

Ngôn ngữ truy vấn ngữ nghĩa SPARQL

Truy vấn dữ liệu

- Tối quan trọng – trong việc sử dụng dữ liệu RDF
- ☐ Cho phép tổng hợp dữ liệu trong các kho chứa khác nhau
- Cho phép các dữ liệu được lựa chọn có thể được
 - ☐ sử dụng lại
 - ☐ biểu diễn lại
- XML và SPARQL

2

Cơ bản về SPARQL

- Thực hiện các truy vấn phức tạp trên dữ liệu RDF phân tán
- W3C recommendation 2008
- ☐ Ví dụ: “Tìm cặp tài nguyên (a,b), sao cho tồn tại x là cha của a và b là anh của x”
 - ☐ Kết quả mong đợi: cặp bác – cháu
- Cú pháp Turtle (N-triples)

3

Cú pháp Turtle

- Turtle = Terse RDF Triple Language
- Hỗ trợ namespace
- Cho phép gom nhóm các bộ ba theo Subject
- Ký pháp tắt cho tập nhiều phần tử

4

Cú pháp SPARQL: Triple

■ Triple đơn

- subject predicate object .

:john rdf:label "John" .

■ Triple ghép

- subject predicate object ; predicate object ...

```
:john
  rdf:label "John" ;
  rdf:type ex:Person ;
  ex:homePage http://example.org/johnspage/ .
```

5

prefixes

■ Cú pháp:

- @prefix abbr: <URI>

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

6

Định danh

■ URI

- <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

■ QName (Qualified name)

- namespace-abbr? :localname

■ Literal

- "John"
- true
- "Hello"@en-GB
- "1.4"^^xsd:decimal

7

Biến

■ ?x ?name ?test

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?mbox WHERE
{ ?x foaf:name "Johnny Lee Outlaw" .
  ?x foaf:mbox ?mbox }
```

8

Nút trắng

- Đơn:
- [] hoặc _:x
 - :john ex:hasFather [] .
 - :john ex:hasFather _:x .
- làm subject:
- [predicate object ; predicate object ...] .
 - [ex:hasName "John"] .
 - [ex:authorOf :lotr ; ex:hasName "Tolkien"] .

9

Nút trắng

- _:b c:speed ?v
- ?x c:speed []
- Nút trắng tương đương với một biến không được thể hiện trong kết quả truy vấn

10

Nút trắng

?x c:speed [rdf:value ?val ;
 c:unit 'km/h']

- Tương đương :

?x c:speed _:b
_:b rdf:value ?val
_:b c:unit 'km/h'

```
<c:Car>
  <c:speed rdf:parseType='Resource'>
    <rdf:value>100</rdf:value>
    <c:unit>km/h</c:unit>
  </c:speed>
</c:Car>
```

11

Tập nhiều đối tượng

- (object1... objectn)
 - :doc1 ex:hasAuthor (:john :mary) .
- Tương đương với
 - :doc1 ex:hasAuthor
 - [rdf:first :john;
 - rdf:rest [rdf:first :mary;
 - rdf:rest rdf:nil]
 -] .

12

Bộ ba

subject property object

```
<http://soict.hut.edu.vn/dungct>
  foaf:name ?name
```

```
?x foaf:name 'Dung'
```

```
?x ?p 'Dung'
_:b1 foaf:age 33
```

13

Cú pháp SPARQL

```
?x c:firstName ?y .
```

```
?x c:lastName ?z.
```

⇔

```
?x c:firstName ?y ;
```

```
c:lastName ?z
```

14

Cú pháp SPARQL

```
?x c:name "Hoang" .
```

```
?x c:name "Minh" .
```

⇔

```
?x c:name "Hoang" , "Minh"
```

15

Cú pháp SELECT

- Ấn định số thể hiện liên quan đến một thuộc tính

