

MOIR RITS  
NITE 10CT RIARGB

TIE  
OF PIKE  
MARS

WILLIAM KATZ WITH CECILIE A. BILKOW  
ILLUSTRATED BY J. J. HARRIS  
HARVARD UNIVERSITY PRESS

100% COTTON  
MADE IN THE USA

# Knowledge Graphs

## Lecture 3 – Querying Knowledge Graphs with SPARQL

### 3.3 More Complex Queries with SPARQL

Prof. Dr. Harald Sack

FIZ Karlsruhe – Leibniz Institute for Information Infrastructure

AIFB – Karlsruhe Institute of Technology

Autumn 2023



FIZ Karlsruhe  
Leibniz-Institut für Informationsinfrastruktur

### 3.1 How to Query RDF(S)

Excursion 3: DBpedia Knowledge Graph

Excursion 4: Wikidata Knowledge Graph

### 3.2 Complex Queries with SPARQL

### **3.3 More Complex SPARQL Queries**

### 3.4 SPARQL Sub-Select and Property Paths

### 3.5 SPARQL is more than a Query Language

### 3.6 Quality Assurance with SHACL Constraints

MOIR RITS  
NITE 10CT RIAGE

# TIE OF THE MARS

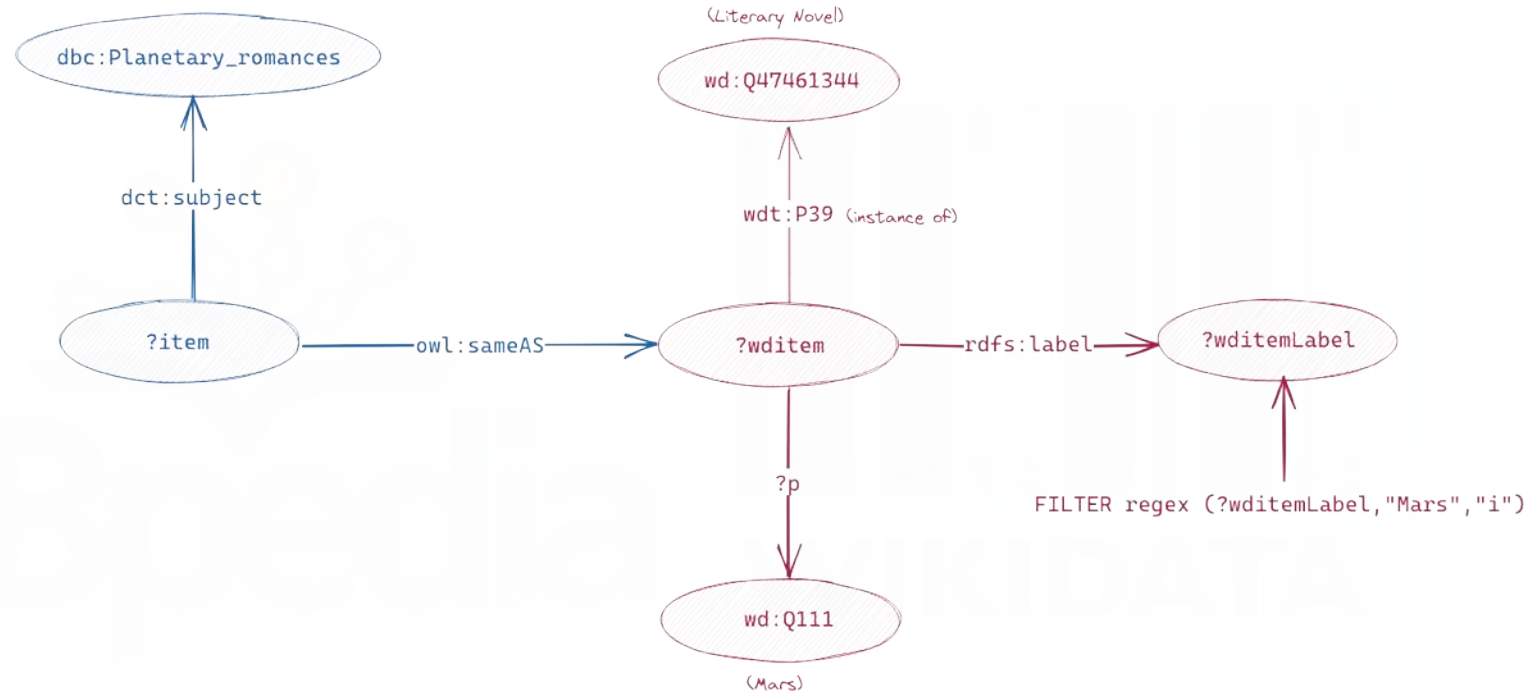
WANDA KAMRITH CRICH & BIKOS  
MARTIN RAY & JON  
AND MARGARET V. D.

