UNSW
THE UNIVERSITY OF NEW SOUTH WALES

More on SPARQL

Madhushi Bandara and Fethi Rabhi

Different types of SPARQL queries

- So far, all queries we saw returned a table (SELECT statement)

SPARQL Explorer for http://dbpedia.org/

SPARQL:

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX : <http://dbpedia.org/resource/>
PREFIX dbpedia2: <http://dbpedia.org/property/>
PREFIX dbpedia: <http://dbpedia.org/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
SELECT ?name ?birth ?death ?person WHERE {      ?person dbo:birthPlace :Berlin .      ?person dbo:birthDate ?birth .      ?person foaf:name ?name .
```

Results: Browse Go! Reset

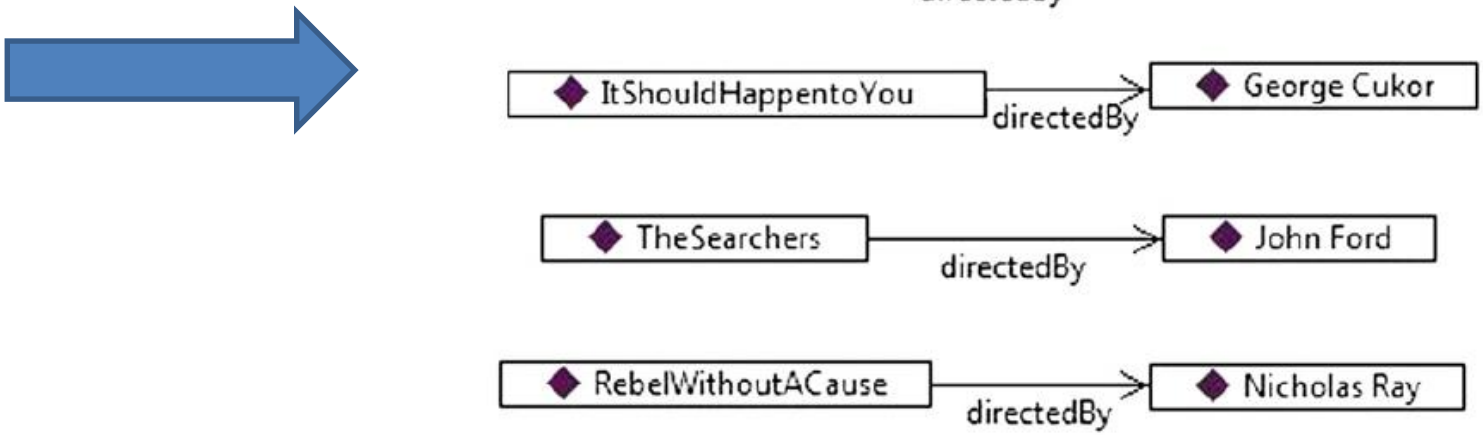
SPARQL results:

name	birth	death	person
""Annot" (Annot Jacobi)"@en	"1894-12-27"^^xsd:date	"1981-10-20"^^xsd:date	:Annot_(artist)
"()"@en	"1811-10-29"^^xsd:date	"1873-06-06"^^xsd:date	:Prince_Adalbert_of_Prussia_(1811–1873)
"()"@en	"1811-10-29"^^xsd:date	"1873-6-6"^^xsd:date	:Prince_Adalbert_of_Prussia_(1811–1873)
"(Henry William Adalbert)"@en	"1811-10-29"^^xsd:date	"1873-06-06"^^xsd:date	:Prince_Adalbert_of_Prussia_(1811–1873)
"(Henry William Adalbert)"@en	"1811-10-29"^^xsd:date	"1873-6-6"^^xsd:date	:Prince_Adalbert_of_Prussia_(1811–1873)
"Abraham Mendelssohn Bartholdy"@en	"1776-12-10"^^xsd:date	"1835-11-19"^^xsd:date	:Abraham_Mendelssohn_Bartholdy
"Achim von Arnim"@en	"1781-01-26"^^xsd:date	"1831-1-21"^^xsd:date	:Ludwig_Achim_von_Arnim
"Adalbert of Prussia"@en	"1811-10-29"^^xsd:date	"1873-06-06"^^xsd:date	:Prince_Adalbert_of_Prussia_(1811–1873)
"Adalbert of Prussia"@en	"1811-10-29"^^xsd:date	"1873-6-6"^^xsd:date	:Prince_Adalbert_of_Prussia_(1811–1873)
"Adam Müller"@en	"1779-06-30"^^xsd:date	"1829-01-17"^^xsd:date	:Adam_Müller
"Adam Müller"@en	"1779-06-30"^^xsd:date	"1829-1-17"^^xsd:date	:Adam_Müller
"Adolf Brand"@en	"1874-11-14"^^xsd:date	"1945-2-2"^^xsd:date	:Adolf_Brand
"Adolf Christen"@en	"1811-08-07"^^xsd:date	"1883-07-13"^^xsd:date	:Adolf_Christen
"Adolf Christen"@en	"1811-08-07"^^xsd:date	"1883-7-13"^^xsd:date	:Adolf_Christen
"Adolf Damaschke"@en	"1865-11-24"^^xsd:date	"1935-7-30"^^xsd:date	:Adolf_Damaschke
"Adolf Erman"@en	"1854-10-31"^^xsd:date	"1937-6-26"^^xsd:date	:Adolf_Erman
"Adolf Gärtner"@en	"1879-03-24"^^xsd:date	"1958-01-09"^^xsd:date	:Adolf_Gärtner

Construct queries

- CONSTRUCT can be used to create new triples

CONSTRUCT {?d rdf:type :Director .
?d rdfs:label ?name . }
WHERE {?any :directedBy ?d .
?d rdfs:label ?name . }



Processing returned triples

- Sophisticated RDF query systems allow you to process newly created triples in different ways:
  - Insert the constructed triples back into the original data source
  - Store the constructed triples as a separate graph
  - Store the constructed triples into a new dataset
  - Serialize the results into a file

Defining rules using SPARQL

- Rule = a way to derive *new* information from the existing one
- We saw that SPARQL (CONSTRUCT) can be used to create new triples
- For example: a system defines the following relationships:
  - Brother
  - Sister
  - Father
  - Mother
- How do you define uncle ?

