SPARQL: Linguagem de Consulta em Ontologias

I Escola de Ontologias UFAL-USP

Endhe Elias e Olavo Holanda

Núcleo de Excelência em Tecnologias Sociais - NEES
Universidade Federal de Alagoas – UFAL

Por que SPARQL?

SPARQL (Simple Protocol and RDF Query Language) é a linguagem de consulta da web semântica. Ele nos permite:

- Recuperar valores de dados estruturados e semiestruturados;
- Explorar dados ao consultar relações desconhecidas;
- Realizar uniões complexas de conjuntos de dados diferentes em uma única e simples consultas.

Padrão SPARQL

SPARQL 1.0 se tornou um padrão em janeiro, 2008:

- SPARQL 1.0 Query Language – para consultar padrões em dados RDF
- SPARQL 1.0 Protocol – para enviar consultas por HTTP
- SPARQL Results XML Format – formato dos resultados em XML
- SPARQL Results JSON Format – formato dos resultados em JSON

SPARQL 1.1 se tornou um padrão em Março, 2013:

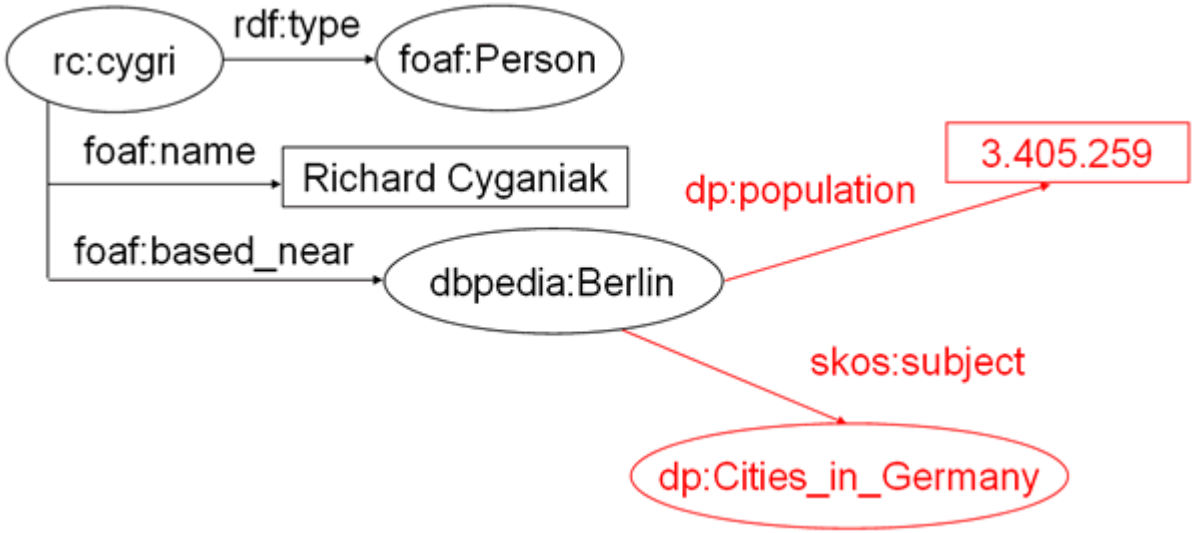
- Versões atualizadas 1.1 da consulta SPARQL e do Protocolo SPARQL
- SPARQL 1.1 Update – para inserção, remoção e modificação de dados RDF
- SPARQL 1.1 Graph Store HTTP Protocol – acesso RESTful de grafos RDF
- SPARQL 1.1 Service Descriptions – descrição de endpoints SPARQL
- SPARQL 1.1 Entailments – como combinar inferência com SPARQL
- SPARQL 1.1 Basic Federated Query - consultando vários endpoints de uma vez
- SPARQL Results CSV/TSV Formats – formato dos resultados em CSV/TSV

Conceitos

RDF é um modelo de dados de grafos baseado em triplas de sujeito, predicado e objeto:

- Sujeitos, predicados e objetos são representados por URIs, que podem ser abreviadas com o uso de prefixos (prefixed names);
- Objetos podem também ser literais: strings, inteiros, booleanos, etc.
- Sintaxe:
  - URIs: <http://example.com/resource> ou prefixo:nome
  - Literais: "string pura" "13.4"^^xsd:float ou "string com linguagem"@pt
  - Tripla: prefixo:sujeito outro_prefixo:predicado "objeto"

Modelo RDF


rc:cygri
rdf:type
foaf:Person
foaf:name
Richard Cyganiak
foaf:based_near
dbpedia:Berlin
dp:population
3.405.259
skos:subject
dp:Cities_in_Germany

Estrutura de uma Consulta SPARQL

Uma consulta SPARQL é composta, em ordem, por:

- Declarações de prefixos, para abreviar URIs
- Definição do conjunto de dados, informando quais grafo(s) RDF estão sendo consultados
- A cláusula de resultado, identificando que informação deve ser retornada a partir da consulta
- O padrão de consulta, especificando o que consultar dentro do conjunto de dados
- Modificadores de consulta, limites, ordenação, e outros que podem modificar o resultado final

```
# declarações de prefixo
PREFIX foo: <http://example.com/resources/>
...
# definição de conjunto de dados
FROM ...
# cláusula de resultado
SELECT ...
# padrão de consulta
WHERE {
   ...
}
# modificadores de consulta
ORDER BY ...
```

Exemplos

Ontologia Viagem:

- Consultar todos os voos da ontologia:

```
SELECT ?voo
WHERE{ ?voo rdf:type :Voo}
```

- rdf:type significa “é um tipo de”;

- Seleciona todas as instâncias onde essas instâncias são um tipo de Voo (Classe Voo da ontologia).

Exemplos

- Consultar todas as passagens:

```
SELECT ?passagem
WHERE{ ?passagem rdf:type :Passagem}
```

- Retorna na variável todas as instâncias pertencentes a classe Passagem.

Exemplos

- Exemplo com duas variáveis;

- Mostrar todas as passagens e seus respectivos preços:

```
SELECT ?passagem ?preco
WHERE{ ?passagem rdf:type :Passagem.
       ?passagem :preco ?preco}
```

Exemplos

- Podemos também ordenar o resultado: cláusula ORDER BY;

- Seleciona todas as passagens e preços ordenando pelo preço da passagem:

```
SELECT ?passagem ?preco
WHERE{ ?passagem rdf:type :Passagem.
                    ?passagem :preco ?preco}
ORDER BY ?preco
```

Exemplo Complexo

- Mostrar o preço e a passagem do voo que tem origem em Brasília e destino Guarulhus, pela companhia aérea GOL.

```
SELECT ?passagem ?preco
WHERE{ ?passagem rdf:type :Passagem.
       ?passagem :preco ?preco.
       ?passagem :relacionado_ao_voo ?voo.
       ?voo rdf:type :Voo.
       ?voo :aeroporto_origem <http://nees.com.br/tut/Viagem.owl#Aeroporto_Brasilia>.
       ?voo :aeroporto_destino <http://nees.com.br/tut/Viagem.owl#Aeroporto_Guarulhos>.
       ?voo :gerenciado_por_companhiaAerea
    <http://nees.com.br/tut/Viagem.owl#Companhia_Aerea_GOL>}
```

Praticar

- Selecionar todas as companhias aéreas;
- Selecionar todos os aeroportos;
- Selecionar todos os voos da GOL;
- Selecionar passagem e preço dos voos com origem em Salvador;
- Selecionar todas as passagens e preços da TAM ordenando pelo preço;
- Selecionar todos os voos com duração de 90.

Praticar

- Selecionar todas as companhias aéreas;
- Selecionar todos os aeroportos;
- Selecionar todos os voos da GOL;
- Selecionar passagem e preço dos voos com origem em Salvador;
- Selecionar todas as passagens e preços da TAM ordenando pelo preço;
- Selecionar todos os voos com duração de 90.

Praticar

- Selecionar todas as companhias aéreas;
- Selecionar todos os aeroportos;
- Selecionar todos os voos da GOL;
- Selecionar passagem e preço dos voos com origem em Salvador;
- Selecionar todas as passagens e preços da TAM ordenando pelo preço;
- Selecionar todos os voos com duração de 90.

