
QUESTIONS FROM THE COURSES

Day 01: questions from the course.

Q1.1 Practice XML replace missing parts

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

<ANSWER HERE/>

A: short_title

B: /main_author

C:/

D:/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>

<ANSWER HERE/>

1. <http://www.wikidata.org/entity/Q23014205>

<!DOCTYPE html>	
--------------------	--

	<html class="client-nojs" lang="en" dir="ltr">
--	---------------------------------------------------

2. <http://www.wikidata.org/entity/Q23014205.json>

```
{"entities":{ "Q23014205":{ "pageid":25028548,"ns":0,"title":"Q23014205","lastrevid":872123137,"modified":"2019-03-02T12:03:45Z","type":"item","id":"Q23014205","labels":{ "fr":
```

3. <http://www.wikidata.org/entity/Q23014205.rdf>

```
<?xml version="1.0"?>
```

4. <http://www.wikidata.org/entity/Q23014205.ttl>

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

5. <http://www.wikidata.org/entity/Q23014205.nt>

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

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?

<ANSWER HERE/>

2.Londoner

4.<http://dbpedia.org/resource/Vienna>

5. Wien

Q1.4 WHO.IS?

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

<ANSWER HERE/>

1.Florian DUFOUR

2.Not available

3. SOCIETE EDITRICE DU MONDE

Q1.5 CURL

4. 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"/>
  </foaf:PersonalProfileDocument>

```

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/"
```

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=" http://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"/>

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 &amp; 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.css2020_01" type="text/css" rel="stylesheet"/><script type="text/javascript">

var BASE = '/';

var ua = window.navigator.userAgent;

var direactory = (~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.js2020_01" type="text/javascript"></script><script type="text/javascript">

uniprot.namespace = 'uniprot';

uniprot.releasedate = '2020_01';

</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/" />
```

6. Topic: Xavier Dolan; Format: jason

7. Topic: Distributed Artificial Intelligence for Distributed Corporate Knowledge Management
; Format:turtle

Q1.6 Recall five best practices of linked open data



<ANSWER HERE/>

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

Q1.7 Spotlight demo

Reproduce the demo:

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

<ANSWER HERE WITH THE COPY-PASTED ANOTATED TEXT OR SCREENSHOT/>

Muse (band)

From Wikipedia, the free encyclopedia

See also: [Muse \(disambiguation\)](#)

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

The screenshot shows the DBpedia Spotlight web interface. At the top, there is a logo with the text "DBpedia Spotlight". Below the logo, there are several input fields and controls. On the left, there is a "Confidence:" slider set to 0.5. To the right of the slider is a "Language:" dropdown menu set to "English". Below these are two checkboxes: "n-best candidates" (unchecked) and "SELECT TYPES..." (unchecked). To the right of these is an "ANNOTATE" button. In the center, there is a text area containing the Wikipedia text about Muse. At the bottom right of this text area is a "BACK TO TEXT" button.

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

This demo uses the statistical DBpedia Spotlight web service at <https://api.dbpedia-spotlight.org/en>.

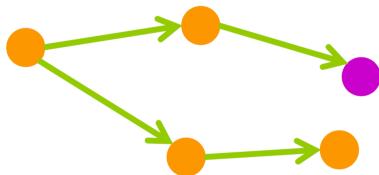
[How to cite this work](#)

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)

<ANSWER HERE/>

Subject; predicate; object



RDF triples can be seen as arcs
of a graph (**vertex, edge, vertex**)

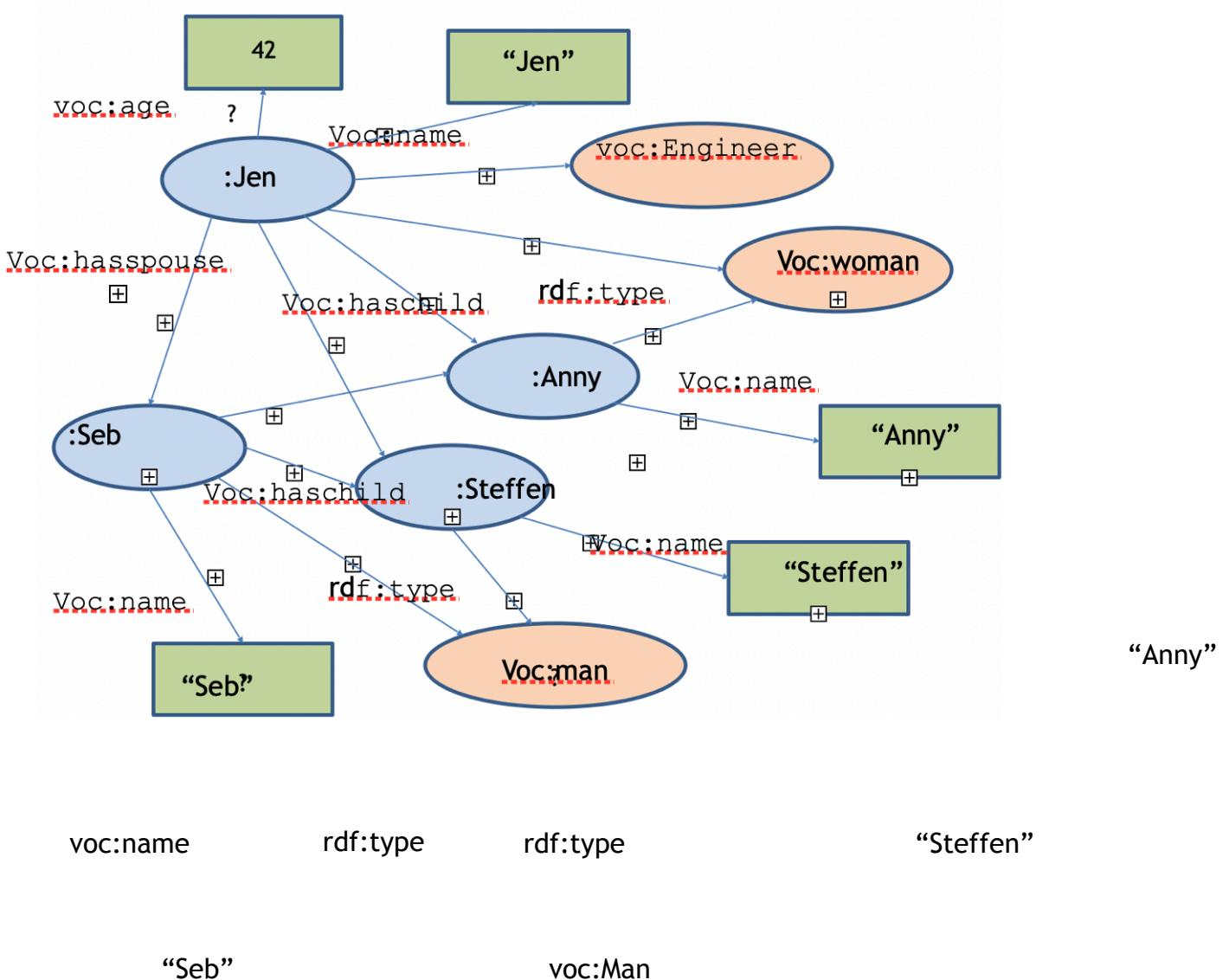
Q2.1 Fill the blanks

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

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

For each person we also explicitly specify the name

<ANSWER HERE BY REPLACING ALL THE QUESTION MARKS/>:



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">AAA </
voc:age>
  <BBB rdf:resource="#Seb"></BBB>
  <voc:hasChild rdf:resource="#Steffen"></voc:hasChild>
  <voc:hasChild>
    <rdf:Description rdf:about="#Anny">
      <voc:name>Anny</voc:name>
      <rdf:type CCC="&vocab;#Woman"></rdf:type>
    </rdf:Description>
  </voc:hasChild>
  <DDD rdf:resource="&vocab;#Engineer"></DDD>

```

```

</voc:Woman>
<EEE rdf:about="#Seb">
  <voc:name>Seb</voc:name>
  <voc:hasChild rdf:resource="#Steffen"></voc:hasChild>
  <voc:hasChild rdf:resource="#Anny"></voc:hasChild>
</EEE>
<voc:Man rdf:about="#Steffen">
  <voc:name>Steffen</voc:name>
</voc:Man>
</rdf:RDF>
A. 42
B. voc:hasspouse
C. rdf:about
D. voc:type
E. voc:Man

```

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 AAA BBB ;
  voc:age "42"^^xsd:string ;
  voc:hasChild <http://www.unice.fr/data#Anny>, <CCC>;
  voc:hasSpouse <http://www.unice.fr/data#Seb> ;
  voc:name "Jen" .
<http://www.unice.fr/data#Seb> DDD 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" .
<EEE> a FFF ;
  GGG HHH .

```

- A. ,
- B. voc:Woman
- C. <http://www.unice.fr/data#Steffen>
- D. a
- E. <http://www.unice.fr/data#Steffen>
- F. voc:Man
- G. voc:name
- H. "Steffen"

Q2.4 Visit me please

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

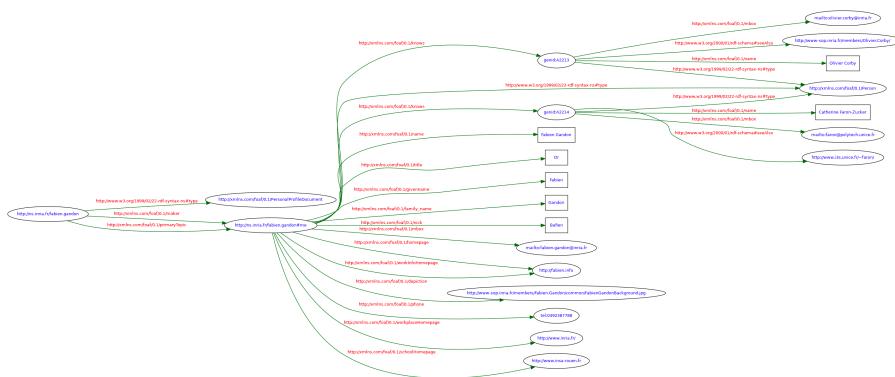
1. Get the RDF data from: <http://ns.inria.fr/fabien.gandon#me>
2. What is the syntax used?

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.easym2rdf.org/converter>
5. Visualize it also with:
<http://cltl.nl/visualrdf/>
<http://www.easym2rdf.org/converter> (PNG, SVG)
6. Adapt to your data and do it again

<ANSWER HERE/>

2.xml/rdf

3.



4.

Turtle/N3:

```
@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. svg 10 first lines

Number of triples parsed: 26

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<!-- Generated by graphviz version 2.40.1 (20161225.0304)
-->
<!-- Title: %3 Pages: 1 -->
<svg width="3713pt" height="307pt"
viewBox="0.00 0.00 3712.84 307.00" xmlns="http://www.w3.org/2000/
svg" xmlns:xlink="http://www.w3.org/1999/xlink">
<g id="graph0" class="graph" transform="scale(1 1) rotate(0)
translate(4 303)">
<title>%3</title>
<polygon fill="#ffffff" stroke="transparent" points="-4,4 -4,-303
3708.8362,-303 3708.8362,4 -4,4"/>

```

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>

```

<ANSWER HERE/>

73 pages of a report about music and danse

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?

