

## Practical Session 3 – More SPARQL

### 1 More advanced SPARQL

**T1:** Install blazegraph. Go to <http://blazegraph.com>, go to the download link (bottom of the page), download and install/run the relevant package for your system (e.g. download `blazegraph.jar` and run the command line `java -jar blazegraph.jar`). If it works, you should see the blazegraph interface at the URL <http://localhost:9999> in your browser. Note that you might have to run it in ‘administrator mode’ for it to work.

If you have trouble installing blazegraph, a shared version is available at:

<http://212.227.190.14/>

**T2:** At <https://mdaquin.github.io/d/cheese.ttl> you should find an RDF graph in turtle format about cheeses. Download and open it in a text editor to see if you understand what it means. Then load it into blazegraph through the UPDATE tab on the interface, selecting the `cheese.ttl` file in the file upload facility at the bottom of the page. The UPDATE tab of the interface should normally be at <http://localhost:9999/blazegraph/#update>.

**T3:** In the query tab, use a SPARQL query to better understand what is in the cheese dataset (you can use the specification of SPARQL<sup>1</sup> to help you). In particular :

1. What are the classes used in the dataset ?
2. What are the predicates that apply to things of type Cheese.
3. For each of those predicates, how many times are they used on average per resource of type cheese, i.e., what is the proportion of cheeses that have values for each of the properties?
4. How many of the objects of triples using each of those predicates are IRIs and how many are literals?
5. What languages are used in the dataset for string literals?

**T5:** Write a query finding all cheeses that have names that contain the string “cor” in the graph, their names, their countries and their municipality if they have one. Why are some results duplicated?

**T6:** Write a query that finds all the geographical regions in the dataset, which are all the regions mentioned as countries, regions, provinces, or municipalities for cheeses.

**T7:** Write a construct query that states, for each geographical region in the graph, which cheeses they produce (using a new property for the predicate “produces”). For cheeses, keep only their names in English (that is, each region is connected through “produces” with a string that is the name of the cheese in English).

**T8:** Unacceptably, Gruyere is written as coming from France in this dataset. Using the Update endpoint (UPDATE TAB), write a query that removes the triple stating this, and write another query that inserts a triple stating that it is from Switzerland.

### 2 Project

For the project, you are asked to build a knowledge graph of children stories. The idea is that your knowledge graph could be used by someone researching a particular theme, plot, type of character, or other aspects of children stories to find relevant ones, or to compare stories with each other on those

---

<sup>1</sup><https://www.w3.org/TR/sparql11-query/>

aspects, or to analyze trends in the way stories have evolved over time and cultures. As the lectures and practical sessions go, we will learn more about how that could be done and how we could use it.

At the end of all the practical sessions, you will have to submit:

1. The RDF code of the knowledge graph.
2. A short report briefly describing the steps you have gone through, the choices you have made, any SPARQL query you have used, and a description of any code you might have written.

So, keep notes of what you do and find!

The following task is to be started at the end of this practical session and completed before the next one.

You should have, since last time, found useful things that you could reuse, integrate, and take inspiration from in the SPARQL endpoints listed below and potentially others. Create SPARQL queries that make, from those endpoints, lists of resources that could be reused in your knowledge graph (stories, authors, characters, formats, etc.).

Inspect their representation and write construct queries that extract the parts of the representation that you think you could reuse from those endpoints. Save the results either in files or in your own blazegraph installation (preferably in different named graphs).

- BNF: <https://data.bnf.fr/sparql/>
- DBpedia: <https://dbpedia.org/sparql>
- Wikidata: <https://query.wikidata.org/>
- Europeana: <https://sparql.europeana.eu/>
- National Library of Finland: <https://data.nationallibrary.fi/bib/sparql>
- Linked Movie Database: <https://triplydb.com/Triply/linkedmdb/sparql/linkedmdb>