```
PREFIX dc: http://purl.org/dc/elements/1.1/
SELECT ?title
WHERE { <http://example.org/book/book1> dc:title ?title }
```

→ cơ chế Prefix – viết tắt URI

→ Các biến trả về

→ Mẫu truy vấn

16

SELECT

- Lựa chọn các biến
- Biến: `?x ?title ?name`
- Cú pháp
- `SELECT var1, var2,... var3`
 - `SELECT ?name`
 - `SELECT ?x,?title`

17

WHERE

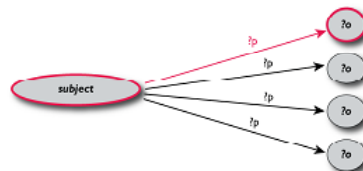
- Mẫu đồ thị đối sánh
- Tập các bộ ba
 - `{ (subject predicate object .)* }`
- Subject: URI, QName, nút trắng, Literal, Biến
- Predicate: URI, QName, nút trắng, biến
- Object: như Subject

18

Mẫu đồ thị (truy vấn)

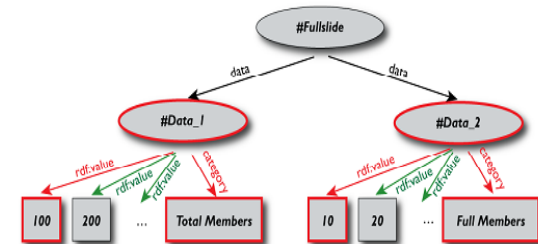
- Thông qua kết nối ký hiệu → truy vấn đồ thị con RDF
- VD:
- Kết quả: trả về các cặp p. o thỏa mãn

```
SELECT ?p ?o
WHERE {subject ?p ?o}
```



19

Ví dụ



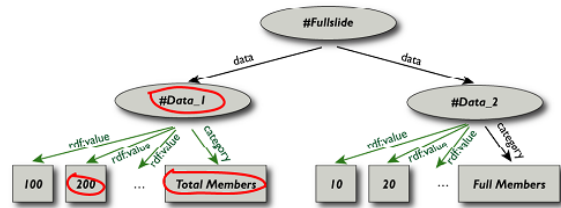
```
SELECT ?cat ?val
WHERE { ?x rdf:value ?val.
        ?x category ?cat }
```



```
[["Total Members",100],["Total
Members",200],... , ["Full
Members",10],...]
```

20

Ví dụ

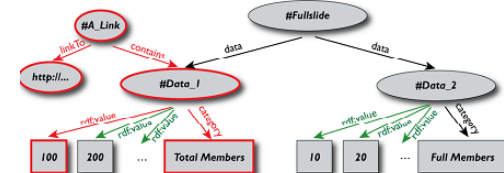


```
SELECT ?cat ?val
WHERE { ?x rdf:value ?val. {
  ?x category ?cat.
  FILTER(?val >= 200). }
```

[["Total Members", 200], ...]

21

Ví dụ

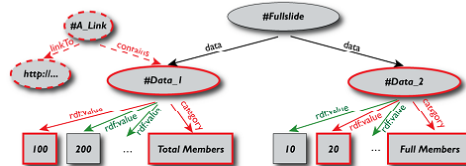


```
SELECT ?cat ?val ?uri
WHERE { ?x rdf:value ?val.
  ?x category ?cat.
  ?al contains ?x.
  ?al linkTo ?uri }
```

[["Total Members", 100, http:// , ...]

22

Ví dụ



```
SELECT ?cat ?val ?uri
WHERE { ?x rdf:value ?val.
  ?x category ?cat.
  OPTIONAL ?al contains ?x.
  ?al linkTo ?uri }
```

[["Total Members", 100, http://...], ...,
["Full Members", 20,], ...,]

23

OPTIONAL

```
SELECT * WHERE {
  ?x :hasCreated ?doc .
  OPTIONAL {
    ?x :isMemberOf ?org
  }
}
```

hasCreated: Bắt buộc
isMemberOf: tùy chọn

24

OPTIONAL

```
:John :hasCreated :d1
:John :hasCreated :d2
:Jack :hasCreated :d3
:Jack :isMemberOf :club
```

- Kết quả
- (1) x = :John ; doc = :d1 ; org = unbound
- (2) x = :John ; doc = :d2 ; org = unbound
- (3) x = :Jack ; doc = :d3 ; org = :club

25

OPTIONAL lồng nhau

