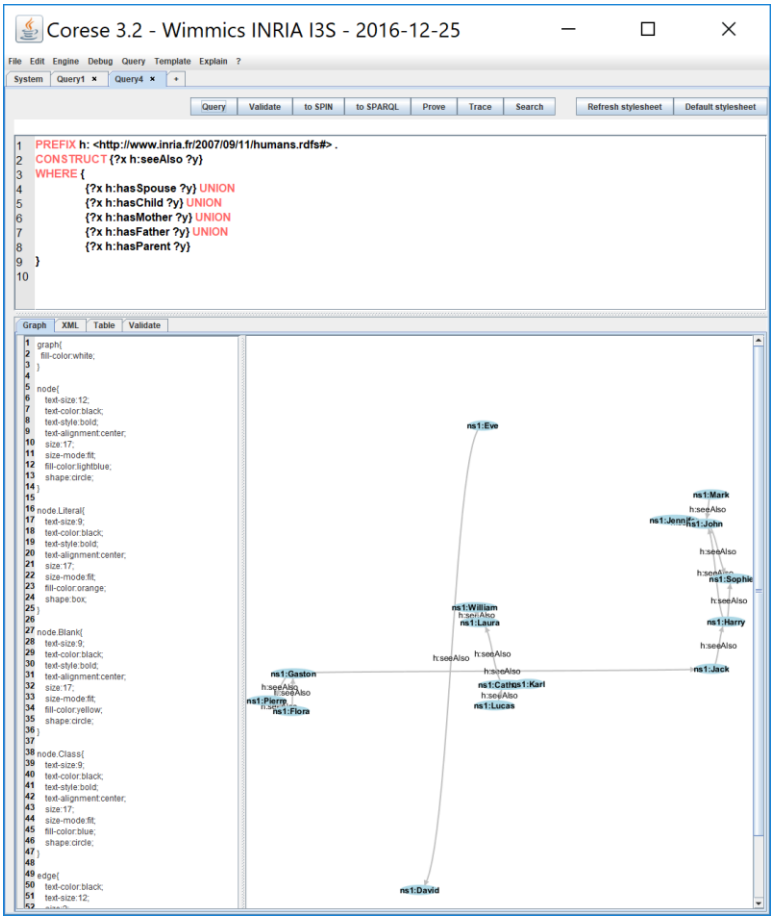
Question 15

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 h:hasSpouse ?y} UNION
        {?x h:hasChild ?y} UNION
        {?x h:hasMother ?y} UNION
        {?x h:hasFather ?y} UNION
        {?x h:hasParent ?y}
}
```

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

> Visualize the graph of first 20 relations from spouse relations and name relation of each member in the couples, extracted by remote access to “Virtuoso SPARQL Query Editor”.

modify it to <u>insert</u> new persons in the base and check the results.

query:

> [Create a new graph]
> PREFIX sw: <http://www.somewhare.com#>
> CREATE GRAPH sw:g1

[Pull some triples from dbpedia remotely]

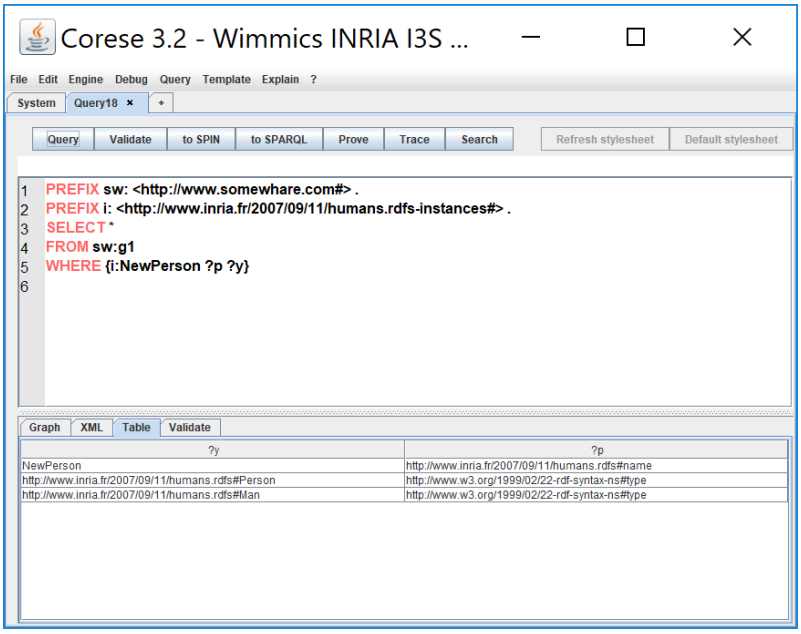
```
PREFIX sw: <http://www.somewhare.com#> .
PREFIX h: <http://www.inria.fr/2007/09/11/humans.rdfs#> .
WITH sw:g1
INSERT {?x ?p ?y .}
WHERE { SERVICE <http://fr.dbpedia.org/sparql/> { SELECT * WHERE {?x ?p ?y.} LIMIT 20 } }
```

[Insert some new triples]

```
PREFIX sw: <http://www.somewhare.com#> .
PREFIX h: <http://www.inria.fr/2007/09/11/humans.rdfs#> .
PREFIX i: <http://www.inria.fr/2007/09/11/humans.rdfs-instances#> .
WITH sw:g1
INSERT {i:NewPerson a h:Person, h:Man . i:NewPerson h:name "NewPerson" .}
WHERE {?x ?p ?y}
```

[Check if they were actually inserted]

```
PREFIX sw: <http://www.somewhare.com#> .
PREFIX i: <http://www.inria.fr/2007/09/11/humans.rdfs-instances#> .
SELECT *
FROM sw:g1
WHERE {i:NewPerson ?p ?y}
```



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: http://wimmics.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.

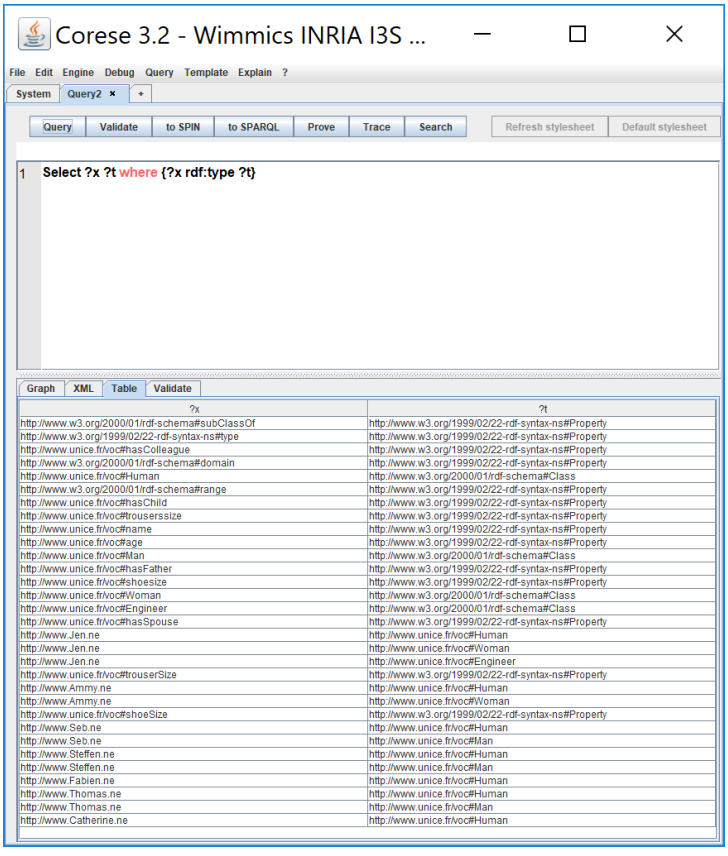
Your schema:

```
<?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#hasColleague">
  <rdfs:domain rdf:resource="http://www.unice.fr/voc#Human"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:range rdf:resource="http://www.unice.fr/voc#Human"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://www.unice.fr/voc#hasChild">
  <rdfs:range rdf:resource="http://www.unice.fr/voc#Human"/>
  <rdfs:domain rdf:resource="http://www.unice.fr/voc#Human"/>
  <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#trouserssize">
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:domain rdf:resource="http://www.unice.fr/voc#Human"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://www.unice.fr/voc#name">
  <rdfs:domain rdf:resource="http://www.unice.fr/voc#Human"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://www.unice.fr/voc#age">
  <rdfs:domain rdf:resource="http://www.unice.fr/voc#Human"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://www.unice.fr/voc#Human">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
 </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#Human"/>
 </rdf:Description>
```