<ANSWER HERE/>

XML syntax

```
@prefix bf: <http://id.loc.gov/ontologies/bibframe/> .  
@prefix bflc: <http://id.loc.gov/ontologies/bflc/> .  
@prefix cs: <http://purl.org/vocab/changeset/schema#> .  
@prefix identifiers: <http://id.loc.gov/vocabulary/identifiers/> .  
@prefix madsrdf: <http://www.loc.gov/mads/rdf/v1#> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .  
@prefix ri: <http://id.loc.gov/ontologies/RecordInfo#> .  
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .  
@prefix skosxl: <http://www.w3.org/2008/05/skos-xl#> .  
@prefix xml: <http://www.w3.org/XML/1998/namespace> .  
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

Q2.7
is

```
<http://id.loc.gov/authorities/subjects/sh85009907> a madsrdf:Occupation ;  
    madsrdf:authoritativeLabel "Authors, French" .  
  
<http://id.loc.gov/authorities/subjects/sh86005256> a madsrdf:Authority,  
    madsrdf:DeprecatedAuthority ;  
    rdfs:label "sh86005256" .
```

What
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> .  
}
```

<ANSWER HERE/>

syntax: TriG

In the graph algebra, <http://inria.fr/rr/doc.html> has a metadata subject
<http://data.bnf.fr/ark:/12148/cb121105993>

Q2.8 Visit Leukocyte surface antigen CD53

1. See HTML data from:
<http://www.uniprot.org/uniprot/Q61451>
2. Get RDF data from:
<http://www.uniprot.org/uniprot/Q61451.rdf>

3. What is the syntax?
4. Translate into Turtle/N3:
<http://rdf-translator.appspot.com/>
5. Any remark about the structure of the data?

<ANSWER HERE/>

Syntax: 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 . }
```

<ANSWER HERE/>

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
```

<ANSWER HERE/>

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

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

<ANSWER HERE/>

wd:Q23	George Washington
wd:Q76	Barack Obama
wd:Q91	Abraham Lincoln
wd:Q207	George W. Bush
wd:Q1124	Bill Clinton
wd:Q8007	Franklin Delano Roosevelt
wd:Q8612	Andrew Johnson
wd:Q9582	Gerald Ford

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

<ANSWER HERE/>

Select * where {?x oaf:Person ?y} limit 20

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

<ANSWER HERE/>

Select * where {?x oaf:Person ?y} 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

<ANSWER HERE/>

1.

Select ?age where{ ?x c:name ‘Fabien’;c:age ?age}

2.

Select ?name where{ ?x c:name ?name; c:age ?age. filter(?age<50)}

3.

Select * where{ ?x c:name ‘Fabien’;c:age ?age. filter(?age<50)}

4.

Select ?name where {?x c:name ‘Fabien’,?name .filter(?name!=‘Fabien’)}

5.

Select * where {?x ?p ?y; ?q ?y . filter(?p!=?q)}

6.

Select * where {?x ?p ?y, ?z . filter(?y!=?z)}

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 }
}
```

<ANSWER HERE/>

Yes.

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

<ANSWER HERE/>

Select ?x

Where{ ?x ex:memberof ?org.

Filter (not exist{?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))
```

<ANSWER HERE/>

The ranking of the authors and the corresponding number of their published 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

<ANSWER HERE/>

A

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 }
}
```

<ANSWER HERE/>

Showing all the people who have acquaintances, if the acquaintance's name is recorded, showing the name, otherwise put John Doe instead.

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
```

<ANSWER HERE/>

Showing the name of people who have acquaintances and the corresponding average age of their acquaintances.

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

<ANSWER HERE/>

The screenshot shows a Mac OS X desktop with a GoToMeeting window open. The window title is "GoToMeeting File Edit View Webcams Zoom Window Audio Help". The status bar at the top right shows "26% battery" and the date "Wed 11:47 AM". The main content area displays a presentation slide with the following text:

practice:

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

Q 3.13

1. Find resources that study informatics or mathematics

```
select * where
{{?x c:study c:Informatics} union {?x c:study c:Mathematics}}
```

2. In addition return the name of the resource if it has a name

```
select * where
{{?x c:study c:Informatics} union {?x c:study c:Mathematics}}
optional {?x c:name ?name}
```

3. In addition return the graph where the name is given

```
select * where
{ {?x c:study c:Informatics} union {?x c:study c:Mathematics}
  optional {graph ?g {?x c:name ?name}} }
```

The slide features a small image of a hand writing on a whiteboard in the top right corner. A red bullseye icon with an arrow is positioned in the bottom right corner of the slide area. The Mac OS X dock at the bottom contains icons for various applications like Finder, Mail, and Safari.

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 } }
```

<ANSWER HERE/>

?x ?p ?y: g1

graph ?g { ?y ?q ?z }: g1

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

<ANSWER HERE/>

```
delete { ?x foaf:name ?n }
insert { ?x rdfs:label ?n }
where { ?x foaf:name ?n }
```

Q3.16 what is this query performing?

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

<ANSWER HERE/>

To change the age from string type to integer type

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

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

<ANSWER HERE/>

Construct

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

<ANSWER HERE/>

BCDE

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

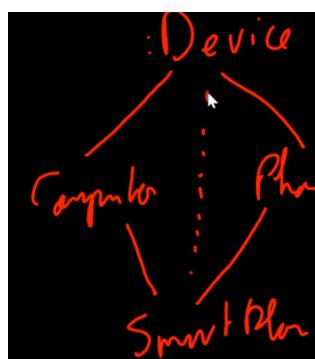
- describe classes of resources
- describe formulas of calculation for values of properties
- describe types of properties of resources
- document definitions in natural language
- sign and authenticate the authors of the definitions of classes and properties

<ANSWER HERE/>

ACD

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 .
```

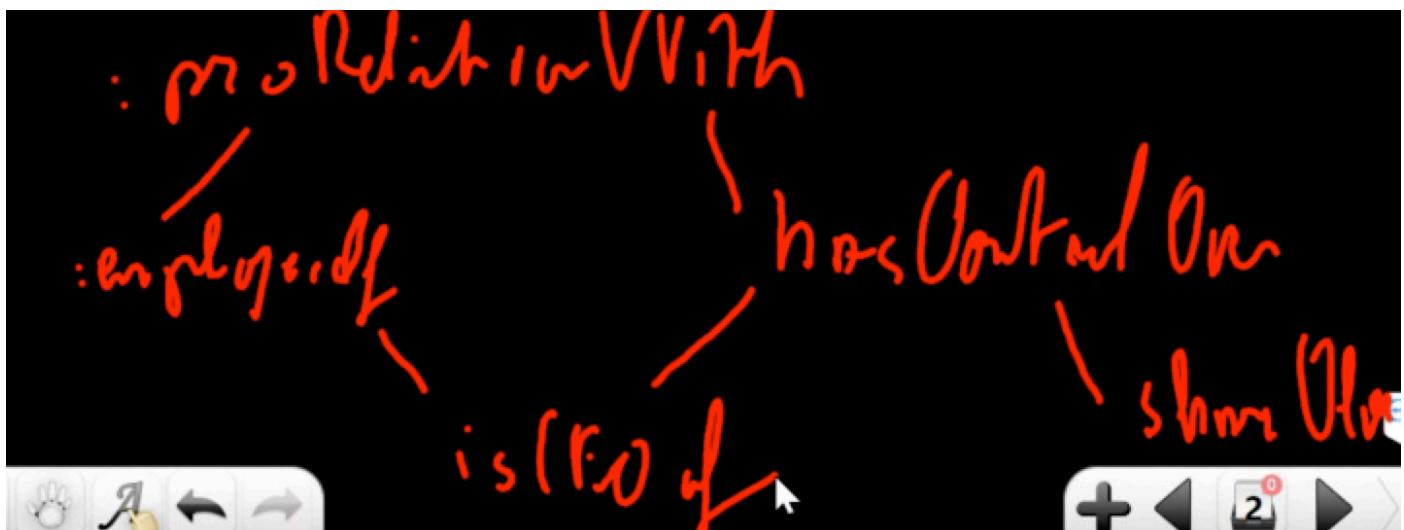


<ANSWER HERE/>

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 .
```

<ANSWER HERE/>



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 :isControoling .
:piloteOf rdfs:subPropertyOf :isControoling .
:isControoling rdfs:domain :Human ; rdfs:range :Object .
:driverOf rdfs:range :Car .
:piloteOf rdfs:domain :Adult ; rdfs:range :Plane .
```

<ANSWER HERE/>

Driver and pilote are types of controlling;

Human controls object;

Car is driven;

Adulte pilote plane

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

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

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

<ANSWER HERE/>

CDF

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.

Document

Agent

Report

Person

SecretReport

PressArticle

hasAuthor

Person

rdfs:label

isMemberOf

rdfs:seeAlso

hasAuthor

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 .
```

Jane: Woman

John: (Male, Human)

James: Male

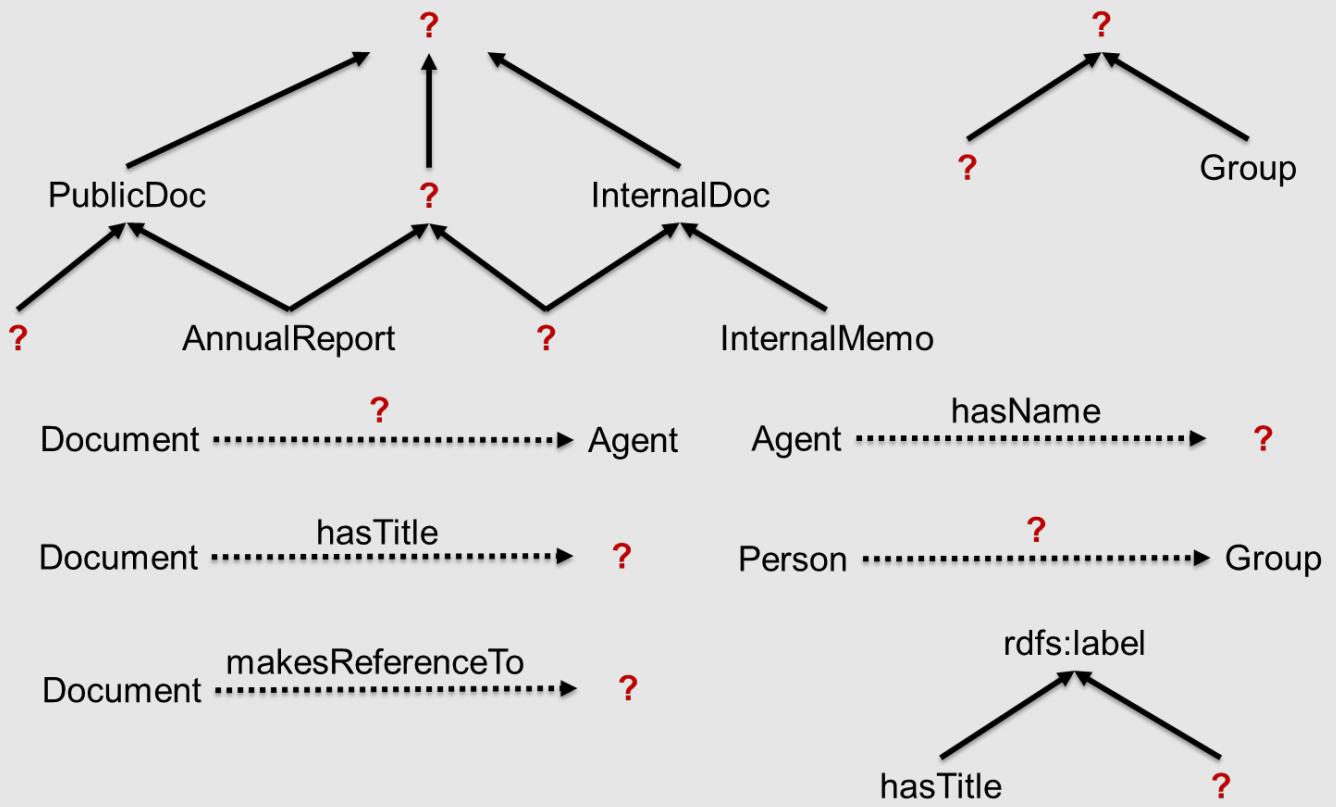
<ANSWER HERE/>

Q5.2 What are we defining and inferring?

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

C

ex:Jim a ex:Man, ex:Parent .



