



# Nested Queries in SPARQL

Renzo Angles    Claudio Gutierrez

Presented by: Nuno Lopes



## Query nesting enables:

- composition / reuse of queries
- rewriting / distributed execution
- optimization

## Query nesting enables:

- composition / reuse of queries
- rewriting / distributed execution
- optimization

## Approaches for nesting

SQL: allows nesting in FROM and WHERE clauses

XQuery: (arbitrary) nesting of expressions

## Query nesting enables:

- composition / reuse of queries
- rewriting / distributed execution
- optimization

## Approaches for nesting

SQL: allows nesting in FROM and WHERE clauses

XQuery: (arbitrary) nesting of expressions

## foaf.ttl

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

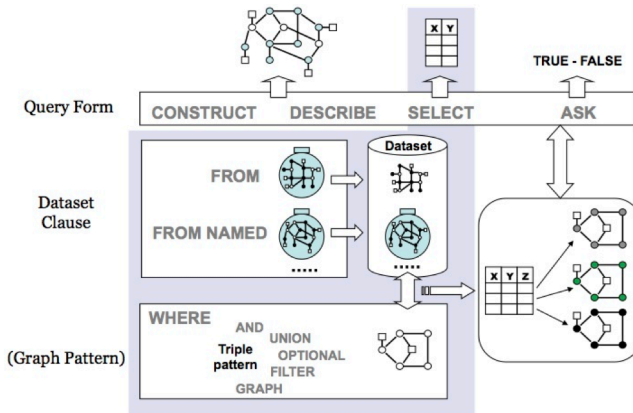
<http://nunolopes.org/foaf.rdf#me> a foaf:Person;
    foaf:name "Nuno Lopes";
    foaf:mbox <mailto:nuno.lopes@deri.org>;
    foaf:nick "nl","nunolopes","nlopes";
    foaf:age 30;
    foaf:knows <http://www.polleres.net/foaf.rdf#me> .

<http://www.polleres.net/foaf.rdf#me> a foaf:Person;
    foaf:name "Axel Polleres";
    foaf:nick "droxel","xel","xl";
    foaf:mbox <mailto:axel.polleres@deri.org>;
    foaf:givenName "Axel";
    foaf:familyName "Polleres".
```

## bib.ttl

```
@prefix bibtex: <http://zeitkunst.org/bibtex/0.1/bibtex.owl#> .

<http://example.org/bibtex#xsparql> a bibtex:Article ;
  bibtex:hasJournal "DERI Journal" ;
  bibtex:hasTitle "XSPARQL" ;
  bibtex:hasAuthor <http://nunolopes.org/foaf.rdf#me> ;
  bibtex:hasAuthor <http://www.polleres.net/foaf.rdf#me> .
```



## Names of people age 18–30 or names having (or not) an email

```
SELECT DISTINCT ?Name
FROM <foaf.ttl>
WHERE {
  {
    { ?X foaf:name ?Name . ?X foaf:age ?Age }
    FILTER (?Age >= 18 && ?Age <= 30)
  }
  UNION {
    { ?X foaf:name ?Name } OPTIONAL { ?X foaf:mbox ?Email }
  }
}
```



## Names of people age 18–30 or names having (or not) an email

```
SELECT DISTINCT ?Name
FROM <foaf.ttl>
WHERE {
  {
    { ?X foaf:name ?Name . ?X foaf:age ?Age }
    FILTER (?Age >= 18 && ?Age <= 30)
  }
  UNION {
    { ?X foaf:name ?Name } OPTIONAL { ?X foaf:mbox ?Email }
  }
}
```

## Names of people age 18–30 or names having (or not) an email

```
SELECT DISTINCT ?Name
FROM <foaf.ttl>
WHERE {
  {
    { ?X foaf:name ?Name . ?X foaf:age ?Age }
    FILTER (?Age >= 18 && ?Age <= 30)
  }
  UNION {
    { ?X foaf:name ?Name } OPTIONAL { ?X foaf:mbox ?Email }
  }
}
```

## Names of people age 18–30 or names having (or not) an email

```
SELECT DISTINCT ?Name
FROM <foaf.ttl>
WHERE {
  {
    { ?X foaf:name ?Name . ?X foaf:age ?Age }
    FILTER (?Age >= 18 && ?Age <= 30)
  }
  UNION {
    { ?X foaf:name ?Name } OPTIONAL { ?X foaf:mbox ?Email }
  }
}
```

## Names of people age 18–30 or names having (or not) an email

