



SPARQL

**Comment illuminer vos mashups en consommant
les données du Linked Data ?**

SemWeb.pro, Paris, 18 janvier 2011



RDF RAPPEL

1- Prenez des choses/concepts/entités du monde réel et placez les dans le monde numérique, en les nommant et en les identifiant par des URIs.

**Signifié
dans le monde réel**

Tim Berners-Lee

Une personne

L'article « Semantic Web » de 2001

est

Un texte

**Signifiant
dans le monde numérique**

<<http://www.w3.org/People/Berners-Lee/card#i>>

<<http://xmlns.com/foaf/0.1/Person>>

<<http://www.sciam.com/article.cfm?id=the-semantic-web>>

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

<<http://purl.org/dc/dcmitype/Text>>

Puisque ces choses/concepts/entités sont identifiés par une URI,
ils sont assimilables à des ressources (RFC 3986)

RDF : le triplet

2- Exprimez des relations entre ces ressources sous la forme de triplets ou comment décrire l'information à son niveau de granularité le plus basique, la donnée ?

Structure d'un triplet RDF = Structure d'une phrase simple

Sujet Verbe Complément = (Sujet, Prédicat, Objet)

L'article « Semantic Web » de 2001 est un texte

(<http://www.sciam.com/article.cfm?id=the-semantic-web>, <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>, <http://purl.org/dc/dcmitype/Text>)

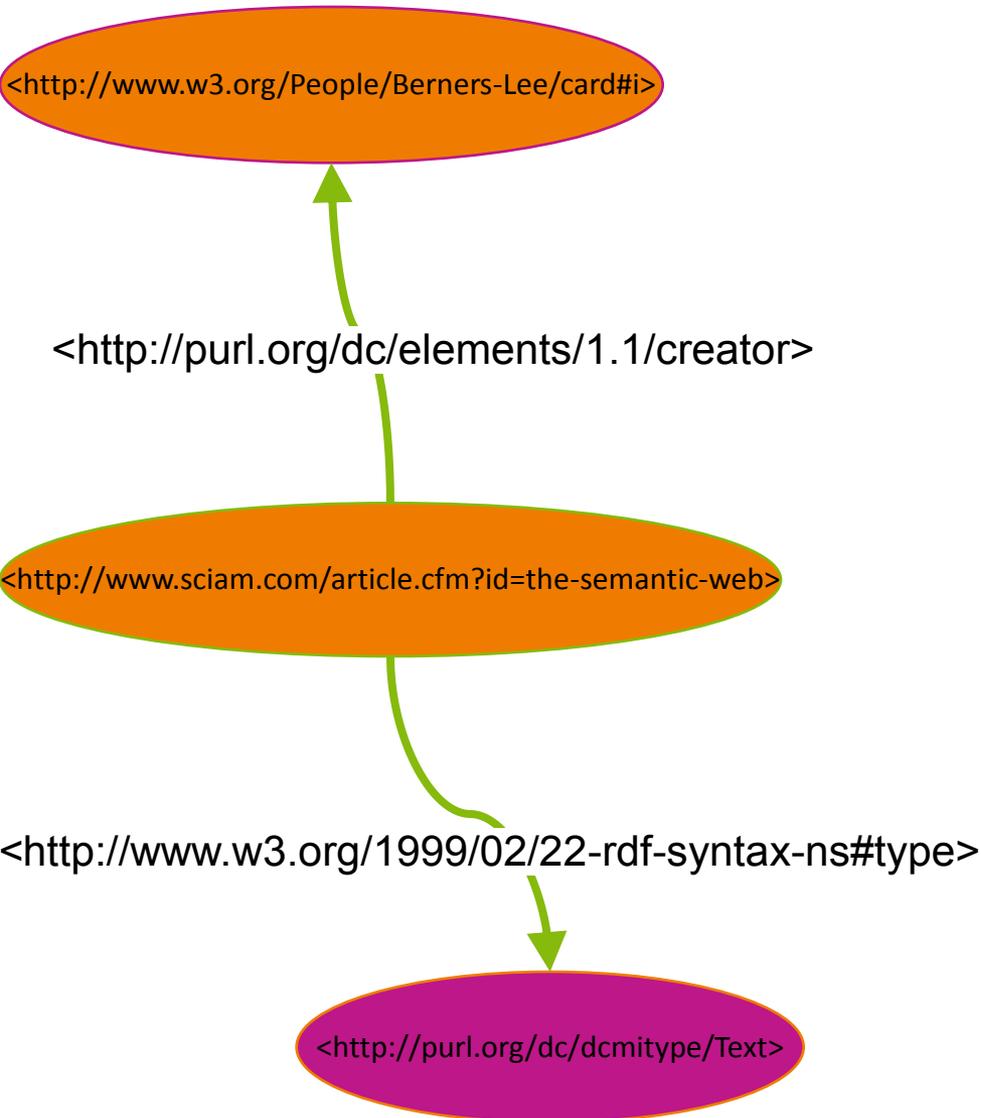
L'article Semantic Web de 2001 a pour créateur Tim Berners-Lee

(<http://www.sciam.com/article.cfm?id=the-semantic-web>,
<http://purl.org/dc/elements/1.1/creator>, <http://www.w3.org/People/Berners-Lee/card#i>)

Tim Berners-Lee est une personne

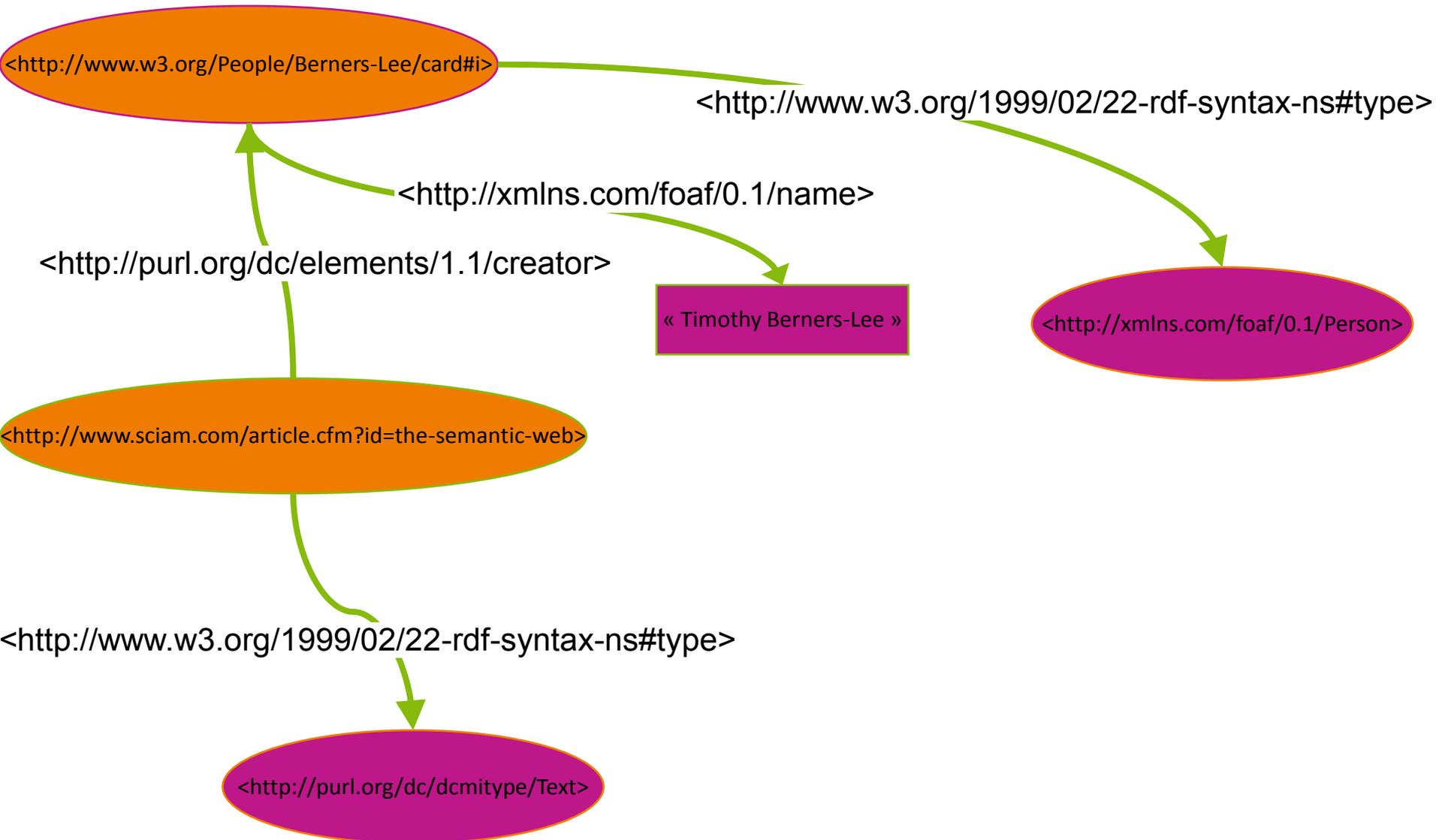
(<http://www.w3.org/People/Berners-Lee/card#i>, <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>, <http://xmlns.com/foaf/0.1/Person>)

3- Représentez et reliez les triplets sous la forme de graphes orientés



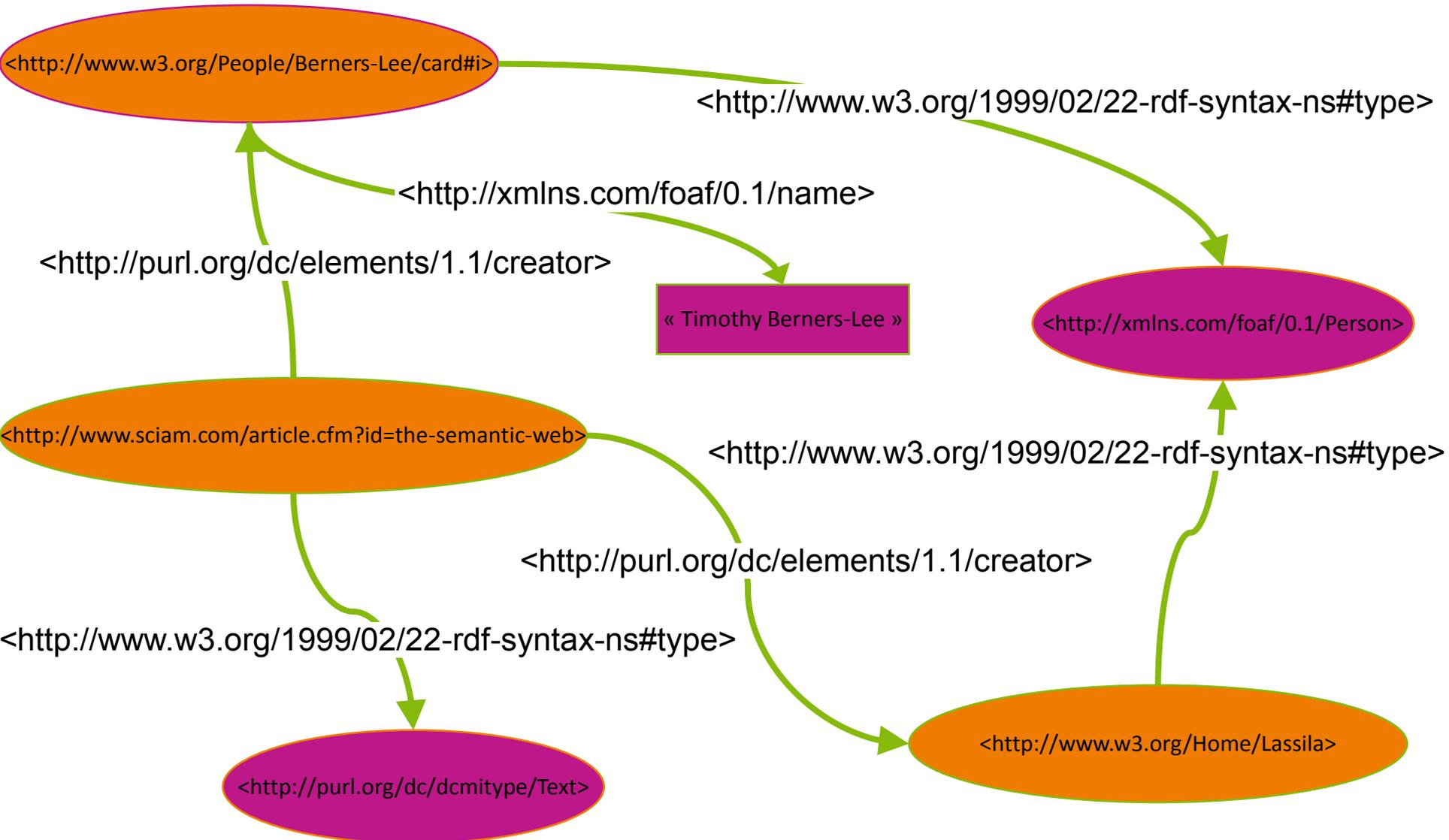
RDF : le graphe orienté

3- Représentez et reliez les triplets sous la forme de graphes orientés



RDF : le graphe orienté

3- Représentez et reliez les triplets sous la forme de graphes orientés

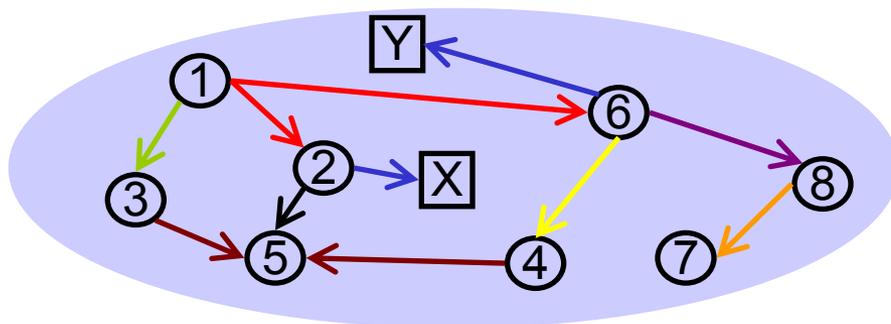


SPARQL PRINCIPES GÉNÉRAUX

Le parcours de graphe pas à pas : le graphe

SPARQL permet d'extraire des informations d'un graphe par l'expression de contraintes sous la forme d'équations qui permettent de parcourir le graphe interrogé.

