

FUNDAMENTOS DE LA WEB SEMANTICA – 2023 UNIDAD 5: SPARQL

Edelweis Rohrer InCo- Fing - Universidad de la República

Parte II - Integración de grafos locales y remotos

Grafos nombrados

En la Parte I estudiamos conceptos básicos del lenguaje SPARQL, y en particular las posibilidades del lenguaje para obtener información en el contexto abierto de la web semántica, donde la información no siempre está completa, o encontramos la misma información en varios sitios con diferente nivel de granularidad. Vimos varios ejemplos de consultas, que nos dieron una idea de la capacidad de SPARQL para navegar por los grafos RDF y devolver únicamente los datos que satisfacen las condiciones de la consulta. Vimos que el lenguaje permite presentar los resultados tal como se encuentran en los grafos, agrupados, aplicando diferentes operaciones, y además serializados en diferentes formatos. En los ejemplos presentados, no se incluye la palabra FROM, lo que significa que estamos consultando el **grafo por defecto** de la base de datos RDF. Por ejemplo, la siguiente consulta formulada en el endpoint de DBPEDIA, devuelve los datos que recupera del grafo por defecto: <http://dbpedia.org>.

```
select ?lugarNac ?pais
where
{
  dbr:María_Noel_Ricchetto dbo:birthPlace ?lugarNac .
  ?lugarNac dbo:country ?pais .
}
```

Un **RDF Dataset** es un **conjunto de grafos RDF** que tiene un **grafo por defecto**, y cero o más **grafos nombrados** (*named graphs*), cada uno de ellos **identificados por una IRI**.

En particular, el grafo por defecto de DBPEDIA incluye un conjunto de subgrafos con todo el vocabulario e instancias de DBPEDIA, por ejemplo, todo el vocabulario con prefijo <http://dbpedia.org/ontology/>. Además, el endpoint de DBPEDIA accede en forma local a grafos identificados por IRIs, por ejemplo: a <http://www.w3.org/2002/07/owl#>, que contiene todo el vocabulario del lenguaje de ontologías OWL, y a <http://dbpedia.org/resource/classes#>, que contiene el conjunto de clases y propiedades del repositorio DBPEDIA, entre otros.

La siguiente consulta recupera el conjunto de recursos que son instancias de las clases del grafo nombrado <http://dbpedia.org/resource/classes#>. La palabra “graph” en el where indica que la evaluación de las condiciones entre llaves a continuación de la IRI se realiza en el grafo <http://dbpedia.org/resource/classes#>.

```

select ?recurso ?nombre ?clase
FROM named <http://dbpedia.org/resource/classes#> ➡ Grafo nombrado
where
{
  ?recurso rdf:type ?clase . ➡ La consulta se evalúa en el grafo por defecto
  ?recurso rdfs:label ?nombre .
  {
    graph <http://dbpedia.org/resource/classes#>
      {?clase rdf:type owl:Class .} ➡ La consulta se evalúa el grafo nombrado
    }
  }
}

```

Como resultado se obtiene la lista de recursos, su nombre y la clase a que pertenece, del grafo por defecto, y al consultar el grafo <http://dbpedia.org/resource/classes#> se impone la condición de que la clase pertenece a este grafo. La siguiente figura muestra el resultado.

