

**CC6202-1**

**LA WEB DE DATOS**

**PRIMAVERA 2016**

## **Lecture 11: Conclusion**

Aidan Hogan

aidhog@gmail.com

# We've gotten from here ...

## The current Web is document-centric

<http://ex.org/Ireland>



<http://ex.org/Dublin>



```
<html>
  <body>
    <h1>Ireland</h1>
    <div class="flag"></div>
    <p>Ireland is a European country.</p>
    <p>Its capital is <a href="http://ex.org/Dublin">Dublin</a>.</p>
    ...
  </body>
</html>
```

# To here (with (4) Links!)

(1) Data, (2) Rules/Ontologies, (3) Query

INPUT:  $(x, \text{partOf}, y)$

DATA:

<http://ex.org/Ireland>



<http://ex.org/Dublin>



RULES:  $(a, \text{capital}, b) \rightarrow (b, \text{partOf}, a)$   
 $(c, \text{partOf}, d), (d, \text{partOf}, e) \rightarrow (c, \text{partOf}, e)$

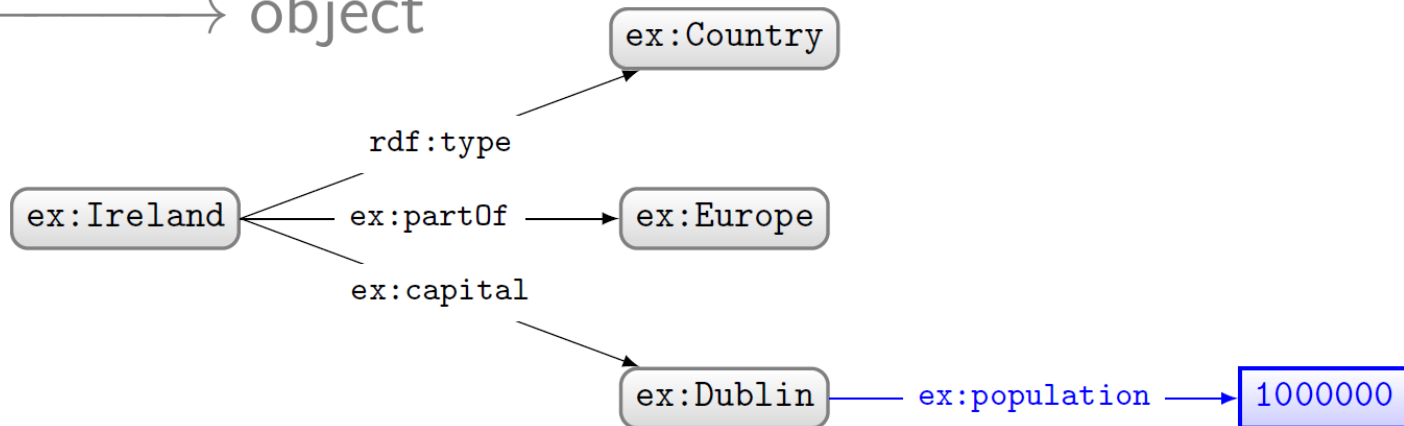
OUTPUT:  $\{(x \mapsto \text{Ireland}, y \mapsto \text{Europe}), (x \mapsto \text{Dublin}, y \mapsto \text{Ireland})$   
 $(x \mapsto \text{Dublin}, y \mapsto \text{Europe})\}$

**DATA? RDF!**

# Use RDF as a core data model

<i>subject</i>	<i>predicate</i>	<i>object</i>
ex:Ireland	ex:partOf	ex:Europe
ex:Ireland	rdf:type	ex:Country
ex:Ireland	ex:capital	ex:Dublin
ex:Dublin	ex:population	1,000,000

subject  $\xrightarrow{\text{predicate}}$  object



# A Summary of RDF Terms

## 1. IRIs (Internationalised Resource Identifiers)

- Used to name generic things

## 2. Literals

- Used to refer to datatype values
- Strings may have a language tag

## 3. Blank Nodes

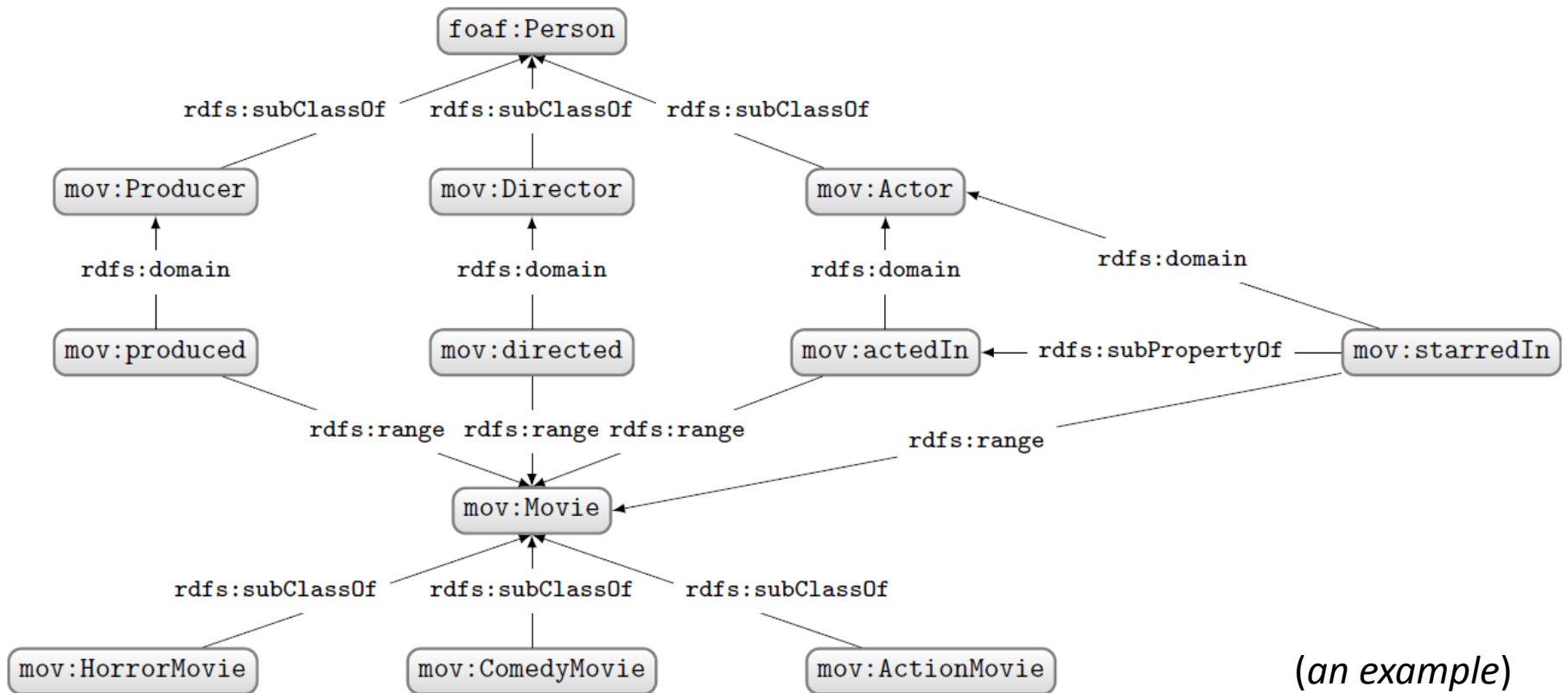
- Used to avoid naming things
- A little mysterious right now

<i>subject</i>	<i>predicate</i>	<i>object</i>
[IRI, Blank Node]	[IRI]	[IRI, Blank Node, Literal]

**RULES/ONTOLOGIES? RDFS!**

# RDF Schema: lightweight semantics

Let's model an RDF Schema for Movies, including different types of movies, some different types of people involved, and how they are related.



