

Web ngữ nghĩa

Soạn bởi: Nguyễn Bá Ngọc

Chương 8

Hà Nội-2021

Chương 8.

SPARQL

Nội dung

8.1. Ngôn ngữ SPARQL

8.2. Đại số SPARQL

8.3. Biểu diễn luật trong SPARQL: SPIN

8.4. SPARQL và OWL

Nội dung

8.1. Ngôn ngữ SPARQL

8.2. Đại số SPARQL

8.3. Biểu diễn luật trong SPARQL: SPIN

8.4. SPARQL và OWL

Các ngôn ngữ truy vấn và Web ngữ nghĩa

- Làm sao để truy cập dữ liệu RDF(S) và OWL?
- Dữ liệu RDF(S)
 - Suy diễn (Entailment) đơn giản
 - Suy diễn RDF
 - Suy diễn RDFS
 - “Một đồ thị RDF liệu có được suy ra từ 1 đồ thị RDF khác?”
- Các ontology OWL
 - Suy diễn logic
 - “Có thể suy diễn quan hệ kế thừa giữa 2 lớp?”
 - “Làm sao để tìm tất cả các phần tử của 1 lớp?”

Các thành phần cơ bản của SPARQL

URIs

URIs đầy đủ:
<http://semweb.edu.vn/example/ontology#keyword>

Viết tắt URIs với các tiền tố:
PREFIX ex:

<http://semweb.edu.vn/example/ontology#>

ex:abc =

http://semweb.edu.vn/example/ontology#abc

Hằng giá trị

Chuỗi ký tự:

"Tháng ba mùa hoa gạo"

Chuỗi ký tự với thẻ ngôn ngữ:

"Hà Nội"@vi

Hằng có định kiểu:

"99"^^xsd:integer

Các giản lược:

true = "true"^^xsd:boolean

3.14 = "3.14"^^xsd:decimal

Các biến

Dấu ? đứng trước tên biến:

?x, ?y, ?z

Ghi chú

Ghi chú:

Ghi chú bắt đầu từ # đến hết dòng

Các mẫu bộ ba

ex:myWidget ex:hasId "abc123" .

?person foaf:name "Nguyen Van A" .

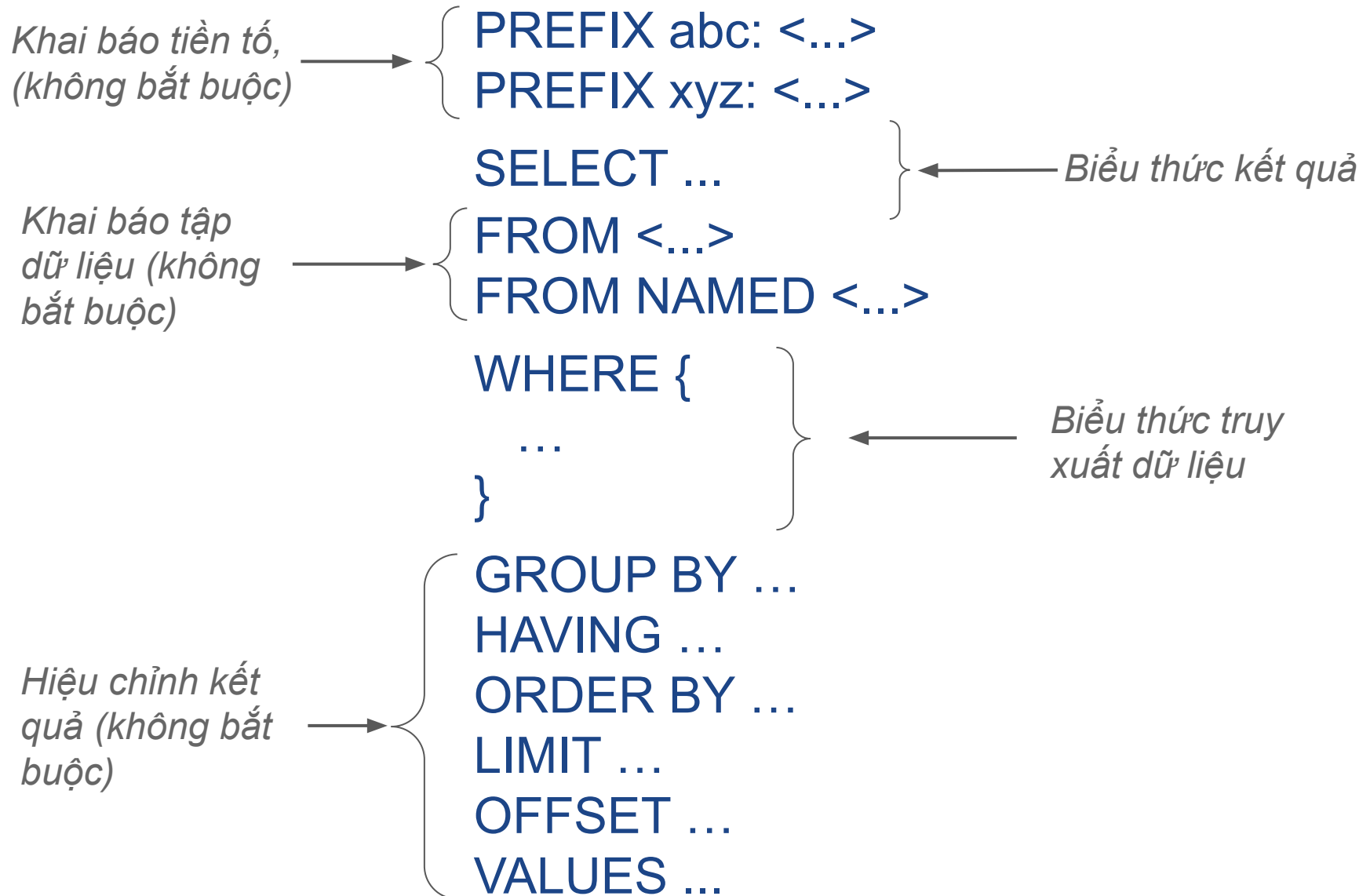
course:SemTech2021 ?property ?value .

So khớp chính xác 1 bộ ba RDF:

So khớp với 1 biến:

So khớp với nhiều biến:

Cấu trúc của câu truy vấn



Phân loại truy vấn SPARQL

Truy vấn SELECT

Xuất các biến và biểu thức cụ thể:

```
SELECT ?c ?cap (1000 * ?people AS ?pop)
```

Xuất tất cả các biến:

```
SELECT *
```

Xuất các bộ giá trị duy nhất:

```
SELECT DISTINCT ?country
```

Kết quả dạng bảng, (XML hoặc JSON):

?c	?cap	?pop
ex:Vietnam	ex:Hanoi	95 500 000

Truy vấn CONSTRUCT

Tổng hợp 1 đồ thị RDF:

```
CONSTRUCT {  
  ?country a ex:HolidayDestination ;  
  ex:arrive_at ?capital ;  
  ex:population ?population .  
}
```

Kết quả là tài liệu RDF:

Truy vấn ASK

Kiểm tra liệu xem có hay không bộ ba theo mẫu:

```
ASK
```

Kết quả có thể là true hoặc false:

```
ask where {  
  dbr:Vietnam dbo:capital dbr:Hanoi  
} => true
```

Truy vấn DESCRIBE

Mô tả tài nguyên khớp với biến được cho:

```
DESCRIBE ?c
```

Kết quả là tài liệu RDF:

```
describe dbc:Hanoi  
=> dbc:Hanoi rdfs:label "Hanoi"@en ;  
      skos:broader dbc:Cities_in_Vietnam ;  
...
```


Các câu lệnh cập nhật trong SPARQL 1.1

Các câu lệnh cập nhật bộ ba

INSERT DATA { triples }

DELETE DATA {triples}

[DELETE { template }] [INSERT { template }] WHERE { pattern }

LOAD <uri> [INTO GRAPH <uri>]

CLEAR GRAPH <uri>

CREATE GRAPH <uri>

DROP GRAPH <uri>

[...] = các thành phần không bắt buộc trong SPARQL 1.1

Kết hợp các mẫu đồ thị SPARQL

Cho A và B là các mẫu đồ thị

- Mẫu đồ thị cơ bản - Basic Graph Pattern (BGP) - bao gồm 1 hoặc nhiều mẫu bộ ba:
 - $A \text{ } \circ \text{ } B$
 - Phép giao \circ - Bao gồm các kết quả phân giải A và B bằng cách khớp giá trị của các biến chung.
- Các mẫu không bắt buộc - Optional:
 - $A \text{ OPTIONAL } \{ B \}$
 - Kết hợp trái (LeftJoin) - Kết hợp các kết quả phân giải A và B bằng cách khớp giá trị cho các biến chung nếu có thể. Lấy tất cả các kết quả từ A dù có kết quả từ B hay không.

