

HANDS-ON INTRO TO RDF & SPARQL

Krzysztof Kutt, PhD
Semantic Internet, WFAIS UJ

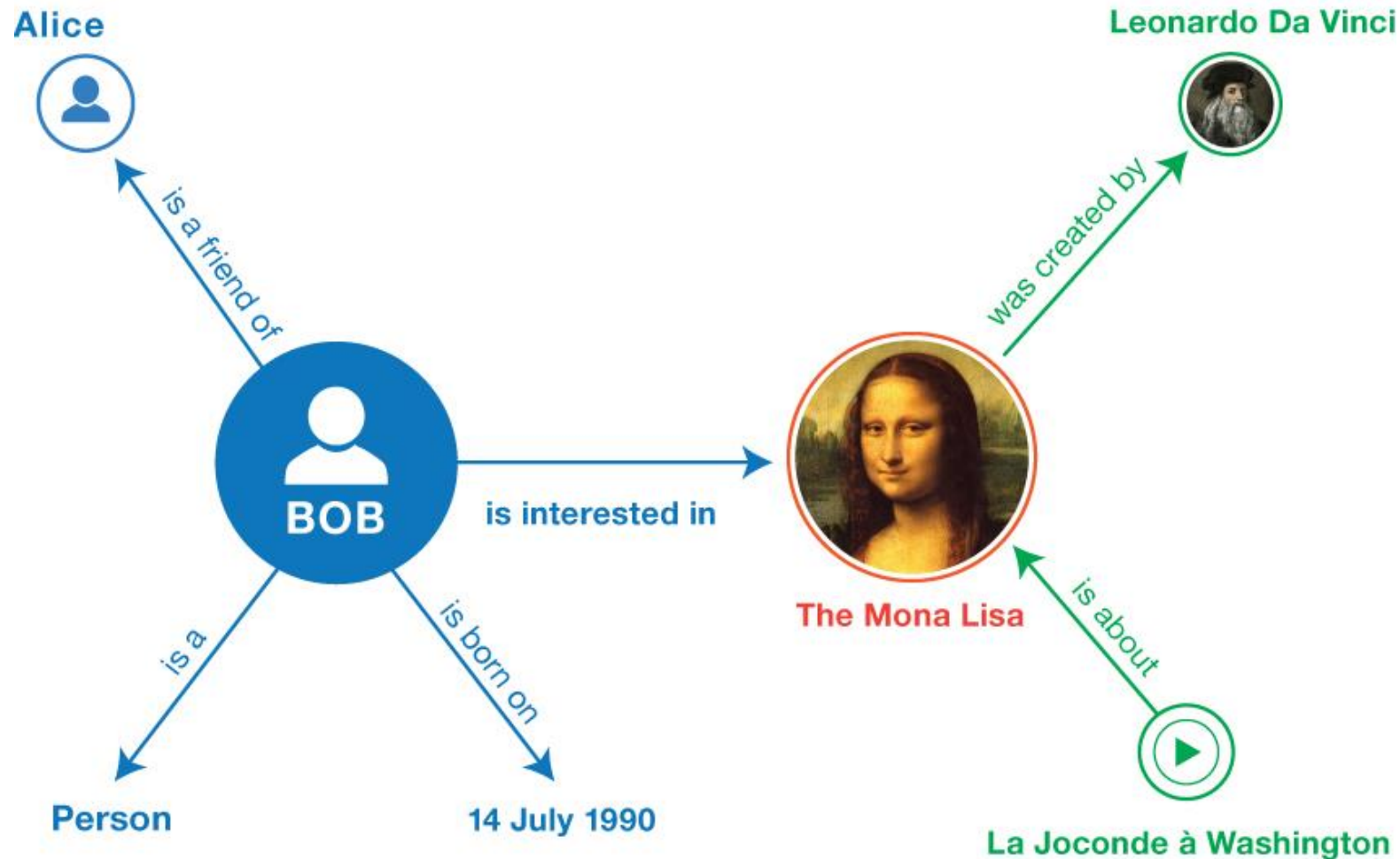
THE GRAPH AND THE TRIPLES

Can the nodes be serialized?

ONCE UPON A TIME, THERE WAS A GRAPH...