*(an example)*



# RDFS: Describe “schema” in RDF

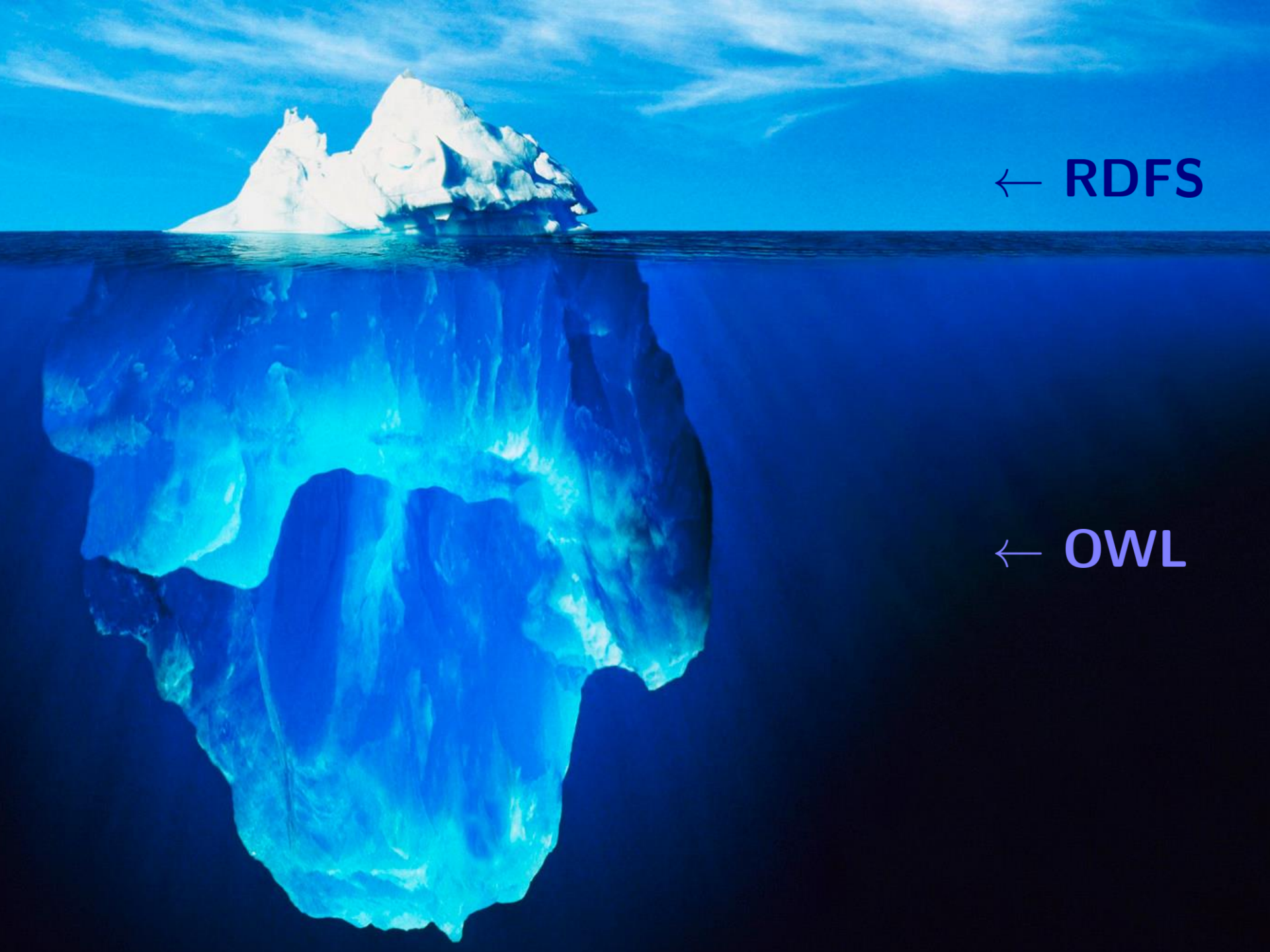
- Sub-class:
  - `ex:CapitalCity rdfs:subClassOf ex:City .`
- Sub-property:
  - `ex:hasCapitalCity rdfs:subPropertyOf ex:hasCity .`
- Domain:
  - `foaf:familyName rdfs:domain foaf:Person .`
- Range:
  - `ex:hasCapitalCity rdfs:range ex:CapitalCity .`
  - `foaf:familyName rdfs:range xsd:string .`

# Apply RDFS reasoning using “rules”

ID	if $G$ matches	then $G$ RDFS <sub>D</sub> -entails
rdfD1	$?x ?p ?l . ( ?l \text{ a literal with datatype IRI } dt(?l) \in D )$	$?x ?p \_ :b . \_ :b \text{ a } dt(?l) .$
rdfD2	$?x ?p ?y .$	$?p \text{ a } rdf:Property .$
rdfs1	$?u \in D$	$?u \text{ a } rdfs:Datatype .$
rdfs2	$?p \text{ rdfs:domain } ?c . ?x ?p ?y .$	$?x \text{ a } ?c .$
rdfs3	$?p \text{ rdfs:range } ?c . ?x ?p ?y .$	$?y \text{ a } ?c .$
rdfs4a	$?x ?p ?y .$	$?x \text{ a } rdfs:Resource .$
rdfs4b	$?x ?p ?y .$	$?y \text{ a } rdfs:Resource .$
rdfs5	$?p \text{ rdfs:subPropertyOf } ?q . ?x ?p ?y .$	$?x ?q ?y .$
rdfs6	$?p \text{ a } rdf:Property .$	$?p \text{ rdfs:subPropertyOf } ?p .$
rdfs7	$?p \text{ rdfs:subPropertyOf } ?q . ?q \text{ rdfs:subPropertyOf } ?r .$	$?p \text{ rdfs:subPropertyOf } ?r .$
rdfs8	$?c \text{ a } rdfs:Class .$	$?c \text{ rdfs:subClassOf } rdfs:Resource .$
rdfs9	$?c \text{ rdfs:subClassOf } ?d . ?x \text{ a } ?c .$	$?x \text{ a } ?d .$
rdfs10	$?c \text{ a } rdfs:Class .$	$?c \text{ rdfs:subClassOf } ?c .$
rdfs11	$?c \text{ rdfs:subClassOf } ?d . ?d \text{ rdfs:subClassOf } ?e .$	$?c \text{ rdfs:subClassOf } ?e .$
rdfs12	$?p \text{ a } rdfs:ContainerMembershipProperty .$	$?p \text{ rdfs:subPropertyOf } rdfs:member .$
rdfs13	$?d \text{ a } rdfs:Datatype .$	$?d \text{ rdfs:subClassOf } rdf:Literal .$

(Don't worry about rdfD1, rdfs1, rdfs12, rdfs13)

**RULES/ONTOLOGIES? OWL!**



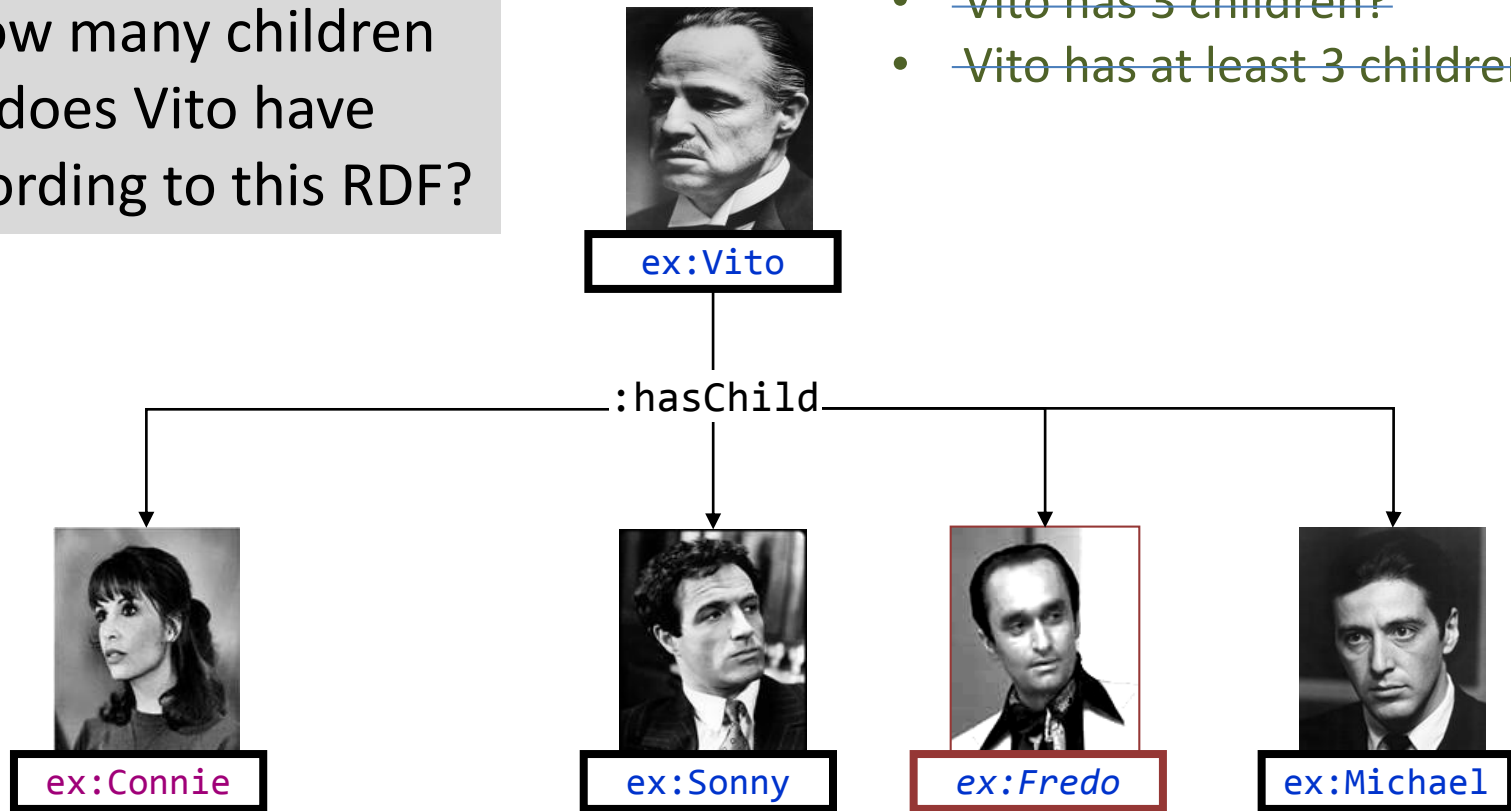
← **RDFS**

← **OWL**

# Open World Assumption (OWA)

How many children does Vito have according to this RDF?

