EURECOM
Sophia Antipolis

# Semantic Web and Information Extraction Technologies

*Semantic Web, Spring 2020*

# Practice SPARQL with CORESE and the Open Web

## Instructions:

- Send all your answers at <raphael.troncy@eurecom.fr> by **Monday 6 April** the latest
- The answers for all questions are gathered in a Word document named: **yourLastName1_yourLastName2.docx**
- Add a short description (5 lines max) in your email to state:
  - Your first name / last name (for the max 2 members of the group);
  - The total time you spent on this assignment;
  - The questions you haven't been able to do and the reasons why you have found difficult to answer these questions.

## Preliminary:

- Download the Corese semantic web engine (client and **NOT** the server) using the direct link at http://wimmics.inria.fr/doc/tutorial/corese-3.2.3c.jar. Do NOT execute it from the desktop (to avoid symbolic link).
- Unzip the data sent on your email. You should have a RDFS ontology (`human_2007_09_11.rdfs`) and an RDF knowledge base (`human_2007_09_11.rdf`).

## Part I: The RDF Knowledge Base

Open the file `human_2007_09_11.rdf` in your favorite text editor.

1. What is the namespace (prefix name and expanded URI) used for the individuals (or instances) in this knowledge base?
2. What is the namespace (prefix name and expanded URI) used by the schema of this simple ontology?
3. Write down all the information you know about John in the Turtle syntax.

## Part II: SPARQL 101

Double click on the file `corese-3.2.3c.jar` to launch the application. The interface enables you to open new tabs (click on the + button). The first tab is used to load files and see execution trace. The following tabs are used to write SPARQL queries and see their results. As you click on +, you can see more query tab.

1. Load the knowledge base contained in the files `human.rdf` **AND** `human.rdfs`.
2. The interface has a general pre-entered SPARQL query. Write instead the following query:

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

Write down in one sentence what this query means?
Run this query, how many answers do you get?
What is/are the type(s) of John?

3. Consider the following SPARQL query:

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

```
{
 ?x humans:hasSpouse ?y
}
```

Write down in one sentence what this query means?
Run this query, how many answers do you get?

4. Look at the knowledge base and write down which RDF property is used for indicating the shoe size of the people?
5. Write down the SPARQL query that provides for all the people their shoe size? How many answers do you get?
6. Write down the SPARQL query that provides for all the people their shoe size **if this information is available**? How many answers do you get?
7. Write down the SPARQL query that provides the people who have a shoe size greater than 8? You can use the `xsd:integer` function to cast a variable into an integer.
8. Write down the SPARQL query that provides the people who have a shoe size greater than 8 or who have the shirt size greater than 12?
9. Look up the URI that identifies John and ask the engine what is the description of this person using the specific SPARQL construct?
10. Write down the SPARQL query that provides the people that have at least one child? How many answers do you get? How many duplicates can you identify? Write down another SPARQL query that will remove the duplicates?
11. Write down the SPARQL query that provides all the men who do not have any children?
12. Write down the SPARQL query that provides all the people who have more than 100 years old?
13. Write down the SPARQL query that provides all the people pairs who have the same shirt size?
14. Write down the SPARQL query that provides all the people who are not men? How many answers do you get?

## ***Part III: The RDFS Ontology***

1. Close and re-launch the corese application and load only the ontology `human.rdfs`.
2. Write down a SPARQL query that provides all the classes defined in this ontology?
3. Write down a SPARQL query that provides all `subClassOf` relationships defined in this ontology?
4. Write down a SPARQL query that provides the definition and the translation of "shoe size"?
5. Write down a SPARQL query that provides all synonyms of the French term "*personne*"?

## ***Part IV: Querying the Open Web***

Let's query the open web. Use now the DBpedia SPARQL endpoint at http://dbpedia.org/sparql (or http://dbpedia.org/snorql/) For doing some queries on DBpedia, you need to know the ontology used. This ontology is described at https://wiki.dbpedia.org/services-resources/ontology (T-BOX). The SPARQL endpoint has the following built-in prefixes:

| | |
|---|---|
| dbpedia | http://dbpedia.org/resource/ |
| dbpedia-owl | http://dbpedia.org/ontology/ |
| dbpprop | http://dbpedia.org/property/ |

1. Write down a SPARQL query that counts the number of classes, object properties and datatype properties contained in the dbpedia ontology.
2. Write down a SPARQL query that lists all winners of the Nobel Prize in Physics sorted from oldest to youngest
3. Write down a SPARQL query that lists the top 10 Universities with most winners of the Nobel Prize in Physics
4. Write down a SPARQL query that provides the number of winners of the Nobel Prize in Physics who are immigrants (i.e. born in a country different from that of the University)
5. Write down the SPARQL query that provides the list of songs from Bruce Springsteen released between 1980 and 1990.

## Part V: Querying Wikidata

Let's now query Wikidata using the SPARQL endpoint at http://query.wikidata.org/. The interface comes with a query builder enabling to assist you when writing your queries as well as vary the output format. Look first at various examples among the nearly 400 provided ones.

1. Write down a SPARQL query that lists all winners of the Nobel Prize in Physics sorted from oldest to youngest
2. Write down a SPARQL query that lists the top 10 Universities with most winners of the Nobel Prize in Physics
3. Write down a SPARQL query that provides the number of winners of the Nobel Prize in Physics who are immigrants (i.e. born in a country different from that of the University)

## Part VI: SPARQL Puzzles

During the first lecture, we saw the Linked Open Numbers dataset (http://km.aifb.kit.edu/projects/numbers/) which defines numbers and their interrelations. The relationships which are defined encompass `previous`, `next`, `lessThan`, `greaterThan` and `primefactor`. Use only those relations to define SPARQL queries for the following:

1. Find all even numbers.
2. Find all numbers that are successors of one of their prime factors.
3. Find all odd numbers.
4. Find all prime numbers.
5. Find all non-prime numbers.
6. Find all twin primes (i.e., two prime numbers at a distance of 2, e.g., 17 and 19)