

Hala Skaf-Molli
Associate Professor
Nantes University
<http://pagesperso.lina.univ-nantes.fr/~skaf-h/>

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

SPARQL 1.1 Update Language

- **Graph Update**
 - addition and removal of triples from some graphs within the **Graph Store**.
- **Graph Management**
 - creating and deletion of graphs within the **Graph Store**

Graph Store

- A mutable container of RDF graphs
- It has one (unnamed) slot holding a *default graph* and zero or more named slots holding *named graphs*. GS can be viewed as a mutable RDF DataSet.
- $GS = \{DG, (iri_1, G_1), \dots, (iri_n, G_n)\}$ where
 - the default graph DG is the RDF graph associated with the unnamed slot
 - $n \geq 0$ and for each $1 \leq i \leq n$, G_i is an RDF graph associated with the named slot identified by IRI iri_i
 - all IRIs are distinct, i.e., $i \neq j$ implies $iri_i \neq iri_j$

Abstract Update Operation

- An Update Operation Op is an **atomic** operation that accepts some arguments Args and transforms a Graph Store GS to another Graph Store GS', denoted as
- $Op(GS, Args) = GS'$
- An Update Operation can create new slots and new RDF graphs, or can remove existing slots and the corresponding graphs. It can also alter the state of each slot individually.

Graph Update Operations

- **Insert Data operation**
- **Delete Data operation**
- **Delete Insert operation**
- **Load operation**
- **Clear operation**

INSERT DATA

- An Update Operation in which new triples, given as a **(ground)** *QuadPattern*, are added in the Graph Store GS, in the default slot or in named.

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

<http://example/book1> ns:price 42 .
```

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
INSERT DATA
{
  <http://example/book1> dc:title "A new book" ;
                        dc:creator "A.N.Other" .
}
```

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

<http://example/book1> ns:price 42 .
<http://example/book1> dc:title "A new book" .
<http://example/book1> dc:creator "A.N.Other" .
```

INSERT DATA into Named Graph

```
# Graph: http://example/bookStore
@prefix dc: <http://purl.org/dc/elements/1.1/> .
<http://example/book1> dc:title "Fundamentals of Compiler Design" .
```

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX ns: <http://example.org/ns#>
INSERT DATA
{ GRAPH <http://example/bookStore> { <http://example/book1> ns:price 42 } }
```

```
# Graph: http://example/bookStore
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix ns: <http://example.org/ns#> .
<http://example/book1> dc:title "Fundamentals of Compiler Design" .
<http://example/book1> ns:price 42 .
```

DELETE DATA

An Update Operation in which triples, given as a (ground) *QuadPattern*, are removed from the Graph Store GS, from the default slot or from named slots.

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

<http://example/book2> ns:price 42 .
<http://example/book2> dc:title "David Copperfield" .
<http://example/book2> dc:creator "Edmund Wells" .
```

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

DELETE DATA
{
  <http://example/book2> dc:title "David Copperfield" ;
                        dc:creator "Edmund Wells" .
}
```

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

<http://example/book2> ns:price 42 .
```


DELETE/INSERT

An Update Operation in which :

1. Triples are deleted from the GS, either from the default slot or from named slot(s), and then
2. New triples are added in the GS, either in the default slot or in named slot(s).

DELETE/INSERT

Groups of triples to be deleted and groups of triples to be added. The specification of the triples is based on query patterns.

Example: rename all people with the given name "Bill" to "William".

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

<http://example/president25> foaf:givenName "Bill" .
<http://example/president25> foaf:familyName "McKinley" .
<http://example/president27> foaf:givenName "Bill" .
<http://example/president27> foaf:familyName "Taft" .
<http://example/president42> foaf:givenName "Bill" .
<http://example/president42> foaf:familyName "Clinton" .
```

DELETE/INSERT

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

```
WITH <http://example/addresses>
```

```
DELETE { ?person foaf:givenName 'Bill' }
```

```
INSERT { ?person foaf:givenName 'William' }
```

```
WHERE
```

```
{ ?person foaf:givenName 'Bill' }
```

(1)

Insert and delete applied
to the graph specified by
WITH

(3)

Results of (2) used to
instantiate DELETE

(4)

Results of (2) used to
instantiate INSERT

(2)

Pattern is evaluated

DELETE/INSERT

Example: rename all people with the given name "Bill" to "William".