- ~~Vito has 3 children?~~
- ~~Vito has at least 3 children?~~



`ex:Vito :hasChild ex:Connie, ex:Sonny, ex:Michael .`

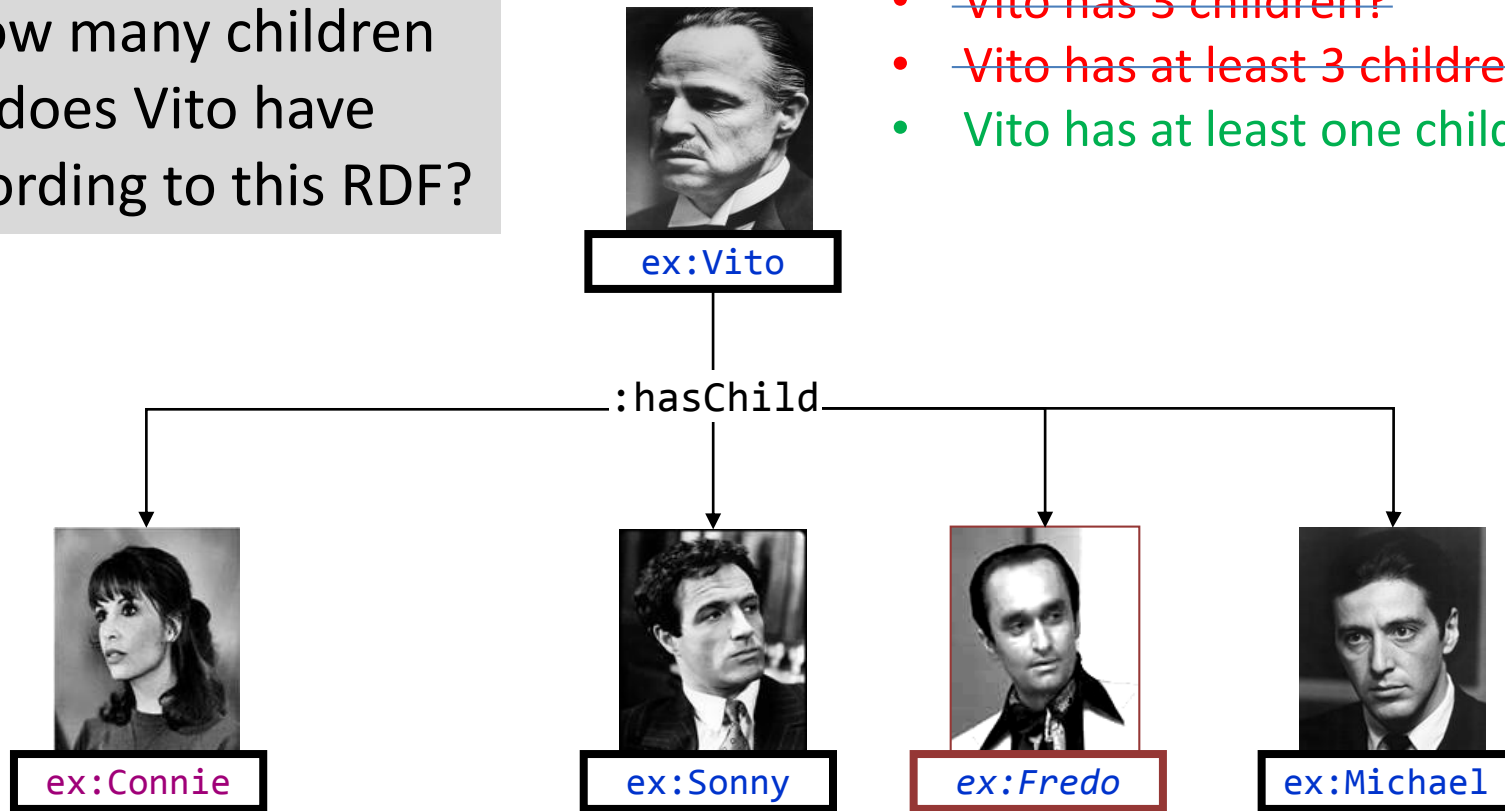
*`ex:Vito :hasChild ex:Fredo .`*

*`... ?`*

# No Unique Name Assumption (No UNA)

How many children does Vito have according to this RDF?

- ~~Vito has 3 children?~~
- ~~Vito has at least 3 children?~~
- Vito has at least one child!

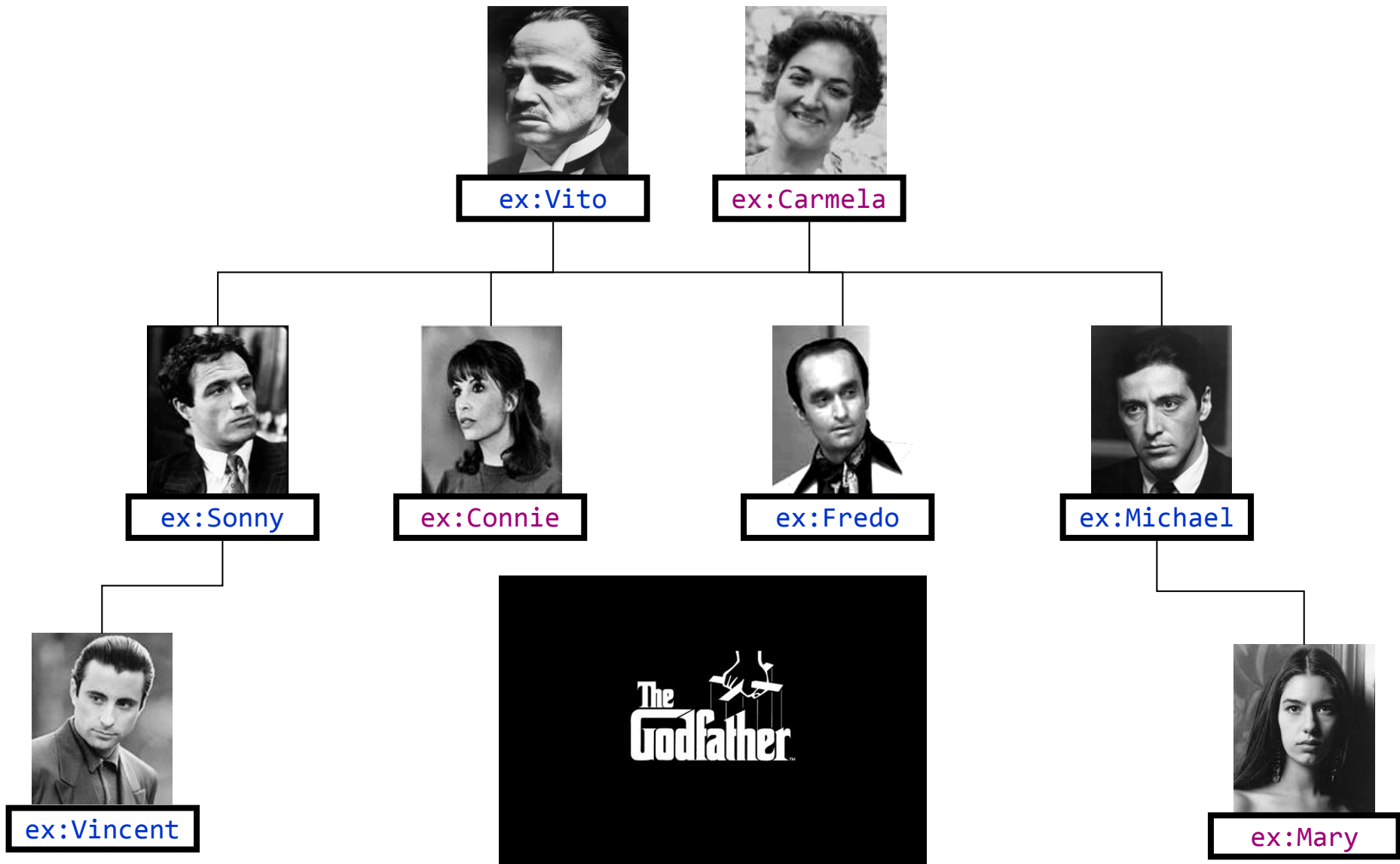


`ex:Vito :hasChild ex:Connie, ex:Sonny, ex:Michael .`

*`ex:Vito :hasChild ex:Fredo .`*

*`... ?`*

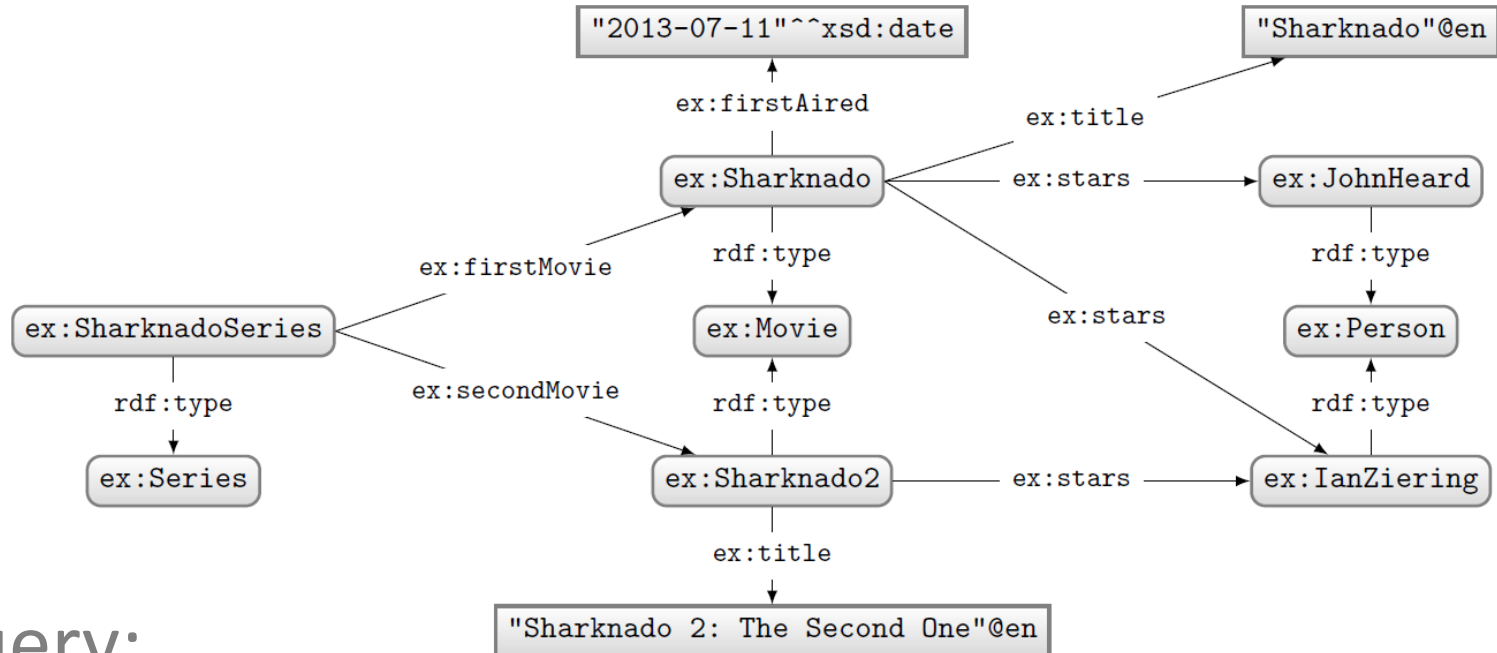
# All the features ...



