

Semantic Web handout including: lecture questions and practical sessions

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

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

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

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

First name: Daisuke

Family name: Kuwabara

Email: daisuke.kuwabara@edu.dsti.institute

QUESTIONS FROM THE COURSES

Day 01: questions from the course.

Q1.1 Practice XML replace missing parts

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

```
<html class="client-nojs" lang="en" dir="ltr">
<head>
<meta charset="UTF-8"/>
<title>Fabien Gandon - Wikidata</title>
<script>document.documentElement.className="client-
js";RLCONF={ "wgBreakFrames":!1,"wgSeparatorTransformTable":["",""],"wgDigitTransformTable":["",
""], "wgDefaultDateFormat":"dmy", "wgMonthNames":["","January","February","March","April","May","J
une","July","August","September","October","November","December"], "wgRequestId":"0be086ca-bcf1-
4fb8-8a74-
5f24fd256045", "wgCSPNonce":!1, "wgCanonicalNamespace":"","wgCanonicalSpecialPageName":!1, "wg
NamespaceNumber":0, "wgPageName":"Q23014205", "wgTitle":"Q23014205", "wgCurRevisionId":134674
3353, "wgRevisionId":1346743353, "wgArticleId":25028548, "wgIsArticle":!0, "wgIsRedirect":!1, "wgActio
n":"view", "wgUserName":null, "wgUserGroups":["*"], "wgCategories":[], "wgPageContentLanguage":"en",
"wgPageContentModel":"wikibase-
item", "wgRelevantPageName":"Q23014205", "wgRelevantArticleId":25028548, "wgIsProbablyEditable":!0
, "wgRelevantPageIsProbablyEditable":!0, "wgRestrictionEdit":[], "wgMediaViewerOnClick":!0, "wgMedia
ViewerEnabledByDefault":!0, "wgVisualEditor"
:{"pageLanguageCode":"en","pageLanguageDir":"ltr","pageVariantFallbacks":"en"},"wgMFDisplayWiki
baseDescriptions":{"search":!0,"nearby":!0,"watchlist":!0,"tagline":!1},"wgWMESchemaEditAttemptStep
Oversample":!1,"wgULSCurrentAutonym":"English","wgNoticeProject":"wikidata","wbIsEditView":!0,"
wbEntityId":"Q23014205","wgEditSubmitButtonLabelPublish":!0,"wbUserSpecifiedLanguages":[],"wbCo
pyright":{"version":"wikibase-1","messageHtml":"By clicking \"publish\", you agree to the \u003Ca
href=\"https://foundation.wikimedia.org/wiki/Terms_of_Use\" class=\"extiw\" title=\"wikimedia:Terms of
Use\"\u003Eterms of use\u003C/a\u003E, and you irrevocably agree to release your contribution under the
\u003Ca rel=\"nofollow\" class=\"external text\"
href=\"https://creativecommons.org/publicdomain/zero/1.0/\" \u003ECreative Commons CC0
```



```

<https://www.wikidata.org/wiki/Special:EntityData/Q23014205> <http://schema.org/softwareVersion>
"1.0.0" .
<https://www.wikidata.org/wiki/Special:EntityData/Q23014205> <http://schema.org/version>
"1346743353"^^<http://www.w3.org/2001/XMLSchema#integer> .
<https://www.wikidata.org/wiki/Special:EntityData/Q23014205> <http://schema.org/dateModified>
"2021-01-23T14:23:53Z"^^<http://www.w3.org/2001/XMLSchema#dateTime> .
<https://www.wikidata.org/wiki/Special:EntityData/Q23014205> <http://wikiba.se/ontology#statements>
"32"^^<http://www.w3.org/2001/XMLSchema#integer> .
<https://www.wikidata.org/wiki/Special:EntityData/Q23014205> <http://wikiba.se/ontology#sitelinks>
"0"^^<http://www.w3.org/2001/XMLSchema#integer> .
<https://www.wikidata.org/wiki/Special:EntityData/Q23014205> <http://wikiba.se/ontology#identifiers>
"12"^^<http://www.w3.org/2001/XMLSchema#integer> .
<http://www.wikidata.org/entity/Q23014205> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://wikiba.se/ontology#Item> .

```

Q1.3 DBpedia

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

Dbp:populationDemonym : Londoner

Vienna URI:

<https://dbpedia.org/resource/Vienna>

Native name:Wien

<ANSWER HERE/>

Q1.4 WHO.IS?

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

<ANSWER HERE/>

Florian DEFOUR

Not foundable

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

<ANSWER HERE/>

```
<!DOCTYPE html>
<html
  prefix= "
    dbp: http://dbpedia.org/property/
    dbo: http://dbpedia.org/ontology/
    dct: http://purl.org/dc/terms/
  "
  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: foaf= "http://xmlns.com/foaf/0.1/"
  xmlns: skos= "http://www.w3.org/2004/02/skos/core#"
  xmlns: dbp= "http://dbpedia.org/property/"
  xmlns: geo= "http://www.w3.org/2003/01/geo/wgs84_pos#"
  xmlns: dbo= "http://dbpedia.org/ontology/"
```

2.

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

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

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

3.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
  <title>406 Not Acceptable</title>
</head><body>
  <h1>406 Not Acceptable</h1>
  <p>An appropriate representation of the requested resource Vienna could not be found on this server.</p>
  Available variant(s):
  <ul>
    <li><a href= "Vienna">Vienna</a> , type text/html, charset UTF-8</li>
  </ul>
```

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

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:skos="http://www.w3.org/2004/02/skos/core#"
  xmlns:dbp="http://dbpedia.org/property/"
  xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
  xmlns:dbpedia="http://dbpedia.org/ontology/"
```

4.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0/EN"
  "http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd">
<html version="XHTML+RDFa 1.0" xmlns="http://www.w3.org/1999/xhtml" xmlns:madsrdf="http://www.loc.gov/mads/rdf/v1#"
  xmlns:ri="http://id.loc.gov/ontologies/RecordInfo#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:skos="http://www.w3.org/2004/02/skos/core#" xmlns:skosxl="http://www.w3.org/2008/05/skos-xl#" xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:cs="http://www.w3.org/2003/06/sw-vocab-status/ns#" xmlns:dcterms="http://purl.org/dc/terms/">
  <head>
    <title>Hugo, Victor, 1802-1885 - LC Linked Data Service: Authorities and Vocabularies | Library of Congress</title>
    <meta name="description" content="The Linked Data Service provides access to commonly found standards and vocabularies promulgated by the Library of Congress. This includes data values and the controlled vocabularies that house them. Datasets available include LCSH, BIBFRAME, LC Name Authorities, LC Classification, MARC codes, PREMIS vocabularies, ISO language codes, and more."/>
    <link rel="schema.DC" href="http://purl.org/dc/elements/1.1/">
    <link rel="dc:relation.isPartOf" href="//www.loc.gov/" title="Library of Congress">
    <meta name="dc:title" content="LC Linked Data Service: Authorities and Vocabularies (Library of Congress)">
    <meta name="dc.contributor" content="The Library of Congress">
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0/EN"
  "http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd">
<html version="XHTML+RDFa 1.0" xmlns="http://www.w3.org/1999/xhtml" xmlns:madsrdf="http://www.loc.gov/mads/rdf/v1#"
  xmlns:ri="http://id.loc.gov/ontologies/RecordInfo#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:skos="http://www.w3.org/2004/02/skos/core#" xmlns:skosxl="http://www.w3.org/2008/05/skos-xl#" xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:cs="http://www.w3.org/2003/06/sw-vocab-status/ns#" xmlns:dcterms="http://purl.org/dc/terms/">
  <head>
    <title>Hugo, Victor, 1802-1885 - LC Linked Data Service: Authorities and Vocabularies | Library of Congress</title>
    <meta name="description" content="The Linked Data Service provides access to commonly found standards and vocabularies promulgated by the Library of Congress. This includes data values and the controlled vocabularies that house them. Datasets available include LCSH, BIBFRAME, LC Name Authorities, LC Classification, MARC codes, PREMIS vocabularies, ISO language codes, and more."/>
    <link rel="schema.DC" href="http://purl.org/dc/elements/1.1/">
    <link rel="dc:relation.isPartOf" href="//www.loc.gov/" title="Library of Congress">
    <meta name="dc:title" content="LC Linked Data Service: Authorities and Vocabularies (Library of Congress)">
    <meta name="dc.contributor" content="The Library of Congress">
```

5.