Basic elements:

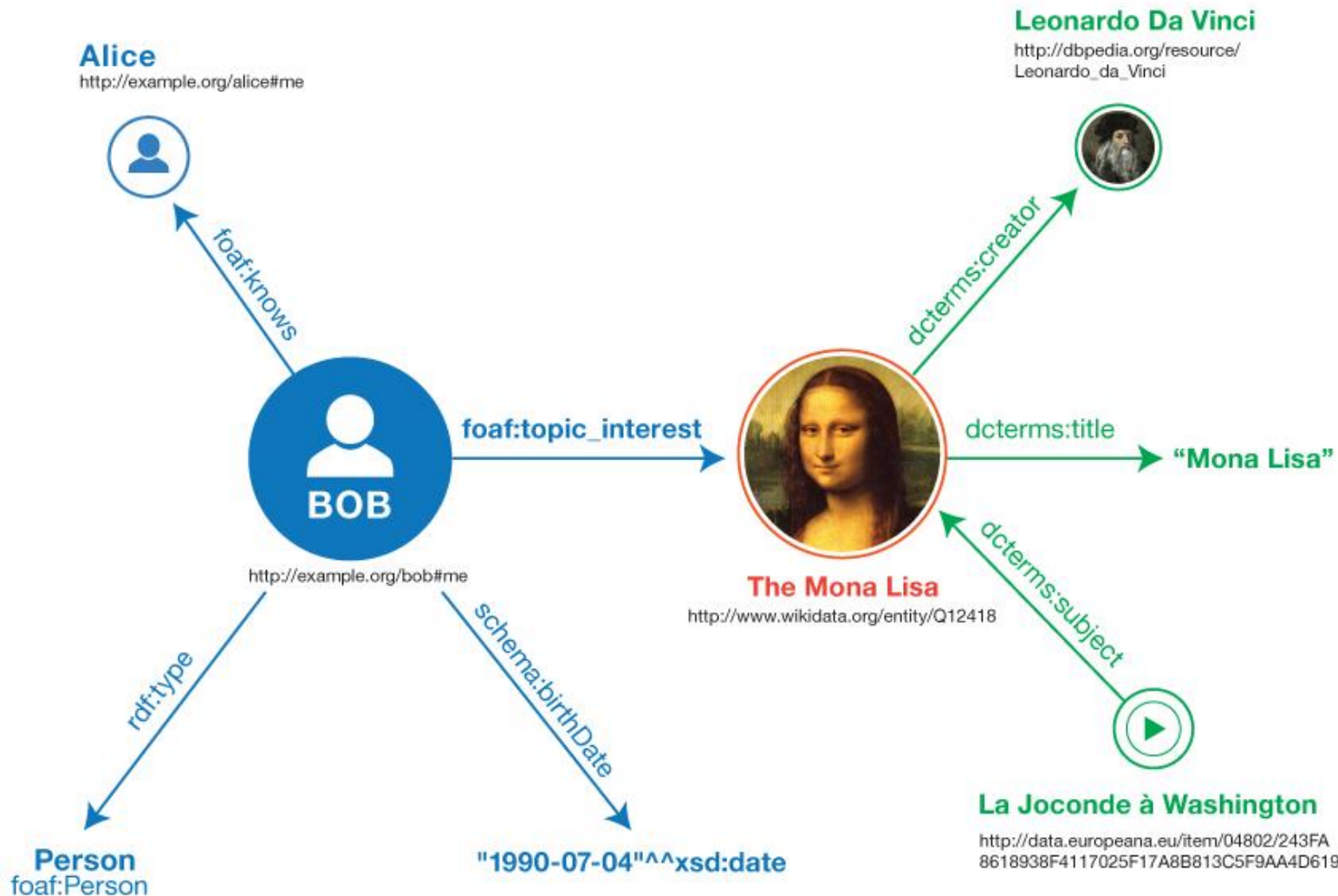
1. Triples
2. Period, semicolon, comma
3. Full URIs
4. Standard prefixes
5. "Base" & own prefixes
6. Datatypes



