Données Semi-Structurées:

Projet Final

Enseignant: *Dario COLAZZO*
Chargée de TD/TP: *Beatrice NAPOLITANO*

Deadline: 25/05/2020

Description du projet

L'objectif de ce projet est de couvrir tous les sujets du cours afin de donner une vision d'ensemble des thèmes abordés. En effet, lorsqu'on travaille avec des données il peut arriver de devoir passer d'un type de structure à un autre pour diverses raisons : pour les besoins des entreprises, des logiciels, des connaissances.
A partir de vocabulaires rdf déjà existants, vous êtes demandé de

1. les étendre en créant un schéma rdf ;
2. réaliser un document rdf en format xml respectant le schéma créé ;
3. transformer ce document en format turtle à l'aide d'un programme python ;
4. saturer le document obtenu en utilisant des fonctions python ;
5. transformer le document saturé en un fichier json ;
6. effectuer des requêtes sur le document obtenu.

À la fin, vous devrez télécharger un dossier avec votre nom au format "NOM_Prenom" dans le dépôt git disponible au lien `https://github.com/xoxor/Donnees_semi_structurees`. Le dossier doit contenir les fichiers suivants :

— un document "1_rdfSchema.rdfs" ;
— un document "2_rdfXml.rdf" ;
— un fichier "3_xmlToTurtle.py" qui produit le document "3_rdfTurtle.rdf" ;
— un fichier "4_saturation.py" qui produit le document "4_rdfSaturated.rdf" ;
— un fichier "5_rdfToJson.py" qui produit le document "5_converted.json" ;
— un fichier contenant vos requêtes PostgreSQL.

1 Créer un vocabulaire RDF

1.1 Théorie

Un vocabulaire est un modèle de données comportant des classes, propriétés et relations qui peut être utilisé pour décrire vos données et métadonnées. C'est-à-dire, un vocabulaire RDF est un ensembles de termes utilisés pour décrire des choses. Un terme est soit une **classe**, soit une propriété.

— Propriétés de type d'objet (les **relations**) ;
— Propriétés de type de données (les **attributs**).

Classe. Une construction qui représente des choses dans le monde réel et/ou des informations, par exemple une personne, une organisation, un concept tel que "santé" ou "liberté".
Relation. Un lien entre deux classes ; par exemple le lien entre un document et l'organisation qui l'a publié (c-à-d organisation *publie* document). En RDF, les relations sont codées comme des propriétés de type d'objet.
Propriété. Une caractéristique d'une classe dans une dimension particulière, comme le nom légal d'une organisation ou la date et l'heure où une observation a été faite.

Les schémas et les vocabulaires RDF comprennent souvent des termes qui sont très génériques. En créant des sous-classes et des relations de sous-propriété, les systèmes qui comprennent la super propriété ou super classe peuvent être capables d'interpréter les données, même si les termes plus spécifiques sont inconnus.

1.1.1 Vocabulaires RDF existants

Une ressource utile pour trouver les vocabulaires RDF existants est le Linked Open Vocabularies repository [LOV], mais plusieurs autres services pour le même usage existent. Nous utiliserons les

vocabulaires de base :

— `RDF URI http://www.w3.org/1999/02/22-rdf-syntax-ns#`
RDF a ses propres propriétés de base et les termes de ce vocabulaire apparaissent dans presque toutes les documents RDF. Les principales propriétés comprennent rdf :type (qui peut être simplement écrit comme "a" dans Turtle). La classe principale est la classe Resource - la super classe pour toutes les classes.
— `RDFS URI http://www.w3.org/2000/01/rdf-schema#`
Le schéma du W3C pour la description des schémas. Les termes de ce vocabulaire sont utilisés pour définir des classes, des propriétés, des sous-classes, des sous-propriétés, etc.
— `XML Schema URI http://www.w3.org/2001/XMLSchema#`
Plutôt que d'inventer ses propres types de données, RDF réutilise ceux définis dans XML Schema. Les principaux types sont les suivants : date, dateTime, anyURI, boolean, integer, float.

Lorsque de nouveaux termes sont nécessaires, créez-les suivant les meilleures pratiques communément admises :

— Les Classes commencent avec une majuscule et sont toujours au singulier.
— Les propriétés commencent par une lettre en minuscule.
— Les propriétés des objets devraient être des verbes.
— Les propriétés de type de données doivent être des noms.
— Utilisez le« Camel Case » si un terme a plus d'un mot.

1.2 Exercice

En lisant la documentation `https://www.w3.org/TR/rdf-primer/` et en partant des vocabulaires rdf existants, créez votre propre vocabulaire qui peut modeler notre collection de Films.

