Création d'une ontologie des langues indo-européennes en Protégé

Étudiants du Groupe 3 :

Alexandra DEMKOVIC
Marceau HERNANDEZ
Didier Kouassi KOUAME

M1 Langue et Informatique

Web sémantique et Big-Data : M2SOL033

Enseignante responsable : Victoria EYHARABIDE

Année universitaire **2022-2023**

Alexandra DEMKOVIC
Marceau HERNANDEZ
Didier Kouassi KOUAME

Table des matières

Alexandra DEMKOVIC
Marceau HERNANDEZ
Didier Kouassi KOUAME

Cours Web sémantique & Big-Data
Master I Langue et Informatique
17/04/2023

Description du sujet d'ontologie

- Les Langues indo-européennes -

Ce projet de groupe porte sur la création d'une ontologie couvrant une partie des langues de la famille de langues indo-européennes qui sont évidemment parlées sur le continent européen, mais qui s'étendent également aux autres continents. Il s'agit de modéliser les langues romanes, les langues slaves, les langues germaniques et les langues celtiques à travers leurs différentes propriétés et caractéristiques communes ou non (équivalences, différences, séparations, oppositions, etc).

Le choix de ce sujet s'est fait de part notre cursus universitaire qui concerne principalement les langues et la linguistique que l'on cherche à manipuler pour les allier à l'informatique. Pour déterminer les différentes propriétés, caractéristiques et éléments des différentes familles de langues choisies, notre projet mêle connaissances personnelles des langues, recherches en ligne de travaux et bibliographies, et supports de cours pour réaliser une description des concepts et des propriétés.

Dans ce rapport sont exposés l'ensemble du projet, les choix faits, le cœur de l'ontologie, les instances peuplant l'ontologie, le processus de création du point d'accès SPARQL, les requêtes SPARQL testées et les résultats, les exemples d'utilisation potentielle en termes de Web sémantique et Big-Data.

Définition du domaine et de sa portée

Le but du projet est de modéliser une ontologie des langues indo-européennes. L'objectif est de pouvoir décrire les relations entre les diverses langues, mais aussi de pouvoir décrire les propriétés des langues. Des relations doivent être établies à plusieurs niveaux au

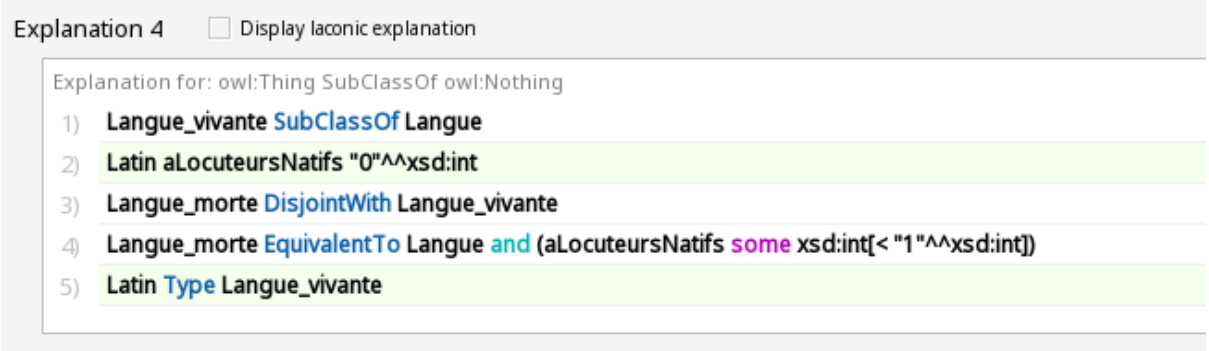
travers des relations entre ces langues (adstrats, superstrats, etc) ou encore par des spécificités propres. Comme le fait de posséder deux alphabets officiels (latin et cyrillique) pour le Serbe.

Voici certaines de ces relations et ce qu'elles impliquent :

- Si deux langues ont une même langue mère, elles sont donc sœurs ;
- Les langues ont un nombre de locuteurs, ainsi on peut interroger sur les langues les plus parlées, trier par nombre de locuteurs, etc. ;
- Une langue peut être une parlée dans plusieurs pays, et peut venir d'un ou plusieurs pays. Par exemple, on peut interroger la langue selon le pays d'origine, les continents dans laquelle elle est parlé, et caetera.

Ainsi, un utilisateur pourrait chercher selon des propriétés ou des ensembles de propriétés, comme lorsqu'on manipule une base de données, mais aussi se servir des inférences pour pouvoir générer des requêtes plus intuitives (par exemple, au lieu de lister les langues sans locuteurs natifs, on peut lister les langues mortes). Ces mêmes inférences nous permettent également d'assurer la cohérence de notre ontologie.

Par exemple, imaginons que par erreur, une langue sans locuteurs natifs se retrouve classée comme langue vivante. Comme une langue sans locuteurs natifs est, dans notre ontologie, considérée comme une langue morte et qu'une langue ne peut être morte et vivante (les deux classes sont donc disjointes). Il serait incohérent d'avoir une langue vivante avec 0 locuteurs. C'est donc ce que l'on a provoqué avec le Latin, en attribuant cette instance à la classe "Langue_vivante" alors que cette dernière n'a pas de locuteurs natifs, le raisonneur détecte une inconsistance et nous le reporte. Vous trouverez ci-dessous l'explication fournie par ce dernier.



Délimitation de la portée

L'ontologie est limitée aux langues indo-européennes, et aux propriétés et relations qui en découlent. La graphie, les caractères spécifiques à la langue comme l'utilisation de diacritiques et de ligatures ne seront pas pris en compte et la modélisation n'inclut pas les dialectes des langues. Car nous avons décidé de concentrer notre approche sur les propriétés qui pourraient être communes entre les différentes langues au dépit de celles qui les différencient.

Dans quel contexte est utilisée l'ontologie

Etant donné que le but est de pouvoir interroger des données sur les langues indo-européennes et de pouvoir les comparer entre elles, il s'agira donc d'assembler les données de façon à pouvoir requêter entre autres sur les propriétés des langues, comme le nombre de locuteurs, ou le nombre de pays où la langue est officielle pour obtenir la liste des objets répondant à ces critères.
Ces requêtes peuvent être les suivantes :

- Quelles sont les langues qui ont un nombre de locuteurs supérieur à 100 millions ?
- Quelles sont les langues qui ont moins de locuteurs que le français ?
- Quelles sont les langues qui sont adstrats du français ?
- Quelles sont les langues qui sont des langues vivantes ?
- Quelles sont les langues qui sont utilisées dans plus de 5 pays ?

- Combien de langues sont de type SVO, et combien sont de type SOV ?
- Combien de locuteurs ont le français comme langue maternelle ?
- Quelles langues sont ancêtres du français ?
- Quelles sont les propriétés communes entre le français et l'allemand ?

