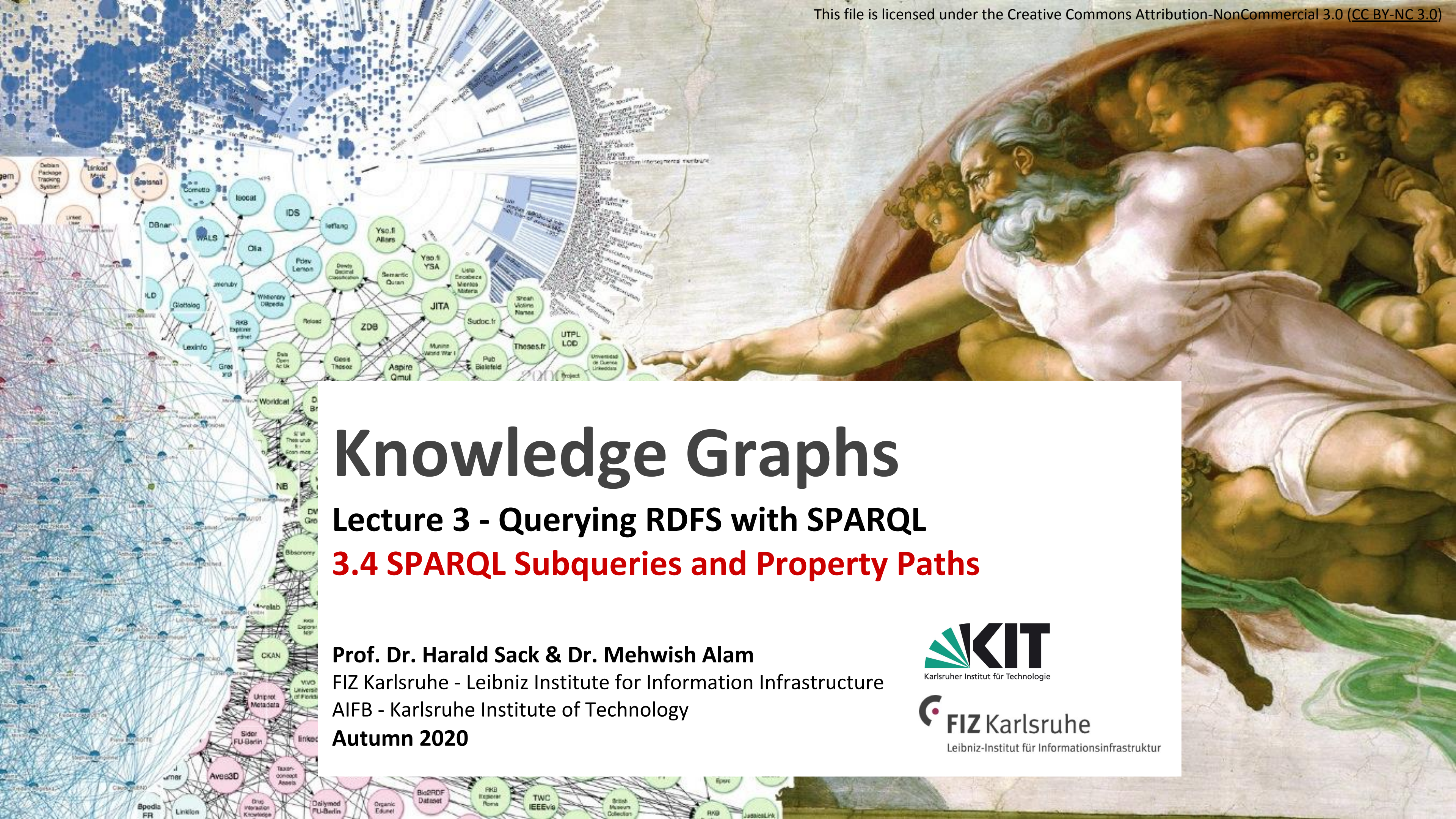


Knowledge Graphs

Lecture 3 - Querying RDFS with SPARQL

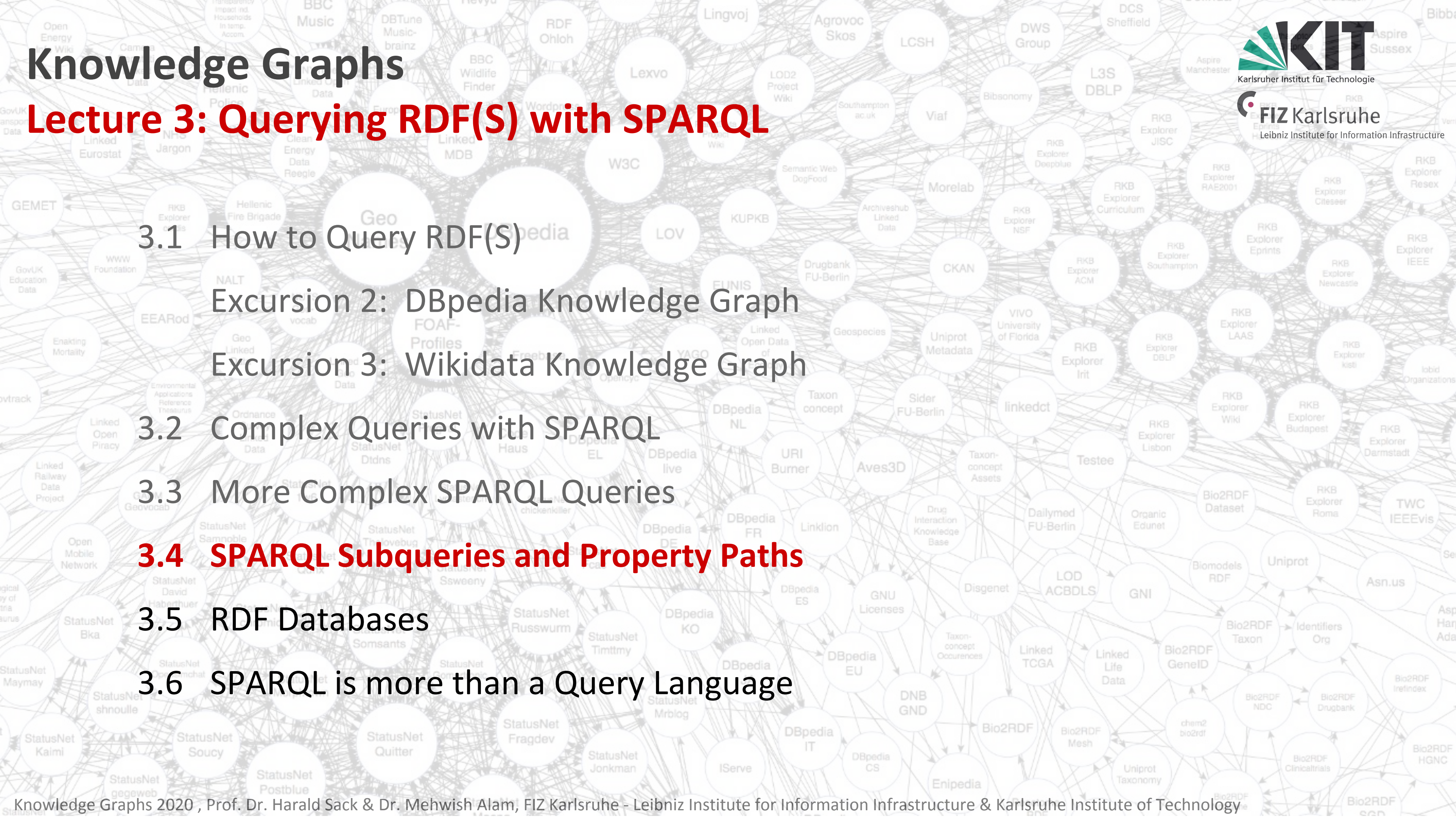
3.4 SPARQL Subqueries and Property Paths

Prof. Dr. Harald Sack & Dr. Mehwish Alam
FIZ Karlsruhe - Leibniz Institute for Information Infrastructure
AIFB - Karlsruhe Institute of Technology

