Llenguatge de consulta SPARQL

PID_00271439

Blas Torregrosa García

Temps mínim de dedicació recomanat: 1 hora

Universitat Oberta de Catalunya

Blas Torregrosa García

Enginyer en Informàtica i màster universitari en Seguretat de les Tecnologies de la Informació i de les Comunicacions (MISTIC) per la Universitat Oberta de Catalunya (UOC). Especialitzat en ciberseguretat. Professor col·laborador del màster de Ciència de Dades de la UOC i professor associat a la Universitat de Valladolid (UVA).

L'encàrrec i la creació d'aquest recurs d'aprenentatge UOC han estat coordinats pel professor: Ferran Prados Carrasco (2020)

Primera edició: febrer 2020

Av. Tibidabo, 39-43, 08035 Barcelona
Realització editorial: FUOC

Índex

Introducció

El llenguatge de consulta semàntic SPARQL[1] permet la recuperació de dades mitjançant la programació. Igual que les bases de dades relacionals tenen el seu propi llenguatge de consulta (SQL o *Structured Query Language*), les tecnologies web semàntiques també tenen el seu propi llenguatge de consulta, que és el mecanisme que permet enviar consultes i obtenir resultats.

(1)Recursivament en anglès, *SPARQL Protocol and RDF Query Language* pronunciat /sparkel/.

El llenguatge es basa en dues especificacions: la de 2008 (**SPARQL 1.0**) i la de 2013 (**SPARQL 1.1**). Des de la versió 1.1, amb aquest llenguatge no solament es pot realitzar consultes sobre les dades, sinó també modificar i inserir dades RDF. En aquest mòdul solament tractarem la consulta de dades amb SPARQL.

Com utilitzar SPARQL

En aquest mòdul es mostra com utilitzar SPARQL com a llenguatge de consulta, similar a SQL, per als models de dades de l'ecosistema d'RDF.

Les característiques de SPARQL són, entre unes altres:

- **Extracció de dades** com a grafs RDF, URI, literals (tipades o no tipades), amb funcions d'agregació, subconsultes, etc.
- **Exploració de dades** mitjançant consultes per a relacions desconegudes.
- **Transformació de dades RDF** d'un vocabulari a un altre.
- **Construcció de nous grafs RDF** basats en grafs de consultes.
- **Actualitzacions de grafs RDF** com a llenguatge de manipulació de dades (*Data Manipulation Language*, DML) complet.
- **Vinculació lògica** per RDF, RDFS i OWL.
- **Consultes federades** distribuïdes en diferents punts d'accés (*Endpoint*) SPARQL.

1. Sintaxi bàsica

Les consultes SPARQL poden tenir diferents formes. La més freqüent és la consulta *Select* que obté informació basada en restriccions sobre les dades.

Cada consulta SELECT d'SPARQL SELECT s'organitza de la manera següent:

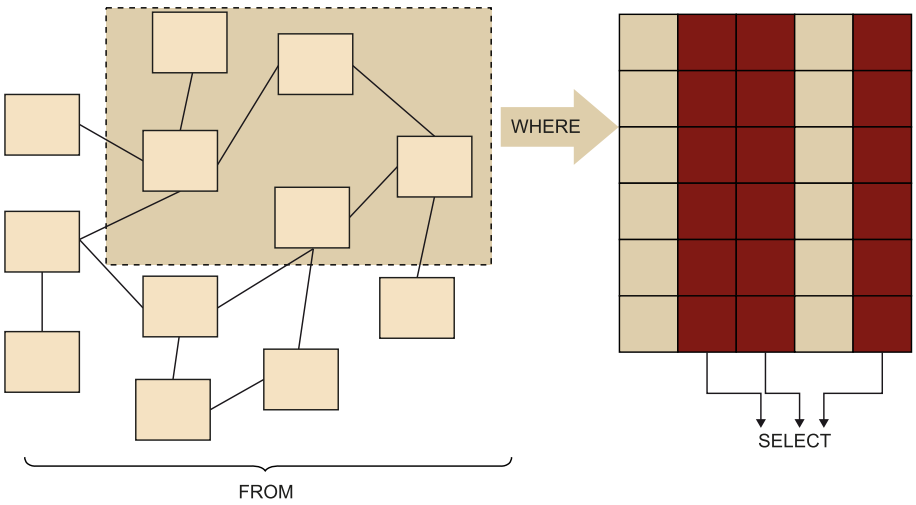
- **PREFIX** (prefixos dels espais de noms).
- **SELECT** (defineix el que es desitja recuperar).
- **FROM** (especifica el conjunt de dades del qual s'extreuen les dades).
- **WHERE** (criteris de restricció de les dades. És una descripció en forma de tripletes de consulta).
- **ORDER BY** (ordenació del resultat).
- **LIMIT** (modificació del resultat).