Etat de l'art

En ce qui concerne la création d'une ontologie linguistique ou de langues plus précisément, des projets ont été menés pour modéliser les différentes propriétés linguistiques des langues, telles que la grammaire, la sémantique et la phonologie, ou encore pour représenter les relations entre les langues, telles que les liens de parenté et les emprunts linguistiques ; cette dernière partie concerne particulièrement le travail que nous allons effectuer.

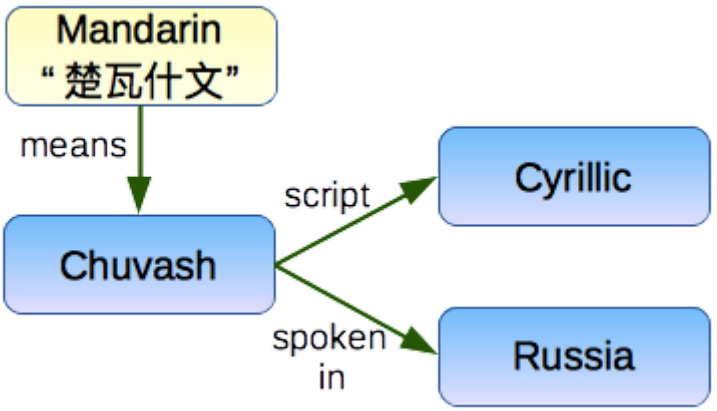
Ces ontologies peuvent être pratiques pour l'indexation de documents multilingues, la recherche d'informations dans plusieurs langues, ou encore la conception de systèmes de traduction automatique.

Parmi ces travaux, le projet GOLD (General Ontology for Linguistic Description) propose une ontologie qui décrit les caractéristiques structurelles des langues du monde entier. L'article "The General Ontology for Linguistic Description (GOLD) : A Case Study of its Use in the Documentation of Wambaya" décrit l'utilisation de GOLD dans un contexte de documentation linguistique.

L'Ontologie générale pour la description linguistique (GOLD) a été imaginée pour la première fois par Scott Farrar en 2003, afin de résoudre le problème de la résolution de schémas de balisage disparates pour les données linguistiques, c'est-à-dire l'utilisation de terme, de formats ou de conventions de balisage différents pour décrire ou étiqueter des données similaires, en particulier celles des langues en danger. Au départ, l'EMELD avait pour objectif de construire un ensemble de termes unique, néanmoins, du fait de la diversité de termes utilisés par les linguistes et les sous-communautés en linguistique, cela n'a pas pu se faire. Ainsi, une ontologie était une solution plus adaptée pour lier ces différents ensembles de termes. Will Lewis a créé la première version de l'ontologie en organisant les informations du glossaire de termes linguistiques en ligne de SIL International, puis une équipe de recherche de l'Université de l'Arizona a doublé la taille de l'ensemble de termes en fouillant la littérature linguistique. Terry Langendoen est également à l'origine de l'acronyme GOLD et a contribué à une grande partie de son contenu intellectuel. Des améliorations majeures ont ensuite été apportées par Gary Simons, Anthony Aristar, Brian Fitzsimons et d'autres personnes impliquées dans le projet E-MELD. En 2005, un atelier sponsorisé par E-MELD a permis de critiquer le contenu et la structure de GOLD, et des révisions sont actuellement en cours sur la base des suggestions des participants. Le groupe de morphologie de Surrey participe également à une série de visites d'échange pour améliorer la couverture du contenu

de GOLD.

De même, le projet Lexvo propose une ontologie pour les langues qui permet de représenter les relations entre les langues et les concepts associés, tels que les pays, les peuples et les cultures. "Lexvo" est un mot-valise créé à partir de deux termes: "Lex" pour lexique (ensemble des mots d'une langue) et "vo" pour vocabulaire. Ainsi, "Lexvo" fait référence au vocabulaire et au lexique des langues humaines. Ce projet est une plateforme qui se spécialise autour des informations relatives aux langues, aux mots, aux caractères et à d'autres éléments liés au langage humain sur le Web de données liées et le Web sémantique. L'objectif de cette initiative mondiale est de créer un Web de données qui met en lumière les relations entre les différentes entités de notre monde. Ce travail permet d'une certaine façon de souligner les liens qui existent entre les éléments linguistiques dans notre monde, tels que les relations sémantiques entre les termes multilingues tels que "livre" ou "New York", peut-on voir sur le site Web du projet. La plateforme crée des identifiants uniques (URI) pour les objets liés au langage, qui sont facilement accessibles et fortement interconnectés avec des ressources sur le Web.
Ci-dessous un exemple d'aperçu de leur projet consultable sur leur site :



Source Image : Site Web du projet Lexvo (http://www.lexvo.org/)

Quant à l'ontologie de Bloom, celle-ci s'inspire de WordNet pour réaliser, sous Protégé, une représentation des mots de la langue française, en se basant sur les verbes. Benjamin Bloom a créé en 1956 une taxonomie, appelée la taxonomie de Bloom, qui permet de classer hiérarchiquement les niveaux d'acquisition des connaissances. Cette méthode est largement utilisée dans l'enseignement par objectifs.

La taxonomie de Bloom, qui a été révisée par Lorin Anderson en 1991, est une grille d'analyse composée de six grandes catégories d'habiletés cognitives, allant de la plus simple à la plus complexe. Chaque catégorie contient une liste de verbes d'action, appelés "habiletés ciblées".

Au final cette ontologie, comme celles précédemment citées, sont assez éloignées de notre ontologie. En effet, nous n'avons pu trouver d'ontologie représentant les langues

comme nous voulions le faire. A l'exception, bien sûr, d'ontologies plus générales comme Dbpedia.

La plupart des ontologies que nous avons trouvé essaient de représenter les langues au travers de leurs formes, de leurs mots lorsque notre idée est de les représenter d'un point de vue plus général, avec des valeurs qui les rendraient plus facilement comparables entre elles.

Construction d'une ontologie

Pour construire notre ontologie des langues indo-européennes, nous avons cherché à créer à partir des codes python étudiés en cours de Web sémantique et Big-Data où il s'agit de créer des classes, des propriétés, des instances de classes et leurs relations. Puis, nous avons eu l'idée de rassembler les différentes données à entrer dans des fichiers .csv qui sont lus par les codes python et ainsi est générée l'ontologie.

Dans le but de réaliser notre ontologie, nous avons d'abord commencé à essayer de se représenter ce qu'on voudrait qu'une ontologie sur le sujet puisse faire, nous l'avons ensuite représentée sur papier avant de passer à Protégé.
Dans Protégé, nous avons alors défini les classes et propriétés (objet et données). Il nous a alors fallu imaginer les différents liens entre les propriétés, définir les domaines et les cibles, les restrictions sur ces propriétés, les équivalences des classes. Le but était de donner un maximum de matière au raisonneur.

