SPARQL

Read

- Semantic Web Programming: chapter 6.
- Foundations of Semantic Web Technologies: chapter 7.

1 Query engine

In this exercise you are asked to make a SPARQL query engine.

1.1 Exercise

Write a java program which reads an RDF graph and a SPARQL query from file, queries the graph and outputs the query results as a table. Your program should accept `SELECT` queries, `CONSTRUCT` queries and `ASK` queries. A messages should be given if the query is of a different type.

1.1.1 Tip

If I query the Simpsons RDF graph (`simpsons.ttl`) we wrote in a previous exercise with my SPARQL query engine and the `SELECT` query

```
1  PREFIX sim: <http://www.ifi.uio.no/INF3580/simpsons#>
2  PREFIX fam: <http://www.ifi.uio.no/INF3580/family#>
3  PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
4  PREFIX foaf: <http://xmlns.com/foaf/0.1/>
5  SELECT ?s ?o
6  WHERE{ ?s foaf:age ?o }
7  LIMIT 1
```

I get[1] the following: (To get the nicely formatted output I use the class ResultSetFormatter.)

```
-------------------------------------------------------------------
| s                                                | o            |
===================================================================
| <http://www.ifi.uio.no/INF3580/simpsons#Maggie> | "1"^^xsd:int |
-------------------------------------------------------------------
```

Executing with the `ASK` query

```
1  ASK{ ?s ?p ?o }
```

gives me

[1] Note that your results may be different according to how your Simpsons RDF file looks like.

```
true
```

Executing with the CONSTRUCT query

```
1  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
2  PREFIX fam: <http://www.ifi.uio.no/INF3580/family#>
3  PREFIX sim: <http://www.ifi.uio.no/INF3580/simpsons#>
4  PREFIX foaf: <http://xmlns.com/foaf/0.1/>
5  CONSTRUCT{ sim:Bart rdfs:label ?name }
6  WHERE{ sim:Bart foaf:name ?name }
```

gives me

```
@prefix rdfs:    <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf:    <http://xmlns.com/foaf/0.1/> .
@prefix xsd:     <http://www.w3.org/2001/XMLSchema#> .
@prefix rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix sim:     <http://www.ifi.uio.no/INF3580/simpsons#> .
@prefix fam:     <http://www.ifi.uio.no/INF3580/family#> .
sim:Bart rdfs:label "Bart Simpsons" .
```

1.1.2 Solution

My program consists of three methods. One creates a model from file, `readModel`. A second, `query`, reads a query from file, executes the query on the given model, and prints the results. The third method is `main`, which reads arguments from command line and calls the two other methods.

First import the necessary APIs. We need `jena.query` and `jena.model` to execute queries on a model. `jena.util` is used for creating a model from a file.

```
1  import org.apache.jena.query.*;
2  import org.apache.jena.rdf.model.*;
3  import org.apache.jena.util.*;
4  import org.apache.jena.shared.PrefixMapping;
5  public class QueryEngine {
```

`readModel`. Loads the model from `file` using Jena's `FileManager`.

```
6  public Model readModel(String file) {
7    return FileManager.get().loadModel(file);
8  }
```

`query`. Reads the query from `file`. Creates a `QueryExecution` from the query and `model`. Executes and prints query results.

`writeQueryResult` examines the type of the query (`SELECT`, `CONSTRUCT`, `ASK`) and presents the results accordingly. If the query is a `SELECT`, then the results are printed using `query`, which implements the `PrefixMapping` interface, to create a compact table without full namespaces, i.e., the prefixes that are defined in the query file are reused for the query result. If it is a `CONSTRUCT` query, the results are printed in the Turtle serialisation format—since that's what I like best. If the query is an `ASK` query, the result, which is either `true` of `false`, is simply printed straight to standard out.

```
 9  public void query(String model_file, String query_file){
10    Model model = readModel(model_file);
11    Query query = QueryFactory.read(query_file);
12    QueryExecution qexec = QueryExecutionFactory.create(query, model);
```

```
13    writeQueryResult(query, qexec);
14  }
15
16  protected void writeQueryResult(Query query, QueryExecution qexec){
17    if(query.isSelectType()){
18      ResultSet rs = qexec.execSelect();
19      ResultSetFormatter.out(rs, query);
20    } else if(query.isConstructType()){
21      Model result = qexec.execConstruct();
22      result.write(System.out, "TTL");
23    } else if(query.isAskType()){
24      boolean result = qexec.execAsk();
25      System.out.println(result);
26    } else{ /* informative error message */ }
27    qexec.close();
28  }
```

`main`. Takes two arguments (model, query) from `args` and sends them to the other methods.

```
29    public static void main(String[] args) {
30      QueryEngine dave = new QueryEngine();
31      dave.query(args[0], args[1]);
32    }
33  } // end QueryEngine
```

2 DBpedia

DBpedia is, according to DBpedia,

> a community effort to extract structured information from Wikipedia and to make this information available on the Web. DBpedia allows you to ask sophisticated queries against Wikipedia, and to link other data sets on the Web to Wikipedia data.

