MAS – Web Sémantique

SPARQL : *S*PARQL *P*rotocol *a*nd *R*DF *Q*uery *L*anguage

Julien TSCHERRIG, Joël DUMOULIN, Omar ABOU KHALED

University of Applied Sciences of Western Switzerland
EIA-FR, Bd de Pérolles 80 - CP 32, CH-1705 Fribourg
julien.tscherrig@hefr.ch| Tél: +41 26 429 69 64
joel.dumoulin@hefr.ch| Tél: +41 26 429 69 60
omar.aboukhaled@hefr.ch |Tél: +41 26 429 65 89
http://humantech.eia-fr.ch | Fax: +41 26 429 66 00

Plan

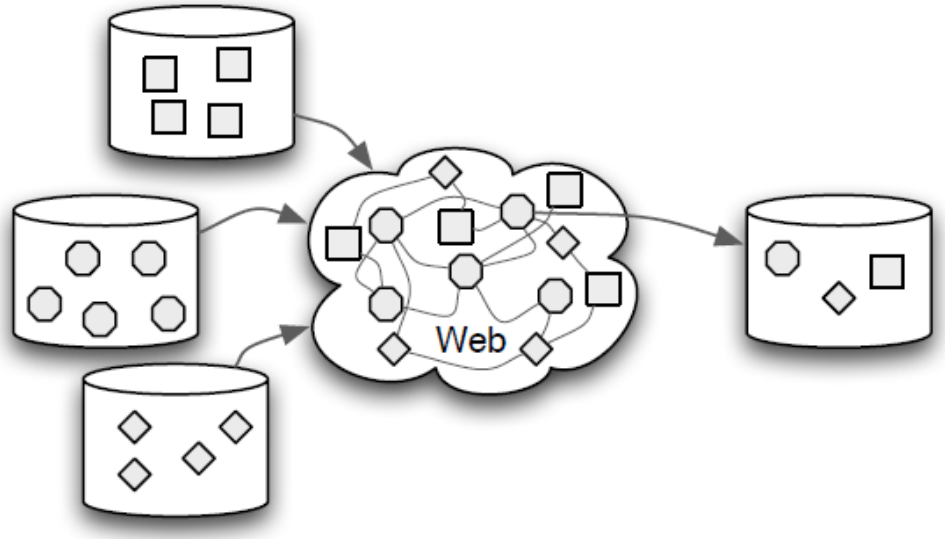
- Recall (LinkedData / RDF)
- Databases
- SPARQL
- DBPedia

Recall (LinkedData / RDF)

MAS – Web Sémantique

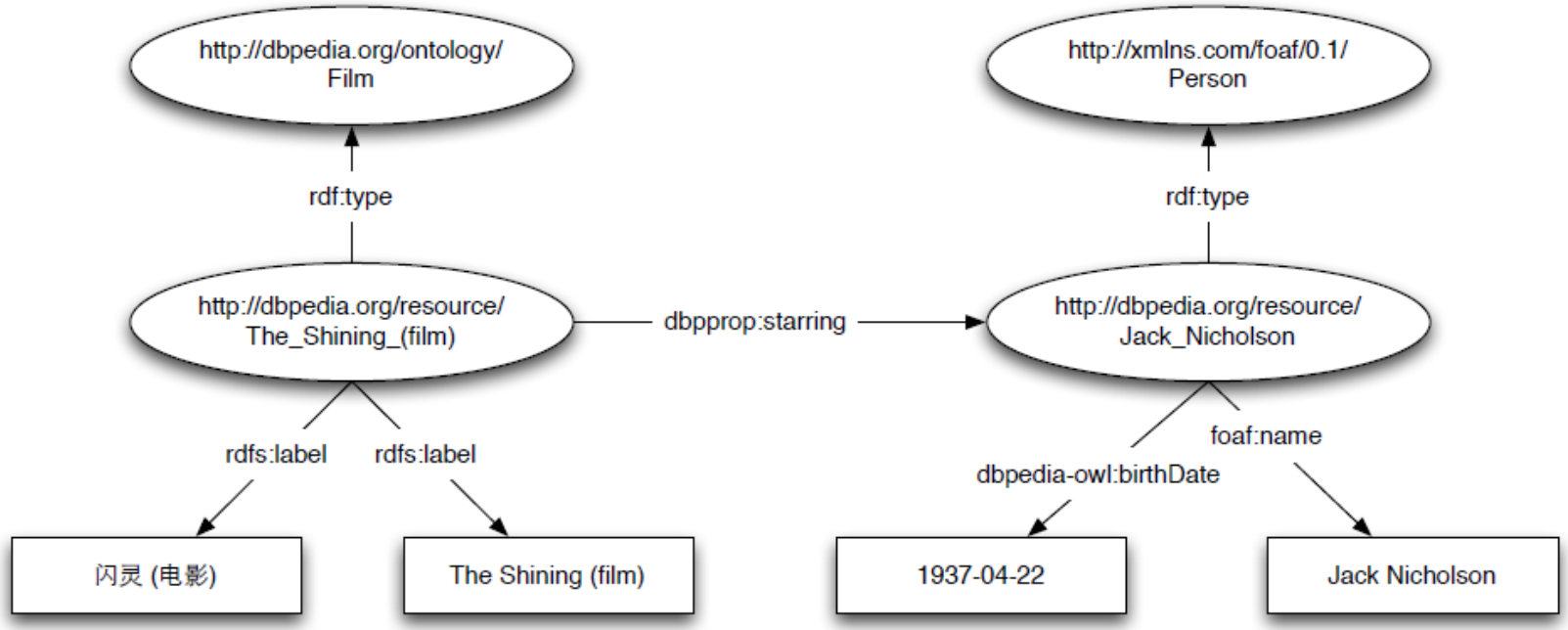
What is Linked Data?

