

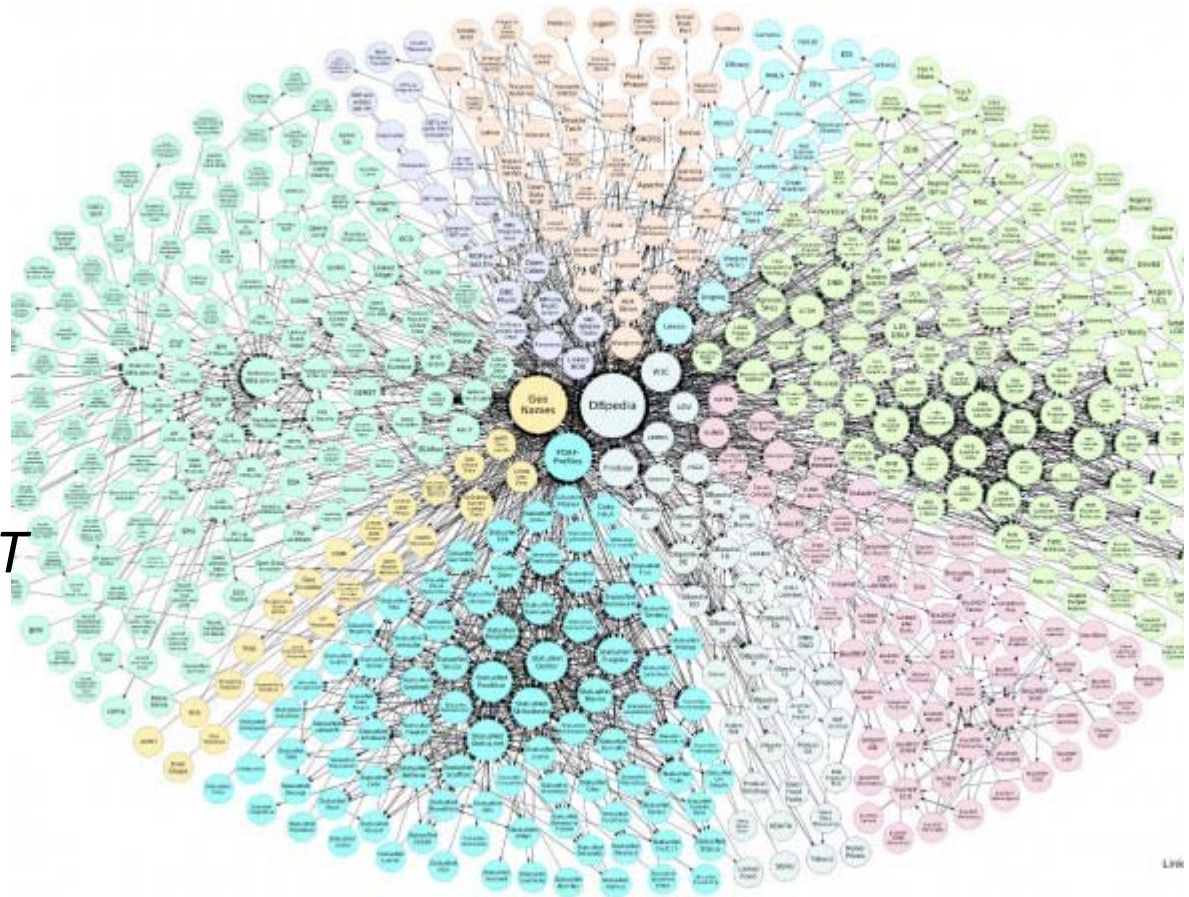
Interrogation/Manipulation de données RDF - SPARQL – (suite)

Catherine COMPAROT

IRIT/UT2J

Equipe MELODI

Mars 2017



SPARQL 1.1 : Des fonctionnalités pour manipuler des datasets RDF

- Interroger plusieurs endpoints SPARQL
- Raisonner avec SPARQL
 - Différents niveaux : RDF, RDFS, OWL, RIF
- Mettre à jour des bases RDF avec SPARQL :
 - Mise à jour de graphes : INSERT, INSERT DATA, DELETE DATA, DELETE, DELETE WHERE, LOAD, CLEAR)
 - Gestion de graphes : CREATE, DROP, COPY, MOVE, ADD
- Etc.