Soit le graphe suivant sous la forme d'une représentation graphique :

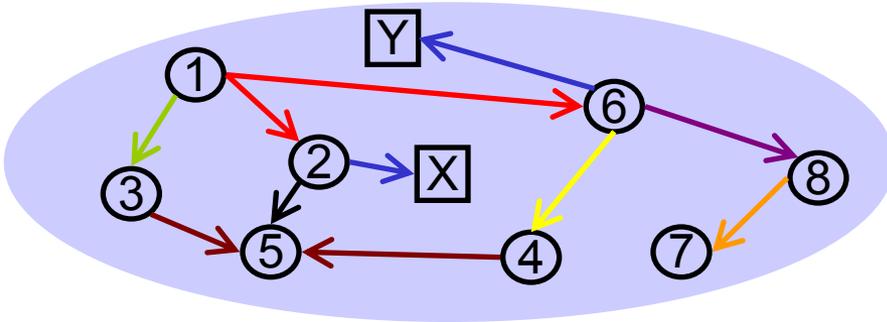


Le même graphe sérialisé en Turtle :

```
<1> <rouge> <2>.
<1> <rouge> <6>.
<1> <vert> <3>.
<2> <bleu> "X".
<2> <noir> <5>.
<3> <marron> <5>.
<4> <marron> <5>.
<4> <j jaune> <6>.
<5> <noir> <4>.
<6> <violet> <8>.
<6> <bleu> "Y".
<7> <orange> <8>.
<8> <orange> <7>.
```

Le parcours de graphe pas à pas

Rechercher un prédicat



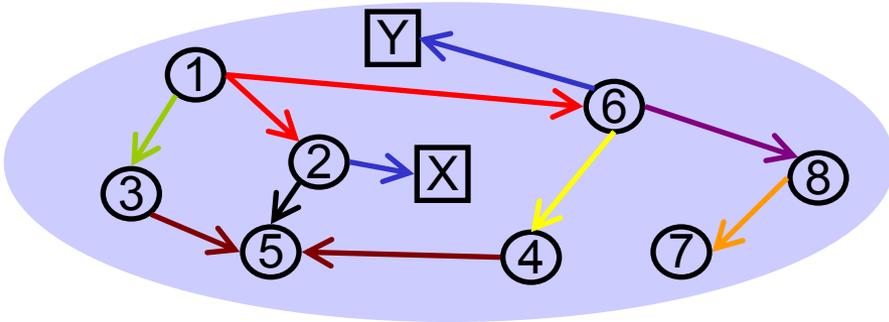
<1> <rouge> <2>.
<1> <rouge> <6>.
<1> <vert> <3>.
<2> <bleu> "X".
<2> <noir> <5>.
<6> <bleu> "Y".

<6> <violet> <8>.
<8> <orange> <7>.
<6> <jaune> <4>.
<4> <marron> <5>.
<3> <marron> <5>.

Comment atteindre les ressources <2> et <6> à partir de <1> ?

Le parcours de graphe pas à pas

Rechercher un prédicat



<1> <rouge> <2>.
<1> <rouge> <6>.
<1> <vert> <3>.
<2> <bleu> "X".
<2> <noir> <5>.
<6> <bleu> "Y".

<6> <violet> <8>.
<8> <orange> <7>.
<6> <jaune> <4>.
<4> <marron> <5>.
<3> <marron> <5>.

Comment atteindre les ressources <2> et <6> à partir de <1> ?

Reformuler la question

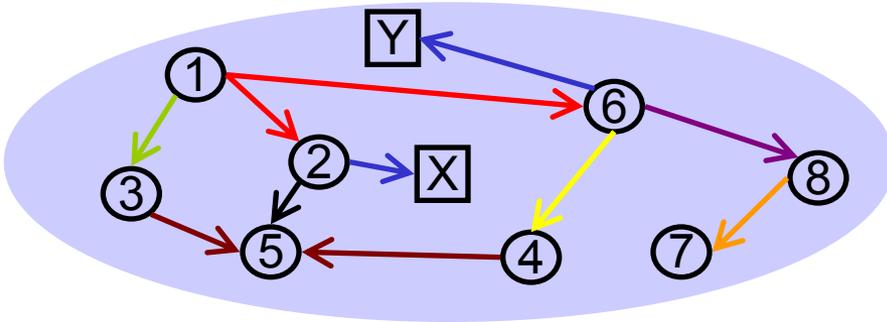
Sujet = <1>

Prédicat = ?

Objet = <2> et <6>

Le parcours de graphe pas à pas

Rechercher un prédicat



<1> <rouge> <2>.
<1> <rouge> <6>.
<1> <vert> <3>.
<2> <bleu> "X".
<2> <noir> <5>.
<6> <bleu> "Y".

<6> <violet> <8>.
<8> <orange> <7>.
<6> <jaune> <4>.
<4> <marron> <5>.
<3> <marron> <5>.

Comment atteindre les ressources <2> et <6> à partir de <1> ?

Reformuler la question

Sujet = <1>
Prédicat = ?
Objet = <2> et <6>

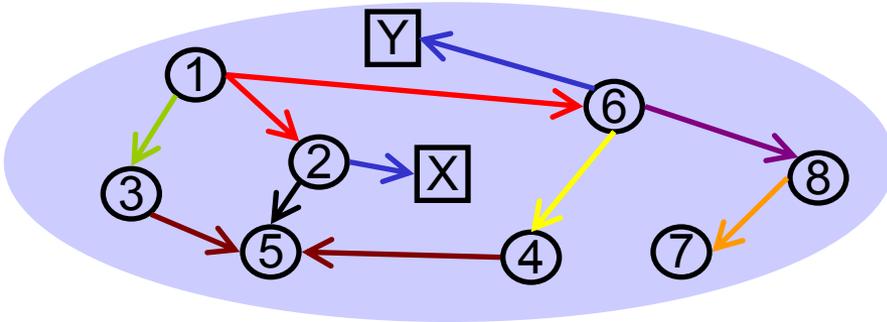


Sous forme de triplets

<1> ? <2>.
<1> ? <6>.

Le parcours de graphe pas à pas

Rechercher un prédicat



<1> <rouge> <2>.
<1> <rouge> <6>.
<1> <vert> <3>.
<2> <bleu> "X".
<2> <noir> <5>.
<6> <bleu> "Y".

<6> <violet> <8>.
<8> <orange> <7>.
<6> <jaune> <4>.
<4> <marron> <5>.
<3> <marron> <5>.

Comment atteindre les ressources <2> et <6> à partir de <1> ?

Reformuler la question

Sujet = <1>
Prédicat = ?
Objet = <2> et <6>

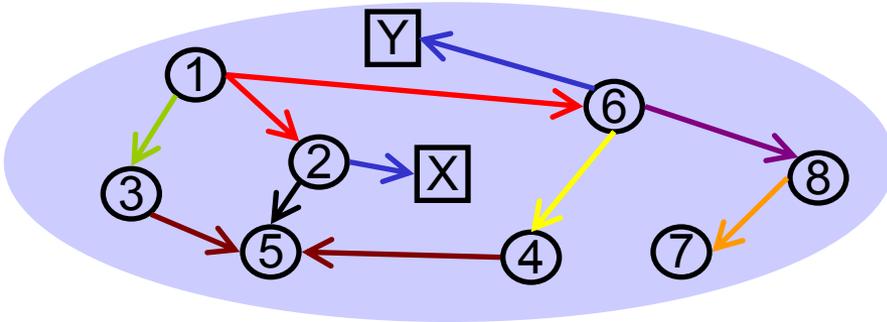


Sous forme de triplets

<1> ?predicat <2>.
<1> ?predicat <6>.

Le parcours de graphe pas à pas

Rechercher un prédicat



<1> <rouge> <2>.
<1> <rouge> <6>.
<1> <vert> <3>.
<2> <bleu> "X".
<2> <noir> <5>.
<6> <bleu> "Y".

<6> <violet> <8>.
<8> <orange> <7>.
<6> <jaune> <4>.
<4> <marron> <5>.
<3> <marron> <5>.

Comment atteindre les ressources <2> et <6> à partir de <1> ?

Reformuler la question

Sujet = <1>
Prédicat = ?
Objet = <2> et <6>



Sous forme de triplets

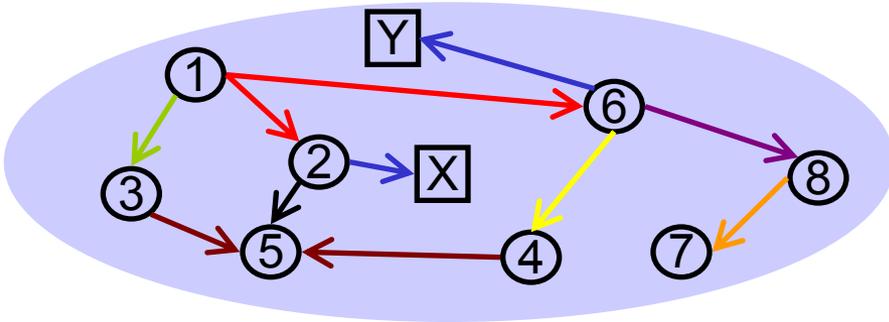
<1> ?predicat <2>.
<1> ?predicat <6>.

Réponse

?predicat
<rouge>

Le parcours de graphe pas à pas

Rechercher un objet



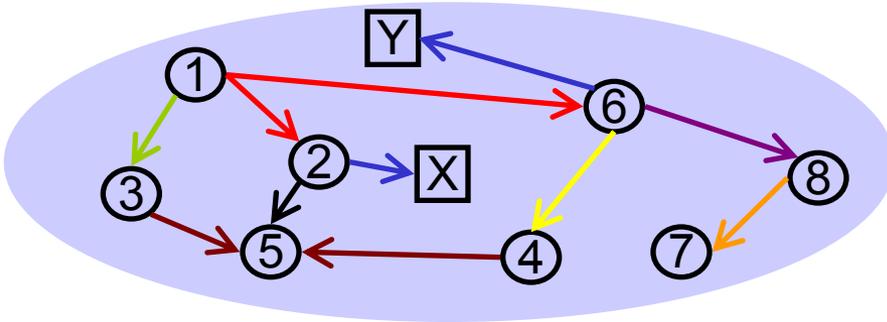
<1> <rouge> <2>.
<1> <rouge> <6>.
<1> <vert> <3>.
<2> <bleu> "X".
<2> <noir> <5>.
<6> <bleu> "Y".

<6> <violet> <8>.
<8> <orange> <7>.
<6> <j jaune> <4>.
<4> <marron> <5>.
<3> <marron> <5>.

Quelle est la ressources liée en tant qu'objet à la ressource <6> par le prédicat <violet> ?

Le parcours de graphe pas à pas

Rechercher un objet



<1> <rouge> <2>.
<1> <rouge> <6>.
<1> <vert> <3>.
<2> <bleu> "X".
<2> <noir> <5>.
<6> <bleu> "Y".

<6> <violet> <8>.
<8> <orange> <7>.
<6> <jaune> <4>.
<4> <marron> <5>.
<3> <marron> <5>.

Quelle est la ressource liée en tant qu'objet à la ressource <6> par le prédicat <violet> ?

Reformuler la question

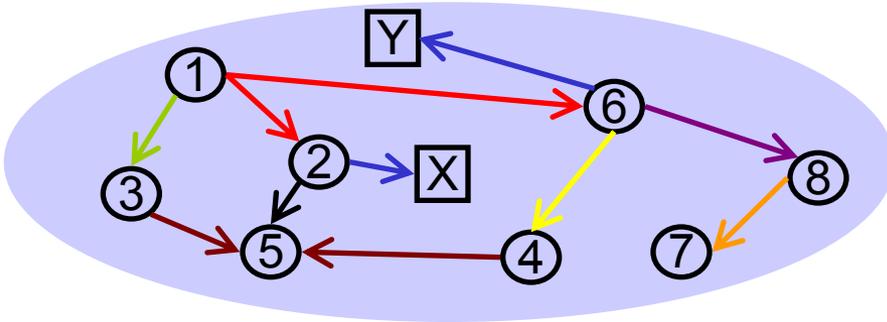
Sujet = <6>

Prédicat = <violet>

Objet = ?

Le parcours de graphe pas à pas

Rechercher un objet



<1> <rouge> <2>.
<1> <rouge> <6>.
<1> <vert> <3>.
<2> <bleu> "X".
<2> <noir> <5>.
<6> <bleu> "Y".

