

SPARQL: RDF Query Language

Maha Mallek
Université Aix Marseille et ENSI

SPARQL

- Comme nous l'avons appris, RDF est un modèle de données en forme de graphique.
- SPARQL est le langage standard pour interroger les données de graphes représentés par des triplets RDF.
 - L'acronyme signifie "SPARQL Protocol and RDF Query Language"
 - C'est l'une des trois normes fondamentales du Web sémantique, avec RDF et OWL.
 - C'est devenu un standard du W3C en Janvier 2008.
 - SPARQL 1.1 est pour l'instant une version de travail et donc au statut de brouillon.

Slide

SPARQL

Une requête SPARQL indique généralement «Je souhaite obtenir ces informations du sous-ensemble de données répondant à ces conditions».

Q1: What is the name of director D3?

```
SELECT ?directorName  
WHERE { :D3 :Name ?directorName }
```

S	P	O
...
D2	Name	Mel Gibson
D2	actedIn	M3
D3	Name	Nadine Labaki
D3	Country	C2
D3	hasWonPrizeIn	P3
D3	actedIn	M4
...

Variable

La variable:

- Il indique au moteur de requête que les triples avec une valeur quelconque dans cette position sont acceptables pour correspondre à ce modèle triple.
- Les valeurs sont stockées dans la variable afin que nous puissions les utiliser ailleurs dans la requête.

Q2: What is the name of the director of the movie M1?

```
SELECT ?directorName
WHERE
{
    :M1 :directedBy ?director .
    ?director :name ?directorName
}
```

S	P	O
M1	year	2007
M1	Name	Sicko
M1	directedBy	D1
...
M4	Name	Caramel
D1	Name	Michael Moore
D1	hasWonPrizeIn	P1
D1	Country	C1
...

Exemple

Q3: List all the movies who have directors from the USA and their directors.

Select ?movie ?director

Where {?movie :directedBy ?director.

?director :country ?country.

?country :name 'USA'}

Answer: M1 D1; M2 D1; M3 D2

S	P	O
M1	year	2007
M1	Name	Sicko
M1	directedBy	D1
M2	directedBy	D1
M2	Year	2009
M2	Name	Capitalism
M3	Year	1995
M3	directedBy	D2
M3	Name	Brave Heart
...
D1	Name	Michael Moore
D1	hasWonPrizeIn	P1
D1	Country	C1
D2	Counrty	C1
D2	hasWonPrizeIn	P2
D2	Name	Mel Gibson
D2	actedIn	M3
...
C1	Name	USA
C1	Capital	Washington DC
C2	Name	Lebanon
C2	Capital	Beirut
...

Exemple

Q4: List all the names of the directors from Lebanon who have won prizes and the prizes they have won.


```
Select ?directorName ?prize
Where { ?director :name ?directorName.
        ?director :country ?c.
        ?c :name 'Lebanon'.
        ?director :hasWonPrizeIn ?prize
}
```

Answer: 'Nadine Labaki' , P3

S	P	O
D1	Name	Michael Moore
D1	hasWonPrizeIn	P1
D1	Country	C1
D2	Counrty	C1
D2	hasWonPrizeIn	P2
D2	Name	Mel Gibson
D2	actedIn	M3
D3	Name	Nadine Labaki
D3	Country	C2
D3	hasWonPrizeIn	P3
D3	actedIn	M4
...
C1	Name	USA
C1	Capital	Washington DC
C2	Name	Lebanon
C2	Capital	Beirut
...

SPARQL

Basic Graph Pattern



```
SELECT ?directorName  
WHERE {da:D3 ab:name ?directorName}
```

```
SELECT ?directorName  
WHERE  
{  
  da:M1 ab:directedBy ?director .  
  ?director ab:name ?directorName  
}
```

Group Graph Pattern



SPARQL

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

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

```
:book1 dc:title "SPARQL Tutorial" .
```

```
:book1 ns:price 42 .
```

```
:book2 dc:title "The Semantic Web" .
```

```
:book2 ns:price 23 .
```

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

```
PREFIX ns: <http://example.org/ns#>
```

```
SELECT ?title ?price
```

```
WHERE { ?x ns:price ?price .
```

```
    FILTER ?price < 30 .
```

```
    ?x dc:title ?title . }
```



Query Results

title	price
"The Semantic Web"	23

Type de requête SPARQL

- SELECT Retour d'une table de tous X, Y, etc satisfaisant aux conditions suivantes ...
- CONSTRUCT Trouvez tous X, Y, etc répondant aux conditions suivantes ... et les remplacer dans le modèle ci-dessous afin de générer des (potentiellement nouvelles) déclarations RDF, créant ainsi un nouveau graphe.
- DESCRIBE Trouvez toutes les déclarations dans l'ensemble de données qui fournissent des informations sur la ou les ressource(s) suivante(s) ... (identifiées par leurs noms ou descriptions)
- ASK Y a-t-il une quelconque X, Y, etc qui satisfait les conditions suivantes ...

