Σημασιολογικός και Κοινωνικός Ιστός

Διάλεξη 07 - SPARQL

Γεώργιος Δημητρακόπουλος
dimitrakopoulos@ionio.gr

Τμήμα Πληροφορικής
Ιόνιο Πανεπιστήμιο

SPARQL

- **Simple Protocol and RDF Query Language**
- **Ρωτάει γράφους RDF (Τριπλέτες)**

- RDF γράφοι: (υποκείμενο, κατηγόρημα, αντικείμενο) που προέρχονται από RDF τεκμήρια, XML ή σχεσιακές βάσεις δεδομένων
- SPARQL: γλώσσα επερωτήσεων και πρωτόκολλο για την ανάκτηση πληροφορίας από RDF γράφους
  - με μορφή URI, κενών κόμβων και τιμών (literals)
  - εξαγωγή RDF υπογράφων
  - δημιουργία νέων RDF γράφων που βασίζονται στην πληροφορία των επερωτώμενων γράφων
- Βασική τεχνική η δημιουργία υποδείγματος γράφου (graph pattern) για το οποίο γίνεται αναζήτηση/ταίριασμα σε ένα RDF γράφο

Χρήσεις

- Εύρεση τιμών από μερικώς γνωστούς γράφους
- Ανάκτηση πληροφορίας για ένα αντικείμενο με μη γνωστές ιδιότητες (properties)
- Εκτέλεση επερωτήσεων σε RDF γράφους
- Εκτέλεση επερωτήσεων με περιορισμούς σύμφωνα με τον τύπο των δεδομένων
- Ερώτηση σε απομακρυσμένους RDF servers
- Χρήση RDF query service σε Web Services

SPARQL - Παράδειγμα

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

SELECT ?c
WHERE
{
?c rdf:type rdfs:Class .
}
```

- Ανακτά όλες τις τριπλέτες για τις οποίες:
- -το property είναι rdf:type
- -το object είναι rdfs:Class
- Δηλ. ανακτά όλες τις κλάσεις της RDF!

SPARQL – Παράδειγμα 2

- Ανάκτησε όλα τα στιγμιότυπα μιας κλάσης (π.χ. course) :
- (οι δηλώσεις των rdf, rdfs prefixes έχουν παραληφθεί για απλότητα)

```
PREFIX uni: <http://www.mydomain.org/uni-ns#>
SELECT ?i
WHERE
{
?i rdf:type uni:course .
}
```

Γενική Σύνταξη

- IRI: περικλείονται σε <>
- τα IRI μπορεί να είναι URI, Qnames (ζεύγος "x:p” {URI, local-name}, π.χ. {"http://example.com/ns/foo", "p"}.)
- PREFIX:δεσμεύει ένα πρόθεμα σε ένα namespace IRI.
  - PREFIX foaf: http://xmlns.com/foaf/0.1/
- BASE: δηλώνει ένα βασικό IRI, που χρησιμοποιείται για την ανάλυση – δήλωση σχετικών IRI
  - BASE http://example.org/book/

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
SELECT ?title
WHERE { <book1> dc:title ?title }
```

- Μεταβλητές: δηλώνονται με πρώτο χαρακτήρα τα «?» ή «$». Το $abc και το ?abc είναι η ίδια μεταβλητή.
- Απλοί όροι (literals): δηλώνονται με “” ή ‘’. Μπορούν να φέρουν ετικέτα της γλώσσας που ανήκουν ("Literal"@language) ή τον τύπο δεδομένων που ανήκουν ("10"^^xsd:integer)

Γενική Σύνταξη

- Όπως η SQL

```
SELECT DISTINCT ?concept
WHERE {
?s a ?concept .
}
```

Γενική Σύνταξη

```
# Δηλώσεις προθεμάτων
PREFIX foo: <http://example.com/resources/>
...
# Δηλώσεις συνόλων δεδομένων
FROM ...
# Πρόταση αποτελέσματος
SELECT ...
# Μοτίβο ερωτήματος
WHERE {
...
}
# Μετατροπείς ερωτημάτων
ORDER BY ...
```

