

LABD

Master Info M1 2017-2018

LABD 9 : Vocabulaire RDF-RDFS

Rappel : déclaration RDF

Une **déclaration RDF** est un triplet constitué d'un **sujet**, d'un **prédicat** et d'un **objet** :

Un **sujet** (obligatoirement une ressource) : C'est un objet du domaine nommé par un identifiant (URI ou ressource anonyme)

Un **prédicat** : C'est une relation binaire sur le domaine entre un sujet et un objet nommé par un identifiant qui est une URI.

Un **objet** : C'est la valeur du prédicat pour le sujet. Il s'agit soit d'une ressource (identifiée par une URI ou anonyme), soit un littéral (chaîne de caractères ou nombre ou ...)



Rappel : Turtle

La ville de Lille se trouve dans le département du Nord.

```
@prefix dbon: <http://dbpedia.org/ontology/> .  
@prefix dbpr: <http://fr.dbpedia.org/property/> .  
@prefix labd: <http://labd/2015#> .  
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

labd:V_Lille

```
dbpr:nom "Lille"@fr ;  
labd:département "Nord"@fr .
```

Rappel : Turtle

La ville de Lille se trouve dans le département du Nord dont la première attestation documentaire date du 4 mars 1790.

```
@prefix dbon: <http://dbpedia.org/ontology/> .
@prefix dbpr: <http://fr.dbpedia.org/property/> .
@prefix labd: <http://labd/2015#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

labd:V_Lille

```
dbpr:nom "Lille"@fr ;
labd:département labd:D_Nord .
```

labd:D_Nord

```
dbpr:nom "Nord"@fr ;
labd:date-création "1790-03-04"^^xsd:date .
```

Rappel : Turtle

La ville de Lille se trouve dans le département du Nord dont la première attestation documentaire date du 4 mars 1790. La population de la ville de Lille est de 227560.

```
@prefix dbon: <http://dbpedia.org/ontology/> .
@prefix dbpr: <http://fr.dbpedia.org/property/> .
@prefix labd: <http://labd/2015#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

labd:V_Lille

```
dbon:populationTotal "227560"^^xsd:integer ;
dbpr:nom "Lille"@fr ;
labd:département labd:D_Nord .
```

labd:D_Nord

```
dbpr:nom "Nord"@fr ;
labd:date-création "1790-03-04"^^xsd:date .
```

Rappel : Turtle

La ville de Lille se trouve dans le département du Nord dont la première attestation documentaire date du 4 mars 1790. La population de la ville de Lille est de 227560 . Ce recensement date de 1er janvier 2010.

```
@prefix dbon: <http://dbpedia.org/ontology/> .
@prefix dbpr: <http://fr.dbpedia.org/property/> .
@prefix labd: <http://labd/2015#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

labd:V_Lille

```
dbon:demographics _:R3 ;
dbpr:nom "Lille"@fr ;
labd:département labd:D_Nord .
```

labd:D_Nord

```
dbpr:nom "Nord"@fr ;
labd:date-création "1790-03-04"^^xsd:date .
```

_:R3

```
dbon:populationTotal "227560"^^xsd:integer ;
dbon:year "2010-01-01"^^xsd:date .
```

Rappel : Turtle

```
@prefix dbon: <http://dbpedia.org/ontology/> .
@prefix dbpr: <http://fr.dbpedia.org/property/> .
@prefix labd: <http://labd/2015#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

labd:V_Lille

```
dbon:demographics [
    dbon:populationTotal 227560 ;
    dbon:year "2010-01-01"^^xsd:date
] ;
dbpr:nom "Lille"@fr ;
labd:département labd:D_Nord .
```

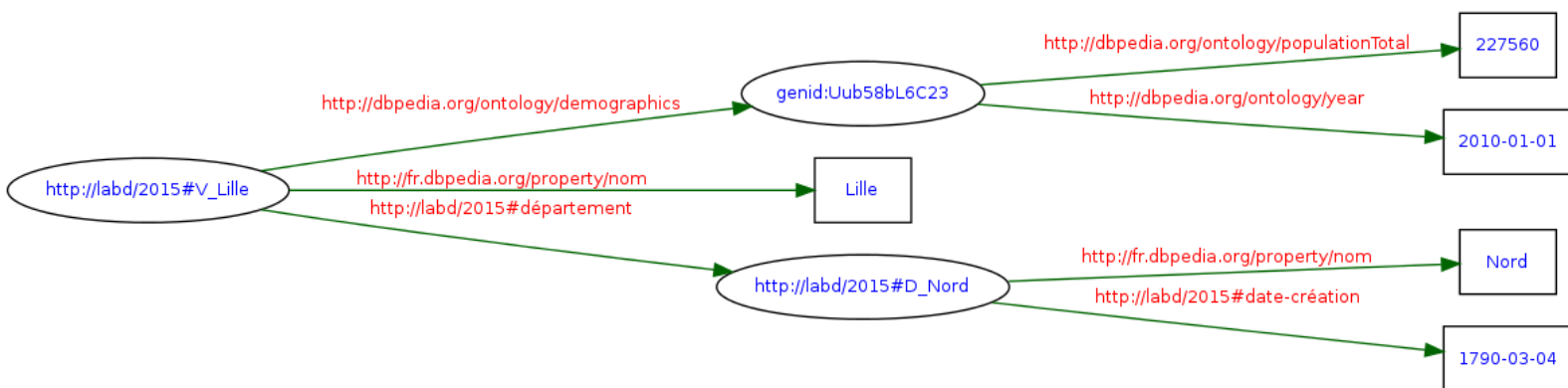
notation Turtle pour nœud blanc



labd:D_Nord

