

SPARQL

Julian Richen

November 29th, 2017

UCF

Table of contents

What is SPARQL?

RDF Format

Query Language

Should I Use It?

Resources

What is SPARQL?

SPARQL Protocol and RDF Query Language

sparkle /ˈspɑːrk(ə)l/

recursive acronym

an RDF Query Language

used to retrieve data stored in Resource Description Framework Format (RDF for short)

developed by the W3C

released in 2008 (version 1.0)

latest release was March, 2013 (version 1.1)



But wait, there's more

SPARQL queries against data stored in key-value pairs

Data in RDF is formatted as subject-predicate-object, similar to other semi-structured databases

Think NoSQL or MongoDB (“document-oriented database”)

The “subject-predicate-object” are defined by Ontologies at specified URIs

Relational databases' concept of rows and columns don't really exist

However, it can help to visualize the data like a relational database

But first

“We need to understand the database, before we can query it”

- some philosopher

RDF Format

What's RDF

It's a Resource Description Framework

Data is defined by web resources (URIs) with data serialization

Stores key-value data in “triples” or “RDF Stores”

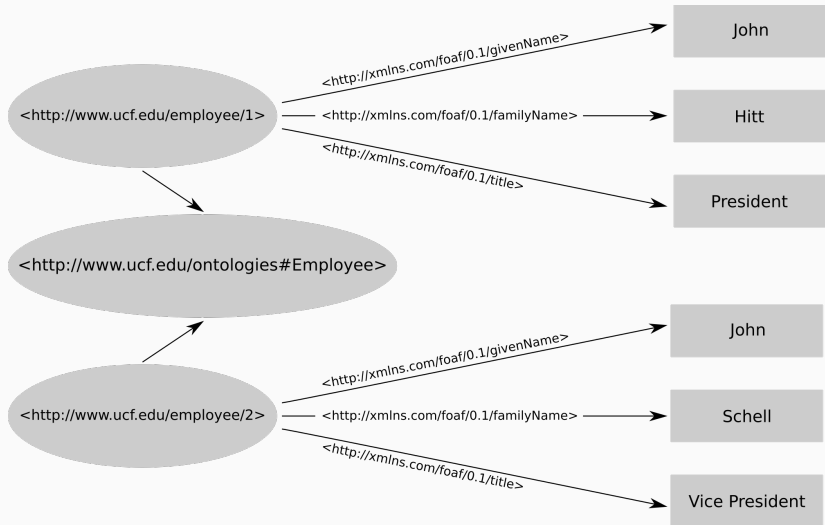
Can store data in three formats:

- RDF/XML

- N-Triples

- Turtle

Example



Visualized as a table in a relational database

employee			
id	givenName	familyName	title
1	John	Hitt	President
2	John	Schell	Vice President

Triple or RDF Store

subject	predicate	object
entity identifier	attribute name	attribute value
<code><http://www.ucf.edu/employee/1></code>	<code><http://xmlns.com/foaf/0.1/givenName></code>	<code>"John" .</code>
<code></ucf/employee/1></code>	<code></foaf/givenName></code>	<code>"John"</code>
<code>employee:1¹</code>	<code>foaf:givenName</code>	<code>"John" .</code>

¹Shorthand using prefix, explained later

RDF/XML Format

```
<?xml version="1.0" encoding="utf-8" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
  ↪ syntax-ns#"
    xmlns:foaf="http://xmlns.com/foaf/0.1/">
  <rdf:Description rdf:about="http://www.ucf.edu/
    ↪ employee/1">
    <foaf:givenName>John</foaf:givenName>
    <foaf:familyName>Hitt</foaf:familyName>
    <foaf:title>President</foaf:title>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.ucf.edu/
    ↪ employee/2">
    <foaf:givenName>John</foaf:givenName>
    <foaf:familyName>Schell</foaf:familyName>
    <foaf:title>Vice President</foaf:title>
  </rdf:Description>
</rdf:RDF>
```

N-Triples

```
<http://www.ucf.edu/employee/1> <http://xmlns.com/foaf  
  ↪ /0.1/givenName> "John" .
```

```
<http://www.ucf.edu/employee/1> <http://xmlns.com/foaf  
  ↪ /0.1/familyName> "Hitt" .
```

```
<http://www.ucf.edu/employee/1> <http://xmlns.com/foaf  
  ↪ /0.1/title> "President" .
```

```
<http://www.ucf.edu/employee/2> <http://xmlns.com/foaf  
  ↪ /0.1/givenName> "John" .
```

```
<http://www.ucf.edu/employee/2> <http://xmlns.com/foaf  
  ↪ /0.1/familyName> "Schell" .
```

```
<http://www.ucf.edu/employee/2> <http://xmlns.com/foaf  
  ↪ /0.1/title> "Vice_␣President" .
```

```
@prefix employee: <http://www.ucf.edu/employee/> .
```