Autumn 2020

KIT Karlsruher Institut für Technologie

FIZ Karlsruhe
Leibniz-Institut für Informationsinfrastruktur



Knowledge Graphs

Lecture 3: Querying RDF(S) with SPARQL

3.1 How to Query RDF(S)

Excursion 2: DBpedia Knowledge Graph

Excursion 3: Wikidata Knowledge Graph

3.2 Complex Queries with SPARQL

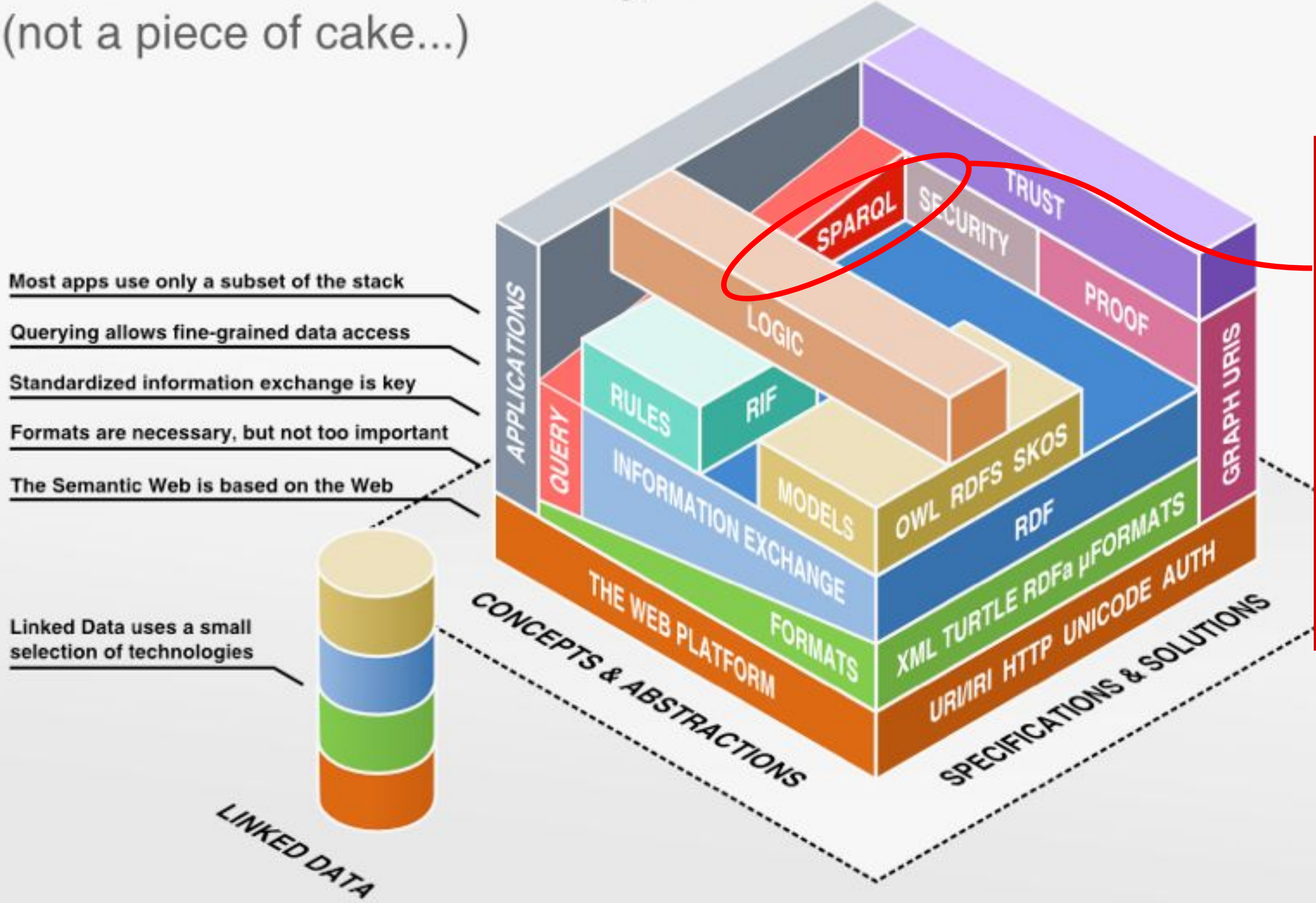
3.3 More Complex SPARQL Queries

3.4 SPARQL Subqueries and Property Paths

3.5 RDF Databases

3.6 SPARQL is more than a Query Language

The Semantic Web Technology Stack
(not a piece of cake...)



