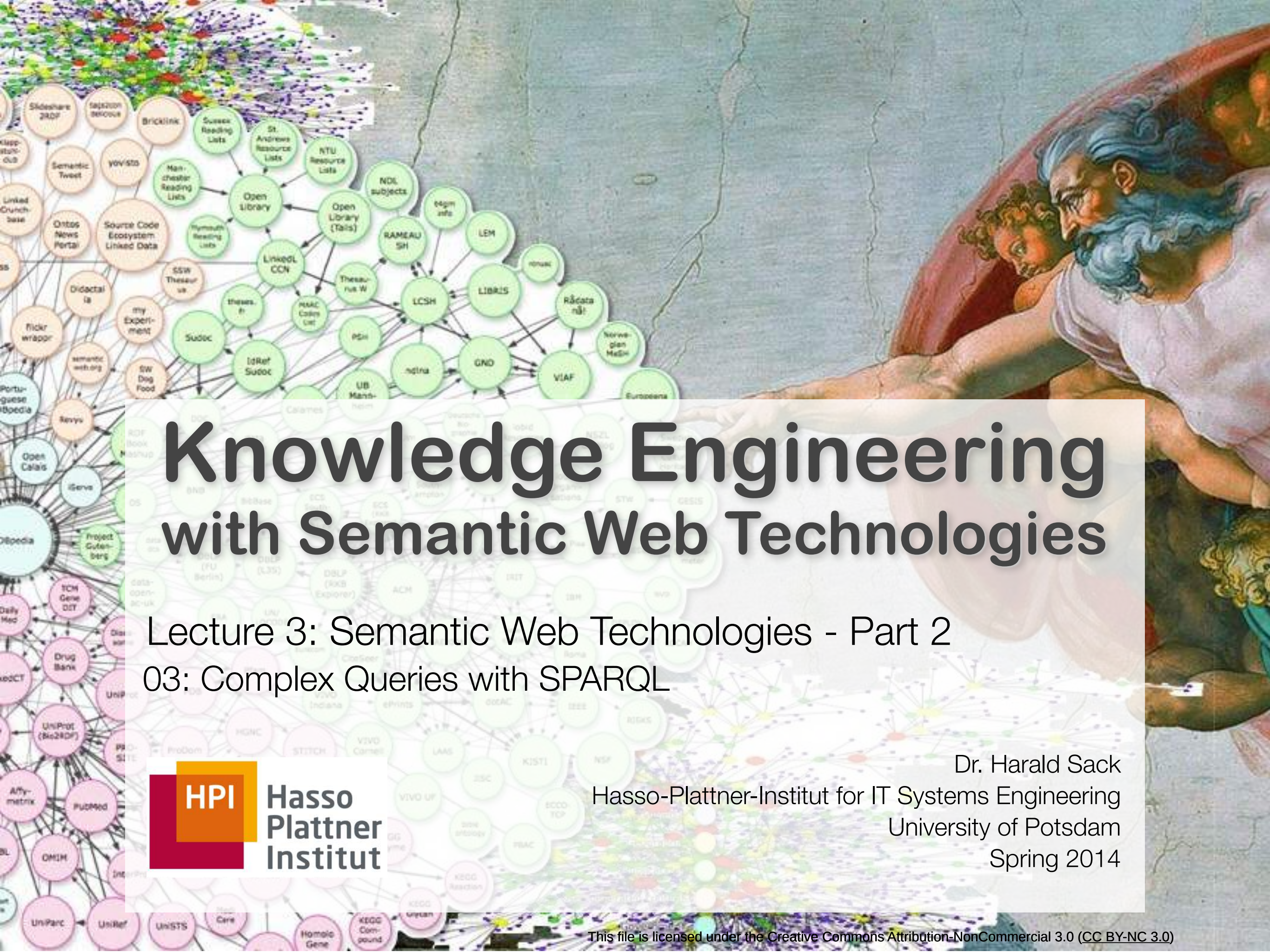


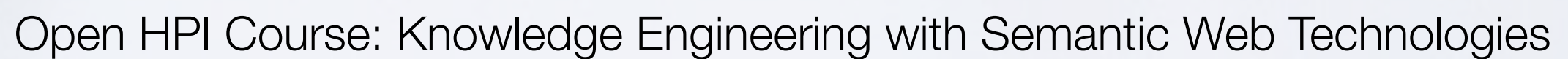


Knowledge Engineering with Semantic Web Technologies

Lecture 3: Semantic Web Technologies - Part 2 03: Complex Queries with SPARQL



Dr. Harald Sack
Hasso-Plattner-Institut for IT Systems Engineering
University of Potsdam
Spring 2014