```
 <rdf:Description rdf:about="http://www.unice.fr/voc#hasFather">
  <rdfs:range rdf:resource="http://www.unice.fr/voc#Man"/>
  <rdfs:domain rdf:resource="http://www.unice.fr/voc#Human"/>
  <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#shoesize">
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:domain rdf:resource="http://www.unice.fr/voc#Human"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://www.unice.fr/voc#Woman">
  <rdfs:subClassOf rdf:resource="http://www.unice.fr/voc#Human"/>
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://www.unice.fr/voc#Engineer">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdfs:subClassOf rdf:resource="http://www.unice.fr/voc#Human"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://www.unice.fr/voc#hasSpouse">
  <rdfs:range rdf:resource="http://www.unice.fr/voc#Human"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:domain rdf:resource="http://www.unice.fr/voc#Human"/>
 </rdf:Description>
</rdf:RDF>
```

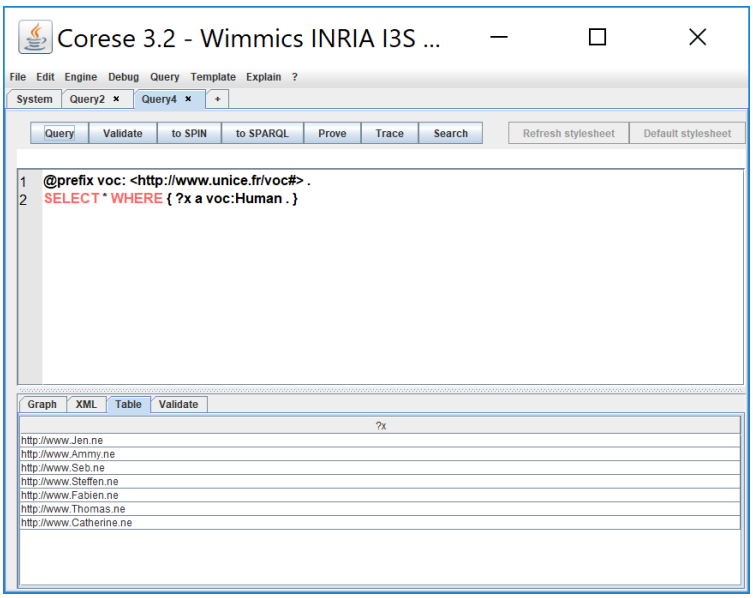
- Check that your RDF schema and RDF files are valid using the W3C's RDF validation service.
- Launch the standalone interface of Corese and load your files Family.rdfs and Jen.rdf
- The interface contains a default SPARQL query:
  `Select ?x ?t where {?x rdf:type ?t}`
  Launch the query and look at the results.

Screenshot:



- 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:



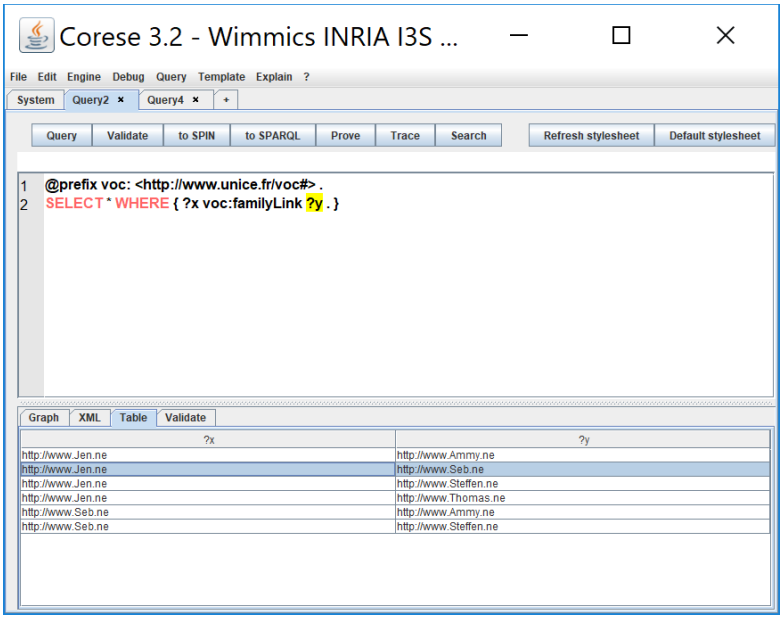
Explanation:

I have already added Human above. I needed it for super class of Engineer, Man, and Woman; and for assignment of the domains or ranges. The screenshot shows the assignment of both super class and sub class.

Nobody has rdf:type Human in Jen.rdf but thanks to the class hierarchy defined in Family.rdfs, everyone is now pulled as Human.

- 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:

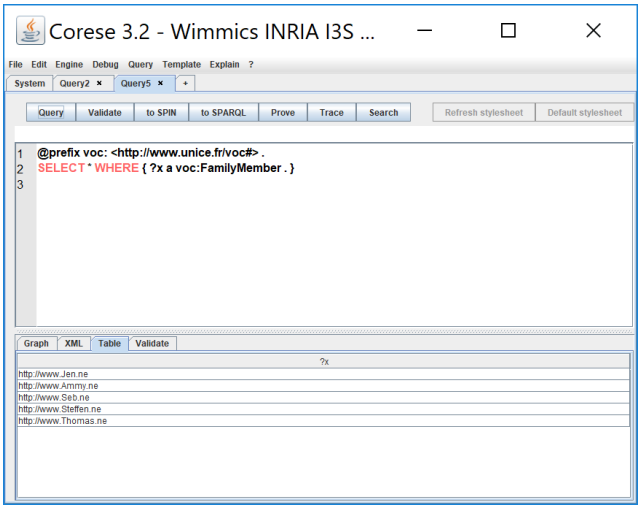


Explanation:

Though no resource has directly familyLink in its property definition in Jen.rdf, thanks to property hierarchy defined in Family.rdfs, now those properties have familyLink as property.

- 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:



Explanation:

Though no Human in Jen.rdf is directly FamilyMember, thanks to signature of property on familyLink, every Human is inferred as FamilyMember and derived by the query.

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?

The namespaces are declared at the first line of rdfs file as such *<rdf:RDF* ***xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"*** *....>*. It is associated to Class or Property by *rdf:[class name/property name]*.

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.
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?

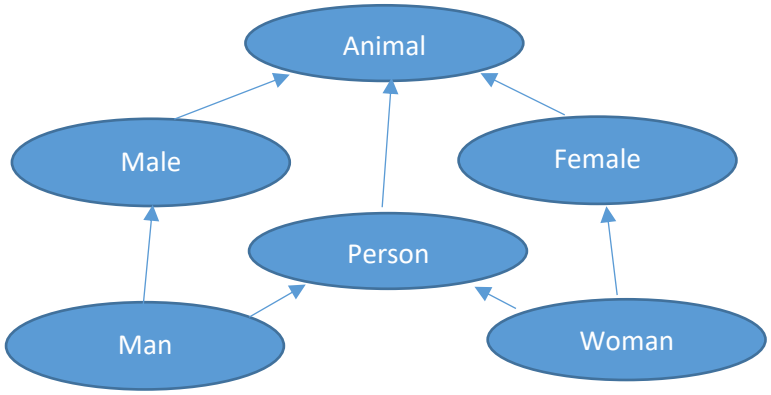
They are associated to the namespace *rdf or rdfs*.

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

Conceptually, resources from Animal class or its sub classes like Male, Female, Person, Lecturer, and ect. can have age property.
Technically, since humans.rdfs does not restrict anything for age property, any class of resources can have.

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:



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 * WHERE { ?x a rdfs:Class . }
```

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

query:

```
SELECT * WHERE { ?x rdfs:subClassOf ?y . }
```

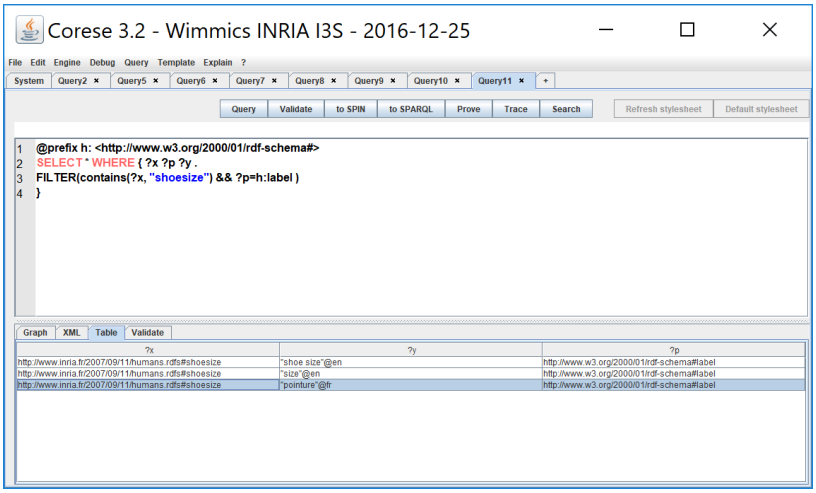
3. Write a query to find the definitions and translations of "shoe size" (*other* labels and comments in different languages for the resource labeled "shoe size").

query:

```
@prefix h: <http://www.w3.org/2000/01/rdf-schema#>
SELECT * WHERE { ?x ?p ?y .
```

FILTER(contains(?x, "shoesize") && ?p=h:label)
}

answers:

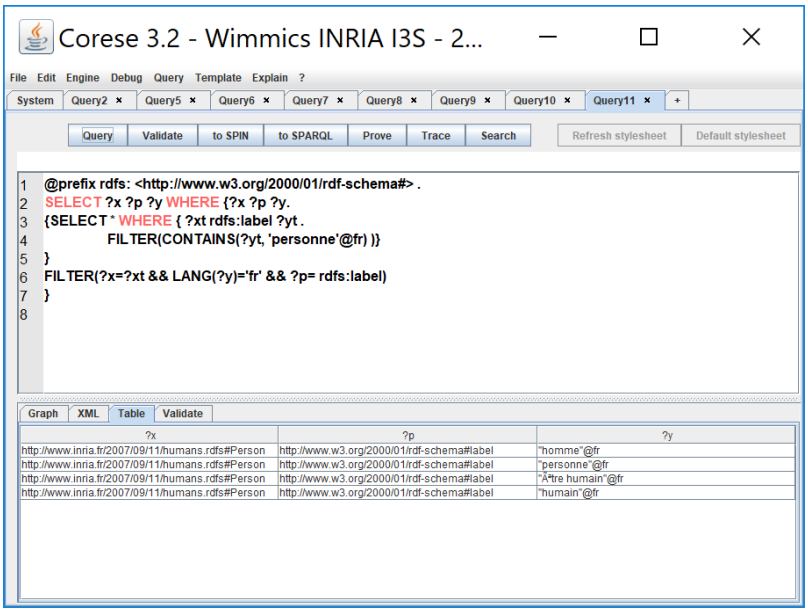


4. Write a query to find the synonyms in French of the word 'personne' in French (*other* labels in the same language for the same resource/class/property). What are the answers?

query:

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
SELECT ?x ?p ?y WHERE {?x ?p ?y.
        {SELECT * WHERE { ?xt rdfs:label ?yt .
                FILTER(CONTAINS(?yt, 'personne'@fr) )}
        }
        FILTER(?x=?xt && LANG(?y)='fr' && ?p= rdfs:label)
}
```

answers:



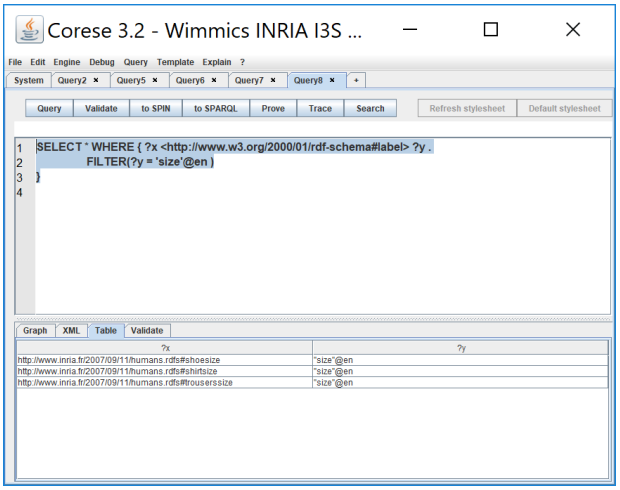
5. Write a query to find the different meaning of the term "size" (disambiguation using the different comments attached to different resources/classes/properties having the label "size"). What are the answers?

query:

```
SELECT * WHERE { ?x <http://www.w3.org/2000/01/rdf-schema#label> ?y .
        FILTER(?y = 'size'@en )
}
```

(* note: here I added “@en” because “size” is an English word.)

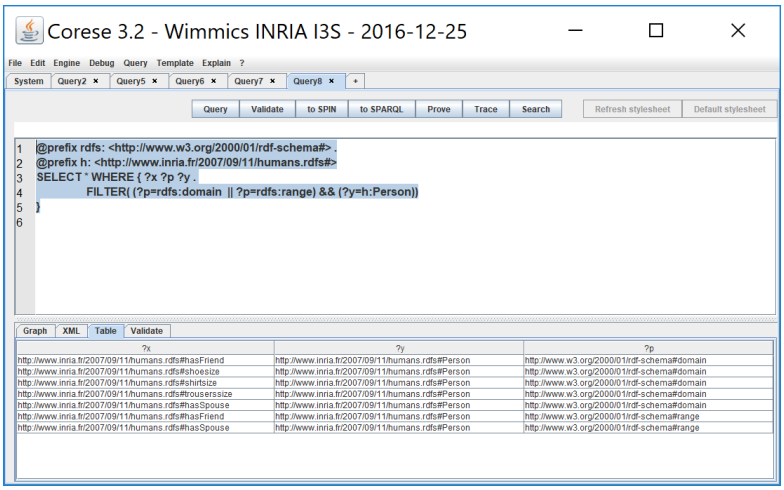
answers:



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

query:

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
SELECT * WHERE { ?x ?p ?y .
        FILTER( (?p=rdfs:domain  || ?p=rdfs:range) && (?y=h:Person))
}
```



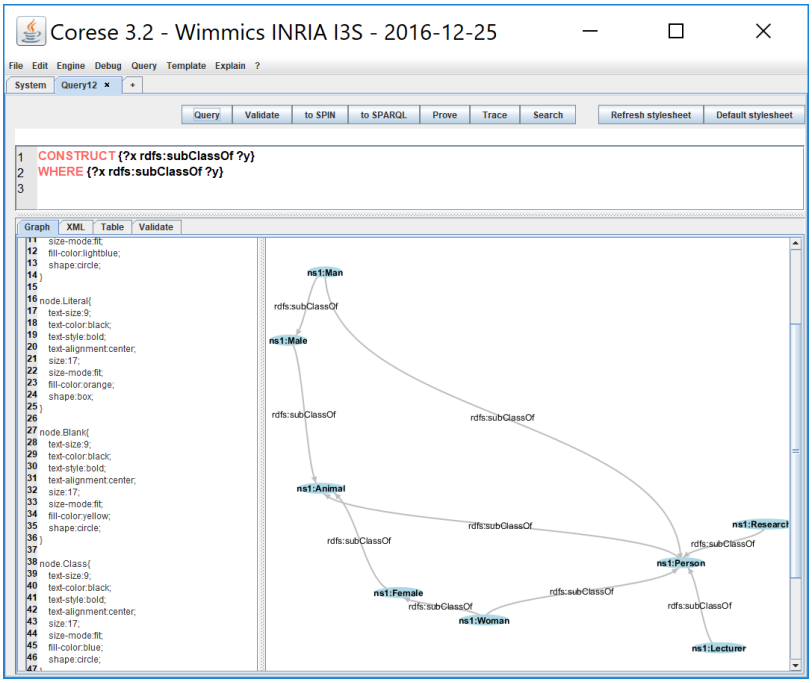
7. Rebuild the hierarchy of Classes (CONSTRUCT) considering only the classes in the humans.rdfs schema

query:

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

WHERE {?x rdfs:subClassOf ?y}

screenshot:



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

Query (Depending on the interpretation of the question, solution 1 gives signature only when the class has subclass, while solution 2 gives every signature relations appeared on the ontology file.):

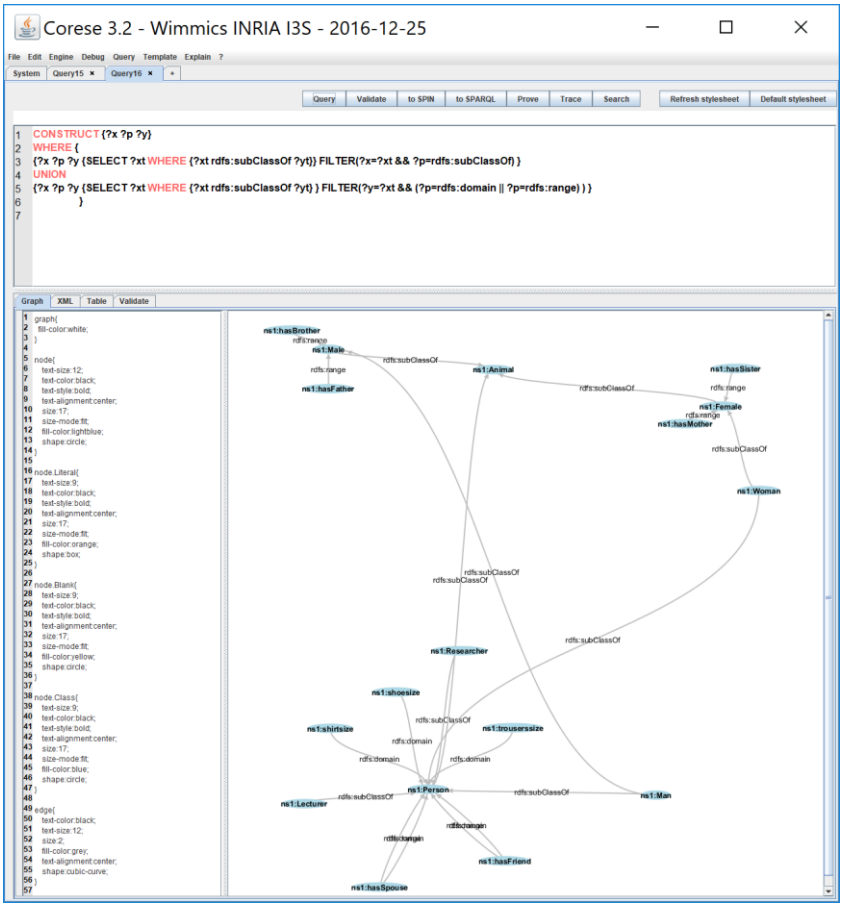
[Solution 1: my solution]

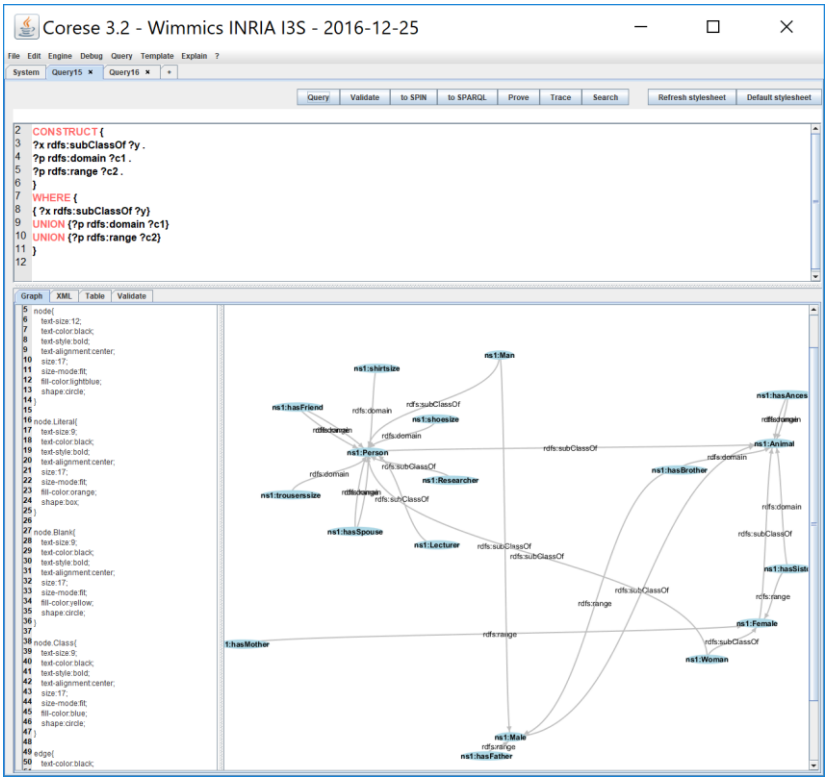
```
CONSTRUCT {?x ?p ?y}
WHERE {
        {?x ?p ?y {SELECT ?xt WHERE {?xt rdfs:subClassOf ?yt}} FILTER(?x=?xt && ?p=rdfs:subClassOf) }
        UNION
        {?x ?p ?y {SELECT ?xt WHERE {?xt rdfs:subClassOf ?yt} } FILTER(?y=?xt && (?p=rdfs:domain ||
?p=rdfs:range) ) }
        }
```

[Solution 2: inclass solution]

```
CONSTRUCT {
        ?x rdfs:subClassOf ?y .
        ?p rdfs:domain ?c1 .
        ?p rdfs:range ?c2 .
}
WHERE {
        { ?x rdfs:subClassOf ?y}
        UNION {?p rdfs:domain ?c1}
        UNION {?p rdfs:range ?c2}
}
```

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 h: <http://www.inria.fr/2007/09/11/humans.rdfs#> .
SELECT * WHERE {?x a h:Person .}
```

Number of results before:

7 results.

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:

Now I had 17 results, which are everyone appeared in the .rdf file. Many of them are registered not on Person class in .rdf, stead on another classes such as Man, Woman, etc. Thanks to the class hierarchy derived from .rdfs file, all of them are inferred as offsprings Person class.

Question 2

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

Query:

```
@prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#> .
SELECT ?x WHERE {
        {?x a h:Male .}
        UNION
        {{SELECT ?xt WHERE {?xt a h:Male .}}
        ?x h:hasSpouse ?y FILTER(?y=?xt) .}
        UNION
        {{SELECT ?xt WHERE {?xt a h:Male .}}
        ?y h:hasSpouse ?x FILTER(?y=?xt) .}
}
```

Number of results and explanation:

9 results.

I interpreted this question is asking for query to pull 1) all Males regardless they have wives or not, and 2) their wives if any, while some of my friends thought it for wives and males only when males have wives. My query consists of three parts and their union: the first for Male resources, and the second and third for their Spouses using subquery. There are three Males with wives, three Males without wives, and three wives. These results do not include the Spouse for Person with male names if they are not registered as Male explicitly (e.g. David.)

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:

```
<Man rdf:ID="Lucas">
```

```
        <shoesize rdf:datatype="&xsd;integer" >7</shoesize>
        <trouserssize rdf:datatype="&xsd;integer" >28</trouserssize>
        <age rdf:datatype="&xsd;integer" >12</age>
        <shirtsize rdf:datatype="&xsd;integer" >8</shirtsize>
        <name>Lucas</name>
        <hasMother rdf:resource="#Catherine"/>
        <hasFather rdf:resource="#Karl"/>
    </Man>
```

Number of results before and after and explanation:

11 results. Karl and his wife Catherine were added. The ontology file has the property hasFather with class Male as its domain, therefore, when Karl was registered as the domain of hasFather in .rdf file, the query inferred Karl was a Male. Also, since Karl was a Male, his Spouse Christine was also identified and derived.

Please note that from here on, I did not reload the data prior to addition of {Lucas hasFather Karl}, just because there was no instruction to do so. Therefore, results may be different from the ones derived from the data before the triple addition.

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 h: <http://www.inria.fr/2007/09/11/humans.rdfs#> .
SELECT ?x ?y WHERE {
        {SELECT * WHERE {?xt a h:Lecturer .}}
        ?x a ?y FILTER(?x= ?xt)
}
```

Number of results and your explanation:

7 results but only 2 distinct persons appeared. The same person belongs to multiple types and all the types appeared on the results, and since Lecturer has no sub class, those 2 persons directly belong to Lecturer in .rdf file.

