

Ontologies et

Web Sémantique

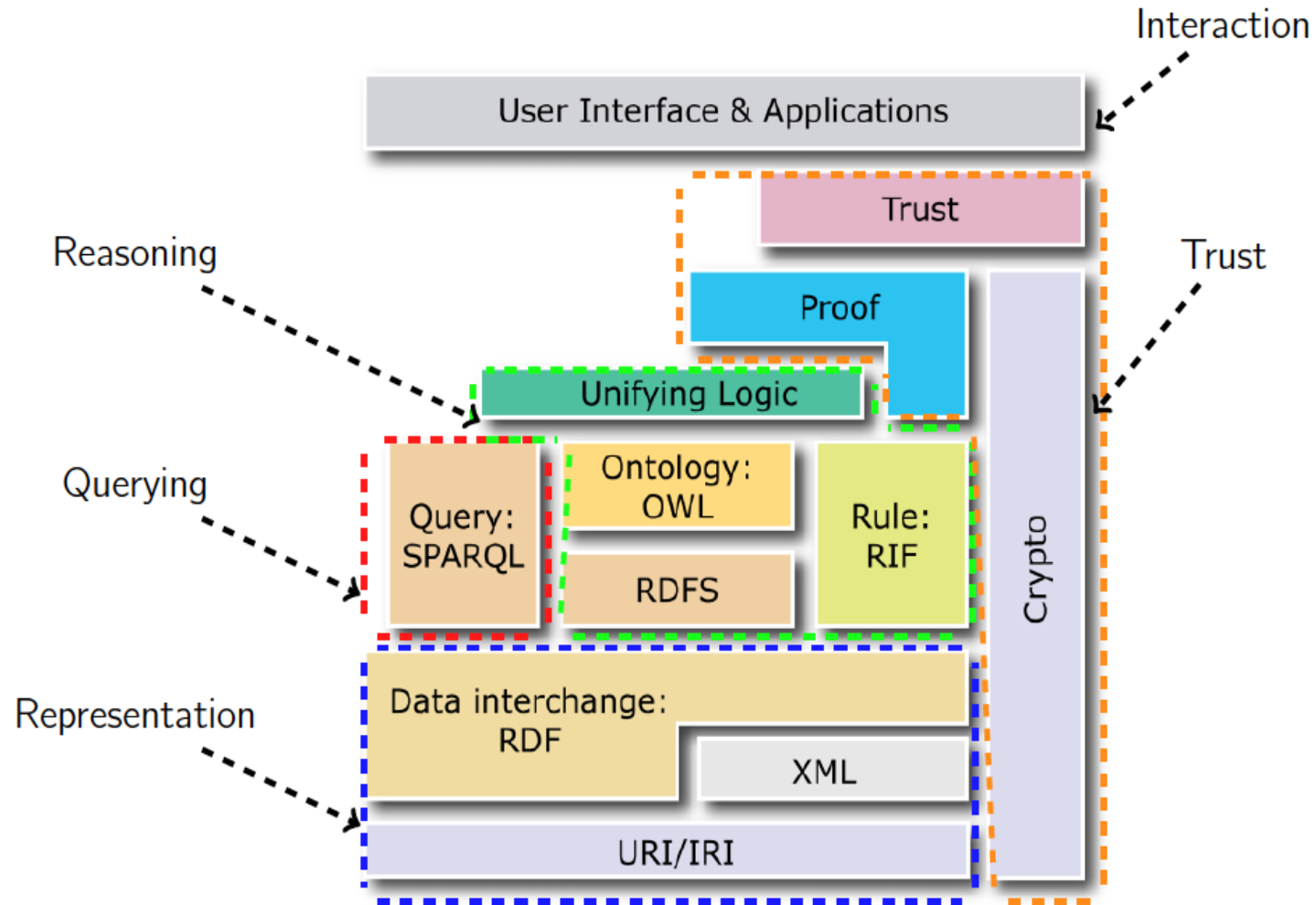
Interroger avec SPARQL

Plan du cours

1. Le langage SPARQL
2. Appariement de graphe – Pattern Matching
3. Syntaxe d'une requête SPARQL
4. Filtres, contraintes, et fonctions
5. Pré et post traitements
6. Différentes formes de requêtes

Le Langage SPARQL

Triplestore



Le Langage SPARQL

SPARQL Protocol and RDF Query Language (**SPARQL**) :

- Standard et recommandation W3C.
 - ✓ [http ://www.w3.org/TR/rdf-sparql-query](http://www.w3.org/TR/rdf-sparql-query)
- Langage d'interrogation du Web Sémantique.
- Syntaxe basée sur Turtle.
- Proche des langages d'interrogation relationnel (opérateurs).
- Interrogation d'un graphe.
 - ✓ Pattern Matching - Appariement de graphe

