



SPARQL 1.1

Quoi de neuf pour manipuler les données sur le Web ?

Alexandre Passant
DERI, NUI Galway

<http://apassant.net> - @terraces

SemWeb.pro

Paris, 18 Janvier 2011



■ Charte SPARQL WG

- <http://www.w3.org/2009/05/sparql-phase-II-charter.html>
- The scope of this charter is to extend SPARQL technology to include some of the features **that the community has identified as both desirable and important for interoperability based on experience** with the initial version of the standard

■ SPARQL 1.1

- Compatibilité ascendante avec SPARQL 1.0
- Update incrémentale et non fondamentale

Tour d'horizon des nouveautés



Digital Enterprise Research Institute

www.deri.ie

SPARQL New Features and Rationale

W3C Working Draft 2 July 2009

This version:

<http://www.w3.org/TR/2009/WD-sparql-features-20090702/>

Latest version:

<http://www.w3.org/TR/sparql-features/>

Editors:

[Kjetil Kjernsmo, Computas AS](#)

[Alexandre Passant, DERI Galway at the National University of Ireland, Galway, Ireland](#)

Copyright © 2009 W3C® (MIT, ERCIM, Keio), All Rights Reserved. W3C [liability](#), [trademark](#) and [document use](#) rules apply.



Enabling **networked** knowledge.

- **En cours de développement au sein du W3C**
 - SPARQL Working Group démarré début 2009
 - Successeur du DAWG (SPARQL “1.0”)
 - Last Call prévue d’ici quelques semaines

- **Commentaires appréciés et bienvenus !**
 - Mailing-list: public-rdf-dawg-comments@w3.org

Statut des implémentations



Digital Enterprise Research Institute

www.deri.ie

- SPARQL 1.1 déjà supporté en partie par différentes implémentations
 - ARQ - <http://openjena.org>
 - OpenLink Virtuoso - <http://virtuoso.openlinksw.com/>
 - OpenAnzo - <http://www.openanzo.org/>
 - Corese - <http://www-sop.inria.fr/teams/edelweiss/wiki/wakka.php?wiki=CoreseDownloads>
 - PerlRDF - <http://github.com/kasei/perlrdf/>



Enabling **networked** knowledge.

Agenda du tutoriel



Digital Enterprise Research Institute

www.deri.ie

- SPARQL 1.1 Query
 - Interrogation
- SPARQL 1.1 Update
 - Mise a jour
- Service Description
 - Découverte

- SPARQL et inférence
 - RDFS, OWL, etc.
- SPARQL distribué
 - Utilisations de sources multiples



Enabling **networked** knowledge.

- <http://apassant.net/home/2011/01/semwebpro/>

- Données: sparql/books.ttl
- Requetes: sparql/*.sparql
- Archive: sparql.tgz

- **Demonstrations avec ARQ (Query)**

- <http://openjena.org>
- ./bin/query -data=books.ttl -query=xxx.sparql
- NB: Prefixes omits sur les slides pour lisibilité

SPARQL 1.1 Query



Digital Enterprise Research Institute

www.deri.ie

SPARQL 1.1 Query Language

W3C Working Draft 14 October 2010

This version:

<http://www.w3.org/TR/2010/WD-sparql11-query-20101014/>

Latest version:

<http://www.w3.org/TR/sparql11-query/>

Previous version:

<http://www.w3.org/TR/2010/WD-sparql11-query-20100601/>

Editors:

Steve Harris, Garlik

Andy Seaborne, Invited Expert <andy.seaborne@epimorphics.com>

Previous Editor:

Eric Prud'hommeaux, W3C <eric@w3.org>

Please refer to the [errata](#) for this document, which may include some normative corrections.

The [previous errata](#) for this document, are also available.

See also [translations](#).

[Copyright](#) © 2010 W3C® (MIT, [ERCIM](#), [Keio](#)), All Rights Reserved. W3C [liability](#), [trademark](#) and [document use](#) rules apply.



Enabling **networked** knowledge.

- **Nouveautés SPARQL 1.1 Query**
 - Projection (Project Expression)
 - Agrégation (Aggregates)
 - Requêtes imbriquées (Subqueries)
 - Négation
 - Chemins (Property-path)
 - Librairie de fonctions (Function library)

■ AS

- Assignation et création de nouvelles valeurs
- Utilisable avec les agrégats, fonctions mathématiques et librairie de fonctions

■ Deux achats, prix double – double.sparql

```
SELECT (?price * 2 AS ?double)
```

```
WHERE {
```

```
  ?book ns:price ?price
```

```
}
```

double
80
60
60
70
60

■ AS

- Assignation et création de nouvelles valeurs
- Utilisable avec les agrégats, fonctions mathématiques et librairie de fonctions

■ Reduction en cours – discount.sparql

```
SELECT ?price (?price * (1-?discount) AS ?newprice)
```

```
WHERE {
```

```
  ?book ns:price ?price ;
```

```
  ns:discount ?discount .
```

```
}
```

price	newprice
30	24.0

■ AS

- Assignation et création de nouvelles valeurs
- Utilisable avec les agrégats, fonctions mathématiques et librairie de fonctions
- Attention aux conflits de nommage !

