

## Série TD 3

---

### Web Sémantique. Interroger avec SPARQL.

---

#### Exercice 1

Soit le graphe RDF ci-dessous, extrait de *DBpedia* :

```
@prefix dbr:      <http://fr.dbpedia.org/resource/> .
@prefix dbo:      <http://dbpedia.org/ontology/> .
@prefix prop-fr:  <http://fr.dbpedia.org/property/> .
@prefix foaf:     <http://xmlns.com/foaf/0.1/> .

dbr:Iron_Maiden   rdf:type          <http://schema.org/MusicGroup> ;
                  dbo:bandMember    dbr:Bruce_Dickinson ;
                  dbo:genre          dbr:Heavy_Metal ;
                  dbo:recordLabel    dbr:EMI_Group ;
                  dbo:wikiPageID     "1494"^^xsd:integer ;
                  rdfs:label         "Iron Maiden"@fr ;
                  foaf:homepage      <http://www.ironmaiden.com/> .

dbr:Seventh_Son_of_a_Seventh_Son  dbo:artist dbr:Iron_Maiden ;
                  rdfs:label         "Seventh Son of a Senventh Son"@fr ;
                  prop-fr:charte     "album"@fr ;
                  dbo:releaseDate    "1988-04-23"^^xsd:date ;
                  dbo:wikiPageID     "16944"^^xsd:integer ;
                  dbo:producer       dbr:Martin_Birch ;
                  dbo:recorderIn     dbr:Munich .

dbr:Bruce_Dickinson  dbo:artist      dbr:Iron_Maiden ;
                  prop-fr:charte     "artiste"@fr ;
                  dbo:birthName      "Paul Bruce Dickinson"@fr ;
                  dbo:instrument     dbr:Chant ;
                  dbo:wikiPageID     "15967"^^xsd:integer ;
                  dbo:profession     dbr:Pilote_d'avion .

dbr:El_Dorado        dbo:artist      dbr:Iron_Maiden ;
                  prop-fr:charte     "single"@fr ;
                  rdfs:label         "El Dorado" ;
                  dbo:album          dbr:The_Final_Frontier ;
                  dbo:composer       dbr:Steve_Harris .
```

Exprimer les requêtes suivantes en SPARQL :

1. Les albums (URI et nom) sortis le *05 octobre 1988*.
2. Les ressources de type groupe musical ayant pour label discographique la ressource *Universal Music Group*.

3. Les artistes qui exercent à la fois comme *Musicien* et *Acteur*.
4. Le nom des albums en langue française produits par *Bruce Dickinson* , ainsi que les artistes membre d'Iron Maiden.
5. 5 albums produits par le même producteur de l'album nommé : *Brave New World*.
6. Le nom et la date de sortie de 5 albums produits par le même producteur de l'album nommé en langue fr : *Brave New World (Album d'Iron Maiden)*.
7. Les albums enregistrés à Munich après 1987. Utiliser la fonction prédéfinie *year*(var).
8. Les chansons/singles, distinctes, ordonnées, comportant le mot *Peace* dans leur titre. Utiliser la fonction prédéfinie *regex*(var, exp).
9. Les albums *d'Iron Maiden* et le nombre de chansons/singles composées par *Steve Harris* de chacun de ces albums.
10. Existe-t-il une chanson nommée "*Disco Maghreb*" ?
11. L'ID des tous les artistes, sauf ceux qui jouent de l'instrument guitare.

**NB :** Il est possible d'exécuter et de tester les requêtes en allant sur le SPARQL Endpoint de DBpedia : <http://fr.dbpedia.org/sparql>.

## Exercice 2

Soient deux fichiers Turtle *fileexo1.ttl* et *newfile.ttl*, représentant respectivement le graphe RDF de l'exercice 1 et le graphe RDF ci-dessous qui décrit des données sur deux autres groupes :

```
# filename: newfile.ttl
-----
@prefix dbr: <http://fr.dbpedia.org/resource/> .
@prefix dbo: <http://dbpedia.org/ontology/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix dc: <http://purl.org/dc/terms/> .

dbr:Freeklane    rdf:type          <http://schema.org/MusicGroup> ;
                  dbo:bandMember   dbr:Chems_Eddine_Abbacha ;
                  dbo:genre         dbr:Fusion ;
                  dbo:recordLabel   dbr:Padidou ;
                  rdfs:label        "Freeklane"@fr ;
                  foaf:homepage     <http://www.freeklane.net/> .

dbr:El_Dey      rdf:type          <http://schema.org/MusicGroup> ;
                  dbo:bandMember   dbr:Sami_Boukhechba ;
                  dbo:genre         dbr:Flamengnawa ;
                  dbo:recordLabel   dbr:Padidou ;
                  rdfs:label        "El Dey"@fr .

<fileexo1.ttl>   dc:date           "2017-11-20"^^xsd:date .
<newfile.ttl>    dc:date           "2018-11-20"^^xsd:date .
```

### a - Exprimer les requêtes suivantes en SPARQL :

1. Liste de tous les groupes musicaux (URI et nom) – depuis les deux fichiers.
2. Liste ordonnée des noms de tous les groupes musicaux ainsi que leur page personnelle si elle existe – depuis les deux fichiers.
3. Les URIs des groupes nommés : *Iron Maiden*, *El Dey*, *Jarka*, et *Freeklane* – depuis les deux fichiers.
4. Le type du littéral que retourne la propriété *dbo:wikiPageID*. Utiliser la fonction *datatype(lit-var)*. Depuis le *fileexo1* uniquement.