We will use their SPARQL endpoint, http://dbpedia.org/sparql, to extract some information.

There is also a more fancy GUI available, if you can get anything out of it: http://dbpedia.org/isparql/.

Links:

- http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/VOSSparqlProtocol

Note that DBpedia can be slow to respond.

2.1 Exercise

Extend your query engine so that it can query a SPARQL endpoint and not just a file.

A simple solution to differentiate between if the source to query is a file or a SPARQL endpoint is to let the program read three arguments, where the first argument specifies the source, e.g., running

```
java_program file simpsons.ttl sparql_query.rq
```

results in running the query `sparql_query.rq` on the file `simpsons.ttl`, just like the query engine you have already written in a previous exercise, while running

```
java_program endpoint http://some.sparql/endpoint/ sparql_query.rq
```

returns the result of querying the endpoint `http://some.sparql/endpoint/` with the query `sparql_query.rq`.

2.1.1 Solution

The key to querying a SPARQL endpoint instead of a file is to create a `QueryExecution` object using the method `QueryExecutionFactory.sparqlService` instead of `QueryExecutionFactory.create`, which is used in the previous SPARQL query engine exercise.

The query engine is an extension of the previously created `QueryEngine` class. The method `queryFile` is (just) a renaming of the `query` method in the superclass to have more informative method names.

```
 1  import org.apache.jena.query.*;
 2  public class IrresistibleQueryEngine extends QueryEngine {
 3    public void queryEndpoint(String service, String query_file){
 4      Query query = QueryFactory.read(query_file);
 5      QueryExecution qexec = QueryExecutionFactory.sparqlService(service, query);
 6      writeQueryResult(query, qexec);
 7    }
 8    public void queryFile(String model_file, String query_file){
 9      super.query(model_file, query_file);
10    }
11    public static void main(String[] args) {
12      IrresistibleQueryEngine dave = new IrresistibleQueryEngine();
13      if(args[0].equals("endpoint")){
14        dave.queryEndpoint(args[1], args[2]);
15      } else if(args[0].equals("file")){
16        dave.queryFile(args[1], args[2]);
17      } else{ /* informative error message */ }
18    }
19  }//end IrresistibleQueryEngine
```

2.2 Exercise

Find all Simpsons characters in DBpedia and list their name, gender and relatives. Let gender and relatives be optional and list only names in English. Assume all characters have a `dcterms:subject` relation to Category:The_Simpsons_characters[2] , i.e.,

```
?character dcterms:subject
    <http://dbpedia.org/resource/Category:The_Simpsons_characters> .
```

Browse DBpedia to find the correct resource and property identifiers to use in your query.

2.2.1 Solution

The correct properties for getting to a character's gender and relatives is `dpprop:gender` and `dbprop:relatives`. A character's name is found by following the `rdfs:label` property. I found this by going to, e.g., Bart Simpson's page on dbpedia[3] .

[2] http://dbpedia.org/resource/Category:The_Simpsons_characters
[3] http://dbpedia.org/page/Bart_Simpson

Language is restricted to English by using `FILTER` and the `lang` function. See func-lang[4] .

Note that the resource `http://dbpedia.org/resource/Category:The_Simpsons_characters` cannot be written using a prefix, since the localname of the resource, `Category:The_Simpsons_characters`, contains a colon. This is a "feature" of the RDF turtle syntax.

```
 1  PREFIX dcterms: <http://purl.org/dc/terms/>
 2  PREFIX dbpedia: <http://dbpedia.org/resource/>
 3  PREFIX dbprop: <http://dbpedia.org/property/>
 4  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
 5
 6  SELECT ?person ?name ?gender ?relative
 7  WHERE {
 8    ?person dcterms:subject
 9      <http://dbpedia.org/resource/Category:The_Simpsons_characters> ;
10     rdfs:label ?name .
11  OPTIONAL{ ?person dbprop:gender ?gender }
12  OPTIONAL{ ?person dbprop:relatives ?relative }
13  FILTER (lang(?name) = 'en')
14  }
```

