

Données Semi-Structurées:

SPARQL

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

02 Mars 2020

1 Avant de commencer

Jena is an open source Semantic Web framework for Java. It provides an API to extract data from and write to RDF graphs. The graphs are represented as an abstract "model". A model can be sourced with data from files, databases, URIs or a combination of these. A Model can also be queried through SPARQL and updated through SPARUL. It can be downloaded at <http://jena.apache.org/download/index.cgi> and installed following the instructions at <http://jena.apache.org/documentation/tools/index.html>.

ARQ is a query engine for Jena that supports the SPARQL RDF Query language. It comes with Jena so there is no need of installation.

In order to apply a query *query.rq* on the data *data.rdf*, open a terminal in the folder that contains the files and type :

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

The data format can also be N3 (with extension *.n3*), Turtle (*.ttl*) or N-Triples (*.nt*)

Note : replace `"/home/bea/snap/"` with the path of the folder you downloaded Jena to.

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.

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 Exercice

Considérer l'ontologie "example.rdfs" et l'ensemble d'annotations RDF qui utilise cette ontologie "example.rdf".

Exercice 1 *Retrouve tous les hommes.*

Exercice 2 *Retrouve tous les couples mariés.*

Exercice 3 *Retrouver les hommes et leurs épouses.*

Exercice 4 *Ecrire 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 5 *Retrouver les personnes avec leur âge s'il est connu.*

Exercice 6 *Identifiez les adultes, c'est-à-dire les personnes dont l'âge est supérieur ou égal à 18 ans.*

Exercice 7 *Est-ce que Mark est un adulte ? L'URI de Mark est `http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark`*

Exercice 8 *Regardez les professeurs (lecturer) et interrogez leur type. Comment peuvent-ils avoir plusieurs types ?*

Exercice 9 *Retrouvez toutes les instances qui sont à la fois Male et Person.*

Exercice 10 *Retrouver toutes les instances de Lecturers ou Researchers.*

Exercice 11 *Utilisez la clause describe pour obtenir la description de Laura.*

Exercice 12 *Construisez tous les triplets infèrent les instances de type Man en utilisant les hommes connus et les personnes mâles connues.*

Exercice 13 *Retrouvez toutes les personnes dont le nom contient "ar", à l'aide de la fonction `regex(String, PATTERN)`.*