

# Chapter 3

## Querying RDF stores with SPARQL



# TL;DR

- We will want to query large RDF datasets, e.g. LOD
- SPARQL is the SQL of RDF
- SPARQL is a language to query and update triples in one or more triples stores
- It's key to exploiting Linked Open Data

## Three RDF use cases

- *Markup web documents* with semi-structured data for better understanding by search engines
- Use as a *data interchange language* that's more flexible and has a richer semantic schema than XML or SQL
- Assemble and link large datasets and publish as knowledge bases to support a domain (e.g., genomics) or in general (DBpedia)

# Three RDF use cases

- *Markup web documents* with semi-structured data for better understanding by search engines (Microdata)
- Use as a *data interchange language* that's more flexible and has a richer semantic schema than XML or SQL
- Assemble and link large datasets and publish as knowledge bases to support a domain (e.g., genomics) or in general (DBpedia)
  - Such knowledge bases may be very large, e.g., Dbpedia has ~300M triples, Freebase has ~3B
  - Using such large datasets requires a language to query and update it

# Semantic Web

Use Semantic Web Technology to  
publish shared data & knowledge

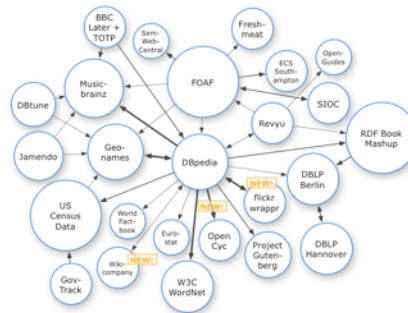
Semantic web technologies  
allow machines to share data  
and knowledge using common  
web language and protocols.