Most apps use only a subset of the stack

Querying allows fine-grained data access

Standardized information exchange is key

Formats are necessary, but not too important

The Semantic Web is based on the Web

Linked Data uses a small selection of technologies

USE
SPARQL
and
QUERY
ON

[1,2]

SPARQL Subqueries

- Example: **what books** were written by the **30 most influential authors**?

```
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 ?influencerLabel ?bookLabel ?authorCount
WHERE {
  { SELECT ?influencer (COUNT(?author) AS ?authorCount)
    WHERE {
      ?author wdt:P106 wd:Q36180 ;
              wdt:P737 ?influencer .
      ?influencer wdt:P106 wd:Q36180 .
    }  GROUP BY ?influencer ORDER BY DESC(?authorCount)
       LIMIT 30
  }
  ?influencer wdt:P800 ?book .
  SERVICE wikibase:label { bd:serviceParam wikibase:language "en" }
} ORDER BY DESC(?authorCount) ?influencerLabel
```

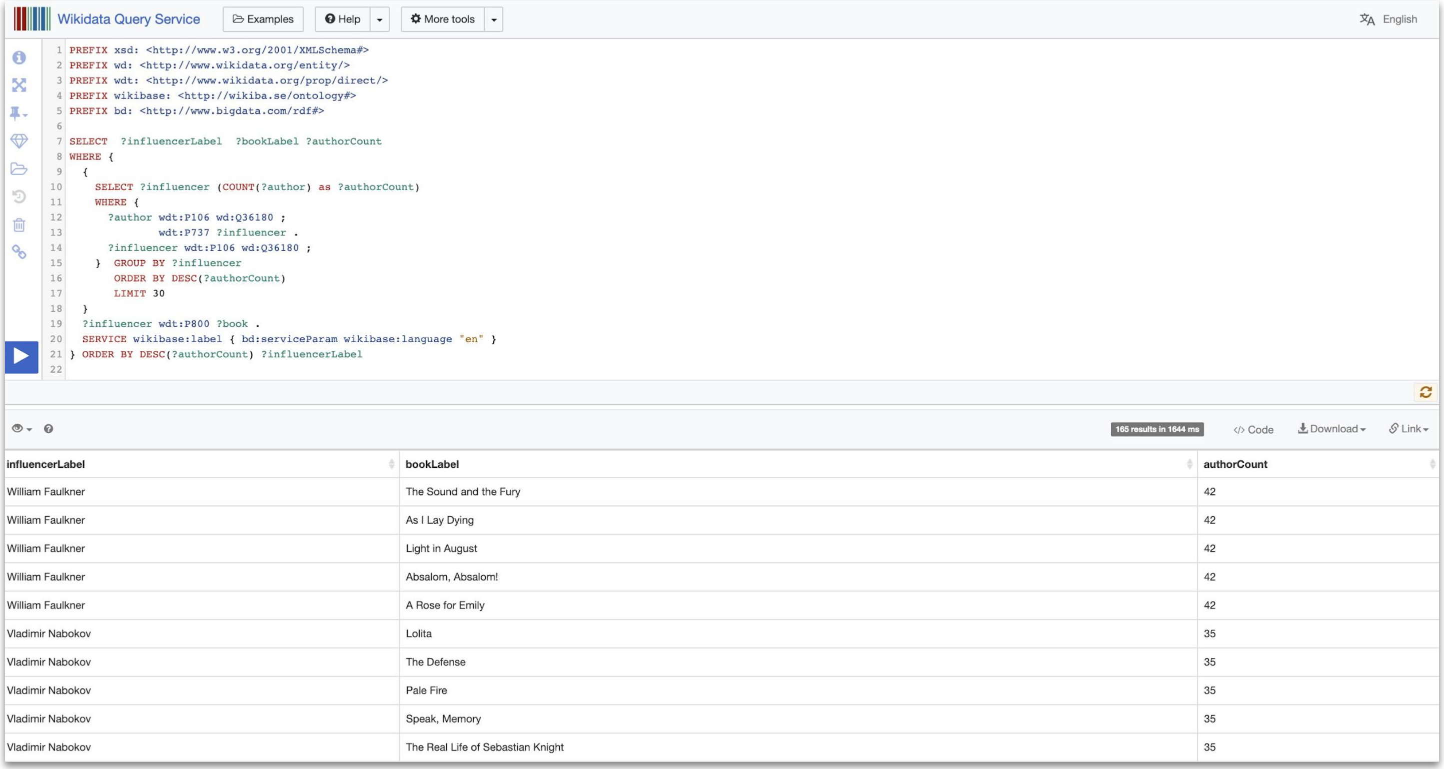
inner Subquery

- Subqueries are a way to embed SPARQL queries within other queries
- Result is achieved by first evaluating the inner query