```
SELECT DISTINCT ?Name
FROM <foaf.ttl>
WHERE {
  {
    { ?X foaf:name ?Name . ?X foaf:age ?Age }
    FILTER (?Age >= 18 && ?Age <= 30)
  }
  UNION {
    { ?X foaf:name ?Name } OPTIONAL { ?X foaf:mbox ?Email }
  }
}
```

```
{ ?X foaf:name ?Name . ?X foaf:age ?Age }
```

```
{ ?X foaf:name ?Name . ?X foaf:age ?Age }
```

X	Name
<a href="http://nunolopes.org/foaf.rdf#me">http://nunolopes.org/foaf.rdf#me</a>	"Nuno Lopes"
<a href="http://www.polleres.net/foaf.rdf#me">http://www.polleres.net/foaf.rdf#me</a>	"Axel Polleres"

```
{ ?X foaf:name ?Name . ?X foaf:age ?Age }
```

X	Name
<code>&lt;http://nunolopes.org/foaf.rdf#me&gt;</code>	"Nuno Lopes"
<code>&lt;http://www.polleres.net/foaf.rdf#me&gt;</code>	"Axel Polleres"

X	Age
<code>&lt;http://nunolopes.org/foaf.rdf#me&gt;</code>	30

```
{ ?X foaf:name ?Name . ?X foaf:age ?Age }
```

X	Name
<a href="http://nunolopes.org/foaf.rdf#me">http://nunolopes.org/foaf.rdf#me</a>	"Nuno Lopes"
<a href="http://www.polleres.net/foaf.rdf#me">http://www.polleres.net/foaf.rdf#me</a>	"Axel Polleres"

⊗

X	Age
<a href="http://nunolopes.org/foaf.rdf#me">http://nunolopes.org/foaf.rdf#me</a>	30



```
{ ?X foaf:name ?Name . ?X foaf:age ?Age }
```

X	Name
<code>&lt;http://nunolopes.org/foaf.rdf#me&gt;</code>	"Nuno Lopes"
<code>&lt;http://www.polleres.net/foaf.rdf#me&gt;</code>	"Axel Polleres"

⊗

X	Age
<code>&lt;http://nunolopes.org/foaf.rdf#me&gt;</code>	30

=

X	Name	Age
<code>&lt;http://nunolopes.org/foaf.rdf#me&gt;</code>	"Nuno Lopes"	30

## Noteworthy new features in SPARQL 1.1

- Aggregates
- Subqueries
- Federation Extensions
- Negation
- Expressions in the SELECT clause
- Property Paths
- Assignment

## Noteworthy new features in SPARQL 1.1

- Aggregates
- **Subqueries**
- Federation Extensions
- Negation
- Expressions in the SELECT clause
- Property Paths
- Assignment

## Noteworthy new features in SPARQL 1.1

- Aggregates
- Subqueries
- Federation Extensions
- Negation
- Expressions in the SELECT clause
- Property Paths
- Assignment

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

SELECT ?y ?minNick
WHERE {
  <http://nunolopes.org/foaf.rdf#me> foaf:knows ?y . {
    SELECT ?y (MIN(?nick) AS ?minNick)
    WHERE { ?y foaf:nick ?nick .
    } GROUP BY ?y
  }
}
```

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

SELECT ?y ?minNick
WHERE {
  <http://nunolopes.org/foaf.rdf#me> foaf:knows ?y . {
    SELECT ?y (MIN(?nick) AS ?minNick)
    WHERE { ?y foaf:nick ?nick .
    } GROUP BY ?y
  }
}
```

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

SELECT ?y ?minNick
WHERE {
  <http://nunolopes.org/foaf.rdf#me> foaf:knows ?y . {
    SELECT ?y (MIN(?nick) AS ?minNick)
    WHERE { ?y foaf:nick ?nick .
    } GROUP BY ?y
  }
}
```

y	minNick
<nunolopes.org/foaf.rdf#me>	"nl"
<polleres.net/foaf.rdf#me>	"droxel"

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

SELECT ?y ?minNick
WHERE {
  <http://nunolopes.org/foaf.rdf#me> foaf:knows ?y . {
    SELECT ?y (MIN(?nick) AS ?minNick)
    WHERE { ?y foaf:nick ?nick .
    } GROUP BY ?y
  }
}
```

y	minNick
<nunolopes.org/foaf.rdf#me>	"nl"
<polleres.net/foaf.rdf#me>	"droxel"



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

SELECT ?y ?minNick
WHERE {
  <http://nunolopes.org/foaf.rdf#me> foaf:knows ?y . {
    SELECT ?y (MIN(?nick) AS ?minNick)
    WHERE { ?y foaf:nick ?nick .
    } GROUP BY ?y
  }
}
```

