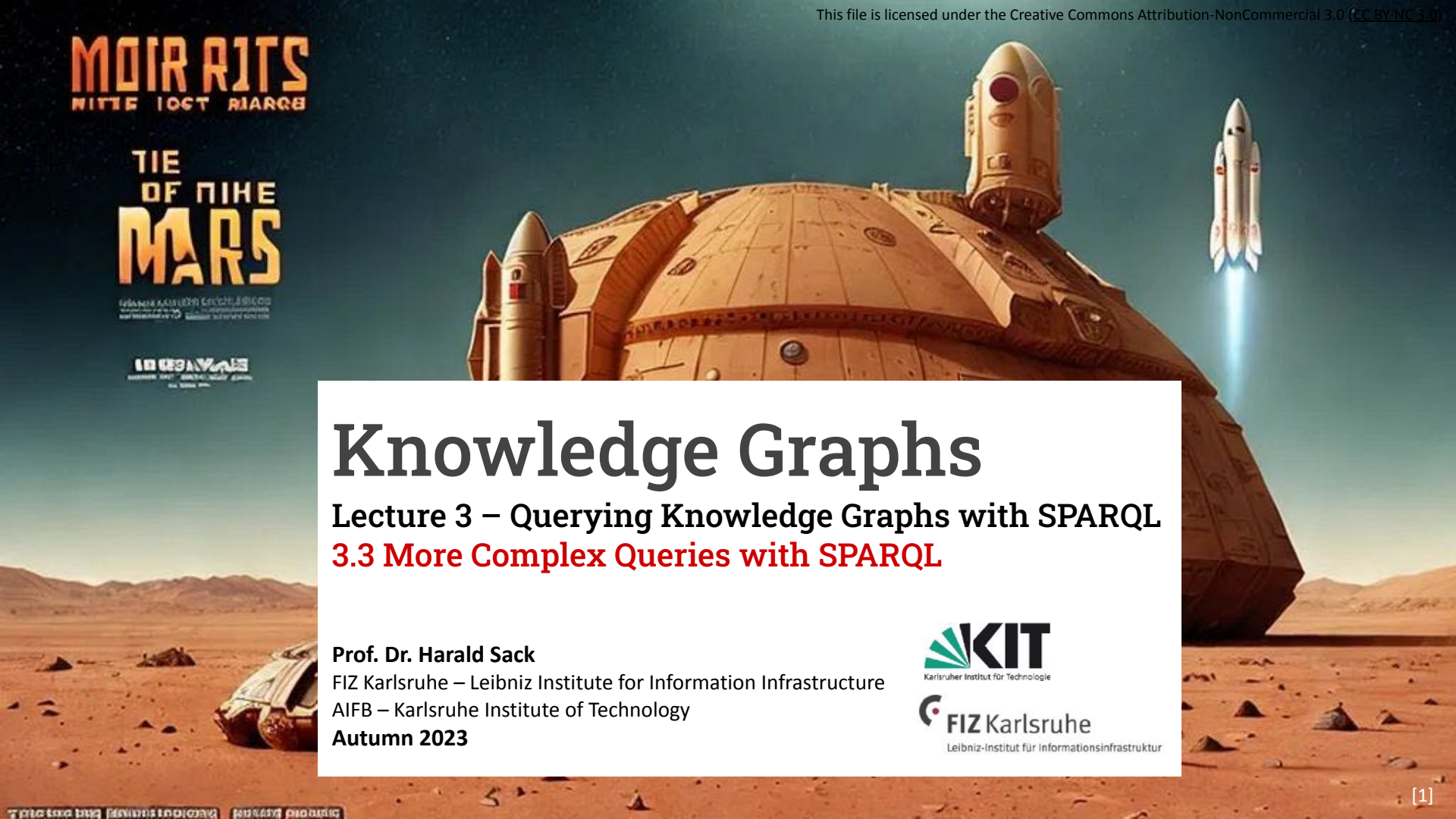
This file is licensed under the Creative Commons Attribution-NonCommercial 3.0 (CC BY-NC 3.0)