```
dbpr:nom "Nord"@fr ;
labd:date-création "1790-03-04"^^xsd:date .
```

Rappel : modèle de graphe



<https://www.w3.org/RDF/Validator/>

RDF/XML - Littéraux typés

On utilise l'attribut `rdf:datatype` pour indiquer le type du contenu d'un élément.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF [<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>
<rdf:RDF
  xmlns:rdf=... xmlns:dc=... xmlns:exterms=... >
  ...
  <rdf:Description rdf:about="http://www.example.org/index.html">
    <exterms:creation-date rdf:datatype="&xsd;date">
      1999-08-16
    </exterms:creation-date>
    ...
  </rdf:Description>
</rdf:RDF>
```

Typage des ressources

La valeur de la propriété `rdf:type` indique le type/la classe dont la ressource est [instance](#). On parle alors de [ressource typée](#).

```
<ex2terms:book78354>    rdf:type          <ex2terms:Book> .
<ex2terms:book78354>    ex2terms:author  _:author78354 .
_:author78354            rdf:type          <ex2terms:Person> .
_:author78354            ex2terms:name    "Jane Smith" .
```

Une ressource peut avoir plusieurs types.

Typage de ressource (2)

```
@prefix dbon: <http://dbpedia.org/ontology/> .
@prefix dbpr: <http://fr.dbpedia.org/property/> .
@prefix labd: <http://labd/2015#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
```

```
labd:V_Lille
  rdf:type labd:Ville ;
  dbon:demographics [
    dbon:populationTotal 227560 ;
    dbon:year "2010-01-01"^^xsd:date
  ] ;
  dbpr:nom "Lille"@fr ;
  labd:département labd:D_Nord .
```

```
labd:D_Nord
  a labd:Departement ;
  dbpr:nom "Nord"@fr ;
  labd:date-création "1790-03-04"^^xsd:date .
```

a est un raccourci Turtle pour la propriété **rdf:type**

Les Containers

Ils permettent de décrire des [regroupements](#) de ressources. Les éléments d'un container sont appelées [membres](#) du groupe.

Il existe [3 types](#) de containers prédéfinis :

- **rdf:Bag** : [multi-ensemble](#) de ressources ou littéraux
- **rdf:Seq** : [séquence](#) de ressources ou littéraux (l'ordre est important)
- **rdf:Alt** : [alternative](#) entre plusieurs ressources ou littéraux

Pour indiquer qu'une ressource est un container, on utilise la propriété **rdf:type**. Les containers sont qualifiés d'[ouverts](#), i.e. il peut exister d'autres membres du container que ceux indiqués par la description dont on dispose.

rdf:Bag

Multi-ensemble de valeurs : des valeurs peuvent être dupliquées, l'ordre n'a pas d'importance.

Exemple

```
<?xml version="1.0"?>
```

```
<rdf:RDF
```

```
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cd="http://www.recshop.fake/cd#">
```

```
  <rdf:Description rdf:about="http://www.recshop.fake/cd/Beatles">
```

```
    <cd:artist>
```

```
      <rdf:Bag>
```

```
        <rdf:li>John</rdf:li>
```

```
        <rdf:li>Paul</rdf:li>
```

```
        <rdf:li>George</rdf:li>
```

```
        <rdf:li>Ringo</rdf:li>
```

```
      </rdf:Bag>
```

```
    </cd:artist>
```

```
  </rdf:Description>
```

```
</rdf:RDF>
```

rdf:Bag

Multi-ensemble de valeurs : des valeurs peuvent être dupliquées, l'ordre n'a pas d'importance.

Exemple

```
@prefix ns0: <http://www.recshop.fake/cd#> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
```

```
<http://www.recshop.fake/cd/Beatles> ns0:artist [  
  a rdf:Bag ;  
  rdf:_1 "John" ;  
  rdf:_2 "Paul" ;  
  rdf:_3 "George" ;  
  rdf:_4 "Ringo"  
] .
```

a est un raccourci Turtle pour
la propriété `rdf:type`

notation Turtle pour nœud blanc

rdf:Seq

Liste de valeurs : des valeurs peuvent être dupliquées, l'ordre est important.

