Données Semi-Structurées:

Projet Final

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

Deadline: 31/03/2021

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. **[BONUS]** effectuer des requêtes sur le document obtenu.

Instructions

Il est possible de travailler sur votre projet en binôme, mais le plagiat entre projets invalidera les projets soumis par tous les binômes en question. Aucune tricherie ne sera tolérée.

À la fin, vous devrez déposer tous vos fichiers sur MyCourse. Les fichiers suivants doivent être présents :

- un document "1_rdfSchema.rdfs" ;
- un document "2_rdfXml.rdf" ;
- un fichier "3_xmlToTurtle.py" qui produit le document "3_rdfTurtle.ttl" ;
- un fichier "4_saturation.py" qui produit le document "4_rdfSaturated.ttl" ;
- un fichier "5_rdfToJson.py" qui produit le document "5_converted.json" (ou ".jsonld") ;
- **[BONUS]** un fichier "6_requetes.psql" contenant vos requêtes effectuées avec PostgreSQL ;
- un rapport détaillé "7_relation.pdf" avec une explication de toutes les étapes réalisées.

1 Créer un vocabulaire RDF

En suivant les instructions données dans le TP 4 et en considérant le schéma d'une cinémathèque vu dans le TP 1, créez un vocabulaire RDF qui peut modéliser notre collection de Films.

En cas de doute, consultez la documentation `https://www.w3.org/TR/rdf-primer/`.

Exemple 1. *Schema RDF (en 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

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

Exemple 2. *Document RDF (en 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/`.

Voici les principales règles pour lire ou écrire des triplets en Turtle :

- Un triplet simple est une séquence de termes (sujet, prédicat, objet) séparés par des espaces et terminée par un point (.).
- Il y a trois types de termes :
  - URI écrit entre < et >; un URI peut aussi s'exprimer par un nom qualifié (QNAME), i.e. précédé par un préfixe défini en utilisant les définitions @prefix, suivi d'un " :" et du reste de l'URI. L'URI complet s'obtient en remplaçant le préfixe par le contenu de sa définition. Un préfixe peut être vide (mais on garde alors le " :");
  - littéral entre guillemets ou accolades, possiblement suivie d'une définition de type suivi d'un nom de type souvent de la forme xsd: . . . pour les types standards de XML Schema ;
  - noeud vide indiqué par _:nodeId ou un groupe entre crochets.
- Un préfixe p est défini par @prefix p : URI. La virgule répète le sujet et le prédicat de triplets qui ne varient que par l'objet. p.e. $ex : a \quad ex : b \quad ex : c, \quad ex : d. \Rightarrow ex : a \quad ex : b \quad ex : c. \quad ex : a \quad ex : b \quad ex : d.$
- Le point-virgule répète le sujet de triplets qui ne varient que par le prédicat et l'objet. p.e. $ex : a \quad ex : b \quad ex : c; \quad ex : d \quad ex : e. \Rightarrow ex : a \quad ex : b \quad ex : c. \quad ex : a \quad ex : d \quad ex : e.$
- Les parenthèses regroupent les éléments d'une collection.

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. Vous pouvez utiliser la bibliothèque lxml pour gérer le document xml (ou *etree* de la bibliothèque *xml* ou ce que vous voulez) et utiliser la bibliothèque rdflib pour créer le fichier Turtle (comme dans l'exemple ici)

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

La saturation d'un document RDF permet de rendre explicite tous les triplets implicites d'une base de données RDF. Ce qui peut être saturé :

— le schéma : ne fait pas grand sens.
— les données basées sur le schéma associé : pour garantir l'exhaustivité des réponses aux requêtes sur la base de données.

On peut inférer de nouveaux triplets, à partir de :

— triplets existants ;
— relations de sous-classe ;
— relations de sous-propriété ;
— domaines et co-domaines.

