

# Getting Started with SPARQL


Bob DuCharme  
November 4, 2010

New York Semantic Web Meetup



## Introductions


- Presentation and all its URLs:  
<http://www.snee.com/semwebmeetup/2010-11-04>
- Me: SGML and XML at Moody's, LexisNexis, Innodata Isogen, TopQuadrant
- Weblog: <http://www.snee.com/bobdc.blog>
- Twitter: @bobdc



## Outline

- SPARQL and the Semantic Web
- Simple queries of local data and remote data
- Building on the simple queries
- SPARQL 1.1
- SPARQL tools, SPARQL in web apps
- Questions


© Copyright 2007-2010 TopQuadrant Inc. Slide 3



## What is SPARQL?

- **SPARQL Protocol and RDF Query Language**
- “Trying to use the Semantic Web without SPARQL is like trying to use a relational database without SQL” – Tim Berners-Lee


© Copyright 2007-2010 TopQuadrant Inc. Slide 4



## The Semantic Web

A set of standards and best practices for sharing data and the semantics of that data over the web for use by applications.

© Copyright 2007-2010 TopQuadrant Inc. Slide 5




## The Semantic Web

A **set of standards** and best practices for sharing data and the semantics of that data over the web for use by applications.

- RDF
- (OWL, RDFS)
- SPARQL


© Copyright 2007-2010 TopQuadrant Inc. Slide 6



## RDF

- Resource Description Framework
- Store data about anything, but especially metadata about resources
- Stored where?
- Very easily aggregated

© Copyright 2007-2010 TopQuadrant Inc. Slide 7



## An RDF “statement”: the triple

- (Subject, predicate, object)
- “index.html has the title ‘My Home Page’.”
- Easily stores (resource ID, propertyName, propertyValue) assertions

© Copyright 2007-2010 TopQuadrant Inc. Slide 8

TopQuadrant

## Triples

```
# rdf1.nt: sample RDF file in n-triples format.

<http://www.snee.com/bob/index.html>
<http://purl.org/dc/elements/1.1/title>
"My Home Page".

<http://www.someclub.org/memberID/4329>
<http://xmlns.com/foaf/0.1/homepage>
<http://www.snee.com/bob/index.html>.
```


© Copyright 2007-2010 TopQuadrant Inc. Slide 9

TopQuadrant

## Linking triples into a “graph”

```
graph TD
  A([http://www.someclub.org/memberID/4329]) -- "http://xmlns.com/foaf/0.1/homepage" --> B([http://www.snee.com/bob/index.html])
  B -- "http://purl.org/dc/elements/1.1/title" --> C[\"My Home Page\"]
```


© Copyright 2007-2010 TopQuadrant Inc. Slide 10



## SPARQL specs

- 1.0 became W3C spec January 2008:
  - SPARQL Query Language for RDF
  - SPARQL Protocol for RDF
  - SPARQL Query Results XML Format
- 1.1 still in Working Draft status, adds:
  - UPDATE
  - Several smaller documents that are being gradually rolled into the ones above

© Copyright 2007-2010 TopQuadrant Inc. Slide 11



## Data to query

```
# filename: ex2.ttl
@prefix ab: <http://learningsparql.com/ns/demo#> .

ab:richard ab:homeTel "(229) 276-5135" .
ab:richard ab:email "richard49@hotmail.com" .

ab:cindy ab:homeTel "(245) 646-5488" .
ab:cindy ab:email "cindym@gmail.com" .

ab:craig ab:homeTel "(194) 966-1505" .
ab:craig ab:email "craigellis@yahoo.com" .
ab:craig ab:email "c.ellis@usairwaysgroup.com" .
```

© Copyright 2007-2010 TopQuadrant Inc. Slide 12

TopQuadrant

## Our first query

```
# filename: ex2b.spq
PREFIX ab: <http://learningsparql.com/ns/demo#>

SELECT ?craigEmail
WHERE {
  ab:craig ab:email ?craigEmail .
}
```

© Copyright 2007-2010 TopQuadrant Inc. Slide 13

TopQuadrant

## Our first query

```
# filename: ex2b.spq
PREFIX ab: <http://learningsparql.com/ns/demo#>

SELECT ?craigEmail
WHERE {
  ab:craig ab:email ?craigEmail .
}
```

© Copyright 2007-2010 TopQuadrant Inc. Slide 14

**Running it with ARQ**

```

arq --data ex2a.ttl --query ex2b.spq

```

Result:

```

-----
| craigEmail |
=====
| "c.ellis@usairwaysgroup.com" |
| "craigellis@yahoo.com" |
-----

```

© Copyright 2007-2010 TopQuadrant Inc. Slide 15

**Running it with TopBraid Composer**

The screenshot shows the TopBraid Composer interface. The 'Query Editor' tab is active, displaying the following SPARQL query:

```

PREFIX ab: <http://learningsparql.com/ns/addressbook#>

SELECT ?craigEmail
WHERE {
  ab:craig ab:email ?craigEmail .
}