Exemple

```
@prefix ns0: <http://www.recshop.fake/cd#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

<http://www.recshop.fake/cd/Beatles> ns0:artist [
  a rdf:Seq ;
  rdf:_1 "George" ;
  rdf:_2 "John" ;
  rdf:_3 "Paul" ;
  rdf:_4 "Ringo"
] .
```

rdf:Alt

Alternative : une valeur parmi un choix de valeurs.

Exemple

```
@prefix ns0: <http://www.recshop.fake/cd#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

<http://www.recshop.fake/cd/Beatles> ns0:format [
  a rdf:Alt ;
  rdf:_1 "CD" ;
  rdf:_2 "Record" ;
  rdf:_3 "Tape"
] .
```


Collection : `rdf:List`

En RDF, une collection est une liste "*à la LISP*" : de type `rdf:List`, avec un premier élément `rdf:first` et une suite `rdf:rest`. La liste vide a la valeur `rdf:nil`.

Une collection est une liste **fermée** : elle forme un groupe qui ne contient que les membres spécifiés lors de la déclaration de la collection.

Exemple

```
@prefix ns0: <http://example.org/stuff/1.0/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

ns0:a ns0:b
  [ rdf:first "apple";
    rdf:rest [ rdf:first "banana";
               rdf:rest rdf:nil ] ]
```

Collection : `rdf:List`

En RDF, une collection est une liste "*à la LISP*" : de type `rdf:List`, avec un premier élément `rdf:first` et une suite `rdf:rest`. La liste vide a la valeur `rdf:nil`.

Une collection est une liste **fermée** : elle forme un groupe qui ne contient que les membres spécifiés lors de la déclaration de la collection.

Exemple

```
@prefix ns0: <http://example.org/stuff/1.0/> .
```

```
<http://example.org/stuff/1.0/a> ns0:b (
  "apple"
  "banana"
) .
```

Réification

Alice demande que Bob accepte les connexions entrantes

Réification

Alice demande que Bob accepte les connexions entrantes

Réification

sujet

prédicat

objet



Alice demande (Bob accepte connexions_entrantes)

Réification

Alice demande BobAccepteConnexions qui est un triplet dont le sujet est Bob, le prédicat est accepte et l'objet est connexions_entrantes

Réification

Alice demande BobAccepteConnexions.

BobAccepteConnexions est_un triplet.

BobAccepteConnexions a_comme_sujet Bob,

BobAccepteConnexions a_comme_prédicat accepte

BobAccepteConnexions a_comme_objet connexions_entrantes

Réification (N-triples)

```
<http://labd/2015/reification#Alice>  
<http://labd/2015/reification#demande>  
_:genid1 .
```

```
_:genid1  
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  
<http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement> .
```

```
_:genid1  
<http://www.w3.org/1999/02/22-rdf-syntax-ns#subject>  
<http://labd/2015/reification#Bob> .
```

```
_:genid1  
<http://www.w3.org/1999/02/22-rdf-syntax-ns#predicate>  
<http://labd/2015/reification#accepte> .
```

```
_:genid1  
<http://www.w3.org/1999/02/22-rdf-syntax-ns#object>  
<http://labd/2015/reification#connexions_entrantes> .
```


Réification (Turtle)

```
@prefix ns0: <http://labd/2015/reification#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

<http://labd/2015/reification#Alice> ns0:demande [
  a rdf:Statement ;
  rdf:subject ns0:Bob ;
  rdf:predicate ns0:accepte ;
  rdf:object ns0:connexions_entrantes
] .
```

Sémantique d'un vocabulaire

RDF permet de définir des graphes étiquetés, en utilisant des ressources du web.

Comment donner [le sens des "étiquettes"](#) du graphe RDF ?

Donner la sémantique des vocabulaires permet d'améliorer l'indexation des contenus en ligne, un traitement plus efficace des requêtes, des réponses plus pertinentes, et une meilleure interopérabilité des systèmes.

Outils : les ontologies. Elles permettent aussi un enrichissement des données en utilisant de l'inférence.

[Ontologies "légères"](#), RDFS.

Tout est ressource!

- les **classes** de ressources qui représentent des types de ressources, des ensembles de ressources
- les **propriétés** qui représentent des types de relations, des ensembles de relations possibles entre les ressources.
- la **relation de typage** (instanciation) pour dire qu'une ressource est d'un certain type;
- la **relation de sous-type** (subsomption) pour dire qu'une classe/propriété est sous classe /propriété d'une autre et que ses instances sont aussi instances de l'autre.
- les triplets!

`rdfs:Class`

Les ressources peuvent être "*rangées*" dans des groupes appelés [classes](#).

La propriété `rdf:type` permet de dire qu'une ressource est élément (on dit [instance](#)) d'une classe.

L'ensemble des instances d'une classe est appelée son [extension](#).

Deux [classes différentes](#) peuvent avoir la [même extension](#).

