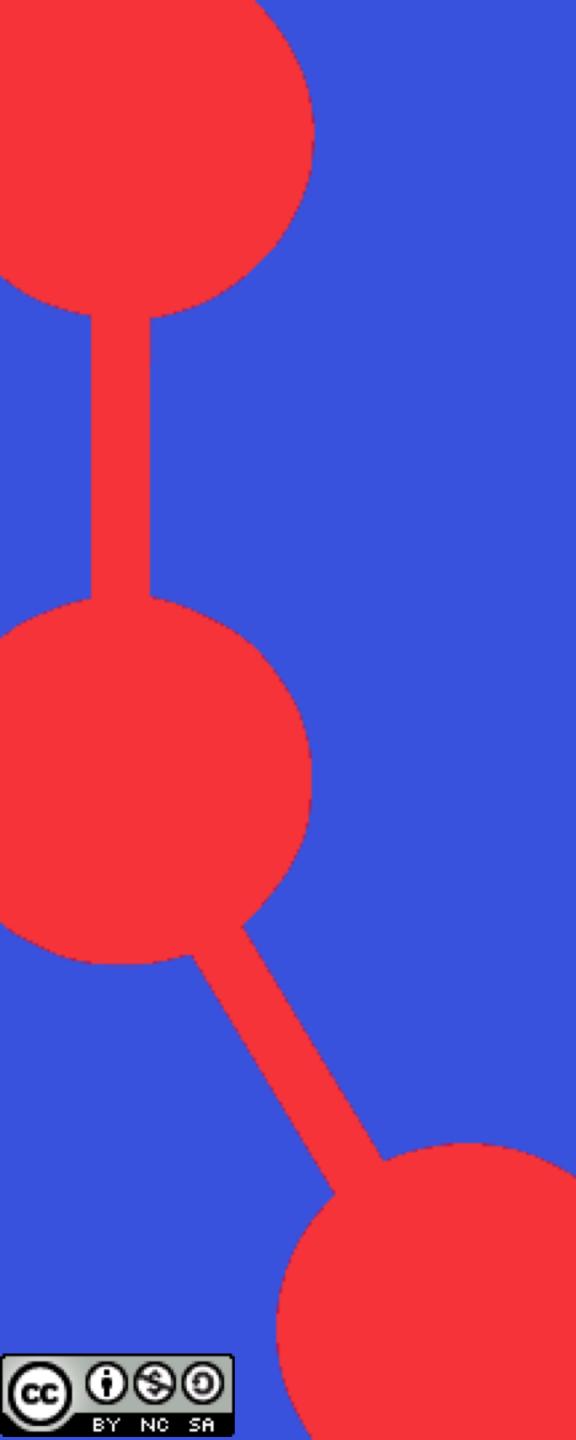


SPARQL

Raúl García-Castro, Óscar Corcho

Facultad de Informática, Universidad Politécnica de Madrid
Campus de Montegancedo sn, 28660 Boadilla del Monte, Madrid

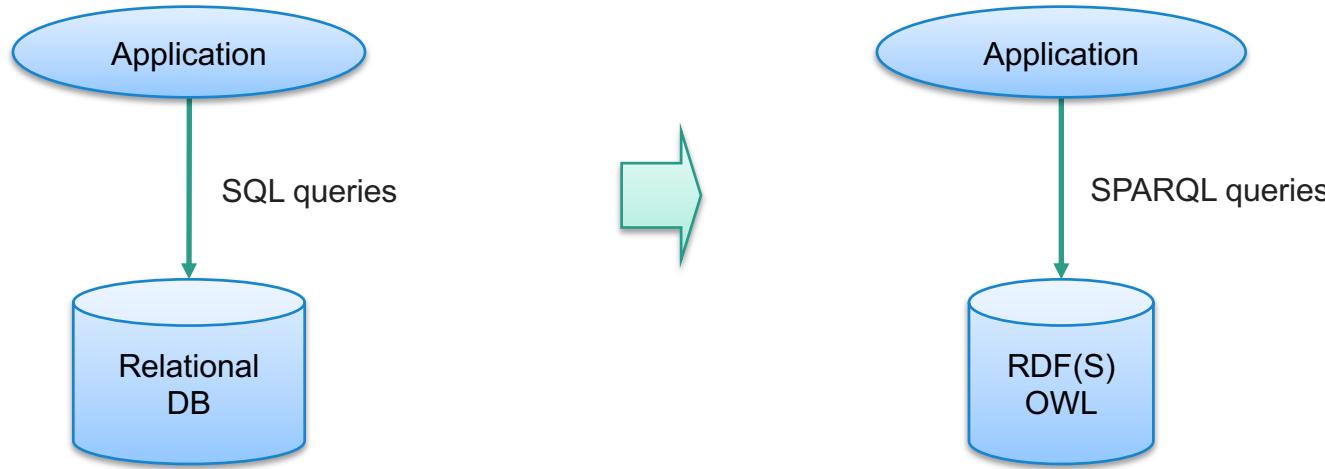
<http://www.oeg-upm.net>
{rgarcia,ocorcho}@fi.upm.es



