

IN3060/4060 – MANDATORY EXERCISE no. 3

Published date: 16.02.2023.

Due date: 23.02.2023 23:59.

Delivery files: 7: exercise_1.rq, ..., exercise_7.rq.

Delivery attempts: 1.

It is wise to read through the whole exercise text first. In these exercises we will write SPARQL queries to retrieve data from a Simpsons RDF file similar to the file we wrote in a previous mandatory exercise. For each of the exercises below write a SPARQL query whose results answer the question in the exercise. The data source will use the same vocabulary as in the first mandatory exercise. An example file is found at <https://sws.ifi.uio.no/in3060/v23/oblig/3/simpsons.ttl>

The same file is loaded into a SPARQL endpoint, which can be queried for instance using the SPARQL editor at <https://sws.ifi.uio.no/yasgui/> (slow to load, be patient!). As SPARQL endpoint you should enter the following: <https://sws.ifi.uio.no/fuseki/simpsons/query>. (Note that the latter is the address of the SPARQL protocol endpoint, there is no web page to be seen there.)

Each exercise which requires a **SELECT** query specifies the result variable names you must use, i.e., the variable names that shall follow the **SELECT** word. We require this in order to easier be able to test your delivery.

See W3C's SPARQL 1.1 Query Language¹ for definitions and examples. The SPARQLer Validator² might also come in handy.

Exercise 1

Query: Find all Persons and order them by identifier, list also optionally their name. Use the result variable names `?person`, `?name`.

Tip: This is a simple query, where you will need to use **SELECT**, **WHERE**, **OPTIONAL** and **ORDER BY**. It is also important to set namespaces correctly in order for the query to work. If you do not get any results, try the query

¹<https://www.w3.org/TR/rdf-sparql-query/>

²<http://sparql.org/query-validator.html>

```
SELECT ?s ?p ?o
WHERE {?s ?p ?o}
```

and see if you get the expected results and namespaces. This query lists *all* triples in the graph.

Mr. Oblig should also be helpful in weeding out typos and similar simple mistakes.

Exercise 2

Query: Find everyone who has a mother or a father and list both the person and the mother or father. Order by mother/father. Use the result variable name `?person`, `?parent`.

Exercise 3

Query: Find everyone with a name with 'M' as first letter. Result variable name: `?person`.

Exercise 4

Query: Find all of Maggie's grandmothers. Result variable name: `?grandmother`.

Exercise 5

Query: Find everyone older than 10. Order by age, oldest first. Output name and age. Result variable names: `?person`, `?age`.

Exercise 6

Query: Is Herb the brother of Homer?

Tip: Use `ASK`.

Exercise 7

Write a `CONSTRUCT` SPARQL query that produces a FOAF file for Homer, adding his name, a `foaf:knows` relationship to his spouse, and the name of his spouse.

Ending notes

Delivery, Devilry

Mandatory exercises are to be handed in using Devilry. Make sure that you are registered in the system by logging on and finding that an `oblig3` is available as an assignment in IN3060 or IN4060. *Check this before you start solving the exercises!* If you are not registered in the system, give notice to `jieyingc@ifi.uio.no`.

You shall deliver one file per exercise. Name the files `exercise_X.rq`, where X is the exercise number.

Mr. Oblig

You can, and should, use Mr. Oblig to test your delivery before handing it in. Mr. Oblig is located at `http://sws.ifi.uio.no/mroblig/`. Note that Mr. Oblig comes without any warranty. A non-functioning Mr. Oblig will not have any effect on the due date of this mandatory exercise, and a flawless test report from Mr. Oblig does not guarantee that your delivery will be graded with *passed*. Also, since there may be many correct answers to an exercise, it is possible that Mr. Oblig does not agree with your solution even though it is correct. However, a perfect score from Mr. Oblig is an indication that your delivery is good and that it does not contain any "stupid" errors.

Good Luck!