```

The results pane on the right, titled '[craigEmail]', shows two rows of results:

c.ellis@usairwaysgroup.com
craigellis@yahoo.com

© Copyright 2007-2010 TopQuadrant Inc. Slide 16



 **More realistic data**

```
@prefix ab: <http://learningsparql.com/ns/demo#> .

ab:i0432 ab:firstName "Richard" .
ab:i0432 ab:lastName  "Mutt" .
ab:i0432 ab:homeTel   "(229) 276-5135" .
ab:i0432 ab:email     "richard49@hotmail.com" .

ab:i9771 ab:firstName "Cindy" .
ab:i9771 ab:lastName  "Marshall" .
ab:i9771 ab:homeTel   "(245) 646-5488" .
ab:i9771 ab:email     "cindym@gmail.com" .

ab:i8301 ab:firstName "Craig" .
ab:i8301 ab:lastName  "Ellis" .
ab:i8301 ab:email     "craigellis@yahoo.com" .
ab:i8301 ab:email     "c.ellis@usairwaysgroup.com" .
```

© Copyright 2007-2010 TopQuadrant Inc. Slide 17

 **Finding Craig's email address**


```
PREFIX ab: <http://learningsparql.com/ns/demo#>

SELECT ?craigEmail
WHERE {
  ?person ab:firstName "Craig" .
  ?person ab:email ?craigEmail .
}
```

**Answer:**

```
-----
| craigEmail |
=====
| "c.ellis@usairwaysgroup.com" |
| "craigellis@yahoo.com" |
-----
```

© Copyright 2007-2010 TopQuadrant Inc. Slide 18


 **Finding Craig Ellis's email address**

```
# Assume ab: prefix is declared in remaining examples
SELECT ?craigEmail
WHERE {
  ?person ab:firstName "Craig" .
  ?person ab:lastName  "Ellis" .
  ?person ab:email ?craigEmail .
}
```

Or...

```
SELECT ?craigEmail
WHERE {
  ?person ab:firstName "Craig" ;
         ab:lastName  "Ellis" ;
         ab:email ?craigEmail .
}
```

© Copyright 2007-2010 TopQuadrant Inc. Slide 19

 **Who just called me?**

```
SELECT ?person
WHERE {
  ?person ab:homeTel "(229) 276-5135" .
}
```

Answer:

```
-----
| person |
=====
| ab:i0432 |
-----
```

© Copyright 2007-2010 TopQuadrant Inc. Slide 20

TopQuadrant

## Yeah, but what's their name?

```

SELECT ?first ?last
WHERE {
  ?person ab:homeTel "(229) 276-5135" .
  ?person ab:firstName ?first .
  ?person ab:lastName ?last .
}

```

Answer:

```

-----
| first      | last      |
=====
| "Richard" | "Mutt"    |
-----

```

© Copyright 2007-2010 TopQuadrant Inc. Slide 21

TopQuadrant

## Searching for strings

```

SELECT *
WHERE {
  ?s ?p ?o FILTER (regex(?o, "yahoo","i")).
}

```


Answer:

```

-----
| s          | p          | o          |
=====
| ab:i8301  | ab:email   | "craigellis@yahoo.com" |
-----

```

© Copyright 2007-2010 TopQuadrant Inc. Slide 22

 **What could go wrong?**

```

SELECT ?craigEmail ?homeTel
WHERE {
  ?person ab:firstName "Craig" .
  ?person ab:lastName "Ellis" .
  ?person ab:email ?craigEmail .
  ?person ab:homeTel ?homeTel .
}

```


**Answer:**

```

-----
| craigEmail | homeTel |
=====
-----

```

© Copyright 2007-2010 TopQuadrant Inc. Slide 23

 **Data that might not be there**

```

SELECT ?firstName ?lastName ?homeTel
WHERE {
  ?s ab:firstName ?firstName ;
     ab:lastName ?lastName ;
     ab:homeTel ?homeTel .
}

```

**Answer:**

```

-----
| firstName | lastName | homeTel |
=====
| "Cindy"   | "Marshall" | "(245) 646-5488" |
| "Richard" | "Mutt"     | "(229) 276-5135" |
-----

```

**Where's Craig?**

© Copyright 2007-2010 TopQuadrant Inc. Slide 24

TopQuadrant

## Data that might not be there

```

SELECT ?firstName ?lastName ?homeTel
WHERE {
  ?s ab:firstName ?firstName ;
     ab:lastName ?lastName .
  OPTIONAL {
    ?s ab:homeTel ?homeTel.
  }
}

```

Answer:

firstName	lastName	homeTel
"Craig"	"Ellis"	
"Cindy"	"Marshall"	"(245) 646-5488"
"Richard"	"Mutt"	"(229) 276-5135"