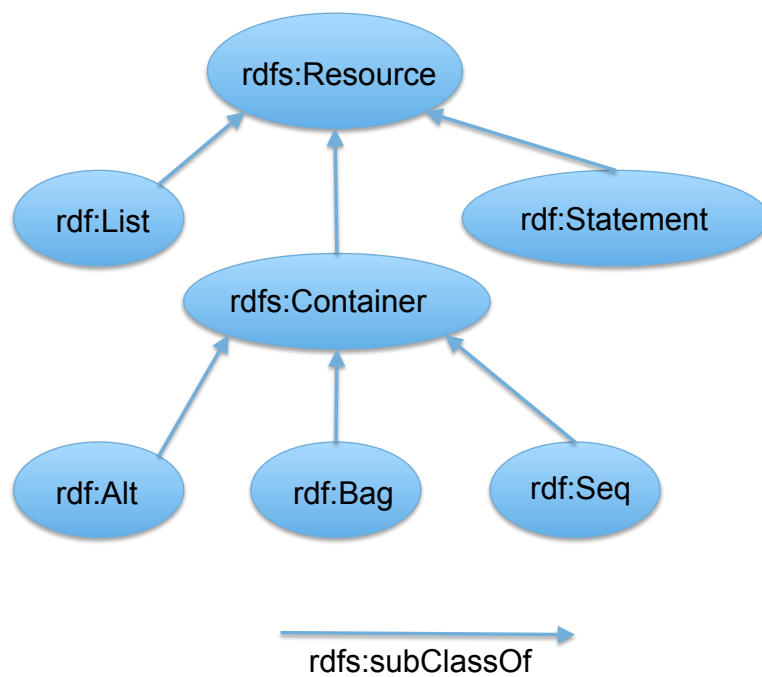
Une [classe est une ressource](#).

Les ressources qui sont des classes forment une classe appelée `rdfs:Class`

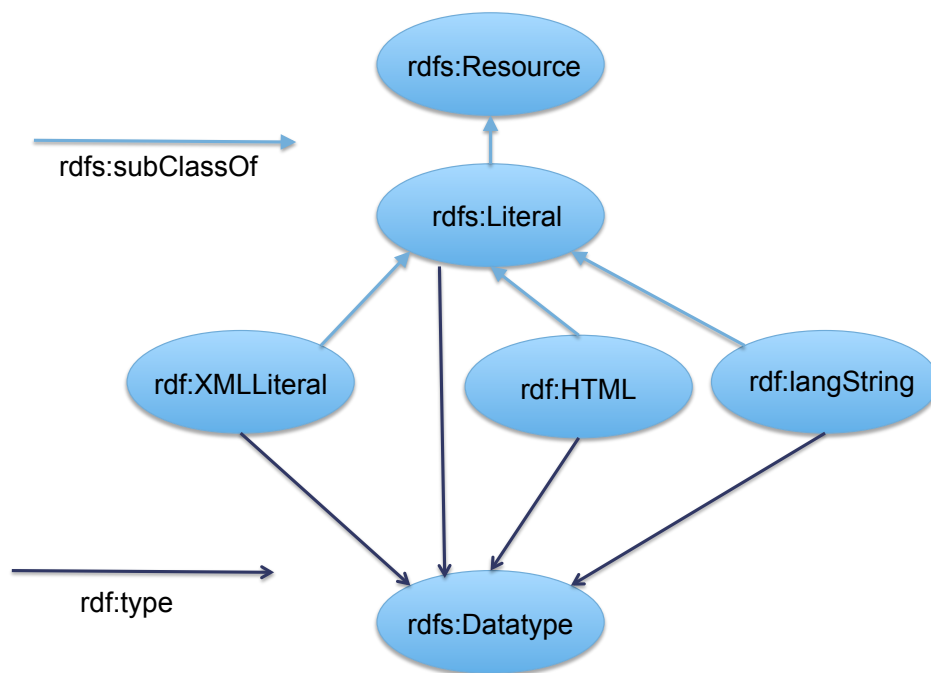
```
(ex:Alice , rdf:type , ex:Personne)
```

```
(ex:Personne , rdf:type , rdfs:Class)
```