- This work is licensed under the license CC BY-NC-SA 4.0 International
 - <http://purl.org/NET/rdflicense/cc-by-nc-sa4.0>
- You are free:
 - to Share — to copy, distribute and transmit the work
 - to Remix — to adapt the work
- Under the following conditions
 - Non-commercial — You cannot use it for commercial purposes, nor for training inside a commercial company
 - Attribution — You must attribute the work by inserting
 - “[source <http://www.oeg-upm.net/>]” at the footer of each reused slide
 - a credits slide stating: “These slides are partially based on “RDF, RDF Schema y SPARQL” by R. García-Castro, O. Corcho”
 - Share-Alike



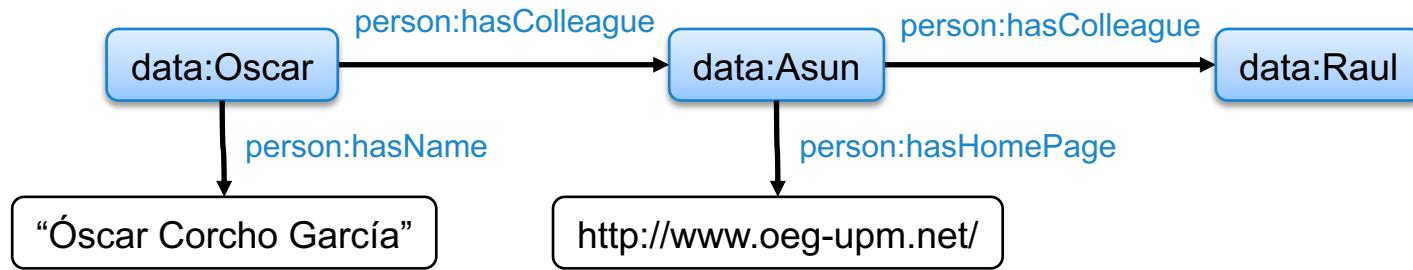
Querying over RDF. The Basics



- Supported by the most relevant APIs and triple stores
 - Jena, Sesame, etc.
 - Virtuoso, Stardog, 4store, Fuseki, etc.
 - ...

Querying over RDF. Examples (I)

- Sample RDF graph



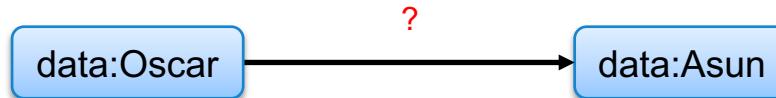
- **Query:** “Give me the persons that have Raúl as a colleague”



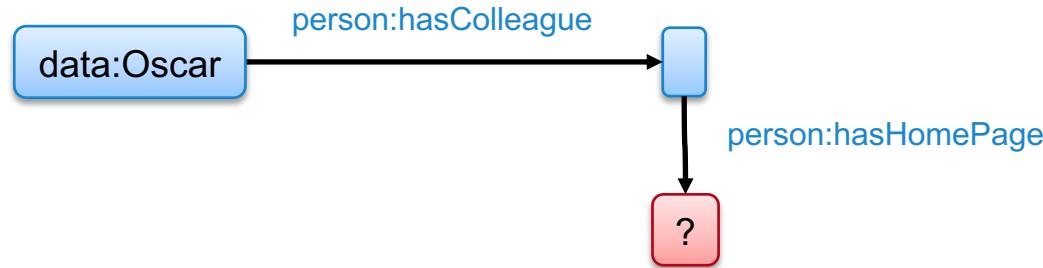
- **Result:** data:Asun

Querying over RDF. Examples (II)

- **Query:** “Which relations do exist between Oscar and Asun?”



- **Result:** person:hasColleague
- **Consulta:** “Give me the web pages of Oscar’s colleagues”



- **Result:** “<http://www.oeg-upm.net/>”

SPARQL is also a protocol

SPARQL is a Query Language ...

Find names and websites of contributors to PlanetRDF:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?website
FROM <http://planetrdf.com/bloggers.rdf>
WHERE {
    ?person foaf: weblog ?website .
    ?person foaf: name ?name .
    ?website a foaf: Document }
```

... and a Protocol

<http://.../qps?query-lang=http://www.w3.org/TR/rdf-sparql-query/&graph-id=http://planetrdf.com/bloggers.rdf&query=PREFIXfoaf:<http://xmlns.com/foaf/0.1/...>

This comprises:

- Services evaluating SPARQL queries over an RDF dataset (a set of graphs)
- + a transport protocol to invoke the services
- + service description using web services
- + several output formats

SPARQL Endpoints

- SPARQL protocol services
 - Enables users (human or other) to query a knowledge base using SPARQL
 - Results are typically returned in one or more machine-processable formats
- Programmatic access using libraries:
 - ARC, RAP, Jena, Sesame, Javascript SPARQL, PySPARQL, R SPARQL package, etc.
- Examples:

| Project | Endpoint |
|--------------------------|---|
| DBpedia | http://dbpedia.org/sparql |
| BBC Programmes and Music | http://bbc.openlinksw.com/sparql/ |
| data.gov | http://semantic.data.gov/sparql |
| data.gov.uk | http://data.gov.uk/sparql |
| Musicbrainz | http://dbtune.org/musicbrainz/sparql |

Example: Querying DBpedia

- People who were born in Berlin before 1900

