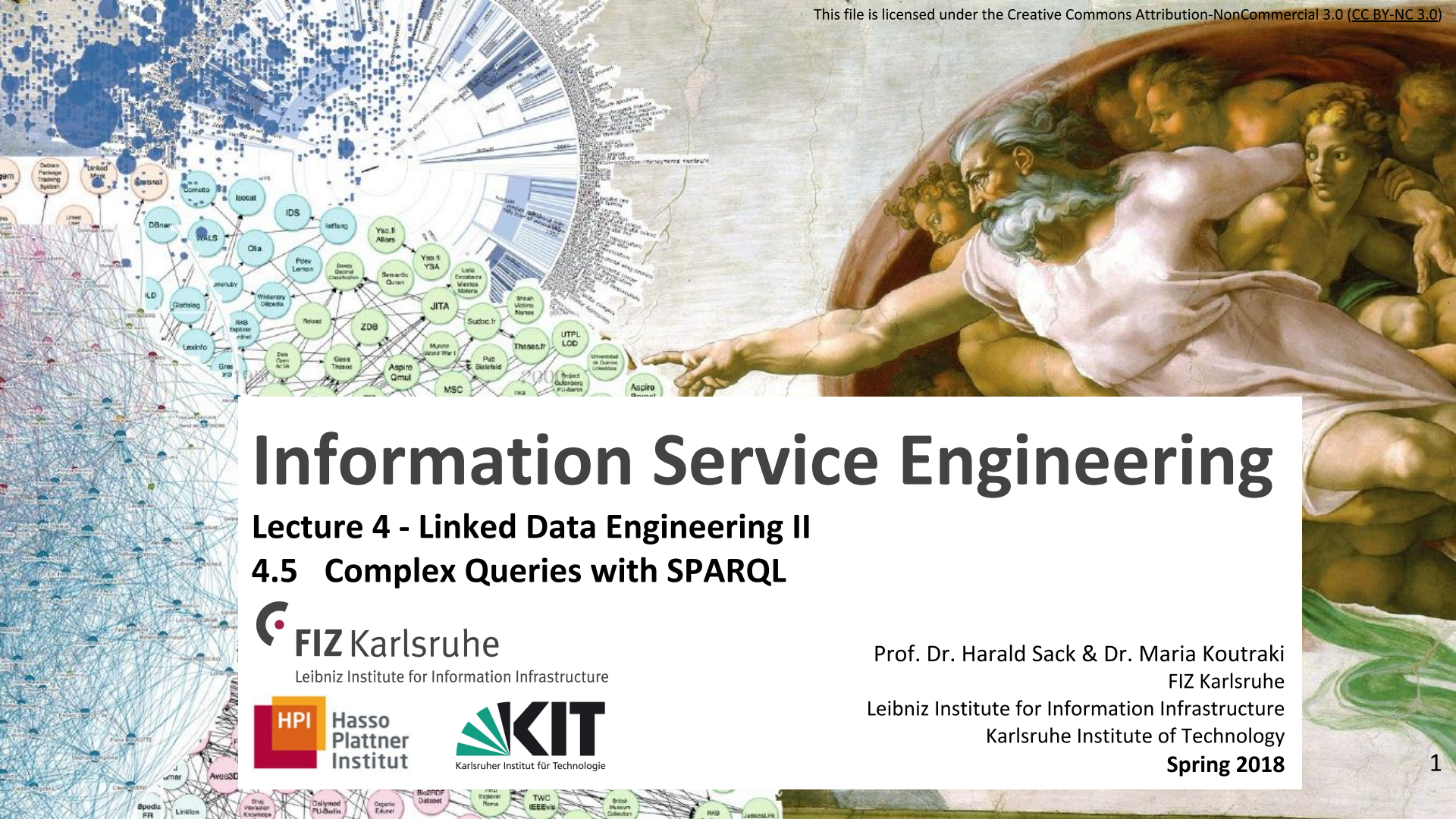




Information Service Engineering

Lecture 4 - Linked Data Engineering II

4.5 Complex Queries with SPARQL



FIZ Karlsruhe

Leibniz Institute for Information Infrastructure



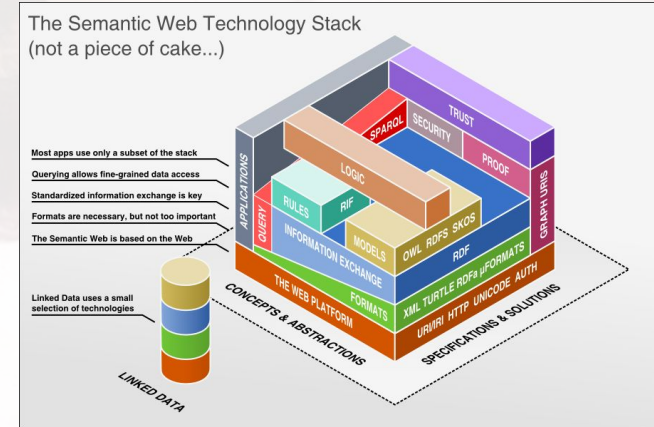
Prof. Dr. Harald Sack & Dr. Maria Koutraki
FIZ Karlsruhe

Leibniz Institute for Information Infrastructure
Karlsruhe Institute of Technology
Spring 2018

Information Service Engineering

Lecture 4: Linked Data Engineering II

- 4.1 Linked Data Principles
- 4.2 Web Ontology Language OWL
- 4.3 Web Ontology Language OWL II
- 4.4 How to Query RDF with SPARQL
- 4.5 Complex Queries with SPARQL**
- 4.6 More Complex SPARQL Queries
- 4.7 Linked Data Knowledge Graphs



SPARQL Filter Constraints

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.5 Complex Queries with SPARQL