<6> <violet> <8>.
<8> <orange> <7>.
<6> <jaune> <4>.
<4> <marron> <5>.
<3> <marron> <5>.

Quelle est la ressources liée en tant qu'objet à la ressource <6> par le prédicat <violet> ?

Reformuler la question

Sujet = <6>
Prédicat = <violet>
Objet = ?

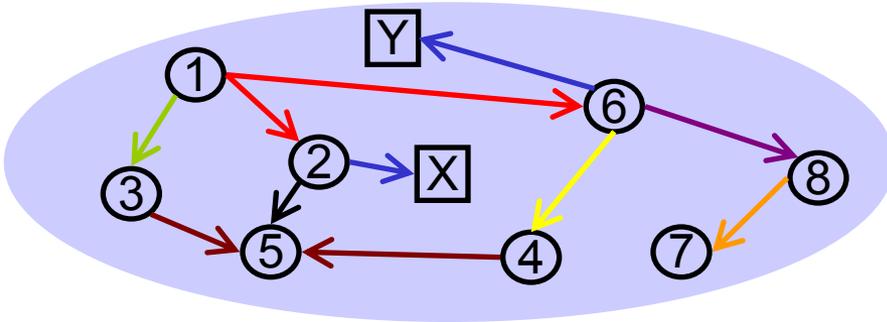


Sous forme de triplets

<6> <violet> ?objet.

Le parcours de graphe pas à pas

Rechercher un objet



<1> <rouge> <2>.
<1> <rouge> <6>.
<1> <vert> <3>.
<2> <bleu> "X".
<2> <noir> <5>.
<6> <bleu> "Y".

<6> <violet> <8>.
<8> <orange> <7>.
<6> <jaune> <4>.
<4> <marron> <5>.
<3> <marron> <5>.

Quelle est la ressource liée en tant qu'objet à la ressource <6> par le prédicat <violet> ?

Reformuler la question

Sujet = <6>
Prédicat = <violet>
Objet = ?



Sous forme de triplets

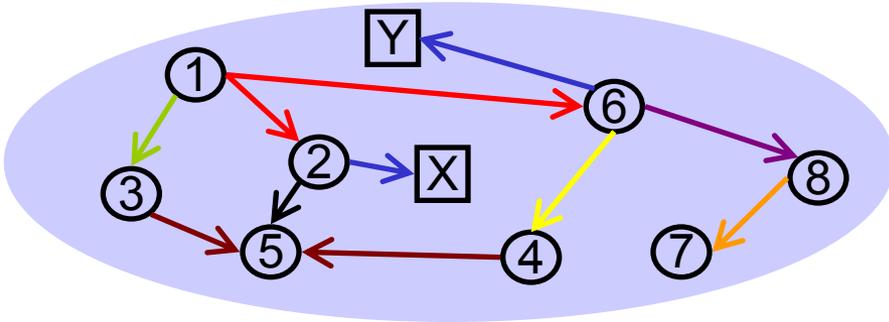
<6> <violet> ?objet.

Réponse

?objet
<8>

Le parcours de graphe pas à pas

Rechercher un sujet



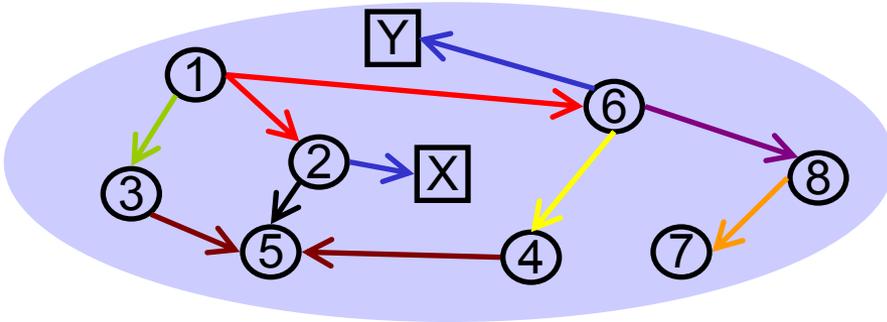
<1> <rouge> <2>.
<1> <rouge> <6>.
<1> <vert> <3>.
<2> <bleu> "X".
<2> <noir> <5>.
<6> <bleu> "Y".

<6> <violet> <8>.
<8> <orange> <7>.
<6> <jaune> <4>.
<4> <marron> <5>.
<3> <marron> <5>.

Quelle est la ressource sujet dont le prédicat est <orange> et l'objet <7> ?

Le parcours de graphe pas à pas

Rechercher un sujet



<1> <rouge> <2>.
<1> <rouge> <6>.
<1> <vert> <3>.
<2> <bleu> "X".
<2> <noir> <5>.
<6> <bleu> "Y".

<6> <violet> <8>.
<8> <orange> <7>.
<6> <jaune> <4>.
<4> <marron> <5>.
<3> <marron> <5>.

Quelle est la ressource sujet dont le prédicat est <orange> et l'objet <7> ?

Reformuler la question

Sujet = ?

Prédicat = <orange>

Objet = <7>

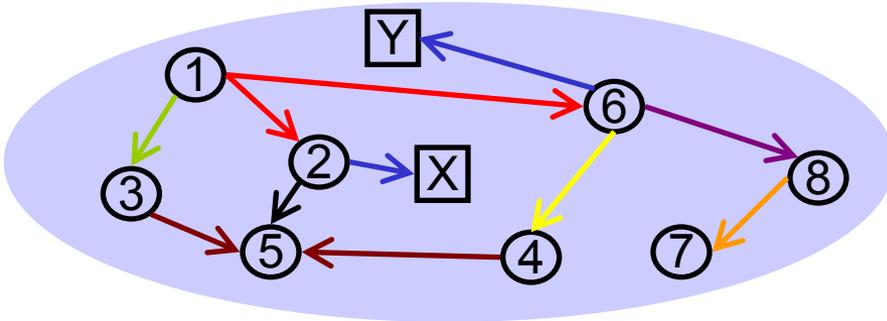


Sous forme de triplets

?sujet <orange> <7>.

Le parcours de graphe pas à pas

Rechercher un sujet



<1> <rouge> <2>.
<1> <rouge> <6>.
<1> <vert> <3>.
<2> <bleu> "X".
<2> <noir> <5>.
<6> <bleu> "Y".

<6> <violet> <8>.
<8> <orange> <7>.
<6> <jaune> <4>.
<4> <marron> <5>.
<3> <marron> <5>.

Quelle est la ressource sujet dont le prédicat est <orange> et l'objet <7> ?

Reformuler la question

Sujet = ?
Prédicat = <orange>
Objet = <7>



Sous forme de triplets

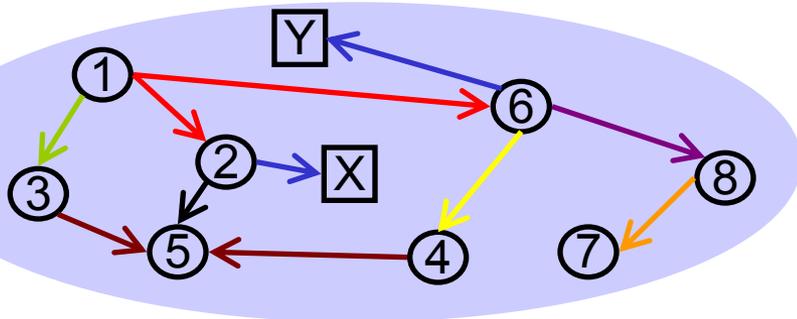
?sujet <orange> <7>.

Réponse

?sujet
<8>

Le parcours de graphe pas à pas

Rechercher des sujets et des objets



<1> <rouge> <2>.

<1> <rouge> <6>.

<1> <vert> <3>.

<2> <bleu> "X".

<2> <noir> <5>.

<6> <bleu> "Y".

<6> <violet> <8>.

<8> <orange> <7>.

<6> <j jaune> <4>.

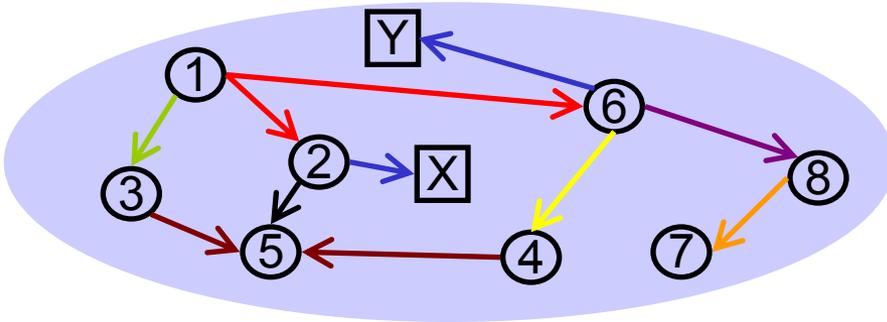
<4> <marron> <5>.

<3> <marron> <5>.

Quelles sont les littéraux liés par le prédicat <bleu> aux ressources liés par le prédicat <rouge> à la ressource <1> ?

Le parcours de graphe pas à pas

Rechercher des sujets et des objets



<1> <rouge> <2>.
<1> <rouge> <6>.
<1> <vert> <3>.
<2> <bleu> "X".
<2> <noir> <5>.
<6> <bleu> "Y".

<6> <violet> <8>.
<8> <orange> <7>.
<6> <jaune> <4>.
<4> <marron> <5>.
<3> <marron> <5>.

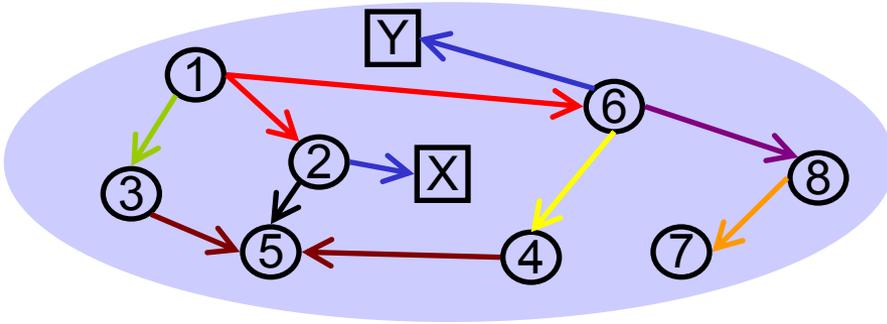
Quelles sont les littéraux liés par le prédicat <bleu> aux ressources liés par le prédicat <rouge> à la ressource <1> ?

Reformuler la question

Sujet = <1>	Sujet = ?ressources
Prédicat = <rouge>	Prédicat = <bleu>
Objet = ?ressources	Objet = ?littéraux

Le parcours de graphe pas à pas

Rechercher des sujets et des objets



<1> <rouge> <2>.
<1> <rouge> <6>.
<1> <vert> <3>.
<2> <bleu> "X".
<2> <noir> <5>.
<6> <bleu> "Y".

<6> <violet> <8>.
<8> <orange> <7>.
<6> <jaune> <4>.
<4> <marron> <5>.
<3> <marron> <5>.

Quelles sont les littéraux liés par le prédicat <bleu> aux ressources liés par le prédicat <rouge> à la ressource <1> ?

Reformuler la question

Sujet = <1>	Sujet = ?ressources
Prédicat = <rouge>	Prédicat = <bleu>
Objet = ?ressources	Objet = ?littéraux

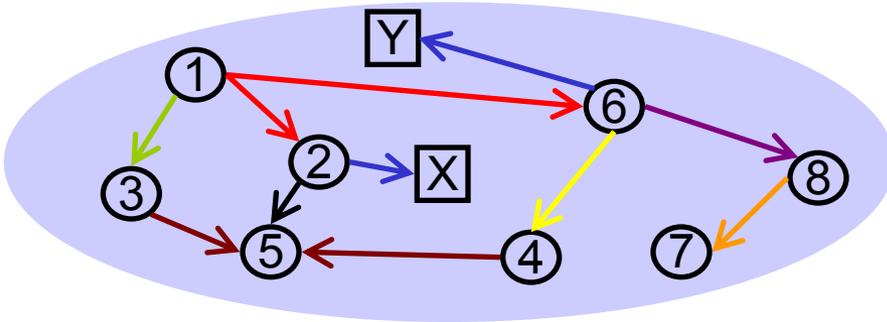


Sous forme de triplets

<1> <rouge> ?ressources.
?ressources <bleu> ?littéraux

Le parcours de graphe pas à pas

Rechercher des sujets et des objets



<1> <rouge> <2>.
<1> <rouge> <6>.
<1> <vert> <3>.
<2> <bleu> "X".
<2> <noir> <5>.
<6> <bleu> "Y".

<6> <violet> <8>.
<8> <orange> <7>.
<6> <jaune> <4>.
<4> <marron> <5>.
<3> <marron> <5>.

Quelles sont les littéraux liés par le prédicat <bleu> aux ressources liés par le prédicat <rouge> à la ressource <1> ?

Reformuler la question

Sujet = <1> Sujet = ?ressources
Prédicat = <rouge> Prédicat = <bleu>
Objet = ?ressources Objet = ?littéraux



Sous forme de triplets

<1> <rouge> ?ressources.
?ressources <bleu> ?littéraux

Réponse

?ressources	?littéraux
<2>	"X"
<6>	"Y"

Construction d'une requête SPARQL

Une question et son équivalence sous forme de phrases simples avec des inconnus.