```
:John a :Man.
:Joe a :Man.
:Eunice a :Woman .
:Maria a :Woman .
:Caroline a :Woman .
:Ted a :Man .
:Socrates a :Man .
:Caroline :hasFather :John .
:Ted :hasBrother :John .
:John :hasFather :Joe .
:Maria :hasMother :Eunice .
:Maria :hasFather :Sargent .
:Ted :hasSister :Eunice .
```

Solution 1

- Defining a SPARQL construct query as

```
CONSTRUCT {?q1 :hasUncle ?q2}
WHERE {?q2 :hasSister ?s .
?q1 :hasMother ?s .}
```

- What is the problem here ?

Solution 2

- First define sibling

```
CONSTRUCT {?q1 :hasSibling ?q2} WHERE {?q1 :hasBrother ?q2}
CONSTRUCT {?q1 :hasSibling ?q2} WHERE {?q1 :hasSister ?q2}
CONSTRUCT {?q1 :hasParent ?q2} WHERE {?q1 :hasFather ?q2}
CONSTRUCT {?q1 :hasParent ?q2} WHERE {?q1 :hasMother ?q2}
```

- Then define uncle

```
CONSTRUCT {?q1 :hasUncle ?q2}
WHERE {?q2 :hasSibling ?parent .
?q2 a :Man .
?q1 :hasParent ?parent }
```

Federated SPARQL queries

Querying large datasets

- Problem
  - when data sets are very large, it can be impractical to merge them together before querying them
  - Copying whole datasets may not be possible
- Federating data sources
  - means to virtually combine the data sources in the query, while leaving each component with its own identity
  - both endpoints and named graphs can participate in federated SPARQL queries

Example

Ask:

```
SELECT ?entry
WHERE {?actor :playedIn :Giant .
   ?actor rdfs:label ?name .
       SERVICE <http://dbpedia.org/sparql>
          {?entry rdfs:label ?name .}
       }
```

Answer:

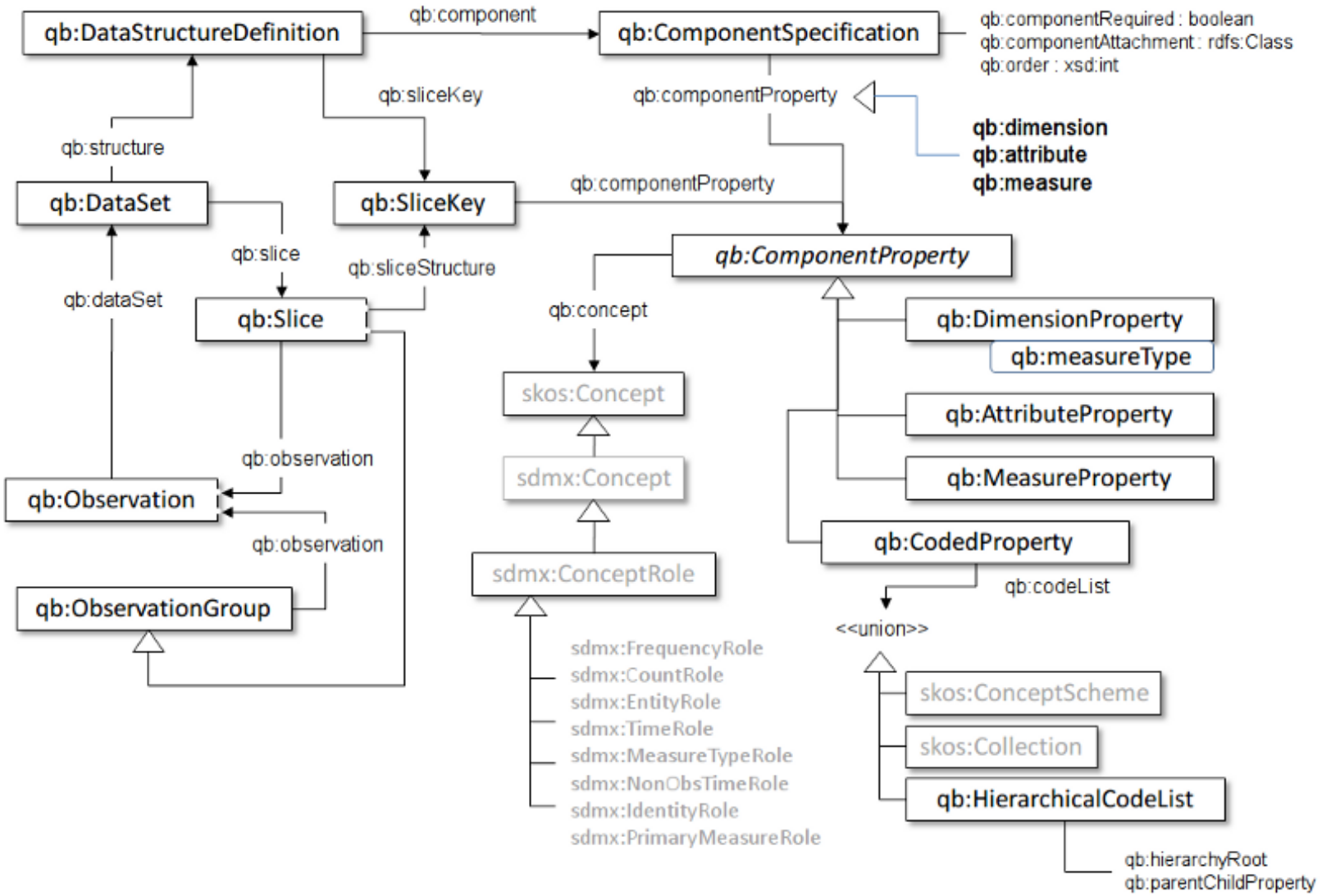
?entry
<http://dbpedia.org/resource/Carroll_Baker>
<http://dbpedia.org/resource/Elizabeth_Taylor>
<http://dbpedia.org/resource/James_Dean>
<http://dbpedia.org/resource/Mercedes_McCambridge>
<http://dbpedia.org/resource/Rock_Hudson>
<http://dbpedia.org/resource/Sal_Mineo>

Querying RDF Cube

RDF Data Cube

- Standard for sharing statistical datasets
- Published by W3C
- Builds on other standards

RDF Cube Ontology



Example Data Cube
Table 1

	2004-2006		2005-2007		2006-2008	
	Male	**Female**	**Male**	**Female**	**Male**	**Female**
Newport	76.7	80.7	77.1	80.9	77.0	81.5
Cardiff	78.7	83.3	78.6	83.7	78.7	83.4
Monmouthshire	76.6	81.3	76.5	81.5	76.6	81.7
Merthyr Tydfil	75.5	79.1	75.5	79.4	74.9	79.6

*data set extracted from StatsWales report number 003311 which describes life expectancy broken down by region (unitary authority), age and time

Example Implemented

- You can find the complete implementation of Table 1 in following URL:
- http://adage.cse.unsw.edu.au/Resources/SampleData/TripleData/LifeExpectancyDataSet.ttl

- You can import it to Protégé and explore

Querying RDF Cube

- Once you import a dataset in RDF format you can query is using standard vocabulary and answer questions such as:
  - What is the label and description?
  - What are the dimensions specified in the dataset?
  - Given values for each dimension, what is the measured value of observation?
  - Etc…
- Following slides provide example SPARQL queries for these questions.

Explore Data Sets

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX qb:   <http://purl.org/linked-data/cube#>
PREFIX dct:  <http://purl.org/dc/terms/>
SELECT DISTINCT  ?label ?description
WHERE {          ?id a qb:DataSet.
                 ?id rdfs:label ?label.
                 ?id dct:description ?description
}
```

