



UNIVERSITÉ  
DE MONTPELLIER

---

**Rapport de - HMIN234 Web Sémantique -**  
*TP2 - Web Sémantique et Social : Interrogation de  
données RDF*

---

22008346 ADOLPHE Benjamin

22014998 LAAROUSSI Laila

M1 DECOL

UNIVERSITÉ DE MONTPELLIER - FACULTÉ DES SCIENCES

25 février 2021

# 1 Le graphe des Participants à l'UE Web Sémantique et Social :

Voir le fichier Exo1.ttl.

## 2 SPARQL : Interrogation et méta-interrogation :

Modèle : Voir dans le dossier SPARQL src/main/java/Movies.java

1. Séparer les triplets contenant des connaissances ontologiques des triplets représentant des données.

```
@prefix movies: <http://www.lirmm.fr/ulliana/movies#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix dbp: <http://dbpedia.org/> .
```

```
movies:directedBy rdfs:domain movies:Movie .
movies:playsIn rdfs:domain movies:Actor .
movies:playsIn rdfs:range movies:Movie .
movies:Actor rdfs:subClassOf movies:Artist .
movies:Director rdfs:subClassOf movies:Artist .
movies:Director rdfs:subClassOf movies:Artist .
movies:title rdf:type owl:DatatypeProperty .
movies:title rdfs:domain movies:Movie .
```

```
//Données
movies:m2 movies:title "Vertigo" .
movies:m1 rdf:type movies:Movie .
movies:m3 movies:directedBy dbp:Alfred_Hitchcock .
movies:a1 movies:playsIn movies:m4 .
```

2. Donner la requête SPARQL qui sélectionne tous les (identifiants des) films.

```
String q2_1 = "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> "
              + "PREFIX mov: <http://www.lirmm.fr/ulliana/movies#> "
              + "SELECT ?s "
              + "WHERE {?s rdf:type mov:Movie}";
```

3. Donner la requête SPARQL qui sélectionne toutes les sous-classes de la classe artiste.

```
String q2_2 ="PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> "
+ "PREFIX mov: <http://www.lirmm.fr/ulliana/movies#> "
+ "SELECT ?movies "
+ "WHERE{{ ?movies rdf:type mov:Movie . }}"
+ " UNION { ?movies mov:title ?x . }"
+ " UNION { ?movies mov:directedBy ?x . }"
+ " UNION { ?x mov:playsIn ?movies . }}";
```

4. Donner la requête SPARQL qui sélectionne tous les acteurs

```
String q2_3 = "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> "
+ "PREFIX mov: <http://www.lirmm.fr/ulliana/movies#> "
+ "SELECT ?s "
+ "WHERE {?s rdfs:subClassOf mov:Artist}";
```

5. Pour chaque requête, dire s'il s'agit d'une interrogation ou d'une méta-interrogation
  - Q\_1 : interrogation
  - Q\_2 : interrogation
  - Q\_3 : méta-interrogation

### 3 L'exploration d'un endpoint SPARQL : le cas DB-Pedia :

1. Choisissez une classes de l'ontologie qui commence par la première lettre de votre nom, et donnez la liste de ses sous-classes.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
SELECT ?subclass
WHERE {
?subclass rdfs:subClassOf ?varClass .
FILTER regex(?varClass, "http://dbpedia.org/ontology/SportsLeague").
}
LIMIT 100
```

<b>subclass</b>
<a href="http://dbpedia.org/ontology/BoxingLeague">http://dbpedia.org/ontology/BoxingLeague</a>
<a href="http://dbpedia.org/ontology/FieldHockeyLeague">http://dbpedia.org/ontology/FieldHockeyLeague</a>
<a href="http://dbpedia.org/ontology/VolleyballLeague">http://dbpedia.org/ontology/VolleyballLeague</a>
<a href="http://dbpedia.org/ontology/VideogamesLeague">http://dbpedia.org/ontology/VideogamesLeague</a>
<a href="http://dbpedia.org/ontology/MixedMartialArtsLeague">http://dbpedia.org/ontology/MixedMartialArtsLeague</a>
<a href="http://dbpedia.org/ontology/CurlingLeague">http://dbpedia.org/ontology/CurlingLeague</a>
<a href="http://dbpedia.org/ontology/BasketballLeague">http://dbpedia.org/ontology/BasketballLeague</a>
<a href="http://dbpedia.org/ontology/HandballLeague">http://dbpedia.org/ontology/HandballLeague</a>
<a href="http://dbpedia.org/ontology/IceHockeyLeague">http://dbpedia.org/ontology/IceHockeyLeague</a>
<a href="http://dbpedia.org/ontology/SoftballLeague">http://dbpedia.org/ontology/SoftballLeague</a>

## 4 Partie 3