© Copyright 2007-2010 TopQuadrant Inc. Slide 25

TopQuadrant

## Querying a public data source

SPARQL Explorer for <http://dbpedia.org/sparql>

```

SPARQL:
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdf: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <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 [com.hackplayers.com](http://com.hackplayers.com) #10 200808

© Copyright 2007-2010 TopQuadrant Inc. Slide 26

## Dbpedia: information from Wikipedia infoboxes

The screenshot shows the Wikipedia article for "Tennis the Menace". A red circle highlights the infobox area on the right side of the page, which contains a cartoon illustration of Bart Simpson and other characters from The Simpsons. The infobox includes the title "Tennis the Menace", a small image, and a list of categories and references.

© Copyright 2007-2010 TopQuadrant Inc. Slide 27

## Everything Bart wrote on blackboard in season 12


```

SELECT ?episode ?chalkboard_gag
WHERE {
?episode skos:subject
<http://dbpedia.org/resource/Category:The_Simpsons_episodes%2C_season_12>.
?episode dbpedia2:blackboard ?chalkboard_gag }
    
```

SPARQL results:

episode	chalkboard_gag
:A_Tale_of_Two_Springfields	"I will not plant subliminal" messa" gore"s""@en
:Bye_Bye_Nerdie	"I will not scare the vice president""@en
:Children_of_a_Lesser_Cloud	"Today is not Mothra's Day""@en
:Day_of_the_Jackanapes	"The hamster did not have a full life""@en
:HOMR	:Network_television
:Homer_vs_Dignity	"I will not surprise the incontinent""@en
:Hungry%2C_Hungry_Homer	"Temptation Island was not a sleazy piece of crap""@en
:%27m_Goin%27_to_Praiseland	"Genetics is not an excuse""@en
:Insane_Clown_Poppy	"I will not surprise the incontinent.""@en
:Lisa_the_Tree_Hugger	"I am not the acting President.""@en
:New_Kids_on_the_Bleech	"I will not buy a presidential pardon""@en
:Pokey_Mom	:Who_Let_the_Dogs_Out%3F
:Simpson_Safari	"I will not flush evidence""@en
:Simpsons_Tall_Tales	"I should not be twenty-one by now""@en
:Skinner%27s_Sense_of_Snow	"Science class should not end in tragedy""@en
:Tennis_the_Menace	"I will not publish the principal's credit report""@en
:The_Computer_Wore_Menace_Shoes	"I will only provide a urine sample when asked""@en
:The_Great_Money_Caper	"The nurse is not dealing""@en
:Trilogy_of_Error	"Fire is not the cleanser""@en


TopQuadrant™



## The Semantic Web

A set of standards and best practices for sharing data and the semantics of that data **over the web for use by applications.**

© Copyright 2007-2010 TopQuadrant Inc. Slide 29




## Retrieving DBpedia data

Bart blackboard query can be stored in a URL like this:

```
http://dbpedia.org/sparql?default-graph-uri=http%3A%2F%2Fdbpedia.org&query=PREFIX%20dbpedia%20%3A%20%3Chttp%3A%2F%2Fdbpedia.org%2Fproperty%2F%3E%0ASELECT%20%3Fepisode%20%3Fchalkboard_gag%0AWHERE%20%7B%20%3Fepisode%20skos%3Asubject%20%3Chttp%3A%2F%2Fdbpedia.org%2Fresource%2FCategory%3AThe_Simpsons_episodes%252C_season_12%3E.%20%3Fepisode%20dbpedia%20%3Ablackboard%20%3Fchalkboard_gag%20%7D
```


© Copyright 2007-2010 TopQuadrant Inc. Slide 30



## Review

- RDF basics
- SPARQL 1.0 SELECT queries of local and web-based data
- Data that might not be there (OPTIONAL key word)
- Filtering data based on a regular expression

© Copyright 2007-2010 TopQuadrant Inc. Slide 31




## Other options

- UNION: specify two sets of data to pull at once
- ORDER BY ?variableName: sort output
- LIMIT
- DISTINCT: don't show duplicates
- Data types
- Functions (regex() earlier; more)

© Copyright 2007-2010 TopQuadrant Inc. Slide 32






## Query Forms

- SELECT
- DESCRIBE
- ASK
- CONSTRUCT

© Copyright 2007-2010 TopQuadrant Inc. Slide 33




## ASK whether solution exists

```
#SELECT ?craigEmail ?homeTel
ASK
WHERE {
  ?person ab:firstName "Craig" .
  ?person ab:lastName "Ellis" .
  ?person ab:email ?craigEmail .
  ?person ab:homeTel ?homeTel .
}
```

**Answer:**

Ask => No

© Copyright 2007-2010 TopQuadrant Inc. Slide 34

 **Creating new data with CONSTRUCT**


```

PREFIX ab: <http://learningsparql.com/demo#>
PREFIX v: <http://www.w3.org/2001/vcard-rdf/3.0#>