Output for Example LifeExpectancy Dataset:

"Life expectancy"@en "Life expectancy within Welsh Unitary authorities - extracted from Stats Wales"@en

Explore Dimensions

```
PREFIX qb:      <http://purl.org/linked-data/cube#>
SELECT DISTINCT  ?dim
WHERE {
                    ?id a qb:DataSet.
                    ?id qb:structure ?dsd.
                    ?dsd qb:component ?comp.
                    ?comp qb:dimension ?dim.

        }
```

Alternative Query Format

```
PREFIX qb:      <http://purl.org/linked-data/cube#>
SELECT DISTINCT  ?dim
WHERE {
                    ?id a qb:DataSet.
                    ?id qb:structure/qb:component/qb:dimension ?dim.
}
```

Output for Example LifeExpectency Dataset:

http://purl.org/linked-data/sdmx/2009/dimension#sex
http://example.org/ns#refPeriod
http://example.org/ns#refArea

Retrieve Specific Value

```
PREFIX qb:      <http://purl.org/linked-data/cube#>
PREFIX sdmx-code:      <http://purl.org/linked-data/sdmx/2009/code#>
PREFIX sdmx-dimension:  <http://purl.org/linked-data/sdmx/2009/dimension#>
PREFIX ex-geo:   <http://example.org/geo#>
PREFIX eg:      <http://example.org/ns#>
 SELECT DISTINCT  ?val
  WHERE {
        ?data  eg:refPeriod <http://reference.data.gov.uk/id/gregorian-interval/2004-01-01T00:00:00/P3Y> .
                                                                              #2004-2007
        ?data sdmx-dimension:sex  sdmx-code:sex-M.                            #Male
        ?data eg:refArea  ex-geo:newport_00pr .                               #From Newport
        ?data eg:lifeExpectancy ?val
}
```