Je veux connaître l'URI et les noms des personnes que connaît Tim Berners-Lee à partir de son fichier FOAF

Tim Berners-Lee connaît des personnes.

Ces personnes s'appellent ???.

Construction d'une requête SPARQL

Les phrases simples sont transformés en triplet RDF avec la syntaxe turtle.

Je veux connaître l'URI et les noms des personnes que connaît Tim Berners-Lee
à partir de son fichier FOAF

```
<http://www.w3.org/People/Berners-Lee/card#i> foaf:knows ?personnes.  
?personnes foaf:name ?nom.
```

Ajout de l'URI du graphe sur lequel porte la recherche

Je veux connaître l'URI et les noms des personnes que connaît Tim Berners-Lee

FROM <http://www.w3.org/People/Berners-Lee/card>

```
<http://www.w3.org/People/Berners-Lee/card#i> foaf:knows ?personnes.  
?personnes foaf:name ?nom.
```

Construction d'une requête SPARQL

Ajout du mot-clé SELECT et précision des variables recherchées

```
SELECT ?personnes ?nom
```

```
FROM <http://www.w3.org/People/Berners-Lee/card>
```

```
<http://www.w3.org/People/Berners-Lee/card#i> foaf:knows ?personnes.
```

```
?personnes foaf:name ?nom.
```

Ajout des éléments obligatoires de la syntaxe SPARQL

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?personnes ?nom
FROM <http://www.w3.org/People/Berners-Lee/card>
WHERE {
    <http://www.w3.org/People/Berners-Lee/card#i> foaf:knows ?personnes.
    ?personnes foaf:name ?nom.
}
```

Description de la syntaxe SPARQL

```
PREFIX dbpedia: <http://dbpedia.org/resource/>  
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

Déclaration des préfixes
et des espaces de nom



```
SELECT ?oeuvre
```

```
WHERE {
```

```
  ?oeuvre <http://dbpedia.org/ontology/author> dbpedia:Victor_Hugo.
```

```
  ?oeuvre <http://www.w3.org/2000/01/rdf-schema#label> "Les Misérables"@en.
```

```
}
```

Description de la syntaxe SPARQL

```
PREFIX dbpedia:
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

Déclaration du type de requête

SELECT ?oeuvre

```
WHERE {
  ?oeuvre <http://dbpedia.org/ontology/author> dbpedia:Victor_Hugo.
  ?oeuvre <http://www.w3.org/2000/01/rdf-schema#label> "Les Misérables"@en.
}
```

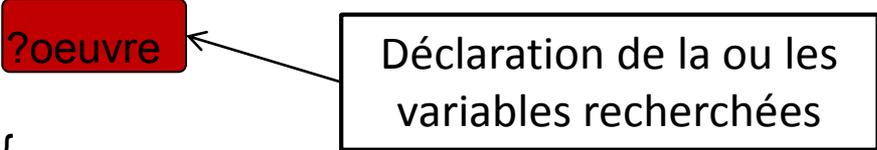
Description de la syntaxe SPARQL

```
PREFIX dbpedia: <http://dbpedia.org/resource/>
```

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
SELECT ?oeuvre
```

Déclaration de la ou les
variables recherchées



```
WHERE {
```

```
  ?oeuvre <http://dbpedia.org/ontology/author> dbpedia:Victor_Hugo.
```

```
  ?oeuvre <http://www.w3.org/2000/01/rdf-schema#label> "Les Misérables"@en.
```

```
}
```

Description de la syntaxe SPARQL

```
PREFIX dbpedia: <http://dbpedia.org/resource/>
```

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
SELECT ?oeuvre
```

Déclaration des
contraintes



```
WHERE {  
  ?oeuvre <http://dbpedia.org/ontology/author> dbpedia:Victor_Hugo.  
  
  ?oeuvre <http://www.w3.org/2000/01/rdf-schema#label> "Les Misérables"@en.  
}
```

Description de la syntaxe SPARQL

```
PREFIX dbpedia: <http://dbpedia.org/resource/>
```

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
SELECT ?oeuvre
```

```
WHERE {
```

```
?oeuvre <http://dbpedia.org/ontology/author> dbpedia:Victor_Hugo;
```

```
?oeuvre <http://www.w3.org/2000/01/rdf-schema#label> "Le Misérables"@fr;
```

```
}
```

Déclaration d'un triplet
on note l'utilisation des chevrons si l'URI
est exprimé complètement

Description de la syntaxe SPARQL

```
PREFIX dbpedia: <http://dbpedia.org/resource/>
```

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
SELECT ?oeuvre
```

```
WHERE {
```

```
  ?oeuvre <http://dbpedia.org/ontology/author> dbpedia:Victor_Hugo.
```

```
  ?oeuvre <http://www.w3.org/2000/01/rdf-schema#label> "Les Misérables"@en.
```

```
}
```

Déclaration du littéral
entre guillemets

Le format XML pour les résultats

```
<?xml version="1.0" ?>
<sparql
  xmlns="http://www.w3.org/2005/sparql-results#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/2001/sw/DataAccess/rf1/result2.xsd">
```

<!-- En-tête -->

```
<head>
<variable name="personnes"/>
<variable name="nom"/>
</head>
```

Rappel des variables déclarées

<!-- Résultats -->

```
<results distinct="false" ordered="true">
```

```
<result>
  <binding name="personnes"><uri>http://dbpedia.org/resource/John_Seely_Brown</uri></binding>
  <binding name="nom"><literal>John Seely Brown</literal></binding>
</result>
```

```
<result>
  <binding name="personnes"><uri>http://dbpedia.org/resource/John_Gage</uri></binding>
  <binding name="nom"><literal>John Gage</literal></binding>
</result>
```

```
</results>
```

```
</sparql>
```

Le format XML pour les résultats

```
<?xml version="1.0" ?>
<sparql
  xmlns="http://www.w3.org/2005/sparql-results#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/2001/sw/DataAccess/rf1/result2.xsd">
```

<!-- En-tête -->

```
<head>
  <variable name="personnes"/>
  <variable name="nom"/>
</head>
```

<!-- Résultats -->

```
<results distinct="false" ordered="true">
```

```
<result>
  <binding name="personnes"><uri>http://dbpedia.org/resource/John_Seely_Brown</uri></binding>
  <binding name="nom"><literal>John Seely Brown</literal></binding>
</result>
```

```
<result>
  <binding name="personnes"><uri>http://db
  <binding name="nom"><literal>John Gage<
</result>
```

Un élément « result » par combinaisons possibles de variables

```
</results>
```

```
</sparql>
```

Le format JSON pour les résultats

Le format JSON en SPARQL 1.0 n'est pas une recommandation mais une note de travail du W3C.

```
{ "head": { "link": [], "vars": ["oeuvre"] },  
  "results": { "distinct": false, "ordered": true, "bindings": [  
    { "oeuvre": { "type": "uri", "value":  
      "http://dbpedia.org/resource/Les_Mis%C3%A9rables" }} ] } }
```

SPARQL PRATIQUE

Exercice : effectuer des requêtes SPARQL sur dbpedia

Deux interfaces Web sont disponibles pour effectuer des requêtes

- Interface classique : <http://dbpedia.org/sparql>
- Interface améliorée : <http://dbpedia.org/snorql>

N'oubliez pas d'indiquer la syntaxe de sortie HTML pour visualiser directement les résultats.

Découvrir les données d'un sparql endpoint pas à pas

Découvrir les types de ressources présentes dans l'entrepôt :

```
select distinct ?Concept where
{
  ?s a ?Concept
}
```

Le mot « a » en position de prédicat correspond à rdf:type

Explorer les prédicats distincts liés à une ressource d'un type

```
select distinct ?predicat where
{
  ?sujet a <http://xmlns.com/foaf/0.1/Person>.
  ?sujet ?predicat ?objet.
}
```

Rechercher les ressources de ce type en associant un des prédicats trouvés

