

# **SPARQL**

## **(and the Regime Entailments)**

ANNA QUERALT, OSCAR ROMERO

(FACULTAT D'INFORMÀTICA DE BARCELONA)

# SPARQL

---

SPARQL: SPARQL Protocol And RDF Query Language

- Standard query language for RDF(S) Graphs
- Is a W3C Recommendation
- It supports RDFS (or OWL) under specific entailments

Based on (navigational) pattern matching

- Simple RDF graphs are used as query patterns

```
Select x,z where x Lectures y, y TaughtIn z, z rdf:type Faculty
```

**The semantics applied are those of homomorphism**

# SPARQL: Basics

---

4 query forms that retrieve either result sets or RDF graphs

- 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 Endpoints:**

It is an endpoint accepting SPARQL queries and returning results via HTTP

# SPARQL SELECT Example

---

Select all pairs lecturer, course such that the lecturer lectures the course

```
PREFIX fib: <http://www.fib.edu/elements/>
```

```
SELECT ?lecturer ?course
```

OUTPUT

```
WHERE
```

```
{
```

```
  ?lecturer fib:lectures ?course
```

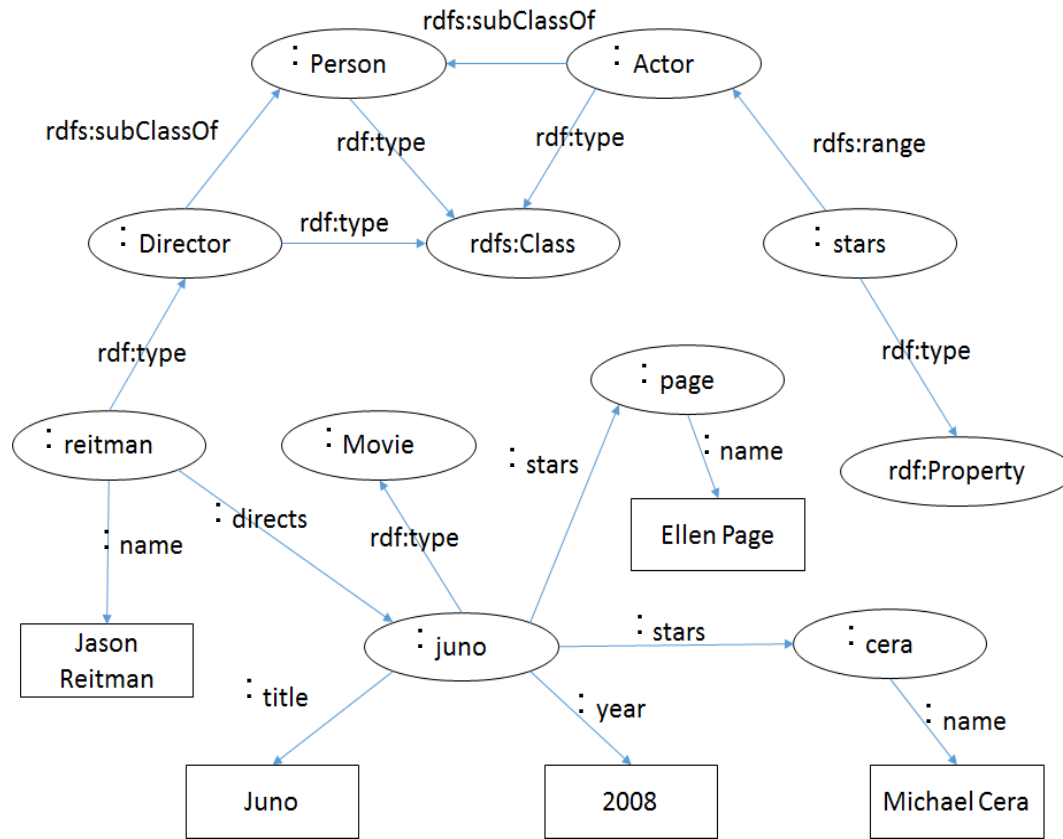
PATTERN

```
}
```

Which is equivalent to: Select x, y Where x lectures y

SPARQL allows property paths based on regular expressions (check the syntax here):  
<https://www.w3.org/TR/sparql11-property-paths/>

# Example of RDF(S) Graph



Write the following queries  
(assuming **no entailment regime**):

- Get the name of all actors that participated in Juno
- Get the name of all directors
- Get the name of all persons
- Get the title of all movies

# SPARQL 1.1 - Entailment Regimes

---

**Simple entailment:** evaluation of basic graph pattern by means of pattern matching under homomorphism

Advanced entailment regimes have been developed based on inference

- To retrieve solutions that *are logical consequences* of the axioms asserted

Most popular **advanced entailments**:

- RDF Schema entailment,
- OWL 2 RDF-Based Semantics entailment,
- Etc.