~ 1997

Semantic Web beginning

# Semantic Web => Linked Open Data

Use Semantic Web Technology to publish shared data & knowledge



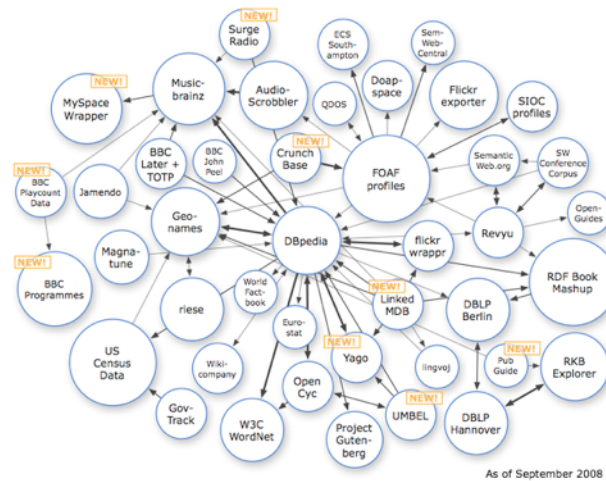
2007

Data is inter-linked to support integration and fusion of knowledge

LOD beginning

# Semantic Web => Linked Open Data

Use Semantic Web Technology to publish shared data & knowledge



As of September 2008

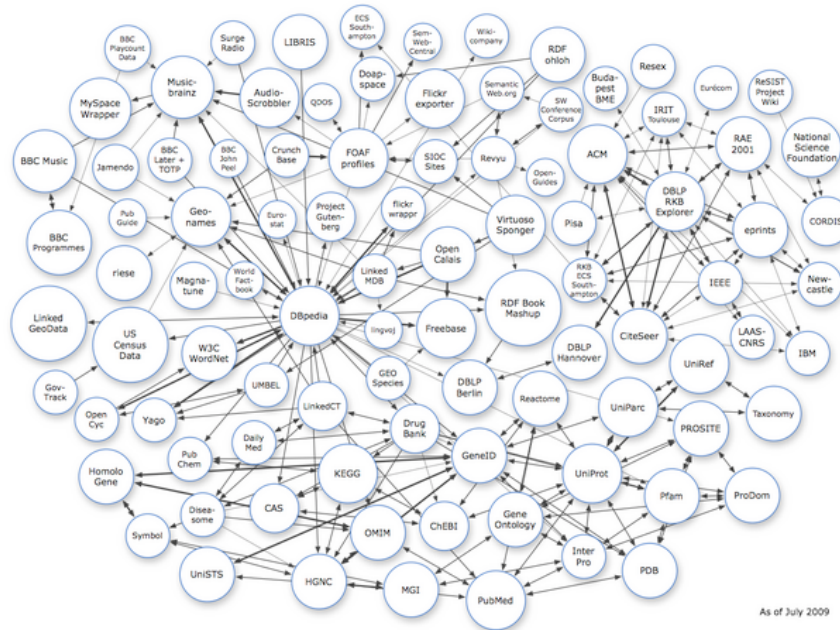
2008

Data is inter-linked to support integration and fusion of knowledge

LOD growing

# Semantic Web => Linked Open Data

Use Semantic Web Technology to publish shared data & knowledge



2009

Data is inter-linked to support integration and fusion of knowledge

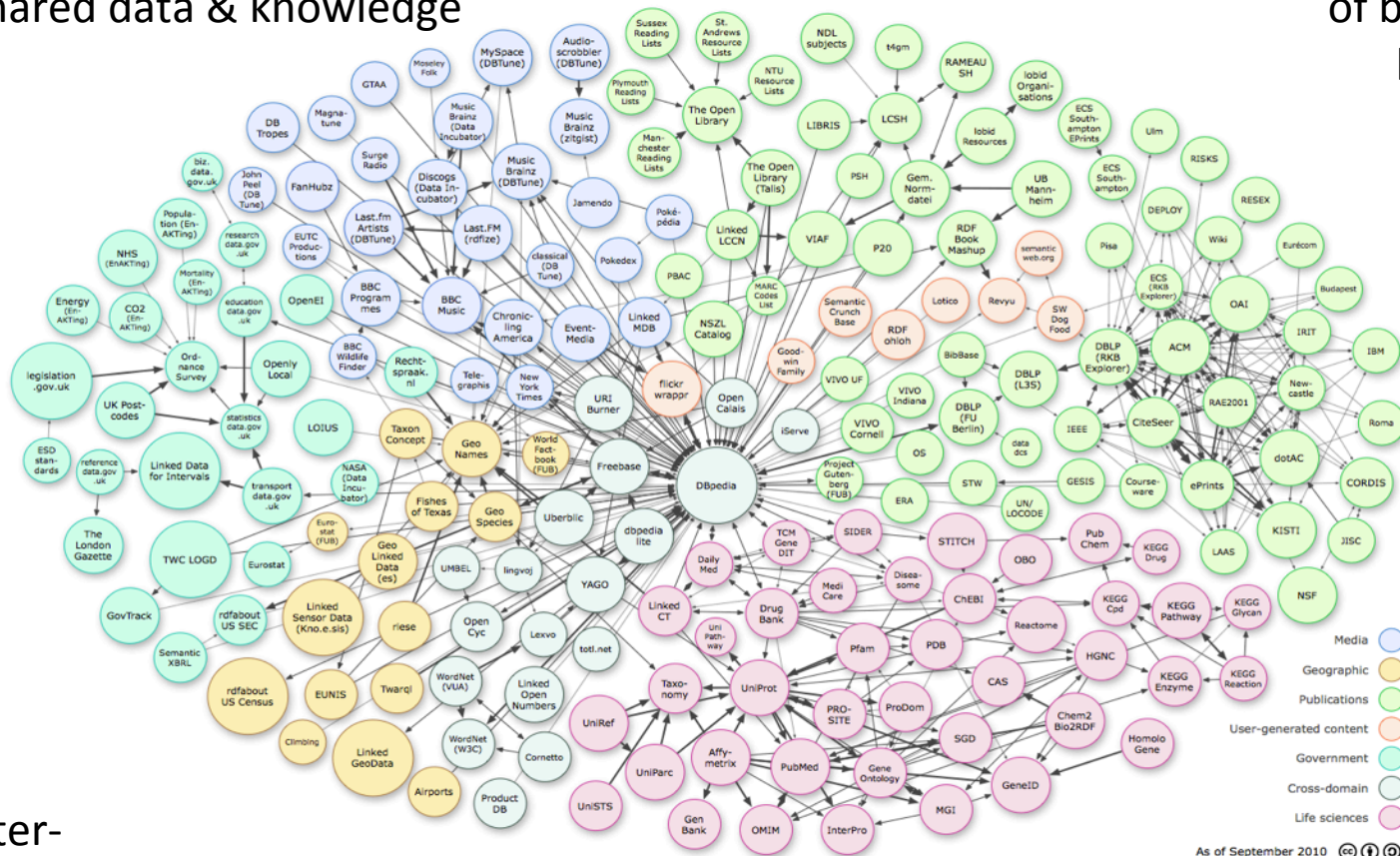
... and growing



# Linked Open Data

Use Semantic Web Technology to publish shared data & knowledge

LOD is the new Cyc: a common source of background knowledge



Data is inter-linked to support integration and fusion of knowledge

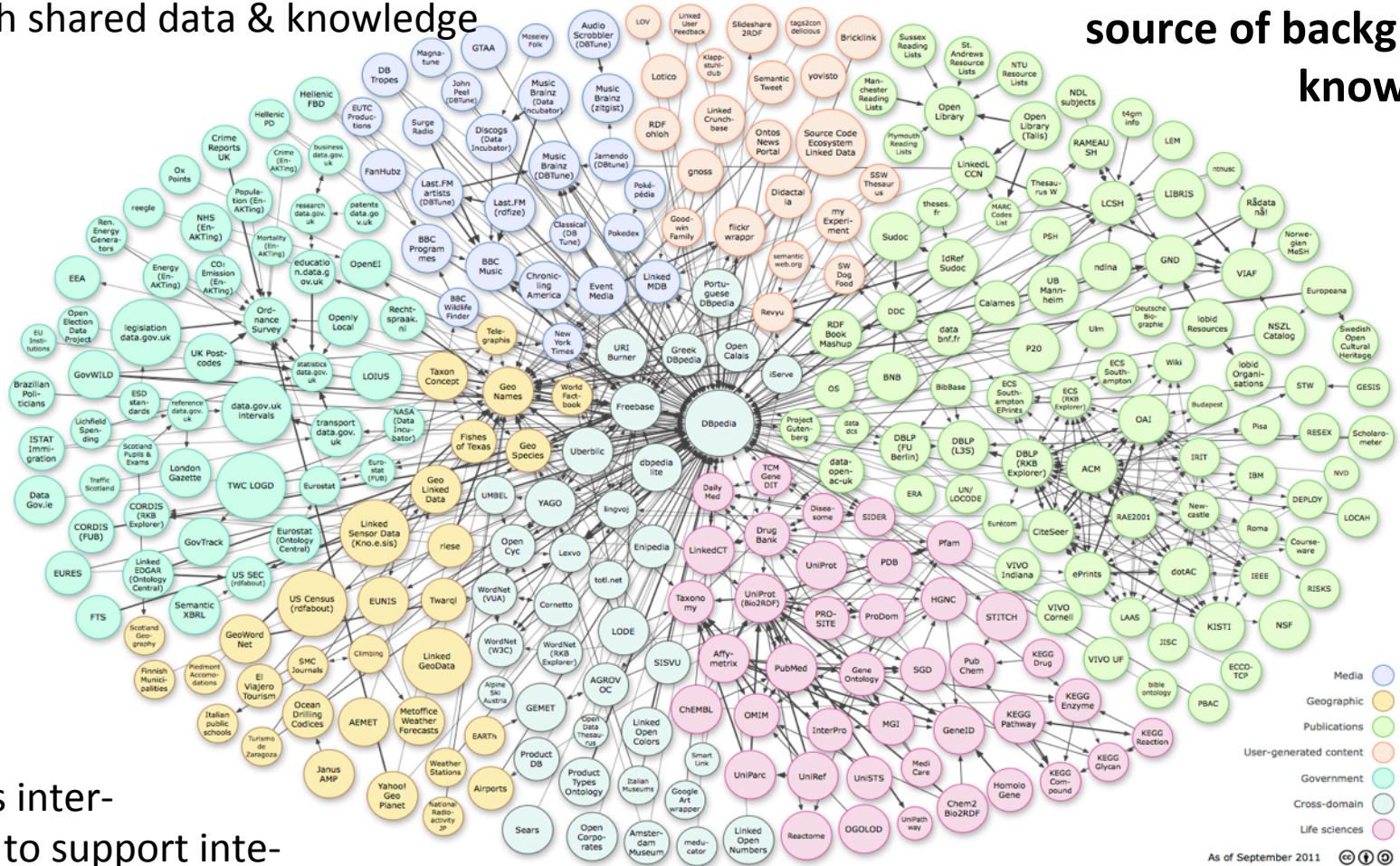
2010

...growing faster

# Linked Open Data

Use Semantic Web Technology to publish shared data & knowledge

LOD is the new Cyc: a common source of background knowledge



Data is inter-linked to support integration and fusion of knowledge

As of September 2011

2011: 31B facts in 295 datasets interlinked by 504M assertions on [ckan.net](http://ckan.net)

# Linked Open Data (LOD)



- Linked **data** is just RDF data, typically just the instances ([ABOX](#)), not schema ([TBOX](#))
- RDF data is a graph of triples
  - URI URI string  
dbr:Barack\_Obama dbo:spouse “Michelle Obama”
  - URI URI URI  
dbr:Barack\_Obama dbo:spouse dbpedia:Michelle\_Obama
- Best **linked** data practice prefers the 2<sup>nd</sup> pattern, using nodes rather than strings for “entities”
- Liked **open** data is just linked data freely accessible on the Web along with any required ontologies

