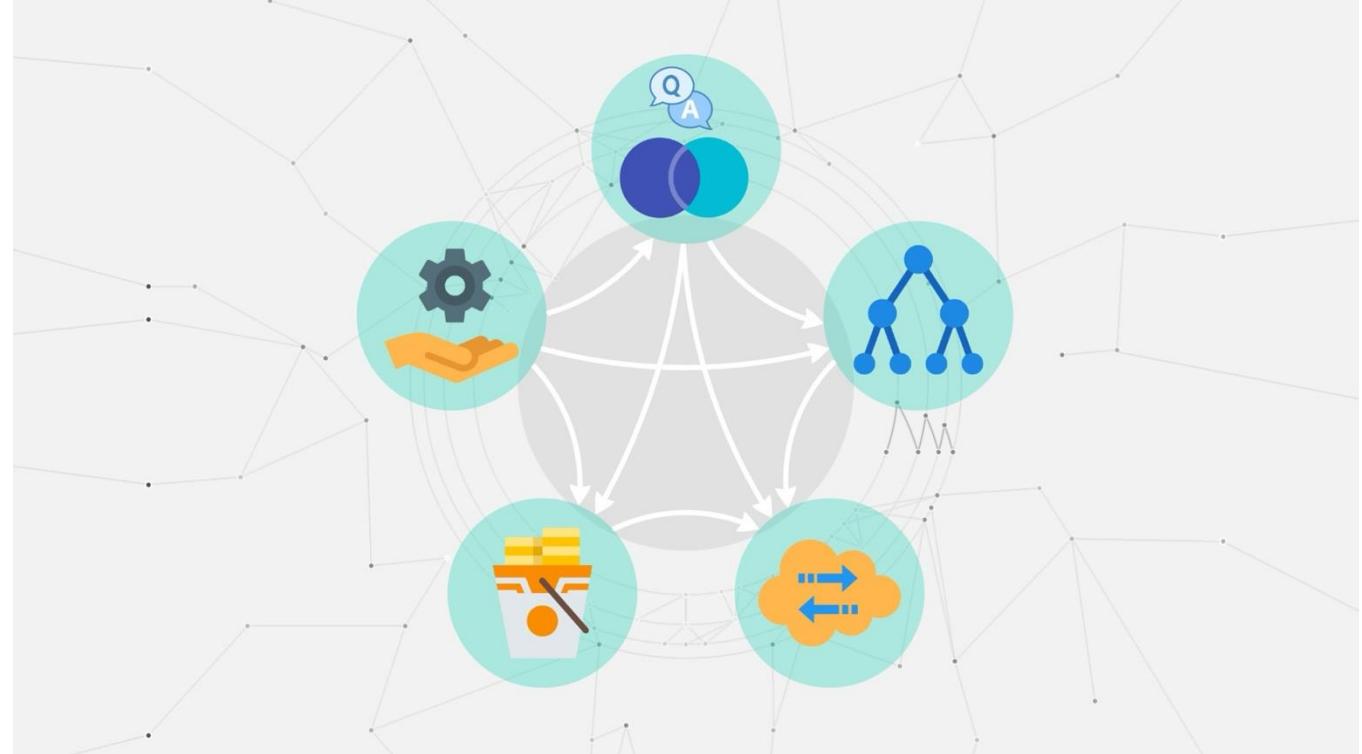


# Practical RDF and SPARQL

---



# 01 Introduction

# Course Audience

! Prerequisite  
• RDF & SPARQL Essentials

Researchers &  
academics

Data scientists

Data consultants

Novice knowledge  
engineers & ontologists

Data leaders

Data & Information  
architects

Software developers



# Learning Outcomes

Progress your understanding of SPARQL query language through practical examples and exercises, including SPARQL Update to perform CRUD operations

Uplift an implicit graph vocabulary into an explicit schema utilising structures provided in the RDF Schema (RDFS) and automated techniques

Perform graph management operations and practice querying the default and named graphs



Define and query RDF Containers and Collections



Become comfortable working with federated queries to exploit the power of Linked Open Data (LOD) sources



Out-of-scope:

- Pretty graph visualisations
- Data transformation pipelines
- Exploratory & predictive AI / ML
- Reification (standard, RDF-star)