```
ex:Jack a ex:GrandFather .  
Jim:not necessarily a grandfather
```

quizz

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 .
```



Jack is a grandfather and he must be both parent and Man at the same time

<ANSWER HERE/>

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 .
```

<ANSWER HERE/>

Q5.4 What can we deduce?

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

JimmyPage:person

JamesPatrickPage = JimmyPage

JimmyHendrix != JimmyPage

<ANSWER HERE/>

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 ]  
    )  
] .
```

<ANSWER HERE/>

Unlucky person: it is a person and opposite of lucky

Q5.6 What can we deduce?

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

<ANSWER HERE/>

James and Jane are both human

Q5.7 What are we defining and inferring?

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

<ANSWER HERE/>

The value is a is from person, while for b and c, values are from personlist

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 .
```

<ANSWER HERE/>

It is impossible that Jane has two biological fathers

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 .
```

<ANSWER HERE/>

The preferred label is “B.A.-BA” or “b.a.-ba” and in English. The alternative label is “B-A-BA” or “b-a-ba” and in English. The hidden label is “BABA” or “baba” and in English

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

<ANSWER HERE/>

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

3.

Super-properties:

[skos:mappingRelation](#)
[skos:narrower](#)

Inverse of:

[skos: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?

<ANSWER HERE/>

Term Name: **FileFormat**

URI	http://purl.org/dc/terms/FileFormat
Label	File Format
Definition	A digital resource format.
Type of Term	Class
Subclass of	http://purl.org/dc/terms/MediaType

```
<http://inria.fr/rr/doc.html>
dc:creator <http://ns.inria.fr/john.doe#me> ; dc:title "Introduction to Algebra"@en ; dc:language "en" ;
dc:subject <#Algebra> ;
dc:date "2016-10-06" ;
dc:publisher <http://inria.fr> ;
dc:format "text/html" ; dc:type dcterms:Text .
```

cc:license [

a cc:License ;
cc:permits cc:CommercialUse cc:HighIncomeNationUse] ;

Such strict restrictions are not appropriate for this educational book.

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?

<ANSWER HERE/>

Type: ObjectProperty; InverseFunctionalProperty

It is the subPropertyOf `page`

domain: Agent

range: Document

Q6.5 practice:

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

<ANSWER HERE/>

```
@prefix admin: <http://webns.net/mvcb/> .  
@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#> .  
@prefix rel: <http://purl.org/vocab/relationship/> .
```

```
<> a foaf:PersonalProfileDocument ;  
    admin:errorReportsTo <mailto:leigh@ldodds.com> ;  
    admin:generatorAgent <http://www.ldodds.com/foaf/foaf-a-matic> ;  
    foaf:maker <#me> ;  
    foaf:primaryTopic <#me> .  
  
<#me> a foaf:Person ;  
    foaf:family_name "Li" ;  
    foaf:givenname "Jin" ;  
    foaf:mbox_sha1sum "92f06930ecc35ef7ce87b452cc9143fb1877eeeb" ;  
    foaf:name "Jin Li" ;  
    foaf:nick "Jinn" ;  
    foaf:schoolHomepage <https://www.datasciencetech.institute/fr/> ;  
    foaf:title "Mrs" .  
  
rel:friendOf rdf:type owl:SymmetricProperty ;  
rel:acquaintanceOf rdf:type owl:SymmetricProperty ;  
rel:neighborOf rdf:type owl:SymmetricProperty ;  
rel:worksWith rdf:type owl:SymmetricProperty ;  
rel:collaboratesWith rdf:type owl:SymmetricProperty ;
```

Q6.6 What does this mean?

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

<ANSWER HERE/>

BioRDF2DBLP is a linkset which links BioRDF and DBLP. The two datasets are Transitive Symmetric.

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.

<ANSWER HERE/>

It shows all the dataset endpoints.

```
SELECT DISTINCT * WHERE { ?ds a void:Dataset . ?ds rdfs:label "DBpedia-fr". ?ds ?p ?o}
```

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" .
```

<ANSWER HERE/>

The bar chart is generated by a plot and this plot is from a cvs file ‘stats1998’. And this file is allowed to be distributed.

Q6.9 What does this mean?

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

:c
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 .
```

<ANSWER HERE/>

C is an sql file which can be distributed ,downloaded, other descriptions are shown by dct.

Dataset R2RTransform12 used three resources:dataset db-employ,R2R-employ-mapping and the site '<http://xmlns.com/foaf/0.1/>'.

FoaFDump is generated by R2RTransform12 and also derived from db-employ dataset.Some descriptions are shown with void, provenance is shown with prov.

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)

<ANSWER HERE/>

3.Music

4.Revision: 2.1.5

5.yes

6.This vocabulary defines 54 classes and 153 properties

7.expressivity: OWL

Subclasses:	MusicGroup, SoloMusicArtist
Parent Class:	foaf:Agent
Properties:	activity, activity_end, activity_start, biography, compiled, discography, djmixed, fanpage, origin, ipi, remixed, sampled, supporting_musician

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

<ANSWER HERE/>

dc:title
2616----->Hypertext Transfer Protocol -- HTTP/1.1

dc:title
2396----->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>
```

<ANSWER HERE/>

cathy; name; Catherine Faron

Cathy; mbox; faron@i3s.unice.fr

Cathy; knows; Fabien Gandon

Q7.3 practice:

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

<ANSWER HERE/>

3. <<https://www.w3.org/TR/xhtml-rdfa-scenarios/scenario-2.html>>

dc:creator "Paul"@en .

The creator of this site is Paul in English.

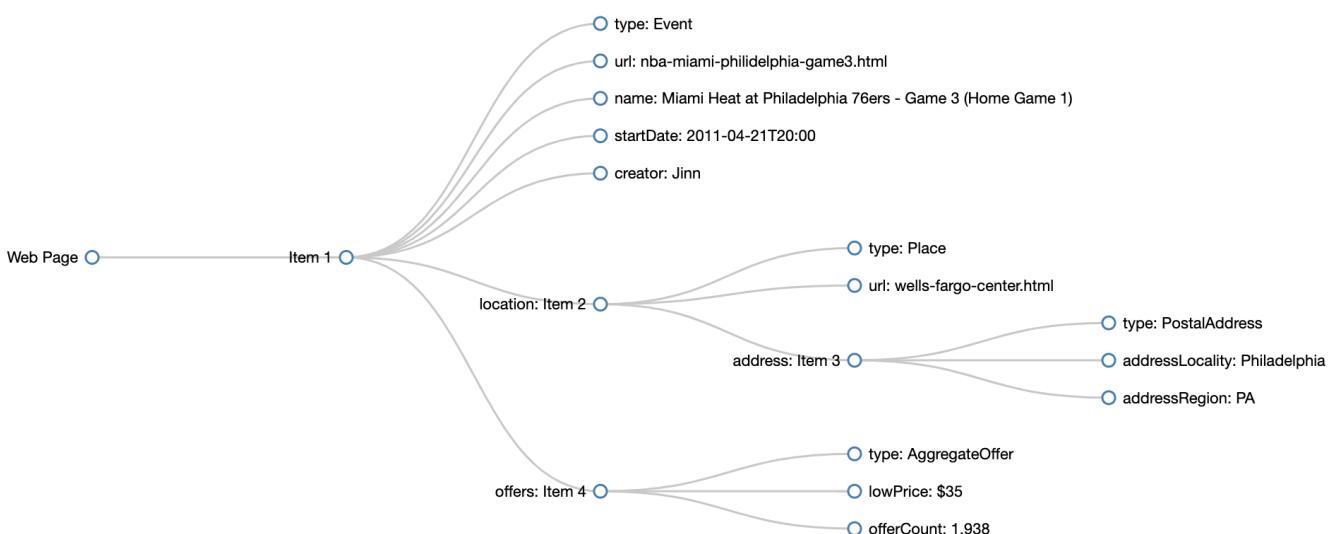
4. Rdaf

5. event;book;personal

Q7.4 Use the online tool to play with RDFa adding for instance a “creator” property

<https://rdfa.info/play/>

<ANSWER HERE/>



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/>

<ANSWER HERE/>

```
@prefix fb: <http://www.facebook.com/2008/fbml> .  
@prefix ns1: <http://www.w3.org/1999/xhtml/vocab#> .  
@prefix ns2: <http://www.facebook.com/2008/> .  
@prefix og: <http://ogp.me/ns#> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .  
@prefix xlink: <http://www.w3.org/1999/xlink#> .  
@prefix xml: <http://www.w3.org/XML/1998/namespace> .  
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .  
  
<https://www.imdb.com/title/tt8946378/?ref\_=hm\_fanfav\_tt\_2\_pd\_fp1> og:description "Directed by Rian Johnson. With Daniel Craig, Chris Evans, Ana de Armas, Jamie Lee Curtis. A detective investigates the death of a patriarch of an eccentric, combative family." ;  
    og:image "https://m.media-amazon.com/images/M/MV5BMGUwZjliMTAtNzAxZi00MWNiLWE2NzgtZGUxMGQxZjhNDRixkEyXkFqcGdeQXVvNjU1NzU3MzE@._V1_UY1200_CR90,0,630,1200_AL_.jpg" ;  
    og:site_name "IMDb" ;  
    og:title "Knives Out (2019) - IMDb" ;  
    og:type "video.movie" ;  
    og:url "http://www.imdb.com/title/tt8946378/" ;  
    ns2:fbmlapp_id "115109575169727" .  
  