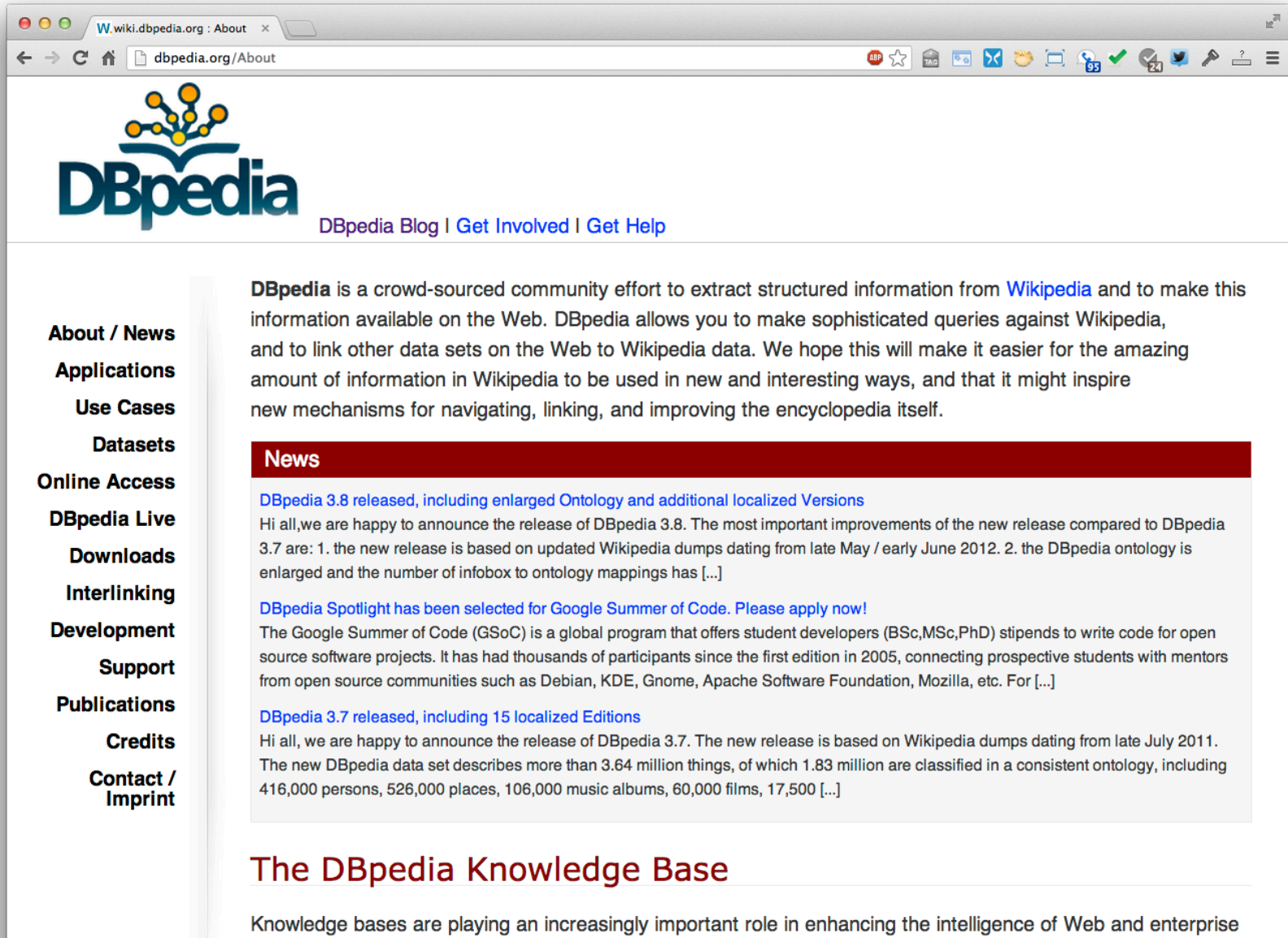


# The Linked Data Mug



See [Linked Data Rules](#), Tim Berners-Lee, circa 2006

# Dbpedia: Wikipedia data in RDF



The screenshot shows a web browser window with the URL `W.wiki.dbpedia.org : About` and `dbpedia.org/About`. The page features the DBpedia logo, which consists of a stylized tree of nodes above the text "DBpedia". Below the logo are links for "DBpedia Blog", "Get Involved", and "Get Help".

**About / News**  
**Applications**  
**Use Cases**  
**Datasets**  
**Online Access**  
**DBpedia Live**  
**Downloads**  
**Interlinking**  
**Development**  
**Support**  
**Publications**  
**Credits**  
**Contact / Imprint**

**DBpedia** is a crowd-sourced community effort to extract structured information from [Wikipedia](#) and to make this information available on the Web. DBpedia allows you to make sophisticated queries against Wikipedia, and to link other data sets on the Web to Wikipedia data. We hope this will make it easier for the amazing amount of information in Wikipedia to be used in new and interesting ways, and that it might inspire new mechanisms for navigating, linking, and improving the encyclopedia itself.

### News

[DBpedia 3.8 released, including enlarged Ontology and additional localized Versions](#)  
Hi all, we are happy to announce the release of DBpedia 3.8. The most important improvements of the new release compared to DBpedia 3.7 are: 1. the new release is based on updated Wikipedia dumps dating from late May / early June 2012. 2. the DBpedia ontology is enlarged and the number of infobox to ontology mappings has [...]

[DBpedia Spotlight has been selected for Google Summer of Code. Please apply now!](#)  
The Google Summer of Code (GSoC) is a global program that offers student developers (BSc, MSc, PhD) stipends to write code for open source software projects. It has had thousands of participants since the first edition in 2005, connecting prospective students with mentors from open source communities such as Debian, KDE, Gnome, Apache Software Foundation, Mozilla, etc. For [...]

[DBpedia 3.7 released, including 15 localized Editions](#)  
Hi all, we are happy to announce the release of DBpedia 3.7. The new release is based on Wikipedia dumps dating from late July 2011. The new DBpedia data set describes more than 3.64 million things, of which 1.83 million are classified in a consistent ontology, including 416,000 persons, 526,000 places, 106,000 music albums, 60,000 films, 17,500 [...]

## The DBpedia Knowledge Base

Knowledge bases are playing an increasingly important role in enhancing the intelligence of Web and enterprise

# Available for download

**3. Canonicalized Datasets**

These datasets contain triples extracted from the respective Wikipedia whose subject and object resource have an equivalent English article. [more...](#)

All DBpedia IRIs/URIs in the canonicalized datasets use the generic namespace `http://dbpedia.org/resource/`. For backwards compatibility, the N-Triples files (.nt, .nq) use URIs, e.g. `http://dbpedia.org/resource/Bras%C3%ADlia`. The Turtle (.ttl) files use IRIs, e.g. `http://dbpedia.org/resource/Brasília`.

**NOTE:** You can find DBpedia dumps in 111 languages at our [DBpedia download server](#).

*Click on the dataset names to obtain additional information. Click on the question mark next to a download link to preview file contents.*

Dataset	en	bg	ca	cs	de	el	es	fr	hu	it	ko	pl	pt	ru	sl	tr
<a href="#">Ontology Infobox Types</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>
	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>
	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>
<a href="#">Ontology Infobox Properties</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>
	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>
	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>
<a href="#">Ontology Infobox Properties (Specific)</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>
	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>
	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>
<b>Dataset</b>	<b>en</b>	<b>bg</b>	<b>ca</b>	<b>cs</b>	<b>de</b>	<b>el</b>	<b>es</b>	<b>fr</b>	<b>hu</b>	<b>it</b>	<b>ko</b>					
<a href="#">Titles</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>
	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>
	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>
<a href="#">Short Abstracts</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>
	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>
	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>
<a href="#">Extended Abstracts</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>
	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>	<a href="#">nq ?</a>
	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>	<a href="#">ttl ?</a>
	<a href="#">nt ?</a>	--	--	--	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	<a href="#">nt ?</a>	--	<a href="#">nt ?</a>	--	--	<a href="#">nt ?</a>	<a href="#">nt ?</a>	--	--

- Broken up into files by information type
- Contains all text, links, infobox data, etc.
- Supported by several ontologies
- Updated ~ every 3 months
- About 300M triples!