Le Langage SPARQL

Requête SPARQL

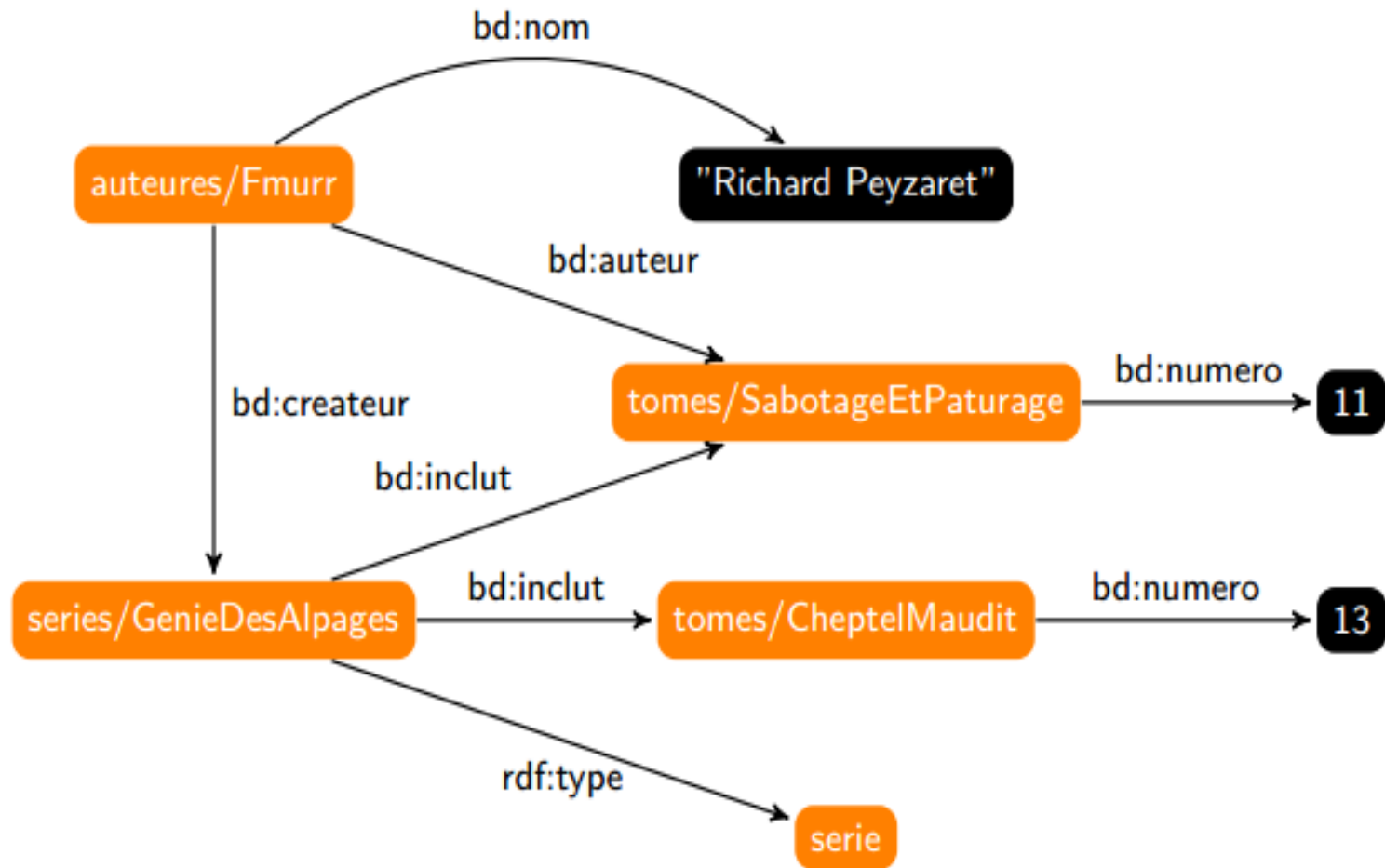
- Une requête est un graphe avec variables.
- Recherche de sous-graphes dans un graphe donné (appariement de graphe).
- Recherche des valeurs des variables qui sont des sous-graphes du graphe représentant les données.

Triplestore



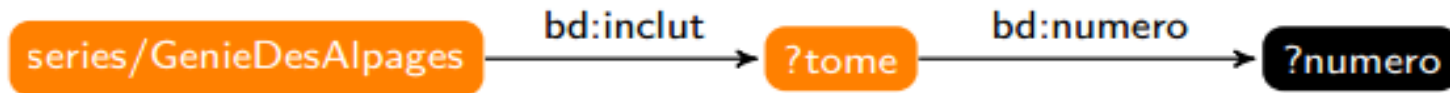
Appariement de graphe - Pattern Matching

Exemple :



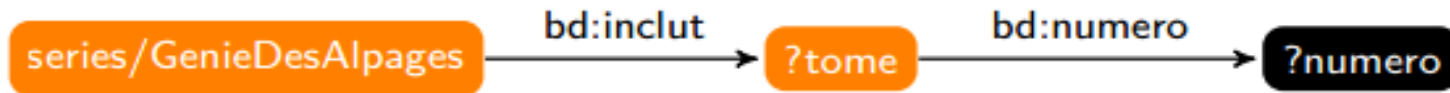
Appariement de graphe - Pattern Matching

