SPARQL

Antonis Loizou
(slides by
Rinke Hoekstra)

Today

- SPARQL - the RDF Query Language

Linked Data in the Wild

- BBC
  http://www.bbc.co.uk/programmes/b006mj59
  http://www.bbc.co.uk/programmes/b006mj59.rdf

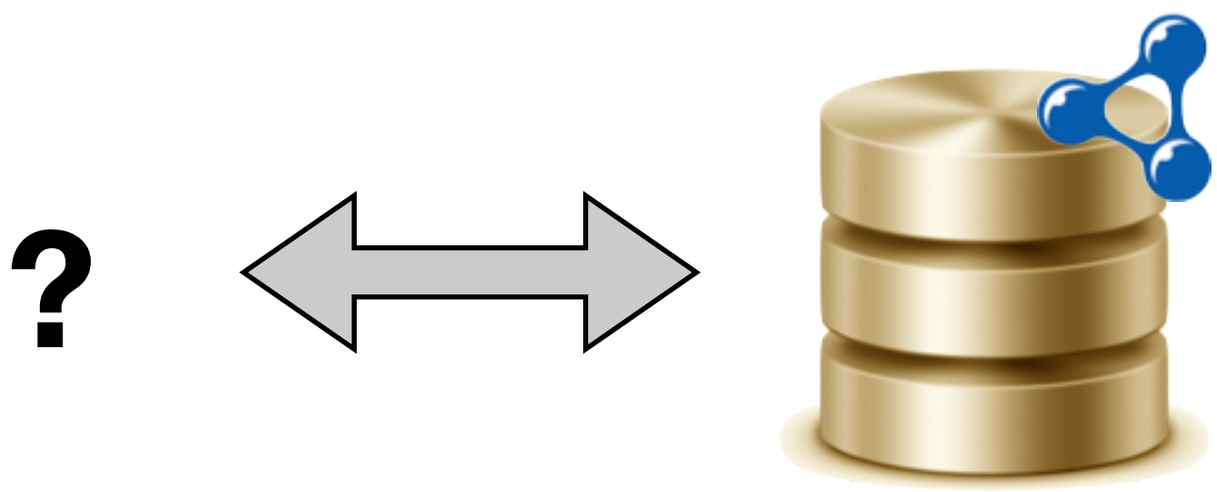
- MusicBrainz
  http://musicbrainz.org/artist/678d88b2-87b0-403b-b63d-5da7465aecc3

- Wikipedia & DBPedia
  http://en.wikipedia.org/wiki/Top_Gear_(2002_TV_series)
  http://live.dbpedia.org/page/Top_Gear_(2002_TV_series)
  http://live.dbpedia.org/data/Top_Gear_(2002_TV_series).n3

- UK Legislation
  http://www.legislation.gov.uk/ukpga/1998/29
  http://www.legislation.gov.uk/ukpga/1998/29/data.rdf

RDF Stores

- Provide a standard **REST API** for querying the store
- Queries are written using the **SPARQL** language
- The REST API of a triple store is called a **SPARQL endpoint**
- **SELECT, CONSTRUCT, INSERT, DELETE, ASK, DESCRIBE**



Triple Patterns

- A **graph pattern** consists of multiple **triple patterns**

- A **triple pattern** is a triple with **zero** or **more variables**
  e.g.