# Queryable

W.wiki.dbpedia.org : Online x SPARQL Explorer for http:// x

dbpedia.org/snorql/

## SPARQL Explorer for http://dbpedia.org/sparql

**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#>
```

```
SELECT * WHERE {
...
}
```

Results:

Powered by [OpenLink Virtuoso](#) and [dbpedia](#)

- You can query any of several RDF triple stores
- Or download the data, load into a store and query it locally

# Browseable

**About: Baltimore**

An Entity of Type : [Independent city \(United States\)](#), from Named Graph : <http://live.dbpedia.org>, within Data Space : [live.dbpedia.org](http://live.dbpedia.org)

Baltimore is the largest city in the U.S. state of Maryland and the 24th largest city in the country. It is located in the central area of the state along the tidal portion of the Patapsco River, an arm of the Chesapeake Bay. The independent city is often referred to as Baltimore City to distinguish it from surrounding Baltimore County.

Property	Value
<a href="#">dbpedia-owl:PopulatedPlace/area</a>	▪ 1.0E-6
<a href="#">dbpedia-owl:PopulatedPlace/areaTotal</a>	▪ 238.4 ▪ 238.41358553264945
<a href="#">dbpedia-owl:PopulatedPlace/populationDensity</a>	▪ 2962.6 ▪ 2961.9827092583737
<a href="#">dbpedia-owl:abstract</a>	▪ Baltimore is the largest city in the U.S. state of Maryland and the 24th largest city in the country. It is located in the central area of the state along the tidal portion of the Patapsco River, an arm of the Chesapeake Bay. The independent city is often referred to as Baltimore City to distinguish it from surrounding Baltimore County. Founded in 1729, Baltimore is the second largest seaport in the Mid-Atlantic United States and is situated closer to Midwestern major seaport on the East Coast. Baltimore's Inner Harbor was once the second leading United States and a major manufacturing center. After a decline in manufacturing, Baltimore economy. At 620,961 residents in 2010, Baltimore's population has decreased by one-third. The Baltimore Metropolitan Area has grown steadily to approximately 2.7 million residents in the country. Baltimore is also a principal city in the larger Baltimore–Washington combined statistical area with 8.4 million residents.
<a href="#">dbpedia-owl:area</a>	▪ 1.000000 (xsd:double)
<a href="#">dbpedia-owl:areaCode</a>	▪ 410, 443
<a href="#">dbpedia-owl:areaLand</a>	▪ 209600000.000000 (xsd:double) ▪ 209643997.603037 (xsd:double)
<a href="#">dbpedia-owl:areaTotal</a>	▪ 238400000.000000 (xsd:double) ▪ 238413585.532649 (xsd:double)
<a href="#">dbpedia-owl:areaWater</a>	▪ 28769587.929612 (xsd:double) ▪ 28800000.000000 (xsd:double)
<a href="#">dbpedia-owl:elevation</a>	▪ 10.000000 (xsd:double) ▪ 10.058400 (xsd:double)
<a href="#">dbpedia-owl:isPartOf</a>	▪ <a href="#">dbpedia:Maryland</a>
<a href="#">dbpedia-owl:leaderName</a>	▪ <a href="#">dbpedia:Stephanie_C._Rawlings-Blake</a>
<a href="#">dbpedia-owl:leaderTitle</a>	▪ Mayor ▪ State Senate ▪ U.S. House

- There are also RDF browsers
- These are driven by queries against a RDF triple store loaded with the DBpedia data



# Why an RDF Query Language?

- Why not use an XML query language?
- XML at a lower level of abstraction than RDF
- There are various ways of syntactically representing an RDF statement in XML
- Thus we would require several XPath queries, e.g.
  - **//uni:lecturer/uni:title** if **uni:title** element
  - **//uni:lecturer/@uni:title** if **uni:title** attribute
  - Both XML representations equivalent!

# SPARQL

- A key to exploiting such large RDF data sets is the SPARQL query language
- Sparql Protocol And Rdf Query Language
- W3C began developing a spec for a query language in 2004
- There were/are other [RDF query languages](#), and extensions, e.g., RQL and Jena's [ARQ](#)
- [SPARQL](#) a W3C recommendation in 2008 and [SPARQL 1.1](#) in 2013
- Most triple stores support SPARQL 1.1

# SPARQL Example

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?age
WHERE {
    ?person a foaf:Person.
    ?person foaf:name ?name.
    ?person foaf:age ?age
}
ORDER BY ?age DESC
LIMIT 10
```

# SPARQL Protocol, Endpoints, APIs

- SPARQL query language
- SPROT = SPARQL Protocol for RDF
  - Among other things specifies how results can be encoded as RDF, XML or JSON
- SPARQL endpoint
  - Service that accepts queries and returns results via HTTP
  - Either generic (fetching data as needed) or specific (querying an associated triple store)
  - May be a service for federated queries

# SPARQL Basic Queries

- SPARQL is based on matching graph patterns
- The simplest graph pattern is the triple pattern
  - *?person foaf:name ?name*
  - Like an RDF triple, but with variables
  - Variables begin with a question mark
- Combining triple patterns gives a graph pattern; an exact match to a graph is needed
- Like SQL, returns a set of results, one for for each way the graph pattern can be instantiated

# Turtle Like Syntax

As in Turtle and N3, we can omit a common subject in a graph pattern.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
SELECT ?name ?age
```

```
WHERE {
```

```
  ?person a foaf:Person;
```

```
    foaf:name ?name;
```

```
    foaf:age ?age
```

```
}
```

# Optional Data

- Query fails unless the entire pattern matches
- We often want to collect some information that might not always be available
- Note difference with relational model

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
SELECT ?name ?age
```

```
WHERE {
```

```
  ?person a foaf:Person;
```

```
    foaf:name ?name.
```

```
OPTIONAL {?person foaf:age ?age}
```

```
}
```

# Example of a Generic Endpoint

- Use the sparql endpoint at
  - <http://demo.openlinksw.com/sparql>
- To query graph at
  - <http://ebiq.org/person/foaf/Tim/Finin/foaf.rdf>

- For foaf knows relations

```
SELECT ?name ?p2
```

```
WHERE { ?person a foaf:Person;
```

```
        foaf:name ?name;
```

```
        foaf:knows ?p2. }
```



# Example

The screenshot shows the Virtuoso SPARQL Query Editor interface. The browser address bar displays `demo.openlinksw.com/sparql`. The page title is "Virtuoso SPARQL Query Editor".