Before:

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

<http://example/president25> foaf:givenName "Bill" .
<http://example/president25> foaf:familyName "McKinley" .
<http://example/president27> foaf:givenName "Bill" .
<http://example/president27> foaf:familyName "Taft" .
<http://example/president42> foaf:givenName "Bill" .
<http://example/president42> foaf:familyName "Clinton" .
```

After:

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

<http://example/president25> foaf:givenName "William" .
<http://example/president25> foaf:familyName "McKinley" .
<http://example/president27> foaf:givenName "William" .
<http://example/president27> foaf:familyName "Taft" .
<http://example/president42> foaf:givenName "William" .
<http://example/president42> foaf:familyName "Clinton" .
```

Exercise (1)

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

DELETE
{ ?book ?p ?v }
WHERE
{ ?book dc:date ?date .
  FILTER ( ?date > "1970-01-01T00:00:00-02:00"^^xsd:dateTime )
  ?book ?p ?v
}
```

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

<http://example/book1> dc:title "Principles of Compiler Design" .
<http://example/book1> dc:date "1977-01-01T00:00:00-02:00"^^xsd:dateTime .

<http://example/book2> ns:price 42 .
<http://example/book2> dc:title "David Copperfield" .
<http://example/book2> dc:creator "Edmund Wells" .
<http://example/book2> dc:date "1948-01-01T00:00:00-02:00"^^xsd:dateTime .

<http://example/book3> dc:title "SPARQL 1.1 Tutorial" .
```

Exercise (2)

```
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 > "1970-01-01T00:00:00-02:00"^^xsd:dateTime )
    ?book ?p ?v
  }
}
```

Data before:

```
# Graph: http://example/bookStore
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://example/book1> dc:title "Fundamentals of Compiler Design" .
<http://example/book1> dc:date "1977-01-01T00:00:00-02:00"^^xsd:dateTime .

<http://example/book2> ns:price 42 .
<http://example/book2> dc:title "David Copperfield" .
<http://example/book2> dc:creator "Edmund Wells" .
<http://example/book2> dc:date "1948-01-01T00:00:00-02:00"^^xsd:dateTime .

<http://example/book3> dc:title "SPARQL 1.1 Tutorial" .
```

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

<http://example/book4> dc:title "SPARQL 1.0 Tutorial" .
```

Exercise 3

- Write a SPARQL query that copies triples of name (*foaf:name*) and email (*foaf:mbox*) from one named graph(<http://example/people>) to another (<http://example/addresses>). Some individuals may not have an address, but the name is copied regardless.

Exercise (3bis)

- Apply the previous query to:

```
# Graph: http://example/people
@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         "Alice" .
_:a  foaf:mbox         <mailto:alice@example.com> .

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

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

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

_:a  foaf:name         "Alice" .
_:a  foaf:mbox         <mailto:alice@example.com> .