Requête : le **numéro** des **tomes** de la série Génie des alpages



Appariement de graphe - Pattern Matching

Requête : le numéro des tomes de la série Génie des alpages



Résultat de la requête :



Syntaxe d'une requête SPARQL

```
PREFIX prefix1: <uri1>
```

```
SELECT ?var1 ... ?varn
```

```
WHERE { triplet1 .
```

```
...
```

```
tripletk .
```

```
}
```

Décrire les motifs de graphe à trouver

- Possibilité d'avoir zéro ou plusieurs préfixes
- Clause SELECT WHERE {}
 - ✓ Une variable s'écrit **?nom_variable**
 - ✓ Tout URI, littéral, et prédicat peut être remplacé par une variable
- Un motif de graphe est un ensemble de motifs de triplets

Syntaxe d'une requête SPARQL

```
PREFIX prefix1: <uri1>
```

```
SELECT ?var1 ... ?varn
```

```
WHERE { triplet1 .
```

```
...
```

```
tripletk .
```

```
}
```

Décrire les motifs de graphe à trouver

➤ Un motif de graphe est un ensemble de motifs de triplets :

1. *Motif de graphe élémentaire* : ensemble de motifs de triplets entre { };
2. *Motif de graphe de groupe* : groupe de groupe de motifs entre { }

```
SELECT .....
```

```
WHERE {
```

```
    triplet1.
```

```
    triplet2.
```

```
}
```

```
SELECT .....
```

```
WHERE {
```

```
    { triplet1. }
```

```
    { triplet2. }
```

```
}
```

Syntaxe d'une requête SPARQL

Exemple de requête de base - Endpoint : *<http://fr.dbpedia.org/sparql>*

Requête : Philosophes ayant influencés Bernard Stiegler

```
PREFIX    dbr: <http://fr.dbpedia.org/resource/>
PREFIX    dbo: <http://dbpedia.org/ontology/>
SELECT    ?phils
WHERE {
    dbr:Bernard_Stiegler    dbo:influencedBy    ?phils .
}
```

Résultat de requête :

phils
http://fr.dbpedia.org/resource/Gilbert_Simondon
http://fr.dbpedia.org/resource/André_Leroi-Gourhan
http://fr.dbpedia.org/resource/Aristote
http://fr.dbpedia.org/resource/Gérard_Granel
http://fr.dbpedia.org/resource/Edmund_Husserl
http://fr.dbpedia.org/resource/Jacques_Derrida

Syntaxe d'une requête SPARQL

Exemple de requête de base - Endpoint : <http://fr.dbpedia.org/sparql>

Requête : Philosophes ayant influencés Bernard Stiegler

```
PREFIX    dbr: <http://fr.dbpedia.org/resource/>
PREFIX    dbo: <http://dbpedia.org/ontology/>
SELECT    ?phils
WHERE {
    dbr:Bernard_Stiegler    dbo:influencedBy    ?phils .
}
```

Résultat de requête :

(Security restrictions of this server do not allow you to retrieve results in this format)

Results Format:

Execution timeout:

Options:

(The result can only be sent in one of the following formats)

- Turtle
- Auto
- HTML
- Spreadsheet
- XML
- JSON
- Javascript
- Turtle
- RDF/XML
- N-Triples
- CSV
- TSV

Syntaxe d'une requête SPARQL

Exemple de requête de base - Endpoint : <http://fr.dbpedia.org/sparql>

Requête : Philosophes ayant influencés Bernard Stiegler et qui travaillent sur la Technique.

```
PREFIX    dbr: <http://fr.dbpedia.org/resource/>
PREFIX    dbo: <http://dbpedia.org/ontology/>
SELECT    ?phils
WHERE {
    dbr:Bernard_Stiegler    dbo:influencedBy    ?phils .
    ?phils    dbo:mainInterest    dbr:Technique .
}
```

Résultat de requête :

phils
http://fr.dbpedia.org/resource/Gilbert_Simondon

Syntaxe d'une requête SPARQL

Exemple de requête de base - Endpoint : <http://fr.dbpedia.org/sparql>

Requête : Philosophes et leurs noms ayant influencés Bernard Stiegler et qui travaillent sur la Technique.

```
PREFIX dbr: <http://fr.dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?phils ?name
WHERE {
    dbr:Bernard_Stiegler  dbo:influencedBy  ?phils .
    ?phils  dbo:mainInterest  dbr:Technique .
    ?phils  foaf:name  ?name .
}
```

Résultat de requête :

phils	name
http://fr.dbpedia.org/resource/Gilbert_Simondon	"Gilbert Simondon"

Syntaxe d'une requête SPARQL

Exemple de requête de base - Endpoint : <http://fr.dbpedia.org/sparql>

Requête : Philosophes et leurs noms ayant influencés Bernard Stiegler et qui travaillent sur la Technique.

```
PREFIX dbr: <http://fr.dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?phils ?name
WHERE {
    dbr:Bernard_Stiegler    dbo:influencedBy    ?phils .
    ?phils    dbo:mainInterest    dbr:Technique ;
              foaf:name    ?name .
}
```