query SPARQL endpoint

SPARQL Subqueries

- Example: **what books** were written by the **30 most influential authors**?



```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
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  ?influencerLabel  ?bookLabel ?authorCount
WHERE {
  {
    SELECT ?influencer (COUNT(?author) as ?authorCount)
    WHERE {
      ?author wdt:P106 wd:Q36180 ;
              wdt:P737 ?influencer .
      ?influencer wdt:P106 wd:Q36180 ;
    }  GROUP BY ?influencer
       ORDER BY DESC(?authorCount)
       LIMIT 30
  }
  ?influencer wdt:P800 ?book .
  SERVICE wikibase:label { bd:serviceParam wikibase:language "en" }
} ORDER BY DESC(?authorCount) ?influencerLabel

```

influencerLabel	bookLabel	authorCount
William Faulkner	The Sound and the Fury	42
William Faulkner	As I Lay Dying	42
William Faulkner	Light in August	42
William Faulkner	Absalom, Absalom!	42
William Faulkner	A Rose for Emily	42
Vladimir Nabokov	Lolita	35
Vladimir Nabokov	The Defense	35
Vladimir Nabokov	Pale Fire	35
Vladimir Nabokov	Speak, Memory	35
Vladimir Nabokov	The Real Life of Sebastian Knight	35

query SPARQL endpoint

SPARQL Subqueries

- Example: **what books** were written by the **30 most influential authors**?

```
#defaultView:Graph
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
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 ?influencer ?influencerLabel ?book ?bookLabel ?authorCount
WHERE {
  { SELECT ?influencer (COUNT(?author) AS ?authorCount)
    WHERE {
      ?author wdt:P106 wd:Q36180 ;
              wdt:P737 ?influencer .
      ?influencer wdt:P106 wd:Q36180 .
     }  GROUP BY ?influencer ORDER BY DESC(?authorCount)
        LIMIT 30
  }
  ?influencer wdt:P800 ?book .
  SERVICE wikibase:label { bd:serviceParam wikibase:language "en" }
} ORDER BY DESC(?authorCount) ?influencerLabel
```