2.2.2 Results

person	name	gender	relative
dbpedia:Apu_Nahasapeemapetilon	"Apu Nahasapeemapetilon"@en	"Male"@en	"Brother: Sanjay"@en
dbpedia:Apu_Nahasapeemapetilon	"Apu Nahasapeemapetilon"@en	"Male"@en	"Daughters: Uma, Poonam, Pria and Sashi"@en
dbpedia:Apu_Nahasapeemapetilon	"Apu Nahasapeemapetilon"@en	"Male"@en	"Nephew: Jamshed"@en
dbpedia:Apu_Nahasapeemapetilon	"Apu Nahasapeemapetilon"@en	"Male"@en	"Niece: Pahusacheta"@en
dbpedia:Apu_Nahasapeemapetilon	"Apu Nahasapeemapetilon"@en	"Male"@en	"Sons: Anoop, Nabendu, Sandeep and Gheet"@en
dbpedia:Apu_Nahasapeemapetilon	"Apu Nahasapeemapetilon"@en	"Male"@en	"Wife: Manjula"@en
dbpedia:Barney_Gumble	"Barney Gumble"@en	"Male"@en	"Al Gumble"@en
dbpedia:Barney_Gumble	"Barney Gumble"@en	"Male"@en	"Arnie Gumble"@en
dbpedia:Barney_Gumble	"Barney Gumble"@en	"Male"@en	"Mrs. Gumble"@en
dbpedia:Barney_Gumble	"Barney Gumble"@en	"Male"@en	"Others:"@en
dbpedia:Barney_Gumble	"Barney Gumble"@en	"Male"@en	"Parents:"@en
dbpedia:Bart_Simpson	"Bart Simpson"@en	dbpedia:Male	"Aunts: Patty and Selma Bouvier"@en
dbpedia:Bart_Simpson	"Bart Simpson"@en	dbpedia:Male	"Parents: Homer and Marge"@en
dbpedia:Bart_Simpson	"Bart Simpson"@en	dbpedia:Male	"Sisters: Lisa and Maggie"@en
dbpedia:Bart_Simpson	"Bart Simpson"@en	dbpedia:Male	"Grandparents: Abe, Mona Simpson, Jacqueline
dbpedia:Chief_Wiggum	"Chief Wiggum"@en	dbpedia:Male	"Cousin: Mark"@en
dbpedia:Chief_Wiggum	"Chief Wiggum"@en	dbpedia:Male	"Father: Iggy"@en
dbpedia:Chief_Wiggum	"Chief Wiggum"@en	dbpedia:Male	"Son: Ralph"@en
dbpedia:Chief_Wiggum	"Chief Wiggum"@en	dbpedia:Male	"Wife: Sarah"@en
dbpedia:Comic_Book_Guy	"Comic Book Guy"@en	"Male"@en	"Cousin: Comic Book Gay"@en
dbpedia:Comic_Book_Guy	"Comic Book Guy"@en	"Male"@en	"Wife: Kumiko Nakamura"@en
dbpedia:Dr._Hibbert	"Dr. Hibbert"@en	dbpedia:Male	"Children:"@en
dbpedia:Dr._Hibbert	"Dr. Hibbert"@en	dbpedia:Male	"Wife: Bernice"@en
dbpedia:Dr._Hibbert	"Dr. Hibbert"@en	dbpedia:Male	"Brothers: Bleeding Gums Murphy and Shelbyvil
dbpedia:Edna_Krabappel	"Edna Krabappel"@en	dbpedia:Female	"Ex-husband: Mr. Krabappel"@en
dbpedia:Edna_Krabappel	"Edna Krabappel"@en	dbpedia:Female	"Husband: Ned Flanders"@en
dbpedia:Edna_Krabappel	"Edna Krabappel"@en	dbpedia:Female	"Step-sons: Rod and Todd Flanders"@en
dbpedia:Fat_Tony	"Fat Tony"@en	"Male"@en	dbpedia:Patty_and_Selma
dbpedia:Fat_Tony	"Fat Tony"@en	"Male"@en	"Cousin: Fit Tony"@en
dbpedia:Fat_Tony	"Fat Tony"@en	"Male"@en	"Son: Michael"@en
dbpedia:Fat_Tony	"Fat Tony"@en	"Male"@en	"Wife: Anna Maria"@en
dbpedia:Grampa_Simpson	"Grampa Simpson"@en	"Male"@en	"Children: Homer Simpson Abbie and Herb Powel
dbpedia:Grampa_Simpson	"Grampa Simpson"@en	"Male"@en	"Grandchildren: Bart, Lisa and Maggie"@en
dbpedia:Grampa_Simpson	"Grampa Simpson"@en	"Male"@en	"Wife: Mona , Rita LaFleur"@en
dbpedia:Homer_Simpson	"Homer Simpson"@en	dbpedia:Male	"Children: Bart, Lisa and Maggie"@en
dbpedia:Homer_Simpson	"Homer Simpson"@en	dbpedia:Male	"Half-brother: Herbert Powell"@en
dbpedia:Homer_Simpson	"Homer Simpson"@en	dbpedia:Male	"Parents: Abraham and Mona"@en
dbpedia:Homer_Simpson	"Homer Simpson"@en	dbpedia:Male	"Wife: Marge"@en
dbpedia:Kent_Brockman	"Kent Brockman"@en	"Male"@en	"Daughter: Brittany"@en
dbpedia:Kent_Brockman	"Kent Brockman"@en	"Male"@en	"Nephew: Unnamed"@en
dbpedia:Kent_Brockman	"Kent Brockman"@en	"Male"@en	"Sister: Unnamed"@en
dbpedia:Kent_Brockman	"Kent Brockman"@en	"Male"@en	"Wife: Stephanie"@en
dbpedia:Krusty_the_Clown	"Krusty the Clown"@en	"Male"@en	"Daughter: Sophie"@en
dbpedia:Krusty_the_Clown	"Krusty the Clown"@en	"Male"@en	"Father: Hyman Krustofski"@en
dbpedia:Lisa_Simpson	"Lisa Simpson"@en	"Female"@en	"Aunts: Patty and Selma Bouvier"@en
dbpedia:Lisa_Simpson	"Lisa Simpson"@en	"Female"@en	"Parents: Homer and Marge"@en
dbpedia:Lisa_Simpson	"Lisa Simpson"@en	"Female"@en	"Siblings: Bart and Maggie"@en
dbpedia:Lisa_Simpson	"Lisa Simpson"@en	"Female"@en	"Grandparents: Abraham Simpson, Mona Simpson,
dbpedia:Maggie_Simpson	"Maggie Simpson"@en	"Female"@en	"Aunts: Patty and Selma Bouvier"@en
dbpedia:Maggie_Simpson	"Maggie Simpson"@en	"Female"@en	"Parents: Homer and Marge"@en
dbpedia:Maggie_Simpson	"Maggie Simpson"@en	"Female"@en	"Siblings: Bart and Lisa"@en

