






RDF, RDF Schema y 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



License

- This work is licensed under the Creative Commons Attribution – Non Commercial – Share Alike License
- You are free:
 -  to Share — to copy, distribute and transmit the work
 -  to Remix — to adapt the work
- Under the following conditions 
 - 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”
 - Non-commercial
 - Share-Alike









2


Índice	
• Introducción general	
• RDF	
– Componentes de RDF	
– Serialización de RDF	
• RDF Schema	
– Componentes de RDFS	
– Serialización de RDFS	

RDF y RDF Schema

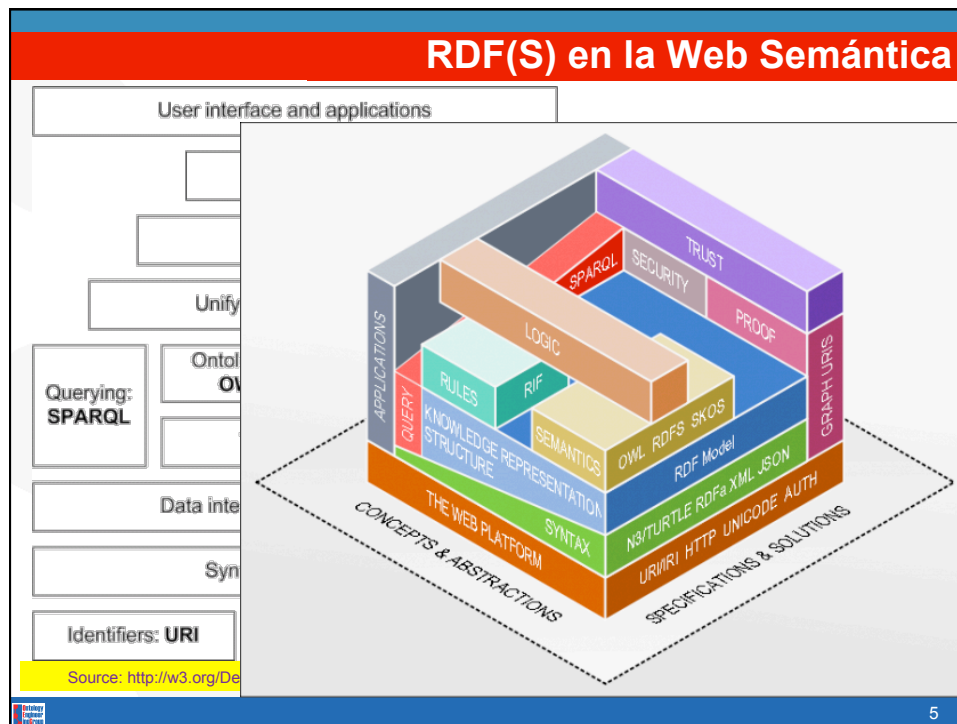
- RDF: Resource Description Framework
- Objetivo
 - Describir la semántica de la información de manera procesable por máquinas

	Database	XML	RDF(S)	
Esquema				RDF Schema
Datos				RDF

- Recomendación del W3C
 - Modelo
 - Sintaxis
 - Semántica



4


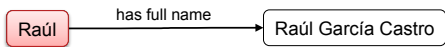
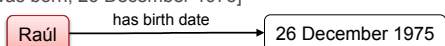
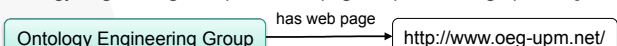


Índice

- **Introducción general**
- **RDF**
 - Componentes de RDF
 - Serialización de RDF
- **RDF Schema**
 - Componentes de RDFS
 - Serialización de RDFS

6

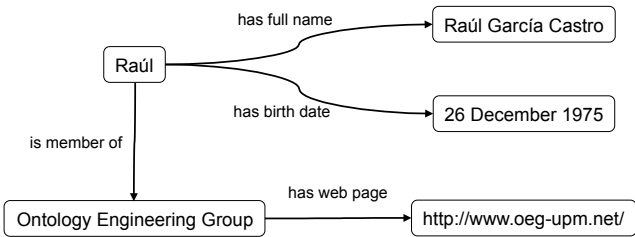
Componentes de RDF

- También conocido como “triples” (tripletas)
 - [Sujeto, Predicado, Objeto]
- “Raúl es un miembro del Ontology Engineering Group”
 - [Raúl, is member of, Ontology Engineering Group]
- “El nombre completo de Raúl es Raúl García Castro”
 - [Raúl, has full name, Raúl García Castro]
- “Raúl nació el 26 de Diciembre de 1975”
 - [Raúl, was born, 26 December 1975]
- “La página web del Ontology Engineering Group es http://www.oeg-upm.net/”
 - [Ontology Engineering Group, has web page, http://www.oeg-upm.net/]

7

Grafos y Datasets RDF

- Los grafos RDF son conjuntos de tripletas



- Y un conjunto de grafos conforma un RDF Dataset (en RDF1.1)
 - Hay un grafo por defecto (default graph)
 - Y cero o más grafos con nombre (named graph)

8

Literales RDF

- Los objetos de las tripletas pueden ser literales (cadenas de caracteres)
 - El sujeto y el predicado son siempre recursos
- Los literales pueden estar tipados
 - Normalmente usando XML Schema datatypes
 - Y algunos extra (RDF1.1)
 - `rdf:langString` (por ejemplo, "Spain"@en)
 - `rdf:HTML` and `rdf:XMLLiteral`

```

graph LR
    Raúl -- "has full name" --> "Raúl García Castro"
    Raúl -- "has birth date" --> "1975-12-26"^^xsd:date
    Raúl -- "is member of" --> OEG[Ontology Engineering Group]
    OEG -- "has web page" --> "http://www.oeg-upm.net/"
    
```

9

Literales RDF. Tipos de datos comunes (I)

A list of the RDF-compatible XSD types, with short descriptions"

	Datatype	Value space (informative)
Core types	<code>xsd:string</code>	Character strings (but not all Unicode character strings)
	<code>xsd:boolean</code>	true, false
	<code>xsd:decimal</code>	Arbitrary-precision decimal numbers
IEEE floating-point numbers	<code>xsd:integer</code>	Arbitrary-size integer numbers
	<code>xsd:double</code>	64-bit floating point numbers incl. $\pm\text{Inf}$, ± 0 , NaN
Time and date	<code>xsd:float</code>	32-bit floating point numbers incl. $\pm\text{Inf}$, ± 0 , NaN
	<code>xsd:date</code>	Dates (yyyy-mm-dd) with or without timezone
	<code>xsd:time</code>	Times (hh:mm:ss.sss...) with or without timezone
	<code>xsd:dateTime</code>	Date and time with or without timezone
Recurring and partial dates	<code>xsd:dateTimeStamp</code>	Date and time with required timezone
	<code>xsd:gYear</code>	Gregorian calendar year
	<code>xsd:gMonth</code>	Gregorian calendar month
	<code>xsd:gDay</code>	Gregorian calendar day of the month
	<code>xsd:gYearMonth</code>	Gregorian calendar year and month
	<code>xsd:gMonthDay</code>	Gregorian calendar month and day
	<code>xsd:duration</code>	Duration of time
	<code>xsd:yearMonthDuration</code>	Duration of time (months and years only)
	<code>xsd:dayTimeDuration</code>	Duration of time (days, hours, minutes, seconds only)

10

Literales RDF. Tipos de datos comunes (II)		
Limited-range integer numbers	xsd:byte	-128...+127 (8 bit)
	xsd:short	-32768...+32767 (16 bit)
	xsd:int	-2147483648...+2147483647 (32 bit)
	xsd:long	-9223372036854775808...+9223372036854775807 (64 bit)
	xsd:unsignedByte	0...255 (8 bit)
	xsd:unsignedShort	0...65535 (16 bit)
	xsd:unsignedInt	0...4294967295 (32 bit)
	xsd:unsignedLong	0...18446744073709551615 (64 bit)
	xsd:positiveInteger	Integer numbers >0
	xsd:negativeInteger	Integer numbers <0
Encoded binary data	xsd:hexBinary	Hex-encoded binary data
	xsd:base64Binary	Base64-encoded binary data
Miscellaneous XSD types	xsd:anyURI	Absolute or relative URIs and IRIs
	xsd:language	Language tags per [BCP47]
	xsd:normalizedString	Whitespace-normalized strings
	xsd:token	Tokenized strings
	xsd:NMTOKEN	XML NMTOKENs
	xsd:Name	XML Names
	xsd:NCName	XML NCNames

