

Linked data - SPARQL -

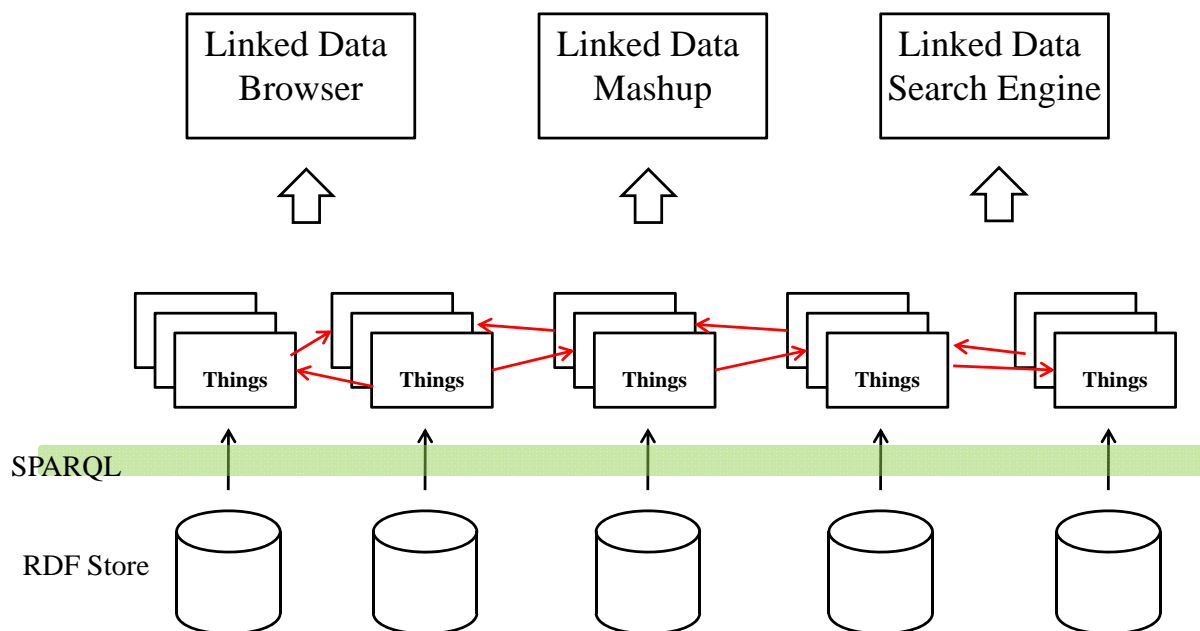
Hideaki Takeda
National Institute of Informatics
takeda@nii.ac.jp