Dans un premier temps, nous avions commencé à écrire un code en python pour créer les classes, les instances et des relations, comme ci-dessous :

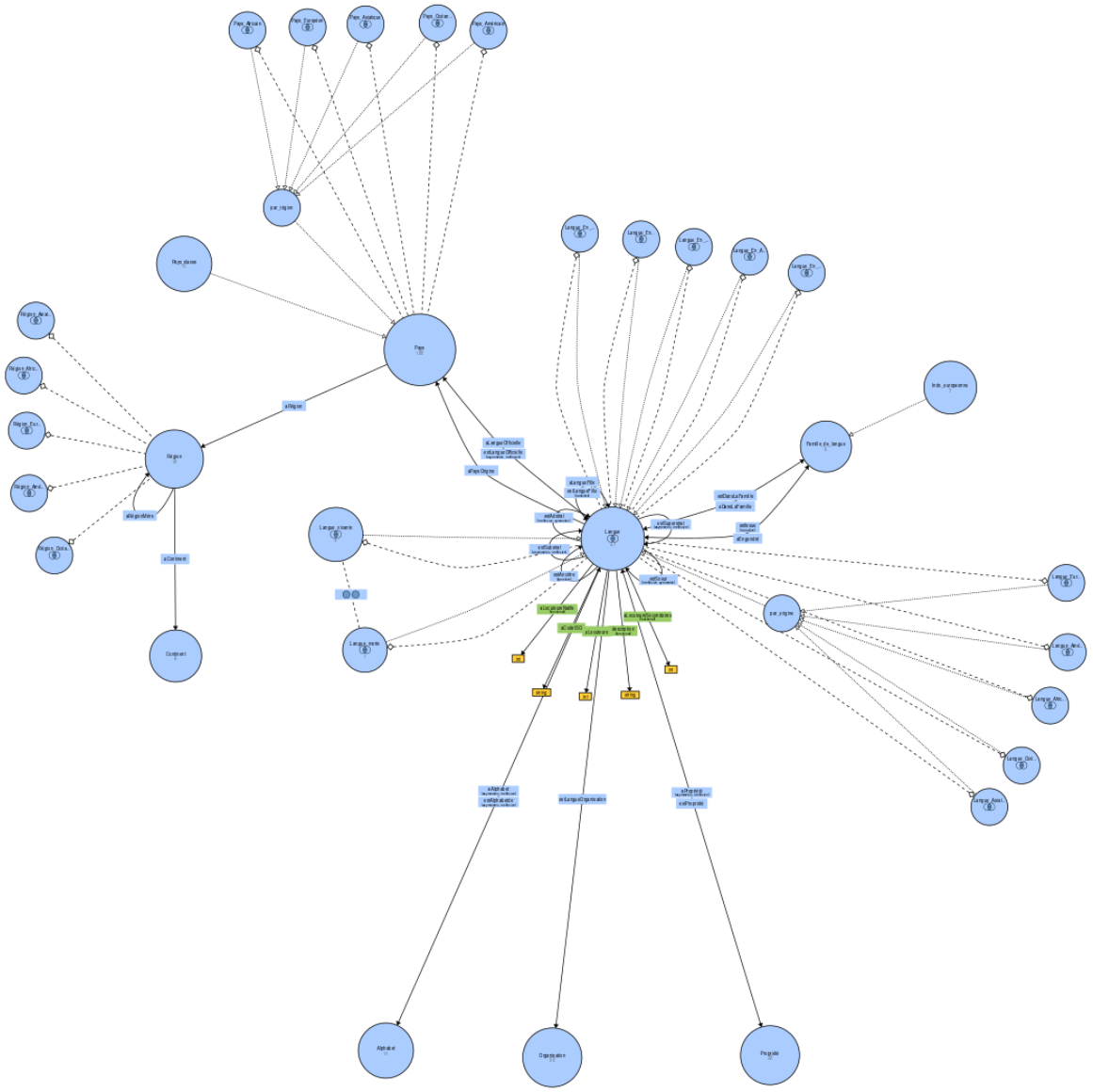
```
graph = Graph()
graph.bind("languesSlaves", languesSlaves)
LanguesSlaves = createClass("LanguesSlaves",None,None,None,None)
pays = createClass("Pays",None,None,None,None)
langue = createClass("Langue",None,None,None,None)
alphabet = createClass("Alphabet",None,None,None,None)

estDansLaFamille = createObjectRelation("estDansLaFamille","Langue","LanguesSlaves")
est_parlé_en = createObjectRelation("est_parlé_en","Langue","Pays")

#Alphabets:
latin = addAlphabet("Latin")
cyrillique = addAlphabet("Cyrillique")


#Langues slaves
Serbe = addLangue("Serbe")
Croate = addLangue("Croate")
Russe = addLangue("Russe")
Ukrainien = addLangue("Ukrainien")
Biélorusse = addLangue("Biélorusse")
Polonais = addLangue("Polonais")
Tchèque = addLangue("Tchèque")
Slovaque = addLangue("Slovaque")
Slovène = addLangue("Slovène")
Macédonien = addLangue("Macédonien")
Bulgare = addLangue("Bulgare")

#Pays slaves
Serbie = addPays("Serbie")
```

```
#relation langues-pays
#relation langues-famille de langue
addLanguePaysRelation(Serbe,Serbie)
addLangueFamilleRelation(Serbe,LanguesSlaves)
```

Puis, nous avons eu l'idée d'automatiser le peuplage des différents éléments en instanciant l'ontologie à l'aide de tableaux (voir les fichiers CSV dans le dossier), correspondant à divers types d'instance, que nous remplirons afin de produire relations et instances. Après avoir commencé à instancier les langues, nous avons alors ajouté différentes propriétés et classes, notamment le code ISO de la langue.

```
    if continent:
        continent_uri = create_instance(graph, continent, classes["Continent"])
        graph.add((region_uri, properties["a_continent"], continent_uri))

    if region_mere:
        region_mere_uri = create_instance(graph, region_mere, classes["Région"])
        graph.add((region_uri, properties["a_region_mere"], region_mere_uri))

Marceau-h
def parse_lang_csv(path):
    df = pd.read_csv(path).fillna("")
    for index, row in df.iterrows():
        label = row["Langue"]
        pays = parse_multiple_values(row, "Pays")
        alphabet = parse_multiple_values(row, "Alphabet")
        locuteurs_natifs = parse_int(row["Locuteurs_n"])
        locuteurs_secondaires = parse_int(row["Locuteurs_s"])
        famille = parse_multiple_values(row, "Famille")
        proprietes = parse_multiple_values(row, "Propriété")
        organisations = parse_multiple_values(row, "Organisations")
        code_iso = parse_multiple_values(row, "ISO_639-3")
        pays_origine = parse_multiple_values(row, "Pays_origine")
        langue_mere = parse_multiple_values(row, "Langue_origine")

        langue_uri = create_instance(graph, label, classes["Langue"])
        instances["langues"][label] = langue_uri

        for a in alphabet:
            a_uri = create_instance(graph, a, classes["Alphabet"])
            graph.add((langue_uri, properties["a_alphabet"], a_uri))

        for f in famille:
            if f == "Créole":
                f_uri = create_instance(graph, f, classes["Famille_de_langue"])
            else:
                f_uri = create_instance(graph, f, classes["Indo_europeenne"])
```

