UNIVERSITAT POLITÈCNICA DE CATALUNYA

School of Professional & Executive Development

Master Big Data Management and Analytics
Training Linked Open Data with SPARQL

Emmanuel Jamin

Open Data and Semantic Web Consultant in Everis

emmanuel.jamin@gmail.com

Jovan Varga

Postdoc Researcher at Universitat Politècnica de Catalunya

jvarga@essi.upc.edu

Linked Open Data with SPARQL

Answers

Fill the names of the participants:

- Antonio García Serrano, antonio.garcia.serrano@gmail.com

Part B: In class practice (2h 30)

Write the SPARQL queries as a solution for each exercise here:

1. **Basic SPARQL queries (15 min)**

   a. Create the SPARQL query to find all the locations which are a country:

You have to find all the values of the locations which are countries. For this, you will first have to find URI of the Country concept in the knowledge base.

```
SELECT
    ?c
WHERE {
    ?c rdf:type <http://dbpedia.org/ontology/Country> .
}

# :Akkadian_Empire [http]
# :Almoravid_dynasty [http]
# :Ammon [http]
# :Antigonid_dynasty [http]
# :Archbishopric_of_Riga [http]
# ...
```

   b. Create the SPARQL query to find all the politicians:

You have to find all the values of the people who are politicians. For this, you will first have to find the URI of the Politician concept in the knowledge base.

```
SELECT
    str(?p)
WHERE {
    ?p rdf:type <http://dbpedia.org/ontology/Politician> .
}

# "http://dbpedia.org/resource/Blake_Richards"
# "http://dbpedia.org/resource/Cicero"
#
"http://dbpedia.org/resource/David_Charles_(Australian_politician)"
# "http://dbpedia.org/resource/Dean_Martin_(politician)"
# ...
```

   c. Create the SPARQL query to find all the companies:

You have to find all the values of the organizations which are companies. For this, you will first have to find URI of the Company concept in the knowledge base.

```
SELECT
    ?p
WHERE {
    ?p rdf:type <http://dbpedia.org/ontology/Company> .
}

# :Airbus_Helicopters [http]
# :Daimler_AG [http]
# :Dassault_Aviation [http]
# :Maxwell_Technologies [http]
# ...
```

2. **Discovering the main entities and the semantic relations (30 min)**

   a. Create the SPARQL query to find all the sub-concepts of the concept Person in the knowledge base:

You have to find all the concepts that are sub-concepts of the concept Person. For this, you will have to use the specific property that exists between the concepts.

This query should be generic and will be used as a template to find other kind of concepts.

```
SELECT
      ?class
WHERE {
      ?class rdfs:subClassOf <http://dbpedia.org/ontology/Person> .
}

# dbpedia:ontology/Ambassador [http]
# dbpedia:ontology/Archeologist [http]
# dbpedia:ontology/Architect [http]
# dbpedia:ontology/Aristocrat [http]
# dbpedia:ontology/Artist [http]
# ...
```

b. Create the SPARQL query to find all the properties that can be used to describe a person:

You have to find all the properties which belongs to the domain of the concept Person. All the retrieved properties can be used to describe a person.

This query should be generic and will be used as a template to find other kind of concepts.

```
SELECT
      DISTINCT ?property
WHERE {
      ?p rdf:type <http://dbpedia.org/ontology/Person> .
      ?p ?property ?p2
}

# rdf:type [http]
# rdfs:label [http]
# rdfs:comment [http]
# rdfs:seeAlso [http]
# owl:sameAs [http]
# dbpedia2:hometown [http]
# ...
```

c. Create the SPARQL query to find all the properties that can be used to describe a company:

You have to find all the properties which belongs to the domain of the concept Company. All the retrieved properties can be used to describe a company. For this, you can just adapt the previous query (b) to the concept Company.

```
SELECT
      DISTINCT ?property
WHERE {
      ?p rdf:type <http://dbpedia.org/ontology/Company> .
      ?p ?property ?p2 .
}

# rdf:type [http]
# rdfs:label [http]
# rdfs:comment [http]
# rdfs:seeAlso [http]
# owl:sameAs [http]
# <http://www.w3.org/ns/prov#wasDerivedFrom> [http]
# dbpedia:ontology/abstract [http]
# dbpedia:ontology/alliance [http]
# dbpedia:ontology/assets [http]
# ...
```

d. Create the SPARQL query to find a specific Property in the knowledge base

You have to find the property of the concept “Company” that refers to the industry that the company belongs to.