Kết hợp các mẫu đồ thị₍₂₎

- Hợp
 - $\{A\}$ Union $\{B\}$
 - Phép hợp - bao gồm các kết quả phân giải A và các kết quả phân giải B.
- Trừ (SPARQL 1.1)
 - A MINUS $\{B\}$
 - Giao với phần bù. Phân giải A. Phân giải B. Chỉ lấy các kết quả phân giải A nhưng không phải là kết quả phân giải B.

Truy vấn con (SPARQL 1.1)

Cho A, B và C là các mẫu đồ thị

```
A . {  
    SELECT ...  
    WHERE {  
        B  
    }  
}  
C .
```

=> Kết hợp các kết quả của truy vấn con với các kết quả phân giải A và C.

Các bộ lọc SPARQL

- Biểu thức lọc (FILTER) lấy những kết quả phân giải thỏa mãn điều kiện lọc (Biểu-thức-lọc = true)
 - A . B . FILTER (Biểu-thức-lọc)


Danh mục	Hàm/Toán tử	Ví dụ
Lô-gic & so sánh	!, &&, , =, !=, <, <=, >, >=, IN, NOT IN	?hasPermit ?age < 25
Điều kiện (SPARQL 1.1)	EXISTS, NOT EXISTS, IF, COALESCE	NOT EXISTS { ?p foaf:mbox ?email }
Toán học	+, -, *, /, abs, round, ceil, floor, RAND	?decimal * 10 > ?minPercent
Chuỗi (SPARQL 1.1)	STRLEN, SUBSTR, UCASE, LCASE, STRSTARTS, CONCAT, STRENDS, CONTAINS, STRBEFORE, STRAFTER	STRLEN(?desc) < 255

Các bộ lọc SPARQL₍₂₎


Danh mục	Hàm/Toán tử	Ví dụ
Ngày/giờ (SPARQL 1.1)	now, year, month, day, hours, minutes, seconds, timezone, tz	month(now()) < 4
Các kiểm tra	isURI, isBlank, isLiteral, isNumeric, bound	isURI(?person) !bound(?person)
Các hàm tạo (SPARQL 1.1)	URI, BNODE, STRDT, STRLANG, UUID, STRUUID	STRLANG(?text, “en”) = “hello”@en
Truy cập	str, lang, datatype	lang(?title) = “en”
Hàm băm (SPARQL 1.1)	MD5, SHA1, SHA256, SHA512	BIND(SHA256(?email) AS ?hash)
Thành phần khác	sameTerm, langMatches, regex, REPLACE	regex(?ssn, “\\d{3}-\\d{2}-\\d{4}”)

Lệnh tổng hợp (SPARQL 1.1)


1. Chia kết quả thành các nhóm dựa trên biểu thức được cung cấp bởi **GROUP BY**
2. Thực hiện các hàm chiếu và tổng hợp trong biểu thức **SELECT** để lấy từ mỗi nhóm 1 kết quả.
3. Lọc các kết quả đã tổng hợp theo biểu thức **HAVING**.



?key	?val	?other1
1	4	...
1	4	...
2	5	...
2	4	...
2	10	...
2	2	...
2	1	...
3	3	...



?key	?sum_of_val
1	8
2	22
3	3



?key	?sum_of_val
1	8
3	3

SPARQL 1.1 bao gồm: **COUNT** , **SUM** , **AVG** , **MIN** , **MAX**, **SAMPLE**, **GROUP_CONCAT**

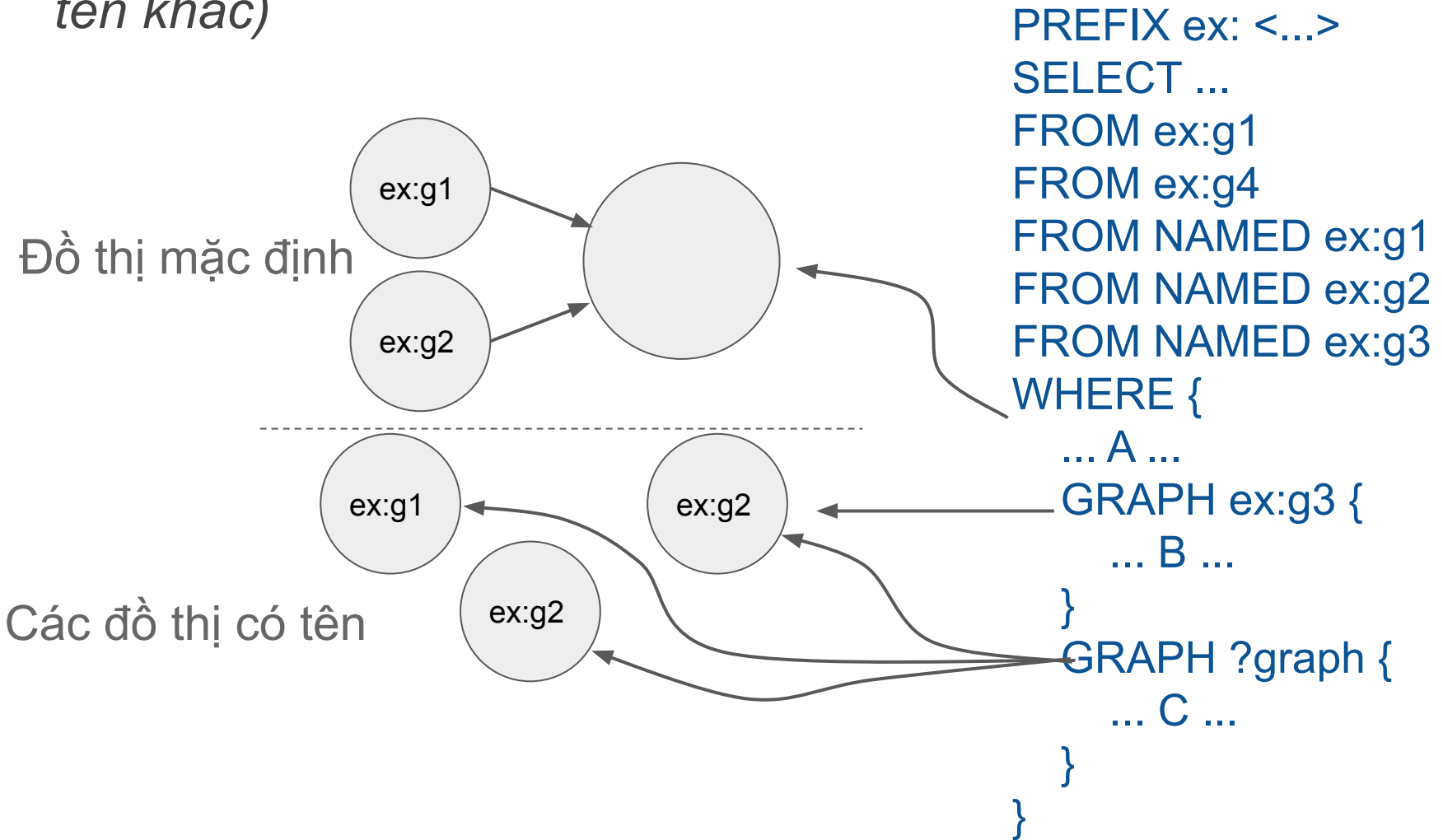
Đường dẫn thuộc tính (SPARQL 1.1)

- Đường dẫn thuộc tính cho phép khớp mẫu bộ ba với các đường dẫn độ dài tùy ý trong 1 đồ thị