```
PREFIX dbo: <http://dbpedia.org/ontology/>
```

```
SELECT ?name ?birth ?death ?person WHERE {
    ?person dbpedia2:birthPlace <http://dbpedia.org/resource/Berlin> .
    ?person foaf:name ?name .
    ?person dbo:birthDate ?birth .
    ?person dbo:deathDate ?death
    FILTER (xsd:dateTime(?birth) <= "1900-01-01T00:00:00Z"^^xsd:dateTime) .
} ORDER BY ?name
```

Example: Querying DBpedia

| name | birth | death | person |
|---|------------------------|------------------------|---|
| "Adolf von Brauchitsch"@en | "1876-11-7"^^xsd:date | "1935-1-21"^^xsd:date | :Adolf_von_Brauchitsch ↗ |
| "Albert Wodrig"@en | "1883-7-16"^^xsd:date | "1972-10-31"^^xsd:date | :Albert_Wodrig ↗ |
| "Albrecht Ritschl"@en | "1822-3-25"^^xsd:date | "1889-3-20"^^xsd:date | :Albrecht_Ritschl ↗ |
| "Alexander von Hartmann"@en | "1890-12-11"^^xsd:date | "1943-1-26"^^xsd:date | :Alexander_von_Hartmann ↗ |
| "Anton Schmid"@en | "1897-7-23"^^xsd:date | "1945-4-29"^^xsd:date | :Anton_Schmid_(Oberst) ↗ |
| "Arthur Boje"@en | "1895-8-3"^^xsd:date | "1976-4-18"^^xsd:date | :Arthur_Boje ↗ |
| "Arthur Nebe"@en | "1894-11-13"^^xsd:date | "1945-3-21"^^xsd:date | :Arthur_Nebe ↗ |
| "Bruno Loerzer"@en | "1891-1-22"^^xsd:date | "1960-8-23"^^xsd:date | :Bruno_Loerzer ↗ |
| "Carl-Heinrich von Stülpnagel"@en | "1886-1-2"^^xsd:date | "1944-8-30"^^xsd:date | :Carl-Heinrich_von_Stülpnagel ↗ |
| "Charlotte Oelschlagel"@en | "1898-8-14"^^xsd:date | "1984-11-14"^^xsd:date | :Charlotte_Oelschlägel ↗ |
| "David ben Naphtali Fränkel"@en | "1707-1-1"^^xsd:date | "1762-1-1"^^xsd:date | :David_ben_Naphtali_Fränel ↗ |
| "Dr. Ernst Meiners"@en | "1893-6-13"^^xsd:date | "1959-6-11"^^xsd:date | :Ernst_Meiners ↗ |
| "Eberhard Kinzel"@en | "1897-10-18"^^xsd:date | "1945-5-23"^^xsd:date | :Eberhard_Kinzel ↗ |
| "Ellen Brockhöft"@en | "1895-4-25"^^xsd:date | "1977-12-19"^^xsd:date | :Ellen_Brockhöft ↗ |
| "Erich Marcks"@en | "1891-6-6"^^xsd:date | "1944-6-12"^^xsd:date | :Erich_Marcks ↗ |
| "Erich von Manstein"@en | "1887-11-24"^^xsd:date | "1973-6-10"^^xsd:date | :Erich_von_Manstein ↗ |
| "Ernst Meiners"@en | "1893-6-13"^^xsd:date | "1959-6-11"^^xsd:date | :Ernst_Meiners ↗ |
| "Ernst-Anton von Krosigk"@en | "1898-3-5"^^xsd:date | "1945-3-16"^^xsd:date | :Ernst-Anton_von_Krosigk ↗ |
| "Erwin Rauch"@en | "1889-10-19"^^xsd:date | "1969-2-26"^^xsd:date | :Erwin_Rauch ↗ |
| "Erwin Sander"@en | "1892-3-5"^^xsd:date | "1962-12-7"^^xsd:date | :Erwin_Sander ↗ |
| "Erwin Vierow"@en | "1890-5-15"^^xsd:date | "1982-2-1"^^xsd:date | :Erwin_Vierow ↗ |
| "Franz Breithaupt"@en | "1880-12-8"^^xsd:date | "1945-4-29"^^xsd:date | :Franz_Breithaupt ↗ |
| "Friedrich Fromm"@en | "1888-10-8"^^xsd:date | "1945-3-12"^^xsd:date | :Friedrich_Fromm ↗ |
| "Friedrich Graf Kleist von Nollendorf"@en | "1762-4-9"^^xsd:date | "1823-2-17"^^xsd:date | :Friedrich_Graf_Kleist_von_Nollendorf ↗ |
| "Friedrich von Rabenau"@en | "1884-10-10"^^xsd:date | "1945-4-15"^^xsd:date | :Friedrich_von_Rabenau ↗ |
| "Fritz Thiele"@en | "1894-4-14"^^xsd:date | "1944-9-4"^^xsd:date | :Fritz_Thiele ↗ |
| "Georg Bruchmüller"@en | "1863-12-11"^^xsd:date | "1948-1-26"^^xsd:date | :Georg_Bruchmüller ↗ |
| "Georg Freiherr von Gayl"@en | "1850-2-25"^^xsd:date | "1927-5-3"^^xsd:date | :Georg_Freiherr_von_Gayl ↗ |
| "Gotthard Frantz"@en | "1888-5-5"^^xsd:date | "1973-1-21"^^xsd:date | :Gotthard_Frantz ↗ |
| "Hans Hahne"@en | "1894-11-30"^^xsd:date | "1944-6-26"^^xsd:date | :Hans_Hahne_(general) ↗ |
| "Hans von Ahlfen"@en | "1897-2-20"^^xsd:date | "1966-9-11"^^xsd:date | :Hans_von_Ahlfen ↗ |
| "Hans-Levin von Barby"@en | "1899-7-25"^^xsd:date | "1942-5-27"^^xsd:date | :Hans-Levin_von_Barby ↗ |
| "Hellmut von der Chevallerie"@en | "1896-11-9"^^xsd:date | "1965-6-1"^^xsd:date | :Hellmut_von_der_Chevallerie ↗ |
| "Hellmuth Felmy"@en | "1885-5-28"^^xsd:date | "1965-12-14"^^xsd:date | :Hellmuth_Felmy ↗ |
| "Herbert Rieckhoff"@en | "1898-12-25"^^xsd:date | "1948-11-30"^^xsd:date | :Herbert_Rieckhoff ↗ |
| "Hermann von Oppeln-Bronikowski"@en | "1899-1-2"^^xsd:date | "1966-9-19"^^xsd:date | :Hermann_von_Oppeln-Bronikowski ↗ |
| "Joachim Lemelsen"@en | "1888-9-28"^^xsd:date | "1954-3-30"^^xsd:date | :Joachim_Lemelsen ↗ |

Example: Querying esDBpedia (II)

- Which Spanish movie directors have been married to actors/actresses?

