Study report-SP

Task1	Time Spent	Comment
SPARQL revision	1 hour	Get some basic syntax and knowledge about SPARQL via lecture notes.

I spent about an hour quickly look through relevant SPARQL lecture. Afterwards, I start to find endpoint for countries based on given query on coursework specification.

Task2-SPARQL Country	Time Spent	Comment
Find country endpoints	3.5 hours	a) DBPedia was often down b) relavant DBPedia document was hard to find c) query refactoring

Query 1:

before

```
SELECT count(?country) WHERE {
  ?country a dbo:Country .
  ?country  <http://dbpedia.org/ontology/populationTotal> ?population
}
```

after

```
PREFIX pop: <http://dbpedia.org/ontology/populationTotal>
SELECT count(?country) WHERE {
  ?country a dbo:Country .
  ?country pop:?population
}
```

It took me a lot of time in finding correct query. Basically, methodology of this task is add more condition in selection part. In other word, correct result was based on previous result. Due to the uncertainty of DBPedia, it took much time. Firstly, I can see that almost 244 countries in Mondial DB. Then the goal of this queries is about to retrieve 244 countries from DBPedia. It turned out that DBPedia has a wide definition of what counts as a "Country". Based on definition of country (i.e. Brazil), I tried several conditions one after another until I found out population actually gave the closest result. In this case, the number of country is 255. After all, I also did query formatting using PREFIX.

In conclusion, it is really inconvenient by using SPAQL to find a certain result. In order to get expected output, I had to try different conditions one by one in selection. SPARQL took much more time in finding the expected result than SQL. By the way it is case sensitive.

Task 3–connected query	**Time Spent**	**Comment**
Find corresponding city in country	3 hours	a) DBPedia was often down b) relavant DBPedia document was hard to find c) unfamiliar with SPARQL

Query 2:

```
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>

SELECT DISTINCT ?city ?country
WHERE { ?city rdf:type dbpedia-owl:City ;
          rdfs:label ?label ;
          dbpedia-owl:country ?country
}
```

Due to special feature of graph, I start to write query for retrieving “connected” attribute from country. The goal of this query is to provide corresponding country for city in DBPedia. In this case, there are two entities which are city and country for ER design. Assume database is well-designed based on normalisation and entity relationship, join operation has to be used between city and country in SQL scenario. However, in this case, SPARQL is simpler than SQL.

Task 4–LIKE operation	**Time Spent**	**Comment**
LIKE in SPARQL	2 hours	a) DBPedia was often down

Query 3:

```
SELECT DISTINCT ?C
WHERE {?C a dbo:City .
    FILTER (regex(?C, 'Man'))
}
```

This query provide city contains string 'Man'. It is quite similar with operation LIKE in SQL. But regex expression could be widely used in many scenarios. Due to its flexibility, SPARQL can work across in multiple semantic cases. Nevertheless, it is still hard to use in most occasions especially on scenario 1 of this report.

Task 5-Operational	**Time Spend**	**Comment**
Operational in SPARQL	5 hours	a) DBPedia was often down

```
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
PREFIX dbpedia: <http://dbpedia.org/resource/>

SELECT DISTINCT ?person
WHERE {
   ?person a dbpedia-owl:Person.
Optional{
   ?person dbpedia-owl:birthPlace ?country.
}
Optional{
   ?person dbpedia-owl:birthPlace ?place.
   ?place dbpedia-owl:country ?country
}
filter(?country= dbpedia:Brazil)
}
```

Again, it is very inconvenient in SPARQL. In addition, the query depends on data graph structure. Purpose of this query is to find people who born in Brazil. In relational database scenario, the result can be easily retrieved by a simple join operation.

In conclusion, the complexity and difficulty of SPARQL depend on purpose and graph structure. SQL is preferable in most cases.