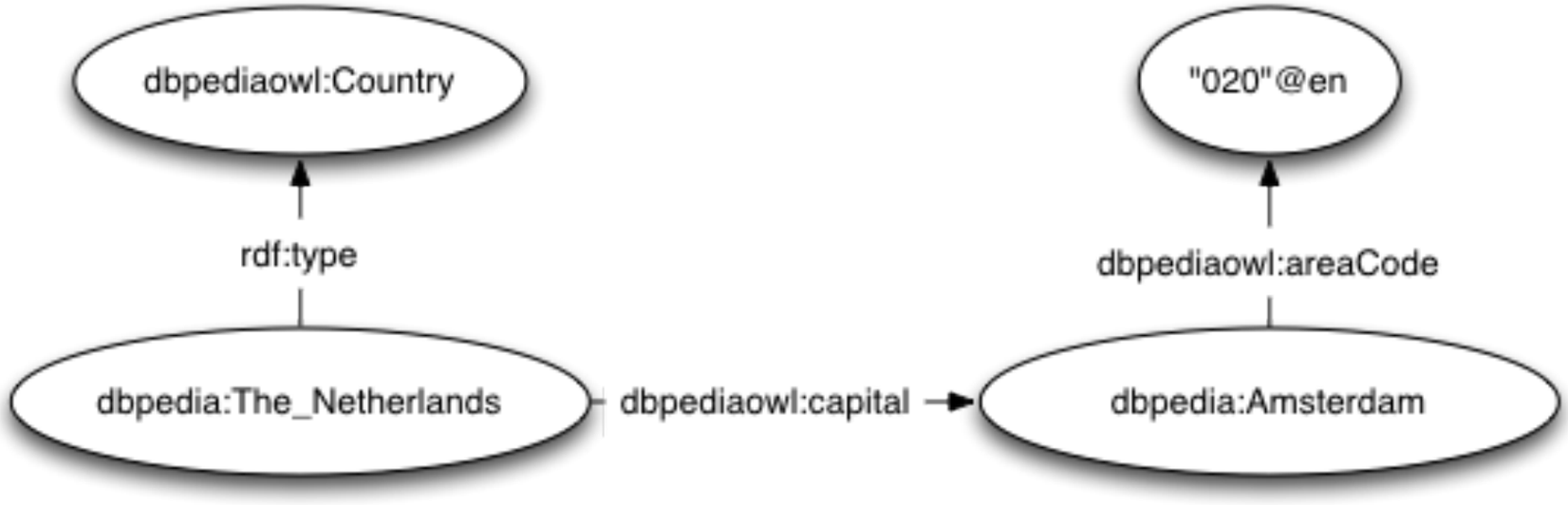
```
?x dbpediaowl:capital dbpedia:Amsterdam
```

```
?x dbpediaowl:capital ?y
```

```
?x dbpediaowl:areaCode "020"@en
```

```
?x ?p ?y
```

- All of them **match**!



Triple Patterns

- A **graph pattern** consists of multiple **triple patterns**

- A **triple pattern** is a triple with **zero** or **more variables**
  e.g.

```
?x dbpediaowl:capital dbpedia:Amsterdam
```

```
?x dbpediaowl:capital ?y
```

```
?x dbpediaowl:areaCode "020"@en
```

```
?x ?p ?y
```

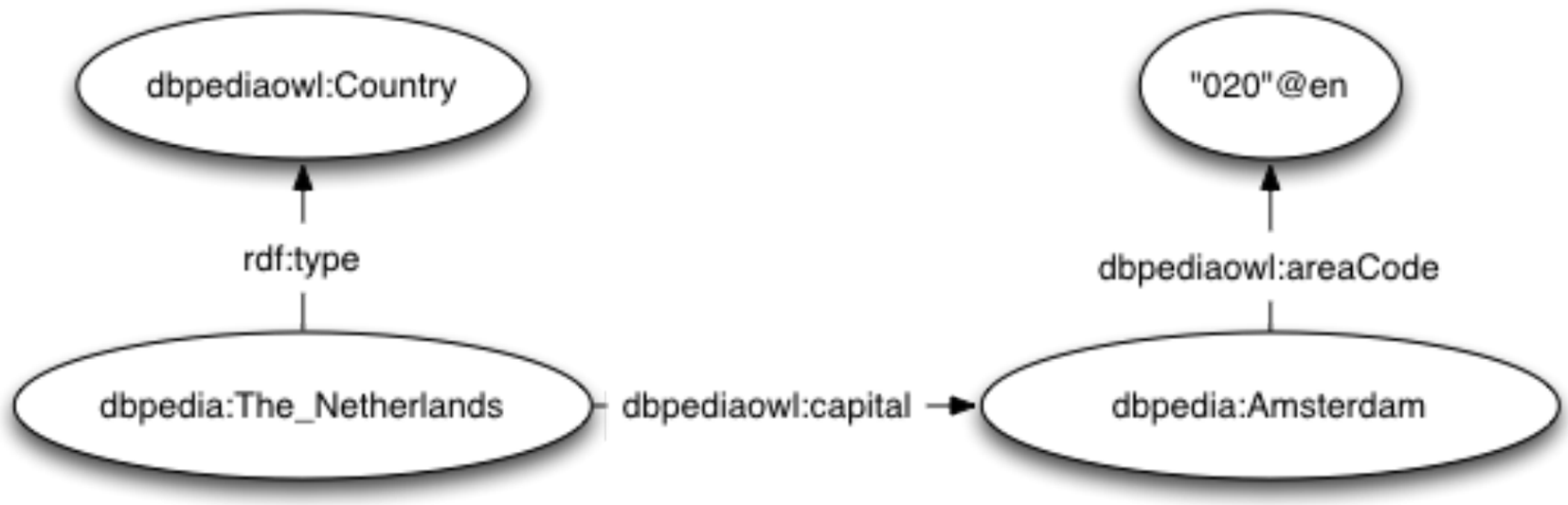
- All of them **match**!

```
dbpedia:Netherlands rdf:type dbpediaowl:Country ;
                    dbpediaowl:capital dbpedia:Amsterdam .
dbpedia:Amsterdam dbpediaowl:areaCode "020"@en .
```