```
<!DOCTYPE html SYSTEM "about:legacy-compat">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en"><head><title>MCAM - Cell surface glycoprotein MUC18 precursor - Homo sapiens (Human) - MCAM gene & protein</title><meta content="IE=edge" http-equiv="X-UA-Compatible"/><meta content="text/html; charset=UTF-8" http-equiv="Content-Type"/><meta content="width=device-width, initial-scale=1" name="viewport"/><link href="/" rel="home"/><link href="https://creativecommons.org/licenses/by/4.0/" rel="license"/><link type="image/vnd.microsoft.icon" href="/favicon.ico" rel="shortcut icon"/><link href="/uniprot.min.css2021_02" type="text/css" rel="stylesheet"/><link href="/tippy.css" type="text/css" rel="stylesheet"/><script type="text/javascript">
  var BASE = '/';
</script><script src="/js-compr.js2021_02" type="text/javascript"></script><script type="text/javascript">
  uniprot.isInternal = false;
  uniprot.namespace = 'uniprot';
  uniprot.releasedate = '2021_02';
</script><script type="text/javascript">
  ;
</script><link href="opensearch.xml" title="UniProtKB" type="application/opensearchdescription+xml" rel="search"/><link href="https://www.uniprot.org/uniprot/P43121" rel="canonical"/><link href="P43121.rdf" title="RDF" type="application/rdf+xml" rel="alternate"/><link href="P43121.rss?version=*" type="application/rss+xml" title="RSS" rel="alternate"/><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:ECO="http://purl.obolibrary.org/obo/ECO_"
  xmlns:annotation="http://purl.uniprot.org/annotation/" xmlns:citation="http://purl.uniprot.org/citations/" xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:disease="http://purl.uniprot.org/diseases/" xmlns:enzyme="http://purl.uniprot.org/enzyme/" xmlns:faldo="http://biohackathon.org/resource/faldo#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/" xmlns:go="http://purl.obolibrary.org/obo/GO_" xmlns:isoform="http://purl.uniprot.org/isoforms/"
  xmlns:keyword="http://purl.uniprot.org/keywords/" xmlns:location="http://purl.uniprot.org/locations/" xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:position="http://purl.uniprot.org/position/" xmlns:pubmed="http://purl.uniprot.org/pubmed/" xmlns:range="http://purl.uniprot.org/range/"
```

```

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:taxon="http://purl.uniprot.org/taxonomy/" xmlns:tissue="http://purl.uniprot.org/tissues/">
<owl:Ontology rdf:about="http://purl.uniprot.org/uniprot/">
<owl:imports rdf:resource="http://purl.uniprot.org/core/">
</owl:Ontology>
<rdf:Description rdf:about="http://purl.uniprot.org/uniprot/P43121">
<rdf:type rdf:resource="http://purl.uniprot.org/core/Protein"/>
<reviewed rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean">true</reviewed>
<created rdf:datatype="http://www.w3.org/2001/XMLSchema#date">1995-11-01</created>
<modified rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2021-02-10</modified>

```

6.
Xavier Dolan
json

7.
Distributed Artificial Intelligence for Distributed Corporate Knowledge Management
Turtle/n3

Q1.6 Recall five best practices of linked open data



<ANSWER HERE/>

On the web with open license
Machine readable data
Non proprietary format
RDF standards
Linked RDF

Q1.7 Spotlight demo

Reproduce the demo:

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

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

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

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

Black Holes and Revelations (2006) incorporated [electronic](#) and [pop](#) elements, displayed in singles such as "Supermassive Black Hole",^[1] and brought Muse wider international success. *The Resistance* (2009) and *The 2nd Law* (2012) explored themes of government oppression and civil uprising and cemented Muse as one of the world's major stadium acts. *Rolling Stone* stated the band possessed "stadium-crushing songs".^[2] Topping the US *Billboard 200*, their seventh album, *Drones* (2015), was a [concept album](#) about [drone warfare](#) and returned to a harder rock sound. Their eighth album, *Simulation Theory* (2018), prominently featured [synthesisers](#) and was influenced by [science fiction](#) and the [simulation hypothesis](#).

Muse have won [numerous awards](#), including two [Grammy Awards](#), two [Brit Awards](#), five [MTV Europe Music Awards](#) and eight [NME Awards](#). In 2012 they received the [Ivor Novello Award](#) for International Achievement from the [British Academy of Songwriters, Composers and Authors](#). As of June 2016, they have sold over 20 million albums worldwide.^[3]

[https://dbpedia.org/page/Muse_\(band\)](https://dbpedia.org/page/Muse_(band))

Muse are an English rock band from Teignmouth, Devon, formed in 1994. The band consists of Matt Bellamy (lead vocals, guitar, keyboards), Chris Wolstenholme (bass guitar, backing vocals), and Dominic Howard (drums). Muse have won numerous awards, including two Grammy Awards, two Brit Awards, five MTV Europe Music Awards and eight NME Awards. In 2012 they received the Ivor Novello Award for International Achievement from the British Academy of Songwriters, Composers and Authors. As of June 2016, they have sold over 20 million albums worldwide.

ミューズ (英: **Muse**) は、[イギリスのロックバンド](#)。1994年にデヴォン州ティンマス (英語版) で結成。グラミー賞のノミネート3回 (うち受賞2回)、全世界でのセールスは1500万枚以上を数える^[3]。

- ミューズ (英: **Muse**) は、イギリスのロックバンド。1994年にデヴォン州で結成。グラミー賞のノミネート3回 (うち受賞2回)、全世界でのセールスは1500万枚以上を数える。(ja)

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/>

RDF is a model for directed labeled multigraph

Directed: edges have a direction

Labeled: edges and nodes are labeled

Multigraph: there can be several edges/arcs between nodes/vertices

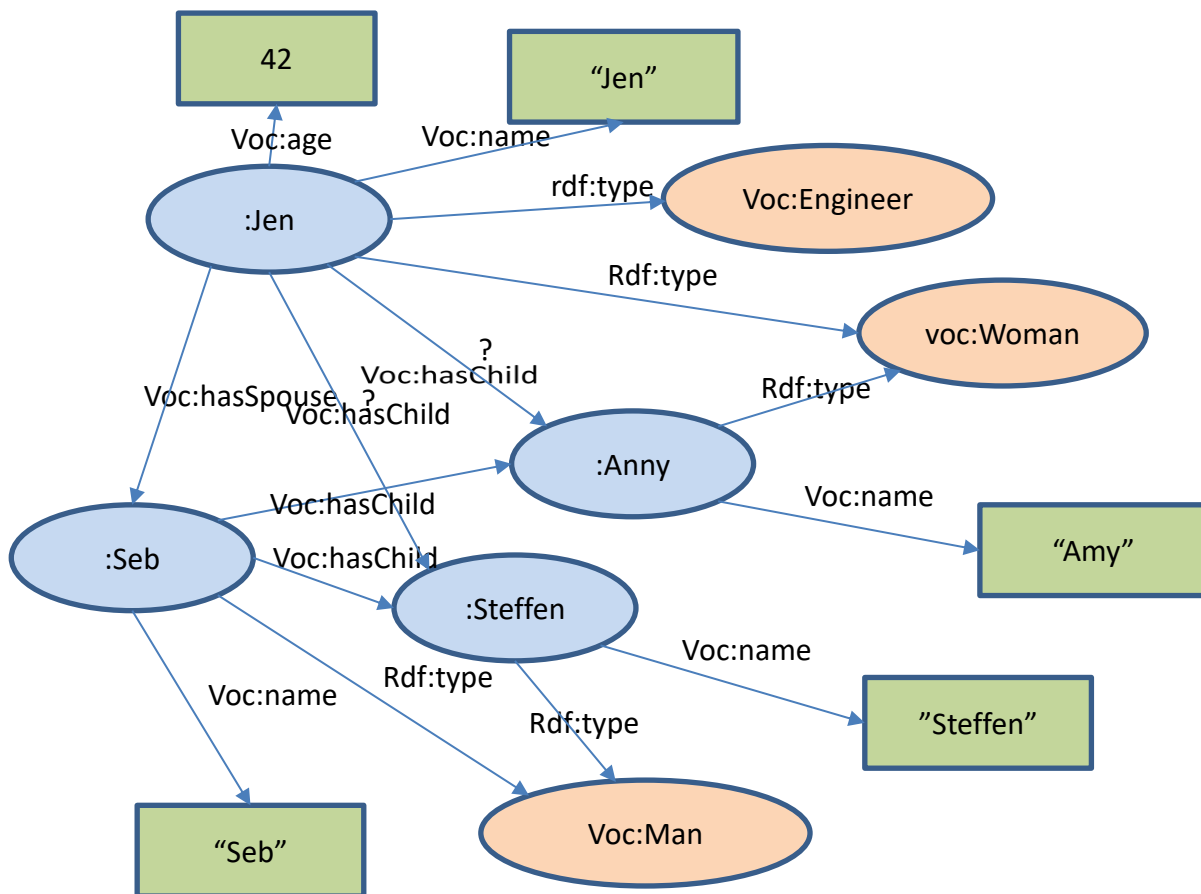
Q2.1 Fill the blanks

"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`, `vac:name`, `vac:hasChild`, `vac:age`, `vac:hasSpouse`, `rdf:type`, `vac:Engineer`, `vac: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">
  <vac:Woman rdf:about="#Jen">
    <vac:name>Jen</vac:name>
    <vac:age
rdf:datatype="http://www.w3.org/2001/XMLSchema#integer"><42> </vac:age>
    <vac rdf:resource="#Seb"></vac: rdf>
    <vac:hasChild rdf:resource="#Steffen"></vac:hasChild>
    <vac:hasChild>
      <rdf:Description rdf:about="#Anny">
        <vac:name>Anny</vac:name>
        <rdf:type rdf:resource="&vocab;#Woman"></rdf:type>
      </rdf:Description>
    </vac:hasChild>
    <rdftype rdf:resource="&vocab;#Engineer"></rdftype>
  </vac:Woman>
  <vac: man rdf:about="#Seb">
    <vac:name>Seb</vac:name>
    <vac:hasChild rdf:resource="#Steffen"></vac:hasChild>
    <vac:hasChild rdf:resource="#Anny"></vac:hasChild>
```

```

</voc: man>
<voc:Man rdf:about="#Steffen">
  <voc:name>Steffen</voc:name>
</voc:Man>
</rdf:RDF>