```
SELECT * WHERE {
  ?x :hasCreated ?doc .
  OPTIONAL {
    ?x :isMemberOf ?org
    OPTIONAL {?org :name ?name}
  }
}
```

26

OPTIONAL lồng nhau

```
SELECT * WHERE {
  ?x :hasCreated ?doc .
  OPTIONAL {
    ?x :isMemberOf ?org
    OPTIONAL {?org :name ?name}
  }
}
```

Kết quả:

- (1) x = :John ; doc = :d1 ; org = unbound ; name = unbound
- (2) x = :John ; doc = :d2 ; org = unbound ; name = unbound
- (3) x = :Jack ; doc = :d3 ; org = :club ; name = unbound
- (4) x = :Jack ; doc = :d3 ; org = :assoc ; name = 'ASPTT'

```
:John :hasCreated :d1
:John :hasCreated :d2
:Jack :hasCreated :d3
:Jack :isMemberOf :club
:Jack :isMemberOf :assoc
:assoc :name 'ASPTT'
```

27

OPTIONAL lồng nhau

```
SELECT * WHERE {
  ?x :hasCreated ?doc .
  OPTIONAL {
    ?x :age ?age .
    ?x :isMemberOf ?org
  }
}
```

```
:John :hasCreated :d1
:John :hasCreated :d2
:Jack :hasCreated :d3
:Jack :isMemberOf :club
:Jim :hasCreated :d4
:Jim :isMemberOf :assoc
:Jim :age 45
```

age và isMemberOf phải cùng xuất hiện trong kết quả nếu có mặt.

28

OPTIONAL lồng nhau

```
SELECT * WHERE {
  ?x :hasCreated ?doc .
  OPTIONAL {
    ?x :age ?age .
    ?x :isMemberOf ?org
  }
}
```

```
:John :hasCreated :d1
:John :hasCreated :d2
:Jack :hasCreated :d3
:Jack :isMemberOf :club
:Jim :hasCreated :d4
:Jim :isMemberOf :assoc
:Jim :age 45
```

Kết quả:

- (1) x = :John ; doc = :d1 ; org = unbound ; age = unbound
- (2) x = :John ; doc = :d2 ; org = unbound ; age = unbound
- (3) x = :Jack ; doc = :d3 ; org = unbound ; age = unbound
- (4) x = :Jim ; doc = :d4 ; org = :assoc ; age = 45

29

OPTIONAL lồng nhau

```
SELECT * WHERE {
  ?x :hasCreated ?doc .
  OPTIONAL { ?x :age ?age }
  OPTIONAL { ?x :isMemberOf ?org }
}
```

```
:John :hasCreated :d1
:John :hasCreated :d2
:Jack :hasCreated :d3
:Jack :isMemberOf :club
:Jim :hasCreated :d4
:Jim :isMemberOf :assoc
:Jim :age 45
```

Kết quả:

- (1) x = :John ; doc = :d1 ; org = unbound ; age = unbound
- (2) x = :John ; doc = :d2 ; org = unbound ; age = unbound
- (3) x = :Jack ; doc = :d3 ; org = club ; age = unbound
- (4) x = :Jim ; doc = :d4 ; org = :assoc ; age = 45

30

Lọc kết quả

- **FILTER** (?age >= 7 && ?age <= 77)
- So sánh : < <= = >= > !=
- Toán tử : + * / -
- Logic : && (and) || (or) ! (not)
- Hàm : isBlank(?x) my:fun(?y)

31

Cú pháp lọc

- **EXP** ::= **TERM** | **EXP OPER EXP** | (**EXP**)
- **TERM** ::= **VAR** | **CST** | **FUN**
- **CST** ::= **URI** | **Literal**
- **OPER** ::= < <= = >= > + - * / && || !
- **FUN** ::= **NAME** (**EXP***)

32

Kiểu dữ liệu - hàm

- `1 = 1.0`
- `'1' ^^xsd:integer = '1.0' ^^xsd:decimal`
- `"engineer" != "engineer"@en`
- `"engineer" = str("engineer"@en)`

33

Hàm

- `isURI(?x)`
- `isLiteral(?y)`
- `isBlank(?z)`
- `bound(?t)`

34

Kiểm tra điều kiện

- `lang(?x) = 'en'`
 - `engineer@en -> true`
 - `kỹ sư@vn -> false`
- `datatype(?y) = xsd:string`
- `! (?x || ?y && ?z)`

35

Xử lý kết quả

- `SELECT * WHERE`
- `SELECT DISTINCT ?x ?y WHERE`
- `ORDER BY ?x DESC(?y)`
- `LIMIT 10`
- `OFFSET 10`

36

DISTINCT