PREFIX ns: <http://example.org/ns#>

SELECT (MAX(?price) AS ?price) ## /!\ Invalide

WHERE {

 ?book ns:price ?price

}

■ SPARQL 1.0:

□ FILTER + BOUND

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

SELECT ?name WHERE {

 ?x foaf:givenName ?name .

 OPTIONAL { ?x foaf:knows ?who } . FILTER (!BOUND(?who))

}

■ SPARQL 1.1

□ MINUS et NOT EXISTS

NOT EXISTS



■ Fonctionnalité

- Simplifier la négation en SPARQL 1.1
- Identifier les patrons de requête non existants

■ Livres sans prix – notesists.sparql

```
SELECT ?book ?title WHERE {  
  ?book dc:title ?title .
```

```
  FILTER NOT EXISTS { ?book ns:price ?price }
```

```
}
```

```
-----  
| book | title |  
-----  
| <http://example.org/book/book?> | "Harry Potter and the Deathly Hallows" |  
-----
```

■ Fonctionnalité

- Simplifier la négation en SPARQL 1.1
- Supprimer certaines valeurs des résultats (Evaluer le MINUS et le soustraire des résultats)

■ Livres sans prix – minus.sparql

```
SELECT ?book ?title WHERE {  
  ?book dc:title ?title .  
  MINUS { ?book ns:price ?price }  
}
```

book	title
<http://example.org/book/book?>	"Harry Potter and the Deathly Hallows"

■ MINUS versus NOT EXISTS ?

- Différence basée sur le nommage des variables
- <http://www.w3.org/TR/sparql11-query/#neg-notexists-minus>

```
@prefix : <http://example/> .  
:a :b :c .
```

```
SELECT * { ?s ?p ?o FILTER NOT EXISTS { ?x ?y ?z } }
```

evaluates to a result set with no solutions because { ?x ?y ?z } matches given any ?s ?p ?o, SO NOT EXISTS { ?x ?y ?z } eliminates any solutions.

s	p	o
---	---	---

whereas with MINUS, there is no shared variable between the first part (?s ?p ?o) and the second (?x ?y ?z) so no bindings are eliminated.

```
SELECT * { ?s ?p ?o MINUS { ?x ?y ?z } }
```

Results:

s	p	o
<http://example/a>	<http://example/b>	<http://example/c>

■ SPARQL 1.0

- Besoin de passer par un script externe pour les fonctions agrégats « classiques » et disponibles en SQL
 - Compter un nombre de résultats
 - Trouver une valeur moyenne
 - Prendre une valeur aléatoire

■ SPARQL 1.1

- COUNT, SUM, MIN, MAX, AVG, SAMPLE, GROUP_CONCAT
- Combinés avec les projections précédentes

■ COUNT

□ Nombre d'éléments associés a une expression

■ Nombre de personnes - count.sparql

```
SELECT (COUNT(?person) as ?count)
WHERE {
  ?person a foaf:Person .
}
```

```
-----
| count |
=====
| 3     |
-----
```

■ SUM

- Sommes des valeurs associées a une expression

■ Somme de prix - sum.sparql

```
SELECT (SUM(?price) as ?total)
WHERE {
  ?book ns:price ?price .
}
```

```
-----
| total |
=====
| 165   |
-----
```

SUM



■ SUM

- Sommes des valeurs associées a une expression

■ Somme des achats – sum2.sparql

```
SELECT (SUM(?price) as ?total) WHERE {  
  :c ns:owns [  
    ns:price ?price  
  ]  
}
```

```
-----  
| total |  
=====  
| 105   |  
-----
```

MIN / MAX / AVG



■ MIN / MAX / AVG

- Valeurs min / max / moyenne des valeurs associées a une expression
- MIN / MAX sont réalisables en SPARQL 1.0 avec FILTER

■ Prix moyen – avg.sparql

```
SELECT (AVG(?price) as ?avg) WHERE {  
  ?book ns:price ?price  
}
```

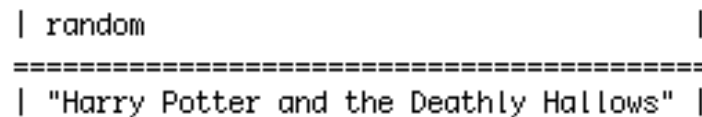
```
-----  
| avg |  
=====  
| 33.000000000000000000000000000000 |  
-----
```

■ SAMPLE

- Valeur aléatoire dans un jeu de données

■ Titre au hasard – sample.sparql

```
SELECT (SAMPLE(?title) as ?random) WHERE {  
  { ?book dc:title ?title }  
  UNION  
  { ?book rdfs:label ?title }  
} LIMIT 1
```

A diagram showing a query result table. The table has two rows. The first row contains the value 'random'. The second row contains the value 'Harry Potter and the Deathly Hallows'. The table is enclosed in a dashed border. A large circle is drawn around the second row, and a smaller circle is drawn around the first row.

random
"Harry Potter and the Deathly Hallows"