# Course Structure

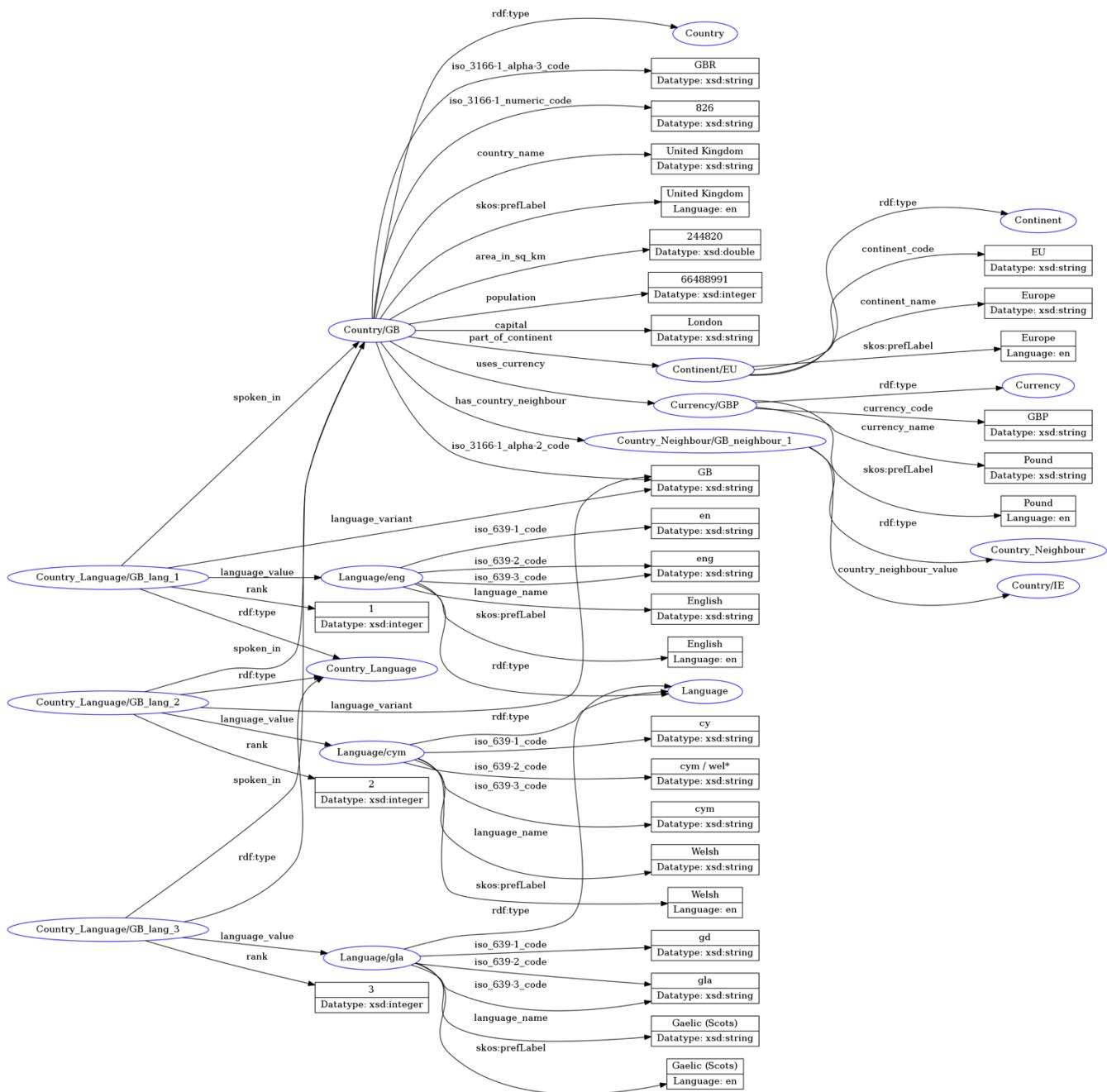


## 02 Familiarising with the dataset

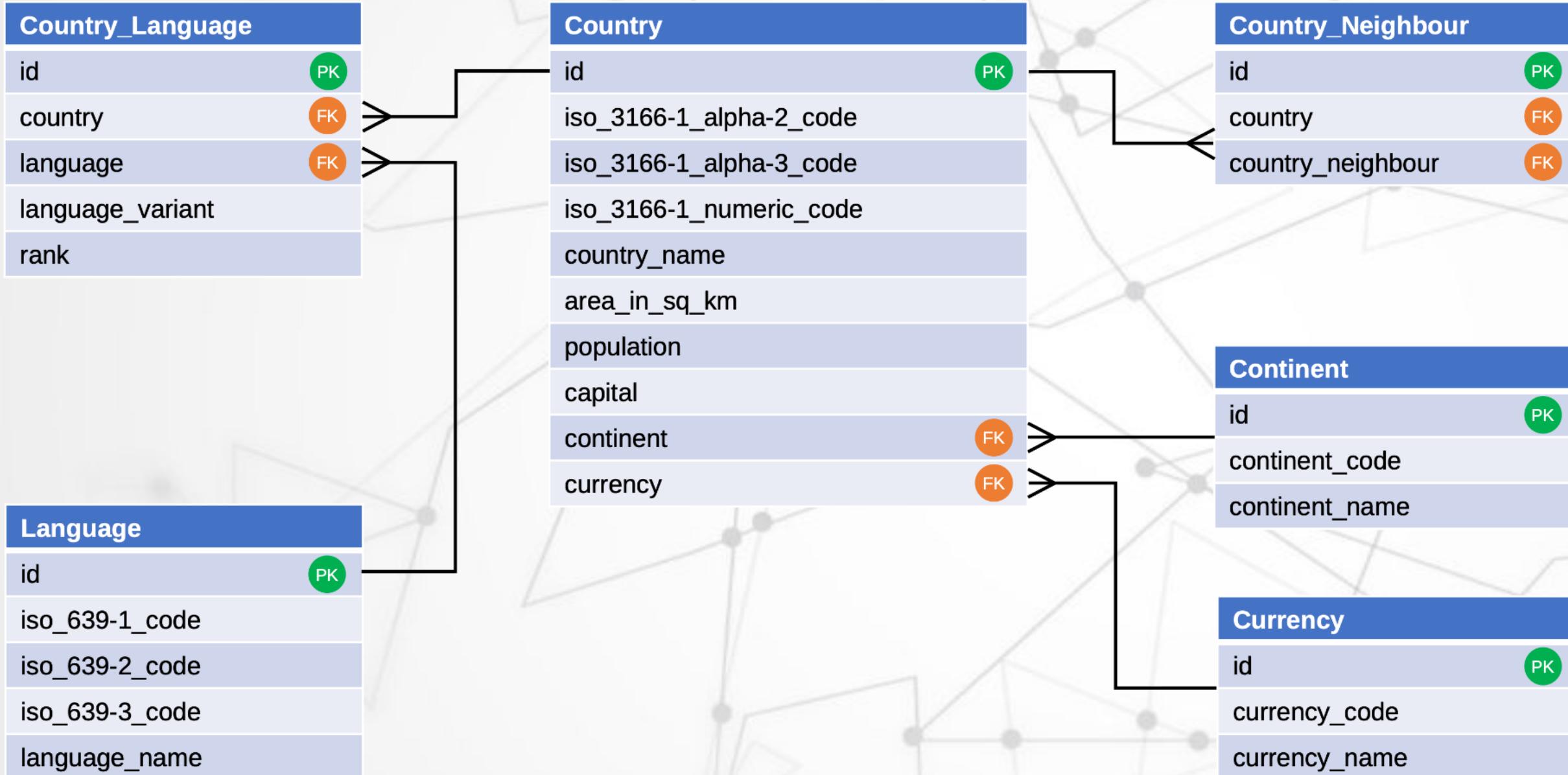
≡ GB\_Extended\_Graph.ttl ×

Volumes &gt; Extreme Pro &gt; Courses &gt; Practical RDF and SPARQL &gt; 2-Dataset Familiarisation &gt; 1a-Extended Graph for Country &gt; ≡ GB\_Extended\_Graph.ttl

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
3 PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
4 PREFIX : <http://example.com/demo/>
5
6 # Country info
7 :Country\GB a :Country ;
8   :iso_3166-1_alpha-2_code 'GB'^^xsd:string ;
9   :iso_3166-1_alpha-3_code 'GBR'^^xsd:string ;
10  :iso_3166-1_numeric_code '826'^^xsd:string ;
11  :country_name 'United Kingdom'^^xsd:string ;
12  skos:prefLabel 'United Kingdom"@en ;
13  :area_in_sq_km '244820'^^xsd:double ;
14  :population '66488991'^^xsd:integer ;
15  :capital 'London'^^xsd:string ;
16  :part_of_continent :Continent\EU ;
17  :uses_currency :Currency\GBP ;
18  :has_country_neighbour :Country_Neighbour\GB_neighbour_1 .
19
20 # Country neighbours
21 :Country_Neighbour\GB_neighbour_1 a :Country_Neighbour ;
22   :country_neighbour_value :Country\IE .
23
24 # Country languages
25 :Country_Language\GB_lang_1 a :Country_Language ;
26   :spoken_in :Country\GB ;
27   :language_value :Language\eng ;
28   :language_variant 'GB'^^xsd:string ;
29   :rank '1'^^xsd:integer .
30
31 :Country_Language\GB_lang_2 a :Country_Language ;
32   :spoken_in :Country\GB ;
33   :language_value :Language\cym ;
34   :language_variant 'GB'^^xsd:string ;
35   :rank '2'^^xsd:integer .
36
37 :Country_Language\GB_lang_3 a :Country_Language ;
38   :spoken_in :Country\GB ;
39   :language_value :Language\gla ;
40   :rank '3'^^xsd:integer .
41
42 # Continent
43 :Continent\EU a :Continent ;
44   :continent_code 'EU'^^xsd:string ;
```



Namespaces:  
 rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  
 xsd: <http://www.w3.org/2001/XMLSchema#>  
 skos: <http://www.w3.org/2004/02/skos/core#>  
<http://example.com/demo/>



Country\_Language

<b>id</b>	<b>country</b>	<b>language</b>	<b>language_variant</b>	<b>rank</b>
GB_lang_1	GB	eng	GB	1
GB_lang_2	GB	cym	GB	2
GB_lang_3	GB	gla		3
IE_lang_1	IE	eng	IE	1
IE_lang_2	IE	gle	IE	2

Language

<b>id</b>	<b>iso_639-1_code</b>	<b>iso_639-2_code</b>	<b>iso_639-3_code</b>	<b>Language_name</b>
eng	en	eng	eng	English
cym	cy	cym / wel*	cym	Welsh
gla	gd	gla	gla	Gaelic (Scots)
gle	ga	gle	gle	Irish

Country

<b>id</b>	<b>iso_3166-1_alpha-2_code</b>	<b>iso_3166-1_alpha-3_code</b>	<b>iso_3166-1_numeric_code</b>	<b>country_name</b>	<b>area_in_sq_km</b>	<b>population</b>	<b>capital</b>	<b>continent</b>	<b>currency</b>
GB	GB	GBR	826	United Kingdom	244820	66488991	London	EU	GBP
IE	IE	IRL	372	Ireland	70280	4853506	Dublin	EU	EUR

Country\_Neighbour

<b>id</b>	<b>country</b>	<b>country_neighbour</b>
GB_neighbour_1	GB	IE
IE_neighbour_1	IE	GB

Continent

<b>id</b>	<b>continent_code</b>	<b>continent_name</b>
EU	EU	Europe

Currency

<b>id</b>	<b>currency_code</b>	<b>currency_name</b>
GBP	GBP	Pound
EUR	EUR	Euro

# 03 Warming up with SPARQL

PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt;

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
3 PREFIX : <http://example.com/demo/>
4
5 # What is the total land mass area (to the nearest sq km) of a given continent, e.g. "Asia"?
6
7 SELECT (ROUND(SUM(?cArea)) AS ?ctArea)
8 WHERE {
9     ?c :part_of_continent/:continent_name "Asia" ;
10    :area_in_sq_km ?cArea .
11 }
12
13
14 # What is the total population of each continent, excluding "Antarctica" and "Oceania"?
15
16 SELECT ?ctName (SUM(?cPop) AS ?ctPop)
17 WHERE {
18     ?c :part_of_continent/:continent_name ?ctName ;
19     :population ?cPop .
20
21     FILTER(?ctName NOT IN("Antarctica", "Oceania"))
22 }
23 GROUP BY ?ctName
24
25
26 # List the country names of countries in the continent "Asia" whose names end in 'stan'. For these countries, list the names of their associated languages and the rank of the languages.
27 Order the results in ascending order of country name and language rank.
28
29 SELECT ?cName ?langN ?r
30 WHERE {
31     ?c :country_name ?cName ;
32     :part_of_continent/:continent_name "Asia" .
33
34     ?clang :spoken_in ?c ;
35     :language_value/:language_name ?langN ;
36     :rank ?r .
37
38     FILTER(STREND($cName, "stan"))
39 }
40 ORDER BY ASC(?cName) ASC(?r)
41
42 # For the countries in "Asia" whose names end in 'stan', list the names of their associated languages as data transposed in such a way that the result table displays, for each country, its
43 rank 1 language as primary language in one column, rank 2 language as secondary language in another column, and rank 3 language as tertiary language in another column.
44
45 SELECT ?cName ?primaryLang ?secondaryLang ?tertiaryLang
46 WHERE {
```