recurso	nombre	clase
http://dbpedia.org/resource/OpenLink_Software	"OpenLink Software"@en	http://dbpedia.org/ontology/Company
http://dbpedia.org/resource/OpenLink_Software	"OpenLink Software"@nl	http://dbpedia.org/ontology/Company
http://dbpedia.org/resource/2015_Games_ILC	"2015 2015"@ar	http://dbpedia.org/ontology/Company
http://dbpedia.org/resource/2015_Games_ILC	"2015, Inc."@ca	http://dbpedia.org/ontology/Company
http://dbpedia.org/resource/2015_Games_ILC	"2015 Games"@de	http://dbpedia.org/ontology/Company
http://dbpedia.org/resource/2015_Games_ILC	"2015 Games, LLC."@en	http://dbpedia.org/ontology/Company
http://dbpedia.org/resource/2015_Games_ILC	"2015 Inc"@fr	http://dbpedia.org/ontology/Company
http://dbpedia.org/resource/2015_Games_ILC	"2015 (企業)"@ja	http://dbpedia.org/ontology/Company
http://dbpedia.org/resource/2015_Games_ILC	"2015 (기업)"@ko	http://dbpedia.org/ontology/Company
http://dbpedia.org/resource/2015_Games_ILC	"2015, Inc."@nl	http://dbpedia.org/ontology/Company
http://dbpedia.org/resource/2015_Games_ILC	"2015, Inc."@pt	http://dbpedia.org/ontology/Company
http://dbpedia.org/resource/2015_Games_ILC	"2015, Inc."@ru	http://dbpedia.org/ontology/Company
http://dbpedia.org/resource/5_Rue_Christine	"5 Rue Christine"@en	http://dbpedia.org/ontology/Company
http://dbpedia.org/resource/5_Rue_Christine	"5 Rue Christine"@nl	http://dbpedia.org/ontology/Company
http://dbpedia.org/resource/5th_Cell	"5th Cell"@de	http://dbpedia.org/ontology/Company
http://dbpedia.org/resource/5th_Cell	"5th Cell"@en	http://dbpedia.org/ontology/Company
http://dbpedia.org/resource/5th_Cell	"5th Cell"@fr	http://dbpedia.org/ontology/Company
http://dbpedia.org/resource/606_Films	"606 Films"@en	http://dbpedia.org/ontology/Company
http://dbpedia.org/resource/6126_(clothing_line)	"6126 (clothing line)"@en	http://dbpedia.org/ontology/Company

También es posible **cambiar el grafo por defecto** en una consulta, es decir, para esa consulta se declara que el grafo por defecto es otro grafo. La siguiente consulta recupera las propiedades del grafo <http://dbpedia.org/resource/classes#> que tienen propiedades equivalentes en el grafo http://dbpedia.org/schema/property_rules#, su nombre y su dominio.

```

select ?prop ?nombre ?dominio ?propEquiv
FROM <http://dbpedia.org/resource/classes#> ➡ Grafo por defecto en esta consulta
FROM named <http://dbpedia.org/schema/property_rules#>
where
{
  ?prop rdf:type rdf:Property .
  ?prop rdfs:label ?nombre .
  ?prop rdfs:domain ?dominio .
  graph <http://dbpedia.org/schema/property_rules#>
  {
    ?prop owl:equivalentProperty ?propEquiv .
  }
}

```

El resultado es el que se muestra en la figura siguiente:

prop	nombre	dominio	propEquiv
http://dbpedia.org/ontology/deathPlace	"death place"@en	http://dbpedia.org/ontology/Person	http://dbpedia.org/property/deathPlace
http://dbpedia.org/ontology/deathPlace	"death place"@en	http://dbpedia.org/ontology/Person	http://dbpedia.org/property/deathPlace
http://dbpedia.org/ontology/deathPlace	"Sterbeort"@de	http://dbpedia.org/ontology/Person	http://dbpedia.org/property/deathPlace
http://dbpedia.org/ontology/deathPlace	"Sterbeort"@de	http://dbpedia.org/ontology/Person	http://dbpedia.org/property/deathPlace
http://dbpedia.org/ontology/deathPlace	"lieu de décès"@fr	http://dbpedia.org/ontology/Person	http://dbpedia.org/property/deathPlace
http://dbpedia.org/ontology/deathPlace	"lieu de décès"@fr	http://dbpedia.org/ontology/Person	http://dbpedia.org/property/deathPlace
http://dbpedia.org/ontology/deathPlace	"plaats van overlijden"@nl	http://dbpedia.org/ontology/Person	http://dbpedia.org/property/deathPlace
http://dbpedia.org/ontology/deathPlace	"plaats van overlijden"@nl	http://dbpedia.org/ontology/Person	http://dbpedia.org/property/deathPlace
http://dbpedia.org/ontology/deathPlace	"τόπος θανάτου"@el	http://dbpedia.org/ontology/Person	http://dbpedia.org/property/deathPlace
http://dbpedia.org/ontology/deathPlace	"τόπος θανάτου"@el	http://dbpedia.org/ontology/Person	http://dbpedia.org/property/deathPlace
http://dbpedia.org/ontology/deathPlace	"死没地"@ja	http://dbpedia.org/ontology/Person	http://dbpedia.org/property/deathPlace
http://dbpedia.org/ontology/deathPlace	"死没地"@ja	http://dbpedia.org/ontology/Person	http://dbpedia.org/property/deathPlace
http://dbpedia.org/ontology/deathPlace	"death place"@en	http://dbpedia.org/ontology/Person	http://dbpedia.org/property/deathPlace
http://dbpedia.org/ontology/deathPlace	"death place"@en	http://dbpedia.org/ontology/Person	http://dbpedia.org/property/deathPlace
http://dbpedia.org/ontology/deathPlace	"Sterbeort"@de	http://dbpedia.org/ontology/Person	http://dbpedia.org/property/deathPlace

