



SPARQL

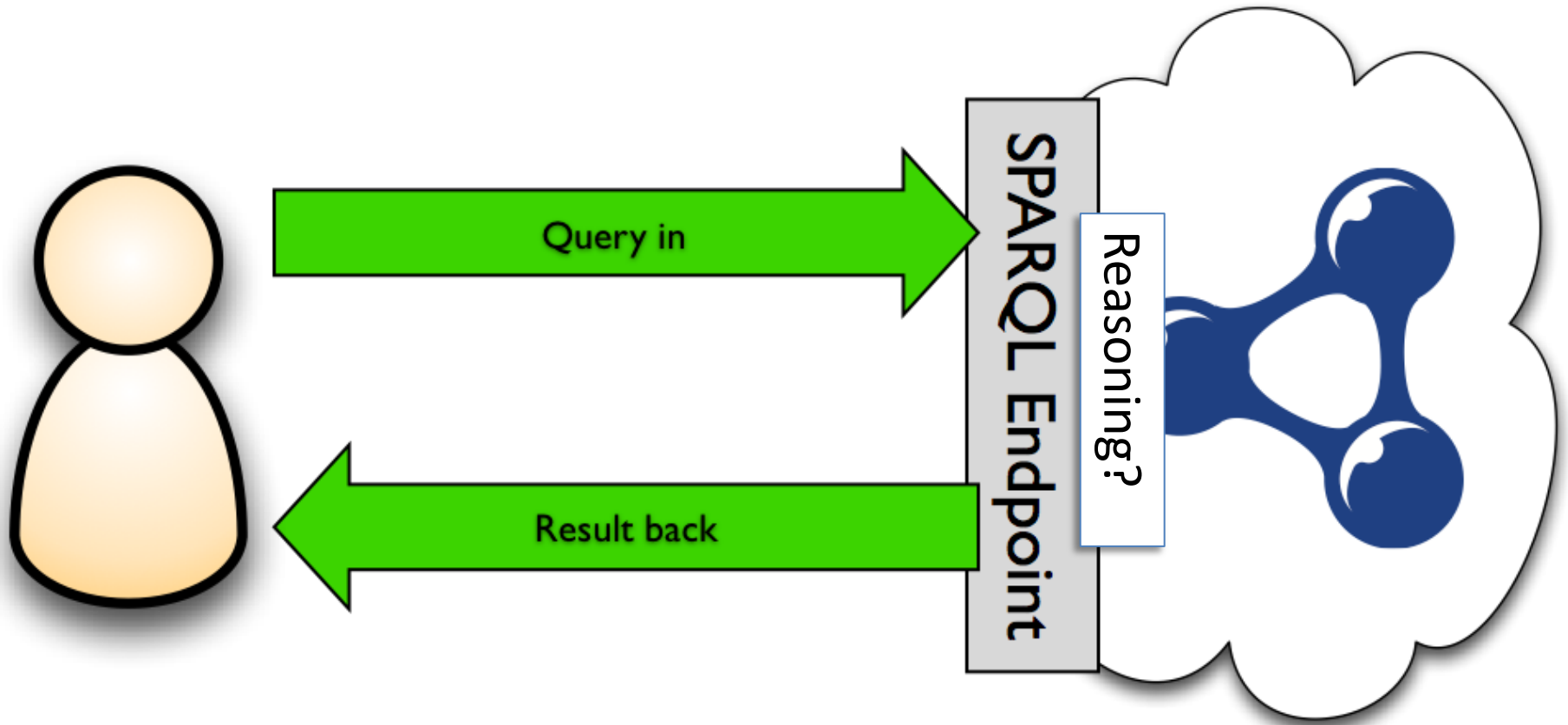
A Query-language for the Web of Data

SPARQL – Querying the Web of Data

- query language for RDF graphs (i.e., linked data)
- extract specific information out of a dataset (or several datasets)
- "The SQL for the Web of Data"



SPARQL Endpoint



```
PREFIX mo: <http://purl.org/ontology/mo/>  
PREFIX dce: <http://purl.org/dc/elements/1.1/>  
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
SELECT DISTINCT ?album ?title ?release_date
```

```
WHERE {
```