Knowledge Graphs

Lecture 3 – Querying Knowledge Graphs with SPARQL

3.3 More Complex Queries with SPARQL

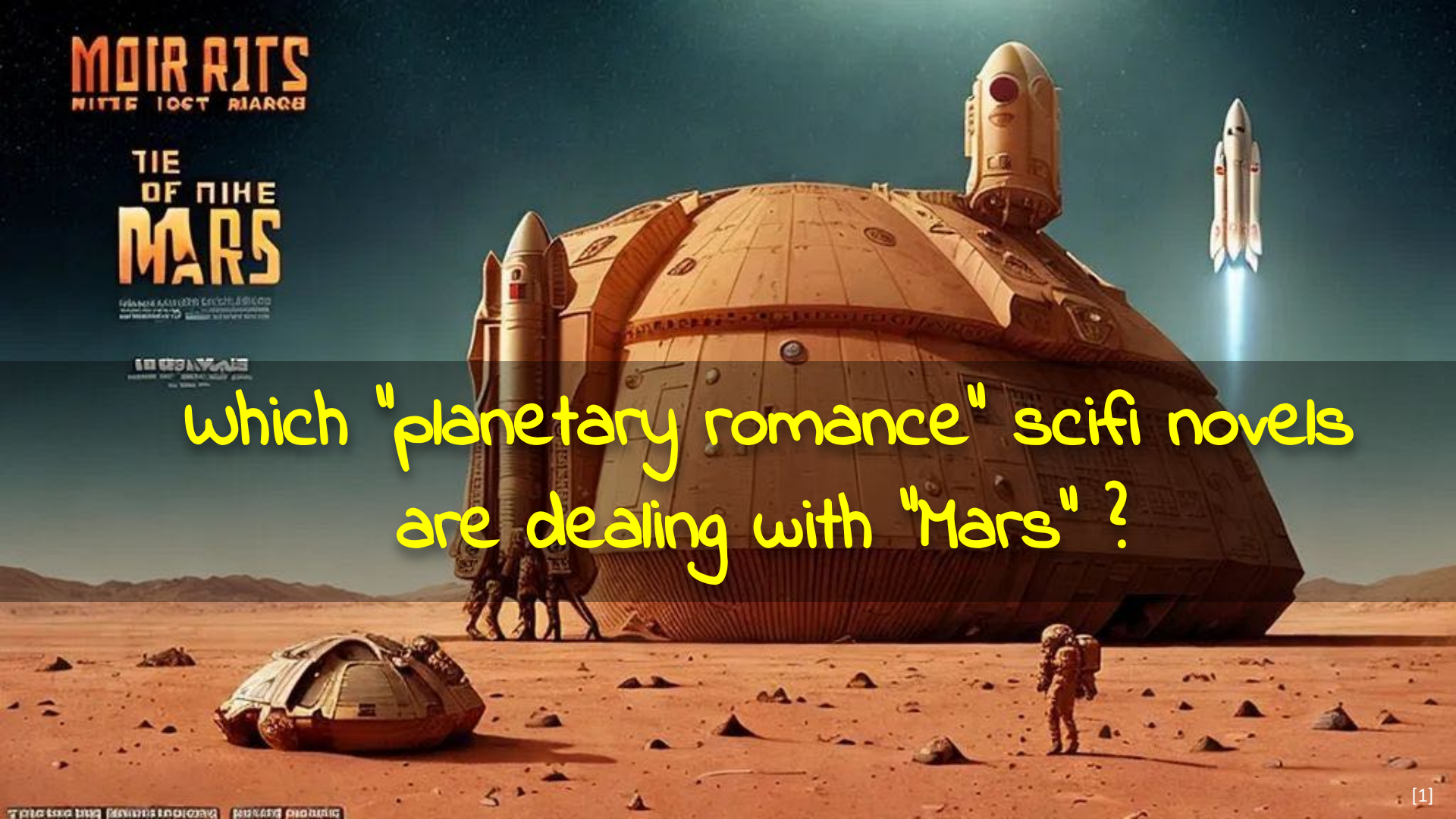
Prof. Dr. Harald Sack
FIZ Karlsruhe – Leibniz Institute for Information Infrastructure
AIFB – Karlsruhe Institute of Technology
Autumn 2023