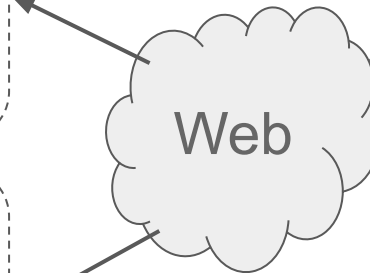
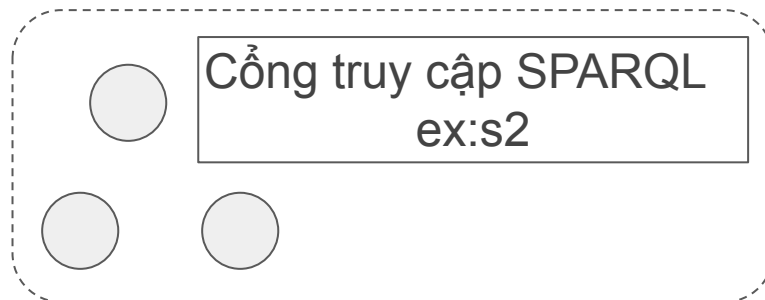
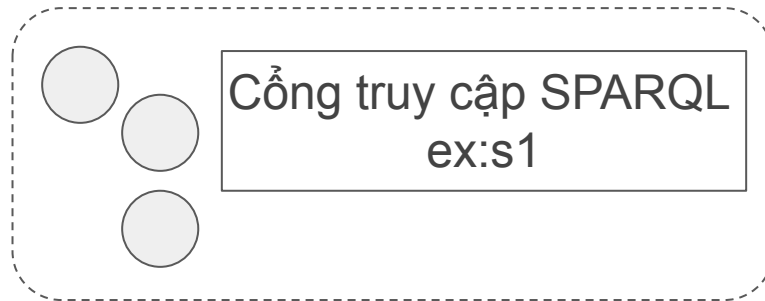
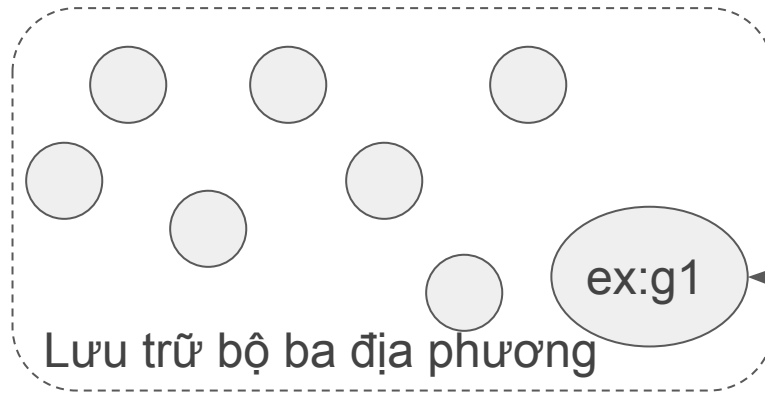
Biểu thức	Ý nghĩa
path1/path2	path1 được nối tiếp bởi path2
^path1	Liên kết ngược (giá trị tới tài nguyên)
path1 path2	path1 hoặc path2
path1*	path1, được lặp lại nhiều lần hoặc không
path1+	path1 được lặp lại 1 hoặc nhiều lần
path1?	không bắt buộc
!uri	Bất kỳ thuộc tính nào ngoại trừ uri
!^uri	Bất kỳ liên kết ngược nào ngoại trừ uri

Các bộ dữ liệu RDF

Câu truy vấn SPARQL (thường) được dùng để xuất dữ liệu từ 1 đồ thị mặc định (và có thể được kết hợp với các đồ thị có tên khác)



Truy vấn liên kết (SPARQL 1.1)



```
PREFIX ex: <...>
SELECT ...
FROM ex:g1
WHERE {
  ... A ...
  SERVICE ex:s1 {
    ... B ...
  }
  SERVICE ex:s2 {
    ... C ...
  }
}
```

Minh họa truy vấn SPARQL

Apache Jena Fuseki Server: Cổng SPARQL

The screenshot displays the Apache Jena Fuseki Server web interface in a browser. The address bar shows the URL `localhost:3030/dataset.html?tab=query&ds=/learn-sparql`. The interface includes a navigation bar with the Apache Jena Fuseki logo, a 'dataset' tab, and links for 'manage datasets' and 'help'. A 'Server status' indicator shows a green dot.

The main content area is titled 'Dataset: /learn-sparql'. Below this, there are tabs for 'query', 'upload files', 'edit', and 'info'. The 'query' tab is active, showing a 'SPARQL query' section with the instruction: 'To try out some SPARQL queries against the selected dataset, enter your query here.'

Under 'EXAMPLE QUERIES', there are two buttons: 'Selection of triples' (highlighted) and 'Selection of classes'. Below this, the 'PREFIXES' section shows buttons for 'rdf', 'rdfs', 'owl', 'xsd', and a plus icon.

The 'SPARQL ENDPOINT' field contains `/learn-sparql/query`. The 'CONTENT TYPE (SELECT)' dropdown is set to 'JSON', and the 'CONTENT TYPE (GRAPH)' dropdown is set to 'Turtle'.

The query editor shows the following SPARQL query:

```
1 SELECT ?subject ?predicate ?object
2 WHERE {
3   ?subject ?predicate ?object
4 }
5 LIMIT 25
```

At the bottom, the 'QUERY RESULTS' section has tabs for 'Table' (selected) and 'Raw Response', along with a download icon. It indicates 'Showing 1 to 12 of 12 entries'. A search bar and a 'Show 50 entries' dropdown are also present.

Ví dụ 8.1. Dữ liệu RDF

filename: ex012.ttl

@prefix ab: <http://learningsparql.com/ns/addressbook#> .

@prefix d: <http://learningsparql.com/ns/data#> .

d:i0432 ab:firstName "Richard" .

d:i0432 ab:lastName "Mutt" .

d:i0432 ab:homeTel "(229) 276-5135" .

d:i0432 ab:email "richard49@hotmail.com" .

d:i9771 ab:firstName "Cindy" .

d:i9771 ab:lastName "Marshall" .

d:i9771 ab:homeTel "(245) 646-5488" .

d:i9771 ab:email "cindym@gmail.com" .

d:i8301 ab:firstName "Craig" .

d:i8301 ab:lastName "Ellis" .

d:i8301 ab:email "craigellis@yahoo.com" .

d:i8301 ab:email "c.ellis@usairwaysgroup.com" .

Ví dụ 8.2. SELECT

filename: ex174.rq

PREFIX ab: <http://learningsparql.com/ns/addressbook#>

PREFIX d: <http://learningsparql.com/ns/data#>

SELECT ?person ?p ?o

WHERE

```
{  
  ?person ab:firstName "Craig" ;  
          ab:lastName  "Ellis" ;  
          ?p ?o .  
}
```

arq --data=data.ttl --query=q1.rq

	person	p	o
1	d:i8301	ab:email	"c.ellis@usairwaysgroup.com"
2	d:i8301	ab:email	"craigellis@yahoo.com"
3	d:i8301	ab:lastName	"Ellis"
4	d:i8301	ab:firstName	"Craig"

Ví dụ 8.3. CONSTRUCT

PREFIX ab: <http://learningsparql.com/ns/addressbook#>

PREFIX d: <http://learningsparql.com/ns/data#>

CONSTRUCT

{ ?person ?p ?o . }

WHERE {

 ?person ab:firstName "Craig" ;

 ab:lastName "Ellis" ;

 ?p ?o .

}

```
1 @prefix ab:    <http://learningsparql.com/ns/addressbook#> .
2 @prefix d:     <http://learningsparql.com/ns/data#> .
3
4 d:i8301 ab:email      "craigellis@yahoo.com" ,
5         "c.ellis@usairwaysgroup.com" ;
6         ab:firstName  "Craig" ;
7         ab:lastName   "Ellis" .
```

Ví dụ 8.4. DESCRIBE

PREFIX d: <http://learningsparql.com/ns/data#>

PREFIX ab: <http://learningsparql.com/ns/addressbook#>

DESCRIBE ?x

WHERE { ?x ab:email "craigellis@yahoo.com" . }

```
1 @prefix ab:    <http://learningsparql.com/ns/addressbook#> .
2 @prefix d:    <http://learningsparql.com/ns/data#> .
3
4 d:i8301 ab:email "craigellis@yahoo.com" ,
5         "c.ellis@usairwaysgroup.com" ;
6         ab:firstName "Craig" ;
7         ab:lastName  "Ellis" .
```


Ví dụ 8.5. ASK

PREFIX d: <http://learningsparql.com/ns/data#>

PREFIX ab: <http://learningsparql.com/ns/addressbook#>

ASK WHERE

{ ?x ab:firstName "Craig" . }

=> Yes

Ví dụ 8.6 OPTIONAL

Bổ xung các bộ ba sau:

```
d:i88888 ab:email "nva@gmail.com" ; ab:firstName "A" ;  
    ab:lastName "Nguyen Van" ; ab:nick "Super" .
```

```
PREFIX ab: <http://learningsparql.com/ns/addressbook#>
```

```
SELECT ?firstName ?last
```

```
WHERE { ?s ab:lastName ?last; ab:firstName ?first .
```

```
    OPTIONAL { ?s ab:nick ?nickname . }
```

```
    BIND (COALESCE(?nickname,?first) AS ?firstName)
```

```
}
```

	firstName	last
1	"Super"	"Nguyen Van"
2	"Cindy"	"Marshall"
3	"Craig"	"Ellis"
4	"Richard"	"Mutt"

Ví dụ 8.7. FILTER

PREFIX ab: <http://learningsparql.com/ns/addressbook#>

SELECT *

WHERE {

 ?s ?p ?o .

