

Why, What, How Practical introduction to SPARQL for biologists and informaticians

Using the real world UniProt and neXtProt
databases as illustrative examples



Swiss Institute of
Bioinformatics

Jerven Bolleman (Swiss-Prot)
Daniel Teixeira (CALIPHO)
Pierre-André Michel (CALIPHO)

Global picture

Semantic Web - Linked data

Based on W3C standards & recommendations

SPARQL

SPARQL Protocol and RDF Query Language

RDF

Resource Description Framework

HTTP

HyperText Transfer Protocol

Usual Web / Semantic web

Website	Dataset
Page / URL	Resource / URI
document, textual	formal description
HTML: presentation	RDF: semantic
Human readable	Machine readable

RDF data

- Atomic description unit = triple = statement
- Triple = subject + property + object

<<http://example.org/tuto/resource#William>>

<<http://example.org/tuto/ontology#pet>>

<<http://example.org/tuto/resource#RexDog>>

RDF / SPARQL namespace & prefix

```
@prefix tto: <http://example.org/tuto/ontology#> .  
@prefix ttr: <http://example.org/tuto/resource#> .
```

<<http://example.org/tuto/resource#William>>

<<http://example.org/tuto/ontology#pet>>

<<http://example.org/tuto/resource#RexDog>>

RDF / SPARQL namespace & prefix

```
@prefix tto: <http://example.org/tuto/ontology#> .  
@prefix ttr: <http://example.org/tuto/resource#> .
```

Resources (data)

```
ttr:William tto:pet ttr:RexDog .
```

ttr:William

tto:pet

ttr:RexDog



RDF serialization: *turtle* format

```
@prefix tto: <http://example.org/tuto/ontology#> .  
@prefix ttr: <http://example.org/tuto/resource#> .
```

```
# Resources (data)  
ttr:William tto:pet ttr:RexDog .
```

my-rdf-data.ttl

ttr:William

tto:pet

ttr:RexDog



First SPARQL queries

[Hide SPARQL prefixes](#)

SPARQL Endpoint: <http://localhost:8080/sparql>

```
PREFIX tto:<http://example.org/tuto/ontology#>
PREFIX ttr:<http://example.org/tuto/resource#>
```

```
SELECT ?somePet WHERE {
  ttr:William tto:pet ?somePet .
}
```

Which is William's pet ?

html ▾

Go

Reset

First SPARQL queries

[Hide SPARQL prefixes](#)

SPARQL Endpoint: <http://localhost:8080/sparql>

```
PREFIX tto:<http://example.org/tuto/ontology#>
PREFIX ttr:<http://example.org/tuto/resource#>
```

```
SELECT ?somePet WHERE {
  ttr:William tto:pet ?somePet .
}
```

html ▾

Go

Reset

Query time is 0.123[s] for 1 rows

somePet

ttr:RexDog

First SPARQL queries

[Hide SPARQL prefixes](#)

SPARQL Endpoint: <http://localhost:8080/sparql>

```
PREFIX tto:<http://example.org/tuto/ontology#>
PREFIX ttr:<http://example.org/tuto/resource#>
```

```
SELECT ?someone WHERE {
  ?someone tto:pet ttr:RexDog .
}
```

Whose pet is Rex?

html ▾

Go

Reset

First SPARQL queries

[Hide SPARQL prefixes](#)

SPARQL Endpoint: <http://localhost:8080/sparql>

```
PREFIX tto:<http://example.org/tuto/ontology#>
PREFIX ttr:<http://example.org/tuto/resource#>
```

```
SELECT ?someone WHERE {
  ?someone tto:pet ttr:RexDog .
}
```

html ▾

Go

Reset

Query time is 0.045[s] for 1 rows

someone

ttr:William

First SPARQL queries

[Hide SPARQL prefixes](#)

SPARQL Endpoint: <http://localhost:8080/sparql>

```
PREFIX tto:<http://example.org/tuto/ontology#>
PREFIX ttr:<http://example.org/tuto/resource#>

SELECT ?relation WHERE {
  ttr:William ?relation ttr:RexDog .
}
```

What's the relation
between William and
Rex ?

html ▾

Go

Reset

First SPARQL queries

[Hide SPARQL prefixes](#)

SPARQL Endpoint: <http://localhost:8080/sparql>

```
PREFIX tto:<http://example.org/tuto/ontology#>
PREFIX ttr:<http://example.org/tuto/resource#>
```

```
SELECT ?relation WHERE {
  ttr:William ?relation ttr:RexDog .
}
```

html ▾

Go

Reset

Query time is 0.042[s] for 1 rows

relation

tto:pet

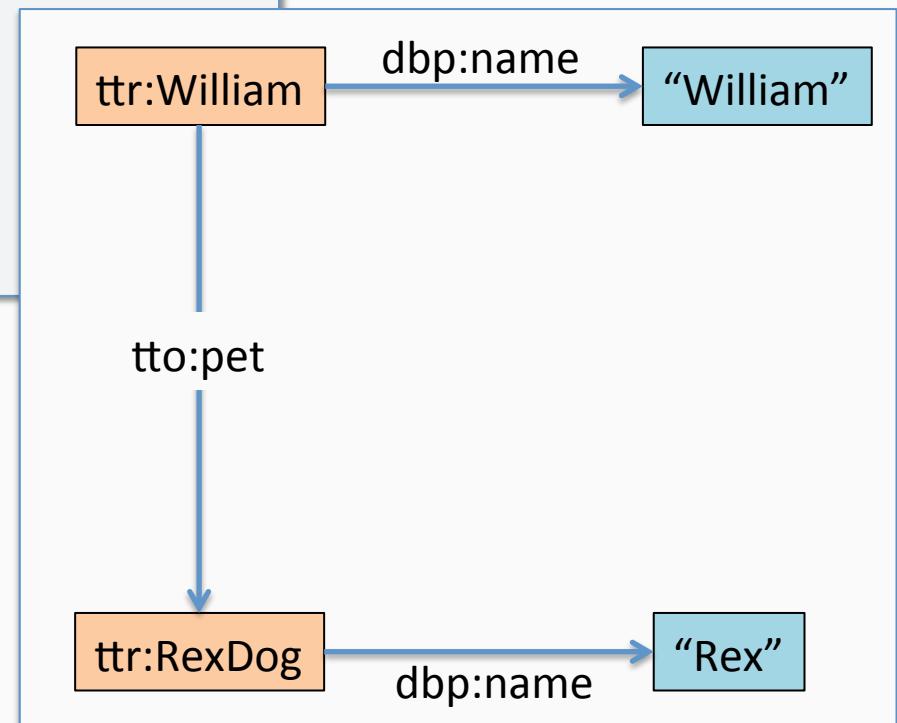
Extending descriptions

- Using existing ontologies
- Using built-in datatypes
- Building an ontology
- Linking data

Giving a name to our *things*

```
@prefix tto: <http://example.org/tuto/ontology#> .  
@prefix ttr: <http://example.org/tuto/resource#> .
```

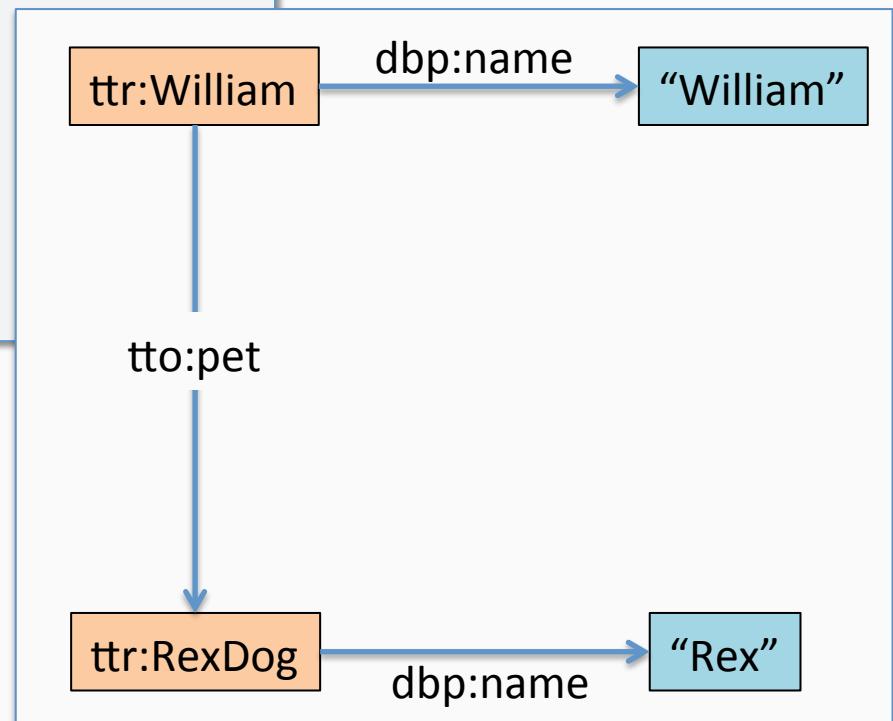
```
ttr:William tto:pet ttr:RexDog .
```