`rdfs:subClassOf`

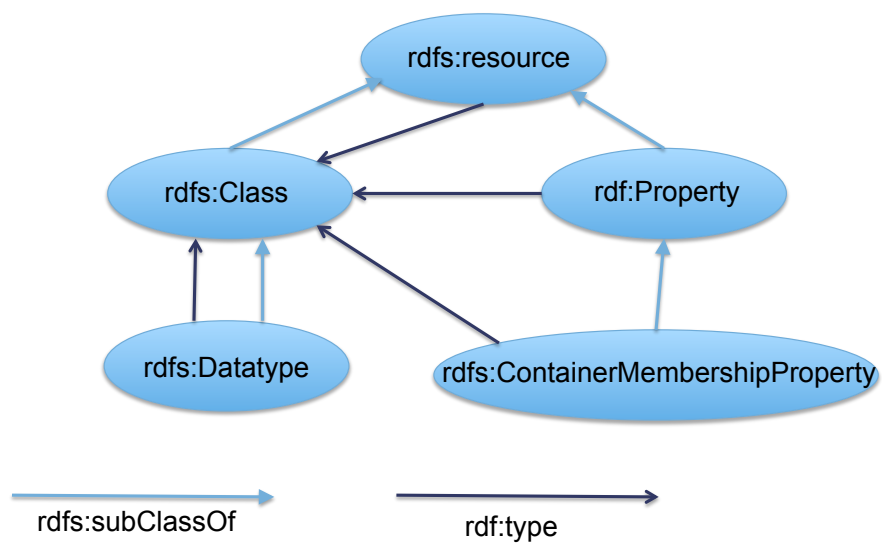


Types de données



Toute instance de **rdfs:Datatype** est sous-classe de **rdfs:Literal**

Tout est ressource!



Les propriétés

- Une **propriété** est de type `rdf:Property`.
- La relation de **sous-propriété** entre 2 propriétés est définie par `rdfs:subPropertyOf`.
- On peut définir le type du sujet (**domaine**) et/ou de l'objet (**co-domaine**) d'une propriété :
 - `rdfs:domain` : domaine d'une propriété
 - `rdfs:range` : co-domaine d'une propriété

```
(ex:habite, rdfs:domain, ex:Personne)
(ex:habite, rdfs:range, ex:Ville)
```
- Le **domaine** ou **co-domaine** d'une propriété peut contenir **plusieurs classes**.

```
(ex:habite, rdfs:range, ex:Departement)
```


Les classes RDF/RDFS

Classe	Commentaire
<code>rdfs:Resource</code>	Tout est ressource
<code>rdfs:Literal</code>	Donc du texte
<code>rdf:langString</code>	Du texte avec indication de langue (@fr)
<code>rdf:HTML</code>	Littéral HTML
<code>rdf:XMLLiteral</code>	Littéral XML
<code>rdfs:Class</code>	Une classe
<code>rdf:Property</code>	Une propriété
<code>rdfs:Datatype</code>	Un type de données
<code>rdf:Statement</code>	Un triplet
<code>rdf:Bag</code>	Un multi-ensemble
<code>rdf:Seq</code>	Une séquence
<code>rdf:Alt</code>	Une alternative
<code>rdfs:Container</code>	Un container
<code>rdfs:ContainerMembershipProperty</code>	<code>rdf:_1</code> , <code>rdf:_2</code> , ..., <code>rdfs:member</code>
<code>rdf:List</code>	Une liste (<code>rdf:nil</code> est instance de <code>rdf:List</code>)

Les propriétés RDF/RDFS

Propriété	Domaine	Co-domaine
<code>rdf:type</code>	<code>rdfs:Resource</code>	<code>rdfs:Class</code>
<code>rdfs:subClassOf</code>	<code>rdfs:Class</code>	<code>rdfs:Class</code>
<code>rdfs:subPropertyOf</code>	<code>rdf:Property</code>	<code>rdf:Property</code>
<code>rdfs:domain</code>	<code>rdf:Property</code>	<code>rdfs:Class</code>
<code>rdfs:range</code>	<code>rdf:Property</code>	<code>rdfs:Class</code>
<code>rdfs:label</code>	<code>rdfs:Resource</code>	<code>rdfs:Literal</code>
<code>rdfs:comment</code>	<code>rdfs:Resource</code>	<code>rdfs:Literal</code>
<code>rdfs:member</code>	<code>rdfs:Resource</code>	<code>rdfs:Resource</code>
<code>rdf:first</code>	<code>rdf:List</code>	<code>rdfs:Resource</code>
<code>rdf:rest</code>	<code>rdf:List</code>	<code>rdf:List</code>
<code>rdfs:seeAlso</code>	<code>rdfs:Resource</code>	<code>rdfs:Resource</code>
<code>rdfs:isDefinedBy</code>	<code>rdfs:Resource</code>	<code>rdfs:Resource</code>
<code>rdf:value</code>	<code>rdfs:Resource</code>	<code>rdfs:Resource</code>
<code>rdf:subject</code>	<code>rdfs:Statement</code>	<code>rdfs:Resource</code>
<code>rdf:predicate</code>	<code>rdfs:Statement</code>	<code>rdfs:Resource</code>
<code>rdf:object</code>	<code>rdfs:Statement</code>	<code>rdfs:Resource</code>

À ces propriétés, il faut ajouter `rdf:_1`, `rdf:_2`, ... qui sont toutes sous-propriétés de `rdfs:member` et instances de `rdfs:ContainerMembershipProperty`.

Exemple de schéma RDFS

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

<http://www.labd.org/2015/voitures/schema#Personne>
  a rdfs:Class ;
  rdfs:comment "La classe personne" .