Figura 1. Consulta SPARQL que pregunta quan es va descobrir Plutó

```
BASE  <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>


SELECT  ?x
WHERE {
  <Pluto> dbo:discovered ?x.
}
```

Figura 2. Consultes SPARQL



1.1. Pròleg

El pròleg permet definir els espais de noms, és a dir, les abreviatures dels vocabularis que utilitzarem en la consulta. Mitjançant prefixos es poden afegir tants espais de noms com calgui. Per fer-ho, haurem d'utilitzar la paraula reservada «PREFIX».

Per exemple, podríem definir prefixos per als vocabularis de DBpedia i de Foaf de la forma següent:

Figura 3. Declaració de prefixos

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

Una vegada definits, podem referir-nos a les propietats dels vocabularis de la forma següent:

Figura 4. Ús de vocabularis declarats

```
dbpedia:Solar_System
foaf:name
```

El terme «BASE» es pot utilitzar solament una vegada en una consulta SPARQL i permet definir el vocabulari per defecte de la consulta. Per tant, se sobreentén que tots els URI que no utilitzin espais de noms pertanyeran al vocabulari base.

1.2. Select

Indica l'operació a executar (SELECT) i el resultat esperat de la consulta, és a dir, els valors que ha de retornar la consulta SPARQL. És un element obligatori d'una consulta.

De la mateixa manera que amb SQL, amb SPARQL es pot usar un asterisc (*) per expressar que volem que la consulta retorni totes les dades. També és possible indicar una llista d'expressions per determinar les dades desitjades. I igual que amb SQL es pot usar la clàusula «DISTINCT» per indicar que no s'han de retornar dades duplicades.

Encara que la consulta es realitza sobre un graf, la sortida de SELECT és una taula. CONSTRUCT permet obtenir els resultats en forma de graf RDF, en lloc de en forma tabular.

Figura 5. Consulta SPARQL que retorna les astronautes dones en forma de tripletes

```
CONSTRUCT  {?x foaf:gender ?g }
WHERE {
   ?x  rdf:type   dbo:Astronaut;
       foaf:name   ?nom ;
       foaf:gender ?g ..
   FILTER ((¿g = "female”@en) ).
}
```

Cal indicar a CONSTRUCT la plantilla de com volem obtenir els resultats, i aquesta plantilla és un conjunt de tripletes amb variables.

ASK permet executar una consulta per comprovar si una determinada condició es compleix en el conjunt de dades consultades. ASK retorna un valor booleà que indica si el patró de consulta se satisfà o no. No retorna el resultat.

DESCRIBE retorna un graf RDF que descriu un recurs RDF. El recurs es pot expressar mitjançant un URI o mitjançant una variable resultant d'un patró de consulta.

1.3. Cos

El cos de la consulta permetrà determinar quins elements del graf s'han de recuperar i en quin format (ordre i agrupació) hauran de ser lliurats. Per a això, el cos de la consulta consta de tres parts:

1) Origen de la consulta: permet definir el conjunt de dades que s'han de consultar. Fa referència al graf RDF (o a un fragment d'aquest) que serà l'origen de les dades. Per indicar el conjunt de dades que utilitzarem s'usen les paraules clau «FROM» (graf RDF per defecte) i «FROM NAMED» o «GRAPH» (graf RDF amb nom). Aquesta clàusula és opcional (els punts d'accés SPARQL solen tenir un conjunt de dades de referència i la clàusula FROM sol estar buida).