4.1 Règles d'inférences/de saturation

4.1.1 Règles basées sur les relations de sous-classes et de sous-propriétés :

R1 : Si x de type C **ET** C $\subseteq$ C' **ALORS** x de type C'

$$(x, rdf : type, C) \wedge (C, rdfs : subclassOf, C') \Rightarrow (x, rdf : type, C')$$

Ex :
(Tom, rdf :type, Man) **ET** (Man, rdfs :subClassOf, Person) **ALORS** (Tom, rdf :type, Person)

R2 : Si p(x,y) **ET** p sous-propriétés de p' **ALORS** p'(x,y)

$$(x, p, y) \wedge (p, rdfs : subPropertyOf, p') \Rightarrow (x, p', y)$$

Ex :
(Tom, author, Report) **ET** (author, rdfs :subPropertyOf, creator) ALORS (Tom, creator, Report)

4.1.2 Règles basées sur la transitivité :

R3 : Transitivité des relations de sous-classe :
Si C sous-classe de C' **ET** C' sous-classe de C" **ALORS** C sous-classe de C"

$(C, rdfs : subclassOf, C') \wedge (C', rdfs : subclassOf, C'') \Rightarrow (C, rdfs : subclassOf, C'')$

R4 : Transitivité des relations de sous-propriété :
Si p sous-propriété de p' **ET** p' sous-propriété de p" **ALORS** p sous-propriété de p'

$(p, rdfs : subPropertyOf, p') \wedge (p', rdfs : subPropertyOf, p'') \Rightarrow (p, rdfs : subPropertyOf, p'')$

4.1.3 Règles basées sur les domaines et co-domaines

R5 : Si p propriété de domaine C **ET** p(x,y) **ALORS** x de type C

$$(x, p, y) \wedge (p, rdfs : domain, C) \Rightarrow (x, rdf : type, C)$$

R6 : Si p propriété de co-domaine C **ET** p(x,y) **ALORS** y de type C

$$(x, p, y) \wedge (p, rdfs : range, C) \Rightarrow (y, rdf : type, C)$$

4.2 Exercice

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

Conseil : Utilisez les fonctions de navigation sur Graph de la bibliothèque rdflib `https://rdflib.readthedocs.io/en/stable/intro_to_graphs.html`.

Exemple 5. *Règle R2 : (x, p, y)* ***ET*** *(p, rdfs : subPropertyOf, p1)* ***ALORS*** *(x, p1, y)*

```
def r2(x,p,y):
  predicates = mySchemaRDF.objects(subject=p, predicate=rdflib.RDFS.subPropertyOf)
  for p1 in predicates:
        if((x, p1, y) not in newSaturatedFile):
          newSaturatedFile.add((x, p1, y))
```

5 Transformation Turtle -> JSON

JSON est un langage d'échange de données, basé sur Javascript, et très utilisé en développement web. JSON-LD (JSON Linked Data) fournit une syntaxe JSON pour les graphes et les jeux de données RDF. JSON-LD propose des identificateurs universels pour les objets JSON, un mécanisme par lequel un document JSON peut faire référence à un objet décrit dans un autre document JSON ailleurs sur le Web, ainsi que des types de données et la gestion de la langue. JSON-LD fournit également un moyen de sérialiser les jeux de données RDF à l'aide du mot-clé **@graph**.
Un exemple de conversion d'un document rdf écrit en Turtle en JSON-LD peut être trouvé dans la documentation `https://www.w3.org/TR/json-ld/#turtle`.

Note : Outre JSON-LD, il existe une autre spécification pour la sérialisation des RDF dans JSON : RDF/JSON. Elle vise à sérialiser RDF dans une structure facile à utiliser pour les développeurs.

5.1 Exercice

Écrire un programme Python, qui partant d'un document rdf au format Turtle le transforme en format JSON. Un algorithme qui explique les étapes nécessaires pour effectuer cette transformation peut être trouvé ici :
`https://www.w3.org/TR/json-ld11-api/#serialize-rdf-as-json-ld-algorithm`.

6 BONUS - Requêtes avec PostgreSQL

PostgreSQL est un système de gestion de base de données relationnelle et objet.

Pour importer un documents json, il suffit de charger le fichier json dans une table temporaire en utilisant la syntaxe suivante :

```
CREATE TEMP TABLE file(info json);
\SET content `cat path/file.json`
INSERT INTO file VALUES(json_array_elements(:'content'));
```

Ici un exemple. En cas de doute, consultez la documentation `https://docs.postgresql.fr/9.5/index.html`

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.
- Les titres des films avec Bruce Willis.
- Quels films ont un résumé ?
- Quels films n'ont pas de résumé ?
- Les titres des films vieux de plus de trente ans.
- Quel rôle joue Harvey Keitel dans Reservoir dogs ?