[4] http://www.w3.org/TR/rdf-sparql-query/

| dbpedia:Maggie_Simpson | "Maggie Simpson"@en | "Female"@en | "Grandparents: Abraham Simpson, Mona Simpson,
| dbpedia:Mayor_Quimby | "Mayor Quimby"@en | "Male"@en | "Children: Numerous illegitimate"@en
| dbpedia:Mayor_Quimby | "Mayor Quimby"@en | "Male"@en | "Nephew: Freddie"@en
| dbpedia:Mayor_Quimby | "Mayor Quimby"@en | "Male"@en | "Wife: Martha"@en
| dbpedia:Nelson_Muntz | "Nelson Muntz"@en | "Male"@en | "Father: Mr Muntz"@en
| dbpedia:Nelson_Muntz | "Nelson Muntz"@en | "Male"@en | "Grandfather: Judge Muntz"@en
| dbpedia:Nelson_Muntz | "Nelson Muntz"@en | "Male"@en | "Mother: Mrs. Muntz"@en
| dbpedia:Nelson_Muntz | "Nelson Muntz"@en | "Male"@en | "Sister: unnamed"@en
| dbpedia:Otto_Mann | "Otto Mann"@en | "Male"@en | "Father: Admiral Mann"@en
| dbpedia:Otto_Mann | "Otto Mann"@en | "Male"@en | "Mother: Elisabeth Mann"@en
| dbpedia:Professor_Frink | "Professor Frink"@en | "Male"@en | "Father: John Frink, Sr."@en
| dbpedia:Professor_Frink | "Professor Frink"@en | "Male"@en | "Son: John Frink III"@en
| dbpedia:Professor_Frink | "Professor Frink"@en | "Male"@en | "Wife: name unknown"@en
| dbpedia:Reverend_Lovejoy | "Reverend Lovejoy"@en | "Male"@en | "Daughter: Jessica"@en
| dbpedia:Reverend_Lovejoy | "Reverend Lovejoy"@en | "Male"@en | "Wife: Helen"@en
| dbpedia:Sideshow_Bob | "Sideshow Bob"@en | "Male"@en | "Brother: Cecil"@en
| dbpedia:Sideshow_Bob | "Sideshow Bob"@en | "Male"@en | "Ex-wife: Selma Bouvier"@en
| dbpedia:Sideshow_Bob | "Sideshow Bob"@en | "Male"@en | "Father: Dr. Robert Terwilliger"@en
| dbpedia:Sideshow_Bob | "Sideshow Bob"@en | "Male"@en | "Mother: Dame Judith Underdunk"@en
| dbpedia:Sideshow_Bob | "Sideshow Bob"@en | "Male"@en | "Son: Gino"@en
| dbpedia:Sideshow_Bob | "Sideshow Bob"@en | "Male"@en | "Wife: Francesca"@en
| dbpedia:Troy_McClure | "Troy McClure"@en | "Male"@en | "Ex-wife: Selma Bouvier"@en
| dbpedia:Waylon_Smithers | "Waylon Smithers"@en | "Male"@en | "Unnamed mother"@en
| dbpedia:Waylon_Smithers | "Waylon Smithers"@en | "Male"@en | "Waylon Smithers, Sr."@en
| dbpedia:Marge_Simpson | "Marge Simpson"@en | "Female"@en | "Children: Bart, Lisa, Maggie"@en
| dbpedia:Marge_Simpson | "Marge Simpson"@en | "Female"@en | "Husband: Homer"@en
| dbpedia:Marge_Simpson | "Marge Simpson"@en | "Female"@en | "Parents: Clancy and Jacqueline"@en
| dbpedia:Marge_Simpson | "Marge Simpson"@en | "Female"@en | "Sisters: Patty and Selma"@en
| dbpedia:Principal_Skinner | "Principal Skinner"@en | "Male"@en | "Adoptive father: Sheldon Skinner"@en
| dbpedia:Principal_Skinner | "Principal Skinner"@en | "Male"@en | "Adoptive mother: Agnes Skinner"@en
| dbpedia:Patty_and_Selma | "Patty and Selma"@en | "Female"@en | "Brother-in-law: Homer"@en
| dbpedia:Patty_and_Selma | "Patty and Selma"@en | "Female"@en | "Daughter : Ling"@en
| dbpedia:Patty_and_Selma | "Patty and Selma"@en | "Female"@en | "Nephew: Bart"@en