I used some slides by Fumi Kato by his courtesy

Hideaki Takeda / National Institute of Informatics



How to use Linked Data

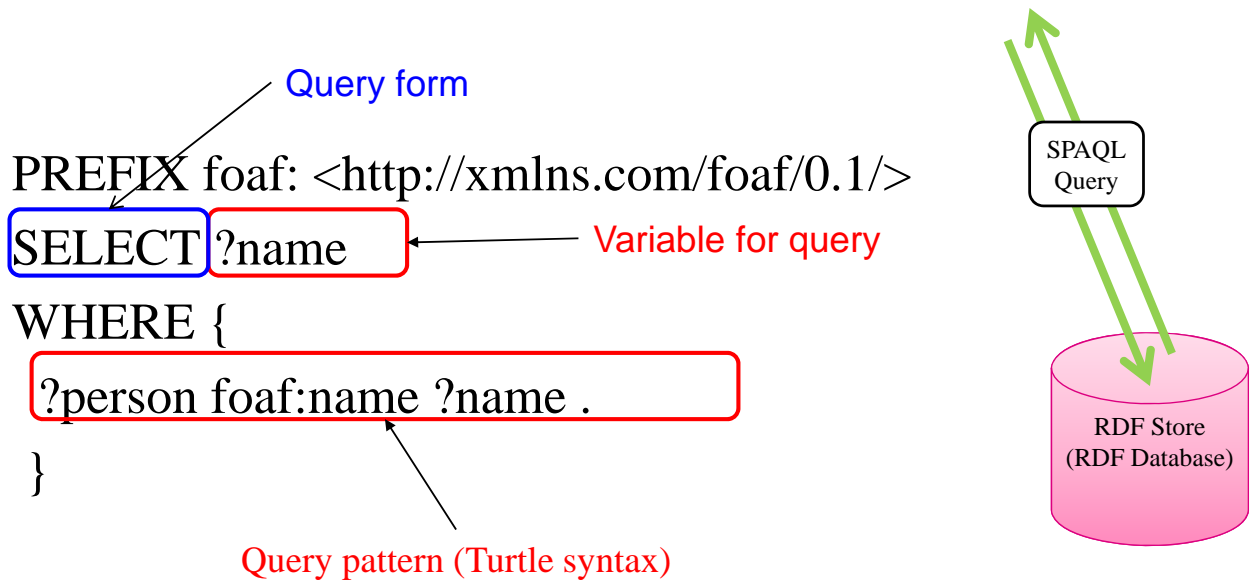


Hideaki Takeda / National Institute of Informatics



SPARQL

- Data query language for RDF
- Similar syntax with SQL



Query form and results

Query form	Results
SELECT, ASK	XML, (JSON)
CONSTRUCT, DESCRIBE	RDF Graph

CONSTRUCT

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
CONSTRUCT {
  ?person foaf:name ?name .
}
FROM <http://fumi.me/foaf.rdf>
WHERE {
  ?person foaf:name ?name .
}
```

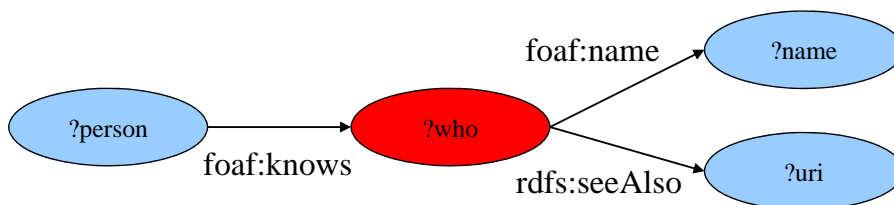
Multiple query patterns

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT * ← All variables
WHERE {
  ?person foaf:name ?name .
  ?person foaf:mbox ?email .
} } Matching multiple patterns
```

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

Connected graph

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?name ?uri
WHERE {
  ?person foaf:knows ?who .
  ?who foaf:name ?name;
      rdfs:seeAlso ?uri .
}
```



Optional values

- OK if it is not matched but return values if matched

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?name ?uri
WHERE {
  ?person foaf:knows ?who .
  ?who foaf:name ?name .
  OPTIONAL {
    ?who rdfs:seeAlso ?uri .
  }
}
```

Modifiers

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?name ?uri
WHERE {
  ?person foaf:knows ?who .
  ?who foaf:name ?name;
  OPTIONAL {
    rdfs:seeAlso ?uri .
  }
}
ORDER BY ?name
LIMIT 10
```

Ascending Sort
(if descending ORDER BY DESC(?name))

Upper bound on the number of solutions

UNION

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE {
  { ?person foaf:name ?name;
    foaf:mbox ?mbox .
  } UNION {
    ?person foaf:name ?name;
    foaf:mbox_sha1sum ?mbox .
  }
}
```

FROM

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
SELECT ?name
```

```
FROM <http://fumi.me/foaf.rdf>
```

```
WHERE {
```

```
  ?person foaf:name ?name .
```

```
}
```

Default dataset

Multiple Dataset

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
SELECT DISTINCT ?name
```

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

```
FROM <http://fumi.me/foaf.rdf>
```

```
FROM <http://www.tom.sfc.keio.ac.jp/~hagino/foaf.rdf>
```

```
WHERE {
```

```
  ?person foaf:name ?name .
```

```
}
```

重複削除

Merging dataset

Named Graph

- Distinguish data resources by URI/IRI
- Query for specifying Graphs

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
SELECT ?g ?name
```

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

```
FROM NAMED <http://fumi.me/foaf.rdf>
```

```
WHERE {
```

```
  GRAPH ?g {
```

```
    ?person foaf:name ?name .
```

```
  }
```

```
}
```

← Specifying graph resource

Default graph and Named graph

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
PREFIX rss: <http://purl.org/rss/1.0/>
```