### b - Quel sera le résultat des requêtes SPARQL suivantes :

```
# filename: req1.sparql
-----
PREFIX dbr: <http://fr.dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>

SELECT ?res ?single
FROM <fileexo1.ttl>
FROM NAMED <newfile.ttl>
WHERE {
  { ?single dbo:composer dbr:Steve_Harris . }
  UNION
  {GRAPH <newfile.ttl> {?res rdf:type <http://schema.org/MusicGroup> }}
}
```

```
# filename: req2.sparql
```

```
-----  
PREFIX dbr: <http://fr.dbpedia.org/resource/>  
PREFIX dbo: <http://dbpedia.org/ontology/>
```

```
SELECT ?res ?graph  
FROM NAMED <fileexo1.ttl>  
FROM NAMED <newfile.ttl>  
WHERE {  
    GRAPH ?graph {?res rdf:type <http://schema.org/MusicGroup> }  
}
```

```
# filename: req3.sparql
```

```
-----  
PREFIX dc: <http://purl.org/dc/terms/> .
```

```
SELECT ?res ?file  
FROM <newfile.ttl>  
FROM NAMED <newfile.ttl>  
FROM NAMED <fileexo1.ttl>  
WHERE {  
    ?file dc:date "2017-11-20"^^xsd:date .  
    {GRAPH <fileexo1.ttl> {?res rdf:type <http://schema.org/MusicGroup> }}  
}
```

**NB:** Il est possible d'exécuter et de tester les fichiers/requêtes en utilisant le moteur de requêtes SPARQL ARQ de Apache Jena : <https://jena.apache.org/>

### Exercice 3

Soit le schéma RDFS suivant, extrait de *The Linked Movie Database (LinkedMDB)*:

- Voir Page 3.

Exprimer les requêtes suivantes en SPARQL :

1. La liste des films réalisés par *Joseph Losey* (URI = <http://data.linkedmdb.org/resource/director/8710>) ainsi que les films sortis en 1979.
2. La liste des films (URI et titre) ayant plus d'un réalisateur (director).
3. La liste distincte des noms des réalisateurs ayant déjà travaillés avec le compositeur de musique ayant pour nom *Hans Zimmer*.
4. 4 films et leur titre respectifs qui ont pour genre nommé *Superhero*.
5. La liste des acteurs (URI et nom) qui ont joué dans plus de 30 films.
6. La liste ordonnée de toutes les classes du schéma RDFS.
7. Existe-t-il un film nommé *Fi Rassi Rond Point* ?
8. La liste de tous les films de guerre (URI et titre) réalisés par *Denis Villeneuve* sauf ceux sortis en 2006.

**NB :** Il est possible d'exécuter et de tester les requêtes en allant sur le SPARQL Endpoint de The Linked Movie Database (LinkedMDB) : <http://data.linkedmdb.org/snorql/>.

### Remarque :

La majorité des ressources de la base de données liées *LinkedMDB* ont aussi leurs descriptions sur DBpedia.

Par exemple, le film *Star Wars : The Empire Strikes Back* est décrit dans les deux bases via les deux URIs suivantes : [http://dbpedia.org/page/The\\_Empire\\_Strikes\\_Back](http://dbpedia.org/page/The_Empire_Strikes_Back) et <http://data.linkedmdb.org/page/film/76> .

A ce titre, la propriété *owl:sameAs* est utilisée pour mentionner qu'une URI décrit la même ressource que dans une autre URI.

```

@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>
@prefix dc: <http://purl.org/dc/terms/> .
@prefix movie: <http://data.linkedmdb.org/resource/movie/> .

movie:film      a      rdfs:Class, rdf:Resource ;
                  Rdfs:label      "film".
movie:actor     a      rdfs:Class, rdf:Resource ;
                  rdfs:label      "actor".
movie:director  a      rdfs:Class, rdf:Resource ;
                  rdfs:label      "director".
movie:music_contributor  a rdfs:Class, rdf:Resource ;
                  rdfs:label      "music_contributor".
movie:film_genre      a      rdfs:Class, rdf:Resource ;
                  rdfs:label      "film_genre".

movie:actor      a      rdf:Property, rdf:Resource ;
                  rdfs:label      "actor" ;
                  rdfs:domain      movie:film ;
                  rdfs:range      movie:actor .

movie:director   a      rdf:Property, rdf:Resource ;
                  rdfs:label      "director" ;
                  rdfs:domain      movie:film ;
                  rdfs:range      movie:director .

movie:genre      a      rdf:Property, rdf:Resource ;
                  rdfs:label      "genre" ;
                  rdfs:domain      movie:film ;
                  rdfs:range      movie:film_genre .

movie:music_contributor  a      rdf:Property, rdf:Resource ;
                  rdfs:label      "music_contributor" ;
                  rdfs:domain      movie:film ;
                  rdfs:range      movie:music_contributor .

dc:title         a      rdf:Property, rdf:Resource ;
                  rdfs:label      "title" ;
                  rdfs:domain      movie:film ;
                  rdfs:range      rdfs:Literal .

dc:date          a      rdf:Property, rdf:Resource ;
                  rdfs:label      "date" ;
                  rdfs:domain      movie:film ;
                  rdfs:range      xsd:date .

xsd:date         a      rdfs:Datatype .

movie:actor_name a      rdf:Property, rdf:Resource ;
                  rdfs:label      "actor_name" ;
                  rdfs:domain      movie:actor ;
                  rdfs:range      rdfs:Literal .

movie:music_contributor_name  a      rdf:Property, rdf:Resource ;
                  rdfs:label      "music_contributor_name" ;
                  rdfs:domain      movie:music_contributor ;
                  rdfs:range      rdfs:Literal .

```