CONSTRUCT {
  ?s v:given-name ?firstName ;
    v:family-name ?lastName ;
    v:homeTel ?homeTel .
}
WHERE {
  ?s ab:firstName ?firstName ;
    ab:lastName ?lastName .
  OPTIONAL {
    ?s ab:homeTel ?homeTel.
  }
}

```

© Copyright 2007-2010 TopQuadrant Inc. Slide 35


 **Inferencing with CONSTRUCT, part 1**

```

CONSTRUCT {
  ?person ab:mother ?parent .
}
WHERE {
  ?person ab:parent ?parent .
  ?parent :sex "female" .
}

```

© Copyright 2007-2010 TopQuadrant Inc. Slide 36



## Inferencing with CONSTRUCT, part 2

```
CONSTRUCT {  
  ?person ab:grandmother ?parent2 .  
  ?parent2 rdfs:type ab:Grandmothers .  
}  
WHERE {  
  ?person ab:parent ?parent1 .  
  ?parent1 ab:parent ?parent2 .  
  ?parent2 :sex "female" .  
}
```

© Copyright 2007-2010 TopQuadrant Inc. Slide 37



## SPARQL 1.1

- Still at least a few months to go
- Supported by ARQ
- Adding on to what we've seen so far:
  - Projected expressions
  - Aggregates and functions
  - Better negation
  - Subqueries and federated queries
  - Property paths
- New: UPDATE

© Copyright 2007-2010 TopQuadrant Inc. Slide 38

TopQuadrant

## Projected expressions

```
PREFIX ab: <http://learningsparql.com/demo#>
PREFIX fn: <http://www.w3.org/2005/xpath-functions#>

SELECT ( fn:concat(?firstName, " ", ?lastName) AS ?name )
WHERE {
  ?s ab:firstName ?firstName ;
     ab:lastName ?lastName .
}
```

**Answer:**

```
-----
| name |
=====
| "Craig Ellis" |
| "Cindy Marshall" |
| "Richard Mutt" |
-----
```

© Copyright 2007-2010 TopQuadrant Inc. Slide 39

TopQuadrant

## Projected expressions as variables?


- **Or**

```
SELECT (( ?bill * .2) AS ?tip )
```
- **In ARQ but not in SPARQL 1.1**

```
SELECT ?tip
WHERE {
  ab:meal ab:bill ?bill .
  LET (?tip := ?bill * .2) .
}
```
- **Maybe in 1.1 spec**

```
SELECT ?tip
WHERE {
  ab:meal ab:bill ?bill .
  BIND (( ?bill * .2) AS ?tip ) .
}
```

© Copyright 2007-2010 TopQuadrant Inc. Slide 40




## Aggregates

```

PREFIX : <http://books.example/>
SELECT (SUM(?lprice) AS ?totalPrice)
WHERE {
  ?org :affiliates ?auth .
  ?auth :writesBook ?book .
  ?book :price ?lprice .
}
GROUP BY ?org
HAVING (SUM(?lprice) > 10)

```


© Copyright 2007-2010 TopQuadrant Inc. Slide 41



## Functions

- “SPARQL provides a subset of the functions and operators defined by XQuery [Operator Mapping](#). XQuery 1.0 section [2.2.3 Expression Processing](#) describes the invocation of XPath functions.”
- More functions: sum, count, avg, min, max, regex, isURI, isNumeric, casting...
- More supported by ARQ: contains, starts-with, lower-case... see <http://jena.sourceforge.net/ARQ/library-function.html>

© Copyright 2007-2010 TopQuadrant Inc. Slide 42




## Negation in SPARQL 1.0

```

SELECT ?first ?last
WHERE {
  ?s ab:firstName ?first ;
     ab:lastName ?last .
  OPTIONAL {
    ?s ab:homeTel ?homeTel .
  }
  FILTER(!bound(?homeTel))
}

```

© Copyright 2007-2010 TopQuadrant Inc. Slide 43



## Negation in SPARQL 1.1


```

SELECT ?first ?last
WHERE {
  ?s ab:firstName ?first ;
     ab:lastName ?last .
  MINUS {
    ?s ab:homeTel ?homeTel .
  }
}

```

- Or, instead of **MINUS**, **FILTER NOT EXISTS**

© Copyright 2007-2010 TopQuadrant Inc. Slide 44




## Subqueries and federation

```

SELECT ?birthDate ?spouseName ?movieTitle ?movieDate {
  { SERVICE <http://dbpedia.org/sparql>
    { SELECT ?birthDate ?spouseName WHERE {
      ?actor rdfs:label "Arnold Schwarzenegger"@en ;
      dbpo:birthDate ?birthDate ;
      dbpo:spouse ?spouseURI .
      ?spouseURI rdfs:label ?spouseName .
      FILTER ( lang(?spouseName) = "en" )
    }
  }
}
{ SERVICE <http://data.linkedmdb.org/sparql>
  { SELECT ?actor ?movieTitle ?movieDate WHERE {
    ?actor imdb:actor_name "Arnold Schwarzenegger".
    ?movie imdb:actor ?actor ;
    dcterms:title ?movieTitle ;
    dcterms:date ?movieDate .
  }
}
}

```

© Copyright 2007-2010 TopQuadrant Inc. Slide 45



## Property paths

- sample data
 