| dbpedia:Patty_and_Selma | "Patty and Selma"@en | "Female"@en | "Nieces: Lisa, Maggie"@en
| dbpedia:Patty_and_Selma | "Patty and Selma"@en | "Female"@en | "Parents: Jackie, Clancy"@en
| dbpedia:Patty_and_Selma | "Patty and Selma"@en | "Female"@en | "Sister: Marge"@en
| dbpedia:Groundskeeper_Willie | "Groundskeeper Willie"@en | dbpedia:Male | "Cousins: Gravedigger Billy, Seamus"@en
| dbpedia:Groundskeeper_Willie | "Groundskeeper Willie"@en | dbpedia:Male | "Mother and father: Unnamed, appeared in \"Mo
| dbpedia:Cletus_Spuckler | "Cletus Spuckler"@en | "Male"@en | "Wife: Brandine"@en
| <http://dbpedia.org/resource/Mona_Simpson_(The_Simpsons)> | "Mona Simpson (The Simpsons)"@en | "Female"@en | "Grandchildren: Bart, Lisa, and Maggie"@en
| <http://dbpedia.org/resource/Mona_Simpson_(The_Simpsons)> | "Mona Simpson (The Simpsons)"@en | "Female"@en | "Husband: Abraham"@en
| <http://dbpedia.org/resource/Mona_Simpson_(The_Simpsons)> | "Mona Simpson (The Simpsons)"@en | "Female"@en | "Son: Homer"@en
| dbpedia:Mr._Burns | "Mr. Burns"@en | dbpedia:Male | "Father: Colonel Clifford Burns"@en
| dbpedia:Mr._Burns | "Mr. Burns"@en | dbpedia:Male | "Mother: Daphne Charles"@en
| dbpedia:Mr._Burns | "Mr. Burns"@en | dbpedia:Male | "Son: Larry"@en
| dbpedia:Ned_Flanders | "Ned Flanders"@en | "Male"@en | "Brother: Ted Flanders"@en
| dbpedia:Ned_Flanders | "Ned Flanders"@en | "Male"@en | "Children: Rod and Todd"@en
| dbpedia:Ned_Flanders | "Ned Flanders"@en | "Male"@en | "Nieces: Bonnie & Connie Flanders"@en
| dbpedia:Ned_Flanders | "Ned Flanders"@en | "Male"@en | "Parents: Mona and Nedward Flanders, Sr."@en
| dbpedia:Ned_Flanders | "Ned Flanders"@en | "Male"@en | "Wife: Maude , Ginger, Edna Krabappel"@en
| dbpedia:Ralph_Wiggum | "Ralph Wiggum"@en | "Male"@en | "Grandparents: Iggy Wiggum"@en
| dbpedia:Ralph_Wiggum | "Ralph Wiggum"@en | "Male"@en | "Parents: Clancy and Sarah"@en
| dbpedia:Milhouse_Van_Houten | "Milhouse Van Houten"@en | dbpedia:Male | "Bastardo"@en
| dbpedia:Milhouse_Van_Houten | "Milhouse Van Houten"@en | dbpedia:Male | "Grandmother: Sofia"@en
| dbpedia:Milhouse_Van_Houten | "Milhouse Van Houten"@en | dbpedia:Male | "Maternal Relatives:"@en
| dbpedia:Milhouse_Van_Houten | "Milhouse Van Houten"@en | dbpedia:Male | "Parents: Kirk and Luann"@en
| dbpedia:Milhouse_Van_Houten | "Milhouse Van Houten"@en | dbpedia:Male | "Paternal Relatives: Norbert \"Zack\" Van Hou
dbpedia:Dr._Nick	"Dr. Nick"@en	"Male"@en
dbpedia:Hans_Moleman	"Hans Moleman"@en	"Male"@en
dbpedia:Lionel_Hutz	"Lionel Hutz"@en	"Male"@en
dbpedia:Moe_Szyslak	"Moe Szyslak"@en	dbpedia:Male
<http://dbpedia.org/resource/Santa's_Little_Helper>	"Santa's Little Helper"@en	"Male greyhound"@en
dbpedia:Lenny_and_Carl	"Lenny and Carl"@en	"Male"@en
dbpedia:Kang_and_Kodos	"Kang and Kodos"@en		"Each other"@en
dbpedia:Selma_Bouvier	"Selma Bouvier"@en	
<http://dbpedia.org/resource/The_Itchy_&_Scratchy_Show>	"The Itchy & Scratchy Show"@en	
dbpedia:Simpson_family	"Simpson family"@en	