SPARQL



03 Complex Queries with SPARQL

Open HPI Course: Knowledge Engineering with Semantic Web Technologies
Lecture 03: Semantic Web Technologies Pt.2

```
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 dbpedia-owl: <http://dbpedia.org/ontology/>
SELECT ?author_name ?title ?pages
FROM <http://dbpedia.org/>
WHERE {
    ?author rdf:type dbpedia-owl:Writer .
    ?author rdfs:label ?author_name .
    ?author dbpedia-owl:notableWork ?work .
    ?work dbpedia-owl:numberOfPages ?pages
    FILTER (?pages > 500) .
    ?work rdfs:label ?title .
} LIMIT 100
```

specifies constraints for the result

- FILTER expressions contain operators and functions
- FILTER can NOT assign/create new values

SPARQL Unary Operators in Constraints

Operator	Type(A)	Result Type
!A	xsd:boolean	xsd:boolean
+A	numeric	numeric
-a	numeric	numeric
BOUND(A)	variable	xsd:boolean
isURI(A)	RDF term	xsd:boolean
isBLANK(A)	RDF term	xsd:boolean
isLITERAL(A)	RDF term	xsd:boolean
STR(A)	literal/URI	simple literal
LANG(A)	literal	simple literal
DATATYPE(A)	literal	URI

- 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 dbpedia-owl: <http://dbpedia.org/ontology/>
SELECT ?author_name ?title ?pages
FROM <http://dbpedia.org/>
WHERE {
    ?author rdf:type dbpedia-owl:Writer .
    ?author rdfs:label ?author_name
    FILTER (LANG(?author_name)="en") .
    ?author dbpedia-owl:notableWork ?work .
    ?work dbpedia-owl:numberOfPages ?pages
    FILTER (?pages > 500) .
    ?work rdfs:label ?title
    FILTER (LANG(?title)="en") .
} LIMIT 100
```

More SPARQL Operators

- 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

- 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 dbpedia-owl: <http://dbpedia.org/ontology/>
SELECT ?author_name ?title ?pages
FROM <http://dbpedia.org/>
WHERE {
    ?author rdf:type dbpedia-owl:Writer .
    ?author rdfs:label ?author_name
    FILTER (LANG(?author_name)="en").
    ?author dbpedia-owl:notableWork ?work .
    ?work dbpedia-owl:numberOfPages ?pages .
    ?work rdfs:label ?title
    FILTER (LANG(?title)="en")
    FILTER REGEX (?title, "love", "i").
} LIMIT 100
```

string (points to "?title")
regular expression (points to "love")
flags (points to "i")

- learn more about regular expressions at <http://regexone.com/>

SPARQL - Filter Constraint Evaluation

- SPARQL Filter Constraints are evaluated in 3-valued Logic
- truth values: **true**, **false**, and **error**

A	B	A B	A && B
T	T	T	T
T	F	T	F
F	T	T	F
F	F	F	F
T	E	T	E
E	T	T	E
F	E	E	F
E	F	E	F
E	E	E	E

binary operators

A	!A
T	F
F	T
E	E

unary operator

SPARQL - Optional Results

- Example: Give 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 dbpedia-owl: <http://dbpedia.org/ontology/>
SELECT ?author_name ?en_title ?de_title
FROM <http://dbpedia.org/>
WHERE {
    ?author rdf:type dbpedia-owl:Writer .
    ?author rdfs:label ?author_name
    FILTER (LANG(?author_name)="en").
    ?author dbpedia-owl: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
```

*optional
constraint*

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

SPARQL - Alternative Results via UNION

- Example: search all influencers of and people influenced by George Orwell

```
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 dbpedia-owl: <http://dbpedia.org/ontology/>
SELECT ?influencer ?influenced
FROM <http://dbpedia.org/>
WHERE {
  { :George_Orwell dbpedia-owl:influenced ?influenced . }
  UNION
  { :George_Orwell dbpedia-owl:influencedBy ?influencer . }
}
```

*logical
disjunction*

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

- Example: Search for 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 dbpedia-owl: <http://dbpedia.org/ontology/>
SELECT ?author_name
FROM <http://dbpedia.org/>
WHERE {
    ?author rdf:type dbpedia-owl:Writer .
    OPTIONAL {?author dbpedia-owl:notableWork ?work . }
    FILTER (!BOUND(?work)) .
} LIMIT 100
```

