TP 4 – introduction à SPARQL

L'objectif de cette séance de TP est de découvrir le langage de requêtes SPARQL

Pour comprendre l'utilisation des éléments SPARQL utilisés dans ce TP, les liens étiquetés en bleu précédés de W3C vous renvoient vers la page correspondante dans le document de référence du W3C consacré à la norme SPARQL 1.1 (https://www.w3.org/TR/sparql11-query/).

Exercice 1: Prise en main du SPARQL endpoint de DBPEDIA

"DBpedia est un projet universitaire et communautaire d'exploration et extraction automatiques de données dérivées de Wikipédia. Son principe est de proposer une version structurée et sous forme de données normalisées au format du web sémantique des contenus encyclopédiques de chaque fiche encyclopédique. DBpedia vise aussi à relier à Wikipédia (et inversement) des ensembles d'autres données ouvertes provenant du Web des données." *(http://fr.wikipedia.org/wiki/DBpedia)*

L'url http://dbpedia.org/sparql correspond à un point d'accès SPARQL (*SPARQL endpoint*) aux données de DBPedia. Vous allez utiliser celui-ci pour explorer les données RDF de DBpedia.

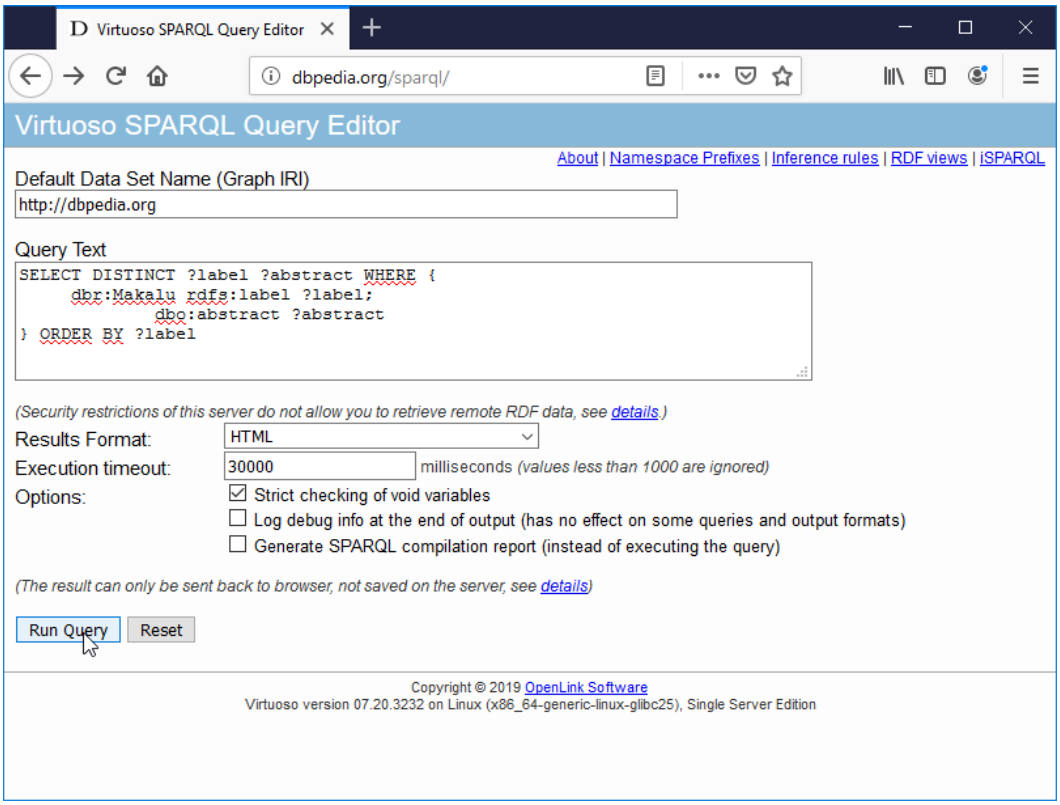


Figure 1: l'interface web du point d'accès SPARQL de DBPedia

Question 1: Dans la zone texte prévue à cet effet tapez la requête SPARQL suivante :

```
SELECT ?m ?a WHERE {
   ?m a dbo:Mountain;
```

```
        dbo:locatedInArea dbr:Nepal;
        dbo:elevation ?a.
}
```

1. D'après vous que fait cette requête ?
2. à quoi correspond le `a` dans le pattern `?m a dbo:Mountain;` ? A quelle URI se substitue-t-il ?
3. A quelles URIs correspondent les préfixes `dbo:`, `dbr:` ?
4. Exécutez la requête et observez le résultat obtenu. Correspond-t-il à ce que vous aviez prévu ?