Résultat de requête :

phils	name
http://fr.dbpedia.org/resource/Gilbert_Simondon	"Gilbert Simondon"

Syntaxe d'une requête SPARQL

Exemple de requête de base - Endpoint : <http://fr.dbpedia.org/sparql>

Requête : Ressource ayant pour nom Bernard Stiegler en langue fr.

```
PREFIX    foaf: <http://xmlns.com/foaf/0.1/>
SELECT    ?resource
WHERE {
    ?resource    foaf:name    "Bernard Stiegler"@fr .
}
```

Résultat de requête :

resource
http://fr.dbpedia.org/resource/Bernard_Stiegler

Syntaxe d'une requête SPARQL

Exemple de requête de base - Endpoint : <http://fr.dbpedia.org/sparql>

Requête :

```
PREFIX    dbo: <http://dbpedia.org/ontology/>
PREFIX    xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT    ?res
WHERE {
    ?res    dbo:birthDate    "1952-04-01+02:00"^^xsd:date .
}
```

Résultat de requête :

resource
http://fr.dbpedia.org/resource/Jeffrey_Steenson
http://fr.dbpedia.org/resource/Bernard_Stiegler
http://fr.dbpedia.org/resource/Vincent_Bolloré
http://fr.dbpedia.org/resource/László_Tőkés
http://fr.dbpedia.org/resource/Annette_O'Toole
http://fr.dbpedia.org/resource/Gérald_Tenenbaum

Syntaxe d'une requête SPARQL

Motif Optionnel - **OPTIONAL** { }

Données :

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

_:a rdf:type foaf:Person .
_:a foaf:name "Stiegler" .
_:a foaf:mbox <mailto:stiegler@example.com> .
_:a foaf:mbox <mailto:stiegler@work.example> .

_:b rdf:type foaf:Person .
_:b foaf:name "Simondon" .
```

Syntaxe d'une requête SPARQL

Motif Optionnel - **OPTIONAL** { }

Requête :

```
PREFIX   foaf: <http://xmlns.com/foaf/0.1/>
SELECT   ?name ?mbox
WHERE {
    ?x     foaf:name      ?name .
    OPTIONAL { ?x foaf:mbox ?mbox . }
}
```

Résultat de requête :

name	mbox
"Stiegler"	<mailto:stiegler@example.com>
"Stiegler"	<mailto:stiegler@work.example>
"Simondon"	

Syntaxe d'une requête SPARQL

Opérateur OU - { } UNION { }

Requête :



```
PREFIX dbr: <http://fr.dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?phils
WHERE {
    dbr:Bernard_Stiegler  dbo:influencedBy  ?phils .
    { ?phils  dbo:mainInterest  dbr:Technique . }
UNION
    { ?phils  dbo:mainInterest  dbr:Politique . }
}
```

Résultat de requête :

phils
http://fr.dbpedia.org/resource/Gilbert_Simondon
http://fr.dbpedia.org/resource/Aristote
http://fr.dbpedia.org/resource/Jacques_Derrida
http://fr.dbpedia.org/resource/Platon
http://fr.dbpedia.org/resource/Gilles_Deleuze

Technique

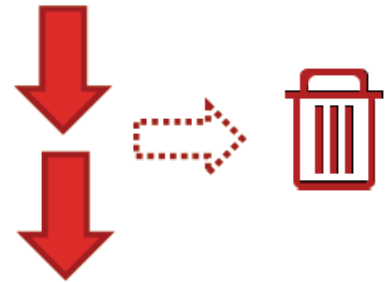
Politique

Syntaxe d'une requête SPARQL

Soustraire un motif du résultat - **MINUS** { }

Requête :

```
PREFIX dbr: <http://fr.dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?phils
WHERE {
    dbr:Bernard_Stiegler  dbo:influencedBy  ?phils .
MINUS
    { ?phils  dbo:mainInterest  dbr:Politique . }
}
```



Résultat de requête :

phils
http://fr.dbpedia.org/resource/Gilbert_Simondon
http://fr.dbpedia.org/resource/André_Leroi-Gourhan
http://fr.dbpedia.org/resource/Gérard_Granel
http://fr.dbpedia.org/resource/Edmund_Husserl
http://fr.dbpedia.org/resource/Martin_Heidegger
http://fr.dbpedia.org/resource/Sigmund_Freud

Syntaxe d'une requête SPARQL

Spécifier des valeurs prédéfinies pour les variables

Requête :