2) Patró de consulta: indica les condicions que han de complir les dades del graf per poder ser recuperades per la consulta. Permetrà seleccionar un conjunt de dades que satisfacin l'estructura i els valors indicats en el patró. El patró estarà contingut dins de la clàusula WHERE i pot ser tan complex com calgui, permetent conjuncions de patrons, disjuncions, parts opcionals variables i restriccions de valors.

3) Modificadors: són operacions que s'executaran sobre les dades seleccionades, sigui per canviar el seu ordre (la clàusula ORDER BY permet ordenar les dades d'acord amb un conjunt de propietats), el seu nivell d'agregació (la clàusula GROUP BY permet agrupar les dades d'acord amb una o més propietats), limitar el seu nombre (la clàusula LIMIT n permet limitar els resultats retornats als *n* primers) i saltar-se'n alguns (la clàusula OFSET i permet començar a mostrar les dades a partir de l'element *i*-èssim retornat per la consulta).

2. Patrons de consulta

Els patrons de consulta són els elements clau per entendre el funcionament de les consultes SPARQL.

> Un **patró de consulta** és una condició que han de satisfer les dades del graf RDF per poder ser seleccionades per la consulta. El patró més freqüent és el de les tripletes.

Un patró és una tripleta RDF (subjecte-predicat-objecte) en què un dels seus components o més són una variable. Les variables es representen mitjançant un símbol d'interrogació (?) i el nom de la variable.

Suposem que estem consultant DBpedia amb SPARQL utilitzant el patró de tripleta següent en el cos d'una consulta SPARQL:

Figura 6. Consulta SPARQL a la DBpedia

```
SELECT  ?satel·lits ?nom
WHERE {
   ?satel·lits  prop:satelliteOf dbpedia:Pluto .
   ?satel·lits rdfs:label ?nom.
   FILTER (lang(?nom) = "ca")
}
```

La consulta retornaria totes aquelles tripletes RDF de la DBpedia que compleixin amb el patró. Una tripleta complirà (*matching*) amb un patró si hi ha valors de la tripleta que apareguin en les dades consultades. En el cas de l'exemple, el patró prefixa l'objecte (recurs Plutó) i el predicat (satelliteOf). Per tant, totes les tripletes amb l'objecte Plutó i el predicat satelliteOf satisfarien el patró plantejat.

El resultat obtingut serà el següent:

Satèl·lits	Nom
http://dbpedia.org/resource/hydra_(moon)	"Hidra (satèl·lit)"@ca
http://dbpedia.org/resource/charon_(moon)	"Caront (satèl·lit)"@ca
http://dbpedia.org/resource/nix_(moon)	"Nix (satèl·lit)"@ca

Apareixeran tants resultats de noms com tripletes en el graf satisfacin la condició: hi ha els noms dels tres satèl·lits en tots els idiomes diferents. Per tant, s'ha afegit un filtre perquè solament mostri els que estan en català.

Com s'ha vist, en una consulta SPARQL es poden definir patrons simples (d'una sola tripleta) o conjunts de patrons (de més d'una tripleta). Un conjunt de patrons es compon de diferents patrons simples concatenats mitjançant un punt. En el nostre cas, tenim dos patrons:

1) El primer patró identificarà aquells recursos que compleixen que són satèl·lits de Plutó.

2) El segon patró identificarà els noms definits en la DBpedia per als noms dels satèl·lits en els diferents idiomes (filtrant solament els que estiguin en català).

Podríem complicar el patró afegint més patrons de tripletes. Les tripletes separades amb el punt (.) es comporten com l'operador `AND (I)`, és a dir, s'han de complir totes (forma conjuntiva).

2.1. Usant literals

Fins ara hem vist com realitzar consultes SPARQL filtrant les dades en funció de l'esquema que segueixen (en quines propietats participa cada recurs) i dels seus URI (amb quins recursos està relacionat). També resulta interessant poder filtrar dades en funció dels literals. Per exemple, identificar els recursos que continguin una certa cadena o que un valor numèric superi una quantitat. Per poder realitzar aquests filtres cal conèixer la representació dels diferents tipus de dades amb SPARQL.

