Semantic Web Systems 2015 - 2016
Sample Solution for Assignment 2 part 1

SPARQL Queries

(1) I have installed Virtuoso Open Link following these instructions:
http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/VOSUsageWindows
The Virtuoso triplestore allows me to load and query Linked Data resources.

(2) I have uploaded my data into Virtuoso. More specifically, in the Virtuoso Conductor page (by default http://localhost:8890/conductor/), I selected "Linked Data" -> "Quad Store Upload". Then I selected the Turtle file created in the previous assignment, and I created a new graph to upload the triples to:
http://vocab.inf.ed.ac.uk/sws/s1054760/assignment1part2

(3) At this point, the data is loaded in the triplestore, and it is ready to be queried. I accessed the SPARQL endpoint of my Virtuoso installation through the conductor interface under the menu "Linked Data" -> "SPARQL"

(4) I can now enter queries into the SPARQL endpoint and visualise the results.

QUERY 1

My dataset describes different types of "persons of concern", such as refugees and asylum seekers which are resident in the United Kingdom. In my first query I want to find out, which type of "persons of concern" is the most common.
I do it with the following query:
I divided the results by 63, to obtain the average across the years included in the dataset: from 1951 to 2014.

```
SELECT ?type ?num
WHERE {
{
 SELECT ?type (SUM(?t1population)/63) AS ?num
 WHERE {
  ?t1 a <http://vocab.inf.ed.ac.uk/sws/s1054760/sws/vocab/Refugees> .
  ?t1 <http://dbpedia.org/ontology/populationTotal> ?t1population .
  BIND("http://vocab.inf.ed.ac.uk/sws/s1054760/sws/vocab/Refugees" AS ?type) .
 }
} UNION
{
 SELECT ?type (SUM(?t1population)/63) AS ?num
 WHERE {
  ?t1 a <http://vocab.inf.ed.ac.uk/sws/s1054760/sws/vocab/AsylumSeekers> .
  ?t1 <http://dbpedia.org/ontology/populationTotal> ?t1population .
  BIND("http://vocab.inf.ed.ac.uk/sws/s1054760/sws/vocab/AsylumSeekers" AS ?type) .
 }
} UNION
{
 SELECT ?type (SUM(?t1population)/63) AS ?num
 WHERE {
  ?t1 a <http://vocab.inf.ed.ac.uk/sws/s1054760/sws/vocab/ReturnedRefugees> .
  ?t1 <http://dbpedia.org/ontology/populationTotal> ?t1population .
  BIND("http://vocab.inf.ed.ac.uk/sws/s1054760/sws/vocab/ReturnedRefugees" AS ?type) .
 }
} UNION
{
 SELECT ?type (SUM(?t1population)/63) AS ?num
 WHERE {
  ?t1 a <http://vocab.inf.ed.ac.uk/sws/s1054760/sws/vocab/IDPs> .
  ?t1 <http://dbpedia.org/ontology/populationTotal> ?t1population .
  BIND("http://vocab.inf.ed.ac.uk/sws/s1054760/sws/vocab/IDPs" AS ?type) .
 }
} UNION
{
 SELECT ?type (SUM(?t1population)/63) AS ?num
 WHERE {
  ?t1 a <http://vocab.inf.ed.ac.uk/sws/s1054760/sws/vocab/ReturnedIDPs> .
  ?t1 <http://dbpedia.org/ontology/populationTotal> ?t1population .
  BIND("http://vocab.inf.ed.ac.uk/sws/s1054760/sws/vocab/ReturnedIDPs" AS ?type) .
 }
} UNION
{
 SELECT ?type (SUM(?t1population)/63) AS ?num
 WHERE {
  ?t1 a <http://vocab.inf.ed.ac.uk/sws/s1054760/sws/vocab/StatelessPersons> .
  ?t1 <http://dbpedia.org/ontology/populationTotal> ?t1population .
  BIND("http://vocab.inf.ed.ac.uk/sws/s1054760/sws/vocab/StatelessPersons" AS ?type) .
 }
} UNION
{
 SELECT ?type (SUM(?t1population)/63) AS ?num
 WHERE {
  ?t1 a <http://vocab.inf.ed.ac.uk/sws/s1054760/sws/vocab/OtherPopulationOfConcern> .
  ?t1 <http://dbpedia.org/ontology/populationTotal> ?t1population .
  BIND("http://vocab.inf.ed.ac.uk/sws/s1054760/sws/vocab/OtherPopulationOfConcern" AS
?type) .
 }
} UNION
{
 SELECT ?type (SUM(?t1population)/63) AS ?num
 WHERE {
  ?t1 a <http://vocab.inf.ed.ac.uk/sws/s1054760/sws/vocab/TotalPopulationOfConcern> .
  ?t1 <http://dbpedia.org/ontology/populationTotal> ?t1population .
  BIND("http://vocab.inf.ed.ac.uk/sws/s1054760/sws/vocab/TotalPopulationOfConcern" AS
?type) .
 }
}
} ORDER BY DESC(?num)
```