<https://www.imdb.com/title/tt8946378/?ref\_=hm\_fanfav\_tt\_2\_pd\_fp1#imdbHeader-navDrawerOpen> ns1:role ns1:button .  
<https://www.imdb.com/title/tt8946378/?ref\_=hm\_fanfav\_tt\_2\_pd\_fp1#imdbHeader-searchClose> ns1:role ns1:button .  
<https://www.imdb.com/title/tt8946378/?ref\_=hm\_fanfav\_tt\_2\_pd\_fp1#imdbHeader-searchOpen> ns1:role ns1:button .  
<https://www.imdb.com/title/tt8946378/?ref\_=hm\_fanfav\_tt\_2\_pd\_fp1#nav-search-form> ns1:role ns1:search .  
[] ns1:role ns1:presentation .
```

Q7.6 Test JSON-LD online

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

<ANSWER HERE/>

Flattened:

```
{  
  "@graph": [  
    {"@id": "file:///base/data/home/apps/s%7Erdf-translator/2.408516547054015808/",  
     "@type": "http://xmlns.com/foaf/0.1/PersonalProfileDocument",  
     "http://webns.net/mvcb/errorReportsTo": {  
       "@id": "mailto:leigh@ldodds.com"  
     },  
     "http://webns.net/mvcb/generatorAgent": {  
       "@id": "http://www.ldodds.com/foaf/foaf-a-matic"  
     }  
  ]  
}
```

```

},
"http://xmlns.com/foaf/0.1/maker": {
    "@id": "file:///base/data/home/apps/s%7Erdf-translator/2.408516547054015808/#me"
},
"http://xmlns.com/foaf/0.1/primaryTopic": {
    "@id": "file:///base/data/home/apps/s%7Erdf-translator/2.408516547054015808/#me"
}
},
{
    "@id": "file:///base/data/home/apps/s%7Erdf-translator/2.408516547054015808/#me",
    "@type": "http://xmlns.com/foaf/0.1/Person",
    "http://xmlns.com/foaf/0.1/family_name": "Li",
    "http://xmlns.com/foaf/0.1/givenname": "Jin",
    "http://xmlns.com/foaf/0.1/mbox_sha1sum": "92f06930ecc35ef7ce87b452cc9143fb1877eeeb",
    "http://xmlns.com/foaf/0.1/name": "Jin Li",
    "http://xmlns.com/foaf/0.1/nick": "Jinn",
    "http://xmlns.com/foaf/0.1/schoolHomepage": {
        "@id": "https://www.datasciencetech.institute/fr/"
    },
    "http://xmlns.com/foaf/0.1/title": "Mrs"
}
]
}

```

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.

<ANSWER HERE/>

B D E

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

<ANSWER HERE/>

B D G H I

PRACTICAL SESSIONS

Day 02: Answers to the practical session on RDF.

Software requirements

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

Create RDF

Read carefully the following statements:

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

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

Code of validated RDF in N3 syntax:

<ANSWER HERE/>

```
@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:hasColleague <http://www.unice.fr/data#Catherine>, <http://www.unice.fr/data#Fabien> ;
voc:hasFather <http://www.unice.fr/data#Thomas> ;
voc:shoesize "36" ;
voc:trousersize "38";
voc:name "Jen" .

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

<http://www.unice.fr/data#Anny> a voc:Woman ;
  voc:name "Anny" .

<http://www.unice.fr/data#Steffen> a voc:Man ;
  voc:name "Steffen" .

<http://www.unice.fr/data#Catherine> a voc:Woman;
  voc:name "Catherine" .

<http://www.unice.fr/data#Fabien> a voc:Man;
  voc:name "Fabien" .

<http://www.unice.fr/data#Thomas> a voc:Man ;
  voc:name "Thomas" .

```

Code of validated RDF in XML syntax:

<ANSWER HERE/>

```

<!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:trousersize rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">38
</voc:trousersize>
  <voc:shoesize rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">36 </
voc:shoesize>
  <voc:hasspouse rdf:resource="#Seb"></voc:hasspouse >
  <voc:hasfather rdf:resource="#Thomas"></voc:hasfather >
  <voc:hasColleague rdf:resource="#Catherine"></voc:hasColleague>
  <voc:hasColleague rdf:resource="#Fabien"></voc:hasColleague>

  <voc:hasChild rdf:resource="#Steffen"></voc:hasChild>
  <voc:hasChild>
    <rdf:Description rdf:about="#Anny">
      <voc:name>Anny</voc:name>

```

```

<rdf:type rdf:about="#Woman"></rdf:type>
</rdf:Description>
</voc:hasChild>
<voc:type rdf:resource="#Engineer"></voc: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>
<voc:Man rdf:about="#Thomas">
  <voc:name>Thomas</voc:name>
</voc:Man>
<voc:Man rdf:about="#Fabien">
  <voc:name>Fabien</voc:name>
</voc:Man>
<voc:Woman rdf:about="#Catherine">
  <voc:name>Catherine</voc:name>
</voc:Woman>
</rdf:RDF>

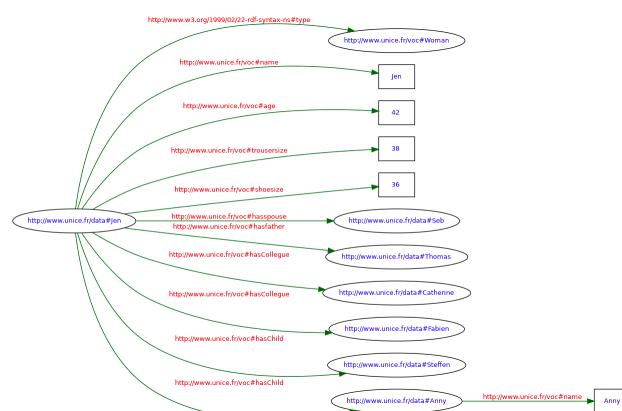
```

Triples:

Triples of the Data Model

Number	Subject	Predicate	Object
1	http://www.unice.fr/data#Jen	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.unice.fr/voc#Woman
2	http://www.unice.fr/data#Jen	http://www.unice.fr/voc#name	"Jen"
3	http://www.unice.fr/data#Jen	http://www.unice.fr/voc#age	"42"^^ http://www.w3.org/2001/XMLSchema#integer
4	http://www.unice.fr/data#Jen	http://www.unice.fr/voc#trousersize	"38"^^ http://www.w3.org/2001/XMLSchema#integer
5	http://www.unice.fr/data#Jen	http://www.unice.fr/voc#shoesize	"36"^^ http://www.w3.org/2001/XMLSchema#integer
6	http://www.unice.fr/data#Jen	http://www.unice.fr/voc#hasspouse	http://www.unice.fr/data#Seb
7	http://www.unice.fr/data#Jen	http://www.unice.fr/voc#hasfather	http://www.unice.fr/data#Thomas
8	http://www.unice.fr/data#Jen	http://www.unice.fr/voc#hasColleague	http://www.unice.fr/data#Catherine
9	http://www.unice.fr/data#Jen	http://www.unice.fr/voc#hasColleague	http://www.unice.fr/data#Fabien
10	http://www.unice.fr/data#Jen	http://www.unice.fr/voc#hasChild	http://www.unice.fr/data#Steffen
11	http://www.unice.fr/data#Jen	http://www.unice.fr/voc#hasChild	http://www.unice.fr/data#Anny
12	http://www.unice.fr/data#Anny	http://www.unice.fr/voc#name	"Anny"

Graph:



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.

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:

<ANSWER HERE/>

```
@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>

<ANSWER HERE/>

<http://ns.inria.fr/fabien.gandon#me> is the belongs to <http://ns.inria.fr/fabien.gandon>

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:

<ANSWER HERE/>

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?

<ANSWER HERE/>

xmlns="http://www.inria.fr/2007/09/11/humans.rdfs#"

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

<ANSWER HERE/>

By the base mechanism,

xmlns="http://www.inria.fr/2007/09/11/humans.rdfs#"

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?

<ANSWER HERE/>

rdf:resource

prefix + ":" + local name

Prefix of Namespace=rdf
Tag=resource

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

<ANSWER HERE/>

The name space is composed with the address of humans and ends with #

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

all the information:

<ANSWER HERE/>

Type: person

Name: "John"

Shoe size: 14

Age: 37

Shirtsize:12

Trouser size: 44

Has parent: Sophie

His father: Harry

His child: Mark

His friend: Alice

His spouse: Jennifer

6. Translate the file in turtle and save it as human_2013.ttl

10 first lines:

<ANSWER HERE/>

@prefix ns0: <<http://www.inria.fr/2007/09/11/humans.rdf#>> .

<<http://www.inria.fr/2007/09/11/humans.rdf#instances#Harry>>
a <<http://www.inria.fr/2007/09/11/humans.rdf#Man>> ;
ns0:name "Harry" ;
ns0:hasChild <<http://www.inria.fr/2007/09/11/humans.rdf#instances#John>> ;
ns0:hasSpouse <<http://www.inria.fr/2007/09/11/humans.rdf#instances#Sophie>> .

<<http://www.inria.fr/2007/09/11/humans.rdf#instances#John>>
a ns0:Person ;
ns0:name "John" ;

```

ns0:shoesize 14 ;
ns0:age 37 ;
ns0:shirtsize 12 ;
ns0:trouserssize 44 ;
ns0:hasParent <http://www.inria.fr/2007/09/11/humans.rdf-
instances#Sophie> .

```

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

<ANSWER HERE/>

```

<http://www.inria.fr/2007/09/11/humans.rdf-
instances#Laura>
a ns0:Person, ns0:Lecturer, ns0:Researcher ;
ns0:hasFriend <http://www.inria.fr/2007/09/11/humans.rdf-
instances#Alice> ;
ns0:name "Laura" .
<http://www.inria.fr/2007/09/11/humans.rdf-
instances#William>
a ns0:Person ;
ns0:hasSpouse <http://www.inria.fr/2007/09/11/humans.rdf-
instances#Laura> ;
<http://www.inria.fr/2007/09/11/humans.rdf-
instances#Catherine>
a ns0:Woman ;
ns0:hasMother <http://www.inria.fr/2007/09/11/humans.rdf-
instances#Laura> .

```

Information about Laura: person; lecturer; Researcher; name is “Laura”; husband is William; daughter is Laura; Friend is Alice

Day 02: Answers to the practical session on SHACL.

Software requirements

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

What is that shape

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

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

<ANSWER HERE/>

What is the type of that shape:

<ANSWER HERE/>

What is the target of that shape:

<ANSWER HERE/>

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

<ANSWER HERE/>

What is the severity level of that constraint?

<ANSWER HERE/>

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

<ANSWER HERE/>

Add your constraints

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

<ANSWER HERE/>

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

<ANSWER HERE/>

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

<ANSWER HERE/>

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

<ANSWER HERE/>

Day 03: Answers to the practical session on SPARQL.

Software requirements

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

Basic query on RDF human.rdf

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

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

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

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

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

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

Question 1:

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

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

Explanation:

<ANSWER HERE/>

It shows the information of three variables: ?s ,?p, ?o. (subject;prediction;object)

Screenshot:

<ANSWER HERE/>

?s	?o	?p
http://www.inria.fr/2007/09/11/humans.rdfsInstances#John	37	http://www.inria.fr/2007/09/11/humans.rdfs#age
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Mark	14	http://www.inria.fr/2007/09/11/humans.rdfs#age
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Gaston	102	http://www.inria.fr/2007/09/11/humans.rdfs#age
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Flora	95	http://www.inria.fr/2007/09/11/humans.rdfs#age
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Pierre	71	http://www.inria.fr/2007/09/11/humans.rdfs#age
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Lucas	12	http://www.inria.fr/2007/09/11/humans.rdfs#age
http://www.inria.fr/2007/09/11/humans.rdfsInstances#William	42	http://www.inria.fr/2007/09/11/humans.rdfs#age
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Karl	36	http://www.inria.fr/2007/09/11/humans.rdfs#age
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Harry	http://www.inria.fr/2007/09/11/humans.rdfsInstances#John	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Gaston	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Jack	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Gaston	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Pierre	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Jack	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Harry	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Jack	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Pierre	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Flora	http://www.inria.fr/2007/09/11/humans.rdfsInstances#John	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Flora	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Pierre	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Lucas	http://www.inria.fr/2007/09/11/humans.rdfsInstances#John	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Lucas	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Pierre	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Karl	http://www.inria.fr/2007/09/11/humans.rdfsInstances#John	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Harry	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Jack	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Gaston	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Pierre	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#John	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Harry	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#John	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Pierre	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Jack	http://www.inria.fr/2007/09/11/humans.rdfsInstances#John	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Jack	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Pierre	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Flora	http://www.inria.fr/2007/09/11/humans.rdfsInstances#John	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Flora	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Pierre	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Lucas	http://www.inria.fr/2007/09/11/humans.rdfsInstances#John	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Lucas	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Pierre	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Karl	http://www.inria.fr/2007/09/11/humans.rdfsInstances#John	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Harry	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Jack	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Gaston	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Pierre	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#John	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Harry	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#John	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Pierre	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Jack	http://www.inria.fr/2007/09/11/humans.rdfsInstances#John	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Jack	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Pierre	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Flora	http://www.inria.fr/2007/09/11/humans.rdfsInstances#John	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Flora	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Pierre	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Lucas	http://www.inria.fr/2007/09/11/humans.rdfsInstances#John	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Lucas	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Pierre	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Karl	http://www.inria.fr/2007/09/11/humans.rdfsInstances#John	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Harry	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Jack	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Gaston	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Pierre	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#John	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Harry	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#John	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Pierre	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Jack	http://www.inria.fr/2007/09/11/humans.rdfsInstances#John	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Jack	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Pierre	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Flora	http://www.inria.fr/2007/09/11/humans.rdfsInstances#John	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Flora	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Pierre	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Lucas	http://www.inria.fr/2007/09/11/humans.rdfsInstances#John	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Lucas	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Pierre	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Karl	http://www.inria.fr/2007/09/11/humans.rdfsInstances#John	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Harry	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Jack	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#Gaston	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Pierre	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#John	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Harry	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
http://www.inria.fr/2007/09/11/humans.rdfsInstances#John	http://www.inria.fr/2007/09/11/humans.rdfsInstances#Pierre	http://www.inria.fr/2

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.