```

@prefix : <http://learningsparql.com/demo#> .
:jane :knows :frank .
:frank :knows :sarah .
:sarah :knows :steve .


```
- query
 

```

PREFIX : <http://learningsparql.com/demo#>
SELECT ?person
WHERE {
  :jane :knows+ ?person .
}

```


© Copyright 2007-2010 TopQuadrant Inc. Slide 46



## SPARQL UPDATE

```
DELETE DATA {  
  ?s ab:firstName "Robert" .  
}  
INSERT DATA {  
  ?s ab:firstName "Bob" .  
}
```

© Copyright 2007-2010 TopQuadrant Inc. Slide 47



## SPARQL Tools

- <http://esw.w3.org/SparqlImplementations>
- ARQ: command line tools, Jena Java library
- Libraries for other languages: Java, JavaScript, Python, Perl, PHP...
- Built into:
  - websites (SPARQL endpoints)
  - triplestores
  - TopBraid Composer
- SPARQL engine web service...

© Copyright 2007-2010 TopQuadrant Inc. Slide 48



**Website as SPARQL web service**

- sparql.org SPARQLer
- My FOAF file:  
http://www.snee.com/bob/foaf.rdf
- This query:

```

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX air: <http://www.megginson.com/exp/ns/airports#>
SELECT ?personName ?airportCode WHERE {
    ?person foaf:name ?personName ;
            foaf:nearestAirport ?airport .
    ?airport air:iata ?airportCode .
}

```

© Copyright 2007-2010 TopQuadrant Inc. Slide 49

**Entering query on sparql.org**

SPARQLer - General purpose processor

General SPARQL query : input query, set any options and press "Get Results"

```

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX air: <http://www.megginson.com/exp/ns/airports#>
SELECT ?personName ?airportCode WHERE {
    ?person foaf:name ?personName ;
            foaf:nearestAirport ?airport .
    ?airport air:iata ?airportCode .
}