```
select distinct ?x ?z
```

```
  where {
    ?x :friend ?y
    ?y :friend ?z
  }
```

- Không trả về 2 lần giá trị của x, z như nhau

37

DISTINCT

```
select distinct ?x ?z
```

```
  where {
    ?x :friend ?y
    ?y :friend ?z
  }
```

```
:Jules :friend :Jim
:Jim   :friend :Jack
:Jules :friend :James
:James :friend :Jack
```

■ Kết quả

- (1) x = :Jules ; z = :Jack

38

Không DISTINCT

```
Select ?x ?z
```

```
  where {
    ?x :friend ?y
    ?y :friend ?z
  }
```

```
:Jules :friend :Jim
:Jim   :friend :Jack
:Jules :friend :James
:James :friend :Jack
```

■ Kết quả

- (1) x = :Jules ; z = :Jack
- (2) x = :Jules ; z = :Jack

39

ORDER BY

```
select ?pers ?date
```

```
  where {
    ?pers :author ?doc
    ?doc :date ?date
  }
```

```
:Jim   :author :d2
:Jack  :author :d1
:d2    :date 2008-01-01
:d1    :date 2007-12-31
```

order by ?date

Kết quả:

- (1) pers = :Jim ; date = 2007-12-31
- (2) pers = :Jack ; date = 2008-01-01

40

ORDER BY

```
select ?doc ?date
  where {
    ?pers :author ?doc
    ?doc :date ?date
  }
order by ?date
        desc(?date)
```

```
:Jim   :author :d2
:Jack  :author :d1
:Jack  :author :d3
:d2    :date 2008-01-01
:d1    :date 2007-12-31
:d3    :date 2007-12-31
```

Kết quả:

- 1) doc = :d3 ; date = 2007-12-31
- (2) doc = :d1 ; date = 2007-12-31
- (3) doc = :d2 ; date = 2008-01-01

41

LIMIT/OFFSET

```
select * where {
  PATTERN
}
LIMIT 20
```

Chỉ hiện tối đa 20 kết quả

42

LIMIT/OFFSET

```
select * where {
  PATTERN
}
LIMIT 20
OFFSET 10
```

Chỉ hiện tối đa 20 kết quả, sau 10 kết quả đầu (Các kết quả từ 11-30)

43

Một số bài tập

- Viết query chuyển đổi ngôn ngữ nhãn của khái niệm person từ tiếng Anh sang tiếng Pháp – nếu có.

```
select ?label where
  ?x rdfs:label ?l@en
  ?l = "person"
  ?x rdfs:label ?label@fr
```

44

Lấy lớp cha của 1 tài nguyên

- Cho 1 resource – xác định lớp cha của lớp tương ứng với resource đó

select ?class where

?x rdf:type ?c

?c rdfs:subClassOf ?class

45

Thực hành truy vấn SPARQL

- Sparqler

□ <http://sparql.org/sparql.html>

- OpenLink's Virtuoso

□ <http://lod.openlinksw.com/sparql>

□ Chọn "Retrieve remote RDF data for all missing source graphs"

46

SPARQL

SPARQLer

SPARQLer - General purpose processor

General SPARQL query : input query, set any options and press "Get Results"

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name
WHERE {
  ?person foaf:name ?name .
}
```

SPARQL query

Dataset

Target graph URI (or use FROM in the query) <http://www.w3.org/People/Ber/>

Output XML: ☒ with XSLT style sheet (leave blank for none); /xml to html.xsl

or JSON output: ☐

or text output: ☐

Force the accept header to text/plain regardless ☐

Get Results

47

OpenLink Virtuoso

OpenLink Virtuoso SPARQL Query

This query page is designed to help you test OpenLink Virtuoso SPARQL protocol endpoint. Consult the [Virtuoso Wiki page](#) describing the service or the [Online Virtuoso Documentation](#) section [RDF Database and SPARQL](#).