- A method to build a Web of Data
- Architectural style, set of standards



Resource Description Framework (RDF)

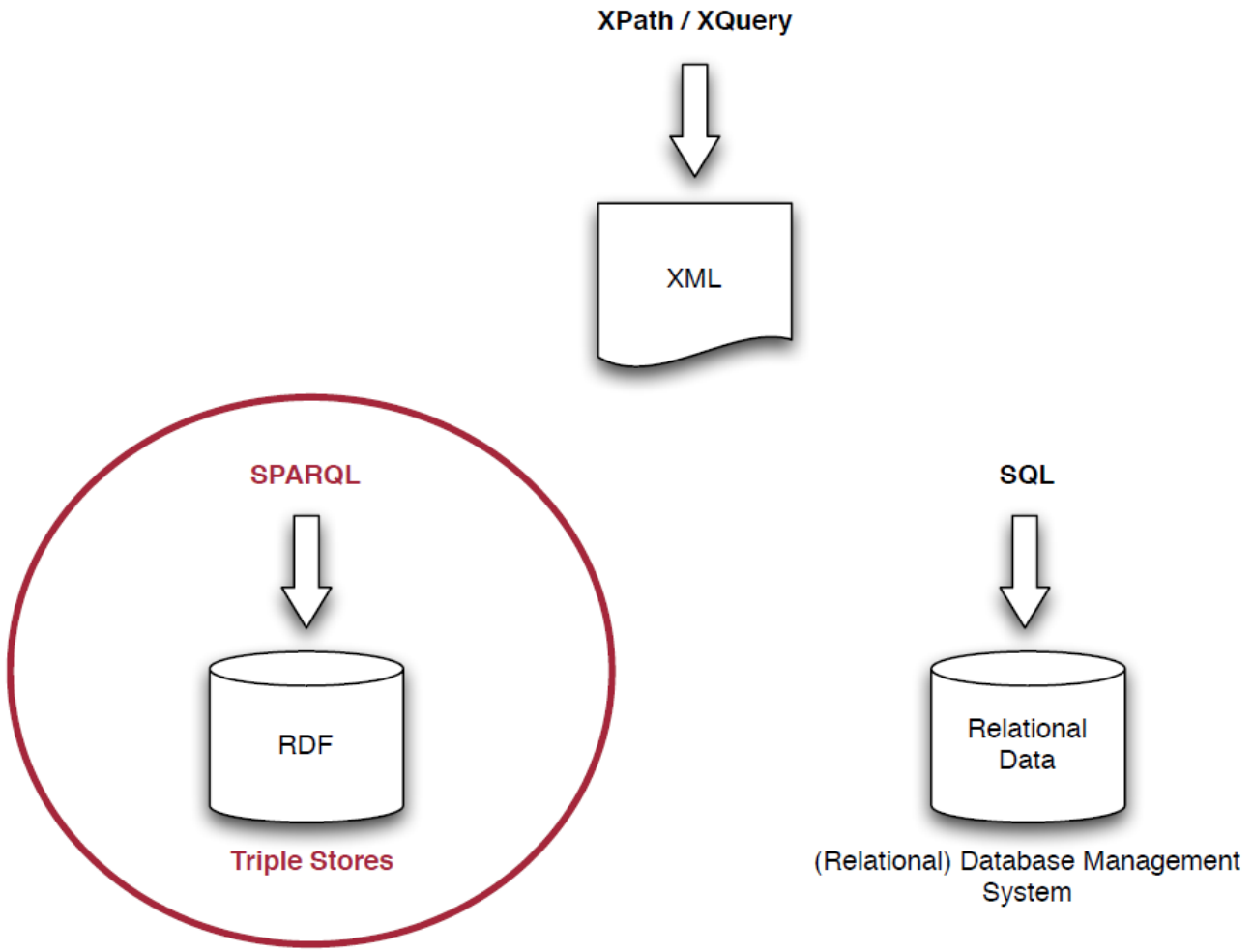
- A model for representing data on the Web
- Several statements (triples) form a graph



Databases

MAS – Web Sémantique

Databases



Databases

- RDF is a model for exposing / exchanging data
- How you store your data internally and how you expose them externally are two pairs of shoes
- Working with RDF does not imply that you must store your data in RDF

Databases (Triple Store)

- Small RDF graphs can be efficiently handled in computers' main memory
- Larger RDF graphs (e.g., DBPedia) require some kind of database management system
- Triple Stores are purpose-built databases for storage and retrieval of data expressed in RDF