y	minNick
<nunolopes.org/foaf.rdf#me>	"nl"
<polleres.net/foaf.rdf#me>	"droxel"

y
<polleres.net/foaf.rdf#me>

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

SELECT ?y ?minNick
WHERE {
  <http://nunolopes.org/foaf.rdf#me> foaf:knows ?y . {
    SELECT ?y (MIN(?nick) AS ?minNick)
    WHERE { ?y foaf:nick ?nick .
    } GROUP BY ?y
  }
}
```

y	minNick
<nunolopes.org/foaf.rdf#me>	"nl"
<polleres.net/foaf.rdf#me>	"droxel"

y
<polleres.net/foaf.rdf#me>

y	minNick
<polleres.net/foaf.rdf#me>	"droxel"

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

SELECT ?y ?minNick
WHERE {
  <http://nunolopes.org/foaf.rdf#me> foaf:knows ?y . {
    SELECT ?y (MIN(?nick) AS ?minNick)
    WHERE { ?y foaf:nick ?nick .
    } GROUP BY ?y
  }
}
```

**FROM and FROM NAMED are not allowed in subqueries!**

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

SELECT ?name ?bDay
WHERE {
  SERVICE <http://example.org/foaf> {
    ?uri foaf:name      ?name .
  }
  SERVICE <http://example.org/birthdays> {
    ?uri foaf:birthday  ?bDay .
  }
}
```

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

SELECT ?name ?bDay
WHERE {
  SERVICE <http://example.org/foaf> {
    ?uri foaf:name      ?name .
  }
  SERVICE <http://example.org/birthdays> {
    ?uri foaf:birthday  ?bDay .
  }
}
```

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

SELECT ?name ?bDay
WHERE {
  SERVICE <http://example.org/foaf> {
    ?uri foaf:name      ?name .
  }
  SERVICE <http://example.org/birthdays> {
    ?uri foaf:birthday  ?bDay .
  }
}
```

Execution method is undefined

```
SELECT ?name ?bDay
WHERE {
  SERVICE <http://example.org/foaf> {
    ?uri foaf:name      ?name . }
  SERVICE <http://example.org/birthdays> {
    ?uri foaf:birthday  ?bDay . }
} BINDINGS ?uri {
  (<http://nunolopes.org/foaf.rdf#me>)
  (<http://www.polleres.net/foaf.rdf#me>) }
```

```
SELECT ?name ?bDay
WHERE {
  SERVICE <http://example.org/foaf> {
    ?uri foaf:name      ?name .  }
  SERVICE <http://example.org/birthdays> {
    ?uri foaf:birthday  ?bDay .  }
} BINDINGS ?uri {
  (<http://nunolopes.org/foaf.rdf#me>)
  (<http://www.polleres.net/foaf.rdf#me>) }
```

- allows to specify filters for a query without changing the graph pattern (good for reuse of queries)
- does not allow to specify values that are taken from another SPARQL query



## Negation

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

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

```
SELECT ?nick
```

```
WHERE { ?person foaf:nick ?nick .
```

```
    FILTER NOT EXISTS { ?uri foaf:nick "nl" }
```

```
}
```

## Negation

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

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

```
SELECT ?nick
```

```
WHERE { ?person foaf:nick ?nick .
```

```
    FILTER NOT EXISTS { ?uri foaf:nick "nl" }
```

```
}
```

unbound  
filters remove  
all solutions!

## Negation

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

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

```
SELECT ?nick
```

```
WHERE { ?person foaf:nick ?nick .
```

```
    FILTER NOT EXISTS { ?uri foaf:nick "nl" }
```

```
}
```

EXISTS filter  
also available

## Negation

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

SELECT ?nick
WHERE { ?person foaf:nick ?nick .
       FILTER NOT EXISTS { ?uri foaf:nick "nl" }
}
```

## Removing solutions

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

SELECT DISTINCT ?name
WHERE { ?person foaf:name ?name.
       MINUS { ?person foaf:givenName "Axel". }
}
```

## Negation

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

SELECT ?nick
WHERE { ?person foaf:nick ?nick .
       FILTER NOT EXISTS { ?uri foaf:nick "nl" }
}
```

## Removing solutions

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

SELECT DISTINCT ?name
WHERE { ?person foaf:name ?name.
       MINUS { ?person foaf:givenName "Axel". }
}
```

Removes compatible solutions: requires shared variables!

## Creating new values

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

SELECT ?person (CONCAT(?given, " ", ?family) AS ?fullName)
FROM <foaf.ttl>
WHERE { ?person foaf:givenName ?given;
          foaf:familyName ?family
        }
```

## Types of nesting

*SPARQL<sub>From</sub>*: Subqueries in dataset clauses

*SPARQL<sub>SubS</sub>*: Subqueries as graph patterns.