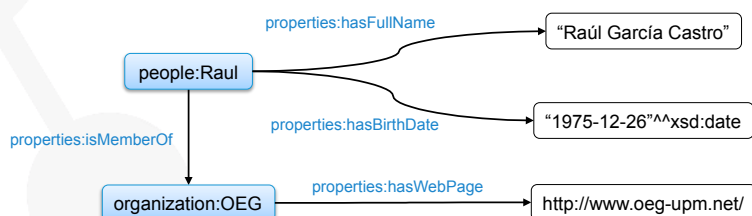
IRIs y URIs en RDF

- Componentes de una URI (RFC3986) o IRI (RFC3987)
 - `http://www.oeg-upm.net:8080/Info/People?position=current#Raul`
 - Scheme
 - Authority
 - Path
 - Query
 - Fragment
- URIs vs IRIs en RDF:
 - Uniform vs Internationalized Resource Identifier
 - Las IRIs pueden contener caracteres Unicode

Espacios de nombres (namespaces) en RDF

- Las IRIs bajo un *namespace* se llaman vocabularios

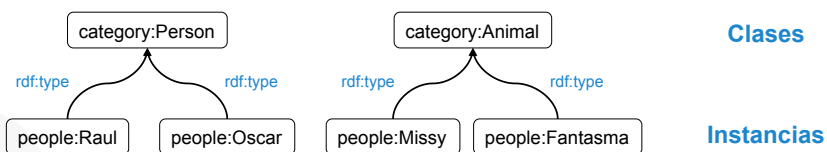
Prefijo	URI
people	http://www.oeg-upm.net/People#
organization	http://www.oeg-upm.net/Organization#
properties	http://www.oeg-upm.net/Properties#
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
xsd	http://www.w3.org/2001/XMLSchema#



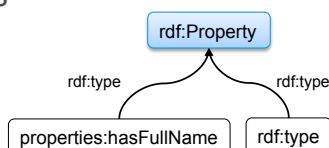
13

Clasificando recursos

- La propiedad ***rdf:type*** se usa para clasificar recursos en categorías/clases



- La clase ***rdf:Property*** es la clase de todas las propiedades



14

Nodos en blanco Valores de propiedades estructuradas

- En ocasiones, se utilizan estructuras de datos más complejas que las tripletas (aunque todo se sigue reduciendo a tripletas)

```
graph LR; Raul[people:Raul] -- properties:hasFullName --> BlankNode[ ]; BlankNode -- properties:firstName --> Raul1["Raúl"]; BlankNode -- properties:lastName --> Raul2["García Castro"];
```

Esta URI intermedia realmente no necesita tener un nombre específico

15

Contenedores RDF

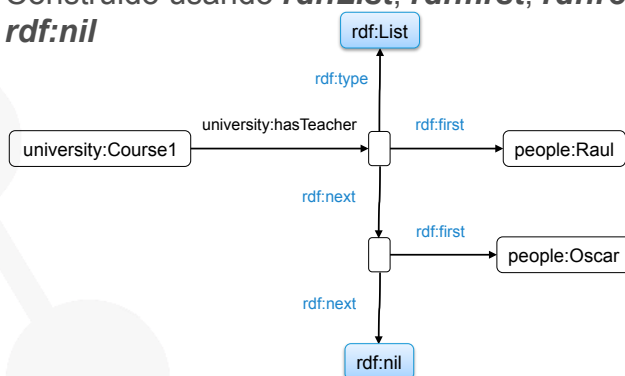
- Describen grupos de cosas
 - Un libro fue creado por varios autores
 - Una clase es impartida por varias personas
 - etc.
- RDF tiene varios tipos de contenedores
 - rdf:Bag**. Grupo de recursos o literales, incluyendo duplicados, donde el orden no es significativo
 - rdf:Seq**. Grupo de recursos o literales, incluyendo duplicados, donde el orden es significativo
 - rdf:Alt**. Grupo de recursos o literales que son alternativas (normalmente para un valor simple de una propiedad)

```
graph LR; Raul[people:Raul] -- properties:hasEmailAddress --> BlankNode[ ]; BlankNode -- rdf:type --> Seq[rdf:Seq]; BlankNode -- rdf:_1 --> Email1["rgarcia@fi.upm.es"]; BlankNode -- rdf:_2 --> Email2["r.garcia@upm.es"];
```

16

Colecciones en RDF

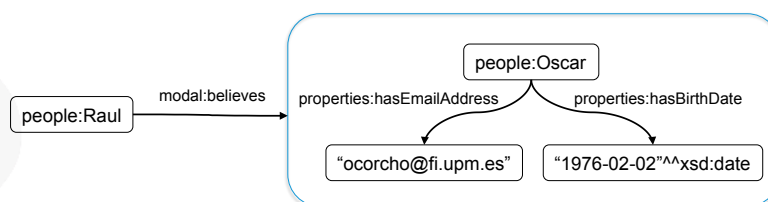
- Grupos de cosas representadas como una lista estructurada
 - “A container with limits”
- Construido usando ***rdf:List***, ***rdf:first***, ***rdf:rest***, y ***rdf:nil***



17

Reificación en RDF

- Tripletas en RDF sobre otras tripletas en RDF
 - “Raúl cree que la fecha de nacimiento de Oscar es el 2 de febrero de 1976 y que su dirección de correo electrónico es ocorcho@fi.upm.es”
- Expresado usando ***rdf:Statement***, ***rdf:subject***, ***rdf:predicate***, y ***rdf:object***

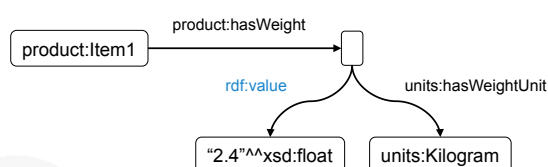


- Reificación en RDF
 - Permite expresar creencias (y otras modalidades)
 - Permite expresar modelos de confianza, firmas digitales, etc.
 - Permite expresar metadatos sobre metadatos

18

Valor principal de un valor estructurado

- Algunas veces uno de los valores de un valor estructurado es el principal
 - El peso de un artículo es 2.4 kilogramos
 - El valor principal es 2.4, lo que se expresa con *rdf:value*
- Se usa muy poco



19

Resumen del vocabulario RDF

Clases	Propiedades	Individuos
Classification		
<i>rdf:Property</i>	<i>rdf:type</i>	
Containers		
<i>rdf:Bag</i>	<i>rdf:_1</i> , <i>rdf:_2</i> , <i>rdf:_3...</i>	
<i>rdf:Seq</i>		
<i>rdf:Alt</i>		
Collections		
<i>rdf:List</i>	<i>rdf:first</i>	<i>rdf:nil</i>
	<i>rdf:rest</i>	
Reification		
<i>rdf:Statement</i>	<i>rdf:subject</i>	
	<i>rdf:predicate</i>	
	<i>rdf:object</i>	
Values		
<i>rdf:XMLLiteral</i>	<i>rdf:value</i>	

20

Serializaciones en RDF

- Varias serializaciones (muchas de ellas finalmente aprobadas en Feb2014)
 - RDF/XML (www.w3.org/TR/rdf-syntax-grammar/)
 - Turtle (<http://www.w3.org/TR/turtle/>)
 - N-Triples (<http://www.w3.org/TR/n-triples/>)
 - TriG (<http://www.w3.org/TR/trig/>)
 - RDFa (<http://www.w3.org/TR/xhtml-rdfa-primer/>)
 - JSON-LD (<http://www.w3.org/TR/json-ld/>)
 - N-Quads (<http://www.w3.org/TR/n-quads/>)