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 h: <http://www.inria.fr/2007/09/11/humans.rdfs#> .
SELECT ?x WHERE {
        ?x a h:Person .
        ?x a h:Male .
}
```

Your explanation of the result:

In .rdf file, Jack is a Man. In ontology file, Man is a subclass of both Person and Male. Therefore, he meets the query conditions.

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 h: <http://www.inria.fr/2007/09/11/humans.rdfs#> .
SELECT * WHERE {
        ?x h:hasAncestor ?y .
}
```

Your explanation of the result:

The hasAncestor is never used in the data, but in the ontology file, hasParent is a sub property of hasAncestor, and both hasFather has Mother are sub properties of hasParent. All of these are reflected in the results.

It is also worth noting that there's no transitivity in this results as such Catherine is mother of Lucas, and Laura is mother of Catherine, then Laura should be an ancestor of Lucas, but that relation is not in the results.

Question 5

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

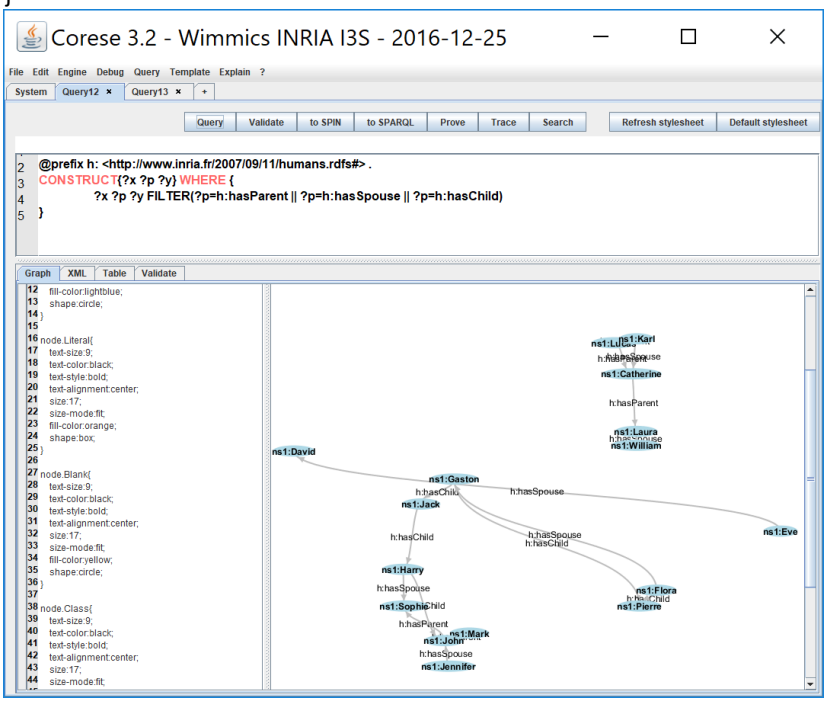
Query **(*note: I interpreted that this meant we needed couples no matter with or without children):**

```
@prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#> .
SELECT ?x ?p ?y WHERE {
        ?x ?p ?y FILTER(?p=h:hasParent || ?p=h:hasSpouse || ?p=h:hasChild)
}
```

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

Query:

```
@prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#> .
CONSTRUCT{?x ?p ?y} WHERE {
        ?x ?p ?y FILTER(?p=h:hasParent || ?p=h:hasSpouse || ?p=h:hasChild)
}
```



Question 6

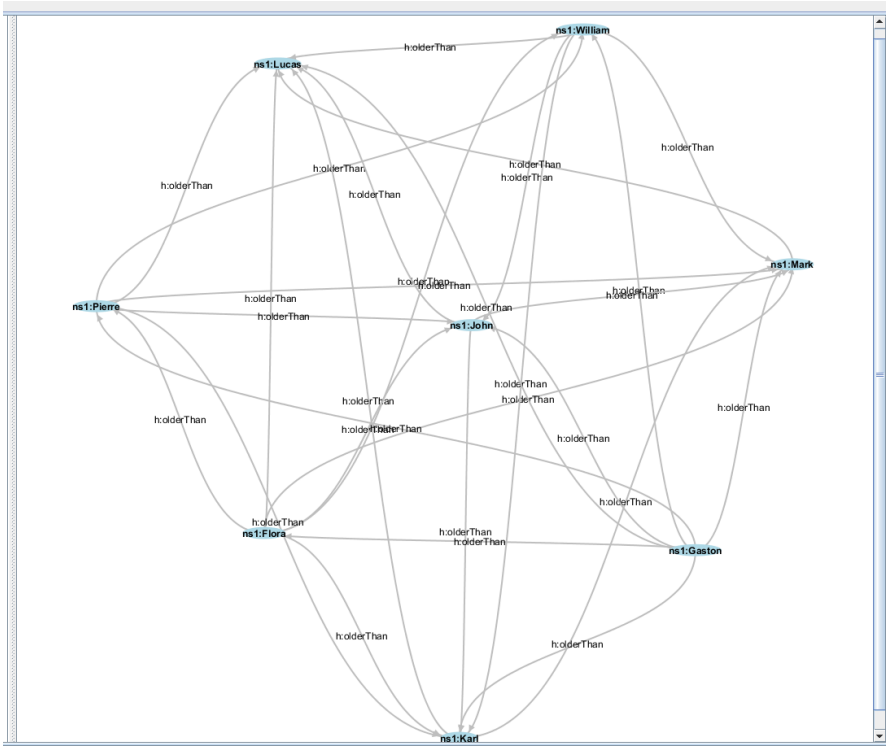
1. Declare the olderThan relationship in the schema to indicate between two people which is eldest and construct the arcs between peoples with a SPARQL query

Addition to schema:

```
<rdf:Property rdf:ID="olderThan">
        <domain rdf:resource="#Animal"/>
        <range rdf:resource="#Animal"/>
</rdf:Property>
```

Query:

```
@prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#> .
CONSTRUCT{?x1 h:olderThan ?x2} WHERE {
        ?x1 h:age ?a1 .
        ?x2 h:age ?a2 .
        FILTER (?a1 > ?a2)
}
        {?x1 h:age ?a1}
        UNION
        {?y h:hasParent ?x}
}
```

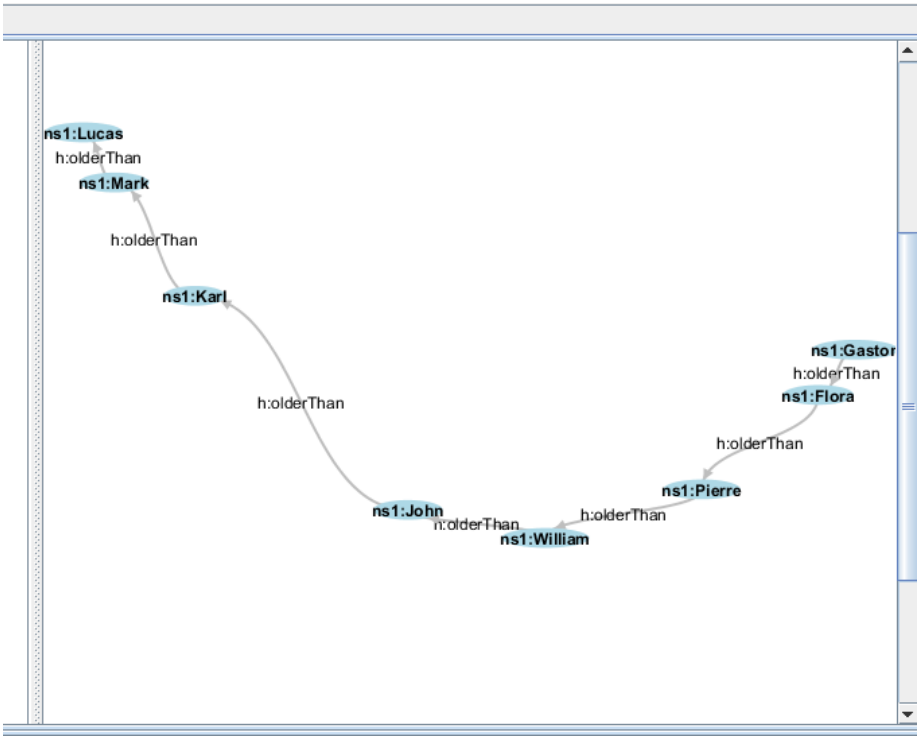


2. Find a query that generates only the minimum number of links without redundancy with olderThan transitivity.

Query:

```
@prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#> .
CONSTRUCT{ ?x1 h:olderThan ?x2}  WHERE {
        ?x1 h:age ?a1 .
        ?x2 h:age ?a2 .
        {
                SELECT ?x1t (MIN(?age_diff) as ?min_age_diff)  WHERE {
                        ?x1t h:age ?a1t.
```

```
                    ?x2t h:age ?a2t .
                    FILTER(?a1t > ?a2t)
                    BIND(?a1t - ?a2t as ?age_diff)
                } GROUP BY ?x1t
        }
        FILTER ((?a1 - ?a2) = ?min_age_diff && ?x1t=?x1)
} ORDER BY ?x1
```