QUERY? **SPARQL (1.1)**



# SPARQL 1.0: Query Features



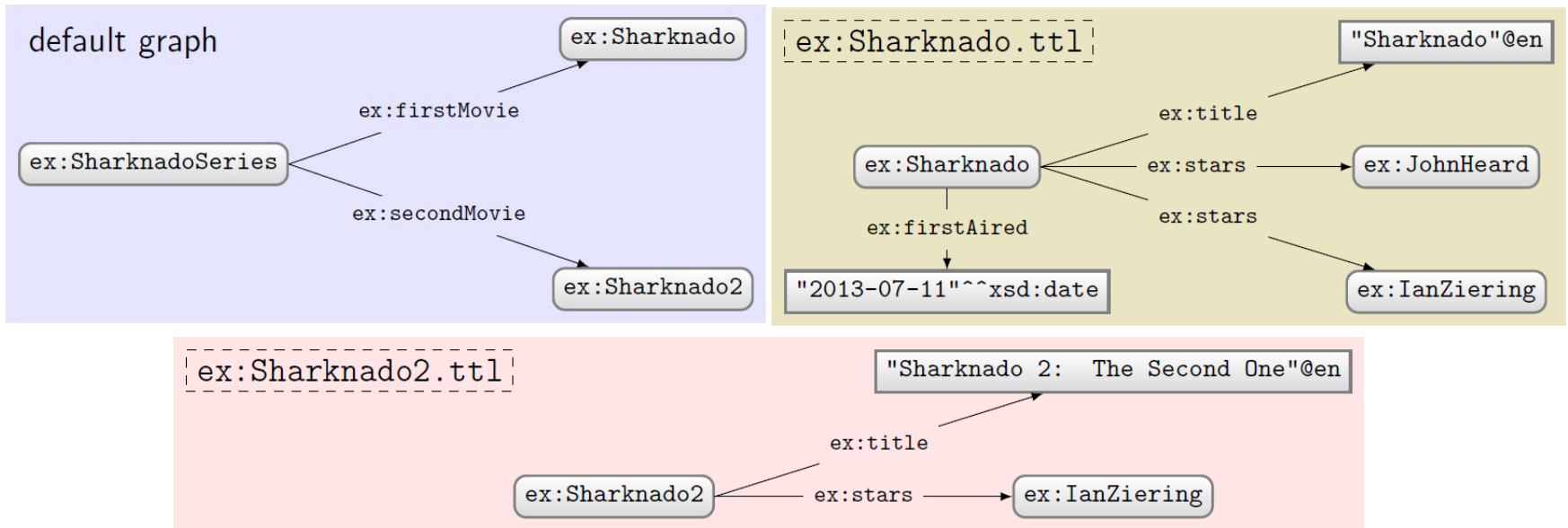
Query:

```
PREFIX ex: <http://ex.org/voc#>
SELECT *
WHERE {
  { ex:SharknadoSeries ex:firstMovie ?movie . }
  UNION
  { ex:SharknadoSeries ex:secondMovie ?movie . }
  OPTIONAL
  { ?movie ex:firstAired ?date . }
  ?movie ex:title ?title .
  FILTER(REGEX(STR(?title),"*[0-9]*"))
}
```

Solutions:

?movie	?title	?date
ex:Sharknado2	"Sharknado 2: The Second One"@en	

# SPARQL 1.0: Named Graphs



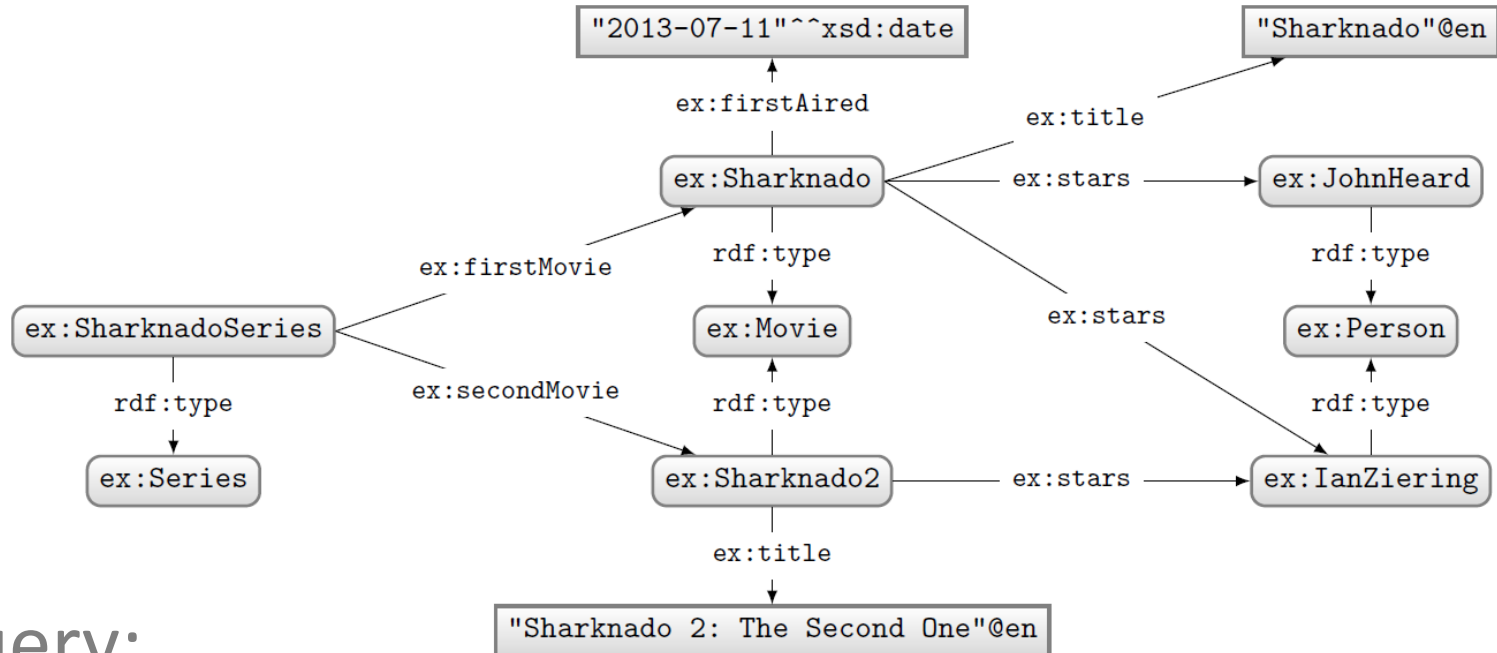
Query:

```
PREFIX ex: <http://ex.org/voc#>
FROM ex:Sharknado2.ttl
FROM NAMED ex:Sharknado.ttl
SELECT DISTINCT ?x ?q
WHERE {
  GRAPH ?g { ?s ?p ?o }
  ?x ?q ?o .
}
```

Solutions:

?x	?q
ex:Sharknado2	ex:stars

# SPARQL 1.1: Query Features



Query:

```
PREFIX ex: <http://ex.org/voc#>
SELECT (COUNT(?star) as ?count)
WHERE {
    ?movie ex:stars ?star .
}
```

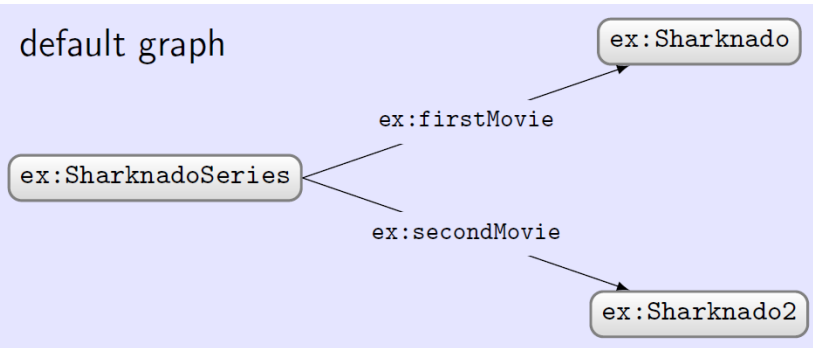
Solutions:

