

# SPARQL

Michal Med

October 8, 2020

## 1 Exercises – DBPedia

Use the <http://dbpedia.org/sparql> endpoint for posing queries. Answer the following questions using relevant sources described in the DBPedia ontology <http://mappings.dbpedia.org/server/ontology/classes/>, or the browser available at

[http://live.dbpedia.org/page/Czech\\_Republic](http://live.dbpedia.org/page/Czech_Republic). For the DBPedia resources and properties and for DC terms use prefixes, e.g.

```
PREFIX dct: <http://purl.org/dc/terms/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX dbrc: <http://dbpedia.org/resource/Category:>
```

**Ex. 1** — Find all types (classes), the resource describing Prague belongs to.

**Answer (Ex. 1)** —

```
SELECT DISTINCT ?c
WHERE
{
  <http://dbpedia.org/resource/Prague> a ?c
}
```

**Ex. 2** — Find labels of Prague in english and german.

**Answer (Ex. 2)** —

```
SELECT DISTINCT ?l
WHERE
{
  <http://dbpedia.org/resource/Prague> rdfs:label ?l.
```

```
FILTER (lang(?l) in ("en", "de"))
}
```

**Ex. 3** — List all EU members (countries) and show what is the relative strength of a voter in different EU countries (i.e. the ratio of total population to the number of seats in the EU parliament). Take the Czech Republic ([http://dbpedia.org/resource/Czech\\_Republic](http://dbpedia.org/resource/Czech_Republic)) as an example to explore the properties. Note that not all information is available for all countries. (Hint: Explore the type [http://dbpedia.org/resource/Category:Member\\_states\\_of\\_the\\_European\\_Union](http://dbpedia.org/resource/Category:Member_states_of_the_European_Union))

**Answer (Ex. 3)** —

```
PREFIX dct: <http://purl.org/dc/terms/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX dbrc: <http://dbpedia.org/resource/Category:>
SELECT DISTINCT ?country ?seats
    (xsd:integer(?pT) AS ?populationTotal) ?ratio
{
  ?country dct:subject dbrc:Member_states_of_the_European_Union
    a dbo:Country .

  OPTIONAL {
    ?country dbp:euseats ?seats .
  }

  OPTIONAL {
    ?country dbo:populationTotal ?pT .
  }

  BIND((xsd:double(?seats)/xsd:double(?pT)) AS ?ratio)
} ORDER BY desc(?ratio) desc(?pT)
```

**Ex. 4** — How many actors died in another country than they got born according to DBPedia. Group by countries where the actors were born. (Hint: Start with the actor type <http://dbpedia.org/resource/Actor>.)

**Answer (Ex. 4)** —

```
PREFIX dct: <http://purl.org/dc/terms/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbp: <http://dbpedia.org/property/>
SELECT ?bp (count(*) AS ?count)
{
```

```

?p dbo:occupation <http://dbpedia.org/resource/Actor> ;
  dbo:deathPlace ?dp ;
  dbo:birthPlace ?bp .
?dp a dbo:Country .
?bp a dbo:Country .
FILTER(?dp != ?bp)
} GROUP BY ?bp
ORDER BY desc(?count)

```

**Ex. 5** — Which world countries have more than one official language ? (Hint: Use HAVING)

**Answer (Ex. 5)** —

```

PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?p (count(?l) AS ?languageCount)
{
  ?p dbo:officialLanguage ?l
}
GROUP BY ?p
HAVING(count(?l) > 1)
ORDER BY desc(count(?l))

```

**Ex. 6** — Find ten countries with the highest population and show their gross domestic product per capita.

**Answer (Ex. 6)** —

```

PREFIX dct: <http://purl.org/dc/terms/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbp: <http://dbpedia.org/property/>
SELECT DISTINCT ?country ?cnt (xsd:integer(?gdp) AS ?gdpPerCapita)
WHERE
{
  ?country a dbo:Country .
  ?country dbp:populationCensus ?cnt .
  ?country dbp:gdpNominalPerCapita ?gdp .
}
ORDER BY DESC(?cnt)
LIMIT 10