<https://www.w3.org/TR/sparql11-overview/>

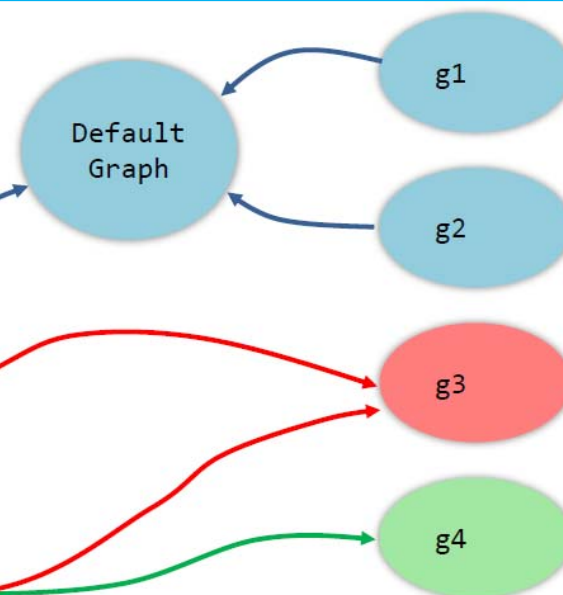
Interroger des Datasets RDF : **FROM**

Une requête SPARQL peut indiquer explicitement le *Dataset* auquel appliquer les patrons de graphes recherchés :

- **FROM** définit le dataset par défaut (DEFAULT GRAPH), fusionnant éventuellement plusieurs graphes
- **FROM NAMED** définit des graphes nommés (avec des URI) interrogés dans le WHERE avec le mot-clé *GRAPH*

?g désigne tour à tour les graphes nommés (NAMED GRAPH) ; ici g3 puis g4

```
SELECT ...  
FROM <g1>  
FROM <g2>  
FROM NAMED <g3>  
FROM NAMED <g4>  
WHERE {  
    .....  
    GRAPH <g3> {  
        .....  
    }  
    GRAPH ?g {  
        .....  
    }  
}
```



Interroger d'autres endpoints SPARQL

SERVICE *sparql_endpoint* { *patron de graphe* } (SPARQL 1.1 « Federated Query ») : recherche le patron de graphe via le endpoint SPARQL spécifié

Exemple : Les labels de l'aluminium dans le endpoint SPARQL de DBpédia

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

```
SELECT distinct ( lcase( str( ?label ) ) AS ?dbLabel )
```

```
WHERE {
```

```
    SERVICE <http://dbpedia.org/sparql> {
```

```
        <http://dbpedia.org/resource/Aluminium> rdfs:label ?label.
```

```
    }
```

```
}
```

Patron de triplet évalué par le serveur SPARQL distant

dbLabel

"aluminium"

"ألومنيوم"

"aluminio"

"alluminio"

"アルミニウム"

"glin"

"aluminio"

"алюминий"

"铝"

Raisonner avec les requêtes

L'inférence, ou système d'implication, est pris en compte par SPARQL 1.1
« Entailment Regimes »

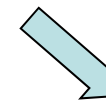
Différents niveaux d'implication sont possibles pour inférer, vérifier : RDF, RDF/S, OWL, RIF

Ex. : Un dataset RDF de 7 triplets (numérotés) en turtle :

```
(1) :V1 rdf:type :Vehicule.  
(2) :V2 rdf:type :Voiture .  
(3) :Voiture rdfs:subClassOf :Vehicule .  
(4) :conduire rdfs:range :Vehicule .  
(5) :Anne :conduire :V3 .  
(6) :Jean :piloter :V4 .  
(7) :piloter owl:equivalentProperty :conduire
```

La requête : « tous les véhicules »

```
SELECT ?v  
WHERE { ?v a :Vehicule . }
```



Rien ?

Raisonner avec les requêtes

- Avec un système d'implication simple :

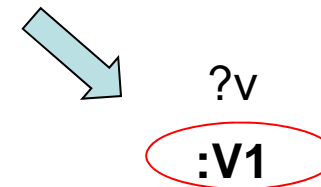
La requête : **SELECT ?v**
WHERE { ?v a :Vehicule . }

a pour résultat :

- (1) :V1 rdf:type :Vehicule.
- (2) :V2 rdf:type :Voiture .
- (3) :Voiture rdfs:subClassOf :Vehicule .
- (4) :conduire rdfs:range :Vehicule .
- (5) :Anne :conduire :V3 .
- (6) :Jean :piloter :V4 .
- (7) :piloter owl:equivalentProperty :conduire

a étant un raccourci pour **rdf:type**

d'après (1) ➔ **:V1 rdf:type :Vehicule**



Raisonner avec les requêtes

- Avec un **système d'implication RDFS** :

La requête : **SELECT ?v**
WHERE { ?v a :Vehicule . }

a pour résultat :

