Semantic Web

Assignment 8

Johannes Härtel   Iryna Dubrovska

Institute of Web Science and Technologies
Department of Computer Science
University of Koblenz-Landau

Some of the tasks may require you to do additional research extending the lecture. Please keep the citation rules in mind.

For all the assignment questions that require you to write a code, make sure to include the code in the answer sheet, along with a separate python file. Where screen shots are required, please add them in the answers directly and not as separate files.

Team Name: XXXX
Team Members: XXXX

1 SPARQL

Consider the following excerpts of RDF datasets containing patient information from a single hospital (1.1) and information about flights (1.2).

Study the data structure apparent from the snippets. Based on that structure, formulate SPARQL queries for each of the specified tasks.

- Specify your queries as precisely as possible.
- You can assume the used prefixes to be known. You don't need to declare them.

1.1 Hospital

```
@prefix x:      <http://example.org/resource/> .
@prefix xo:     <http://example.org/ontology#> .
@prefix rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd:    <http://www.w3.org/2001/XMLSchema#> .

# patient information

x:pat127    rdf:type            xo:Patient.
x:pat127    rdf:type            xo:Female.
x:pat127    xo:name             "Janis Kranowitz".
x:pat127    xo:address          "Whalers Rd 127, Nantucket, MA".
x:pat127    xo:dateOfBirth      "1978-07-24"^^xsd:date.

# admission information

x:pat127    xo:admission        x:adm1272.
x:adm1272   rdf:type            xo:Admission.
x:adm1272   xo:careUnit         x:OBGyn.
x:adm1272   xo:physician        x:DrCarolineSmith.
x:adm1272   xo:reason           "Child Delivery".
x:adm1272   xo:admissionDate    "2017-03-20"^^xsd:date.
x:adm1272   xo:releaseDate      "2017-03-24"^^xsd:date.

# medical reports during patients time of admission

x:adm1272   xo:report       x:rep12721.
x:rep12721  xo:text     "C-section indicated for patient J. Kranowitz...
x:rep12721  xo:date         "2017-03-20T08:28:30"^^xsd:dateTime.

x:adm1272   xo:report       x:rep12722.
x:rep12722  xo:text         "Child delivered. It is a healthy boy.".
x:rep12722  xo:date         "2017-03-20T09:01:00"^^xsd:dateTime.

# medical costs for patients admission

x:adm1272   xo:medicalBill  x:bill362.
x:bill362   xo:description  "Caesarean section".
x:bill362   xo:cost         "9350.00"^^xsd:float.
x:adm1272   xo:medicalBill  x:bill363.
x:bill363   xo:description  "Postnatal child care".
x:bill363   xo:cost         "1980.00"^^xsd:float.
...
```

1.1.1

List all the patients that have been assigned to `x:DrCarolineSmith` for their admissions.

1.1.2

Count the number of medical reports for all the patients ever admitted to `xo:ICU`.

1.1.3

Return the 5 most costly care units of the hospital together with these costs. For that, sum up the total costs of care known for each care unit respectively.

1.2 Flights

```
@prefix ex:  <http://example.org/> .
@prefix foaf:  <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

ex:carrier rdfs:domain          ex:Flight.
ex:carrier rdfs:range           ex:Carrier.
ex:connectsTo rdfs:domain       ex:Flight.
ex:connectsTo rdfs:range        ex:Flight.

ex:LH_4718      rdf:type        ex:Flight.
ex:LH_4718      ex:carrier      ex:LH.
ex:LH_4718      ex:start        ex:Frankfurt.
ex:LH_4718      ex:destination  ex:Toronto.
ex:LH_4718      ex:sharesCodeWith ex:AC_9121.
ex:LH_4718      ex:duration     "8.5"^^xsd:nonNegativeInteger.
ex:LH       foaf:name   "Lufthansa"^^xsd:string.

ex:AZ_317       rdf:type        ex:Flight.
ex:AZ_317       ex:carrier      ex:AZ.
ex:AZ_317       ex:start        ex:Rome.
ex:AZ_317       ex:layoverAt    ex:Munich.
ex:AZ_317       ex:destination  ex:London.
ex:AZ_317       ex:connectsTo   ex:BA_13.
ex:AZ_317       ex:duration     "2.2"^^xsd:nonNegativeInteger.

...
```

1.2.1

Write a query that lists flights from `Rome` to destination `London` that take between 2 and 2.5 hours.

1.2.2

Write a query that retrieves a list of all `flights` that have 2 different intermediate stops (`at` which they have a `layover`).

1.2.3

Write a query that lists all the flights with destination `Los Angeles` starting from `Hamburg` or `Berlin`.

2 Querying DBPedia (10 Points)

In this exercise you will write SPARQL queries to retrieve information from the DBPedia SPARQL endpoint (`http://dbpedia.org/sparql`).

Please take into account that DBPedia may be under maintenance at times, so we recommend you to run your queries early.

Hints:

- `dbo` : `http://dbpedia.org/ontology/`, `dbp` : `http://dbpedia.org/property/`
- `dbr` : `http://dbpedia.org/resource/`, `foaf` : `http://xmlns.com/foaf/0.1/`
- You should use `SELECT DISTINCT` in order to avoid duplicate, if sensible for the respective task.
- You will need sensible FILTERs that modify your query results such that they fit the requirement. You can find possible expression here: `https://www.w3.org/TR/rdf-sparql-query/#OperatorMapping`.
- You will need to familiarize yourself with properties usually found with certain classes (e.g., birth date of a person). For that, explore example resources in DBPedia.

For the following tasks, specify an appropriate query and give the result you are getting (max. 10, if multiple lines are returned):

2.1

List all the countries (instances of class `dbo:Country`) that have the `title` of their leaders containing the word "president".
Note: pay attention to case sensitivity.

2.2

Zombies: List the persons (instances of `dbo:Person`) born before 1867 that are still alive today (if DBPedia is to be believed). List their names and birth dates. Count, how many zombies there are. *Hint:* create a filter also checking, whether an optional pattern (e.g. for matching a death date) was bound.

2.3

Bound by fate: List 10 pairs of (distinct) persons that share the dates for both birth and death (according to DBPedia). They should all have been born in 1927 or after.

Important Notes

Submission

- Solutions have to be submitted to the OLAT repository Submission in the respected folder.
- The name of the group and the names of all participating students with matriculation numbers must be listed on each submission.
- Solution format: all solutions as *one* PDF document. Programming code has to be submitted as Python code to the OLAT repository. Upload *all* `.py` files of your program! Use `UTF-8` as the file encoding. *Other encodings will not be taken into account!*
- Check that your code compiles without errors.
- Make sure your code is formatted to be easy to read.
  - Make sure you code has consistent indentation.
  - Make sure you comment and document your code adequately in English.
  - Choose consistent and intuitive names for your identifiers.
- Do *not* use any accents, spaces or special characters in your filenames.

Acknowledgment

This pdfLaTeX template was adapted by Jun Sun and Iryna Dubrovska based on the LuaLaTeX version by Lukas Schmelzeisen.

LaTeX

Use `pdflatex assignment_X.tex` to build your PDF.