■ GROUP_CONCAT

- Concaténer les résultats d'une agrégation

■ Titre des livres d'une personne – gconcat.sparql

```
SELECT
```

```
  (SAMPLE(?person) as ?someone)
```

```
  (GROUP_CONCAT(?title; SEPARATOR = " + ") AS ?titles )
```

```
WHERE {
```

```
  ?person ns:owns ?book .
```

```
  ?book dc:title ?title .
```

```
} GROUP BY ?person LIMIT 1
```

```
-----  
| someone                | titles                |  
-----  
| <http://example.org/book/b> | "Harry Potter and the Deathly Hallows + Harry Potter and the Philosopher's Stone" |  
-----
```

Chemins (Property path)



- “A property path is a possible route through a graph between two graph node”
 - <http://www.w3.org/TR/sparql11-query/#pp-language>
 - Longueur = 1 => requete classique
 - Basés sur les expressions régulières
- Exemples
 - Identifier les amis d'un ami, et les amis de leurs amis, etc. quelque soit la distance
 - Identifier les sous-catégories de « Web » dans Dbpedia, quelque soit la profondeur de hiérarchie
 - Identifier si deux ressources sont liées par certains liens, tout en excluant d'autres

Property path disponibles



Syntax Form	Matches
<i>uri</i>	A URI or a prefixed name. A path of length one.
<i>^elt</i>	Inverse path (object to subject).
<i>!uri</i> Or <i>!(uri₁/ .../uri_n)</i>	Negated property set. A URI which is not one of <i>uri_i</i>
<i>!^uri</i> and <i>!(uri₁/ .../uri_j/^uri_{j+1}/ .../^uri_n)</i>	Negated property set. A URI which is not one of <i>uri_i</i> , nor one of <i>uri_{j+1}...uri_n</i> as reverse paths
<i>(elt)</i>	A group path <i>elt</i> , brackets control precedence.
<i>elt1 / elt2</i>	A sequence path of <i>elt1</i> , followed by <i>elt2</i>
<i>elt1 elt2</i>	A alternative path of <i>elt1</i> , or <i>elt2</i> (all possibilities are tried).
<i>elt*</i>	A path of zero or more occurrences of <i>elt</i> .
<i>elt+</i>	A path of one or more occurrences of <i>elt</i> .
<i>elt?</i>	A path of zero or one <i>elt</i> .
<i>elt{n,m}</i>	A path between <i>n</i> and <i>m</i> occurrences of <i>elt</i> .
<i>elt{n}</i>	Exactly <i>n</i> occurrences of <i>elt</i> .
<i>elt{n,}</i>	<i>n</i> or more occurrences of <i>elt</i> .
<i>elt{,n}</i>	Between 0 and <i>n</i> occurrences of <i>elt</i> .

■ Simplifier l'union

- Triples utilisant un prédicat parmi plusieurs
- E.g. label
 - rdfs:label, dc:title, ...

■ Titre des livres qq soit le prédicat – altern.sparql

```
SELECT ?name
```

```
WHERE {
```

```
  ?book (dc:title|rdfs:label) ?name
```

```
}
```

name
"Harry Potter and the Philosopher's Stone"
"Harry Potter and the Prisoner Of Azkaban"
"Harry Potter and the Chamber of Secrets"
"Harry Potter and the Order of the Phoenix"
"Harry Potter and the Goblet of Fire"
"Harry Potter and the Deathly Hallows"
"Harry Potter and the Half-Blood Prince"

■ Simplifier les requetes bi-directionnelles

- Identifier des chemins inverse, sans qu'ils soient explicites dans les données
- Simulation de la symétrie (e.g. ex:frere)

■ Relations hiérarchiques

```
SELECT ?c1 ?c2 WHERE { ?c2 ^skos:broader ?c2 }
```

<=>

```
SELECT ?c1 ?c2 WHERE { ?c2 skos:broader ?c2 }
```

■ Simplifier les requetes bi-directionnelles

- Identifier des chemins inverse, sans qu'ils soient explicites dans les données
- Simulation de la symétrie (e.g. ex:frere)

■ Relations entre 2 personnes – inverse.sparql

- Avec alternative

```
SELECT ?a ?b
```

```
WHERE {
```

```
  ?a (foaf:knows|^foaf:knows) ?b
```

```
}
```

- Simplifier les imbrications

- A suivi de B, suivi de C ...

- Titre des livres d'une connaissance – seq1.sparql

```
SELECT ?name
```

```
WHERE {
```

```
  :c foaf:knows/ns:owns/dc:title ?name
```

```
}
```

```
-----  
| name |  
-----  
| "Harry Potter and the Deathly Hallows" |  
| "Harry Potter and the Philosopher's Stone" |  
-----
```

- Simplifier les imbrications

 - A suivi de B, suivi de C ...

