

FUNDAMENTOS DE LA WEB SEMANTICA – 2023 UNIDAD 5: SPARQL

Edelweis Rohrer InCo- Fing - Universidad de la República

Parte I: Conceptos básicos, constructores

El patrón de consulta

Un **patrón de consulta básico** es una *conjunción de triplas* donde *sujeto, predicado y objeto pueden ser una variable*.



Para el grafo de la figura anterior, si queremos obtener el lugar de nacimiento de María Noel Riccetto y país al que pertenece, el patrón de consulta es:

```
{
  dbr:María_Noel_Riccetto dbo:birthPlace ?lugarNac .
  ?lugarNac dbo:country ?pais .
}
```

Variables

Formalmente:

- I**: conjunto de IRIs
 - B**: conjunto de nodos blancos
 - L**: conjunto de literales
 - V**: conjunto de variables
- } disjuntos

Triplas $(s, p, o) \in (I \cup B) \times I \times (I \cup B \cup L)$

Triple pattern $t \in (I \cup B \cup V) \times (I \cup V) \times (I \cup B \cup L \cup V)$

Patrón de consulta básico (Basic graph pattern):

$BGP = \{t \in \mid t (I \cup B \cup V) \times (I \cup V) \times (I \cup B \cup L \cup V)\}$

En resumen:

- Todas las consultas SPARQL se basan en un **patrón de consulta**
- Recuperan los recursos que cumplen con las **condiciones** del patrón de consulta.
- Un patrón de consulta básico se corresponde con un **subgrafo** de datos RDF cuando los **recursos del subgrafo pueden ser sustituidos en las variables** del patrón. Dicho subgrafo es el resultado de la consulta.
- El resultado puede ser serializado en diferentes formatos.

Existen diferentes **tipos de consultas SPARQL**:

- **SELECT**: devuelve el conjunto de datos RDF que coinciden con un patrón de consulta.
- **CONSTRUCT**: devuelve un nuevo grafo RDF a partir de datos RDF que coinciden con un patrón de consulta.
- **ASK**: devuelve si existen datos RDF coincidentes con un patrón de consulta.

- **DESCRIBE:** devuelve un grafo RDF que describe los recursos que cumplen con un patrón de consulta.

La **sintaxis general de una consulta SPARQL** es:

Prefijos	
SELECT.....	Descripción del resultado: qué información se quiere obtener
FROM....	Grafo(s) RDF: dónde buscar
WHERE.....	Patrón de la consulta: qué buscar
ORDER BY....	Modificadores de la consulta: filtrar y organizar los resultados

Por ejemplo:

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

En este ejemplo se omite el *from* porque se está consultando el grafo por defecto de la base RDF (el concepto de “grafo por defecto” se explica más adelante en este documento).

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

Patrones complejos, filtros, modificadores y otros tipos de consulta

Con el patrón de consulta básico el resultado de la consulta debe satisfacer TODAS las condiciones. Sin embargo, **la información en la web está incompleta**, por lo que a veces es necesario recuperar recursos aunque no todas las propiedades tengan un valor asociado.

Los **patrones de consulta complejos** permiten introducir variantes:

- **OPTIONAL:** recupera valores de propiedades en caso que existan, pero igualmente se incluye el recurso (que no posee valor en la propiedad) en el resultado.
- **UNION:** recupera recursos que tienen valores para al menos una de un conjunto de condiciones (por ejemplo, tienen valores para alguna propiedad de un conjunto de propiedades).

Los **filtros** restringen los resultados de una consulta y los **modificadores** permiten organizar los resultados.

Puedes consultar la sintaxis de los patrones de consulta complejos, filtros y modificadores en [1].

La siguiente consulta recupera todas las ciudades de Uruguay y, para las que tienen entre de 20.000 y 500.000 habitantes recupera además su población.

```
SELECT ?x ?population
WHERE
{
  ?x rdf:type dbo:City.
  ?x dbo:country dbr:Uruguay.
  OPTIONAL {?x dbo:populationTotal ?population.
    FILTER (?population > 20000 && ?population < 500000)}
}
```

Existen otros **tipos de consulta diferentes de SELECT**, que son **CONSTRUCT, ASK y DESCRIBE**.

La siguiente consulta construye un grafo que asocia a cada persona su nombre a través de la propiedad foaf:name.

```
CONSTRUCT
{ ?persona foaf:name ?nombre }
WHERE
{
  ?persona rdf:type dbo:Person .
  ?persona dbo:birthName ?nombre.
}
```

** Puedes consultar la sintaxis del Construct, Ask y Describe en [1].**

Antes de continuar con la lectura de esta parte de la unidad puedes hacer las **actividades individuales 1 y 2 para fijar los conceptos hasta este punto. **

Extensión SPARQL 1.1: subconsultas, agregaciones y otros

El lenguaje SPARQL es extendido en 2013 con nuevas funcionalidades, como **expresiones en el SELECT, bind values, agregaciones, subconsultas, property paths y negación**.

** Puedes consultar la sintaxis de los nuevos constructores de la extensión 1.1 en [1].**

La siguiente consulta es un ejemplo de uso de agregaciones, recupera la cantidad de bailarinas y bailarines por lugar de nacimiento.

```
select ?lugarNac (COUNT(?bailarina) as ?cantBailarinas)
where
{
  ?bailarina rdf:type dbo:Artist .
  ?bailarina dbo:birthPlace ?lugarNac .
  {
    {?bailarina dbo:field dbr:Ballet} UNION
    {?bailarina rdf:type yago:BalletDancer109834699} UNION
    {?bailarina rdf:type yago:Ballerina109834592}
  }
}
group by ?lugarNac
```

La siguiente consulta es un ejemplo de uso de subconsultas, recupera la cantidad de departamentos de Uruguay con más de 200.000 habitantes. La subconsulta suma la cantidad de población de cada localidad agrupando por departamento y se queda con aquellos que tienen más de 200.000 habitantes (esta subconsulta realiza una agregación) y la consulta cuenta los departamentos.

```
select (COUNT(?depto) as ?cantidad)
where
{
  select ?depto (SUM(?pop) as ?popdepto)
  where
    {
      ?place dbo:country dbr:Uruguay.
      ?place dbo:populationTotal ?pop.
      ?place dbo:subdivision ?depto.
    }
  group by ?depto
  having (SUM(?pop) > 200000)
}
```

Antes de continuar con la lectura de la Parte II de la unidad puedes hacer las **actividades individuales 3 y 4 para fijar los conceptos hasta este punto. **

Referencia:

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