**Default Data Set Name (Graph IRI)**  
`http://eblquity.umbc.edu/person/foaf/Tim/Finin/foaf.rdf`

**Query Text**

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?p2
WHERE {
  ?person a foaf:Person;
          foaf:name ?name;
          foaf:knows ?p2.
}
```

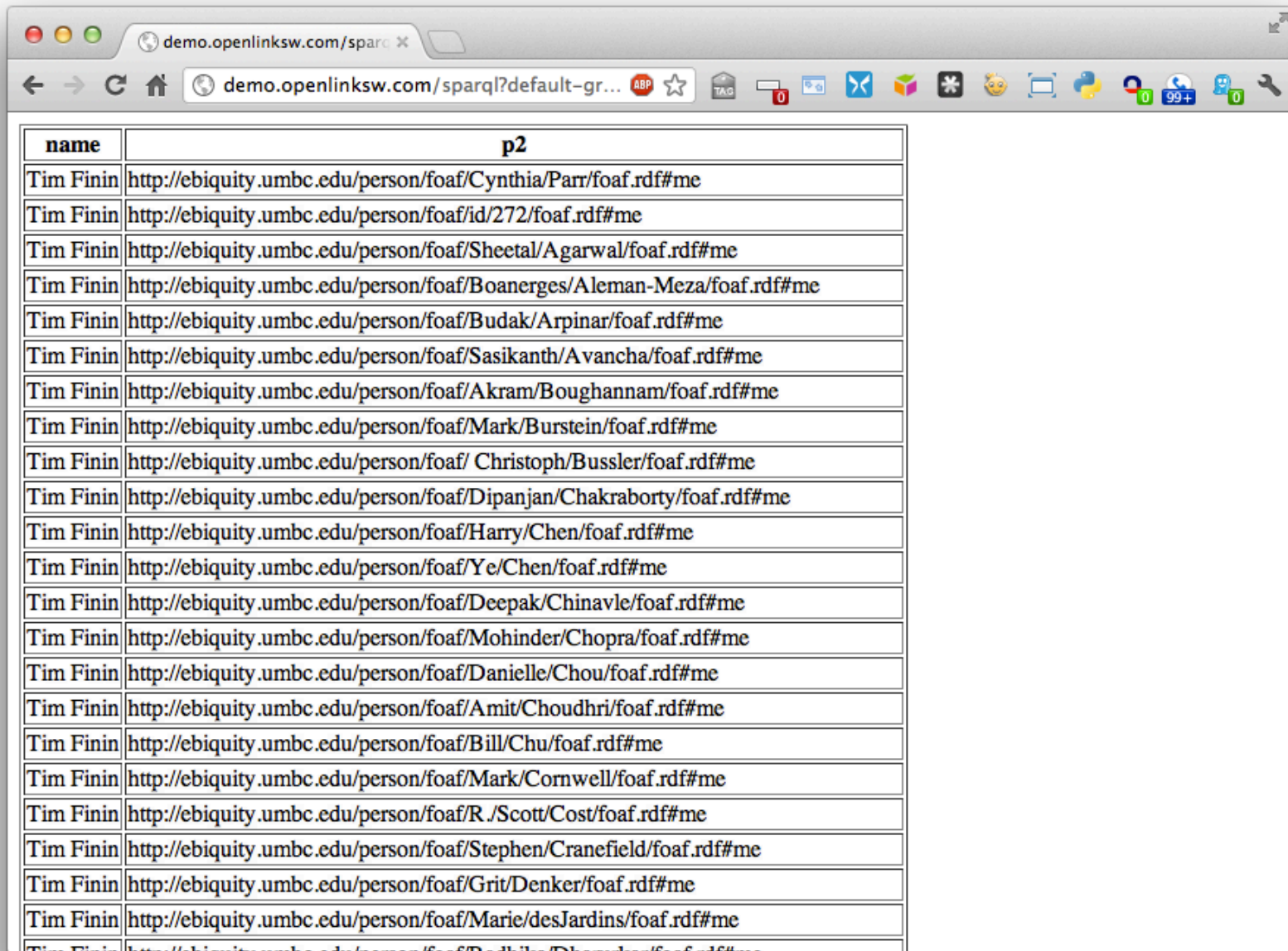
**Options:**

- Sponging:** Retrieve remote RDF data for all missing source graphs
- Results Format:** HTML
- Execution timeout:** 0 milliseconds (values less than 1000 are ignored)
- Options:**  Strict checking of void variables

(The result can only be sent back to browser, not saved on the server, see [details](#))

**Run Query** **Reset**

# Query results as HTML



The image shows a web browser window with the address bar displaying `demo.openlinksw.com/sparql?default-gr...`. The browser's toolbar includes navigation buttons (back, forward, refresh, home), a search bar, and various extension icons. The main content area displays a table with two columns: **name** and **p2**. The **name** column contains the text "Tim Finin" for every row. The **p2** column contains a list of URIs, each representing a person's profile page on the EBIquity system. The URIs are: `http://ebiquity.umbc.edu/person/foaf/Cynthia/Parr/foaf.rdf#me`, `http://ebiquity.umbc.edu/person/foaf/id/272/foaf.rdf#me`, `http://ebiquity.umbc.edu/person/foaf/Sheetal/Agarwal/foaf.rdf#me`, `http://ebiquity.umbc.edu/person/foaf/Boanerges/Aleman-Meza/foaf.rdf#me`, `http://ebiquity.umbc.edu/person/foaf/Budak/Arpinar/foaf.rdf#me`, `http://ebiquity.umbc.edu/person/foaf/Sasikanth/Avancha/foaf.rdf#me`, `http://ebiquity.umbc.edu/person/foaf/Akram/Boughannam/foaf.rdf#me`, `http://ebiquity.umbc.edu/person/foaf/Mark/Burstein/foaf.rdf#me`, `http://ebiquity.umbc.edu/person/foaf/ Christoph/Bussler/foaf.rdf#me`, `http://ebiquity.umbc.edu/person/foaf/Dipanjan/Chakraborty/foaf.rdf#me`, `http://ebiquity.umbc.edu/person/foaf/Harry/Chen/foaf.rdf#me`, `http://ebiquity.umbc.edu/person/foaf/Ye/Chen/foaf.rdf#me`, `http://ebiquity.umbc.edu/person/foaf/Deepak/Chinavle/foaf.rdf#me`, `http://ebiquity.umbc.edu/person/foaf/Mohinder/Chopra/foaf.rdf#me`, `http://ebiquity.umbc.edu/person/foaf/Danielle/Chou/foaf.rdf#me`, `http://ebiquity.umbc.edu/person/foaf/Amit/Choudhri/foaf.rdf#me`, `http://ebiquity.umbc.edu/person/foaf/Bill/Chu/foaf.rdf#me`, `http://ebiquity.umbc.edu/person/foaf/Mark/Cornwell/foaf.rdf#me`, `http://ebiquity.umbc.edu/person/foaf/R./Scott/Cost/foaf.rdf#me`, `http://ebiquity.umbc.edu/person/foaf/Stephen/Cranefield/foaf.rdf#me`, `http://ebiquity.umbc.edu/person/foaf/Grit/Denker/foaf.rdf#me`, `http://ebiquity.umbc.edu/person/foaf/Marie/desJardins/foaf.rdf#me`, and `http://ebiquity.umbc.edu/person/foaf/Redhika/Dhanuvar/foaf.rdf#me`.

name	p2
Tim Finin	<a href="http://ebiquity.umbc.edu/person/foaf/Cynthia/Parr/foaf.rdf#me">http://ebiquity.umbc.edu/person/foaf/Cynthia/Parr/foaf.rdf#me</a>
Tim Finin	<a href="http://ebiquity.umbc.edu/person/foaf/id/272/foaf.rdf#me">http://ebiquity.umbc.edu/person/foaf/id/272/foaf.rdf#me</a>
Tim Finin	<a href="http://ebiquity.umbc.edu/person/foaf/Sheetal/Agarwal/foaf.rdf#me">http://ebiquity.umbc.edu/person/foaf/Sheetal/Agarwal/foaf.rdf#me</a>
Tim Finin	<a href="http://ebiquity.umbc.edu/person/foaf/Boanerges/Aleman-Meza/foaf.rdf#me">http://ebiquity.umbc.edu/person/foaf/Boanerges/Aleman-Meza/foaf.rdf#me</a>
Tim Finin	<a href="http://ebiquity.umbc.edu/person/foaf/Budak/Arpinar/foaf.rdf#me">http://ebiquity.umbc.edu/person/foaf/Budak/Arpinar/foaf.rdf#me</a>
Tim Finin	<a href="http://ebiquity.umbc.edu/person/foaf/Sasikanth/Avancha/foaf.rdf#me">http://ebiquity.umbc.edu/person/foaf/Sasikanth/Avancha/foaf.rdf#me</a>
Tim Finin	<a href="http://ebiquity.umbc.edu/person/foaf/Akram/Boughannam/foaf.rdf#me">http://ebiquity.umbc.edu/person/foaf/Akram/Boughannam/foaf.rdf#me</a>
Tim Finin	<a href="http://ebiquity.umbc.edu/person/foaf/Mark/Burstein/foaf.rdf#me">http://ebiquity.umbc.edu/person/foaf/Mark/Burstein/foaf.rdf#me</a>
Tim Finin	<a href="http://ebiquity.umbc.edu/person/foaf/ Christoph/Bussler/foaf.rdf#me">http://ebiquity.umbc.edu/person/foaf/ Christoph/Bussler/foaf.rdf#me</a>
Tim Finin	<a href="http://ebiquity.umbc.edu/person/foaf/Dipanjan/Chakraborty/foaf.rdf#me">http://ebiquity.umbc.edu/person/foaf/Dipanjan/Chakraborty/foaf.rdf#me</a>
Tim Finin	<a href="http://ebiquity.umbc.edu/person/foaf/Harry/Chen/foaf.rdf#me">http://ebiquity.umbc.edu/person/foaf/Harry/Chen/foaf.rdf#me</a>
Tim Finin	<a href="http://ebiquity.umbc.edu/person/foaf/Ye/Chen/foaf.rdf#me">http://ebiquity.umbc.edu/person/foaf/Ye/Chen/foaf.rdf#me</a>
Tim Finin	<a href="http://ebiquity.umbc.edu/person/foaf/Deepak/Chinavle/foaf.rdf#me">http://ebiquity.umbc.edu/person/foaf/Deepak/Chinavle/foaf.rdf#me</a>
Tim Finin	<a href="http://ebiquity.umbc.edu/person/foaf/Mohinder/Chopra/foaf.rdf#me">http://ebiquity.umbc.edu/person/foaf/Mohinder/Chopra/foaf.rdf#me</a>
Tim Finin	<a href="http://ebiquity.umbc.edu/person/foaf/Danielle/Chou/foaf.rdf#me">http://ebiquity.umbc.edu/person/foaf/Danielle/Chou/foaf.rdf#me</a>
Tim Finin	<a href="http://ebiquity.umbc.edu/person/foaf/Amit/Choudhri/foaf.rdf#me">http://ebiquity.umbc.edu/person/foaf/Amit/Choudhri/foaf.rdf#me</a>
Tim Finin	<a href="http://ebiquity.umbc.edu/person/foaf/Bill/Chu/foaf.rdf#me">http://ebiquity.umbc.edu/person/foaf/Bill/Chu/foaf.rdf#me</a>
Tim Finin	<a href="http://ebiquity.umbc.edu/person/foaf/Mark/Cornwell/foaf.rdf#me">http://ebiquity.umbc.edu/person/foaf/Mark/Cornwell/foaf.rdf#me</a>
Tim Finin	<a href="http://ebiquity.umbc.edu/person/foaf/R./Scott/Cost/foaf.rdf#me">http://ebiquity.umbc.edu/person/foaf/R./Scott/Cost/foaf.rdf#me</a>
Tim Finin	<a href="http://ebiquity.umbc.edu/person/foaf/Stephen/Cranefield/foaf.rdf#me">http://ebiquity.umbc.edu/person/foaf/Stephen/Cranefield/foaf.rdf#me</a>
Tim Finin	<a href="http://ebiquity.umbc.edu/person/foaf/Grit/Denker/foaf.rdf#me">http://ebiquity.umbc.edu/person/foaf/Grit/Denker/foaf.rdf#me</a>
Tim Finin	<a href="http://ebiquity.umbc.edu/person/foaf/Marie/desJardins/foaf.rdf#me">http://ebiquity.umbc.edu/person/foaf/Marie/desJardins/foaf.rdf#me</a>
Tim Finin	<a href="http://ebiquity.umbc.edu/person/foaf/Redhika/Dhanuvar/foaf.rdf#me">http://ebiquity.umbc.edu/person/foaf/Redhika/Dhanuvar/foaf.rdf#me</a>

# Other result format options

The image shows a user interface for a query tool. On the left, there are several settings: 'Sponging:', 'Results Format:', 'Execution timeout:', and 'Options:'. Below 'Options:' is a note: '(The result can only be sent back...'. At the bottom left are two buttons: 'Run Query' and 'Reset'. A dropdown menu is open over the 'Results Format:' label, listing the following options: 'Auto', 'HTML', 'Spreadsheet', 'XML', 'JSON' (which has a checkmark), 'Javascript', 'NTriples', 'RDF/XML', 'CSV', 'CXML (Pivot Collection)', and 'CXML (Pivot Collection with QRcode)'. In the background, there are faint labels for 'source graphs' and 'less than 1000 are ignored)'.

# Example of a dedicated Endpoint

- Use the sparql endpoint at
  - <http://dbpedia.org/sparql>
- To query DBpedia
- Discover places associated with Pres. Obama