- Titre des livres d'une connaissance – seq2.sparql

```
SELECT ?name
```

```
WHERE {
```

```
  :c foaf:knows/ns:owns/(dc:title|rdfs:label) ?name
```

```
}
```

```
-----  
| name |  
-----  
| "Harry Potter and the Deathly Hallows" |  
| "Harry Potter and the Chamber of Secrets" |  
| "Harry Potter and the Philosopher's Stone" |  
-----
```

- **Simplifier les imbrications**
 - A suivi de B, suivi de C ...
- **Titre des livres d'une connaissance – seq3.sparql**
 - seq2.sparql est un raccourci pour

```
SELECT ?name WHERE {  
  :c foaf:knows [ ns:owns ?book ]  
  { ?book dc:title ?name } UNION { ?book rdfs:label ?name }  
}
```

■ Sous-classes d'une classes donnée

- Quelque soit la profondeur, avec possibilité de mentionner un intervalle {1,4}
- Simulation de l'inférence RDFS

■ Sous-classes d' Agent dans MO – musicontology.sparql

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

```
SELECT ?s
```

```
FROM <http://motools.sourceforge.net/doc/musicontology.n3>
```

```
WHERE {
```

```
  ?s rdfs:subClassOf+ <http://xmlns.com/foaf/0.1/Agent>
```

```
}
```


Chemins et ontologies



```
Alexandre-Passants-MacBook-Pro:Jena-2.6.4 alex$ ./bin/query -query=./semwebpro/musicontology.sparql
```

```
-----  
| s |  
-----  
| <http://purl.org/ontology/mo/SoundEngineer> |  
| <http://purl.org/ontology/mo/Composer> |  
| <http://purl.org/ontology/mo/Arranger> |  
| <http://purl.org/ontology/mo/Performer> |  
| <http://purl.org/ontology/mo/Listener> |  
| <http://purl.org/ontology/mo/MusicArtist> |  
| <http://purl.org/ontology/mo/Conductor> |  
-----
```

```
Alexandre-Passants-MacBook-Pro:Jena-2.6.4 alex$ ./bin/query -query=./semwebpro/musicontology2.sparql
```

```
-----  
| s |  
-----  
| <http://purl.org/ontology/mo/SoundEngineer> |  
| <http://purl.org/ontology/mo/Composer> |  
| <http://purl.org/ontology/mo/Arranger> |  
| <http://purl.org/ontology/mo/Performer> |  
| <http://purl.org/ontology/mo/Listener> |  
| <http://purl.org/ontology/mo/MusicArtist> |  
| <http://purl.org/ontology/mo/MusicGroup> |  
| <http://purl.org/ontology/mo/SoloMusicArtist> |  
| <http://purl.org/ontology/mo/Conductor> |  
-----
```

Requetes imbriquées



■ Sous-requetes

- Nécessité de passer par un langage externe en SPARQL 1.0

■ Cas du « Limit per resource »

- Identifier un seul nom par connaissance

```
$query = "SELECT ?person WHERE { :Alice :knows ?person .}";  
$res = do_query($query);  
foreach ($res as $r) {  
  $person = $r->person->value;  
  $query = "SELECT ?name WHERE { ?person foaf:name ?  
    name . } LIMIT 1";  
}
```


■ Sous-requetes

- Nécessité de passer par un langage externe en SPARQL 1.0

■ Cas du « Limit per resource » - subselect.sparql

- Identifier un seul nom par connaissance
- **!/ \ portée / nommage des variables**

```
SELECT ?who ?name WHERE {  
  :c foaf:knows ?who .  
  {  
    SELECT ?name WHERE {  
      ?who foaf:name ?name .  
    } LIMIT 1  
  }  
}
```



who	name
:b	"Alex P"

- **Intégrer les fonctions utilisés par les différents intégrateurs**
 - Utilisés fréquemment mais non standardisés en SPARQL 1.0 (Principalement Xpath)
 - En supplément des fonctions habituelles, et de nouvelles
 - **Présents dans le prochain WD**

- **Exemples**
 - STRLEN – longueur d’une chaîne
 - CONCAT – concaténation
 - COALESCE – première expression sans erreur
 - CONTAINS – sous-chaîne de caractère

SPARQL 1.1 Update



Digital Enterprise Research Institute

www.deri.ie

SPARQL 1.1 Update

W3C Working Draft 14 October 2010

This version:

<http://www.w3.org/TR/2010/WD-sparql11-update-20101014/>

Latest version:

<http://www.w3.org/TR/sparql11-update/>

Previous version:

<http://www.w3.org/TR/2010/WD-sparql11-update-20100601/>

Editors:

Simon Schenk <sschenk@uni-koblenz.de>

Paul Gearon <gearon@computer.org>

Alexandre Passant, DERI Galway at the National University of Ireland, Galway, Ireland <alexandre.passant@deri.org>

Copyright © 2010 W3C® (MIT, ERCIM, Keio), All Rights Reserved. W3C [liability](#), [trademark](#) and [document use](#) rules apply.



Enabling **networked** knowledge.