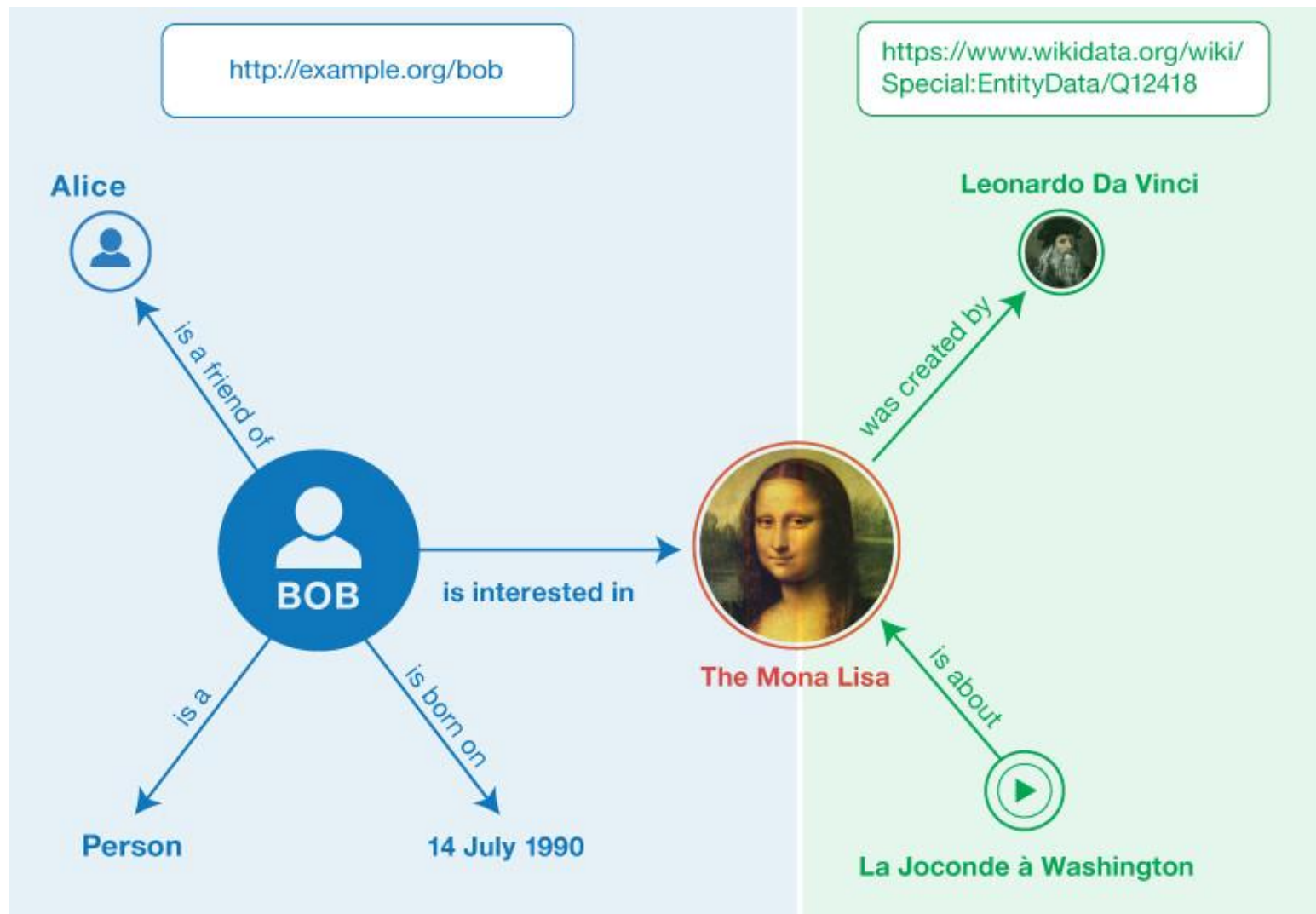
ONCE UPON A TIME, THERE WAS A GRAPH...

Basic elements:

1. Triples
2. Period, semicolon, comma
3. Full URIs
4. Standard prefixes
5. "Base" & own prefixes
6. Datatypes



...STORED ON MANY SERVERS



VALIDATION?

- RDFShape Data analysis: <https://rdfshape.weso.es/dataInfo>
- Anything to Triples (any23) <https://any23.apache.org/>
- W3C RDF Validator <https://www.w3.org/RDF/Validator/>

MORE COMPLEX TRIPLES

But wait, there's more!