There is also a rich Web based user interface with sample queries. [SPARQL](#)

Dataset

Query

Default Graph URI

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

Retrieve remote RDF data for all missing source graphs

Query text

```
SELECT ?name
WHERE {
  ?person foaf:name ?name .
}
```

SPARQL query

Display Results As: HTML

☒ Rigorous check of the query

Run Query

Reset

Dữ liệu RDF

- Tự tạo ontology – up lên một địa chỉ URI
- Ontology FOAF của Tim Berners Lee:
 - <http://www.w3.org/People/Berners-Lee/card>

49

Truy vấn 1

- Truy vấn 1: Tìm tất cả tên người trong contact FOAF trên

name
Dan Connolly
Henry Story
Timothy Berners-Lee
Norman Walsh
World Wide Web Consortium
Ralph R. Swick
Daniel Krech
Christoph Bussler
Nicholas Gibbins
Wendy Hall
Nigel Shadbolt
Les Carr
Charles McCallieNeville
Håkon Wium Lie
Peter Szolovits

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name
WHERE {
  ?person foaf:name ?name .
}
```

50

Truy vấn 2

- Tìm tất cả những người trong contact của foaf có tên và địa chỉ mail
- Sử dụng *

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT * WHERE {
  ?person foaf:name ?name .
  ?person foaf:mbox ?email .
}
```

51

Truy vấn 3

- Tìm các homepage của những người mà TBL (<http://www.w3.org/People/Berners-Lee/card>) biết



52

```

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX card: <http://www.w3.org/People/Berners-Lee/card#>
SELECT ?homepage
  FROM <http://www.w3.org/People/Berners-Lee/card>
 WHERE {
   card:i    foaf:knows    ?known .
   ?known    foaf:homepage ?homepage .
 }

```

53

Bài tập

Cho dữ liệu sau

```

@prefix foaf: <http://xmlns.com/foaf/0.1/>
_:a foaf:name "Anna" ;
foaf:mbox <mailto:anna@example.net> .
_:b foaf:name "Ioan" .

```

Cho biết kết quả trả về bởi câu SPARQL

```

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name WHERE {
  ?node foaf:name ?name .
  ?node foaf:mbox ?mbox .
}

```

54

Bài tập

Cho dữ liệu sau

```

@prefix dc10: <http://purl.org/dc/elements/1.0/> .
@prefix dc11: <http://purl.org/dc/elements/1.1/> .
_:a dc10:title "SPARQL Query Language Tutorial" .
_:b dc11:title "SPARQL Query Language (2nd ed)" .
_:c dc10:title "SPARQL" .
_:c dc11:title "SPARQL" .

```

Cho biết kết quả trả về bởi câu SPARQL

```

PREFIX dc10: <http://purl.org/dc/elements/1.0/>
PREFIX dc11: <http://purl.org/dc/elements/1.1/>
SELECT DISTINCT ?title
WHERE { { ?book dc10:title ?title } UNION
{ ?book dc11:title ?title } }

```

55

Bài tập

Cho CSDL RDF sau

```

D := { (⊔P1,rdf:type,Person), (⊔P1,name,"Pete"), (⊔P1,age,"17"), (⊔P1,email,"pete@abc.com"),
       (⊔P2,rdf:type,Person), (⊔P2,name,"John"), (⊔P2,email,"john@abc.com"),
       (⊔P3,rdf:type,Person), (⊔P3,name,"Sue"), (⊔P3,age,"21"),
       (⊔P1, knows, ⊔P2), (⊔P1, knows, ⊔P3), (⊔P2, knows, ⊔P1), (⊔P2, knows, ⊔P3) }.

```

Vẽ đồ thị RDF. Cho biết kết quả trả về bởi câu SPARQL

- ((?p,rdf:type,Person) AND (?p,age,?age)) FILTER (?age>20)
- ((?p,rdf:type,Person) AND (?p,name,?name)) OPT (?p,age,?age)
- ((?p,rdf:type,Person) OPT (?p,email,?email)) FILTER (!bound(?email))

56