<ANSWER HERE/>

Select all where the variable x has type of variable t. And the t must starts with the prefix h:<http://www.inria.fr/2007/09/11/humans.rdfs#> .

Run this query. How many answers do you get?

<ANSWER HERE/>

21 answers

Find John and his types in the answers.

<ANSWER HERE/>

John's types: Person

▼ result 2

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

<ANSWER HERE/>

```
SELECT ?p ?o
```

```
{
```

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

```
}
```

Results:

<ANSWER HERE/>

?p	?o
http://www.inria.fr/2007/09/11/humans.rdfs#age	37
http://www.inria.fr/2007/09/11/humans.rdfs#hasParent	http://www.inria.fr/2007/09/11/humans.rdfs-instances#Sophie
http://www.inria.fr/2007/09/11/humans.rdfs#name	John
http://www.inria.fr/2007/09/11/humans.rdfs#shirtsize	12
http://www.inria.fr/2007/09/11/humans.rdfs#shoesize	14
http://www.inria.fr/2007/09/11/humans.rdfs#trouserssize	44
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.inria.fr/2007/09/11/humans.rdfs#Person

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

Query

<ANSWER HERE/>

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

Results:

<ANSWER HERE/>

?_ast_v_0	?_ast_p_0	?_ast_v_1	?_ast_p_1
37	http://www.inria.fr/2007/09/11/humans.rdfs#age		
http://www.inria.fr/2007/09/11/humans.rdfs-instances...	http://www.inria.fr/2007/09/11/humans.rdfs#hasParent		
John	http://www.inria.fr/2007/09/11/humans.rdfs#name		
12	http://www.inria.fr/2007/09/11/humans.rdfs#shirtsize		
14	http://www.inria.fr/2007/09/11/humans.rdfs#shoesize		
44	http://www.inria.fr/2007/09/11/humans.rdfs#trouserss...		
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...	http://www.inria.fr/2007/09/11/humans.rdfs#hasChild
			http://www.inria.fr/2007/09/11/humans.rdfs-instances...
			http://www.inria.fr/2007/09/11/humans.rdfs#hasFather
			http://www.inria.fr/2007/09/11/humans.rdfs-instances...
			http://www.inria.fr/2007/09/11/humans.rdfs#hasFriend
			http://www.inria.fr/2007/09/11/humans.rdfs-instances...
			http://www.inria.fr/2007/09/11/humans.rdfs#hasSpouse

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.

<ANSWER HERE/>

Select all the information about the relation ‘someone and his/her spouse’

Run this query. How many answers do you get?

<ANSWER HERE/>

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:

<ANSWER HERE/>

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

select * where { ?x h:shoesize ?y }

Result:

<ANSWER HERE/>

?x	?y
http://www.inria.fr/2007/09/11/humans.rdfs-instances#John	14
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark	8
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston	11
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre	8
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas	7
http://www.inria.fr/2007/09/11/humans.rdfs-instances#William	10
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl	7

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

Query:

<ANSWER HERE/>

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

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

Result:

<ANSWER HERE/>

?x	?y
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Harry	
http://www.inria.fr/2007/09/11/humans.rdfs-instances#John	14
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark	8
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve	
http://www.inria.fr/2007/09/11/humans.rdfs-instances#David	
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Alice	
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston	11
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack	
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Flora	
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre	8
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura	
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jennifer	
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas	7
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Catherine	
http://www.inria.fr/2007/09/11/humans.rdfs-instances#William	10
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl	7

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

Query:

<ANSWER HERE/>

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
select ?x where { ?x h:shoesize ?y . ?x h:shirtsize ?z . filter(?y>8 || ?z>12)
}
```

Result:

<ANSWER HERE/>

?x

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#William

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:

<ANSWER HERE/>

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

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

<ANSWER HERE/>

5 answers with 2 duplicates

2. Find a way to avoid duplicates.

Query:

<ANSWER HERE/>

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

How many answers do you get then?

<ANSWER HERE/>

4

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

Query:

<ANSWER HERE/>

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

select distinct ?x where {?x a ?t . filter(strstarts(?t, h:))} minus {?x h:hasChild ?y} }

Answer: 12

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:

<ANSWER HERE/>

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

select ?x ?y where {?x h:age ?y}

Result:

<ANSWER HERE/>

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

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

Query:

<ANSWER HERE/>

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

select ?x ?y where {?x h:age ?y. filter(?y<18)}

How many answers do you get?

<ANSWER HERE/>

2

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:

<ANSWER HERE/>

```

prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
select * where { <http://www.inria.fr/2007/09/11/humans.rdfs#instances#Mark> h:age ?age.
bind(if (?age>18,"True","False")
as ?Adult)
}

```

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

Query:

<ANSWER HERE/>

```

prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
select * where { ?x h:age ?age.
bind(if (?age/2=FLOOR(?age/2),"True","False")
as ?even)
}

```

Question 8

- 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:

<ANSWER HERE/>

```

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

```

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

Query:

<ANSWER HERE/>

```

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

```

Question 9

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

Query:

<ANSWER HERE/>

Q6.1

```

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

```

Transfor to Construct

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

construct {?x h:hasChild ?y} where {?x h:hasChild ?y}

Result:

<ANSWER HERE/>

?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

Question 10

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

screenshots:

<ANSWER HERE/>

```
1 prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
2 Insert data {
3   h:Jin h:age 24
4 }
5 select * where {?x h:age 24}
```

Graph XML Table Validate

?x
http://www.inria.fr/2007/09/11/humans.rdfs#Jin

Question 11

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

Query:

<ANSWER HERE/>

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

select * where{ ?x h:shirtsize ?o . ?y h:shirtsize ?o. filter(?x != ?y) }

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

Query:

<ANSWER HERE/>

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

construct {?x h:seealso ?y} where{ ?x h:shirtsize ?o . ?y h:shirtsize ?o. filter(?x != ?y) }

3. Change the query into an insert.

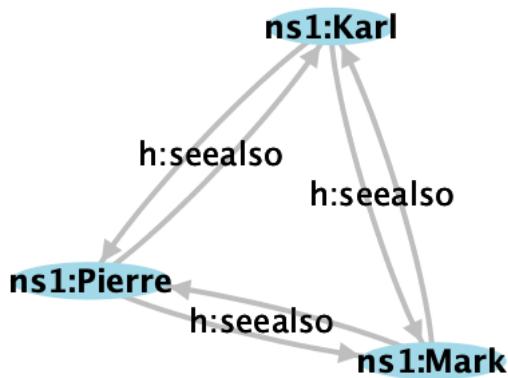
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>

Insert {?x h:seealso ?y} where{ ?x h:shirtsize ?o . ?y h:shirtsize ?o. filter(?x != ?y) }

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

screenshot:

<ANSWER HERE/>



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

Query:

<ANSWER HERE/>

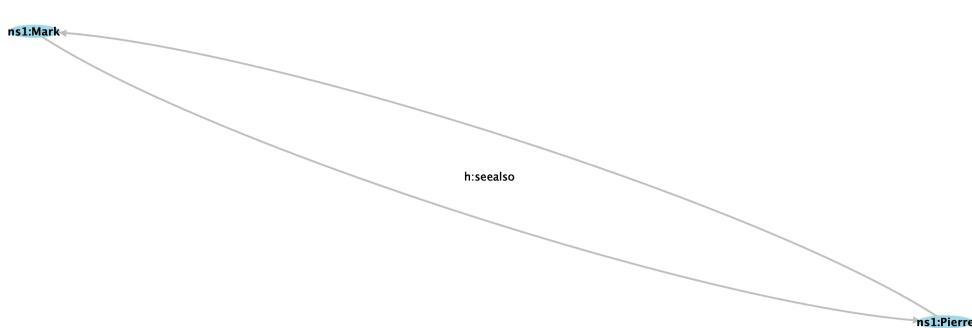
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>

Insert {?x h:seealso ?y} where{ ?x h:shoesize ?o . ?y h:shoesize ?o. filter(?x != ?y) }

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

screenshot:

<ANSWER HERE/>



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

Query:

<ANSWER HERE/>

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

select ?x where {?x h:seealso ?y}

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

?x	?o	?y
http://www.inria.fr/2007/09/11/humans.rdfs#Mark	8	http://www.inria.fr/2007/09/11/humans.rdfs#Pierre
http://www.inria.fr/2007/09/11/humans.rdfs#Pierre	8	http://www.inria.fr/2007/09/11/humans.rdfs#Mark
http://www.inria.fr/2007/09/11/humans.rdfs#Lucas	7	http://www.inria.fr/2007/09/11/humans.rdfs#Karl
http://www.inria.fr/2007/09/11/humans.rdfs#Karl	7	http://www.inria.fr/2007/09/11/humans.rdfs#Lucas

Question 12

1. Find the largest shoe size

Query:

<ANSWER HERE/>

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

SELECT (max(?shoesize) as ?maxshoesize)

WHERE { ?s h:shoesize ?shoesize }

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

Query:

<ANSWER HERE/>

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

SELECT (max(?shoesize) as ?maxshoesize) ?s

WHERE { ?s h:shoesize ?shoesize }

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

Query:

<ANSWER HERE/>

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

SELECT (avg(?shoesize) as ?aveshoesize)

WHERE { ?s h:shoesize ?shoesize }

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

Query:

<ANSWER HERE/>

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

SELECT (sum(?shoesize)/count(?shoesize) as ?avs)

WHERE { ?s h:shoesize ?shoesize }

Question 13

Find couples without children

Query:

<ANSWER HERE/>

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

select ?x ?y

where{{?x h:hasSpouse ?y} minus{?x h:hasChild ?o}}

Question 14

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

Insert:

<ANSWER HERE/>

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

