# COMP 474/6741 Intelligent Systems (Winter 2024)

## Worksheet #3: Knowledge Base Queries & Linked Open Data

**Task 1.** 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 2.** Your first SPARQL query: What can you find in DBpedia with

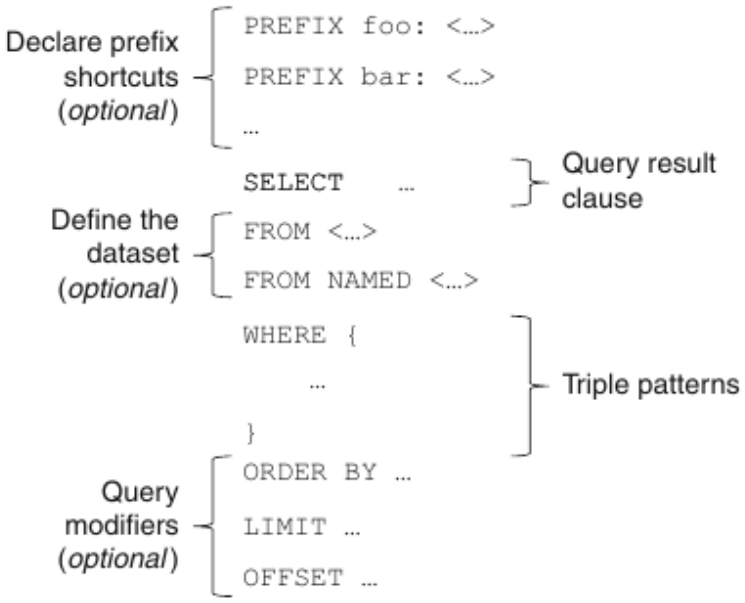
```
SELECT ?o
WHERE {
    <http://dbpedia.org/resource/Concordia_University> dbp:location ?o
}
```

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

**Task 3.** Let's try out `DESCRIBE`: Can you explain the result from

```
DESCRIBE ?s
WHERE { ?s geo:lat "45.496944"^^xsd:float .
        ?s geo:long "-73.578056"^^xsd:float . }
```

Note that the prefix abbreviations `geo` and `xsd` are pre-defined in the query interface.[1]




[1] You can see all the pre-defined prefixes here: https://dbpedia.org/sparql/?help=nsdecl

**Task 4.** Now find all *predicates* and *objects* that have `dbr:Concordia_University` as the *subject*:

```
SELECT .   .   .
WHERE {

  .   .   .
}
```

*Hint:* the subject URI is given and you need variables for the predicate and the object. Note that you can use the pre-defined prefix `dbr` (for `http://dbpedia.org/resource/`) in your query.

**Task 5.** Create a query that prints out the URI and **optionally** the *foaf:homepage* of all universities and colleges located in Montreal:

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

  .   .   .
}
```

*Hint:* Look for a property that gives you "all colleges and universities in Montreal." Note that Concordia doesn't have a `foaf:homepage` listed in DBpedia, but *Dawson College* has one.

**Task 6.** 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 7.** 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.