Question 7

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

Query:

```
@prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#> .
SELECT DISTINCT ?x ?p ?y
WHERE {
        {SELECT ?xt WHERE{?xt h:name “John”}} .
        {SELECT ?xtt WHERE{?xtt rdfs:label ?ytt FILTER(CONTAINS(?ytt, “size”)) }} .
        ?x ?p ?y FILTER(?x=?xt && ?p=?xtt)
}
```

➔ Results: shoesize - 14, shirtsize – 12, trouserssize - 44

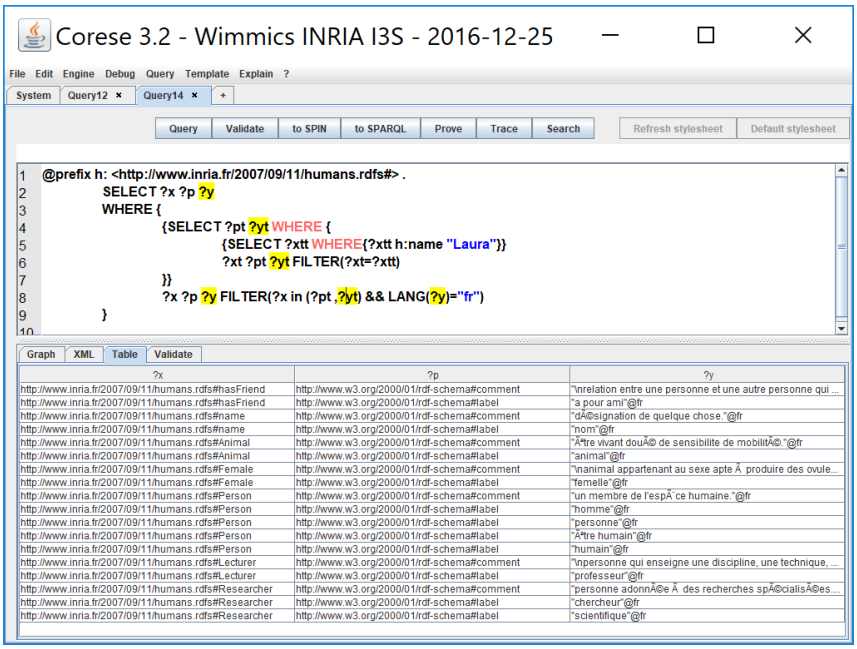
Question 8

Use the ontology to document your answers in natural language: write a query to find the types and properties of Laura in French. **(*Note: I interpreted this that I need to pull classed and properties which Laura has and their label or comment is in French.)**

Query:

```
SELECT ?x ?p ?y
```

```
WHERE {
        {SELECT ?pt ?yt WHERE {
                ?xt h:name "Laura" .
                ?xt ?pt ?yt .
        }}
?x ?p ?y .
FILTER(?x in (?pt, ?yt) && LANG(?y)="fr")
}
```



Day 05: 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: http://wimmics.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. **(*Note: I interpreted this that we were supposed to compare the very original ontology .rdfs file and incrementally modified ontology .owl file. Therefore, the result comparison "before addition to the schema" vs. "after addition to the schema" in each question shows more than what is caused solely by the addition to schema in each question, rather shows accumulated changes from the very original ontology .rdfs file.)**

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

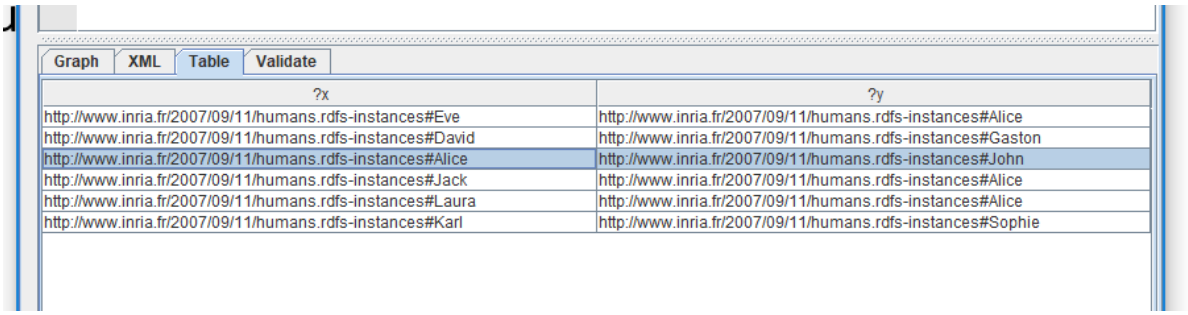
   Code added to the schema:

```
<rdf:Property rdf:ID="hasFriend">
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#SymmetricProperty"/>
        …..
</rdf:Property>
…..
<rdf:Property rdf:ID="hasSpouse">
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#SymmetricProperty"/>
        …..
</rdf:Property>
```

Query:

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

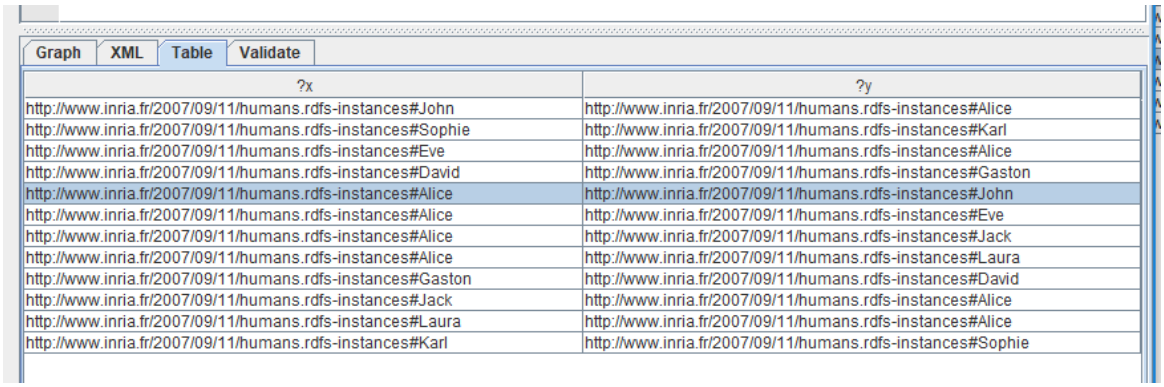
Result before addition to the schema:



Graph | XML | Table | Validate

?x	?y
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve	http://www.inria.fr/2007/09/11/humans.rdfs-instances#Alice
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#Alice	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#Alice
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura	http://www.inria.fr/2007/09/11/humans.rdfs-instances#Alice
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl	http://www.inria.fr/2007/09/11/humans.rdfs-instances#Sophie

Result after addition to the schema:



Graph | XML | Table | Validate

?x	?y
http://www.inria.fr/2007/09/11/humans.rdfs-instances#John	http://www.inria.fr/2007/09/11/humans.rdfs-instances#Alice
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Sophie	http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve	http://www.inria.fr/2007/09/11/humans.rdfs-instances#Alice
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#Alice	http://www.inria.fr/2007/09/11/humans.rdfs-instances#John
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Alice	http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Alice	http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Alice	http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston	http://www.inria.fr/2007/09/11/humans.rdfs-instances#David
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack	http://www.inria.fr/2007/09/11/humans.rdfs-instances#Alice
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura	http://www.inria.fr/2007/09/11/humans.rdfs-instances#Alice
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl	http://www.inria.fr/2007/09/11/humans.rdfs-instances#Sophie

Explanation:

There are many couples or friends who have only uni-directional hasSpouse or hasFriend relations in original .rdf file. Thanks to owl:SymmetricProperty, all of them are inferred to be bi-directional couples or friends.

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

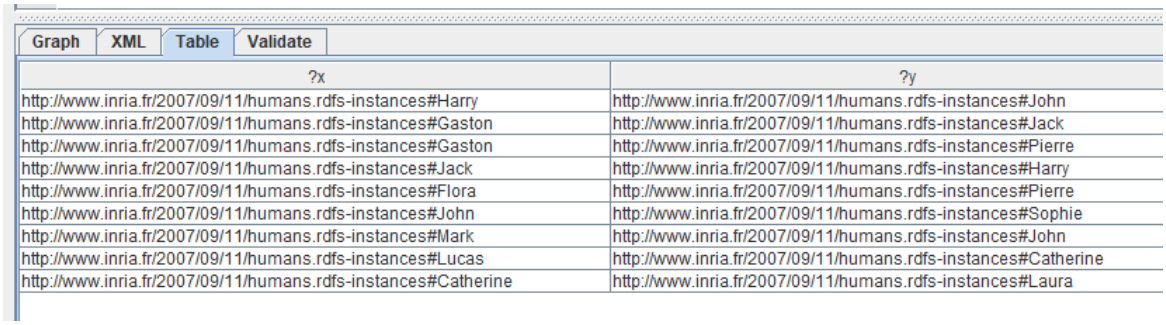
Code added to the schema:

```
<rdf:RDF xmlns:owl="http://www.w3.org/2002/07/owl#" ….>
…..
<rdf:Property rdf:ID="hasChild">
        <owl:inverseOf rdf:resource="#hasParent"/>
…..
```

Query:

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

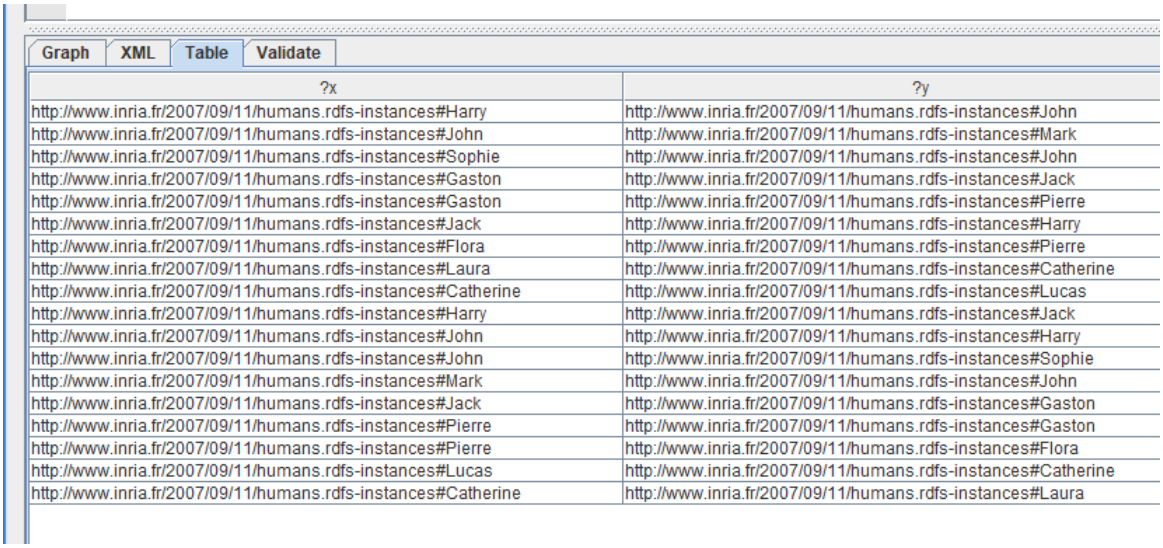
Result before addition to the schema:



Graph | XML | Table | Validate

?x	?y
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#Gaston	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#Jack	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-instances#Pierre
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#Catherine	http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura

Result after addition to the schema:



Graph | XML | Table | Validate

?x	?y
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#John	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-instances#John
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston	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#Jack	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-instances#Pierre
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura	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-instances#Lucas
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#Catherine	http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura

Explanation:

There are many parents-children who have only uni-directional hasChild or hasParent relations in original .rdf file. Thanks to owl:inverseOf defined in hasChild property, all of them are inferred to be bi-directional parents-children.

