

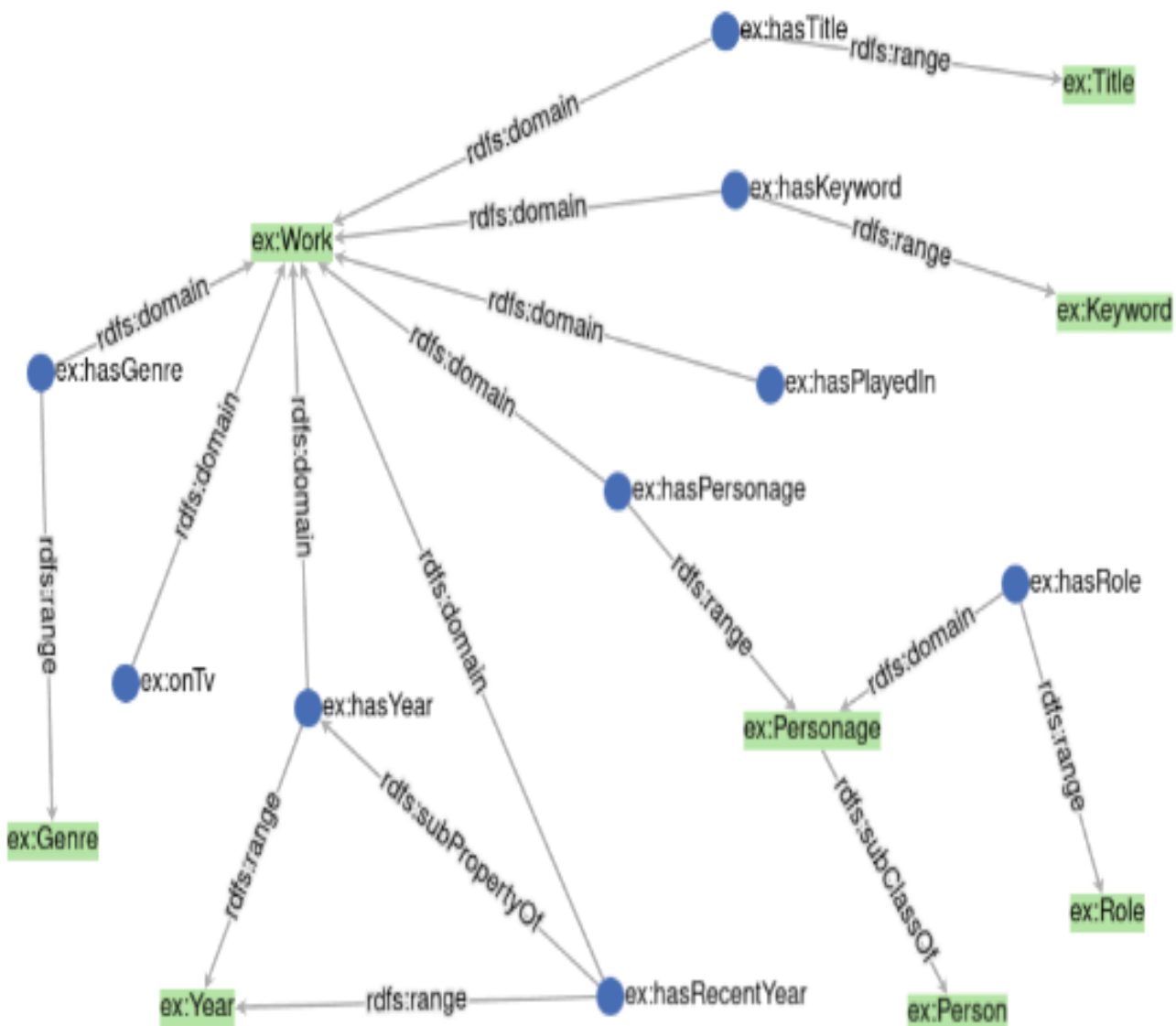
TP: Système OBDA avec Obi-Wan

1) Description de l'ontologie

Dans le cadre de ce projet, nous avons choisi comme thématique la caractérisation des films sous différentes propriétés avec l'ensemble des personnes y participant.