```
PREFIX dcterms: <http://purl.org/dc/terms/>
```

```
PREFIX dbo: <http://dbpedia.org/ontology/>
```

```
SELECT DISTINCT ?director ?actor WHERE {
    ?director rdf:type dbo:Person .
    ?director dbo:birthPlace ?place .
    ?place dbo:country <http://es.dbpedia.org/resource/Spain>.
    ?movie dbo:director ?director .
    ?movie rdf:type dbo:Film .
    ?director dbo:spouse ?actor .
    ?actor rdf:type dbo:Actor .
}
```

ORDER BY ?director

Example: Querying esDBpedia (II)

| director | actor |
|---|---|
| http://es.dbpedia.org/resource/Agustín_Navarro | http://es.dbpedia.org/resource/Carmen_de_la_Maza |
| http://es.dbpedia.org/resource/Antonia_San_Juan | http://es.dbpedia.org/resource/Luis_Miguel_Seguí |
| http://es.dbpedia.org/resource/Antonio_Banderas | http://es.dbpedia.org/resource/Ana_Leza |
| http://es.dbpedia.org/resource/Antonio_Banderas | http://es.dbpedia.org/resource/Melanie_Griffith |
| http://es.dbpedia.org/resource/Antonio_Isasi-Isasmendi | http://es.dbpedia.org/resource/Marisa_Paredes |
| http://es.dbpedia.org/resource/Antonio_Ozores | http://es.dbpedia.org/resource/Elisa_Montés |
| http://es.dbpedia.org/resource/Carlos_Saura | http://es.dbpedia.org/resource/Eulalia_Ramón |
| http://es.dbpedia.org/resource/Daniel_Calparsoro | http://es.dbpedia.org/resource/Patricia_Vico |
| http://es.dbpedia.org/resource/Daniel_Calparsoro | http://es.dbpedia.org/resource/Najwa_Nimri |
| http://es.dbpedia.org/resource/Fernando_Guillén_Cuervo | http://es.dbpedia.org/resource/Ana_Milán |
| http://es.dbpedia.org/resource/Javier_Aguirre_Fernández | http://es.dbpedia.org/resource/Enriqueta_Carballeira |
| http://es.dbpedia.org/resource/Javier_Aguirre_Fernández | http://es.dbpedia.org/resource/Esperanza_Roy |
| http://es.dbpedia.org/resource/Jesús_Franco | http://es.dbpedia.org/resource/Lina_Romay |
| http://es.dbpedia.org/resource/Jorge_Mistral | http://es.dbpedia.org/resource/Graciela_Dufau |
| http://es.dbpedia.org/resource/José_Corbacho | http://es.dbpedia.org/resource/Anna_Barrachina |
| http://es.dbpedia.org/resource/José_Luis_Garcí | http://es.dbpedia.org/resource/Andrea_Tenuta |
| http://es.dbpedia.org/resource/José_Mota | http://es.dbpedia.org/resource/Patricia_Rivas |
| http://es.dbpedia.org/resource/Karra_Elejalde | http://es.dbpedia.org/resource/Silvia_Bel |
| http://es.dbpedia.org/resource/Álex_de_la_Iglesia | http://es.dbpedia.org/resource/Carolina_Bang |

A simple SPARQL query

Data:

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .  
@prefix : <http://example.org/book/> .  
:book1 dc:title "SPARQL Tutorial" .
```