SPARQL

MAS – Web Sémantique

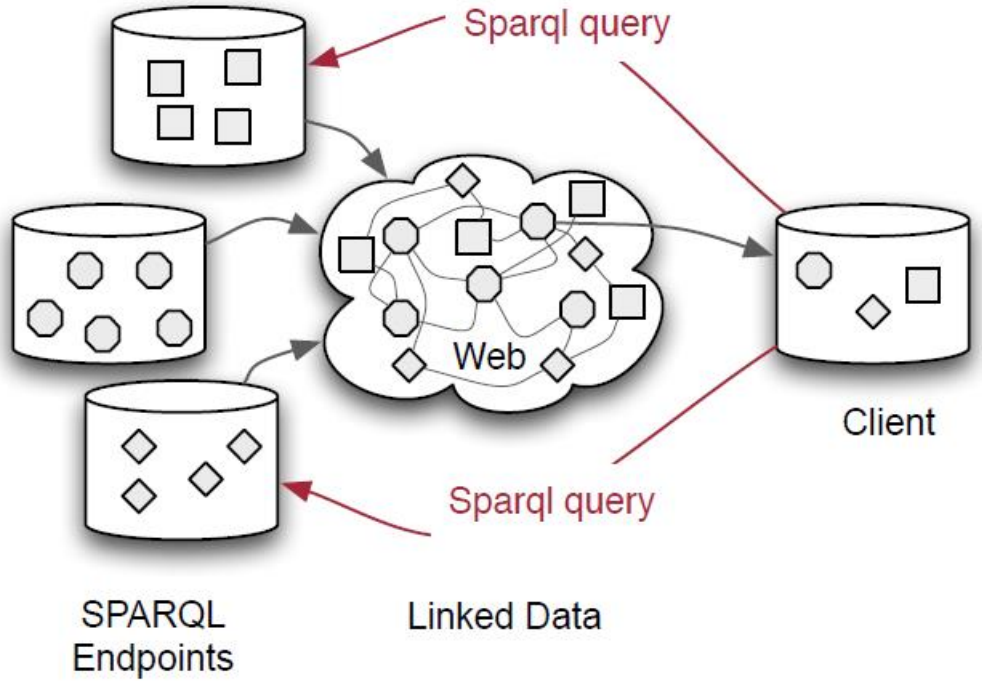
SPARQL

- A query language and protocol for accessing
- RDF data on the Web

```
SELECT DISTINCT ?x
        WHERE {
                ?x
                skos:subject
                <http:dbpedia.org/resource/Category:1980s_horror_films>
        }
LIMIT 10
```

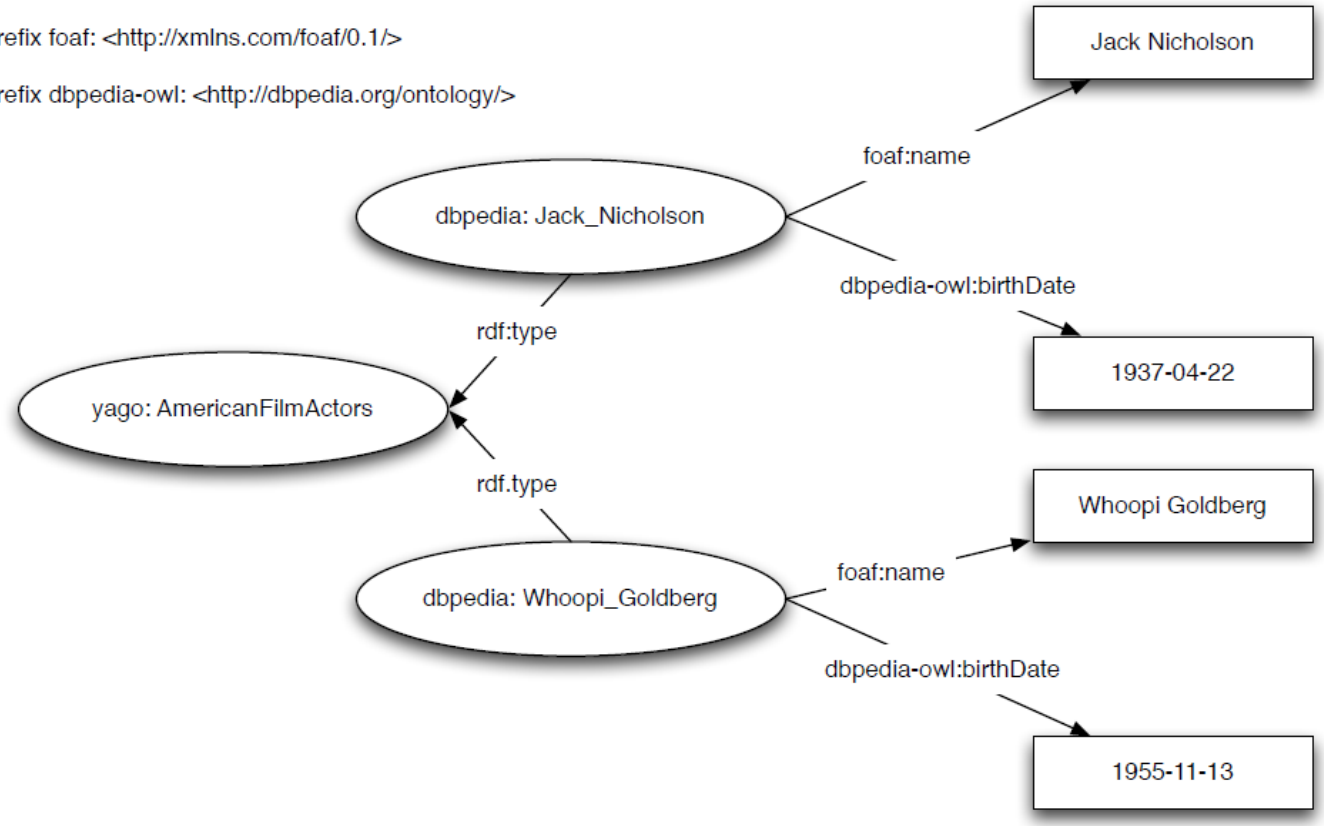
What is SPARQL?

- SPARQL is a query language for accessing RDF data
- SPARQL is a protocol that defines how queries and results can be transported over a network