```
PREFIX dbp: <http://dbpedia.org/resource/>
PREFIX dbpo: <http://dbpedia.org/ontology/>
SELECT distinct ?Property ?Place
WHERE {dbp:Barack_Obama ?Property ?Place .
       ?Place rdf:type dbpo:Place .}
```

http://dbpedia.org/sparql

dbpedia lookup

```

PREFIX dbp: <http://dbpedia.org/resource/>
PREFIX dbpo: <http://dbpedia.org/ontology/>
SELECT distinct ?Property ?Place
WHERE {dbp:Barack_Obama ?Property ?Place .
       ?Place rdf:type dbpo:Place .}

```

Property	Place
<a href="http://dbpedia.org/property/birthPlace">http://dbpedia.org/property/birthPlace</a>	<a href="http://dbpedia.org/resource/Hawaii">http://dbpedia.org/resource/Hawaii</a>
<a href="http://dbpedia.org/property/birthPlace">http://dbpedia.org/property/birthPlace</a>	<a href="http://dbpedia.org/resource/Honolulu%2C_Hawaii">http://dbpedia.org/resource/Honolulu%2C_Hawaii</a>
<a href="http://dbpedia.org/property/birthPlace">http://dbpedia.org/property/birthPlace</a>	<a href="http://dbpedia.org/resource/United_States">http://dbpedia.org/resource/United_States</a>
<a href="http://dbpedia.org/property/state">http://dbpedia.org/property/state</a>	<a href="http://dbpedia.org/resource/Illinois">http://dbpedia.org/resource/Illinois</a>
<a href="http://dbpedia.org/property/nationality">http://dbpedia.org/property/nationality</a>	<a href="http://dbpedia.org/resource/United_States">http://dbpedia.org/resource/United_States</a>
<a href="http://dbpedia.org/ontology/nationality">http://dbpedia.org/ontology/nationality</a>	<a href="http://dbpedia.org/resource/United_States">http://dbpedia.org/resource/United_States</a>
<a href="http://dbpedia.org/ontology/birthplace">http://dbpedia.org/ontology/birthplace</a>	<a href="http://dbpedia.org/resource/Hawaii">http://dbpedia.org/resource/Hawaii</a>
<a href="http://dbpedia.org/ontology/birthplace">http://dbpedia.org/ontology/birthplace</a>	<a href="http://dbpedia.org/resource/Honolulu%2C_Hawaii">http://dbpedia.org/resource/Honolulu%2C_Hawaii</a>
<a href="http://dbpedia.org/ontology/birthplace">http://dbpedia.org/ontology/birthplace</a>	<a href="http://dbpedia.org/resource/United_States">http://dbpedia.org/resource/United_States</a>

# To use this you must know

- Know: RDF data model and SPARQL
- Know: Relevant [ontology terms](#) and [CURIEs](#) for individuals
- More difficult than for a typical database because the schema is so large
- Possible solutions:
  - Browse the KB to learn terms and individual CURIEs
  - Query using `rdf:label` and strings
  - Use Lushan Han's intuitive KB

# Search for: dbpedia barack obama



**About: [Barack Obama](#)**

An Entity of Type : [agent](#), from Named Graph : <http://dbpedia.org>, within Data Space : [dbpedia.org](#)

Barack Hussein Obama II is the 44th and current President of the United States, in office since 2009. He is the first African American to hold the office. Born in Honolulu, Hawaii, Obama is a graduate of Columbia University and Harvard Law School, where he was president of the Harvard Law Review. He was a community organizer in Chicago before earning his law degree.

Property	Value
<a href="#">dbpedia-owl:abstract</a>	<ul style="list-style-type: none"><li>Barack Hussein Obama II [bəˈrɑːk hʊˈseɪn oʊˈbɑːmə] ist ein US-ameri Präsident der Vereinigten Staaten. Er wurde bei der Präsidentschafts 2012 für eine zweite Amtsperiode als US-Präsident bestätigt. Obama Kenianers, ist der erste Afroamerikaner in diesem Amt. Obama ist au 1992 Politiker der Demokratischen Partei. Von 2005 bis 2008 gehört Senat der Vereinigten Staaten an. Am 10. Dezember 2009 wurde ihr sich als erster US-Präsident im Amt öffentlich für die Legalisierung vo</li><li>Barack Hussein Obama II is the 44th and current President of the Ur American to hold the office. Born in Honolulu, Hawaii, Obama is a gra where he was president of the Harvard Law Review. He was a comm He worked as a civil rights attorney in Chicago and taught constitutio to 2004. He served three terms representing the 13th District in the Il the United States House of Representatives in 2000. In 2004, Obam represent Illinois in the United States Senate with his victory in the M Democratic National Convention in July, and his election to the Sena 2007, and in 2008, after a close primary campaign against Hillary Ro</li></ul>



# Query using labels