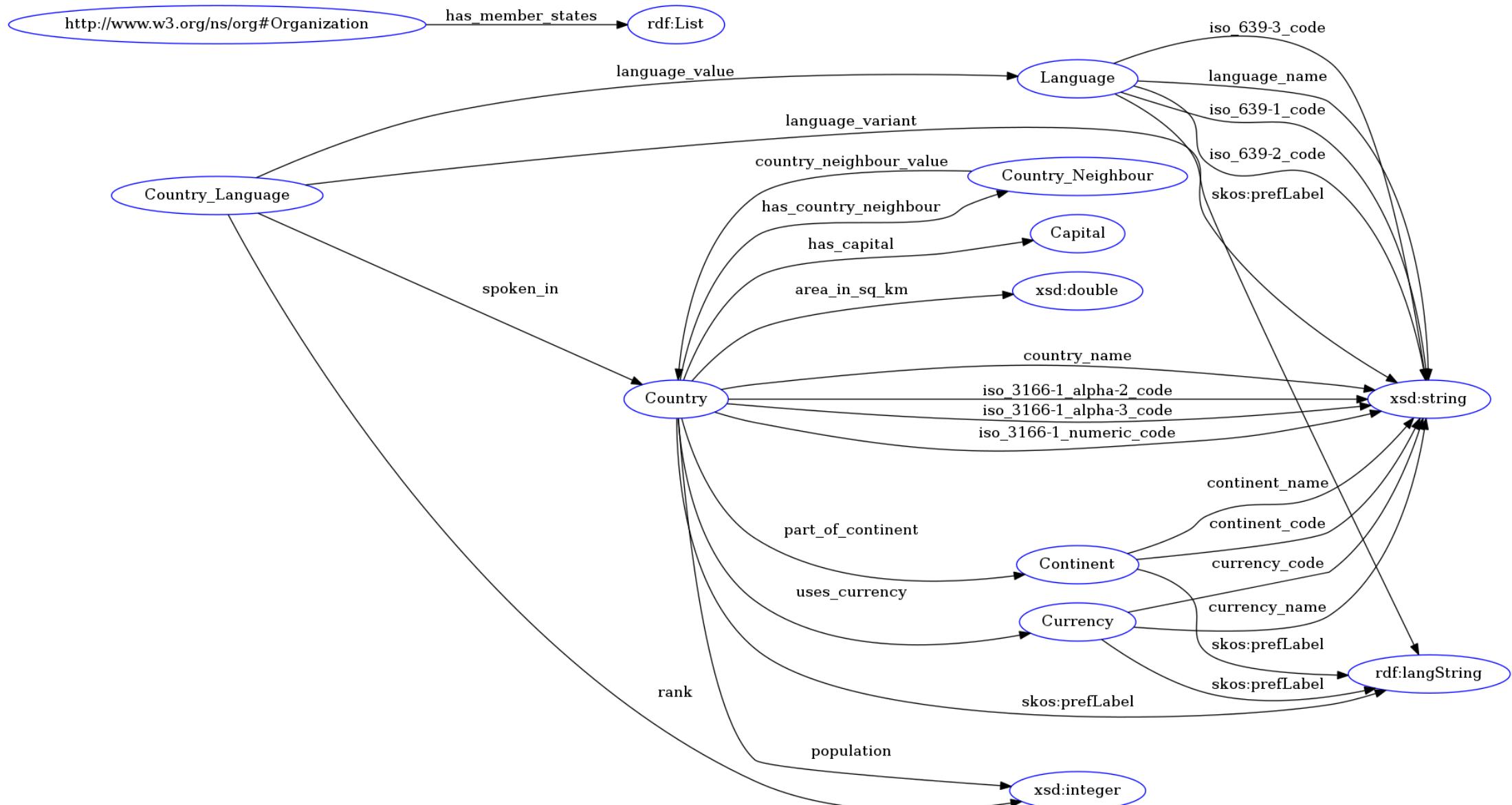


# 04 Uplifting the graph vocabulary with RDF Schema

PREFIX rdf: &lt;http://www.w3.org/1999/02/2 Untitled-1

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
4 PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
5 PREFIX : <http://example.com/demo/>
6
7 # Class declarations
8
9 :Country a rdfs:Class ;
10    skos:prefLabel "Country"@en ;
11    rdfs:comment "Manually declared this class"@en ;
12    rdfs:isDefinedBy <http://example.com/demo> .
13
14 :Continent a rdfs:Class ;
15    skos:prefLabel "Continent"@en ;
16    rdfs:comment "Manually declared this class"@en ;
17    rdfs:isDefinedBy <http://example.com/demo> .
18
19 # Property declarations
20
21 :part_of_continent a rdf:Property ;
22    skos:prefLabel "Part of continent"@en ;
23    rdfs:domain :Country ;
24    rdfs:range :Continent ;
25    rdfs:comment "Manually declared this property"@en ;
26    rdfs:isDefinedBy <http://example.com/demo> .
27
28 :country_name a rdf:Property ;
29    skos:prefLabel "Country name"@en ;
30    rdfs:domain :Country ;
31    rdfs:range xsd:string ;
32    rdfs:comment "Manually declared this property"@en ;
33    rdfs:isDefinedBy <http://example.com/demo> .
34
35
36
37 # INSERT query to transform types of things into RDFS Classes, in a specific named graph.
38