Query:

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

Query result:

| title |
|-------------------|
| "SPARQL Tutorial" |

- A pattern is *matched* against the RDF data
- Each way a pattern can be matched yields a solution
- The sequence of solutions is filtered by: Project, distinct, order, limit/offset
- One of the result forms is applied: SELECT, CONSTRUCT, DESCRIBE, ASK

Multiple matches (joins)

```
@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> .  
_:c foaf:mbox <mailto:carol@example.org> .
```

```
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> |

Matching RDF literals

```
@prefix dt: <http://example.org/datatype#> .  
@prefix ns: <http://example.org/ns#> .  
@prefix : <http://example.org/ns#> .  
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .  
  
:x ns:p "cat"@en .  
:y ns:p "42"^^xsd:integer .  
:z ns:p "abc"^^dt:specialDatatype .
```

```
SELECT ?v WHERE { ?v ?p "cat" }
```

v

```
SELECT ?v WHERE { ?v ?p "cat"@en }
```

v

<http://example.org/ns#x>

```
SELECT ?v WHERE { ?v ?p 42 }
```

v

<http://example.org/ns#y>

```
SELECT ?v WHERE { ?v ?p "abc"^^<http://example.org/datatype#specialDatatype> }
```

v

<http://example.org/ns#z>

Blank node labels in query results

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

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

| x | name |
|------------|---------|
| <u>_:c</u> | "Alice" |
| <u>_:d</u> | "Bob" |

=

| x | name |
|------------|---------|
| <u>_:r</u> | "Alice" |
| <u>_:s</u> | "Bob" |

Optional graph patterns (left outer join)

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

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

| name | mbox |
|---------|-----------------------------|
| "Alice" | <mailto:alice@example.com> |
| "Alice" | <mailto:alice@work.example> |
| "Bob" | |

Multiple optional graph patterns

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .  
  
_:a foaf:name "Alice" .  
_:a foaf:homepage <http://work.example.org/alice/> .  
  
_:b foaf:name "Bob" .  
_:b foaf:mbox <mailto:bob@work.example> .
```

```
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 }  
}
```

| name | mbox | hpage |
|---------|---------------------------|----------------------------------|
| "Alice" | | <http://work.example.org/alice/> |
| "Bob" | <mailto:bob@work.example> | |

Alternative graph patterns

```
@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" .  
_:a dc10:creator "Alice" .  
_:b dc11:title "SPARQL Protocol Tutorial" .  
_:b dc11:creator "Bob" .  
_: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 ?title  
WHERE { { ?book dc10:title ?title } UNION  
       { ?book dc11:title ?title } }
```

```
SELECT ?x ?y  
WHERE { { ?book dc10:title ?x } UNION  
       { ?book dc11:title ?y } }
```

```
SELECT ?title ?author  
WHERE  
{ { ?book dc10:title ?title . ?book dc10:creator ?author }  
UNION  
{ ?book dc11:title ?title . ?book dc11:creator ?author } }
```

| |
|----------------------------------|
| title |
| "SPARQL Protocol Tutorial" |
| "SPARQL" |
| "SPARQL (updated)" |
| "SPARQL Query Language Tutorial" |

| x | y |
|----------------------------------|----------------------------|
| | "SPARQL (updated)" |
| | "SPARQL Protocol Tutorial" |
| "SPARQL" | |
| "SPARQL Query Language Tutorial" | |

| author | title |
|---------|----------------------------------|
| "Alice" | "SPARQL Protocol Tutorial" |
| "Bob" | "SPARQL Query Language Tutorial" |

Patterns on named graphs

```
# Named graph: http://example.org/foaf/aliceFoaf
@prefix foaf:<http://.../foaf/0.1/> .
@prefix rdf:<http://.../1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:<http://.../2000/01/rdf-schema#> .

_:a  foaf:name      "Alice" .
_:a  foaf:mbox      <mailto:alice@work.example> .
_:a  foaf:knows     _:b .

_:b  foaf:name      "Bob" .
_:b  foaf:mbox      <mailto:bob@work.example> .
_:b  foaf:nick       "Bobby" .
_:b  rdfs:seeAlso   <http://example.org/foaf/bobFoaf> .

<http://example.org/foaf/bobFoaf>
    rdf:type      foaf:PersonalProfileDocument .
```

```
# Named graph: http://example.org/foaf/bobFoaf
@prefix foaf:<http://.../foaf/0.1/> .
@prefix rdf:<http://.../1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:<http://.../2000/01/rdf-schema#> .

_:z  foaf:mbox      <mailto:bob@work.example> .
_:z  rdfs:seeAlso   <http://example.org/foaf/bobFoaf> .
_:z  foaf:nick       "Robert" .

<http://example.org/foaf/bobFoaf>
    rdf:type      foaf:PersonalProfileDocument .
```