?count
--------

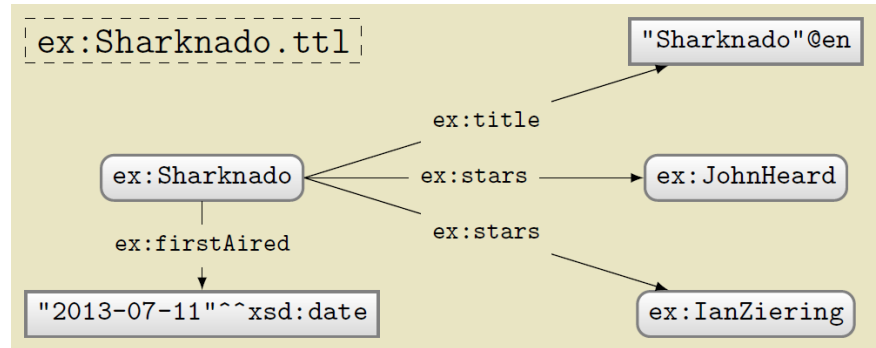
3
---

# SPARQL 1.1: Update

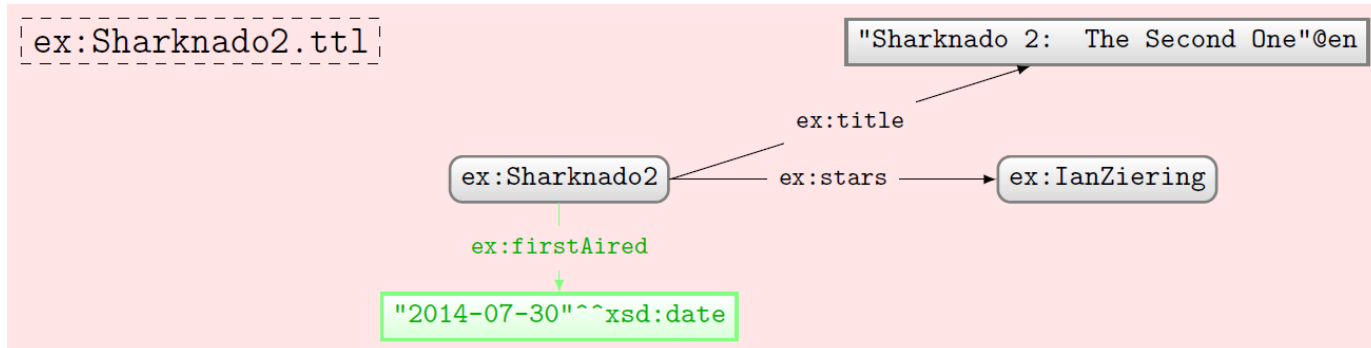
default graph



`ex:Sharknado.ttl`



`ex:Sharknado2.ttl`



```
PREFIX ex: <http://ex.org/voc#>
INSERT DATA {
  GRAPH ex:Sharknado2.ttl
    { ex:Sharknado2 ex:firstAired "2014-07-30"^^xsd:date . }
}
```

**LINKS? LINKED DATA**

# IRIs link to other RDF documents ...

```
@base <http://anakena.dcc.uchile.cl/~calvarez/foaf.ttl> .
```

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
```

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
```

```
<#me> a foaf:Person;  
      foaf:name "Camila Andrea Álvarez"@es;  
      foaf:firstName "Camila"@es;  
      foaf:surname "Álvarez"@es;  
      foaf:birthday "02-24";  
      foaf:gender "female";  
      foaf:homepage <http://anakena.dcc.uchile.cl/~calvarez/cv.html>;  
      foaf:based_near <http://sws.geonames.org/3871336/>;  
      foaf:interest <http://dbpedia.org/resource/Tales_(series)>,  
      <http://dbpedia.org/resource/Embroidery>;  
      foaf:img <http://images.evisos.cl/2009/06/03/erizo-de-tierra_9de6128c_3.jpg>;  
      foaf:schoolHomepage <http://www.uchile.cl/>, <http://www.dcc.uchile.cl/>;  
      foaf:knows <http://anakena.dcc.uchile.cl/~jasalas/web/foaf.ttl#me>,  
      <http://anakena.dcc.uchile.cl/~jogarriid/web/foaf.ttl#yo>,  
      <http://anakena.dcc.uchile.cl/~ekauffma/foaf.ttl#eli> .
```

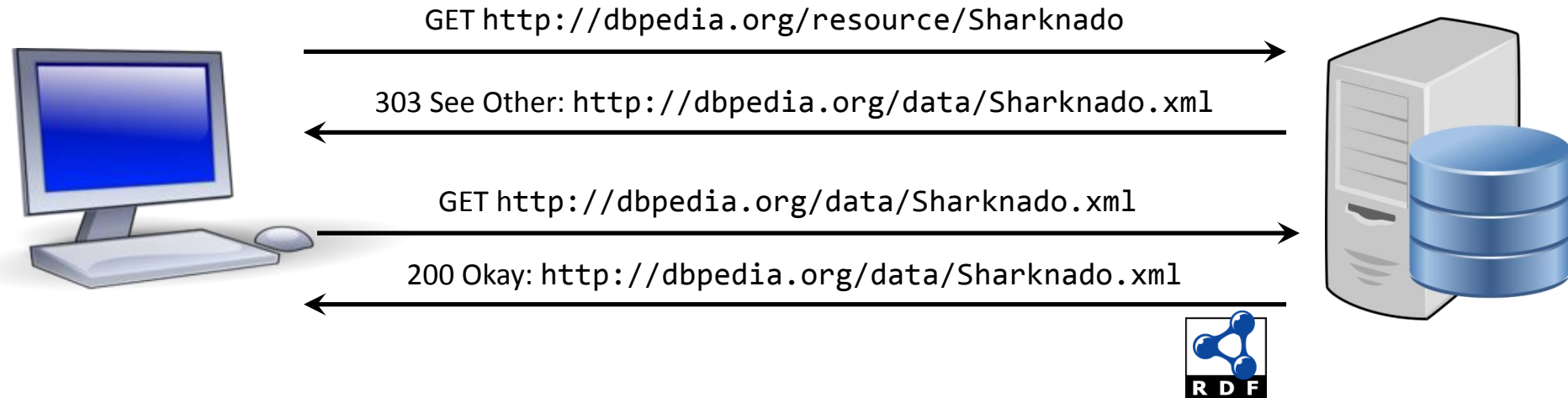
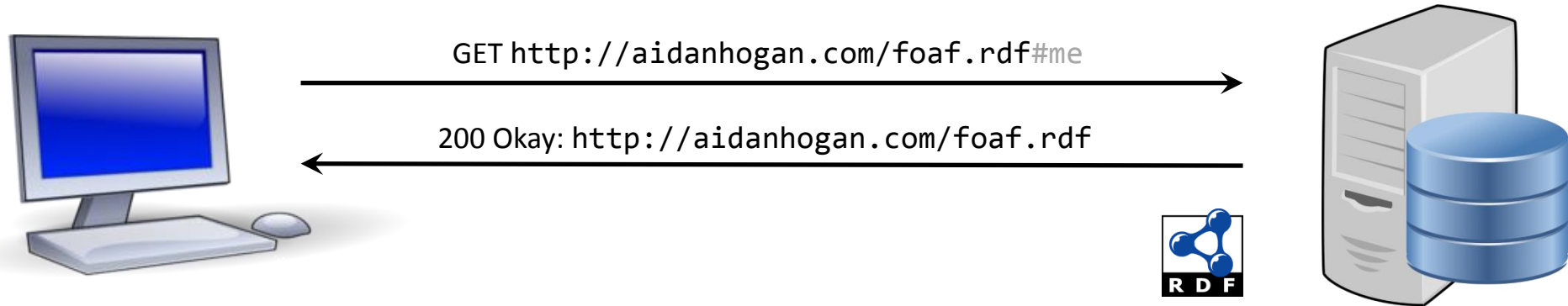
# Four Principles of Linked Data

<http://www.w3.org/DesignIssues/LinkedData.html>



1. Use URIs as names for things
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF\*, SPARQL)
4. Include links to other URIs. so that they can discover more things.

# Hash vs. Slash





# The 5 ★'s of Linked Open Data

- ★ Publish data under open licence
- ★★ Make the data “machine readable”
  - e.g., a Spreadsheet better than a PDF table
- ★★★ Use non-proprietary formats
  - e.g., a CSV text file better than Excel
- ★★★★★ Use URIs to name your stuff ([hint: RDF](#))
  - use unambiguous identifiers that can be linked/looked up
- ★★★★★★ Provide links to other content
  - so consumers can follow links to find out more





# Basics in place (with (4) Links!)

(1) Data, (2) Rules/Ontologies, (3) Query

INPUT:  $(x, \text{partOf}, y)$

DATA:

<http://ex.org/Ireland>



<http://ex.org/Dublin>



RULES:  $(a, \text{capital}, b) \rightarrow (b, \text{partOf}, a)$   
 $(c, \text{partOf}, d), (d, \text{partOf}, e) \rightarrow (c, \text{partOf}, e)$

OUTPUT:  $\{(x \mapsto \text{Ireland}, y \mapsto \text{Europe}), (x \mapsto \text{Dublin}, y \mapsto \text{Ireland})$   
 $(x \mapsto \text{Dublin}, y \mapsto \text{Europe})\}$