```

Q2.3 Fill the blanks (N3/Turtle)

```

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

```

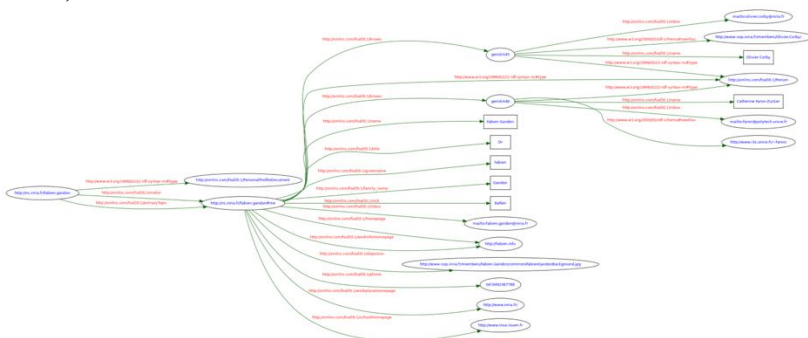
Q2.4 Visit me please

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

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

<ANSWER HERE/>

2. RDF, XML



3. _____

4.

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

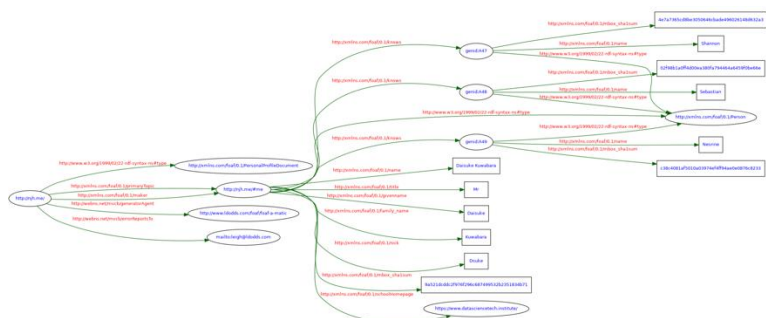
<http://ns.inria.fr/fabien.gandon>
 a foaf:PersonalProfileDocument ;
 foaf:maker <http://ns.inria.fr/fabien.gandon#me> ;
 foaf:primaryTopic <http://ns.inria.fr/fabien.gandon#me> .

<http://ns.inria.fr/fabien.gandon#me>
 a foaf:Person ;
 foaf:name "Fabien Gandon" ;
 foaf:title "Dr" ;
 foaf:givenname "Fabien" ;
 foaf:family_name "Gandon" ;
 foaf:nick "Bafien" ;
 foaf:mbox <mailto:fabien.gandon@inria.fr> ;
 foaf:homepage <http://fabien.info> ;
 foaf:depiction <http://www-sop.inria.fr/members/Fabien.Gandon/common/FabienGandonBackground.jpg> ;
 foaf:phone <tel:0492387788> ;
 foaf:workplaceHomepage <http://www.inria.fr/> ;
 foaf:workInfoHomepage <http://fabien.info> ;
 foaf:schoolHomepage <http://www.insa-rouen.fr> ;
 foaf:knows [
 a foaf:Person ;
 foaf:name "Olivier Corby" ;
 foaf:mbox <mailto:olivier.corby@inria.fr> ;
 rdfs:seeAlso <http://www-sop.inria.fr/members/Olivier.Corby/>
], [
 a foaf:Person ;
 foaf:name "Catherine Faron-Zucker" ;
 foaf:mbox <mailto:faron@polytech.unice.fr> ;
 rdfs:seeAlso <http://www.i3s.unice.fr/~faron/>
] .

5.



6.



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/>

Report with a Music and Danse theme with 73 pages

Q2.6 Visit to Victor Hugo

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

<ANSWER HERE/>

RDF

Name translation seems somewhat wrong

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

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

<ANSWER HERE/>

<http://inria.fr/rr/doc.html>
has the subject
<http://data.bnf.fr/ark:/12148/cb121105993>
which is in the graph of
<http://inria.fr/data/algebra>

Q2.8 Visit Leukocyte surface antigen CD53

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

5. Any remark about the structure of the data?

<ANSWER HERE/>

RDF, XML

Owl ontology

Reification

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 . }
```

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
```

[:Paris](#) [rdf:type](#) [owl:Thing](#)

[:Paris](#) [rdf:type](#) [dbpedia:ontology/Place](#)

[:Paris](#) [rdf:type](#) [dbpedia:ontology/Location](#)

[:Paris](#) [rdf:type](#) [http://www.w3.org/2003/01/geo/wgs84_pos#SpatialThing](#)

[:Paris](#) [rdf:type](#) [dbpedia:class/yago/Object100002684](#)

[:Paris](#) [rdf:type](#) [dbpedia:class/yago/PhysicalEntity100001930](#)

[:Paris](#)  [rdf:type](#)  [dbpedia:class/yago/Prefecture108626947](#) 

[:Paris](#)  [rdf:type](#)  [dbpedia:class/yago/Region108630985](#) 

[:Paris](#)  [rdf:type](#)  [dbpedia:class/yago/Seat108647945](#) 

[:Paris](#)  [rdf:type](#)  [dbpedia:class/yago/Site108651247](#) 

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") }
```

wd:Q23	George Washington
wd:Q76	Barack Obama
wd:Q91	Abraham Lincoln
wd:Q207	George W. Bush
wd:Q6279	Joe Biden
wd:Q1124	Bill Clinton
wd:Q8007	Franklin Delano Roosevelt
wd:Q9640	Lyndon B. Johnson
wd:Q9588	Richard Nixon
wd:Q8612	Andrew Johnson
wd:Q9696	John F. Kennedy
wd:Q9916	Dwight D. Eisenhower
wd:Q11613	Harry S. Truman
wd:Q9582	Gerald Ford
wd:Q9960	Ronald Reagan
wd:Q11812	Thomas Jefferson
wd:Q11815	James Monroe
wd:Q11813	James Madison
wd:Q11869	William Henry Harrison

wd:Q11816	John Quincy Adams
wd:Q11806	John Adams
wd:Q12306	Millard Fillmore
wd:Q11881	John Tyler
wd:Q12312	Franklin Pierce
wd:Q11891	James K. Polk
wd:Q22686	Donald Trump
wd:Q12325	James Buchanan
wd:Q11896	Zachary Taylor
wd:Q23505	George H. W. Bush
wd:Q11817	Andrew Jackson
wd:Q23685	Jimmy Carter
wd:Q34597	James A. Garfield
wd:Q34836	Ulysses S. Grant
wd:Q35041	William McKinley
wd:Q35171	Grover Cleveland
wd:Q11820	Martin Van Buren
wd:Q35286	Warren G. Harding
wd:Q35236	Herbert Hoover
wd:Q33866	Theodore Roosevelt
wd:Q35498	Chester A. Arthur
wd:Q35678	Benjamin Harrison
wd:Q35648	William Howard Taft
wd:Q34296	Woodrow Wilson
wd:Q36023	Calvin Coolidge
wd:Q35686	Rutherford B. Hayes

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/>

<https://www.wikidata.org/wiki/Q30>: United States of America

<https://www.wikidata.org/wiki/Property:P6>: Head of government

The q-name of “given name” is P735

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

<ANSWER HERE/>

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

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

<ANSWER HERE/>

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

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

1. Find the age of resources whose name is ‘Fabien’
2. Find the name of resources whose age is less than 50
3. Find property values of resources whose name is ‘Fabien’ and whose age is less than 50
4. Find other names of resources whose name is ‘Fabien’
5. Find resources which have two different properties with the same value
6. Find resources which have the same property with two different values

<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 ?p ?v WHERE{
?x c:name "Fabien";
c:age ?age;
?p ?v .
FILTER(?age<50)}

4.
SELECT ?name WHERE{
?x c:name "Fabien" ,
?name .
FILTER(?name != "Fabien")}

5.
SELECT * WHERE{
?x ?p ?v ;
?q ?v .
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, as there is no member of variables between a and b

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 c:memberOf ?org .  
filter(! exists{?x c:author ?doc})}
```

Q3.9 what is retrieving this query ?