Patterns on named graphs II

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

```
SELECT ?src ?bobNick
FROM NAMED <http://example.org/foaf/aliceFoaf>
FROM NAMED <http://example.org/foaf/bobFoaf>
WHERE
{
  GRAPH ?src
  { ?x foaf:mbox <mailto:bob@work.example> .
    ?x foaf:nick ?bobNick
  }
}
```

| | |
|-------------------------------------|----------|
| src | bobNick |
| <http://example.org/foaf/aliceFoaf> | "Bobby" |
| <http://example.org/foaf/bobFoaf> | "Robert" |

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

```
SELECT ?nick
FROM NAMED <http://example.org/foaf/aliceFoaf>
FROM NAMED <http://example.org/foaf/bobFoaf>
WHERE
{
  GRAPH data:bobFoaf {
    ?x foaf:mbox <mailto:bob@work.example> .
    ?x foaf:nick ?nick }
}
```

| |
|----------|
| nick |
| "Robert" |

Restricting values

```
@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 .
```

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>  
SELECT ?title  
WHERE { ?x dc:title ?title  
       FILTER regex(?title, "SPARQL")  
 }
```

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>  
SELECT ?title  
WHERE { ?x dc:title ?title  
       FILTER regex(?title, "web", "i")  
 }
```

```
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 < 30.5)  
       ?x dc:title ?title . }
```

| |
|-------------------|
| title |
| "SPARQL Tutorial" |

| |
|--------------------|
| title |
| "The Semantic Web" |

| | |
|--------------------|-------|
| title | price |
| "The Semantic Web" | 23 |

- Based on XQuery 1.0 and XPath 2.0 Function and Operators
- XSD boolean, string, integer, decimal, float, double, dateTIme
- Notation <, >, =, <=, >= and != for value comparison applied to any type
- BOUND, isURI, isBLANK, isLITERAL
- REGEX, LANG, DATATYPE, STR (lexical form)
- Function call for casting and extensions functions

Solution sequences and modifiers

- **Order modifier:** put the solutions in order
- **Projection modifier:** choose certain variables
- **Distinct modifier:** ensure solutions in the sequence are unique
- **Reduced modifier:** permit elimination of some non-unique solutions
- **Limit modifier:** restrict the number of solutions
- **Offset modifier:** control where the solutions start from in the overall sequence of solutions

```
SELECT ?name  
WHERE { ?x foaf:name ?name ; :empId ?emp }  
ORDER BY ?name DESC(?emp)
```

```
SELECT ?name  
WHERE  
{ ?x foaf:name ?name }
```

```
SELECT DISTINCT ?name  
WHERE { ?x foaf:name ?name }
```

```
SELECT REDUCED ?name  
WHERE { ?x foaf:name ?name }
```

```
SELECT ?name  
WHERE { ?x foaf:name ?name }  
LIMIT 20
```

```
SELECT ?name WHERE { ?x foaf:name ?name }  
ORDER BY ?name  
LIMIT 5  
OFFSET 10
```

- **SELECT**
 - Returns all, or a subset of, the variables bound in a query pattern match
- **CONSTRUCT**
 - Returns an RDF graph constructed by substituting variables in a set of triple templates
- **ASK**
 - Returns a boolean indicating whether a query pattern matches or not
- **DESCRIBE**
 - Returns an RDF graph that describes the resources found

SPARQL query forms: SELECT

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

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>  
SELECT ?nameX ?nameY ?nickY  
WHERE  
{ ?x foaf:knows ?y ;  
  foaf:name ?nameX .  
  ?y foaf:name ?nameY .  
  OPTIONAL { ?y foaf:nick ?nickY }  
}
```

| nameX | nameY | nickY |
|---------|---------|-------|
| "Alice" | "Bob" | |
| "Alice" | "Clare" | "CT" |

SPARQL query forms: CONSTRUCT

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

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>  
PREFIX vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>  
  
CONSTRUCT { <http://example.org/person#Alice> vcard:FN ?name }  
  
WHERE { ?x foaf:name ?name }
```

Query result:

```
@prefix vcard: <http://www.w3.org/2001/vcard-rdf/3.0#> .  
  
<http://example.org/person#Alice> vcard:FN "Alice" .
```

SPARQL query forms: ASK

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .  
  
_:a foaf:name "Alice" .  
_:a foaf:homepage <http://work.example.org/alice/> .  
  