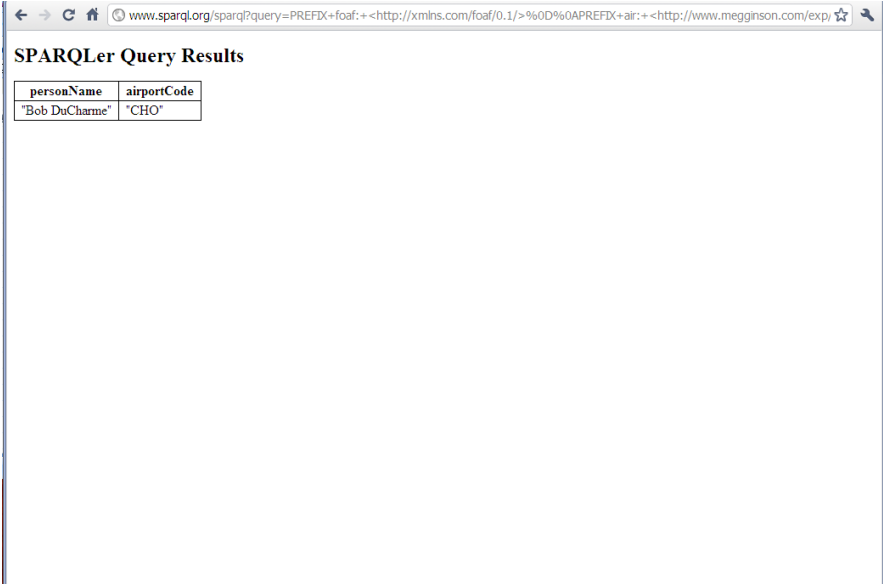
```

Target graph URI (or use FROM in the query) 
Output XML:  with XSLT style sheet (leave blank for none): 
or JSON output: 
or text output: 
or CSV output: 
or TSV output: 
Force the accept header to text/plain regardless

© Copyright 2007-2010 TopQuadrant Inc. Slide 50

TopQuadrant

## sparql.org query result



www.sparql.org/sparql?query=PREFIX+foaf:+<http://xmlns.com/foaf/0.1/>%0D%0APREFIX+air:+<http://www.megginson.com/exp/

SPARQLer Query Results

personName	airportCode
"Bob DuCharme"	"CHO"

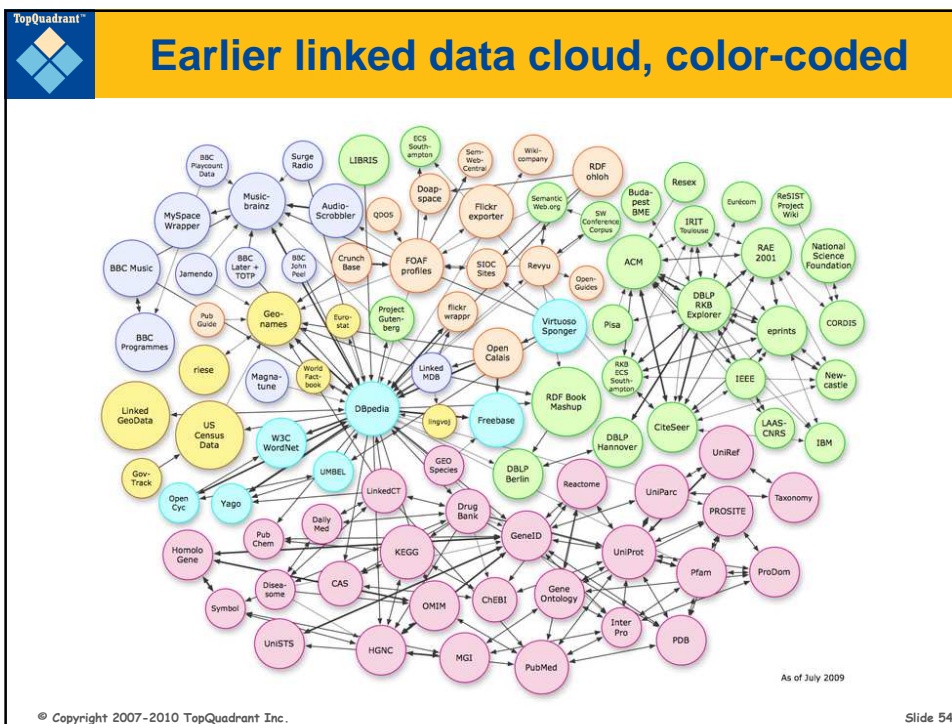
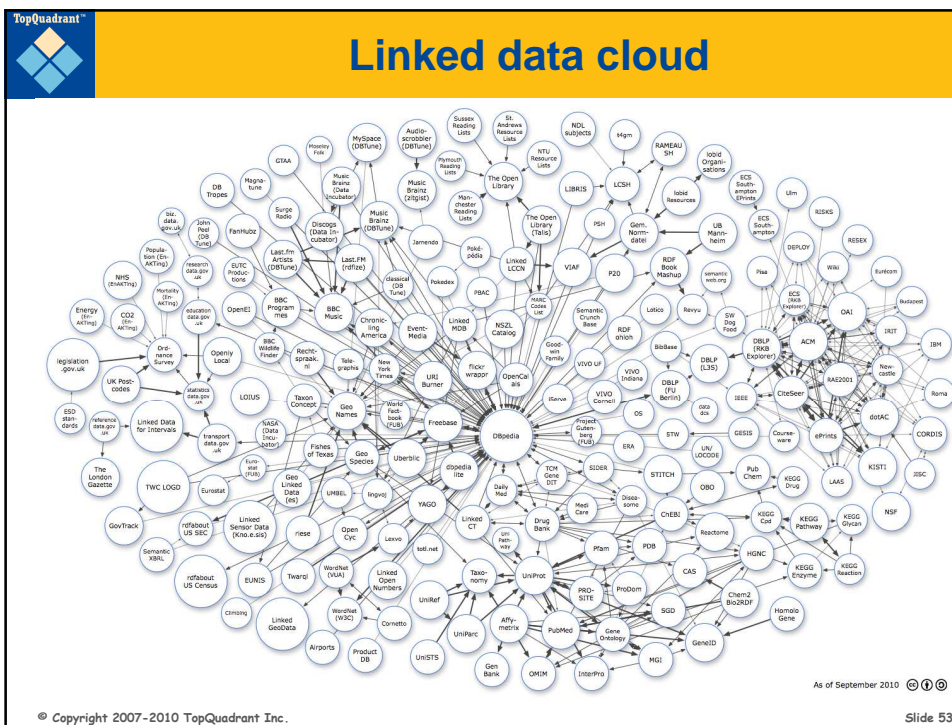
© Copyright 2007-2010 TopQuadrant Inc. Slide 51

TopQuadrant

## URL for sparql.org query

```
http://www.sparql.org/sparql?query=PREFIX+foaf:
+%3Chttp://xmlns.com/foaf/0.1/%3E%0D%0APREF
IX+air:+%3Chttp://www.megginson.com/exp/ns/ai
rports%23%3E%0D%0ASELECT+%3FpersonName+%3Fair
portCode+WHERE+{%0D%0A++%3Fperson+foaf:name+%
3FpersonName+;%0D%0A+++++++foaf:nearestAi
rport+%3Fairport+.+%0D%0A++%3Fairport+air:iat
a+%3FairportCode+.+%0D%0A}&default-graph-uri=
http://www.snee.com/bob/foaf.rdf+&output=xml&
stylesheet=/xml-to-html.xsl
```

© Copyright 2007-2010 TopQuadrant Inc. Slide 52



TopQuadrant™

## Is SPARQL Difficult?



© Copyright 2007-2010 TopQuadrant Inc. Slide 55

TopQuadrant™

## Is SPARQL Difficult?

“Consider, for instance, SPARQL, a query language. To find, say, music artists associated with the producer Timbaland, you'd have to type a long piece of convoluted code that most of us wouldn't bother to do.”

© Copyright 2007-2010 TopQuadrant Inc. Slide 56

**This query...**