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

<ANSWER HERE/>

Author and the number of documents they publish in the descending order with respect to the publishing number

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

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

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

<ANSWER HERE/>

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 }
}
```

People and their friends name unless name would be “John Doe” if not known

<ANSWER HERE/>

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/>

Facebook query:

People and the average age of friends they know.

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/>

1.
SELECT * WHERE{
{?x c:study c:Informatics}UNION{?x c:study c:Mathematics}}

2.
SELECT * WHERE{
{?x c:study c:Informatics}UNION{?x c:study c:Mathematics}
OPTIONAL{?x c:name ?name}}

3. SELECT * WHERE{
{?x c:study c:Informatics}UNION{?x c:study c:Mathematics}
OPTIONAL{GRAPH ?g {?x c:name ?name}}}

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

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

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

<ANSWER HERE/>

G1 as it is not in the graph{ } and thus using from

G2 as it is in the graph and using from named

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/>

Delete the string of age property, insert the integer of it.

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{ }where{ }

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

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

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

<ANSWER HERE/>

Insert will push the data to the dataset while the construct just give me the result(data)

Day 04: questions from the course on RDFS.

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

- ☐ **X** 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/>

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

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

<ANSWER HERE/>

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/>

Phone and computer is the subclass of device

Smartphone is subclass of computer and phone → smartphone is subclass of device

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/>

EmployeeOf and isShareholderOf is subproperty of proRelationWith
Isshareholder and isCEOof is the subploerty of hasControlOver
proRelationWith is subPropertyOf isShareholderOf and isCEOof

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

<ANSWER HERE/>

Attributes of piloteOf are type Object and Plane

Attributes of driverOf are type Object and Car

Subject of piloteOf are type Human and Adult

Subject of driverOf are type Human

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

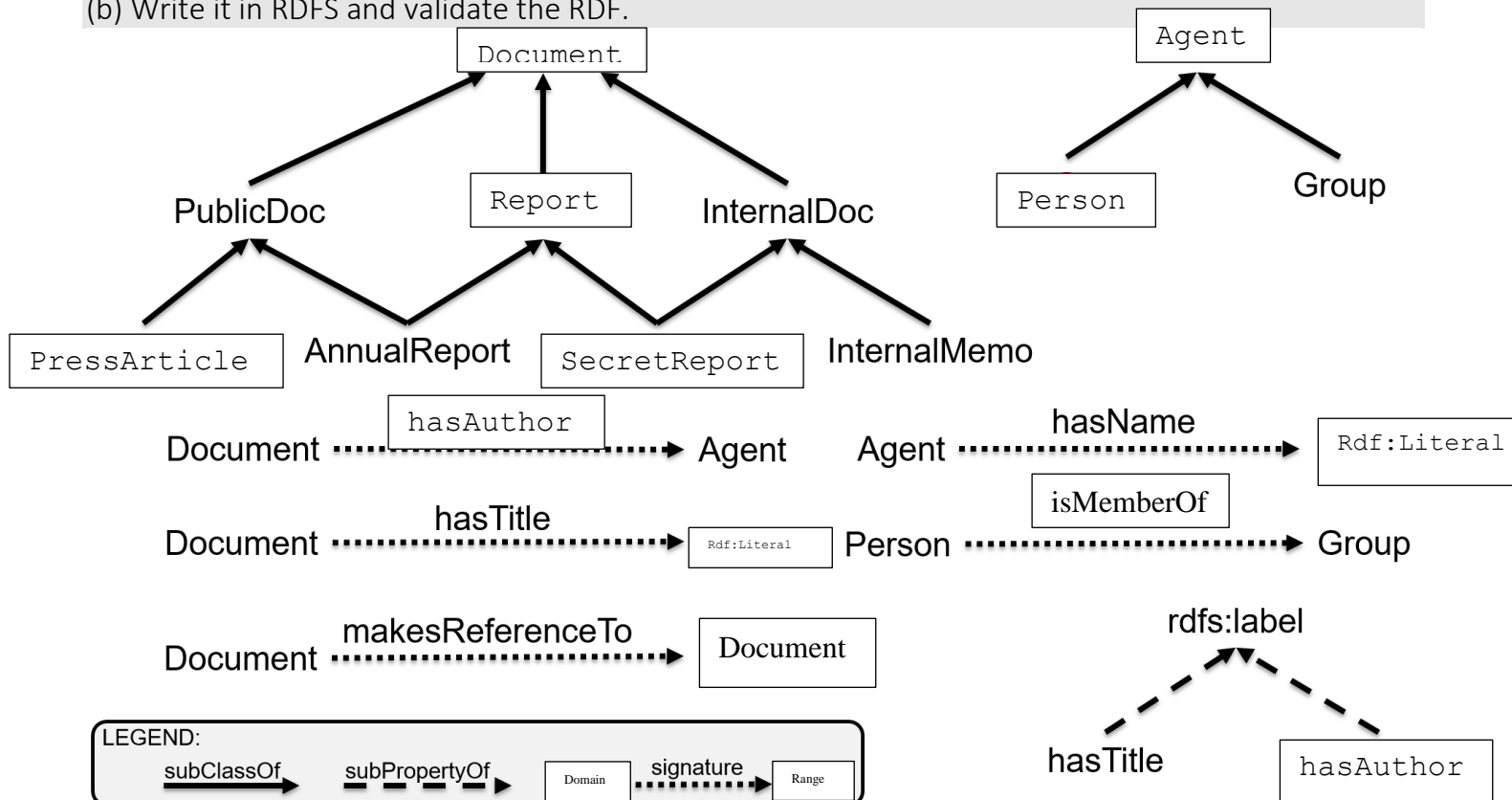
```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
@base <http://inria.fr/2005/humans.rdfs>
<p1> a rdf:Property ; rdfs:label "age"@fr .
<c1> a rdfs:Class; rdfs:comment "un être humain"@fr .
```

- ☐ <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/>

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

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



Day 04: questions from the course on OWL.

Q5.1 What can we deduce?

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

<ANSWER HERE/>

Jane is Human

John is Female and Human

Q5.2 What are we defining and inferring?

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

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

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

<ANSWER HERE/>

Jim is Man and Parent ← The information GrandFather cannot be known with this information
Jack is Man and Parent

Q5.3 What can we deduce?

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

<ANSWER HERE/>

Jane hasParent Jim. Jane hasAncestor Jim and James
John hasSpouse Jane
Jim has Ancestor James whose child is Jim

Q5.4 What can we deduce?

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

<ANSWER HERE/>

JimmyPage are is different from JimmyHendrix which is from JamesPatrickPage

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 is person who are intersected with the complement of Lucky.

Q5.6 What can we deduce?

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

<ANSWER HERE/>

James and Jane are 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/>

a is Person and b and c are 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/>

James and Jhon URL are identified as the same person
As human can have only 1 biological Father and Mother

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/>

There are two labels preferred.

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

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

2. <http://www.w3.org/TR/skos-reference/skos.rdf>

3.

Inverse of `broadMatch`

Sub property of `mappingRelation`

Sub property of `narrower`

Q6.3 practice:

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

<ANSWER HERE/>

MediaType

2.

PREFIX cd

PRFIX cc:

PREFIX xhtml:license <http://creativecommons.org/ns#>

[https://www.amazon.com/Deep-Learning-Adaptive-Computation-](https://www.amazon.com/Deep-Learning-Adaptive-Computation-Machine/dp/0262035618/ref=sr_1_1?dchild=1&keywords=deep+learning&qid=1617951041&sr=8-1)

[Machine/dp/0262035618/ref=sr_1_1?dchild=1&keywords=deep+learning&qid=1617951041&sr=8-1](https://www.amazon.com/Deep-Learning-Adaptive-Computation-Machine/dp/0262035618/ref=sr_1_1?dchild=1&keywords=deep+learning&qid=1617951041&sr=8-1)

dc:title Deep-Learning-Adaptive-Computation-Machine

dc: date November 18 2016

dc:format "text/html"

dc:Language English

dc:type

<xhtml:license rdf="http://creativecommons.org/licenses/by-nc-nd/4.0/">

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?

1. Type: ObjectProperty, InverseFunctionalProperty
4. Inherits from page
5. The signature of weblog:Domain:Agent Range:Document,
The signature of Domain:Thing 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/>

RDF/XML

<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/"
  xmlns:admin="http://webns.net/mvcb/">
<foaf:PersonalProfileDocument rdf:about="">
  <foaf:maker rdf:resource="#me"/>
  <foaf:primaryTopic rdf:resource="#me"/>
  <admin:generatorAgent rdf:resource="http://www.ldodds.com/foaf/foaf-a-matic"/>
  <admin:errorReportsTo rdf:resource="mailto:leigh@ldodds.com"/>
</foaf:PersonalProfileDocument>
<foaf:Person rdf:ID="me">
  <foaf:name>Daisuke Kuwabara</foaf:name>
  <foaf:title>Mr</foaf:title>
  <foaf:givenname>Daisuke</foaf:givenname>
  <foaf:family_name>Kuwabara</foaf:family_name>
  <foaf:nick>Dsuke</foaf:nick>
  <foaf:mbox_sha1sum>9a521cdcdc2f976f296c687499532b2351834b71</foaf:mbox_sha1sum>
  <foaf:schoolHomepage rdf:resource="https://www.datasciencetech.institute/">
  <foaf:knows>
  <foaf:Person>
  <foaf:name>Shannon</foaf:name>
  <foaf:mbox_sha1sum>4e7a7365cd8be3050646cbade496026148d632a3</foaf:mbox_sha1sum></foaf:Person>
  <foaf:knows>
  <foaf:Person>
  <foaf:name>Sebastian</foaf:name>
```



```

<foaf:mbox_sha1sum>02f98b1a0ff4d00ea380fa794464a6459f0be66e</foaf:mbox_sha1sum></foaf:Person>
></foaf:knows>
<foaf:knows>
<foaf:Person>
<foaf:name>Nesrine</foaf:name>
<foaf:mbox_sha1sum>c38c4081af5010a03974ef4ff94ae0e0876c8233</foaf:mbox_sha1sum></foaf:Person>
></foaf:knows></foaf:Person>
</rdf:RDF>

