

Étudiant 1: Yu, chen bao, 12012026

Étudiant 2: Chaid, Fatoumbi, 12006030

### **Devoir – Classification de Musiciens**

A rendre en binôme, au dernier TP , le 19 décembre 11 :55.

Avec compléments



---

## **1) Toutes les données sont représentées sous forme de triplets .**

A ce niveau on sélectionne un certain nombre de propriétés utiles et permettant de classer les artistes. Ci-dessous la liste des propriétés choisies:

- genre Ex:K-pop,rock,classique,etc.
  - instrument Ex:piano,guitare,flûte,etc.
  - langage Ex:anglais,chinois,espagnol,etc. (on ne peut pas aimer un chanteur qui chante dans une langue qu'on ne connaît pas)
  - nationalité Ex:chinois,français,espagnol,etc.
  - sexe
  - voiceType Ex:soprano,ténor,contralto,etc. ( C'est un critère très important musicien )
- 

## **2) Environ 120 musiciens .**

### **2.a) Le code pour le construc:**

```
PREFIX yu:<http://www.semanticweb.org/yu#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX wd: <http://www.wikidata.org/entity/>
```

```
CONSTRUCT {
```

```
    #Je donne les classes des chaque objet ?xx
```

```
    ?x rdf:type yu:musicien.
```

```
        ?x rdfs:label ?nom.
```

```
    ?instrument rdf:type yu:instrument.
```

```
        ?instrument rdfs:label ?instrumentLabel.
```

```
    ?sexe rdf:type yu:sexe.
```

```
        ?sexe rdfs:label ?sexeLabel.
```

```
    ?langage rdf:type yu:langage.
```

```
        ?langage rdfs:label ?langageLabel.
```

```
    ?nationalite rdf:type yu:nationalité.
```

```
        ?nationalite rdfs:label ?nationaliteLabel.
```

```
    ?voiceType rdf:type yu:voiceType.
```

```
        ?voiceType rdfs:label ?voiceTypeLabel.
```

```
    ?genre rdf:type yu:genre.
```

```
        ?genre rdfs:label ?genreLabel.
```

```
    #Je definie les différentes propriétés
```

```
    ?x
```

```
    yu:pinstrument ?instrument;
```

```
    yu:pgenre ?genre;
```

```
    yu:psexe ?sexe;
```

```

        yu:plangage ?langage;
        yu:pnationalité ?nationalite;
        yu:pvoiceType ?voiceType.
    }
WHERE {
    ?x wdt:P106 wd:Q639669. #musicien;
    ?x wdt:P1477 ?nom.filter(lang(?nom) = "en").
    ?x wdt:P1303 ?instrument.
        ?instrument rdfs:label ?instrumentLabel.filter(lang(?instrumentLabel) = "en").
    ?x wdt:P136 ?genre.
        ?genre rdfs:label ?genreLabel.filter(lang(?genreLabel) = "en").
    ?x wdt:P21 ?sexe.
        ?sexe rdfs:label ?sexeLabel.filter(lang(?sexeLabel) = "en").
    ?x wdt:P103 ?langage.
        ?langage rdfs:label ?langageLabel.filter(lang(?langageLabel) = "en").
    ?x wdt:P27 ?nationalite.
        ?nationalite rdfs:label ?nationaliteLabel.filter(lang(?nationaliteLabel) = "en").
    ?x wdt:P412 ?voiceType.
        ?voiceType rdfs:label ?voiceTypeLabel.filter(lang(?voiceTypeLabel) = "en").
}

```

## 2.b) Commentaires du code précédent :

- Dans construc:
  - ?instrument rdf:type yu:instrument.
  - ?instrument rdfs:label ?instrumentLabel.

On donne un type chaque objet dans notre ontologie ainsi qu'un label à chaque pour faciliter la lecture.

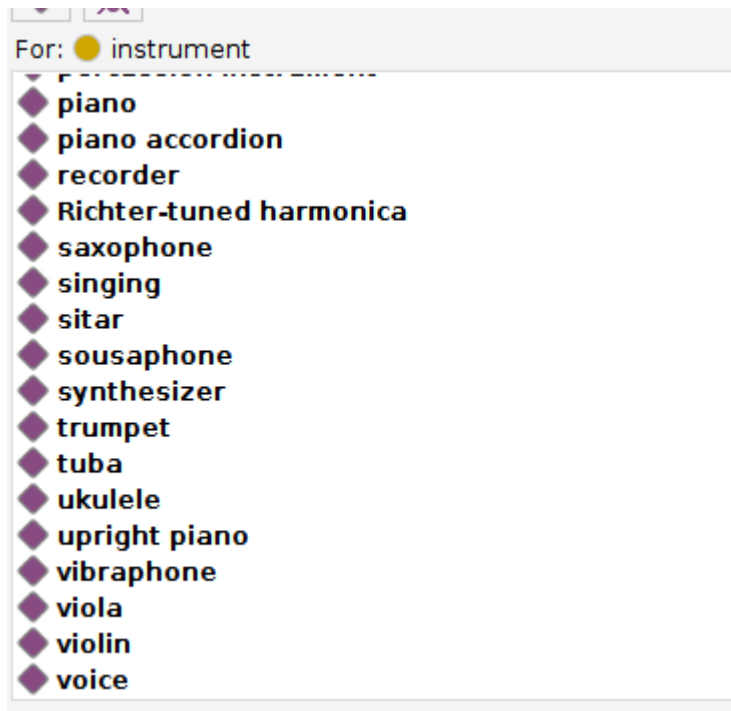
**Ex :** avoir un label au lieu d'un code de type Q673291