 FILTER (regex(?o, "yahoo","i"))

}

	s	p	o
1	<http://learningsparql.com/ns/data#i8301>	ab:email	"craigellis@yahoo.com"

Ví dụ 8.8. Union

```
d:i0432 ab:firstName "Richard" ; ab:lastName "Mutt" ;  
      ab:instrument "sax" ; ab:instrument "clarinet" .
```

```
d:i8301 ab:firstName "Craig" ; ab:lastName "Ellis" ;  
      ab:instrument "trumpet" .
```

```
PREFIX ab: <http://learningsparql.com/ns/addressbook#>
```

```
SELECT ?first ?last ?instrument
```

```
WHERE { { ?person ab:firstName ?first ; ab:lastName ?last ;  
      ab:instrument "trumpet" ; ab:instrument ?instrument . }  
UNION
```

```
{ ?person ab:firstName ?first ; ab:lastName ?last ;  
      ab:instrument "sax" ; ab:instrument ?instrument . } }
```

	first	last	instrument
1	"Craig"	"Ellis"	"trumpet"
2	"Richard"	"Mutt"	"sax"
3	"Richard"	"Mutt"	"clarinet"

Ví dụ 8.9 SERVICE (SPARQL 1.1)

SELECT ?p ?o

WHERE {

SERVICE <http://DBpedia.org/sparql>

{ SELECT ?p ?o

WHERE { <http://dbpedia.org/resource/Joseph_Hocking> ?p ?o . } }

Showing 1 to 10 of 98 entries

Search: Show entries

	p	o
1	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Thing
2	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://dbpedia.org/class/yago/Object100002684
3	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://dbpedia.org/class/yago/Organism100004475
4	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://dbpedia.org/class/yago/Person100007846
5	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://dbpedia.org/class/yago/PhysicalEntity100001930
6	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://dbpedia.org/class/yago/Whole100003553
7	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://dbpedia.org/class/yago/Wikicat19th-centuryMethodistMinisters
8	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://dbpedia.org/class/yago/WikicatPeopleFromStStephen-in-Brannel
9	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://dbpedia.org/class/yago/Writer110794014
10	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://dbpedia.org/class/yago/YagoLegalActor

Showing 1 to 10 of 98 entries

First Previous **1** 2 3 4 5 ... 10 Next Last

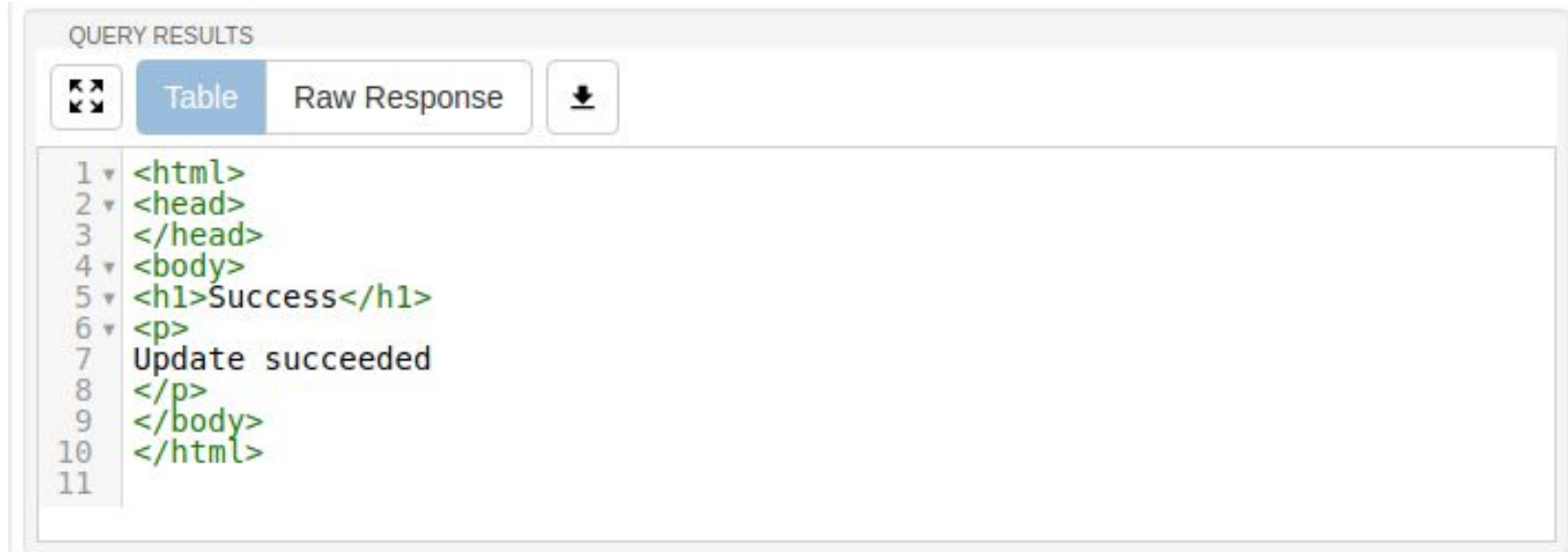
Ví dụ 8.10. INSERT (SPARQL 1.1)

PREFIX d: <http://learningsparql.com/ns/data#>

PREFIX ab: <http://learningsparql.com/ns/addressbook#>

INSERT DATA

```
{  
  d:99999 ab:firstName "Nguyen" ;  
  ab:lastName "VB" .  
}
```



Nội dung

8.1. Ngôn ngữ SPARQL

8.2. Đại số SPARQL

8.3. Biểu diễn luật trong SPARQL: SPIN

8.4. SPARQL và OWL

Biên dịch sang đại số SPARQL

```
{ ?book ex:price ?price  
  FILTER (?price < 15)  
  OPTIONAL { ?book ex:title ?title }  
  { ?book ex:author ex:Shakespeare } UNION  
  { ?book ex:author ex:Marlowe }  
}
```

!Ghi chú: Bộ lọc được áp dụng cho nhóm chứa nó.

Biên dịch sang đại số SPARQL₍₂₎

B1. Mở rộng các IRIs thành dạng đầy đủ (bước này không bắt buộc - có thể bỏ qua nếu tiếp tục sử dụng các tiền tố trong biểu thức đại số)

```
{ ?book <http://ex.org/price> ?price  
  OPTIONAL { ?book <http://ex.org/title> ?title }  
  { ?book <http://ex.org/author>  
    <http://ex.org/Shakespeare> } UNION  
  { ?book <http://ex.org/author>  
    <http://ex.org/Marlowe> }  
  FILTER (?price < 15)  
}
```

Biên dịch sang đại số SPARQL₍₂₎

B2. Đưa các nhóm mẫu bộ ba vào các biểu thức BGP(.)

```
{ BGP(?book <http://ex.org/price> ?price)
  OPTIONAL {BGP(?book <http://ex.org/title> ?title)}
  {BGP(?book <http://ex.org/author>
        <http://ex.org/Shakespeare>)} UNION
  {BGP(?book <http://ex.org/author>
        <http://ex.org/Marlowe>)}
  FILTER (?price < 15)
}
```

Biên dịch sang đại số SPARQL₍₃₎

B3. Sử dụng toán tử LeftJoin(.) cho các thành phần không bắt buộc

```
{ LeftJoin(BGP(?book <http://ex.org/price> ?price),  
           BGP(?book <http://ex.org/title> ?title), true)  
  {BGP(?book <http://ex.org/author>  
       <http://ex.org/Shakespeare>)} UNION  
  {BGP(?book <http://ex.org/author>  
       <http://ex.org/Marlowe>)}  
  FILTER (?price < 15)  
}
```

Biên dịch sang đại số SPARQL₍₄₎

B4. Kết hợp các mẫu đồ thị tương đương bằng toán tử Union(.)

```
{ LeftJoin(BGP(?book <http://ex.org/price> ?price),  
           BGP(?book <http://ex.org/title> ?title), true)  
  Union(BGP(?book <http://ex.org/author>  
          <http://ex.org/Shakespeare>),  
        BGP(?book <http://ex.org/author>  
          <http://ex.org/Marlowe>))  
  FILTER (?price < 15)  
}
```

Biên dịch sang đại số SPARQL₍₅₎

B5. Áp dụng toán tử Join(.) để liên kết các thành phần

```
{ Join(  
  LeftJoin(BGP(?book <http://ex.org/price> ?price),  
            BGP(?book <http://ex.org/title> ?title), true)  
  Union(BGP(?book <http://ex.org/author>  
            <http://ex.org/Shakespeare>),  
         BGP(?book <http://ex.org/author>  
            <http://ex.org/Marlowe>)))  
  FILTER (?price < 15)  
}
```

Biên dịch sang đại số SPARQL₍₆₎

B6. Áp dụng toán tử `Filter(.)` cho nhóm có bộ lọc

`Filter(?price < 15,`

`Join(`

`LeftJoin(BGP(?book <http://ex.org/price> ?price),`

`BGP(?book <http://ex.org/title> ?title), true),`

`Union(BGP(?book <http://ex.org/author>`

`<http://ex.org/Shakespeare>),`

`BGP(?book <http://ex.org/author>`

`<http://ex.org/Marlowe>))))`

Công cụ trực tuyến: <http://sparql.org/query-validator.html>

Thực hiện biểu thức đại số SPARQL

Xử lý mẫu đồ thị cơ bản

Định nghĩa lời giải:

Đặt P là một mẫu đồ thị cơ bản. Một hàm thành phần μ là 1 lời giải cho $BGP(P)$ trên đồ thị G nếu:

- (1) Miền của μ là tập các biến trong P
- (2) Tồn tại phép gán σ từ các nút rỗng trong P tới IRIs, nút rỗng hoặc hằng giá trị RDF sao cho:
- (3) Đồ thị RDF $\mu(\sigma(P))$ là đồ thị con của G

- Kết quả xử lý $BGP(P)$ trên G được viết là $[BGP(P)]_G$
- Kết quả là 1 tập đa giá trị của μ
- Cơ số của μ = số lượng phép gán σ khác nhau để μ là lời giải.
 - (σ - Ánh xạ các nút rỗng).