Γενική Σύνταξη

- Υποστηρίζονται επιπλέον οι παρακάτω εντολές ακριβώς όπως στην SQL
- ORDER BY
- GROUP BY
- HAVING
- LIMIT

Φίλτρο

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX type: <http://dbpedia.org/class/yago/>
PREFIX prop: <http://dbpedia.org/property/>
SELECT DISTINCT ?country_name ?population
WHERE {
?country a type:LandlockedCountries ;
rdfs:label ?country_name ;
prop:populationEstimate ?population .
FILTER (?population > 15000000) .
}
```

Aggregations

```
PREFIX books: <http://books.example/>
SELECT (SUM(?lprice) AS ?totalPrice)
WHERE {
?org books:affiliates ?auth .
?auth books:writesBook ?book .
?book books:price ?lprice .
}
GROUP BY ?org
HAVING (SUM(?lprice) > 10)
```

Subqueries

```
PREFIX people: <http://people.example/>
SELECT ?y ?minName
WHERE {
people:alice people:knows ?y .
{
SELECT ?y (MIN(?name) AS ?minName)
WHERE {
?y people:name ?name .
} GROUP BY ?y
}
}
```

Αρνηση

- PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
- PREFIX foaf: <http://xmlns.com/foaf/0.1/>
- SELECT ?person
- WHERE
- {
- ?person rdf:type foaf:Person .
- FILTER NOT EXISTS { ?person foaf:name ?name }
- }

Εισαγωγη-Διαγραφη

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

INSERT DATA { <http://www.example.org/alice#me> foaf:knows ?x .
?x foaf:name "Dorothy" . } ;

DELETE { ?person foaf:name ?mbox }
WHERE { <http://www.example.org/alice#me> foaf:knows ?person .
?person foaf:name ?name FILTER ( lang(?name) = "EN" ) .}
```

ASK

- ASK: απαντά με μια true ή false τιμή

- PREFIX prop: <http://dbpedia.org/property/>
- ASK
- {
- <http://dbpedia.org/resource/Amazon_River> prop:length ?amazon .
- <http://dbpedia.org/resource/Nile> prop:length ?nile .
- FILTER(?amazon > ?nile) .
- }

DESCRIBE

- DESCRIBE:δημιουργεί RDF γράφο που περιγράφει τους πόρους που ανακτώνται
- DESCRIBE http://dbpedia.org/resource/Paris_Hilton
- Δοκιμάστε το ερώτημα: http://dbpedia.org/sparql

Implicit join

- Ανάκτησε όλους τους lecturers και τα τηλέφωνα τους:

```
SELECT ?x ?y
WHERE
{ ?x rdf:type uni:Lecturer ;
 uni:phone ?y . }
```

- Implicit join: Επιβάλουμε στο δεύτερο σκέλος να ανακτήσουμε μόνον εκείνες τις τριπλέτες των οποίων το resource είναι η μεταβλητή ?x
- Συντακτική συντόμευση
- Μια Semicolon (;) μας δηλώνει ότι η δεύτερη τριπλέτα έχειτο ίδιο subject με την προηγούμενη

Implicit join

- Εναλλακτικά, γράφουμε:

```
SELECT ?x ?y
WHERE
{
?x rdf:type uni:Lecturer .
?x uni:phone ?y .
}
```

Τριπλέτες

- Δεδομένα

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
_:a foaf:name "Johnny Lee Outlaw" .
_:a foaf:mbox <mailto:jlow@example.com> .
_:b foaf:name "Peter Goodguy" .
_:b foaf:mbox <mailto:peter@example.org> .
```

- Ερώτημα

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE
{ ?x foaf:name ?name .
 ?x foaf:mbox ?mbox }
```

Αποτέλεσμα

name	mbox
"Johnny Lee Outlaw"	<mailto:jlow@example.com>
"Peter Goodguy"	<mailto:peter@example.org>

Τριπλέτες

- Δεδομένα

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix : <http://example.org/book/> .
@prefix ns: <http://example.org/ns#> .
:book1 dc:title "SPARQL Tutorial" .
:book1 ns:price 42 .
:book2 dc:title "The Semantic Web" .
:book2 ns:price 23 .
```

- Ερώτημα

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX ns: <http://example.org/ns#>
SELECT ?title ?price
WHERE { ?x ns:price ?price .
 FILTER ?price < 30 .
 ?x dc:title ?title . }
```