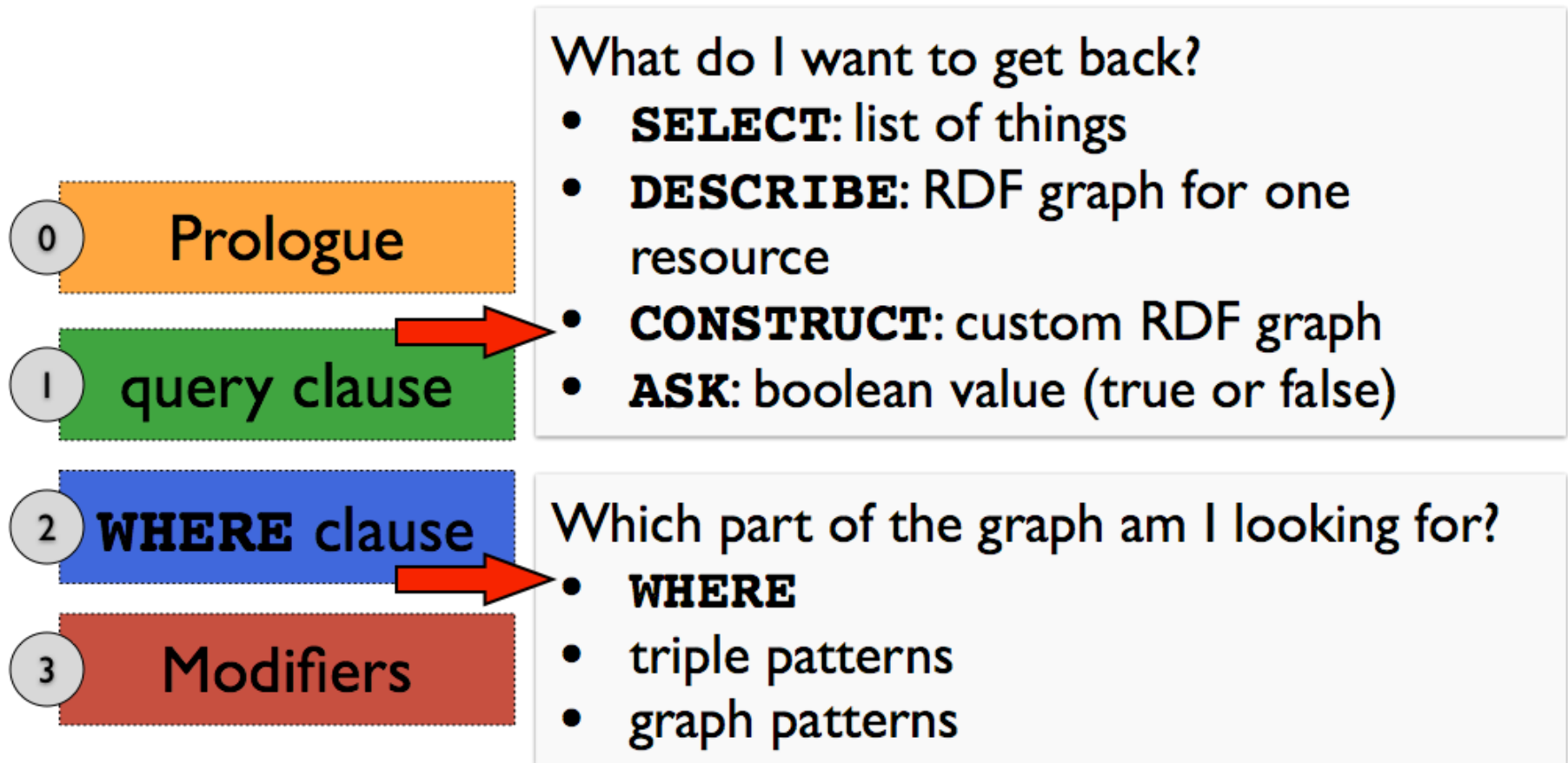
```
    ?album a mo:Record ;  
           dce:date ?release_date ;  
           dce:title ?title .
```

```
    FILTER (year(?release_date) = 2007 &&  
            month(?release_date) = 7)
```

```
}
```

```
ORDER BY ?release_date
```

SPARQL in a Nutshell



```
PREFIX mo: <http://purl.org/ontology/mo/>
```

```
PREFIX dce: <http://purl.org/dc/elements/1.1/>
```

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
SELECT DISTINCT ?album ?title ?release_date
```

