BASI DI DATI II – 2 modulo
Parte VIII: SPARQL

Prof. Riccardo Torlone

Università Roma Tre

Outline

- Querying RDF
- SPARQL

Query Languages: SQL

- A language for querying collections of tuples:

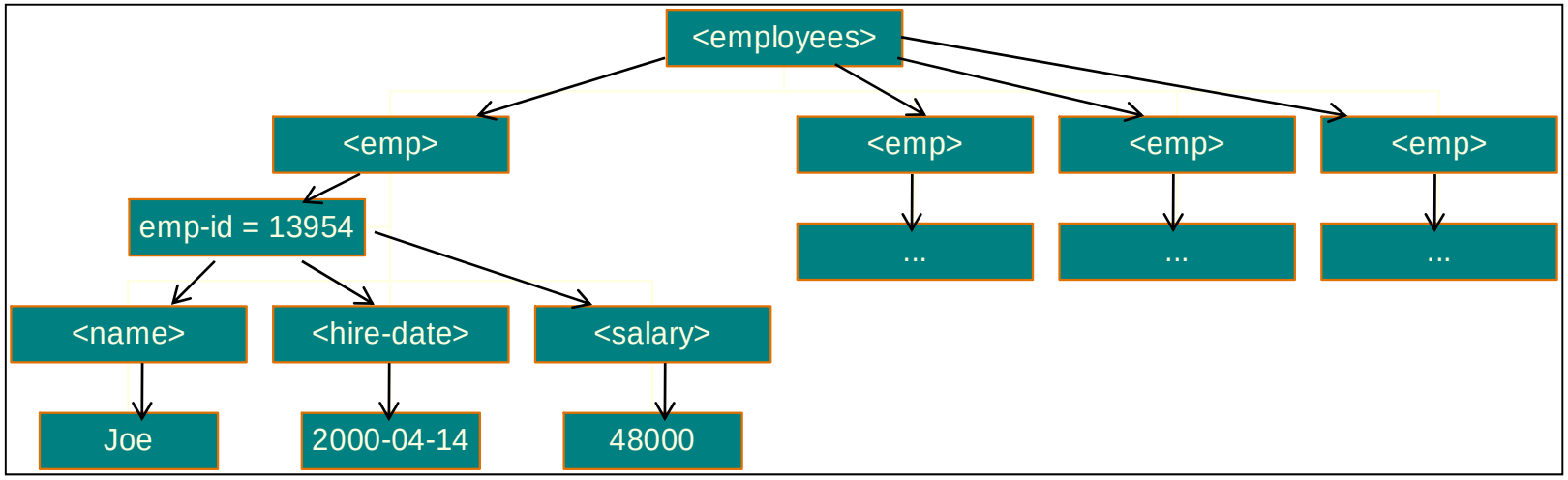
```
SELECT SALARY, HIRE_DATE
FROM EMPS
WHERE EMP_ID = 13954
```

EMP_ID	NAME	HIRE_DATE	SALARY
13954	Joe	2000-04-14	48000
10335	Mary	1998-11-23	52000
…	…	…	…
04182	Bob	2005-02-10	21750

Query Languages: XQuery

- A language for querying trees of XDM nodes:

```
for $e in fn:doc("my_employees.xml”)
where $e//emp/@emp-id = 13954
return $e//emp/salary
```



Why an RDF Query Language?

- XML at a lower level of abstraction than RDF
- There are various ways of syntactically representing an RDF statement in XML
- Thus we would require several XQuery queries, e.g.
  - //uni:lecturer/uni:title if uni:title element
  - //uni:lecturer/@uni:title if uni:title attribute
  - Both XML representations equivalent!

Families of SQL-like langages for RDF(S)

- RDQL
  - Implementations: Jena, Sesame, RDFStore, ...
- RQL
  - Implementations: RQL, SPARQL, ...
- SPARQL adopted by W3C