2.3 Exercise

Make a `CONSTRUCT` query which creates a graph based on the `SELECT` query you made above.
Type a character as `foaf:Person` and use the properties

- `foaf:name`
- `dbpfam:hasGenderResource`
- `dbpfam:hasGenderLiteral`
- `dbpfam:hasRelative`
- `rdfs:label`

to relate the person to its name, gender and relatives. Use `hasGenderLiteral` if the value of gender is a literal, and `hasGenderResource` if the value is a resource[5]. The object values for `fam:hasRelationshipTo` must be resources and not literals. This means that you should ignore values of relatives given as literals. `rdfs:label` shall hold the name of the character.

2.3.1 Tip

Assume the table

Person	Name	Gender	Relative
dbp:Marge_Simpson	"Marge Simpson"	"Female"	dbp:Maggie_Simpson
dbp:Lisa_Simpson	"Lisa Simpson"	dbp:Female	"Father: Homer"

is the result of running your `SELECT` query. Then your `CONSTRUCT` query should produce the following RDF graph

```
dbp:Marge_Simpson a foaf:Person ;
  foaf:name "Marge Simpson" ;
  fam:hasGenderLiteral "Female" ;
  fam:hasRelative dbp:Maggie_Simpson ;
  rdfs:label "Marge Simpson" .

dbp:Lisa_Simpson a foaf:Person ;
 foaf:name "Lisa Simpson" ;
 fam:hasGenderResource dbp:Female ;
 rdfs:label "Lisa Simpson" .
```

2.3.2 Solution

The query is listed below. The interesting bit is the `CONSTRUCT` block. Instead of a `SELECT` block, which lists which variables to be output, `CONSTRUCT` prescribes also the graph pattern that the variables should be output in.

```
 1  PREFIX dcterms: <http://purl.org/dc/terms/>
 2  PREFIX dbp: <http://dbpedia.org/resource/>
 3  PREFIX dbprop: <http://dbpedia.org/property/>
 4  PREFIX dbpfam: <http://www.ifi.uio.no/inf3580/dbpfam.rdf#>
 5  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
 6  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
 7  PREFIX foaf: <http://xmlns.com/foaf/0.1/>
 8  CONSTRUCT {
 9    ?person rdf:type foaf:Person ;
10     foaf:name ?name ;
11     dbpfam:hasGenderLiteral ?genderLiteral ;
12     dbpfam:hasGenderResource ?genderResource ;
13     dbpfam:hasRelationshipTo ?relative .
14  }
15  WHERE {
16    ?person dcterms:subject
17      <http://dbpedia.org/resource/Category:The_Simpsons_characters> ;
18     rdfs:label ?name .
19  FILTER (lang(?name) = 'en')
```

[5] We create this odd construction because we are going to re-use this ontology later, and in OWL DL object properties and datatype properties are disjoint.

```
20  OPTIONAL{ ?person dbprop:gender ?genderLiteral
21    FILTER(isLiteral(?genderLiteral)) }
22  OPTIONAL{ ?person dbprop:gender ?genderResource
23    FILTER(isURI(?genderResource)) }
24  OPTIONAL{ ?person dbprop:relatives ?relative
25    FILTER(isURI(?relative)) }
26  }
```