```
select distinct ?sujet ?objet
where
{
  ?sujet a <http://xmlns.com/foaf/0.1/Person>.
  ?sujet <http://www.w3.org/2000/01/rdf-schema#label> ?objet.
}
```

Rechercher tous les triplets dont une ressource particulière est sujet

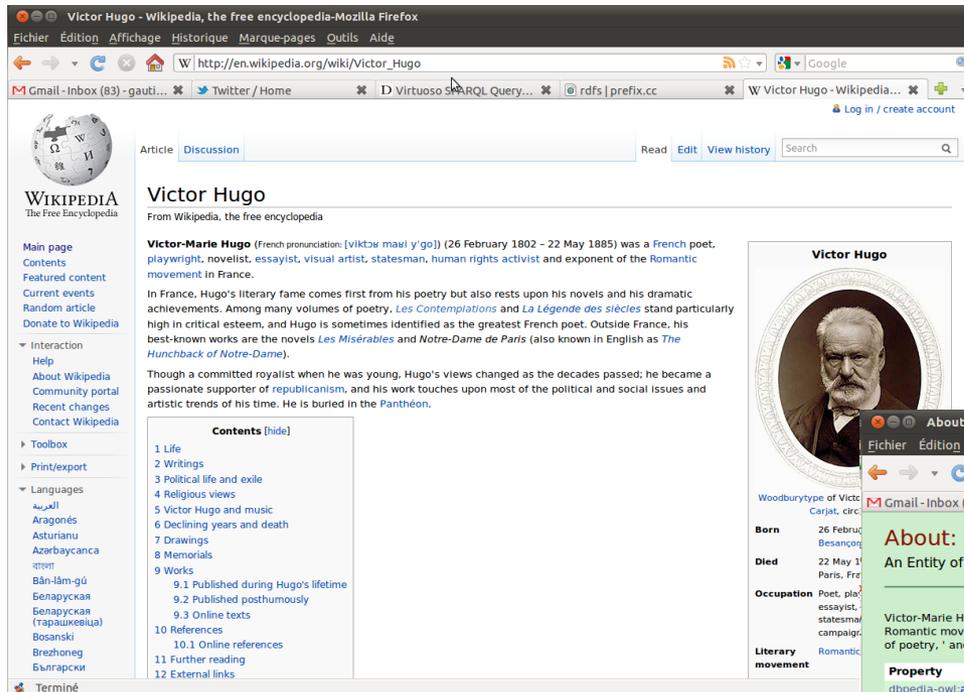
```
select distinct ?predicat ?objet
where
{
  <http://dbpedia.org/resource/Bill_Clinton> ?predicat ?objet.
}
```

Complicquer la requête pour commencer à parcourir le graphe

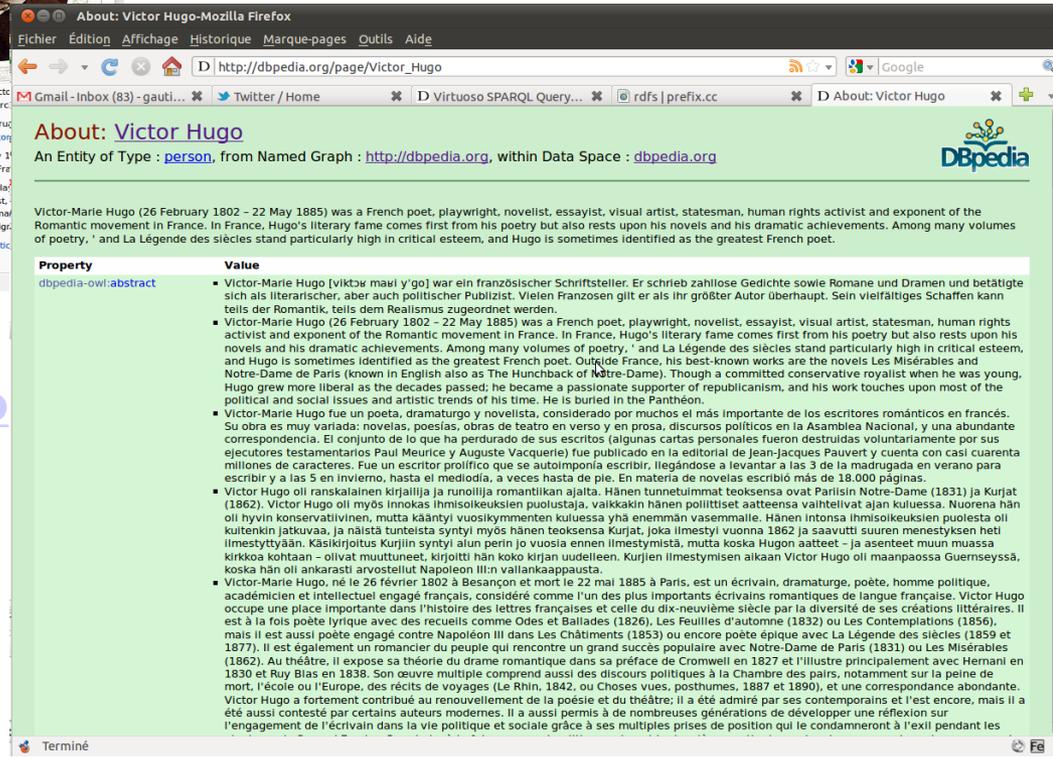
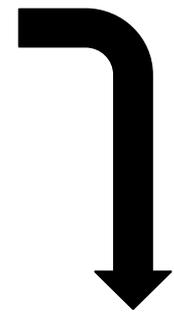
```
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix dbpr: <http://dbpedia.org/property/>
select distinct ?label
where
{
  <http://dbpedia.org/resource/Bill_Clinton> dbpr:placeOfBirth ?objet.
  ?objet rdfs:label ?label
}
```

Trouver une URI dans Dbpedia

http://en.wikipedia.org/wiki/Victor_Hugo



The screenshot shows the Wikipedia article for Victor Hugo. The browser address bar displays the URL http://en.wikipedia.org/wiki/Victor_Hugo. The article text includes: "Victor-Marie Hugo (French pronunciation: [viktɔʁ mɑʁi yˈɡo]) (26 February 1802 – 22 May 1885) was a French poet, playwright, novelist, essayist, visual artist, statesman, human rights activist and exponent of the Romantic movement in France. In France, Hugo's literary fame comes first from his poetry but also rests upon his novels and his dramatic achievements. Among many volumes of poetry, *Les Contemplations* and *La Légende des siècles* stand particularly high in critical esteem, and Hugo is sometimes identified as the greatest French poet. Outside France, his best-known works are the novels *Les Misérables* and *Notre-Dame de Paris* (also known in English as *The Hunchback of Notre-Dame*). Though a committed royalist when he was young, Hugo's views changed as the decades passed; he became a passionate supporter of republicanism, and his work touches upon most of the political and social issues and artistic trends of his time. He is buried in the Panthéon.



The screenshot shows the Dbpedia page for Victor Hugo. The browser address bar displays the URL http://dbpedia.org/page/Victor_Hugo. The page title is "About: Victor Hugo" and it identifies the entity as a person. The main text states: "Victor-Marie Hugo (26 February 1802 – 22 May 1885) was a French poet, playwright, novelist, essayist, visual artist, statesman, human rights activist and exponent of the Romantic movement in France. In France, Hugo's literary fame comes first from his poetry but also rests upon his novels and his dramatic achievements. Among many volumes of poetry, ' and La Légende des siècles stand particularly high in critical esteem, and Hugo is sometimes identified as the greatest French poet."

Property	Value
<code>dbpedia-owl:abstract</code>	<ul style="list-style-type: none">Victor-Marie Hugo [viktɔʁ mɑʁi yˈɡo] war ein französischer Schriftsteller. Er schrieb zahllose Gedichte sowie Romane und Dramen und betätigte sich als literarischer, aber auch politischer Publizist. Vielen Franzosen gilt er als ihr größter Autor überhaupt. Sein vielfältiges Schaffen kann teils der Romantik, teils dem Realismus zugeordnet werden.Victor-Marie Hugo (26 February 1802 – 22 May 1885) was a French poet, playwright, novelist, essayist, visual artist, statesman, human rights activist and exponent of the Romantic movement in France. In France, Hugo's literary fame comes first from his poetry but also rests upon his novels and his dramatic achievements. Among many volumes of poetry, ' and La Légende des siècles stand particularly high in critical esteem, and Hugo is sometimes identified as the greatest French poet. Outside France, his best-known works are the novels <i>Les Misérables</i> and <i>Notre-Dame de Paris</i> (known in English also as <i>The Hunchback of Notre-Dame</i>). Though a committed conservative royalist when he was young, Hugo grew more liberal as the decades passed; he became a passionate supporter of republicanism, and his work touches upon most of the political and social issues and artistic trends of his time. He is buried in the Panthéon.Victor-Marie Hugo fue un poeta, dramaturgo y novelista, considerado por muchos el más importante de los escritores románticos en francés. Su obra es muy variada: novelas, poesías, obras de teatro en verso y en prosa, discursos políticos en la Asamblea Nacional, y una abundante correspondencia. El conjunto de lo que ha perdurado de sus escritos (algunas cartas personales fueron destruidas voluntariamente por sus ejecutores testamentarios Paul Meurice y Auguste Vacquerie) fue publicado en la editorial de Jean-Jacques Pauvert y cuenta con casi cuarenta millones de caracteres. Fue un escritor prolífico que se autoimponía escribir, ilegándose a levantar a las 3 de la madrugada en verso para escribir y a las 5 en invierno, hasta el mediodía, a veces hasta de pie. En materia de novelas escribió más de 18.000 páginas.Victor Hugo oli ranskalainen kirjailija ja runoilija romantiikan ajalta. Hänen tunnetuimmat teoksensa ovat Pariisin Notre-Dame (1831) ja Kurjat (1862). Victor Hugo oli myös innokas ihmisoikeuksien puolustaja, vaikkakin hänen poliittiset aatteensa vaihtelivat ajan kuluessa. Nuorena hän oli hyvin konservatiivinen, mutta kääntyi vuosikymmenten kuluessa yhä enemmän vasemmalle. Hänen intonsa ihmisoikeuksien puolesta oli kuitenkin jatkuvaa, ja näistä tunteista syntyi myös hänen teoksensa Kurjat, joka ilmestyi vuonna 1862 ja saavutti suuren menestyksen heti ilmestyttyään. Käsitkirjoitus Kurjien syntyi alun perin jo vuosia ennen ilmestymistä, mutta koska Hugon aatteet – ja asenteet muun muassa kirkkoa kohtaan – olivat muuttuneet, kirjoitti hän koko kirjan uudelleen. Kurjien ilmestymisen aikaan Victor Hugo oli maanpaossa Guernerseysä, koska hän oli ankarasti arvostellut Napoleon III:n vallankaappausa.Victor-Marie Hugo, né le 26 février 1802 à Besançon et mort le 22 mai 1885 à Paris, est un écrivain, dramaturge, poète, homme politique, académicien et intellectuel engagé français, considéré comme l'un des plus importants écrivains romantiques de langue française. Victor Hugo occupe une place importante dans l'histoire des lettres françaises et celle du dix-neuvième siècle par la diversité de ses créations littéraires. Il est à la fois poète lyrique avec des recueils comme <i>Odes</i> et <i>Ballades</i> (1826), <i>Les Feuilles d'automne</i> (1832) ou <i>Les Contemplations</i> (1856), mais il est aussi poète engagé contre Napoléon III dans <i>Les Châtiments</i> (1833) ou encore poète épique avec <i>La Légende des siècles</i> (1859 et 1877). Il est également un romancier du peuple qui rencontre un grand succès populaire avec <i>Notre-Dame de Paris</i> (1831) ou <i>Les Misérables</i> (1862). Au théâtre, il expose sa théorie du drame romantique dans sa préface de <i>Cromwell</i> en 1827 et l'illustre principalement avec <i>Hernani</i> en 1830 et <i>Ruy Blas</i> en 1838. Son œuvre multiple comprend aussi des discours politiques à la Chambre des pairs, notamment sur la peine de mort, l'école ou l'Europe, des récits de voyages (<i>Le Rhin</i>, 1842, ou <i>Choses vues</i>, posthumes, 1887 et 1890), et une correspondance abondante. Victor Hugo a fortement contribué au renouvellement de la poésie et du théâtre; il a été admiré par ses contemporains et l'est encore, mais il a été aussi contesté par certains auteurs modernes. Il a aussi permis à de nombreuses générations de développer une réflexion sur l'engagement de l'écrivain dans la vie politique et sociale grâce à ses multiples prises de position qui le condamneront à l'exil pendant les

http://dbpedia.org/resource/Victor_Hugo



La redirection du Linked Data → page

Exercice 1

Formuler une requête qui affiche toutes les informations associées à l'URI de Victor Hugo.

URI utile

Victor Hugo : http://dbpedia.org/resource/Victor_Hugo

Exercice 1

Formuler une requête qui affiche toutes les informations associées à l'URI de Victor Hugo.

URI utile

Victor Hugo : http://dbpedia.org/resource/Victor_Hugo

Requête

```
select ?predicat ?objet
where
{
  <http://dbpedia.org/resource/Victor\_Hugo> ?predicat ?objet.
}
```

Exercice 2

Rechercher une ressource ayant pour auteur Victor Hugo et pour type « Livre ».

URI utile

Victor Hugo : http://dbpedia.org/resource/Victor_Hugo

Propriété auteur : <http://dbpedia.org/ontology/author>

Type Livre : <http://dbpedia.org/ontology/Book>

Exercice 2

Rechercher une ressource ayant pour auteur Victor Hugo et pour type « Livre ».

URI utile

Victor Hugo : http://dbpedia.org/resource/Victor_Hugo

Propriété auteur : <http://dbpedia.org/ontology/author>

Type Livre : <http://dbpedia.org/ontology/Book>

Requête

```
select ?livres
where
{
  ?livres <http://dbpedia.org/ontology/author> <http://dbpedia.org/resource/Victor\_Hugo>.
  ?livres a <http://dbpedia.org/ontology/Book>
}
```

Exercice 3

Rechercher des romanciers français nés à Paris.

URI utile

Prédicat « Sujet » : <http://www.w3.org/2004/02/skos/core#subject>

Romanciers français : http://dbpedia.org/resource/Category:French_novelists

Prédicat « Lieu de naissance » : <http://dbpedia.org/ontology/birthPlace>

Paris : <http://dbpedia.org/resource/Paris>

Exercice 3

Rechercher des romanciers français nés à Paris.

URI utile

Prédicat « Sujet » : <http://www.w3.org/2004/02/skos/core#subject>

Romanciers français : http://dbpedia.org/resource/Category:French_novelists

Prédicat « Lieu de naissance » : <http://dbpedia.org/ontology/birthPlace>

Paris : <http://dbpedia.org/resource/Paris>

Requête

```
select ?personnes
where
{
  ?personnes <http://purl.org/dc/terms/subject>
  <http://dbpedia.org/resource/Category:French_novelists>.
  ?personnes <http://dbpedia.org/ontology/birthPlace> <http://dbpedia.org/resource/Paris>
}
```

**SPARQL
ALLER PLUS LOIN**

CONSTRUCT : récupérer ou créer des triplets

Ce type de requêtes permet de récupérer un ensemble de triplets stockés dans le triple store pour les manipuler localement, de « re-modéliser » des nouveaux triplets à partir des triplets existants voire d'appliquer des règles d'inférence.

Deux parties

Construction des triplets cible

Une ressource a pour titre une étiquette

Recherche dans les triplets existants

Une ressource a pour étiquette une étiquette.

CONSTRUCT : récupérer ou créer des triplets

Ce type de requêtes permet de récupérer un ensemble de triplets stockés dans le triple store pour les manipuler localement, de « re-modéliser » des nouveaux triplets à partir des triplets existants voire d'appliquer des règles d'inférence.

Deux parties

Construction des triplets cible

?resource dc:title ?label

Recherche dans les triplets existants

?resource rdfs:label ?label

CONSTRUCT : récupérer ou créer des triplets

Ce type de requêtes permet de récupérer un ensemble de triplets stockés dans le triple store pour les manipuler localement, de « re-modéliser » des nouveaux triplets à partir des triplets existants voire d'appliquer des règles d'inférence.

Deux parties

CONSTRUCT {

?resource **dc:title** **?label**

}

WHERE {

?resource **rdfs:label** **?label**

}

ASK : vérifier la présence d'un graphe

Ce type de requêtes permet de vérifier la présence dans l'ensemble de données RDF interrogé des contraintes exprimées.

Y-a-t'il un triplet dont le prédicat est dc:title ?

Une ressource a pour titre une étiquette

ASK : vérifier la présence d'un graphe

Ce type de requêtes permet de vérifier la présence dans l'ensemble de données RDF interrogé des contraintes exprimées.

Y-a-t'il un triplet dont le prédicat est dc:title ?

?resource dc:title ?label

ASK : vérifier la présence d'un graphe

Ce type de requêtes permet de vérifier la présence dans l'ensemble de données RDF interrogé des contraintes exprimées.