<http://www.labd.org/2015/voitures/schema#Vehicule>
  a rdfs:Class ;
  rdfs:comment "La classe vehicule" .

<http://www.labd.org/2015/voitures/schema#Voiture>
  a rdfs:Class ;
  rdfs:comment "La classe voiture" ;
  rdfs:subClassOf <http://www.labd.org/2015/voitures/schema#Vehicule> .

<http://www.labd.org/2015/voitures/schema#conducteur>
  a rdf:Property ;
  rdfs:range <http://www.labd.org/2015/voitures/schema#Personne> ;
  rdfs:domain <http://www.labd.org/2015/voitures/schema#Vehicule> .
```

Exemple d'instance du schéma précédent

```
@prefix ns0: <http://www.labd.org/2015/voitures/schema#> .

<http://www.labd.org/2015/voitures/data#vo001>
  a ns0:Voiture ;
  ns0:conducteur <http://www.labd.org/2015/voitures/data#p101> .

<http://www.labd.org/2015/voitures/data#vo002>
  a ns0:Voiture ;
  ns0:conducteur <http://www.labd.org/2015/voitures/data#p102> .

<http://www.labd.org/2015/voitures/data#p102> a ns0:Personne .

<http://www.labd.org/2015/voitures/data#p101> a ns0:Personne .
```

Règles d'inférences - sous-typage

①

$$\frac{(x, \text{rdf:type}, C) \wedge (C, \text{rdfs:subClassOf}, C')}{(x, \text{rdf:type}, C')}$$

②

$$\frac{(x, p, y) \wedge (p, \text{rdfs:subPropertyOf}, p')}{(x, p', y)}$$

Règles d'inférences - transitivité

$$\textcircled{3} \frac{(C, \text{rdfs:subclassOf}, C') \wedge (C', \text{rdfs:subclassOf}, C'')}{(C, \text{rdfs:subclassOf}, C')}$$

$$\textcircled{4} \frac{(p, \text{rdfs:subPropertyOf}, p') \wedge (p', \text{rdfs:subPropertyOf}, p'')}{(p, \text{rdfs:subPropertyOf}, p')}$$

Règles d'inférences - domaine/co-domaine

$$\textcircled{5} \quad \frac{(x, p, y) \wedge (p, \text{rdfs:domain}, C)}{(x, \text{rdf:type}, C)}$$

$$\textcircled{6} \quad \frac{(x, p, y) \wedge (p, \text{rdfs:range}, C)}{(y, \text{rdf:type}, C)}$$

Exercice : Donner les inférences faites par RDFS

```
c:creator rdfs:domain c:Person
i:Man241 c:creator i:Image262
i:Man241 rdf:type c:Person
c:author rdfs:subPropertyOf c:creator
c:author rdfs:range c:Document
i:Woman297 c:author i:Book812
i:Book812 rdf:type c:Document
i:Woman297 c:creator i:Book812
i:Woman297 rdf:type c:Person
c:aSoutenu rdfs:domain c:Docteur
c:aSoutenu rdfs:range c:These
i:Woman297 c:aSoutenu i:t127
i:Woman297 rdf:type c:Docteur
i:t127 rdf:type c:These
c:nbDeRoues rdfs:domain c:Vehicule
i:Car207 c:nbDeRoues "4"^^xsd:integer
i:Car207 rdf:type c:Vehicule
```


Exercice : Donner les inférences faites par RDFS

```
c:creator rdfs:domain c:Person
i:Man241 c:creator i:Image262
i:Man241 rdf:type c:Person
c:author rdfs:subPropertyOf c:creator
c:author rdfs:range c:Document
i:Woman297 c:author i:Book812
i:Book812 rdf:type c:Document
i:Woman297 c:creator i:Book812
i:Woman297 rdf:type c:Person
c:aSoutenu rdfs:domain c:Docteur
c:aSoutenu rdfs:range c:These
i:Woman297 c:aSoutenu i:t127
i:Woman297 rdf:type c:Docteur
i:t127 rdf:type c:These
c:nbDeRoues rdfs:domain c:Vehicule
i:Car207 c:nbDeRoues "4"^^xsd:integer
i:Car207 rdf:type c:Vehicule
```

Exercice : Donner les inférences faites par RDFS

```
c:creator rdfs:domain c:Person
i:Man241 c:creator i:Image262
i:Man241 rdf:type c:Person

c:author rdfs:subPropertyOf c:creator
c:author rdfs:range c:Document
i:Woman297 c:author i:Book812
i:Book812 rdf:type c:Document
i:Woman297 c:creator i:Book812
i:Woman297 rdf:type c:Person

c:aSoutenu rdfs:domain c:Docteur
c:aSoutenu rdfs:range c:These
i:Woman297 c:aSoutenu i:t127
i:Woman297 rdf:type c:Docteur
i:t127 rdf:type c:These

c:nbDeRoues rdfs:domain c:Vehicule
i:Car207 c:nbDeRoues "4"^^xsd:integer
i:Car207 rdf:type c:Vehicule
```

Exercice : Donner les inférences faites par RDFS