2.3.3 Result

```
@prefix rdfs:    <http://www.w3.org/2000/01/rdf-schema#> .
@prefix ns3:     <http://www.ifi.uio.no/inf3580/dbpfam.rdf#> .
@prefix foaf:    <http://xmlns.com/foaf/0.1/> .
@prefix rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

<http://dbpedia.org/resource/Professor_Frink>
      a       foaf:Person ;
      ns3:hasGenderLiteral
              "Male"@en ;
      foaf:name "Professor Frink"@en .

<http://dbpedia.org/resource/Hans_Moleman>
      a       foaf:Person ;
      ns3:hasGenderLiteral
              "Male"@en ;
      foaf:name "Hans Moleman"@en .

<http://dbpedia.org/resource/Sideshow_Bob>
      a       foaf:Person ;
      ns3:hasGenderLiteral
              "Male"@en ;
      foaf:name "Sideshow Bob"@en .

<http://dbpedia.org/resource/Maggie_Simpson>
      a       foaf:Person ;
      ns3:hasGenderLiteral
              "Female"@en ;
      foaf:name "Maggie Simpson"@en .

<http://dbpedia.org/resource/Bart_Simpson>
      a       foaf:Person ;
      ns3:hasGenderResource
              <http://dbpedia.org/resource/Male> ;
      foaf:name "Bart Simpson"@en .

<http://dbpedia.org/resource/Edna_Krabappel>
      a       foaf:Person ;
      ns3:hasGenderResource
              <http://dbpedia.org/resource/Female> ;
      foaf:name "Edna Krabappel"@en .

<http://dbpedia.org/resource/Reverend_Lovejoy>
      a       foaf:Person ;
      ns3:hasGenderLiteral
              "Male"@en ;
      foaf:name "Reverend Lovejoy"@en .

<http://dbpedia.org/resource/Moe_Szyslak>
      a       foaf:Person ;
      ns3:hasGenderResource
              <http://dbpedia.org/resource/Male> ;
      foaf:name "Moe Szyslak"@en .

<http://dbpedia.org/resource/Waylon_Smithers>
      a       foaf:Person ;
      ns3:hasGenderLiteral
              "Male"@en ;
```

```
      foaf:name "Waylon Smithers"@en .

<http://dbpedia.org/resource/Simpson_family>
      a       foaf:Person ;
      foaf:name "Simpson family"@en .

<http://dbpedia.org/resource/Groundskeeper_Willie>
      a       foaf:Person ;
      ns3:hasGenderResource
              <http://dbpedia.org/resource/Male> ;
      foaf:name "Groundskeeper Willie"@en .

<http://dbpedia.org/resource/Dr._Hibbert>
      a       foaf:Person ;
      ns3:hasGenderResource
              <http://dbpedia.org/resource/Male> ;
      foaf:name "Dr. Hibbert"@en .

<http://dbpedia.org/resource/Apu_Nahasapeemapetilon>
      a       foaf:Person ;
      ns3:hasGenderLiteral
              "Male"@en ;
      foaf:name "Apu Nahasapeemapetilon"@en .

<http://dbpedia.org/resource/Patty_and_Selma>
      a       foaf:Person ;
      ns3:hasGenderLiteral
              "Female"@en ;
      foaf:name "Patty and Selma"@en .

<http://dbpedia.org/resource/Milhouse_Van_Houten>
      a       foaf:Person ;
      ns3:hasGenderResource
              <http://dbpedia.org/resource/Male> ;
      foaf:name "Milhouse Van Houten"@en .

<http://dbpedia.org/resource/Santa's_Little_Helper>
      a       foaf:Person ;
      ns3:hasGenderLiteral
              "Male greyhound"@en ;
      foaf:name "Santa's Little Helper"@en .

<http://dbpedia.org/resource/Grampa_Simpson>
      a       foaf:Person ;
      ns3:hasGenderLiteral
              "Male"@en ;
      foaf:name "Grampa Simpson"@en .

<http://dbpedia.org/resource/Fat_Tony>
      a       foaf:Person ;
      ns3:hasGenderLiteral
              "Male"@en ;
      ns3:hasRelationshipTo
              <http://dbpedia.org/resource/Patty_and_Selma> ;
      foaf:name "Fat Tony"@en .

<http://dbpedia.org/resource/Mayor_Quimby>
      a       foaf:Person ;
      ns3:hasGenderLiteral
              "Male"@en ;
      foaf:name "Mayor Quimby"@en .

<http://dbpedia.org/resource/Homer_Simpson>
      a       foaf:Person ;
      ns3:hasGenderResource
              <http://dbpedia.org/resource/Male> ;
      foaf:name "Homer Simpson"@en .

<http://dbpedia.org/resource/Troy_McClure>