IN THE NAME  
OF THE "MARTIAN" SPACE  
THE MARS RITE

which "planetary romance" scifi novels  
are dealing with "Mars" ?

# SPARQL Federated Queries

Example: which "planetary romances" sci-fi novels are dealing with "Mars" ?



[2,3]



# SPARQL Federated Queries

SPARQL enables federated queries over several RDF datasets or SPARQL endpoints via the **SERVICE** objective.



[2,3]

```

PREFIX dct: <http://purl.org/dc/terms/>
PREFIX dbc: <http://dbpedia.org/resource/Category:>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX owl: <http://www.w3.org/2002/07/owl#>

SELECT DISTINCT ?wditemLabel WHERE {
  SERVICE <http://dbpedia.org/sparql> {
    ?item dct:subject dbc:Planetary_romances ;
          owl:sameAs ?wditem FILTER regex (?wditem, "wikidata.org") .
  }
  SERVICE <https://query.wikidata.org/sparql> {
    ?wditem rdfs:label ?wditemLabel FILTER(LANG(?wditemLabel)="en").
    { ?wditem ?p wd:Q111 .}
    UNION
    { ?wditem rdfs:label ?wditemLabel FILTER regex (?wditemLabel,"Mars","i")}
  }
}

```

- Example: Connect **DBpedia** with **Wikidata** "which 'planetary romances' scifi novels are dealing with 'Mars' ?"
- Only possible, if SPARQL endpoints permit federation.

[query SPARQL endpoint](#)  
[query SPARQL endpoint again](#)

# SPARQL Federated Queries

Example: which "planetary romances" sci-fi novels are dealing with "Mars" ?


Wikidata Query Service

```

1 #defaultView:ImageGrid
2 PREFIX dct: <http://purl.org/dc/terms/>
3 PREFIX dbc: <http://dtpedia.org/resource/Category/>
4 PREFIX wdt: <http://www.wikidata.org/prop/direct/>
5 PREFIX owl: <http://www.w3.org/2002/07/owl#>
6
7 SELECT DISTINCT ?wditemLabel ?date ?image WHERE {
8   SERVICE <http://dtpedia.org/sparql/> {
9     ?item dct:subject dbc:Planetary_romances ;
10     owl:sameAs ?wditem FILTER regex (?wditem, "wikidata.org") .
11   }
12   SERVICE <https://query.wikidata.org/sparql/> {
13     ?wditem wdt:P31 wdt:Q7725634 ;
14     wdt:P577 ?date .
15     ?wditem rdfs:label ?wditemLabel FILTER(LANG(?wditemLabel)="en").
16     OPTIONAL {?wditem wdt:P18 ?image .}
17     { ?wditem ?p wdt:Q111 .}
18     UNION
19     { ?wditem rdfs:label ?wditemLabel FILTER regex (?wditemLabel,"Mars","i")}
20   }
21 }
22 ORDER BY ?date
23

```

5 results in 27 ms



commons:Warlord of Mars-1913.jpg  
1 January 1913  
The Warlord of Mars

commons:Princess of Mars large.jpg  
1 January 1917  
A Princess of Mars

commons:Aelita (cover by Shilling...  
1 January 1923  
Aelita

[query SPARQL endpoint](#)

# SPARQL Variable Assignments

Example:

Select authors with their notable scifi works *ordered by year of publication.*



The **BIND** form allows a value to be assigned to a variable.

```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX wikibase: <http://wikiba.se/ontology#>
PREFIX bd: <http://www.bigdata.com/rdf#>
```

```
SELECT ?authorLabel ?bookLabel ?book ?author ?year
```

```
WHERE {
```

```
    ?author wdt:P106 wd:Q36180 ;           # ?author occupation Wrote
        wdt:P800 ?book .                 # notableWork ?book
    ?book wdt:P577 ?date ;                 # ?book publicationDate ?date
        wdt:P136 wd:Q24925 .             # genre ScienceFiction
```

*Binding a new variable*

```
    BIND (YEAR(?date) AS ?year) FILTER (BOUND(?year))
```

```
    SERVICE wikibase:label { bd:serviceParam wikibase:language "en" }
```

```
} ORDER BY ?year
```




[query SPARQL endpoint](#) [3]