L'ontologie peut décrire, sous forme de prédicat binaire, les films avec leur titre, leur année, leur genre, les rôles des personnes dans un film, mais également peut afficher toutes les informations d'un film juste avec un mot clé (le keyword associé au film). Avec cette ontologie, nous pouvons aussi déterminer les œuvres récentes c'est-à-dire dont la date de production est supérieure à l'an 2000. Par ailleurs, on peut même connaître les films avec différents genres.

2) Visualisation de l'ontologie



3) Description des mappings

1. “work_info” :

- rôle: décrit pour chaque film le titre et la date de production.
- requête SQL associée: “select id, title, production_year from title”
- requête SPARQL: ?work <ex:hasTitle> ?title. ?work <ex:hasYear> ?year.

http://localhost:7001/sparql

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3 SELECT * WHERE {
4   ?work <ex:hasTitle> ?title.
5   ?work <ex:hasYear> ?year.
6 }
```

Table Response 17514 results in 1.136 seconds Simple view Ellipse Filter query results

	work	title	year
1	<ex:work/55434>	Becket	1964
2	<ex:work/71650>	Boy A	2007
3	<ex:work/87765>	Chaos Theory	2007
4	<ex:work/55913>	Before the Rain	1994
5	<ex:work/1920>	13 Moons	2002
6	<ex:work/23418>	Ai qing wan sui	1994
7	<ex:work/93453>	Cimarron	1931

2. “getFilm_withKeyword” :

- rôle: donne le titre du film à partir d’un mot clé.
- requête SQL associée: “select T.id, T.title, K.keyword from title as T, keyword as K, movie_keyword as MK where K.phonetic_code=T.phonetic_code and MK.keyword_id=K.id”
- requête SPARQL: ?work <ex:hasTitle> ?title. ?work <ex:hasKeyword> ‘keyword’.

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3 SELECT * WHERE {
4   ?work <ex:hasTitle> ?title.
5   ?work <ex:hasKeyword> 'wizard'.
6 }
```

Table Response 2 results in 0.144 seconds

	work	title
1	527823	The Wizard
2	574813	Who's Your Daddy?

3. “person_role” :

- rôle: détermine le rôle de chaque participant d’un film
- requête SQL associée: “select N.name, R.role, T.title from name as N, role_type as R, cast_info as C, title as T where C.role_id=R.id and C.person_id=N.id and T.id=C.movie_id”
- requête SPARQL: ?p <ex:hasPlayedIn> ?w. ?w <ex:hasRole> ?r.

```
2 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3 SELECT * WHERE {
4   ?person <ex:hasPlayedIn> ?work.
5   ?person <ex:hasRole> ?role.
6 }
```

Table Response 823161 results in 367.974 seconds Simple view

	person	work	role
1	'K', Murray the	I Wanna Hold Your Hand	actor
2	'K', Murray the	The Beatles Anthology	actor
3	'The Jeweler', Jacob	State Property	actor
4	1, Todd	Who's the Man?	actor
5	13, Nick	Inland Empire	actor
6	2 Dope, Shaggy	Big Money Hustlas	actor

4. "film_with_given_person" :

- rôle: pour savoir dans quel(s) film(s) apparaît un personnage donné.
- requête SQL associée: "select N.name, T.title from name as N, role_type as R, cast_info as C, title as T where C.role_id=R.id and C.person_id=N.id and T.id=C.movie_id"
- requête SPARQL: 'person name' <ex:hasPlayedIn> ?work.

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3 SELECT * WHERE {
4   'Bowen, Vincent' <ex:hasPlayedIn> ?work.
5 }
```

Table Response 6 results in 0.143 seconds

	work
1	36 Quai des Orfèvres
2	Comme une image
3	Le temps qui reste
4	Munich
5	Palais royal!
6	Un long dimanche de fiançailles

5. "film_on_tv" :

- rôle: connaître les oeuvres télévisuelles (séries, téléfilm, épisode).
- requête SQL associée: "select T.title, K.kind from kind_type as K, title as T where T.kind_id=K.id and kind='episode' or kind='tv movie' or kind='tv series'"
- requête SPARQL: ?work <ex:onTv> ?genre.

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3 SELECT * WHERE {
4   ?work <ex:onTv> ?genre.
5 }
```