```
WHERE {
```

```
  ?album a mo:Record ;  
         dce:date ?release_date ;  
         dce:title ?title .
```

 graph pattern

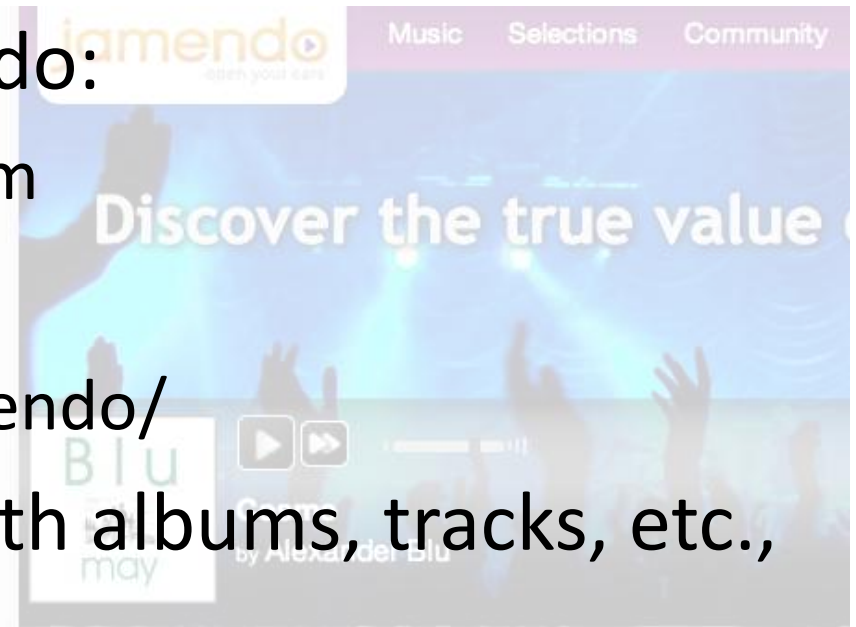
```
FILTER (year(?release_date) = 2007 &&  
        month(?release_date) = 7)
```

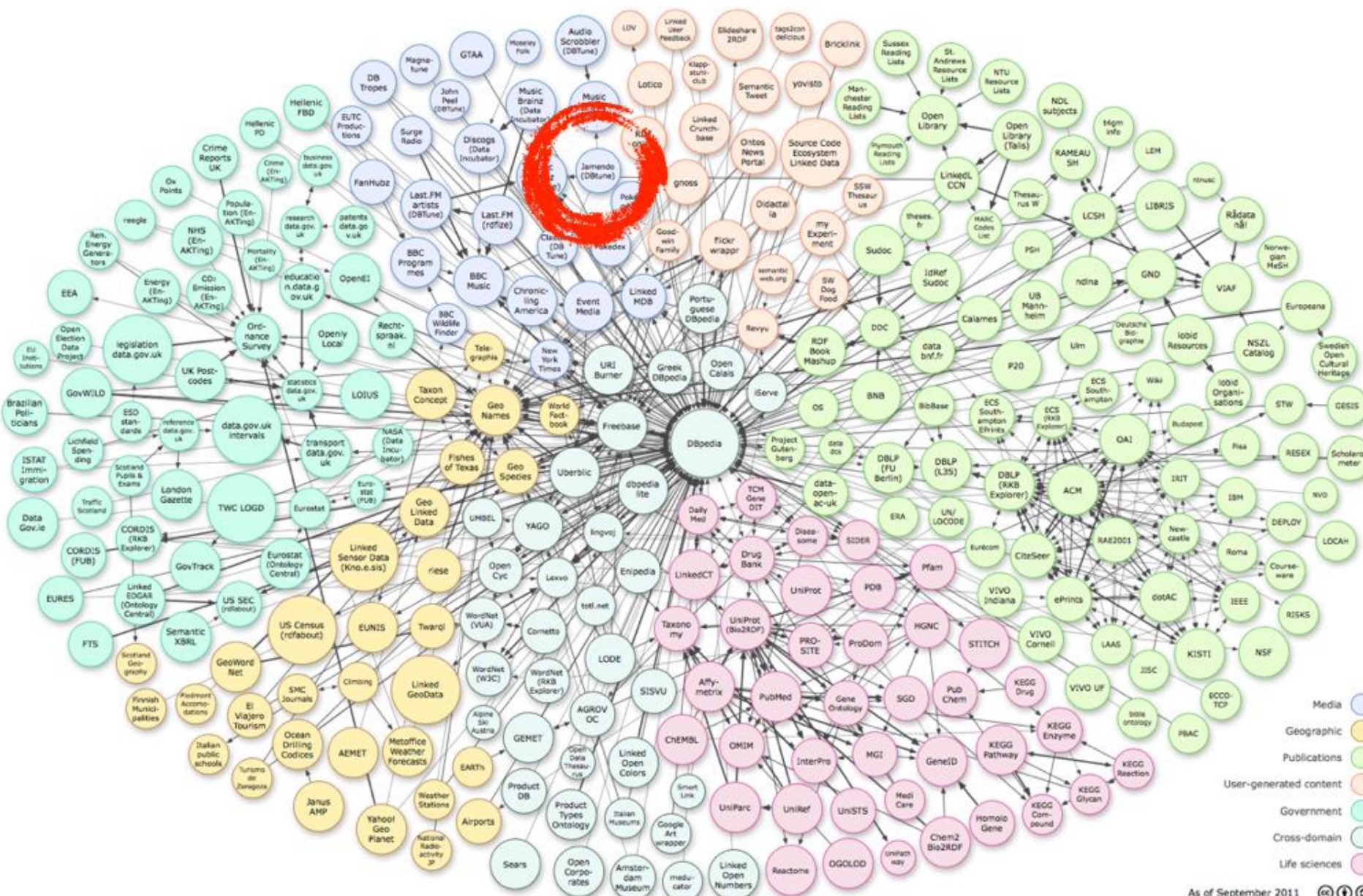
```
}
```

