



MASTER 1 INFORMATIQUE - WEB SÉMANTIQUE

TP3 - Système OBDA avec Obi-Wan

Hugo Maitre
Adrien Linares

Enseignant :
Pr. Marie-Laure MUGNIER
Créateur du système :
M. Maxime BURON

27 avril 2022

Table des matières

| | | |
|-----|--|---|
| 1 | Description de l'ontologie | 2 |
| 2 | Description des mappings | 3 |
| 3 | Listes des requêtes effectués et nombres de réponses | 4 |
| 4 | Conclusion | 5 |
| 4.1 | Principales difficultés | 5 |
| 4.2 | Perspectives d'amélioration | 5 |
| 4.3 | Bilan Appréciation | 5 |

1 Description de l'ontologie

Nous avons choisi d'orienter la construction de notre ontologie sur 2 principaux concepts. Dans un premier temps la caractérisation des films : leur titre, année de sortie, genre et mots clés en définissant des prédicats binaires associant à un film les divers attributs cités plus haut. On a donc articulé tous les prédicats binaires autour de la classe Movie. Aucune autre classe n'a dû être implémentée hormis la classe Movie pour ce premier concept car n'apportant pas grand chose à l'ontologie. Le second concept a pour objectif de décrire les personnes impliquées dans un film, leur rôle ; il peut s'agir ici de 3 classes différentes : les acteurs, les producteurs et le réalisateur. On a décidé d'introduire la classe Personne (classe mère dont le nom vient assez naturellement) qui a pour but d'abstraire les 3 classes citées précédemment, on pourra de cette manière plus facilement considérer la totalité des personnes impliquées dans un film particulier. Cette ontologie s'articule donc principalement sur les propriétés d'un film donné, aux personnes reliées à celui-ci et leurs rôles respectifs.

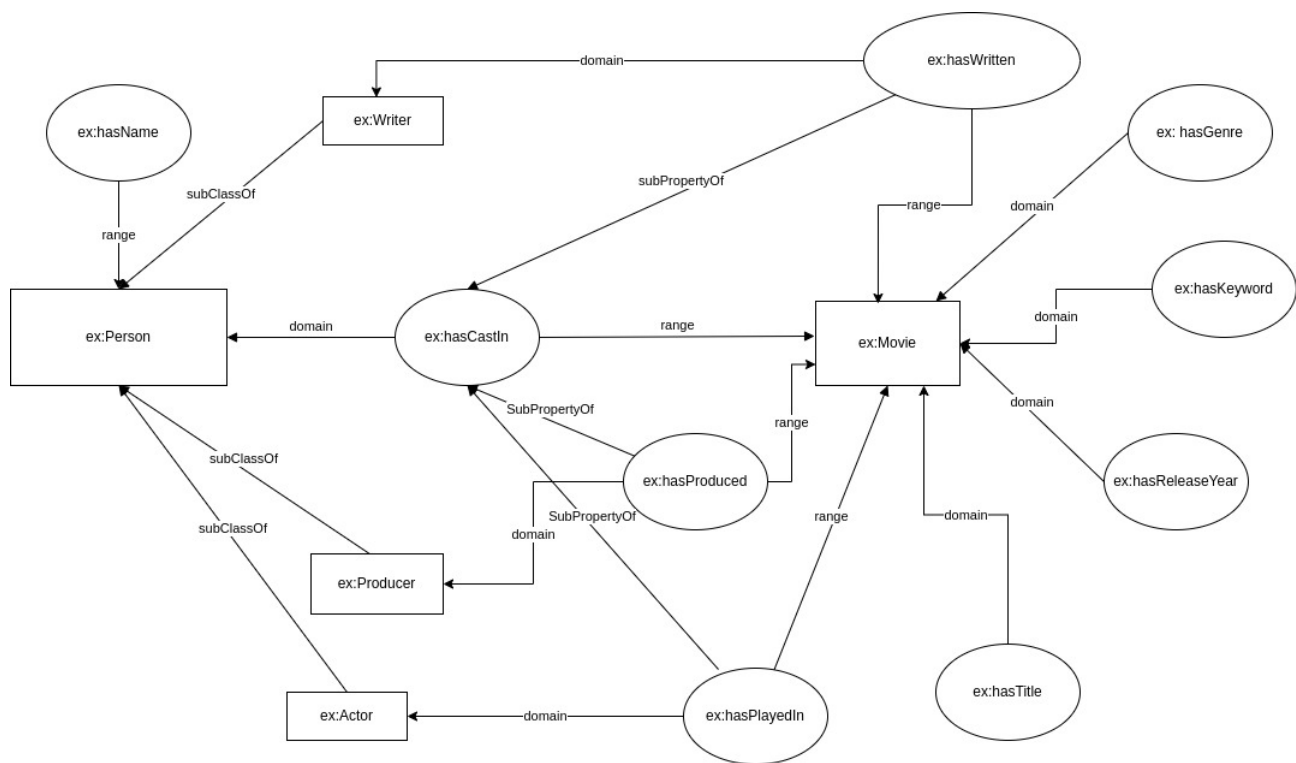


FIGURE 1 – Ontologie

Ci-dessus est représenté le graphe RDF de notre ontologie. Il nous a servi de base pour ensuite construire cette dernière dans le format N-triples.