3. Declare `hasAncestor` as transitive property.

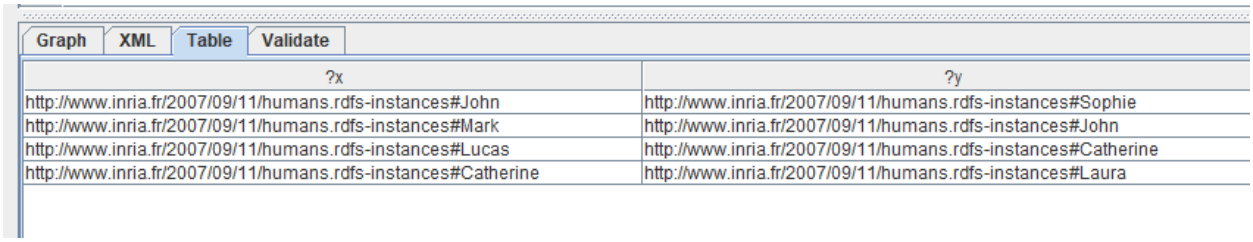
Code added to the schema:

```
…..
<rdf:Property rdf:ID="hasAncestor">
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#TransitiveProperty"/>
…..
```

Query:

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

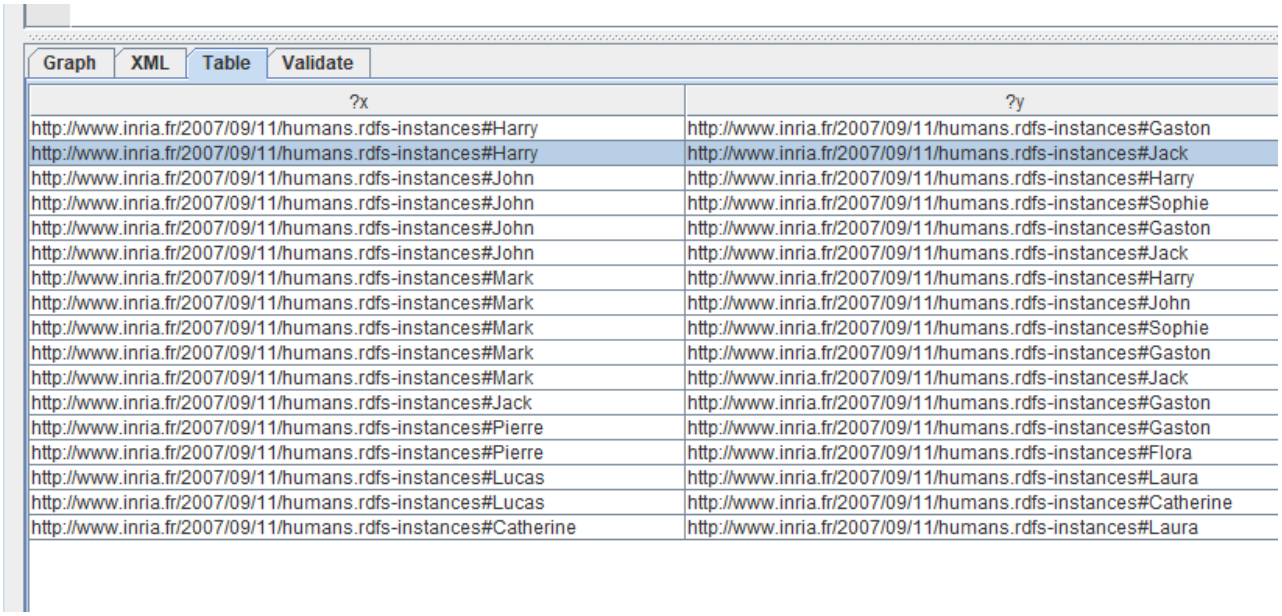
Result before addition to the schema:



Graph | XML | Table | Validate

?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#Catherine	http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura

Result after addition to the schema:



Graph | XML | Table | Validate

?x	?y
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Harry	http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston
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#John	http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston
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#Mark	http://www.inria.fr/2007/09/11/humans.rdfs-instances#Harry
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#Mark	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#Gaston
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark	http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack
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#Laura
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#Catherine	http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura

Explanation:

There's no directly assigned hasAncestor in .rdf file but thanks to hasParent being sub property of hasAncestor the result before addition to the schema already has some triples. The result after addition to the schema even has more triples thanks to inverseOf assignment between hasChild and hasParent, and transitivity of hasAncestor, which represents every possible combination of offspring-ancestor relations.

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:

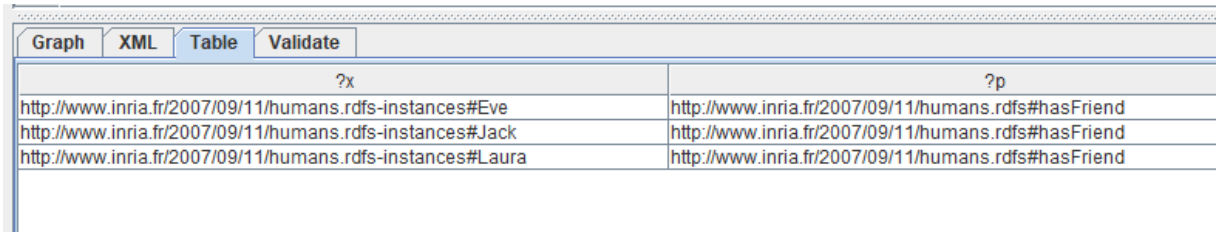
```
…..
<Class rdf:ID="Male">
        <owl:disjointWith rdf:resource="#Female"/>
        …..
</Class>
…..
```

➔ Now, data is with a constraint violation on Alice who is intentionally set Woman and Man at the same time.

Query:

```
select * where {?x ?p <http://www.inria.fr/2007/09/11/humans.rdfs-instances#Alice>}
```

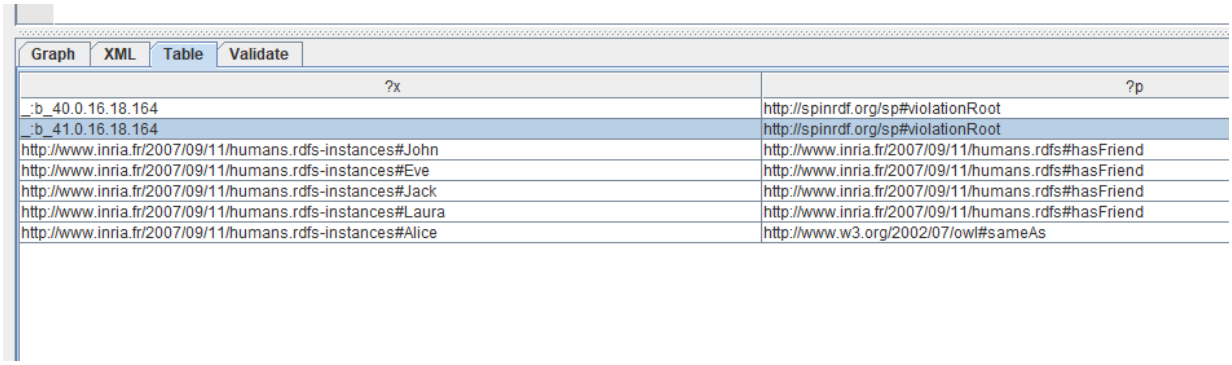
Result before addition to the schema:



Graph | XML | Table | Validate

?x	?p
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#Jack	http://www.inria.fr/2007/09/11/humans.rdfs#hasFriend
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura	http://www.inria.fr/2007/09/11/humans.rdfs#hasFriend

Result after addition to the schema:



Graph | XML | Table | Validate

?x	?p
_:b_40.0.16.18.164	http://spinrdf.org/sp#violationRoot
_:b_41.0.16.18.164	http://spinrdf.org/sp#violationRoot
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#Eve	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#hasFriend
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	http://www.w3.org/2002/07/owl#sameAs

Explanation:

Result after addition to the schema additionally has a few more friends due to SymmetricProperty of hasFriend in the prior question, and two properties saying violationRoot and one sameAs. They seem to represent the violation of disjoint classes Male vs. Female over their sub classes Man vs. Woman.