Using a dbpedia property

```
@prefix tto: <http://example.org/tuto/ontology#> .  
@prefix ttr: <http://example.org/tuto/resource#> .  
@prefix dbp: <http://dbpedia.org/property/>
```

```
ttr:William tto:pet ttr:RexDog .
```

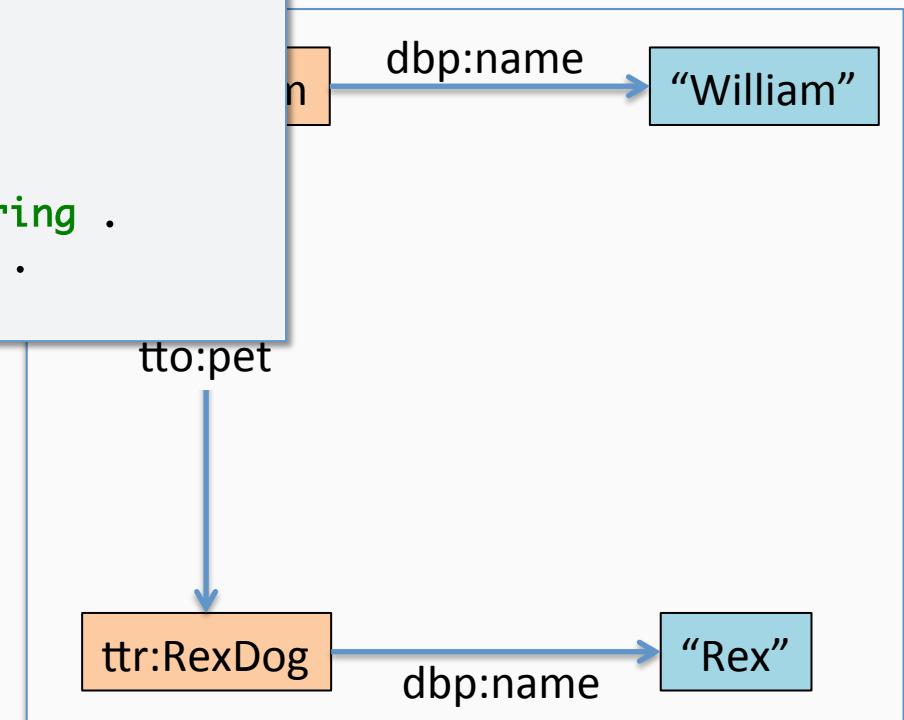


Using literals as property values

```
@prefix tto: <http://example.org/tuto/ontology#> .  
@prefix ttr: <http://example.org/tuto/resource#> .  
@prefix dbp: <http://dbpedia.org/property/> .  
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

```
ttr:William tto:pet ttr:RexDog .
```

```
ttr:William dbp:name "William"^^xsd:string .  
ttr:RexDog dbp:name "Rex"^^xsd:string .
```

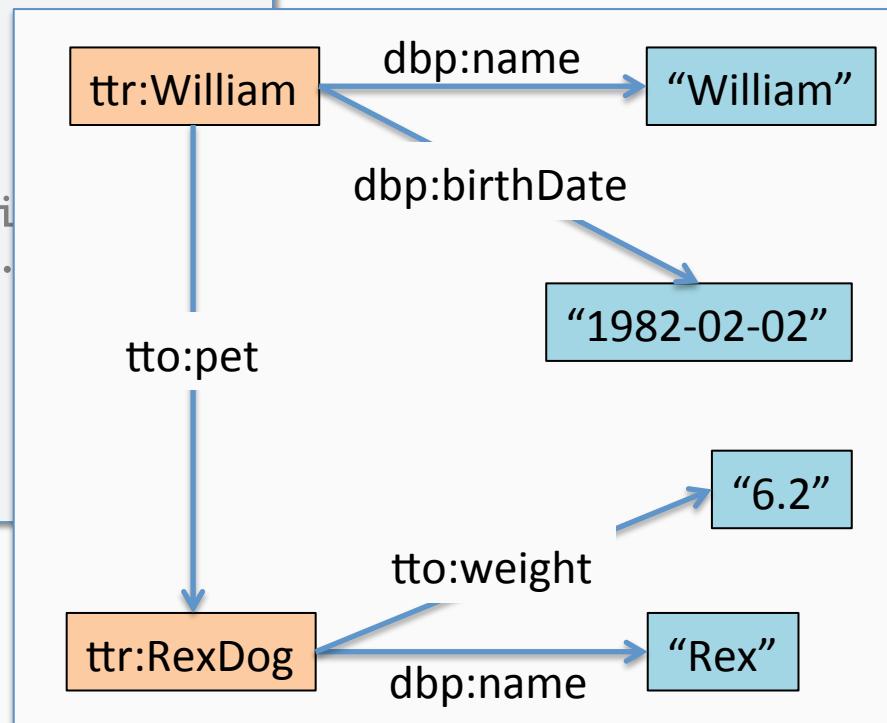


Using literals as property values

```
@prefix tto: <http://example.org/tuto/ontology#> .  
@prefix ttr: <http://example.org/tuto/resource#> .  
@prefix dbp: <http://dbpedia.org/property/> .  
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

```
ttr:William tto:pet ttr:RexDog .
```

```
ttr:William dbp:name "William"^^xsd:string  
ttr:RexDog dbp:name "Rex"^^xsd:string .
```

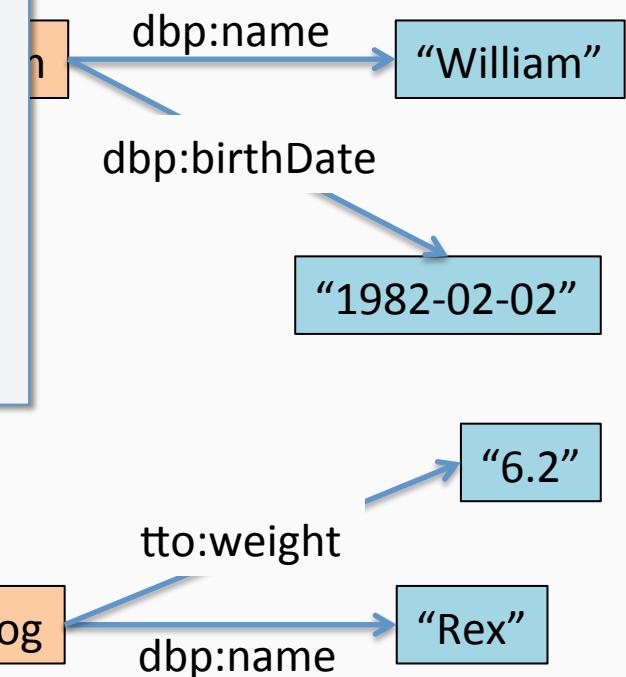


Using literals as property values

```
@prefix tto: <http://example.org/tuto/ontology#> .  
@prefix ttr: <http://example.org/tuto/resource#> .  
@prefix dbp: <http://dbpedia.org/property/> .  
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