En una consulta es posible incluir más de un grafo nombrado. El lenguaje permite también usar una variable en el where para indicar que las condiciones se evalúen para cada uno de los grafos nombrados. La siguiente consulta recupera las propiedades del grafo <http://dbpedia.org/resource/classes#> con su dominio, tales que tienen propiedades equivalentes en el grafo http://dbpedia.org/schema/property_rules#, y que no existen como propiedades en http://dbpedia.org/schema/property_rules#.

```

select ?prop ?dominio ?propEquiv
FROM named <http://dbpedia.org/resource/classes#>
FROM named <http://dbpedia.org/schema/property_rules#>
where
{
  graph ?g1
  {
    ?prop rdf:type rdf:Property .
    ?prop rdfs:domain ?dominio .
  }
  graph ?g2
  {
    ?prop owl:equivalentProperty ?propEquiv .
    FILTER not exists {?prop rdf:type rdf:Property .}
  }
}

```

Se obtiene el siguiente resultado:

prop	dominio	propEquiv
http://dbpedia.org/ontology/deathPlace	http://dbpedia.org/ontology/Person	http://dbpedia.org/property/deathPlace
http://dbpedia.org/ontology/deathDate	http://dbpedia.org/ontology/Person	http://dbpedia.org/property/death
http://dbpedia.org/ontology/birthPlace	http://dbpedia.org/ontology/Person	http://dbpedia.org/property/birthPlace
http://dbpedia.org/ontology/birthDate	http://dbpedia.org/ontology/Person	http://dbpedia.org/property/birth

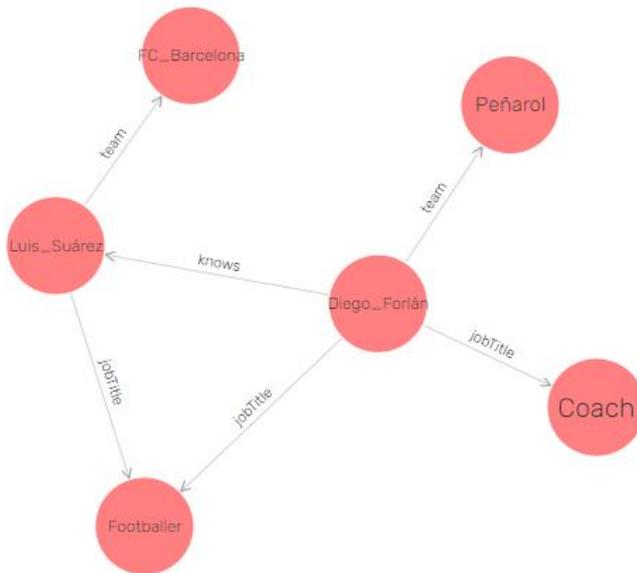
*Puedes ejecutar las consultas de ejemplo de este material accediendo al endpoint de DBPEDIA en <http://dbpedia.org/sparql>.

Para profundizar más en este tema puedes consultar [1].