```
PREFIX dbp: <http://dbpedia.org/resource/>
```

```
PREFIX dbpo: <http://dbpedia.org/ontology/>
```

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-  
schema#>
```

```
SELECT distinct ?Property ?Place
```

```
WHERE {?P a dbpo:Person;
```

```
    rdfs:label "Barack Obama"@en;
```

```
    ?Property ?Place .
```

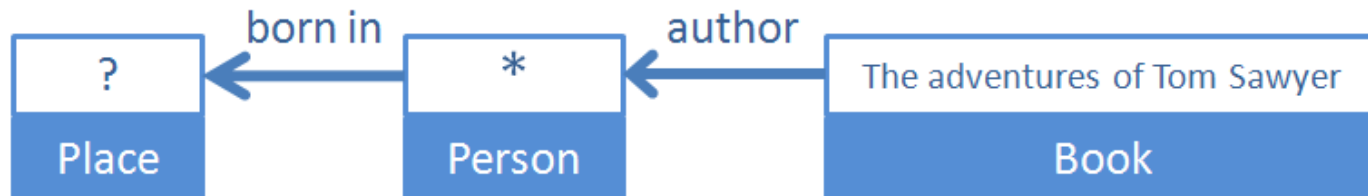
```
?Place rdf:type dbpo:Place .}
```



# Query using labels

```
PREFIX dbp: <http://dbpedia.org/resource/>
PREFIX dbpo: <http://dbpedia.org/ontology/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-
schema#>
SELECT distinct ?P ?Property ?Place
WHERE {?P a dbpo:Person;
       rdfs:label ?Name.
       FILTER regex(?Name, 'obama', 'i')
       ?P ?Property ?Place .
       ?Place rdf:type dbpo:Place .
}
```

# Structured Keyword Queries

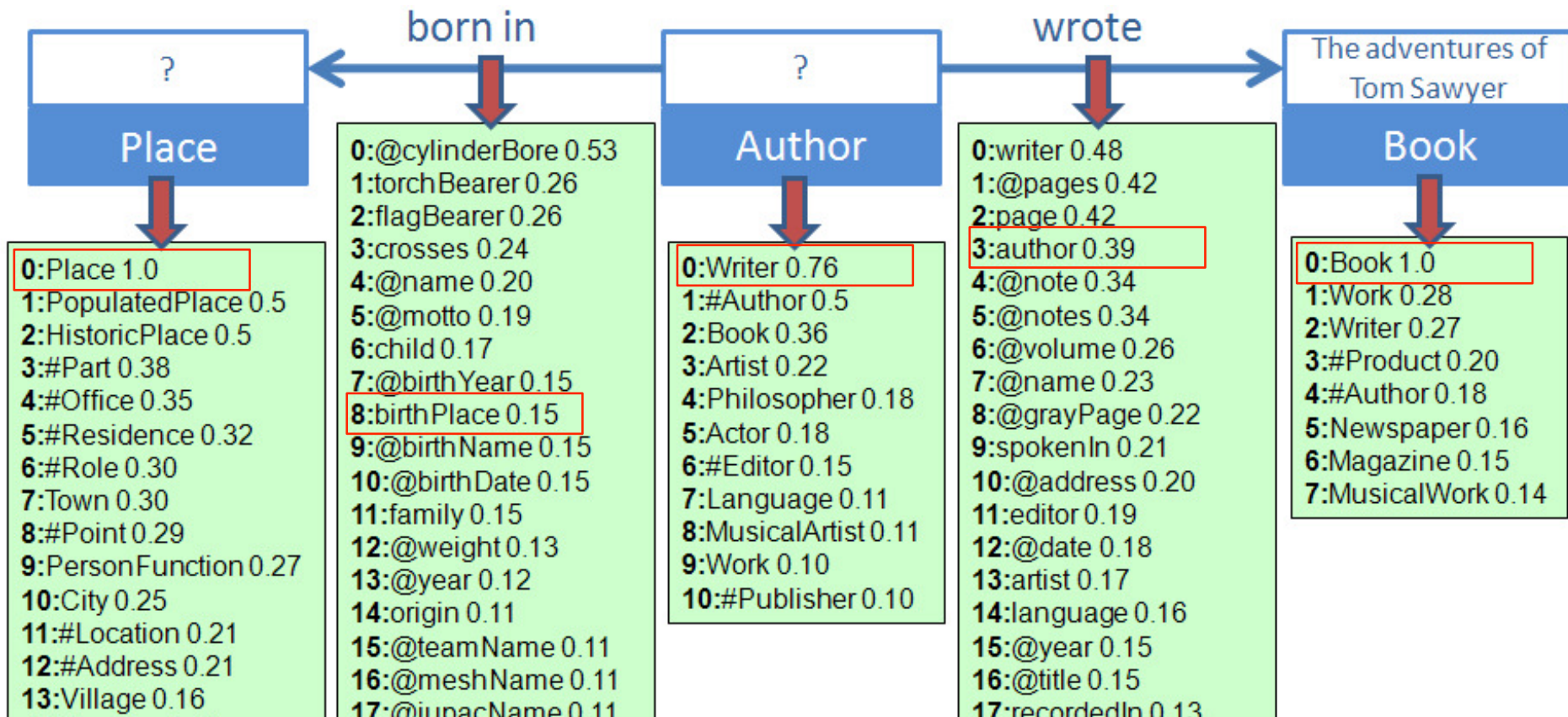


- Nodes are entities and links binary relations
- Entities described by two unrestricted terms: *name* or value and *type* or concept
- Outputs marked with ?
- Compromise between a natural language Q&A system and formal query
  - Users provide compositional structure of the question
  - Free to use their own terms to annotate structure

# Translation result

*Concepts:* Place => Place, Author => Writer, Book => Book

*Properties:* born in => birthPlace, wrote => author (inverse direction)



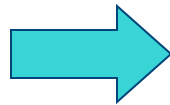
# SPARQL Generation



The translation of a semantic graph query to SPARQL is straightforward given the mappings