Graph Pattern

- The WHERE clause specifies a **graph pattern**
  - that should be matched
  - can match multiple times

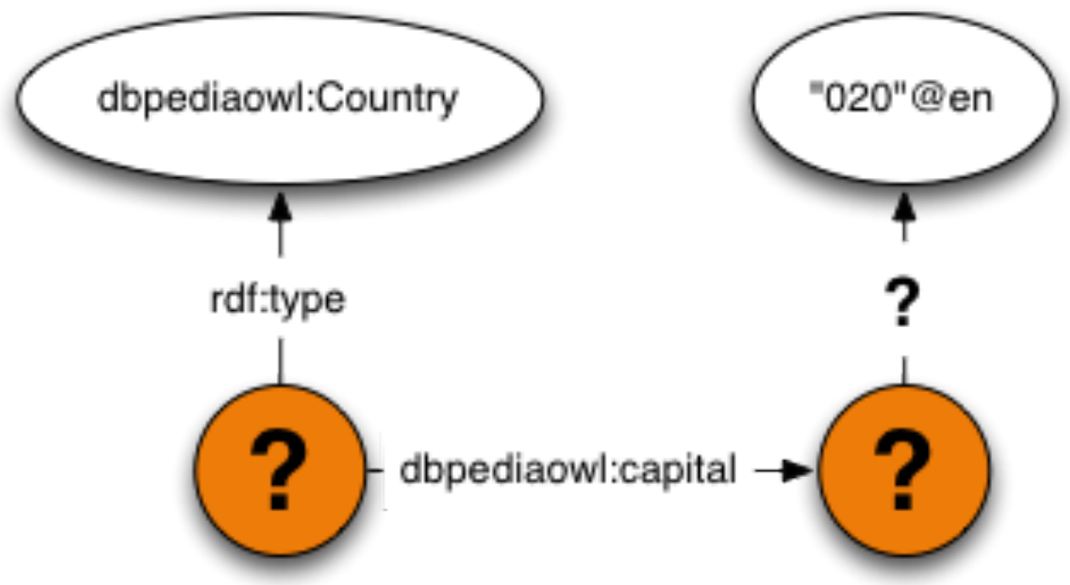
- A **graph pattern** is an **RDF graph** with some nodes & edges as **variables**



Graph Pattern

- The WHERE clause specifies a **graph pattern**
  - that should be matched
  - can match multiple times

- A **graph pattern** is an **RDF graph** with some nodes & edges as **variables**



SPARQL Query Syntax

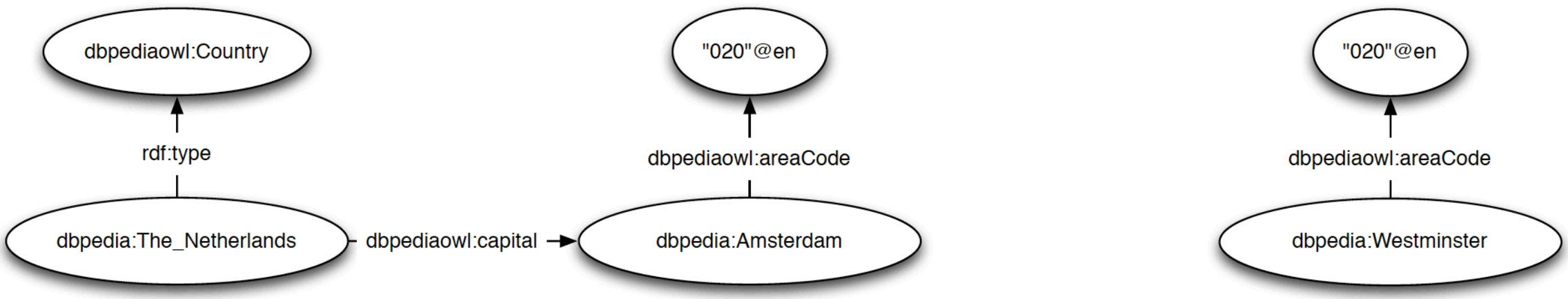
- PREFIX: the **namespace prefixes** used in the SPARQL query
- SELECT: the entities (variables) you want to **return**
- WHERE: the graph pattern that should be matched.

```
PREFIX dbpedia: <http://dbpedia.org/resource/>
PREFIX dbpediaowl: <http://dbpedia.org/ontology/>
SELECT ?country ?capital ?rel
WHERE {
 ?country rdf:type dbpediaowl:Country ;
                   dbpediaowl:capital ?capital .
 ?capital ?rel "020"@en .
}
```