*Antes de continuar con la lectura de esta parte de la unidad puedes hacer la **actividad individual 5** para fijar los conceptos hasta este punto. *

SERVICE para acceder a grafos remotos

En la actividad individual 5 de la Unidad Modelos Semánticos RDF(S) construimos y subimos al endpoint GraphDB el grafo que se ilustra a continuación:



Si para cada jugador, queremos conocer el equipo, la profesión y si conoce a otro jugador, podemos formular la siguiente consulta en el editor de consultas SPARQL de GraphDB:

```
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX schema: <http://schema.org/>
select ?jugador ?equipo ?profesion ?conocido
where {
    ?jugador dbo:team ?equipo .
    ?jugador schema:jobTitle ?profesion .
    OPTIONAL{?jugador schema:knows ?conocido .}
}
```

El resultado de la consulta es el que aparece en la figura siguiente:

jugador	equipo	profesion	conocido
dbr:Luis_Suárez	dbr:FC_Barcelona	dbr:Footballer	
dbr:Diego_Forlán	dbr:Peñarol	dbr:Footballer	dbr:Luis_Suárez
dbr:Diego_Forlán	dbr:Peñarol	dbr:Coach	dbr:Luis_Suárez

Si queremos agregar el lugar de nacimiento de cada jugador, que sabemos que lo encontramos en la DBPEDIA, podemos ampliar la consulta en el endpoint de GraphDB, para obtener además el valor que toma la propiedad *dbo:birthPlace* de la siguiente manera:

```

PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX schema: <http://schema.org/>
select ?jugador ?equipo ?profesion ?lugarNac ?conocido
where {
    ?jugador dbo:team ?equipo .
    ?jugador schema:jobTitle ?profesion .
    OPTIONAL{?jugador schema:knows ?conocido .}
    OPTIONAL{?jugador dbo:birthPlace ?lugarNac .}
}

```

➡ de DBPEDIA

Sin embargo, la consulta nos devuelve lo que aparece en la siguiente figura:

jugador	equipo	profesion	lugarNac	conocido
dbr:Luis_Suárez	dbr:FC_Barcelona	dbr:Footballer		
dbr:Diego_Forlán	dbr:Peñarol	dbr:Footballer		dbr:Luis_Suárez
dbr:Diego_Forlán	dbr:Peñarol	dbr:Coach		dbr:Luis_Suárez

No se obtiene el lugar de nacimiento de Suárez ni de Forlán (que existe en DBPEDIA) porque es un dato que está en un grafo de un endpoint remoto (DBPEDIA).

Para consultar información de RDF datasets remotos, integrándola con los datos de datasets locales, usamos el constructor **SERVICE**. Para el ejemplo anterior, debemos formular la consulta de la siguiente manera:

```

PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX schema: <http://schema.org/>
select ?jugador ?equipo ?profesion ?lugarNac ?conocido
where {
    ?jugador dbo:team ?equipo .
    ?jugador schema:jobTitle ?profesion .
    OPTIONAL{?jugador schema:knows ?conocido .}
    SERVICE <https://dbpedia.org/sparql>
        {?jugador dbo:birthPlace ?lugarNac . }
}

```

➡ endpoint remoto (DBPEDIA)

Ahora sí se obtiene el lugar de nacimiento de cada jugador, como se muestra en la figura siguiente:

	jugador	equipo	profesion	lugarNac	conocido
1	dbr:Diego_Forlán	dbr:Peñarol	dbr:Coach	dbr:Montevideo	dbr:Luis_Suárez
2	dbr:Diego_Forlán	dbr:Peñarol	dbr:Footballer	dbr:Montevideo	dbr:Luis_Suárez
3	dbr:Luis_Suárez	dbr:FC_Barcelona	dbr:Footballer	dbr:Salto_Uruguay	

La extensión **SERVICE** de SPARQL 1.1. permite **ejecutar parte de una consulta SPARQL en un endpoint remoto**. Los resultados son retornados al procesador de la consulta y son combinados con los resultados del resto de la consulta.

Puedes ejecutar las consultas de ejemplo de este material accediendo al endpoint de GraphDB.

Para profundizar más en este tema puedes consultar [2].

Referencias:

[1] SPARQL 1.1 Query Language - W3C Recommendation 21 March 2013.
<https://www.w3.org/TR/sparql11-query/>

[2] SPARQL 1.1 Federated Query - W3C Recommendation 21 March 2013.
<https://www.w3.org/TR/2013/REC-sparql11-federated-query-20130321/>