■ SPARQL 1.0

- Read-only: Interrogation mais pas de mise a jour
- Nécessité de passer par des langage externes

■ SPARQL Update Member Submission

- <http://www.w3.org/Submission/SPARQL-Update/>
- Base de travail pour le WG
- Quelques nouveautés

■ SPARQL et Graph Stores

- <http://www.w3.org/TR/sparql11-update/#t31>
- Un entrepôt SPARQL peut supporter les graphes nommés
 - Granularité et provenance
 - Ajout de suppression de graphs
 - Pas nécessaire d'être « authoritative »
- Et un graphe par défaut (qui peut être l'agrégation des graphes nommés, selon les implémentations)

■ 2 type d'opérations

- Opérations sur les graphes (graph management)
- Opération sur les données au sein de graphes (graph update)

Opérations sur les graphes



Digital Enterprise Research Institute

www.deri.ie

- Créer et supprimer des graphes
 - CREATE et DROP

- CREATE
 - Création d'un graphe nommé, vide
 - CREATE [SILENT] GRAPH <uri>

- DROP
 - Suppression d'un graphe
 - DROP [SILENT] (GRAPH <uri> | DEFAULT | NAMED | ALL)
 - Le graphe DEFAULT existe toujours, i.e. recrée si supprimé

■ LOAD et CLEAR

- Ajout / suppression de triplets dans un graphe

■ LOAD

- Chargement de données depuis un graphe
- `LOAD [SILENT] <documentURI> [INTO GRAPH <uri>]`
 - documentURI peut être le résultat d'une requête SPARQL
 - `LOAD <http://example.org/sparql?q=CONSTRUCT...> INTO <http://example.org/data/Paris>`

■ CLEAR

- `CLEAR [SILENT] (GRAPH <uri> | DEFAULT | NAMED | ALL)`
- Suppression des données d'un graphe local
- Le graphe existe toujours

■ INSERT DATA

□ Insertion de données sans variables

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
INSERT DATA
{
  <http://example/book3> dc:title "A new book" ;
                        dc:creator "A.N.Other" .
}
```

Data before:

```
# Default graph
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix ns: <http://example.org/ns#> .

<http://example/book3> ns:price 42 .
```

Data after:

```
# Default graph
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix ns: <http://example.org/ns#> .

<http://example/book3> ns:price 42 .
<http://example/book3> dc:title "A new book" .
<http://example/book3> dc:creator "A.N.Other" .
```

■ DELETE DATA

- Idem pour la suppression de données

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

DELETE DATA
{
  <http://example/bookx> dc:title "David Copperfield" ;
                        dc:creator "Edmund Wells" .
}
```

Data before:

```
# Default graph
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix ns: <http://example.org/ns#> .

<http://example/bookx> ns:price 42 .
<http://example/bookx> dc:title "David Copperfield" .
<http://example/bookx> dc:creator "Edmund Wells" .
```

Data after:

```
# Default graph
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix ns: <http://example.org/ns#> .

<http://example/bookx> ns:price 42 .
```

Opérations de triplets



■ INSERT

- Insertion avec un BGP classique
- Possibilité de copie de données, en combinant avec la clause GRAPH
- **#!/\ DELETE/INSERT est actuellement redondant**

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

INSERT
  { GRAPH <http://example/bookStore2> { ?book ?p ?v } }
WHERE
  { GRAPH <http://example/bookStore>
    { ?book dc:date ?date .
      FILTER ( ?date < "2000-01-01T00:00:00-2:00"^^xsd:dateTime )
      ?book ?p ?v
    } }
  }
```

```

PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

INSERT
  { GRAPH <http://example/bookStore2> { ?book ?p ?v } }
WHERE
  { GRAPH <http://example/bookStore>
    { ?book dc:date ?date .
      FILTER ( ?date < "2000-01-01T00:00:00-2:00"^^xsd:dateTime )
      ?book ?p ?v
    } }

```

```

# Graph: http://example/bookStore
@prefix dc: <http://purl.org/dc/elements/1.1/> .

<http://example/book3> dc:title "Fundamentals of Compiler Design" .
<http://example/book3> dc:date "1996-01-01T00:00:00-2:00" .

<http://example/book> dc:title "SPARQL 1.1 Tutorial" .

<http://example/bookx> ns:price 42 .
<http://example/bookx> dc:title "David Copperfield" .
<http://example/bookx> dc:creator "Edmund Wells" .
<http://example/bookx> dc:date "2002-01-01T00:00:00-2:00" .

```

```

# Graph: http://example/bookStore2
@prefix dc: <http://purl.org/dc/elements/1.1/> .

<http://example/book> dc:title "SPARQL 1.0 Tutorial" .

```

```

# Graph: http://example/bookStore
@prefix dc: <http://purl.org/dc/elements/1.1/> .

<http://example/book3> dc:title "Fundamentals of Compiler Design" .
<http://example/book3> dc:date "1996-01-01T00:00:00-2:00" .

<http://example/book> dc:title "SPARQL 1.1 Tutorial" .

<http://example/bookx> ns:price 42 .
<http://example/bookx> dc:title "David Copperfield" .
<http://example/bookx> dc:creator "Edmund Wells" .
<http://example/bookx> dc:date "2002-01-01T00:00:00-2:00" .

```

```

# Graph: http://example/bookStore2
@prefix dc: <http://purl.org/dc/elements/1.1/> .

<http://example/book> dc:title "SPARQL 1.0 Tutorial" .