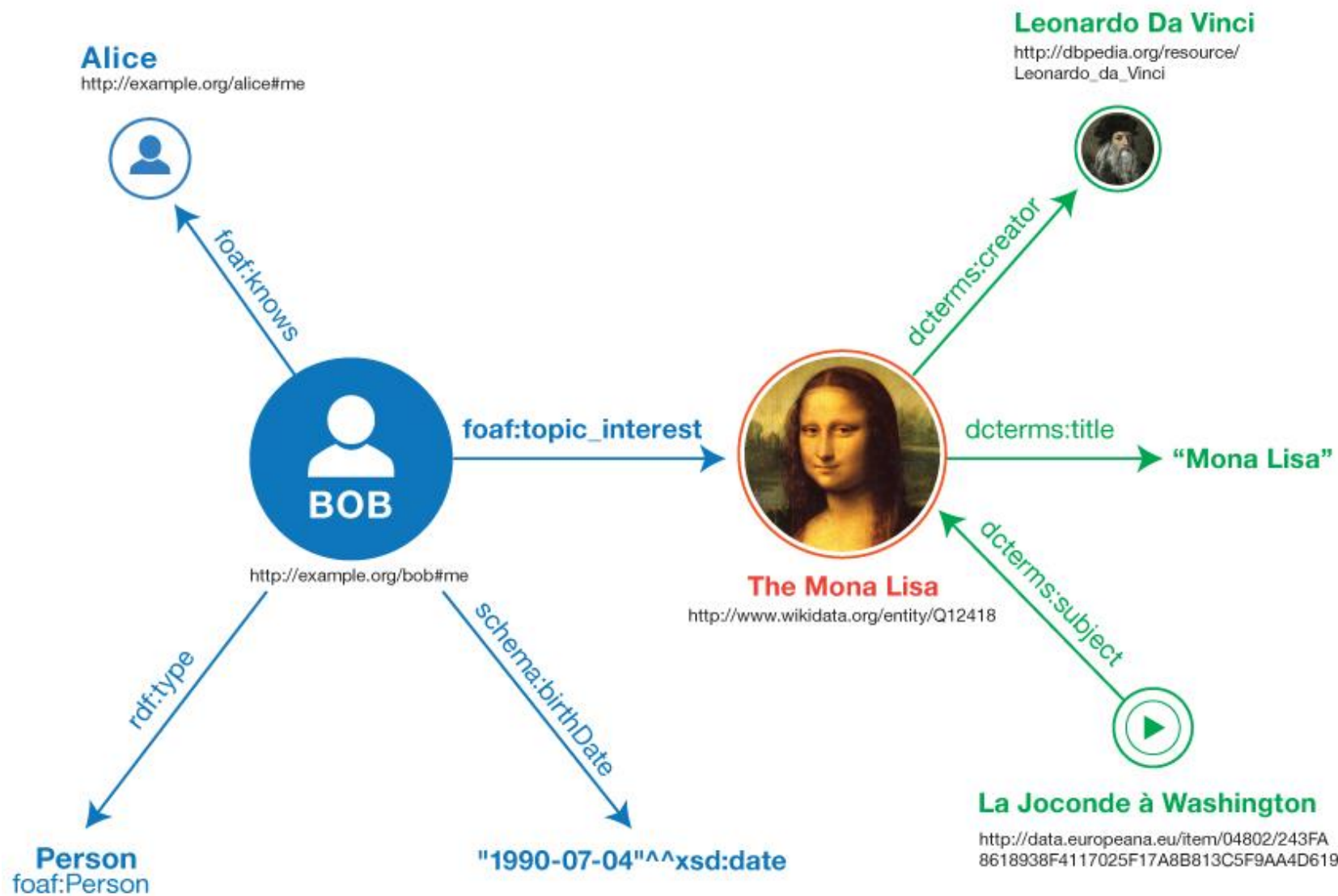
1. Blank nodes:
 - a) Bob knows some person. We know s/he teaches physics, but we don't know anything else.
 - b) Bob killed Alice yesterday, because she doesn't like The Mona Lisa.
2. Containers:
 - a) Bob has two children.
 - b) Bob shook hands with his children.
3. Collection (List):

Bob shook hands with his children (exactly three actions!)
4. Classes:

Bob is a Person. But what the Mona Lisa is?
5. Properties:

Bob likes the Mona Lisa and the movie?
6. Non-modeling properties:

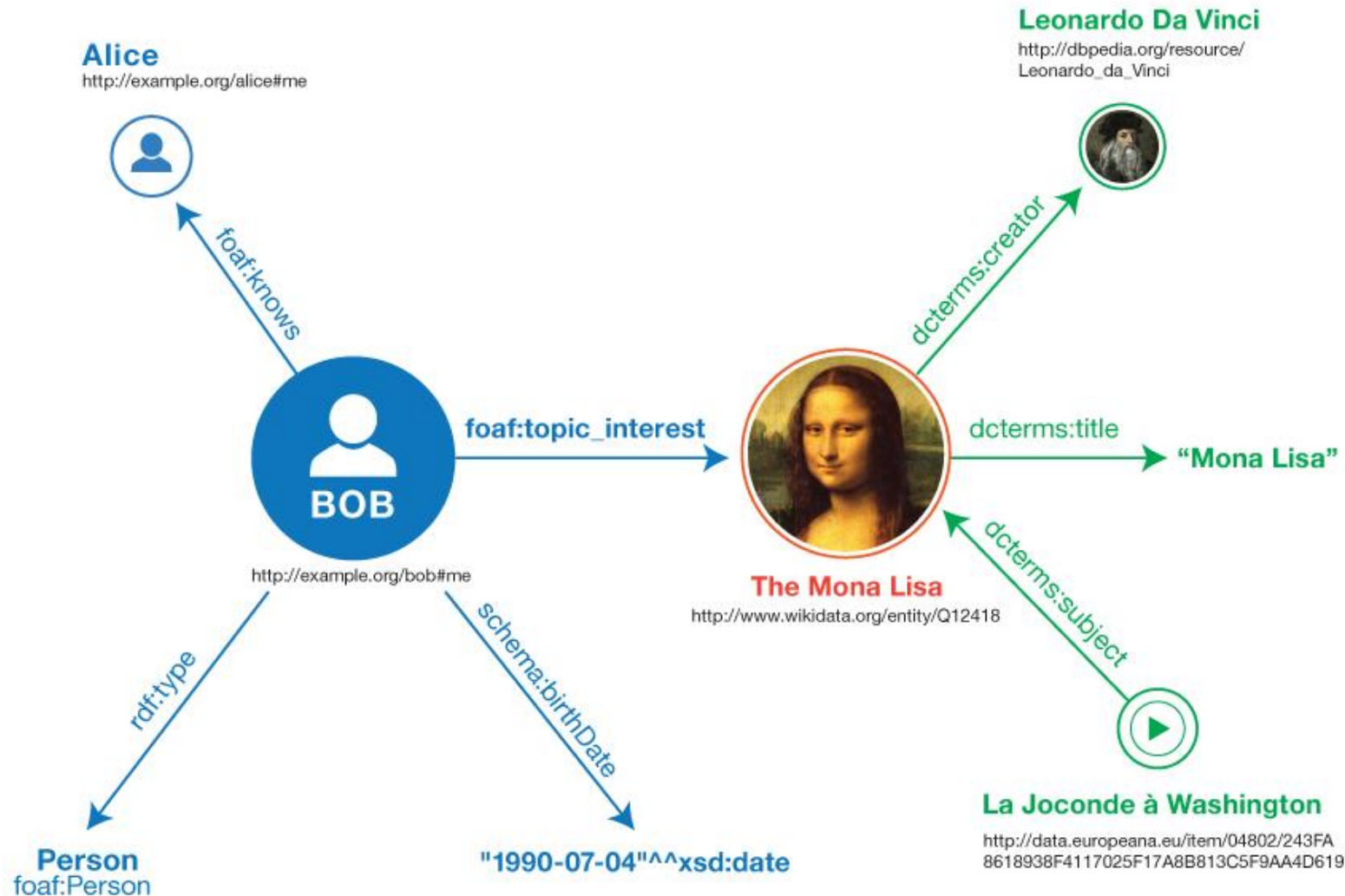
rdfs:label, rdfs:seeAlso, rdfs:isDefinedBy, rdfs:comment



PATTERN MATCHING

Patterns, patterns everywhere...

A QUERY IS A PATTERN



SPARQL ~ TURTLE

SPARQL

Turtle

```
@base <http://example.org/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix schema: <http://schema.org/> .
@prefix wd: <http://www.wikidata.org/entity/> .
@prefix kkutt: <https://krzysztof.kutt.pl/foaf.rdf#> .
```

```
<bob#me>
  a foaf:Person ;
  foaf:knows <alice#me> ,
             kkutt:me ;
  schema:birthDate "1990-07-04"^^xsd:date ;
  foaf:topic_interest wd:Q12418 .
```

HOW TO EXECUTE?