```
ORDER BY ?release_date
```

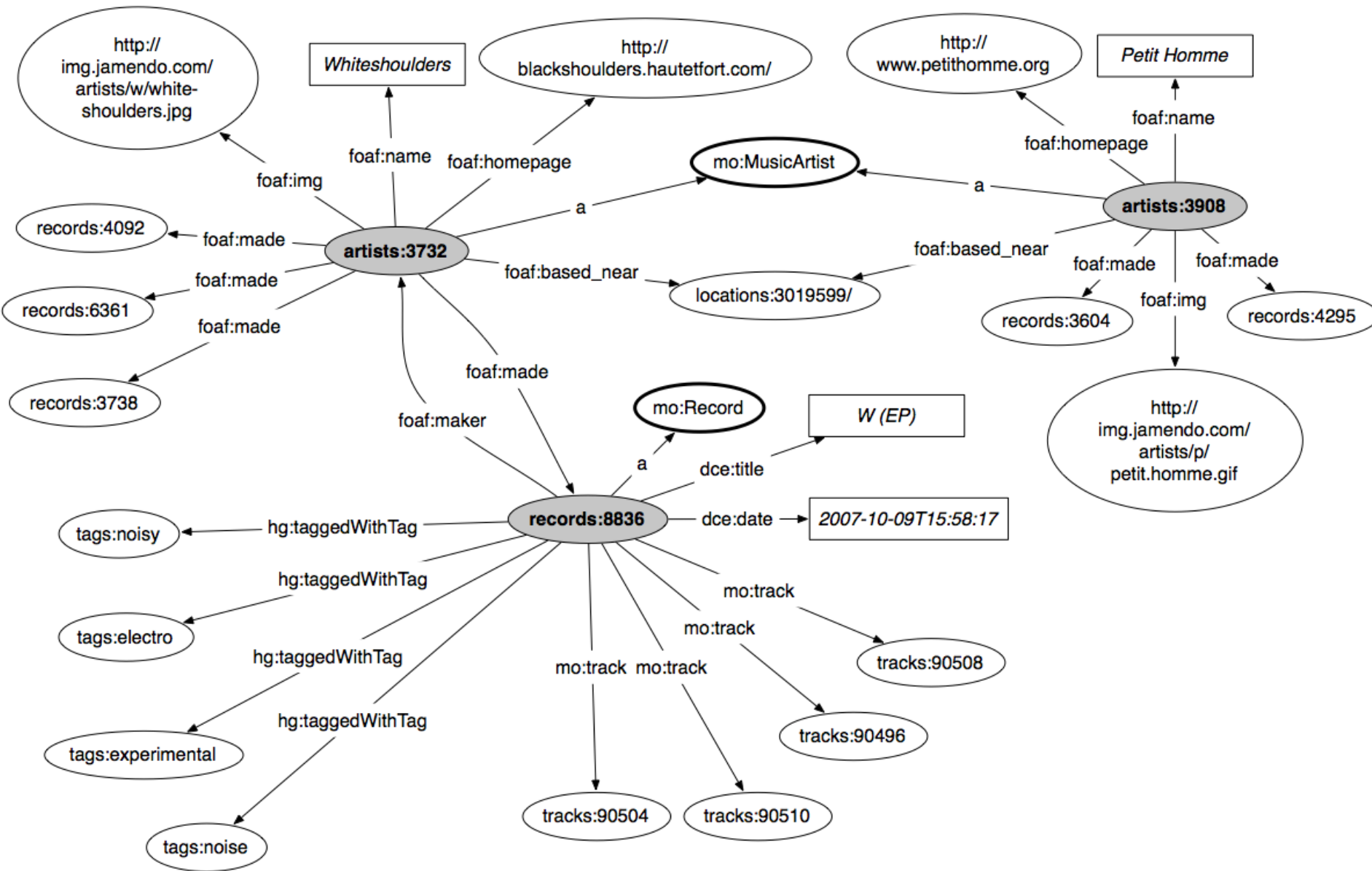
Example Data

- from music site Jamendo:
 - <http://www.jamendo.com>
- linked data at DBTune:
 - <http://dbtune.org/jamendo/>
- thousands of artists with albums, tracks, etc., all creative commons
- > 1,000,000 triples





- Media ●
- Geographic ●
- Publications ●
- User-generated content ●
- Government ●
- Cross-domain ●
- Life sciences ●



```
<artist/3908> a mo:MusicArtist .
<artist/3908> foaf:based_near <http://sws.geonames.org/3019599/> .
<artist/3908> foaf:homepage <http://www.petithomme.org> .
<artist/3908> foaf:img <http://img.jamendo.com/artists/p/petit.homme.gif> .
<artist/3908> foaf:made <record/3604> .
<artist/3908> foaf:made <record/4295> .
<artist/3908> foaf:name "Petit Homme"^^xsd:string .