<http://example/book3> dc:title "Fundamentals of Compiler Design" .
<http://example/book3> dc:date "1996-01-01T00:00:00-2:00" .

```

Opérations de triplets



■ DELETE

- Suppression avec un BGP classique
- USING pour spécifier le graph dans les 2 clauses

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

USING <http://example/addresses>
DELETE { ?person ?property ?value }
WHERE { ?person foaf:firstName 'Fred' }
```

Data before:

```
# Graph: http://example/addresses
@prefix foaf: <http://xmlns.com/foaf/0.1/>

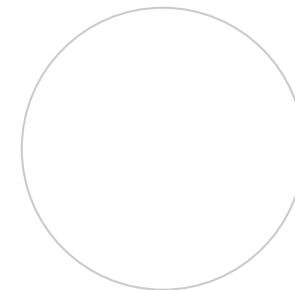
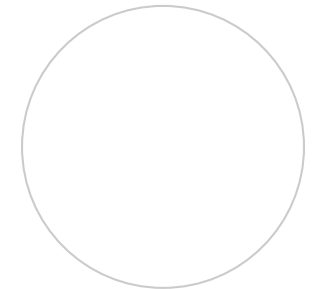
<http://example/william> a foaf:Person .
<http://example/william> foaf:firstName "William" .
<http://example/william> foaf:mbox "bill@example" .

<http://example/fred> a foaf:Person .
<http://example/fred> foaf:firstName "Fred" .
<http://example/fred> foaf:mbox "fred@example" .
```

Data after:

```
# Graph: http://example/addresses
@prefix foaf: <http://xmlns.com/foaf/0.1/>

<http://example/william> a foaf:Person .
<http://example/william> foaf:firstName "William" .
<http://example/william> foaf:mbox "bill@example" .
```



- Raccourcis pour la manipulation de graphs
 - Prochain WD
 - Commentaires bienvenus d'utilisateurs et développeurs

- Inspirés des commandes fichiers UNIX
 - COPY [SILENT] ([GRAPH] <fromURI> | DEFAULT) TO ([GRAPH] <toURI> | DEFAULT)
 - MOVE [SILENT] ([GRAPH] <fromURI> | DEFAULT) TO ([GRAPH] <toURI> | DEFAULT)
 - ADD [SILENT] ([GRAPH] <fromURI> | DEFAULT) TO ([GRAPH] <toURI> | DEFAULT)

Raccourcis UPDATE



■ Raccourcis pour la manipulation de graphs

- Prochain WD
- Commentaires bienvenus d'utilisateurs et développeurs

```
MOVE DEFAULT TO <http://example.org/named>
```

■ Inspi

- CO ([G
- MC ([G
- AD <tc

Data before:

```
# Default graph
@prefix foaf: <http://xmlns.com/foaf/0.1/>

<http://example/william> a foaf:Person .
<http://example/william> foaf:firstName "William" .
<http://example/william> foaf:mbox "bill@example" .

# Graph http://example.org/named
<http://example/fred> a foaf:Person .
```

Data after:

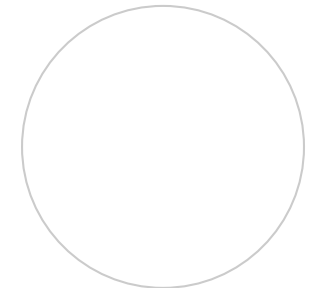
```
# Default graph

# Graph http://example.org/named
<http://example/william> a foaf:Person .
<http://example/william> foaf:firstName "William" .
<http://example/william> foaf:mbox "bill@example" .
```

) TO

) TO

TO ([GRAPH]



SPARQL 1.1 Uniform HTTP Protocol for Managing RDF Graphs

W3C Working Draft 14 October 2010

This version:

<http://www.w3.org/TR/2010/WD-sparql11-http-rdf-update-20101014/>

Latest version:

<http://www.w3.org/TR/sparql11-http-rdf-update/>

Previous version:

<http://www.w3.org/TR/2010/WD-sparql11-http-rdf-update-20100601/>

Editor:

Chimezie Ogbuji, Cleveland Clinic Foundation ogbujic@ccf.org

Copyright © 2009 W3C[®] (MIT, ERCIM, Keio), All Rights Reserved. W3C [liability](#), [trademark](#) and [document use](#) rules apply.

■ RESTful design et HTTP Verbs

- PUT, DELETE, POST, associés aux requêtes SPARQL 1.1 Update
- PATCH (Informative)

Service Description



Digital Enterprise Research Institute

www.deri.ie

SPARQL 1.1 Service Description

W3C Working Draft 14 October 2010

This version:

<http://www.w3.org/TR/2010/WD-sparql11-service-description-20101014/>

Latest version:

<http://www.w3.org/TR/sparql11-service-description/>

Previous version:

<http://www.w3.org/TR/2010/WD-sparql11-service-description-20100601/>

Editor:

Gregory Todd Williams, Rensselaer Polytechnic Institute <greg@evilfunhouse.com>

Copyright © 2010 W3C® (MIT, ERCIM, Keio), All Rights Reserved. W3C [liability](#), [trademark](#) and [document use](#) rules apply.



Enabling **networked** knowledge.

