Exercise-3

Fork: https://github.com/Scraylex/exercise-4

Findings to 1.

Query 1

```
comunica-sparql
https://solid.interactions.ics.unisg.ch/lukas/profile/card\#me \
    "SELECT ?person WHERE { ?p foaf:knows ?person. }"
```

Retrieves all the values of the variable ?person where the triple pattern ?p foaf:knows ?person holds. This query is requesting all the persons that are known by a resource identified by the variable ?p via one triple pattern. The Request sent by this query will be a GET request to the SPARQL endpoint requesting the directly known friends of the resource identified by https://solid.interactions.ics.unisg.ch/lukas/profile/card#me. No further request will be sent.

Query 2

```
comunica-sqarql-link-traversal
https://solid.interactions.ics.unisg.ch/lukas/profile/card\#me \
   "SELECT ?name WHERE { ?person foaf:knows ?friend . ?friend foaf:knows ?
foaf . ?foaf foaf:name ?name . FILTER (?person =
<https://solid.interactions.ics.unisg.ch/lukas/profile/card#me>) }"
```

Retrieves the name of the friends of the person identified by the URI https://solid.interactions.ics.unisg.ch/lukas/profile/card#me. This query uses three triple patterns and a FILTER expression to restrict the results to exclude the person from which the graph traversal is started.

1. identifies the friends of the person
2. identifies the friends of those friends
3. identifies the name of those friends of friends.

The HTTP request sent by this query will be a GET request to the SPARQL endpoint with the query string. Transitive queries in SPARQL are sent recursively to follow the path of the query. Each HTTP GET request retrieves a set of resources that are related to the previous resources in the path. These resources are then used to generate the next set of HTTP GET requests until the entire path is traversed. The final set of resources that are retrieved is the result set of the transitive query.