SELECT – renvoyer le nom d'une organisation avec un URI spécifique

Données d'échantillon

```
comp:A rov:haslegalName "Niké" .  
comp:A org:hasRegisteredSite site:1234 .  
  
Comp:B rov:haslegalName "BARCO" .  
  
site:1234 locn:fullAddress "Dahliastraat 24, 2160 Wommelgem .
```

Requête

```
PREFIX comp: < http://example.org/org/>  
PREFIX org: < http://www.w3.org/TR/vocab-regorg/ >  
PREFIX site: <http://example.org/site/>  
PREFIX rov: <http://www.w3.org/TR/vocab-regorg/>  
  
SELECT ?name  
  
WHERE  
{ ?x org:hasRegisteredSite site:1234 .  
  ?x rov:haslegalName ?name .}
```

Résultat

name
"Niké"

SELECT - renvoyer le nom et l'adresse des organisations

Données d'échantillon

```
comp:A rov:haslegalName "Niké" .
comp:A org:hasRegisteredSite site:1234 .

Comp:B rov:haslegalName "BARCO" .

site:1234 locn:fullAddress "Dahliastraat 24, 2160 Wommelgem" .
```

Requête

```
PREFIX org: < http://www.w3.org/TR/vocab-regorg/ >
PREFIX locn:< http://www.w3.org/ns/locn#>
PREFIX rov:<http://www.w3.org/TR/vocab-regorg/>

SELECT ?name ?address

WHERE
{ ?x org:hasRegisteredSite ?site.
  ?x rov:haslegalName ?name .
  ?site locn:fullAddress ?address . }
```

Résultat

name	address
"Niké"	"Dahliastraat 24, 2160 Wommelgem"

CONSTRUCT – créer un nouveau graphe avec une autre étiquette pour le nom

Données d'échantillon

```
comp:A rov:haslegalName "Niké" .  
comp:A org:hasRegisteredSite site:1234 .  
  
comp:B rov:haslegalName "BARCO" .  
  
site:1234 locn:fullAddress "Dahliastraat 24, 2160 Wommelgem" .
```

Requête

```
PREFIX comp: < http://example/org/org/>  
PREFIX org: < http://www.w3.org/TR/vocab-regorg/ >  
PREFIX rdfs: <http://www.w3.org/TR/rdf-schema/>  
  
CONSTRUCT {?comp rdfs:label ?name}  
  
WHERE  
{ ?comp org:haslegalName ?name. }
```

Graphe résultant

```
@prefix comp: <http://example/org/> .  
@prefix rdfs: <http://www.w3.org/TR/rdf-schema/>  
  
comp:a rdfs:label "Niké" .  
comp:b rdfs:label "BARCO" .
```

DESCRIBE - renvoyer tous les triplets des organisations inscrites sur un site particulier

Données d'échantillon

```
comp:A rov:haslegalName "Niké" .  
comp:A org:hasRegisteredSite site:1234 .  
  
comp:B rov:haslegalName "BARCO" .  
  
site:1234 locn:fullAddress "Dahliastraat 24, 2160 Wommelgem" .
```

Requête

```
PREFIX comp: <http://example/org/>  
PREFIX site: <http://example/site>  
PREFIX org: <http://www.w3.org/TR/vocab-regorg/  
  
DESCRIBE ?organisation  
  
WHERE  
{?organisation org:hasRegisteredSite site:1234}
```

Résultat

```
@prefix comp: <http://example/org/> .  
@prefix org: <http://www.w3.org/TR/vocab-regorg/> .  
  
comp:A has:legalName "Niké" .  
comp:A org:hasRegisteredSite site:1234 .
```

ASK – vérifier s’il y a des organisations ayant "1234" comme leur site enregistré?

Données d'échantillon

comp:A rov:haslegalName "Niké" .
comp:A org:hasRegisteredSite site:1234 .

comp:B rov:haslegalName "BARCO" .

site:1234 locn:fullAddress "Dahliastraat 24, 2160 Wommelgem" .

Requête

PREFIX org: < http://www.w3.org/TR/vocab-regorg/

ASK

WHERE

{?organisation org:hasRegisteredSite site:1234}

Résultat

TRUE

ASK – vérifier s’il y a un site agréé pour l’organisation “BARCO”?

Données d'échantillon

comp:A rov:haslegalName “Niké” .
comp:A org:hasRegisteredSite site:1234 .

comp:B rov:haslegalName “BARCO” .

site:1234 locn:fullAddress “Dahliastraat 24, 2160 Wommelgem” .

Requête

PREFIX comp: <http://example/org/>
PREFIX org: <http://www.w3.org/TR/vocab-regorg/>

ASK

WHERE

{comp:B org:hasRegisteredSite ?site .}

Résultat

FALSE

SPARQL Update

Peut être utilisé pour ...