Exemple 1. *Schema XML :*

```
<rdf:RDF
        xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema#">

        <rdfs:Class rdf:ID="Person">
          <rdfs:label xml:lang="en">person</rdfs:label>
          <rdfs:label xml:lang="en">human being</rdfs:label>
          <rdfs:label xml:lang="en">human</rdfs:label>
          <rdfs:comment xml:lang="en">a member of the human species</rdfs:comment>
          <rdfs:label xml:lang="fr">personne</rdfs:label>
          <rdfs:label xml:lang="fr">etre humain</rdfs:label>
          <rdfs:label xml:lang="fr">humain</rdfs:label>
          <rdfs:label xml:lang="fr">homme</rdfs:label>
          <rdfs:comment xml:lang="fr">un membre de l'espece humaine.</rdfs:comment>
        </rdfs:Class>


        <rdfs:Class rdf:ID="Actor">
          <rdfs:subClassOf rdf:resource="#Person"/>
          <rdfs:label xml:lang="en">actor</rdfs:label>
          <rdfs:comment xml:lang="en">a person who portrays a character
            in a performance</rdfs:comment>
          <rdfs:label xml:lang="fr">acteur</rdfs:label>
          <rdfs:comment xml:lang="fr">une personne qui joue un role
            dans un spectacle</rdfs:comment>
        </rdfs:Class>

        <rdfs:Property rdf:ID="surname">
          <rdfs:label xml:lang="en">last name</rdfs:label>
          <rdfs:label xml:lang="fr">nom</rdfs:label>
          <rdfs:comment xml:lang="en">last name of a person</rdfs:comment>
          <rdfs:comment xml:lang="fr">nom de famille d'une personne</rdfs:comment>
        </rdfs:Property>

</rdf:RDF>
```

2 Réaliser un document RDF

2.1 Exercice

En partant du schéma que vous venez de réaliser, créez un document décrivant notre cinémathèque. Essayez de créer un document non saturé afin de pouvoir tester le programme au point 4.

Exemple 2. *Document XML :*

```
<rdf:RDF
xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#">

  <rdfs:Person rdf:ID="JTravolta">
    <rdf:type rdf:resource="#Actor"/>
    <rdfs:surname>Travolta</rdfs:surname>
  </rdfs:Person>

</rdf:RDF>
```

Le validateur RDF en ligne fourni par W3C (`https://www.w3.org/RDF/Validator/`) analyse votre document RDF, vérifie votre syntaxe et génère des vues tabulaires et graphiques de votre document RDF.

3 Transformation XML -> Turtle

Turtle est une syntaxe d'un langage qui permet une sérialisation non-XML des modèles RDF. Documentation : `https://www.w3.org/2007/02/turtle/primer/`. Le schéma en notation Turtle correspondant au schéma de l'Exemple 1 est le suivant :

Exemple 3. *Schema Turtle :*

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.

rdfs:Person a rdfs:Class;
rdfs:label "person"@en, "human"@en, "humain"@fr;
rdfs:comment "human being"@en, "etre humain"@fr.

rdfs:Actor a rdfs:Class;
rdfs:subClassOf rdfs:Person;
rdfs:label "actor"@en, "acteur"@fr;
rdfs:comment "a person who portrays a character in a performance"@en;
rdfs:comment "une personne qui joue un role
dans un spectacle"@fr.

rdfs:surname a rdf:Property;
rdfs:label "last name"@en, "nom"@fr.
```

Le datatype *xsd :integer* est identifié par son URIref (l'URIref complet étant

http ://www.w3.org/2001/XMLSchema#integer). Cet URIref peut être utilisé sans indiquer explicitement dans le schéma qu'il identifie un datatype. Cependant, il est utile d'indiquer explicitement qu'un URIref identifie un datatype. Cela peut être fait en utilisant la classe RDF Schema *rdfs :Datatype* :

```
xsd:integer rdf:type rdfs:Datatype .
```

3.1 Exercice

Écrire un programme python, qui partant d'un document rdf au format xml le transforme en format Turtle. Par exemple, à partir du document rdf de l'Exemple 2 nous devons obtenir :

Exemple 4. *Document Turtle :*

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.

<JTravolta> a rdfs:Actor;
rdfs:surname "Travolta".
```

4 Saturation du document RDF

4.1 Exercice

Écrire un programme python, qui à partir d'un document rdf non saturé, applique récursivement les règles de saturation.

5 Transformation Turtle -> JSON

5.1 Exercice

Écrire un programme python, qui partant d'un document rdf au format Turtle le transforme en format JSON.

6 Requêtes PostgreSQL

6.1 Exercice

Écrivez en utilisant PostgreSQL les requêtes suivantes :
— La liste des titres de films.
— Les titres des films parus en 1990
— Le résumé d'Alien
— Titre des films avec Bruce Willis
— Quels films ont un résumé ?
— Quels films n'ont pas de résumé ?
— Donner les titres des films vieux de plus de trente ans.
— Quel rôle joue Harvey Keitel dans Reservoir dogs ?
— . . . more TODO