



SPARQL

Michel Gagnon



SPARQL

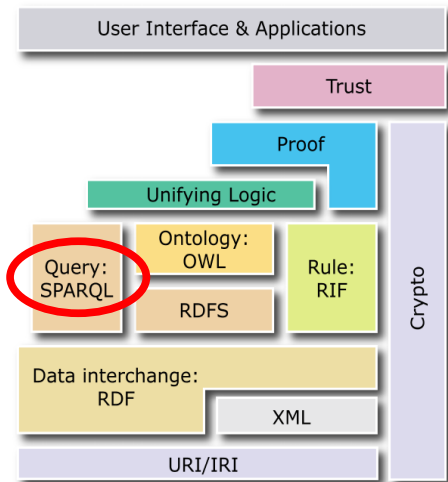
Soit une base de données en RDF

Comment accéder à l'information qui y est contenue?

Il faut pour cela un langage de requête

Comme RDF est un modèle de graphe, il est logique que le langage de requête soit basé sur un processus d'appariement de graphes

W3C propose le langage SPARQL





SPARQL

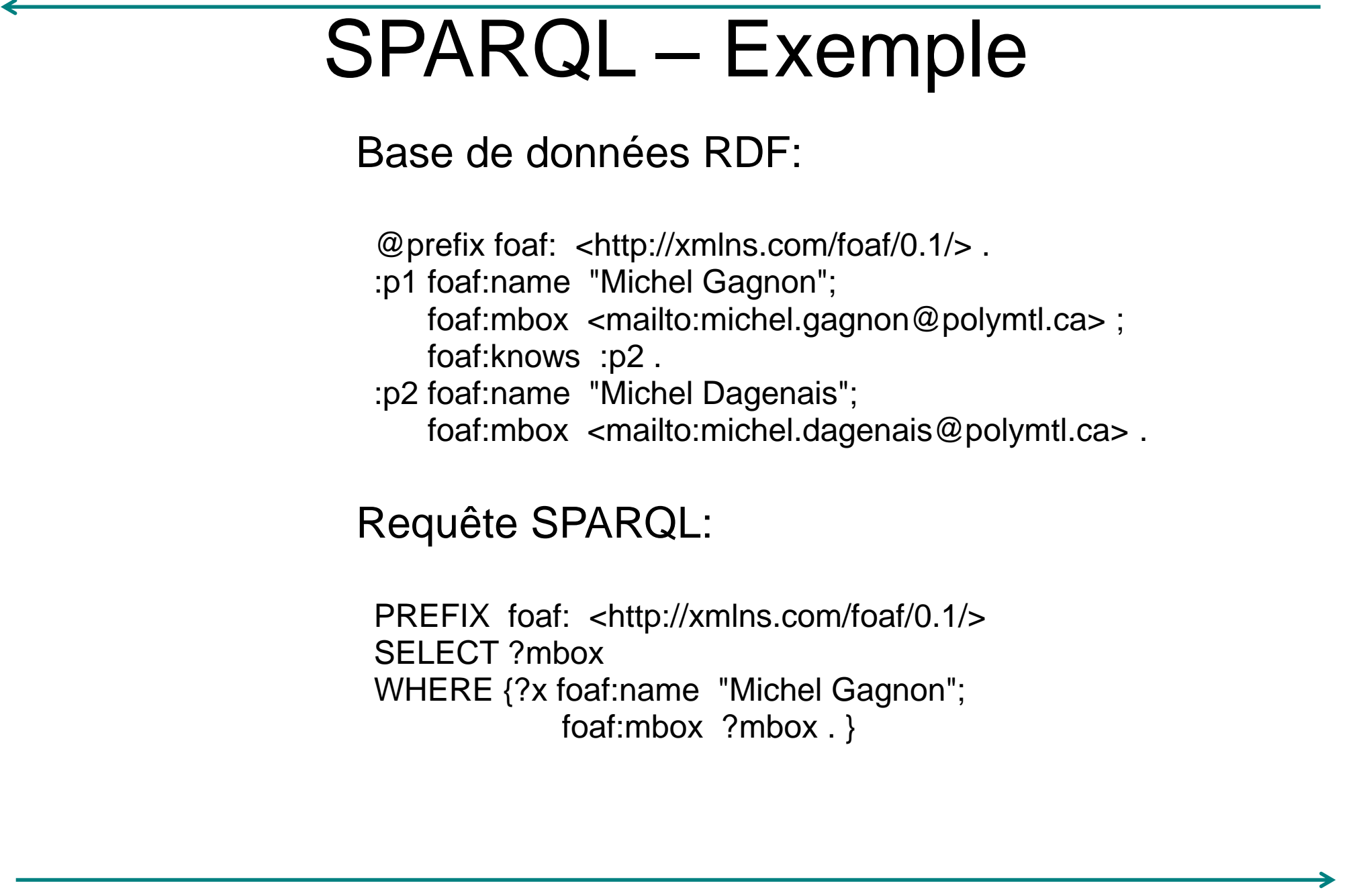
Ce que permet SPARQL:

de Extraire l'information sous forme de URI,
de noeuds vides ou de littéraux

Extraire des sous-graphes RDF

Construire de nouveaux graphes RDF à
partir de l'information obtenue






SPARQL – Exemple

Base de données RDF:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .  
:p1 foaf:name "Michel Gagnon";  
    foaf:mbox <mailto:michel.gagnon@polymtl.ca> ;  
    foaf:knows :p2 .  
:p2 foaf:name "Michel Dagenais";  
    foaf:mbox <mailto:michel.dagenais@polymtl.ca> .
```

Requête SPARQL:

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



SPARQL – Exemple

Base de données RDF:


```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .  
:p1 foaf:name "Michel Gagnon";  
    foaf:mbox <mailto:michel.gagnon@polymtl.ca>;  
    foaf:knows :p2 .  
:p2 foaf:name "Michel Dagenais";  
    foaf:mbox <mailto:michel.dagenais@polymtl.ca> .
```

Requête SPARQL:

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

<mailto:michel.gagnon@polymtl.ca>





SPARQL – Exemple

Base de données RDF:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
:p1 foaf:name "Michel Gagnon";
    foaf:mbox <mailto:michel.gagnon@polymtl.ca>;
    foaf:knows :p2 .
:p2 foaf:name "Michel Dagenais";
    foaf:mbox <mailto:michel.dagenais@polymtl.ca> .
```

Requête SPARQL:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?n1 ?n2
WHERE {?x foaf:knows ?y ;
        foaf:name ?n1 .
        ?y foaf:name ?n2 . }
```



SPARQL – Exemple

Base de données RDF:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
:p1 foaf:name "Michel Gagnon";
    foaf:mbox <mailto:michel.gagnon@polymtl.ca>;
    foaf:knows :p2 .
:p2 foaf:name "Michel Dagenais";
    foaf:mbox <mailto:michel.dagenais@polymtl.ca> .
```

Requête SPARQL:


```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?n1 ?n2
WHERE {?x foaf:knows ?y ;
        foaf:name ?n1 .
        ?y foaf:name ?n2 . }
```

?n1

?n2

Michel Gagnon

Michel Dagenais



SPARQL – Exemple

Base de données RDF:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .  
@prefix local: <http://www.polymt.ca/vocab/> .  
:p1 foaf:name "Michel Gagnon";  
    local:worksAt local:dgi;  
    foaf:knows :p2 .  
:p2 foaf:name "Michel Dagenais";  
    local:worksAt local:dgi .
```

Requête SPARQL:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>  
PREFIX local: <http://www.polymt.ca/vocab/>  
CONSTRUCT {?n1 local:colleagueOf ?n2 . }  
WHERE {?n1 local:worksAt ?w.  
       ?n2 local:worksAt ?w . }
```



SPARQL – Exemple

Base de données RDF:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix local: <http://www.polymt.ca/vocab/> .
:p1 foaf:name "Michel Gagnon";
    local:worksAt local:dgi;
    foaf:knows :p2 .
:p2 foaf:name "Michel Dagenais";
    local:worksAt local:dgi .
```

Requête SPARQL:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX local: <http://www.polymt.ca/vocab/>
CONSTRUCT {?n1 local:colleagueOf ?n2 . }
WHERE {?n1 local:worksAt ?w.
       ?n2 local:worksAt ?w . }
:p1 local:colleagueOf :p1 ;
    local:colleagueOf :p2 .
:p2 local:colleagueOf :p2 ;
    local:colleagueOf :p1 .
```

SPARQL – Exemple

Base de données RDF:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix local: <http://www.polymt.ca/vocab/> .
:p1 foaf:name "Michel Gagnon";
    local:worksAt local:dgi;
    foaf:knows :p2 .
:p2 foaf:name "Michel Dagenais";
    local:worksAt local:dgi .
```

Requête SPARQL:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX local: <http://www.polymt.ca/vocab/>
CONSTRUCT {?n1 local:colleagueOf ?n2 . }
WHERE {?n1 local:worksAt ?w.
       ?n2 local:worksAt ?w . }
```

Forme
équivalente



```
:p1 local:colleagueOf :p1, :p2 .
:p2 local:colleagueOf :p2, :p1 .
```



SPARQL – Exemple


Base de données RDF:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix local: <http://www.polymt.ca/vocab/> .
:p1 foaf:name "Michel Gagnon";
    local:worksAt local:dgi;
    foaf:knows :p2 .
:p2 foaf:name "Michel Dagenais";
    local:worksAt local:dgi .
```

Requête SPARQL:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX local: <http://www.polymt.ca/vocab/>
CONSTRUCT {?n1 local:colleagueOf ?n2 . }
WHERE {?n1 local:worksAt ?w.
       ?n2 local:worksAt ?w .
       FILTER(?n1 != ?n2)}
```