Some tools, like GraphDB, allow you to define your own entailment regime or define their own

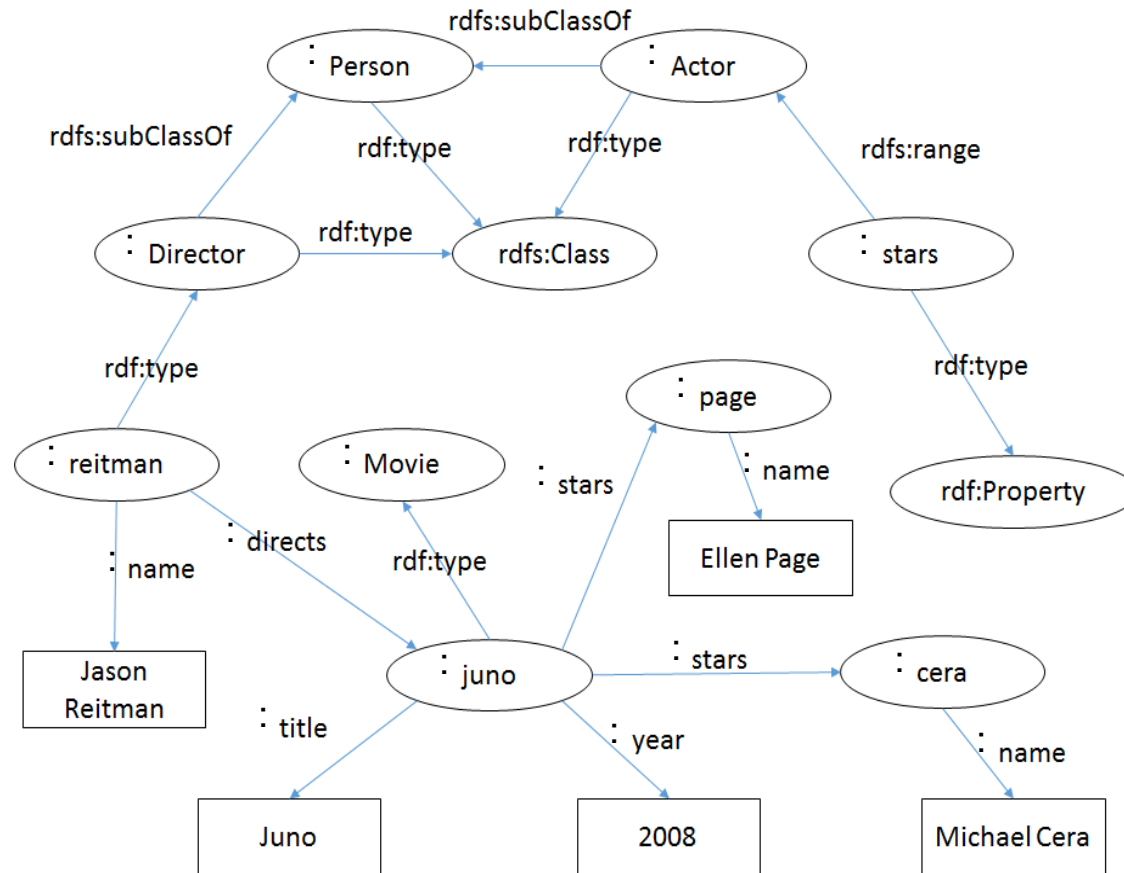
# RDFS Regime Entailment (Inference Rules)

RDFS entailment patterns.