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

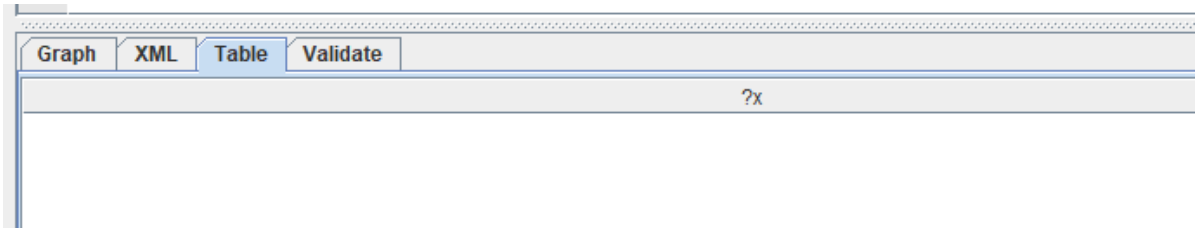
Code added to the schema:

```
…..
<rdf:Description rdf:about="#Professor">
        <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
        <owl:intersectionOf rdf:parseType ="Collection">
                <Class rdf:about="#Lecturer"/>
                <Class rdf:about="#Researcher"/>
        </owl:intersectionOf>
</rdf:Description>
…..
```

Query:

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

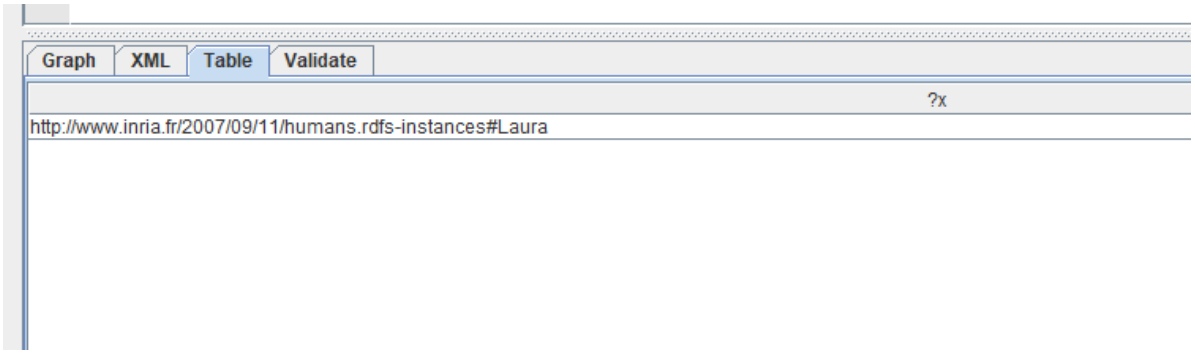
Result before addition to the schema:



Graph | XML | Table | Validate

?x

Result after addition to the schema:



Explanation:

Laura as an only person in .rdf file who is Lecturer and Researcher at the same time appears as Professor in OWL version as expected.

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

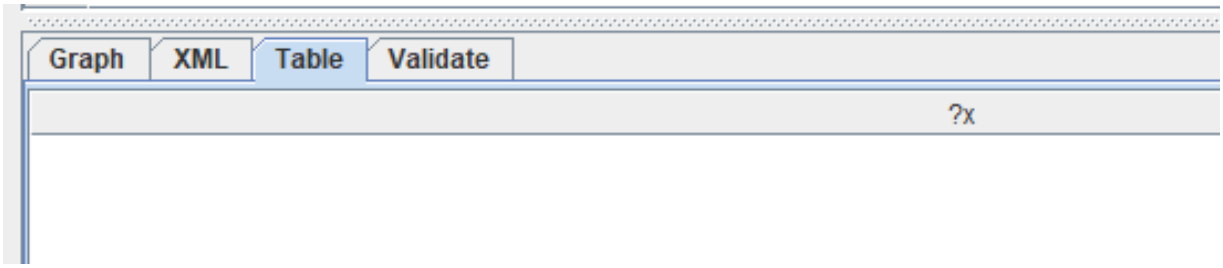
Code added to the schema:

```
…..
<rdf:Description rdf:about="#Academic">
        <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
        <owl:unionOf rdf:parseType ="Collection">
                <Class rdf:about="#Lecturer"/>
                <Class rdf:about="#Researcher"/>
        </owl:unionOf>
</rdf:Description>
…..
```

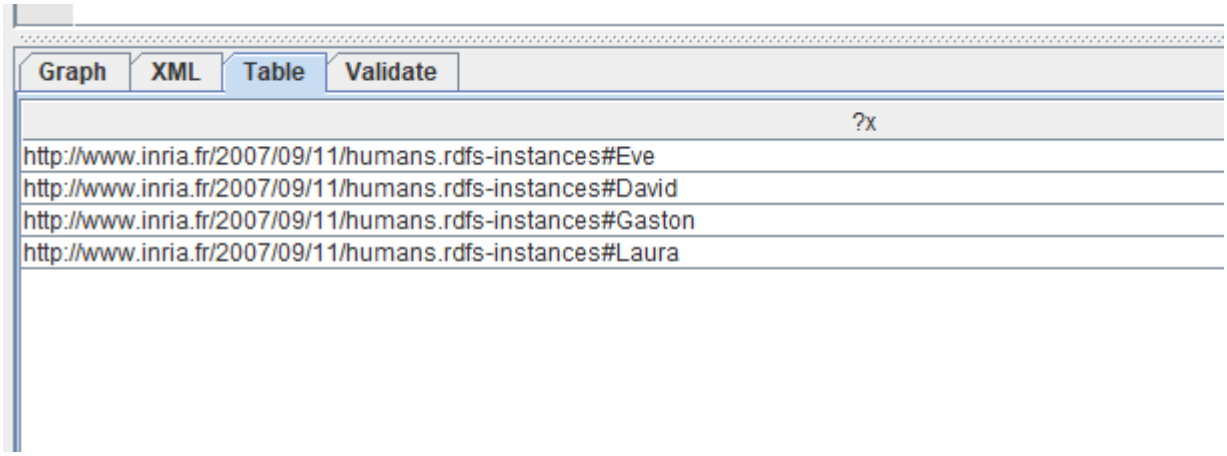
Query:

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

Result before addition to the schema:



Result after addition to the schema:



Explanation:

Results after the addition include many people who are either of Researcher or Lecturer including Laura who is both, just as I expect. The one before the addition has nothing.

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):

```
<rdf:Description rdf:about="#Professor">
        <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
        <owl:intersectionOf rdf:parseType ="Collection">
                <Class rdf:about="#Lecturer"/>
                <Class rdf:about="#Researcher"/>
        </owl:intersectionOf>
        <rdfs:subClassOf>
                <owl:Restriction>
                        <owl:onProperty rdf:resource="#mainEmployer" />
                        <owl:onClass rdf:resource="#University" />
                        <owl:qualifiedCardinality>1</owl:qualifiedCardinality>
                </owl:Restriction>
        </rdfs:subClassOf>
</rdf:Description>

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

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

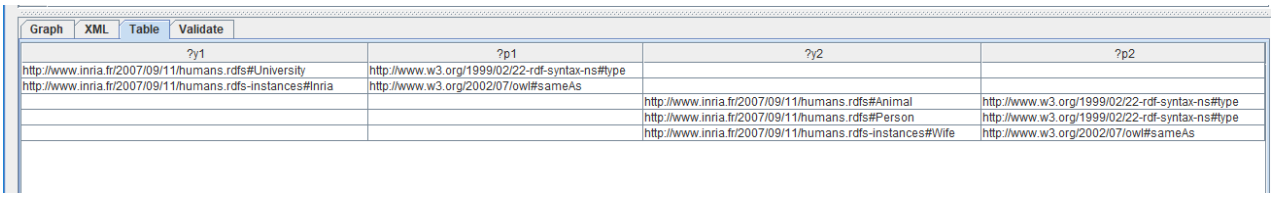
Code added to the data (just declare the main employer of a Professor):

```
<University rdf:ID="Inria" />

<Person rdf:ID="Wife" />

<Person rdf:about="#Fabien">
        <rdf:type rdf:resource="&humans;#Researcher"/>
        <rdf:type rdf:resource="&humans;#Lecturer"/>
        <mainEmployer rdf:resource="#Inria" />
</Person>

<Person rdf:about="#Moto">
        <rdf:type rdf:resource="&humans;#Researcher"/>
        <rdf:type rdf:resource="&humans;#Lecturer"/>
        <mainEmployer rdf:resource="#Wife" />
</Person>
```

Query:

```
@prefix i: <http://www.inria.fr/2007/09/11/humans.rdfs-instances#> .
select * where { {i:Inria  ?p1 ?y1} UNION {i:Wife ?p2 ?y2} }
```

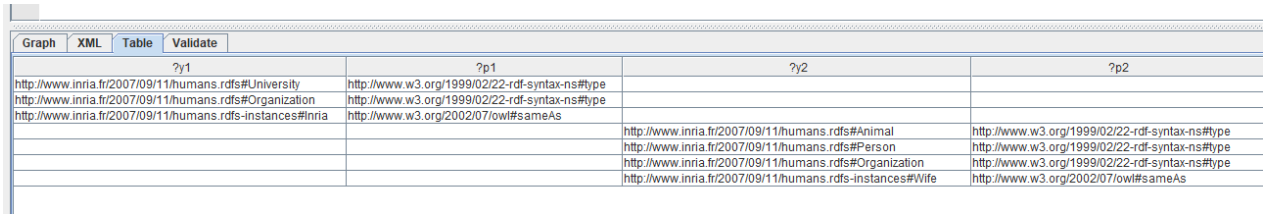
Result before addition to the schema:



Graph | XML | Table | Validate

?y1	?p1	?y2	?p2
http://www.inria.fr/2007/09/11/humans.rdfs#University	http://www.w3.org/1999/02/22-rdf-syntax-ns#type		
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Inria	http://www.w3.org/2002/07/owl#sameAs		
		http://www.inria.fr/2007/09/11/humans.rdfs#Animal	http://www.w3.org/1999/02/22-rdf-syntax-ns#type
		http://www.inria.fr/2007/09/11/humans.rdfs#Person	http://www.w3.org/1999/02/22-rdf-syntax-ns#type
		http://www.inria.fr/2007/09/11/humans.rdfs-instances#Wife	http://www.w3.org/2002/07/owl#sameAs

Result after addition to the schema:



Graph | XML | Table | Validate

?y1	?p1	?y2	?p2
http://www.inria.fr/2007/09/11/humans.rdfs#University	http://www.w3.org/1999/02/22-rdf-syntax-ns#type		
http://www.inria.fr/2007/09/11/humans.rdfs#Organization	http://www.w3.org/1999/02/22-rdf-syntax-ns#type		
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Inria	http://www.w3.org/2002/07/owl#sameAs		
		http://www.inria.fr/2007/09/11/humans.rdfs#Animal	http://www.w3.org/1999/02/22-rdf-syntax-ns#type
		http://www.inria.fr/2007/09/11/humans.rdfs#Person	http://www.w3.org/1999/02/22-rdf-syntax-ns#type
		http://www.inria.fr/2007/09/11/humans.rdfs#Organization	http://www.w3.org/1999/02/22-rdf-syntax-ns#type
		http://www.inria.fr/2007/09/11/humans.rdfs-instances#Wife	http://www.w3.org/2002/07/owl#sameAs

Explanation:

OWL file defines that University is a sub class of Organization, therefore one class is automatically added to #inria instance, Organization class. Wife instance is not an Organization without OWL but since I set the restriction of range of mainEmployer must be Organization, as long as Moto, a professor, has Wife as his mainEmployer, it is automatically inferred that Wife is an instance of Organization in the result after the addition to schema (,which is not true.)

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" >
        .....
        <owl:equivalentClass>
                <owl:Restriction>
                        <owl:onProperty rdf:resource="#hasParent" />
                        <owl:someValuesFrom rdf:resource="#Woman" />
                </owl:Restriction>
        </owl:equivalentClass>
</Class>
```