_:b foaf:name "Bob" .  
_:b foaf:mbox <mailto:bob@work.example> .
```

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

Query result:

yes

SPARQL query forms: DESCRIBE

```
PREFIX ent: <http://org.example.com/employees#>  
  
DESCRIBE ?x WHERE { ?x ent:employeeId "1234" }
```

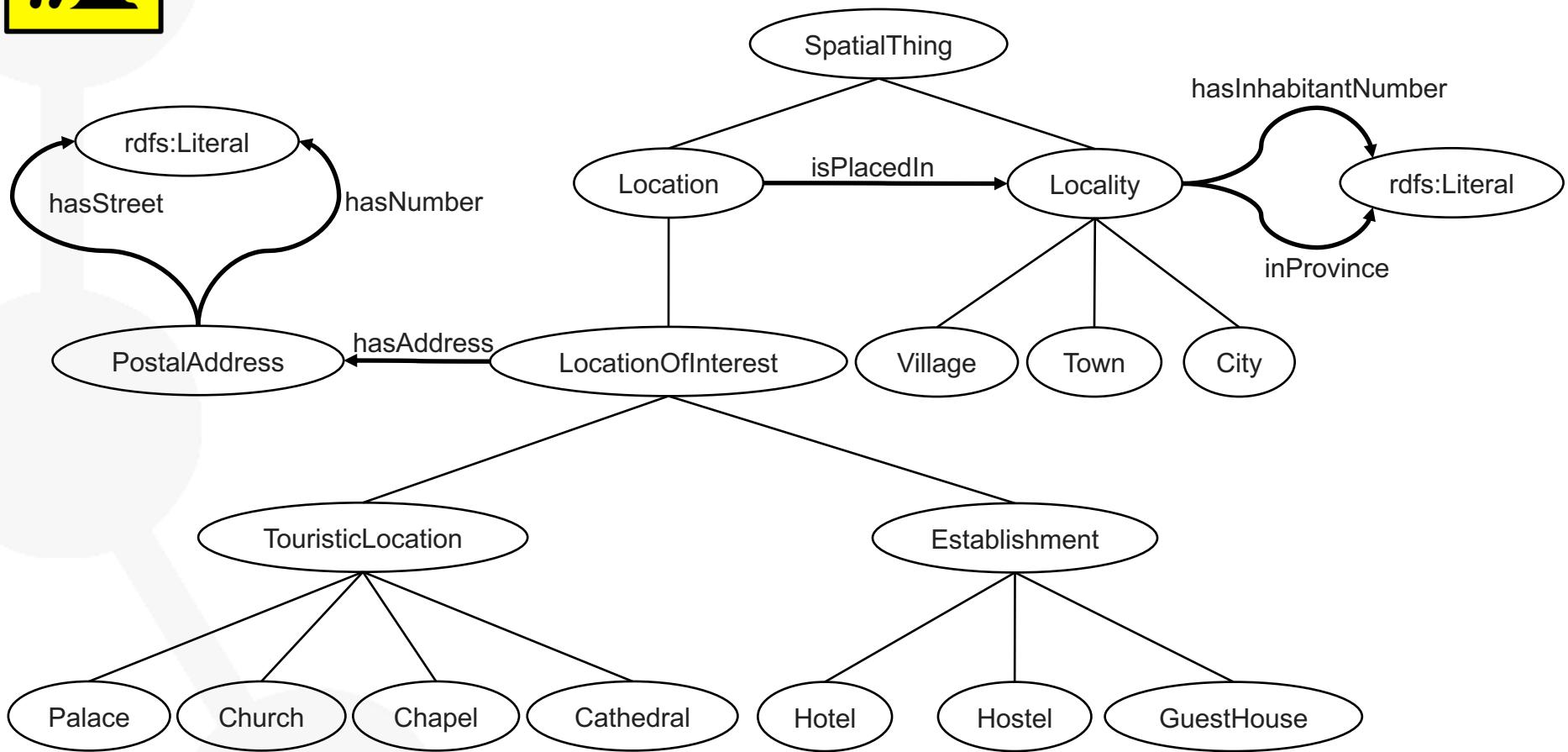
Query result:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .  
@prefix vcard: <http://www.w3.org/2001/vcard-rdf/3.0> .  
@prefix exOrg: <http://org.example.com/employees#> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix owl: <http://www.w3.org/2002/07/owl#>  
  
_:a exOrg:employeeId "1234" ;  
  
      foaf:mbox_sha1sum "ABCD1234" ;  
      vcard:N  
      [ vcard:Family "Smith" ;  
        vcard:Given "John" ] .  
  
foaf:mbox_sha1sum rdf:type owl:InverseFunctionalProperty .
```



- **Objective**
 - Understand how to perform SPARQL queries
- **Tasks**
 - Perform a set of SPARQL queries over a sample ontology
 - <http://sandbox.linkeddata.es/sparql>
 - In the graph: http://sandbox.linkeddata.es/Grado_20122013