This query should be generic and will be used as a template to find other properties.

```
SELECT
      ?company, ?p, ?c
WHERE {
      ?company rdf:type <http://dbpedia.org/ontology/Company> .
      ?company ?p ?c .
      FILTER regex(str(?p), "industry", "i") .
}

#   :Airbus_Helicopters   [http]   dbpedia:ontology/industry   [http]
:Aerospace_manufacturer [http]
#  :Daimler_AG              [http]  dbpedia:ontology/industry  [http]
:Automotive_industry [http]
#   :Dassault_Aviation     [http]   dbpedia:ontology/industry   [http]
:Aerospace_engineering [http]
#   :Dassault_Aviation     [http]   dbpedia:ontology/industry   [http]
:Military [http]
# ...
```

e. Create the SPARQL query to find a specific value in the knowledge base

You have to find the Company that is called “IBM”. Also based on regular expression, you can search for the specific instance that contains the word “IBM”.

This query should be generic and will be used as a template to find other values.

```
SELECT
      ?company, ?name
WHERE {
      ?company rdf:type <http://dbpedia.org/ontology/Company> .
      ?company foaf:name ?name .
      FILTER regex(str(?name), "IBM") .
}

# :IBM_Toronto_Software_Lab [http] "IBM Toronto Software Lab"@en
# :Unyte                    [http] "IBM Lotus Sametime Unyte"@en
# :Kenexa                   [http] "Kenexa, an IBM Company"@en
# ...
```

3. Creating SPARQL queries (45 min)

With the previous queries, you should be able to find any kind of concepts and their semantic relations with other concepts. For the mandatory queries, you have to create at least 3 of the following queries:

a. Create the SPARQL query to find all the companies that have the United States listed as location and that work in the Media industry . In the results, you should display the Company name, the industry and the location.

```
SELECT
      DISTINCT ?company, ?typeind, ?typeloc
WHERE {
      ?company rdf:type <http://dbpedia.org/ontology/Company> .
      ?company ?industry ?typeind .
      FILTER regex(str(?industry), "industry", "i") .
      FILTER regex(str(?typeind), "media", "i") .
      ?company ?location ?typeloc .
```

```
      FILTER regex(str(?location), "location", "i") .
      FILTER regex(str(?typeloc), "united states", "i") .
}

# :Spinner_(website) [http]    :Media_player_(software)       [http]
      "United States"@en
# :Internet_Brands [http]      :Mass_media [http] "El        Segundo,
California, United States"@en
# :NGP_VAN [http]  :New_media [http]  "Somerville,     Massachusetts,
United States"@en
# ...
```

b. Create the SPARQL query to find all the companies in the Automotive industry ordered by number of employees (in descending order). In the results, you should display the Company name, the industrial sector and the number of employees.

```
SELECT
      DISTINCT ?company, ?typeind, ?num
WHERE {
      ?company rdf:type dbo:Company .
      ?company ?industry ?typeind .
      FILTER regex(str(?industry), "industry", "i") .
      FILTER regex(str(?typeind), "automotive_industry", "i") .
      ?company ?numEmployees ?num .
      FILTER regex(str(?numEmployees), "numberOfEmployees", "i") .
}
ORDER BY DESC (?num)
# :Volkswagen [http]     :Automotive_industry [http]    588902
# :Volkswagen_Group [http]     :Automotive_industry [http]    588902
# :Toyota [http]   :Automotive_industry [http]    344109
# :Daimler_AG [http]     :Automotive_industry [http]    279972
# ...
```

c. Create the SPARQL query to find all the Politicians born in United States. You should display in the results the name of the politician, the city where he was born and his function. The results should be order by cities.

```
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbpedia: <http://dbpedia.org/resource/>
SELECT ?thing WHERE {
   ?thing a dbo:Politician .
   ?thing dbo:birthPlace dbpedia:United_States.
}
# :Kenneth_Mitchell_(politician) [http]
# :Bill_Clinton [http]
# :Gérard_Ouellet [http]
# ...
```

d. Create the SPARQL query to find the Leaders of all the countries with a population higher than 10 million. In the results, you should display the country name, the leader name and the population. The list should be ordered by the country's population (descending order).

```
PREFIX o: <http://dbpedia.org/ontology/>
PREFIX p: <http://dbpedia.org/property/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?name ?pop WHERE {
    ?country a o:Country ;
        foaf:name ?name ;
             p:populationEstimate ?pop .
             FILTER (?pop >= 10000000) .
}
```

e. Create the SPARQL query to find all the movies directed by Scorsese and published after 1990. In the results, you should display the name of the director, the name of the movie and the date of the release.

<SPARQL query 3.e> ….