Αποτέλεσμα

title	price
"The Semantic Web"	23

Ετικέτα Optional

- Δεδομένα

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix : <http://example.org/book/> .
@prefix ns: <http://example.org/ns#> .
:book1 dc:title "SPARQL Tutorial" .
:book1 ns:price 42 .
:book2 dc:title "The Semantic Web" .
:book2 ns:price 23 .
```

- Ερώτημα

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX ns: <http://example.org/ns#>
SELECT ?title ?price
WHERE { ?x dc:title ?title .
 OPTIONAL { ?x ns:price ?price .
 FILTER ?price < 30 }}
```

Η έκφραση OPTIONAL{pattern} είναι ίση με {pattern} OR { NOT(pattern)}

title	price
“SPARQL Tutorial“	
"The Semantic Web"	23

Construct (δημιουργία τριπλετών)

- Δεδομένα:

```
@prefix org: <http://example.com/ns#> .
_:a org:employeeName "Alice" .
_:a org:employeeId 12345 .
_:b org:employeeName "Bob" .
_:b org:employeeId 67890 .
```

- Ερώτημα:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX org: <http://example.com/ns#>
CONSTRUCT
{ ?x foaf:name ?name }
WHERE
{ ?x org:employeeName ?name }
```

Αποτέλεσμα:

```
@prefix org: <http://example.com/ns#> .

_:x    foaf:name        "Alice" .
_:y    foaf:name        "Bob" .
```

Αποτέλεσμα σε RDF

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-
rdfsyntax-ns#"
xmlns:foaf="http://xmlns.com/foaf/0.1/" >
<rdf:Description>
<foaf:name>Alice</foaf:name>
</rdf:Description>
<rdf:Description>
<foaf:name>Bob</foaf:name>
</rdf:Description>
</rdf:RDF>
```

Παράδειγμα

- Στις παρακάτω διαφάνειες, θα συνενώσουμε 2 RDF αρχεία (μαζί τις τις RDFS κλάσεις τους) και θα εκτελέσουμε ένα σύνθετο ερώτημα σε SPARQL
- Αρχικά θα βλέπουμε τα δεδομένα σε XML, μετά σε SPARQL-Turtle μορφή και τέλος σε RDF γράφο.