Sample vocabulary



Queries en este modelo

1. Get all the classes
2. Get the subclasses of the class Establishment
3. Get the instances of the class City
4. Get the number of inhabitants of Santiago de Compostela
5. Get the number of inhabitants of Santiago de Compostela and of Arzua
6. Get different places with the inhabitants number, ordering the results by name of the place (ascending)
7. Get all the instances of Locality with their inhabitant number (if it exists)
8. Get all the places with more than 200.000 inhabitants
9. Get postal data of Pazo_Breogan (street, number, locality, province)
10. Get the subclasses of class Location
11. Get the instances of class Locality
12. Describe the resource with rdfs:label "Madrid"
13. Construct the RDF(S) graph that directly relates all the touristic places with their respective provinces, using a new property called "isIn"
14. Ask if there is some instance of Town
15. Ask if there is some instance of Chapel



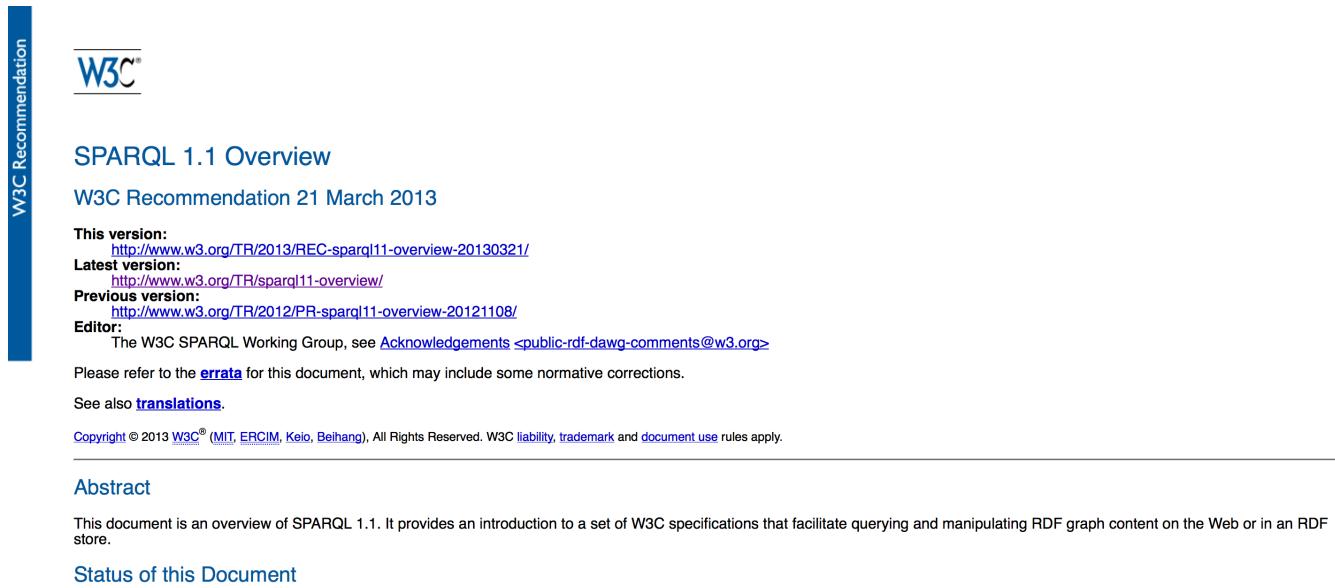
- **Objective**
 - Understand how to perform more complex SPARQL queries
- **Tasks**
 - Perform a set of SPARQL queries over DBpedia
 - <http://dbpedia.org/snorql/>
 - In the default graph

Queries to perform

1. Get all the properties that can be applied to instances of the Politician class
([<http://dbpedia.org/ontology/Politician>](http://dbpedia.org/ontology/Politician))
2. Get all the properties, except rdf:type, that can be applied to instances of the Politician class
3. Which different values exist for the properties, except rdf:type, of the instances of the Politician class?
4. For each of the properties, except rdf:type, that can be applied to instances of the Politician class, which different values do they take in those instances?
5. For each of the properties, except rdf:type, that can be applied to instances of the Politician class, how many distinct values do they take in those instances?

- SPARQL 1.1 Overview

<http://www.w3.org/TR/sparql11-overview/>



The screenshot shows the W3C SPARQL 1.1 Overview page. On the left, there's a vertical bar with the text "W3C Recommendation". The main content area has a header "SPARQL 1.1 Overview" and "W3C Recommendation 21 March 2013". Below this, there are links for "This version", "Latest version", "Previous version", and "Editor". It also includes a note about referring to the errata and translations, and copyright information. At the bottom, there are sections for "Abstract" and "Status of this Document".

- SPARQL validator:

<http://www.sparql.org/query-validator.html>
<http://yasgui.laurensrietveld.nl/>

- SPARQL implementations:

<http://esw.w3.org/topic/SparqlImplementations>

Questions?

Raúl García-Castro, Óscar Corcho

Facultad de Informática, Universidad Politécnica de Madrid
Campus de Montegancedo sn, 28660 Boadilla del Monte, Madrid

<http://www.oeg-upm.net>
{rgarcia,ocorcho}@fi.upm.es