```
ttr:William tto:pet ttr:RexDog .
```

```
ttr:William dbp:name "William"^^xsd:string .  
ttr:RexDog dbp:name "Rex"^^xsd:string .  
ttr:William dbp:birthDate "1982-02-02"^^xsd:date .  
ttr:RexDog tto:weight "6.2"^^xsd:decimal .
```



Building an ontology

What is an ontology ?

- A conceptualization of some domain
- A vocabulary = classes + properties

How ?

- On top of existing ontologies (RDF Schema, OWL,..., dbpedia)
- Describe new classes and properties using the vocabulary of other existing ontologies

RDF Schema vocabulary

Classes	Properties
rdfs:Resource	rdf:type, rdfs:label, rdfs:comment, rdfs:seeAlso, rdfs:isDefinedBy, rdf:value, rdfs:member
rdf:Property	rdfs:subPropertyOf, rdfs:domain, rdfs:range
rdfs:Class	rdfs:subClassOf
rdf:Statement	rdf:subject, rdf:predicate, rdf:object
rdf>List	rdf:first, rdf:rest
rdfs:Datatype	
rdfs:Literal : rdf:HTML, rdf:langString, rdf:XMLLiteral	
rdfs:Container: rdf:Bag, rdf:Seq, rdf:Alt	
rdfs:ContainerMembershipProperty	

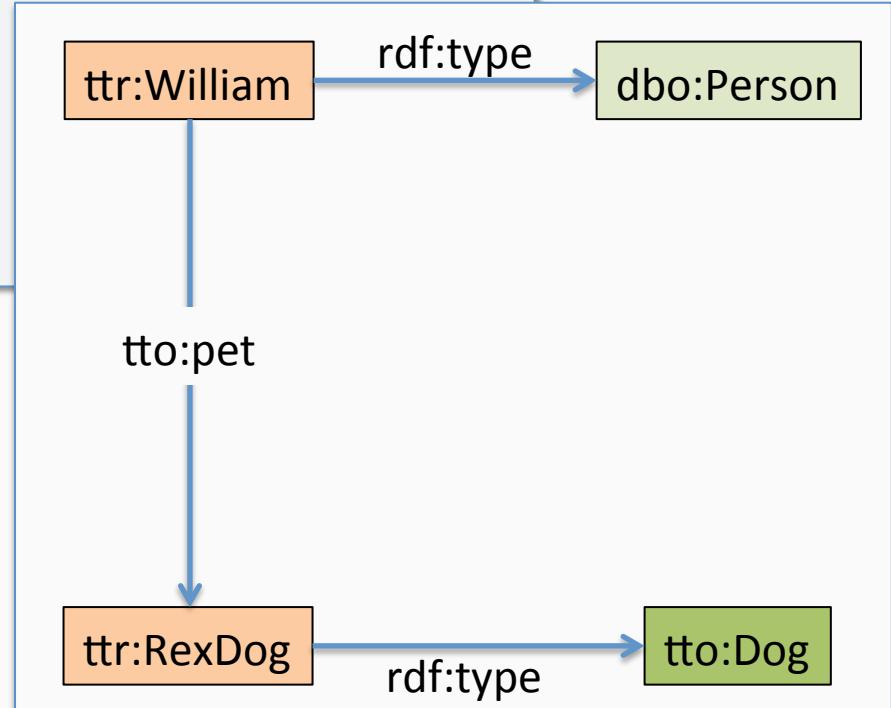
Classifying resources with *rdf:type*

```
@prefix tto: <http://example.org/tuto/ontology#> .  
@prefix ttr: <http://example.org/tuto/resource#> .
```

Resources (data)

```
ttr:William tto:pet ttr:RexDog .
```

Classes (ontology)



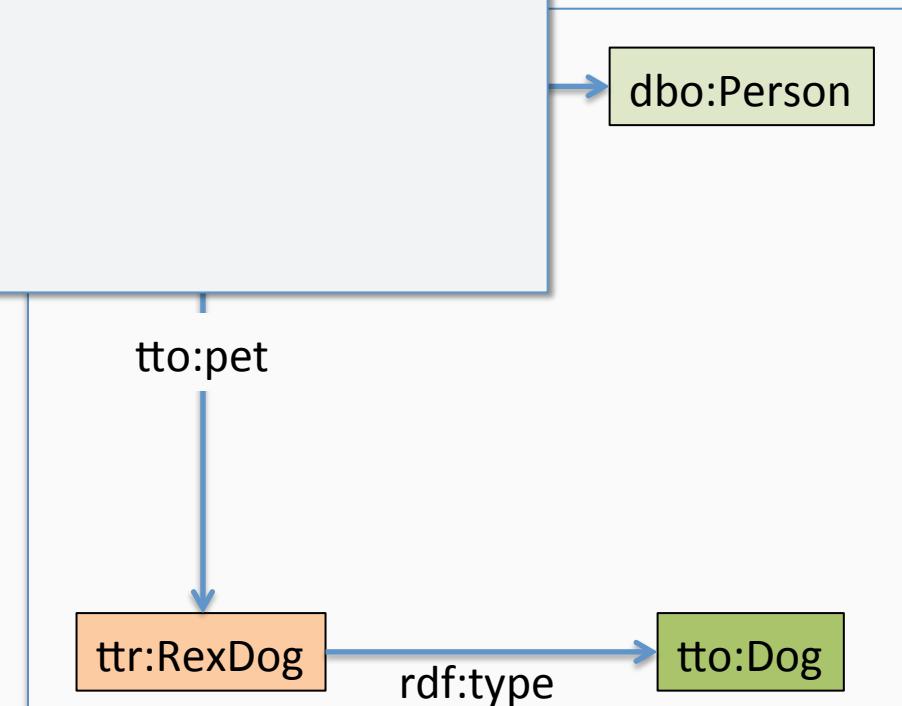
Classifying resources with *rdf:type*

```
@prefix tto: <http://example.org/tuto/ontology#> .  
@prefix ttr: <http://example.org/tuto/resource#> .  
@prefix dbo: <http://dbpedia.org/ontology/> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#>
```

Resources (data)

```
ttr:William tto:pet ttr:RexDog .  
ttr:William rdf:type dbo:Person .  
ttr:RexDog rdf:type tto:Dog .
```

Classes (ontology)



Describing the class *tto:Dog*

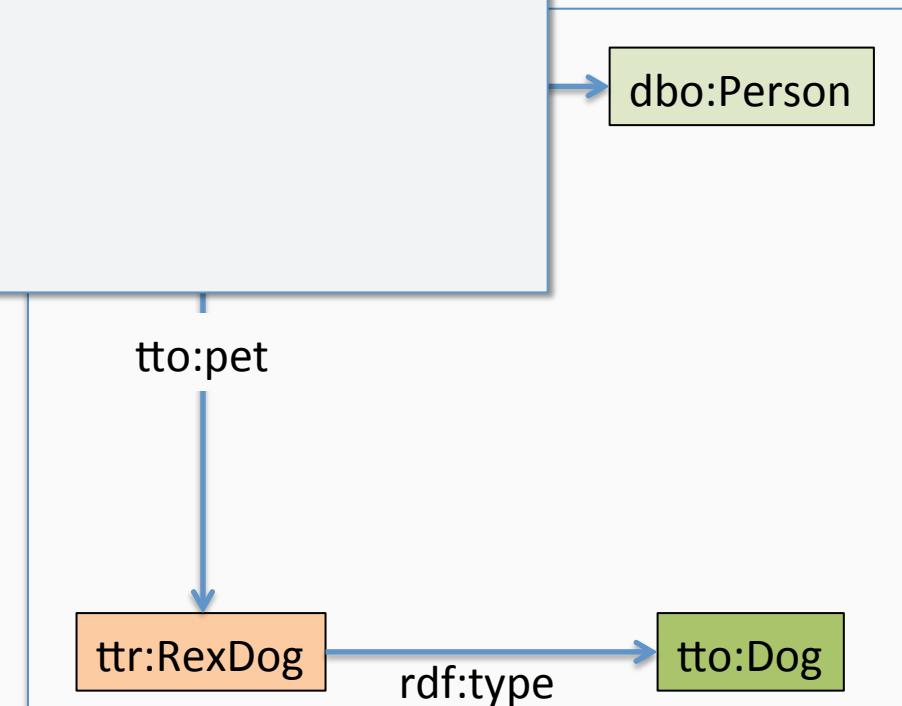
```
@prefix tto: <http://example.org/tuto/ontology#> .  
@prefix ttr: <http://example.org/tuto/resource#> .  
@prefix dbo: <http://dbpedia.org/ontology/> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#>
```

Resources (data)

```
ttr:William tto:pet ttr:RexDog .  
ttr:William rdf:type dbo:Person .  
ttr:RexDog rdf:type tto:Dog .
```

Classes (ontology)

```
tto:Dog rdf:type rdfs:Class .  
tto:Dog rdfs:label "dog"^^xsd:string .
```



Syntax shortcuts

```
@prefix tto: <http://example.org/tuto/ontology#> .  
@prefix ttr: <http://example.org/tuto/resource#> .  
@prefix dbo: <http://dbpedia.org/ontology/> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#>
```

Resources (data)