```

**Ex. 7** — Find all parliamentary republics optionally together with their english label (if exist).

Answer (Ex. 7) —

```
PREFIX dct: <http://purl.org/dc/terms/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX dbr: <http://dbpedia.org/resource/>
SELECT DISTINCT ?country ?l
WHERE
{
  ?country a dbo:Country .
  ?country dbo:governmentType dbr:Parliamentary_republic .
  OPTIONAL {
    ?country rdfs:label ?l
    FILTER (lang(?l)="en")
  }
}
```

Ex. 8 — Find all female ancestors of the Prince Charles. (Hint: Start with the resource [http://dbpedia.org/resource/Charles,\\_Prince\\_of\\_Wales](http://dbpedia.org/resource/Charles,_Prince_of_Wales) and use property path expressions.)

Answer (Ex. 8) —

```
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX dbr: <http://dbpedia.org/resource/>

SELECT DISTINCT ?s WHERE {
  <http://dbpedia.org/resource/Charles,_Prince_of_Wales> dbo:parent+ ?s .
  ?s <http://xmlns.com/foaf/0.1/gender> "female"@en .
}
```

Ex. 9 — Using the CSSZ SPARQL endpoint <https://data.cssz.cz/sparql-query-editor>, create a SPARQL query finding the ratio of newly approved disability pensions comparing to the total number of conducted medical reports related to disabilities in individual years.

Compare your solution to the corresponding exercise in Seminar 1.

Answer (Ex. 9) —

```
PREFIX cssz-dimension: <https://data.cssz.cz/ontology/dimension/>
PREFIX cssz-measure: <https://data.cssz.cz/ontology/measure/>
PREFIX cssz-typ-posudku: <https://data.cssz.cz/resource/ciselnik-typy-posudku-lps/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

SELECT ?rok ?celkovy_pocet_posudku ?pocet_novych_duchodu ?pomer {
#   ?s cssz-dimension:podskupina-diagnoz-dle-who ?o
```

```

{SELECT (SUM(?pocetr) AS ?pocet_novych_duchodu) ?rok {
  ?x cssz-dimension:refPeriod [skos:notation ?rok] .
  ?x cssz-dimension:pohlavi <https://data.cssz.cz/ontology/sdmx/code/sex-T> .
  ?x cssz-measure:pocet-nove-priznanych-duchodu ?pocetr .
  ?x cssz-dimension:vekova-kategorie <https://data.cssz.cz/generated/resource/age/Y_GEO> .
  ?x cssz-dimension:skupina-diagnoz-dle-who <https://data.cssz.cz/resource/icd-10/chapter/C_T> .
} GROUP BY ?rok}

{SELECT (SUM(?pocet_posudku) AS ?celkovy_pocet_posudku) ?rok {
  ?posudek cssz-dimension:datum [skos:notation ?datum] .
  ?posudek cssz-dimension:kraj ?kraj .
  ?posudek cssz-dimension:typ-posudku ?typ_posudku .
  ?posudek cssz-measure:pocet-posudku ?pocet_posudku .
  BIND(substr(str(?datum),0,5) AS ?rok)
  FILTER(?typ_posudku IN (cssz-typ-posudku:invalidita-typ-rizeni-zjistovaci))
} GROUP BY ?rok}

BIND((?pocet_novych_duchodu/?celkovy_pocet_posudku) AS ?pomer)
} ORDER BY ASC(?rok)

```

## 2 Relevant References

- DBPedia SPARQL endpoint <http://dbpedia.org/sparql>
- CSSZ SPARQL endpoint <https://data.cssz.cz/sparql-query-editor>
- OSW ontology SPARQL endpoint <http://onto.fel.cvut.cz:7300/repositories/osw2018-ontology>
- SPARQL endpoint list <http://www.w3.org/wiki/SparqlEndpoints>