```

PREFIX : <http://dbpedia.org/resource/>
PREFIX d: <http://dbpedia.org/ontology/>

SELECT DISTINCT ?artist WHERE {
  ?album d:producer :Timbaland .
  ?album d:musicalArtist ?artist .
}

```

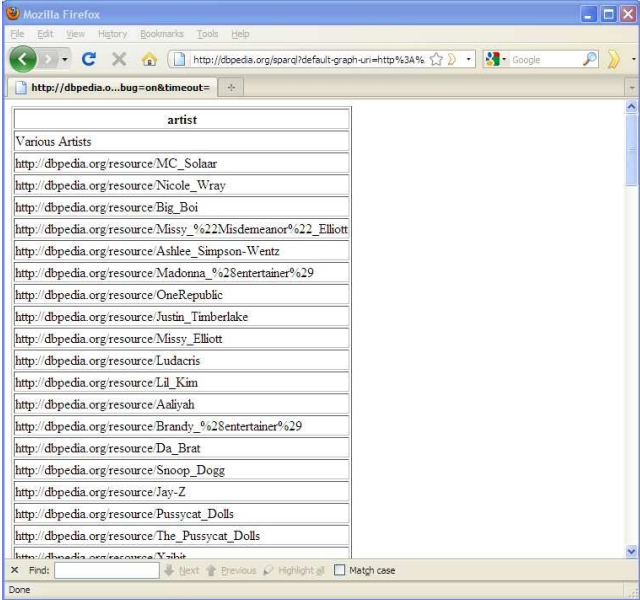
© Copyright 2007-2010 TopQuadrant Inc. Slide 57

**entered here...**

© Copyright 2007-2010 TopQuadrant Inc. Slide 58

TopQuadrant™

## and there they are.



© Copyright 2007-2010 TopQuadrant Inc. Slide 59

TopQuadrant™

## On the other hand...

Some JavaScript from a View Source of that same CNN page:

```

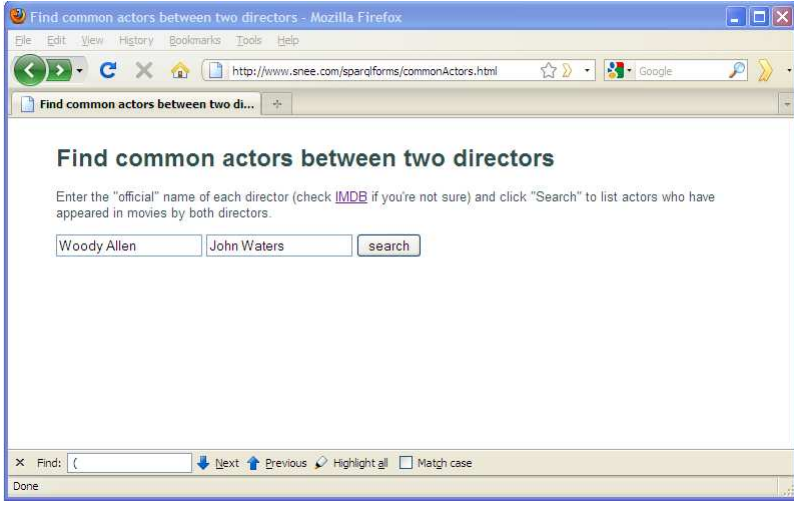
if (cnnWinExtraRegExp.test(cnnWinExtra)) { var cnnOmniExtra =
cnnWinExtraRegExp.split(cnnWinExtra); cnnWinLoc = cnnWinLoc +
cnnOmniExtra[0]; } else { cnnWinLoc = cnnWinLoc +
cnnWinExtra; } } if (typeof(cnnPageName) != "undefined")
{ s.pageName = cnnPageName; s.eVar1 = cnnPageName; } else
{ s.pageName = cnnWinLoc; s.eVar1 = cnnWinLoc; } if
(typeof(cnnSectionName) != "undefined")
{ s.channel = cnnSectionName; s.eVar2 = cnnSectionName; } else
{ s.channel = "Nonlabeled"; s.eVar2 = "Nonlabeled"; } if
(typeof(cnnSubSectionName) != "undefined")
{ s.server = cnnSubSectionName; s.eVar3 = cnnSubSectionName; } else
{ s.server = ""; s.eVar3 = ""; } if (typeof(cnnSectionFront) !=
"undefined") { s.prop1 = cnnSectionFront; } if
(typeof(cnnContentType) != "undefined")
{ s.prop4 = cnnContentType; s.prop6 = s.pageName; }

```

© Copyright 2007-2010 TopQuadrant Inc. Slide 60

TopQuadrant

## Form-based SPARQL app



Find common actors between two directors

Enter the "official" name of each director (check [IMDB](#) if you're not sure) and click "Search" to list actors who have appeared in movies by both directors.

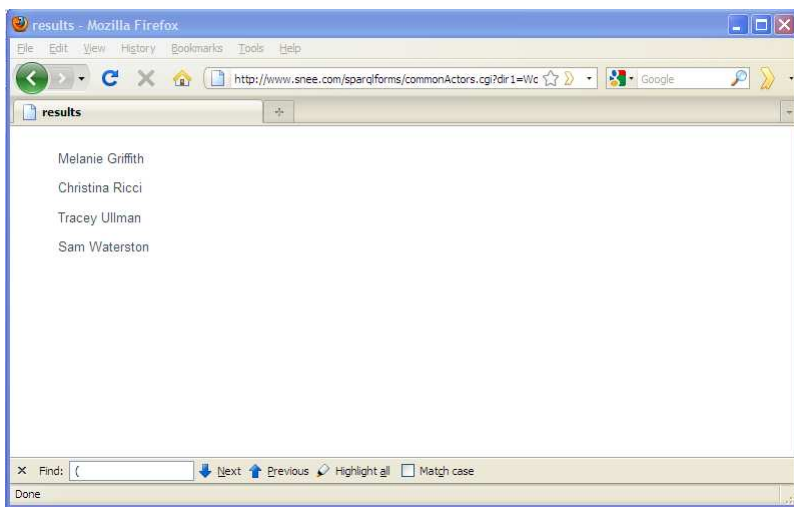
Woody Allen John Waters search

Find: ( ) Next Previous Highlight all Match case  
Done

© Copyright 2007-2010 TopQuadrant Inc. Slide 61

TopQuadrant

## Form-based SPARQL app: results




results

- Melanie Griffith
- Christina Ricci
- Tracey Ullman
- Sam Waterston

Find: ( ) Next Previous Highlight all Match case  
Done

© Copyright 2007-2010 TopQuadrant Inc. Slide 62



## commonactors.cgi main() part 1