Example Dataset

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix dbpedia: <http://dbpedia.org/resource/>
@prefix yago: <http://dbpedia.org/class/yago/>
@prefix foaf: <http://xmlns.com/foaf/0.1/>
@prefix dbpedia-owl: <http://dbpedia.org/ontology/>
```



A Simple SPARQL Query

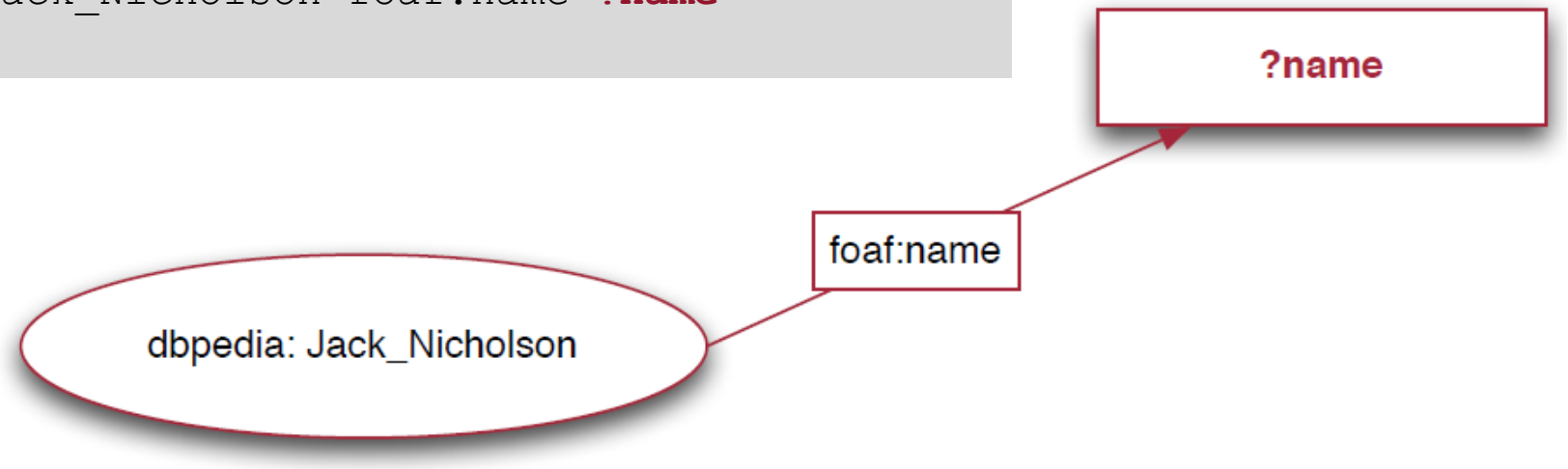
```
SELECT ?name
WHERE {
        dbpedia:Jack_Nicholson foaf:name ?name
}
```

name
"Jack Nicholson"

Simple Query Illustrated

- In SPARQL we formulate triple / graph patterns

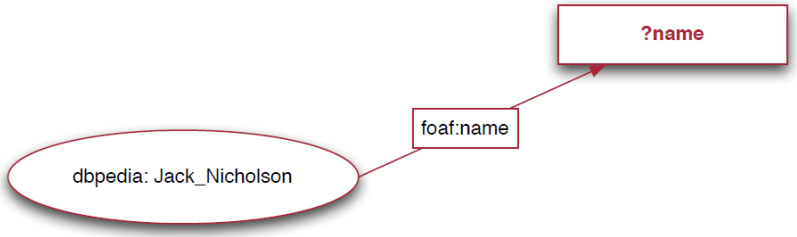
```
SELECT ?name
WHERE {
        dbpedia:Jack_Nicholson foaf:name ?name
}
```



Simple Query Illustrated

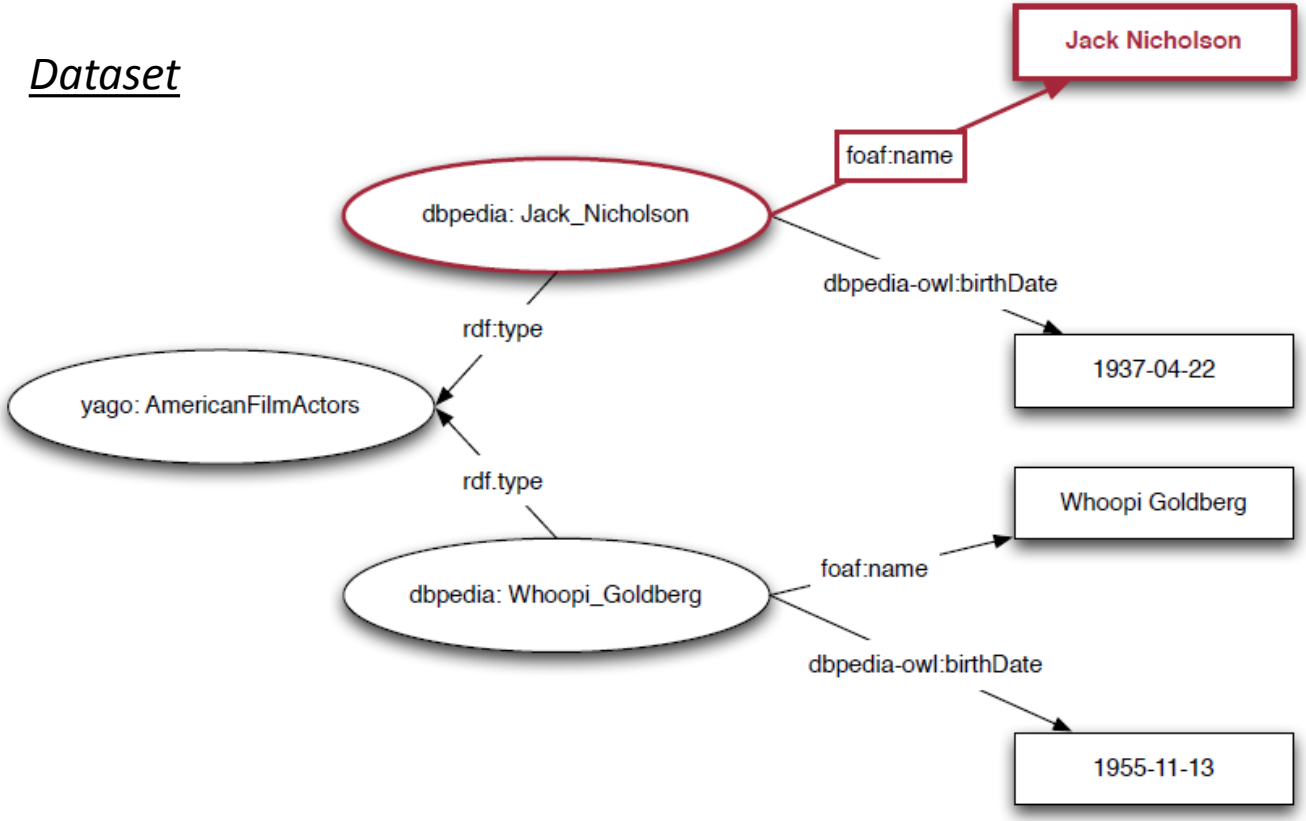
- Patterns are matched against the dataset

Researched Pattern