La sintaxi general per a literals és una cadena de caràcters (entre cometes dobles " o simples ') i, opcionalment, el nom del seu tipus precedit de ^^. Per exemple:

- `"1"^^xsd:integer` per indicar que 1 és un nombre sencer.
- `"1"^^xsd:string` per indicar que és una cadena de caràcters amb l'1.
- `"3,1415"^^xsd:decimal` per indicar el número 3,1415.
- `"1.0e3"^^xsd:double` per indicar que és un nombre real.
- `"true"^^xsd:boolean` per indicar el valor *true* de tipus booleà.
- `"1930-02-18"^^xsd:date` per indicar que és una data.

Per afegir patrons que utilitzin comparacions amb possibles valors literals de les tripletes podem utilitzar la clàusula FILTER. Aquesta funció ens permet filtrar solament aquelles tripletes que satisfacin una determinada condició. La condició s'indica mitjançant un conjunt d'expressions booleanes, de la mateixa forma que amb SQL.

La clàusula FILTER funciona com un patró de tripleta més. Per tant, es pot utilitzar la clàusula FILTER diverses vegades en una mateixa consulta. FILTER no pot assignar ni crear nous valors.

Les expressions de la clàusula FILTER poden utilitzar diferents operadors. Els més comuns són:

- Els operadors booleans (! per representar `NOT` («no» lògic), && per representar un `AND` («i» lògic) i || per representar un `OR` («o» lògic).
- Els operadors de comparació (=, !=, >=, <, <=, > i >=).
- Els operadors matemàtics (+, -, *, /, %).

Les cadenes de caràcters amb RDF poden tenir associat un idioma. Això permet definir una mateixa propietat en diferents idiomes. Per exemple, el nom del planeta nan Plutó és **Pluto** en anglès o *Plutón* en castellà. Per indicar això, amb RDF s'afegeix un sufix a la cadena de caràcters que conté una @ i el codi de l'idioma. Així, si consultem el nom de Plutó en la DBpedia, tindríem «"Plutón"@es», «"Pluto"@en» o «冥王星@ja» entre altres valors. Els tres valors representen el nom del planeta nan en castellà, anglès i japonès, respectivament.

Per filtrar per cadenes de caràcters es pot utilitzar la clàusula FILTER en combinació amb la funció regex. La funció *regex* permet definir l'expressió regular que els valors de la variable hauran de satisfer (d'una manera molt semblant al LIKE amb SQL).

Funció regex

Més informació sobre el funcionament de regex a www.w3.org/tr/xpath-functions/#regex-syntax.

La funció regex permet utilitzar caràcters comodí, com per exemple el ^ per indicar l'inici d'una cadena de caràcters, el $ per indicar el final i el * per indicar una cadena de zero, un o més caràcters.

A més de la funció regex hi ha altres funcions útils per filtrar els valors:

Filtrar valors

La llista completa de les funcions útils per filtrar valors es pot trobar a: www.w3.org/tr/sparql11-query/#SparqlOps.

- `datatype`: retorna el tipus de dades d'un element,
- `str`: converteix a text un literal,
- `isUri`, indica si un recurs és un URI,
- `lang`, indica l'idioma associat a una cadena de text,
- `bound`, indica si el literal té assignat un valor.

2.2. Patrons opcionals

Les consultes vistes fins ara exigeixen que se satisfacin tots els patrons per obtenir els resultats. En alguns casos això pot ser massa restrictiu. Per resoldre aquest problema, hi ha els **patrons opcionals** que es defineixen mitjançant una clàusula OPTIONAL.

Tant les tripletes que satisfacin el patró opcional com les que no ho facin se seleccionaran en la consulta. Per tant, els resultats de la cerca seran aquells en què es compleix el patró opcional, però també les dades en què no es compleixi. Les variables lligades al patró opcional no tindran valor per les tripletes en què el patró no s'ha complert.

Figura 7. Consulta SPARQL sobre els membres de la missió Apol·lo 11

```
SELECT  ?nom ?fn ?fd
WHERE {
   dbp:Apollo_11 prop:crewMembers ?x .

  ?x rdf:type dbo:Astronaut ;
     foaf:name ?nom ;
     dbo:birthDate ?fn .

   OPTIONAL { ?x dbo:deathDate ?fd }
}
```

A continuació podem veure el resultat de la consulta anterior, que ara retorna els tres membres de la tripulació. Per a alguns, la data de defunció està buida, ja que, si escau, el patró de la tripleta amb el qual es calculava no se satisfà.