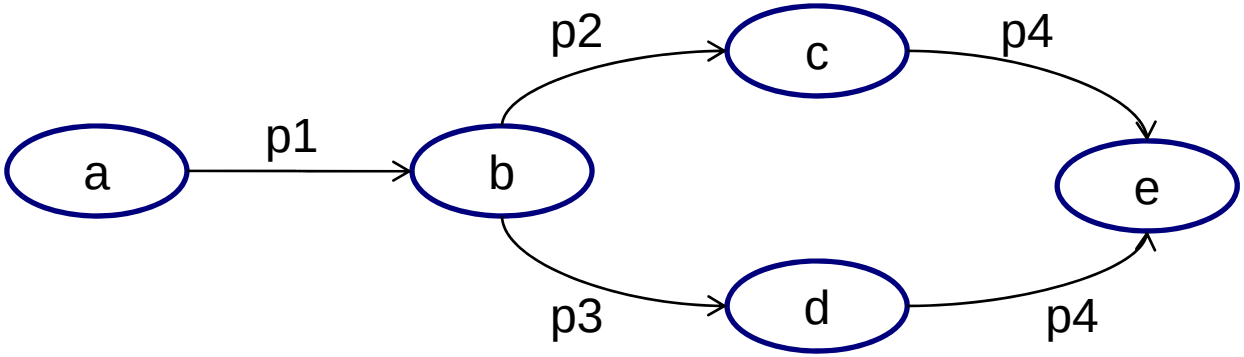
SPARQL

- SPARQL is a recursive acronym standing for
  - SPARQL Protocol and RDF Query Language
- It provides facilities to:
  - extract information in the form of URIs, blank nodes, plain and typed literals.
  - extract RDF subgraphs.
  - construct new RDF graphs based on information in the queried graphs

RDF and graphs

- A set of RDF statements can be always represented as a graph

subject	property	object
a	p1	b
b	p2	c
b	p3	d
c	p4	e
d	p4	e



SPARQL query

- Basic format:

  SELECT list of **variables**

  WHERE { a **graph pattern** with variables }

  - variables: any identifier preceded by “?”
  - graph pattern: “describes” a graph
- Example:

  SELECT ?x ?y

  WHERE { ?x p4 ?y }
- Intuitive semantics: any binding of the variables such that the graph pattern in the query matches a sub-graph of the underlying RDF graph
- Query result:

  c e

  d e

Graph Patterns

- Basic Graph Pattern
  - a set of triple patterns
- Group Pattern
  - a set of graph patterns: must all match
- Value Constraints
  - restrict RDF terms in a solution
- Optional Graph Patterns
  - additional patterns may extend the solution
- Alternative Graph Pattern
  - two or more possible patterns are tried
- Patterns on Named Graphs
  - patterns are matched against named graphs

Graph Patterns

- Basic Graph Pattern
  - a set of triple patterns
- Group Pattern
  - a set of graph patterns: must all match
- Value Constraints
  - restrict RDF terms in a solution
- Optional Graph Patterns
  - additional patterns may extend the solution
- Alternative Graph Pattern
  - two or more possible patterns are tried
- Patterns on Named Graphs
  - patterns are matched against named graphs

Basic Graph Pattern

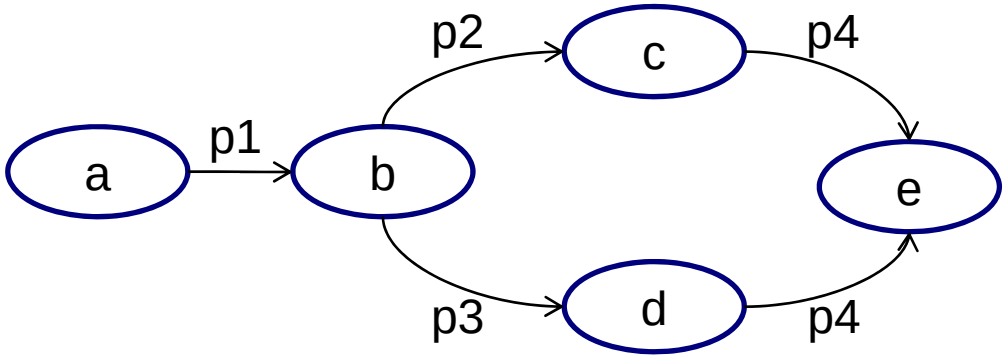
- Basic Graph Pattern (BGP): set of triple patterns
- Triple Pattern: similar to an RDF Triple (subject, predicate, object), but any component can be a query variable (preceded by “?”); literal (constant) subjects are allowed:

  ?book dc:title ?title

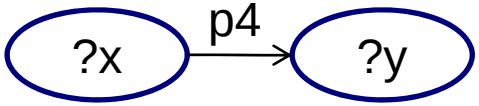
- Matching a triple pattern to a graph: bindings between variables and RDF Terms
- Matching a BGP to a graph: a **Pattern Solution** of BGP P on an RDF graph G is any substitution S such that S(P) is a subgraph of G.