Voici ci-dessous une représentation des classes, de la hiérarchie entre ces dernières, des propriétés d'objet et des propriétés de données :



L'image étant trop grande pour être lisible, nous vous invitons à la consulter au sein du dépot github du projet à l'adresse suivante :
https://raw.githubusercontent.com/Marceau-h/ontologie-s8/main/onto.svg

Les langues ont ainsi plusieurs propriétés d'objet entre elles, mais également un grand nombre de propriétés vers d'autres classes, comme les pays, les "propriétés" et caetera.
Cela nous permettra alors d'interroger les langues selon leur relation avec une autre langue, mais également selon leur relation avec d'autres classes. Par exemple on pourrait se

demander : “Quelles sont les langues sœurs de l’Italien” ou “Quelles langues (dans notre ontologie) sont parlées en Amérique du nord”.

Création des classes et des attributs (propriétés)

Pour peupler notre ontologie nous avons créé des classes d'ensemble permettant de regrouper les langues selon les régions dans lesquelles elles sont parlées, mais également selon la, ou dans de rares cas les, région de laquelle la langue origine. Cela nous permet alors de visualiser avec facilité, une fois avoir lancé le raisonneur, les langues en fonction de ces deux critères. Cela permet aussi de se rendre compte du fait que certaines langues indo-européennes ont été exportées dans le monde sur des régions alentours ou dans le monde.

Nous avons alors relié nos diverses classes et les propriétés d’objets pour décrire les relations entre toutes les différentes instances. Nous avons ainsi établi des relations permettant de relier les langues à leur famille de langue avec la relation “estDansLaFamille”, ou pour les relier à leur type d’alphabet avec la relation “aAlphabet”. Ou encore pour associer chaque langue à la fois aux pays dans lesquelles ces langues sont parlées par le prédicat “estLangue”, mais aussi aux organisations dans lesquelles ces langues sont officialisées par le prédicat “estLangueOrganisation”, mais aussi des relations telles que “estSoeur” pour signifier que deux langues sont sœurs, ou encore “estLangueFille” pour préciser que l’une hérite de l’autre.

Ensuite, nous avons aussi établi des relations entre les classes ou individus avec leurs propriétés de données (valeur de données) pour associer des attributs comme le nombre de locuteurs natifs avec “aLocuteursNatifs” d’une langue ou de locuteurs ayant une langue pour langue seconde avec “aLocuteursSecondaires”, ou encore associer des chaînes de caractères comme les noms d’organisations mondiales (ONU, OTAN, CEE, etc) aux différentes langues.

Création d’instances

Les classes permettent de regrouper des instances possédant des propriétés similaires ; il peut y avoir plusieurs ensembles d’individus parmi les individus qui seront regroupés par classes ou sous-classes notamment. Ces instances vont permettre de représenter concrètement les concepts abstraits définis dans l’ontologie des langues indo-européennes.

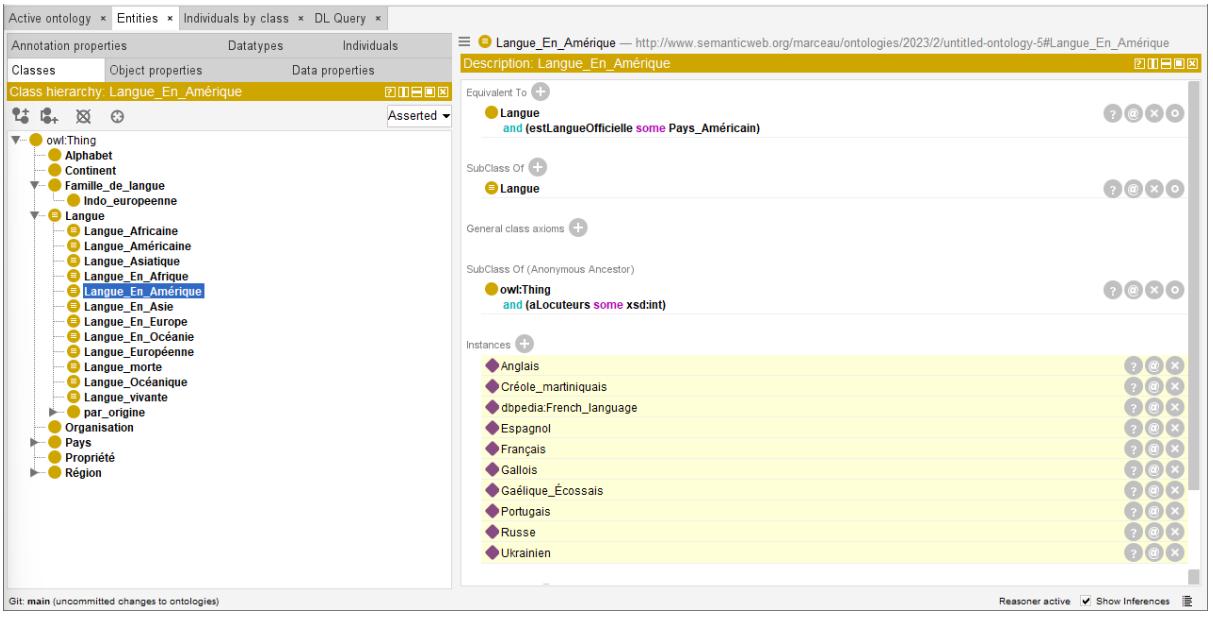
Nous avons alors en instance différentes langues indo-européennes (allemand, anglais, celte, croate, espagnol, français, russe, serbe, et caetera). Ainsi que leurs sous-familles (langues slaves, langues romanes, langues celtiques, langues germaniques). Mais également les alphabets de ces langues (latin, cyrillique, celtique, gaulois).
Nous avons, dans un second temps, ajouté les différents pays (France, Serbie, Russie, Allemagne, Espagne, …), continents (Europe, Asie, Afrique, les Amériques, …) et régions (Europe de l'Ouest, de l'Est, et centrale, Amérique du Nord et du Sud,...).