- L'ajout de données (INSERT)
- La suppression de données (DELETE)
- Le chargement de graphe RDF (LOAD / LOAD .. INTO)
- L'effacement d'un graphe RDF (CLEAR GRAPH)
- Créer des graphes RDF (CREATE GRAPH)
- Retirer des graphes RDF (DROP GRAPH)
- Copier des graphes RDF (COPY GRAPH ... TO GRAPH)
- Déplacer des graphes RDF (MOVE GRAPH ... TO GRAPH)
- Ajouter des graphes RDF (ADD GRAPH TO GRAPH)

INSERT – ajouter un site agréé pour “BARCO”?

Données d'échantillon

```
comp:A rov:haslegalName "Niké" .  
comp:A org:hasRegisteredSite site:1234 .  
  
comp:B rov:haslegalName "BARCO" .  
  
site:1234 locn:fullAddress "Dahliastraat 24, 2160 Wommelgem" .
```

Requête

```
PREFIX comp: <http://example/org/>  
PREFIX org: <http://www.w3.org/TR/vocab-regorg/>  
  
INSERT DATA  
{  
  site:5678 locn:fullAddress "President Kennedypark 35, 8500 Kortrijk" .  
  comp:B org:hasRegisteredSite site:5678 .  
}
```

Résultat

```
comp:A rov:haslegalName "Niké" .  
comp:A org:hasRegisteredSite site:1234 .  
  
comp:B rov:haslegalName "BARCO" .  
comp:B org:hasRegisteredSite site:5678 .  
  
site:1234 locn:fullAddress "Dahliastraat 24, 2160 Wommelgem" .  
site:5678 locn:fullAddress "President Kennedypark 35, 8500 Kortrijk" .
```

INSERT/DELETE – changer l'adresse de “Niké”

Données d'échantillon

```
comp:A rov:haslegalName “Niké” .  
comp:A org:hasRegisteredSite site:1234 .  
  
comp:B rov:haslegalName “BARCO” .  
  
site:1234 locn:fullAddress “Dahliastraat 24, 2160 Wommelgem” .
```

Requête

```
PREFIX comp: <http://example/org/>  
PREFIX org: <http://www.w3.org/TR/vocab-regorg/>  
  
DELETE DATA  
{  
  comp:A org:hasRegisteredSite site:1234 .  
}  
  
INSERT DATA  
{  
  site:5678 locn:fullAddress “Rue de Loi 34, 1000 Bruxelles” .  
  comp:A org:hasRegisteredSite site:5678 .  
}
```

Résultat

```
comp:A rov:haslegalName “Niké” .  
comp:A org:hasRegisteredSite site:1000.  
  
site:1234 locn:fullAddress “Dahliastraat 24, 2160 Wommelgem” .  
site:1000 locn:fullAddress “Rue de Loi 34, 1000 Bruxelles” .
```

Sort in ascending order:

```
PREFIX e: <http://learningsparql.com/ns/expenses#>
```

```
SELECT ?description ?date ?amount
WHERE
{
  ?meal e:description ?description ;
        e:date ?date ;
        e:amount ?amount .
}
```

```
ORDER BY ?amount
```

Result Set:

description	date	amount
"breakfast"	"2011-10-15T08:32"	4.32
"breakfast"	"2011-10-14T08:53"	6.53
"breakfast"	"2011-10-16T09:05"	6.65
"lunch"	"2011-10-15T12:55"	9.45
"lunch"	"2011-10-16T13:24"	10.00
"lunch"	"2011-10-14T13:19"	11.13
"dinner"	"2011-10-16T19:44"	25.05
"dinner"	"2011-10-14T19:04"	28.30
"dinner"	"2011-10-15T18:54"	31.45

How to sort in descending order?

```
PREFIX e: <http://learningsparql.com/ns/expenses#>

SELECT ?description ?date ?amount
WHERE
{
    ?meal e:description ?description ;
          e:date ?date ;
          e:amount ?amount .
}

ORDER BY DESC(?amount)
```

Max et AVG

```
PREFIX e: <http://learningsparql.com/ns/expenses#>

SELECT (MAX(?amount) as ?maxAmount)
WHERE { ?meal e:amount ?amount . }
```

	maxAmount	
=====		
	31.45	

```
PREFIX e: <http://learningsparql.com/ns/expenses#>

SELECT (AVG(?amount) as ?avgAmount)
WHERE { ?meal e:amount ?amount . }
```

	avgAmount	
=====		
	14.764444444444444444444444444444	

Group By

```
PREFIX e: <http://learningsparql.com/ns/expenses#>

SELECT ?description (SUM(?amount) AS ?mealTotal)
WHERE
{
  ?meal e:description ?description ;
        e:amount ?amount .
}
GROUP BY ?description
```

description	mealTotal	
=====		
"dinner"	84.80	
"lunch"	30.58	
"breakfast"	17.50	

Having Function

```
PREFIX e: <http://learningsparql.com/ns/expenses#>

SELECT ?description (SUM(?amount) AS ?mealTotal)
WHERE
{
    ?meal e:description ?description ;
          e:amount ?amount .
}
GROUP BY ?description
HAVING (SUM(?amount) > 20)
```

description	mealTotal	
=====		
"dinner"	84.80	
"lunch"	30.58	

Merci

mallek.maha@gmail.com