- Example: Filter results **only for English labels**

```
PREFIX : <http://dbpedia.org/resource/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?author_name ?title ?pages
FROM <http://dbpedia.org/>
WHERE {
    ?author rdf:type dbo:Writer .
    ?author rdfs:label ?author_name .
    FILTER (LANG(?author_name)="en")
    ?author dbo:notableWork ?work .
    ?work dbo:numberOfPages ?pages .
    FILTER (?pages > 500)
    ?work rdfs:label ?title .
    FILTER (LANG(?title)="en")
} LIMIT 100
```

More SPARQL Operators

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.5 Complex Queries with SPARQL

- **Logical connectives** `&&` and `||` for `xsd:boolean`
- **Comparison operators** `=`, `!=`, `<`, `>`, `<=`, and `>=` for numeric datatypes, `xsd:dateTime`, `xsd:string`, and `xsd:boolean`
- **Comparison operators** `=` and `!=` for other datatypes
- **Arithmetic operators** `+`, `-`, `*`, and `/` for numeric datatypes
- and in addition:
 - `REGEX(String,Pattern)` or `REGEX(String,Pattern,Flags)`
 - `sameTERM(A,B)`
 - `langMATCHES(A,B)`

SPARQL RegEx Filter Constraints

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.5 Complex Queries with SPARQL

- Example: Book titles that **contain the word "love"**

```
PREFIX : <http://dbpedia.org/resource/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?author_name ?title
FROM <http://dbpedia.org/>
WHERE {
  ?author rdf:type dbo:Writer .
  ?author rdfs:label ?author_name .
  FILTER (LANG(?author_name)="en")
  ?author dbo:notableWork ?work .
  ?work rdfs:label ?title .
  FILTER (LANG(?title)="en")
  FILTER REGEX (?title, "love", "i")
} LIMIT 100
```

[query SPARQL endpoint](#)

string

regular
expression

flags

SPARQL RegEx Filter Constraints

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.5 Complex Queries with SPARQL

- Example: Book titles that start or end with the word “love”

```
PREFIX : <http://dbpedia.org/resource/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?author_name ?title
FROM <http://dbpedia.org/>
WHERE {
    ?author rdf:type dbo:Writer .
    ?author rdfs:label ?author_name .
    FILTER (LANG(?author_name)="en")
    ?author dbo:notableWork ?work .
    ?work rdfs:label ?title .
    FILTER (LANG(?title)="en")
    FILTER REGEX (?title, "^love|love$", "i")
} LIMIT 100
```

regular
expression

[query SPARQL endpoint](#)

SPARQL Optional Queries

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.5 Complex Queries with SPARQL

- Example: Retrieve also the German book title, **if available**

```
PREFIX : <http://dbpedia.org/resource/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?author_name ?en_title ?de_title
FROM <http://dbpedia.org/>
WHERE {
    ?author rdf:type dbo:Writer .
    ?author rdfs:label ?author_name .
    FILTER (LANG(?author_name)="en")
    ?author dbo:notableWork ?work .
    ?work rdfs:label ?en_title .
    FILTER (LANG(?en_title)="en")
    OPTIONAL { ?work rdfs:label ?de_title .
                FILTER (LANG(?de_title)="de")
            }
} LIMIT 100
```

- The keyword **OPTIONAL** selects optional elements from the RDF graph
- complies to a Left Outer Join

*optional
constraint*

[query SPARQL endpoint](#)

SPARQL Alternative Queries via UNION

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.5 Complex Queries with SPARQL

- Example: Retrieve all influencers of and people influenced by **Miguel_de_Cervantes**

```
PREFIX : <http://dbpedia.org/resource/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>

SELECT ?influencer ?influenced
FROM <http://dbpedia.org/>
WHERE {
    { :Miguel_de_Cervantes dbo:influenced ?influenced . }
    UNION
    { :Miguel_de_Cervantes dbo:influencedBy ?influencer . }
}
```

*logical
disjunction*

- The keyword **UNION** allows for alternatives (logical disjunction)

[query SPARQL endpoint](#)

SPARQL Negated Queries

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.5 Complex Queries with SPARQL

- Example: Retrieve authors that **don't have an entry for "notable work"**

```
PREFIX : <http://dbpedia.org/resource/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>

SELECT ?author
FROM <http://dbpedia.org/>
WHERE {
    ?author rdf:type dbo:Writer .
    FILTER NOT EXISTS {?author dbo:notableWork ?work .}
} LIMIT 100
```

*filter query
result for
existency*

- SPARQL 1.1 offers several variants for negation:
 - **FILTER NOT EXISTS**
 - **MINUS**
 - **! BOUND ()**

[query SPARQL endpoint](#)

SPARQL Federated Queries

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.5 Complex Queries with SPARQL

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

```
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>

SELECT ?item ?childName WHERE {
  SERVICE <http://dbpedia.org/sparql> {
    ?item dbo:occupation dbr:Poet ;
    owl:sameAs ?wdItem FILTER regex (?wdItem, "wikidata.org") .
  }
  SERVICE <https://query.wikidata.org/sparql> {
    ?wdItem wdt:P40 ?child .
    ?child rdfs:label ?childName FILTER (LANG(?childName)="en").
  }
}

ORDER BY ?item
```

- Example: connect the **DBpedia** with **Wikidata**
- only possible, if SPARQL endpoints permit federation



More Complex Queries with SPARQL

Next Lecture...

Picture References:

- P.11: Albrecht Dürer, Stag Beetle (1505), [https://commons.wikimedia.org/wiki/File:Albrecht_D%C3%BCrer_-_Hirschk%C3%A4fer_\(1505\).jpg](https://commons.wikimedia.org/wiki/File:Albrecht_D%C3%BCrer_-_Hirschk%C3%A4fer_(1505).jpg)