- Dans where:
  - ?x wdt:P1303 ?instrument.
  - ?instrument rdfs:label ?instrumentLabel.filter(lang(?instrumentLabel) = "en").

**?x** sera un **objet musicien** , qui utilise les instruments **?instrument** relié par la propriété **p1303** (INSTRUMENT).

**rdfs:label** nous permet d'obtenir un label qui correspond à chaque instrument .

## 2.c) Exemple d'affichage (dans Protégé) :



### **3) L'ontologie que nous avons réalisé sous Protégé :**

#### **3.a) les classes :**



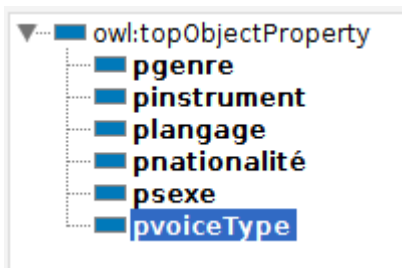
La classe critère contient toutes les sous classes permettant de classifier un musicien.

La classe musicien contient tous les musiciens.

La classe voters contient les personnes pouvant voter (Ici ce n'est pas utilisé, explication dans la question 5).

#### **3.b) Object properties :**

Le p devant tous les criteres veut dire propriété, et tous les propriétés ont domaine qui est "musicien".



### 3.c) Data properties



Ici on en a deux, commentaire\_manuelle et commentaire\_auto.  
avec "+" si c'est en opinion positive, et "-" si c'est en opinion négative.

### 4)

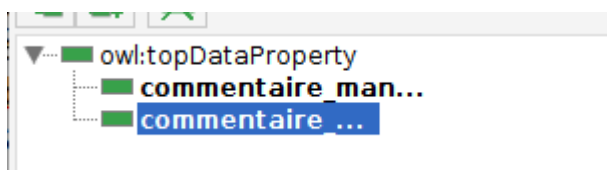
Pour réaliser cette tâche , nous avons suivi la deuxième directive donnée dans l'énoncé: .  
Les étapes sont les suivantes :

- On télécharge sous format rdf/xml depuis yasgui
- On ouvre dans visual studio
- On remplace tous les rdf:Description par owl:NamedIndividual
- On fait un copié collé dans l'ontologie , dans la partie des "individuals"
- On ajoute `xmlns:yu="http://www.semanticweb.org/yu#"`

Vous pouvez voir le résultat dans le fichier rendu.

### 5)

Ici nous avons fait un modèle pour représenter une seule personne. On a donc utilisé les **Data properties** :



avec "+" si c'est en opinion positive,  
et "-" si c'est en opinion négative .

En effet , au départ j'ai utilisé une autre modélisation , et je pense que c'est plus adapté pour la question 9 (expliqué dans la question 9)

Donc l'autre modélisation est avec 4 object properties :

- aimerPar\_auto ,Domain:musicien
- aimerPar\_manuelle,Domain:musicien
- detesterPar\_autp,Domain:musicien
- detesterPar\_manuelle,Domain:musicien

Et on peut établir les relations correspondantes pour chaque musiciens .

Et cette modélisation ce sera plus utile pour la question( on peut compter facilement le nombre des utilisateurs qui ont donné un avis positif pour un musicien. )

## 6)

A ce niveau on a rajouté manuellement "+" ou "-" pour ces 16 musiciens :

Show: ☒ this ☒ disjoints

Found 36 uses of commentaire\_manuelle

- Adam Noah Levine
- Adele Laurie Blue Adkins
- Alicia Augello Cook
- Amy Lee
- Christina Flores
- commentaire\_manuelle
- David James Archuleta
- Joanna Caroline Newsom
- John Anthony Gillis
- Michael Trent Reznor
- Nicole Prascovia Elikolani Valiente
- Norma Deloris Egstrom
- Peter Brian Gabriel
- Richard Wayne Penniman
- Roberta Joan Anderson
- Tionne Tenese Watkins
- Wesly Reid Scantlin