Ví dụ 8.11. Khớp mẫu

```
ex:NVA ex:gives [  
    a ex:Lesson ; ex:hasTopic "SPARQL" ] .
```

```
ex:NVB ex:gives [  
    a ex:Lesson ; ex:hasTopic "DLs and OWL" ] .
```

```
BGP(?who ex:gives _ :x . _ :x ex:hasTopic ?what)
```

Ví dụ 8.11. Khớp mẫu₍₂₎

ex:NVA ex:gives _ :a .

_ :a rdf:type ex:Lesson; ex:hasTopic "SPARQL" .

ex:Sebastian ex:gives _ :b .

_ :b rdf:type ex:Lesson; ex:hasTopic "DLs and OWL" .

BGP(?who ex:gives _ :x . _ :x ex:hasTopic ?what)

μ_1 : ?who \rightarrow ex:NVA,

σ_1 : _ :x \rightarrow _ :a ?what \rightarrow "SPARQL"

μ_2 : ?who \rightarrow ex:NVB,

σ_2 : _ :x \rightarrow _ :b ?what \rightarrow "DLs and OWL"

Hai lời giải với cơ số = 1

Khái niệm lời giải tương thích

Định nghĩa:

Hai lời giải μ_1 và μ_2 được gọi là tương thích nếu $\mu_1(x) = \mu_2(x)$, với tất cả x là biến chung của μ_1 và μ_2 .

$\mu_1 : ?x \rightarrow ex:a, ?y \rightarrow ex:b$

$\mu_2 : ?y \rightarrow ex:b, ?z \rightarrow ex:c$ ✓

$\mu_3 : ?x \rightarrow ex:a, ?y \rightarrow ex:b$

$\mu_4 : ?x \rightarrow ex:b, ?z \rightarrow ex:c$ ✗

$\mu_5 : ?x \rightarrow ex:a$

$\mu_6 : ?y \rightarrow ex:b$ ✓

Hợp của 2 lời giải tương thích

Hợp của 2 lời giải tương thích μ_1 và μ_2 được ký hiệu là : $\mu_1 \cup \mu_2$ và thỏa mãn:

$\text{dom}(\mu_1 \cup \mu_2) = \text{dom}(\mu_1) \cup \text{dom}(\mu_2)$, và
với $\forall x \in \text{dom}(\mu_1) \cup \text{dom}(\mu_2)$

$$(\mu_1 \cup \mu_2)(x) = \begin{cases} \mu_1(x) & \text{nếu } x \in \text{dom}(\mu_1) \\ \mu_2(x) & \text{nếu ngược lại} \end{cases}$$

Toán tử Union

Kết quả thực hiện $\text{Union}(A_1, A_2)$ trên đồ thị G với các biểu thức A_1, A_2 được ký hiệu là $[\text{Union}(A_1, A_2)]_G$ và

$$[\text{Union}(A_1, A_2)]_G = \{\mu | \mu \in [A_1]_G \text{ hoặc } \mu \in [A_2]_G\}$$

Toán tử Join

$[\text{Join}(A_1, A_2)]_G = \{\mu_1 \cup \mu_2 \mid \mu_1 \in [A_1]_G, \mu_2 \in [A_2]_G \text{ và } \mu_1 \text{ tương thích với } \mu_2\}$

Ví dụ 8.12. Join

Chúng ta xét $\text{Join}(A_1, A_2)$ trên một đồ thị G với:

$$[A_1]_G = \{(\mu_1 : ?x \rightarrow \text{ex}:a, ?y \rightarrow \text{ex} : b), (\mu_2 : ?x \rightarrow \text{ex}: d)\}$$

$$[A_2]_G = \{(\mu_3 : ?y \rightarrow \text{ex} : b, ?z \rightarrow \text{ex} : c)\}$$

$$[\text{Join}(A_1, A_2)]_G = \{((\mu_1 \cup \mu_3) : ?x \rightarrow \text{ex}:a, ?y \rightarrow \text{ex}:b, ?z \rightarrow \text{ex}:c), \\ ((\mu_2 \cup \mu_3) : ?x \rightarrow \text{ex}:d, ?y \rightarrow \text{ex}:b, ?z \rightarrow \text{ex}:c)\}$$

Toán tử Filter

$\text{Filter}(F, A)$ với F là 1 biểu thức điều kiện lọc và A là 1 biểu thức đại số có kết quả là:

$$[\text{Filter}(F, A)]_G = \{\mu \mid \mu \in [A]_G \text{ và } \mu(F) == \text{true}\},$$

trong đó $\mu(F)$ là giá trị biểu thức F với bộ giá trị từ μ .

Toán tử LeftJoin

$$[\text{LeftJoin}(A_1, A_2, F)]_G =$$

$\{\mu_1 \cup \mu_2 \mid \mu_1 \in [A_1]_G, \mu_2 \in [A_2]_G \text{ và } \mu_1 \text{ tương thích với } \mu_2 \text{ và } (\mu_1 \cup \mu_2)(F) == \text{true}\}$

\cup

$\{\mu_1 \mid \mu_1 \in [A_1]_G \text{ và với } \forall \mu_2 \in [A_2]_G \text{ thì } \mu_1 \text{ và } \mu_2 \text{ không tương thích hoặc } (\mu_1 \cup \mu_2)(F) == \text{false}\}$

Ví dụ 8.13. Biểu thức đại số SPARQL

@prefix ex: <http://eg.org/> .

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

ex:Hamlet ex:author ex:Shakespeare ; ex:price "10.50"^^xsd:decimal .

ex:Macbeth ex:author ex:Shakespeare .

ex:Tamburlaine ex:author ex:Marlowe ; ex:price "17"^^xsd:integer .

ex:DoctorFaustus ex:author ex:Marlowe ; ex:price "12"^^xsd:integer ;

ex:title "The Tragical History of Doctor Faustus" .

ex:RomeoJulia ex:author ex:Brooke ; ex:price "9"^^xsd:integer .

prefix ex: <http://eg.org/>

select ?book ?price ?title

where { ?book ex:price ?price . FILTER (?price < 15)

OPTIONAL { ?book ex:title ?title . }

{ ?book ex:author ex:Shakespeare . } UNION

{ ?book ex:author ex:Marlowe . }

}

Ví dụ 8.13. Biểu thức đại số SPARQL₍₂₎

@prefix ex: <http://eg.org/> .

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

ex:Hamlet ex:author ex:Shakespeare ; ex:price "10.50"^^xsd:decimal .

ex:Macbeth ex:author ex:Shakespeare .

ex:Tamburlaine ex:author ex:Marlowe ; ex:price "17"^^xsd:integer .

ex:DoctorFaustus ex:author ex:Marlowe ; ex:price "12"^^xsd:integer ;

ex:title "The Tragical History of Doctor Faustus" .

ex:RomeoJulia ex:author ex:Brooke ; ex:price "9"^^xsd:integer .

```
{ ?book ex:price ?price . FILTER (?price < 15)
  OPTIONAL { ?book ex:title ?title . }
  { ?book ex:author ex:Shakespeare . } UNION
  { ?book ex:author ex:Marlowe . }
}
```

Ví dụ 8.13. Biểu thức đại số SPARQL₍₃₎

@prefix ex: <http://eg.org/> .

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

ex:Hamlet ex:author ex:Shakespeare ; ex:price "10.50"^^xsd:decimal .

ex:Macbeth ex:author ex:Shakespeare .

ex:Tamburlaine ex:author ex:Marlowe ; ex:price "17"^^xsd:integer .

ex:DoctorFaustus ex:author ex:Marlowe ; ex:price "12"^^xsd:integer ;

ex:title "The Tragical History of Doctor Faustus" .

ex:RomeoJulia ex:author ex:Brooke ; ex:price "9"^^xsd:integer .