<artist/3732> a mo:MusicArtist .
<artist/3732> foaf:based_near <http://sws.geonames.org/3019599/> .
<artist/3732> foaf:homepage <http://blackshoulders.hautetfort.com/> .
<artist/3732> foaf:img <http://img.jamendo.com/artists/w/white-shoulders.jpg> .
<artist/3732> foaf:made <record/3738> .
<artist/3732> foaf:made <record/4092> .
<artist/3732> foaf:made <record/6361> .
<artist/3732> foaf:made <record/8836> .
<artist/3732> foaf:name "Whiteshoulders"^^xsd:string .

<record/8836> dce:date "2007-10-09T15:58:17"^^xsd:dateTime .
<record/8836> dce:title "W (EP)"^^xsd:string .
<record/8836> mo:track <track/90496> .
<record/8836> mo:track <track/90504> .
<record/8836> mo:track <track/90508> .
<record/8836> mo:track <track/90510> .
<record/8836> hg:taggedWithTag tags:electro .
<record/8836> hg:taggedWithTag tags:experimental .
<record/8836> hg:taggedWithTag tags:noise .
<record/8836> hg:taggedWithTag tags:noisy .
<record/8836> a mo:Record .
<record/8836> foaf:maker <artist/3732> .
```

Triple and Graph Patterns

```
WHERE {  
  ?album a mo:Record ;  
    dce:date ?release_date ;  
    dce:title ?title .  
  FILTER (year(?release_date) = 2007  
&&  
          month(?release_date) = 7)  
}
```

One Variable

<artist/3732> foaf:made ?record .



```
<artist/3908> a mo:MusicArtist .
<artist/3908> foaf:based_near <http://sws.geonames.org/3019599/> .
<artist/3908> foaf:homepage <http://www.petithomme.org> .
<artist/3908> foaf:img <http://img.jamendo.com/artists/p/petit.homme.gif> .
<artist/3908> foaf:made <record/3604> .
<artist/3908> foaf:made <record/4295> .
<artist/3908> foaf:name "Petit Homme"^^xsd:string .

<artist/3732> a mo:MusicArtist .
<artist/3732> foaf:based_near <http://sws.geonames.org/3019599/> .
<artist/3732> foaf:homepage <http://blackshoulders.hautetfort.com/> .
<artist/3732> foaf:img <http://img.jamendo.com/artists/w/white-shoulders.jpg> .
<artist/3732> foaf:made <record/3738> .
<artist/3732> foaf:made <record/4092> .
<artist/3732> foaf:made <record/6361> .
<artist/3732> foaf:made <record/8836> .
<artist/3732> foaf:name "Whiteshoulders"^^xsd:string .

<record/8836> dce:date "2007-10-09T15:58:17"^^xsd:dateTime .
<record/8836> dce:title "W (EP)"^^xsd:string .
<record/8836> mo:track <track/90496> .
<record/8836> mo:track <track/90504> .
<record/8836> mo:track <track/90508> .
<record/8836> mo:track <track/90510> .
<record/8836> hg:taggedWithTag tags:electro .
<record/8836> hg:taggedWithTag tags:experimental .
<record/8836> hg:taggedWithTag tags:noise .
<record/8836> hg:taggedWithTag tags:noisy .
<record/8836> a mo:Record .
<record/8836> foaf:maker <artist/3732> .
```

Two Variables

?artist foaf:made ?record .

```
<artist/3908> a mo:MusicArtist .
<artist/3908> foaf:based_near <http://sws.geonames.org/3019599/> .
<artist/3908> foaf:homepage <http://www.petithomme.org> .
<artist/3908> foaf:img <http://img.jamendo.com/artists/p/petit.homme.gif> .
<artist/3908> foaf:made <record/3604> .
<artist/3908> foaf:made <record/4295> .
<artist/3908> foaf:name "Petit Homme"^^xsd:string .

<artist/3732> a mo:MusicArtist .
<artist/3732> foaf:based_near <http://sws.geonames.org/3019599/> .
<artist/3732> foaf:homepage <http://blackshoulders.hautetfort.com/> .
<artist/3732> foaf:img <http://img.jamendo.com/artists/w/white-shoulders.jpg> .
<artist/3732> foaf:made <record/3738> .
<artist/3732> foaf:made <record/4092> .
<artist/3732> foaf:made <record/6361> .
<artist/3732> foaf:made <record/8836> .
<artist/3732> foaf:name "Whiteshoulders"^^xsd:string .