39 ## Option 1: Makes use of the the full IRI for the named graph. The named graph is specified in the INSERT clause and patterns in the WHERE clause are matched against statements present in
40 the named graph through the USING clause
41
42 INSERT {
43   GRAPH <http://example.com/demo/General_Country_Info_Graph> {
44     ?t a rdfs:Class ;
45     skos:prefLabel ?tLabelFull ;
46     rdfs:comment "Transformation into RDF Schema performed automatically"@en .
```





Namespaces:

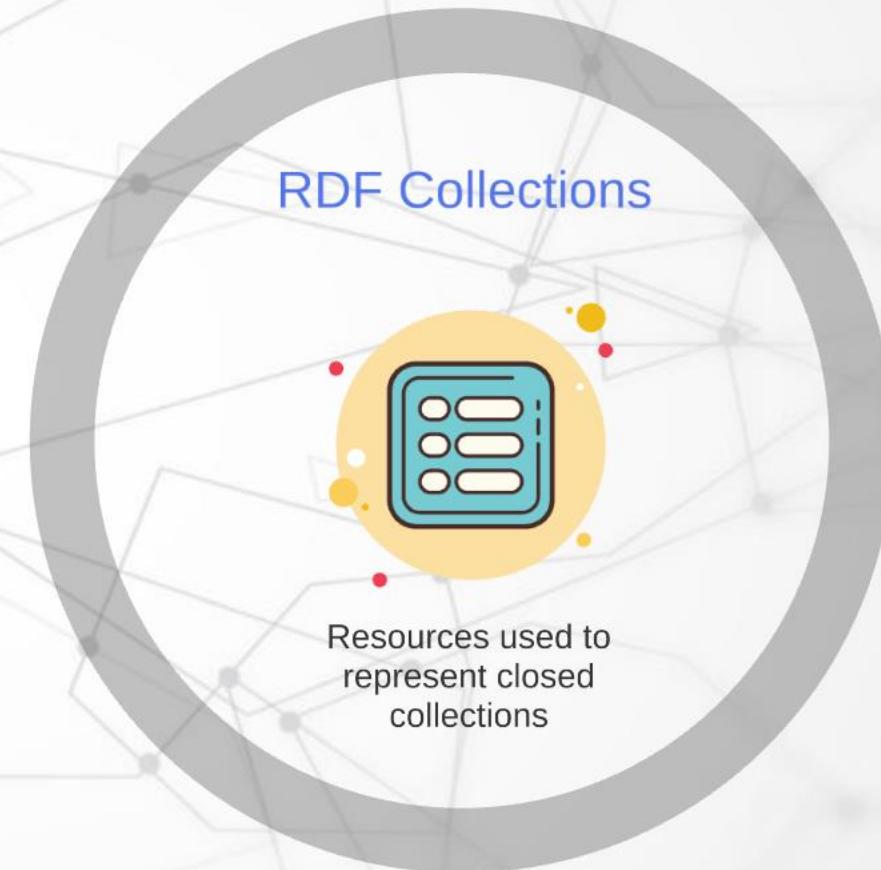
```

rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs: http://www.w3.org/2000/01/rdf-schema#
xsd: http://www.w3.org/2001/XMLSchema#
skos: http://www.w3.org/2004/02/skos/core#
http://example.com/demo/

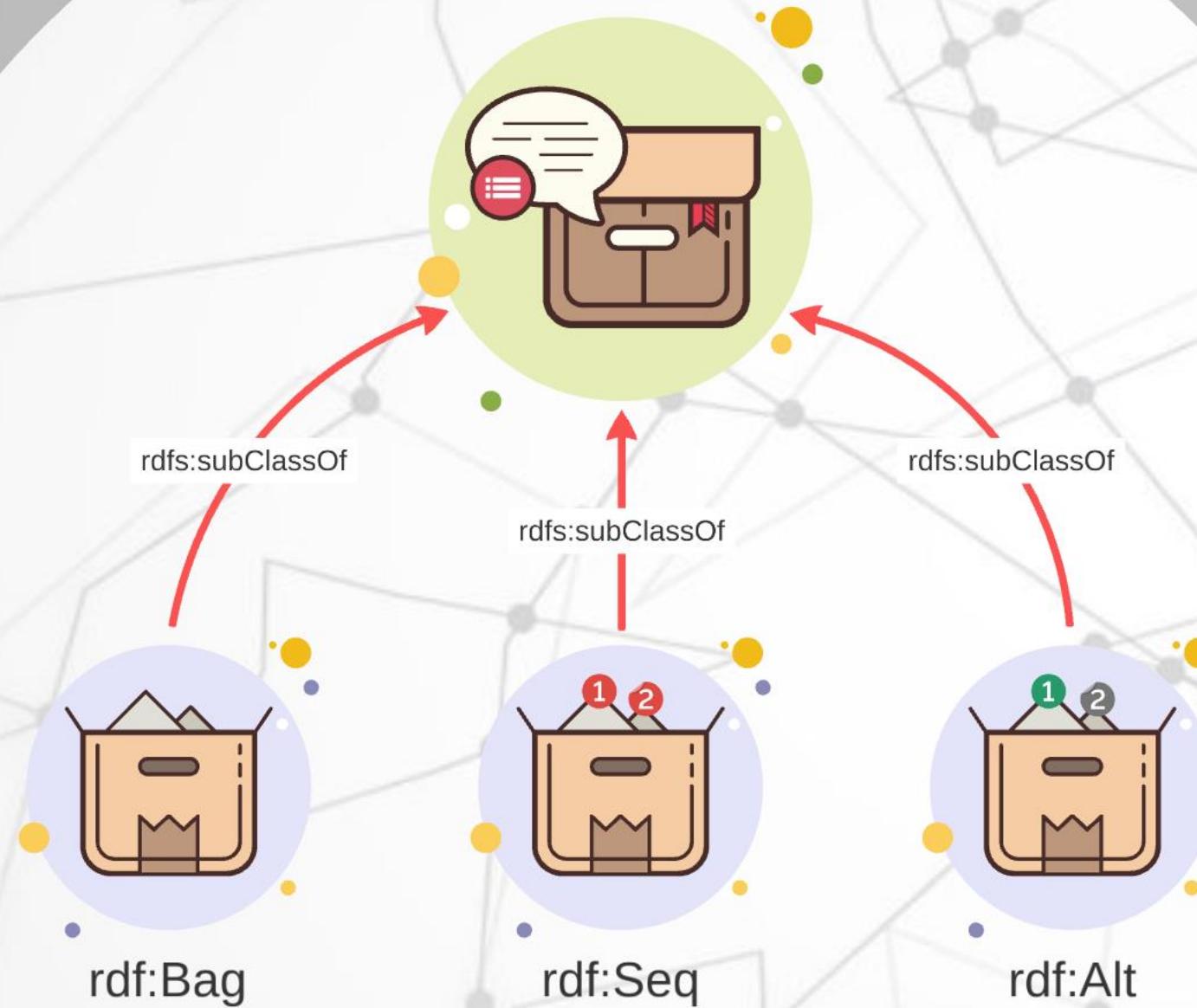
```

# 05 RDF Containers and Collections

# RDF Containers and Collections



# rdfs:Container



rdfs:subClassOf



## rdf:Bag

Conventionally used to indicate that the items in a container are unordered



## rdf:Seq

Conventionally used to indicate that the items in a container are ordered



## rdf:Alt

Conventionally used to indicate that one of the items in the container is to be selected, where the default choice is the first item



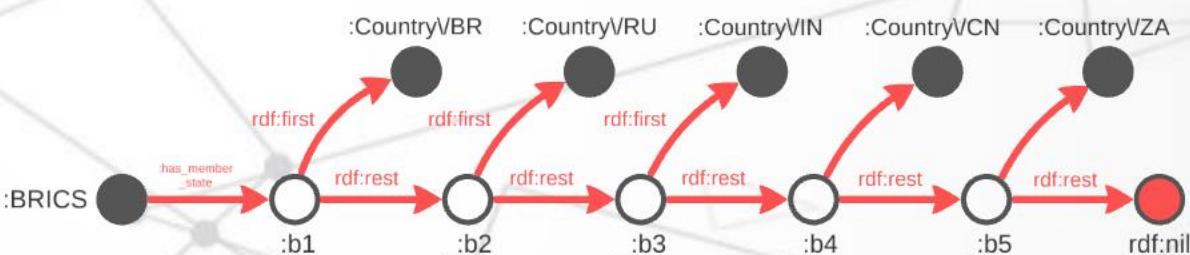
## ≡ Operationalisation.sparql ×

Volumes > Extreme Pro > Courses > Practical RDF and SPARQL > Planning > Demo > ≡ Operationalisation.sparql