- Permettre a un SPARQL endpoint de fournir des informations a son sujet
 - Une méthode pour découvrir les informations
 - Un vocabulaire pour les représenter
- Type d'informations modélisables
 - Fonctions disponibles
 - Quantité de données
 - Vocabularies utilisés
 - Graphes
 - Inférence
 - ...

■ Découvrir les informations a propos d'un SPARQL endpoint

- <http://www.w3.org/TR/sparql11-service-description/>
- SPARQL services made available via the SPARQL Protocol **should** return a service description document at the service URL. This service description
 - **must** be made available in an RDF serialization
 - **may** be provided embedded in (X)HTML by RDFa, and
 - **should** use content-negotiation if available in other RDF representations

■ Exemple

- <http://myrdf.us/sparql11> (cf. source)
- `curl -H "Accept: application/rdf+xml" http://myrdf.us/sparql11`

- <http://www.w3.org/ns/sparql-service-description#>
 - 10 Classes
 - 7 Instances
 - [Unique URIs for Semantic Web Entailment Regimes](#)
 - [Unique URIs for OWL 2 Profiles](#)
 - [Unique URIs for File Formats](#)
 - 8 Propriétés
- **Combinaison avec d'autres vocabulaires pour des informations statistiques (e.g. nombre de triplets)**
 - SCOVO - <http://sw.joanneum.at/scovo/schema.html>
 - void - <http://vocab.deri.ie/void/guide>

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
  xmlns:ex="http://example/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:scovo="http://purl.org/NET/scovo#"
  xmlns:sd="http://www.w3.org/ns/sparql-service-description#"
  xmlns:void="http://rdfs.org/ns/void#">
  <sd:Service>
    <sd:url rdf:resource="http://www.example/sparql/" />
    <sd:feature rdf:resource="http://www.w3.org/ns/sparql-service-description#DereferencesURIs" />
    <sd:extensionFunction>
      <sd:ScalarFunction rdf:about="java:com.ldodds.sparql.Distance" />
    </sd:extensionFunction>
    <sd:defaultDatasetDescription>
      <sd:Dataset>
        <sd:defaultGraph>
          <sd:Graph>
            <void:statItem>
              <rdf:Description>
                <scovo:dimension rdf:resource="http://rdfs.org/ns/void#numberOfTriples" />
                <rdf:value rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">100</rdf:value>
              </rdf:Description>
            </void:statItem>
          </sd:Graph>
        </sd:defaultGraph>
        <sd:namedGraph>
          <rdf:Description>
            <sd:name rdf:resource="http://www.example/named-graph" />
            <sd:graph>
              <sd:Graph>
                <void:statItem>
                  <rdf:Description>
                    <scovo:dimension rdf:resource="http://rdfs.org/ns/void#numberOfTriples" />
                    <rdf:value rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">2000</rdf:value>
                  </rdf:Description>
                </void:statItem>
              </sd:Graph>
            </sd:graph>
          </rdf:Description>
        </sd:namedGraph>
      </sd:Dataset>
    </sd:defaultDatasetDescription>
  </sd:Service>
</rdf:RDF>
```

SPARQL 1.1 Entailment Regimes

W3C Working Draft 14 October 2010

This version:

<http://www.w3.org/TR/2010/WD-sparql11-entailment-20101014/>

Latest version:

<http://www.w3.org/TR/sparql11-entailment/>

Previous version:

<http://www.w3.org/TR/2010/WD-sparql11-entailment-20100601/>

Editors:

Birte Glimm, Oxford University Computing Laboratory <birte.glimm@comlab.ox.ac.uk>

Chimezie Ogbuji, Cleveland Clinic <ogbujic@ccf.org>

Contributors:

Sandro Hawke, W3C <sandro@w3.org>

Ivan Herman, W3C <ivan@w3.org>

Bijan Parsia, University of Manchester <bparsia@cs.manchester.ac.uk>

Axel Polleres, Digital Enterprise Research Institute <axel@polleres.net>

Andy Seaborne, Invited Expert <andy.seaborne@epimorphics.com>

Copyright © 2010 W3C® (MIT, ERCIM, Keio), All Rights Reserved. W3C [liability](#), [trademark](#) and [document use](#) rules apply.

■ Associer SPARQL et inférence Web pour la manipulation de données

- <http://www.w3.org/TR/sparql11-entailment/>
- RDF Entailment Regime
- RDFS Entailment Regime
- D-Entailment Regime
- OWL 2 RDF-Based Semantics Entailment Regime
- OWL 2 Direct Semantics Entailment Regime
- RIF Core Entailment
- Time permitting

■ RDFS et SPARQL 1.1 + inférence

- Dériver des faits a partir de l'ontologie
- foaf:knows => foaf:Person
 - foaf:knows rdfs:range foaf:Person

■ Données / requête

- :Alex a foaf:Person ; foaf:knows :Gautier .
- SELECT ?x WHERE { ?x a foaf:Person }

■ Réponses

- 1.0 => Alex
- 1.1 + RDFS => :Alex + :Gautier

SPARQL 1.1 Federation Extensions

W3C Working Draft 1 June 2010

This version:

<http://www.w3.org/TR/2010/WD-sparql11-federated-query-20100601/>

Latest version:

<http://www.w3.org/TR/sparql11-federated-query/>

Editor:

Eric Prud'hommeaux, W3C <eric@w3.org>

Contributor:

Andy Seaborne, Talis <andy.seaborne@talis.com>

Please refer to the [errata](#) for this document, which may include some normative corrections.

The [previous errata](#) for this document, are also available.

See also [translations](#).

[Copyright](#) © 2010 [W3C](#)[®] ([MIT](#), [ERCIM](#), [Keio](#)), All Rights Reserved. W3C [liability](#), [trademark](#) and [document use](#) rules apply.

■ SPARQL 1.0

- FROM et FROM NAMED pour interroger des graphes distants
- SELECT * FROM <<http://example.org>> WHERE { ?s ?p ?o }

■ SPARQL 1.1

- Interrogation d'endpoints distants
- SERVICE et BINDINGS
- Time permitting

■ SERVICE

- Interrogation d'endpoints distants
- Possibilité de combiner les services

■ Personne définie dans 2 entrepots