The result of this query are as follows:

type	num
http://vocab.inf.ed.ac.uk/sws/s1054760/sws/vocab/TotalPopulationOfConcern	170936
http://vocab.inf.ed.ac.uk/sws/s1054760/sws/vocab/Refugees	164708
http://vocab.inf.ed.ac.uk/sws/s1054760/sws/vocab/AsylumSeekers	5970
http://vocab.inf.ed.ac.uk/sws/s1054760/sws/vocab/ReturnedRefugees	227
http://vocab.inf.ed.ac.uk/sws/s1054760/sws/vocab/StatelessPersons	29

This query uses all the numerical information about the population sizes, but it does not use information about the various years and countries of origin.

QUERY 2

Next, I ask from which countries most asylum seekers come from:

```
SELECT ?country (SUM(?t1population)/63) AS ?num
 WHERE {
  ?t1 a <http://vocab.inf.ed.ac.uk/sws/s1054760/sws/vocab/AsylumSeekers> .
  ?t1 <http://dbpedia.org/ontology/populationTotal> ?t1population .
  ?t1 <http://dbpedia.org/ontology/countryOrigin> ?country .
  FILTER(isLiteral(?country))
 } GROUP BY ?country ORDER BY DESC(?num)
```

The first 10 rows of the result are as follows:

country	num
"Various/Unknown"	4538
"Pakistan"	196
"Sri Lanka"	138

"Iran (Islamic Rep. of)"	137
"Afghanistan"	95
"Nigeria"	72
"Albania"	65
"China"	61
"Eritrea"	52
"Syrian Arab Rep."	42

This queries does not use information about the years, and it only focuses on one type of population.

QUERY 3

Next, I ask which years have seen the largest pupulation of returned refugees.

```
SELECT ?year ?t1population
 WHERE {
  ?t1 a <http://vocab.inf.ed.ac.uk/sws/s1054760/sws/vocab/ReturnedRefugees> .
  ?t1 <http://www.europeana.eu/schemas/edm/year> ?year .
  ?t1 <http://dbpedia.org/ontology/populationTotal> ?t1population .
 } ORDER BY DESC(?t1population)
```

The first 10 rows of the result are as follows:

year	**population**
2002	1771
2000	1685
1999	1428
2001	936
2012	840
2011	835
2003	823
2014	738

2006	523
2013	465

This queries does not use information about the countries of origin, and it only focuses on one type of population.

FEDERATED QUERY

In the following query, I want to obtain the same information as per Query 2, but limited to European countries. Since the information about whether a country is european or not is not included in the dataset, I run a federated query that includes DBpedia data using the following DBpedia endpoint:

http://dbpedia.org/sparql

The federated query is as follows:

```
SELECT ?countryLabel (SUM(?t1population)/63) AS ?num
WHERE
{

  {
  SELECT ?dbCountry ?alias WHERE {
  SERVICE <http://dbpedia.org/sparql>
    {
      {
      ?dbCountry <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
                     <http://dbpedia.org/class/yago/EuropeanCountries> .
      ?dbCountry <http://www.w3.org/2002/07/owl#sameAs> ?alias .
      }
     UNION
      {
      ?dbCountry <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
                     <http://dbpedia.org/class/yago/EuropeanCountries> .
      ?alias <http://www.w3.org/2002/07/owl#sameAs> ?dbCountry .
      }
     }
   }
  }

  ?t1 a <http://vocab.inf.ed.ac.uk/sws/s1054760/sws/vocab/Refugees> .
  ?t1 <http://dbpedia.org/ontology/populationTotal> ?t1population .
  ?t1 <http://dbpedia.org/ontology/countryOrigin> ?alias .
  ?t1 <http://dbpedia.org/ontology/countryOrigin> ?countryLabel .
  FILTER(isLiteral(?countryLabel))
} GROUP BY ?countryLabel ORDER BY DESC(?num)
```

The first 10 rows of the result are as follows:

countryLabel	**num**
"Russian Federation"	356
"Albania"	274
"Romania"	103
"Ukraine"	81
"The former Yugoslav Republic of Macedonia"	65
"Poland"	47
"Czech Rep."	24
"Belarus"	15
"Croatia"	10
"Azerbaijan"	10

This query uses the same data as Query 2, with the addition of DBpedia data about countries.