```
ASK {
```

```
  ?resource dc:title ?label
```

```
}
```

Le mot-clé DISTINCT

```
SELECT ?predicat
WHERE {
  <http://dbpedia.org/resource/Victor_Hugo> ?predicat ?objet.
}
```

Autant de résultats que de triplets
dont Victor Hugo est sujet, plusieurs
résultats identiques

Le mot-clé DISTINCT

```
SELECT ?predicat
WHERE {
  <http://dbpedia.org/resource/Victor_Hugo> ?predicat ?objet.
}
```

Autant de résultats que de triplets dont Victor Hugo est sujet, plusieurs résultats identiques

```
SELECT DISTINCT ?predicat
WHERE {
  <http://dbpedia.org/resource/Victor_Hugo> ?predicat ?objet.
}
```

Dédoublonnage automatique des résultats

Le mot-clé DISTINCT

```
SELECT ?predicat
WHERE {
  <http://dbpedia.org/resource/Victor_Hugo> ?predicat ?objet.
}
```

Autant de résultats que de triplets dont Victor Hugo est sujet, plusieurs résultats identiques

```
SELECT DISTINCT ?predicat
WHERE {
  <http://dbpedia.org/resource/Victor_Hugo> ?predicat ?objet.
}
```

Dédoublonnage automatique des résultats

Le mot-clé OPTIONAL

```
PREFIX dbpedia: <http://dbpedia.org/ontology/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/ >
SELECT ?auteur ?depiction WHERE {
  ?auteur a <http://dbpedia.org/class/yago/FrenchNovelists>.
  ?auteur foaf:depiction ?depiction.
  ?auteur dbpedia:birthPlace <http://dbpedia.org/resource/Paris>.
}
```

Résultats que si toutes les contraintes sont vérifiées

Le mot-clé DISTINCT

```
SELECT ?predicat
WHERE {
  <http://dbpedia.org/resource/Victor_Hugo> ?predicat ?objet.
}
```

Autant de résultats que de triplets dont Victor Hugo est sujet, plusieurs résultats identiques

```
SELECT DISTINCT ?predicat
WHERE {
  <http://dbpedia.org/resource/Victor_Hugo> ?predicat ?objet.
}
```

Dédoublonnage automatique des résultats

Le mot-clé OPTIONAL

```
PREFIX dbpedia: <http://dbpedia.org/ontology/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/ >
SELECT ?auteur ?depiction WHERE {
  ?auteur a <http://dbpedia.org/class/yago/FrenchNovelists>.
  ?auteur foaf:depiction ?depiction.
  ?auteur dbpedia:birthPlace <http://dbpedia.org/resource/Paris>.
}
```

Résultats que si toutes les contraintes sont vérifiées

```
SELECT ?auteur ?depiction WHERE {
  ?auteur a <http://dbpedia.org/class/yago/FrenchNovelists>.
  OPTIONAL {?auteur foaf:depiction ?depiction.}
  ?auteur dbpedia:birthPlace <http://dbpedia.org/resource/Paris>.
}
```

Résultats même si la contrainte optionnelle n'est pas vérifiée (mais affichage du résultat si elle existe)

Le mot-clé UNION

```
SELECT ?auteur WHERE {  
  ?auteur a <http://dbpedia.org/class/yago/FrenchNovelists>.  
  {  
    {?auteur <http://dbpedia.org/ontology/birthPlace> <http://dbpedia.org/resource/Paris>.  
    UNION  
    {?auteur <http://dbpedia.org/ontology/birthPlace> <http://dbpedia.org/resource/Besan%C3%A7on>.  
    }  
  }  
}
```

UNION permet d'exprimer l'opérateur booléen « OU » (inclusif)

Le mot-clé UNION

```
SELECT ?auteur WHERE {  
  ?auteur a <http://dbpedia.org/class/yago/FrenchNovelists>.  
  {  
    {?auteur <http://dbpedia.org/ontology/birthPlace> <http://dbpedia.org/resource/Paris>.  
    UNION  
    {?auteur <http://dbpedia.org/ontology/birthPlace> <http://dbpedia.org/resource/Besan%C3%A7on>.  
    }  
  }  
}
```

UNION permet d'exprimer l'opérateur booléen « OU » (inclusif)

Le mot-clé ORDER

```
SELECT ?auteur ?name WHERE {  
  ?auteur a <http://dbpedia.org/class/yago/FrenchNovelists>;  
    foaf:name ?name  
  }  
ORDER BY ?name
```

ORDER BY permet de classer les résultats suivants une variable déclarée

Le mot-clé UNION

```
SELECT ?auteur WHERE {  
  ?auteur a <http://dbpedia.org/class/yago/FrenchNovelists>.  
  {  
    {?auteur <http://dbpedia.org/ontology/birthPlace> <http://dbpedia.org/resource/Paris>.  
    UNION  
    {?auteur <http://dbpedia.org/ontology/birthPlace> <http://dbpedia.org/resource/Besan%C3%A7on>.  
    }  
  }  
}
```

UNION permet d'exprimer l'opérateur booléen « OU » (inclusif)

Le mot-clé ORDER

```
SELECT ?auteur ?name WHERE {  
  ?auteur a <http://dbpedia.org/class/yago/FrenchNovelists>;  
    foaf:name ?name  
}  
ORDER BY ?name
```

ORDER BY permet de classer les résultats suivants une variable déclarée

Le mot-clé LIMIT

```
SELECT ?auteur ?name WHERE {  
  ?auteur a <http://dbpedia.org/class/yago/FrenchNovelists>;  
    foaf:name ?name  
}  
ORDER BY ?name  
LIMIT 20
```

LIMIT permet de limiter le nombre de résultats

Il est possible d'appliquer des filtres à une requête SPARQL pour limiter les résultats.

Filtre lang()

```
SELECT ?abstract
WHERE {
  <http://dbpedia.org/resource/Victor_Hugo> <http://dbpedia.org/ontology/abstract> ?abstract.
  FILTER (lang(?abstract)="fr")
}
```

Filtrer selon la langue du littéral

Il est possible d'appliquer des filtres à une requête SPARQL pour limiter les résultats.

Filtre lang()

```
SELECT ?abstract
WHERE {
  <http://dbpedia.org/resource/Victor_Hugo> <http://dbpedia.org/ontology/abstract> ?abstract.
  FILTER (lang(?abstract)="fr")
}
```

Filtrer selon la langue du littéral

Filtre de comparaison

```
SELECT distinct ?auteur ?date WHERE {
  ?auteur a <http://dbpedia.org/class/yago/FrenchNovelists>.
  ?auteur <http://dbpedia.org/ontology/birthDate> ?date.
  FILTER ( xsd:dateTime(?date) > xsd:dateTime("1950-01-01")).
}
```

Filtrer les données avec les signes de comparaison mathématique

Il est possible d'appliquer des filtres à une requête SPARQL pour limiter les résultats.

Filtre lang()

```
SELECT ?abstract
WHERE {
  <http://dbpedia.org/resource/Victor_Hugo> <http://dbpedia.org/ontology/abstract> ?abstract.
  FILTER (lang(?abstract)="fr")
}
```

Filtrer selon la langue du littéral

Filtre de comparaison

```
SELECT distinct ?auteur ?date WHERE {
  ?auteur a <http://dbpedia.org/class/yago/FrenchNovelists>.
  ?auteur <http://dbpedia.org/ontology/birthDate> ?date.
  FILTER ( xsd:dateTime(?date) > xsd:dateTime("1950-01-01")).
}
```

Filtrer les données avec les signes de comparaison mathématique

Filtre de regexp

```
SELECT ?auteur ?name WHERE {
  ?auteur a <http://dbpedia.org/class/yago/FrenchNovelists>.
  ?auteur foaf:name ?name
  FILTER regex(?name, '^Jean').
}
```

Filtrer un littéral en y appliquant une regexp

Il est possible d'appliquer des filtres à une requête SPARQL pour limiter les résultats.

Filtre lang()

```
SELECT ?abstract
WHERE {
  <http://dbpedia.org/resource/Victor_Hugo> <http://dbpedia.org/ontology/abstract> ?abstract.
  FILTER (lang(?abstract)="fr")
}
```

Filtrer selon la langue du littéral

Filtre de comparaison

```
SELECT distinct ?auteur ?date WHERE {
  ?auteur a <http://dbpedia.org/class/yago/FrenchNovelists>.
  ?auteur <http://dbpedia.org/ontology/birthDate> ?date.
  FILTER ( xsd:dateTime(?date) > xsd:dateTime("1950-01-01")).
}
```

Filtrer les données avec les signes de comparaison mathématique

Filtre de regexp

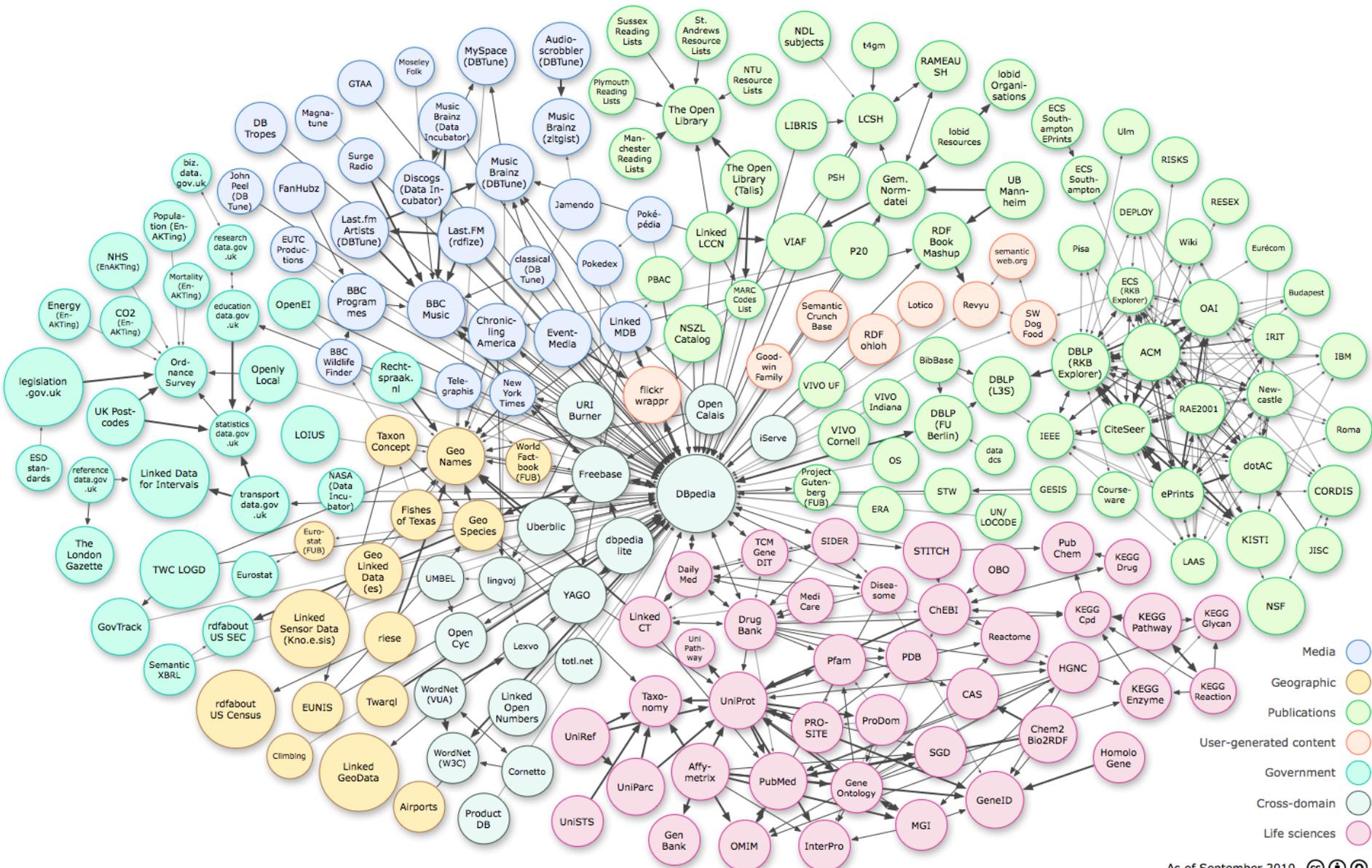
```
SELECT ?auteur ?name WHERE {
  ?auteur a <http://dbpedia.org/class/yago/FrenchNovelists>.
  ?auteur foaf:name ?name
  FILTER regex(?name, '^Jean').
}
```

Filtrer un littéral en y appliquant une regexp

Consulter la [recommandation SPARQL](#) pour découvrir les autres filtres

CONSOMMER LES DONNÉES DU LINKED DATA AVEC SPARQL

Linked Open Data cloud diagram



Sparql endpoint disponibles

Sparql endpoint lié à un ensemble de données

- Dbpedia : <http://dbpedia.org/sparql>
- Data.gov : <http://services.data.gov/sparql>
- Data.gov.uk : <http://services.data.gov.uk/sparql>
- Music Brainz : <http://dbtune.org/musicbrainz/>
- Linked Geo Data : <http://linkedgeodata.org/sparql/>
- GeoLinkedData : <http://geo.linkeddata.es/web/guest/endpoints>
- BBC programmes : <http://dbtune.org/bbc/programmes/test/>
- Données bibliographiques de la British Library : <http://bnb.bibliographica.org/sparql>
- Open University : <http://data.open.ac.uk/query>
- ...

Sparql endpoint agrégeant plusieurs ensemble de données

- Uberblic : <http://api.talis.com/stores/uberblic/services/sparql>
- Linked Data semantic repository : <http://www.ontotext.com/ldsr/>
- LOD cache : <http://lod.openlinksw.com/sparql>
- Linked Open Commerce : <http://linkedopencommerce.com/sparql>

Librairies pour manipuler du RDF

Il existe de très nombreuses librairies pour manipuler du RDF. Quelques exemples :

- Java (triples) :
 - [Jena](#)
 - [Sesame](#)
 - [Triax SCB](#)
 - [RDF2Go](#)
 - [Topaz](#)
 - [RDFReactor](#)
 - [So\(m\)mer](#)
 - [Elmo](#)
 - [jenabean](#)
 - [Linked Data API](#)
- C
 - [Redland](#)
- Python
 - [RDFlib](#)
- Ruby
 - [ActiveRDF](#)
- Scala
 - [Scardf](#)
- Javascript
 - [Jquery-SPARQL](#)
- PHP :
 - [RAP](#)
 - [ARC](#)

Interroger directement un sparql endpoint : JQuery SPARQL

Jquery-SPARQL est un module JQuery qui permet d'interroger un sparql endpoint et de récupérer les données en JSON.

<https://github.com/jgeldart/jquery-sparql>

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.4.4/jquery.min.js"></script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<script src="urlencode.js"></script>
<script src="jquery.sparql.js"></script>
<script>
  var cbfunc = function(results) {
    $("#results").val(JSON.stringify(results));
  };

  $(document).ready(function() {
    $("#run_sparql").click(function(e) {
      $.sparql("http://dbpedia.org/sparql")
        .prefix("rdfs", "http://www.w3.org/2000/01/rdf-schema#")
        .select(["?label"])
        .where("<http://dbpedia.org/resource/Tim_Berners-Lee>", "rdfs:label", "?label")
        .execute(cbfunc);
      return false;
    });
  });
</script>
```

<http://www.lespetitescases.net/semweblabs/semwebpro/sparql/jquery-sparql/example.htm>

Interroger directement un sparql endpoint : ARC2 et PHP

ARC2 est un framework PHP mis au point par Benjamin Nowack qui permet de créer, gérer et interroger un triple store local ou interroger un sparql endpoint distant.