Δεδομένα XML

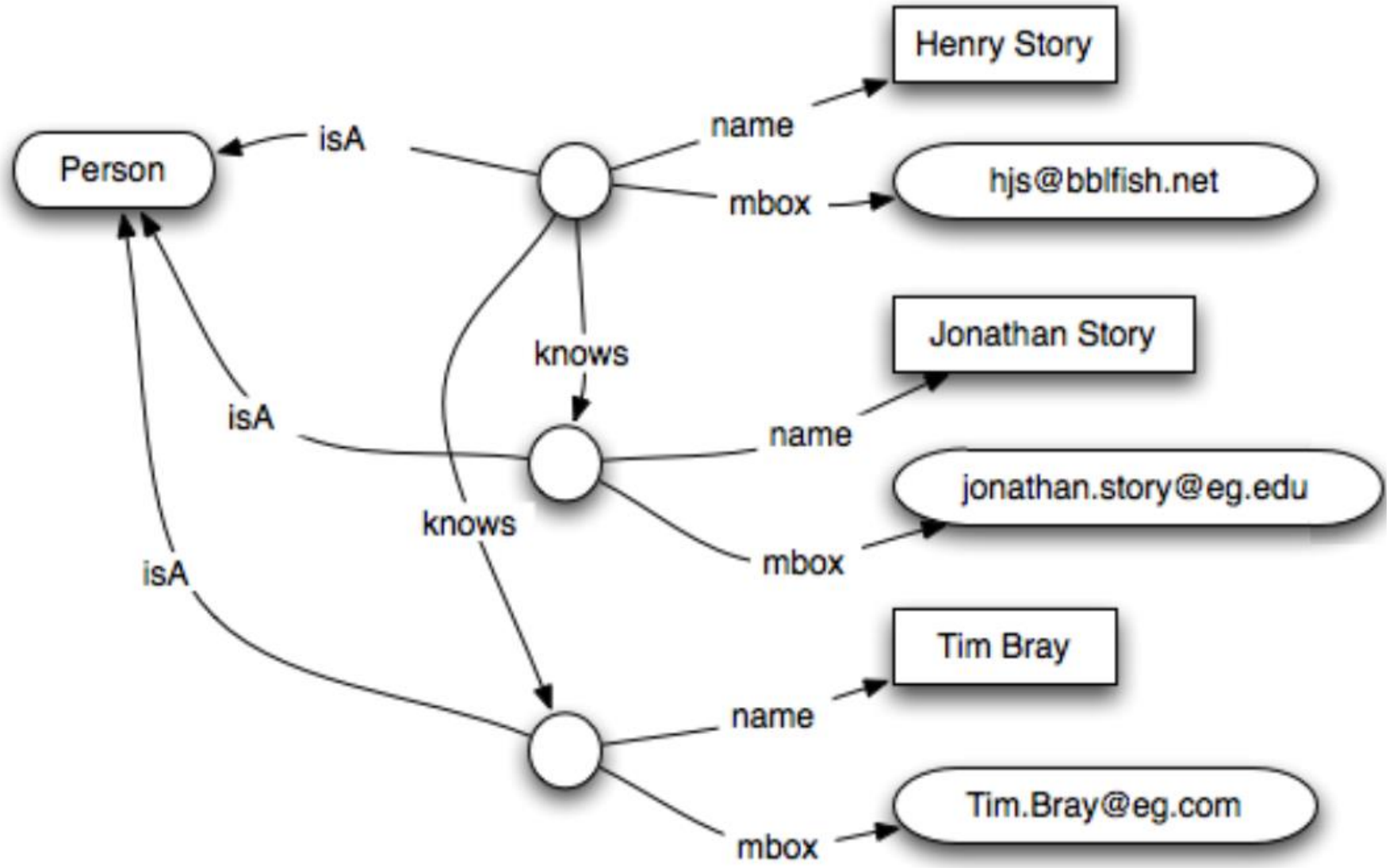
```
<Person>
 <name>Henry Story</name>
 <mbox>hs@bblfish.net</mbox>
 <knows>
<Person><name>Tim Bray</name> <mbox>tb@eg.com</mbox>
</Person>
<Person><name>Jonathan Story</name> <mbox>js@eg.edu</mbox>
</Person>
 </knows>
</Person>
```

Δεδομένα σε Turtle

```
[ a :Person;
:name "Henry Story";
:mbox <mailto:hs@insead.edu>;
:knows [ a :Person;
:name "Tim Bray";
:mbox <mailto:tb@eg.com
];
:knows [ a :Person;
:name "Jonathan Story";
:mbox <mailto:js@eg.edu> ];
] .
```