```
insert data {h:Jolin h:age 33; h:shoesize 44}
```

```
select * where {h:Jolin ?d ?e}
```

Screenshot result:

<ANSWER HERE/>

?e	?d
33	http://www.inria.fr/2007/09/11/humans.rdfs#age
44	http://www.inria.fr/2007/09/11/humans.rdfs#shoesize

Question 15

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

query:

<ANSWER HERE/>

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

```
SELECT *
```

```
WHERE {
```

```
?s ?p ?o.
```

```
FILTER (?p IN (h:hasChild,h:hasSpouse, h:hasFather, h:hasMother, h:hasParent) )
```

```
}
```

screenshot:

<ANSWER HERE/>

?s	?o	?p
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#hasChild
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#hasChild
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#hasChild
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#hasChild
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#hasChild
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#hasChild
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#hasMother
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Catherine	http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura	http://www.inria.fr/2007/09/11/humans.rdfs#hasMother
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#hasParent
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Harry	http://www.inria.fr/2007/09/11/humans.rdfs-instances#Sophie	http://www.inria.fr/2007/09/11/humans.rdfs#hasSpouse
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve	http://www.inria.fr/2007/09/11/humans.rdfs-instances#David	http://www.inria.fr/2007/09/11/humans.rdfs#hasSpouse
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Flora	http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston	http://www.inria.fr/2007/09/11/humans.rdfs#hasSpouse
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jennifer	http://www.inria.fr/2007/09/11/humans.rdfs-instances#John	http://www.inria.fr/2007/09/11/humans.rdfs#hasSpouse
http://www.inria.fr/2007/09/11/humans.rdfs-instances#William	http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura	http://www.inria.fr/2007/09/11/humans.rdfs#hasSpouse
http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl	http://www.inria.fr/2007/09/11/humans.rdfs-instances#Catherine	http://www.inria.fr/2007/09/11/humans.rdfs#hasSpouse

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

<ANSWER HERE/>

Finding spouses in dbpedia and constructing the couple relation and showing their names

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

query:

<ANSWER HERE/>

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

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

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

insert data {db:AA db:spouse db:BB. db:AA foaf:name "AA@en". db:BB foaf:name "BB@en"}

select * where{db:AA ?x ?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: <https://project.inria.fr/corese/>

Create your own schema Family.rdfs

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

Your schema:

```
<ANSWER HERE/>

<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:voc="http://www.unice.fr/voc#"

>
<rdf:Description rdf:about="http://www.unice.fr/data#Steffen">
  <rdf:type rdf:resource="http://www.unice.fr/voc#Man"/>
  <voc:name>Steffen</voc:name>
</rdf:Description>
<rdf:Description rdf:about="http://www.unice.fr/data#Fabien">
  <voc:name>Fabien</voc:name>
  <rdf:type rdf:resource="http://www.unice.fr/voc#Man"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.unice.fr/data#Seb">
  <rdf:type rdf:resource="http://www.unice.fr/voc#Man"/>
  <voc:hasChild rdf:resource="http://www.unice.fr/data#Steffen"/>
  <voc:name>Seb</voc:name>
  <voc:hasChild rdf:resource="http://www.unice.fr/data#Anny"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.unice.fr/data#Jen">
  <voc:hasSpouse rdf:resource="http://www.unice.fr/data#Seb"/>
  <rdf:type rdf:resource="http://www.unice.fr/voc#Woman"/>
  <voc:hasColleague rdf:resource="http://www.unice.fr/data#Fabien"/>
  <voc:hasFather rdf:resource="http://www.unice.fr/data#Thomas"/>
```

```

<voc:hasColleague rdf:resource="http://www.unice.fr/data#Catherine"/>
<voc:hasChild rdf:resource="http://www.unice.fr/data#Anny"/>
<voc:shoesize>36</voc:shoesize>
<voc:age rdf:datatype="http://www.w3.org/2001/XMLSchema#string">42</voc:age>
<voc:trousersize>38</voc:trousersize>
<voc:name>Jen</voc:name>
<voc:hasChild rdf:resource="http://www.unice.fr/data#Steffen"/>
<rdf:type rdf:resource="http://www.unice.fr/voc#Engineer"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.unice.fr/data#Catherine">
<voc:name>Catherine</voc:name>
<rdf:type rdf:resource="http://www.unice.fr/voc#Woman"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.unice.fr/data#Anny">
<rdf:type rdf:resource="http://www.unice.fr/voc#Woman"/>
<voc:name>Anny</voc:name>
</rdf:Description>
<rdf:Description rdf:about="http://www.unice.fr/data#Thomas">
<rdf:type rdf:resource="http://www.unice.fr/voc#Man"/>
<voc:name>Thomas</voc:name>
</rdf:Description>
</rdf: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:

<ANSWER HERE/>

?x	?t
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/1999/02/22-rdf-syntax-ns#Property
http://www.unice.fr/data#Steffen	http://www.unice.fr/voc#Man
http://www.unice.fr/voc#name	http://www.w3.org/1999/02/22-rdf-syntax-ns#Property
http://www.unice.fr/data#Fabien	http://www.unice.fr/voc#Man
http://www.unice.fr/data#Seb	http://www.unice.fr/voc#Man
http://www.unice.fr/voc#hasChild	http://www.w3.org/1999/02/22-rdf-syntax-ns#Property
http://www.unice.fr/data#Anny	http://www.unice.fr/voc#Woman
http://www.unice.fr/data#Jen	http://www.unice.fr/voc#Woman
http://www.unice.fr/data#Jen	http://www.unice.fr/voc#Engineer
http://www.unice.fr/voc#hasSpouse	http://www.w3.org/1999/02/22-rdf-syntax-ns#Property
http://www.unice.fr/voc#hasColleague	http://www.w3.org/1999/02/22-rdf-syntax-ns#Property
http://www.unice.fr/voc#hasFather	http://www.w3.org/1999/02/22-rdf-syntax-ns#Property
http://www.unice.fr/data#Thomas	http://www.unice.fr/voc#Man
http://www.unice.fr/data#Catherine	http://www.unice.fr/voc#Woman
http://www.unice.fr/voc#shoesize	http://www.w3.org/1999/02/22-rdf-syntax-ns#Property
http://www.unice.fr/voc#age	http://www.w3.org/1999/02/22-rdf-syntax-ns#Property
http://www.unice.fr/voc#trousersize	http://www.w3.org/1999/02/22-rdf-syntax-ns#Property

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

Screenshot:

<ANSWER HERE/>

?p	?o
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2000/01/rdf-schema#Class
http://www.w3.org/2000/01/rdf-schema#subClassOf	file:///Users/jinli/Desktop/Family.rdfs.rdf#Woman
http://www.w3.org/2000/01/rdf-schema#subClassOf	file:///Users/jinli/Desktop/Family.rdfs.rdf#Man

Explanation:

<ANSWER HERE/>

Because we added the class Human, now it is shown in the result that Woman and Man are subclass of 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:

<ANSWER HERE/>

?x	?t
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/1999/02/22-rdf-syntax-ns#Property
http://www.w3.org/2000/01/rdf-schema#subPropertyOf	file:///Users/jinli/Desktop/Family.rdfs.rdf#hasChild
http://www.w3.org/2000/01/rdf-schema#subPropertyOf	file:///Users/jinli/Desktop/Family.rdfs.rdf#hasSpouse

Explanation:

<ANSWER HERE/>

Because we added the property familylink, now it is shown in the result that haschild and hasspouse are subproperty of Human

- 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:

<ANSWER HERE/>

?t	?x
file:///Users/jinli/Desktop/Family.rdfs.rdf#familyLink	http://www.w3.org/2000/01/rdf-schema#domain

Explanation:

<ANSWER HERE/>

The domain of familylink is FamilyMember

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?

<ANSWER HERE/>

```
<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="http://www.w3.org/2000/01/rdf-schema#" xml:base="http://www.inria.fr/2007/09/11/humans.rdfs">
```

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

<ANSWER HERE/>

Class: rdfs

subClassOf: rdfs

comment: rdfs

label: rdfs

Property: rdf

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

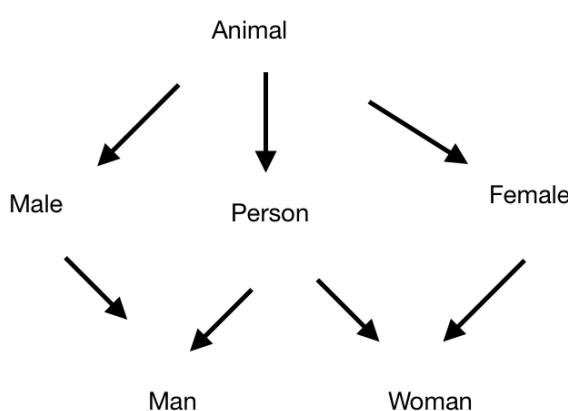
<ANSWER HERE/>

All classes can have age property because in the declaration of age property, no domain or range are specified.

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:

<ANSWER HERE/>



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:

<ANSWER HERE/>

SELECT DISTINCT ?type

WHERE {

?s a ?type.

}

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

query:

<ANSWER HERE/>

SELECT DISTINCT ?subject ?label ?supertype

WHERE {

{ ?subject a rdf:Class . } UNION { ?individual a ?subject . } .

OPTIONAL { ?subject rdfs:subClassOf ?supertype } .

OPTIONAL { ?subject rdfs:label ?label }

} ORDER BY ?subject

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:

<ANSWER HERE/>

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

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

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

SELECT *

WHERE {

h:shoesize rdfs:label ?shoesize_name; rdfs:comment ?shoesize_definition

}

answers:

<ANSWER HERE/>

?shoesize_name	?shoesize_definition
"shoe size"@en	"express in some way the approximate length of the shoes for a person."@en
"shoe size"@en	"taille, exprimée en points, des chaussures d'une personne."@fr
"size"@en	"express in some way the approximate length of the shoes for a person."@en
"size"@en	"taille, exprimée en points, des chaussures d'une personne."@fr
"pointure"@fr	"express in some way the approximate length of the shoes for a person."@en
"pointure"@fr	"taille, exprimée en points, des chaussures d'une personne."@fr

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:
<ANSWER HERE/>

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

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

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

SELECT *

WHERE {

 h:Person rdfs:label ?syno

 FILTER (langMatches(lang(?syno), "FR")).