Filter(?price < 15,

Join(LeftJoin(BGP(?book <http://eg.org/price> ?price),

BGP(?book <http://eg.org/title> ?title), true),

Union(BGP(?book <http://eg.org/author> <http://eg.org/Shakespeare>),

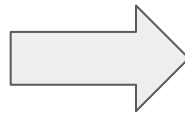
BGP(?book <http://eg.org/author> <http://eg.org/Marlowe>))))

Ví dụ 8.13. Biểu thức đại số SPARQL₍₄₎

```
Filter(?price < 15,  
  Join(LeftJoin(BGP(?book <http://eg.org/price> ?price),  
    BGP(?book <http://eg.org/title> ?title), true),  
  Union(BGP(?book <http://eg.org/author> <http://eg.org/Shakespeare>),  
    BGP(?book <http://eg.org/author> <http://eg.org/Marlowe>))))
```

book
ex:Macbeth
ex:Hamlet

book
ex:Tamburlaine
ex:DoctorFaustus



book
ex:Hamlet
ex:Macbeth
ex:Tamburlaine
ex:DoctorFaustus

Ví dụ 8.13. Biểu thức đại số SPARQL₍₅₎

Filter(?price < 15,

Join(LeftJoin(BGP(?book <http://eg.org/price> ?price),

BGP(?book <http://eg.org/title> ?title), true),

Union(BGP(?book <http://eg.org/author> <http://eg.org/Shakespeare>),

BGP(?book <http://eg.org/author> <http://eg.org/Marlowe>))))

book	price
ex:Hamlet	10.5
ex:Tamburlaine	17
ex:DoctorFaustus	12
ex:RomeoJulia	9

book	title
ex:DoctorFaustus	"The Tragical History of Doctor Faustus"

book	price	title
ex:Hamlet	10.5	
ex:Tamburlaine	17	
ex:DoctorFaustus	12	"The Tragical History of Doctor Faustus"
ex:RomeoJulia	9	

Ví dụ 8.13. Biểu thức đại số SPARQL₍₆₎

```
Filter(?price < 15,  
  Join(LeftJoin(BGP(?book <http://eg.org/price> ?price),  
    BGP(?book <http://eg.org/title> ?title), true),  
  Union(BGP(?book <http://eg.org/author> <http://eg.org/Shakespeare>),  
    BGP(?book <http://eg.org/author> <http://eg.org/Marlowe>))))
```

book	price	title
ex:Hamlet	10.5	
ex:Tamburlaine	17	
ex:DoctorFaustus	12	"The Tragical History of Doctor Faustus"

Ví dụ 8.13. Biểu thức đại số SPARQL₍₇₎

Filter(?price < 15,

Join(LeftJoin(BGP(?book <http://eg.org/price> ?price),

BGP(?book <http://eg.org/title> ?title), true),

Union(BGP(?book <http://eg.org/author> <http://eg.org/Shakespeare>),

BGP(?book <http://eg.org/author> <http://eg.org/Marlowe>))))

book	price	title
ex:Hamlet	10.5	
ex:DoctorFaustus	12	"The Tragical History of Doctor Faustus"

Các toán tử điều chỉnh

ToList(M)	Tạo 1 danh sách từ 1 đa tập với cùng các phần tử và cơ sở (thứ tự tùy ý, các trùng lặp không bắt buộc liên nhau)
OrderBy(M, comparators)	Sắp xếp lời giải
Distinct(M)	Loại bỏ các trùng lặp
Reduced(M)	Giảm lược, có thể loại bỏ trùng lặp
Slice(M, o, l)	Cắt các lời giải thành một danh sách độ dài l bắt đầu từ vị trí o
Project(M, vars)	Hình chiếu của M trong không gian với các biến được nhắc tới

Nội dung

8.1. Ngôn ngữ SPARQL

8.2. Đại số SPARQL

8.3. Biểu diễn luật trong SPARQL: SPIN

8.4. SPARQL và OWL

Kiến trúc SPIN

Các mô-đun thư viện tiêu chuẩn SPIN

<http://spinrdf.org/spl>

Các cấu trúc mô hình hóa có thể tái sử dụng như cơ sở, liên kết thành phần và lớp

Bộ từ vựng SPIN để mô hình hóa

<http://spinrdf.org/spin>

Luật và ràng buộc

Xác định ngữ nghĩa của lớp và các mối quan hệ

Các hàm và khuôn mẫu

Đóng gói các truy vấn có thể tái sử dụng với các tham số mẫu

Cú pháp SPARQL SPIN

<http://spinrdf.org/sp>

Một bộ từ vựng RDF để biểu diễn các truy vấn, các biến, và các biểu thức lọc v.v..

Cú pháp SPIN RDF cho SPARQL

SPIN cung cấp 1 bộ từ vựng để biểu diễn các truy vấn SPARQL như những bộ ba RDF.

```
ASK WHERE {  
  ?this ss:width ?width .  
  ?this ss:height ?height .  
  FILTER (?width != ?height) .  
}
```

Các lợi ích:

- Lưu các truy vấn SPARQL cùng với mô hình
- Dễ dàng chia sẻ trên WNN
- Tính nhất quán tham chiếu (liên hệ đúng nguồn).
- Các không gian tên được thiết lập một lần thay vì cho từng truy vấn.

```
[ a sp:Ask ;  
  rdfs:comment "Width and height must not be  
equal"^^xsd:string ;  
  sp:where ([ sp:object _:b1 ; sp:predicate ss:width ;  
    sp:subject spin:_this]  
    [ sp:object _:b2 ; sp:predicate ss:height ;  
      sp:subject spin:_this]  
  [ a sp:Filter ; sp:expression  
    [ a sp:ne ; sp:arg1 _:b1 ; sp:arg2 _:b2]])  
] .
```

Ngôn ngữ ràng buộc SPIN cho SPARQL

Thuộc tính spin:constraint có thể được sử dụng để liên kết một lớp với các truy vấn SPARQL để hình thức hóa các bất biến cho các phần tử của lớp

The screenshot shows a 'Class Form' window for the class 'ss:Square'. It includes sections for 'Annotations' (with 'rdfs:label' set to 'Square'), 'Class Axioms' (with 'rdfs:subClassOf' set to 'ss:Rectangle'), and 'Other Properties' (with 'spin:constraint' set to a SPARQL query). The SPARQL query is:

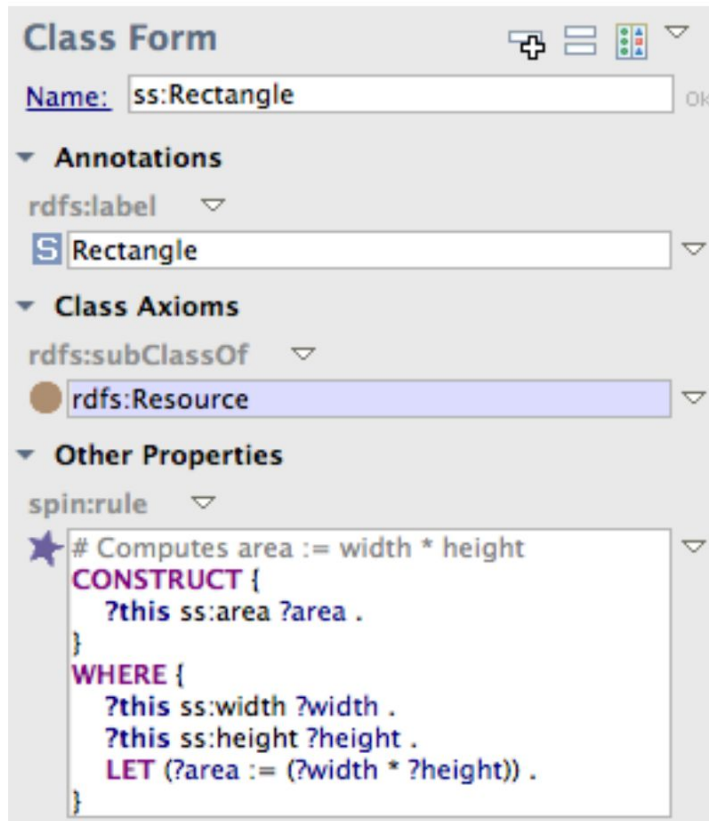
```
# Width and height must be equal
ASK WHERE {
  ?this ss:width ?width .
  ?this ss:height ?height .
  FILTER (?width != ?height) .
}
```

Các lợi ích:

- Cách tiếp cận mô hình hóa HDT
- SPARQL có tính diễn đạt cao
- Các ràng buộc có thể được thực thi bởi chính các mô-tơ SPARQL của CSDL
- Dễ dàng kết hợp với các ngôn ngữ dựa trên ràng buộc khác như SKOS SPIN

Ngôn ngữ luật SPIN cho SPARQL

Thuộc tính spin:rule có thể được sử dụng để liên kết một lớp với các truy vấn SPARQL CONSTRUCT để thiết lập các luật suy diễn cho các phần tử của lớp



Các lợi ích:

- Cách tiếp cận mô hình hóa HĐT
- SPARQL có tính diễn đạt cao
- Các ràng buộc có thể được thực thi bởi chính các mô-tơ SPARQL của CSDL
- Dễ dàng kết hợp với các ngôn ngữ luật khác như OWL RL

SPIN có thể định nghĩa các hàm SPARQL

spin:Function có thể được sử dụng để định nghĩa các hàm SPARQL mới với truy vấn SPARQL như thân hàm

spin:body ▾

```
★ SELECT (?width * ?height)
WHERE {
  ?arg1 ss:width ?width .
  ?arg1 ss:height ?height .
}
```

spin:constraint ▾

★ Argument sp:arg1 : ss:Rectangle

spin:returnType ▾

ⓓ xsd:integer

rdf:type ▾

● spin:Function

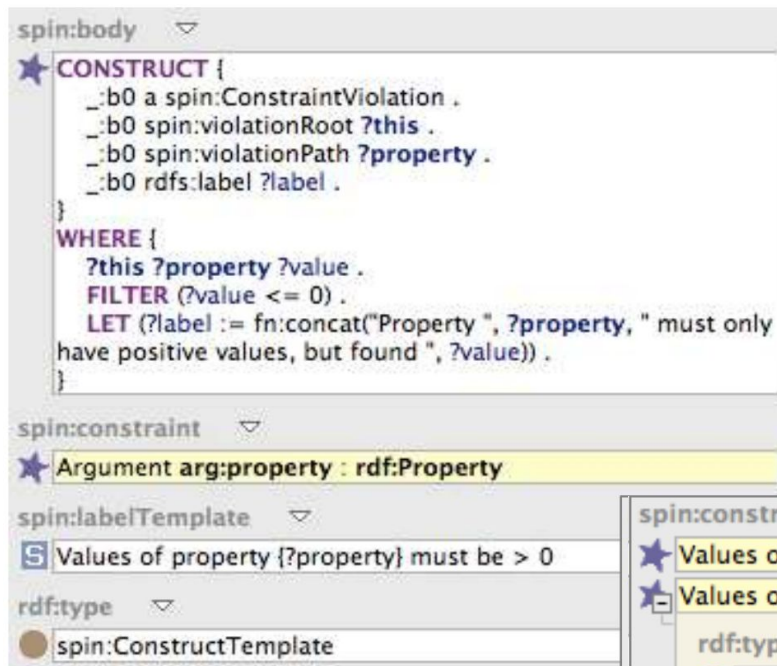
Các lợi ích:

- Cách tiếp cận mô hình hóa HĐT
- SPARQL có tính diễn đạt cao
- Các ràng buộc có thể được thực thi bởi chính các mô-tơ SPARQL của CSDL
- Dễ dàng kết hợp với các ngôn ngữ luật khác như OWL RL

```
CONSTRUCT {
  ?this ss:area ?area .
}
WHERE {
  LET (?area := ss:computeArea(?this)) .
}
```

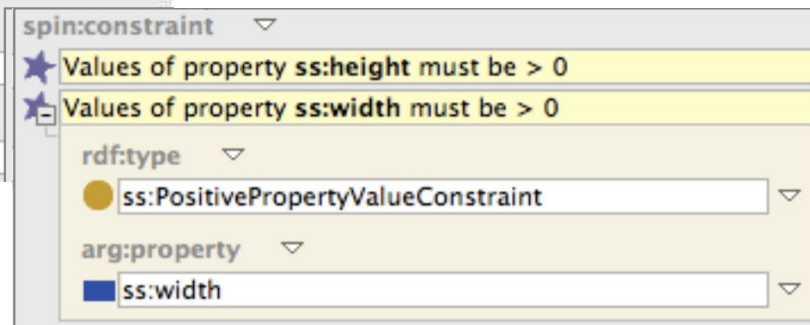
SPIN có thể tái sử dụng truy vấn SPARQL

spin:Template có thể được sử dụng để đóng gói các truy vấn SPARQL trong các đơn vị có thể được khởi tạo với các tham số để tái sử dụng



Các lợi ích:

- Hỗ trợ khả năng tái sử dụng các mẫu mô hình hóa theo HĐT
- Định nghĩa các ngôn ngữ chuyên dụng cho từng lĩnh vực
- Không cần chuyên môn về SPARQL để cung cấp các tham số.



SPIN và SHACL

SHACL - **SH**Apes **C**onstraint **L**anguage

- Quy chuẩn W3C để định nghĩa hình trạng của dữ liệu RDF
 - Có thể được hiểu như ngôn ngữ lược đồ cho RDF
 - Các hình được định nghĩa như những đồ thị RDF được gọi là các hình đồ thị
 - Đồ thị được kiểm tra được gọi là "Đồ thị dữ liệu"
 - Ảnh hưởng nhiều từ SPIN

Tham khảo

<https://spinrdf.org/spin-shacl.html>

<https://www.w3.org/TR/shacl/>

<https://jena.apache.org/documentation/shacl/>

<https://www.w3.org/TR/shacl-js/>

Thử khảo sát các triển khai SHACL ??

Dữ liệu

@prefix ex: <http://semweb.edu.vn/ex#> .

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

ex:Alice

 a ex:Person ;

 ex:ssn "987-65-432A" .

ex:Bob

 a ex:Person ;

 ex:ssn "123-45-6789" ;

 ex:ssn "124-35-6789" .

ex:Calvin

 a ex:Person ;

 ex:birthDate "1971-07-07"^^xsd:date ;

 ex:worksFor ex:UntypedCompany .

Ràng buộc

```
@prefix ex: <http://semweb.edu.vn/ex#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
ex:PersonShape
  a sh:NodeShape ;
  sh:targetClass ex:Person ;    # Applies to all persons
  sh:property [                # _:b1
    sh:path ex:ssn ;           # constrains the values of ex:ssn
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
    sh:pattern "^\\d{3}-\\d{2}-\\d{4}$" ;
  ] ;
  sh:property [                # _:b2
    sh:path ex:worksFor ;
    sh:class ex:Company ;
    sh:nodeKind sh:IRI ;
  ] ;
  sh:closed true ;
  sh:ignoredProperties ( rdf:type ) .
```

Nội dung

8.1. Ngôn ngữ SPARQL

8.2. Đại số SPARQL

8.3. Biểu diễn luật trong SPARQL: SPIN

8.4. SPARQL và OWL



Xử lý truy vấn với các kết quả suy diễn

<http://dbpedia.org/resource/Tim_Berners-Lee>

foaf:homepage

<http://www.w3.org/People/Berners-Lee/> .

foaf:name rdfs:subPropertyOf rdfs:label .

foaf:homepage a owl:InverseFunctionalProperty .

<http://dblp.l3s.de/d2r/page/authors/Tim_Berners-Lee>

foaf:homepage

<http://www.w3.org/People/Berners-Lee/> ;

foaf:name "Tim Berners-Lee".

SELECT ?P ?O

WHERE { <http://dbpedia.org/resource/Tim_Berners-Lee> rdfs:label ?O }

?O
"Tim Berners-Lee"

Các khó khăn

- Có thể có vô số kết quả
 - Do thuộc tính ContainerMembership (RDFS)
 - Suy diễn kiểu dữ liệu (OWL)
 - => Thiết lập các hạn chế phạm vi đối với kết quả
- Các biến không chính thức
 - Biến chính là biến có xuất hiện trong đầu ra và được lưu trong toàn bộ tiến trình xử lý
- Các tính năng SPARQL 1.1 khác, ví dụ, tổng hợp

Khả năng có vô số kết quả suy diễn

Đồ thị:

```
:rr2010Proceedings :hasEditors [ a rdf:Seq;  
    rdf:_1 :pascal_hitzler.  
    rdf:_2 :thomas_lukasiewicz.  
]
```

Truy vấn với suy diễn RDFS:

```
SELECT ?CM { ?CM a rdfs:ContainerMembershipProperty }
```

Các kết quả suy diễn RDFS (bộ ba mệnh đề):

```
rdfs:_1 a rdfs:ContainerMembershipProperty .  
rdfs:_2 a rdfs:ContainerMembershipProperty .  
rdfs:_3 a rdfs:ContainerMembershipProperty .  
...
```


Khả năng có vô số kết quả suy diễn₍₂₎

Đồ thị:

```
:rr2010Proceedings :hasEditors [ a rdf:Seq;  
    rdf:_1 :pascal_hitzler.  
    rdf:_2 :thomas_lukasiewicz.  
]
```

Truy vấn với suy diễn RDFS:

```
SELECT ?CM { ?CM a rdfs:ContainerMembershipProperty }
```

SPARQL 1.1 lọc các từ `rdf:_1`, `rdf:_2`, ..., `rdf:_n` không xuất hiện trong đồ thị dữ liệu.