<record/8836> dce:date "2007-10-09T15:58:17"^^xsd:dateTime .
<record/8836> dce:title "W (EP)"^^xsd:string .
<record/8836> mo:track <track/90496> .
<record/8836> mo:track <track/90504> .
<record/8836> mo:track <track/90508> .
<record/8836> mo:track <track/90510> .
<record/8836> hg:taggedWithTag tags:electro .
<record/8836> hg:taggedWithTag tags:experimental .
<record/8836> hg:taggedWithTag tags:noise .
<record/8836> hg:taggedWithTag tags:noisy .
<record/8836> a mo:Record .
<record/8836> foaf:maker <artist/3732> .
```

Three Variables

?subject ?predicate ?object .

```
<artist/3908> a mo:MusicArtist .
<artist/3908> foaf:based_near <http://sws.geonames.org/3019599/> .
<artist/3908> foaf:homepage <http://www.petithomme.org> .
<artist/3908> foaf:img <http://img.jamendo.com/artists/p/petit.homme.gif> .
<artist/3908> foaf:made <record/3604> .
<artist/3908> foaf:made <record/4295> .
<artist/3908> foaf:name "Petit Homme"^^xsd:string .

<artist/3732> a mo:MusicArtist .
<artist/3732> foaf:based_near <http://sws.geonames.org/3019599/> .
<artist/3732> foaf:homepage <http://blackshoulders.hautetfort.com/> .
<artist/3732> foaf:img <http://img.jamendo.com/artists/w/white-shoulders.jpg> .
<artist/3732> foaf:made <record/3738> .
<artist/3732> foaf:made <record/4092> .
<artist/3732> foaf:made <record/6361> .
<artist/3732> foaf:made <record/8836> .
<artist/3732> foaf:name "Whiteshoulders"^^xsd:string .

<record/8836> dce:date "2007-10-09T15:58:17"^^xsd:dateTime .
<record/8836> dce:title "W (EP)"^^xsd:string .
<record/8836> mo:track <track/90496> .
<record/8836> mo:track <track/90504> .
<record/8836> mo:track <track/90508> .
<record/8836> mo:track <track/90510> .
<record/8836> hg:taggedWithTag tags:electro .
<record/8836> hg:taggedWithTag tags:experimental .
<record/8836> hg:taggedWithTag tags:noise .
<record/8836> hg:taggedWithTag tags:noisy .
<record/8836> a mo:Record .
<record/8836> foaf:maker <artist/3732> .
```

A full Graph Pattern

```
?artist a mo:MusicArtist ;  
    foaf:name ?artist_name ;  
    foaf:homepage ?homepage .
```

<artist/3908> a mo:MusicArtist .

<artist/3908> foaf:based_near <<http://sws.geonames.org/3019599/>> .

<artist/3908> foaf:homepage <<http://www.petithomme.org>> .

<artist/3908> foaf:img <<http://img.jamendo.com/artists/p/petit.homme.gif>> .

<artist/3908> foaf:made <record/3604> .

<artist/3908> foaf:made <record/4295> .

<artist/3908> foaf:name "Petit Homme"^^xsd:string .

<artist/3732> a mo:MusicArtist .

<artist/3732> foaf:based_near <<http://sws.geonames.org/3019599/>> .

<artist/3732> foaf:homepage <<http://blackshoulders.hautetfort.com/>> .

<artist/3732> foaf:img <<http://img.jamendo.com/artists/w/white-shoulders.jpg>> .

<artist/3732> foaf:made <record/3738> .

<artist/3732> foaf:made <record/4092> .

<artist/3732> foaf:made <record/6361> .

<artist/3732> foaf:made <record/8836> .

<artist/3732> foaf:name "Whiteshoulders"^^xsd:string .

<record/8836> dce:date "2007-10-09T15:58:17"^^xsd:dateTime .

<record/8836> dce:title "W (EP)"^^xsd:string .

<record/8836> mo:track <track/90496> .

<record/8836> mo:track <track/90504> .

<record/8836> mo:track <track/90508> .

<record/8836> mo:track <track/90510> .

<record/8836> hg:taggedWithTag tags:electro .

<record/8836> hg:taggedWithTag tags:experimental .

<record/8836> hg:taggedWithTag tags:noise .

<record/8836> hg:taggedWithTag tags:noisy .

<record/8836> a mo:Record .

<record/8836> foaf:maker <artist/3732> .

Example Scenario

- Get all albums for a particular month (July 2007)
- explore datasets and build final query bit by bit
- learn about different features of SPARQL along the way
- cannot cover every aspect of SPARQL! E.g., property paths are missing.
- Check <http://piratepad.net/csws-sparql>

Example Server

- Cliopatria instance with Jamendo dataset pre-loaded:
- <http://eculture.cs.vu.nl:1235/home>
- User: test
- Password: tester

01: List all Classes