# SPARQL Variable Assignments

Example:

Select authors with their notable scifi works *ordered by year of publication.*


Wikidata Query Service
Examples
Help
More tools
Query Builder
English

```

1 PREFIX wd: <http://www.wikidata.org/entity/>
2 PREFIX wdt: <http://www.wikidata.org/prop/direct/>
3 PREFIX wikibase: <http://wikiba.se/ontology#>
4 PREFIX bd: <http://www.bigdata.com/rdf#>
5
6 SELECT ?authorLabel ?bookLabel ?book ?author ?year
7 WHERE {
8   ?author wdt:P106 wd:Q36180 ;
9           wdt:P800 ?book .
10  ?book wdt:P577 ?date ;
11        wdt:P136 wd:Q24925 .
12  BIND (YEAR(?date) AS ?year) FILTER (BOUND(?year))
13  SERVICE wikibase:label { bd:serviceParam wikibase:language "en" }
14 } ORDER BY ?year
15

```

Table
459 results in 1146 ms
Code
Download
Link

| authorLabel  | bookLabel   | book                            | author                        | year |
|--|---|---------------------------------|-------------------------------|------|
| Antonius Diogenes                                  | The Wonders beyond Thule                            | <a href="#">Q wd:Q4315715</a>   | <a href="#">Q wd:Q606249</a>  | 100  |
| Francis Bacon                                      | New Atlantis  | <a href="#">Q wd:Q1233830</a>   | <a href="#">Q wd:Q37388</a>   | 1627 |
| Margaret Cavendish, Duchess of Newcastle-upon-Tyne | The Blazing World                                   | <a href="#">Q wd:Q1082871</a>   | <a href="#">Q wd:Q242640</a>  | 1666 |
| John Bunyan  | The Pilgrim's Progress                              | <a href="#">Q wd:Q268211</a>    | <a href="#">Q wd:Q108206</a>  | 1678 |
| Marie-Anne de Roumier-Robert                       | Les Voyages de Milord Céton dans les sept planettes | <a href="#">Q wd:Q106370743</a> | <a href="#">Q wd:Q3291524</a> | 1765 |
| Louis-Sébastien Mercier                            | L'An 2440, rêve s'il en fut jamais                  | <a href="#">Q wd:Q82530</a>     | <a href="#">Q wd:Q709670</a>  | 1771 |



# SPARQL Aggregate Functions

Example: How many authors are there and  
how many of their notable works are science fiction novels?



```

PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX wikibase: <http://wikiba.se/ontology#>
PREFIX bd: <http://www.bigdata.com/rdf#>

SELECT (COUNT(?book) AS ?bookcount)
       (COUNT(DISTINCT(?author)) AS ?authorcount)
WHERE {
    ?author wdt:P106 wd:Q36180 ; # ?author :occupation :Writer
           wdt:P800 ?book ;      #           :notableWork ?book
           wdt:P136 wd:Q24925 . #           :genre ?ScienceFiction
    SERVICE wikibase:label { bd:serviceParam wikibase:language "en" }
}

```

aggregate  
functions


- **COUNT** is a SPARQL aggregate function which counts the number of times a given expression has a bound.
- More aggregate functions:
  - SUM
  - AVG
  - MIN/MAX
  - SAMPLE

[3]

[query SPARQL endpoint](#)

# SPARQL Aggregate Functions

Example: How many authors are there and  
how many of their notable works are science fiction novels?


**Wikidata Query Service**
Examples
Help
More tools
Query Builder
English

```

1 PREFIX wd: <http://www.wikidata.org/entity/>
2 PREFIX wdt: <http://www.wikidata.org/prop/direct/>
3 PREFIX wikibase: <http://wikiba.se/ontology#>
4 PREFIX bd: <http://www.bigdata.com/rdf#>
5
6 SELECT (COUNT(?book) AS ?bookcount)
7        (COUNT(DISTINCT(?author)) AS ?authorcount)
8 WHERE {
9     ?author wdt:P106 wd:Q36180 ; # ?author :occupation :Writer
10           wdt:P800 ?book ;      #           :notableWork ?book
11           wdt:P136 wd:Q24925 .#           :genre ?ScienceFiction
12     SERVICE wikibase:label { bd:serviceParam wikibase:language "en" }
13 }
14

```

Table
1 result in 513 ms
Code
Download
Link

| bookcount | authorcount |
|-----------|-------------|
| 526       | 181         |