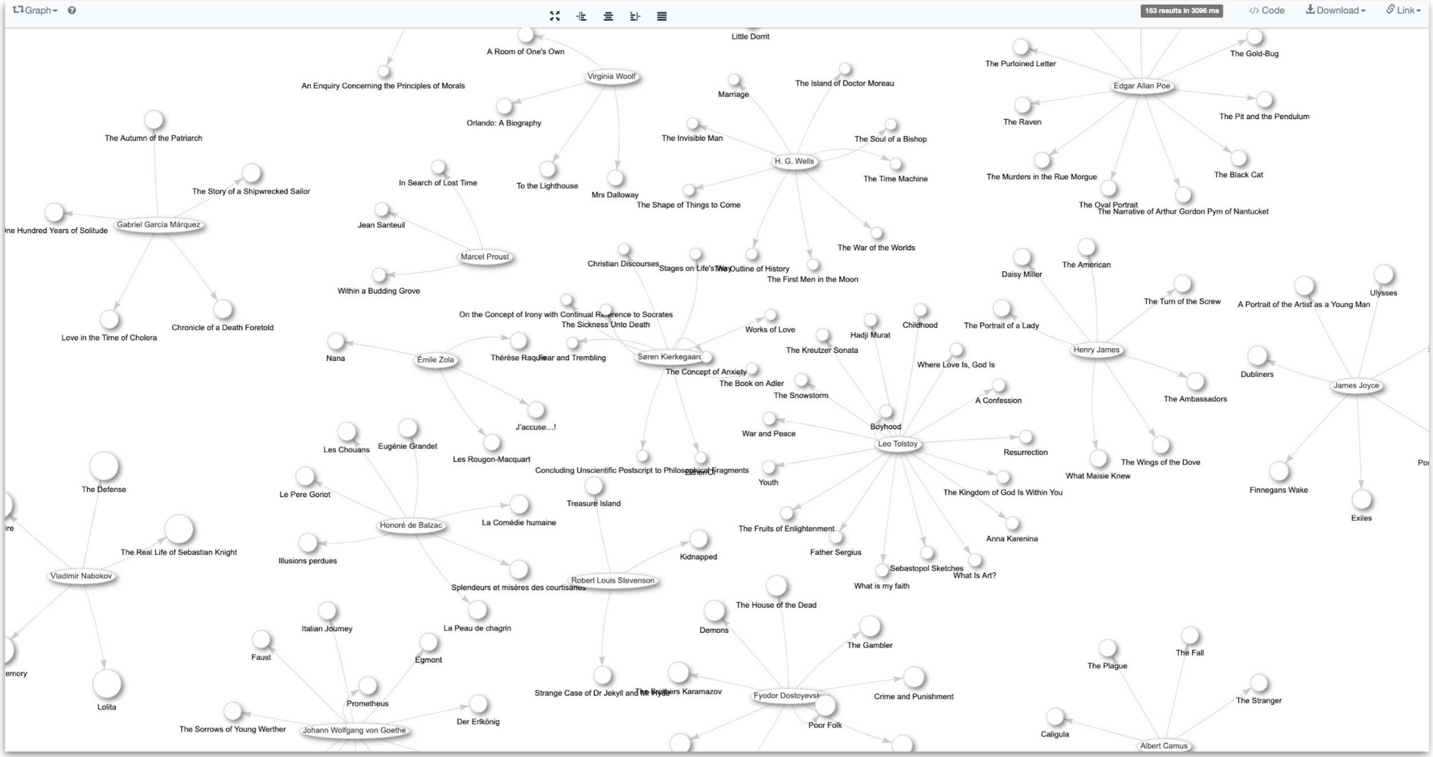
- With the Wikidata SPARQL endpoint, we are able to display the result as a graph

WIKIDATA

[query SPARQL endpoint]

SPARQL Subqueries

- Example: **what books** were written by the **30 most influential authors**?



- With the Wikidata SPARQL endpoint, we are able to display the result as a **graph**

query SPARQL endpoint

SPARQL Property Paths

- A **property path** is a possible route through an RDF graph between two graph nodes.
  - trivial case: property path of length 1, i.e. a triple pattern
  - **alternatives**: match one or both possibilities

```
{ :book1 dc:title|rdfs:label ?displayString . }
```

  - **sequence**: property path of length >1

```
{ :alice foaf:knows/foaf:knows/foaf:name ?name . }
```

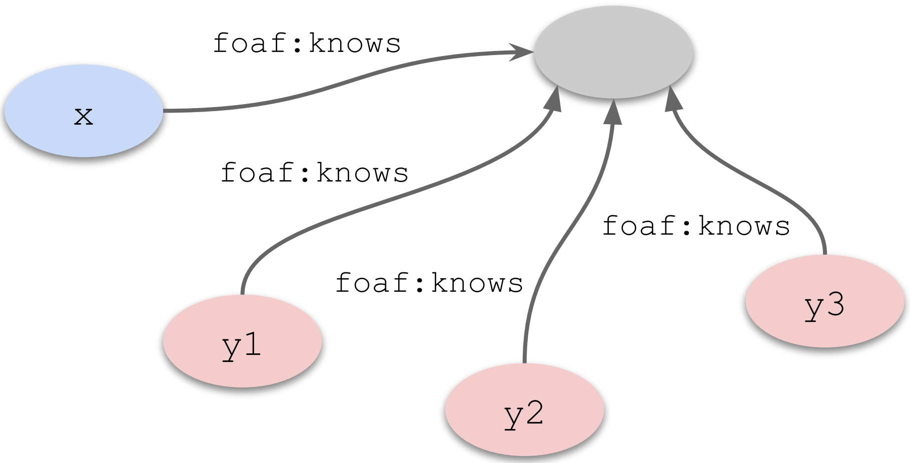
  - **inverse property paths**: reversing the direction of the triple

```
{ ?x foaf:mbox <mailto:alice@example> . }
                   =
{ <mailto:alice@example> ^foaf:mbox ?x . }
```