Praticar

- Selecionar todas as companhias aéreas;
- Selecionar todos os aeroportos;
- Selecionar todos os voos da GOL;
- Selecionar passagem e preço dos voos com origem em Salvador;
- Selecionar todas as passagens e preços da TAM ordenando pelo preço;
- Selecionar todos os voos com duração de 90.

Praticar

- Selecionar todas as companhias aéreas;
- Selecionar todos os aeroportos;
- Selecionar todos os voos da GOL;
- Selecionar passagem e preço dos voos com origem em Salvador;
- Selecionar todas as passagens e preços da TAM ordenando pelo preço;
- Selecionar todos os voos com duração de 90.

Praticar

- Selecionar todas as companhias aéreas;
- Selecionar todos os aeroportos;
- Selecionar todos os voos da GOL;
- Selecionar passagem e preço dos voos com origem em Salvador;
- Selecionar todas as passagens e preços da TAM ordenando pelo preço;
- Selecionar todos os voos com duração de 90.

Praticar

- Selecionar todas as companhias aéreas;
- Selecionar todos os aeroportos;
- Selecionar todos os voos da GOL;
- Selecionar passagem e preço dos voos com origem em Salvador;
- Selecionar todas as passagens e preços da TAM ordenando pelo preço;
- Selecionar todos os voos com duração de 90.

DBPedia

- DBPedia é uma versão em RDF da informação do Wikipedia.
- DBPedia contém dados derivados dos quadros de informação do Wikipedia, hierarquia de categoria, resumos de artigos e outros links externos.
- DBpedia contém quase 2 bilhões (aprox. 1.89) de triplas RDF.

DBPedia Endpoint

- DBpedia endpoint: http://dbpedia.org/sparql
- Selecione 50 conceitos presentes no DBPedia:

```
SELECT DISTINCT ?concept
WHERE {
        ?s rdf:type ?concept .
} LIMIT 50
```

DBPedia Endpoint

- DBpedia endpoint: http://dbpedia.org/sparql
- Selecione 50 conceitos presentes no DBPedia:

```
SELECT DISTINCT ?concept
WHERE {
        ?s rdf:type ?concept .
} LIMIT 50
```

- LIMIT é um modificador de resultado que limita o número de linhas retornado por uma consulta;

DBPedia Endpoint

- DBpedia endpoint: http://dbpedia.org/sparql
- Selecione 50 conceitos presentes no DBPedia:

```
SELECT DISTINCT ?concept
WHERE {
        ?s rdf:type ?concept .
} LIMIT 50
```

- LIMIT é um modificador de resultado que limita o número de linhas retornado por uma consulta;
- Pode ser usado junto com ORDER BY e OFFSET para recuperar um pedaço de um conjunto ordenado de soluções.

DBPedia

- Selecione todos os países sem litoral com uma população maior que 15 milhões:

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX type: <http://dbpedia.org/class/yago/>
PREFIX prop: <http://dbpedia.org/property/>
SELECT ?country_name ?population
WHERE {
   ?country rdf:type type:LandlockedCountries ;
            rdfs:label ?country_name ;
            prop:populationEstimate ?population .
   FILTER (?population > 15000000) .
}
```

DBPedia

- Selecione todos os países sem litoral com uma população maior que 15 milhões:

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX type: <http://dbpedia.org/class/yago/>
PREFIX prop: <http://dbpedia.org/property/>
SELECT ?country_name ?population
WHERE {
   ?country rdf:type type:LandlockedCountries ;
            rdfs:label ?country_name ;
            prop:populationEstimate ?population .
   FILTER (?population > 15000000) .
}
```

- A restrição FILTER usa condições booleanas para filtrar resultados indesejados;

DBPedia

- Selecione todos os países sem litoral com uma população maior que 15 milhões:

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX type: <http://dbpedia.org/class/yago/>
PREFIX prop: <http://dbpedia.org/property/>
SELECT ?country_name ?population
WHERE {
   ?country rdf:type type:LandlockedCountries ;
         rdfs:label ?country_name ;
         prop:populationEstimate ?population .
   FILTER (?population > 15000000) .
}
```

- A restrição FILTER usa condições booleanas para filtrar resultados indesejados;
- Atalho: o ponto e vírgula (;) pode ser usado para separar dois padrões de triplas que compartilham o mesmo sujeito (?country é um sujeito compartilhado);

DBPedia

- Selecione todos os países sem litoral com uma população maior que 15 milhões:

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX type: <http://dbpedia.org/class/yago/>
PREFIX prop: <http://dbpedia.org/property/>
SELECT ?country_name ?population
WHERE {
   ?country rdf:type type:LandlockedCountries ;
         rdfs:label ?country_name ;
         prop:populationEstimate ?population .
   FILTER (?population > 15000000) .
}
```

- A restrição FILTER usa condições booleanas para filtrar resultados indesejados;
- Atalho: o ponto e vírgula (;) pode ser usado para separar dois padrões de triplas que compartilham o mesmo sujeito (?country é um sujeito compartilhado);
- rdfs:label é um predicado comum para fornecer um rótulo amigável para humanos de um determinado recurso.

DBPedia

- Várias diferentes traduções para o mesmo país;
- Podemos especificar uma determinada linguagem:

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX type: <http://dbpedia.org/class/yago/>
PREFIX prop: <http://dbpedia.org/property/>
SELECT ?country_name ?population
WHERE {
   ?country rdf:type type:LandlockedCountries ;
         rdfs:label ?country_name ;
         prop:populationEstimate ?population .
   FILTER (?population > 15000000 &&
   langMatches(lang(?country_name), "EN")) .
}
```

DBPedia

- Várias diferentes traduções para o mesmo país;
- Podemos especificar uma determinada linguagem:

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX type: <http://dbpedia.org/class/yago/>
PREFIX prop: <http://dbpedia.org/property/>
SELECT ?country_name ?population
WHERE {
   ?country rdf:type type:LandlockedCountries ;
            rdfs:label ?country_name ;
            prop:populationEstimate ?population .
   FILTER (?population > 15000000 &&
  langMatches(lang(?country_name), "EN")) .
}
```

- lang extraí a tag de línguagem do literal, se existir

DBPedia

- Várias diferentes traduções para o mesmo país;
- Podemos especificar uma determinada linguagem:

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX type: <http://dbpedia.org/class/yago/>
PREFIX prop: <http://dbpedia.org/property/>
SELECT ?country_name ?population
WHERE {
   ?country rdf:type type:LandlockedCountries ;
            rdfs:label ?country_name ;
            prop:populationEstimate ?population .
   FILTER (?population > 15000000 &&
  langMatches(lang(?country_name), "EN")) .
}
```

- lang extraí a tag de línguagem do literal, se existir
- langMatches compara tags de linguagem

SPARQL

- *Logical:* !, &&, ||
- *Math:* +, -, *, /
- *Comparison:* =, !=, >, <, IN, NOT IN...
- *SPARQL tests:* isURI, isBlank, isLiteral, isNumeric, bound
- *SPARQL accessors:* str, lang, datatype
- *Other:* sameTerm, langMatches, regex, REPLACE
- *Conditionals (SPARQL 1.1):* IF, COALESCE, EXISTS, NOT EXISTS
- *Constructors (SPARQL 1.1):* URI, BNODE, STRDT, STRLANG, UUID, STRUUID
- *Strings (SPARQL 1.1):* STRLEN, SUBSTR, UCASE, LCASE, STRSTARTS, STRENDS, CONTAINS, STRBEFORE,STRAFTER, CONCAT, ENCODE_FOR_URI
- *More math (SPARQL 1.1):* abs, round, ceil, floor, RAND
- *Date/time (SPARQL 1.1):* now, year, month, day, hours, minutes, seconds, timezone, tz
- *Hashing (SPARQL 1.1):* MD5, SHA1, SHA256, SHA384, SHA512

Referências

- SPARQL By Example: A Tutorial, Lee Feigenbaum e Eric Prud'hommeaux;
- SPARQL 1.1 Query Language: http://www.w3.org/TR/2013/REC-sparql11-query-20130321/

OBRIGADO!