- SPARQLer Query Validator: <http://sparql.org/query-validator.html>
- For querying RDF documents:
 - SPARQLer <http://sparql.org/sparql.html> (for RDF/XML files)
 - RDFShape Data query <https://rdfshape.weso.es/dataQuery> (only SELECT; but for all notations)
- Sample RDF/XML file: <https://krzysztof.kutt.pl/foaf.rdf>

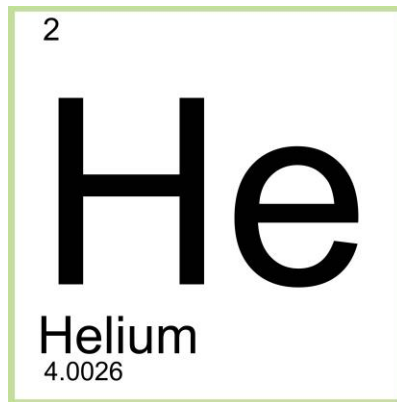
BASIC ELEMENTS AND FORMS

Keep calm and develop queries

PERIODIC TABLE OF ELEMENTS

<http://www.daml.org/.../PeriodicTable.owl>

Mirror: <https://krzysztof.kutt.pl/.../PeriodicTable.owl>



1. What is inside?
2. What are the predicates?
3. List all elements with name and symbol defined
4. List all elements with name and symbol defined, placed in period_7
5. List all elements with name and symbol defined, placed in period_7, with optional color
6. List all elements with name and symbol defined, placed in period_7, with optional color, sorted by name (desc)
7. List all elements with name and symbol defined, placed in period_7, with optional color, sorted by name (desc), where name starts with "U"
8. List three top elements with name and symbol defined, placed in period_7, with optional color, sorted by name (desc), where name starts with "U"

Anatomy of a Query

Declare prefix shortcuts
(*optional*)

→ {
 PREFIX foo: <...>
 PREFIX bar: <...>
 ...

Define the dataset
(*optional*)

{
 SELECT ... ← Query result clause
 FROM <...>
 FROM NAMED <...>
 WHERE {
 ...
 } ← Query pattern

Query modifiers
(*optional*)

{
 GROUP BY ...
 HAVING ...
 ORDER BY ...
 LIMIT ...
 OFFSET ...
 VALUES ...

QUERY FORMS

2. CONSTRUCT

Returns an RDF graph specified by a set of triple templates

4. DESCRIBE

Returns an RDF graph that describes the resources found

1. SELECT

Returns all, or a subset of, the variables bound in a query pattern match

3. ASK

Returns a boolean indicating whether a query pattern matches or not

5. UPDATE

Addition and removal of triples.
Graph management

SPARQL ENDPOINTS

Mis datos son sus datos

SPARQL ENDPOINTS

1. Wikidata ([GUI](#))
<https://query.wikidata.org/sparql>
2. DBpedia ([GUI](#))
<https://dbpedia.org/sparql>
3. YAGO ([GUI](#))
<https://yago-knowledge.org/sparql/query>
4. Linked Brain Data ([GUI](#))
<http://www.linked-neuron-data.org:8890/sparql>

Wrapper (syntax highlighting, hints, auto prefix addition): <https://yasgui.triply.cc/>

```
# Wikidata (example) query (seen during the 1st lab)
# Countries sorted by population
```

```
PREFIX bd: <http://www.bigdata.com/rdf#>
PREFIX wikibase: <http://wikiba.se/ontology#>
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
```

```
SELECT DISTINCT ?countryLabel ?population {
  # ?country <instance of> <country> ;
  #           <population> ?population .
  ?country wdt:P31 wd:Q6256 ;
            wdt:P1082 ?population .
  # for generation of ?countryLabel; equivalent of:
  # ?country rdfs:label ?countryLabel
  # FILTER(LANG(?countryLabel) = "en")
  SERVICE wikibase:label { bd:serviceParam
wikibase:language "en" }
}
ORDER BY DESC(?population)
```

THE 15 MOST POPULOUS COUNTRIES IN EUROPE

How to finish the query from the 1st lab?

1. In Europe?
2. 15 top countries?

```
# Wikidata (example) query (seen during the 1st lab)
# Countries sorted by population
```

```
PREFIX bd: <http://www.bigdata.com/rdf#>
PREFIX wikibase: <http://wikiba.se/ontology#>
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
```

```
SELECT DISTINCT ?countryLabel ?population {
  # ?country <instance of> <country> ;
  #           <population> ?population .
  ?country wdt:P31 wd:Q6256 ;
            wdt:P1082 ?population .
  # for generation of ?countryLabel; equivalent of:
  # ?country rdfs:label ?countryLabel
  # FILTER(LANG(?countryLabel) = "en")
  SERVICE wikibase:label { bd:serviceParam
wikibase:language "en" }
}
ORDER BY DESC(?population)
```

THANK YOU FOR
YOUR ATTENTION!

GEIST Research Group: <https://geist.re/>

Krzysztof Kutt: <https://krzysztof.kutt.pl/>



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).



KEEP
CALM

AND

ASK
QUESTIONS!