Table Response 33960 results in 0.436 seconds

	work	genre
33959	Üç maymun	tv series
33960	Üç maymun	tv movie
33957	Üvegtigris 2	tv series
33958	Üvegtigris 2	tv movie
33955	Üvegtigris	tv series
33956	Üvegtigris	tv movie
33953	Ödipussi	tv series
33954	Ödipussi	tv movie
33951	Öritsu uchûgun Oneamisû no tsubasa	tv series
33952	Öritsu uchûgun Oneamisû no tsubasa	tv movie

6. "film_with_more_genres" :

- rôle: trouver les film qui ont différents genres.
- requête SQL associée: "select T.title, K.kind, K2.kind from kind_type as K, title as T, kind_type as K2 where T.kind_id=K.id and K.kind!=K2.kind"
- requête SPARQL: ?work <ex:hasMoreGenre> ?genres.

```
4 ?work <ex:hasMoreGenre> ?genres.
5 }
```

Table Response 117439 results in 1.67 seconds

	work	genres
117433	Üç maymun	episode
117434	Üç maymun	movie
117435	Üç maymun	tv mini series
117436	Üç maymun	tv movie
117437	Üç maymun	tv series
117438	Üç maymun	video game
117439	Üç maymun	video movie
117426	Üvegtigris 2	episode
117427	Üvegtigris 2	movie
117428	Üvegtigris 2	tv mini series
117429	Üvegtigris 2	tv movie

7. "film_after2000" :

- rôle: trouver les œuvres produites après l'an 2000.
- requête SQL associée: "select T.title, N.name, T.production_year from name as N, role_type as R, cast_info as C, title as T where C.role_id=R.id and C.person_id=N.id and T.id=C.movie_id and T.production_year > 2000"
- requête SPARQL: **?work <ex:hasRecentYear> ?year.**

```

3 SELECT ↑ WHERE {
4   ?work <ex:hasRecentYear> ?year.
5 }

```

Table Response 6178 results in 3.082 seconds

work	year
6178 Üç maymun	2008
6177 Üvegtigris 2	2006
6176 Üvegtigris	2001
6175 Ônibus 174	2002
6174 Être et avoir	2002
6173 Éloge de l'amour	2001
6172 Çilgin dersane	2007

8. "film_genre" :

- rôle: donne le genre d'un film.
- requête SQL associée: "select T.title, K.kind from kind_type as K, title as T where T.kind_id=K.id"
- requête SPARQL: **?work <ex:hasGenre> ?genre.**

```

4   ?work <ex:hasGenre> ?genre.
5 }

```

Table Response 17068 results in 0.221 seconds

work	genre
16923 Young at Heart	movie
16922 Young and Innocent	movie
16911 Young Adam	movie
16910 You, Me and Dupree	movie
16909 You've Got Mail	movie
16908 You're a Big Boy Now	movie
16907 You'll Never Get Rich	movie
16905 You Wish!	tv movie
16904 You Were Never Lovelier	movie
16903 You Stupid Man	movie
16902 You Only Move Twice	episode

3) Liste de requêtes effectuées avec leur nombre de réponses

- a) Quel est le genre d'un film donné par son titre?
req: **"Young Adam" <ex:hasGenre> ?genre.**
→ 1 réponse : "Young Adam" => movie
- b) Quelles sont les œuvres avec plusieurs genres ?
- c) req: **?work <ex:hasMoreGenre> ?genres.**
→ 117439 réponses : ex "Ôdishon" en a 7
- d) Quel rôle joue chaque personnage du film dont le mot clé est 'wizard' ?
req: **?work <hasPersonage> ?person.**
 ?person <hasRole> ?role.
 ?work <hasKeyword> 'wizard'.
→ 107 réponse
- e) Quels mots clés sont associés au film numéro '77' ?
req: **'77' <ex:hasKeyword> ?keyword.**
→ 9 réponses : age, ak-47, ascii,,,
- f) Quels sont les films télévisuels en épisodes ?
req: **?work <ex:onTv> 'episode'.**
→ 16777 réponses : ex 'Carry on Constable'