<https://github.com/semsol/arc2>

```
<?php
/* configuration */
$remoteconfig = array(

    /* remote endpoint */
    'remote_store_endpoint' => 'http://dbpedia.org/sparql',
);

/* instantiation */
$remotestore = ARC2::getRemoteStore($remoteconfig);

$query="SELECT ?label WHERE {<http://dbpedia.org/resource/Tim_Berners-Lee> rdfs:label ?label FILTER
(lang(?label)='fr')}";

$result = $remotestore->query($query, 'row');
print $result['label'];
?>
```

Exemple :

<http://www.lespetitescases.net/semweblabs/semwebpro/sparql/arc2/remote.php>

et source :

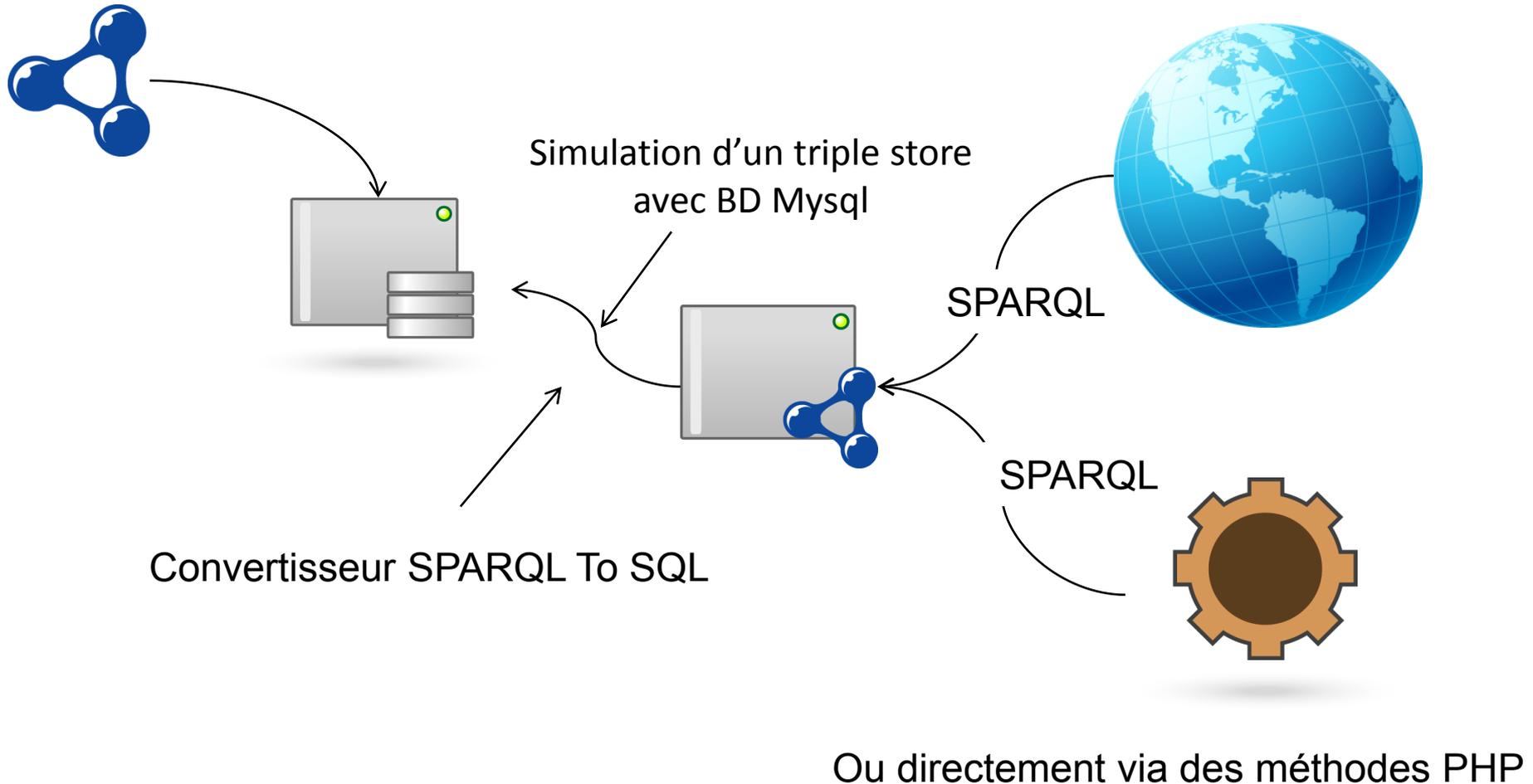
<http://www.lespetitescases.net/semweblabs/semwebpro/sparql/arc2/remote.php>

- Stocker les données dans un triple store RDF
 - Mémoire : [Corese](#), [Redstore](#)
 - Triple store natif : [Mulgara](#), [AllegroGraph](#), [BigOWLIM](#), [4store](#), [Neo4j](#)
 - BDR paramétrée : [CubicWeb](#), [Open Anzo](#), [Virtuoso](#), [ARC](#), [Oracle 11g](#), [Sesame](#), [3store](#)
 - Column store : [Cstore](#), [Heart](#), [BigData](#), [Cloudera](#)

Principes de fonctionnement de ARC

Toutes les interactions avec la base de données MySQL se font en SPARQL

Depuis l'extérieur via un sparql endpoint



Configurer ARC

```
<?php
```

```
/* Fichier de configuration pour ARC 2 */
```

```
$arc_config = array(
```

```
/* Configuration de l'accès à la base de données MySQL */
```

```
'db_host' => "", /* Domaine de la base de données ; par défaut : localhost */
```

```
'db_name' => "", /* Nom de la base de données */
```

```
'db_user' => "", /* Nom de l'utilisateur de la base de données */
```

```
'db_pwd' => "", /* Mot de passe de l'utilisateur de la base de données */
```

```
/* Nom de l'entrepôt RDF
```

```
    Cela permet de créer plusieurs entrepôts distincts avec la même base de données
```

```
*/
```

```
'store_name' => "",
```

```
/* Nombre d'erreurs maximums avant l'arrêt d'un script
```

```
'max_errors' => 100,
```

```
*/
```

```
/* Configuration complémentaire en cas de présence d'un proxy
```

```
'proxy_host' => '192.168.1.1',
```

```
'proxy_port' => 8080,
```

```
*/
```

```
/* Préfixe pour les nœuds blancs
```

```
'bnode_prefix' => 'bn',
```

```
*/
```

```
/* Formats supportés par l'extracteur de données structurées dans les pages Web */
```

```
'sem_html_formats' => 'rdfa microformats',
```

```
);
```

```
?>
```

LOAD : Charger des triplets

```
<?php
$url=$_GET['url'];
include_once('arc/ARC2.php');
include_once('config.php');

/*Instantiation*/
$store = ARC2::getStore($arc_config);

/*Si les tables ne sont pas créées, création des tables */
if (!$store->isSetUp()) {
    $store->setUp();
}

/* Chargement du flux distant */
if (isset($url)) {
    $load="LOAD <".$url.">";
    if ($rows = $store->query($load)) {
        $added_triples = $rows['result']['t_count'];
        echo $added_triples." triplets ont été ajoutés<br/>";
    }
}
else {
    print "Vous n'avez pas saisi d'URL à charger";
}
?>
```

Renvoie un tableau

Exploitation du tableau

SELECT : effectuer une requête

```
<?php
include_once('arc/ARC2.php');
include_once('config.php');

/* Instantiation de l'entrepôt à partir de la configuration */
$store=ARC2::getStore($arc_config);

/*Si les tables ne sont pas créées, création des tables */
if (!$store->isSetUp()) {
    $store->setUp();
}

$query="PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
        PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
        SELECT DISTINCT ?s ?nom WHERE {
            ?s skos:subject <http://dbpedia.org/resource/Category:Star_Wars_characters>;
            rdfs:label ?nom.
            FILTER (lang(?nom)='fr')
        }";

$res=$store->query($query);

if($store->getErrors()) {
    print_r($store->errors);
    echo "***\nQuery = $query";
}
else {
    print_r($res);
}
?>
```

Le tableau de réponse

```
Array
(
  [query_type] => select
  [result] => Array
    (
      [variables] => Array
        (
          [0] => s
          [1] => nom
        )
      [rows] => Array
        (
          [0] => Array
            (
              [s] => http://dbpedia.org/resource/Anakin_Skywalker
              [s type] => uri
              [nom] => Anakin Skywalker
              [nom type] => literal
              [nom lang] => fr
            )
          [1] => Array
            (
              [s] => http://dbpedia.org/resource/Luke_Skywalker
              [s type] => uri
              [nom] => Luke Skywalker
              [nom type] => literal
              [nom lang] => fr
            )
        )
      )
  [query_time] => 0.17298412322998
)
```

Exploiter le tableau de réponses

```
<?php
/* [SNIP] */

$query="PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
SELECT DISTINCT ?s ?nom WHERE {
  ?s skos:subject <http://dbpedia.org/resource/Category:Star_Wars_characters>;
  rdfs:label ?nom.
  FILTER (lang(?nom)='fr')
}";

$res=$store->query($query);

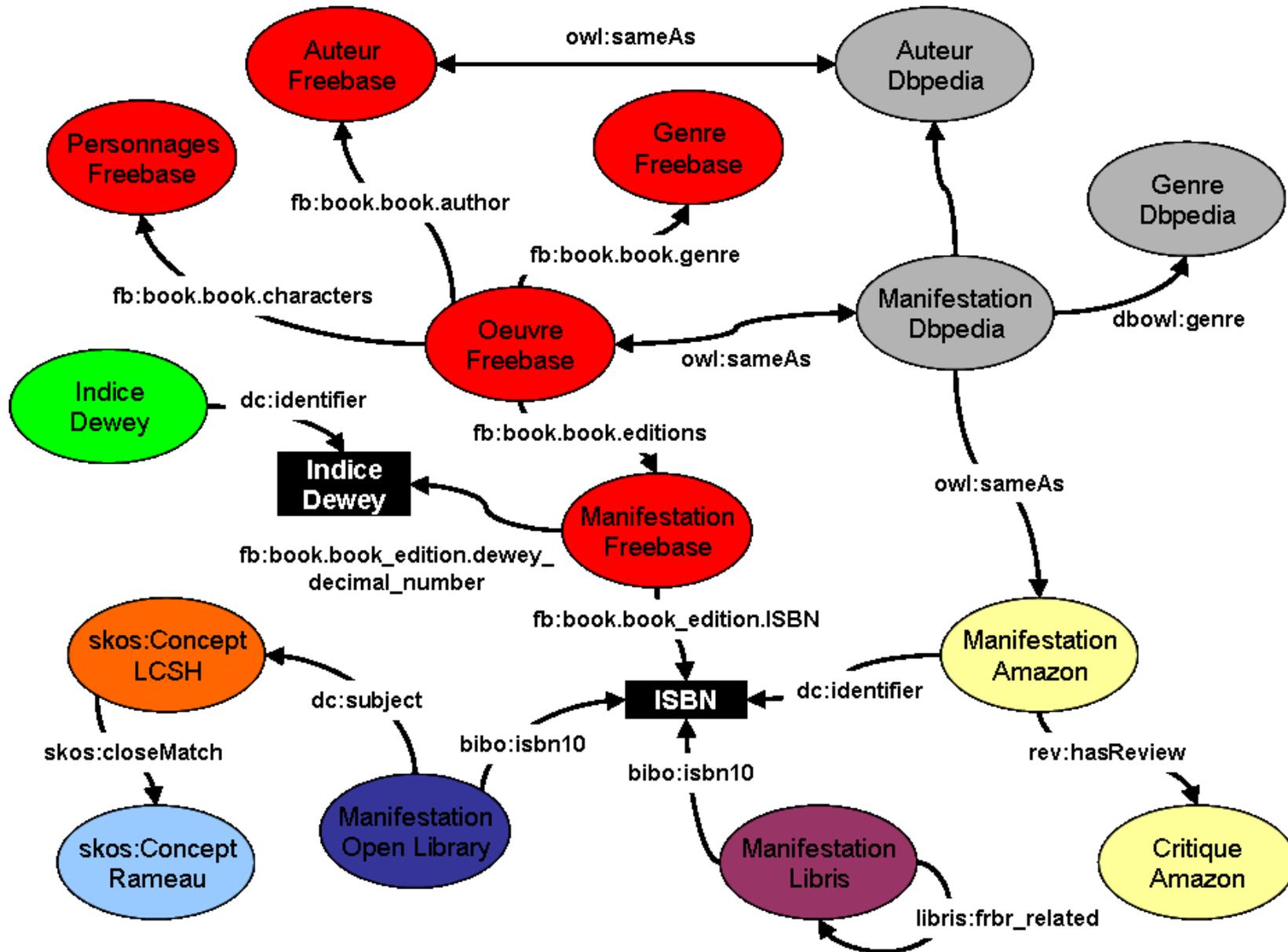
if($store->getErrors()) {
  print_r($store->errors);
  echo "***\nQuery = $query";
}
else {
  print "<ul>";
  foreach ($res['result']['rows'] as $result) {
    print "<li><a href='\\".$result['s'].\"'\>".$result['nom'].\"</a></li>";
  }
  print "</ul>";
}

?>
```