Multiple triple patterns form a **conjunction**

- **Every** triple pattern should **match**

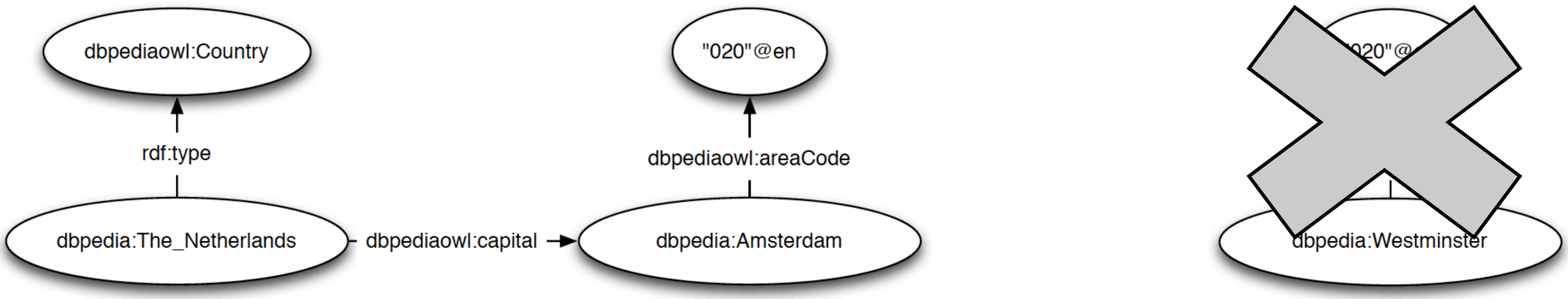
```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbpediaowl: <http://dbpedia.org/ontology/>

SELECT ?x
WHERE {
    ?x dbpediaowl:capital ?y .
    ?y dbpediaowl:areaCode "020"@en .
}
```



Multiple triple patterns form a **conjunction**

- **Every** triple pattern should **match**

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

SELECT ?x
WHERE {
    ?x dbpediaowl:capital ?y .
    ?y dbpediaowl:areaCode "020"@en .
}
```



But we can also use **disjunctions**

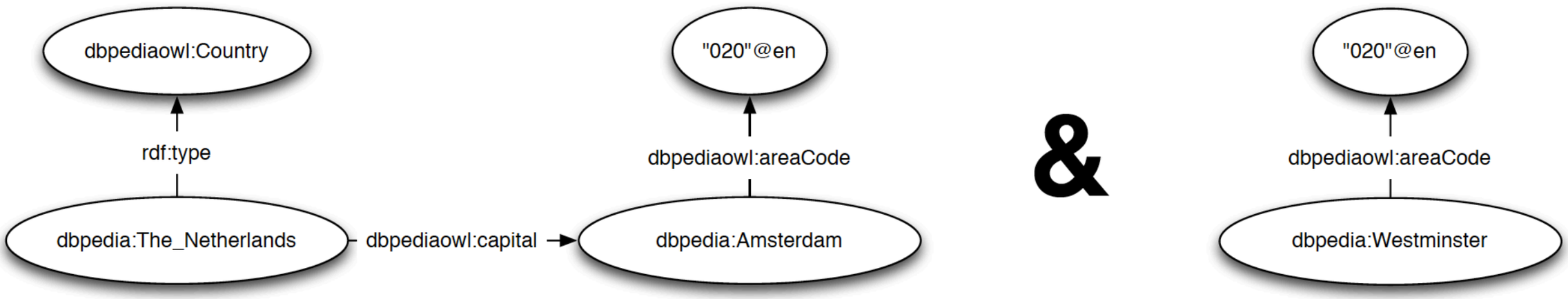
- **At least one** of the *graph* patterns should **match**

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

SELECT ?y
WHERE {
  { ?y dbpediaowl:areaCode "020"@en . }
  UNION
  { ?y dbpediaowl:areaCode "010"@en . }
}
```