Karlsruher Institut für Technologie

FIZ Karlsruhe
Leibniz-Institut für Informationsinfrastruktur

Knowledge Graphs

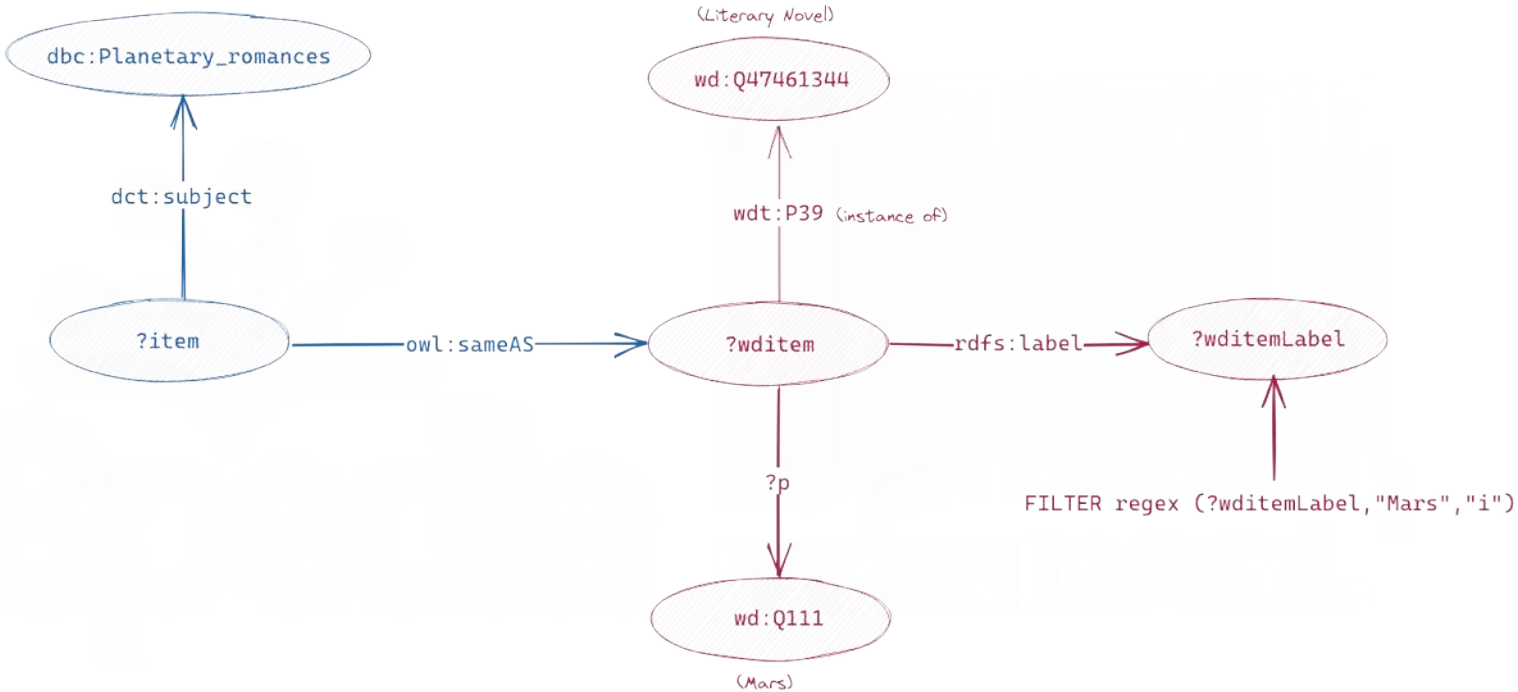
Lecture 3: Querying Knowledge Graphs with SPARQL


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

[1]

SPARQL Federated Queries

Example: which "planetary romances" scifi 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]