Raisonneur : Consistance et déductions

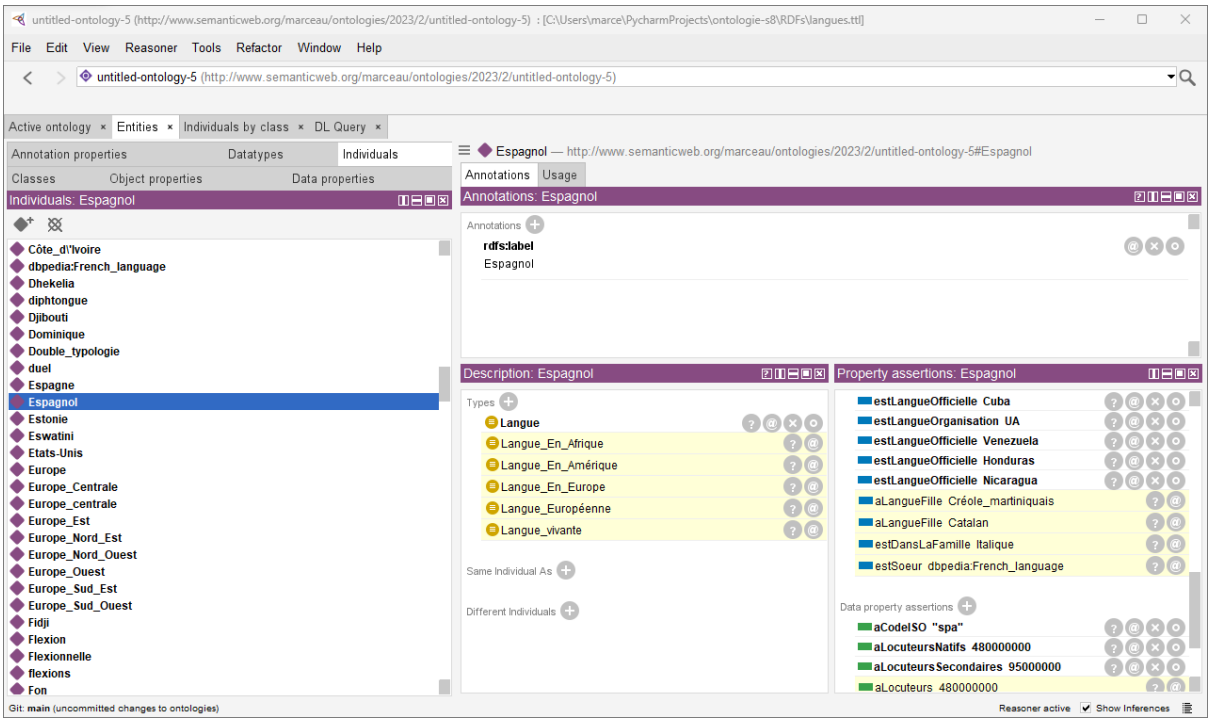
Pour vérifier la consistance de notre ontologie, mais surtout pour compléter notre ontologie à l'aide des déductions, nous avons utilisé le raisonneur HermiT.

Comme nous en avons parlé précédemment, le raisonnement nous permet de s'assurer de la consistance de notre ontologie, ce qui nous a servi à vérifier notre ontologie au fur et à mesure de notre instanciation.

Le raisonneur nous permet également d'ajouter, grâce aux déductions, un grand nombre de relations, c'est-à-dire de propriétés d'objet, mais aussi de "classement" des instances. Ce classement au sein des instances permet alors, avec les bonnes restrictions, de remplir des classes entières à l'aide de différentes propriétés, en voici un exemple dans protégé.



Cela nous permet alors également à l'aide de restrictions de trouver des propriétés d'objets comme ci-dessous :



SPARQL

Requêtes SPARQL

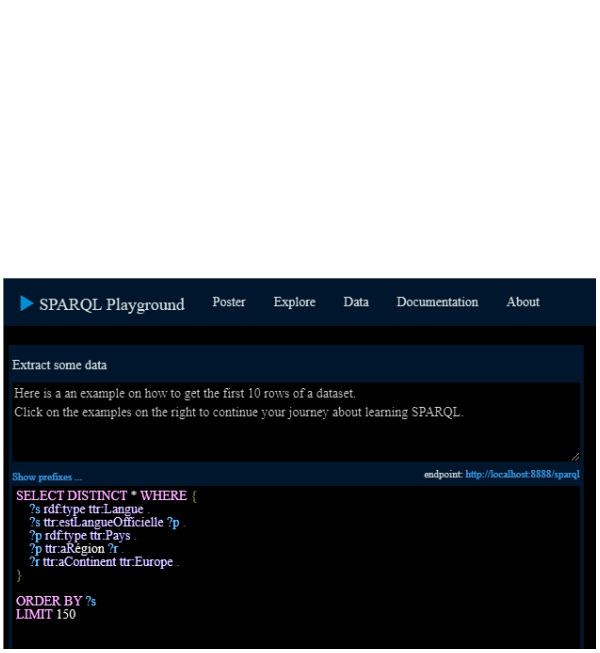
Après à la création de l'ontologie et à son instanciation, nous pouvons alors requêter cette l'ontologie à l'aide de requêtes SPARQL. Pour cela, nous avons réalisé plusieurs requêtes depuis le logiciel Protégé et un point d'accès SPARQL, certaines sont des retranscriptions de nos questions précédemment énoncées dans la partie *Dans quel contexte est utilisée l'ontologie*, alors que d'autres ont été pensées pour présenter les ajouts les plus récents à notre ontologie.
Vous trouverez ci-dessous des exemples de requêtes SPARQL entrées :

Active Ontology | Entities | Individuals by class | DL Query | SPARQL Query

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX untitled-ontology-5: <http://www.semanticweb.org/marceau/ontologies/2023/2/untitled-ontology-5#>