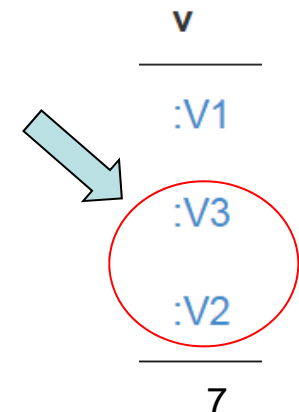
- (1) :V1 rdf:type :Vehicule.
- (2) :V2 rdf:type :Voiture .
- (3) :Voiture rdfs:subClassOf :Vehicule .
- (4) :conduire rdfs:range :Vehicule .
- (5) :Anne :conduire :V3 .
- (6) :Jean :piloter :V4 .
- (7) :piloter owl:equivalentProperty :conduire

règle avec subClassOf : $_x \text{ rdf:type/rdfs:subClassOf } _y \rightarrow _x \text{ rdf:type } _y$

d'où (2) + (3) impliquent \Rightarrow :V2 rdf:type :Vehicule

règle : $_x \text{ rdfs:range } _y \cdot _a _x _b \rightarrow _b \text{ rdf:type } _y$

d'où (4) + (5) impliquent \Rightarrow :V3 rdf:type :Vehicule



Raisonner avec les requêtes

- Avec un **système d'implication OWL** :

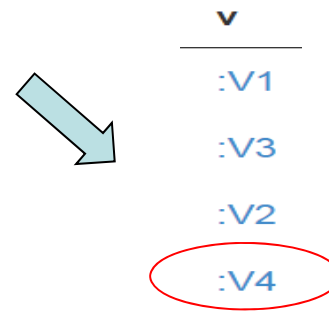
La requête : **SELECT ?v**
WHERE { ?v a :Vehicule . }

a pour résultat :

- (1) :V1 rdf:type :Vehicule.
- (2) :V2 rdf:type :Voiture .
- (3) :Voiture rdfs:subClassOf :Vehicule .
- (4) :conduire rdfs:range :Vehicule .
- (5) :Anne :conduire :V3 .
- (6) :Jean :piloter :V4 .
- (7) :piloter owl:equivalentProperty :conduire

règle avec owl:equivalentProperty : $_x \text{ owl:equivalentProperty } _y \rightarrow _x \text{ rdfs:range } _y$
d'où :

(7) + (6) impliquent \Rightarrow :V4 rdf:type :Vehicule



INSERT DATA { ... } / DELETE DATA { ... }

INSERT DATA { triplets } / DELETE DATA { triplets } : Ajout / suppression de connaissances (triplets)

Pas de variable dans la requête

Exemple : ajout d'une classe *:BigElement*

PREFIX : <http://ex/monVocabulaire#>

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX owl: <http://www.w3.org/2002/07/owl#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

INSERT DATA {

 :BigElement rdf:type owl:Class ;

 rdfs:label "big element"@en ;

 rdfs:comment "set of elements whose weight is over 250"@en.

}

Ajout de 3 triplets

INSERT { ... } WHERE { *patrons de triplets* }

Ajout de connaissances à partir de connaissances existantes

Même principe qu'une requête **CONSTRUCT** (SPARQL « Query Language »)

Exemple 1 : ajouter la relation *:plusLourdQue* entre les atomes

PREFIX : <http://ex/monVocabulaire#>

PREFIX pt: <http://www.daml.org/2003/01/periodictable/PeriodicTable#>

INSERT { ?y :plusLourdQue ?x }

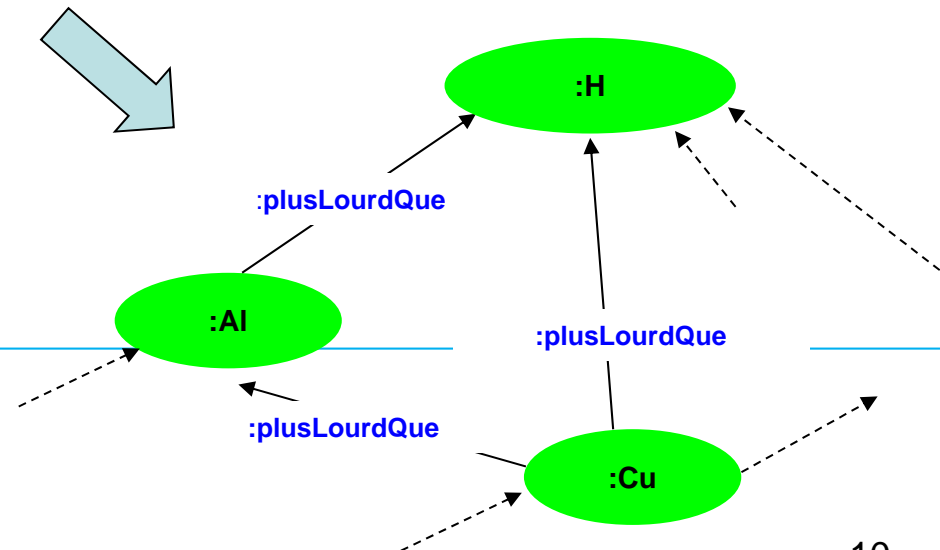
WHERE {

 ?x pt:atomicNumber ?w1.