Question 2:

1. A quoi correspond l'url `http://dbpedia.org/resource/Saipal` qui apparait dans la colonne *m* du tableau résultat ?
2. Cliquez sur ce lien. A quoi correspond le résultat obtenu ?
3. Le lien était bien `https://dbpedia.org/resource/Saipal` (vous pouvez le vérifier en tapant directement cette url dans la barre d'adresse de votre navigateur), pourtant l'url de la page affichée est `https://dbpedia.org/page/Saipal`.
    - Que s'est-il passé ? Pouvez-vous expliquer ce comportement de votre navigateur ?
    - Que faudrait-il modifier dans la requête HTTP pour obtenir en RDF les données de DBpedia concernant Saipal ?
    - Récupérez ces données au format Turtle. Expliquez comment vous avez procédé.

Question 3 : Donnez une requête SPARQL qui permet d'obtenir les URIs et l'altitude des dix plus hauts sommets du Népal.

Question 4 : Complétez la requête précédente pour obtenir en plus de l'altitude, le nom de chaque sommet défini par la valeur de la propriété `rdfs:label`.

1. Qu'observez-vous ? Combien de sommets différents obtenez-vous ? Pourquoi ?
2. Modifiez la requête pour faire en sorte de n'obtenir que les 10 plus hauts sommets du Népal avec uniquement leur nom en anglais.

Question 5 : La requête précédente donne-t-elle tous les sommets du Népal de plus de 8000 m ?

1. Modifiez la requête de façon à obtenir dans l'ordre du plus haut au plus bas, l'altitude et le nom anglais de tous les sommets népalais de plus de 8000 m.
2. Combien de sommets trouvez-vous ? Ecrivez une requête qui vous donne directement ce résultat (le nombre de sommets népalais de plus de 8000 m).

Question 6 : Si tous les sommets népalais de plus de 8000m ont un nom en anglais, DBpedia ne contient le nom en russe que de certains d'entre eux.

1. Pouvez-vous dire combien et lesquels ? Indiquez-la ou les requêtes qui vous a permis de répondre à ces questions.

2. Ecrivez une requête qui pour ces sommets donne simultanément leur URI, leur altitude, leur nom en anglais et leur nom en russe et les trie selon leur altitude (l'ordre décroissant).
3. Inversement donnez une requête qui donne la liste des sommets de plus de 8000 qui n'ont pas de nom (label) en russe.

    Donnez une requête qui donne la liste des tous les sommets de plus de 8000 avec leur altitude, leur nom en anglais et **s'il existe** leur nom en russe.

Question 7 : a) Donnez une requête qui vous permet de trouver les sommets dont la première ascension a été effectuée par Lionnel Terray, l'année de cette première ascension et si DBpedia dispose des données le nom du ou des alpinistes avec qui Lionel Terray a accompli cette première.

b) Donnez une requête qui vous permet de trouver les sommets dont la première ascension a été effectuée par Lionnel Terray, l'année de cette première ascension et si DBpedia dispose des données le nom **et le pays de naissance** du ou des alpinistes avec qui Lionel Terray a accompli cette première.

Vous devriez obtenir un résultat ressemblant à celui obtenu dans l'image ci-dessous

sommet	annee	compagnonCordee	pays
"Makalu"@en	"1955"^^<http://www.w3.org/2001/XMLSchema#gYear>	"Jean Couzy"@en	"France"@en
"Chomo Lonzo"@en	"1954"^^<http://www.w3.org/2001/XMLSchema#gYear>	"Jean Couzy"@en	"France"@en
"Huantsán"@en	"1952"^^<http://www.w3.org/2001/XMLSchema#gYear>		