```
ttr:William tto:pet ttr:RexDog .
```

```
ttr:William a dbo:Person .
```

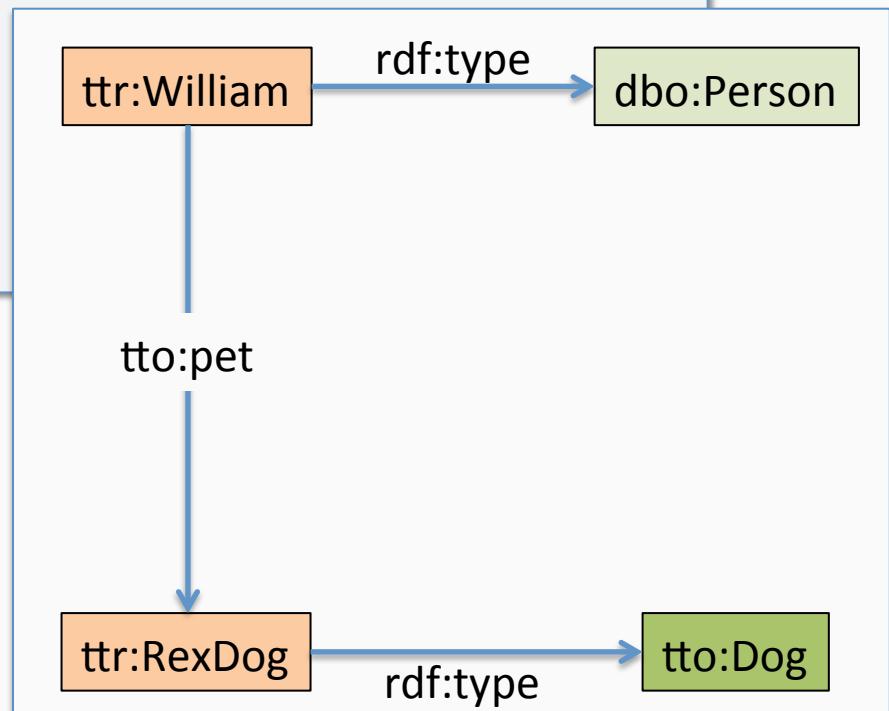
```
ttr:RexDog a tto:Dog .
```

Classes (ontology)

```
tto:Dog a rdfs:Class ;  
rdfs:label "dog"^^xsd:string .
```

a = rdf:type

; make subject of next line
same as subject on current
line



Organizing classes with *rdfs:subClassOf*

```
@prefix tto: <http://example.org/tuto/ontology#> .  
@prefix ttr: <http://example.org/tuto/resource#> .  
@prefix dbo: <http://dbpedia.org/ontology/> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#>
```

Resources (data)

```
ttr:William tto:pet ttr:RexDog .
```

```
ttr:William a dbo:Person .
```

```
ttr:RexDog a tto:Dog .
```

Classes (ontology)

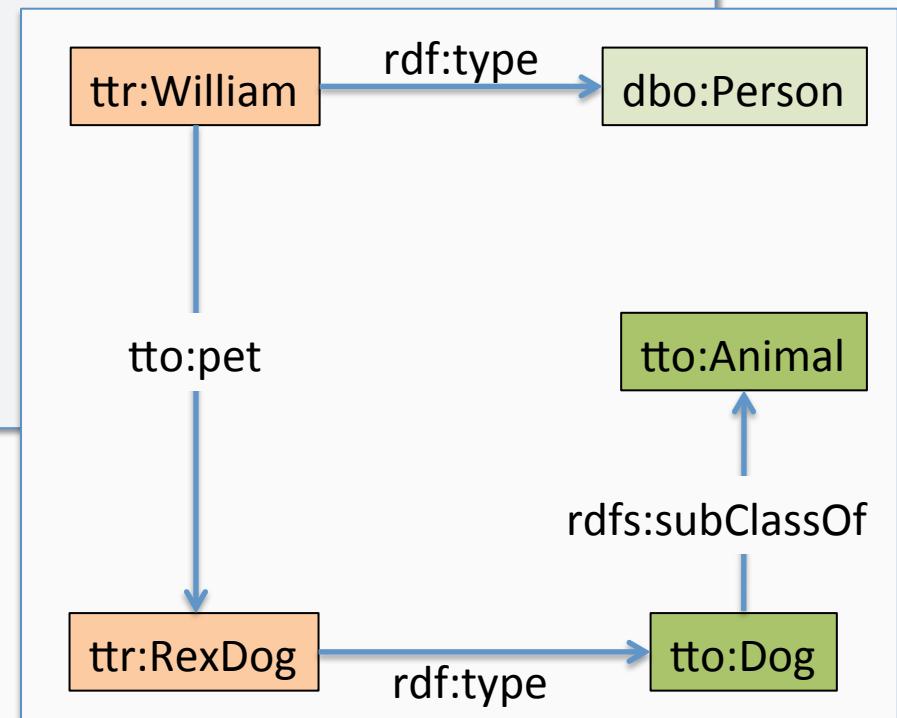
```
tto:Dog a rdfs:Class ;
```

```
    rdfs:label "dog"^^xsd:string ;
```

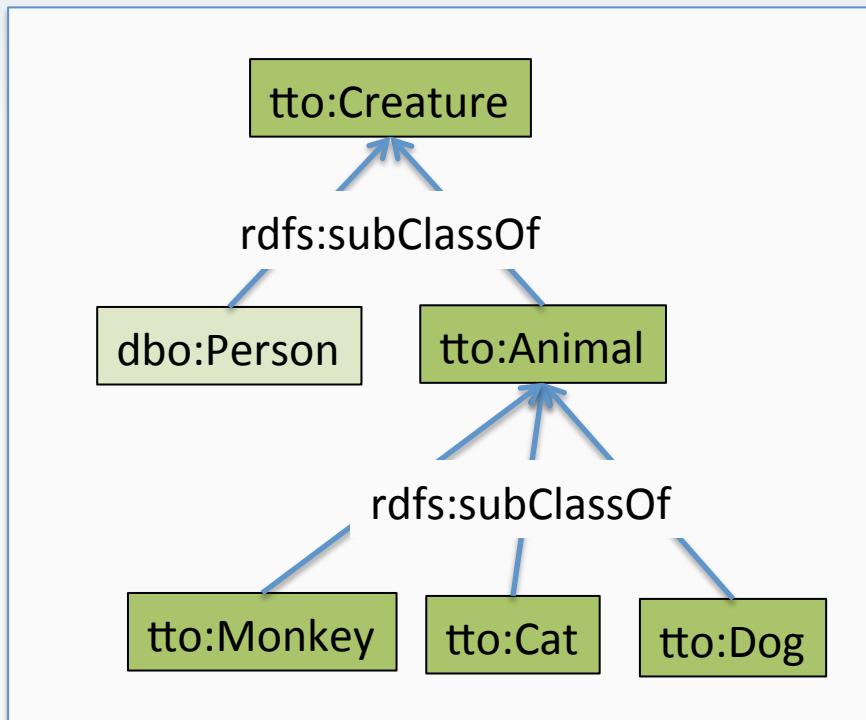
```
    rdfs:subClassOf tto:Animal .
```

```
tto:Animal a rdfs:Class ;
```

```
    rdfs:label "animal"^^xsd:string .
```



Classes of our sample dataset



Describing a property

```
@prefix tto: <http://example.org/tuto/ontology#> .  
@prefix ttr: <http://example.org/tuto/resource#> .  
@prefix dbo: <http://dbpedia.org/ontology/> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#>
```

Resources (data)

```
ttr:William tto:pet ttr:RexDog .
```

```
ttr:William a dbo:Person .
```

```
ttr:RexDog a tto:Dog .
```

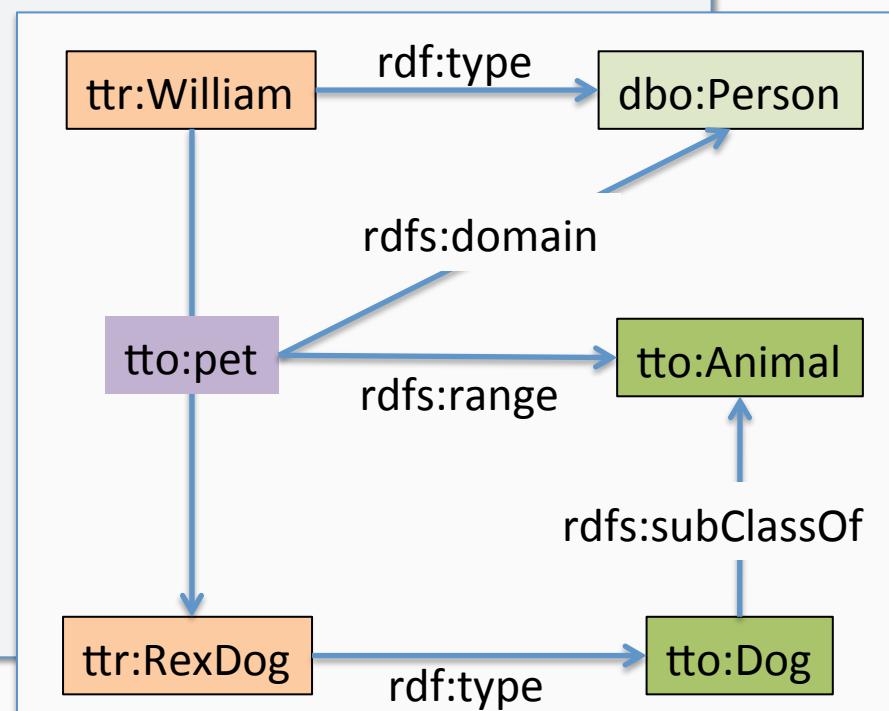
Classes (ontology)