A ➔ rdf:type

Δεδομένα - γράφος



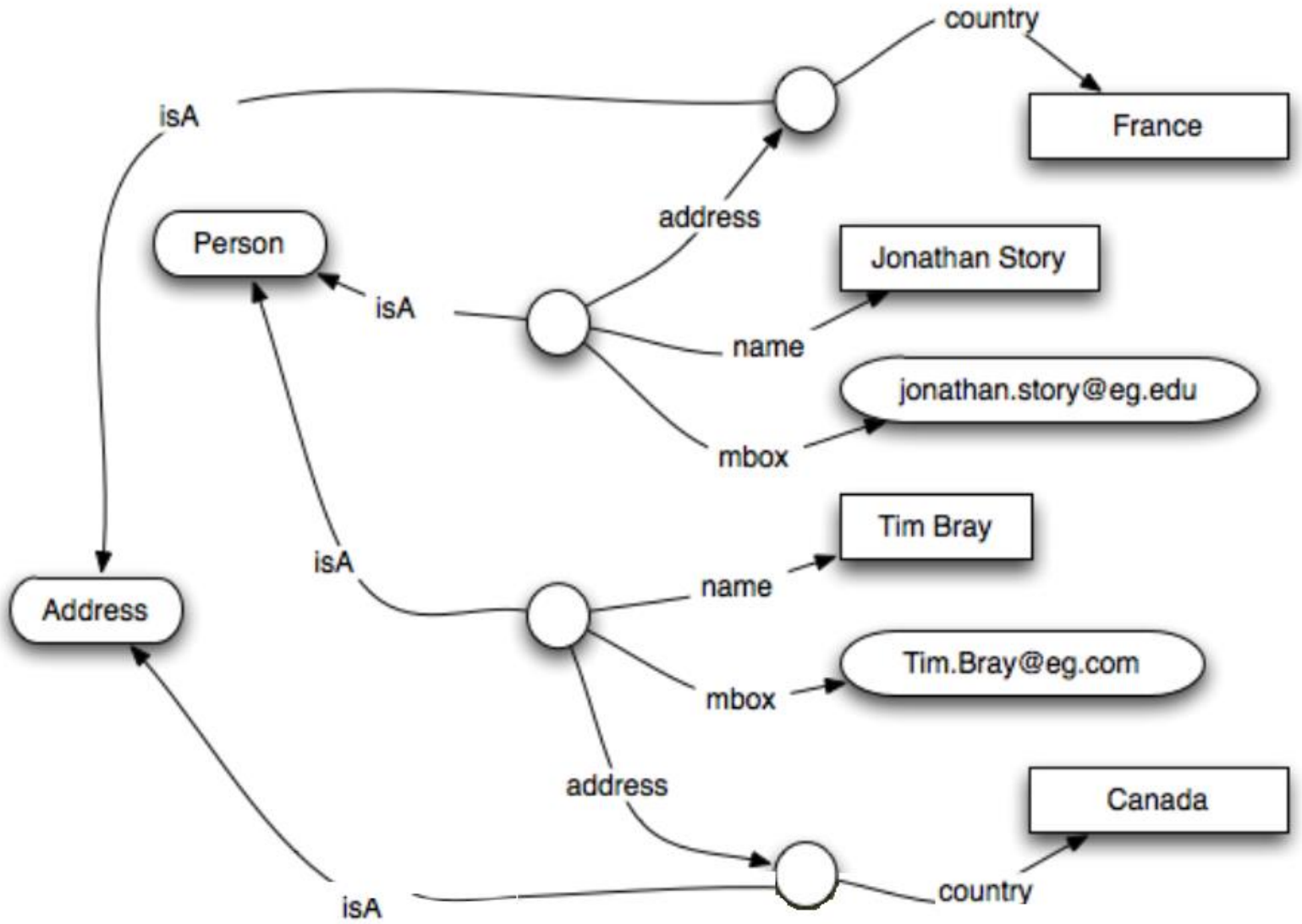
Δεδομένα #2 XML

```
<AddressBook>
<Person>
<name>Jonathan Story</name>
<mbox>Jonathan.Story@eg.edu</mbox>
<address>
<Country>France</Country>
</address>
</Person>
<Person>
<name>Tim Bray</name>
<mbox>Tim.Bray@eg.Com</mbox>
<address>
<Country>Canada</Country>
</address>
</Person>
</AddressBook>
```