```
SELECT ?person
```

```
WHERE {
```

```
  SERVICE <http://dbpedia.org/sparql> { ?p a foaf:Person . }
```

```
  SERVICE <http://example.org/sparql> { ?p a foaf:Person . }
```

```
}
```

■ SERVICE

- Interrogation d'endpoints distants
- Possibilité de combiner les services

■ Membres des Clash – service.sparql

```
SELECT ?members
```

```
WHERE { {
```

```
  SERVICE <http://dbpedia.org/sparql> {
```

```
    dbp:The_Clash dbr:pastMembers ?members .
```

```
  } } UNION {
```

```
  SERVICE <http://dbtune.org/musicbrainz/sparql> {
```

```
    dbt:8f92558c-2baa-4758-8c38-615519e9deda foaf:member ?members .
```

```
  } } }
```

■ SERVICE

- Interrogation d'endpoints distants
- Possibilité de combiner les services

■ Membres des Clash – service.sparql

```
SELECT ?members
```

```
WHERE { {
```

```
  SERVICE <http://dbpedia.org/sp  
    dbp:The_Clash dbr:pastMembe
```

```
  } } UNION {
```

```
  SERVICE <http://dbtune.org/mu  
    dbt:8f92558c-2baa-4758-8c38
```

```
  } } }
```

```
-----  
| members |  
-----  
| <http://dbpedia.org/resource/Mick_Jones_%28The_Clash%29> |  
| dbp:Rob_Harper |  
| dbp:Topper_Headon |  
| dbp:Keith_Levene |  
| dbp:Joe_Strummer |  
| dbp:Terry_Chimes |  
| dbp:Paul_Simonon |  
| dbp:Vince_White |  
| "Nick Sheppard"@en |  
| "Pete Howard"@en |  
| dbt:01f8343c-06e2-4f05-a137-f8ea7efa85c4 |  
| dbt:19237b9f-2457-4eec-99f5-3166fc73fb4d |  
| dbt:2d2d9c00-2e78-4ead-857a-ecbb9e765d6e |  
| dbt:64c43453-f396-42b1-bff5-01644d710d94 |  
| dbt:899a92fe-7ee5-4bc4-9628-0eff26df09e5 |  
| dbt:af4ea2a3-c54d-40b4-8120-a1f0b838ad54 |  
-----
```

■ SERVICE

- Interrogation d'endpoints distants
- Possibilité de combiner les services

■ Identifier les personnes depuis des services pertinents pour un domaine donné

```
SELECT ?person
```

```
WHERE {
```

```
  ?service dc:subject <http://dbpedia.org/resource/Music> .
```

```
  SERVICE ?service { ?p a foaf:Person . }
```

```
}
```

■ BINDINGS

- Propager des contraintes dans différents endpoints
- Simplification du FILTER

■ Personnes avec un prénom « alex »

```
SELECT ?person WHERE {  
  SERVICE <http://ex.orgsparql> {  
    ?p a foaf:Person ; foaf:name ?name }  
  SERVICE <http://example.org/sparql> {  
    ?p a foaf:Person ; foaf:name ?name }  
} BINDINGS ?name { ("alex") }
```


■ SPARQL 1.1

- Nouveautés issues de demandes utilisateurs
- Simplification du processus de requete (SPARQL 1.1 Query)
- Découverte de SPARQL endpoints (SD)
- Mise a jour des données (SPARQL 1.1 Update)
- Mais aussi: Requetes distribuées et inférence

■ Commentaires appréciés et bienvenus !

- Bientot en Last Call !
- Mailing-list: public-rdf-dawg-comments@w3.org

Remerciements



Digital Enterprise Research Institute

www.deri.ie

- **Membres du W3C SPARQL Working Group**
 - Certaines requêtes tirées des spécifications en cours
 - Jeu de données basées sur <http://sparql.org>
- **Utilisateurs, intégrateurs et développeurs**
 - Retour sur expérience de SPARQL 1.0
 - Feedback et commentaires sur SPARQL 1.1
- **Science Foundation Ireland**
 - Lion2 - Enabling Network Knowledge @ DERI
- **SemWeb.pro / LogiLab**
 - <http://semweb.pro>



Enabling **networked** knowledge.