Các kết quả trong SPARQL 1.1 là:

```
{ ?CM/rdfs:_1, ?CM/rdfs:_2, }
```

Các kết quả suy diễn kiểu dữ liệu trong OWL

Một cách phức tạp để mô tả Peter 50 tuổi trong OWL:

```
ex:Peter a [ a owl:Restriction ; owl:onProperty ex:age ;  
            owl:allValuesFrom [ rdf:type rdfs:Datatype .  
            owl:oneOf ("50"^^xsd:integer) ] ]
```

Một truy vấn phức tạp để tìm các giá trị số không phải tuổi của Peter:

```
SELECT ?x WHERE {  
  ex:Peter a [ a owl:Restriction ; owl:onProperty ex:age ;  
              owl:allValuesFrom [ a rdfs:Datatype ;  
              owl:datatypeComplementOf [ a rdfs:Datatype ; owl:oneOf (?x) ] ] ] }
```

Về mặt lý thuyết: Có vô số giá trị khác 50

SPARQL 1.1 giới hạn kết quả trong phạm vi {bộ từ vựng rdf + rdfs + owl} - {rdf:_1, ...rdf:_n} + {các từ xuất hiện trong đồ thị dữ liệu}

Các biến không chính thức

Cho đồ thị:

```
foaf:Person rdfs:subClassOf foaf:Agent .  
foaf:Person rdfs:subClassOf  
  [ a owl:Restriction ; owl:onProperty :hasFather ;  
    owl:someValuesFrom foaf:Person. ]  
foaf:knows rdfs:range foaf:Person.  
:jeff a Person  
:jeff foaf:knows :aidan
```

```
SELECT ?X ?Y { ?X :hasFather ?Y }
```

Không có kết quả, biến ?Y là biến chính thức trong trường hợp này.

Các biến không chính thức₍₂₎

Cho đồ thị:

```
foaf:Person rdfs:subClassOf foaf:Agent .
foaf:Person rdfs:subClassOf
  [ a owl:Restriction ; owl:onProperty :hasFather ;
    owl:someValuesFrom foaf:Person. ]
foaf:knows rdfs:range foaf:Person.
:jeff a Person . :jeff foaf:knows :aidan .
```

```
SELECT ?X { ?X :hasFather ?Y }
```

?Y giống như biến không chính thức trong trường hợp này.

Phép chiếu được thực hiện cuối cùng trên các kết quả xử lý BGP. SPARQL 1.1 coi tất cả biến như các biến chính thức.

=> Vẫn không có kết quả.

Suy diễn SPARQL 1.1 & các mẫu phức tạp

```
:person1 rdf:type [ owl:unionOf (:male :female) ]
```

```
SELECT ?X {  
  { ?X rdf:type :male }  
  UNION  
  { ?X rdf:type :female } }
```

Suy diễn SPARQL được định nghĩa sau so khớp BGP
=> Không có kết quả

Suy diễn SPARQL 1.1 và các tổng hợp

```
foaf:Person rdfs:subClassOf foaf:Agent .
foaf:Person rdfs:subClassOf [ a owl:Restriction ;
    owl:onProperty :hasFather ;
    owl:someValuesFrom foaf:Person. ]
:jeff a Person . :jeff foaf:knows :aidan .
foaf:knows rdfs:range foaf:Person .
:jeff :hasFather [a Person] .
:jeff owl:sameAs :aidan . }
```

Suy diễn
owl:sameAs không
ảnh hưởng đến
thống kê, nhưng
các nút rỗng có ảnh
hưởng.

Các tổng hợp được thực hiện sau so khớp BGP, vì vậy không có tác dụng

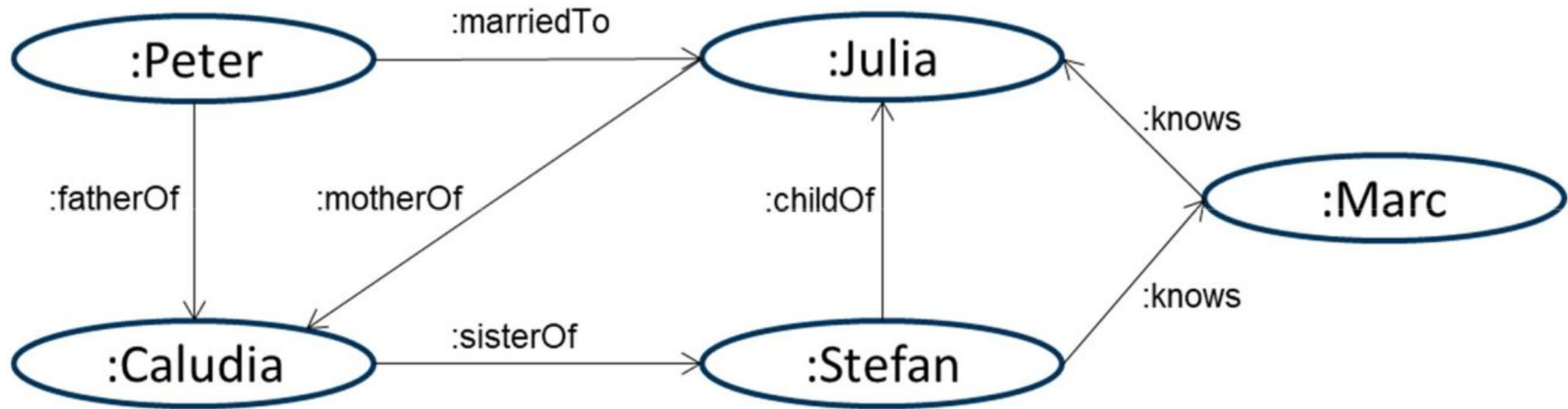
```
SELECT (Count(?X) AS ?Y) { ?X a foaf:Person }
```

Với ngữ nghĩa RDFS/OWL kết quả trả về là: {**?Y/3**}



Thử nghiệm

Viết các truy vấn SPARQL dựa trên đồ thị



1. Whom does Stefan know?
2. Who is a child of Julia?
3. Does Claudia have siblings?
4. Is Julia married?

Chuyển đồ thị thành tài liệu RDF và thực hiện các truy vấn với ARQ hoặc Apache Jena Fuseki?

Thử nghiệm₍₂₎

@prefix : <http://semweb.edu.vn/eg#>.

:Peter :fatherOf :Caludia;

:marriedTo :Julia.

:Julia :motherOf :Caludia.

:Caludia :sisterOf :Stefan.

:Stefan :childOf :Julia;

:knows :Marc.

:Marc :knows :Julia.

filename: ex187.ttl

@prefix d: <http://learningsparql.com/ns/data#> .

@prefix ab:

<http://learningsparql.com/ns/addressbook#> .

d:jane ab:hasParent d:gene .

d:gene ab:hasParent d:pat ;

ab:gender d:female .

d:joan ab:hasParent d:pat ;

ab:gender d:female .

d:pat ab:gender d:male .

d:mike ab:hasParent d:joan .

Viết câu truy vấn tìm ông của jane?

Viết câu truy vấn tìm ông nội của jane?

Kiểm tra xem gene và joan có phải chị em không?

Đếm số lượng anh chị em của gene?

@prefix ex: <http://eg.org/> .

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

ex:Hamlet ex:author ex:Shakespeare ; ex:price

"10.50"^^xsd:decimal .

ex:Macbeth ex:author ex:Shakespeare .

ex:Tamburlaine ex:author ex:Marlowe ; ex:price

"17"^^xsd:integer .

ex:DoctorFaustus ex:author ex:Marlowe ; ex:price

"12"^^xsd:integer ;

ex:title "The Tragical History of Doctor Faustus" .

ex:RomeoJulia ex:author ex:Brooke ; ex:price "9"^^xsd:integer .

-
- a. Thống kê số lượng sách được viết bởi mỗi tác giả?
 - b. Tìm tất cả sách được viết bởi Shakespeare hoặc ex:Brooke.
 - c. Tìm tất cả sách được viết bởi Shakespeare hoặc ex:Brooke và có giá < 15.
 - d. Thực hiện từng bước biểu thức đại số cho ý c.**
 - e. Tìm tất cả sách được viết bởi Shakespeare, sử dụng thuộc tính :**writtenBy**.