SELECT ?s (COUNT(?s) AS ?pays) WHERE {
   ?s rdf:type untitled-ontology-5:Langue .
   ?s untitled-ontology-5:estLangueOfficielle ?p .
   ?p rdf:type untitled-ontology-5:Pays .
}
GROUP BY ?s
HAVING (?pays > 5)
LIMIT 150
```

s	pays
Polonais	"6"^^<http://www.w3.org/2001/XMLSchema#integer>
Russe	"15"^^<http://www.w3.org/2001/XMLSchema#integer>
Français	"27"^^<http://www.w3.org/2001/XMLSchema#integer>
Bulgare	"9"^^<http://www.w3.org/2001/XMLSchema#integer>
Portugais	"47"^^<http://www.w3.org/2001/XMLSchema#integer>
Allemand	"8"^^<http://www.w3.org/2001/XMLSchema#integer>
Serbe	"9"^^<http://www.w3.org/2001/XMLSchema#integer>
Slovaque	"6"^^<http://www.w3.org/2001/XMLSchema#integer>
Anglais	"88"^^<http://www.w3.org/2001/XMLSchema#integer>
Ukrainien	"8"^^<http://www.w3.org/2001/XMLSchema#integer>
Italien	"6"^^<http://www.w3.org/2001/XMLSchema#integer>
Espagnol	"20"^^<http://www.w3.org/2001/XMLSchema#integer>



s	p	r
ttr:Allemand	ttr:Allemagne	ttr:Europe_Nord_Ouest
ttr:Allemand	ttr:Autriche	ttr:Europe_Nord_Ouest
ttr:Allemand	ttr:Belgique	ttr:Europe_Nord_Ouest
ttr:Allemand	ttr:Italie	ttr:Europe_Nord_Ouest
ttr:Allemand	ttr:Liechtenstein	ttr:Europe_Nord_Ouest
ttr:Allemand	ttr:Luxembourg	ttr:Europe_Nord_Ouest
ttr:Allemand	ttr:Suisse	ttr:Europe_Nord_Ouest
ttr:Anglais	ttr:Akrotiri	ttr:Europe_Centrale
ttr:Anglais	ttr:Dhekelia	ttr:Europe_Centrale
ttr:Anglais	ttr:Gibraltar	ttr:Europe_Sud_Ouest
ttr:Anglais	ttr:Irlande	ttr:Europe_Nord_Ouest
ttr:Anglais	ttr:Jersey	ttr:Europe_Nord_Ouest
ttr:Anglais	ttr:Malte	ttr:Europe_Sud_Ouest
ttr:Anglais	ttr:Royaume-Uni	ttr:Europe_Nord_Ouest
ttr:Anglais	ttr:Île_de_Man	ttr:Europe_Nord_Ouest
ttr:Biélorusse	ttr:Biélorussie	ttr:Europe_Nord_Est
ttr:Biélorusse	ttr:Fédération_de_Russie	ttr:Europe_Nord_Est
ttr:Biélorusse	ttr:Pologne	ttr:Europe_Centrale
ttr:Biélorusse	ttr:Ukraine	ttr:Europe_Nord_Est
ttr:Bulgare	ttr:Bulgarie	ttr:Balkans
ttr:Bulgare	ttr:Macédoine_du_Nord	ttr:Balkans
ttr:Bulgare	ttr:Serbie	ttr:Balkans
ttr:Bulgare	ttr:Ukraine	ttr:Europe_Nord_Est
ttr:Catalan	ttr:Espagne	ttr:Europe_Nord_Ouest
ttr:Catalan	ttr:France	ttr:Europe_Nord_Ouest
ttr:Catalan	ttr:Italie	ttr:Europe_Nord_Ouest
ttr:Corse	ttr:France	ttr:Europe_Nord_Ouest