```
tto:Dog a rdfs:Class ;  
    rdfs:label "dog"^^xsd:string ;  
    rdfs:subClassOf tto:Animal .
```

```
tto:Animal a rdfs:Class ;  
    rdfs:label "animal"^^xsd:string .
```

Properties (ontology)

```
tto:pet a rdf:Property ;  
    rdfs:label "pet"^^xsd:string ;  
    rdfs:domain dbo:Person ;  
    rdfs:range tto:Animal .
```



What you know about RDF Schema

Classes	Properties
rdfs:Resource	rdf:type, rdfs:label, rdfs:comment, rdfs:seeAlso, rdfs:isDefinedBy, rdf:value, rdfs:member
rdf:Property	rdfs:domain, rdfs:range, rdfs:subPropertyOf
rdfs:Class	rdfs:subClassOf
rdf:Statement	rdf:subject, rdf:predicate, rdf:object
rdf>List	rdf:first, rdf:rest
rdfs:Datatype	
rdfs:Literal : rdf:HTML, rdf:langString, rdf:XMLLiteral	
rdfs:Container: rdf:Bag, rdf:Seq, rdf:Alt	
rdfs:ContainerMembershipProperty	

Linking data

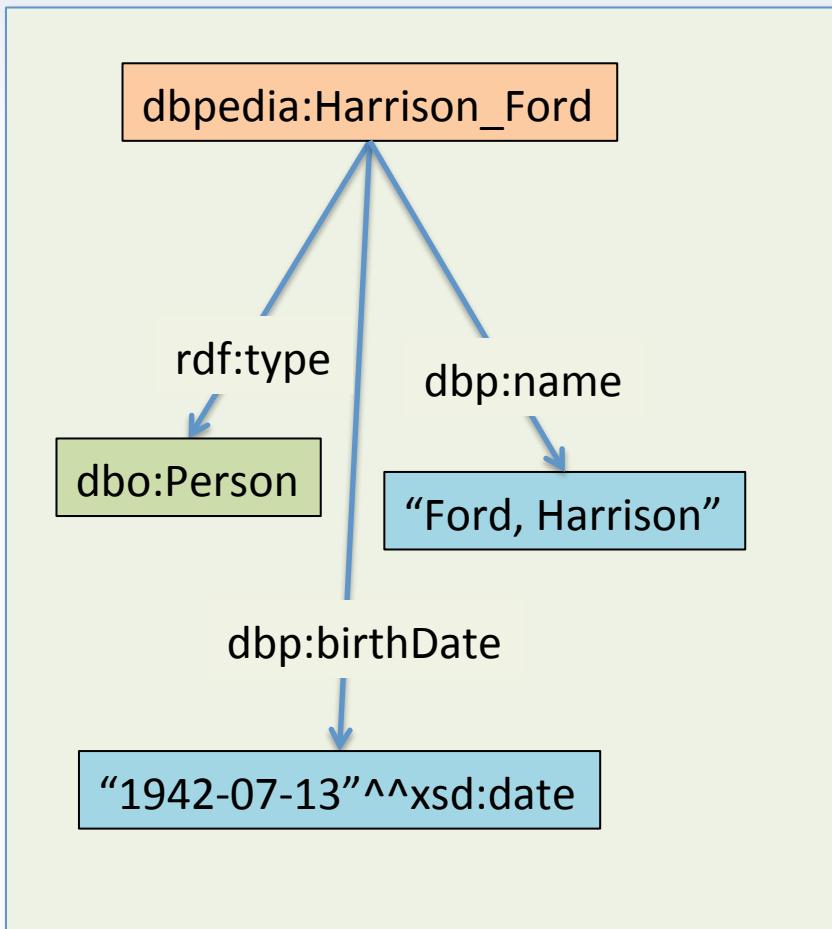


http://dbpedia.org/resource/Harrison_Ford

dbpedia:Harrison_Ford

Linking data

DBPEDIA dataset



Linking data



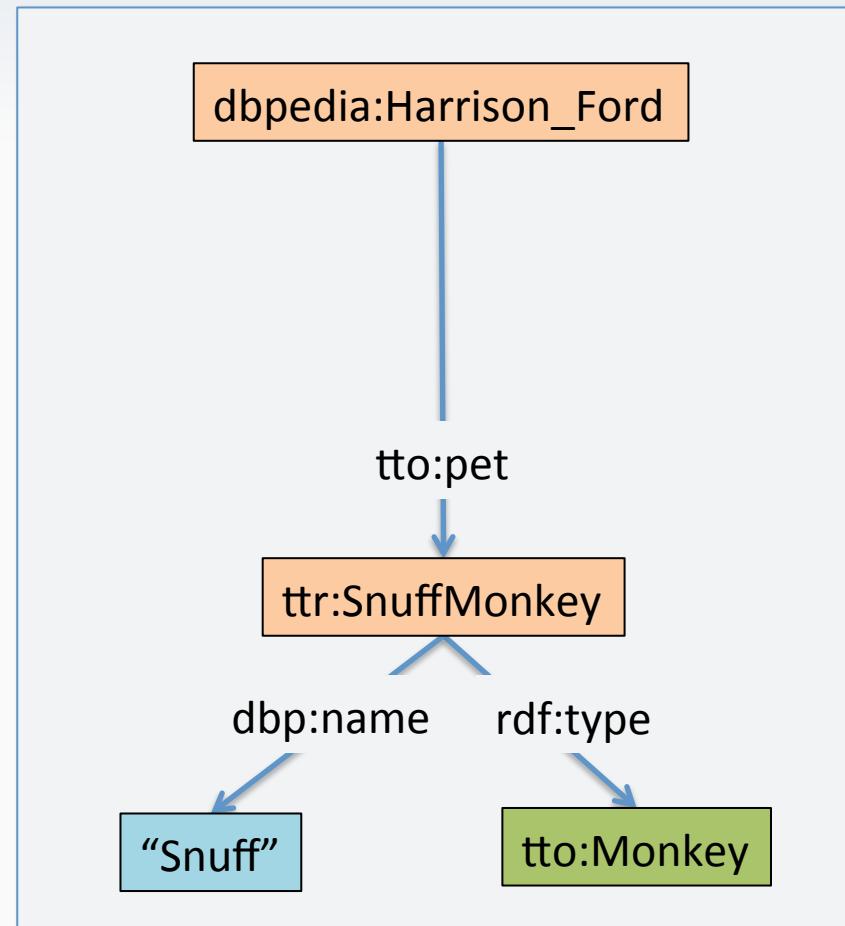
<http://example.org/tuto/resource#SnuffMonkey>

ttr:SnuffMonkey

Linking data



Tutorial dataset



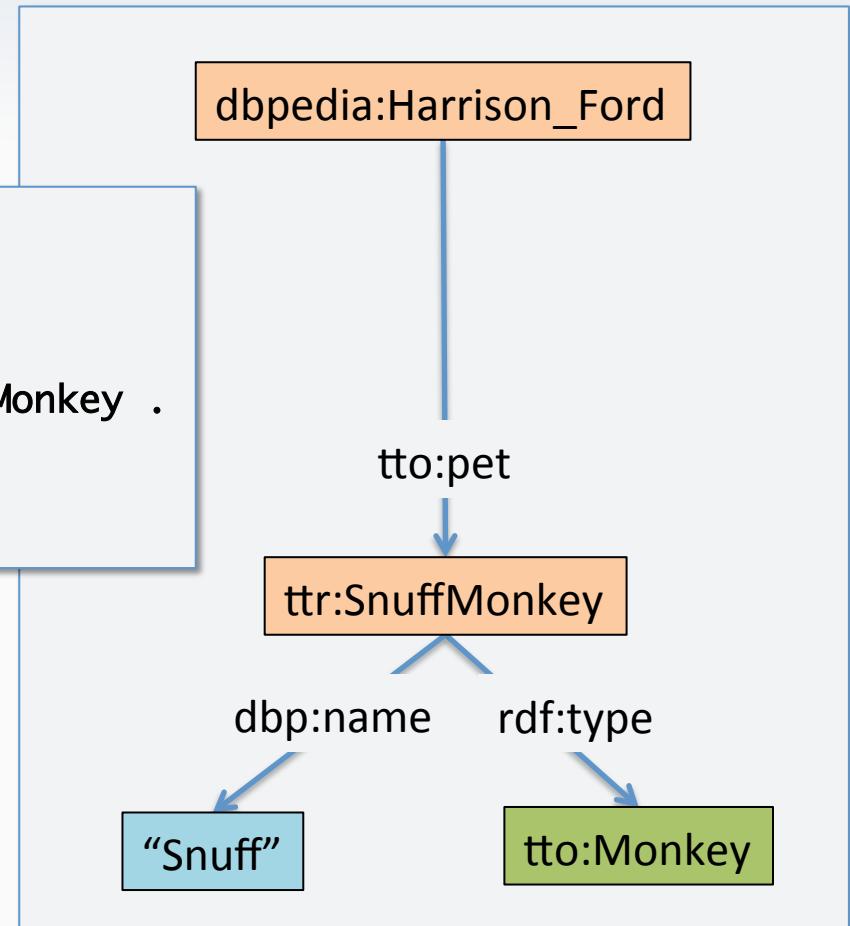
Linking data



```
@prefix ...  
...  
  