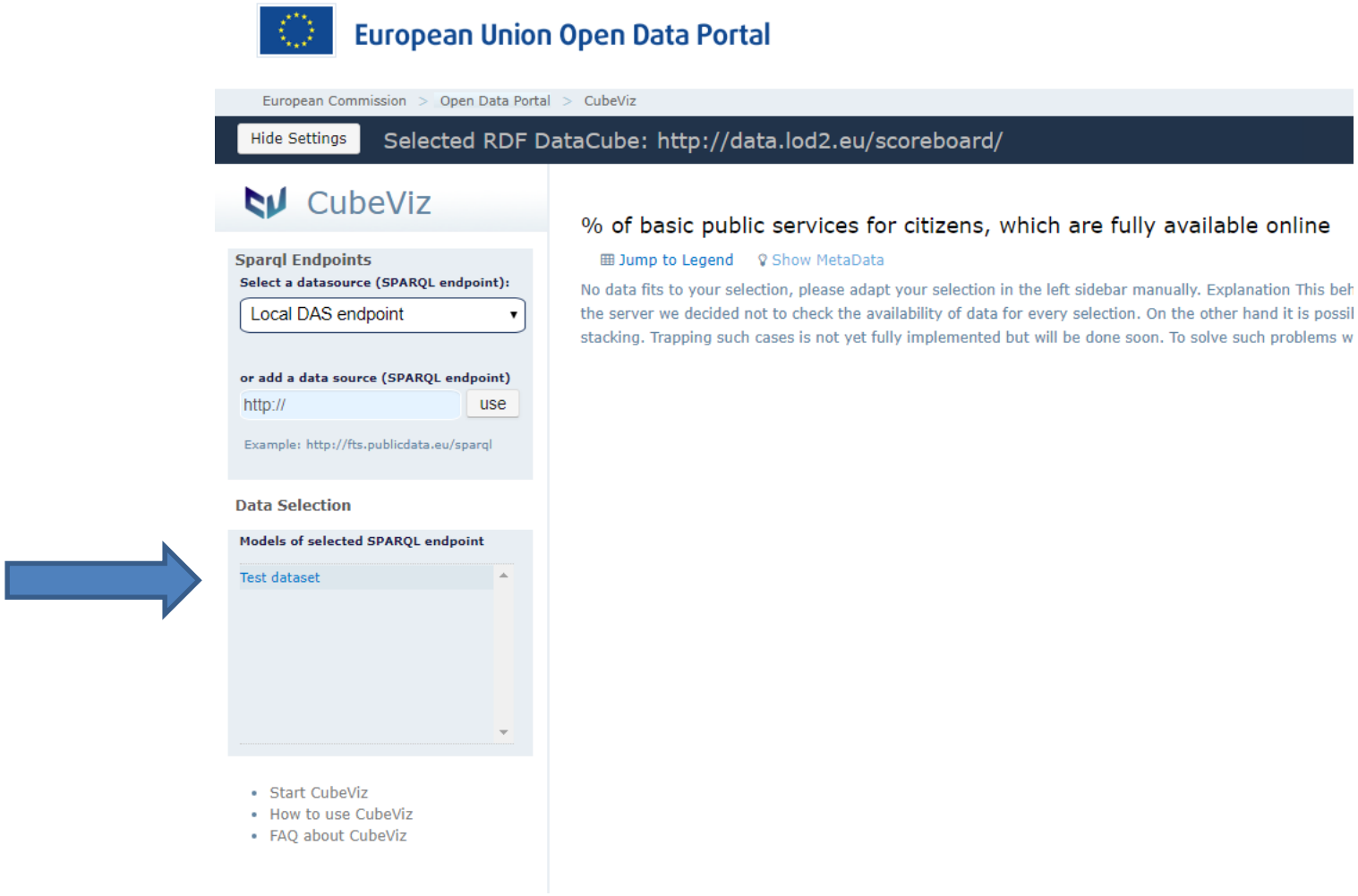
Output for Example LifeExpectency Dataset:

”76.7"^^<http://www.w3.org/2001/XMLSchema#decimal>

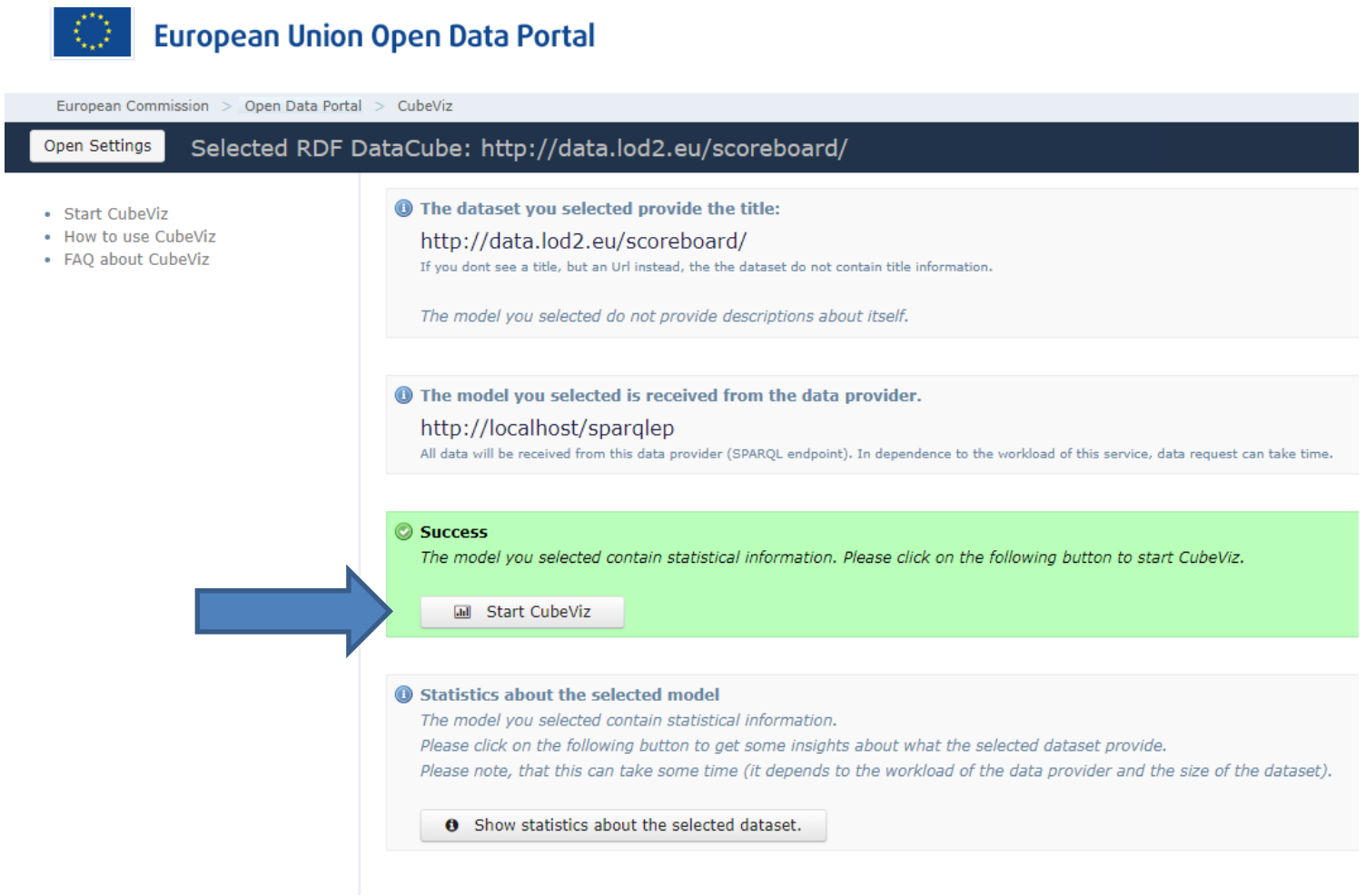
RDF Cube Applications

- Apps can be built on top of SPARQL endpoints
- CubViz
  - a facetted browser for statistical data utilizing the RDF Data Cube vocabulary
  - provides possibilities to interactively filter observations which are to be visualized as charts
  - Available from EU Open Data Portal: https://data.europa.eu/euodp/cubeviz/