```

Turtle

```

@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix ns0: <http://webns.net/mvcb/> .

<http://njh.me/>
  a foaf:PersonalProfileDocument ;
  foaf:maker <http://njh.me/#me> ;
  foaf:primaryTopic <http://njh.me/#me> ;
  ns0:generatorAgent <http://www.ldodds.com/foaf/foaf-a-matic> ;
  ns0:errorReportsTo <mailto:leigh@ldodds.com> .

<http://njh.me/#me>
  a foaf:Person ;
  foaf:name "Daisuke Kuwabara" ;
  foaf:title "Mr" ;
  foaf:givenname "Daisuke" ;
  foaf:family_name "Kuwabara" ;
  foaf:nick "Dsuke" ;
  foaf:mbox_sha1sum "9a521dcddc2f976f296c687499532b2351834b71" ;
  foaf:schoolHomepage <https://www.datasciencetech.institute/> ;
  foaf:knows [
    a foaf:Person ;
    foaf:name "Shannon" ;
    foaf:mbox_sha1sum "4e7a7365cd8be3050646cbade496026148d632a3"
  ], [
    a foaf:Person ;
    foaf:name "Sebastian" ;
    foaf:mbox_sha1sum "02f98b1a0ff4d00ea380fa794464a6459f0be66e"
  ], [
    a foaf:Person ;
    foaf:name "Nesrine" ;
    foaf:mbox_sha1sum "c38c4081af5010a03974ef4ff94ae0e0876c8233"
  ] .

```

- +
- Ref: enemy of “people with authority”
- Ref: Lost Contact With Nika
- Ref: Influenced by “Michal Fabinger”
- Ref: Spouse of “My future wife”

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/>

BioRDF2DBL is linked to BioRDF DBLP using exactMatch predicate
Almost matched with 9000

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/>

Find the datasets which has the sparql endpoints ready to be queried

```
PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:   <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
PREFIX foaf:   <http://xmlns.com/foaf/0.1/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX scovo:  <http://purl.org/NET/scovo#>
PREFIX void:   <http://rdfs.org/ns/void#>
PREFIX akt:    <http://www.aktors.org/ontology/portal#>
select * where{ ?ds a void:Dataset; rdfs:label 'DBpedia-fr'; ?p ?v. }
```

Q6.8 What does this mean?

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

plot generating the bar-char is used for stats 1998 which is the distribution of the csv and

Q6.9 What does this mean?

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

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

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

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

<ANSWER HERE/>

Dump which contains around 13000 triples was generated by the R2RTransform12 in 2015 who derives from df-employ

Q6.10 practice:

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

<ANSWER HERE/>

Mo:MusicArtist

Revision: 2.1.5

Reused: af and theatre

defines 54 classes and 153 properties.

RDFS OWL

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

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

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

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

<ANSWER HERE/>

“<http://.ietf.org/rfc/rfc2717.txt> dc:title “Hypertext Transfer Protocol -- HTTP/1.1”

“<http://.ietf.org/rfc/rfc2717.txt> dc:title “Uniform Resource Identifiers (URI): Generic Syntax”

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

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

<ANSWER HERE/>

Cathy is the foaf Person, foaf name is “Catherine Faron” and her mail box is faron@i3s.unice.fr and she 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/>

```
<div xmlns="http://www.w3.org/1999/xhtml"
  prefix="
    rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
    dc: http://purl.org/dc/terms/
    rdfs: http://www.w3.org/2000/01/rdf-schema#"
  >
  <div typeof="rdfs:Resource" about="http://rdf-translator.appspot.com/">
    <div property="dc:creator" xml:lang="en" content="Paul"></div>
  </div>
</div>
```

Scenario-2 has for creator “Paul”(English language)

4. An ontology
5. Spinoza, Einstein, Schopenhauer among other

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

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

<ANSWER HERE/>

```
<span vocab="http://schema.org/" typeof="TechArticle">
  <a property="url" href="http://www.w3.org/TR/rdfa-primer/">
    <span property="name">RDFa 1.1 Primer</span></a>.
  <span property="creator"> "Daisuke"</span>
</span>
```

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

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

<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/tt1375666/?ref_=tt_mv_close> og:description "Directed by
Christopher Nolan. With Leonardo DiCaprio, Joseph Gordon-Levitt, Elliot Page, Ken
Watanabe. A thief who steals corporate secrets through the use of dream-sharing
technology is given the inverse task of planting an idea into the mind of a C.E.O." ;
  og:image "https://m.media-
amazon.com/images/M/MV5BMjAxMzY3Njc5NF5BMl5BanBnXkFtZTcwNTI5OTM0Mw@@._V1_UY1200_CR90,0,
630,1200_AL_.jpg" ;
  og:site_name "IMDb" ;
  og:title "Inception (2010) - IMDb" ;
  og:type "video.movie" ;
  og:url "http://www.imdb.com/title/tt1375666/" ;
  ns2:fbmlapp_id "115109575169727" .
```

?

Q7.6 Test JSON-LD online

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

From

<ANSWER HERE/>

```
[
  {
    "@id": "",
    "@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": "#me"
      }
    ],
    "http://xmlns.com/foaf/0.1/primaryTopic": [
      {
        "@id": "#me"
      }
    ]
  },
  {
    "@id": "_:N72ec5ecbbaa84046995a5ca355861099",
    "@type": [
      "http://xmlns.com/foaf/0.1/Person"
    ],
    "http://xmlns.com/foaf/0.1/mbox_sha1sum": [
      {
        "@value": "c38c4081af5010a03974ef4ff94ae0e0876c8233"
      }
    ]
  }
]
```



```
    }
  ],
  "http://xmlns.com/foaf/0.1/name": [
    {
      "@value": "Nesrine"
    }
  ]
},
{
  "@id": "#me",
  "@type": [
    "http://xmlns.com/foaf/0.1/Person"
  ],
  "http://xmlns.com/foaf/0.1/family_name": [
    {
      "@value": "Kuwabara"
    }
  ],
  "http://xmlns.com/foaf/0.1/givenname": [
    {
      "@value": "Daisuke"
    }
  ],
  "http://xmlns.com/foaf/0.1/knows": [
    {
      "@id": "_:N4596cc36cc924c2d84309e2bc2fd6758"
    },
    {
      "@id": "_:N5b3e6677b4644dad8ce6f98af3b42ec7"
    },
    {
      "@id": "_:N72ec5ecbbaa84046995a5ca355861099"
    }
  ],
  "http://xmlns.com/foaf/0.1/mbox_sha1sum": [
    {
```