*no variable
binding*

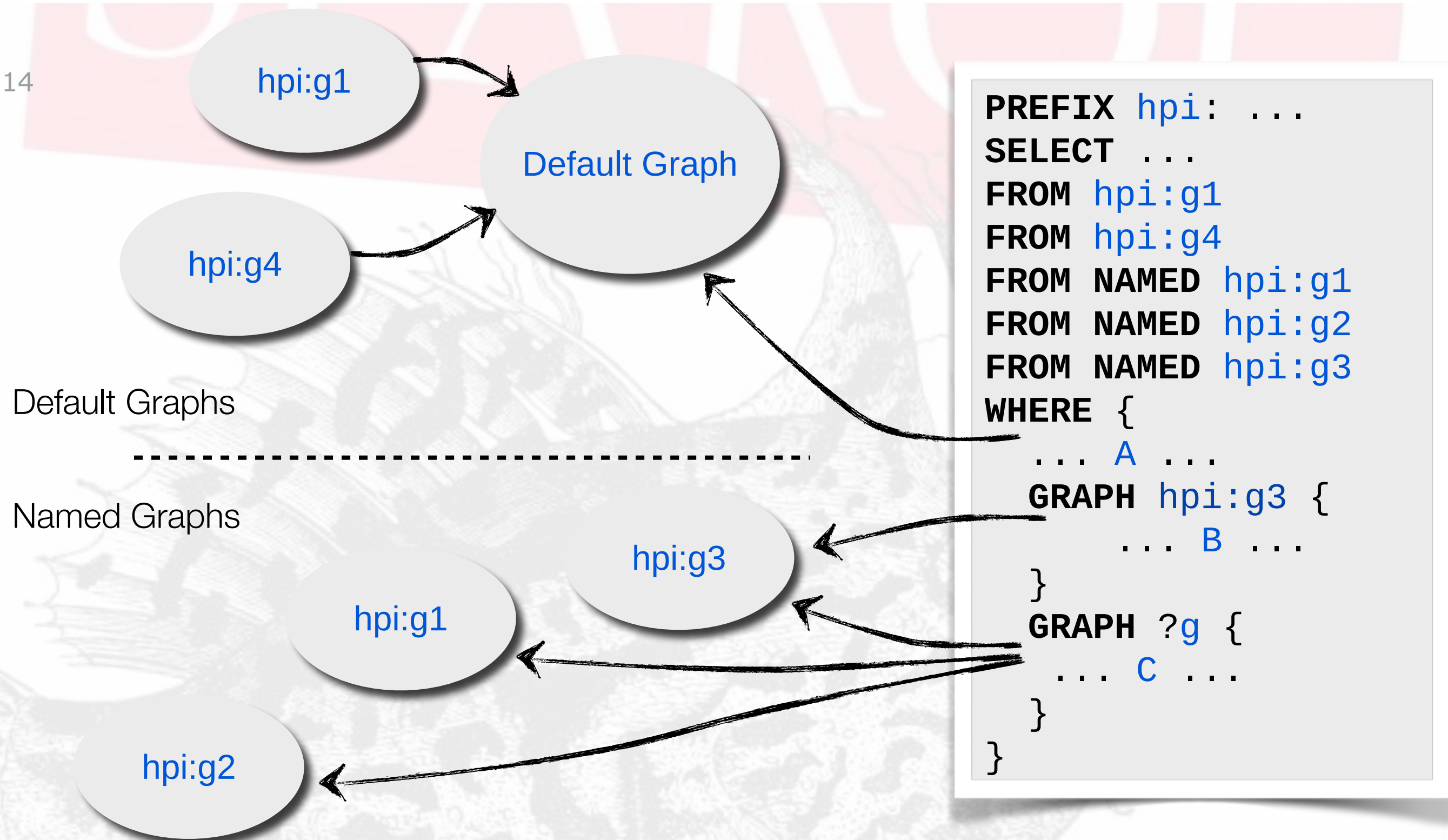
- **Negation** in SPARQL
- (complies to ,NOT EXISTS' in SQL)

- SPARQL queries are executed over an RDF dataset
 - one (or more) **default RDF graph** (FROM)
 - zero or more **named RDF graphs** (FROM NAMED, GRAPH)
- **Named Graphs** can explicitly addressed via the keyword **GRAPH** and the URI of the named graph

```
SELECT ...  
WHERE {  
  ...  
  GRAPH <http://example.org/graph1.rdf> {  
    ?x foaf:mbox ?mbox  
  }  
  ...  
}
```

SPARQL - RDF Graphs

14



- How to ask a SPARQL Endpoint which RDF graphs are available?

```
SELECT DISTINCT ?g  
WHERE {  
  GRAPH ?g { ?s ?p ?o . }  
}
```



04 Even More SPARQL – SPARQL 1.1

Open HPI Course: Knowledge Engineering with Semantic Web Technologies
Lecture 03: Semantic Web Technologies Pt.2