COMP 474/6741 Intelligent Systems (Winter 2022)

Worksheet #3: Knowledge Base Queries with SPARQL

Task 1. Quick refresher: Write two RDF Schema triples in Turtle format that 1. define a new class `ex:Professor` that is 2. a subclass of `foaf:Person` (you can use namespace abbreviations for `foaf`, `rdf` and `rdfs`):

1. .....
2. .....

Task 2. Add another triple stating that `ex:Rene` is a `ex:Professor`:

.....

Together with the triples above, a system can *infer* another triple. What is this triple and where does it come from?

.....

Task 3. How is Concordia University in the DBpedia knowledge graph *linked* to Wikidata? Find the *property* and *object* for:
<http://dbpedia.org/resource/Concordia_University> .....

Task 4. Your first SPARQL query: What can you find in DBpedia with

```
SELECT ?o
WHERE {
    <http://dbpedia.org/resource/Concordia_University>
        <http://dbpedia.org/property/address> ?o
}
```

You can run this query using DBpedia's public SPARQL endpoint at https://dbpedia.org/sparql/.

Task 5. Let's try out `DESCRIBE`: Can you explain the result from

```
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
DESCRIBE ?s
WHERE { ?s geo:lat "45.497002"^^xsd:float .
        ?s geo:long "-73.578003"^^xsd:float . }
```

Note that we started using prefix abbreviations, similar to Turtle.

Task 6. Now find all *predicates* and *objects* that have `dbr:Concordia_University` as the *subject*:

```
PREFIX dbr: <http://dbpedia.org/resource/>
SELECT .   .   .
WHERE {

  .   .   .
}
```

Hint: the subject URI is given and you need variables for the predicate and the object.

Task 7. Ok, something a bit more challenging: Create a query that prints out the *names* and **optionally** the *homepages* of all universities and colleges located in Montreal:

```
PREFIX dbr: <http://dbpedia.org/resource/>
SELECT ?uname ?uhomepage
WHERE {

  .   .   .
}
```

Hint: you can start from Concordia's URI and look for the right values to put into the query pattern.

Task 8. Using a `FILTER`, find all universities and colleges in Montreal that have more than 10000 students (`dbo:numberOfStudents`):

```
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?uni ?num
WHERE {

  .   .   .
  FILTER
}
```

Bonus task: sort the output by the number of students (you'll need an `ORDER BY` clause).

Task 9. If you ask Eliza, *"Is the Yangtze river longer than the Nile River?"*, you'll get a passive-aggressive answer like *"I'll ask the questions, if you don't mind!"*. Can you do better by writing a `SPARQL ASK` query for the DBpedia knowledge graph?

```
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
ASK
{
    .   .   .
}
```

Hint: the URIs for the two rivers are `dbr:Yangtze` and `dbr:Nile`. Find the property for the *length*, bind each value to a variable and add a `FILTER` to check if one is bigger than the other.