ttr:Corse	ttr:Italie	ttr:Europe_Nord_Ouest

Point d'accès SPARQL

Nous avons donc pu lancer des requêtes sur protégé, mais également héberger notre propre endpoint SPARQL, disponible à l'adresse https://sparql.marceau.tech, cela nous a permis de requêter rapidement notre ontologie, sans avoir à lancer protégé. Mais également de la rendre requêtable par tous.

Un ensemble de figures des requêtes SPARQL testées est disponible en annexe, ainsi que sur le dépôt Github.

Alexandra DEMKOVIC
Marceau HERNANDEZ
Didier Kouassi KOUAME

Cours Web sémantique & Big-Data
Master I Langue et Informatique
17/04/2023

Bibliographie

Pour les chiffres (nombres de locuteurs par langue) :
UNESCO : https://fr.unesco.org/
Ethnologue : https://www.ethnologue.com/

Pour les langues parlées officiellement dans les organisations mondiales :
Site web de l'OTSC : http://odkb-csto.org/
Site web de l'UEE : https://www.eurasiancommission.org/
Site web de l'OTAN : https://www.nato.int/
Site web de l'OSCE : https://www.osce.org/
Site web de l'ONU : https://www.un.org/
Site web de l'UE : https://europa.eu/
Site web de l'OCS : http://www.shanghaicooperation.org/

Etat de l'art :
Projet GOLD : http://linguistics-ontology.org/info/about. Dernière consultation le 17 avril 2023.
"The General Ontology for Linguistic Description (GOLD) : A Case Study of its Use in the Documentation of Wambaya" (article)
Utilisation de GOLD dans un contexte de documentation linguistique : https://www.aclweb.org/anthology/W02-0305.pdf
Projet Lexvo **:** http://www.lexvo.org/. Dernière consultation le 17 avril 2023.
Projet BLOOM : Deslis, A. J. (n.d.). Conception de l'ontologie. ILOT. Consulté le 10 avril 2023, from https://ilot.hypotheses.org/category/ontologie/conception-de-lontologie.

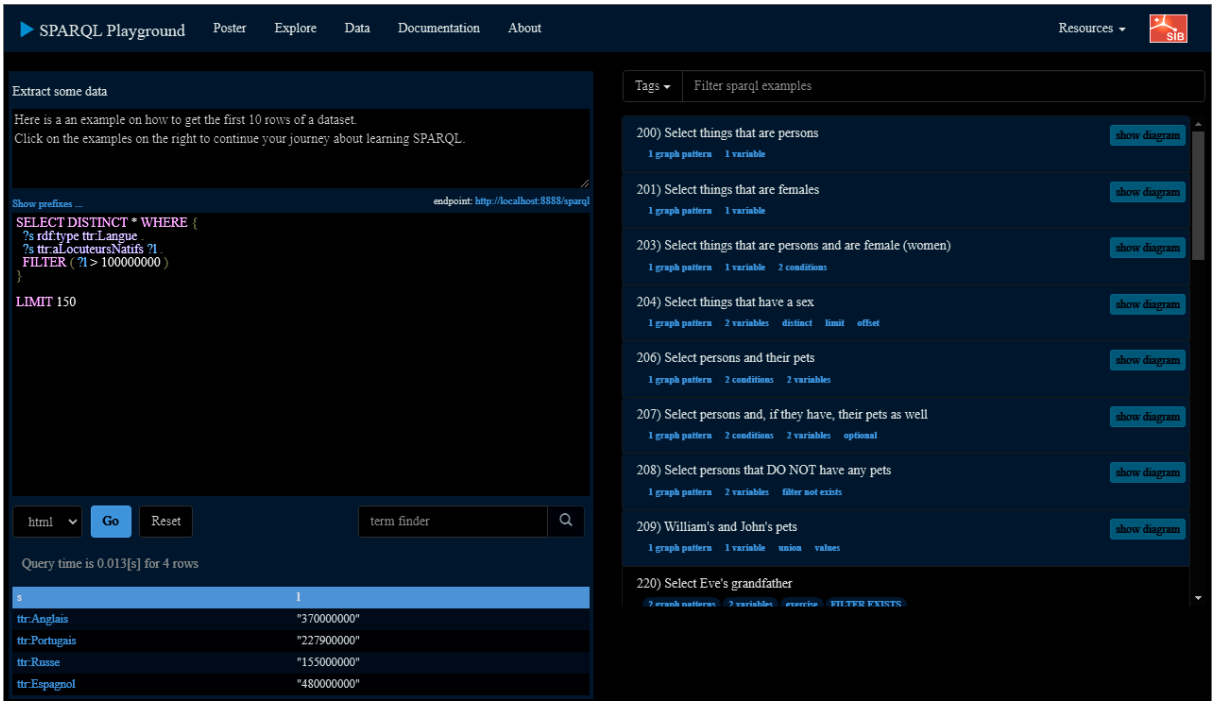
Alexandra DEMKOVIC
Marceau HERNANDEZ
Didier Kouassi KOUAME

Annexe

Annexe 1. Requêtes SPARQL

1. **Quelles sont les langues qui ont un nombre de locuteurs natifs supérieur à 10 millions ?**

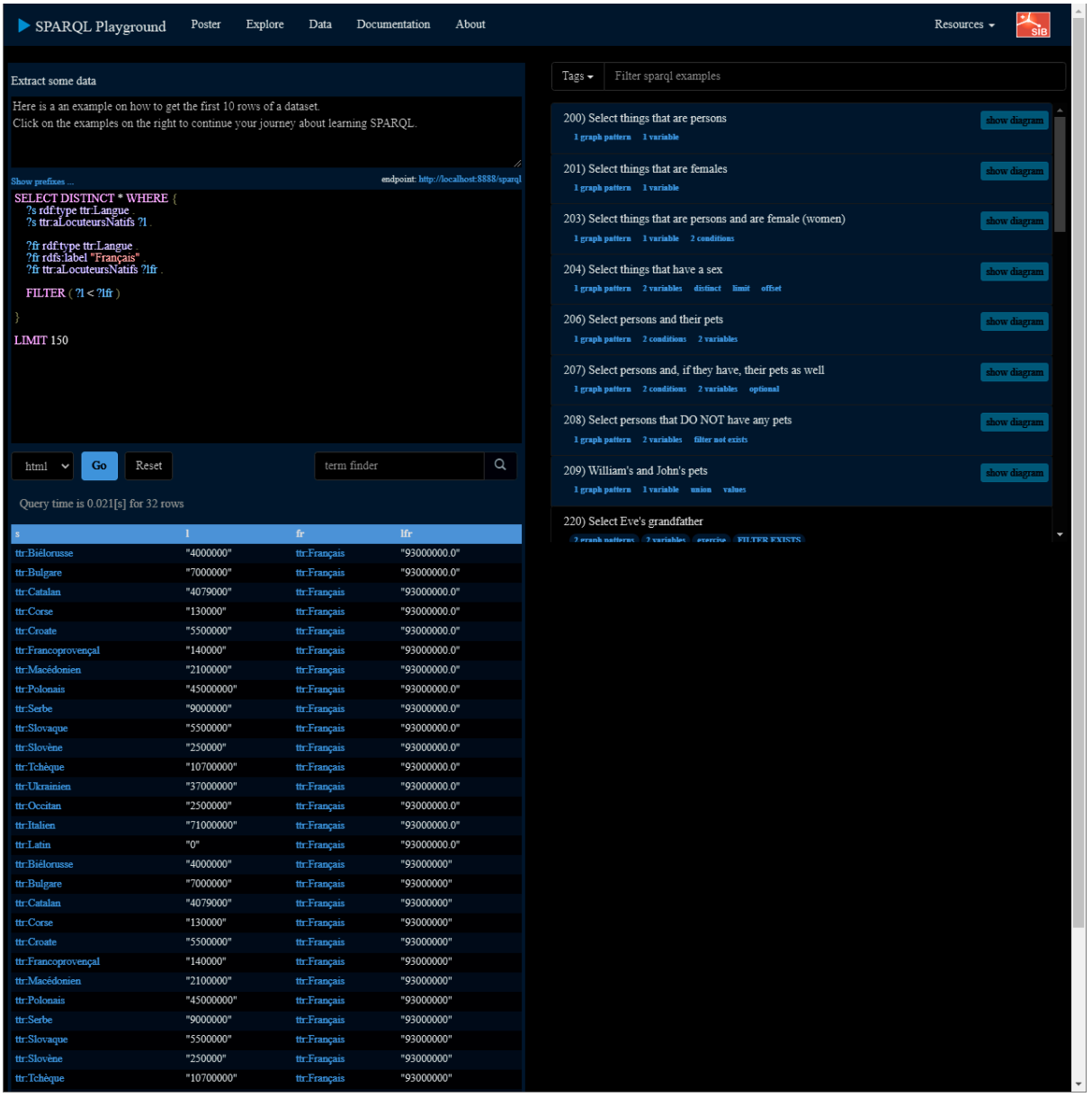
```
SELECT DISTINCT * WHERE {
  ?s rdf:type ttr:Langue .
  ?s ttr:aLocuteursNatifs ?l .
  FILTER ( ?l > 100000000 )
}