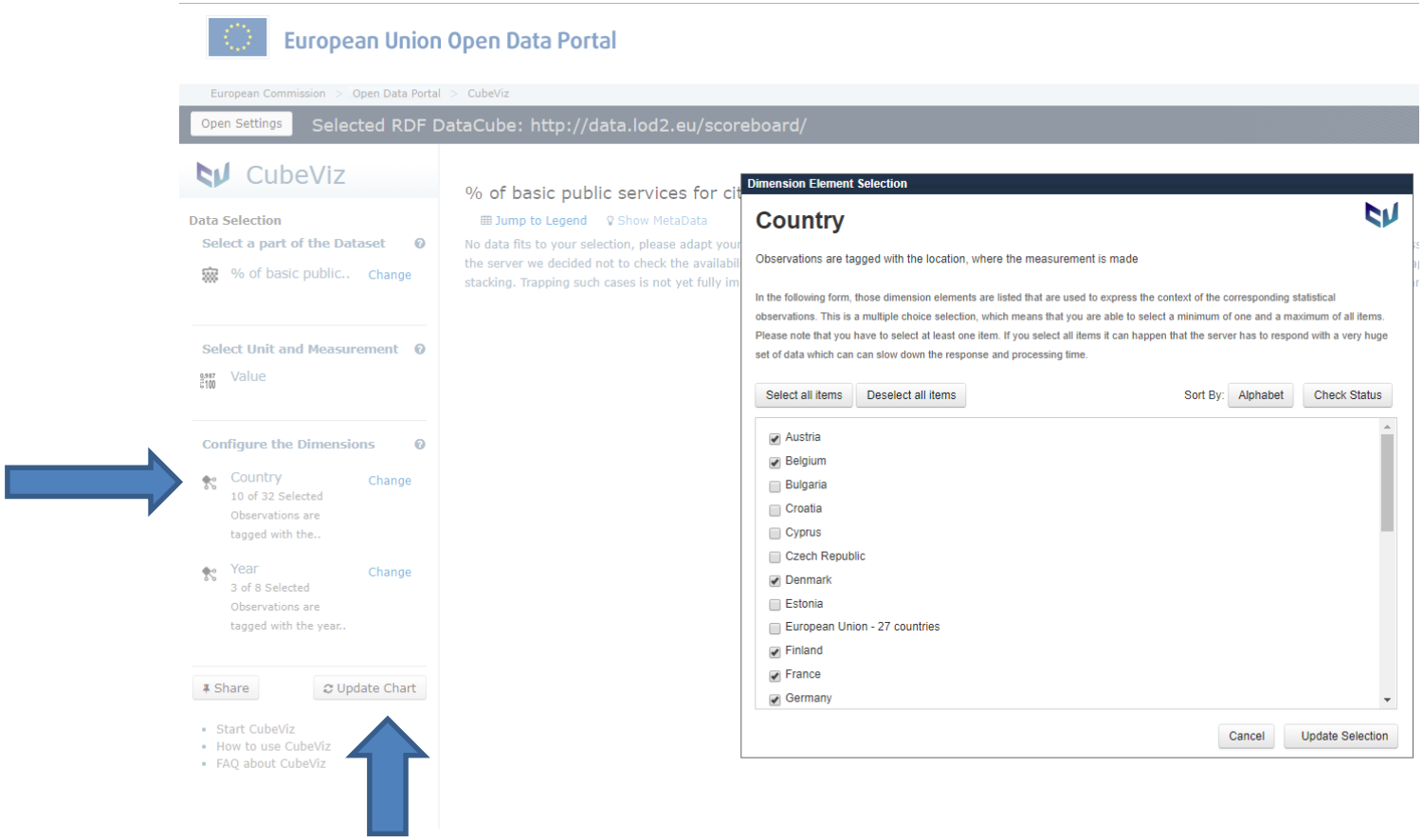
Selecting a statistical dataset



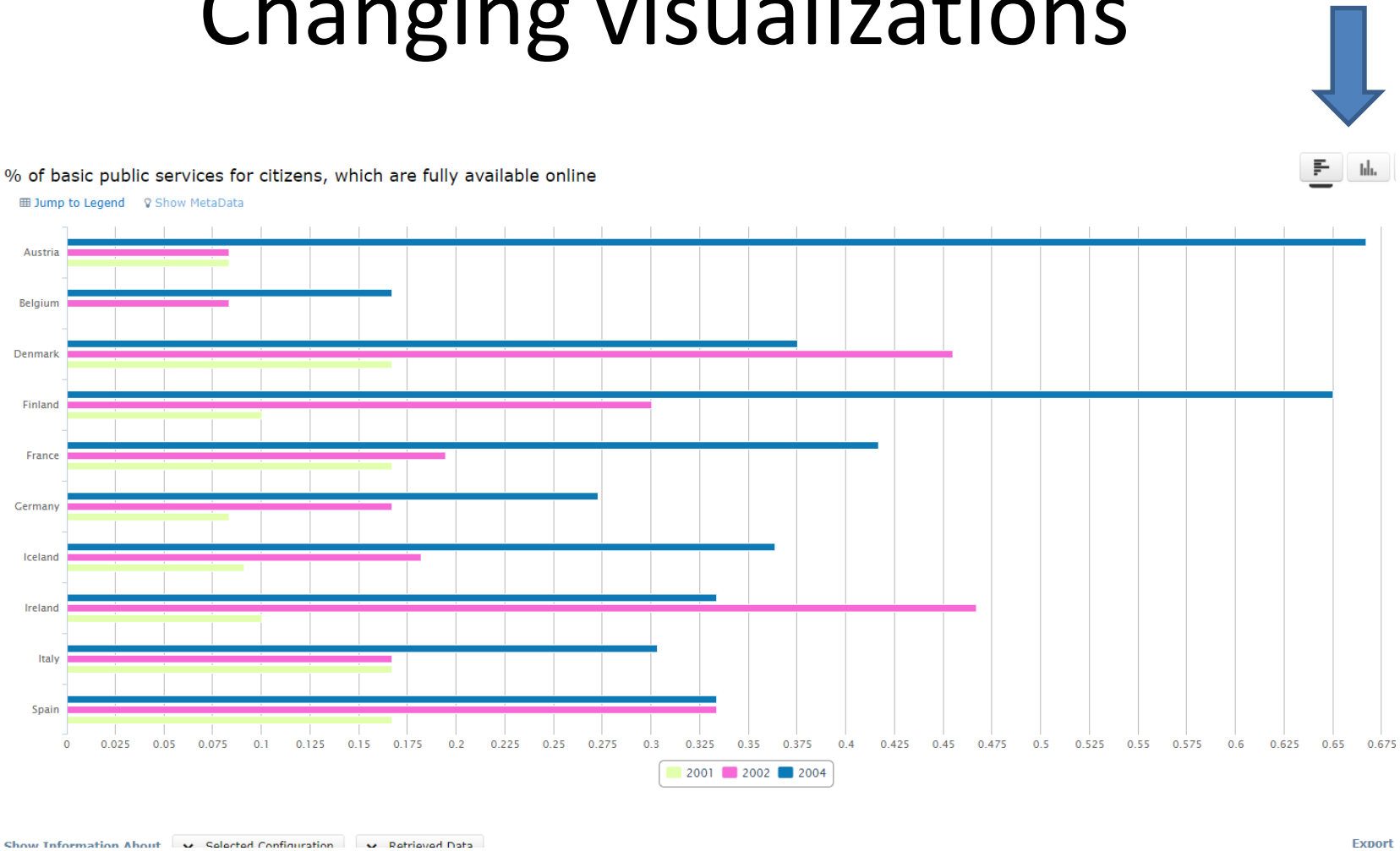
Starting CubViz



Selecting dimensions and creating chart



Changing visualizations



Thank You

UNSW
THE UNIVERSITY OF NEW SOUTH WALES