```
SELECT ?name
WHERE {
   dbpedia:Jack_Nicholson foaf:name ?name
}
```

Dataset



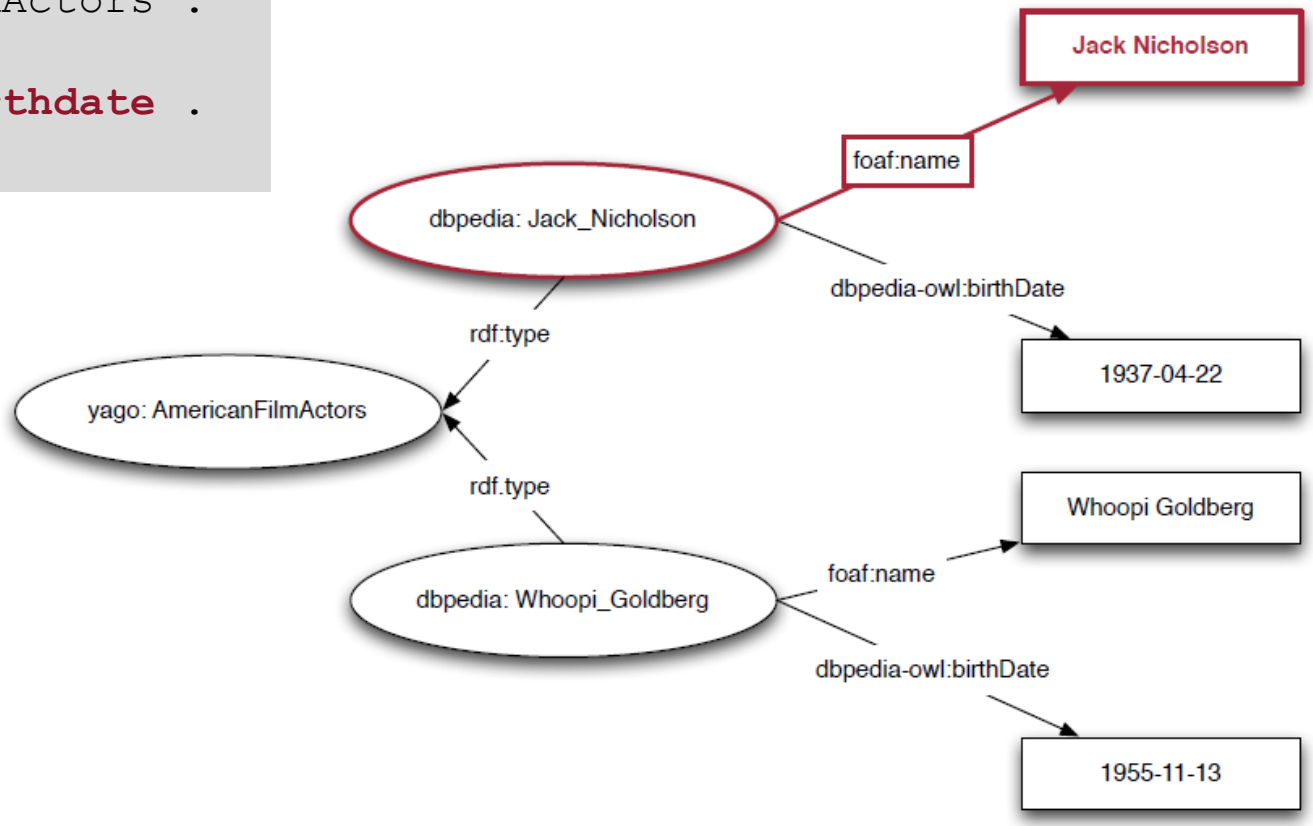
Exercise: draw the graph pattern

```
SELECT ?name ?birthdate
WHERE {
        ?x rdf:type yago:AmericanFilmActors .
        ?x foaf:name ?name .
        ?x dbpedia-owl:birthDate ?birthdate .
}
```

Querying Multiple Values

```
SELECT ?name ?birthdate
WHERE {
        ?x rdf:type yago:AmericanFilmActors .
        ?x foaf:name ?name .
        ?x dbpedia-owl:birthDate ?birthdate .
}
```

name	birthdate
"Jack Nicholson"	1937-04-22
"Whoopi Goldberg"	1955-11-13



Prefixes (namepaces) in SPARQL queries

- The DBPedia SPARQL Web interface knows about prefix/namespace mappings
- But in general SPARQL endpoints don't

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

SELECT ?name
WHERE {
        dbpedia:Jack_Nicholson foaf:name ?name
}
```

SPARQL Filters (1)

- ...restrict the solutions of a graph pattern match according to a given expression
- ...eliminate solutions that, when substituted into the expression, result in Boolean false

```
SELECT ?name
WHERE {
        ?x foaf:name ?name .
        FILTER regex(?name, "Nicholson")
}
```

name
"Jack Nicholson"

SPARQL Filters (2)

```
SELECT ?name ?birthdate
WHERE {
        ?x rdf:type yago:AmericanFilmActors .
        ?x foaf:name ?name .
        ?x dbpedia-owl:birthDate ?birthdate .

        FILTER(?birthdate > "1950-01-01T00:00:00Z"^^xsd:dateTime)
}
```

name	birthdate
"Whoopi Goldberg"	1955-11-13

SPARQL Filters (3)