```
      "@value": "9a521dcddc2f976f296c687499532b2351834b71"
    }
  ],
  "http://xmlns.com/foaf/0.1/name": [
    {
      "@value": "Daisuke Kuwabara"
    }
  ],
  "http://xmlns.com/foaf/0.1/nick": [
    {
      "@value": "Dsuke"
    }
  ],
  "http://xmlns.com/foaf/0.1/schoolHomepage": [
    {
      "@id": "https://www.datasciencetech.institute/"
    }
  ],
  "http://xmlns.com/foaf/0.1/title": [
    {
      "@value": "Mr"
    }
  ]
],
{
  "@id": "_:N5b3e6677b4644dad8ce6f98af3b42ec7",
  "@type": [
    "http://xmlns.com/foaf/0.1/Person"
  ],
  "http://xmlns.com/foaf/0.1/mbox_sha1sum": [
    {
      "@value": "02f98b1a0ff4d00ea380fa794464a6459f0be66e"
    }
  ],
  "http://xmlns.com/foaf/0.1/name": [
    {
```

```

    "@value": "Sebastian"
  }
]
},
{
  "@id": "_:N4596cc36cc924c2d84309e2bc2fd6758",
  "@type": [
    "http://xmlns.com/foaf/0.1/Person"
  ],
  "http://xmlns.com/foaf/0.1/mbox_sha1sum": [
    {
      "@value": "4e7a7365cd8be3050646cbade496026148d632a3"
    }
  ],
  "http://xmlns.com/foaf/0.1/name": [
    {
      "@value": "Shannon"
    }
  ]
}
]

```

N-Quads

```

<https://json-ld.org/playground/#me> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://xmlns.com/foaf/0.1/Person> .
<https://json-ld.org/playground/#me> <http://xmlns.com/foaf/0.1/family_name> "Kuwabara" .
<https://json-ld.org/playground/#me> <http://xmlns.com/foaf/0.1/givenname> "Daisuke" .
<https://json-ld.org/playground/#me> <http://xmlns.com/foaf/0.1/knowns> _:b0 .
<https://json-ld.org/playground/#me> <http://xmlns.com/foaf/0.1/knowns> _:b1 .
<https://json-ld.org/playground/#me> <http://xmlns.com/foaf/0.1/knowns> _:b2 .
<https://json-ld.org/playground/#me> <http://xmlns.com/foaf/0.1/mbox_sha1sum>
"9a521dcd2f976f296c687499532b2351834b71" .
<https://json-ld.org/playground/#me> <http://xmlns.com/foaf/0.1/name> "Daisuke Kuwabara" .
<https://json-ld.org/playground/#me> <http://xmlns.com/foaf/0.1/nick> "Dsuke" .
<https://json-ld.org/playground/#me> <http://xmlns.com/foaf/0.1/schoolHomepage>
<https://www.datasciencetech.institute/> .
<https://json-ld.org/playground/#me> <http://xmlns.com/foaf/0.1/title> "Mr" .
<https://json-ld.org/playground/> <http://webns.net/mvcb/errorReportsTo> <mailto:leigh@ldodds.com> .
<https://json-ld.org/playground/> <http://webns.net/mvcb/generatorAgent>
<http://www.ldodds.com/foaf/foaf-a-matic> .
<https://json-ld.org/playground/> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://xmlns.com/foaf/0.1/PersonalProfileDocument> .

```

```
<https://json-ld.org/playground/> <http://xmlns.com/foaf/0.1/maker> <https://json-ld.org/playground/#me> .  
<https://json-ld.org/playground/> <http://xmlns.com/foaf/0.1/primaryTopic> <https://json-  
ld.org/playground/#me> .  
_:b0 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .  
_:b0 <http://xmlns.com/foaf/0.1/mbox_sha1sum> "c38c4081af5010a03974ef4ff94ae0e0876c8233" .  
_:b0 <http://xmlns.com/foaf/0.1/name> "Nesrine" .  
_:b1 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .  
_:b1 <http://xmlns.com/foaf/0.1/mbox_sha1sum> "4e7a7365cd8be3050646cbade496026148d632a3" .  
_:b1 <http://xmlns.com/foaf/0.1/name> "Shannon" .  
_:b2 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .  
_:b2 <http://xmlns.com/foaf/0.1/mbox_sha1sum> "02f98b1a0ff4d00ea380fa794464a6459f0be66e" .  
_:b2 <http://xmlns.com/foaf/0.1/name> "Sebastian"
```

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

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

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

```
GET /catalog/tv/ HTTP/1.1
```

```
Host: example.org
```

```
Accept: text/turtle; charset=UTF-8
```

and we receive the following answer:

```
HTTP/1.1 200 OK
```

```
Content-Type: text/turtle; charset=UTF-8
```

```
Link: <http://www.w3.org/ns/ldp#Resource>; rel="type",
```

```
<http://www.w3.org/ns/ldp#DirectContainer>; rel="type"
```

```
Allow: OPTIONS, HEAD, GET, POST, PUT
```

```
Accept-Post: text/turtle, application/ld+json
```

```
Content-Length: 232
```

```
ETag: W/"90231678"
```

```
@prefix ldp: <http://www.w3.org/ns/ldp#> .
```

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

```
@prefix cat: <http://example.org/vocab/catalog#> .
```

```
<> a ldp:DirectContainer; ldp:membershipResource <#cat>;
```

```
ldp:hasMemberRelation cat:hasProduct;
```

```
    dcterms:title "Container of the TV descriptions";
```

```
    ldp:contains <tv1>, <tv2> .
```

```
<#cat> a cat:Catalog; dcterms:title "Catalog of TVs"; cat:hasProduct <tv1>,  
<tv2> .
```

Which ones of the following statements are true?

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

PRACTICAL SESSIONS

Day 02: Answers to the practical session on RDF.

Software requirements

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

Create RDF

Read carefully the following statements:

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

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

Code of validated RDF in N3 syntax:

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

Code of validated RDF in XML syntax:

<ANSWER HERE/>

```
<?xml version="1.0", encoding="UTF-8"?>
<!DOCTYPE rdf:RDF [<!ENTITY vocab "http://www.unice.fr/vocab"><!ENTITY xsd
"http://www.w3.org/2001/XMLSchema#">]>
<rdf:RDF xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns# xmlns:vocab="&vocab;#"
xml:base=http://www.unice.fr/data>
<vocab:Woman rdf:about="#Jen">
<vocab:name>Jen</vocab:name>
<vocab:age rdf:datatype=http://www.w3.org/2001/XMLSchema#integer>42</vocab:age>
<vocab:shoesize rdf:datatype=http://www.w3.org/2001/XMLSchema#integer>36</vocab:shoeSize>
<vocab:trouseSize rdf:datatype=http://www.w3.org/2001/XMLSchema#integer>38</vocab:trouserSize>
<vocab:hasSpouse rdf:resource="#Seb"></vocab:hasSpouse>
<vocab:hasColleague rdf:resource="#Catherine"></vocab:hasColleague>
<vocab:hasColleague rdf:resource="#Fabien"></vocab:hasColleague>
<vocab:hasFather rdf:resource="#Thomas"></vocab:hasFather>
```

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:

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

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

<http://ns.inria.fr/fabien.gandon#me> a foaf:Person ;
```

```

foaf:depiction                                     <http://www-
sop.inria.fr/members/Fabien.Gandon/common/FabienGandonBackground.jpg> ;

foaf:family_name "Gandon" ;

foaf:givenname "Fabien" ;

foaf:homepage <http://fabien.info> ;

foaf:knows [ a foaf:Person ;

            rdfs:seeAlso <http://www-sop.inria.fr/members/Olivier.Corby/> ;

            foaf:mbox <mailto:olivier.corby@inria.fr> ;

            foaf:name "Olivier Corby" ],

[ a foaf:Person ;

  rdfs:seeAlso <http://www.i3s.unice.fr/~faron/> ;

  foaf:mbox <mailto:faron@polytech.unice.fr> ;

  foaf:name "Catherine Faron-Zucker" ] ;

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>> has foaf:maker and foaf:primaryTopic for
<<http://ns.inria.fr/fabien.gandon#me>>

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

The triple is: dbr:Paris

Rdf:type yago:WikicatCapitalsInEurope

<ANSWER HERE/>

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

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/>

“<http://www.inria.fr/2007/09/11/humans.rdfs>-instances”

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

<ANSWER HERE/>

Rdf:ID

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/>

As there is no prefix before each resource vocabulary, the default one is invoked

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

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

<ANSWER HERE/>

The default namespace is the url between brackets. Here it actually refers to a shortcut defined at the beginning of the document with the code `<!ENTITY humans`

`"http://www.inria.fr/2007/09/11/humans.rdfs">`

Hence the default namespace is <http://www.inria.fr/2007/09/11/humans.rdfs>

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

<ANSWER HERE/>

John is 37, his father of Mark, friend of Alice, husband of Jennifer son of Sophie and Harry.
His shoe size is 14, shirt size is 12, trouser size is 44

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

<ANSWER HERE/>

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

`<name>John</name>`

`<shoesize rdf:datatype="&xsd;integer">14</shoesize>`

`<age rdf:datatype="&xsd;integer">37</age>`

`</Person>`

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

`<name>Mark</name>`

`<shoesize rdf:datatype="&xsd;integer">8</shoesize>`

`<age rdf:datatype="&xsd;integer">14</age>`

`<shirtsize rdf:datatype="&xsd;integer">9</shirtsize>`

`<trousersize rdf:datatype="&xsd;integer">36</trousersize>`

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

`</Person>`