- Importante:** las serializaciones permiten diferentes variantes sintácticas.
 - El orden en el que las tripletas RDF aparecen en un documento no es relevante

Serializaciones en RDF. RDF/XML

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:person="http://www.ontologies.org/ontologies/people#"
  xmlns="http://www.oeg-upm.net/ontologies/people#"
  xml:base="http://www.oeg-upm.net/ontologies/people">

  <rdf:Property rdf:about="http://www.ontologies.org/ontologies/people#hasHomePage"/>
  <rdf:Property rdf:about="http://www.ontologies.org/ontologies/people#hasColleague"/>
  <rdf:Property rdf:about="http://www.ontologies.org/ontologies/people#hasName"/>

  <rdf:Description rdf:about="#Raul"/>
  <rdf:Description rdf:about="#Asun">
    <person:hasColleague rdf:resource="#Raul"/>
    <person:hasHomePage>http://www.fi.upm.es</person:hasHomePage>
  </rdf:Description>
  <rdf:Description rdf:about="#Oscar">
    <person:hasColleague rdf:resource="#Asun"/>
    <person:hasName>Oscar Corcho García</person:hasName>
  </rdf:Description>

</rdf:RDF>
```

Serializaciones en RDF. Turtle

```

graph LR
    Oscar[people:Oscar] -- people:hasColleague --> Asun[people:Asun]
    Asun -- people:hasColleague --> Raul[people:Raul]
    Oscar -- people:hasName --> Name["Óscar Corcho García"]
    Asun -- people:hasHomePage --> Home["http://www.oeg-upm.net/"]
  
```

@base <<http://www.oeg-upm.net/ontologies/people/>>
@prefix vocab: <<http://www.ontologies.org/ontologies/people#>>
 :Asun vocab:hasColleague :Raul ;
 vocab:hasHomePage "http://www.fi.upm.es/".
 :Oscar vocab:hasColleague :Asun ;
 vocab:hasName "Óscar Corcho García".

23

RDF inference. Entailment rules

Rule Name	if E contains	then add
rdf1	uuu aaa yyy .	aaa rdf:type rdf:Property .