Δεδομένα #2 σε Turtle

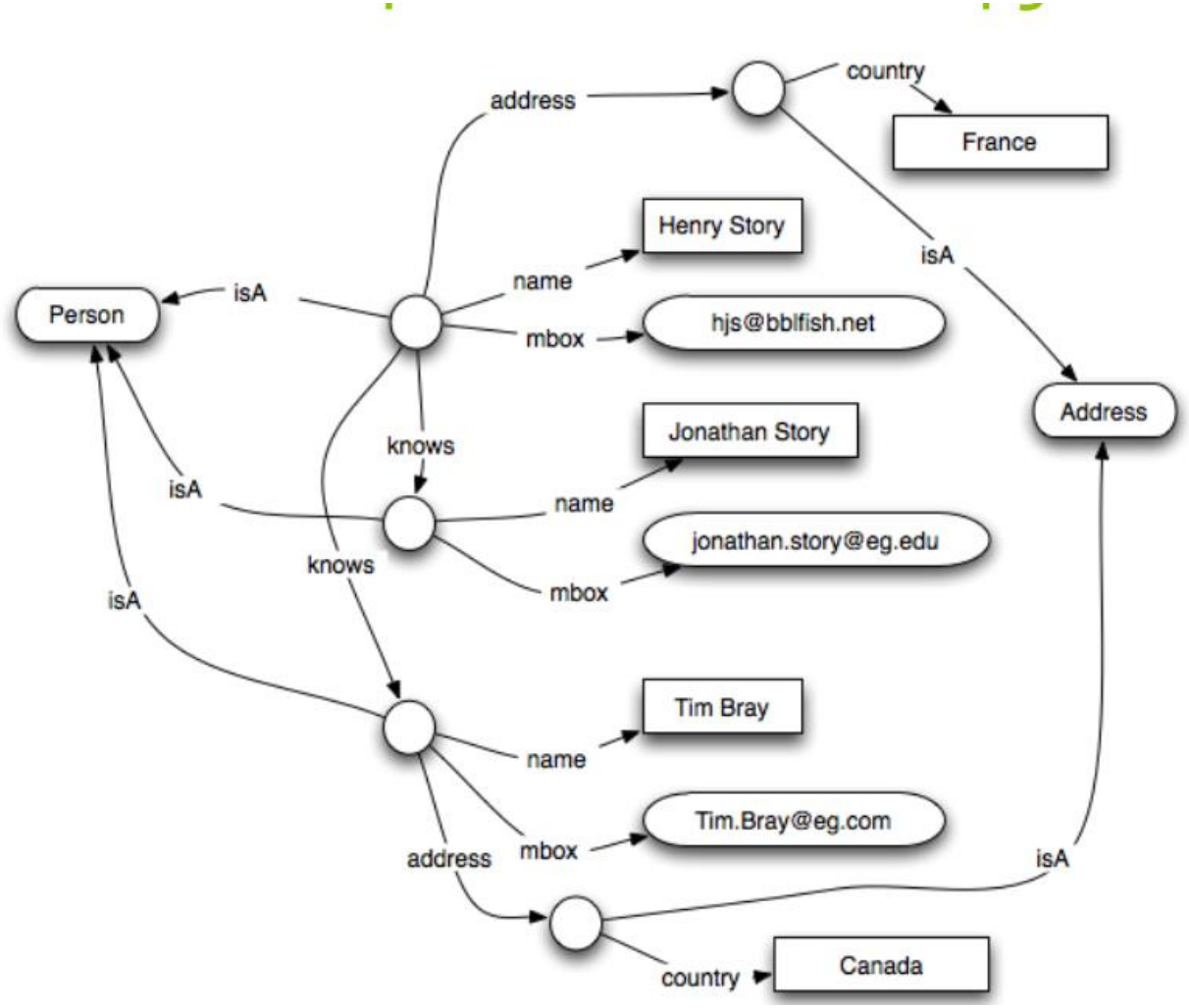
```
[ a :Person;
:name "Tim Bray";
:mbox <mailto:Tim.Bray@eg.com>
:address [ a :Address;
:country "Canada"@en
] ].
[ a :Person;
:name "Jonathan Story";
:mbox <mailto:Jonathan.Story@eg.edu>
:address [
a :Address;
:country "France"@en ]
].
```

Δεδομένα #2 - Γράφος



Αποτέλεσμα συνένωσης

- Οι 2 γράφοι μπορούν να συνενωθούν στον ακόλουθο



Παράδειγμα SPARQL

- Ποιον/ποιούς ξέρει ο Henry που ζει στον Καναδά και ποιο το email του;

- Μπορεί να απαντηθεί μόνο με SPARQL μιας και οι γλώσσες επερώτησης σε XML δουλεύουν μόνο τοπικά σε κάθε έγγραφο.

