4. SPARQL 1.1

The purpose of this lab is to learn how to formulate queries in SPARQL 1.1 to perform simple data quality analysis using SPARQL 1.1.

Formulating SPARQL 1.1 Queries

Formulate these natural language queries as SPARQL 1.1 queries and execute them against the DBpedia SPARQL endpoint.[1]

Hint: To understand which vocabulary to use for the queries, you can first access DBpedia pages that might relate to the resources that the query looks for. From these pages, you can then figure out what are the right names for classes, properties and individual resources . For instance, to find the right terms for Query 1 below, you may want to look up to some actor and movie pages on DBpedia. From there, you can learn that the property `dbo:starring`[2] might be relevant. Note that DBpedia uses the same structure as Wikipedia for its URIs. For instance, the corresponding DBpedia URI of the Wikipedia page `http://en.wikipedia.org/wiki/Quentin_Tarantino` is `http://dbpedia.org/resource/Quentin_Tarantino`.

Queries about Actors and Movies

1. Find all actors together with the number of their movies.
2. Find all productive actors (starring $\geq$ 100 movies) together with number of their movies.
3. Find all productive, single actors (starring $\geq$ 100 movies but having no spouse) together with number of their movies.
4. Find all productive, single actors (starring $\geq$ 100 movies but having no spouse), together with the number of their movies and the release date of their first movie.
5. Find all productive, single actors (starring $\geq$ 100 movies but having no spouse), together with the number of their movies, their first movie, and the release date of their first movie.
6. Classify actors whose birth place was in India based on their productivity: A for starring > 100 movies, B for starring 21-100 movies, C for starring < 20 movies.

[1] `http://dbpedia.org/sparql/`

[2] `http://dbpedia.org/ontology/starring`

7. Select five movies released in the same year as that of the latest Arnold Schwarzenegger's movie.
8. FInd all actors whose birth names end with 'Stone'.
9. Find all actors who are influenced (either directly or indirectly) by Harold Pinter.

SPARQL 1.1 Queries to Analyze Data Quality

DBpedia uses the URI `u:Actor`[3] and the URI `yago:Actor109765278`[4] to describe if someone is an actor. Analyze using SPARQL queries, how many people are there whose types are described using:

- both `u:Actor` and `yago:Actor109765278`
- only `u:Actor`
- only `yago:Actor109765278`
- only `u:Actor` and only `yago:Actor109765278`
- `u:Actor` or `yago:Actor109765278` (if a person has both, he/she must be counted only once)

What data quality observation can you draw from the results of the SPARQL queries?

[3] `http://umbel.org/umbel/rc/Actor`
[4] `http://dbpedia.org/class/yago/Actor109765278`