

Software requirements

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

Basic query on RDF human.rdf

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

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

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

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

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

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

Question 1:

Create a new tab to enter the following query and explain what it does and the results you get:

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

This is a good way to familiarize yourself with the data.

Question 2:

Create a new tab to enter the following query:

```
prefix h: <http://www.inria.fr/2007/09/11/humans.rdfs#>
select * where { ?x a ?t . filter(strstarts(?t, h:)) }
```

Translate this query in plain English.

Run this query. How many answers do you get?

Find John and his types.

Question 3:

In the previous answer, locate the URI of John.

1. formulate a SELECT query to find all the properties of John
2. request a description of John using the SPARQL clause for this.

Question 4

Create a new tab to enter the following query:

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

Translate this query in plain English.

Run this query. How many answers do you get?

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.
2. Change this query to retrieve all persons and, if available, their shoe size.
3. Change this query to retrieve all persons whose shoesize is greater than 8 or whose shirt size is greater than 12.

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.
How many answers do you get? How many duplicates do you identify in these responses?
2. Find a way to avoid duplicates.
How many answers do you get then?
3. Rewrite a query to find the Persons who have no child.

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 who are not adults.
How many answers do you get?
2. 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.
3. Write a query that indicates for each person if her age is even (true or false).

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

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.

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.

Question 11

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

2. Find the persons who have the same size shirt (shirtsize) and construct a seeAlso relationship between them.
3. Change the query into an insert.
4. Visualize the resources connected by seeAlso (use the CONSTRUCT clause).
5. Adapt the first query to find persons who have same size shoe (shoesize) and insert a seeAlso relationship between them.
6. Visualize the resources connected by seeAlso (use the CONSTRUCT clause)
7. Change the query to find the resources connected by a path consisting of one or several seeAlso.
8. Reload the engine (option reload in the menu) and rerun the last visualization query.

Question 12

1. Find the largest shoe size
2. Find people who have the biggest size of shoe (subquery + aggregate)
3. Calculate the average shoe size using the appropriate aggregation operator
4. Check the average with your own calculation using `sum()` and `count()`

Question 13

Find couples without children

Question 14

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

Question 15

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

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

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