Pour éviter les redondances



SPARQL – Exemple

Base de données RDF:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix local: <http://www.polymt.ca/vocab/> .
:p1 foaf:name "Michel Gagnon";
    local:worksAt local:dgi;
    foaf:knows :p2 .
:p2 foaf:name "Michel Dagenais";
    local:worksAt local:dgi .
```

Requête SPARQL:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX local: <http://www.polymt.ca/vocab/>
CONSTRUCT {?n1 local:colleagueOf ?n2 . }
WHERE {?n1 local:worksAt ?w.
       ?n2 local:worksAt ?w .
       FILTER(?n1 != ?n2)}
:p1 local:colleagueOf :p2 .
:p2 local:colleagueOf :p1 .
```





Requêtes sur DBpedia

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?p ?o WHERE {
  [] foaf:name "Barack Obama"@en ;
    ?p ?o .
}
LIMIT 100
```

Les requêtes sur Dbpedia peuvent être faites grâce à l'éditeur en ligne suivant : <https://dbpedia.org/sparql>



Requêtes sur DBpedia

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?p ?o WHERE {
  [] foaf:name "Barack Obama"@en ;
    ?p ?o .
}
LIMIT 100
```

On cherche tous les triplets qui ont comme sujet quelqu'un dont le nom est « Barack Obama » (en anglais). On extrait, pour chaque triplet, sa propriété et sa valeur.



Requêtes sur DBpedia

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>  
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>  
PREFIX dbo: <http://dbpedia.org/ontology/>
```

```
SELECT DISTINCT ?s ?b WHERE {  
  ?s a foaf:Person;  
    foaf:name ?name ;  
    dbo:birthPlace ?b .  
  FILTER contains(?name,"Obama")  
}
```



Requêtes sur DBpedia

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>
```

```
SELECT DISTINCT ?s ?b WHERE {
  ?s a foaf:Person;
     foaf:name ?name ;
     dbo:birthPlace ?b .
  FILTER contains(?name,"Obama")
}
```


On recherche toutes personnes dont le nom contient « Obama », ainsi que leur lieu de naissance.



Requêtes sur DBpedia

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>

SELECT DISTINCT ?s ?b ?y WHERE {
  ?s a foaf:Person;
     foaf:name ?name ;
     dbo:birthDate ?y ;
     dbo:birthPlace ?b .
  FILTER contains(?name,"Obama")
}
LIMIT 100
```



Requêtes sur DBpedia

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>

SELECT DISTINCT ?s ?b ?y WHERE {
  ?s a foaf:Person;
    foaf:name ?name ;
    dbo:birthDate ?y ;
    dbo:birthPlace ?b .
  FILTER contains(?name,"Obama")
}
LIMIT 100
```

On extrait les URI de tous ceux dont le nom contient « Obama », ainsi que leur lieu et date de naissance.



Requêtes sur DBpedia

Les PREFIX ne sont pas obligatoires, mais réduisent la taille des requêtes. Sans PREFIX, les < > sont obligatoires


```
PREFIX dbo: <http://dbpedia.org/ontology/>
```

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

```
SELECT DISTINCT ?c ?pc WHERE {  
  dbr:Canada dbo:capital ?c .  
  ?c dbo:postalCode ?pc.  
}
```

VS

```
SELECT DISTINCT ?c ?pc WHERE {  
  <http://dbpedia.org/resource/Canada> <http://dbpedia.org/ontology/capital> ?c .  
  ?c <http://dbpedia.org/ontology/postalCode> ?pc.  
}
```






Requêtes sur DBpedia

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>

SELECT DISTINCT ?p ?o WHERE {
    <http://dbpedia.org/resource/SPARQL> ?p ?o
    FILTER (!isliteral(?o) &&
           ?p not in (rdf:type, owl:sameAs, dbo:wikiPageExternalLink))
}
LIMIT 100
```



Requêtes sur DBpedia

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>

SELECT DISTINCT ?p ?o WHERE {
  <http://dbpedia.org/resource/SPARQL> ?p ?o
  FILTER (!isliteral(?o) &&
    (?o rdfs:type dbo:WikiPageExternalLink))
}
LIMIT 10
```


Recherche tous les triplets ayant comme sujet l'URI suivante:
<http://dbpedia.org/resource/SPARQL>.
On extrait les propriétés et leurs valeurs,
en excluant celles dont la valeur est un littéral et les trois propriétés
suivantes: rdf:type, owl:sameAs et dbo:wikiPageExternalLink.



Requêtes sur DBpedia

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?s WHERE {
    ?s rdfs:label ?l .
    FILTER( contains(?l, "Computer") )
}
LIMIT 100
```



Requêtes sur DBpedia

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?s WHERE {
  ?s rdfs:label ?l .
  FILTER( contains(?l, "Computer") )
}
LIMIT 100
```


On cherche les URI qui sont associées
à un label qui contient "Computer"



Requêtes sur DBpedia

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?s WHERE {
    ?s ?p ?o
    FILTER( contains(str(?s), "Computer") )
}
LIMIT 100
```



Requêtes sur DBpedia

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?s WHERE {
  ?s ?p ?o
  FILTER( contains(str(?s), "Computer") )
}
LIMIT 100
```

Ici, c'est l'URI qui doit contenir "Computer".
À noter qu'il faut convertir l'URI en chaîne de caractères avant de tester.