Résultats attendus

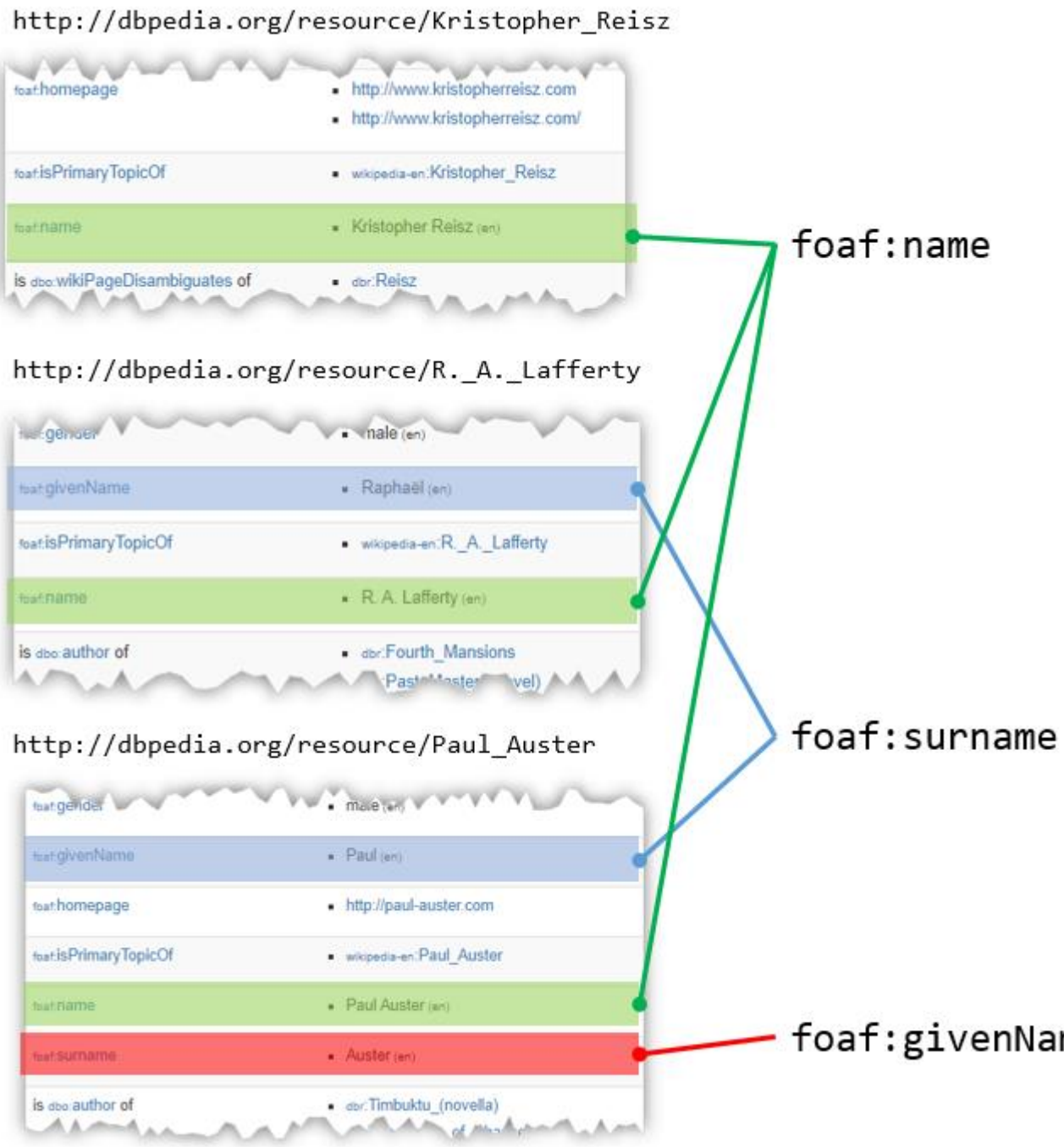
Exercice 2 : Recherche dans DBPEDIA des informations relatives aux écrivains américains

Vous allez maintenant utiliser DBpedia pour retrouver des informations qui vous permettront d'enrichir les données sur les écrivains américains sur lesquelles vous avez travaillé la semaine dernière.

1. Dans DBpedia quelles sont les propriétés de l'écrivain Paul Auster ? Ecrivez une requête SPARQL permettant d'obtenir la liste des URI de ces propriétés, ordonnée par ordre alphabétique.
2. Parmi ces propriétés, lesquelles permettent selon vous de savoir quand et où il est né ? Ecrivez une requête SPARQL permettant d'obtenir ces réponses.
3. Ecrivez une requête SPARQL permettant de savoir si Paul Auster est mort. (hint: requête ASK)
4. Ecrivez une requête permettant de trouver les classes dont Paul Auster est instance. Parmi celles-ci, laquelle définie dans l'ontologie DBpedia (préfixe `dbo:`) vous semble la plus appropriée pour déterminer que Paul Auster est un écrivain ?

5. En utilisant ce type, écrivez une requête permettant de déterminer le nombre d'écrivains présents dans DBpedia. Combien en avez-vous trouvé ? (hint: utilisez la fonction d'aggrégation COUNT)
6. Mais êtes-vous sûr d'avoir retrouvé toutes les personnes ayant une activité d'écrivain ? Par exemple le français Bernard Rapp, journaliste à la base, a aussi eu une activité d'écriture et peut à ce titre être aussi considéré comme un écrivain.