LIMIT 150
```



2. **Quelles sont les langues qui ont moins de locuteurs que le français ?**

```
SELECT DISTINCT * WHERE {
   ?s rdf:type ttr:Langue .
   ?s ttr:aLocuteursNatifs ?l .

   ?fr rdf:type ttr:Langue .
   ?fr rdfs:label "Français" .
```

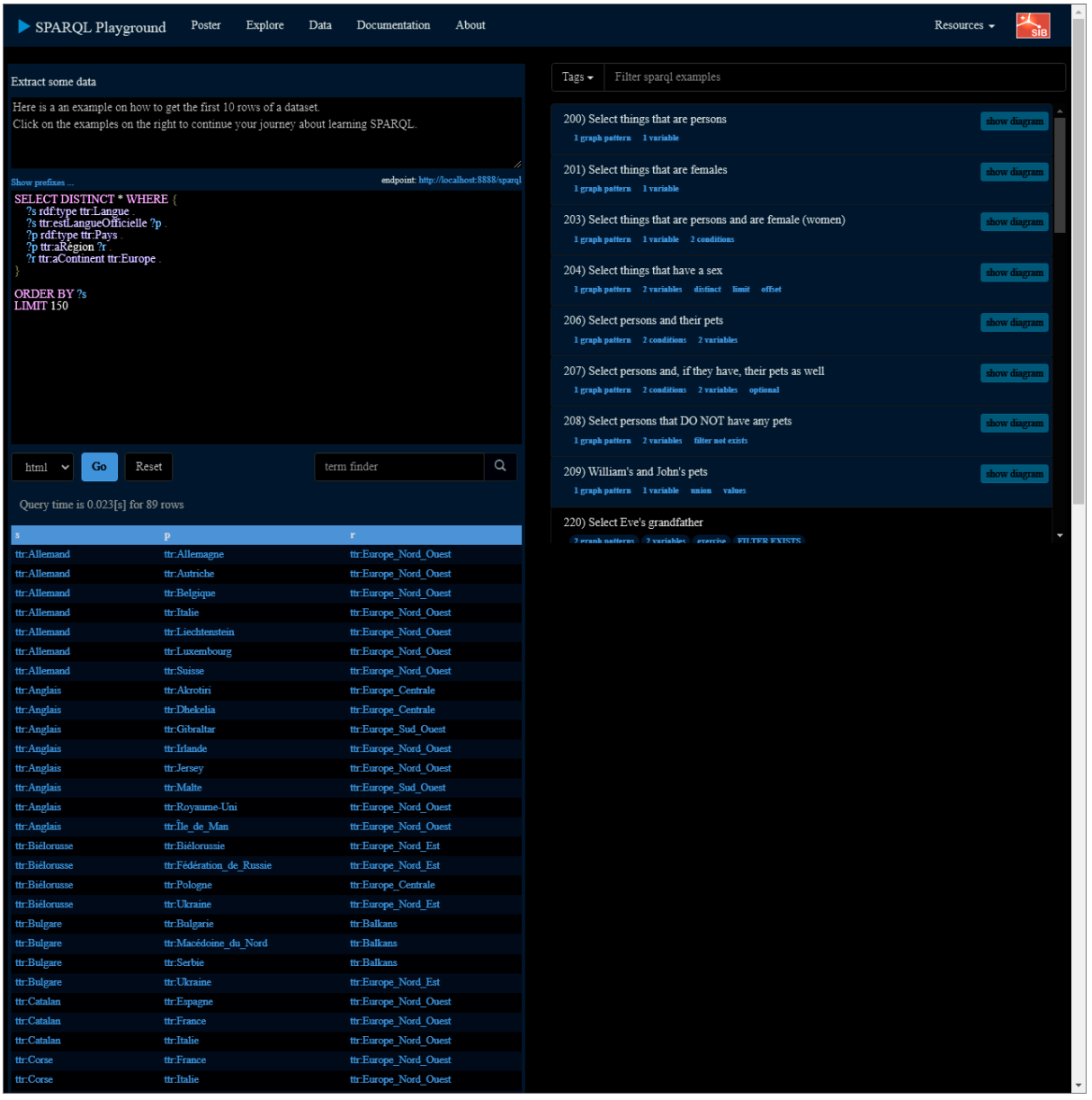
```
    ?fr ttr:aLocuteursNatifs ?lfr .

    FILTER ( ?l < ?lfr )
}

LIMIT 150
```



3. Quelles sont les langues parlées en Europe ?

```
SELECT DISTINCT * WHERE {
    ?s rdf:type ttr:Langue .
    ?s ttr:estLangue ?p .
```

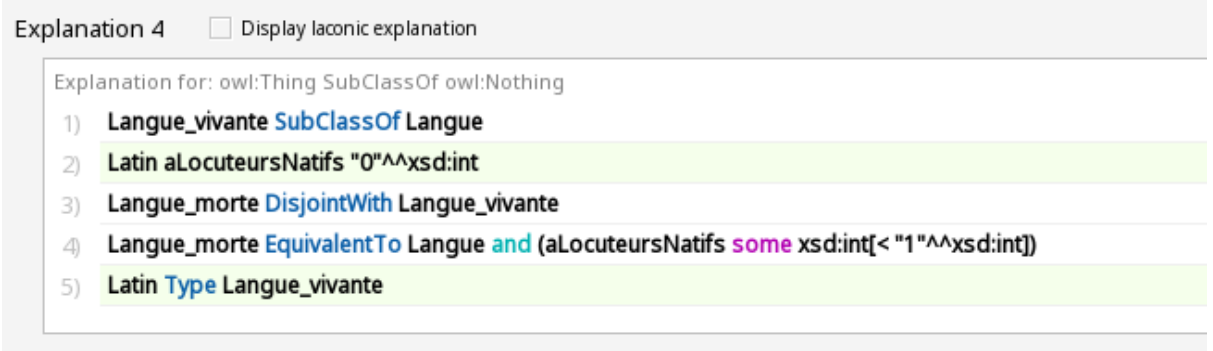
```
    ?p rdf:type ttr:Pays .
    ?p ttr:aRégion ?r .
    ?r ttr:aContinent ttr:Europe .
}

ORDER BY ?s
LIMIT 150
```



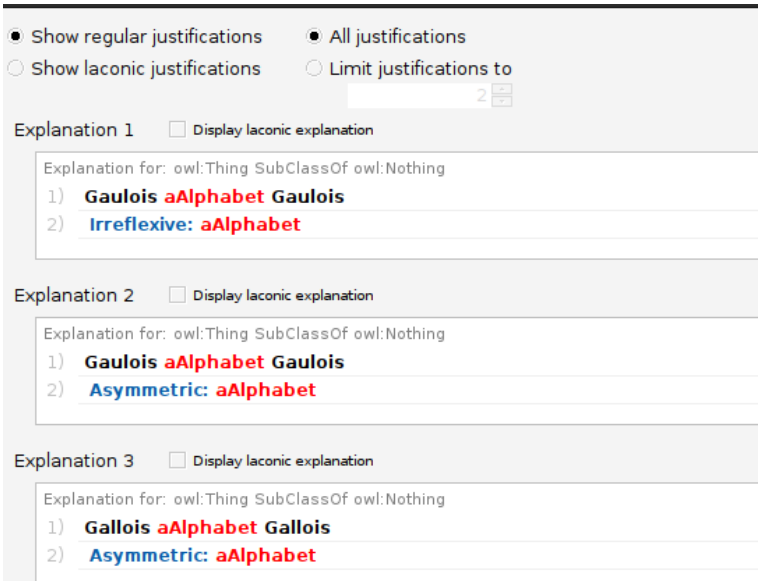
Annexe 2: Inconsistances

Aucune connaissance utile ne peut être déduite de l'ontologie ou aucune des classes de l'ontologie ne peut avoir d'instances.

Exemple 1


Explanation 4
Display laconic explanation
Explanation for: owl:Thing SubClassOf owl:Nothing
1) Langue_vivante SubClassOf Langue
2) Latin aLocuteursNatifs "0"^^xsd:int
3) Langue_morte DisjointWith Langue_vivante
4) Langue_morte EquivalentTo Langue and (aLocuteursNatifs some xsd:int[< "1"^^xsd:int])
5) Latin Type Langue_vivante

Exemple 2


Show regular justifications
All justifications
Show laconic justifications
Limit justifications to
2
Explanation 1
Display laconic explanation
Explanation for: owl:Thing SubClassOf owl:Nothing
1) Gaulois aAlphabet Gaulois
2) Irreflexive: aAlphabet
Explanation 2
Display laconic explanation
Explanation for: owl:Thing SubClassOf owl:Nothing
1) Gaulois aAlphabet Gaulois
2) Asymmetric: aAlphabet
Explanation 3
Display laconic explanation
Explanation for: owl:Thing SubClassOf owl:Nothing
1) Gallois aAlphabet Gallois
2) Asymmetric: aAlphabet