```

def main():
    form = cgi.FieldStorage()
    dir1name = form.getvalue('dir1')
    dir2name = form.getvalue('dir2')

    sparql = SPARQLWrapper("http://data.linkedmdb.org/sparql")
    queryString = ""

    PREFIX m: <http://data.linkedmdb.org/resource/movie/>
    SELECT DISTINCT ?actorName WHERE {


        ?dir1      m:director_name "DIR1-NAME".
        ?dir2      m:director_name "DIR2-NAME".
        ?dir1film m:director ?dir1;
                  m:actor ?actor.

        ?dir2film m:director ?dir2;
                  m:actor ?actor.

        ?actor    m:actor_name ?actorName.
    }
    """

```

© Copyright 2007-2010 TopQuadrant Inc. Slide 63



## commonactors.cgi main() part 2

```

queryString = queryString.replace("DIR1-NAME",dir1name)
queryString = queryString.replace("DIR2-NAME",dir2name)


sparql.setQuery(queryString)
sparql.setReturnFormat(JSON)

try:
    ret = sparql.query()
    results = ret.convert()
    requestGood = True
except Exception, e:
    results = str(e)
    requestGood = False

```

© Copyright 2007-2010 TopQuadrant Inc. Slide 64





## commonactors.cgi main() part 3

```

print ""Content-type: text/html

<html>
<head>
<title>results</title>
<link href="simple.css" type="text/css" rel="stylesheet" />
</head>
<body>
""


if requestGood == False:
    print "<h1>Problem communicating with the server</h1>"
    print "<p>" + results + "</p>"
elif (len(results["results"]["bindings"]) == 0):
    print "<p>No results found.</p>"

else:
    for result in results["results"]["bindings"]:
        print "<p>" + result["actorName"]["value"] + "</p>"

print "</body></html>"

```

© Copyright 2007-2010 TopQuadrant Inc. Slide 65



## SPARQL Query Results XML Format


```

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX air:
<http://www.megginson.com/exp/ns/airports#>

SELECT ?personName ?airportCode WHERE {
    ?person foaf:name ?personName ;
           foaf:nearestAirport ?airport .
    ?airport air:iata ?airportCode .
}

```

© Copyright 2007-2010 TopQuadrant Inc. Slide 66



## SPARQL Query Results XML Format


```
<sparql xmlns="http://www.w3.org/2005/sparql-results#">

  <head>
    <variable name="personName"/>
    <variable name="airportCode"/>
  </head>

  <results>
    <result>
      <binding name="personName">
        <literal>Bob DuCharme</literal>
      </binding>
      <binding name="airportCode">
        <literal>CHO</literal>
      </binding>
    </result>
  </results>

</sparql>
```

© Copyright 2007-2010 TopQuadrant Inc. Slide 67



## In conclusion...

When you know some SPARQL, you can do more with:

- RDFS
- OWL
- triplestores
- SPARQL endpoints
- Linked Data

© Copyright 2007-2010 TopQuadrant Inc. Slide 68