```
PREFIX dbr: <http://fr.dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?phils ?name
WHERE {
    dbr:Bernard_Stiegler  dbo:influencedBy  ?phils .
    ?phils    foaf:name ?name .
}
VALUES ?name {"Gilbert Simondon"@fr "Edmund Husserl"@fr}
```



Résultat de requête :

phils	name
http://fr.dbpedia.org/resource/Gilbert_Simondon	"Gilbert Simondon"
http://fr.dbpedia.org/resource/Edmund_Husserl	"Edmund Husserl"

Filtres, contraintes, et fonctions

Filtrer les résultats sur les valeurs – **FILTER (contrainte)**

- Déclarer des contraintes supplémentaires notamment sur les variables.
- Contrainte construite avec opérateurs et fonctions.
- Opérateurs :
 - ✓ Logiques : ! , && , ||
 - ✓ Mathématiques : +, - , * , /
 - ✓ De comparaison : = , !=, <, >
- Fonctions prédéfinies :
 - ✓ Tests : isURI, isBlank, isLiteral
 - ✓ Accesseurs et transformateurs d'un type : str, lang, datatype
 - ✓ Expression régulière : regex

Filtres, contraintes, et fonctions

Filtrer les résultats sur les valeurs – **FILTER (contrainte)**

➤ Fonctions prédéfinies :

Fonctions prédéfinies

- **isURI**(term) : renvoie vrai ssi term est un URI
- **isBlank**(term) : renvoie vrai ssi term est un noeud anonyme
- **isLiteral**(term) : renvoie vrai ssi term est un littéral
- **bound**(? var) : renvoie vrai ssi var est associé à une valeur
- **str**(term) : renvoie la chaîne de caractères correspondant à term
- **lang**(lit) : renvoie le code langue de lit
- **datatype**(lit) : renvoie l'URI du type de lit
- **sameTerm**(term1, term2) : renvoie vrai si les termes sont égaux
- **langMatches**(lang, rang) : renvoie vrai si le langage lang appartient au domaine rang
- **regex**(text, motif, options) : renvoie vrai si l'expression donnée par motif correspond à la chaîne text selon l'option spécifiée

Filtres, contraintes, et fonctions

Filtrer les résultats sur les valeurs – **FILTER (contrainte)**

Requête :

```
PREFIX ex: <http://www.example.org/resource#>
SELECT ?person ?name ?age
WHERE {
    ?person  rdf:type  ex:Person ;
             ex:name  ?name ;
             ex:age   ?age .
    FILTER (?age > "18"^^xsd:integer)
}
```

```
PREFIX ex: <http://www.example.org/resource#>
SELECT ?person ?name ?age
WHERE {
    ?person  rdf:type  ex:Person ;
             ex:age   ?age .
    FILTER (?age > 18)
    ?person ex:name  ?name .
}
```

Filtres, contraintes, et fonctions

Filtrer les résultats sur les valeurs – **FILTER (contrainte)**

- Présence d'une valeur dans une liste: **IN**

Requête :

```
PREFIX    ex: <http://www.example.org/resource/>
SELECT    ?x
WHERE {
    ?x    ex:name ?n ;
    FILTER (?n    IN    ("Stiegler"^^xsd:string,
                        "Simondon"^^xsd:string))
}
```

Filtres, contraintes, et fonctions

Filtrer les résultats sur les valeurs – **FILTER (contrainte)**

Données:

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .

_:a dc:title "Arrival"@en .
_:a dc:title "Premier Contact"@fr .
_:a dc:title "L'Arrivée"@fr-BE .
_:a dc:title "La Llegada"@es .

_:b dc:title "Il Buono, il Bruto, il Cattivo" .
```

Requête:

Utiliser les fonctions langMatches et lang pour trouver les titres en français du film Arrival.

Filtres, contraintes, et fonctions

Filtrer les résultats sur les valeurs – **FILTER (contrainte)**

Requête:

```
PREFIX    dc: <http://purl.org/dc/elements/1.1/>
SELECT    ?title
WHERE {
    ?x dc:title "Arrival"@en ;
        dc:title ?title .
    FILTER langMatches (lang(?title), "FR")
}
```

Résultat de requête :

title
"Premier Contact"@fr
"L' Arrivée"@fr-BE

Pré et Post traitements

Supprimer les doublons – **DISTINCT**

```
PREFIX dbr: <http://fr.dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT DISTINCT ?phils
WHERE {
    dbr:Bernard_Stiegler  dbo:influencedBy  ?phils .
}
```

Limiter le nombre des résultats – **LIMIT**

```
PREFIX dbr: <http://fr.dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?phils
WHERE {
    dbr:Bernard_Stiegler  dbo:influencedBy  ?phils .
}
LIMIT 4
```

Pré et Post traitements

Trier les résultats – **ORDER BY**

```
PREFIX dbr: <http://fr.dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?phils ?name
WHERE {
    dbr:Bernard_Stiegler  dbo:influencedBy  ?phils .
    ?phils foaf:name ?name .
}
ORDER BY ?name
```

Grouper, Agréger, et Filtrer les résultats – **GROUP BY - HAVING**

- Une ou plusieurs variables de regroupement.
- Condition sur les groupes.
- Fonctions d'agrégation (COUNT, MAX, AVG, etc.)

Pré et Post traitements

Grouper, Agréger, et Filtrer les résultats – **GROUP BY - HAVING**

```
PREFIX  univ: <http://www.univ.org/resource#>
SELECT ?student AVG(?note)
WHERE {
    ?student univ:note ?note .
}
GROUP BY ?student
```

Grouper, Agréger, et Filtrer les résultats – **GROUP BY - HAVING**

```
PREFIX  univ: <http://www.univ.org/resource#>
SELECT ?student AVG(?note)
WHERE {
    ?student univ:note ?note .
}
GROUP BY ?student
HAVING (AVG(?note) >= 10)
```

Pré et Post traitements

Choisir les sources de données à interroger – clause **FROM**

1. **FROM URI** : choisir le graphe à interroger, identifié par son URI.
2. **FROM NAMED GRAPH** : choisir le graphe à interroger et possibilité de faire référence à ce graphe dans la clause WHERE avec **GRAPH**.