Optional graphs

- A **part** of the graph pattern is **optional**, and does **not need to match**

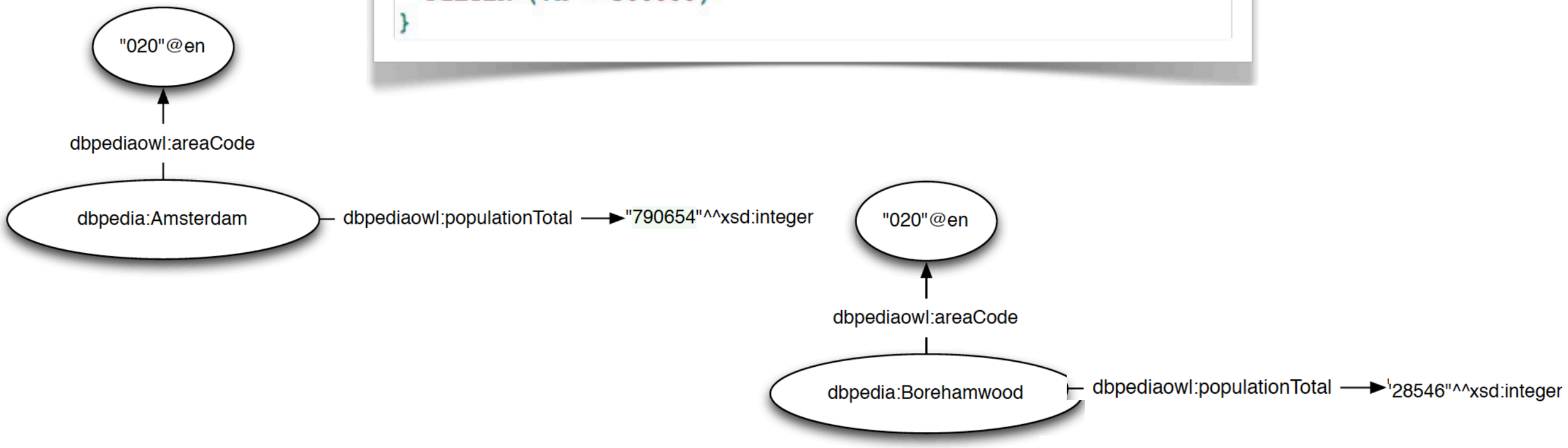
```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbpediaowl: <http://dbpedia.org/ontology/>

SELECT ?y ?x
WHERE {
  ?y dbpediaowl:areaCode "020"@en .
  OPTIONAL { ?x dbpediaowl:capital ?y . }
}
```



Filtering query results

- Tests in the **FILTER** clause have to be validated for matching subgraphs

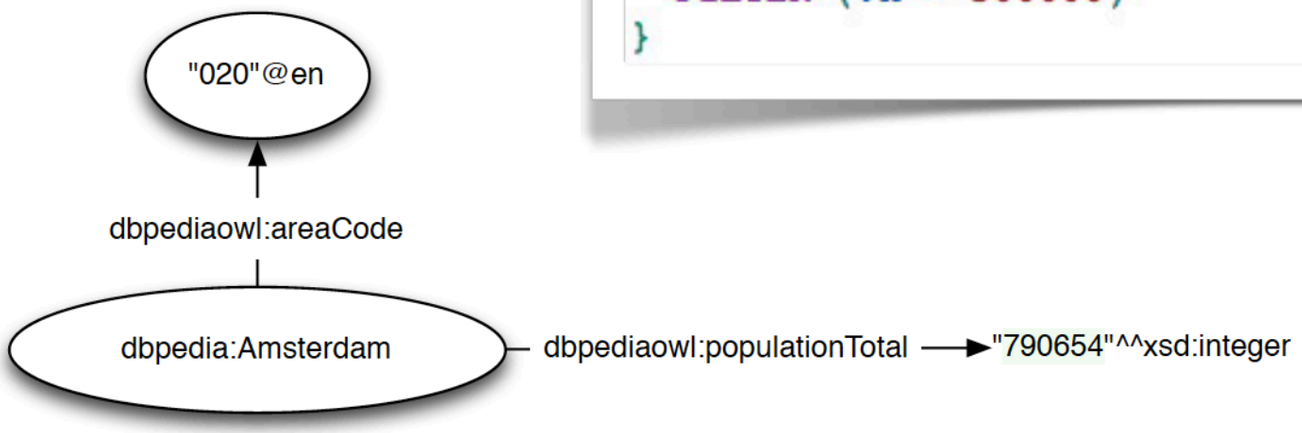
```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbpediaowl: <http://dbpedia.org/ontology/>

SELECT ?y ?x
WHERE {
  ?y dbpediaowl:areaCode "020"@en .
  ?y dbpediaowl:populationTotal ?x .
  FILTER (?x > 500000)
}
```