Απάντηση σε SPARQL

```
SELECT ?name ?mail
WHERE {
[a :Person;
:name "Henry Story";
:knows [ :name ?name;
:mbox ?mail;
:address [ a :Address;
:country
"Canada"@en;
] ] ].
}
```

SPARQL Query Editor

- https://dbpedia.org/sparql

- Drugbank (βάση φαρμάκων)
- https://drugbank.bio2rdf.org/sparql

SPARQL Query Editor

- Βρείτε όλα τα φάρμακα μαζί με τη δοσολογία τους αλλά και τα αντίστοιχα συμπτώματα
- Δοκιμάστε το ερώτημα: http://drugbank.bio2rdf.org/sparql

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX db: <http://bio2rdf.org/drugbank_vocabulary:>
SELECT ?drug_name ?dosage ?indication
WHERE { ?drug a db:Drug .
?drug rdfs:label ?drug_name .
?drug db:dosage ?dosage .
?drug db:indication ?indication . }
```

- 3000 αποτελέσματα

SPARQL Query Editor

- Χωρίς συμπτώματα:

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX db: <http://bio2rdf.org/drugbank_vocabulary:>
SELECT ?drug_name
WHERE {
?drug a db:Drug .
?drug rdfs:label ?drug_name .
}
```

- 7700 αποτελέσματα

SPARQL Query Editor

- Το 2ο ερώτημα είχε παραπάνω αποτελέσματα γιατί δεν έχουν όλα τα φάρμακα πληροφορίες για δοσολογία και συμπτώματα. Η προσθήκη της λέξης OPTIONAL στο μοτίβο ερωτήματος επιτρέπει το ταίριασμα φαρμάκων που δεν συμμετέχουν αναγκαστικά σε τριάδες οι οποίες αναφέρονται σε δοσολογία ή/και συμπτώματα:

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX db: <http://bio2rdf.org/drugbank_vocabulary:>
SELECT ?drug_name ?dosage ?indication
WHERE {
?drug a db:Drug .
?drug rdfs:label ?drug_name .
OPTIONAL { ?drug db:dosage ?dosage . }
OPTIONAL { ?drug db:indication ?indication . }
}
```

Παράδειγμα

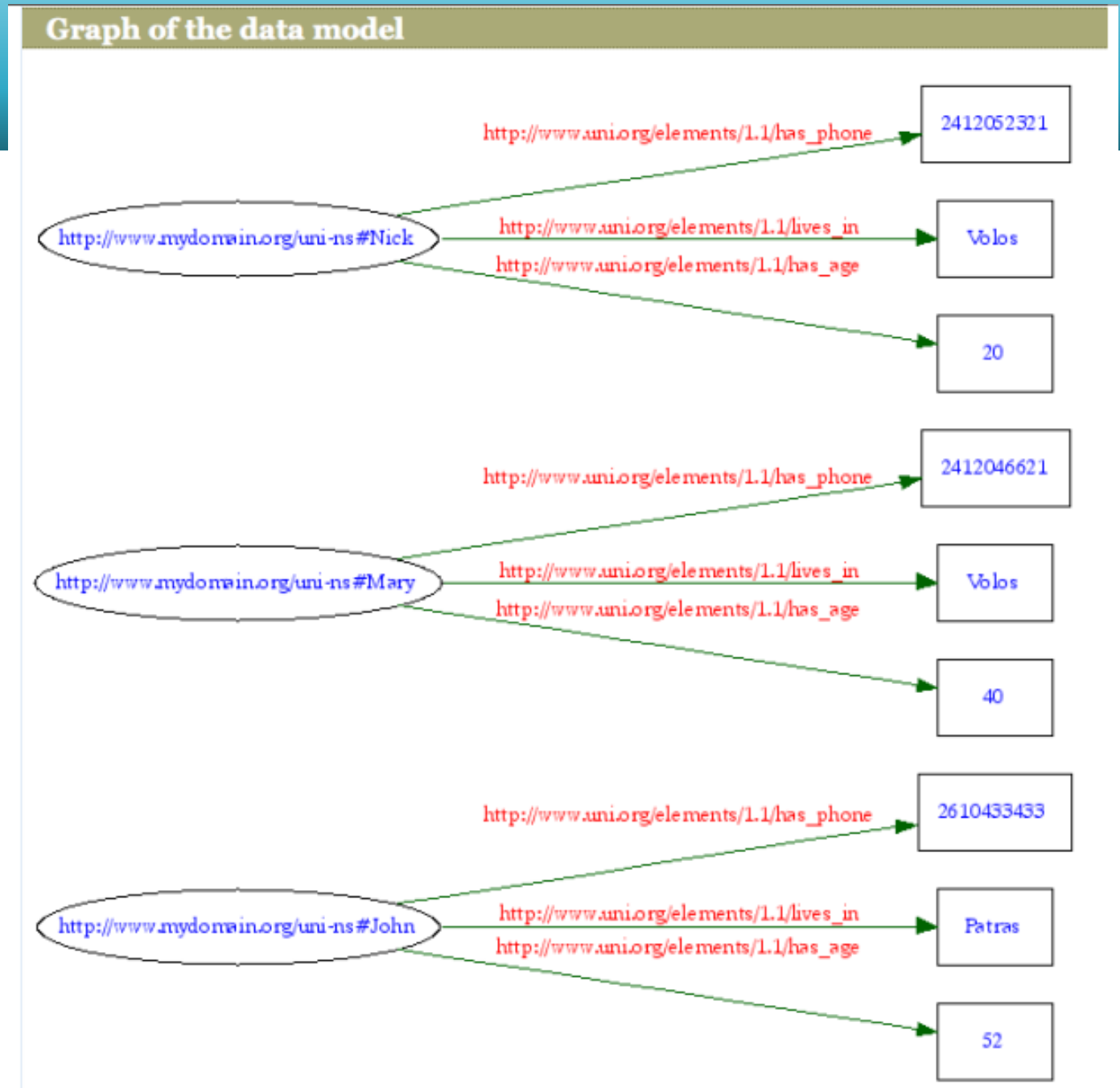
Έστω ότι θέλουμε να κάνουμε SPARQL Query σε μια οντολογία που περιέχει και τα παρακάτω instances