Requête:

```
PREFIX   foaf: <http://xmlns.com/foaf/0.1/>
SELECT  ?mbox
FROM    <http://www.example.org/data/graph1.ttl>
FROM    <http://www.example.org/data/graph2.ttl>
WHERE   {
    ?x    foaf:mbox    ?mbox .
}
```

} Default Graph

Pré et Post traitements

Choisir les sources de données à interroger – clause **FROM NAMED**

1. **FROM URI** : choisir le graphe à interroger, identifié par son URI.
2. **FROM NAMED GRAPH** : choisir le graphe à interroger et possibilité de faire référence à ce graphe dans la clause WHERE avec **GRAPH**.

Requête:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX g: <http://www.example.org/data#>
SELECT ?mbox ?age
FROM <http://www.example.org/data/graph1.ttl> Default Graph
FROM NAMED g:graph2
WHERE {
    {?x foaf:mbox ?mbox .}
    UNION
    {GRAPH g:graph2 {?x foaf:age ?age} .}
}
```

Différentes formes de requête

Quatre formes de requête, dont les clauses :

➤ **SELECT**

- ✓ Recherche à apparier les termes RDF (noeud anonyme, IRI, littéraux) et les variables du motif du graphe.
- ✓ Les résultats (appariements) sont retournés sous forme de table.

➤ **CONSTRUCT**

- ✓ Retourne un résultat sous forme de graphe RDF.
- ✓ Reformule des variables sous forme de graphe RDF.
- ✓ Transforme les données d'un graphe RDF à un autre graphe.

➤ **ASK**

- ✓ Teste l'existence d'un résultat non vide.
- ✓ Retourne une valeur booléenne.

➤ **DESCRIBE**

- ✓ Donne des informations sur le graphe RDF
- ✓ Retourne un graphe

Différentes formes de requête

Clause **ASK**

Requête:

```
PREFIX dbr: <http://fr.dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
ASK
WHERE {
    dbr:Bernard_Stiegler    dbo:influencedBy dbr:Gilbert_Simondon .
}
```

Résultat de requête :

true

Différentes formes de requête

Clause **DESCRIBE**

Requête:

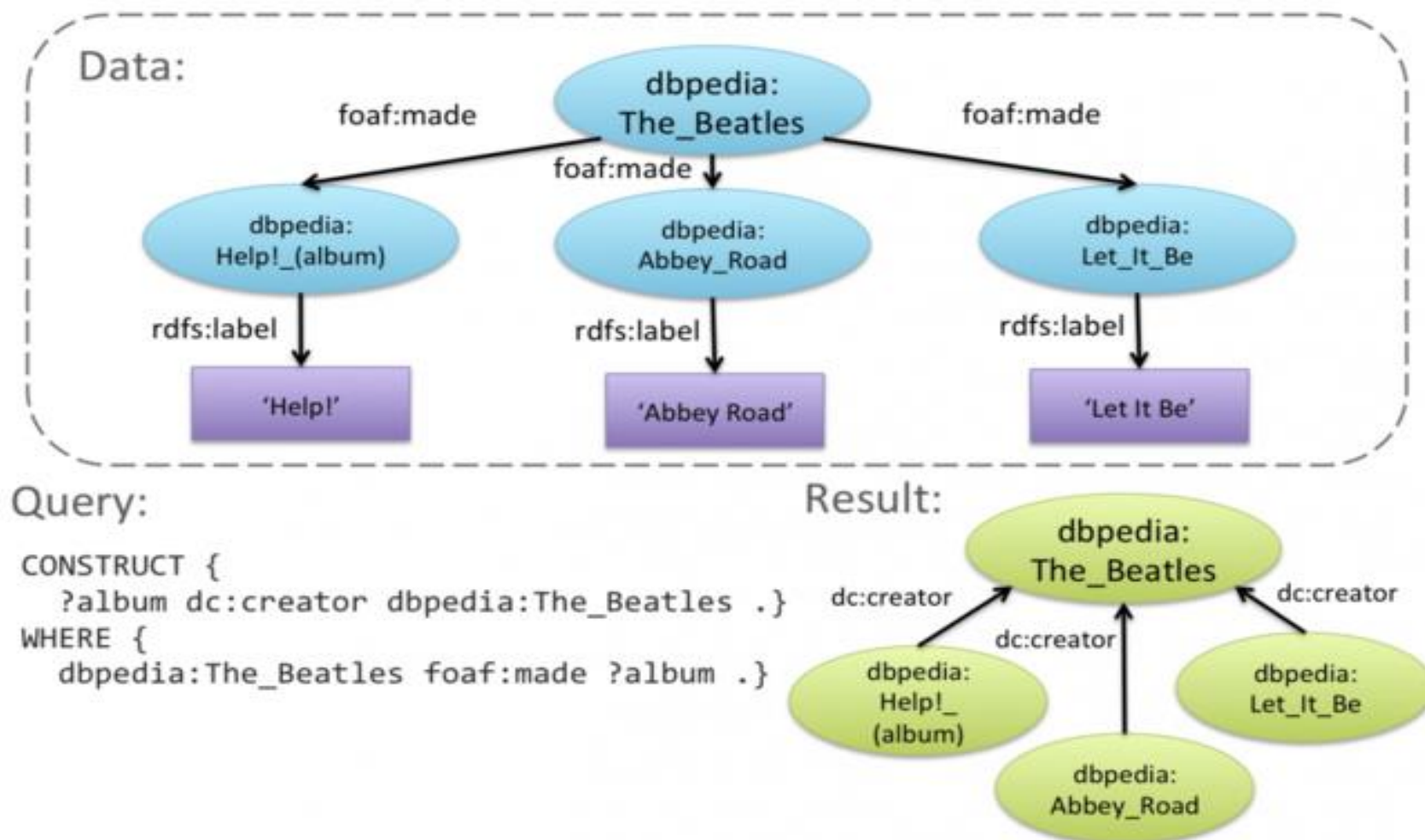
```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
DESCRIBE ?res
WHERE {
    ?res foaf:name "Bernard Stiegler"@fr .
}
```

Résultat de requête :