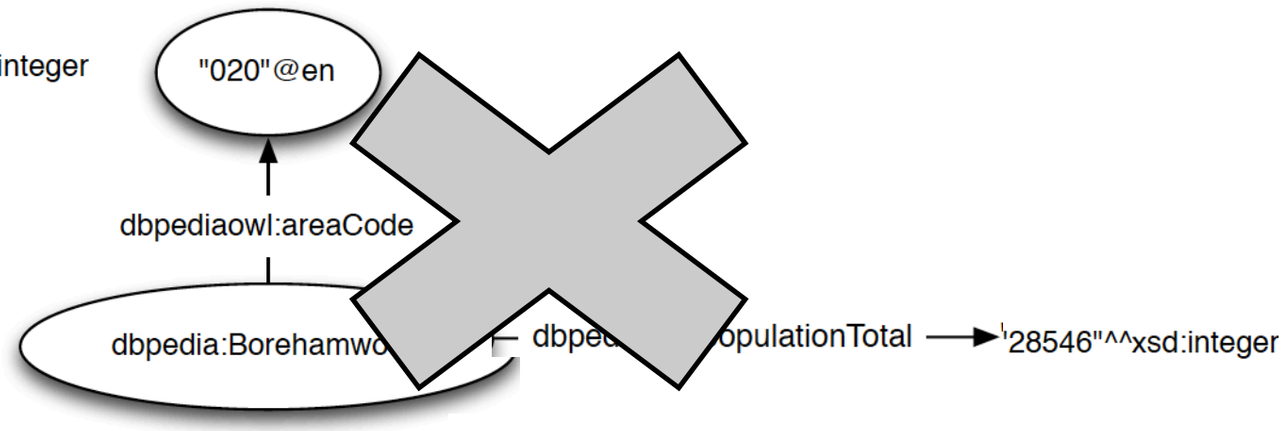


Filtering query results

- Tests in the **FILTER** clause have to be validated for matching subgraphs

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

SELECT ?y ?x
WHERE {
  ?y dbpediaowl:areaCode "020"@en .
  ?y dbpediaowl:populationTotal ?x .
  FILTER (?x > 500000)
}
```





Solution modifiers

- Sorting using **ORDER BY**

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbpediaowl: <http://dbpedia.org/ontology/>
SELECT ?label
WHERE {
 ?place dbpediaowl:areaCode "020"@en ;
        rdfs:label ?label
}
ORDER BY DESC(?label)
```

- **LIMIT**ing the number of results

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbpediaowl: <http://dbpedia.org/ontology/>
SELECT ?label
WHERE {
 ?place dbpediaowl:areaCode "020"@en ;
        rdfs:label ?label .
  FILTER ( lang(?label) = "en" )
}
ORDER BY DESC(?label)
LIMIT 10
```

- **DISTINCT** results

```
PREFIX dbpediaowl: <http://dbpedia.org/ontology/>
PREFIX dbpedia: <http://dbpedia.org/resource/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT DISTINCT ?name WHERE {
  ?country rdf:type dbpediaowl:Country ;
           dbpediaowl:capital ?ams .
  ?ams     dbpediaowl:country dbpedia:Netherlands ;
           rdfs:label ?name
  FILTER ( lang(?name) = "en" )
}
```

Query Types

- **SELECT** returns a table with **variable bindings**
- **CONSTRUCT** returns a RDF **graph**
- **ASK** returns **yes** or **no**
- **DESCRIBE** returns a RDF graph
- **INSERT** is like **CONSTRUCT**, but inserts the graph into the triple store

Query Types

- **SELECT** returns a table with **variable bindings**

- **CONSTRUCT** returns a RDF **graph**