**IMPORTING LEGACY DATA  
(E.G., RDB2RDF)**

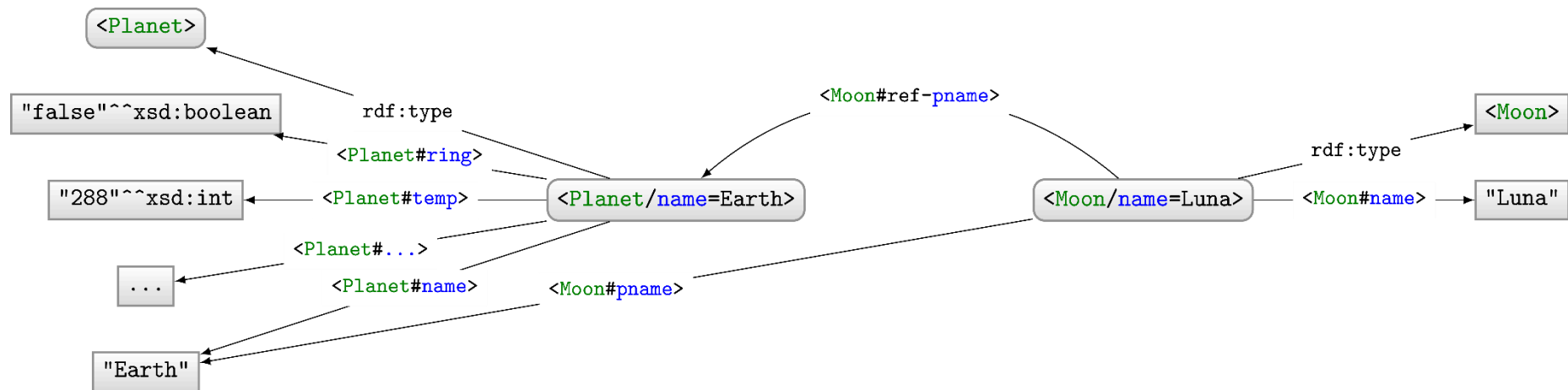
# Direct Mapping

## Planet

name	dist	radius	grav	day	year	temp	ring
Mercury	0.39	0.38	2.8	58.646	0.241	440	false
Venus	0.72	0.95	8.9	-243.019	0.615	730	false
Earth	1.00	1.00	9.8	0.997	1.000	288	false
Mars	1.52	0.53	3.7	1.026	1.880	186	false
Jupiter	5.20	10.97	22.9	0.414	11.862	152	true
Saturn	9.54	9.14	9.1	0.444	29.447	134	true
Uranus	19.19	3.98	7.8	-0.719	84.017	76	true
Neptune	30.07	3.86	11.0	0.671	164.791	53	true

## Moon

name	pname	discoverer	year
Luna	Earth	⊥	⊥
Ganymedes	Jupiter	Galileo Galilei	1610
Calisto	Jupiter	Galileo Galilei	1610
Europa	Jupiter	Galileo Galilei	1610
Io	Jupiter	Galileo Galilei	1610
Titan	Saturn	Christiaan Huygens	1655
Triton	Neptune	William Lassell	1846





# R2RML

```
@prefix ...
```

```
<OuterPlanetMap> a rr:TriplesMap ;
```

```
  rr:logicalTable [
```

```
    rr:sqlQuery "SELECT * FROM Planet WHERE dist>1" ;
```

```
    rr:sqlVersion rr:SQL2008
```

```
  ] ;
```

```
  rr:subjectMap [
```

```
    rr:template "http://ex.org/Planet/name={name}" ;
```

```
    rr:class <http://ex.org/OuterPlanet>
```

```
  ] .
```

## Planet

name	dist	radius	grav	day	year	temp	ring
Mercury	0.39	0.38	2.8	58.646	0.241	440	false
Venus	0.72	0.95	8.9	-243.019	0.615	730	false
Earth	1.00	1.00	9.8	0.997	1.000	288	false
Mars	1.52	0.53	3.7	1.026	1.880	186	false
Jupiter	5.20	10.97	22.9	0.414	11.862	152	true
Saturn	9.54	9.14	9.1	0.444	29.447	134	true
Uranus	19.19	3.98	7.8	-0.719	84.017	76	true
Neptune	30.07	3.86	11.0	0.671	164.791	53	true

```
@base <http://ex.org/>
```

```
@prefix ...
```

```
<Planet/name=Mars> a <OuterPlanet> .
```

```
<Planet/name=Jupiter> a <OuterPlanet> .
```

```
<Planet/name=Saturn> a <OuterPlanet> .
```

```
<Planet/name=Neptune> a <OuterPlanet> .
```

**... THE WEB OF DATA**

# The Web of Data (with (4) Links!)

(1) Data, (2) Rules/Ontologies, (3) Query

INPUT:  $(x, \text{partOf}, y)$

DATA:

<http://ex.org/Ireland>



<http://ex.org/Dublin>



RULES:  $(a, \text{capital}, b) \rightarrow (b, \text{partOf}, a)$   
 $(c, \text{partOf}, d), (d, \text{partOf}, e) \rightarrow (c, \text{partOf}, e)$

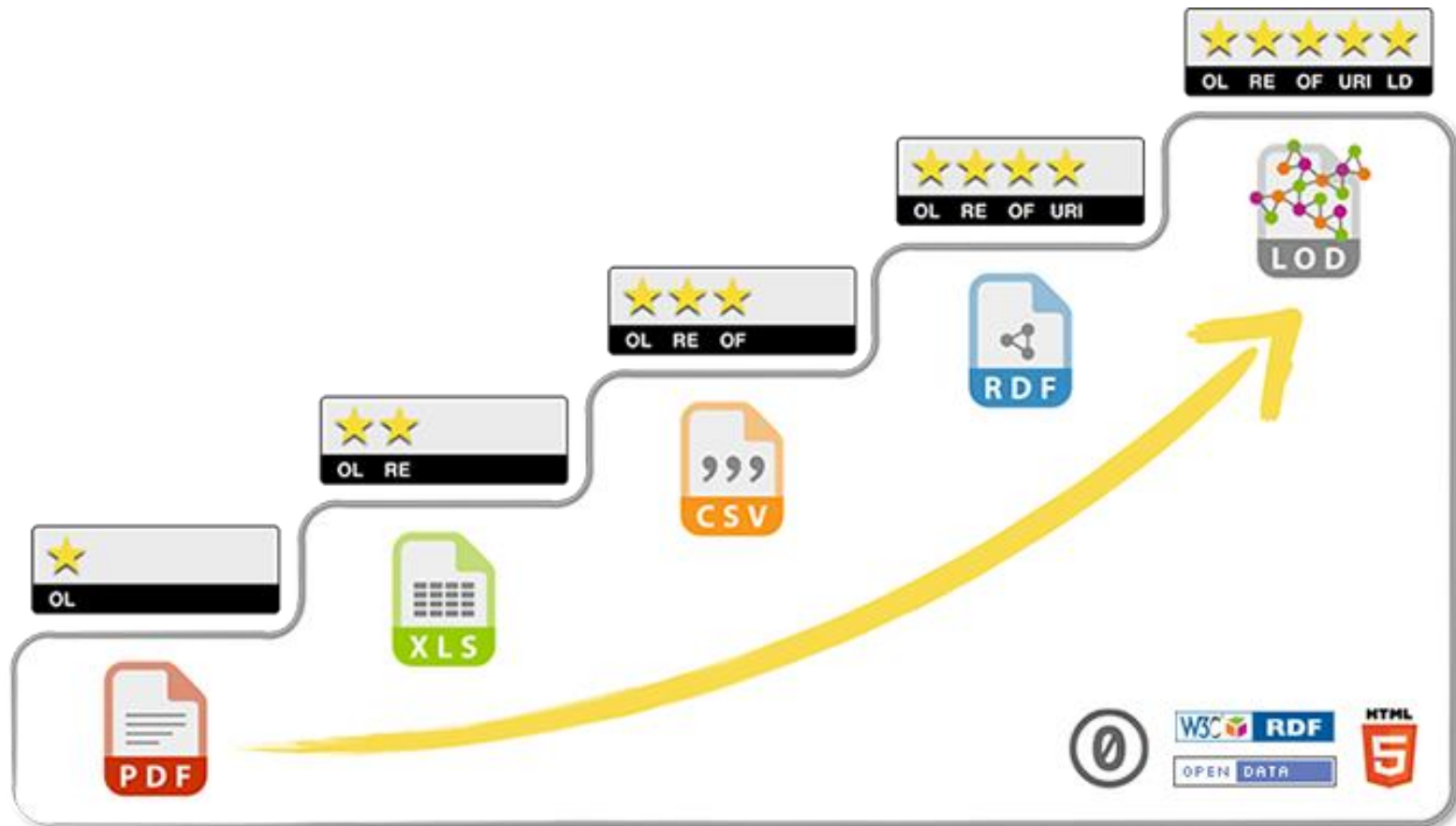
OUTPUT:  $\{(x \mapsto \text{Ireland}, y \mapsto \text{Europe}), (x \mapsto \text{Dublin}, y \mapsto \text{Ireland})$   
 $(x \mapsto \text{Dublin}, y \mapsto \text{Europe})\}$