```
86
87 # -----
88 # RDF CONTAINERS & COLLECTIONS
89 # -----
90
91 CREATE GRAPH :Language_Families ;
92
93 CREATE GRAPH :Travel_Bucket_Lists ;
94
95 INSERT DATA {
96     GRAPH :Language_Families {
97         :Romance_Languages a rdf:Bag ;
98         |   rdf:_1 :Language\lat ; # Latin
99         |   rdf:_2 :Language\ita ; # Italian
100        |   rdf:_3 :Language\fra ; # French
101        |   rdf:_4 :Language\spa ; # Spanish
102        |   rdf:_5 :Language\por ; # Portuguese
103        |   rdf:_6 :Language\ron ; # Romanian
104        |   rdf:_7 :Language\cat ; # Catalan
105        |   rdf:_8 :Language\oci ; # Occitan
106        |   rdf:_9 :Language\srd ; # Sardinian
107        |   rdf:_10 :Language\cos . # Corsican
108    }
109    GRAPH :Travel_Bucket_Lists {
110        :My_Travel_Bucket_List a rdf:Seq ;
111        |   rdf:_1 :Country\NO ; # Norway
112        |   rdf:_2 :Country\JP ; # Japan
113        |   rdf:_3 :Country\US ; # US
114        |   rdf:_4 :Country\TH ; # Thailand
115        |   rdf:_5 :Country\ZA . # South Africa
116    }
117 };
118
119
120
121
```



Under the hood...



☰ Operationalisation.sparql ×

Volumes &gt; Extreme Pro &gt; Courses &gt; Practical RDF and SPARQL &gt; Planning &gt; Demo &gt; ☰ Operationalisation.sparql



```
119 CREATE GRAPH :International_Organisations ;
120
121 INSERT DATA {
122     GRAPH :International_Organisations {
123
124         ## Declare organisations and associate them with RDF Lists
125
126         :BRICS a :Intergovernmental_Organisation ;
127             skos:prefLabel "BRICS"@en ;
128             rdfs:comment "Intergovernmental organisation consisting of five major emerging economies: Brazil, Russia, India, China, and South Africa"@en ;
129             :has_member_states ( :Country\BR :Country\RU :Country\IN :Country\CN :Country\ZA ) .
130
131         :NATO a :Military_Alliance ;
132             skos:prefLabel "NATO"@en ;
133             rdfs:comment "North Atlantic Treaty Organisation"@en ;
134             :has_member_states ( :Country\AL :Country\BE :Country\BG :Country\CA :Country\HR :Country\CZ :Country\DK :Country\EE :Country\FR
135             :Country\DE :Country\GR :Country\HU :Country\IS :Country\IT :Country\LV :Country\LT :Country\LU :Country\ME :Country\NL :Country\MK
136             :Country\NO :Country\PL :Country\PT :Country\R0 :Country\SK :Country\SI :Country\ES :Country\TR :Country\GB :Country\US ) .
137
138         ## New graph schema structures
139
140         :Intergovernmental_Organisation a rdfs:Class ;
141             rdfs:subClassOf org:Organization .
142
143         :Military_Alliance a rdfs:Class ;
144             rdfs:subClassOf org:Organization .
145
146         :has_member_states a rdf:Property ;
147             rdfs:domain org:Organization ;
148             rdfs:range rdf:List .
149
150
151
152 }
```



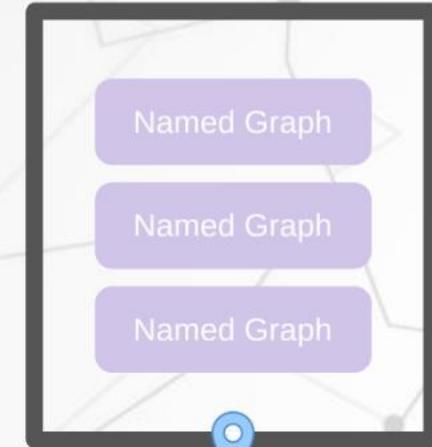
PREFIX rdf: &lt;http://www.w3.org/1999/02/2 Untitled-1 ●

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
4 PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
5 PREFIX org: <http://www.w3.org/ns/org#>
6 PREFIX : <http://example.com/demo/>
7
8 # List all the country names of countries that are member states of the NATO, in the specified order.
9
10 ## Initial query
11
12 SELECT ?cName
13 WHERE {
14   :NATO :has_member_states ?m .
15   ?m rdfs:rest*/rdf:first ?c .
16   ?c :country_name ?cName .
17 }
18
19 ## Leaner query using Property Paths
20
21 SELECT ?cName
22 WHERE {
23   :NATO :has_member_states/rdf:rest*/rdf:first/:country_name ?cName .
24 }
25
26 # What is the first country name in the list of member states of the NATO?
27
28 SELECT ?cName
29 WHERE {
30   :NATO :has_member_states/rdf:first/:country_name ?cName .
31 }
32
33 # What is the third country name in the list of member states of the NATO?
34
35 SELECT ?cName
36 WHERE {
37   :NATO :has_member_states/rdf:rest/rdf:rest/rdf:rest/rdf:first/:country_name ?cName .
38 }
39
40 # In which international organisation(s) is a given country a member state?
41
42 SELECT ?org
43 WHERE {
44   VALUES ?cName { "South Africa" }
45   ?org :has_member_states/rdf:rest*/rdf:first/:country_name ?cName
```

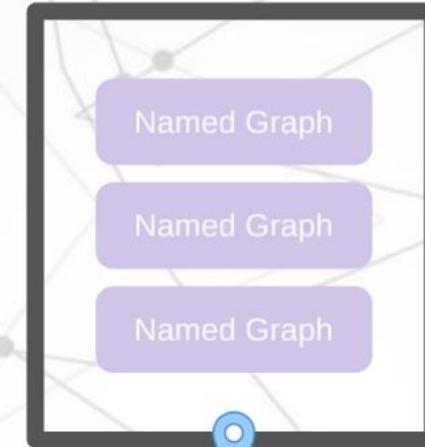
# 06 Federated queries

# Federated Queries

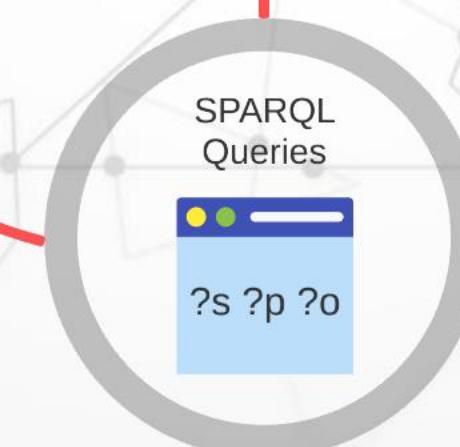
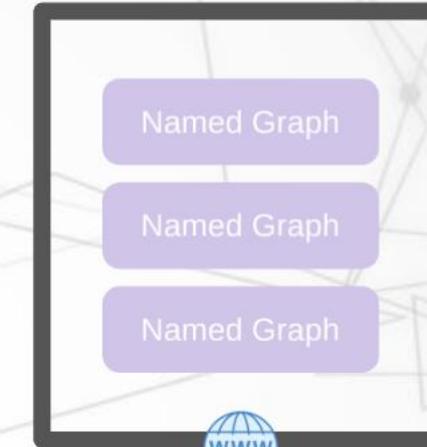
Graph Database 1