rdf2	uuu aaa lll . where lll is a well-typed XML literal .	_:nnn rdf:type rdf:XMLLiteral . where _:nnn identifies a blank node allocated to lll by rule lg.

24

Exercise



- **Objective**
 - Get used to the different syntaxes of RDF
- **Tasks**
 - Take the text of an RDF file and create its corresponding graph
 - Take an RDF graph and create its corresponding RDF/XML and Turtle files
 - Note: you can use <http://rdf.greggkellogg.net/distiller> to check them

25

Exercise 1.a. Create a graph from a file



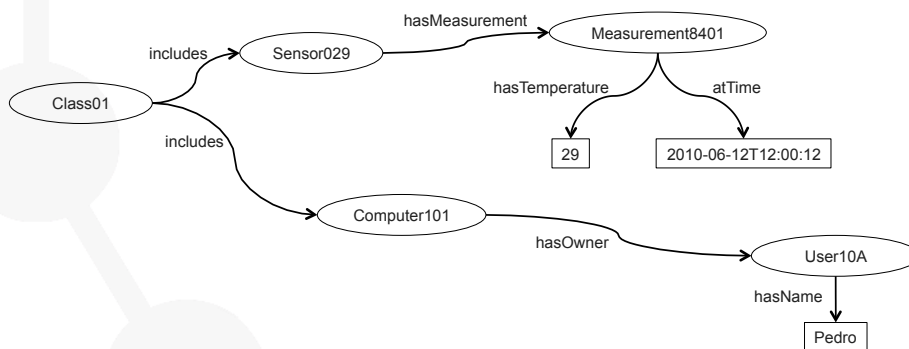
- Open the file StickyNote_PureRDF.rdf
- Create the corresponding graph from it
- Compare your graph with those of your colleagues

26

Exercise 1.b. Create files from a graph



- Transform the following graph into the RDF/XML and Turtle syntaxes

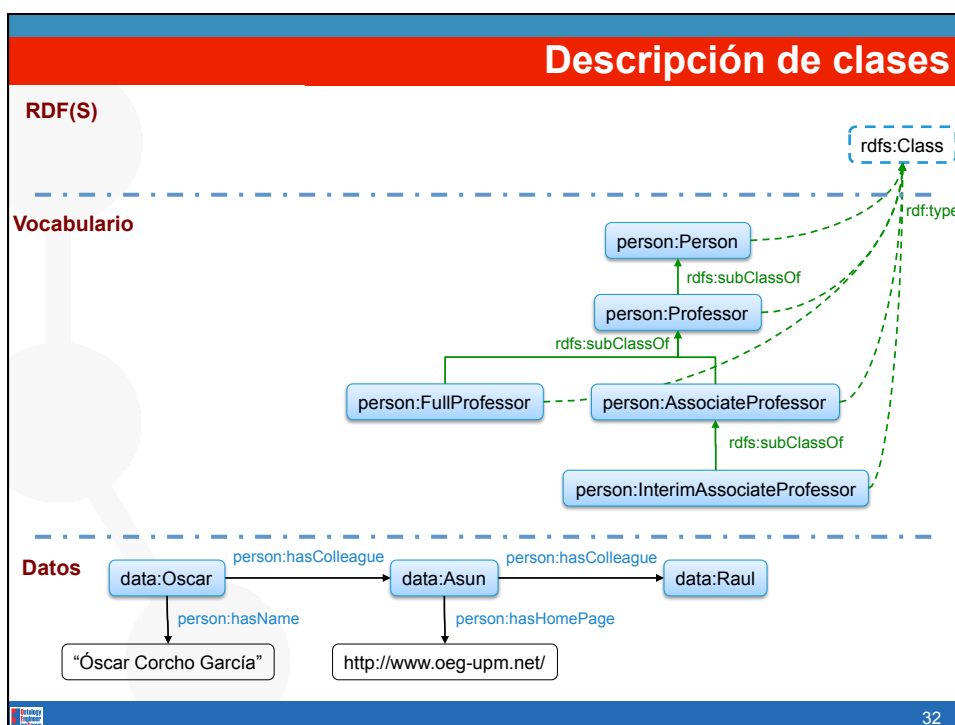
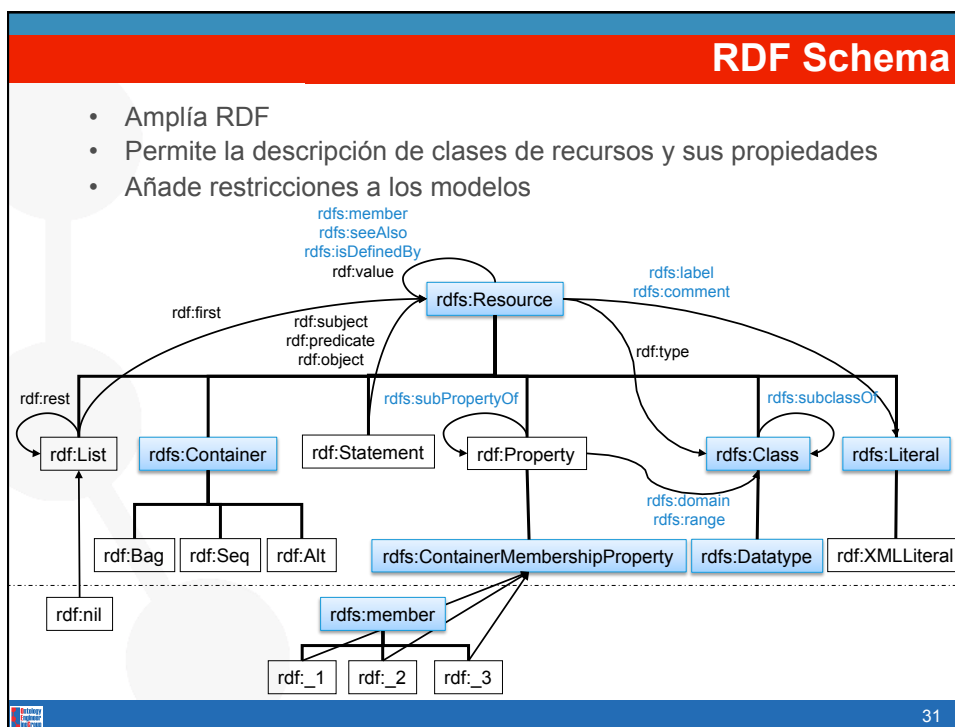


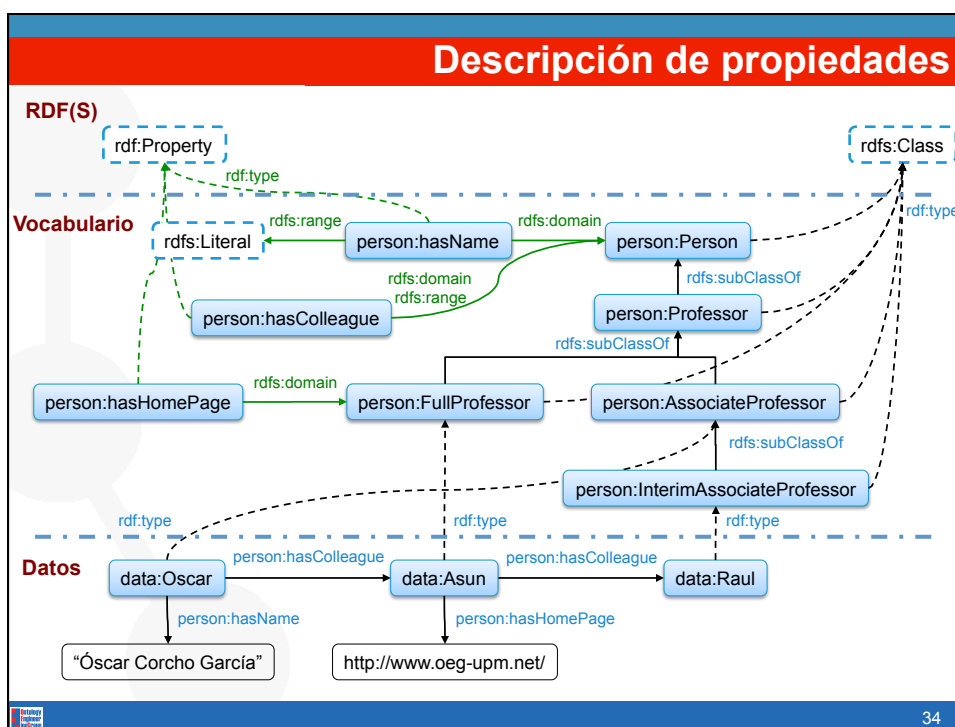
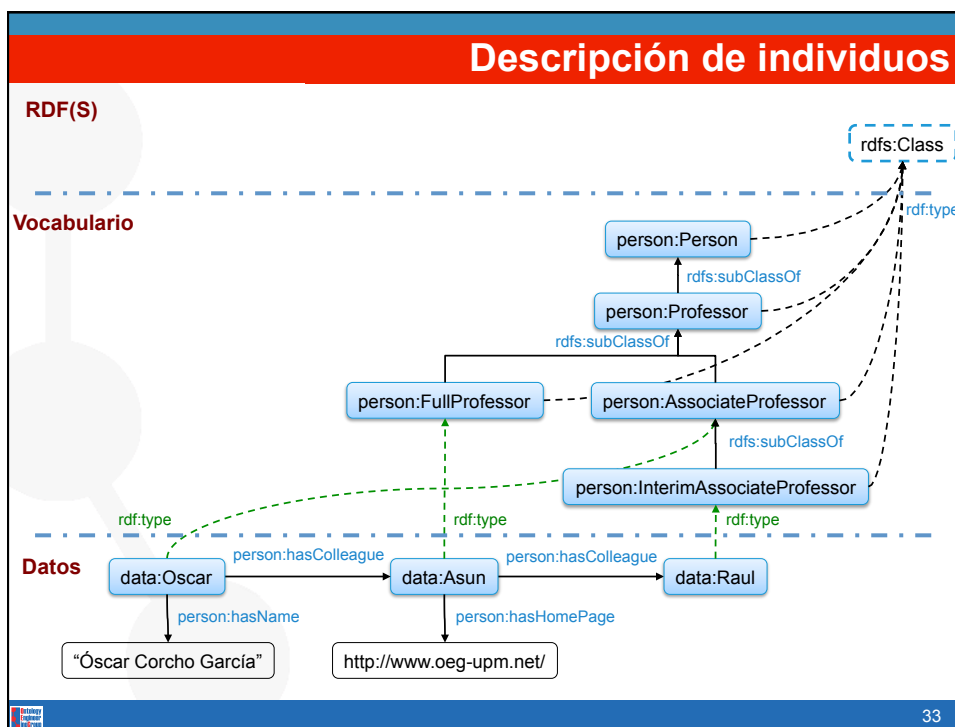
28

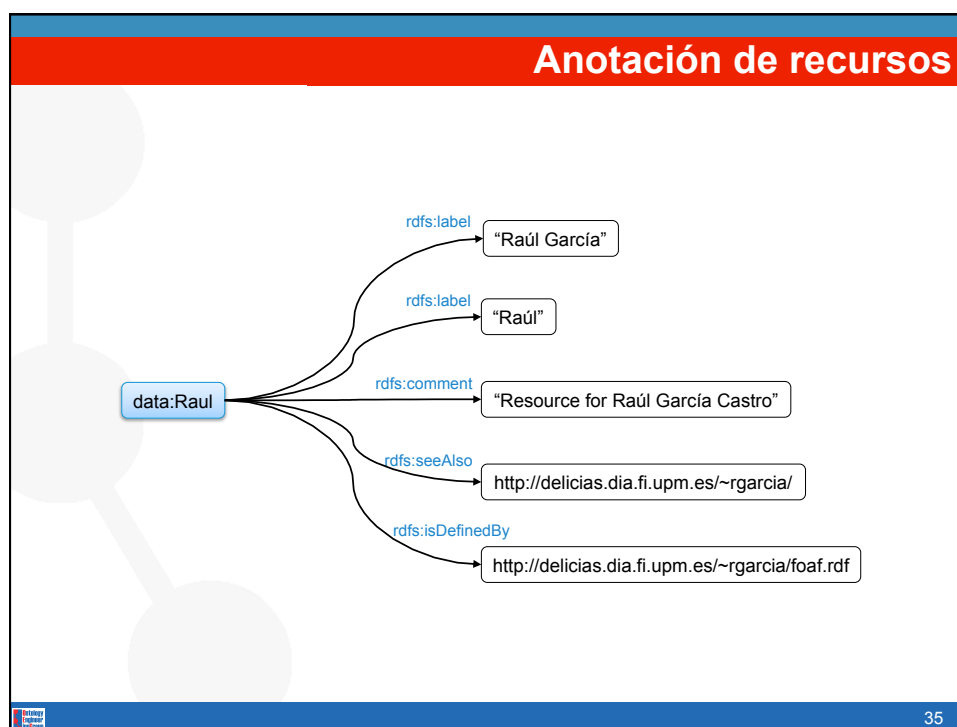
Índice

- **Introducción general**
- **RDF**
 - Componentes de RDF
 - Serialización de RDF
- **RDF Schema**
 - Componentes de RDFS
 - Serialización de RDFS

30







Resumen del vocabulario RDFS

Clases	Propiedades	Individuos
Classification		
<code>rdfs:Resource</code>	<code>rdfs:subClassOf</code>	
<code>rdfs:Class</code>		
Properties		
	<code>rdfs:domain</code>	
	<code>rdfs:range</code>	
	<code>rdfs:subPropertyOf</code>	
Datatypes		
<code>rdfs:Literal</code>		
<code>rdfs:Datatype</code>		
Containers		
<code>rdfs:Container</code>	<code>rdfs:member</code>	
<code>rdfs:ContainerMembershipProperty</code>		
Annotation		
	<code>rdfs:label</code>	
	<code>rdfs:comment</code>	
	<code>rdfs:seeAlso</code>	
	<code>rdfs:isDefinedBy</code>	

36

Serialización en RDF/XML (1/2)

