

Données Semi-Structurées:

SPARQL

Enseignant: *Dario COLAZZO*
Chargée de TD/TP: *Beatrice NAPOLITANO*

15 Février 2021

1 Avant de commencer

Jena est un framework de sémantique web pour Java. Ce framework fournit une API pour l'extraction et l'écriture de données au format RDF. Les graphes sont représentés en un "modèle" abstrait. Ce modèle peut provenir d'une combinaison de fichiers, base de données et d'URIs. Jena peut être téléchargé à l'adresse <http://jena.apache.org/download/index.cgi> et installé en suivant les instructions à l'adresse <http://jena.apache.org/documentation/tools/index.html>.

ARQ est un moteur de requêtes supportant le langage SPARQL RDF. Jena contient ARQ, il n'est donc pas nécessaire de l'installer.

Pour faire une requête *query.rq* sur un jeu de données *data.rdf*, ouvrir un terminal dans le dossier contenant *data.rdf* et taper :

```
/home/bea/snap/apache-jena-3.14.0/bin/arq --data "data.rdf" --query "query.rq"
```

Le format de données peut aussi être N3 (avec l'extension *.n3*), Turtle (*.ttl*) ou N-Triples (*.nt*).

Note : remplacer `"/home/bea/snap/"` avec le chemin vers le dossier où vous avez téléchargé Jena.

Pour plus d'information <https://jena.apache.org/documentation/query/>.

2 Rappel

SPARQL est un langage de requête et un protocole qui permet de rechercher, d'ajouter, de modifier ou de supprimer des données RDF disponibles à travers Internet. <https://www.w3.org/TR/sparql11-query/>

SPARQL possède quatre formes de résultat :

- SELECT : retourne un tableau de résultats ;
- CONSTRUCT : retourne un graphe RDF, basé sur un modèle dans la requête ;
- DESCRIBE : retourne un graphe RDF, basé sur ce que le processeur de requêtes est configuré pour renvoyer ;
- ASK : pose une requête booléenne.

La forme SELECT retourne directement un tableau de solutions comme un ensemble de résultats, tandis que DESCRIBE et CONSTRUCT utilisent les résultats de l'appariement pour construire des graphes RDF.

La correspondance de modèles produit un ensemble de solutions. Cet ensemble peut être modifié de diverses manières :

- Projection : ne garde que les variables sélectionnées ;
- OFFSET / LIMIT : couper le nombre de solutions (mieux utiliser avec ORDER BY) ;
- ORDER BY : trier les résultats ;
- DISTINCT : ne rendre qu'une seule ligne pour une combinaison de variable et de valeur.

Les modificateurs de solutions OFFSET / LIMIT et ORDER BY s'appliquent toujours à toutes les formes de résultats.

3 Example

Example 1 *Retrouver tous les hommes.*

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

SELECT ?homme
WHERE
{
    ?homme rdf:type humans:Man
}
```

Example 2 *Retrouver tous les couples mariés.*

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

SELECT ?spouse1 ?spouse2
WHERE
{
    ?spouse1 humans:hasSpouse ?spouse2 .
}
```

Example 3 *Est-ce que la pointure de John est supérieur à 10 ?*

```
PREFIX humans: <http://humans/#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX john: <http://humans/-instances#John>

ASK
WHERE
{
    ?john humans:shoesize ?shoesize
    FILTER ( xsd:integer(?shoesize) >= 10 )
}
```

4 Exercice

Considérer l'ontologie "schema.rdfs" et l'ensemble d'annotations RDF qui utilise cette ontologie "document.rdf" créées la semaine dernière.

Exercice 1 *Retrouver tous les femmes.*

Exercice 2 *Retrouver les hommes et leurs épouses.*

Exercice 3 *Écrire une requête pour extraire tous les ressources pour lesquelles on connaît au moins un parent.*

Certains apparaissent 2 fois parce qu'on retrouve leur 2 parents ; comment éviter les doublons ?

Exercice 4 *Retrouver les personnes avec leur âge s'il est connu.*

Exercice 5 *Identifier les adultes, c'est-à-dire les personnes dont l'âge est supérieur ou égal à 18 ans.*

Exercice 6 *Est-ce que Mark est un adulte ? L'URI de Mark est <http://humans/-instances#Mark> Utiliser la clause ASK.*

Exercice 7 *Regardez les professeurs (lecturer) et interrogez leur type. Comment peuvent-ils avoir plusieurs types ?*

Exercice 8 *Retrouver toutes les instances qui sont à la fois Male et Person.*

Exercice 9 *Retrouver toutes les instances de Lecturers ou Researchers.*

Exercice 10 *Utiliser la clause describe pour obtenir la description de Laura. (<http://humans/-instances#Laura>)*

Exercice 11 *Construisez tous les triplets infèrent les instances de type Man en utilisant les hommes connus et les personnes mâles connues.*

Exercice 12 *Retrouvez toutes les personnes dont le nom contient "ar", à l'aide de la fonction `regex(String, PATTERN)`.*