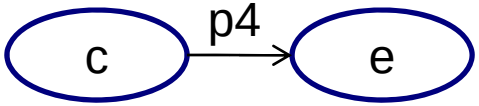
Example

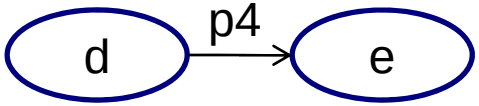
subject	property	object
a	p1	b
b	p2	c
b	p3	d
c	p4	e
d	p2	e



SELECT ?x ?y
WHERE { ?x p4 ?y }







Basic Graph Pattern - Multiple Matches

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
 _:a foaf:name "Johnny Lee Outlaw" .
_:a foaf:mbox <mailto:jlow@example.com> .
_:b foaf:name "Peter Goodguy" .
_:b foaf:mbox <mailto:peter@example.org> .
```

Data

Query

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

name	mbox
"Johnny Lee Outlaw"	<mailto:jlow@example.com>
"Peter Goodguy"	<mailto:peter@example.org>

Query Result

Basic Graph Pattern - Blank Nodes

Data

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
_:a foaf:name "Alice" .
_:b foaf:name "Bob" .
```

Query

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

Query Result

x	name
_:c	“Alice“
_:d	“Bob”

Graph Patterns

- Basic Graph Pattern
  - a set of triple patterns
- Group Pattern
  - a set of graph patterns: must all match
- Value Constraints
  - restrict RDF terms in a solution
- Optional Graph Patterns
  - additional patterns may extend the solution
- Alternative Graph Pattern
  - two or more possible patterns are tried
- Patterns on Named Graphs
  - patterns are matched against named graphs

Group Pattern

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

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

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

Graph Patterns

- Basic Graph Pattern
  - a set of triple patterns
- Group Pattern
  - a set of graph patterns: must all match
- Value Constraints
  - restrict RDF terms in a solution
- Optional Graph Patterns
  - additional patterns may extend the solution
- Alternative Graph Pattern
  - two or more possible patterns are tried
- Patterns on Named Graphs
  - patterns are matched against named graphs

Value Constraints

Data

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix : <http://example.org/book/> .
@prefix ns: <http://example.org/ns#> .
:book1 dc:title "SPARQL Tutorial" .
:book1 ns:price 42 .
:book2 dc:title "The Semantic Web" .
:book2 ns:price 23 .
```

Query

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX ns: <http://example.org/ns#>
SELECT ?title ?price
WHERE { ?x ns:price ?price .
        FILTER ?price < 30 .
        ?x dc:title ?title . }
```

Query Result

title	price
"The Semantic Web"	23

Regular expressions can be used

```
PREFIX dc:    <http://purl.org/dc/elements/1.1/>
PREFIX ldap:  <http://ldap.hp.com/people#>
PREFIX foaf:

SELECT ?name ?name2
{
   ?doc          dc:title     ?title .
   FILTER regex(?title, “SPARQL”) .
   ?doc          dc:creator   ?reseacher .
   ?researcher   ldap:email      ?email .
   ?researcher   ldap:name       ?name
}
```

- “Find the name and email addresses of authors of a paper about SPARQL”

Graph Patterns

- Basic Graph Pattern
  - a set of triple patterns
- Group Pattern
  - a set of graph patterns: must all match
- Value Constraints
  - restrict RDF terms in a solution
- Optional Graph Patterns
  - additional patterns may extend the solution
- Alternative Graph Pattern
  - two or more possible patterns are tried
- Patterns on Named Graphs
  - patterns are matched against named graphs

Optional graph patterns

Data

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix : <http://example.org/book/> .
@prefix ns: <http://example.org/ns#> .
:book1 dc:title "SPARQL Tutorial" .
:book1 ns:price 42 .
:book2 dc:title "The Semantic Web" .
:book2 ns:price 23 .
```

Query

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX ns: <http://example.org/ns#>
SELECT ?title ?price
WHERE { ?x dc:title ?title .
        OPTIONAL { ?x ns:price ?price .
                   FILTER ?price < 30 } }
```