```
<?xml version="1.0"?>

<!DOCTYPE rdf:RDF [
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
  <!ENTITY person "http://www.oeg-upm.net/ontologies/person#" >
]>

<rdf:RDF xmlns="http://www.oeg-upm.net/ontologies/person#"
  xml:base="http://www.oeg-upm.net/ontologies/person"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:person="http://www.oeg-upm.net/ontologies/person#">

  <rdfs:Property rdf:about="#person;hasColleague">
    <rdfs:range rdf:resource="#Person"/>
    <rdfs:domain rdf:resource="#Person"/>
  </rdfs:Property>

  <rdfs:Property rdf:about="#person;hasHomePage">
    <rdfs:domain rdf:resource="#FullProfessor"/>
  </rdfs:Property>

  <rdfs:Property rdf:about="#person;hasName">
    <rdfs:domain rdf:resource="#Person"/>
    <rdfs:range rdf:resource="#rdfs:Literal"/>
  </rdfs:Property>

  ...
```

37

Serialización en RDF/XML (2/2)

```
...
<rdfs:Class rdf:about="#AssociateProfessor">
  <rdfs:subClassOf rdf:resource="#Professor"/>
</rdfs:Class>

<rdfs:Class rdf:about="#FullProfessor">
  <rdfs:subClassOf rdf:resource="#Professor"/>
</rdfs:Class>

<rdfs:Class rdf:about="#InterimAssociateProfessor">
  <rdfs:subClassOf rdf:resource="#AssociateProfessor"/>
</rdfs:Class>

<rdfs:Class rdf:about="#Person"/>

<rdfs:Class rdf:about="#Professor">
  <rdfs:subClassOf rdf:resource="#Person"/>
</rdfs:Class>

<FullProfessor rdf:about="#Asun">
  <person:hasHomePage>http://www.fi.upm.es</person:hasHomePage>
  <person:hasColleague rdf:resource="#Raul"/>
</FullProfessor>

<AssociateProfessor rdf:about="#Oscar">
  <person:hasName>Oscar Corcho García</person:hasName>
  <person:hasColleague rdf:resource="#Asun"/>
</AssociateProfessor>

<InterimAssociateProfessor rdf:about="#Raul"/>
</rdf:RDF>
```

38

Serialización en Turtle (1/2)

```
@base <http://www.oeg-upm.net/ontologies/person> .
@prefix : <http://www.oeg-upm.net/ontologies/person#> .
@prefix person: <http://www.oeg-upm.net/ontologies/person#> .

:hasColleague a rdfs:Property .
  rdfs:domain :Person .
  rdfs:range :Person ;

:hasHomePage a rdfs:Property .
  rdfs:domain :FullProfessor .

:hasName a rdfs:Property .
  rdfs:domain :Person ;
  rdfs:range rdfs:Literal .
...
```

a is equivalent to rdfs:type

39

Serialización en Turtle (2/2)

```
...
:Person a rdfs:Class .

:Professor a rdfs:Class ;
  rdfs:subClassOf :Person .

:FullProfessor a rdfs:Class ;
  rdfs:subClassOf :Professor .

:AssociateProfessor a rdfs:Class ;
  rdfs:subClassOf :Professor .

:InterimAssociateProfessor a rdfs:Class ;
  rdfs:subClassOf :AssociateProfessor .

:Asun a :FullProfessor ;
  :hasHomePage "http://www.fi.upm.es" ;
  :hasColleague :Raul .

:Oscar a :AssociateProfessor ;
  :hasName "Oscar Corcho García" ;
  :hasColleague :Asun .

:Raul a :InterimAssociateProfessor .
```

a is equivalent to rdfs:type

40

RDF(S) inference. Entailment rules

Rule Name	If E contains:	then add:
rdfs1	UUU aaa III. where III is a plain literal (with or without a language tag).	_:nnn rdf:type rdfs:Literal . where _:nnn identifies a blank node allocated to III by rule lg.
rdfs2	aaa rdfs:domain XXX . UUU aaa yyy .	UUU rdf:type XXX .
rdfs3	aaa rdfs:range XXX . UUU aaa VW .	VW rdf:type XXX .
rdfs4a	UUU aaa XXX .	UUU rdf:type rdfs:Resource .
rdfs4b	UUU aaa VW .	VW rdf:type rdfs:Resource .
rdfs5	UUU rdfs:subPropertyOf VW . VW rdfs:subPropertyOf XXX .	UUU rdfs:subPropertyOf XXX .
rdfs6	UUU rdf:type rdf:Property .	UUU rdfs:subPropertyOf UUU .
rdfs7	aaa rdfs:subPropertyOf bbb . UUU aaa yyy .	uuu bbb yyy .
rdfs8	UUU rdf:type rdfs:Class .	UUU rdfs:subClassOf rdfs:Resource .
rdfs9	UUU rdfs:subClassOf XXX . VW rdf:type UUU .	VW rdf:type XXX .
rdfs10	UUU rdf:type rdfs:Class .	UUU rdfs:subClassOf UUU .
rdfs11	UUU rdfs:subClassOf VW . VW rdfs:subClassOf XXX .	UUU rdfs:subClassOf XXX .
rdfs12	UUU rdf:type rdfs:ContainerMembershipProperty .	UUU rdfs:subPropertyOf rdfs:member .
rdfs13	UUU rdf:type rdfs:Datatype .	UUU rdfs:subClassOf rdfs:Literal .

41

RDF(S) inference. Additional inferences

ext1	UUU rdfs:domain VW . VW rdfs:subClassOf ZZZ .	UUU rdfs:domain ZZZ .
ext2	UUU rdfs:range VW . VW rdfs:subClassOf ZZZ .	UUU rdfs:range ZZZ .
ext3	UUU rdfs:domain VW . WWW rdfs:subPropertyOf UUU .	WWW rdfs:domain VW .
ext4	UUU rdfs:range VW . WWW rdfs:subPropertyOf UUU .	WWW rdfs:range VW .
ext5	rdf:type rdfs:subPropertyOf WWW . WWW rdfs:domain VW .	rdfs:Resource rdfs:subClassOf VW .
ext6	rdfs:subClassOf rdfs:subPropertyOf WWW . WWW rdfs:domain VW .	rdfs:Class rdfs:subClassOf VW .
ext7	rdfs:subPropertyOf rdfs:subPropertyOf WWW . WWW rdfs:domain VW .	rdf:Property rdfs:subClassOf VW .
ext8	rdfs:subClassOf rdfs:subPropertyOf WWW . WWW rdfs:range VW .	rdfs:Class rdfs:subClassOf VW .
ext9	rdfs:subPropertyOf rdfs:subPropertyOf WWW . WWW rdfs:range VW .	rdf:Property rdfs:subClassOf VW .

42

Y de representación taxonómica... ¿Qué podemos usar?

- Clases
- Instancias
- ¿Podemos hacer instancias de instancias?
- Subclase de
- Propiedades de instancia
- Propiedades de clase
- Dominios y rangos (asociados a clases o generales)
- Conocimiento disjunto
- Conocimiento exhaustivo
- Reificación para relaciones n-arias
- Negación

43

RDF(S) limitations

- RDFS **too weak** to describe resources in sufficient detail
 - No **localised range and domain** constraints
 - Can't say that the range of hasChild is person when applied to persons and elephant when applied to elephants
 - No **existence/cardinality** constraints
 - Can't say that all *instances* of person have a mother that is also a person, or that persons have exactly 2 parents
 - No **boolean** operators
 - Can't say or, not, etc.
 - No **transitive, inverse or symmetrical** properties
 - Can't say that isPartOf is a transitive property, that hasPart is the inverse of isPartOf or that touches is symmetrical
- Difficult to provide **reasoning support**
 - No "native" reasoners for non-standard semantics
 - May be possible to reason via FOL axiomatisation

44

Ejemplo de algunas RDF APIs

- Librerías RDF para distintos lenguajes:
 - Java, Python, C, C++, C#, .Net, Javascript, Tcl/Tk, PHP, Lisp, Obj-C, Prolog, Perl, Ruby, Haskell
 - Un listado disponible en <http://www.w3.org/2001/sw/wiki/Tools>
- Multilenguaje:
 - Redland RDF Libraries (C, Perl, PHP, Python and Ruby): <http://librdf.org>
- Java:
 - Jena: <http://jena.apache.org/>
 - Sesame: <http://openrdf.callimachus.net/>
- PHP:
 - RAP - RDF API for PHP: <http://wifo5-03.informatik.uni-mannheim.de/bizer/rdfapi/index.html>
- Python:
 - RDFLib: <https://github.com/RDFLib>
 - Pyrple: <http://infomesh.net/pyrple/>

45

Ejercicio



- **Objetivo**
 - Entender las características de RDF(S) para implementar vocabularios y datos, incluyendo sus limitaciones
- **Tareas**
 - A partir de una descripción de dominio, crear el grafo RDF(S)
 - Primero incluir sólo el vocabulario del dominio
 - Después incluir referencias a RDF y a los vocabularios RDFS

46

Ejercicio 1. Descripción del dominio



- Cierta ubicación puede ser un lugar de interés.
- Los lugares de interés pueden ser lugares turísticos o establecimientos, pero no ambos a la vez.
- Los lugares turísticos pueden ser palacios, iglesias, capillas y catedrales.
- Los establecimientos pueden ser hoteles, hostales, o viviendas en alquiler.
- Un lugar está ubicado en una localidad, que puede ser una ciudad, un pueblo o una ciudad céntrica.
- Un lugar de interés tiene dirección postal, que incluye el nombre de la calle y el número.
- Las localidades tienen un cierto número de habitantes.
- Las localidades pertenecen a una cierta provincia.
- Covarrubias es un pueblo con 634 habitantes en la provincia de Burgos.
- El restaurante “El Galo” está ubicado en Covarrubias, en la calle Mayor, número 5.
- Una de las iglesias de Covarrubias está en la calle Santo Tomás.

47



El Lenguaje de Consultas SPARQL 1.1

Raúl García-Castro, Oscar Corcho

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

<http://www.oeg-upm.net>

ocorcho@fi.upm.es



Lenguajes de consulta sobre RDF(S)

- Soportado por las APIs y *triple stores* más relevantes
 - Jena, Sesame, etc.
 - Virtuoso, Stardog, 4store, Fuseki, etc.
 - ...

50

Consultas sobre RDF. Ejemplos (I)

- Ejemplo de grafo RDF

