



---

## TD/TP n° 2 (partie 1)

### RDF/RDF Schema

#### Master Ingénierie Informatique

#### U.E. « Web sémantique (3.0) »

F. BERTRAND

Année universitaire 2020-2021

---

**Concept abordé :** Le langage de requêtes SPARQL (recherche de motifs dans un graphe)

### Environnement de travail

Pour ce TP nous utiliserons CORESE un interpréteur de requêtes SPARQL développé par l'INRIA de Sophia-Antipolis. Cet interpréteur est téléchargeable sous la forme d'un fichier exécutable .jar.

Son lancement s'effectue avec la commande suivante :

```
java -jar nomDuFichierCorese.jar
```

Son utilisation est simple avec un onglet dédié au chargement de fichier et un autre pour l'interrogation SPARQL.

Un tutoriel SPARQL est également fourni ici.

## 1 Le langage SPARQL

À partir du fichier univ1.rdfs présent dans l'archive du TP (contenant un schéma et des données), du cours, d'une présentation de SPARQL et de la recommandation W3, écrire les requêtes SPARQL permettant d'obtenir les informations suivantes :

1. Déterminer les personnes qui sont enseignantes. Le résultat attendu est le suivant :

```
<results>
<result>
<binding name='x'><uri>http://www.univ-larochelle.fr#FredMartin</uri></binding>
</result>
<result>
<binding name='x'><uri>http://www.univ-larochelle.fr#HerveBlanchon</uri></binding>
</result>
<result>
<binding name='x'><uri>http://www.univ-larochelle.fr#RobertDupond</uri></binding>
</result>
</results>
```

Est-ce que CORESE supporte l'inférence?... Justifiez votre réponse.

Maintenant aller dans le menu Engine, décochez RDFS puis ré-exécutez la requête... Conclusion?...

2. Déterminer les personnes employées par l'université de La Rochelle et leur statut. Le résultat attendu est le suivant :

```

<results>
<result>
<binding name='x'><uri>http://www.univ-larochelle.fr#FredMartin</uri></binding>
<binding name='type'><uri>http://example.org#Person</uri></binding>
</result>
<result>
<binding name='x'><uri>http://www.univ-larochelle.fr#FredMartin</uri></binding>
<binding name='type'><uri>http://example.org#Phd-Student</uri></binding>
</result>
<result>
<binding name='x'><uri>http://www.univ-larochelle.fr#HerveBlanchon</uri></binding>
<binding name='type'><uri>http://example.org#Person</uri></binding>
</result>
<result>
<binding name='x'><uri>http://www.univ-larochelle.fr#HerveBlanchon</uri></binding>
<binding name='type'><uri>http://example.org#Teacher</uri></binding>
</result>
<result>
<binding name='x'><uri>http://www.univ-larochelle.fr#RobertDupond</uri></binding>
<binding name='type'><uri>http://example.org#Person</uri></binding>
</result>
<result>
<binding name='x'><uri>http://www.univ-larochelle.fr#RobertDupond</uri></binding>
<binding name='type'><uri>http://example.org#Teacher</uri></binding>
</result>
</results>

```

3. Par rapport à la requête précédente, on souhaiterait récupérer uniquement les résultats suivants (ne pas récupérer les réponses liées au type Person) :

```

<results>
<result>
<binding name='x'><uri>http://www.univ-larochelle.fr#FredMartin</uri></binding>
<binding name='type'><uri>http://example.org#Phd-Student</uri></binding>
</result>
<result>
<binding name='x'><uri>http://www.univ-larochelle.fr#HerveBlanchon</uri></binding>
<binding name='type'><uri>http://example.org#Teacher</uri></binding>
</result>
<result>
<binding name='x'><uri>http://www.univ-larochelle.fr#RobertDupond</uri></binding>
<binding name='type'><uri>http://example.org#Teacher</uri></binding>
</result>
</results>

```

4. Déterminer les personnes étant en relation avec l'université. Le résultat attendu est le suivant :

```

<results>
<result>
<binding name='x'><uri>http://www.univ-larochelle.fr#RobertDupond</uri></binding>
<binding name='type'><uri>http://example.org#Teacher</uri></binding>
</result>
<result>
<binding name='x'><uri>http://www.univ-larochelle.fr#HerveBlanchon</uri></binding>
<binding name='type'><uri>http://example.org#Teacher</uri></binding>
</result>
<result>
<binding name='x'><uri>http://www.univ-larochelle.fr#HenriDurand</uri></binding>
<binding name='type'><uri>http://example.org#Student</uri></binding>
</result>
<result>
<binding name='x'><uri>http://www.univ-larochelle.fr#FredMartin</uri></binding>
<binding name='type'><uri>http://example.org#Phd-Student</uri></binding>
</result>
</results>

```

5. Déterminer les personnes ayant comme responsable Robert Dupond. Le résultat attendu est le suivant :

```

<results>
<result>
<binding name='x'><uri>http://www.univ-larochelle.fr#FredMartin</uri></binding>
<binding name='x'><uri>http://www.univ-larochelle.fr#HerveBlanchon</uri></binding>
</results>

```

6. Construire le graphe RDF suivant à partir d'une requête SPARQL :

```
<?xml version="1.0" ?>
<rdf:RDF
  xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
  xmlns:ex='http://example.org#'
  xmlns:ulr='http://www.univ-larochelle.fr#'>

  <rdf:Description rdf:about='http://www.univ-larochelle.fr#HerveBlanchon'>
    <ex:is-managed-by rdf:resource='http://www.univ-larochelle.fr#RobertDupond' />
  </rdf:Description>

  <rdf:Description rdf:about='http://www.univ-larochelle.fr#FredMartin'>
    <ex:is-managed-by rdf:resource='http://www.univ-larochelle.fr#RobertDupond' />
  </rdf:Description>

</rdf:RDF>
```