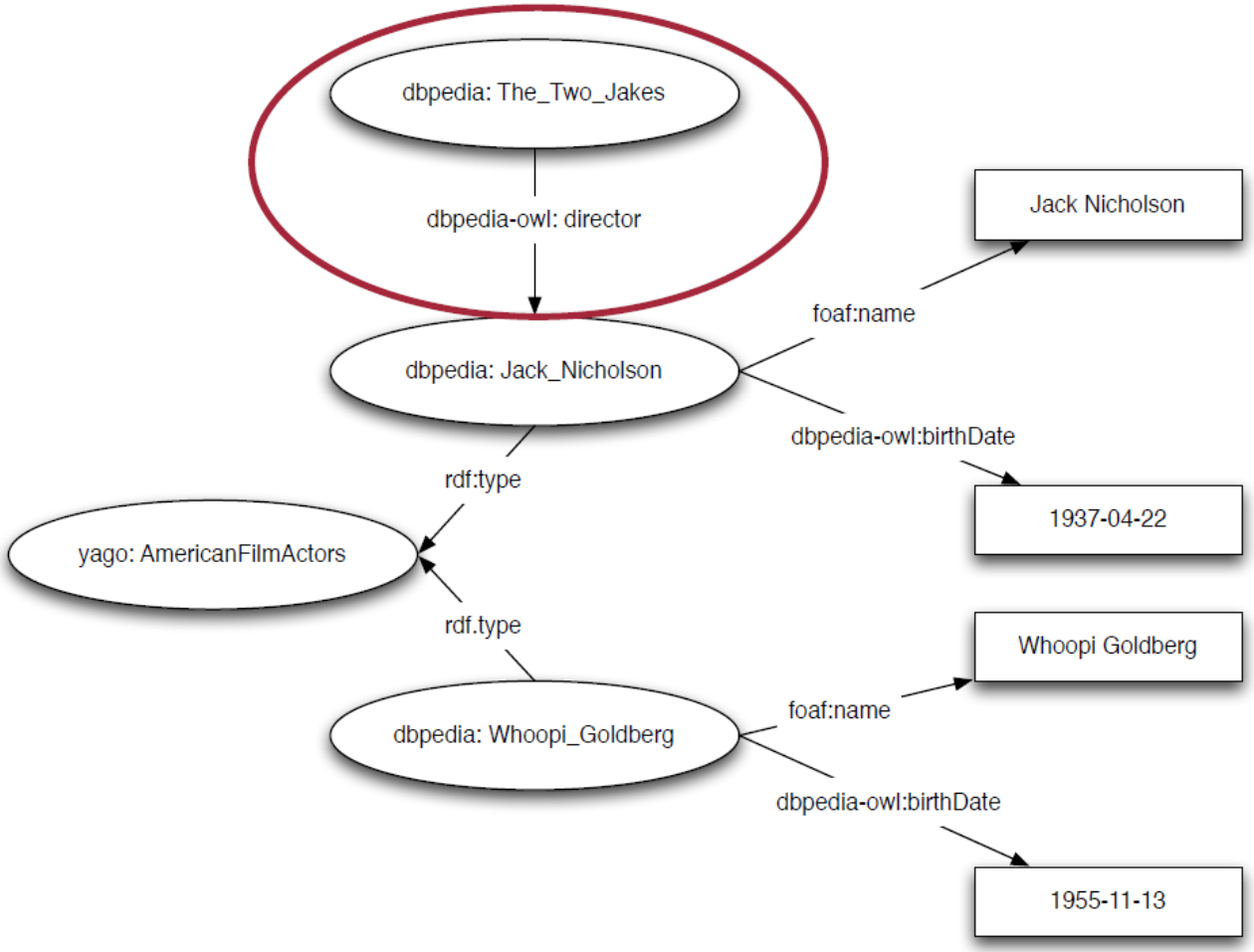
- SPARQL offers lots of FILTERing possibilities
- Boolean operators ! || &&
- Comparison operators = != > < >= <=
- Arithmetic operators * / - +
- RDF element operators bound(), isURI(), LANG(), STR()

http://www.w3.org/TR/rdf-sparql-query/#tests

Optional Graph Patterns

- In a standard SPARQL query the entire graph pattern must match in order to retrieve a result
- But in an open world we cannot always assume complete structures
  - e.g. DBpedia: not all actors have birthdays
- SPARQL allows to formulate queries that include information in the solution if it is available, but does not reject the solution if parts are missing