```
<rdf:Description rdf:about="http://www.mydomain.org/uni-ns#John">
    <uni:has_phone> 2610433433 </uni:has_phone>
    <uni:lives_in> Patras </uni:lives_in>
    <uni:has_age>52</uni:has_age>
</rdf:Description>

<rdf:Description rdf:about="http://www.mydomain.org/uni-ns#Mary">
    <uni:has_phone> 2412046621</uni:has_phone>
    <uni:lives_in> Volos </uni:lives_in>
    <uni:has_age>40</uni:has_age>
</rdf:Description>

<rdf:Description rdf:about="http://www.mydomain.org/uni-ns#Nick">
    <uni:has_phone> 2412052321</uni:has_phone>
    <uni:lives_in> Volos </uni:lives_in>
    <uni:has_age>20</uni:has_age>
</rdf:Description>
```

Παράδειγμα


Graph of the data model
http://www.mydomain.org/uni-ns#Nick
http://www.uni.org/elements/1.1/has_phone
2412052321
http://www.uni.org/elements/1.1/lives_in
Volos
http://www.uni.org/elements/1.1/has_age
20
http://www.mydomain.org/uni-ns#Mary
http://www.uni.org/elements/1.1/has_phone
2412046621
http://www.uni.org/elements/1.1/lives_in
Volos
http://www.uni.org/elements/1.1/has_age
40
http://www.mydomain.org/uni-ns#John
http://www.uni.org/elements/1.1/has_phone
2610433433
http://www.uni.org/elements/1.1/lives_in
Patras
http://www.uni.org/elements/1.1/has_age
52

Παράδειγμα

Ερώτημα για να πάρουμε τα ονόματα και τα τηλέφωνα όσων μένουν στον Βόλο:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX uni: <http://www.uni.org/...#>
SELECT ?name, ?phone
WHERE {
?name uni:lives_in "Volos" .
?name uni:has_phone ?phone .
}
```

Θα επιστραφεί:

name	phone
http://www.mydomain.org/uni-ns#Mary	2412046621
http://www.mydomain.org/uni-ns#Nick	2412052321

Παράδειγμα

Ερώτημα για να πάρουμε τα ονόματα και τα τηλέφωνα όσων μένουν στον Βόλο **και** είναι κάτω από 30 ετών:

```
...
SELECT ?name, ?phone
WHERE {?name uni:lives_in "Volos" .
       ?name uni:has_phone ?phone .
       ?name uni:has_age ?age .
       FILTER (?age < 30) .
}
```

Θα επιστραφεί:

name	phone
http://www.mydomain.org/uni-ns#Nick	2412052321