Graph Database 2



Graph Database 3



## SPARQL Federated Query

PREFIX ex: <...>

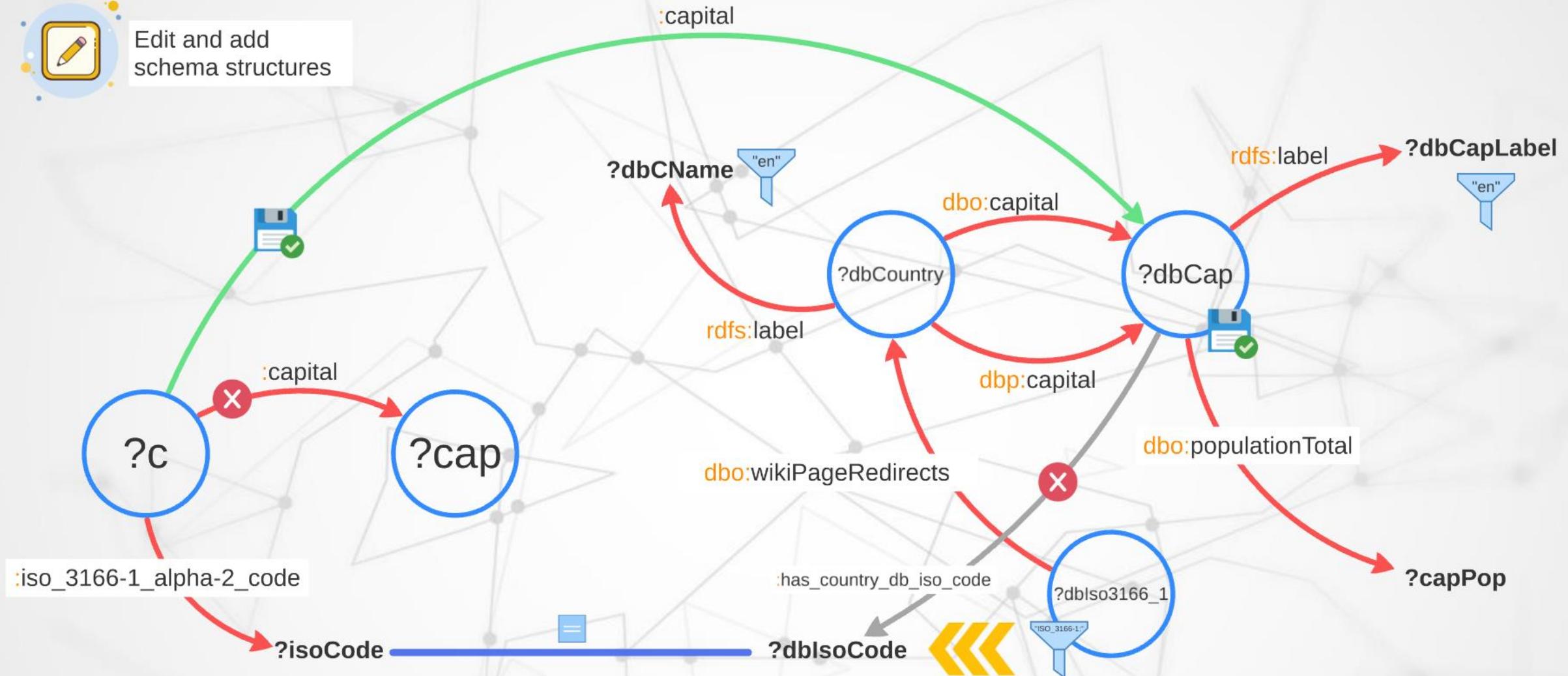
SELECT ...  
WHERE {

**SERVICE** <[endpoint]> {

...  
}  
}



Edit and add  
schema structures



PREFIX `rdf:` <<http://www.w3.org/1999/02/22-rdf-syntax-ns#>>  
PREFIX `rdfs:` <<http://www.w3.org/2000/01/rdf-schema#>>  
PREFIX `xsd:` <<http://www.w3.org/2001/XMLSchema#>>  
PREFIX `skos:` <<http://www.w3.org/2004/02/skos/core#>>  
PREFIX `:` <<http://example.com/demo/>>  
PREFIX `dbr:` <<http://dbpedia.org/resource/>>  
PREFIX `dbp:` <<http://dbpedia.org/property/>>  
PREFIX `dbo:` <<http://dbpedia.org/ontology/>>

≡ Operationalisation.sparql ×

Volumes &gt; Extreme Pro &gt; Courses &gt; Practical RDF and SPARQL &gt; Planning &gt; Demo &gt; ≡ Operationalisation.sparql