EXEMPLES DE MASHUP

Exemple d'utilisation 1 : *Linked book Mashup*

<http://www.lespetitescases.net/semweblabs/linkedbookmashup/>



Exemple d'utilisation 1 : *Linked book Mashup*



Linked book mashup



Le Seigneur des Anneaux

Par J. R. R. Tolkien

Infos bibliographiques

Éditions

Personnages

Infos bibliographiques

Le Seigneur des anneaux (The Lord of the Rings) est un roman en trois volumes de J. R. R. Tolkien paru entre 1954 et 1955. Quoique les deux histoires puissent être lues de façon indépendante, il est la suite de *Bilbo le Hobbit*, suite demandée par son éditeur à Tolkien. Durant les douze années de sa rédaction, il s'attache à faire vivre le monde dont il est le créateur, la Terre du Milieu, en truffant sa nouvelle œuvre de références et d'allusions qui la relient au monde du *Silmarillion*, sur lequel il travaille depuis 1917 et dans lequel Bilbo le Hobbit a été attiré « contre l'intention première » de son auteur. C'est une des œuvres fondamentales de la littérature dite de fantasy, terme que Tolkien explicite dans son essai de 1939 « Du conte de fées », publié dans divers recueils. Tolkien lui-même considérait son livre comme « un conte de fées [...] pour des adultes », écrit « pour amuser ; pour être agréable à lire ». Cette œuvre est composée de six livres, qui ne portent pas de titres. À l'origine, Tolkien souhaite publier *Le Seigneur des anneaux* en un seul volume, mais le prix du papier étant trop prohibitif en cette période d'après-guerre, l'œuvre est divisée en trois volumes : *La Communauté de l'anneau*, *Les Deux Tours*, et *Le Retour du roi* (The Return of the King). On fait souvent référence à cette œuvre comme à « la trilogie du Seigneur des anneaux », terme techniquement incorrect car l'œuvre fut écrite et conçue d'un seul tenant. Néanmoins, Tolkien lui-même reprend dans ses lettres, de temps à autres, le terme de « trilogie » lorsqu'il est employé par ses correspondants.

(Source : [Wikipedia](#))

Langue originale de la publication : Anglais

Indices Dewey : 823 : English fiction ;

Genres : Adventure novel ; Speculative fiction ; Fiction ; High fantasy ; Chivalric romance ; Fantasy ;

Indexation matières avec LCSH : [Fiction](#) ; [Baggins, Frodo \(Fictitious character\)](#) ; [Middle Earth \(Imaginary place\)](#)--Fiction ; [Fantasy fiction, English](#) ;

Correspondances du LCSH avec Rameau : [Nouvelles fantastiques anglaises](#) ;

Biographie de l'auteur

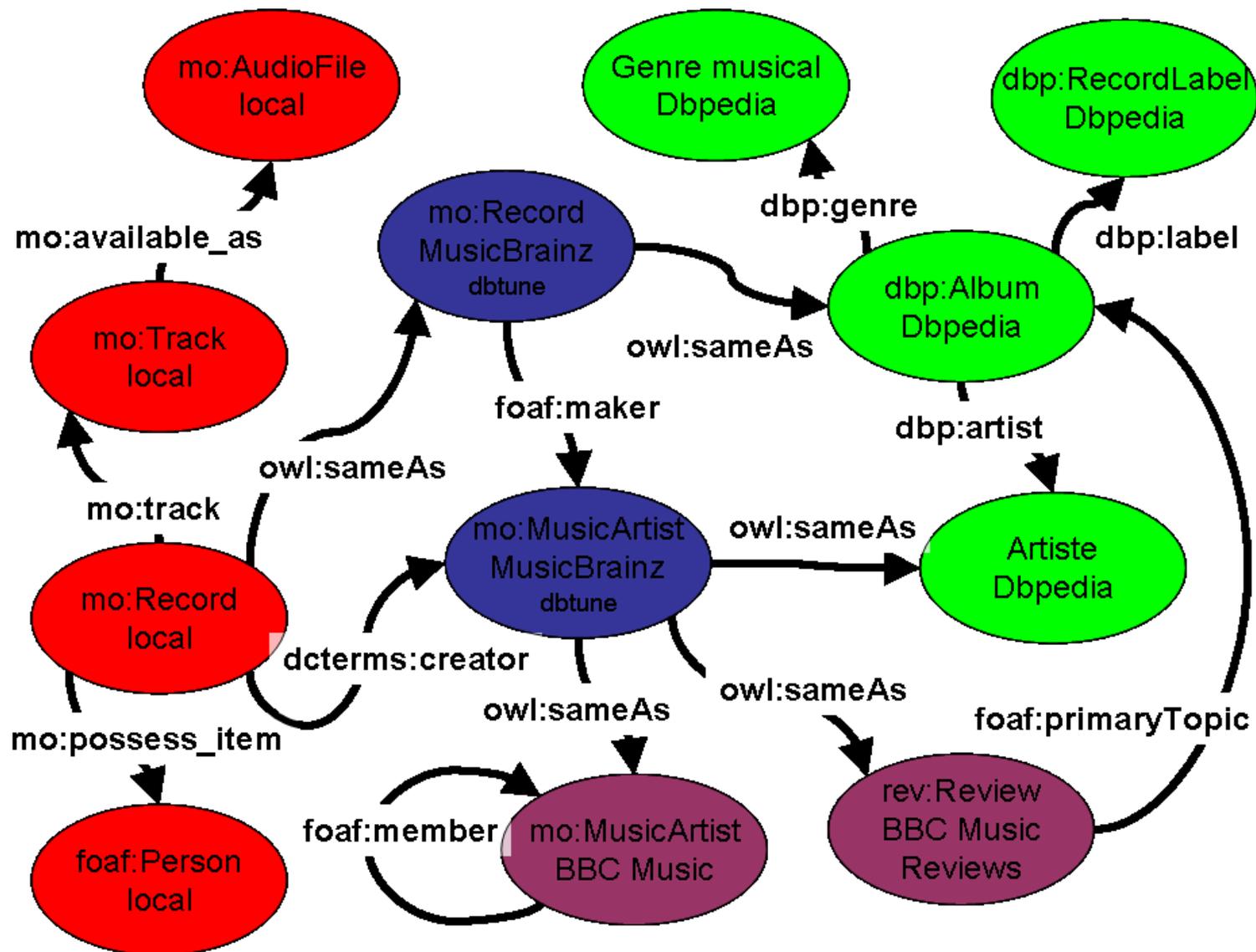


John Ronald Reuel Tolkien, plus connu sous sa signature : J. R. R. Tolkien, est un écrivain, poète, philologue et professeur d'université anglais, né le 3 janvier 1892 à Bloemfontein et mort le 2 septembre 1973 à Bournemouth. Il est principalement connu en tant qu'auteur des romans de high fantasy *Bilbo le Hobbit* et *Le Seigneur des anneaux*. Tolkien est professeur d'anglo-saxon à l'université d'Oxford de 1925 à 1949, et professeur de langue et de littérature anglaise à Merton de 1945 à 1959. Ami proche de C. S. Lewis, il est, comme lui, membre du groupe littéraire connu sous le nom d'*Inklings*. Tolkien est nommé commandeur de l'Ordre de l'Empire britannique par la reine Élisabeth II le 28 mars 1972. Après sa mort, son troisième fils Christopher publie plusieurs ouvrages basés sur les nombreuses notes et manuscrits inédits de son père, dont *Le Silmarillion*. Avec *Bilbo le Hobbit* et *Le Seigneur des anneaux*, ces livres forment un ensemble uni de récits, poèmes, essais et langues construites concernant le monde imaginaire d'Ardà, dont la Terre du Milieu est le continent principal. Dans les années 1950, Tolkien donne le nom de *legendarium* à ces écrits. De nombreux auteurs ont publié des ouvrages de fantasy avant Tolkien, mais le succès majeur remporté par *Bilbo le Hobbit* et *Le Seigneur des anneaux* au moment de leur publication en poche aux États-Unis a eu pour conséquence directe une renaissance populaire du genre, qui a conduit à considérer Tolkien comme le « père » de la fantasy moderne. Ses ouvrages ont eu une influence majeure sur les écrivains de fantasy ultérieurs. En 2008, le *Times* l'a classé sixième d'une liste des « 50 plus grands écrivains britanniques depuis 1945 ».

(Source : [Wikipedia](#))

Exemple d'utilisation 2 : *Linked My Music*

<http://www.lespetitescases.net/semweblabs/linkedmymusic/>



Exemple d'utilisation 2 : *Linked My Music*

Linked My Music

A mashup with linked musical data



[Kid A](#)



[Dummy](#)



[Pocket Symphony](#)



[10 000 Hz Legend](#)



Kid A

[Radiohead](#)

Date de parution : 1999

Nombre de pistes : 10

Genre que vous avez renseigné : Alt. Rock

Label : Parlophone

Genre dans wikipedia : Electronic music

Description

Kid A est le quatrième album du groupe de rock britannique Radiohead, il est sorti en 2000. Alors que les albums précédents restent dans un style rock alternatif, les albums suivants sont beaucoup plus psychédélics : Kid A marque l'apogée de ce style expérimental de Radiohead. Pour cette raison, il est considéré par beaucoup comme un chef-d'œuvre. Dans cet album, les guitares ont quasiment disparu au profit de synthétiseurs et de sampleurs. Le nom donné à l'album, Kid A (littéralement « Enfant A »), évoque pour certains un premier enfant cloné. Pour d'autres, il laisse penser que le groupe le considère comme son premier enfant. Avec Kid A, l'album suivant de Radiohead, Amnesiac, forme un diptyque de musique expérimentale, un prolongement : Kid A et Amnesiac forment en réalité le diptyque Kid Amnesiac. Ce disque comporte une majorité de chansons composées principalement de synthétiseurs et de boîtes à rythmes (Kid A, Idioteque, Everything in Its Right Place...), tout en gardant des sonorités pop/rock (In Limbo) et en explorant d'autres univers comme le free-jazz (The National Anthem). Lorsqu'on lance deux albums Kid A sur une chaîne hifi et sur un PC par exemple, à 17 secondes d'intervalle (16.8 s pour être précis) l'un de l'autre, on obtient quelque chose d'étonnant : les sons se marient à la perfection,

Biographie du créateur



Radiohead est un groupe de rock alternatif

anglais originaire d'Abingdon dans l'Oxfordshire ayant vendu près de 60 millions d'albums depuis leur début. Le groupe est composé de Thom Yorke au chant, de Jonny Greenwood à la guitare, de Colin Greenwood à la basse et au synthétiseur, de Ed O'Brien à la guitare et de Phil Selway à la batterie. Le premier single du groupe "Creep" paraît en 1992 et leur premier album Pablo Honey suivra en 1993. La popularité de Radiohead au Royaume-Uni augmente avec la sortie de leur second album intitulé The Bends en 1995 cependant c'est en 1997 avec la sortie de OK Computer qu'ils acquièrent une réputation mondiale. Les sorties consécutives de Kid A en 2000 et de Amnesiac en 2001 marquent l'apogée du groupe bien que la critique soit divisée. Cette période marque un changement dans la ligne de conduite du groupe, en effet, l'incorporation dans leurs morceaux de musique électronique, de post-punk et de jazz donne une nouvelle dimension à leur musicalité. La sortie de Hail to the Thief en 2003 marque la séparation entre Radiohead et leur principal label EMI.

[En savoir plus](#)

<http://www.lespetitescases.net/souvenirs/>

Boîte à souvenirs



Memories
box

Ce site a deux vocations :

- rassembler différents souvenirs personnels de visites et/ou d'événements sous la forme d'albums photos améliorés ;
- démontrer les avantages des technologies du Web sémantique et les possibilités du Web de données.

En savoir plus



Visite de La Garde-Adhémar

13 August 2007



Visite du Musée Rolin

04 August 2007

Visiter par carte



Visiter par nuage

[architecturereligieuse](#) ;

[herault](#) ; [nature](#) ; [paysage](#) ;

[sculpture](#) ;

[moyenage](#) ;

[sainthertranddecomminges](#) ; [hautegaronne](#) ;

[montagne](#) ; [conques](#) ; [antiquite](#) ; [grece](#) ;

[mosaique](#) ; [morvan](#) ; [bourgogne](#)

; [chateau](#) ; [peinture](#) ; [epoquemoderne](#) ;

[renaissance](#) ; [gardeadhemar](#) ; [drome](#) ; [leur](#) ;

[animal](#) ; [rue](#) ; [aude](#) ; [nuage](#) ; [architecture](#) ;

[mycenes](#) ; [architecturefuneraire](#) ; [louvre](#) ;

[pierrefonds](#) ; [oise](#) ; [d50](#) ;

[epoquecontemporaine](#) ; [rodin](#) ; [gers](#)

; [larressingle](#) ; [meteores](#) ; [meteora](#) ; [paysae](#) ;

[oxford](#) ; [vitrail](#) ; [maison](#) ; [seine](#) ; [75006](#) ;

[75018](#) ;

<http://lespetitescases.net/gautierpoupeurd75019> ;



Merci pour votre attention

Gautier Poupeau

Antidot | GSM: +33 (0)6 45 49 59 77

F-75 Paris | <mailto:gpoupeau@antidot.net>

Blog : [Les petites cases](#) | Twitter : @lespetitescases