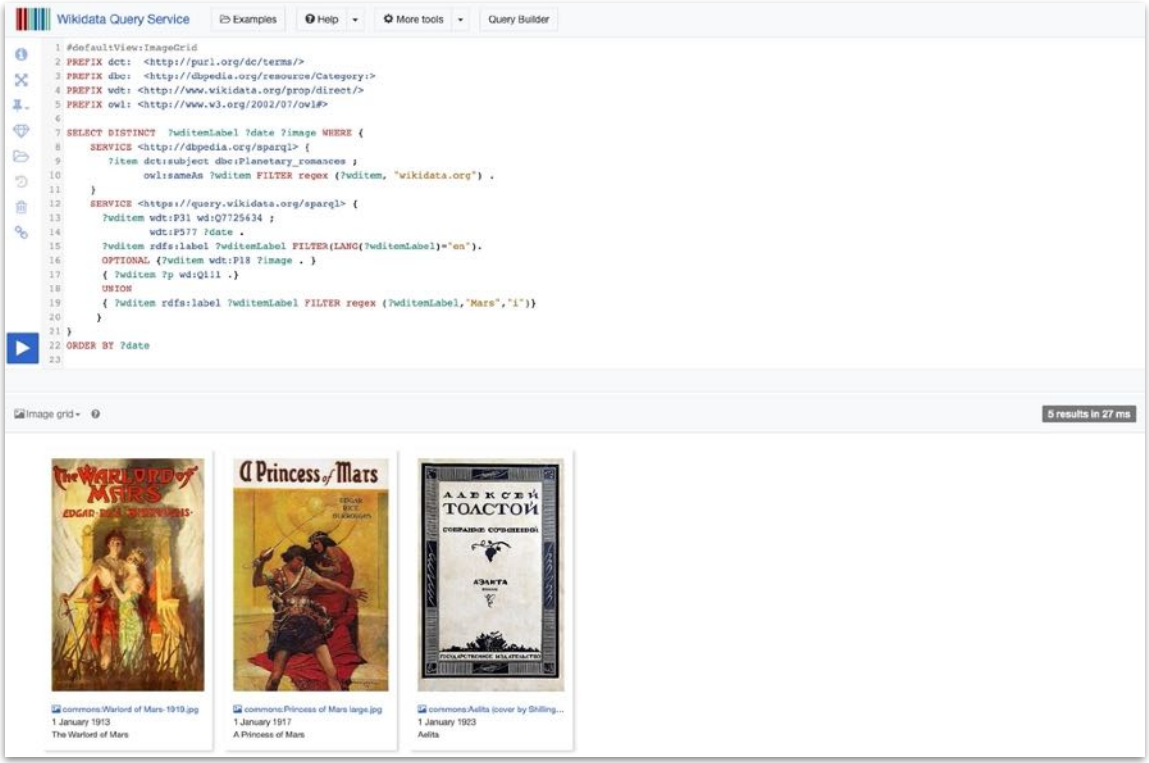
```
PREFIX dct:  <http://purl.org/dc/terms/>
PREFIX dbc:  <http://dbpedia.org/resource/Category:>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX owl: <http://www.w3.org/2002/07/owl#>

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" scifi novels are dealing with "Mars" ?



query SPARQL endpoint

SPARQL Variable Assignments

Example:

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

```
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 Writer
              wdt:P800 ?book .          # notableWork ?book
     ?book    wdt:P577 ?date ;          # ?book publicationDate ?date
              wdt:P136 wd:Q24925 .      # genre ScienceFiction
  BIND (YEAR(?date) AS ?year) FILTER (BOUND(?year))
  SERVICE wikibase:label { bd:serviceParam wikibase:language "en" }
} ORDER BY ?year
```

Binding a new variable

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

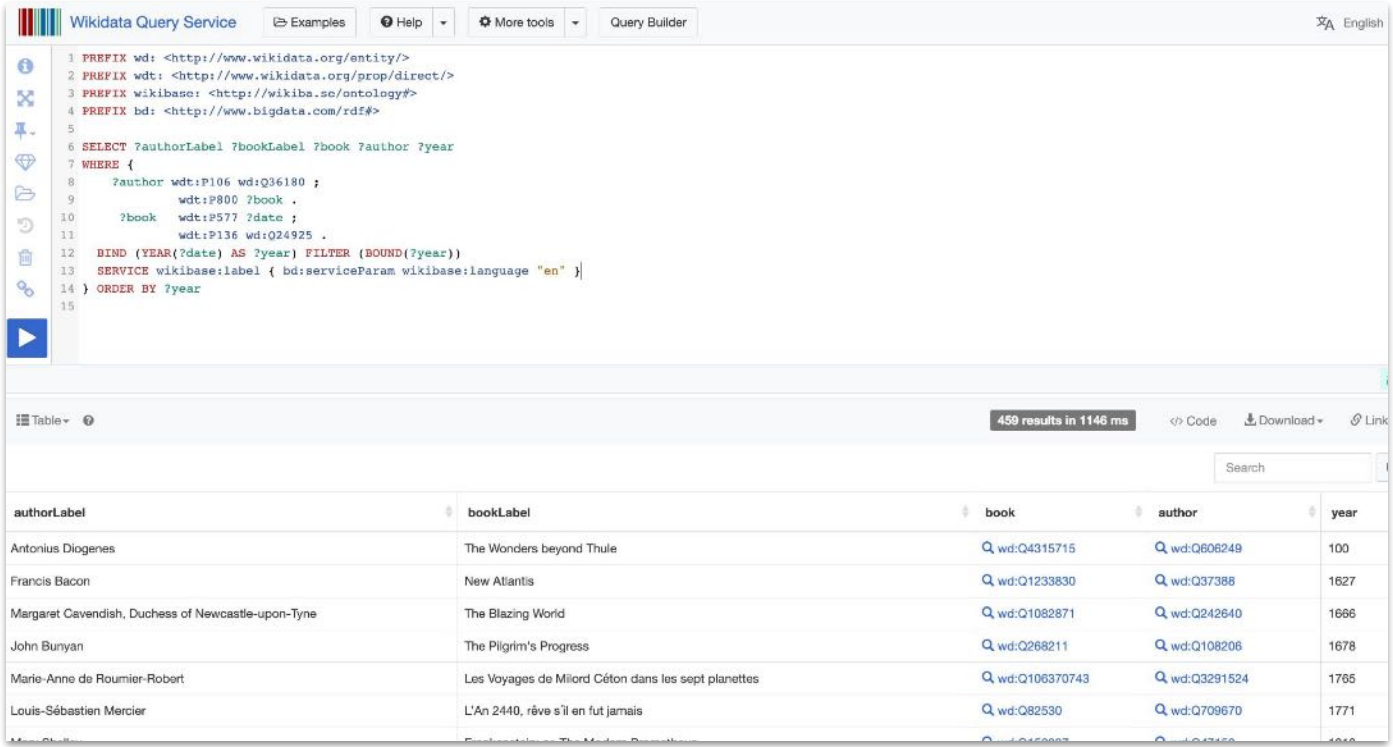
[3]

query SPARQL endpoint

SPARQL Variable Assignments

Example:

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



WIKIDATA [3]

query SPARQL endpoint

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