   Bernard Rapp, n'a pas le type écrivain (`dbo:Writer`) mais il possède la propriété `dbo:occupation` avec la valeur `dbr:Writer`. Ecrivez une requête qui permet de savoir combien de personnes sont dans ce cas dans le graphe DBpedia. (hint: utilisez un filtre NOT EXISTS)
7. Ecrivez une requête qui calcule le nombre d'écrivains présents dans DBpedia, en considérant qu'un écrivain est soit une ressource typée par `dbo:Writer`, soit une ressource ayant la propriété `dbo:occupation dbr:Writer`. Le résultat obtenu devrait être la somme des résultats obtenus aux deux questions précédentes. (hint: utilisez un pattern alternatif dans la requête à l'aide de UNION)
8. Modifiez la requête précédente pour ne retrouver que les écrivains nés aux Etats-Unis. Combien en avez-vous trouvé ?
9. Pour Paul Auster, les propriétés `foaf:givenName` et `foaf:surname` donnent respectivement son prénom et son nom de famille. Mais tous les écrivains américains de DBpedia possèdent-ils ces propriétés ? Ecrivez une requête permettant de répondre à cette question.
10. Pour lier les données d'Artemis Bookstore avec celles de DBpedia, nous allons nous servir des noms et prénoms des différents auteurs. Nous avons vu dans la question précédente que pour certains ces informations sont directement accessibles à partir des propriétés `foaf:givenName` et `foaf:surname`. Mais cela n'est pas le cas pour tous. Certains ont uniquement la propriété `foaf:givenName` (comme par exemple Raphaël Lafferty, d'autres comme Kristopher Reisz n'ont aucune des deux. Par contre tous possèdent une propriété `foaf:name`, le plus souvent il s'agit de la concaténation du nom et prénom comme pour Paul Auster, mais pas toujours. Par exemple, pour Raphaël Lafferty il s'agit des initiales de ses prénoms et de son nom de famille.



utilisation des propriétés foaf pour nommer les écrivains dans DBpedia Ecrire une requête SPARQL qui pour permet de retrouver l'URI, le nom foaf (`foaf:name`) et si ils existent le prénom (`foaf:givenName`) et le nom de famille (`foaf:surname`). Les résultats attendus doivent être de la forme : Resultat de la recherche des propriétés fof des écrivains américains dans DBpedia

Exercice 3: Lier vos données avec DBPedia

Dans cet exercice il s'agit d'enrichir votre jeu de données afin de rajouter des triplets permettant de vous lier avec DBpedia. Par exemple, si dans votre jeu de données Paul Auster est identifé par l'URI `http://artemisBookstore.com/abr/PaulAuster`, le triplet permettant de le lier à sa représentation dans DBpedia sera:

```
abr:PaulAuster owl:sameAs dbr:Paul_Auster
```

avec les préfixes suivants

```
PREFIX          abr:          <http://artemisBookstore.com/abr/>          .
@prefix          owl:          <http://www.w3.org/2002/07/owl#>          .
@prefix dbr: <http://dbpedia.org/resource/> .
```

Pour cela nous allons procéder en deux temps

1. Exécuter sur votre poste de travail une requête permettant de récupérer un fichier CSV contenant l'URI, le nom et le prénom de tous les écrivains américains connus par DBpedia.
2. Ecrire un programme permettant de mettre en correspondance les écrivains de votre jeu de données avec ceux que vous avez récupérés et de générer les triplets `owl:sameAs` reliant vos écrivains à ceux de DBpedia.

a) Récupérer les écrivains de DBpedia

Pour effectuer cette tâche nous allons utiliser le moteur de requête ARQ disponible avec la distribution de JENA. ARQ peut être utilisé soit au travers d'une API dans des applications Java soit comme un outil depuis la ligne de commandes (voir le document *Command-Line SPARQL with Jena* de Richard Cyganiak). C'est cette deuxième possibilité que vous allez utiliser aujourd'hui.

1. sur le SPARQL endpoint de DBpedia écrivez une requête SPARQL permettant de rétrouver l'URI, le nom et le prénom de tous les écrivains américains connus dans DBpedia.
2. recopiez cette requête dans un fichier texte `dbpediaAmericanWriter.rq`
3. exécutez cette requête en utilisant la commande `rsparql` d'ARQ Jena afin de récupérer sur votre poste de travail un fichier contenant le résultat de la requête au format CSV.

b) Créer les liens entre vos données et DBpedia

Ecrivez un programme java qui, à partir des données CSV récupérées lors de l'exercice précédant, crée les liens entre vos données et DBpedia en se basant sur les noms et prénoms des écrivains.

La sortie de votre programme devra être un fichier au format Turtle, contenant les triplets reliant vos ressources aux ressources correspondantes dans DBpedia à l'aide de la propriété `owl:sameAs`