7. À partir du schéma univ2.rdfs (utilisant cette fois-ci des nœuds anonymes) déterminer le nom des personnes qui sont enseignantes. Si elles possèdent un responsable, le faire apparaître. Le résultat attendu est le suivant :

```
<results>
<result>
<binding name='teacher_name'><literal>Herve Blanchon</literal></binding>
<binding name='manager_name'><literal>Robert Dupond</literal></binding>
</result>
<result>
<binding name='teacher_name'><literal>Fred Martin</literal></binding>
<binding name='manager_name'><literal>Robert Dupond</literal></binding>
</result>
<result>
<binding name='teacher_name'><literal>Robert Dupond</literal></binding>
</result>
</results>
```

8. Déterminer le nom des personnes qui sont inscrites ou employées à l'université de La Rochelle. En modifiant votre requête, distinguer les noms dans chaque catégorie. Le premier résultat attendu est le suivant :

```
<results>
<result>
<binding name='name'><literal>Herve Blanchon</literal></binding>
</result>
<result>
<binding name='name'><literal>Fred Martin</literal></binding>
</result>
<result>
<binding name='name'><literal>Robert Dupond</literal></binding>
</result>
<result>
<binding name='name'><literal>Henri Durand</literal></binding>
</result>
</results>
```

Le second résultat attendu est le suivant :

```
<results>
<result>
<binding name='employee_name'><literal>Herve Blanchon</literal></binding>
</result>
<result>
<binding name='employee_name'><literal>Fred Martin</literal></binding>
</result>
<result>
<binding name='employee_name'><literal>Robert Dupond</literal></binding>
</result>
<result>
<binding name='student_name'><literal>Fred Martin</literal></binding>
</result>
<result>
<binding name='student_name'><literal>Henri Durand</literal></binding>
</result>
</results>
</sparql>
```

## 2 Utilisation d'une base RDF de taille importante avec SPARQL

La plate-forme Jamendo offre différents services de diffusion de musique. Une grande partie de son catalogue est disponible sous forme de triplets RDF (1100000 triplets) sur le site dbtune.org et s'appuie sur plusieurs schémas dont « *Music Ontology* ».

À partir des fichiers `jamendo.rdf` (données) et `musicontology-level1.rdfs` (schéma) présents dans l'archive du TP écrire les requêtes SPARQL permettant d'obtenir les informations suivantes en vous limitant à 20 résultats :

Avant de débiter les requêtes, regarder le schéma RDFS, il est relativement important mais il est structuré : d'abord sont présentées les classes puis les relations entre ces classes.

1. donner le nom des artistes musicaux;

```
Cicada
Hace Soul
vincent j
NoU
Margin of Safety
Bobywan
Les Clip's
Carter Hotel
La Tumba
King Dubby
vavrek
Suerte
My Name Is Fantastik
KEPAY
t r y ^ d
Buzzworkers
Mr Nuts
Stian
isotrak
433 er0s
```

2. pour chaque artiste donner, par ordre alphabétique sur le nom de l'artiste, le nom de ses albums;

```
! M U H ? | HUM
# NUTSHELL # | # DRZWI #
#2 Orchestra | Mercutio's Dead
#2 Orchestra | Being Alive Killed The Best In Me
#Blockout | Get High
#Dance 75# | #Dance 75# Volume 1
#Dance 75# | Dakadium
#Dance 75# | #Dance 75 Mixtape#
#Dance 75# | Can You Do Better ?
#NarNaoud# | Green Vision
#ZedMeta# | Petit aperçu
#Zorglups# | First Demo
$ArnoDj13$ | $ArnoDj13$
&ND | Hit the road
(((Niko))) | Geometry Of Art
(own+line) | Down
* Q u i r y * | FÃ©lin Pour L'Autre
* Q u i r y * | Soeurs Siamoises
-;~Â°Â§[ k.ROCKSHIRE ](Â§Â°~- | the doggystyle EP
--Kwada-- | DEMO
```

3. pour l'artiste 2MaTao, donner le nom de ses albums;

```
2MaTao | Synthesizer Project
2MaTao | Synthesizer Project vol.2
```

4. en étudiant la spécification SPARQL 1.1, déterminer comment donner, pour chaque artiste, le nombre d'albums qu'il a enregistré;

```
! M U H ? | 1
# NUTSHELL # | 1
#2 Orchestra | 2
#Blockout | 1
#Dance 75# | 4
```

```
#NarNaoud# | 1
#ZedMeta# | 1
#Zorglups# | 1
$ArnoDj13$ | 1
&ND | 1
(((Niko))) | 1
(own+line) | 1
* Q u i r y * | 2
-;~Â°Â§[ k.ROCKSHIRE ](Â§Â°~- | 1
--Kwada=- | 1
-DEMO- | 1
-mystery- | 3
...ChArLy's... | 1
...anabase* | 1
...with sad adieus | 1
```

5. donner le nom des albums « *taggés* » acoustique mais pas electro;

```
" il me manque..."
.:le canzonì di A+ebu)_AYris:..
15 court-mÃ©trages
1Ã©res Prises
2006
3 Obras de Teatro
5 Songs
A Jour
A MIS HERMANOS
A glimpse inside the bubble
A l'aube
A night in Istanbul
A tranparent pain
Accords et A,mes
Acoustic Demo
After Infinity 1
After Infinity III
Amour critique
Andrew, Just
AprÃ©s la chute
```

6. donner le nom de toutes les propriétés associées à la classe MusicArtist

```
biography
compiled
discography
djmixed
fanpage
remixed
sampled
supporting_musician
compiler
djmixed_by
remixer
sampler
supporting_musician
tribute_to
```