```
@prefix dbpedia-owl: <http://dbpedia.org/ontology/> .
@prefix dbpedia-fr: <http://fr.dbpedia.org/resource/> .
dbpedia-fr:Crise_des_subprimes dbpedia-owl:wikiPageWikiLink dbpedia-fr:Bernard_Stiegler .
dbpedia-fr:Libriste dbpedia-owl:wikiPageWikiLink dbpedia-fr:Bernard_Stiegler .
<http://fr.dbpedia.org/resource/Crise_bancaire_et_financi\u00E9re_de_l\u00027automne_2008> dbpedia-owl:wikiPageWikiLink dbpedia-fr:Bernard_Stiegler .
dbpedia-fr:Allumés_du_Jazz dbpedia-owl:wikiPageWikiLink dbpedia-fr:Bernard_Stiegler .
<http://fr.dbpedia.org/resource/D\u0002ns_le_texte> dbpedia-owl:wikiPageWikiLink dbpedia-fr:Bernard_Stiegler .
dbpedia-fr:Design dbpedia-owl:wikiPageWikiLink dbpedia-fr:Bernard_Stiegler .
dbpedia-fr:Jean-Louis_Déotte dbpedia-owl:wikiPageWikiLink dbpedia-fr:Bernard_Stiegler .
dbpedia-fr:Critiques_de_Facebook dbpedia-owl:wikiPageWikiLink dbpedia-fr:Bernard_Stiegler .
<http://fr.dbpedia.org/resource/Philippe_Petit_(journaliste_et_philosophe)> dbpedia-owl:wikiPageWikiLink dbpedia-fr:Bernard_Stiegler .
dbpedia-fr:Temps_de_cerveau_humain_disponible dbpedia-owl:wikiPageWikiLink dbpedia-fr:Bernard_Stiegler .
<http://fr.dbpedia.org/resource/Centre_national_d\u00027art_et_de_culture_Georges-Pompidou> dbpedia-owl:wikiPageWikiLink dbpedia-fr:Bernard_Stiegler .
dbpedia-fr:Déconstruction dbpedia-owl:wikiPageWikiLink dbpedia-fr:Bernard_Stiegler .
dbpedia-fr:Université_Toulouse_-_Jean_Jaurès dbpedia-owl:wikiPageWikiLink dbpedia-fr:Bernard_Stiegler .
dbpedia-fr:Consommérisme dbpedia-owl:wikiPageWikiLink dbpedia-fr:Bernard_Stiegler .
dbpedia-fr:Technique dbpedia-owl:wikiPageWikiLink dbpedia-fr:Bernard_Stiegler .
@prefix prop-fr: <http://fr.dbpedia.org/property/> .
dbpedia-fr:Gilbert_Simondon prop-fr:aInfluencé dbpedia-fr:Bernard_Stiegler ;
dbpedia-owl:influenced dbpedia-fr:Bernard_Stiegler ;
dbpedia-owl:wikiPageWikiLink dbpedia-fr:Bernard_Stiegler .
```

Différentes formes de requête

Clause **CONSTRUCT**



Différentes formes de requête

Clause **CONSTRUCT**

Données:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .  
_:a foaf:name "Stiegler" .  
_:a foaf:mbox <mailto:stiegler@example.org> .
```

Requête :

```
PREFIX    foaf: <http://xmlns.com/foaf/0.1/>  
CONSTRUCT {<http://example.org/peson#Stiegler> vcard:FN ?name}  
WHERE {  
    ?x foaf:name ?name .  
}
```

Résultat :

```
@prefix vcard: <http://www.w3.org/2001/vcard-rdf/3.0#> .  
<http://example.org/person#Stiegler> vcard:FN "Stiegler" .
```

Web Sémantique - Outils

- **Endpoints SPARQL** (jeux de données fournis) : éditeurs DBpedia et Wikidata, YASQUI, SPARQLer, Twinkle, etc.
- **Librairies** ou **frameworks** : RDFlib (Python), Apache Jena, Apache Marmotta, etc.
- **SGBD** : AllegroGraph, BlazeGraph, BrightstarDB, Dydra, Stardog, etc.