```
<Woman rdf:ID="Alice">
<hasFriend rdf:resource="#John"/>
<name>Alice</name>
</Woman>
```

```
<Woman rdf:ID="Jennifer">
<hasSpouse rdf:resource="#John"/>
<name>Jennifer</name>
</Woman>
```

```
<Person rdf:about="#John">
<shirtsize rdf:datatype="xsd:integer">12</shirtsize>
<trousersize rdf:datatype="xsd:integer">44</trousersize>
<hasParent rdf:resource="#Sophie"/>
</Person>
```

```
<Man rdf:ID="Harry">
<name>Harry</name>
<hasChild rdf:resource="#John"/>
```

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

<ANSWER HERE/>

Laurence is a lecturer, person and researcher.
She's married to William is the mother of Catherine
Has a friend Alice

```
<http://www.inria.fr/2007/09/11/humans.rdfs-instance#William> a:person;
:age 42;
:hasSpouse <http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura> ;
:name "William";
:shirtsize 13;
:shoesize 10;
:trousersize 46.
```

```
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Catherine> a :Woman;
:hasMother<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>
```

```
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura> a :Lecturer,
:Person,
:Researcher ;
:hasFriend<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Alice>;
:name « Laura ».
```

Day 02: Answers to the practical session on SHACL.

Software requirements

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

What is that shape

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

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

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

<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:

Resources

Properties

Values

When we have only variables, the results of the pattern is the exhaustive
Results are entire contents of all the triples

Screenshot:


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

Translate this query in plain English.

Select those who have spouse y

Run this query. How many answers do you get?

6

Question 5:

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

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

Query:

```
select ?name ?shoesize
where { ?x a h:Person;
       h:name ?name;
       h:shoesize ?shoesize }
```

Result:

1	John	14
2	Mark	8
3	William	10

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

Query:

```
select ?name ?shoesize
where { ?x a h:Person;
       h:name ?name
       optional { ?x h:shoesize ?shoesize } }
```

Result:

1	John	14
2	Mark	8
3	Eve	
4	David	
5	Laura	
6	William	10

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

Query:

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
select *
where {
  ?x a h:Person .
  optional { ?x h:shirtsize ?shirt }
  optional { ?x h:shoesize ?shoe }
  FILTER(?shirt > 12 || ?shoe > 8)
}
```

Result:

1	<http://www.inria.fr/2007/09/11/humans.rdfs#>	12	14
2	<http://www.inria.fr/2007/09/11/humans.rdfs#>	13	10

Question 6:

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

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

Query:


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

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

5 answers, 1 duplicate(Gaston)

2. Find a way to avoid duplicates.

Query:

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

How many answers do you get then?

4

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

Query:

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

Question 7

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

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

Query:

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

Result:

1	<http://www.inria.fr/2007/09/11/humans.rdfs...37
2	<http://www.inria.fr/2007/09/11/humans.rdfs...14
3	<http://www.inria.fr/2007/09/11/humans.rdfs...102
4	<http://www.inria.fr/2007/09/11/humans.rdfs...95
5	<http://www.inria.fr/2007/09/11/humans.rdfs...71
6	<http://www.inria.fr/2007/09/11/humans.rdfs...12
7	<http://www.inria.fr/2007/09/11/humans.rdfs...42
8	<http://www.inria.fr/2007/09/11/humans.rdfs...36

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

Query:

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
select ?x ?age
where { ?x h:age ?age . filter(?age<18) }
```

How many answers do you get?

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:

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
ask { ?x h:name "Mark";
      h:age ?age .
      filter(?age>=18) }
```

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

Query:

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
select * ( floor(?age/2) = ?age/2 as ?even)
where { ?x a h:Person .
        ?x h:age ?age. }
```

1	<http://www.inria.fr/2007/09/11/humans.rdfs#>	37	false
2	<http://www.inria.fr/2007/09/11/humans.rdfs#>	14	true
3	<http://www.inria.fr/2007/09/11/humans.rdfs#>	42	true
4	<http://www.inria.fr/2007/09/11/humans.rdfs#>	36	true

Question 8

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

Query:

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

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

Query:

<ANSWER HERE/>

??

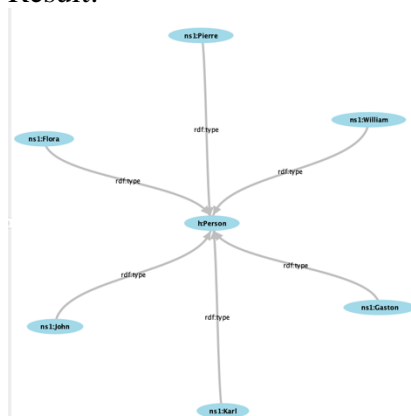
Question 9

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

Query:

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
construct { ?x a h:Person }
where {
    ?x h:age ?age .filter(?age>=18)}
```

Result:



Question 10

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

screenshots:

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

```
insert data {
i:Daisuke a h:Person;
h:name "Kuwabara";
h:hasFriend "Shannon";
h:shoesize 9;
h:age 24
}
```

?	<http://www.inria.fr/2007/09/11/humans.rdfs#a...	24
?	<http://www.inria.fr/2007/09/11/humans.rdfs#h...	Shannon
?	<http://www.inria.fr/2007/09/11/humans.rdfs#n...	Kuwabara
?	<http://www.inria.fr/2007/09/11/humans.rdfs#s...	9
?	rdf:type	<http://www.inria.fr/2007/09/11/humans.rdfs#P...

Question 11

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

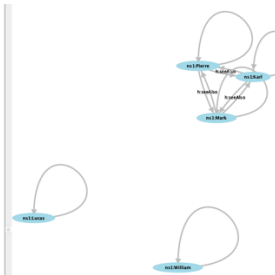
Query:

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

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

Query:

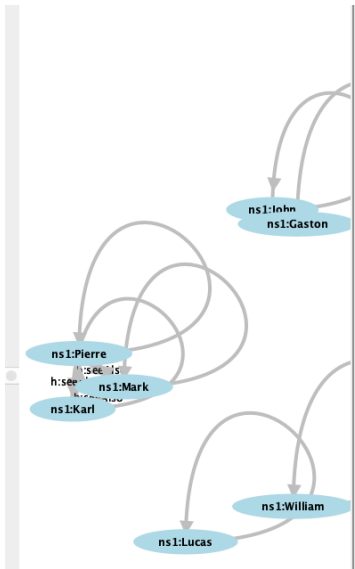
```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
construct { ?x h:seeAlso ?y }
where{
?x h:shirtsize ?s.
?y h:shirtsize ?s.
}
```



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 ?s.
?y h:shirtsize ?s.
}
```

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



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

Query:

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

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/>

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

Question 12

1. Find the largest shoe size

Query:

<ANSWER HERE/>

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

Query:

<ANSWER HERE/>

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

Query:

<ANSWER HERE/>

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

Query:

<ANSWER HERE/>

Question 13

Find couples without children

Query:

<ANSWER HERE/>

Question 14

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

Insert:

<ANSWER HERE/>

Screenshot result:

<ANSWER HERE/>

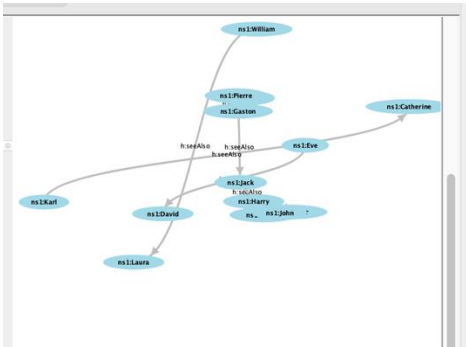
Question 15

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

query:

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

screenshot:



Question 16

Run the following query:

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

Explain what it does

By accessing to DBPedia France, get 20 married couple and show name of both partners and visualize these details using construct

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

query:

```
prefix db: <http://dbpedia.org/ontology/>
prefix foaf: <http://xmlns.com/foaf/0.1/>
```

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

```
insert { ?x h:name ?nx . ?y h:name ?ny . ?x h:hasSpouse ?y }
where {
  service <http://fr.dbpedia.org/sparql/> {
    select * where {
      ?x db:spouse ?y .
      ?x foaf:name ?nx .
      ?y foaf:name ?ny .
    }
    limit 20
  }
}
```

1	db:Guillaume IV (roi du Ro...	db:Adélaïde de Saxe-Meini...	"Guillaume IV"@fr	"Adélaïde de Saxe-Meininge...
2	db:Isabelle Ire de Jérusalem	db:Amaury II de Lusignan	"Isabelle de Jérusalem"@fr	"Amaury II de Lusignan"@fr
3	db:Ariane de Rothschild	db:Benjamin de Rothschild	"Ariane de Rothschild"@fr	"Benjamin de Rothschild"@fr
4	db:Maximilien Ier du Saint...	db:Blanche-Marie Sforza	"Maximilien"@fr	"Blanche-Marie Sforza"@fr
5	db:Philibert Ier de Savoie	db:Blanche-Marie Sforza	"Philibert Ier de Savoie"@fr	"Blanche-Marie Sforza"@fr
6	db:Bruce Paltrow	db:Blythe Danner	"Bruce Paltrow"@fr	"Blythe Danner"@fr
7	db:Dinah le teckel	db:Butch le bouledogue	"Dinah"@fr	"Butch"@fr
8	db:Jérôme Bonaparte	db:Catherine de Wurtemberg	"Jérôme Bonaparte"@fr	"Catherine de Wurtemberg"...
9	db:Gus Glouton	db:Clara Cluck	"Gus Glouton"@fr	"Clara Cluck"@fr
10	db:Panchito Pistoles	db:Clara Cluck	"Panchito Pistoles"@fr	"Clara Cluck"@fr
11	db:Wolfgang Amadeus Moz...	db:Constance Mozart	"Wolfgang Amadeus Mozart"...	"Constance Mozart"@fr
12	db:Antoinette Feuerwerker	db:David Feuerwerker	"Antoinette Feuerwerker"@fr	"David Feuerwerker"@fr
13	db:Butch le bouledogue	db:Dinah le teckel	"Butch"@fr	"Dinah"@fr
14	db:Pluto (Disney)	db:Dinah le teckel	"Pluto"@fr	"Dinah"@fr
15	db:Matilda Picsou	db:Donald Dingue	"Matilda Picsou"@fr	"Donald Dingue"@fr
16	db:Nicolae Ceausescu	db:Elena Ceausescu	"Nicolae Ceausescu"@fr	"Elena Ceausescu"@fr
17	db:Pluto (Disney)	db:Fifi le pékinois	"Pluto"@fr	"Fifi"@fr
18	db:Niall Frossach	db:Flaithbhertach	"Niall mac Fergaile Frossach"...	"Flaithbhertach"@fr
19	db:Hannelore Schmidt	db:Helmut Schmidt	"Hannelore Schmidt"@fr	"Helmut Schmidt"@fr
20	db:Irene Ovtchinnikova	db:Pierre de Grèce	"Irene Ovtchinnikova"@fr	"Pierre de Grèce"@fr

Day 04: Answers to the practical session on RDFS.

Software requirements

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

Create your own schema Family.rdfs

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

Your schema:

```
<rdf:RDF
  xml:base="http://www.unice.fr/voc"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
```

```
<rdf:Property rdf:ID="name">
</rdf:Property>
```

```
<rdf:Property rdf:ID="shoesize">
</rdf:Property>
```

```
<rdf:Property rdf:ID="age">
</rdf:Property>
```

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

```
<rdf:Property rdf:ID="hasSpouse">
</rdf:Property>
```

```
<rdf:Property rdf:ID="hasColleague">
</rdf:Property>
```

```
<rdf:Class rdf:ID="Man">
</rdf:Class>
```

```
<rdf:Class rdf:ID="Engineer">
</rdf:Class>
```

```
</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/>

- 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/>

Explanation:

<ANSWER HERE/>

All the people above have a hasSpouse or hasChild property. As a result, the system applied the property of the parent class, family Link

- 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/>

Explanation:

<ANSWER HERE/>

- 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/>

Explanation:

<ANSWER HERE/>

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/>

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/>

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

<ANSWER HERE/>

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/>

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

query:

<ANSWER HERE/>

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/>

answers:

<ANSWER HERE/>

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/>

answers:

<ANSWER HERE/>

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/>

answers:

<ANSWER HERE/>

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

query:

<ANSWER HERE/>

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

query:

<ANSWER HERE/>

screenshot:

<ANSWER HERE/>

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

query:

<ANSWER HERE/>

screenshot:

<ANSWER HERE/>

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

SELECT*

WHERE{

 ?x a rdfs: Person}

Number of results before:

<ANSWER HERE/>

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/>

17

Question 2

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

Query:

<ANSWER HERE/>

Number of results and explanation:

<ANSWER HERE/>

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/>

Number of results before and after and explanation:

<ANSWER HERE/>

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/>

Number of results and your explanation:

<ANSWER HERE/>

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/>

Your explanation of the result:

<ANSWER HERE/>

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/>

Your explanation of the result:

<ANSWER HERE/>

Question 5

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

Query:

<ANSWER HERE/>

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

Query:

<ANSWER HERE/>

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/>

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/>

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/>

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:

In the RDF <>:

Xmlns: owl = "http://www.w3.org/2002/07/owl#"

Replaced rdf: Property b owl: SymmetricProperty:

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

```
</owl:SymmetricProperty>
```

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

```
</owl:SymmetricProperty>
```

Query:

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

```
SELECT * WHERE(  
    { ?f1 h:hasFriend ?f2.  
    UNION  
    { ?s1 h:hasSpouse ?s2}}
```

Result before addition to the schema : 12 results

Result before addition to the schema:

<ANSWER HERE/>

Result after addition to the schema:

<ANSWER HERE/>

24 = 12*2,

Explanation:

Relationship between `hasFriend` and `hasSpouse` is bi-directional

<ANSWER HERE/>

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

Code added to the schema:

<ANSWER HERE/>

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

Query:

<ANSWER HERE/>

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

```
SELECT* WHERE{  
  ?x h:hasChild ?y  
}
```

Result before addition to the schema:

<ANSWER HERE/>

Result after addition to the schema:

<ANSWER HERE/>

Explanation:

<ANSWER HERE/>

For all the resources that had a `hasParent` relation with other resources,
The latter now have a `hasChild` relation with the former

3. Declare `hasAncestor` as transitive property.

Code added to the schema:

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

```
</owl:TransitiveProperty>
```

Query:

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

```
SELECT* WHERE{  
  ?x h:has Ancestor ?y}
```

Result before addition to the schema:

<ANSWER HERE/>

Result after addition to the schema:

<ANSWER HERE/>

Explanation:

<ANSWER HERE/>

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/>

Query:

<ANSWER HERE/>

Result before addition to the schema:

<ANSWER HERE/>

Result after addition to the schema:

<ANSWER HERE/>

Explanation:

<ANSWER HERE/>

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

Code added to the schema:

<ANSWER HERE/>

Query:

<ANSWER HERE/>

Result before addition to the schema:

<ANSWER HERE/>

Result after addition to the schema:

<ANSWER HERE/>

Explanation:

<ANSWER HERE/>

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

Code added to the schema:

<ANSWER HERE/>

Query:

<ANSWER HERE/>

Result before addition to the schema:

<ANSWER HERE/>

Result after addition to the schema:

<ANSWER HERE/>

Explanation:

<ANSWER HERE/>

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/>

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

<ANSWER HERE/>

Query:

<ANSWER HERE/>

Result before addition to the schema:

<ANSWER HERE/>

Result after addition to the schema:

<ANSWER HERE/>

Explanation:

<ANSWER HERE/>

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:

<ANSWER HERE/>

Result before addition to the schema:

<ANSWER HERE/>

Result after addition to the schema:

<ANSWER HERE/>

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