	If S contains:	then S RDFS entails recognizing D:
<i>rdfs1</i>	xxx aaa yyy .	aaa <i>rdf:type</i> <i>rdf:Property</i> .
<i>rdfs2</i>	aaa <i>rdfs:domain</i> XXX . yyy aaa ZZZ .	yyy <i>rdf:type</i> XXX .
<i>rdfs3</i>	aaa <i>rdfs:range</i> XXX . yyy aaa ZZZ .	ZZZ <i>rdf:type</i> XXX .
<i>rdfs4a</i>	xxx aaa yyy .	XXX <i>rdf:type</i> <i>rdfs:Resource</i> .
<i>rdfs4b</i>	xxx aaa yyy .	yyy <i>rdf:type</i> <i>rdfs:Resource</i> .
<i>rdfs5</i>	XXX <i>rdfs:subPropertyOf</i> yyy . yyy <i>rdfs:subPropertyOf</i> ZZZ .	XXX <i>rdfs:subPropertyOf</i> ZZZ .
<del><i>rdfs6</i></del>	<del>xxx <i>rdf:type</i> <i>rdf:Property</i> .</del>	<del>xxx <i>rdfs:subPropertyOf</i> xxx .</del>
<i>rdfs7</i>	aaa <i>rdfs:subPropertyOf</i> bbb . xxx aaa yyy .	xxx bbb yyy .
<del><i>rdfs8</i></del>	<del>xxx <i>rdf:type</i> <i>rdfs:Class</i> .</del>	<del>xxx <i>rdfs:subClassOf</i> <i>rdfs:Resource</i> .</del>
<i>rdfs9</i>	XXX <i>rdfs:subClassOf</i> yyy . ZZZ <i>rdf:type</i> XXX .	ZZZ <i>rdf:type</i> yyy .
<del><i>rdfs10</i></del>	<del>xxx <i>rdf:type</i> <i>rdfs:Class</i> .</del>	<del>xxx <i>rdfs:subClassOf</i> xxx .</del>
<i>rdfs11</i>	XXX <i>rdfs:subClassOf</i> yyy . yyy <i>rdfs:subClassOf</i> ZZZ .	XXX <i>rdfs:subClassOf</i> ZZZ .
<del><i>rdfs12</i></del>	<del>xxx <i>rdf:type</i> <i>rdfs:ContainerMembershipProperty</i> .</del>	<del>xxx <i>rdfs:subPropertyOf</i> <i>rdfs:member</i> .</del>
<i>rdfs13</i>	XXX <i>rdf:type</i> <i>rdfs:Datatype</i> .	XXX <i>rdfs:subClassOf</i> <i>rdfs:Literal</i> .

+ core class inference

# Example of RDF(S) Graph



Write the following queries (assuming the **RDFS entailment regime**):

- Get the name of all actors that participated in Juno
- Get the name of all directors
- Get the name of all persons
- Get the title of all movies



# *Activity: Learning SPARQL*

---

Go to the last version of the RDF Query Language document by the W3C:  
<http://www.w3.org/TR/2013/REC-sparql11-query-20130321/> and read the following sections:

- 4. SPARQL Syntax,
- 5. Graph Patterns,
- 7. Matching Alternatives,
- 8. Negation,
- 9. Property Paths (equivalent to Navigational Pattern Matching: i.e., regular expressions on paths),
- 10. Assignment,
- 11. Aggregates,
- 12. Subqueries and
- 16. Query forms

A tutorial can be found here: <https://www.w3.org/2009/Talks/0615-qbe/>

# *Activity: Learning SPARQL*

---

Solve the exercise handed out by the lecturer (find it attached to this session in the LearnSQL website)

- This is a set of **basic** queries useful to explore a dataset
- Train yourself later to be able to trigger advanced queries

This exercise requires connecting to the DBPedia SPARQL endpoint. There are several of them, for example: <http://dbpedia.org/snorql/>

# Summary

---

SPARQL is the de facto standard to query knowledge graphs (RDF, RDFS, OWL)

It is based on pattern matching, but it also provides most of the relational operators (e.g., group by, set operators, etc.)

One may want to enable SPARQL entailment regimes, which extend pattern matching with basic reasoning capabilities

# Bibliography

---

SPARQL. W3C Recommendation. Latest at <http://www.w3.org/TR/rdf-sparql-query/>