SPARQL Property Paths

- **inverse path sequences**

```
{ ?x foaf:knows/^foaf:knows ?y .
     FILTER (?x != ?y) }
```



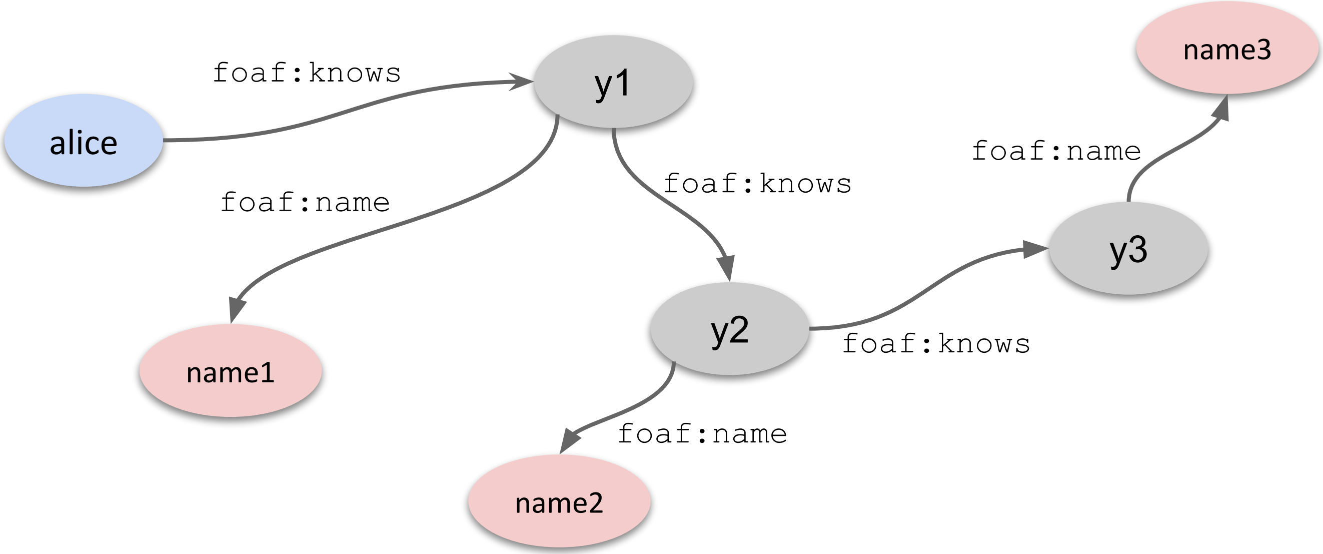
- Example:
  who else besides me knows the people I know?

SPARQL Property Paths

- **arbitrary length match**

```
{ :alice foaf:knows+/foaf:name ?name .      }
```

- Example:
  what are the names of all persons I know and those who they know (and so on...) ?



SPARQL Property Paths

- **inverse path sequences**

```
{ ?x foaf:knows/^foaf:knows ?y .
     FILTER (?x != ?y) }
```

- **arbitrary length match**

```
{ :alice foaf:knows+/foaf:name ?name .       }
```

- **negated property paths**

```
{ ?x !(rdf:type|^rdf:type) ?y . }
```

SPARQL Property Paths

- Example: who else was influenced by the influencers of George Orwell?

