More SPARQL

The first exercise is to set up a real endpoint, and for this, we will use Fuseki, which is a SPARQL endpoint that uses Jena and ARQ and is developed by the same people.

1 Run Fuseki

- Download from `https://jena.apache.org/download/index.cgi`
- Unpack the file, e.g. unzip jena-fuseki-3.7.0.zip
- This will give you a directory named jena-fuseki-3.7.0/
- To start the server, run the fuseki-server script: `fuseki-server --update --mem /ds`
- You should now have a server that supports update queries, keeps the data set in memory and places the default dataset at `http://localhost:3030/ds`
- The documentation is here: `https://jena.apache.org/documentation/fuseki2/`
- Open `http://localhost:3030`. If everything is set up correctly, you will come to the frond page of the Fuseki server. Here you can manage datasets, query dataset, load data from a local file and find some information about your datasets. If you go to query, you will be able to use SPARQL-queries on you dataset.

2 RDF Dataset

Now, we will study the following query:

```
SELECT *
FROM <http://data.lenka.no/dumps/kommune-navn.ttl>
FROM <http://data.lenka.no/dumps/kommunesentre-geonames.ttl>
FROM NAMED <http://data.lenka.no/dumps/kommunesentre-geonames.ttl>
FROM <http://sws.geonames.org/6453350/about.rdf>

WHERE {
```

```
    {
     ?feature gn:officialName "Lillehammer"@no .
    } UNION {
     ?feature gn:name "Lillehammer" .
    }
  OPTIONAL {
    GRAPH <http://data.lenka.no/dumps/kommunesentre-geonames.ttl> {
    ?feature pos:lat ?lat ;
             pos:long ?long ;
             owl:sameAs ?other .
    }
  }
  OPTIONAL {
    ?feature gn:population ?pop .
  }
}
```

2.1 Load data

While some services will allow you to query external sources using the FROM-construct (e.g., `http://sparql.org/sparql`), Fuseki 2 does not. Thus we need to load the data into the triplestore and name the graphs.

1. Download the following 3 datasets:
   - `http://data.lenka.no/dumps/kommune-navn.ttl`
   - `http://data.lenka.no/dumps/kommunesentre-geonames.ttl`
   - `http://sws.geonames.org/6453350/about.rdf`
2. Open `http://localhost:3030`. To the right of the `/ds`-dataset choose **add data**.
3. In the **Destination graph name**-field enter:
   `http://data.lenka.no/dumps/kommune-navn.ttl`,
   select the `kommune-navn.ttl`-file you downloaded and press **upload now**.
4. Change **Destination graph name** to:
   `http://data.lenka.no/dumps/kommunesentre-geonames.ttl`,
   select the `kommunesentre-geonames.ttl`-file you downloaded and press **upload now**.
5. Change **Destination graph name** to:
   `http://sws.geonames.org/6453350/about.rdf`,
   select the `about.rdf`-file you downloaded and press **upload now**.

2.2 Exercise

Find necessary prefixes.

You need to begin your query with the namespace prefixes. They look like this:

```
PREFIX gd: <http://vocab.lenka.no/geo-deling#>
```

You can use the service at `http://prefix.cc/`. What other prefix declarations do you need to run the query?

2.3 Exercise

Paste the query into the text field and run it. What's the result:

2.4 Exercise

Why do you think you get three solutions to the query?

2.5 Exercise

Why is the latitude and longitude only given for the two data.lenka.no sources?

2.6 Exercise

Why is the population only given in the Geonames solution?

3 SPARQL Update

Now, we will turn to SPARQL Update. Go to: `http://localhost:3030/` and go to manage datasets. Add a new in-memory dataset called simpsons. Go back to the start-page and select query (to the right of /simpsons).

3.1 Tip

If you have problems with the update-queries, check that you are using the correct service. If the SPARQL ENDOPOINT-field is set to `/simpsons/query`, it is not possible to do update-queries. It must be set to `/simpsons` (can do both select and update queries).

3.2 Exercise

Using an update-query (in this exercise, don't upload the file!), insert the data about Homer, Marge and Lisa Simpson, available at `https://www.uio.no/studier/emner/matnat/ifi/IN3060/v21/obliger/simpsons.ttl` into the triple store. If an update it is successful, it will output:

```
<html>
<head>
</head>
<body>
<h1>Success</h1>
<p>
Update succeeded
</p>
</body>
</html>
```

You can go to info and select "count triples in all graphs" to check that the triples have been added (it should be 52 triples). You can also run a select-query (change the endpoint back to /simpsons/query) and insert the query `SELECT * WHERE { ?s ?p ?o }`.

3.3 Exercise

Homer Simpson is actually 37 years old now. Update!

3.4 Exercise

Delete the fact that someone is 34 years old.

3.5 Exercise

Delete persons aged less than 10 years.

3.6 Exercise

Delete all the Simpsons triples.

3.7 Exercise

Insert the following triples (look up the prefix):

```
<http://example.org/dahut> rdfs:label "Dahut"@en_US, "Le Dahu"@fr .
```

3.8 Exercise

Find the English-language labels irrespective of locale.

4 Property paths

In this exercise we will use the family vocabulary we have used in previous exercises. Write SPARQL queries which use property path primarily to solve the following questions. Note that some of these queries will not return any results if executed on the simpsons data at `https://www.uio.no/studier/emner/matnat/ifi/IN3060/v21/obliger/simpsons.ttl`

4.1 Exercise

Find all pairs (x,y) where y is the grandmother on the mother's side of x (mormor in Norwegian).

4.2 Exercise

Find all pairs (x,y) where y is the grandparent of x.

4.3 Exercise

Find all pairs (x,y) where y is an ancestor of x.

4.4 Exercise

Find all pairs (x,y) where y is the uncle of x.

4.5 Exercise

Find everyone who is married. List them in one column, and remember, do not use `UNION` but paths solve this.

4.6 Exercise

Find all pairs (x,y) where y is the second cousin (norsk: tremenning) of x.

Why? Because there is nothing that says that the paths must be different. You will end up with the same person as ?x and ?y, siblings, first cousins and second cousins.

5 Aggregate functions

Using the same vocabulary as in the previous exercise, solve these exercises by using aggregate functions.

5.1 Exercise

List everyone and the number of siblings they have.

5.2 Exercise

Find the oldest and youngest person.