# Resources (data)  
dbpedia:Harrison_Ford tto:pet ttr:SnuffMonkey .  
  
ttr:SnuffMonkey a tto:Monkey ;  
    dbp:name "Snuff"^^xsd:string .
```

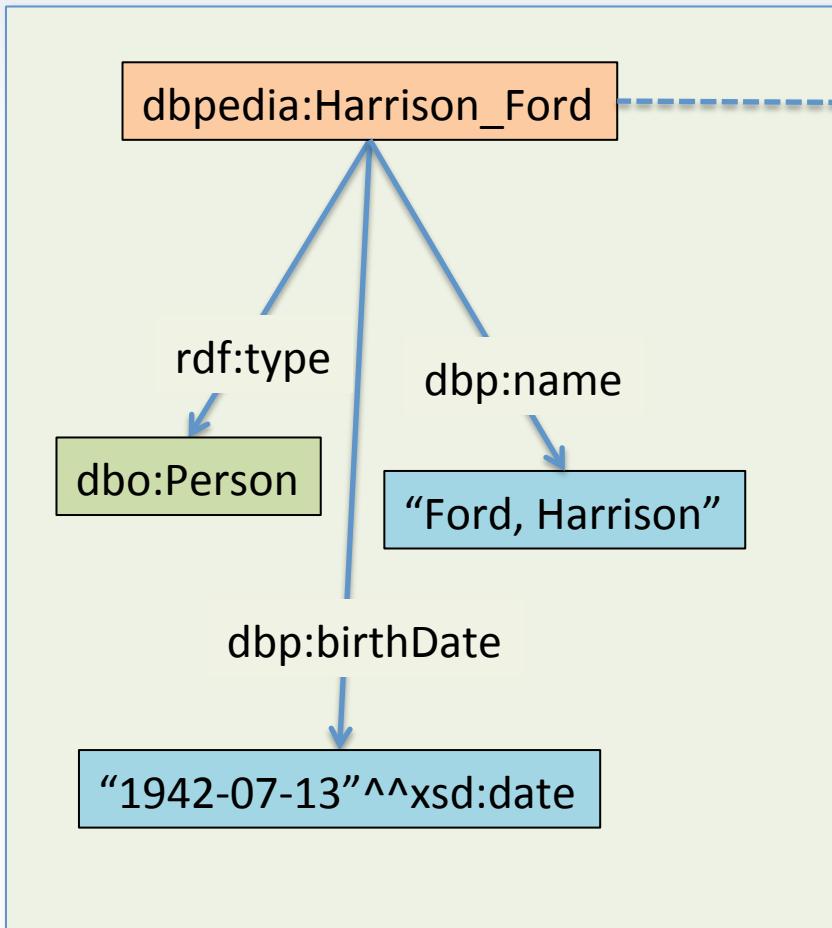


Tutorial dataset

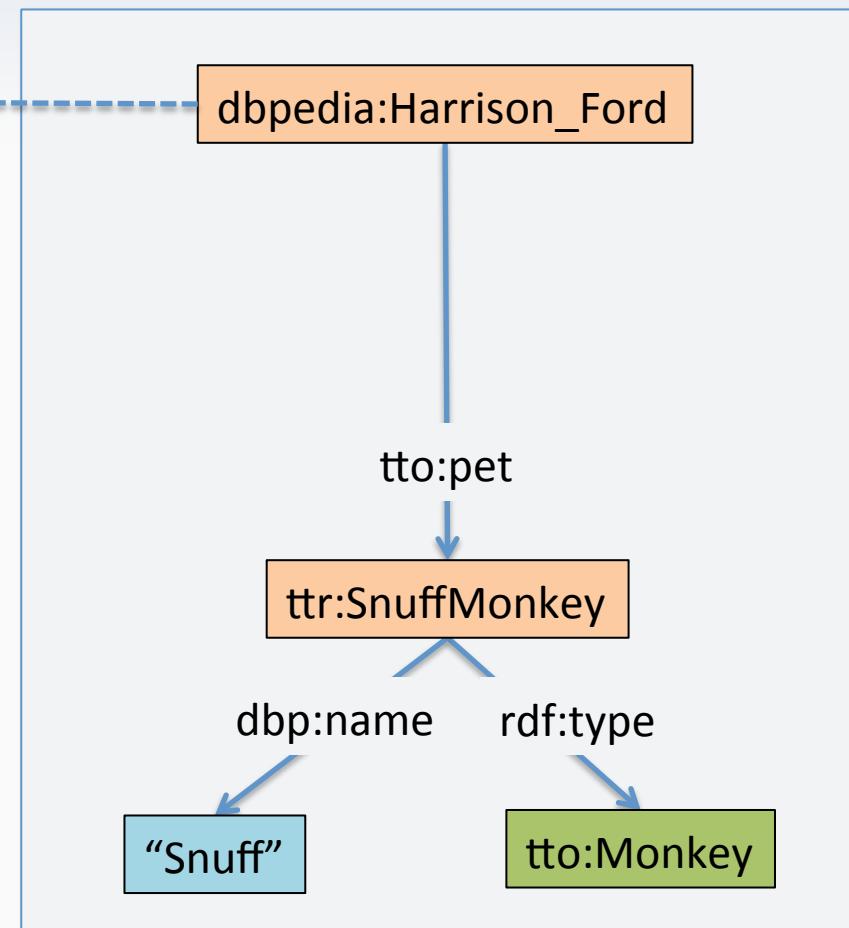


Linking data – reusing URIs

DBPEDIA dataset



Tutorial dataset



Federated query

Hide SPARQL prefixes

SPARQL Endpoint: <http://localhost:8080/sparql>

```
PREFIX dbo:<http://dbpedia.org/ontology/>
PREFIX dbp:<http://dbpedia.org/property/>
PREFIX dbpedia:<http://dbpedia.org/resource/>
PREFIX owl:<http://www.w3.org/2002/07/owl#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX tto:<http://example.org/tuto/ontology#>
PREFIX ttr:<http://example.org/tuto/resource#>
PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
```

```
SELECT * WHERE {
  VALUES ?subj {dbpedia:Harrison_Ford}
  ?subj tto:pet ?pet .
  SERVICE <http://dbpedia.org/sparql> {
    ?subj dbp:name ?name .
    ?subj dbp:birthDate ?bd .
  }
}
```

html ▾

Go

Reset

Federated query

Hide SPARQL prefixes

SPARQL Endpoint: <http://localhost:8080/sparql>

```
PREFIX dbo:<http://dbpedia.org/ontology/>
PREFIX dbp:<http://dbpedia.org/property/>
PREFIX dbpedia:<http://dbpedia.org/resource/>
PREFIX owl:<http://www.w3.org/2002/07/owl#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX tto:<http://example.org/tuto/ontology#>
PREFIX ttr:<http://example.org/tuto/resource#>
PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
```