Listes des relations présentes dans l'ontologie :

1. hasGenre(Movie,Genre)
2. hasKeyWord(Movie,Keyword)
3. hasReleaseYear(Movie,Year)
4. hasTitle(Movie,Title)
5. hasPlayedIn(Actor,Movie)
6. hasProduced(Producer,Movie)
7. hasWritten(Writer,Movie)
8. hasCastIn(Person,Movie)
9. hasName(Person,Name)

2 Description des mappings

Les mappings seront ici nommés tel que dans le fichier *ris.json* par souci de clarté. Ils seront de plus, dans ce rapport, triés par ordre de difficulté croissante.

- **Mapping n°1 "searchGenreFilmByTitle"** : Donne le genre d'un titre de film à partir de son titre.

Requête associée :

```
SELECT T.title, MI.info from movie_info AS MI, info_type AS IT,  
title AS T WHERE MI.info_type_id = IT.id and MI.info_type_id=3 and  
MI.movie_id = T.id
```

Triplet produit : Film <ex :hasGenre> Genre

- **Mapping n°2 "searchFilmByKeyword"** : À partir d'un mot-clé, donne le(s) film(s) associé(s).

Requête associée :

```
SELECT T.title, K.keyword from title as T, movie_keyword as MV,  
keyword as K WHERE T.id = MV.movie_id and MV.keyword_id = K.id
```

Triplet produit : Film <ex :hasKeyword> Keyword

- **Mapping n°3 "searchActorsByFilm"** : À partir d'un titre de film, donne les acteurs y ayant joué.

Requête associée :

```
SELECT T.title, N.name from cast_info AS CAA, title AS T, name AS N
WHERE CAA.movie_id = T.id and CAA.person_id = N.id and CAA.role_id < 3
```

Triplet produit : Acteur <ex :hasPlayedIn> Film

- **Mapping n°4 "searchCastingByFilmTitle"** : À partir d'un titre de film, donne les gens qui y ont participé.

Requête associée :

```
SELECT T.title, N.name from cast_info AS CAA, title AS T, role_type AS RT, name AS N
WHERE movie_id = T.id and CAA.role_id < 5 and CAA.person_id = N.id and CAA.role_id = RT.id
```

Triplet produit : Person <ex :hasCastIn> Film

- **Mapping n°5 "searchFilmsMadeByProducerAgivenYear"** : Donne les films produits par un producteur à telle année.

Requête associée :

```
SELECT T.title, T.production_year, N.name from title as T, cast_info as CAA, name as N, role_type as RT
WHERE N.id = CAA.person_id and CAA.role_id = 4 and CAA.role_id = RT.id and CAA.movie_id = T.id
```

Triplets produits :

Producer <ex :hasProduced> Film

Film <ex :hasReleaseYear> Year

Les triplets produits sont testables à partir du SPARQL endpoint et renverront les résultats escomptés.

3 Listes des requêtes effectués et nombres de réponses

1. **Quel est le genre d'un film à partir de son titre ?**

ex : Quel est le genre du film "Into the Wild" => 3 réponses : Adventure, Biography, Drama

2. **Quels sont les films d'un certain mot clé ?**
ex : Quels sont les films du mot clé "red-star" ? => "Wonder Woman" 1 réponse
3. **Quels sont les acteurs ayant joué dans un film donné ?**
ex : Quels sont les acteurs ayant joué dans "Memento" ? => 11 réponses
4. **Quel est le casting complet d'un film ?**
ex : Quel est le casting complet (seulement réalisateur + scénariste/écrivain + acteurs) du film "Barry Lyndon" ? => 44 réponses
5. **Quels sont les films avec l'année de sortie écrit par un réalisateur ?**
ex : Quels sont les films avec année de production écrit par Oscar Wilde
=> 8 réponses
6. **Quels sont les producteurs ayant collaboré avec un certain réalisateur ?**
=> Nous n'avons pas réussi à exprimer cette requête plus complexe dans un format compréhensible par Obi Wan (sans faire une requête imbriquée).

4 Conclusion

4.1 Principales difficultés

Il a été difficile d'avoir une vision des relations entre chacune des tables sur la base de données IMDB où l'affichage des attributs et clés étrangères avec SQLite3 n'est pas toujours explicite. Une fois que nous avons une vision précise de la BDD, l'écriture de nos requêtes en SQL – d'une manière compréhensible pour Obi-Wan – nécessitant souvent 2 à 3 jointures s'est aussi révélée relativement difficile, en partie dû à notre manque de pratique du SQL.

4.2 Perspectives d'amélioration

On pourrait compléter notre ontologie avec d'autres concepts et relations, produire encore plus de triplets par requête et faire des requêtes relativement plus complexes comme la numéro 6.

4.3 Bilan Appréciation

Nous avons apprécié travailler sur ce projet qui a été un complément au cours et nous a permis de bien mieux comprendre le lien entre la vision bases de données classiques et RDFS ontologie de l'autre. Le contexte était d'autant plus intéressant puisqu'il portait sur le domaine du cinéma.