# SPARQL Aggregate Functions

Example: *which* author wrote *how many* notable Science fiction novels?



```

PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX wikibase: <http://wikiba.se/ontology#>
PREFIX bd: <http://www.bigdata.com/rdf#>

SELECT (?authorLabel (COUNT(?book) AS ?bookcount))
WHERE {
    ?author wdt:P106 wd:Q36180 ; # ?author :occupation :Writer
            wdt:P800 ?book ;      # :notableWork ?book
            wdt:P136 wd:Q24925 . # :genre ?ScienceFiction
    SERVICE wikibase:label { bd:serviceParam wikibase:language "en" }
} GROUP BY ?authorLabel
ORDER BY DESC (?bookcount)
  
```

aggregate  
function


- The solution can be divided into groups via **GROUP BY**.
- The aggregate function is then calculated for each group.

[3]

[query SPARQL endpoint](#)

# SPARQL Aggregate Functions

Example: which author wrote how many notable Science fiction novels?

 Wikidata Query Service

[Examples](#)
[Help](#)
[More tools](#)
[Query Builder](#)

[English](#)

```

1 PREFIX wd: <http://www.wikidata.org/entity/>
2 PREFIX wdt: <http://www.wikidata.org/prop/direct/>
3 PREFIX wikibase: <http://wikiba.se/ontology#>
4 PREFIX bd: <http://www.bigdata.com/rdf#>
5
6 SELECT ?authorLabel (COUNT(?book) AS ?bookcount)
7 WHERE {
8   ?author wdt:P106 wd:Q36180 ; # ?author occupation :Writer
9           wdt:P800 ?book ; # :notableWork ?book
10          wdt:P136 wd:Q24923 . # :genre ?ScienceFiction
11   SERVICE wikibase:label { bd:serviceParam wikibase:language "en" }
12 } GROUP BY ?authorLabel
13 ORDER BY DESC (?bookcount)
14
    
```

181 results in 842 ms
 [Code](#)
[Download](#)
[Link](#)

| authorLabel           | bookcount |
|-----------------------|-----------|
| Stephen King          | 36        |
| Margaret Atwood       | 15        |
| Brandon Sanderson     | 15        |
| Ursula K. Le Guin     | 13        |
| Alastair Reynolds     | 12        |
| H. G. Wells           | 9         |
| Alice Bradley Sheldon | 9         |
| Lucius Shepard        | 9         |
| Samuel R. Delany      | 9         |
| Roger Zelazny         | 8         |
| H. P. Lovecraft       | 8         |
| Philip K. Dick        | 8         |
| C. S. Lewis           | 7         |

# SPARQL Aggregate Functions

Example: *which* author wrote *how many* notable Science fiction novels?

SPARQL 1.1 provides more aggregate functions

- SUM
- AVG
- MIN
- MAX
- SAMPLE – “pick” one non-deterministically
- GROUP\_CONCAT – concatenate values with a designated string separator



CUUCH  
MCARIS

# SPARQL Sub-Select and Property Paths

Next Lecture...

### Bibliographic References:

- Steve Harris, Andy Seaborne (2013), [SPARQL 1.1 Query Language](#), W3C Recommendation 21 March 2013
- Aidan Hogan (2020), [The Web of Data](#), Springer.
  - Chap. 6.2.4 Filtering and Binding Values, pp. 338–345.
  - Chap. 6.2.7 Aggregation, pp. 355–362.
  - Chap. 6.2.8 Solution Modifiers, pp. 363–364.
  - Chap. 6.4 SPARQL Federation, pp. 409–413.

### Picture References:

- [1] “A movie poster for the science fiction novel “the first Men on the Mars” which depicts the first landing on Mars in a retro-futuristic style showing the dry Mars surface, the rocket landing ship standing straight up in the lonely dessert surrounded a few astronauts exploring the environment.”, created via ArtBot, Deliberate, 2023, [CC-BY-4.0], <https://tinybots.net/artbot>
- [2] DBpedia logo, wiki.dbpedia.org, DBpedia Team [Public Domain], <https://commons.wikimedia.org/wiki/File:DBpediaLogo.svg>
- [3] Wikidata logo, Wikimedia Commons [Public Domain], <https://commons.wikimedia.org/wiki/File:Wikidata-logo-en.svg>
- [4] “A science fiction movie poster for “Cthulhu and the Gods of Mars” which depicts the first landing of humans on Mars in a retro-futuristic style showing how the great Cthulhu is hovering over the red dessert facing a few human astronauts surrounded by strange ancient artefacts.”, created via ArtBot, Deliberate, 2023, [CC-BY-4.0], <https://tinybots.net/artbot>