Query Result

title	price
“SPARQL Tutorial“	
"The Semantic Web"	23

Multiple Optional Blocks

Data

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
_:a foaf:name "Alice" .
_:a foaf:homepage <http://work.example.org/alice/> .
_:b foaf:name "Bob" .
_:b foaf:mbox <mailto:bob@work.example> .
```

Query

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox ?hpage
WHERE { ?x foaf:name ?name .
        OPTIONAL { ?x foaf:mbox ?mbox }.
        OPTIONAL { ?x foaf:homepage ?hpage } }
```

Query Result

name	mbox	hpage
“Alice“		<http://work.example.org/alice/>
“Bob“	<mailto:bob@example.com>	

Graph Patterns

- Basic Graph Pattern
  - a set of triple patterns
- Group Pattern
  - a set of graph patterns: must all match
- Value Constraints
  - restrict RDF terms in a solution
- Optional Graph Patterns
  - additional patterns may extend the solution
- Alternative Graph Pattern
  - two or more possible patterns are tried
- Patterns on Named Graphs
  - patterns are matched against named graphs

Alternative Graph Patterns

Data

```
@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 Protocol Tutorial" .
_:c dc10:title "SPARQL" .
_:c dc11:title "SPARQL (updated)" .
```

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

Query

x
"SPARQL Query Language Tutorial"
"SPARQL"
"SPARQL Protocol Tutorial"
"SPARQL (updated)"

Query Result

Graph Patterns

- Basic Graph Pattern
  - a set of triple patterns
- Group Pattern
  - a set of graph patterns: must all match
- Value Constraints
  - restrict RDF terms in a solution
- Optional Graph Patterns
  - additional patterns may extend the solution
- Alternative Graph Pattern
  - two or more possible patterns are tried
- Patterns on Named Graphs
  - patterns are matched against named graphs

RDF Dataset

- RDF data stores may hold multiple RDF graphs:
  - RDF Dataset in SPARQL terminology
  - each graph provides some information
  - different parts of the query may be matched against different graphs
- RDF Dataset
  - one graph, the default graph, which does not have a name
  - zero or more named graphs, each identified by URI
  - { G, (<u_1>, G_1), (<u_2>, G_2), . . . (<u_n>, G_n) }
- A graph pattern P matches an RDF dataset DS with solution S if P matches G (the default graph of DS) with solution S.

Named and Default Graphs

```
# Default graph
@prefix dc: <http://purl.org/dc/elements/1.1/> .
<http://example.org/bob> dc:publisher "Bob" .
<http://example.org/alice> dc:publisher "Alice" .
```

```
# Named graph: http://example.org/bob
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
_:a foaf:name "Bob" .
_:a foaf:mbox <mailto:bob@oldcorp.example.org> .
```

```
# Named graph: http://example.org/alice
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
_:a foaf:name "Alice" .
_:a foaf:mbox <mailto:alice@work.example.org> .
```

Merging the Named Graphs

```
# Deafult graph
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
_:x foaf:name "Bob" .
_:x foaf:mbox <mailto:bob@oldcorp.example.org> .
_:y foaf:name "Alice" .
_:y foaf:mbox <mailto:alice@work.example.org> .
```

```
# Named graph: http://example.org/bob
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
_:a foaf:name "Bob" .
_:a foaf:mbox <mailto:bob@oldcorp.example.org> .
```

```
# Named graph: http://example.org/alice
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
_:a foaf:name "Alice" .
_:a foaf:mbox <mailto:alice@work.example.org> .
```

Querying the Dataset

```
# Named graph: http://example.org/foaf/aliceFoaf
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
_:a foaf:name "Alice" .
_:a foaf:mbox <mailto:alice@work.example> .
_:a foaf:knows _:b .
_:b rdfs:seeAlso <http://example.org/foaf/bobFoaf> .
<http://example.org/foaf/bobFoaf> rdf:type foaf:PersonalProfileDocument .
_:b foaf:name "Bob" .
_:b foaf:mbox <mailto:bob@work.example> .
_:b foaf:age 32 .
```

```
# Named graph: http://example.org/foaf/bobFoaf
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
 _:1 foaf:mbox <mailto:bob@work.example> .
 _:1 rdfs:seeAlso <http://example.org/foaf/bobFoaf> .
_:1 foaf:age 35 .
<http://example.org/foaf/bobFoaf> rdf:type foaf:PersonalProfileDocument .
```