```
graph LR
    Oscar["data:Oscar"] -- "person:hasName" --> Name["Óscar Corcho García"]
    Oscar -- "person:hasColleague" --> Asun["data:Asun"]
    Asun -- "person:hasHomePage" --> Home["http://www.oeg-upm.net/"]
    Asun -- "person:hasColleague" --> Raul["data:Raul"]
```

- **Consulta:** “Dime las personas que tienen a Raúl como colega”

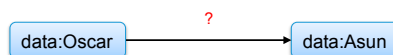
```
graph LR
    Q["?"] -- "person:hasColleague" --> Raul["data:Raul"]
```

- **Resultado:** data:Asun

51

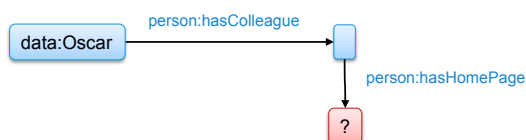
Consultas sobre RDF. Ejemplos (II)

- **Consulta:** “Dime qué relaciones existen entre Oscar y Asun”



- **Resultado:** person:hasColleague

- **Consulta:** “Dime la página Web de los colegas de Oscar”



- **Resultado:** “http://www.oeg-upm.net/”

52

SPARQL también es un protocolo

- SPARQL es un lenguaje de consulta...
Encuentra nombres y websites de colaboradores de 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 }
```
- ... Y un Protocolo

```
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/...
```
- Servicios ejecutando consultas SPARQL sobre un conjunto de grafos
- Un protocolo de transporte para invocar al servicio
- Descripción del servicio con tecnologías de servicios Web

54

Conclusiones SPARQL

- Servicios del protocolo SPARQL
 - Permite a los usuarios (humanos u otros) hacer consultas usando SPARQL
 - Los resultados son devueltos normalmente en uno o más formatos procesables por máquinas
- Listado de puntos de acceso (*endpoints*) de SPARQL
 - <http://esw.w3.org/topic/SparglEndpoints>
 - <http://spargles.okfn.org/>
- Acceso programático usando librerías:
 - ARC, RAP, Jena, Sesame, Javascript SPARQL, PySPARQL, etc.
- Ejemplos:

Proyecto	Conclusión
DBpedia	http://dbpedia.org/sparql
AragoDBpedia	http://opendata.aragon.es/sparql
Biblioteca Nacional	http://datos.bne.es/
Bathing Water Quality UK	http://environment.data.gov.uk/lab/sparql.html
Musicbrainz	http://dbtune.org/musicbrainz/snorql/

55

Ejemplo: consultando DBpedia

- Gente nacida en Berlín antes de 1900

SPARQL:

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX : <http://dbpedia.org/resource/>
PREFIX dbpedia2: <http://dbpedia.org/property/>
PREFIX dbpedia: <http://dbpedia.org/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX dbo: <http://dbpedia.org/ontology/>

SELECT ?name ?birth ?death ?person WHERE {
  ?person dbpedia2:birthPlace <http://dbpedia.org/resource/Berlin> .
  ?person dbo:birthDate ?birth .
  ?person foaf:name ?name .
  ?person dbo:deathDate ?death
  FILTER (?birth < "1900-01-01"^^xsd:date) .
}