```
c:creator rdfs:domain c:Person
i:Man241 c:creator i:Image262
i:Man241 rdf:type c:Person

c:author rdfs:subPropertyOf c:creator
c:author rdfs:range c:Document
i:Woman297 c:author i:Book812
i:Book812 rdf:type c:Document
i:Woman297 c:creator i:Book812
i:Woman297 rdf:type c:Person

c:aSoutenu rdfs:domain c:Docteur
c:aSoutenu rdfs:range c:These
i:Woman297 c:aSoutenu i:t127
i:Woman297 rdf:type c:Docteur
i:t127 rdf:type c:These

c:nbDeRoues rdfs:domain c:Vehicule
i:Car207 c:nbDeRoues "4"^^xsd:integer
i:Car207 rdf:type c:Vehicule
```

Exercice : Donner les inférences faites par RDFS

```
c:creator rdfs:domain c:Person
i:Man241 c:creator i:Image262
i:Man241 rdf:type c:Person

c:author rdfs:subPropertyOf c:creator
c:author rdfs:range c:Document
i:Woman297 c:author i:Book812
i:Book812 rdf:type c:Document
i:Woman297 c:creator i:Book812
i:Woman297 rdf:type c:Person

c:aSoutenu rdfs:domain c:Docteur
c:aSoutenu rdfs:range c:These
i:Woman297 c:aSoutenu i:t127
i:Woman297 rdf:type c:Docteur
i:t127 rdf:type c:These

c:nbDeRoues rdfs:domain c:Vehicule
i:Car207 c:nbDeRoues "4"^^xsd:integer
i:Car207 rdf:type c:Vehicule
```

Exercice

Définir un schéma RDFS pour les concepts suivants :

personne

enseignant

étudiant

cours

et les propriétés suivantes :

nom qui précise le nom d'un enseignant ou d'un cours. Le nom est donné sous la forme d'un littéral `http://www.w3.org/1999/02/22-rdf-syntax-ns#Literal`.

estSuiviPar qui précise que tel cours est suivi par telle liste d'étudiants

aCommeEnseignant qui précise que tel cours a comme enseignant tel enseignant.

Schéma

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

<http://www.labd.org/2015/cours/schema#Personne> a rdfs:Class .
<http://www.labd.org/2015/cours/schema#Cours> a rdfs:Class .
<http://www.labd.org/2015/cours/schema#Etudiant>
  a rdfs:Class ;
  rdfs:subClassOf <http://www.labd.org/2015/cours/schema#Personne> .

<http://www.labd.org/2015/cours/schema#Enseignant>
  a rdfs:Class ;
  rdfs:subClassOf <http://www.labd.org/2015/cours/schema#Personne> .

<http://www.labd.org/2015/cours/schema#estSuiviPar>
  a rdfs:Property ;
  rdfs:domain <http://www.labd.org/2015/cours/schema#Cours> ;
  rdfs:range rdf:Seq .

<http://www.labd.org/2015/cours/schema#nom>
  a rdfs:Property ;
  rdfs:domain <http://www.labd.org/2015/cours/schema#Cours>,
    <http://www.labd.org/2015/cours/schema#Personne> ;
  rdfs:range rdf:Literal .

<http://www.labd.org/2015/cours/schema#aCommeEnseignant>
  a rdfs:Property ;
  rdfs:domain <http://www.labd.org/2015/cours/schema#Cours> ;
  rdfs:range <http://www.labd.org/2015/cours/schema#Enseignant> .
```

Exercice, suite

Définir une instance RDF correspondant au schéma RDFS précédent pour exprimer :

Le cours « Programmation Orientée Objets » est enseigné par Jean-Christophe Routier.

Il est suivi par les étudiants Marie Martin, Pierre Dupont et Alain Durand.

Instance

```
@prefix ns0: <http://www.labd.org/2015/cours/schema#> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
```

```
<http://www.labd.org/2015/cours/data#P00>  
  a ns0:Cours ;  
  ns0:nom "Programmation Orientée Objets" ;  
  ns0:aCommeEnseignant <http://www.labd.org/2015/cours/data#jcr> ;  
  ns0:estSuiviPar [  
    a rdf:Seq ;  
    rdf:_1 <http://www.labd.org/2015/cours/data#etud01> ;  
    rdf:_2 <http://www.labd.org/2015/cours/data#etud02> ;  
    rdf:_3 <http://www.labd.org/2015/cours/data#etud03>  
  ] .
```

```
<http://www.labd.org/2015/cours/data#jcr>  
  a ns0:Enseignant ;  
  ns0:nom "Jean-Christophe Routier" .
```

```
<http://www.labd.org/2015/cours/data#etud01>  
  a <http://www.labd.org/2015/cours/schema#Etudiant> ;  
  ns0:nom "Marie Martin" .
```

```
<http://www.labd.org/2015/cours/data#etud02>  
  a ns0:Etudiant ;  
  ns0:nom "Pierre Dupont" .
```

```
<http://www.labd.org/2015/cours/data#etud03>  
  a ns0:Etudiant ;  
  ns0:nom "Alain Durand" .
```


Exercice, suite

Définir une instance RDF correspondant au schéma RDFS précédent pour exprimer :

Les cours « Programmation Orientée Objets » et « Technologies du Web 2 » sont enseignés par Jean-Christophe Routier.