Figura 8. Resultat de la consulta

Nom	Data Naixement	Data Defunció
"Neil Armstrong"@en	1930-08-05	2012-08-25
"Buzz Aldrin"@en	1930-01-20	
"Michael Collins"@en	1930-10-31	

2.3. Patrons alternatius

Fins ara hem vist com utilitzar combinacions de patrons de tripletes de manera que tots els patrons es compleixin alhora. Algunes vegades caldrà utilitzar patrons de manera que es garanteixi que se satisfaci un patró o un altre, com en una OR lògica.

Aquest tipus de patrons es denominen *patrons alternatius*. Els **patrons alternatius** s'expressen emmarcant els dos patrons disjunts entre els símbols { } i unint-los mitjançant la clàusula UNION.

Figura 9. Consulta SPARQL amb la unió de les tripulacions de les missions Apol·lo 11 i Apol·lo 13

```
SELECT  ?nom ?fn ?fd
WHERE {
  {
      dbp:Apollo_11 prop:crewMembers ?x .
      ?x rdf:type dbo:Astronaut ;
         foaf:name ?nom ;
         dbo:birthDate ?fn .
      OPTIONAL { ?x dbo:deathDate ?fd }
   }
  UNION
   {
      dbp:Apollo_13 prop:crewList ?x .
      ?x rdf:type dbo:Astronaut ;
         foaf:name ?nom ;
         dbo:birthDate ?fn .
      OPTIONAL { ?x dbo:deathDate ?fd }
   }
}
```

Per integrar en una consulta tots dos conjunts de patrons de forma alternativa, s'engloba cada patró entre les marques { } i s'intercala la clàusula UNION al mig. El resultat de la consulta anterior SPARQL és:

Figura 10. Resultat de la consulta

Nom	Data Naixement	Data Defunció
"Neil Armstrong"@en	1930-08-05	2012-08-25
"Buzz Aldrin"@en	1930-01-20	
"Michael Collins"@en	1930-10-31	
"Fred Haise"@en	1933-11-14	
"James Lovell"@en	1928-03-25	
"Jack Swigert"@en	1931-08-30	1982-12-27

3. Punts d'accés SPARQL

Per poder executar consultes SPARQL necessitem un punt d'accés.[2] Aquests punts d'accés serien l'equivalent a la consola d'un sistema gestor de bases de dades. Hi ha diferents punts d'accés disponibles des d'internet que ens permeten consultar conjunts de dades RDF localitzades.

(2) *Endpoint*, en anglès.

Des de la versió SPARQL 1.1, els punts d'accés permeten obtenir els resultats d'una consulta en diferents formats (XML, JSON, CSV, etc.), o crear un nou graf RDF com a resultat d'una consulta. Amb SPARQL també es poden consultar dades RDF de més d'un conjunt de dades mitjançant el que es denominen **consultes federades** (*Federated Query*).

Alguns dels punts d'accés més populars són els següents:

Punts d'accés

La llista dels punts d'accés més rellevants es pot trobar a www.w3.org/wiki/sparqlendpoints.

1) Per consultar dades de caràcter general:

- DBpedia permet accedir a les seves dades.
- Wikidata permet accedir a les dades de www.wikidata.org.

2) Per consultar dades geogràfiques:

- GeoNames permet accedir a les dades geogràfiques disponibles a GeoNames.
- LinkedGeoData permet accedir a les dades geogràfiques de la web d'OpenStreetMaps.

Bibliografia

DuCharme, R. (2013). *Learning SPARQL* (2a. ed.). O'Reilly Media.

Guarino, N.; Oberle, D.; Staab, S. (2009). *What Is an Ontology?* [en línia]. [Data de consulta: gener 2020]. Disponible a: <https://iaoa.org/isc2012/docs/Guarino2009_What_is_an_Ontology.pdf>

Kumar, A. (2018). *Architecting Data-Intensive Applications*. Packt Pub.

Noy, N. F.; McGuinness, D. L. (2001). *Ontology development 101: A guide to creating your first ontology*. Stanford knowledge systems laboratory technical report (Informe SMI-2001-0880).

Powers, S. (2003). *Practical RDF*. O'Reilly Media.