ORDER BY ?name
```

Results:

56

Ejemplo: consultando DBpedia

name	birth	death	person
"Adolf Otto Reinhold Windaus"@en	"1876-12-25"^^xsd:date	"1959-06-09"^^xsd:date	:Adolf_Otto_Reinhold_Windaus
"Adolf von Baeyer"@en	"1835-10-31"^^xsd:date	"1917-08-20"^^xsd:date	:Adolf_von_Baeyer
"Alexander von Humboldt"@en	"1769-09-14"^^xsd:date	"1859-05-06"^^xsd:date	:Alexander_von_Humboldt
"Carl Borchardt"@en	"1817-02-22"^^xsd:date	"1880-06-27"^^xsd:date	:Carl_Wilhelm_Borchardt
"Carl Ludvig Engel"@en	"1778-07-03"^^xsd:date	"1840-05-04"^^xsd:date	:Carl_Ludwig_Engel
"Carl Ludwig Siegel"@en	"1896-12-31"^^xsd:date	"1981-04-04"^^xsd:date	:Carl_Ludwig_Siegel
"Carl Wilhelm Borchardt"@en	"1817-02-22"^^xsd:date	"1880-06-27"^^xsd:date	:Carl_Wilhelm_Borchardt
"Constantin Carathéodory"@en	"1873-09-13"^^xsd:date	"1950-02-02"^^xsd:date	:Constantin_Carath%C3%A9odory
"Eduard von Hartmann"@en	"1842-02-23"^^xsd:date	"1906-06-05"^^xsd:date	:Karl_Robert_Eduard_von_Hartmann
"Ernst Zermelo"@en	"1871-07-27"^^xsd:date	"1953-05-21"^^xsd:date	:Ernst_Zermelo
"Eugen Rosenstock-Huussy"@en	"1888-07-06"^^xsd:date	"1973-02-24"^^xsd:date	:Eugen_Rosenstock-Huussy
"Franz Karl Achard"@en	"1753-04-28"^^xsd:date	"2020-04-21"^^xsd:date	:Franz_Karl_Achard
"Fritz Perls"@en	"1893-07-08"^^xsd:date	"1970-03-14"^^xsd:date	:Fritz_Perls
"Georg Simmel"@en	"1858-03-01"^^xsd:date	"1918-09-28"^^xsd:date	:Georg_Simmel
"Gustav Magnus"@en	"1802-05-02"^^xsd:date	"1870-04-04"^^xsd:date	:Heinrich_Gustav_Magnus
"Hans Luther"@en	"1879-03-10"^^xsd:date	"1962-05-11"^^xsd:date	:Hans_Luther
"Heinrich Gustav Magnus"@en	"1802-05-02"^^xsd:date	"1870-04-04"^^xsd:date	:Heinrich_Gustav_Magnus
"Heinrich Louis d'Arrest"@en	"1822-07-13"^^xsd:date	"1875-06-14"^^xsd:date	:Heinrich_Louis_d%27Arrest
"Herbert Marcuse"@en	"1898-07-19"^^xsd:date	"1979-07-29"^^xsd:date	:Herbert_Marcuse
"Karl Dönitz"@en	"1891-09-16"^^xsd:date	"1980-12-24"^^xsd:date	:Karl_D%C3%B6nitz
"Kurt Eisner"@en	"1867-05-14"^^xsd:date	"1919-02-21"^^xsd:date	:Kurt_Eisner
"Leo Frobenius"@en	"1873-06-29"^^xsd:date	"1938-08-09"^^xsd:date	:Leo_Frobenius
"Moritz Schlick"@en	"1882-04-14"^^xsd:date	"1936-06-22"^^xsd:date	:Moritz_Schlick
"Oskar Messter"@en	"1866-11-21"^^xsd:date	"1943-12-06"^^xsd:date	:Oskar_Messter
"Otto Wilhelm Hermann von Abich"@en	"1806-12-11"^^xsd:date	"1886-07-01"^^xsd:date	:Otto_Wilhelm_Hermann_von_Abich
"Paul Richard Heinrich Blasius"@en	"1883-08-09"^^xsd:date	"1970-04-24"^^xsd:date	:Paul_Richard_Heinrich_Blasius
"Philipp Veit"@en	"1793-02-13"^^xsd:date	"1877-12-18"^^xsd:date	:Philipp_Veit
"Rudolph Schoenheimer"@en	"1898-05-10"^^xsd:date	"1941-09-11"^^xsd:date	:Rudolph_Schoenheimer
"Walter Benjamin"@en	"1892-07-15"^^xsd:date	"1940-09-27"^^xsd:date	:Walter_Benjamin
"Walther Rathenau"@en	"1867-09-29"^^xsd:date	"1924-06-22"^^xsd:date	:Walther_Rathenau

57

También consultas más "españolas"

- ¿Qué toreros han estado casados con cantantes de copla? [Aquí](#)

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

SELECT ?torero ?cantante WHERE{

?torero rdf:type dbpedia-owl:BullFighter .

?torero dbpedia-owl:spouse ?cantante .

?cantante dcterms:subject <http://es.dbpedia.org/resource/

Categoría:Cantantes_de_coplas>

}

torero	cantante
http://es.dbpedia.org/resource/José_Ortega_Cano	http://es.dbpedia.org/resource/Rocío_Jurado
http://es.dbpedia.org/resource/Curro_Romero	http://es.dbpedia.org/resource/Concha_Márquez_Piquer
http://es.dbpedia.org/resource/Francisco_Rivera	http://es.dbpedia.org/resource/Isabel_Pantoja

58

Un consulta sencilla en SPARQL

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"

- Un patrón de consulta se *ajusta* contra los datos en RDF
- En todo caso un patrón puede ajustarse para proporcionar una solución
- La secuencia de soluciones se puede filtrar
- Se aplica uno de los siguientes constructores: SELECT, CONSTRUCT, DESCRIBE, ASK

59

Varios patrones de grafos: *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>

60

Literales RDF

```

@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>

61

Nodos en blanco

```

@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
_:c	"Alice"
_:d	"Bob"

=

x	name
_:r	"Alice"
_:s	"Bob"

62

Grupos de patrones

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

SELECT ?x
WHERE {}

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE { { ?x foaf:name ?name . }
        { ?x foaf:mbox ?mbox . FILTER regex(?name, "Smith")}
      }
```

63

Patrones opcionales (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"	

64

Patrones opcionales múltiples

```
@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>	

65

Patrones de alternativas

```
@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"

66

Grafos con nombre

```
# 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 .
```

67

Grafos con nombre

```
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"

68

Filtros y restricciones

```
@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")
}
```

title
"SPARQL Tutorial"

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

title
"The Semantic Web"

```
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	price
"The Semantic Web"	23

69

Tests de valores

- Basado en funciones y operadores XQuery 1.0 y XPath 2.0
- XSD boolean, string, integer, decimal, float, double, dateTime
- Anotaciones <, >, =, <=, >= y != para comparación de valores
Aplicable a cualquier tipo
- BOUND, isURI, isBLANK, isLITERAL
- REGEX, LANG, DATATYPE, STR ()
- Cualquier función extendida

70

Modificadores

- **Modificador de orden:** pone las soluciones en un cierto orden

```
SELECT ?name
WHERE { ?x foaf:name ?name ; :empId ?emp }
ORDER BY ?name DESC(?emp)
```
- **Modificador de proyección:** selecciona ciertas variables

```
SELECT ?name
WHERE
{ ?x foaf:name ?name }
```
- **Modificador DISTINCT:** asegura que las soluciones son únicas

```
SELECT DISTINCT ?name
WHERE { ?x foaf:name ?name }
```
- **Modificador reductor:** permite la eliminación de algunas soluciones que no son únicas

```
SELECT REDUCED ?name
WHERE { ?x foaf:name ?name }
```
- **Modificador de límite:** restringe el número de soluciones

```
SELECT ?name
WHERE { ?x foaf:name ?name }
LIMIT 20
```
- **Modificador Offset:** controla dónde comienzan las soluciones en la secuencia global de soluciones

