

Web sémantique, SPARQL

Ressources logicielles nécessaires à ce TD :

Le moteur Corese: <http://wimmics.inria.fr/doc/tutorial/corese-3.2.2.jar>

Lancez l'interface standalone du moteur de recherche Corese:

sous Windows, double-cliquez sur le fichier corese-3.2.2.jar

ou bien : `java.exe "-Dfile.encoding=UTF-8" -jar corese-3.2.2.jar`

sous Linux, exécutez la commande `"java - jar corese-3.2.2.jar"`

Cette interface présente à l'utilisateur un onglet permettant de charger des fichiers d'entrée et de visualiser les traces de chargement et permet d'ouvrir des onglets supplémentaires pour y écrire et lancer une requête SPARQL.

A l'URL <http://wimmics.inria.fr/doc/tutorial/human.ttl> se trouve un fichier RDF.

Copiez le et chargez le dans Corese.

Saisissez la requête suivante :

```
prefix h: <http://www.inria.fr/human#>
select * where {
  ?x a ?t
  filter (strstarts(?t, h:))
}
```

En une phrase traduisez ce que demande cette requête.

Exécutez cette requête. Combien de réponses obtenez-vous?

Retrouvez John et son type.

En une phrase traduisez ce que demande la requête SPARQL suivante:

```
prefix h: <http://www.inria.fr/human#>
select * where { ?x h:hasSpouse ?y }
```

Combien de réponses obtenez-vous?

Regardez dans le fichier RDF quelle est la propriété utilisée pour donner la pointure des chaussures des personnes.

1. En déduire une requête pour extraire toutes les personnes (h:Person) avec leur pointure.
2. Modifiez cette requête pour extraire toutes les personnes et, si elle est disponible, leur pointure.
3. Modifiez cette requête pour extraire toutes les personnes dont la pointure est supérieure à 8 ou dont la taille de chemise est supérieure à 12.

Dans la réponse précédente, repérez l'URI de John.

1. Formulez une requête pour trouver toutes les propriétés de John

2. Demandez une description de John au moteur en utilisant la clause SPARQL prévue pour cela.

Regardez dans le fichier RDF quelle est la propriété utilisée pour donner les enfants d'une personne.

1. Formulez une requête pour trouver les personnes qui ont au moins un enfant.

Combien de réponses obtenez-vous? Combien de doublons identifiez-vous dans ces réponses?

2. Donnez un moyen d'éviter les doublons. Combien de réponses obtenez-vous alors?

3. Reformulez une requête pour trouver les Hommes (Man) qui n'ont pas d'enfant.

Trouvez les personnes ayant la même taille de chemise.

Construisez le symétrique de toutes les relations hasFriend en utilisant le bon énoncé SPARQL.

Modifiez une des requêtes SELECT WHERE précédemment écrites en une requête CONSTRUCT WHERE (en conservant la même clause WHERE) afin de visualiser les résultats qui vous intéressent sous forme de graphe RDF.

Ajoutez vos propres annotations au fichier RDF en utilisant des propriétés vues dans ce fichier et en ajoutant de nouvelles propriétés.

Trouver les personnes qui ont même taille de chemise (shirtsize) et insérer une relation rdfs:seeAlso entre ces ressources.

Retrouver les ressources reliées par rdfs:seeAlso.

Ecrire une requête qui trouve les personnes qui ont même taille de chaussure (shoesize) et qui insère une relation rdfs:seeAlso entre ces ressources.

Retrouver les ressources reliées par rdfs:seeAlso, avec un construct qui reproduit l'arc rdfs:seeAlso

Modifier la requête pour trouver les ressources reliées par un chemin constitué de un ou plusieurs rdfs:seeAlso.

Trouver les personnes qui ont la plus grande taille de chaussure (subquery + aggregate)

Calculer la taille moyenne de chaussure

Trouver les couples sans enfant

En utilisant *insert*, créer une nouvelle personne avec ses propriétés. Vérifier ensuite qu'elle a bien été créée.

Trouver les personnes reliées par des chemins de liens de parenté.

Interroger le service <http://fr.dbpedia.org/sparql> sur le modèle :

```
select * where {  
  service <http://fr.dbpedia.org/sparql> {  
    select * where {  
      ?x rdfs:label "Auguste"@fr . ?x ?p ?y  
    } limit 100 }}
```