<https://www.w3.org/wiki/SparqlEndpoints>

<http://dbpedia.org/sparql>

<http://query.wikidata.org/>

<http://legacy.yasgui.org/>

<http://rdflib.readthedocs.io/en/latest/>

<http://jena.apache.org/>

<http://marmotta.apache.org/>

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

Web Sémantique - Outils

Dbpedia SPARQL Endpoint – <http://fr.dbpedia.org/sparql>

Virtuoso SPARQL Query Editor

[About](#) | [Namespace Prefixes](#) | [Inference rules](#)

Default Data Set Name (Graph IRI)

Query Text

```
PREFIX dbr: <http://fr.dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?phils
WHERE {
    dbr:Bernard_Stiegler dbo:influencedBy ?phils .
}
```

(Security restrictions of this server do not allow you to retrieve remote RDF data, see [details](#).)

Results Format:

Execution timeout: milliseconds (values less than 1000 are ignored)

Options: ☒ Strict checking of void variables

(The result can only be sent back to browser, not saved on the server, see [details](#))

Web Sémantique - Outils

Apache Jena – <https://jena.apache.org/>



A free and open source Java framework for building [Semantic Web](#) and [Linked Data](#) applications.

➤ Get started now!

⬇ Download

RDF

RDF API

Interact with the core API to create and read [Resource Description Framework](#) (RDF) graphs. Serialise your triples using popular formats such as [RDF/XML](#) or [Turtle](#).

ARQ (SPARQL)

Query your RDF data using ARQ, a [SPARQL 1.1](#) compliant engine. ARQ supports remote federated queries and free text search.

Triple store

TDB

Persist your data using TDB, a native high performance triple store. TDB supports the full range of Jena APIs.

Fuseki

Expose your triples as a SPARQL end-point accessible over HTTP. Fuseki provides REST-style interaction with your RDF data.

OWL

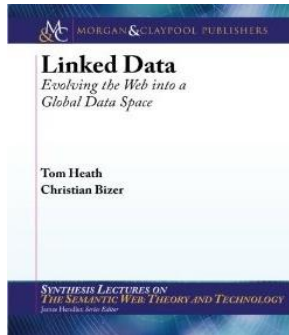
Ontology API

Work with models, RDFS and the [Web Ontology Language](#) (OWL) to add extra semantics to your RDF data.

Inference API

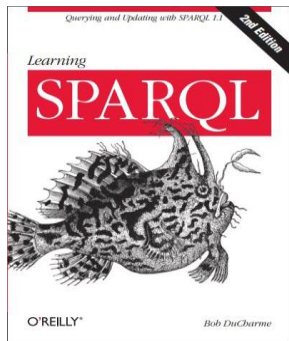
Reason over your data to expand and check the content of your triple store. Configure your own inference rules or use the built-in OWL and RDFS [reasoners](#).

Références



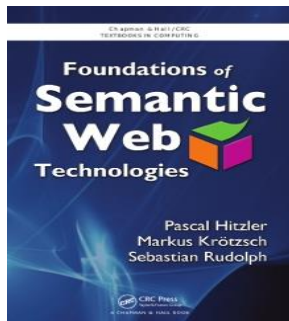
Linked Data: Evolving the Web into a Global Data Space

- ✓ Auteur : Christian Bizer, Tom Heath
- ✓ Éditeur : Morgan & Claypool Publishers
- ✓ Edition : Février 2011 - 136 pages - ISBN 9781608454310



Learning SPARQL : Querying and Updating with SPARQL

- ✓ Auteur : Bob DuCharme
- ✓ Éditeur : O'Reilly Media
- ✓ Edition: Juillet 2013– 386pages -ISBN : 9781449306595



Foundations of Semantic Web Technologies

- ✓ Auteur : Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph
- ✓ Éditeur : CRC Press/Chapman and Hall
- ✓ Edition : 2009 - 455 pages - ISBN : 9781420090505

Références

- W3C – Semantic Web
 - ✓ https://www.w3.org/2001/sw/wiki/Main_Page
- INRIA MOOC - Fabien Gandon – Web Sémantique et Web de Données
 - ✓ https://www.canal-u.tv/producteurs/inria/cours_en_ligne/web_semantique_et_web_de_donnees
- Fabien Duchateau – BDBIO – RDF et SPARQL
 - ✓ <http://liris.cnrs.fr/~fduchate/ens/BDBIO/cm/rdf.pdf>
- Langage d'interrogation SPARQL pour RDF
 - ✓ <http://www.yoyodesign.org/doc/w3c/rdf-sparql-query/>