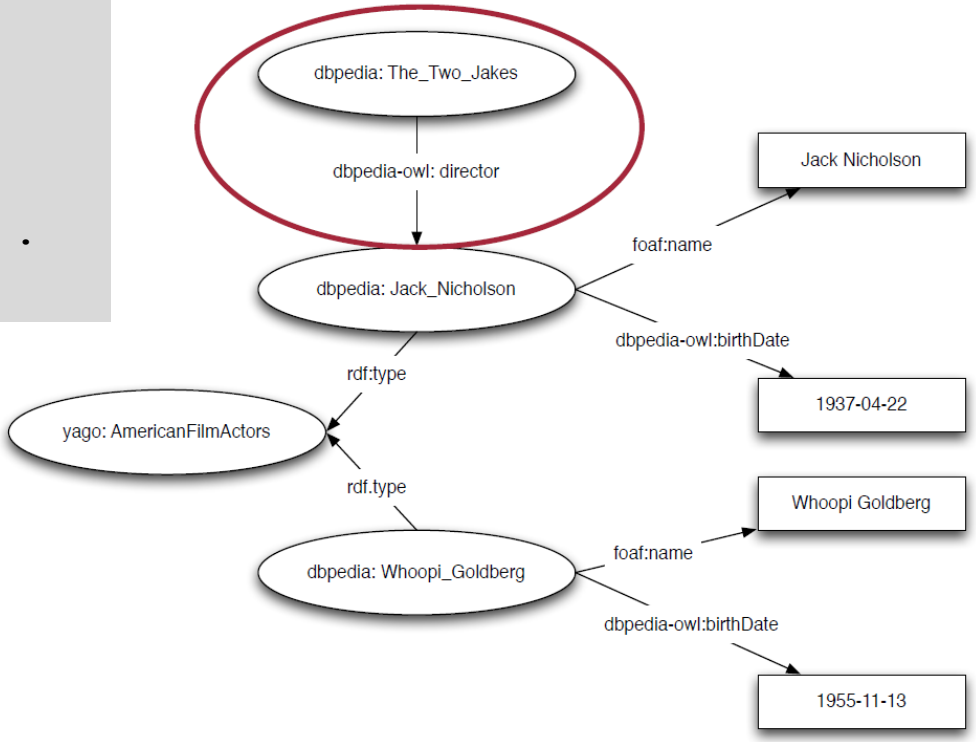
Optional Graph Patterns



Optional Graph Patterns

```
SELECT ?name ?directed_movie
WHERE {
        ?x rdf:type yago:AmericanFilmActors .
        ?x foaf:name ?name .
        ?directed_movie dbpedia-owl:director ?x .
}
```

name	directed_movie
"Jack Nicholson"	dbpedia:The_Two_Jakes

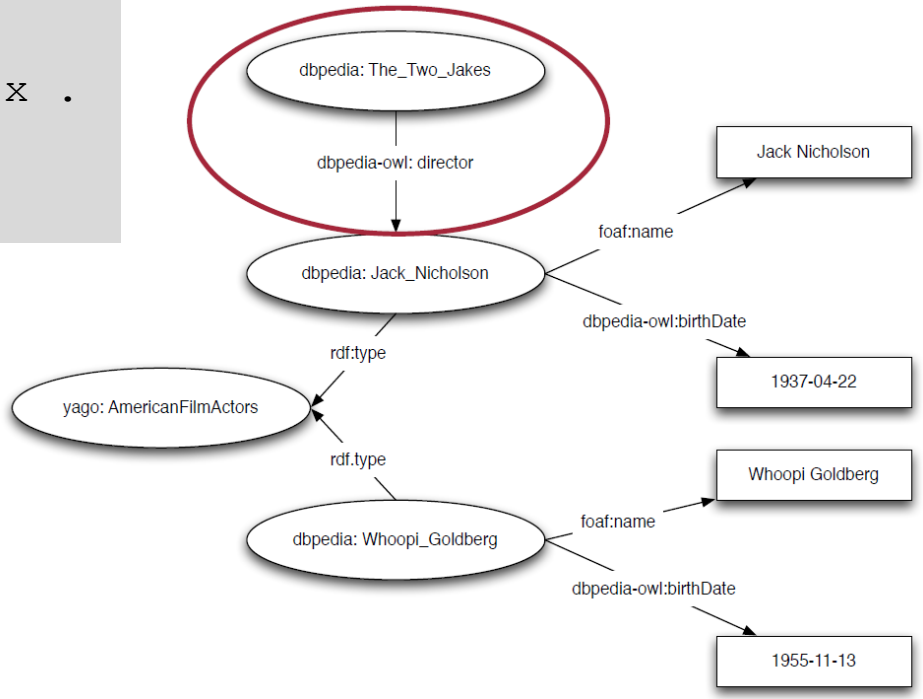


We lost Whoopi. Why?

Optional Graph Patterns

```
SELECT ?name ?directed_movie
WHERE {
        ?x rdf:type yago:AmericanFilmActors .
        ?x foaf:name ?name .
        OPTIONAL {
                ?directed_movie dbpedia-owl:director ?x .
        }
}
```

name	directed_movie
"Jack Nicholson"	dbpedia:The_Two_Jakes
"Whoopi Goldberg"	



SPARQL UNION

- Sometimes you want to express “or” in a graph pattern
  - The graph should match this OR that pattern
  - With the UNION keyword we can define alternative matching graph patterns

SPARQL UNION

```
SELECT ?name ?directed_movie
WHERE {
        {?x rdf:type yago:AmericanFilmActors }

        UNION

        {?x rdf:type yago:GermanFilmActors }

        ?x foaf:name ?name .

        OPTIONAL {
                ?directed_movie dbpedia-owl:director ?x .
        }
}
```

SPARQL Modifiers

- The results returned by a query are by default unordered SPARQL defines the following solution modifiers:
  - ORDER BY - reorder the solution sequence
  - DISTINCT - avoid duplicate solutions
  - OFFSET - start after a certain number of solutions
  - LIMIT - limit the output to a number of solutions

SPARQL Modifiers

```
SELECT ?name
WHERE {
        ?x a foaf:Person .
        ?x foaf:name ?name .
}
LIMIT 100
```

```
SELECT ?person (count(DISTINCT ?spouse) as ?spouses)
where {
        ?person a yago:AmericanFilmActors .
        ?person dbpprop:spouse ?spouse .
}
ORDER BY DESC(?spouses)
LIMIT 100
```

SPARQL Protocol for RDF

- SPARQL also defines how queries and results can be transported over a network:
  - Bindings for HTTP and SOAP
  - http://www.w3.org/TR/rdf-sparql-protocol/

SPARQL Protocol - HTTP

- Queries are sent to an endpoint using
- HTTP GET (default)
- HTTP POST (if encoded query string exceed limits)
- Results are returned either as
  - SPARQL Results Document (SELECT and ASK)
  - Serialized RDF Graph (CONSTRUCT and DESCRIBE)

SPARQL Protocol - Example

HTTP GET

```
SELECT ?name
WHERE {
    dbpedia:Jack_Nicholson foaf:name ?name
}
```