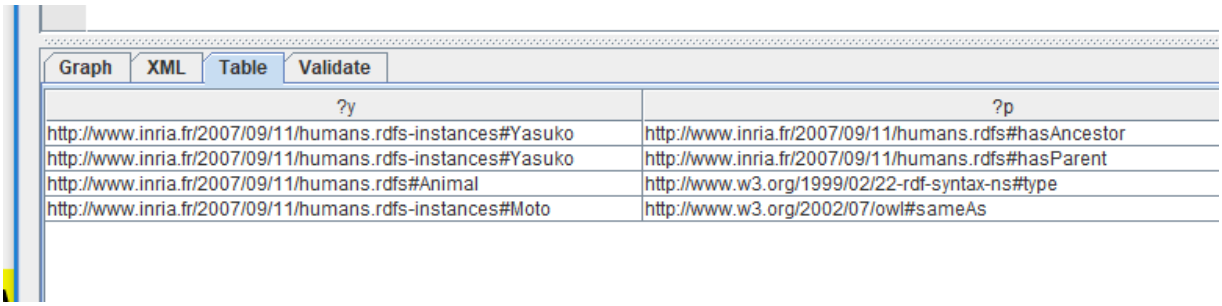
Data added to RDF file:

```
<Animal rdf:ID="Moto">
        <hasParent rdf:resource="#Yasuko"/>
</Animal>
<Woman rdf:ID="Yasuko" />
```

Query:

```
@prefix i: <http://www.inria.fr/2007/09/11/humans.rdfs-instances#>
select * where {
        i:Moto ?p ?y
}
```

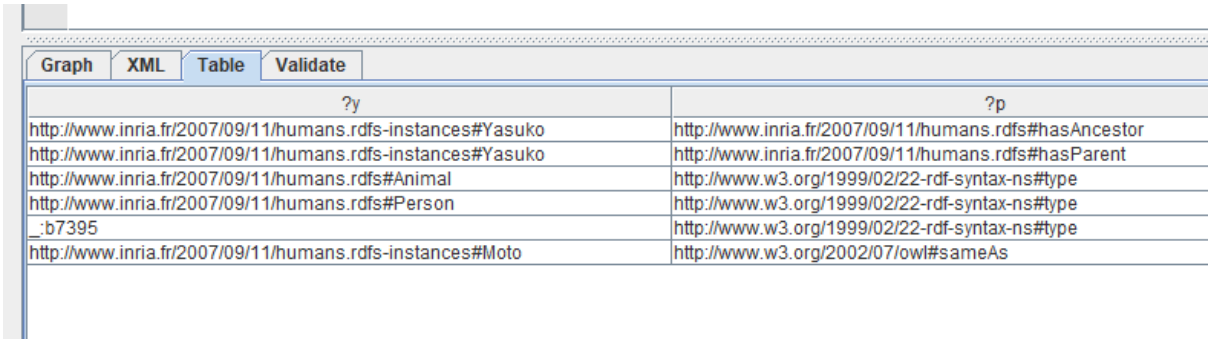
Result before addition to the schema:



Graph | XML | Table | Validate

?y	?p
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Yasuko	http://www.inria.fr/2007/09/11/humans.rdfs#hasAncestor
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Yasuko	http://www.inria.fr/2007/09/11/humans.rdfs#hasParent
http://www.inria.fr/2007/09/11/humans.rdfs#Animal	http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Moto	http://www.w3.org/2002/07/owl#sameAs

Result after addition to the schema:



Graph | XML | Table | Validate

?y	?p
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Yasuko	http://www.inria.fr/2007/09/11/humans.rdfs#hasAncestor
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Yasuko	http://www.inria.fr/2007/09/11/humans.rdfs#hasParent
http://www.inria.fr/2007/09/11/humans.rdfs#Animal	http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.inria.fr/2007/09/11/humans.rdfs#Person	http://www.w3.org/1999/02/22-rdf-syntax-ns#type
_:b7395	http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Moto	http://www.w3.org/2002/07/owl#sameAs

Explanation:

Although I only specify Moto as an Animal, OWL version successfully infers Moto is a Person based on restricted relation that anything (Animal, Male etc.) who hasParent to a Woman is Person through owl:enquivalentClass. (By the way, Yasuko is really my mom's name.)

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
   - in English
     - <DstiMajor> class is either one of <DataScicence> or <DataEngineer>.
   - in OWL
     - <DstiMajor> a owl:Class;
       owl:oneOf
       (<DataScience> <DataEngineer>) .

2. owl:unionOf
   - in English
     - <DstiStudents> class is union of <DataScicenceStudents> or <DataEngineerStudent>.
   - in OWL
     - <DstiStudents> a owl:Class;
       owl:unionOf
       (<DataScienceStudents> <DataEngineerStudents>) .

3. owl:intersectionOf

- in English
    - <HourToMyself> class is intersection of <HourWithoutKids>, <HourWithoutSpouse>, and <HourWithoutJob>.
- in OWL
    - <HourToMyself> a owl:Class;
      owl:intersectionOf
      (<HourWithoutKids> <HourWithoutSpouse> <HourWithoutJob>) .

4. owl:complementOf
    - in English
        - <HourWithoutJob> class is complement of <HourWithJob>.
    - in OWL
        - <HourWithoutJob> a owl:Class;
          owl:complementOf <HourWithJob> .

5. owl:disjointWith
   or owl:AllDisjointClasses
   or owl:disjointUnionOf
    - in English
        - <HourToMyself> class is disjoint with <HourWithKids> .
    - in OWL
        - <HourToMyself> a owl:Class;
          owl:disjointWith <HourWithKids>.

6. owl:ObjectProperty
    - in English
        - <hires> property is a type of owl:ObjectProperty, which represents a relation between classes.
    - in OWL
        - <hires> a owl:ObjectProperty .
          h:CompanyA <hires> h:StaffA .

7. owl:DatatypeProperty
    - in English
        - <isHiredAt> property is a type of owl:DatatypeProperty, which represents an employment date.
    - in OWL
        - <isHiredAt> a owl:DatatypeProperty . h:StaffA <isHiredAt> "2015-07-01"^^xsd:date .

8. owl:SymmetricProperty
   or owl:AsymmetricProperty
    - in English
        - <isInterviewedBy> property is a type of owl:AsymmetricProperty, which represents the object resource interviews the subject resource, with no possibility of the other way around.
    - in OWL
        - <isInterviewedBy> a owl:AsymmetricProperty .
          h:StaffA <isInterviewedBy> h:CompanyA .

9. owl:inverseOf
    - in English
        - <interviews> is a property owl:InverseOf <isInterviewedBy>.
    - in OWL
        - <interviews> owl:InverseOf <isInterviewedBy> .

10. owl:TransitiveProperty
    - in English

    - <largerThan> property as a owl:TransitiveProperty has transitivity like when A < B & B < C then A < C.
- in OWL
    - <largerThan> a owl:TransitiveProperty .
      h:A <largerThan> h:B .

11. owl:propertyDisjointWith
    - in English
        - Property <strictlyLargerThan> is owl:propertyDisjointWith property against <strictlySmallerThan> because they never share the same resources in subject and object.
    - in OWL
        - <strictlyLargerThan> owl:propertyDisjointWith <strictlySmallerThan> .

12. owl:ReflexiveProperty
    or owl:IrreflexiveProperty
    - in English
        - Property <knowsBest> is owl: ReflexiveProperty, because everybody knows themselves best.
    - in OWL
        - <knowsBest> a owl:ReflexiveProperty .

13. owl:propertyChainAxiom
    - in English
        - Property <parentFriend> is a property composed by a chain of properties <hasChild> and <hasParent>.
    - in OWL
        - <parentFriend> a rdf:Property ;
          owl:propertyChainAxiom (<hasChild> <hasParent>) .

14. owl:FunctionalProperty
    - in English
        - Property <hasBiologicalMother> as an owl:FunctionalProperty means that the subject resource only has one biological mother.
    - in OWL
        - <hasBilogicalMother> a owl:FunctionalProperty .
          H:PersonA <hasBilogicalMother> h:MotherA .

15. owl:InverseFunctionalProperty
    - in English
        - Property <hasMailAddress> as an owl:InverseFunctionalProperty means that a mail address belongs to only one subject resource (account).
    - in OWL
        - <hasMailAddress> a owl:InverseFunctionalProperty .
          AccountA <hasMailAddress> AddressA .

16. owl:hasKey
    - in English
        - An owl:hasKey can identify one and only one building through the objects by properties <country> <city> <zipCode> <street> <number>. (Whether this is true or not depends on the country's rule but here I assumed every country in scope has the similar one as France.)
    - in OWL
        - BuildingA owl:hasKey
          (<country> <city> <zipCode> <street> <number>) .

17. owl:allValuesFrom
    - in English
        - For all DSTI students, if they have free time, all the time is for study.

- in OWL
    - <DSTIStudent> a rdf:Class;
        rdfs:subClassOf
            [a owl:Restriction;
            owl:onProperty <hasFreeTime>;
            owl:allValuesFrom <Study>] .

18. owl:someValuesFrom
    - in English
        - For all francophones, they speak at least one language which is French.
    - in OWL
        - <Francophone> a rdf:Class;
            rdfs:subClassOf
                [a owl:Restriction;
                owl:onProperty <speaks>;
                owl:someValuesFrom <French>] .

19. owl:hasValue
    - in English
        - Human has exactly two legs.
    - in OWL
        - <Human> a rdf:Class;
            rdfs:subClassOf
                [a owl:Restriction;
                owl:onProperty <nbLegs>;
                owl:hasValue “2”] .

20. owl:maxCardinality
    or owl:minCardinality
    - in English
        - Undergrad students can belong to university at most for eight years.
    - in OWL
        - <UndergradStudent> a rdf:Class;
            rdfs:subClassOf
                [a owl:Restriction;
                owl:onProperty <nbYearsOfStay>;
                owl:maxCardinality 8] .

21. owl:qualifiedCardinality
    - in English
        - Non-handicapped human has exactly two resources as legs (one left leg and one right leg).
    - in OWL
        - <NonHandicappedHuman> a rdf:Class;
            rdfs:subClassOf
                [a owl:Restriction;
                owl:onProperty <has>;
                owl:onClass <Leg> ;
                owl:qualifiedCardinality 2] .
