Cairo University
Faculty of Computers & Artificial Intelligence
Computer Science Department
Year 2021 – 2022
First Term

Lab-5

SPARQL

Given the attached family-ontology.owl file, answer the following queries.

1. Give me all people (of type Person, showing that superclasses were inferred)

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: http://www.semanticweb.org/ontologies/family-
ontology#>

SELECT ?anyPerson
     WHERE { ?anyPerson a uri:Person }
```

2. Give me every one and his child:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/ontologies/family-
ontology#>

SELECT ?anyPerson ?child
    WHERE { ?anyPerson uri:hasChild ?child.
}
```

Cairo University
Faculty of Computers & Artificial Intelligence
Computer Science Department
Year 2021 – 2022
First Term

3. showing the use of sameAs and that the classes were inferred automatically:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: <http://www.semanticweb.org/ontologies/family-
ontology#>

SELECT * WHERE { <http://www.dbpedia.com#Tim-Berners-Lee>
owl:sameAs ?son. ?son uri:age ?age }
```

Building RDF Graphs using CONSTRUCT Query

The CONSTRUCT query form returns an RDF graph. The graph is built based on a template which is used to generate RDF triples based on the results of matching the graph pattern of the query.

- **Data**

```
@prefix org:    <http://example.com/ns#> .
_:a  org:employeeName   "Alice" .
_:a  org:employeeId     12345 .

_:b  org:employeeName   "Bob" .
_:b  org:employeeId     67890
```

- Query

```
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
PREFIX org:  <http://example.com/ns#>

CONSTRUCT { ?x foaf:name ?name }
WHERE { ?x org:employeeName ?name }
```

Cairo University
Faculty of Computers & Artificial Intelligence
Computer Science Department
Year 2021 – 2022
First Term

- Output

```
@prefix org: <http://example.com/ns#> .

_:x foaf:name "Alice" .
_:y foaf:name "Bob" .
```

ASK Query

Applications can use the ASK form to test whether or not a query pattern has a solution. No information is returned about the possible query solutions, just whether or not a solution exists.

Data

```
@prefix foaf:       <http://xmlns.com/foaf/0.1/> .

_:a  foaf:name       "Alice" .

_:a  foaf:homepage  <http://work.example.org/alice/> .

_:b  foaf:name       "Bob" .

_:b foaf:mbox       <mailto:bob@work.example> .
```

Query

```
PREFIX foaf:   <http://xmlns.com/foaf/0.1/>

ASK  { ?x foaf:name  "Alice" }
```

Output

```
Yes
```

Cairo University
Faculty of Computers & Artificial Intelligence
Computer Science Department
Year 2021 – 2022
First Term

Query

```
PREFIX foaf:   <http://xmlns.com/foaf/0.1/>

ASK  { ?x foaf:name  "Alice" .

       ?x foaf:mbox  <mailto:alice@work.example> }
```

Output

No

Solution Sequences and Modifiers

Query patterns generate an unordered collection of solutions, each solution being a partial function from variables to RDF terms. These solutions are then treated as a sequence (a solution sequence), initially in no specific order; any sequence modifiers are then applied to create another sequence. Finally, this latter sequence is used to generate one of the results of a SPARQL query form.

A solution sequence modifier is one of:

Order modifier: put the solutions in order

```
SELECT ?x WHERE { ?x ?y ?z} ORDER
BY DESC(?z)
```

Distinct modifier: ensure solutions in the sequence are unique

```
SELECT DISTINCT ?x WHERE { ?x ?y ?z}
```

Reduced modifier: permit elimination of some non-unique solutions
SELECT REDUCED ?x WHERE { ?x ?y ?z}

Offset modifier: control where the solutions start from in the overall sequence of solutions
SELECT ?x WHERE { ?x ?y ?z} OFFSET 10

Limit modifier: restrict the number of solutions
SELECT ?x WHERE { ?x ?y ?z} LIMIT 10

SPARQL in JENA

Loading the model

```
//Loading model
model = ModelFactory.createDefaultModel();
InputStream in = new FileInputStream("family-
ontology.owl");
model.read(in, null);
in.close();
```

Write a query

```
String prefix = "
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uri: < http://www.semanticweb.org/ontologies/family-
ontology#> ";

//constructing select query for parents
with children String queryString = prefix+
```

```
"SELECT ?parent ?child WHERE { "+
"?parent uri:hasChild ?child."+
"?child uri:age ?age."+
"}";
```

3. Execute the query

```
Query query = QueryFactory.create(queryString) ;

try(QueryExecution qexec = QueryExecutionFactory.create(query,
model)) {
     ResultSet rs = qexec.execSelect() ;

    ResultSetFormatter.out(rs) ;
    }
```