_:b  foaf:name         "Bob" .
```


Exercise (4)

- Write a request to removes all statements about anything with a given name (foaf:givenName) of "Fred" from the graph <http://example/addresses>.
- Apply your request to:

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

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

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

Exercise (5)

- Write a request to removes all statements about anything with a given name (foaf:givenName) of "Fred" from the default graph
- Apply your request to:

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

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

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

Exercise (6)

- Write a request to removes all statements about anything with a given name (foaf:givenName) of "Fred" from the default graph
- Apply your request to:

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

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

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

Graph Update Operations

- **Insert Data operation**
- **Delete Data operation**
- **Delete Insert operation**
- **Load operation** is an update Operation in which new triples (from a remote graph) are added in the GS, either in the default slot or in a named slot, if
- **Clear operation** is an update Operation in which triples are deleted from the GS, either from a named slot, the default slot, all named slots or all slots.

SPARQL 1.1 Update Language

- **Graph Update**
 - addition and removal of triples from some graphs within the **Graph Store**.
- **Graph Management**
 - creating and deletion of graphs within the **Graph Store**

Graph Management

- **CREATE** is an update Operation in which
 - a new named slot and
 - a new graph G are created in the GS. The new graph is held in the new slot, and is empty. Other slots and graphs are not affected.
- **DROP** is an Update Operation in which
 - one or more slots (a named slot IRI_i , the default slot, all named slots or all slots) and their corresponding graphs are removed from the GS.

Graph Management

- Copy
 - inserting all data from an input graph into a destination graph.
 - Data from the input graph is not affected, but data from the destination graph, if any, is removed before insertion.
- Move
 - moving all data from an input graph into a destination graph.
 - The input graph is removed after insertion and data from the destination graph, if any, is removed before insertion.
- Add
 - inserting all data from an input graph into a destination graph.
 - Data from the input graph is not affected, and initial data from the destination graph, if any, is kept intact.

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

Data before:

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

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

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

<http://example/fred> a foaf:Person .
<http://example/fred> foaf:givenName "Fred" .
```

Data after:

```
# Default graph
```

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

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

Note that the original content in `http://example.org/named` is lost by a `MOVE` operation.

Example

- Adds all statements from the default graph to a named graph.

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

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

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

<http://example/fred> a foaf:Person .
```

Synthesis

- **Graph Update** : Insert Data, Delete Data, Insert/Delete, Load, Clear
- **Graph Management**
 - create, drop, copy, move, add

SPARQL 1.1 Federated Query

- **Simple query to a remote SPARQL endpoint**

- Remote source

`http://people.example.org`

`@prefix foaf:`

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

`@prefix : <http://example.org/> .`

`:people15 foaf:name "Alice" .`

`:people16 foaf:name "Bob" .`

`:people17 foaf:name "Charles" .`

`:people18 foaf:name "Daisy" .`

- Local data:

`http://example.org/myfoaf.rdf`

`<http://example.org/myfoaf/I>`

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

`<http://example.org/people15> .`

SPARQL 1.1 Federated Query

Remote source <<http://people.example.org>>

@prefix foaf: <http://xmlns.com/foaf/0.1/> .

@prefix : <http://example.org/> .

:people15 foaf:name "Alice" .

:people16 foaf:name "Bob" .

:people17 foaf:name "Charles" .

:people18 foaf:name "Daisy" .

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

SELECT ?name

FROM <http://example.org/myfoaf.rdf>

WHERE

{

<http://example.org/myfoaf/I> foaf:knows ?person

SERVICE <http://people.example.org/sparql> {

?person foaf:name ?name . }

}

Local data: <http://example.org/myfoaf.rdf>

<http://example.org/myfoaf/I>

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

<http://example.org/people15> .

name

Alice

Federating SPARQL Queries Across Government Linked Data

```
select ?x ?name ?districtname
where {
  ?x a <http://environment.data.gov.uk/def/bathing-water/BathingWater> .
  ?x <http://www.w3.org/2000/01/rdf-schema#label> ?name .
  ?x <http://statistics.data.gov.uk/def/administrative-geography/district> ?district .

  SERVICE <http://data.ordnancesurvey.co.uk/datasets/boundary-line/apis/sparql>
  {
    ?district <http://data.ordnancesurvey.co.uk/ontology/spatialrelations/within>
      <http://data.ordnancesurvey.co.uk/id/7000000000041421> .
    ?district <http://www.w3.org/2000/01/rdf-schema#label> ?districtname .}
  }
order by ?districtname
```

Try this in SQL