```
SELECT * WHERE {
  VALUES ?subj {dbpedia:Harrison_Ford}
  ?subj tto:pet ?pet .
  SERVICE <http://dbpedia.org/sparql> {
    ?subj dbp:name ?name .
    ?subj dbp:birthDate ?bd .
  }
}
```

html

Go

Reset

Query time is 0.426[s] for 1 rows

subj	pet	name	bd
dbpedia:Harrison_Ford	ttr:SnuffMonkey	"Ford, Harrison"@en	"1942-07-13+02:00"

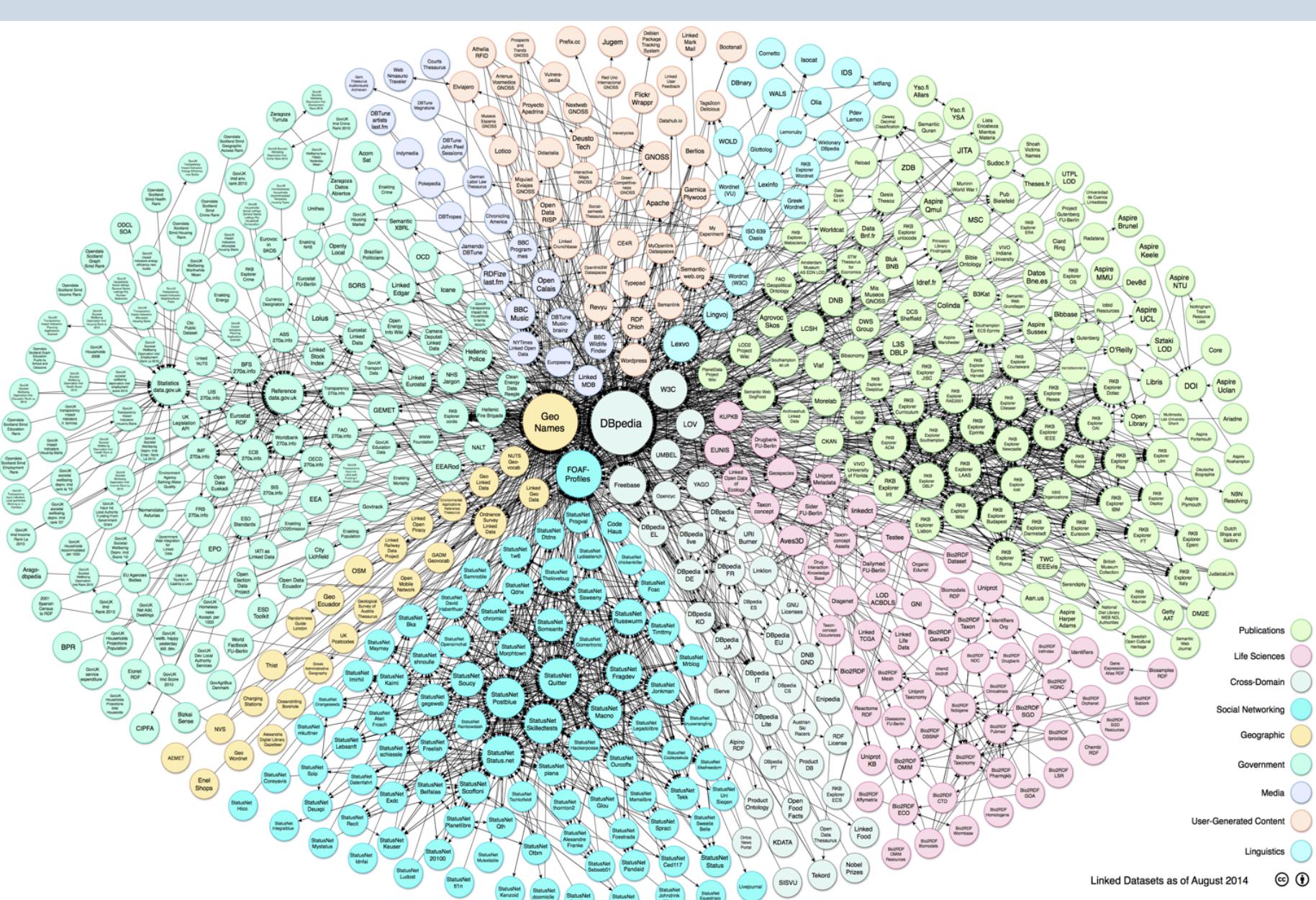
Summary

- How to describe resources
- How to reuse existing ontologies
- How to build an ontology
- How to link data

Who is present in the LOD cloud

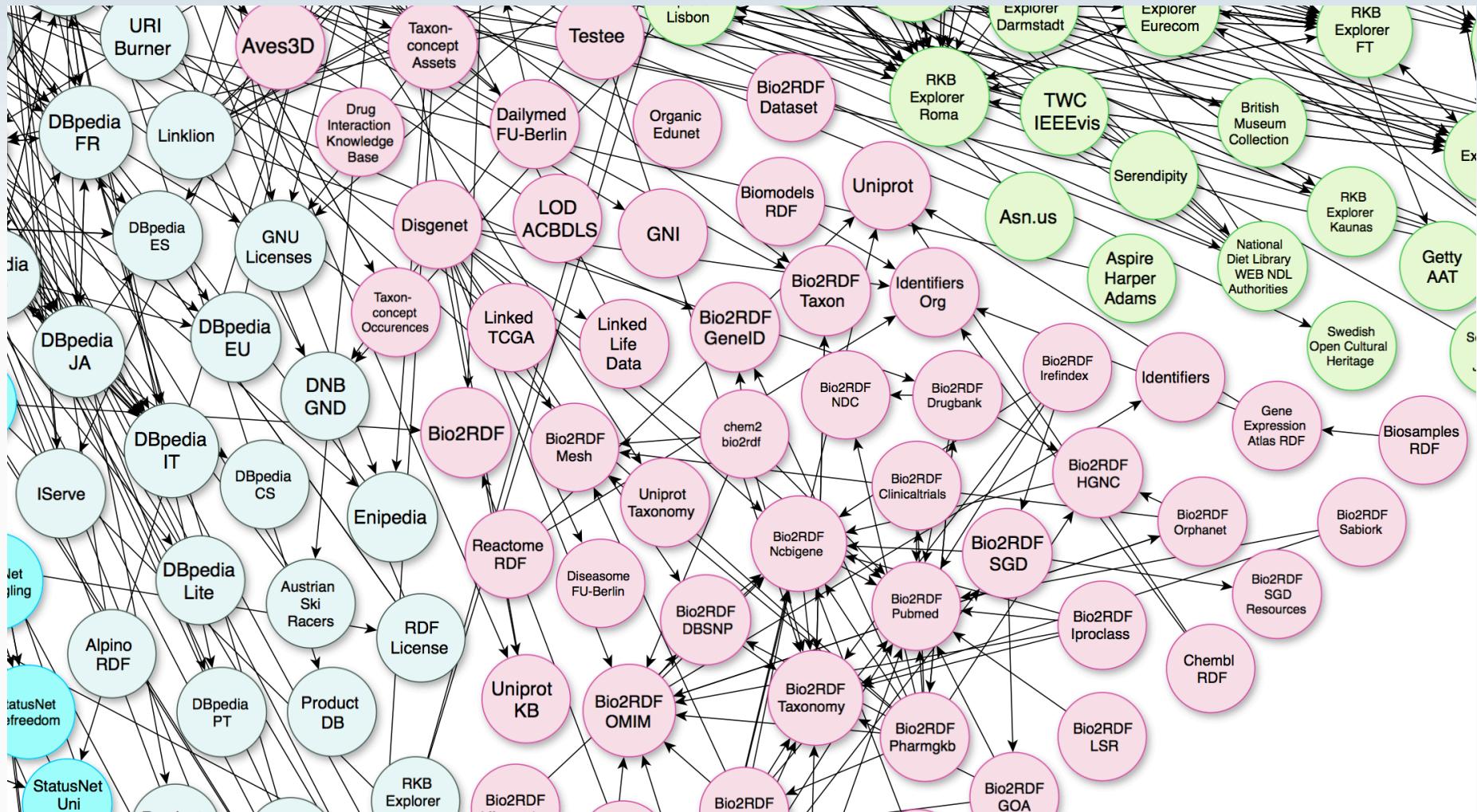
- Social networks
- Governments: UK
- Media: BBC, NY Times, ...
- Publications: DOI, ...
- Life sciences

...



"Linking Open Data cloud diagram 2014, by Max Schmachtenberg, Christian Bizer, Anja Jentzsch and Richard Cyganiak. <http://lod-cloud.net/>"

Life sciences in the LOD cloud



"Linking Open Data cloud diagram 2014, by Max Schmachtenberg, Christian Bizer, Anja Jentzsch and Richard Cyganiak. <http://lod-cloud.net/>"

Life sciences SPARQL endpoints

SIB Swiss Institute of Bioinformatics

- UniProt, neXtProt (CALIPHO) , SubstructureSearch (PIG)

European Bioinformatics Institute (EBI)

- BioSamples, BioModels, ChEMBL, Expression Atlas, Reactome, InterPro, Ensembl
- PDB - via bio2rdf

National Center for Biotechnology Information (NCBI) – via bio2rdf

- dbSNP, HomoloGene, Gene, OMIM, PubMed, Taxonomy

Gene ontology consortium (GOC) – via bio2rdf

- GOA

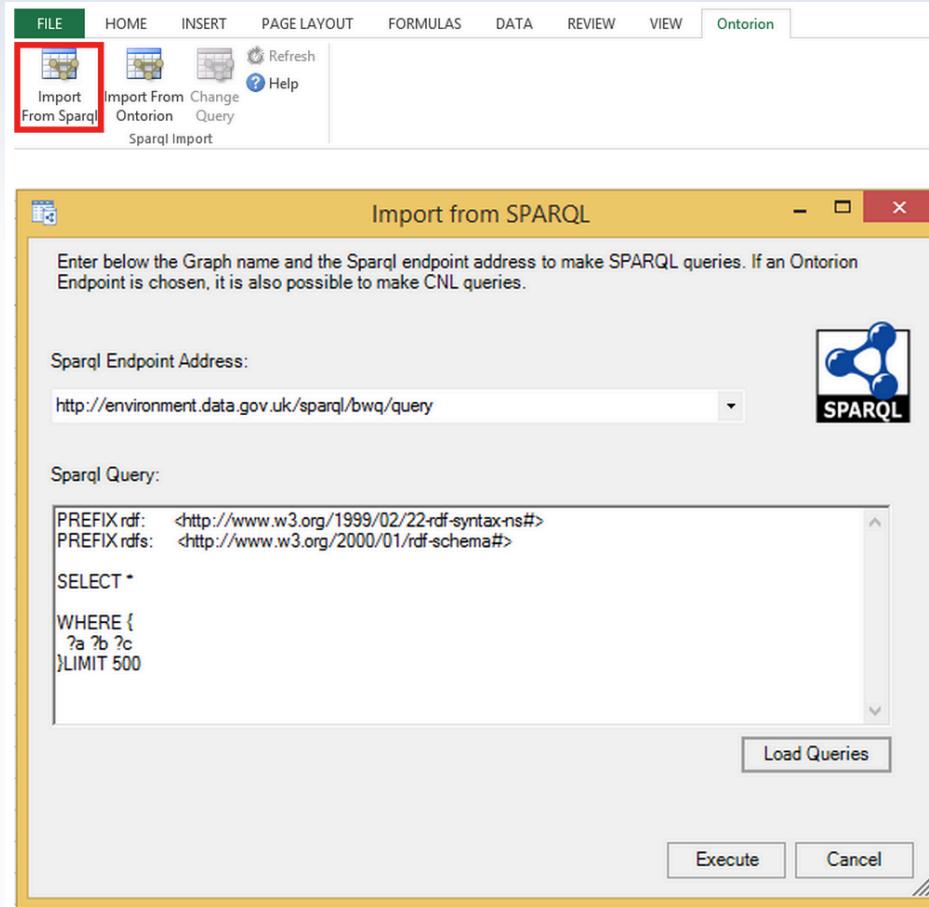
Bio2Rdf – Open source project

- 35 datasets

SubstructureSearch demo: <http://glycoproteome.isb-sib.ch:8081/rdf-service-glassfish/>

How to connect to a SPARQL end point

Example: Excel add-in



Cognitum Ontorion™ add-in: <http://www.cognitum.eu/download/download.aspx?id=1014>

Connecting to a SPARQL end point using the SNORQL explorer

Snorql: Exploring <http://data.semanticweb.org/sparql>

GRAPH: Default graph. [List named graphs](#)

SPARQL:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX swrc: <http://swrc.ontoware.org/ontology#>
PREFIX swrc-ext: <http://www.cs.vu.nl/~mcaklein/onto/swrc_ext/2005/05#>
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX ical: <http://www.w3.org/2002/12/cal/ical#>
PREFIX swc: <http://data.semanticweb.org/ns/swc/ontology#>
```

```
SELECT DISTINCT ?resource ?value
WHERE { ?resource <http://data.semanticweb.org/ns/swc/ontology#address> ?value }
ORDER BY ?resource ?value
```