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
```

```
SELECT ?name ?title ?link ?date
```

```
FROM <http://planetrdf.com/index.rdf>
```

```
FROM NAMED <http://fumi.me/foaf.rdf>
```

```
WHERE {
```

```
  ?item dc:creator ?name .
```

```
  ?item rss:title ?title .
```

```
  ?item rss:link ?link .
```

```
  ?item dc:date ?date .
```

```
  GRAPH <http://fumi.me/foaf.rdf> {
```

```
    ?me foaf:name "加藤文彦"@ja .
```

```
    ?me foaf:knows ?who .
```

```
    ?who foaf:name ?name .
```

```
  } .
```

```
}
```

FILTER: regex

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT ?name ?email
FROM <http://fumi.me/foaf.rdf>
WHERE {
    ?person foaf:name ?name;
           foaf:mbox ?email .
    FILTER regex(str(?email), "w3.org", "i")
}
```

FILTER: xsd:Date

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rss: <http://purl.org/rss/1.0/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
SELECT ?name ?title ?link ?date
FROM <http://planetrdf.com/index.rdf>
WHERE {
    ?item dc:creator ?name .
    ?item rss:title ?title .
    ?item rss:link ?link .
    ?item dc:date ?date .
    FILTER ( xsd:dateTime(?date) > "2008-03-31T00:00:00Z"^^xsd:dateTime )
}
```

Or compare as string **FILTER (str(?date) > "2008-03-31"**)

FILTER

- Logical connectives: !, &&, ||
- Arithmetic connectives: +, -, *, /
- Test: =, !=, >, <, ...
- Unary Test: isURI, isBlank, isLiteral, BOUND
- Data type: xsd:integer, xsd:decimal, xsd:float, xsd:double, xsd:string, xsd:boolean, xsd:dateTime
- Language Tag: @EN, @JP
- regex, str

Language tag

- Specifying language for literals
 - Turtle: “Tokyo”@en
 - RDF/XML: xml:lang=“en“
- FILTER
 - FILTER (lang(?name)=“en”)
- Note: literal with language tag is not equivalent to one without language tag
 - FILTER (?city = “Tokyo”)
 - ◆ No match with ”Tokyo”@en

Data type

- Types for literals
 - “1.3”^^xsd:decimal
 - “2008-05-01T09:12:58Z”^^xsd:dateTime
 - “true”^^xsd:boolean
 - “SFC”^^xsd:string
- Casting
 - Casting may be needed for applying operators.
 - ◆ FILTER regex(str(?title), “Semantic”)
 - ◆ FILTER xsd:int(?age) > 19
 - ◆ FILTER

DERI Pipes

- RDF-version of Yahoo! Pipes

<http://pipes.deri.org/>

The screenshot displays the DERI Pipes Designer interface. On the left, a list of published pipes is shown, including 'HEG/DBPEDIA/get_skos_narrower', 'HEG/DBPEDIA/get_skos_related', 'Idd/DBPediaCompanyLinks', 'Idd/GetDbpediaLocationIsNewYorkCity', 'Idd/New York Merger Acquisition', 'Idd/open calais beta', and 'Idd/rss2rdf'. The main workspace is titled 'Designer' and shows a workflow diagram for a pipe with ID 'HEG/DBPEDIA/get_...'. The workflow consists of several interconnected components: a 'Parameter' component with 'Label: Term', 'Name: Term', and 'Default: Knowledge'; a 'URL builder' component with 'Base: http://dbpedia.org/data/' and 'Path elements', 'Text [wired]', and 'Query parameters'; an 'RDF Fetch' component with 'URL: [wired]' and 'Format: RDF/XML'; and a 'Select' component with 'Query: select ?o where'. An 'Output' component is also present. The interface includes a 'File' menu, 'Debug', and 'Help' options. At the bottom, there are 'text view' and 'table view' tabs, and a page number '18 / 18'.

DERI Pipes Operators

- SELECT
 - Just similar to SPARQL
 - ex.
- `select ?uri where { ?s <http://xmlns.com/foaf/0.1/knowns?> ?uri } ORDER by desc(?uri) LIMIT 2 offset 10`

Usage

- Use Firefox (or Chrome)
- Access Editor
 - <http://pipes.deri.org:8080/pipes/>
- Retrieve pipe
 - <http://pipes.deri.org:8080/pipes/?pipeid=TBLonTheSW>
- Run pipe
 - <http://pipes.deri.org:8080/pipes/pipes/?id=TBLonTheSW>
- My examples
 - <http://pipes.deri.org:8080/pipes/?pipeid=takeda-test-03>

Assignment

- Build some DERI Pipes application
 - Mix two or more resources
- Deadline: August 31, 2011
- Report & Application