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

71

Ejercicio 2



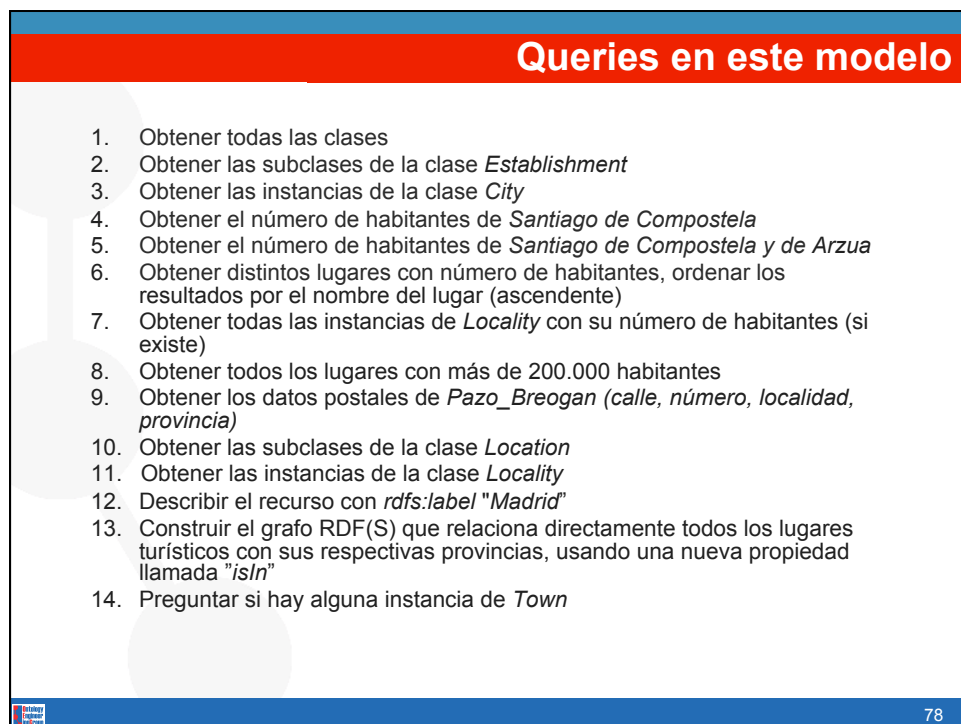
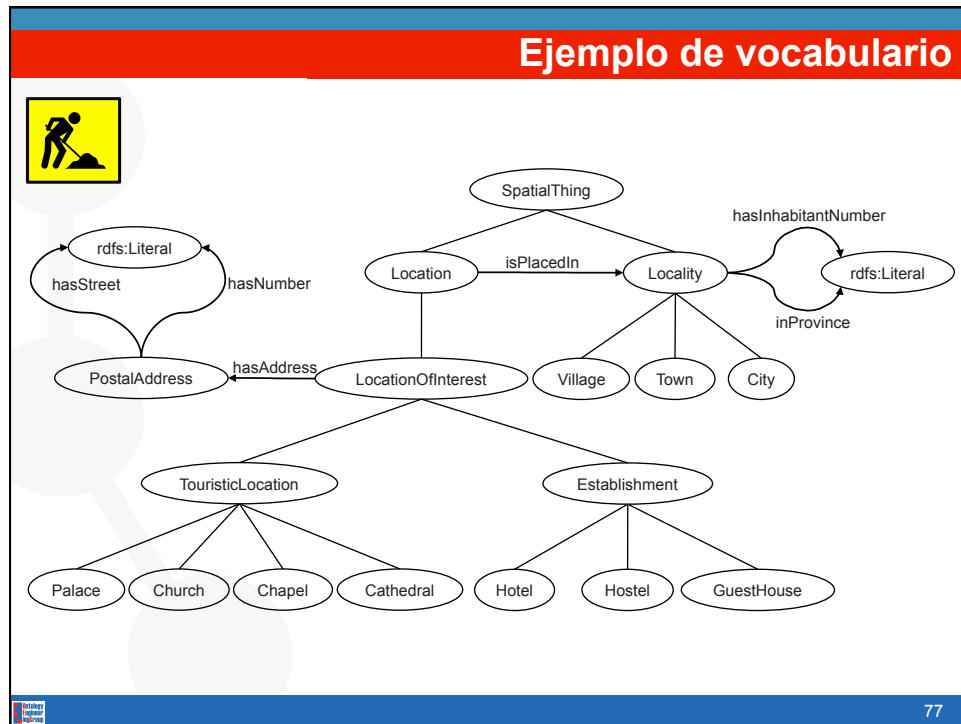
•Objetivo

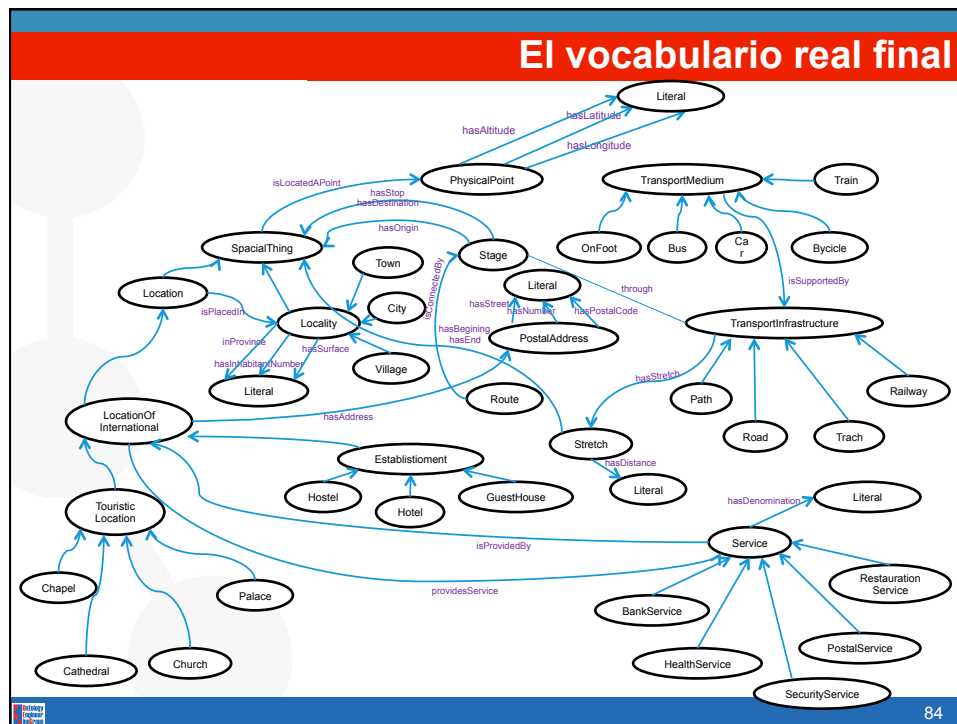
- Comprender cómo trabajar con consultas SPARQL

•Tareas

- Crear un conjunto de consultas SPARQL sobre un vocabulario de ejemplo
 - <http://sandbox.linkeddata.es/sparql>
 - En el grafo: http://sandbox.linkeddata.es/Grado_20122013

76





Ejercicio 2



•**Objetivo**

- Comprender cómo trabajar con consultas SPARQL algo más complejas

•**Tareas**

- Crear un conjunto de consultas SPARQL sobre Dbpedia
 - <http://dbpedia.org/sparql>
 - En el grafo por defecto

85

Consultas a realizar

1. Dame todas las propiedades aplicables a las instancias de la clase Politician (<<http://dbpedia.org/ontology/Politician>>)
2. Dame todas las propiedades, exceptuando rdf:type, aplicables a las instancias de la clase Politician
3. ¿Cuántos valores distintos se pueden encontrar para las propiedades, exceptuando rdf:type, de las instancias de la clase Politician)?
4. Para cada una de las propiedades, exceptuando rdf:type, aplicables a las instancias de la clase Politician, dime cuántos valores distintos toman en dichas instancias.
5. Para cada una de las propiedades, exceptuando rdf:type, aplicables a las instancias de la clase Politician, dime la media de valores distintos que toman en dichas instancias.
6. Para cada una de las propiedades, exceptuando rdf:type, aplicables a las instancias de la clase Politician, dime el máximo número de valores distintos que toman entre todas sus instancias, ordenado de mayor a menor
<<Dicho de otro modo, dime qué propiedades son las más comunes entre las instancias de la clase Politician>>

86

Ejercicio 3



•Objetivo

- Aprender a realizar consultas SPARQL complejas

•Tareas

- Crear un conjunto de consultas SPARQL sobre un vocabulario real
 - <http://es.dbpedia.org/sparql>

90

Queries avanzadas

1. Dame todas las propiedades aplicables a las instancias de la clase Politician (<<http://dbpedia.org/ontology/Politician>>)
2. Dame todas las propiedades, exceptuando `rdf:type`, aplicables a las instancias de la clase Politician (<<http://dbpedia.org/ontology/Politician>>)
3. ¿Cuántos valores distintos se pueden encontrar para las propiedades, exceptuando `rdf:type`, de las instancias de la clase Politician (<<http://dbpedia.org/ontology/Politician>>)?
4. Para cada una de las propiedades, exceptuando `rdf:type`, aplicables a las instancias de la clase Politician (<<http://dbpedia.org/ontology/Politician>>), dime cuántos valores distintos toman en dichas instancias.
5. Para cada una de las propiedades, exceptuando `rdf:type`, aplicables a las instancias de la clase Politician (<<http://dbpedia.org/ontology/Politician>>), dime la media de valores distintos que toman en dichas instancias.
6. Para cada una de las propiedades, exceptuando `rdf:type`, aplicables a las instancias de la clase Politician (<<http://dbpedia.org/ontology/Politician>>), dime el máximo número de valores distintos que toman entre todas sus instancias, ordenado de mayor a menor

91

Main References

- SPARQL 1.1 Overview
<http://www.w3.org/TR/sparql11-overview/>
- SPARQL validator:
<http://www.sparql.org/query-validator.html>
<http://yasgui.laurensrietveld.nl/>
- SPARQL implementations:
<http://esw.w3.org/topic/SparqlImplementations>

92