## Concepts

- Place => Place
- Author => Writer
- Book => Book



## Relations

- born in => birthPlace
- wrote => author

```
PREFIX dbo: <http://dbpedia.org/ontology/>

SELECT DISTINCT ?x, ?y WHERE {
  ?0 a dbo:Book .
  ?0 rdfs:label ?label0 .
  ?label0 bif:contains "'The adventures of Tom Sawyer"' .
  ?x a dbo:Writer .
  ?y a dbo:Place .
  {?0 dbo:author ?x} .
  {?x dbo:birthPlace ?y} .
}
```

# SELECT FROM

- The FROM clause lets us specify the target graph in the query
- SELECT \* returns all

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
SELECT *
```

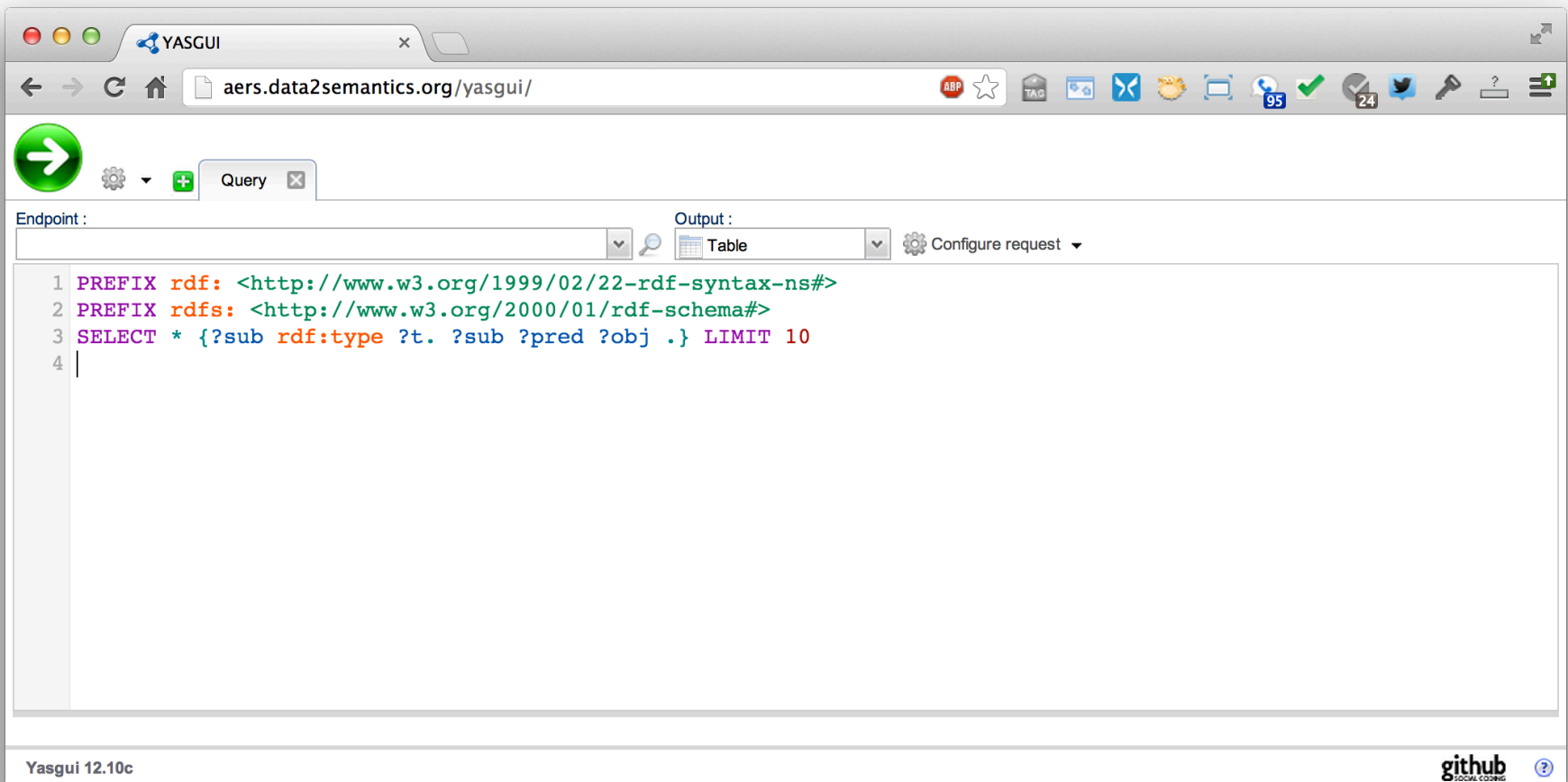
```
FROM <http://ebiq.org/person/foaf/Tim/Finin/foaf.rdf>
```

```
WHERE {
```

```
  ?P1 foaf:knows ?p2
```

```
}
```

# YASGUI generic web client



Try it: <http://aers.data2semantics.org/yasgui/>  
Source: <https://github.com/LaurensRietveld/yasgui>

# FILTER

*Find landlocked countries with a population >15 million*

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

```
PREFIX type: <http://dbpedia.org/class/yago/>
```

```
PREFIX prop: <http://dbpedia.org/property/>
```

```
SELECT ?country_name ?population
```

```
WHERE {
```

```
    ?country a type:LandlockedCountries ;
```

```
        rdfs:label ?country_name ;
```

```
        prop:populationEstimate ?population .
```

```
    FILTER (?population > 15000000) .
```

```
}
```

# FILTER Functions

- Logical: !, &&, ||
- Math: +, -, \*, /
- Comparison: =, !=, >, <, ...
- SPARQL tests: isURI, isBlank, isLiteral, bound
- SPARQL accessors: str, lang, datatype
- Other: sameTerm, langMatches, regex
- Conditionals (SPARQL 1.1): IF, COALESCE
- Constructors (SPARQL 1.1): URI, BNODE, STRDT, STRLANG
- Strings (SPARQL 1.1): STRLEN, SUBSTR, UCASE, ...
- More math (SPARQL 1.1): abs, round, ceil, floor, RAND
- Date/time (SPARQL 1.1): now, year, month, day, hours, ...
- Hashing (SPARQL 1.1): MD5, SHA1, SHA224, SHA256, ...



# Union

- UNION keyword forms disjunction of two graph patterns
- Both subquery results are included

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
PREFIX vCard: <http://www.w3.org/2001/vcard-rdf/3.0#>
```

```
SELECT ?name
```

```
WHERE
```

```
{
```

```
  { [ ] foaf:name ?name } UNION { [ ] vCard:FN ?name }
```

```
}
```

# Query forms

Each form takes a WHERE block to restrict the query

- **SELECT:** Extract raw values from a SPARQL endpoint, the results are returned in a table format
- **CONSTRUCT:** Extract information from the SPARQL endpoint and transform the results into valid RDF
- **ASK:** Returns a simple True/False result for a query on a SPARQL endpoint
- **DESCRIBE** Extract RDF graph from endpoint, the contents of which is left to the endpoint to decide based on what maintainer deems as useful information

# SPARQL 1.1

SPARQL 1.1 includes

- Updated 1.1 versions of SPARQL Query and SPARQL Protocol
- SPARQL 1.1 Update
- SPARQL 1.1 Graph Store HTTP Protocol
- SPARQL 1.1 Service Descriptions
- SPARQL 1.1 Entailments
- SPARQL 1.1 Basic Federated Query

# Summary

- An important usecase for RDF is exploiting large collections of semi-structured data, e.g., the linked open data cloud
- We need a good query language for this
- SPARQL is the SQL of RDF
- SPARQL is a language to query and update triples in one or more triples stores
- It's key to exploiting Linked Open Data