Il sont suivis par les étudiants Marie Martin, Pierre Dupont et Alain Durand.

```
@prefix ns0: <http://www.labd.org/2015/cours/schema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

<http://www.labd.org/2015/cours/data#TW2>
  a ns0:Cours ;
  ns0:nom "Techonolgies du Web 2" ;
  ns0:estSuiviPar <http://www.labd.org/2015/cours/data#L3S4> ;
  ns0:aCommeEnseignant <http://www.labd.org/2015/cours/data#jcr> .

<http://www.labd.org/2015/cours/data#POO>
  a ns0:Cours ;
  ns0:nom "Programmation Orientée Objets" ;
  ns0:estSuiviPar <http://www.labd.org/2015/cours/data#L3S4> ;
  ns0:aCommeEnseignant <http://www.labd.org/2015/cours/data#jcr> .

<http://www.labd.org/2015/cours/data#jcr>
  ns0:nom "Jean-Christophe Routier" ;
  a ns0:Enseignant .

<http://www.labd.org/2015/cours/data#L3S4>
  a rdf:Seq ;
  rdf:_3 <http://www.labd.org/2015/cours/data#etud03> ;
  rdf:_1 <http://www.labd.org/2015/cours/data#etud01> ;
  rdf:_2 <http://www.labd.org/2015/cours/data#etud02> .

<http://www.labd.org/2015/cours/data#etud01>
  a <http://www.labd.org/2015/cours/schema#Etudiant> ;
  ns0:nom "Marie Martin" .

<http://www.labd.org/2015/cours/data#etud02>
  a ns0:Etudiant ;
  ns0:nom "Pierre Dupont" .

<http://www.labd.org/2015/cours/data#etud03>
  a ns0:Etudiant ;
  ns0:nom "Alain Durand" .
```

Exercices, tirés de l'examen de mai 2015

Exercice 1 : Modélisation RDF

Le texte suivant est extrait de la page Wikipedia de Johannes Brahms :

Joachim conseille à Brahms de s'adresser à Franz Liszt qui, à cette époque, est chef d'orchestre à la cour de Weimar. La légende voudrait que Brahms se soit endormi pendant que Liszt exécutait sa célèbre Sonate en si mineur.

Question 1 : Exprimez ces informations sous forme de triplets RDF en utilisant la syntaxe Turtle.

```
@prefix ns1: <http://www.example.org/labd#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

```
ns1:Joachim ns1:dit ns1:parlerALiszt .
```

```
ns1:parlerALiszt ns1:objet ns1:Liszt ;
  ns1:sujet ns1:Brahms ;
  ns1:verbe ns1:parlerA .
```

Joachim conseille à Brahms de s'adresser à Franz Liszt "qui à cette époque, est chef d'orchestre à la cour de Weimar". La légende voudrait que Brahms s'endorme pendant que Liszt exécutait sa célèbre Sonate en si mineur.

```
ns1:_ ns1:dit ns1:BrahmsADormi .
```

```
ns1:BrahmsADormi ns1:en_meme_temps_que ns1:LisztJoue ;
  ns1:objet "endormi" ;
  ns1:sujet ns1:Brahms ;
  ns1:verbe ns1:etat .
```

```
ns1:LisztJoue ns1:objet "Sonate en si mineur" ;
  ns1:sujet ns1:Liszt ;
  ns1:verbe ns1:joue .
```

Exercice 2 : Inférence RDFS

On considère l'ensemble de triplets suivants, donné au format Turtle :

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

uni:Person a rdfs:Class .
uni:Student
  a rdfs:Class ;
  rdfs:subClassOf uni:Person .
uni:Professor
  a rdfs:Class ;
  rdfs:subClassOf uni:Person .
uni:advises
  a rdf:Property ;
  rdfs:domain uni:Professor ;
  rdfs:range uni:Student ;
  rdfs:subPropertyOf uni:knows .
uni:knows a rdf:Property .

uni:alan uni:knows uni:bob .
uni:alan rdf:type uni:Professor .
uni:bob rdf:type uni:Student .
uni:alan rdf:type uni:Person .
uni:bob rdf:type uni:Person .
uni:alan uni:advises uni:bob .
```

Question 1 : Donnez cinq triplets supplémentaires qui peuvent être inférés par application des règles d'inférence de RDFS.