 ?y pt:atomicNumber ?w2.

 filter (?w2 > ?w1)

}



INSERT { ... } WHERE { *patrons de triplets* }

Exemple 2 : peupler la classe *:BigElement* avec les atomes pesant plus de 250

PREFIX : <http://ex/monVocabulaire#>

PREFIX pt: <http://www.daml.org/2003/01/periodictable/PeriodicTable#>

INSERT { ?e a :BigElement }

WHERE {

 ?e a pt:Element.

 ?e pt:atomicWeight ?w.

 filter (?w > 250)

}



Patron des triplets à ajouter

DELETE { ... } WHERE { *patrons de triplets* }

Suppression de données à partir de patrons de triplets

Exemple : suppression de la classe *:BigElement* et des déclarations d'instances associées

PREFIX : <http://ex/monVocabulaire#>

PREFIX pt: <http://www.daml.org/2003/01/periodictable/PeriodicTable#>

DELETE {

 ?e a :BigElement.

 :BigElement ?p ?o.

}

WHERE {

 OPTIONAL { ?e a :BigElement . }

 OPTIONAL { :BigElement ?p ?o . }

}

Patron des triplets à supprimer

DELETE { ... } INSERT { ... } WHERE { ... }

Modifier de données à partir de patrons de triplets

Exemple 1 : Changer les tags de langue des labels d'éléments

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

```
PREFIX pt: <http://www.daml.org/2003/01/periodictable/PeriodicTable#>
```

```
DELETE { ?e rdfs:label ?lab . }
```

```
INSERT { ?e rdfs:label ?labFR . }
```

```
WHERE {
```

```
    ?e a pt:Element ;
```

```
        rdfs:label ?lab .
```

```
    Filter ( lang(?lab) = "fr" )
```

```
    BIND ( strLang( str( ?lab), "fr-FR") AS ?labFR)
```

```
}
```

DELETE { ... } INSERT { ... } WHERE { ... }

Exemple 2 : Mettre la première lettre de chaque label d'élément en majuscule

PREFIX rdfs: <<http://www.w3.org/2000/01/rdf-schema#>>

PREFIX pt: <<http://www.daml.org/2003/01/periodictable/PeriodicTable#>>

DELETE { ?e rdfs:label ?lab . }

```
INSERT { ?e rdfs:label ?newLab . }
```

WHERE {

```
?e a pt:Element ; rdfs:label ?lab .
```

BIND (str(?lab) AS ?strLab)

BIND (lang(?lab) AS ?langLab)

```

BIND ( strLang( concat( ucase( substr( ?strLab, 1, 1 ) ), substr( ?strLab, 2 ) ), ?langLab )

```

AS ?newLab)

}

DELETE { ... } INSERT { ... } WHERE { ... }

Exemple 3 : Créer des URI à partir du nom des éléments chimique

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX pt: <http://www.daml.org/2003/01/periodictable/PeriodicTable#>

INSERT { ?newUri a pt:Element. }

WHERE {

 ?e a pt:Element .

 ?e pt:name ?nom .

 BIND (uri(concat("http://monVoc/", encode_for_uri(?nom))) AS ?newUri)

}

PREFIX pt: <http://www.daml.org/2003/01/periodictable/PeriodicTable#>

SELECT ?e

WHERE { ?e a pt:Element . }



<http://monVoc/krypton>

pt:Os

<http://monVoc/germanium>

pt:Lu

<http://monVoc/lanthanum>

pt:Hf

pt:Co

pt:Tc

pt:Np

Mise en oeuvre : IHM de fuseki 2

localhost:3030/dataset.html?tab=upload&ds=/periodicTable

Apache Jena Fuseki [dataset](#) [manage datasets](#) [help](#) Server status: ●

Dataset:

[query](#) [upload files](#) [edit](#) [info](#)

SPARQL query

To try out some SPARQL queries against the selected dataset, enter your query here.

EXAMPLE QUERIES

[Selection of triples](#) [Selection of classes](#)

PREFIXES

[rdf](#) [rdfs](#) [owl](#) [xsd](#)

SPARQL ENDPOINT CONTENT TYPE (SELECT) CONTENT TYPE (GRAPH)

```
1 PREFIX : <http://www.daml.org/2003/01/periodictable/PeriodicTable#>
2 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 prefix pt:<http://www.daml.org/2003/01/periodictable/PeriodicTable#>
```

:3030/index.html