```
SELECT DISTINCT ?class
WHERE {
    ?s a ?class .
}
```

find all types of things in the dataset (classes)

class
http://purl.org/ontology/mo/MusicArtist
http://www.w3.org/2006/time#Interval
http://purl.org/ontology/mo/Lyrics
foaf:Document
http://purl.org/ontology/mo/Record
http://purl.org/ontology/mo/Signal
http://www.holygoat.co.uk/owl/redwood/0.1/tags/Tag
http://purl.org/ontology/mo/Track
http://purl.org/ontology/mo/Torrent
http://purl.org/ontology/mo/ED2K
http://purl.org/ontology/mo/Playlist

02: Properties of Artists

```
PREFIX mo: <http://purl.org/ontology/mo/>
```

```
SELECT DISTINCT ?pred
```

```
WHERE {
```

```
    ?artist a mo:MusicArtist ;
```

```
    ?pred ?obj .
```

```
}
```

all properties for the mo:MusicArtist class

pred
rdf:type
foaf:based_near
foaf:name
foaf:made
foaf:homepage
foaf:img
http://purl.org/ontology/mo/biography

03: Properties of Artists v02

```
PREFIX mo: <http://purl.org/ontology/mo/>
```

```
SELECT DISTINCT ?pred ?class
```

```
WHERE {
```

```
    ?artist a mo:MusicArtist ;
```

```
        ?pred ?obj .
```

```
    ?obj a ?class .
```

```
}
```

properties with objects

pred	class
foaf:made	http://purl.org/ontology/mo/Record

only one result?

04: Properties of Artists v03

```
PREFIX mo: <http://purl.org/ontology/mo/>
```

```
SELECT DISTINCT ?pred ?class
```

```
WHERE {
```

```
  ?artist a mo:MusicArtist ;  
    ?pred ?obj .
```

```
  OPTIONAL {
```

```
    ?obj a ?class .
```

```
  }
```

```
}
```

OPTIONAL statement

pred	class
rdf:type	rdfs:Resource
foaf:based_near	rdfs:Resource
foaf:name	rdfs:Literal
foaf:made	<http://purl.org/ontology/mo/Record>
foaf:homepage	rdfs:Resource
foaf:img	rdfs:Resource
<http://purl.org/ontology/mo/biography>	rdfs:Literal

05: Properties of Records

```
PREFIX mo: <http://purl.org/ontology/mo/>
```

```
SELECT DISTINCT ?pred ?class
```

```
WHERE {
```

```
  ?artist a mo:Record;
```

```
  ?pred ?obj .
```

```
  OPTIONAL {
```

```
    ?obj a ?class .
```

```
  }
```

```
}
```

pred	class
dc:date	rdfs:Literal
dc:description	rdfs:Literal
dc:title	rdfs:Literal
<http://purl.org/ontology/mo/available as>	<http://purl.org/ontology/mo/Torrent>
<http://purl.org/ontology/mo/available as>	<http://purl.org/ontology/mo/ED2K>
<http://purl.org/ontology/mo/available as>	<http://purl.org/ontology/mo/Playlist>
<http://purl.org/ontology/mo/image>	rdfs:Resource
<http://purl.org/ontology/mo/track>	<http://purl.org/ontology/mo/Track>
<http://www.holygoat.co.uk/owl/redwood/0.1/tags/taggedWithTag>	<http://www.holygoat.co.uk/owl/redwood/0.1/tags/Tag>
rdf:type	rdfs:Resource
foaf:maker	<http://purl.org/ontology/mo/MusicArtist>

06: Properties of Records v02

```
PREFIX mo: <http://purl.org/ontology/mo/>
```

```
SELECT DISTINCT ?pred ?class
```

```
WHERE {
```

```
  ?artist a mo:Record;
```

```
  ?pred ?obj .
```

```
  OPTIONAL {
```

```
    ?obj a ?class .
```

```
  }
```

```
} ORDER BY ASC(?pred) DESC(?class)
```

ORDER BY statement

pred	class
dc:date	rdfs:Literal
dc:description	rdfs:Literal
dc:title	rdfs:Literal
<http://purl.org/ontology/mo/available as>	<http://purl.org/ontology/mo/Torrent>
<http://purl.org/ontology/mo/available as>	<http://purl.org/ontology/mo/Playlist>
<http://purl.org/ontology/mo/available as>	<http://purl.org/ontology/mo/ED2K>
<http://purl.org/ontology/mo/image>	rdfs:Resource
<http://purl.org/ontology/mo/track>	<http://purl.org/ontology/mo/Track>
<http://www.holygoat.co.uk/owl/redwood/0.1/tags/taggedWithTag>	<http://www.holygoat.co.uk/owl/redwood/0.1/tags/Tag>
rdf:type	rdfs:Resource
foaf:maker	<http://purl.org/ontology/mo/MusicArtist>

Problem: Incorrect Data

- date/time related functions in SPARQL expect dates in `xsd:dateTime` format:

```
"2007-06-15T17:13:58"^^xsd:dateTime
```

- however, source data has dates as plain literals with wrong format:

```
"2007-06-15 17:13:58"
```

07: Fixing Data with CONSTRUCT

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
PREFIX dce: <http://purl.org/dc/elements/1.1/>
```

```
CONSTRUCT {  
    ?s2 dce:date ?date_fixed .  
}
```

```
WHERE {  
    ?s2 dce:date ?date .
```

```
@prefix dce:      <http://purl.org/dc/elements/1.1/> .
```

```
@prefix xsd:      <http://www.w3.org/2001/XMLSchema#> .
```

```
<http://dbtune.org/jamendo/record/6015>  
  dce:date        "2007-06-15T17:13:58"^^xsd:dateTime .
```

```
<http://dbtune.org/jamendo/record/6343>  
  dce:date        "2007-07-05T19:32:23"^^xsd:dateTime .
```

```
<http://dbtune.org/jamendo/record/12863>  
  dce:date        "2007-11-19T11:46:34"^^xsd:dateTime .
```

```
<http://dbtune.org/jamendo/record/3993>  
  dce:date        "2007-02-01T18:08:28"^^xsd:dateTime .
```

Public Domain	dc:date	"2006-01-20T19:06:18"^^xsd:dateTime
Golden Opportunity	dc:date	"2007-10-25T15:54:54"^^xsd:dateTime

08: Filtering Results

```
PREFIX dce: <http://purl.org/dc/elements/1.1/>
```

```
CONSTRUCT {  
    ?s ?p ?o .  
}
```

```
WHERE {  
    ?s ?p ?o .  
    FILTER (?p != dce:date)  
}
```

Example Server

- Cliopatria instance with Jamendo dataset pre-loaded:
- <http://eculture.cs.vu.nl:1235/home>
- User: test
- Password:

09: All Albums from July 2007

```
PREFIX mo: <http://purl.org/ontology/mo/>
PREFIX dce: <http://purl.org/dc/elements/1.1/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
SELECT ?album ?title ?release_date
WHERE {
    ?album a mo:Record ;
           dce:date ?release_date ;
           dce:title ?title .
    FILTER (YEAR(?release_date) = 2007 && MONTH(?release_date) =
7)
}
ORDER BY ?release_date
```

album	title	release_date
Demo 2005	"Demo 2005"^^xsd:string	"2007-07-04T16:24:23"^^xsd:dateTime
Cinema	"Cinema"^^xsd:string	"2007-07-04T16:39:11"^^xsd:dateTime
ECLECTEK II	"ECLECTEK II"^^xsd:string	"2007-07-04T17:02:09"^^xsd:dateTime
Unknow user	"Unknow user"^^xsd:string	"2007-07-04T18:06:18"^^xsd:dateTime
Turkish Delight the Opera, Act I, live at the Black Cap	"Turkish Delight the Opera, Act I, live at the Black Cap"^^xsd:string	"2007-07-04T22:17:36"^^xsd:dateTime
Turkish Delight the opera, Act II, live at The Black Cap	"Turkish Delight the opera, Act II, live at The Black Cap"^^xsd:string	"2007-07-04T23:03:26"^^xsd:dateTime
girls always cry-xysides	"girls always cry-xysides"^^xsd:string	"2007-07-04T23:16:26"^^xsd:dateTime

Querying and Updating with SPARQL 1.1

Learning

SPARQL



O'REILLY®

Bob DuCharme

Nice book:

Bob DuCharme, "Learning SPARQL", O'Reilly.

```
examples$ sparql --query=2.3.4.rq --data=jamendo.ttl --quiet
```

pred	class
<http://xmlns.com/foaf/0.1/name>	
<http://xmlns.com/foaf/0.1/made>	mo:Record
<http://xmlns.com/foaf/0.1/img>	
<http://xmlns.com/foaf/0.1/based_near>	
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	
<http://xmlns.com/foaf/0.1/homepage>	
mo:biography	

Apache Jena comes with nice SPARQL command-line tools

<http://jena.apache.org>

```
rappor -i ntriples 1.2.3-literals.ntriples -o turtle  
rappor -i turtle jstar.ttl -o dot
```

Redland / librdf – the “Swiss Army Knife” of RDF tools

<http://librdf.org>