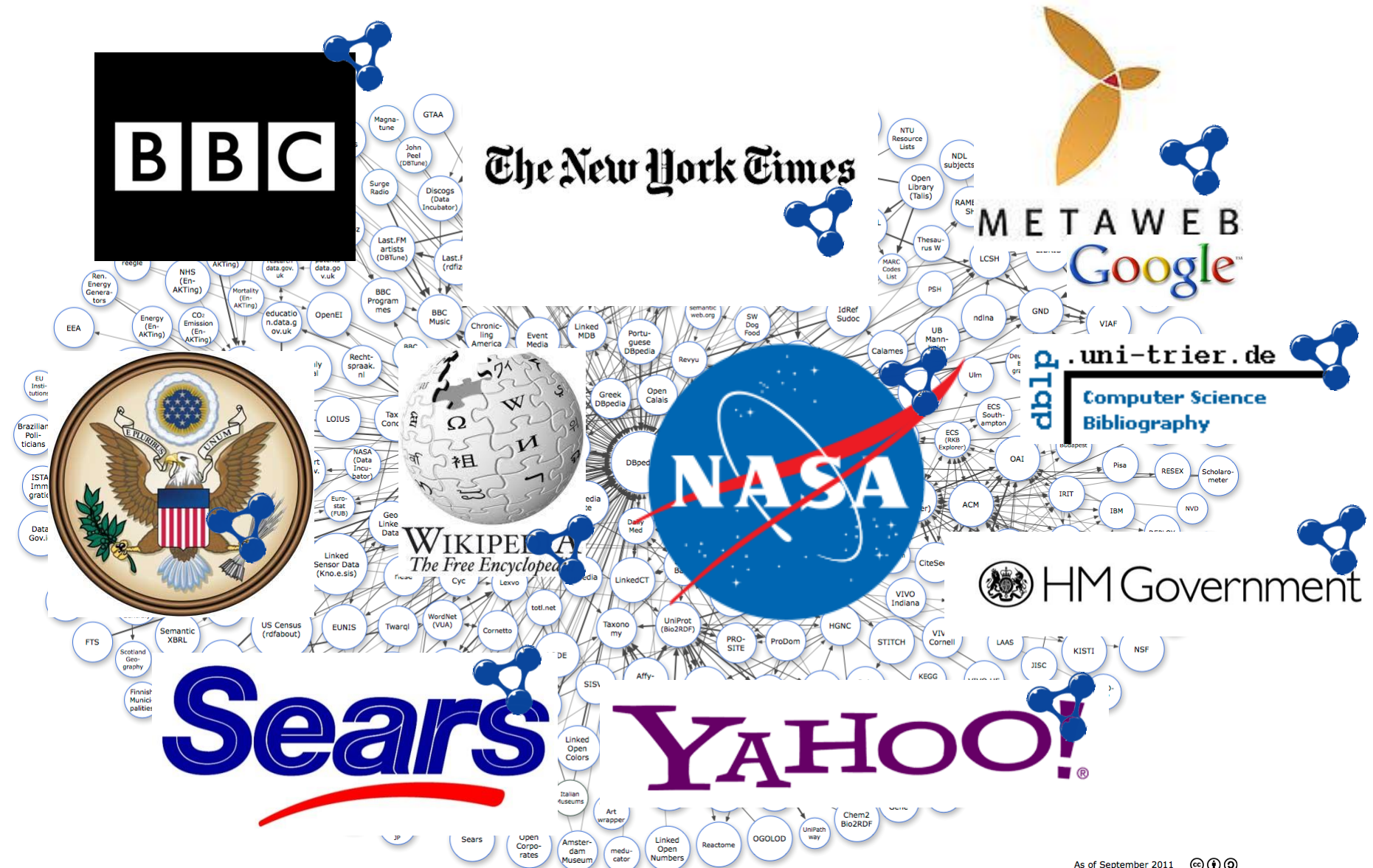
STARTING TO TAKE OFF

# Publishing Open Data

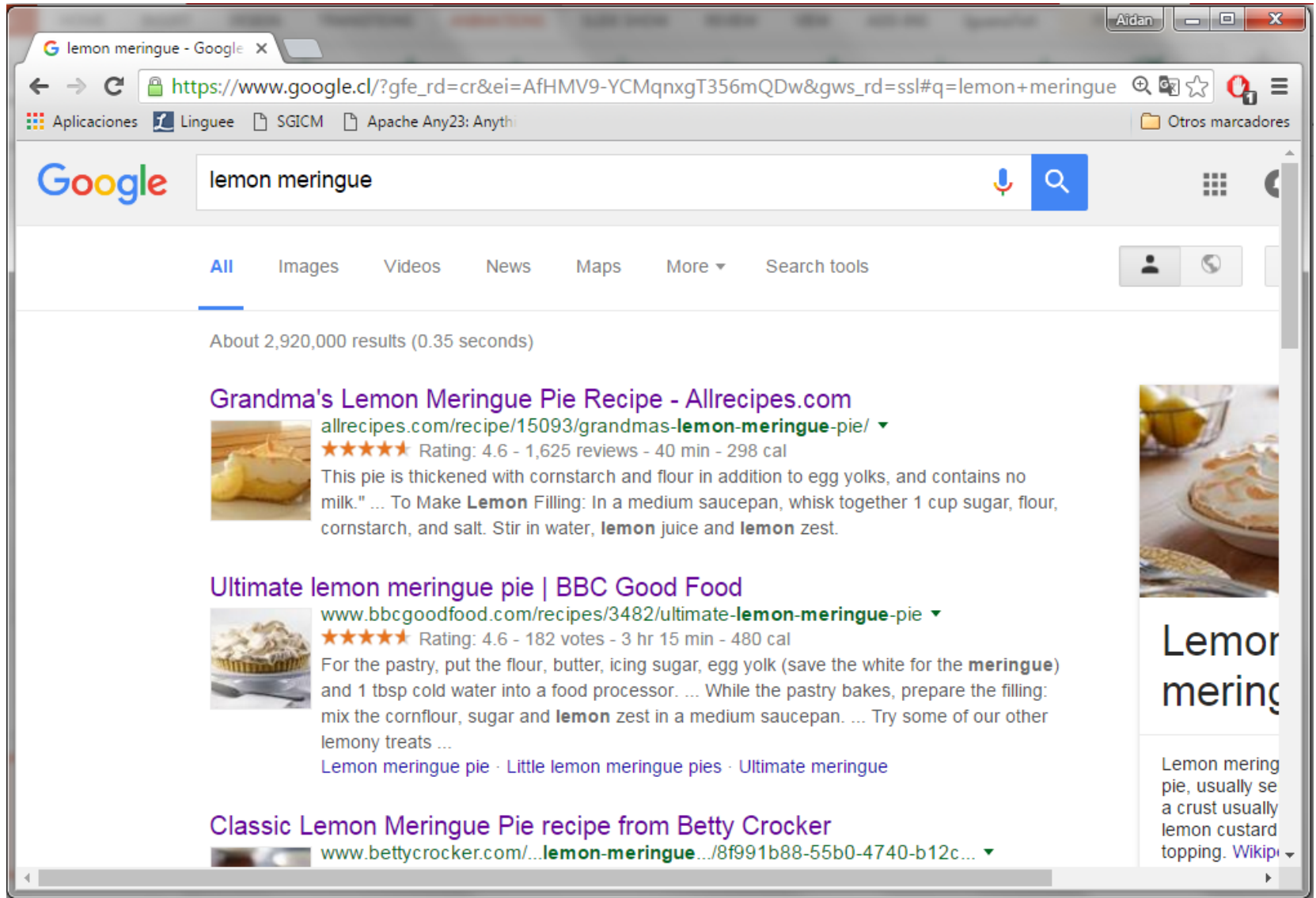




# The Linked Data Cloud



# Google's Rich Snippets



A screenshot of a Google search for "lemon meringue". The browser window shows the Google homepage with the search bar containing "lemon meringue". Below the search bar, the results are displayed under the "All" tab. The first result is "Grandma's Lemon Meringue Pie Recipe - Allrecipes.com", which includes a thumbnail image of a pie, a star rating of 4.6, and a brief description. The second result is "Ultimate lemon meringue pie | BBC Good Food", also with a thumbnail, a 4.6 rating, and a description. The third result is "Classic Lemon Meringue Pie recipe from Betty Crocker". On the right side of the page, there is a vertical sidebar with a large image of a pie and the text "Lemon meringue pie, usually se a crust usually lemon custard topping. Wikipi".

lemon meringue - Google

[https://www.google.cl/?gfe\\_rd=cr&ei=AfHmV9-YCMqnxgT356mQDw&gws\\_rd=ssl#q=lemon+meringue](https://www.google.cl/?gfe_rd=cr&ei=AfHmV9-YCMqnxgT356mQDw&gws_rd=ssl#q=lemon+meringue)

Aplicaciones Linguee SGICM Apache Any23: Anythi Otros marcadores

Google

lemon meringue

All Images Videos News Maps More Search tools

About 2,920,000 results (0.35 seconds)

**Grandma's Lemon Meringue Pie Recipe - Allrecipes.com**

[allrecipes.com/recipe/15093/grandmas-lemon-meringue-pie/](https://allrecipes.com/recipe/15093/grandmas-lemon-meringue-pie/)

★★★★★ Rating: 4.6 - 1,625 reviews - 40 min - 298 cal

This pie is thickened with cornstarch and flour in addition to egg yolks, and contains no milk." ... To Make **Lemon** Filling: In a medium saucepan, whisk together 1 cup sugar, flour, cornstarch, and salt. Stir in water, **lemon** juice and **lemon** zest.

**Ultimate lemon meringue pie | BBC Good Food**

[www.bbcgoodfood.com/recipes/3482/ultimate-lemon-meringue-pie](https://www.bbcgoodfood.com/recipes/3482/ultimate-lemon-meringue-pie)

★★★★★ Rating: 4.6 - 182 votes - 3 hr 15 min - 480 cal

For the pastry, put the flour, butter, icing sugar, egg yolk (save the white for the **meringue**) and 1 tbsp cold water into a food processor. ... While the pastry bakes, prepare the filling: mix the cornflour, sugar and **lemon** zest in a medium saucepan. ... Try some of our other lemony treats ...

[Lemon meringue pie](#) · [Little lemon meringue pies](#) · [Ultimate meringue](#)

**Classic Lemon Meringue Pie recipe from Betty Crocker**

[www.bettycrocker.com/.../lemon-meringue.../8f991b88-55b0-4740-b12c...](https://www.bettycrocker.com/.../lemon-meringue.../8f991b88-55b0-4740-b12c...)

Lemon meringue pie, usually se a crust usually lemon custard topping. Wikipi

# Google's Info-Box

A screenshot of a Google search for "sully prudhomme". The browser window shows the search bar with the text "sully prudhomme" and the Google logo. Below the search bar are tabs for "All", "Images", "Videos", "Books", "News", and "More". The search results show "About 372,000 results (0.32 seconds)". The first result is "Sully Prudhomme - Wikipedia, the free encyclopedia" with a link to [https://en.wikipedia.org/wiki/Sully\\_Prudhomme](https://en.wikipedia.org/wiki/Sully_Prudhomme). The second result is "Sully Prudhomme - Wikipedia, la enciclopedia libre" with a link to [https://es.wikipedia.org/wiki/Sully\\_Prudhomme](https://es.wikipedia.org/wiki/Sully_Prudhomme). The third result is "Sully Prudhomme - Biographical - Nobelprize.org" with a link to [www.nobelprize.org/nobel\\_prizes/literature/laureates/1901/prudhomme](http://www.nobelprize.org/nobel_prizes/literature/laureates/1901/prudhomme). The fourth result is "Sully Prudhomme | French poet | Britannica.com" with a link to <https://www.britannica.com/biography/Sully-Prudhomme>. The fifth result is "Sully Prudhomme - NNDB.com" with a link to [www.nndb.com/people/297/000098003/](http://www.nndb.com/people/297/000098003/).