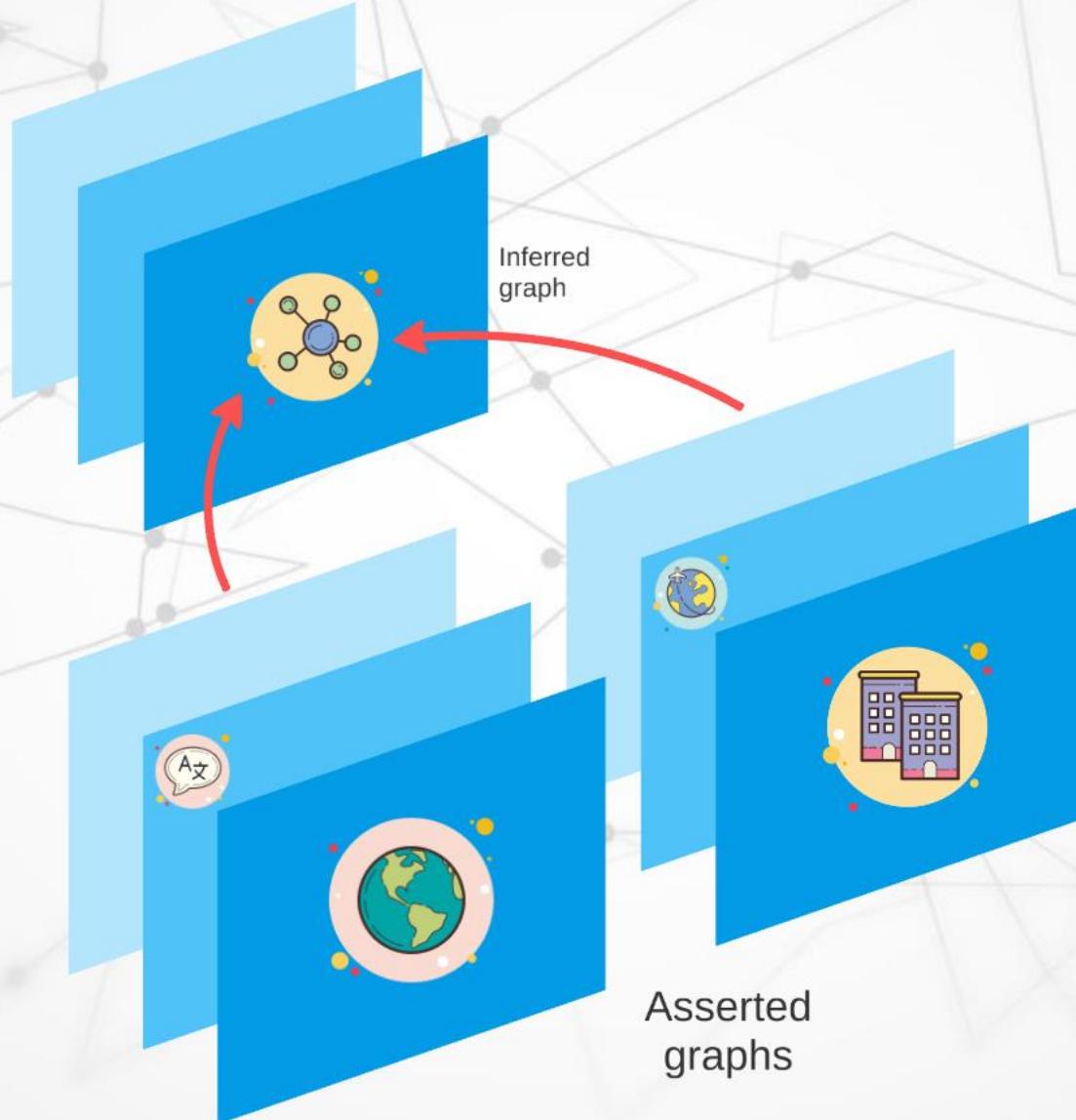


```
150 # =====
151 # FEDERATED QUERIES
152 # =====
153
154 # Create capitals as resources using information from DBpedia.
155
156 WITH :General_Country_Info_Graph
157 INSERT {
158     ?dbCap a :Capital_City ;
159     skos:prefLabel ?dbCapLabel ;
160     dbo:populationTotal ?capPop ;
161     :has_country_db_iso_code ?dbIsoCode .
162 }
163 WHERE {
164     SERVICE <https://dbpedia.org/sparql> {
165         ?dbIso3166_1 dbo:wikiPageRedirects ?dbCountry .
166
167         ?dbCountry a dbo:Country ;
168         rdfs:label ?dbCName ;
169         dbo:capital|dbp:capital ?dbCap .
170
171         ?dbCap rdfs:label ?dbCapLabel .
172
173         OPTIONAL {
174             ?dbCap dbo:populationTotal ?capPop .
175         }
176
177         FILTER(LANG(?dbCName) = "en" && LANG(?dbCapLabel) = "en")
178         FILTER(CONTAINS(STR(?dbIso3166_1), "ISO_3166-1:"))
179     }
180
181     BIND(STRAFTER(STR(?dbIso3166_1), "ISO_3166-1:") AS ?dbIsoCode)
182 };
183
184 # Remove existing literal value capitals and explicitly link countries to new capitals.
185
186 WITH :General_Country_Info_Graph
187 DELETE {
188     ?c :capital ?cap .
189 }
190 INSERT {
191     ?c :capital ?dbCap .
192 }
193 WHERE {
```



# 07 Managing graphs

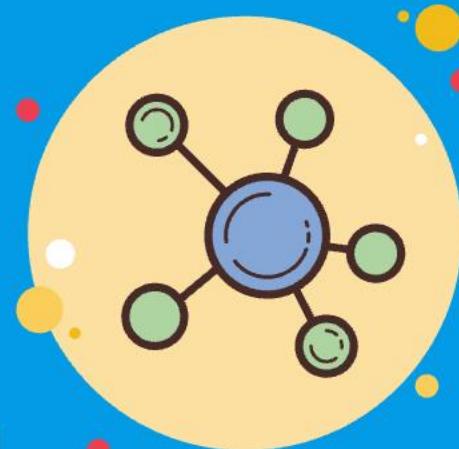
# Named Graph Management



# Inferred graph

- Separate raw data from derived data
- Useful for data provenance & traceability
- Eases responsibility allocation & team dynamics
- Provides a layer for encoding business rules

# Archive graph



# Inferred graph

- Separates
- Useful
- Eases
- Provides



≡ Operationalisation.sparql ×

Volumes &gt; Extreme Pro &gt; Courses &gt; Practical RDF and SPARQL &gt; Planning &gt; Demo &gt; ≡ Operationalisation.sparql



```
231 # =====
232 # GRAPH MANAGEMENT
233 # =====
234
235 # Inferred graph
236
237 CREATE GRAPH :Inferred_Graph ;
238
239 INSERT {
240   GRAPH :Inferred_Graph {
241     ?c :has_neighbour ?cn ;
242     :has_primary_language ?ln1 ;
243     :has_secondary_language ?ln2 ;
244     :has_tertiary_language ?ln3 .
245   }
246 }
247 USING :General_Country_Info_Graph
248 WHERE {
249   ?c a :Country .
250
251   OPTIONAL {
252     ?c :has_country_neighbour/:country_neighbour_value/:country_name ?cn .
253   }
254   OPTIONAL {
255     ?cl1 :spoken_in ?c ;
256     :language_value/:language_name ?ln1 ;
257     :rank 1 .
258   }
259   OPTIONAL {
260     ?cl2 :spoken_in ?c ;
261     :language_value/:language_name ?ln2 ;
262     :rank 2 .
263   }
264   OPTIONAL {
265     ?cl3 :spoken_in ?c ;
266     :language_value/:language_name ?ln3 ;
267     :rank 3 .
268   }
269 };
270
271 # Archive graph and refactoring entities
272
273 ## Refactoring classes
274
```



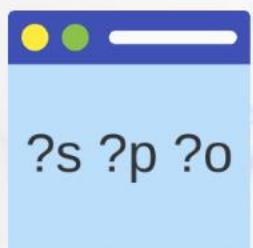
PREFIX : <http://example.com/demo/> • Untitled-1

```
1 PREFIX : <http://example.com/demo/>
2
3 # =====
4 # CREATE
5 # =====
6
7 # Create a new named graph.
8 CREATE GRAPH :International_Organisations
9
10
11 # =====
12 # DROP
13 # =====
14
15 # Remove a specific named graph.
16 DROP GRAPH :International_Organisations
17
18 # Remove the default graph.
19 DROP DEFAULT
20
21 # Remove all named graphs.
22 DROP NAMED
23
24 # Remove all graphs.
25 DROP ALL
26
27 # =====
28 # LOAD
29 # =====
30
31 # Read an RDF document from an IRI or file location and insert its triples into a specific named graph.
32 LOAD <file:../../General_Country_Info_Dataset.ttl> INTO GRAPH :General_Country_Info_Graph
33
34 # =====
35 # CLEAR
36 # =====
37
38 # Remove all triples from a specific named graph.
39 CLEAR GRAPH :General_Country_Info_Graph
40
41 # Remove all triples from the default graph only.
42 CLEAR DEFAULT
43
44 # Remove all triples from all named graphs, where the graphs themselves are not deleted.
45 CLEAR NAMED
46
```