URI-encoded query string *(EncodedQuery)*

```
SELECT%20?name%0AWHERE%20%7B%0A%20%20%20%20dbpedi
a:Jack_Nicholson%20foaf:name%20?name%0A%7D
```

HTTP GET

```
GET /sparql/?query=EncodedQuery HTTP/1.1
Host: dbpedia.org
```

(OR Using CURL)

```
curl -v http://dbpedia.org/sparql/?
query=SELECT%20?name%0AWHERE%20%7B%0A
%20%20%20%20dbpedicholson%20foaf:name
%20?name%0A%7D
```

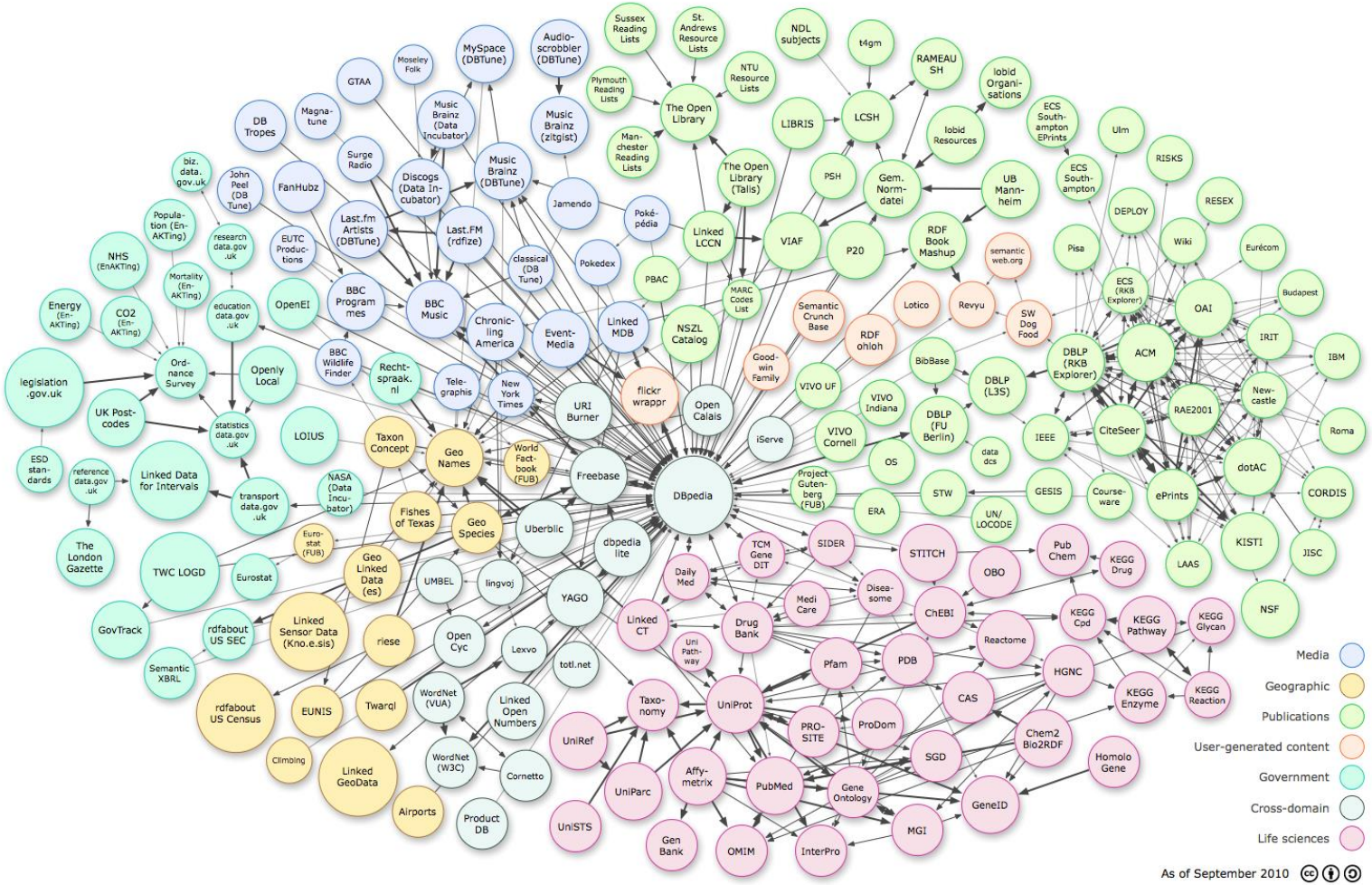
Result

```
<sparql xmlns="...">
<head>
   <variable name="name"/>
</head>
<results distinct="false" ordered="true">
   <result>
   <binding name="name">
      <literal xml:lang="en">
         Jack Nicholson
      </literal>
   </binding>
</result>
</results>
```

DBPedia

MAS – Web Sémantique

LinkedData



DBPedia

- DBpedia is a crowd-sourced community effort to extract structured information from Wikipedia and make this information available on the Web.
- DBpedia allows you to ask sophisticated queries against Wikipedia, and to link the different data sets on the Web to Wikipedia data. We hope that this work will make it easier for the huge amount of information in Wikipedia to be used in some new interesting ways. Furthermore, it might inspire new mechanisms for navigating, linking, and improving the encyclopedia itself.

Demo

- Resources
  - City
    - http://dbpedia.org/page/Fribourg
  - Country
    - http://dbpedia.org/page/Switzerland
  - Person
    - http://dbpedia.org/page/Bill_Gates
- Endpoint
  - http://dbpedia.org/sparql

References

- For smaller datasets
  - TDB works fine
- For larger RDF datasets, you should consider
  - OpenLink Virtuoso (http://virtuoso.openlinksw.com/)
  - Store (http://4store.org/)