      a       foaf:Person ;
      ns3:hasGenderLiteral
```

```
                "Male"@en ;
        foaf:name "Troy McClure"@en .

<http://dbpedia.org/resource/Lionel_Hutz>
        a       foaf:Person ;
        ns3:hasGenderLiteral
                "Male"@en ;
        foaf:name "Lionel Hutz"@en .

<http://dbpedia.org/resource/Mr._Burns>
        a       foaf:Person ;
        ns3:hasGenderResource
                <http://dbpedia.org/resource/Male> ;
        foaf:name "Mr. Burns"@en .

<http://dbpedia.org/resource/Principal_Skinner>
        a       foaf:Person ;
        ns3:hasGenderLiteral
                "Male"@en ;
        foaf:name "Principal Skinner"@en .

<http://dbpedia.org/resource/Dr._Nick>
        a       foaf:Person ;
        ns3:hasGenderLiteral
                "Male"@en ;
        foaf:name "Dr. Nick"@en .

<http://dbpedia.org/resource/Lenny_and_Carl>
        a       foaf:Person ;
        ns3:hasGenderLiteral
                "Male"@en ;
        foaf:name "Lenny and Carl"@en .

<http://dbpedia.org/resource/Kent_Brockman>
        a       foaf:Person ;
        ns3:hasGenderLiteral
                "Male"@en ;
        foaf:name "Kent Brockman"@en .

<http://dbpedia.org/resource/Ned_Flanders>
        a       foaf:Person ;
        ns3:hasGenderLiteral
                "Male"@en ;
        foaf:name "Ned Flanders"@en .

<http://dbpedia.org/resource/Marge_Simpson>
        a       foaf:Person ;
        ns3:hasGenderLiteral
                "Female"@en ;
        foaf:name "Marge Simpson"@en .

<http://dbpedia.org/resource/The_Itchy_&_Scratchy_Show>
        a       foaf:Person ;
        foaf:name "The Itchy & Scratchy Show"@en .

<http://dbpedia.org/resource/Mona_Simpson_(The_Simpsons)>
        a       foaf:Person ;
        ns3:hasGenderLiteral
                "Female"@en ;
        foaf:name "Mona Simpson (The Simpsons)"@en .

<http://dbpedia.org/resource/Ralph_Wiggum>
        a       foaf:Person ;
        ns3:hasGenderLiteral
                "Male"@en ;
        foaf:name "Ralph Wiggum"@en .

<http://dbpedia.org/resource/Kang_and_Kodos>
        a       foaf:Person ;
        foaf:name "Kang and Kodos"@en .
```

```
<http://dbpedia.org/resource/Krusty_the_Clown>
      a       foaf:Person ;
      ns3:hasGenderLiteral
              "Male"@en ;
      foaf:name "Krusty the Clown"@en .

<http://dbpedia.org/resource/Barney_Gumble>
      a       foaf:Person ;
      ns3:hasGenderLiteral
              "Male"@en ;
      foaf:name "Barney Gumble"@en .

<http://dbpedia.org/resource/Chief_Wiggum>
      a       foaf:Person ;
      ns3:hasGenderResource
              <http://dbpedia.org/resource/Male> ;
      foaf:name "Chief Wiggum"@en .

<http://dbpedia.org/resource/Lisa_Simpson>
      a       foaf:Person ;
      ns3:hasGenderLiteral
              "Female"@en ;
      foaf:name "Lisa Simpson"@en .

<http://dbpedia.org/resource/Otto_Mann>
      a       foaf:Person ;
      ns3:hasGenderLiteral
              "Male"@en ;
      foaf:name "Otto Mann"@en .

<http://dbpedia.org/resource/Cletus_Spuckler>
      a       foaf:Person ;
      ns3:hasGenderLiteral
              "Male"@en ;
      foaf:name "Cletus Spuckler"@en .

<http://dbpedia.org/resource/Nelson_Muntz>
      a       foaf:Person ;
      ns3:hasGenderLiteral
              "Male"@en ;
      foaf:name "Nelson Muntz"@en .

<http://dbpedia.org/resource/Comic_Book_Guy>
      a       foaf:Person ;
      ns3:hasGenderLiteral
              "Male"@en ;
      foaf:name "Comic Book Guy"@en .

<http://dbpedia.org/resource/Selma_Bouvier>
      a       foaf:Person ;
      foaf:name "Selma Bouvier"@en .
```

2.4 Exercise

Explain what a `DESCRIBE` SPARQL query is. Make an example using the DBpedia SPARQL endpoint.

2.4.1 Solution

This is an example of a `DESCRIBE` query.

```
DESCRIBE <http://dbpedia.org/resource/Category:The_Simpsons_characters>
```