}

answers:

<ANSWER HERE/>

"homme"@fr
"personne"@fr
"être humain"@fr
"humain"@fr

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

query:
<ANSWER HERE/>

SELECT ?uri ?label

WHERE {

?uri rdfs:label ?label .

filter(?label="size"@en)

}

answers:

<ANSWER HERE/>

?uri	?label
http://www.inria.fr/2007/09/11/humans.rdfs#shoesize	"size"@en
http://www.inria.fr/2007/09/11/humans.rdfs#shirtsize	"size"@en
http://www.inria.fr/2007/09/11/humans.rdfs#trouserssize	"size"@en

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

query:
<ANSWER HERE/>

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

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

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

```
SELECT *
```

```
WHERE {
```

```
    ?p rdfs:subClassOf h:Person
```

```
}
```

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

query:

```
<ANSWER HERE/>
```

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

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

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

```
construct { ?p rdfs:subClassOf ?j }
```

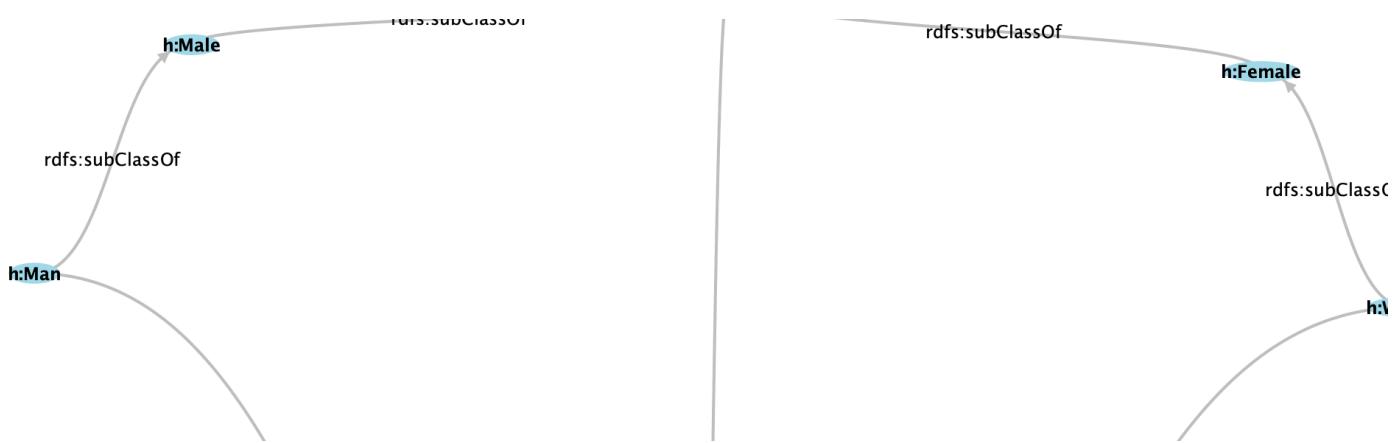
```
WHERE {
```

```
    ?p rdfs:subClassOf ?j
```

```
}
```

screenshot:

```
<ANSWER HERE/>
```



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

query:

```
<ANSWER HERE/>
```

```
select * WHERE {
```

```
    ?j rdfs:label ?lb.
```

```
    ?p rdfs:subClassOf ?j.
```

```
    ?p rdfs:label ?ll
```

```
}
```

screenshot:

```
<ANSWER HERE/>
```

?j	?lb	?p	?li
http://www.inria.fr/2007/09/11/humans.rdfs#Person	"homme"@fr	http://www.inria.fr/2007/09/11/humans.rdfs#Man	"man"@en
http://www.inria.fr/2007/09/11/humans.rdfs#Person	"homme"@fr	http://www.inria.fr/2007/09/11/humans.rdfs#Man	"homme"@fr
http://www.inria.fr/2007/09/11/humans.rdfs#Person	"homme"@fr	http://www.inria.fr/2007/09/11/humans.rdfs#Lecturer	"lecturer"@en
http://www.inria.fr/2007/09/11/humans.rdfs#Person	"homme"@fr	http://www.inria.fr/2007/09/11/humans.rdfs#Lecturer	"professeur"@fr
http://www.inria.fr/2007/09/11/humans.rdfs#Person	"homme"@fr	http://www.inria.fr/2007/09/11/humans.rdfs#Researcher	"researcher"@en
http://www.inria.fr/2007/09/11/humans.rdfs#Person	"homme"@fr	http://www.inria.fr/2007/09/11/humans.rdfs#Researcher	"scientist"@en
http://www.inria.fr/2007/09/11/humans.rdfs#Person	"homme"@fr	http://www.inria.fr/2007/09/11/humans.rdfs#Researcher	"chercheur"@fr
http://www.inria.fr/2007/09/11/humans.rdfs#Person	"homme"@fr	http://www.inria.fr/2007/09/11/humans.rdfs#Researcher	"scientifique"@fr
http://www.inria.fr/2007/09/11/humans.rdfs#Person	"homme"@fr	http://www.inria.fr/2007/09/11/humans.rdfs#Woman	"woman"@en
http://www.inria.fr/2007/09/11/humans.rdfs#Person	"homme"@fr	http://www.inria.fr/2007/09/11/humans.rdfs#Woman	"femme"@fr

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:

<ANSWER HERE/>

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#>
prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
select * where {
?x ?p h:Person}
```

Number of results before:

<ANSWER HERE/>

17

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:

<ANSWER HERE/>

28

Because it includes other primitives except type, like subClassOf, domain and range

Question 2

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

Query:

<ANSWER HERE/>

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#>
prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
select * where {
?x a h:Man; h:hasSpouse ?y }
```

Number of results and explanation:

<ANSWER HERE/>

1

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:

<ANSWER HERE/>

```
insert data{ h:Lucas h:hasFather h:Karl}
```

Number of results and your explanation:

Number of results before and after and explanation:

<ANSWER HERE/>

1

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:

<ANSWER HERE/>

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#>
prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
select * where {
  h:Lecturer rdf:type ?y
}
```

Number of results before and after and explanation:

<ANSWER HERE/>

Only 1 type

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:

<ANSWER HERE/>

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#>
prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
select * where {
  ?x a h:Male.
  ?x a h:Person}

```

Your explanation of the result:

<ANSWER HERE/>

Jack is both Man and Person.

Question 4

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

Query:

<ANSWER HERE/>

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#>
prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
select * where {
?a h:hasAncestor ?b}
```

Your explanation of the result:

<ANSWER HERE/>

It is used in the second position in triples

Question 5

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

Query:

<ANSWER HERE/>

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#>
prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
select * where {
?a h:hasSpouse?b.
?a h:hasChild ?c}
```

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

Query:

<ANSWER HERE/>

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#>
prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
construct {?a ?b ?c}
where {
?a h:hasSpouse?b.
?a h:hasChild ?c
}
```

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:

<ANSWER HERE/>

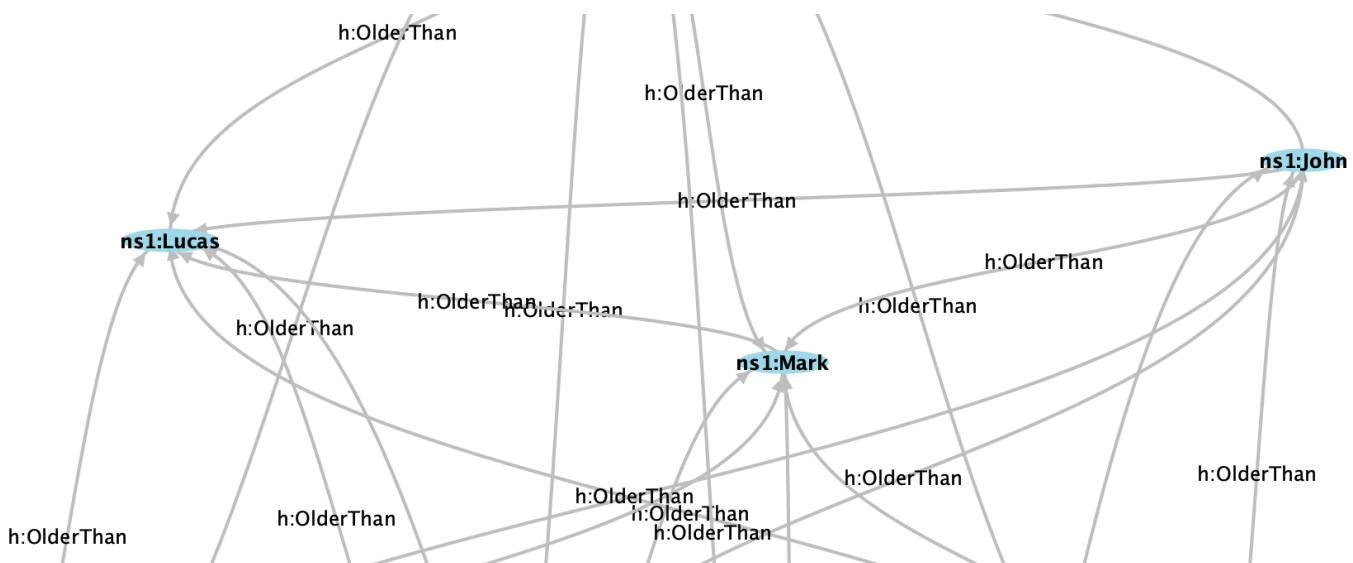
```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#>
```

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

construct{?a h:OlderThan ?c}
where {
  ?a h:age ?b.
  ?c h:age ?d.
  filter(?b > ?d )}
```

Query:

<ANSWER HERE/>



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

Query:

<ANSWER HERE/>

Question 7

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

Query:

<ANSWER HERE/>

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

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

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

```
select * where {
```

h:John ?p ?y.

?p rdfs:label "size"@en

2

Question 8

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

Query:

<ANSWER HERE/>

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

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

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

```
select * where {
```

```
    h:Laura rdf:type ?x
```

```
}
```

Day 04: Answers to the practical session on OWL.

Software requirements

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

A, Query data augmented by an OWL schema

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

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

Code added to the schema:

<ANSWER HERE/>

`h:hasSpouse` a `owl:SymmetricProperty` .

`h:hasFriend` a `owl:SymmetricProperty` .

Query:

<ANSWER HERE/>

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

insert data {`h:hasSpouse` a `owl:SymmetricProperty` .}

insert data {`h:hasFriend` a `owl:SymmetricProperty` .}

Result before addition to the schema:

<ANSWER HERE/>

No owl properties shown for `hasSpouse` and `hasChild`

Result after addition to the schema:

<ANSWER HERE/>

?y http://www.w3.org/2002/07/owl#SymmetricProperty ?p http://www.w3.org/1999/02/22-rdf-syntax-ns#type ?b http://www.w3.org/2002/07/owl#SymmetricProperty ?a http://www.w3.org/1999/02/22-rdf-syntax-ns#type

Explanation:

<ANSWER HERE/>

Owl properties are added for these two classes

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

Code added to the schema:

<ANSWER HERE/>

{`h:hasChild` `owl:InverseOf` `h:hasParent` .}

Query:

<ANSWER HERE/>

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

select * where {

h:hasChild ?p ?y

}

Result before addition to the schema:

<ANSWER HERE/>

No owl relational properties shown for hasChild and hasParent

Result after addition to the schema:

<ANSWER HERE/>

```
?y
http://www.w3.org/1999/02/22-rdf-syntax-ns#Property
"relation between an animal and another animal to which it gave birth."@en
"relation entre un animal et un autre animal auquel il a donné naissance."@fr
"has for child"@en
"a pour enfant"@fr
http://www.inria.fr/2007/09/11/humans.rdfs#hasParent
```

```
?p
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.w3.org/2000/01/rdf-schema#comment
http://www.w3.org/2000/01/rdf-schema#comment
http://www.w3.org/2000/01/rdf-schema#label
http://www.w3.org/2000/01/rdf-schema#label
http://www.w3.org/2002/07/owl#InverseOf
```

Explanation:

<ANSWER HERE/>

Owl properties are added for these two classes

3. Declare `hasAncestor` as transitive property.

Code added to the schema:

<ANSWER HERE/>

h:hasAncestor a owl:TransitiveProperty

Query:

<ANSWER HERE/>

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

insert data {h:hasAncestor a owl:TransitiveProperty .}

Result before addition to the schema:

<ANSWER HERE/>

No owl properties shown for hasAncestor

Result after addition to the schema:

<ANSWER HERE/>

```
http://www.w3.org/1999/02/22-rdf-syntax-ns#Property
http://www.w3.org/2002/07/owl#TransitiveProperty
"relation between an animal and another animal from which it is descended."@en
"relation entre un animal et un autre animal duquel il descend."@fr
http://www.inria.fr/2007/09/11/humans.rdfs#Animal
"has for ancestor"@en
"a pour ancêtre"@fr
http://www.inria.fr/2007/09/11/humans.rdfs#Animal
```

```
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.w3.org/2000/01/rdf-schema#comment
http://www.w3.org/2000/01/rdf-schema#comment
http://www.w3.org/2000/01/rdf-schema#domain
http://www.w3.org/2000/01/rdf-schema#label
http://www.w3.org/2000/01/rdf-schema#label
http://www.w3.org/2000/01/rdf-schema#range
```

Explanation:

<ANSWER HERE/>

Owl properties are added for this class

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:

<ANSWER HERE/>

{h:Male owl:disjointWith h:Female.}

Query:

<ANSWER HERE/>

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

insert data

{h:Male owl:disjointWith h:Female.}

Result before addition to the schema:

<ANSWER HERE/>

No owl relational properties shown for these two classes

Result after addition to the schema:

<ANSWER HERE/>

http://www.w3.org/2000/01/rdf-schema#Class "an animal that produces gametes (spermatozoa) that can fertilize female gametes (ova)."@en "individu appartenant au sexe qui possède le pouvoir de fécondation."@fr "male"@en "mâle"@fr http://www.inria.fr/2007/09/11/humans.rdfs#Animal http://www.inria.fr/2007/09/11/humans.rdfs#Female	http://www.w3.org/1999/02/22-rdf-syntax-ns#type http://www.w3.org/2000/01/rdf-schema#comment http://www.w3.org/2000/01/rdf-schema#comment http://www.w3.org/2000/01/rdf-schema#label http://www.w3.org/2000/01/rdf-schema#label http://www.w3.org/2000/01/rdf-schema#subClassOf http://www.w3.org/2002/07/owl#disjointWith
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Explanation:

<ANSWER HERE/>

Owl properties are added for this class

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

Code added to the schema:

<ANSWER HERE/>

Query:

h:Professor owl:intersectionOf (<Lecturer> <Researcher>)

<ANSWER HERE/>

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

insert data

{

h:Professor owl:intersectionOf (h:Lecturer h:Researcher)

}

Result before addition to the schema:

<ANSWER HERE/>

No owl relational properties shown for these three classes

Result after addition to the schema:

<ANSWER HERE/>

:b0

:b2

http://www.w3.org/2002/07/owl#intersectionOf

http://www.w3.org/2002/07/owl#intersectionOf

Explanation:

<ANSWER HERE/>

Owl properties linked these three classes

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

Code added to the schema:

<ANSWER HERE/>

`h:Academic owl:unionOf (h:Lecturer h:Researcher)`

Query:

<ANSWER HERE/>

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

insert data

{

`h:Academic owl:unionOf (h:Lecturer h:Researcher)`

}

Result before addition to the schema:

<ANSWER HERE/>

No owl relational properties shown for these classes

Result after addition to the schema:

<ANSWER HERE/>

?p
_:b4

?d
<http://www.w3.org/2002/07/owl#unionOf>

Explanation:

<ANSWER HERE/>

Owl properties linked these three classes

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):

<ANSWER HERE/>

`h:Organization rdfs:subClassOf h:University.`

`h:mainEmployer a rdf:Property; rdfs:domain h:Person; rdfs:range h:Organization.`

`h:Professor a owl:Restriction`

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

<ANSWER HERE/>

`h:Professor h:mainEmployer h:University`

Query:

<ANSWER HERE/>

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

```

prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#>
prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
insert data
{
h:Organization rdfs:subClassOf h:University.
h:mainEmployer a rdf:Property; rdfs:domain h:Person; rdfs:range h:Organization.
h:Professor a owl:Restriction
h:Professor h:mainEmployer h:University
}

```

Result before addition to the schema:

<ANSWER HERE/>

No owl relational properties shown for these classes

Result after addition to the schema:

<ANSWER HERE/>

```

http://www.inria.fr/2007/09/11/humans.rdfs#University
http://www.inria.fr/2007/09/11/humans.rdfs#Person
http://www.w3.org/2002/07/owl#Restriction
_:b0
_:b2

```

```

http://www.inria.fr/2007/09/11/humans.rdfs#mainEmployer
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.w3.org/2002/07/owl#intersectionOf
http://www.w3.org/2002/07/owl#intersectionOf

```

Explanation:

<ANSWER HERE/>

Owl properties linked these three classes

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:

<ANSWER HERE/>

Query:

```

prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#>
prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
insert data
{

```

h:Person rdfs:subClassOf h:restriction .

h:restriction a owl:Restriction .

h:restriction owl:onProperty h:hasBiologicalParent.

h:restriction owl:onClass h:Woman.

h:restriction owl:qualifiedCardinality 1.

}

<ANSWER HERE/>

Result before addition to the schema:

<ANSWER HERE/>

Result after addition to the schema:

<ANSWER HERE/>

?y
http://www.w3.org/2000/01/rdf-schema#Class
"a member of the human species"@en
"un membre de l'espèce humaine."@fr
"homme"@fr
"person"@en
"human being"@en
"human"@en
"personne"@fr
"être humain"@fr
"humain"@fr
http://www.inria.fr/2007/09/11/humans.rdfs#Animal
http://www.inria.fr/2007/09/11/humans.rdfs#restriction

?p
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.w3.org/2000/01/rdf-schema#comment
http://www.w3.org/2000/01/rdf-schema#comment
http://www.w3.org/2000/01/rdf-schema#label
http://www.w3.org/2000/01/rdf-schema#label
http://www.w3.org/2000/01/rdf-schema#label
http://www.w3.org/2000/01/rdf-schema#label
http://www.w3.org/2000/01/rdf-schema#label
http://www.w3.org/2000/01/rdf-schema#label
http://www.w3.org/2000/01/rdf-schema#label
http://www.w3.org/2000/01/rdf-schema#label
http://www.w3.org/2000/01/rdf-schema#subClassOf
http://www.w3.org/2000/01/rdf-schema#subClassOf

Explanation:

<ANSWER HERE/>

B, Make your own OWL models:

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

1. owl:oneOf

<YOUR EXAMPLE HERE/>

<university> owl:oneOf <organisation>

University is one of organizations

2. owl:unionOf

<YOUR EXAMPLE HERE/>

<legalagent> owl:unionof (<person> <group>)

Legal agents concludes both person and group

3. owl:intersectionOf

<YOUR EXAMPLE HERE/>

<Man> owl:interseccion (<person><male>)

Man is the intersection of person and male

4. owl:complementOf

<YOUR EXAMPLE HERE/>

<Man> owl:complementof <Woman>

If we consider the class person, it is composed with man and woman (not considering bisexual here)

5. owl:disjointWith

or owl:AllDisjointClasses
or owl:disjointUnionOf

<YOUR EXAMPLE HERE/>

<Man> owl:disjointWith <Woman>

It means that there is no overlap btwn Man and Woman

6. owl:ObjectProperty

<YOUR EXAMPLE HERE/>

owl:ObjectProperty are relations between resources only e.g. hasParent(#thomas,#stephan)

<owl:ObjectProperty rdf:about="hasSon"> <owl:propertyDisjointWith rdf:resource="hasDaughter"/>
</owl:ObjectProperty>

7. owl:DatatypeProperty

<YOUR EXAMPLE HERE/>

owl:DatatypeProperty have a literal value possibly typed ex:hasAge(#thomas,16^^xsd:int)

8. owl:SymmetricProperty
or owl:AsymmetricProperty

<YOUR EXAMPLE HERE/>

a relation that, as soon as it exists, exists in both directions (e.g. to be maried)

<hasSpouse> a owl:SymmetricProperty

9. owl:inverseOf

<YOUR EXAMPLE HERE/>

two relations that exist simultaneously and inversely (ex. parent_of / child_of)

<winedAgainst> owl:inverseOf <lostAgainst>

10. owl:TransitiveProperty

<YOUR EXAMPLE HERE/>

a property propagated from peers to peers (e.g ancestors)

<sameColorAs> a owl:TransitiveProperty .

11. owl:propertyDisjointWith

<YOUR EXAMPLE HERE/>

relations that cannot exist together on the same subject and the same object

<hasSon> owl:propertyDisjointWith <hasDaughter>

12. owl:ReflexiveProperty
or owl:IrreflexiveProperty

<YOUR EXAMPLE HERE/>

a relation that links all individuals to themselves

<hasRelative> a owl:ReflexiveProperty .

13. owl:propertyChainAxiom

<YOUR EXAMPLE HERE/>

relations which combine as a path/chain imply another relation (e.g. parent + brother = uncle)

<uncle> rdf:type owl:ObjectProperty ; owl:propertyChainAxiom
(<parent> <brother>) .

14. owl:FunctionalProperty

<YOUR EXAMPLE HERE/>

a relation for which a resource can have only one value (e.g. birth date)

<birthDate> a owl:FunctionalProperty .

15. owl:InverseFunctionalProperty

<YOUR EXAMPLE HERE/>

a relation for which identical values imply the same subject (e.g. SSN)

socialSecurityNumber a owl:InverseFunctionalProperty

16. owl:hasKey

<YOUR EXAMPLE HERE/>

two resources with the same key values are the same

<Person> owl:hasKey (<name> <firstname> <birthdate> <birthplace>)

17. owl:allValuesFrom

<YOUR EXAMPLE HERE/>

ex:Human rdfs:subClassOf

[a owl:Restriction ;
owl:onProperty ex:hasParent ;

owl:allValuesFrom ex:Human] .

Values of the onProperty is from human, for this example, it means that human's parents must be human too

18. owl:someValuesFrom

<YOUR EXAMPLE HERE/>

```
<owl:Class rdf:ID="Sportive"> <owl:equivalentClass>
<owl:Restriction>
<owl:onProperty rdf:resource="#hobby" /> <owl:someValuesFrom rdf:resource="#Sport" />
</owl:Restriction>
</owl:equivalentClass>
</owl:Class>
```

It shows restriction on some values. Sportive man means that someone who has sport hobbies but also they may have some other types of hobby, so only some hobbies are sport

19. owl:hasValue

<YOUR EXAMPLE HERE/>

```
<owl:Class rdf:ID="Bike">
<subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#nbWheels" /> <owl:hasValue>2</owl:hasValue>
</owl:Restriction>
</subClassOf>
</owl:Class>
```

It restricts to exact values. Bike has 2 wheels, no other options

20. owl:maxCardinality
or owl:minCardinality

<YOUR EXAMPLE HERE/>

It restricts how many times a property is used for a same subject but with different values. This example means that one person only has one name at most.

```
<owl:Class rdf:ID="Person">
<subClassOf> <owl:Restriction>
<owl:onProperty rdf:resource="#name" />
<owl:maxCardinality>1</owl:maxCardinality> </owl:Restriction>
</subClassOf>
</owl:Class>
```

21. owl:qualifiedCardinality

<YOUR EXAMPLE HERE/>

constraint on the number of time a property may be used with values of a given type

with the same subject: minimum, maximum, nombre exact. This example shows human has one biological male parent.

```
<owl:Class rdf:ID="Human"> <rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#hasBiologicalParent" /> <owl:onClass rdf:resource="#Male" />
<owl:qualifiedCardinality>1</owl:qualifiedCardinality>
</owl:Restriction>
</rdfs:subClassOf> </owl:Class>
```