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

```
employee:1 foaf:givenName "John" .
```

```
employee:1 foaf:familyName "Hitt" .
```

```
employee:1 foaf:title "President" .
```

```
employee:2 foaf:givenName "John" .
```

```
employee:2 foaf:familyName "Schell" .
```

```
employee:2 foaf:title "Vice_President" .
```

Turtle Syntax cont.

```
@prefix employee: <http://www.ucf.edu/employee/> .
```

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

```
employee:1 foaf:givenName "John" .
```

```
employee:1 foaf:familyName "Hitt" .
```

```
employee:1 foaf:title "President" .
```

```
employee:2 foaf:givenName "John" .
```

```
employee:2 foaf:familyName "Schell" .
```

```
employee:2 foaf:title "Vice_President" .
```


Match the colors

```
@prefix employee: <http://www.ucf.edu/employee/> .
```

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

```
employee:1
```

```
foaf:givenName
```

```
"John" .
```

```
<http://www.ucf.edu/employee/1>
```

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

```
"John" .
```

Query Language

Supports Create, Read, Update, and Delete operations (CRUD)

Read queries have multiple forms: SELECT, CONSTRUCT, ASK, DESCRIBE

Has helper functions like FILTER, OPTIONAL, EXISTS, NOT EXISTS, BIND, etc . . .

Federated Queries via SERVICE

And more! (check the specs)

Example

```
@prefix employee: <http://www.ucf.edu/employee/> .
```

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

```
employee:1 foaf:givenName "John" .
```

```
employee:1 foaf:familyName "Hitt" .
```

```
employee:1 foaf:title "President" .
```

```
employee:2 foaf:givenName "John" .
```

```
employee:2 foaf:familyName "Schell" .
```

```
employee:2 foaf:title "Vice_President" .
```

Example

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

```
SELECT
```

```
    ?person
```

```
WHERE {
```

```
    ?person foaf:familyName "Hitt" .
```

```
}
```

?person

employee:1

Example

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

```
SELECT
```

```
  ?person ?givenName ?familyName ?title
```

```
WHERE {
```

```
  ?person foaf:familyName "Hitt" .
```

```
  ?person foaf:givenName ?givenName .
```

```
  ?person foaf:familyName ?familyName .
```

```
  ?person foaf:title ?title .
```

```
}
```

?person	?givenName	?familyName	?title
employee:1	John	Hitt	President

Example

```
@prefix employee: <http://www.ucf.edu/employee/> .  
@prefix ucf: <http://www.ucf.edu/ontology#> .  
@prefix foaf: <http://xmlns.com/foaf/0.1/> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  
    ↪ .
```

```
employee:1 foaf:givenName "John" .  
employee:1 foaf:familyName "Hitt" .  
employee:1 foaf:title "President" .  
employee:1 rdf:type ucf:Employee .
```

```
employee:2 foaf:givenName "John" .  
employee:2 foaf:familyName "Schell" .  
employee:2 foaf:title "Vice_President" .  
employee:2 rdf:type ucf:Employee .
```

Example

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/> .  
PREFIX employee: <http://www.ucf.edu/employee/> .  
PREFIX ucf: <http://www.ucf.edu/ontology#> .
```

```
SELECT  
    ?person  
WHERE {  
    ?person a ucf:Employee .  
}
```

?person
employee:1
employee:2

Example

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/> .  
PREFIX employee: <http://www.ucf.edu/employee/> .  
PREFIX ucf: <http://www.ucf.edu/ontoilogies#> .
```

```
SELECT
```

```
    ?person ?givenName ?familyName ?title
```

```
WHERE {
```

```
    ?person a ucf:Employee .
```

```
    ?person foaf:givenName ?givenName .
```

```
    ?person foaf:familyName ?familyName .
```

```
    ?person foaf:title ?title .
```

```
}
```

?person	?givenName	?familyName	?title
employee:1	John	Hitt	President
employee:2	John	Schell	Vice President

Example

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/> .
PREFIX employee: <http://www.ucf.edu/employee/> .
PREFIX ucf: <http://www.ucf.edu/ontoilogies#> .