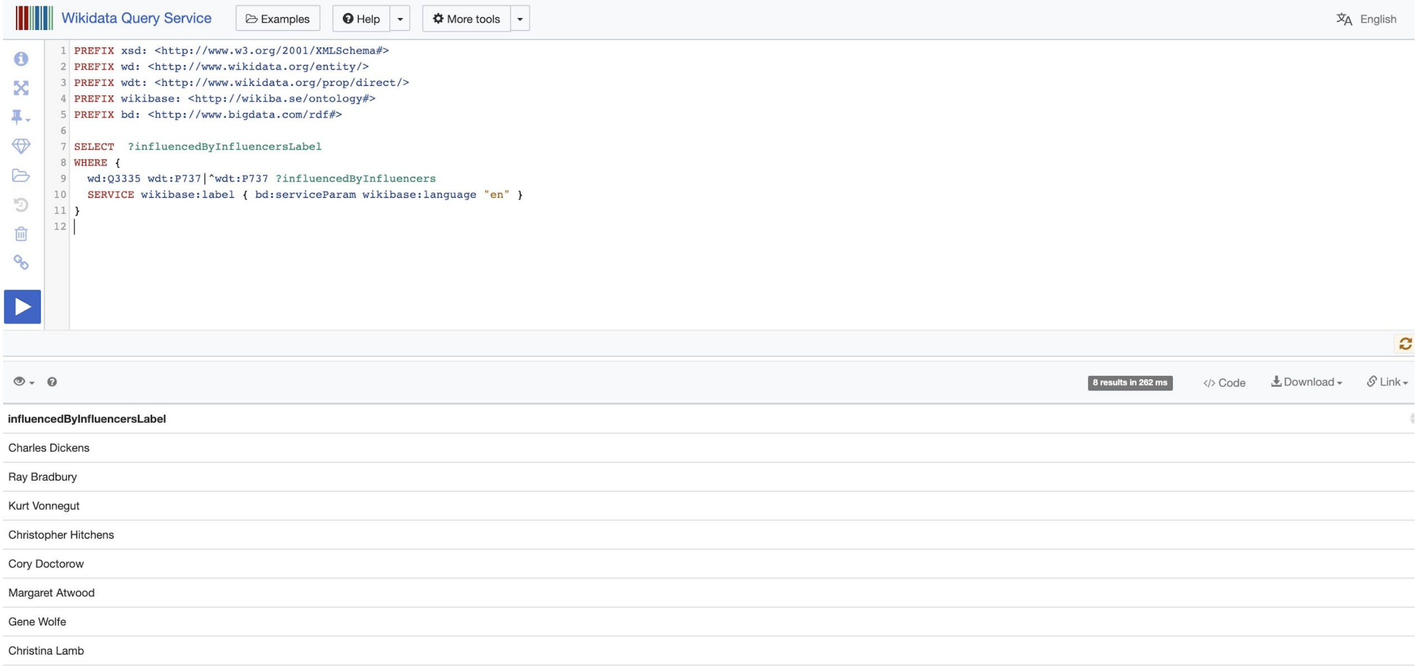
```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX wikibase: <http://wikiba.se/ontology#>
PREFIX bd: <http://www.bigdata.com/rdf#>

SELECT  ?influencedByInfluencersLabel
WHERE {
  wd:Q3335 wdt:P737|^wdt:P737 ?influencedByInfluencers
  SERVICE wikibase:label { bd:serviceParam wikibase:language "en" }
}
```

query SPARQL endpoint

SPARQL Property Paths

- Example: who else was influenced by the influencers of George Orwell?



```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
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  ?influencedByInfluencersLabel
WHERE {
  wd:Q3335 wdt:P737|^wdt:P737 ?influencedByInfluencers
  SERVICE wikibase:label { bd:serviceParam wikibase:language "en" }
}
```

influencedByInfluencersLabel
Charles Dickens
Ray Bradbury
Kurt Vonnegut
Christopher Hitchens
Cory Doctorow
Margaret Atwood
Gene Wolfe
Christina Lamb

query SPARQL endpoint

RDF Databases

Next Lecture...

[4]

Picture References:

- [1] Benjamin Nowack, *The Semantic Web - Not a Piece of cake…*, at bnode.org, 2009-07-08 , [CC BY 3.0] http://bnode.org/blog/2009/07/08/the-semantic-web-not-a-piece-of-cake
- [2] British Crown vector illustration, publicdomainvectors.org, [Public Domain] https://publicdomainvectors.org/en/free-clipart/British-Crown-vector-illustration/12150.html
- [3] WIkidata Logo, Planemad, [Public Domain] https://commons.wikimedia.org/wiki/File:Wikidata-logo-en.svg
- [4] Ernst Haeckel, Kunstformen der Natur (1904), plate 31: Cyrtoidea, [Public Domain] https://commons.wikimedia.org/wiki/File:Haeckel_Cyrtoidea.jpg