## 7)

L'ensemble des règles SWRL que nous avons réalisés :

	nomme	nomme
✓	S1	yu:musicien(?x) ^ yu:psexe(?x, autogen0:Q6581072) ^ yu:pgenre(?x, autogen0:Q546264) -> yu:commentaire_auto(?x, "+")
✓	S2	yu:musicien(?x) ^ yu:psexe(?x, autogen0:Q6581072) ^ yu:pgenre(?x, autogen0:Q213665) -> yu:commentaire_auto(?x, "+")
✓	S3	yu:musicien(?x) ^ yu:pinstrument(?x, autogen0:Q5994) -> yu:commentaire_auto(?x, "+")
✓	S4	yu:musicien(?x) ^ yu:pinstrument(?x, autogen0:Q6607) -> yu:commentaire_auto(?x, "+")
✓	S5	yu:musicien(?x) ^ yu:pinstrument(?x, autogen0:Q11405) -> yu:commentaire_auto(?x, "+")
✓	S6	yu:musicien(?x) ^ yu:psexe(?x, autogen0:Q6581097) ^ yu:pgenre(?x, autogen0:Q7749) -> yu:commentaire_auto(?x, "-")
✓	S7	yu:musicien(?x) ^ yu:psexe(?x, autogen0:Q6581097) ^ yu:pgenre(?x, autogen0:Q11403) -> yu:commentaire_auto(?x, "-")
✓	S8	yu:musicien(?x) ^ yu:psexe(?x, autogen0:Q6581097) ^ yu:pgenre(?x, autogen0:Q1530455) -> yu:commentaire_auto(?x, "-")
✓	S9	yu:musicien(?x) ^ yu:psexe(?x, autogen0:Q6581072) ^ yu:voiceType(?x, autogen0:Q37137) -> yu:commentaire_auto(?x, "-")

Pour "+" :

Pour choisir nos musicien préférés , on met un **commentaire\_auto** ssi:

Il s'agit d'un musicien qui :

est une femme et fait du K-pop

ou une une femme qui fait du beat\_music

ou joue du piano,

ou joue de la guitar,

ou joue de la flûte .

Pour “\_” :

Si c'est un musicien et que :

c'est un homme et dont le genre de musique est rock\_and\_roll

ou c'est un homme et dont le genre de musique est samba

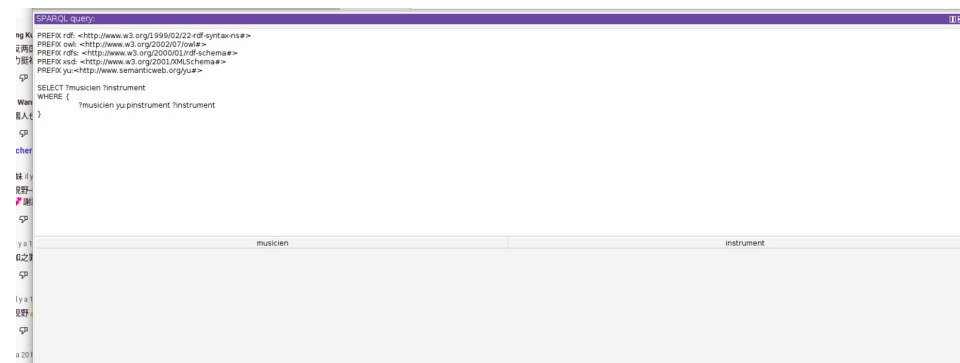
ou c'est un homme et dont le genre de musique est voice\_jazz

ou c'est une femme qui produit un son de type contralto

**8)**

A ce niveau nous avons eu du mal à réaliser les requêtes sql dans protégé. On l'a donc calculé manuellement et on obtient une accuracy de 9/16 soit environ 64% .

Les requêtes que nous avons testé et le résultat :



On n'observe aucun résultat.

**Liste des commentaires manuelle :**

Usage: commentaire\_manuelle

Show: ☒ this ☒ disjoints

Found 36 uses of commentaire\_manuelle

- Adam Noah Levine
- Adele Laurie Blue Adkins
- Alicia Augello Cook
- Amy Lee
- Christina Flores
- commentaire\_manuelle
  - DataProperty: commentaire\_manuelle
  - commentaire\_manuelle Domain musicien
- David James Archuleta
- Joanna Caroline Newsom
- John Anthony Gillis
- Michael Trent Reznor
- Nicole Prascovia Elikolani Valiente
- Norma Deloris Egstrom
- Peter Brian Gabriel
- Richard Wayne Penniman
- Roberta Joan Anderson
- Tionne Tenese Watkins
- Wesly Reid Scantlin

Calcul:

Ceux qui correspond(9)	Ceux qui correspond pas(5)
Adam +	Adele
Alicia +	Nicole
Amy Lee +	Peter br +
David +	Richard wayne
Joanna +	Roberta
John +	
Michel +	
Tionne -	
Wesley +	

**Accuracy : 9/14 ≈ 64%**



9) On veut des "règles de classifications", d'après les questions précédentes, on veut des objets propriétés comme "pgenre", "page" et "pinstrument", mais selon le nombre des gens qui aiment .

Du coup on peut ajouter des règles de classifications comme :

aimer par plus de  $n$  personnes, avec  $n$  un nombre entier .

Cette règle de classification nous permettra de sélectionner des chanteurs qui sont aimés par  $n$  personnes.