```
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" }
}
```

Table | 1 result in 513 ms | Code | Download | Link

bookcount	authorcount
526	181

WIKIDATA [3]

query SPARQL endpoint

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)
```

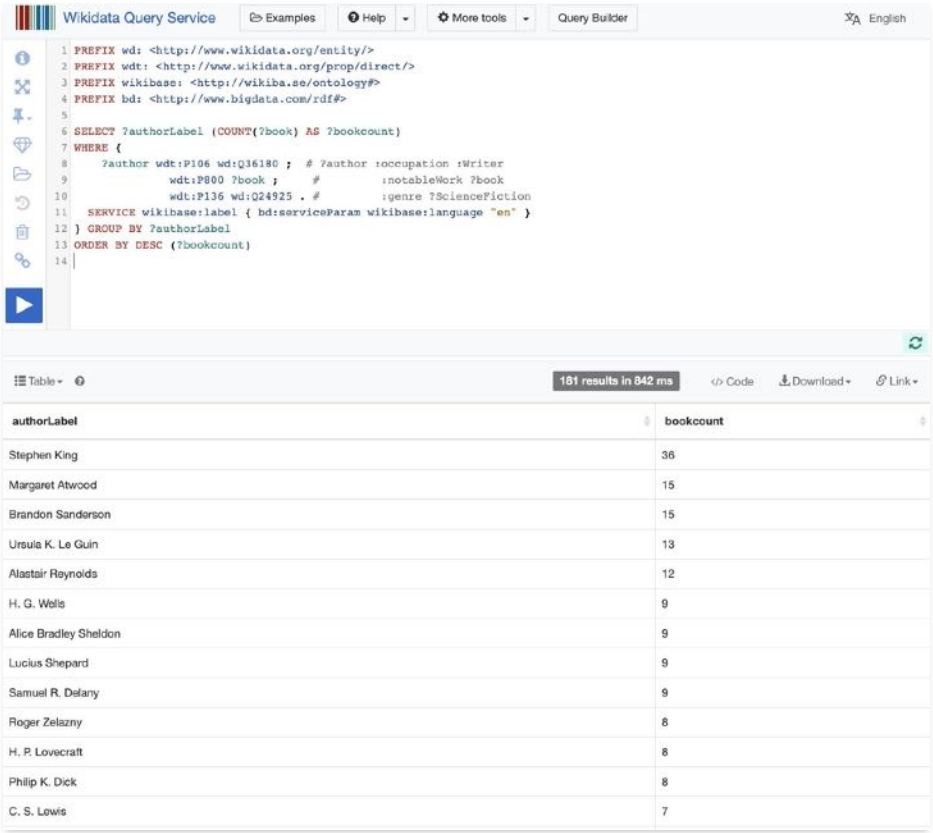
aggregate function

- 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 [3]

query SPARQL endpoint

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

Next Lecture...

[4]

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

[4]