```
PREFIX dbpedia: <http://dbpedia.org/resource/>
PREFIX dbpediaowl: <http://dbpedia.org/ontology/>
PREFIX example: <http://www.example.org/rdf#>
CONSTRUCT {
   ?city example:area-code ?area_code .
}
WHERE {
  ?city dbpediaowl:country dbpedia:Netherlands ;
        dbpediaowl:areaCode ?area_code .
}
```

- **ASK** returns **yes** or **no**

- **DESCRIBE** returns a RDF graph

- **INSERT** is like **CONSTRUCT**, but inserts the graph into the triple store

Query Types

- **SELECT** returns a table with **variable bindings**
- **CONSTRUCT** returns a RDF **graph**
- **ASK** returns **true** or **false**

```
PREFIX dbpedia: <http://dbpedia.org/resource/>
PREFIX dbpediaowl: <http://dbpedia.org/ontology/>
ASK
WHERE {
 dbpedia:Netherlands rdf:type dbpediaowl:Country ;
                     dbpediaowl:capital dbpedia:Amsterdam .
 dbpedia:Amsterdam dbpediaowl:areaCode "020"@en .
}
```

- **DESCRIBE** returns a RDF graph
- **INSERT** is like **CONSTRUCT**, but inserts the graph into the triple store

Query Types

- **SELECT** returns a table with **variable bindings**
- **CONSTRUCT** returns a RDF **graph**
- **ASK** returns **true** or **false**
- **DESCRIBE** returns a RDF graph

```
PREFIX dbpedia: <http://dbpedia.org/resource/>
DESCRIBE dbpedia:Amsterdam
```

- **INSERT** is like **CONSTRUCT**, but inserts the graph into the triple store

Query Types

- **SELECT** returns a table with **variable bindings**
- **CONSTRUCT** returns a RDF **graph**
- **ASK** returns **yes** or **no**
- **DESCRIBE** returns a RDF graph
- **INSERT** is like **CONSTRUCT**, but inserts the graph into the triple store

```
PREFIX dbpedia: <http://dbpedia.org/resource/>
PREFIX dbpediaowl: <http://dbpedia.org/ontology/>
PREFIX example: <http://www.example.org/rdf#>
INSERT {
   ?city example:area-code ?area_code .
}
WHERE {
  ?city dbpediaowl:country dbpedia:Netherlands ;
        dbpediaowl:areaCode ?area_code .
}
```

Querying the Linked Data Cloud

- DBPedia
- Factforge: DBPedia + Geonames + NY Times + MusicBrainz + ...

```
PREFIX dbpedia: <http://dbpedia.org/resource/>
PREFIX dbpediaowl: <http://dbpedia.org/ontology/>
SELECT ?capital WHERE {
  dbpedia:Netherlands dbpediaowl:capital ?capital .
}
```

- Same query, different results!

YasGUI SPARQL Client

- http://yasgui.laurensrietveld.nl/
  u: iwa.examples@gmail.com
  p: yasgui4iwa

- Namespace, predicate and class auto-completion, bookmarked queries, selectable endpoints
  - http://prefix.cc
  - http://sparqles.okfn.org

- Experiment using YasGUI during the Lab Session on Wednesday

More SPARQL features

- **Named Graphs**

  Provide the context for the statements they contain

- **Aggregates**

  COUNT, SUM, MIN, MAX, AVG, GROUP_CONCAT, and SAMPLE

- **GROUP BY** and **HAVING**

- **VALUES**

- **Sub queries**

Even more SPARQL features

- **Property paths**

  Sequence, inverse, alternative, zero-or-more, one-or-more …

- **Negation:** MINUS

- **Functions**

  - IF, BOUND, EXISTS, NOT EXISTS, IN, NOT IN
  - &&, ||, =, <, >, =>, =<
  - isIRI, isBlank, isLiteral, isNumeric, str, lang, datatype
  - STRLEN, SUBSTR, CONTAINS, CONCAT, REGEX
  - now, year, month, day, hours, minutes, seconds
  - and more!!

- http://www.w3.org/TR/sparql11-query/