Querying RDF schema with SPARQL

Shreya Bhajikhaye

Semantic Web and RDF

- Resource Data Framework(RDF):
  - Data model with basic unit of information as a *triple.*
  - Triple consists of a *subject*, *predicate* and *object*. It can also be referred as a resource identifier, an attribute or property name and an attribute or property value.
  - To remove any ambiguity from the information state by triple, the triple's subject and predicate must be URIs.
  - RDF serializations – N3, RFD/XML, Turtle
  - Storing large RDF data files in optimized database management systems called *triplestore*.

SPARQL

- SPARQL Protocol and RDF Query Language
- One of the key technologies of the semantic web due to its flexibility as well as ease of joining complex data structures and detecting intricate patterns in data.
- Relational databases require users to know the structure of the data.
- Conversely, the RDF data model with SPARQL is much more accessible and exploratory with limited knowledge of the data schema.
- Beneficial for implementing graph-oriented databases with complex systems of data

Triple Patterns

- Specifying subject , predicate and object

```
PREFIX  dc: <http://purl.org/dc/elements/1.1/>
PREFIX  : <http://example.org/book/>

SELECT  $title
WHERE   { :book1  dc:title  $title }
```

- Predicate-object lists

```
   ?x  foaf:name   ?name ;
       foaf:mbox   ?mbox .
```

```
?x  foaf:name   ?name .
?x  foaf:mbox   ?mbox .
```

- Object lists

```
 ?x foaf:nick  "Alice" , "Alice_" .
```

```
?x  foaf:nick  "Alice" .
?x  foaf:nick  "Alice_" .
```

Dataset file

```
# filename: ex054.ttl
@prefix ab: <http://learningsparql.com/ns/addressbook#> .
@prefix d:  <http://learningsparql.com/ns/data#> .

d:i0432 ab:firstName "Richard" .
d:i0432 ab:lastName  "Mutt" .
d:i0432 ab:homeTel   "(229) 276-5135" .
d:i0432 ab:nick      "Dick" .
d:i0432 ab:email     "richard49@hotmail.com" .

d:i9771 ab:firstName "Cindy" .
d:i9771 ab:lastName  "Marshall" .
d:i9771 ab:homeTel   "(245) 646-5488" .
d:i9771 ab:email     "cindym@gmail.com" .

d:i8301 ab:firstName "Craig" .
d:i8301 ab:lastName  "Ellis" .
d:i8301 ab:workTel   "(245) 315-5486" .
d:i8301 ab:email     "craigellis@yahoo.com" .
d:i8301 ab:email     "c.ellis@usairwaysgroup.com" .
```

Searching Strings

- FILTER & MINUS – Query to find all employees without a work telephone number

```
# filename: ex067.rq

PREFIX ab: <http://learningsparql.com/ns/addressbook#>

SELECT ?first ?last

WHERE
{
  ?s ab:firstName ?first ;
     ab:lastName ?last .
  FILTER NOT EXISTS { ?s ab:workTel ?workNum }
}
```

```
-------------------------
| first     | last      |
=========================
| "Cindy"   | "Marshall" |
| "Richard" | "Mutt"     |
-------------------------
```

```
# filename: ex068.rq

PREFIX ab: <http://learningsparql.com/ns/addressbook#>

SELECT ?first ?last

WHERE
{
  ?s ab:firstName ?first ;
     ab:lastName ?last .
  MINUS { ?s ab:workTel ?workNum }
}
```

```
-------------------------
| first     | last      |
=========================
| "Cindy"   | "Marshall" |
| "Richard" | "Mutt"     |
-------------------------
```

Selective querying

- OPTIONAL – query to find all employees and their telephone number

```
# filename: ex055.rq

PREFIX ab: <http://learningsparql.com/ns/addressbook#>

SELECT ?first ?last ?workTel
WHERE
{
  ?s ab:firstName ?first ;
     ab:lastName ?last ;
     ab:workTel ?workTel .
}
```

```
---------------------------------------
| first   | last    | workTel         |
=======================================
| "Craig" | "Ellis" | "(245) 315-5486" |
---------------------------------------
```

```
# filename: ex057.rq

PREFIX ab: <http://learningsparql.com/ns/addressbook#>

SELECT ?first ?last ?workTel
WHERE
{
  ?s ab:firstName ?first ;
     ab:lastName ?last .
  OPTIONAL
  { ?s ab:workTel ?workTel . }
}
```

```
-------------------------------------------
| first     | last       | workTel         |
===========================================
| "Craig"   | "Ellis"    | "(245) 315-5486" |
| "Cindy"   | "Marshall" |                 |
| "Richard" | "Mutt"     |                 |
-------------------------------------------
```

- Query to find the names of all employees with their telephone number and nick names

```
# filename: ex059.rq

PREFIX ab: <http://learningsparql.com/ns/addressbook#>

SELECT ?first ?last ?workTel ?nick


 WHERE
 {
   ?s ab:firstName ?first ;
      ab:lastName ?last .
   OPTIONAL
   {
      ?s ab:workTel ?workTel ;
         ab:nick ?nick .
   }
 }
```

```
-----------------------------------------
| first     | last       | workTel | nick |
=========================================
| "Craig"   | "Ellis"    |         |      |
| "Cindy"   | "Marshall" |         |      |
| "Richard" | "Mutt"     |         |      |
-----------------------------------------
```

```
# filename: ex061.rq

PREFIX ab: <http://learningsparql.com/ns/addressbook#>

SELECT ?first ?last ?workTel ?nick
WHERE
{
  ?s ab:firstName ?first ;
     ab:lastName ?last .
  OPTIONAL { ?s ab:workTel ?workTel . }
  OPTIONAL { ?s ab:nick ?nick .  }

}
```

```
-----------------------------------------------------
| first     | last       | workTel          | nick   |
=====================================================
| "Craig"   | "Ellis"    | "(245) 315-5486" |        |
| "Cindy"   | "Marshall" |                  |        |
| "Richard" | "Mutt"     |                  | "Dick" |
-----------------------------------------------------
```

WikiData Service

- Wikidata provides multiple services for its data, one of which is the Label service. It is notified through the SERVICE keyword.
- The Label service can be used to provide descriptions of the entity that is used/returned by the query.

```
SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE]". }
```

- The service also takes the output language as a parameter if the description is to be displayed in a language different from the default(English).

References

- DuCharme, B.(2013). *Learning Sparql: Querying and Updating for SPARQL 1.1*. O'Reilly
- https://www.w3.org/TR/sparql11-query/