Accessing Graph Labels

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?src ?bobAge
WHERE { GRAPH ?src
          { ?x foaf:mbox <mailto:bob@work.example> .
            ?x foaf:age ?bobAge }
        }
```

src	bobAge
<http://example.org/foaf/aliceFoaf>	32
<http://example.org/foaf/bobFoaf>	35

Restricting by Graph Label

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX data: <http://example.org/foaf/>
SELECT ?age
WHERE
    {
        GRAPH data:bobFoaf  {
                ?x foaf:mbox <mailto:bob@work.example> .
                ?x foaf:age ?age  }
     }
```

age
35

Restricting via Query Pattern

```
PREFIX data: <http://example.org/foaf/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?mbox ?age ?ppd
 WHERE
     { GRAPH data:aliceFoaf
          { ?alice foaf:mbox <mailto:alice@work.example> ;
                   foaf:knows ?whom .
            ?whom foaf:mbox ?mbox ;
                   rdfs:seeAlso ?ppd .
            ?ppd a foaf:PersonalProfileDocument . } .
       GRAPH ?ppd { ?w foaf:mbox ?mbox ;
                          foaf:age ?age } }
```

mbox	age	ppd
<mailto:bob@work.example>	35	<http://example.org/foaf/bobFoaf>

Constructing an Output Graph

Data

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
_:a foaf:givenname "Alice" .
_:a foaf:family_name "Hacker" .
_:b foaf:firstname "Bob" .
_:b foaf:surname "Hacker" .
```

Query result

```
@prefix vcard:
    <http://www.w3.org/2001/vcard-rdf/3.0#> .
_:v1 vcard:N        _:x .
 _:x vcard:givenName  "Alice" .
 _:x vcard:familyName "Hacker" .
 _:v2 vcard:N        _:z .
 _:z vcard:givenName  "Bob" .
_:z vcard:familyName "Hacker" .
```

Query

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX vcard:
   <http://www.w3.org/2001/vcard-rdf/3.0#>
CONSTRUCT
{
   ?x vcard:N _:v .
   _:v vcard:givenName ?gname .
   _:v vcard:familyName ?fname
}
WHERE
{
            { ?x foaf:firstname ?gname }
UNION
            { ?x foaf:givenname ?gname } .
            { ?x foaf:surname ?fname }
UNION
            { ?x foaf:family_name ?fname } .
}
```

ASK – A Boolean query

Data

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
_:a foaf:name "Alice" .
_:a foaf:homepage <http://work.example.org/alice/> .
_:b foaf:name "Bob" .
_:b foaf:mbox <mailto:bob@work.example> .
```

Query

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
ASK { ?x foaf:name "Alice" }
```

yes

Query result

SPARQL Serialization

```
PREFIX foaf: http://xmlns.com/foaf/0.1/
PREFIX ex: <http://ns.example.org/#>
SELECT ?x ?hpage ?name ?mbox
                    ?age ?blurb ?friend
FROM <data.n3> WHERE
{
    ?x foaf:name ?name .
    ?x foaf:mbox ?mbox .
    ?x foaf:homepage ?hpage .
    ?x foaf:knows ?friend .
    OPTIONAL { ?x ex:ageInYears ?age } .
    OPTIONAL { ?x ex:blurb ?blurb }
}
ORDER BY ?name
```

XML for SPARQL

```
<?xml version="1.0"?>
<sparql xmlns=http://www.w3.org/2005/sparql-results#
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.w3.org/2001/sw/DataAccess/rf1/result2.xsd">
<head>
  <variable name="x"/> <variable name="hpage"/> <variable name="name"/>
  <variable name="mbox"/> <variable name="age"/> <variable name="blurb"/>
  <variable name="friend"/> <link href="example.rq" />
</head>
<results>
  <result>
    <binding name="x"><bnode>r1</bnode></binding>
    <binding name="hpage"><uri>http://work.example.org/alice/</uri></binding>
    <binding name="name"><literal>Alice</literal></binding>
    <binding name="mbox"><literal></literal></binding>
    <binding name="friend"><bnode>r2</bnode></binding>
    <binding name="blurb"><literal ….>…</literal></binding>
  </result>
      …….
</results>
</sparql>
```

Conclusions

- Some SPARQL features:
  - Once the schemas are defined elsewhere in RDF/RDF(S) too, they can also be queried upon!
  - Queries built similarly to SQL...
  - Many facilities: regex, datatypes, functions,...
  - Queries over graphs (in opposition to flat relational DBs).
  - Implicit and explicit joins

Essential Online Resources

- http://www.w3.org/standards/semanticweb/
- http://www.w3.org/standards/semanticweb/query
- http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/