Browse:

- [Classes](#)
- [Properties](#)
- [Named Graphs](#)

Results: [Browse](#) [Go!](#) [Reset](#)

All uses of property <http://data.semanticweb.org/ns/swc/ontology#address>:

resource	value
< http://data.semanticweb.org/conference/iswc/2001 >	"Stanford University, California, USA"
< http://data.semanticweb.org/conference/iswc/2002 >	"Sardinia, Italy"
< http://data.semanticweb.org/conference/iswc/2003 >	"Sanibel Island, Florida, USA"

SPARQL language

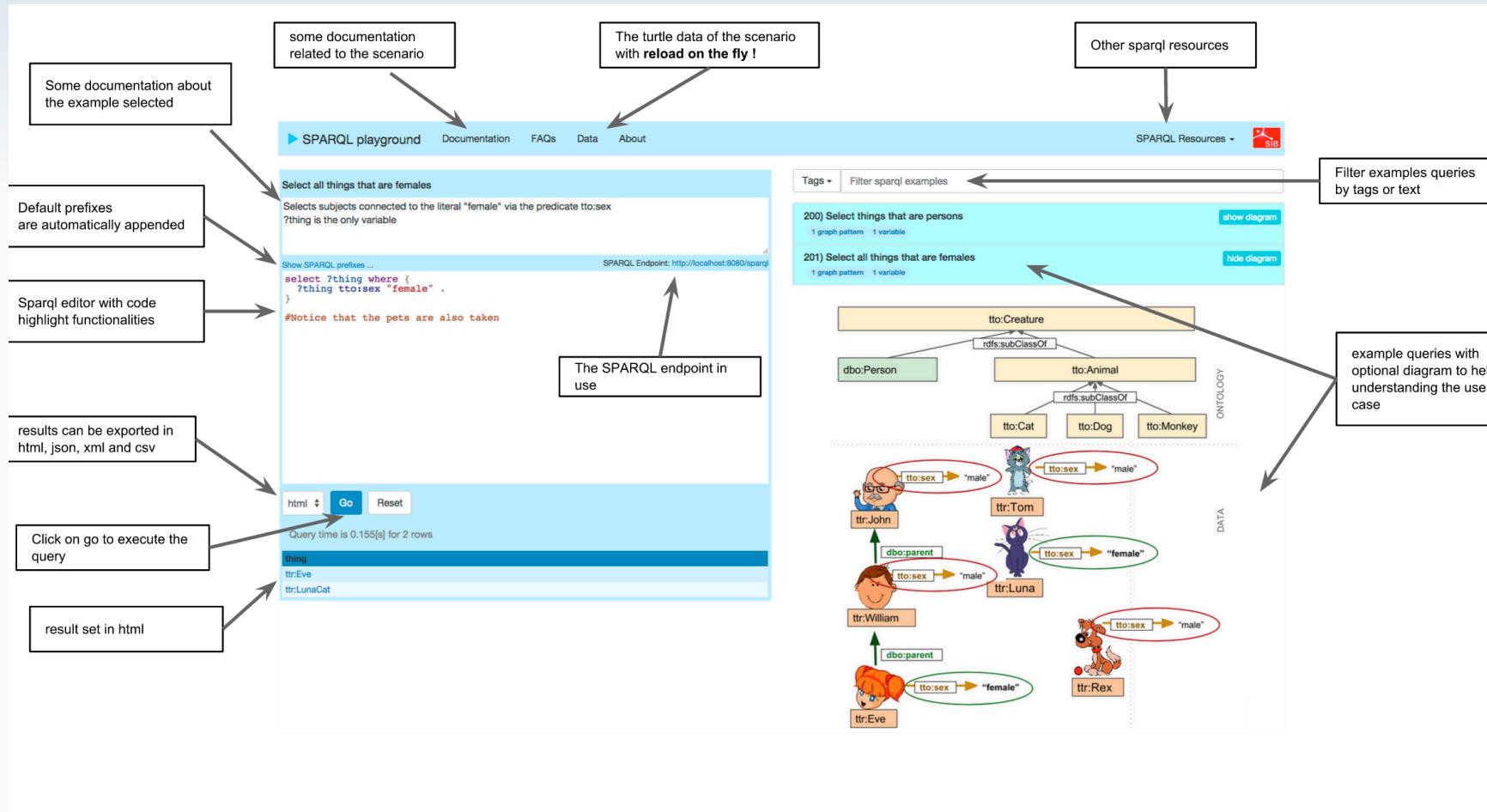
Query language

- SELECT, ASK
- DESCRIBE, CONSTRUCT

Update

- INSERT, DELETE
- CREATE, LOAD, CLEAR, DROP, COPY, MOVE, ADD

sparql-playground



The end

Many thanks to

Jerven Bolleman

Daniel Teixeira

Alain Gateau

Monique Zahn

Pascale Gaudet