Ln 89, Col 1 Spaces: 4 UTF-8 LF SPARQL ✓ Prettier

# 08 Querying default and named graphs

SPARQL  
Queries

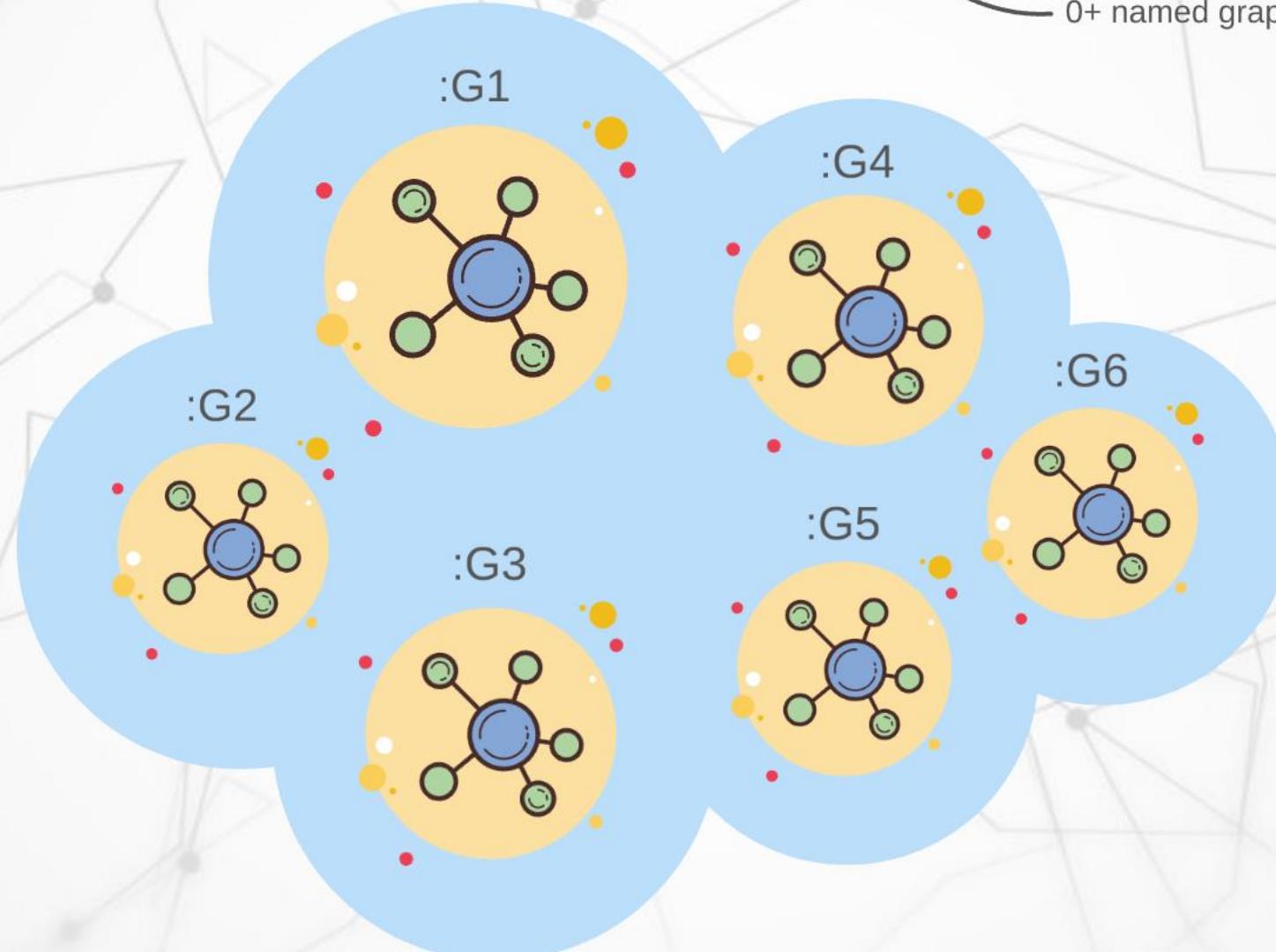


Collection of RDF graphs

1 default (unnamed) graph

&

0+ named graphs



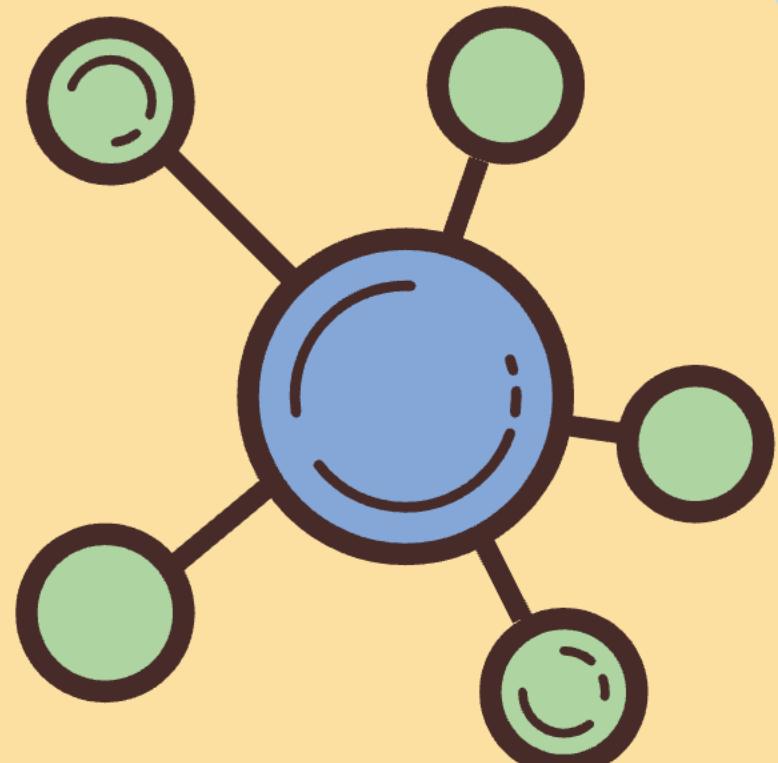
:Latin a :Language .

.G3

:G2

:G1

:Italian :derives\_from :Latin .



```
SELECT ?s ?p ?o  
WHERE {  
    ?s ?p ?o  
}
```

?s	?p	?o
:Latin	rdf:type	:Language
:Italian	:derives_from	:Latin

# Inclusive paradigm

# Exclusive paradigm



# SPARQL Queries

The card features a dark blue header bar with two circular icons: a yellow one on the left and a green one on the right. To the right of the icons are two buttons: "Inclusive paradigm" and "Exclusive paradigm". Below the header, the card has a light blue background. It contains the following text:  
**FROM <...>**  
**FROM NAMED <...>**

PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt;

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
4 PREFIX : <http://example.com/demo/>
5
6 # List all the statements (from the default graph) in which the languages Latin, Italian, French and Portuguese participate as subjects or objects.
7
8 SELECT DISTINCT ?s ?p ?o
9 WHERE {
10   VALUES ?l { :Language\lat :Language\ita :Language\fra :Language\spa :Language\por }
11
12   ?s ?p ?o .
13
14   FILTER(?s = ?l || ?o = ?l)
15 }
16
17 # Define the default graph as the graph consisting of the datasets for :Language_Families and :General_Country_Info_Graph. List all the statements (from the defined default graph) in which
18 # the languages Latin, Italian, French and Portuguese participate as subjects or objects.
19
20 SELECT DISTINCT ?s ?p ?o
21 FROM :Language_Families
22 FROM :General_Country_Info_Graph
23 WHERE {
24   VALUES ?l { :Language\lat :Language\ita :Language\fra :Language\spa :Language\por }
25
26   ?s ?p ?o .
27
28   FILTER(?s = ?l || ?o = ?l)
29 }
30
31 # Define the default graph as the graph consisting of datasets that are not part of any named graph. List all the statements (from the defined default graph) in which the languages Latin,
32 Italian, French and Portuguese participate as subjects or objects.
33
34 SELECT DISTINCT ?s ?p ?o
35 FROM <http://www.bigdata.com/rdf#nullGraph>
36 WHERE {
37   VALUES ?l { :Language\lat :Language\ita :Language\fra :Language\spa :Language\por }
38
39   ?s ?p ?o .
40
41   FILTER(?s = ?l || ?o = ?l)
42 }
43
44 # List the names of languages that start with the capital letter 'S' that are common in both the :General_Country_Info_Graph and :Language_Families graph.
45
46 SELECT DISTINCT ?Name
```



# 09 Course wrap-up