SELECT
    ?person ?givenName
WHERE {
    ?person foaf:givenName ?givenName .
    OPTIONAL { ?person a ucf:Employee . }
}
```

Example

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/> .
PREFIX employee: <http://www.ucf.edu/employee/> .
PREFIX ucf: <http://www.ucf.edu/ontoilogies#> .

SELECT
    ?person ?givenName
WHERE {
    ?person foaf:givenName ?givenName .
    NOT EXISTS { ?person foaf:title "President" . }
}
```

Example

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/> .
PREFIX employee: <http://www.ucf.edu/employee/> .
PREFIX ucf: <http://www.ucf.edu/ontoilogies#> .

SELECT
    ?person ?fullName
WHERE {
    ?person foaf:givenName ?givenName .
    ?person foaf:familyName ?familyName .
    BIND (concat (?givenName, "␣", ?familyName) AS ?
        ↪ fullName)
}
```

Example

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/> .
PREFIX employee: <http://www.ucf.edu/employee/> .
PREFIX ucf: <http://www.ucf.edu/ontoilogies#> .

CONSTRUCT {
    ?person foaf:name ?fullName
} WHERE {
    ?person foaf:givenName ?givenName .
    ?person foaf:familyName ?familyName .
    BIND (concat (?givenName, "␣", ?familyName) AS ?
        ↪ fullName)
}
```

Example

```
@prefix employee: <http://www.ucf.edu/employee/> .  
@prefix ucf: <http://www.ucf.edu/ontology#> .  
@prefix foaf: <http://xmlns.com/foaf/0.1/> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  
    ↪ .
```

```
employee:1 foaf:givenName "John" .  
employee:1 foaf:familyName "Hitt" .  
employee:1 foaf:name "John_Hitt" .  
employee:1 foaf:title "President" .  
employee:1 rdf:type ucf:Employee .
```

```
employee:2 foaf:givenName "John" .  
employee:2 foaf:familyName "Schell" .  
employee:1 foaf:name "John_Schell" .  
employee:2 foaf:title "Vice_President" .  
employee:2 rdf:type ucf:Employee .
```

Example

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/> .
PREFIX employee: <http://www.ucf.edu/employee/> .
PREFIX ucf: <http://www.ucf.edu/ontologies#> .

INSERT {
    ?person foaf:title "Staff"
} WHERE {
    ?person foaf:givenName "John" .
    ?person foaf:familyName "Hitt" .
    ?person foaf:title "President" .
}
```

Example

```
@prefix employee: <http://www.ucf.edu/employee/> .  
@prefix ucf: <http://www.ucf.edu/ontology#> .  
@prefix foaf: <http://xmlns.com/foaf/0.1/> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

↪ .

```
employee:1 foaf:givenName "John" .  
employee:1 foaf:familyName "Hitt" .  
employee:1 foaf:title "Staff" .  
employee:1 rdf:type ucf:Employee .
```


Example

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

```
SELECT ?person
```

```
FROM <local_database.rtf>
```

```
WHERE {
```

```
    ?person foaf:giveName "John" .
```

```
}
```

Example

```
# Named graph <http://www.ucf.edu/employee/1>  
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
```

```
employee:1 foaf:givenName "John" .  
employee:1 foaf:familyName "Hitt" .
```

```
# Query
```

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

```
SELECT ?person ?givenName ?familyName  
FROM NAMED <http://www.ucf.edu/employee/1>  
WHERE {  
    ?person foaf:giveName "John" .  
    ?person foaf:givenName ?givenName .  
    ?person foaf:familyName ?familyName .  
}
```

Example

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/> .
SELECT ?person
WHERE {
    SERVICE <http://www.ucf.edu/database> {
        SELECT ?person
        WHERE {
            ?person foaf:name "John_Doe" .
        }
    }
    SERVICE <http://www.seminolestate.edu/database> {
        SELECT ?person2
        WHERE {
            ?person2 foaf:name "John_Doe" .
        }
    }
    FILTER (?person = ?person2)
}
```

We could keep going ...

Should I Use It?

Should I Use It?

Yes and no
Good answer right?

Flexible

Modern

Standard

²buzzwords

Resources

SPARQL v1.1 Spec

<https://www.w3.org/TR/sparql11-overview/>

SPARQL in 11 minutes

<https://www.youtube.com/watch?v=FvGndkpa4K0>

L^AT_EX