**Sully Prudhomme**  
Poet

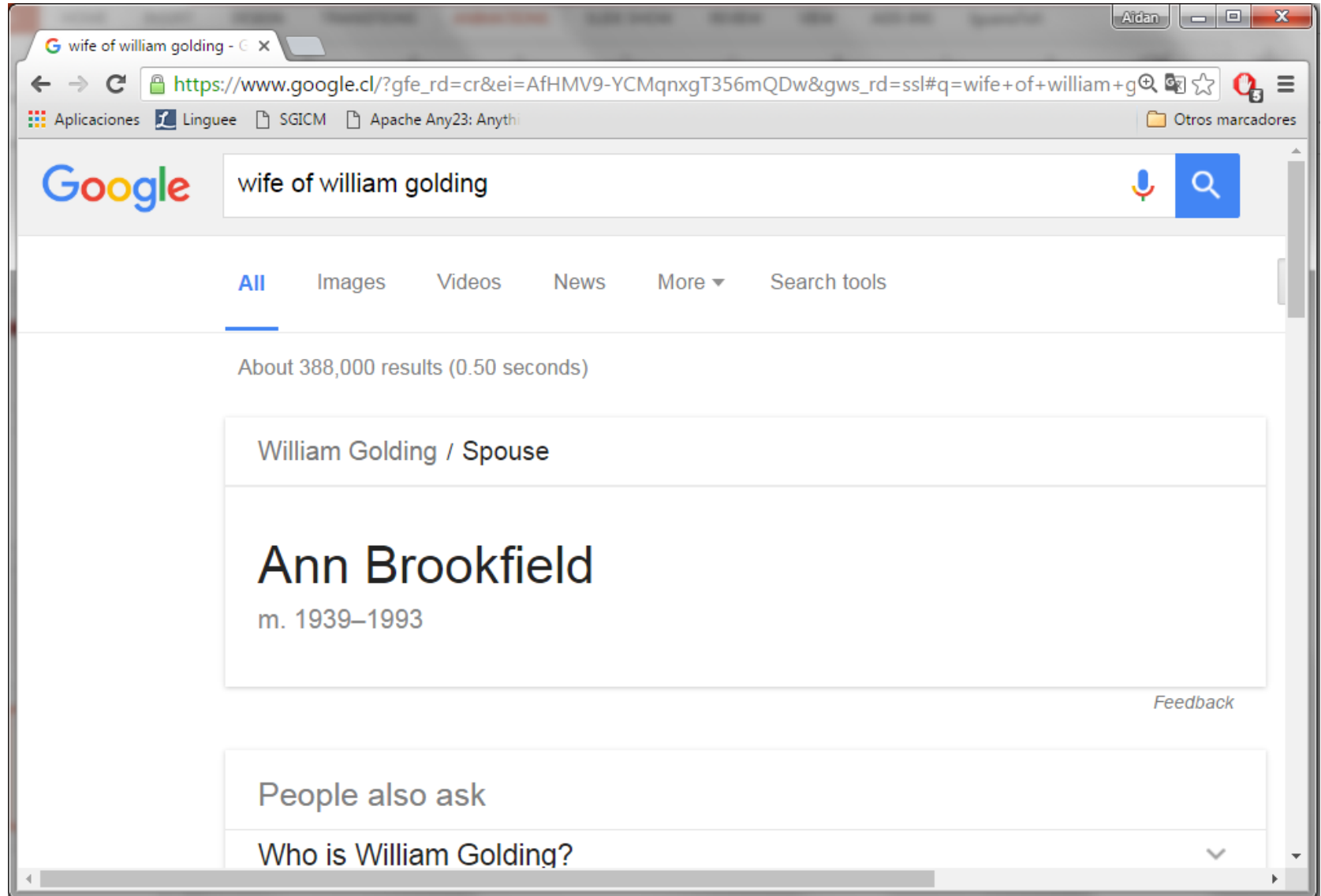
René François Armand Prudhomme was a French poet and essayist. He was the first ever winner of the Nobel Prize in Literature in 1901.  
[Wikipedia](#)

**Born:** March 16, 1839, [Paris, France](#)  
**Died:** September 6, 1907, [Châtenay-Malabry, France](#)  
**Books:** [Les vaines tendresses](#)  
**Awards:** [Nobel Prize in Literature](#)

**People also search for** [View 10+ more](#)

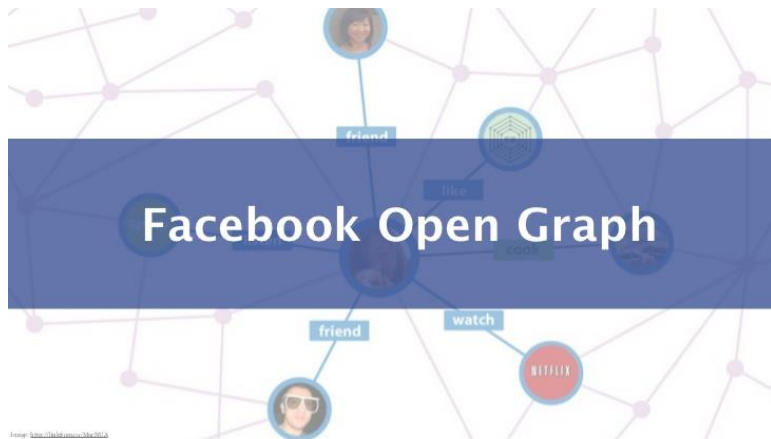
 <a href="#">Leconte de Lisle</a>	 <a href="#">Theodor Mommsen</a>	 <a href="#">Frédéric Mistral</a>	 <a href="#">Paul Verlaine</a>	 <a href="#">Gabriel Fauré</a>
--	--	---	--	--

# Google's Semantic Search





# Facebook's Open Graph Protocol



IMDb

16 hrs ·

The latest trailer for "Game of Thrones" is out, and contains so many fantastic details that we don't even know where to begin. Take a look: <http://imdb.to/1KXLPFJ>



"Game of Thrones": The Wheel

"I'm not going to stop the wheel. I'm going to break the wheel."

IMDB.COM

Like · Comment · Share · Send · 6,469 395 1,381

# Wikidata



## Abraham Lincoln (Q91)

American politician, 16th President of the United States in office from 1861 to 1865

[Abe Lincoln](#) | [Lincoln](#) | [Honest Abe](#)

► [In more languages](#)

place of death



[Petersen House](#)

[\[edit\]](#)

► [1 reference](#)

position held



[President of the United States of America](#)

[start time](#)

[4 March 18](#)

[end time](#)

[15 April 18](#)

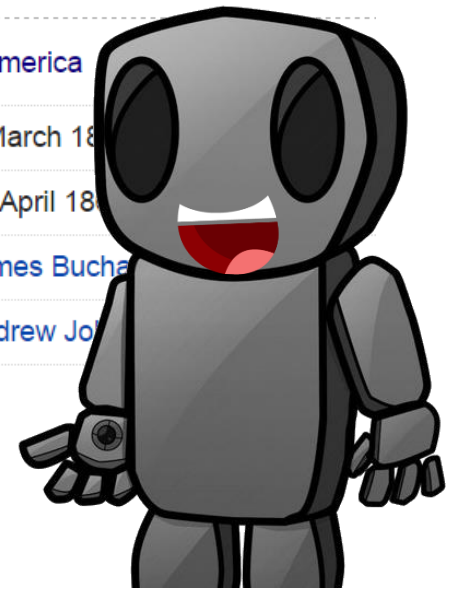
[replaces](#)

[James Bucha](#)

[succeeded by](#)

[Andrew Jo](#)

► [1 reference](#)





The background image is a dark, atmospheric landscape. It features a narrow, light-colored path that winds through a dense forest of bare, dark trees. The scene is heavily shadowed, with a greenish tint overlaying the entire image. The text "MANY OPEN RESEARCH QUESTIONS!" is centered in the lower half of the image in a white, sans-serif font.

MANY OPEN RESEARCH QUESTIONS!

# Our research group ...



**PROJECT ...**

# Project Presentations, December 14th

- Slides:
  - Deadline: **Tuesday 13th, 23:59**
- Code:
  - Submit code/ontology developed to u-cursos
  - Deadline: **Wednesday 14th, 00:59** (*1 hr after slides*)
- Presentation:
  - 9 minutes **strict!**
  - Idea, (source data), pipeline, methods used, example results/demo, limitations/difficulties, lessons learned, (possible future work?)
  - **Wednesday 14th, 10:00-12:00, Sala Auditorio Philippe Flajolet**  
(3er piso, lado poniente)
  - Presentation order selected randomly, not announced  
(Please be there at 10:00!!)
- No report needed 😊

**EXAM ...**



# Exam, December 19th



FINAL BOSS

- Four questions, best of three
  1. RDF (Lecture 2)
    - Good to know how to write Turtle
  2. RDFS/OWL (Lecture 3,4)
    - Lectures 5,6 not important 😊
  3. SPARQL (Lecture 7,8)
    - Will need to write queries
      - *Minor* syntax errors not a problem but should know keywords and query structure
  4. Linked Data/Modelling/RDB2RDF (Lecture 9,10,2)
    - Modelling: given some facts/semantics in natural language, model them in RDF, RDFS, and OWL
    - RDB2RDF: may need to provide mappings

One page (back and front) of original (not copied) handwritten notes permitted

Questions?



Thanks for taking the course!