*SPARQL<sub>Filter</sub>*: Subqueries in filter constraints.

## Query in the FROM clause

```
SELECT ?M1 ?M2
FROM <foaf.ttl>
FROM ( CONSTRUCT { ?A1 co-author ?A2 }
      FROM <bib.ttl>
      WHERE { ?Pub author ?A1 . ?Pub author ?A2 .
              FILTER ( !(?A1 = ?A2))}
      )
WHERE { ?P1 co-author ?P2 . ?P1 mail ?M1 . ?P2 mail ?M2 }
```



## Query in the FROM clause

```
SELECT ?M1 ?M2
FROM <foaf.ttl>
FROM ( CONSTRUCT { ?A1 co-author ?A2 }
      FROM <bib.ttl>
      WHERE { ?Pub author ?A1 . ?Pub author ?A2 .
              FILTER ( !(?A1 = ?A2))}
      )
WHERE { ?P1 co-author ?P2 . ?P1 mail ?M1 . ?P2 mail ?M2 }
```

- Natural extension: CONSTRUCT returns an RDF graph
- Side effect: adds construction of values

## Nested query in the graph pattern

```
SELECT ?Mail
FROM <foaf.ttl>
WHERE { {?X mail ?Mail} . {( SELECT DISTINCT ?X
                              FROM <bib.ttl>
                              WHERE {?Pub author ?X})} }
```

## Nested query in the graph pattern

```
SELECT ?Mail
FROM <foaf.ttl>
WHERE { {?X mail ?Mail} . {( SELECT DISTINCT ?X
                              FROM <bib.ttl>
                              WHERE {?Pub author ?X})} }
```

- projection of variables in graph patterns
- datasets in nested queries

## Nested queries in **FILTER** expressions

```
SELECT ?Name FROM <bib.ttl>
WHERE { ?Aut name ?Name . ?Pub author ?Aut . ?Pub conf ?Conf
        FILTER (?Conf IN ( SELECT ?ConfX
                            FROM <bib.ttl>
                            WHERE { ?ConfX series "ISWC" })))}
```

## Nested queries in **FILTER** expressions

```
SELECT ?Name FROM <bib.ttl>
WHERE { ?Aut name ?Name . ?Pub author ?Aut . ?Pub conf ?Conf
        FILTER (?Conf IN ( SELECT ?ConfX
                            FROM <bib.ttl>
                            WHERE { ?ConfX series "ISWC" })))}
```

- Duplicate elimination (without DISTINCT)
- Similar to subqueries in SQL: introduces IN, SOME, ALL and EXISTS operators

## XSPARQL follows a different semantics for nesting

- Based on XQuery nesting
- Variables are correlated
- Top-down evaluation

## Nesting in **FROM** clauses

```
prefix : <http://zeitkunst.org/bibtex/0.1/bibtex.owl#>
prefix foaf: <http://xmlns.com/foaf/0.1/>

let $ds := construct { $A1 :co-author $A2 }
           from <bib.ttl>
           where { $Pub :hasAuthor $A1 . $Pub :hasAuthor $A2 .
                  filter ( !($A1 = $A2))}

for $M1 $M2
from <foaf.ttl>
from $ds
where { $P1 :co-author $P2 . $P1 foaf:mbox $M1 .
        $P2 foaf:mbox $M2 }
return ($M1, $M2)
```

## Nesting in **FROM** clauses

```
prefix : <http://zeitkunst.org/bibtex/0.1/bibtex.owl#>
prefix foaf: <http://xmlns.com/foaf/0.1/>

let $ds := construct { $A1 :co-author $A2 }
         from <bib.ttl>
         where { $Pub :hasAuthor $A1 . $Pub :hasAuthor $A2 .
                 filter ( !($A1 = $A2))}

for $M1 $M2
from <foaf.ttl>
from $ds
where { $P1 :co-author $P2 . $P1 foaf:mbox $M1 .
        $P2 foaf:mbox $M2 }
return ($M1, $M2)
```



## Nesting in **WHERE** clauses

```
for distinct $X
from <bib.ttl>
where { $Pub author $X }
return
  for $Mail
  from <foaf.ttl>
  where { $X mail $Mail . }
  return $Mail
```

## SPARQL 1.1 SERVICE

```
PREFIX dbpedia2: <http://dbpedia.org/property/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?N ?MyB
FROM <http://polleres.net/foaf.rdf>
WHERE { [ foaf:birthday ?MyB ].

  SERVICE <http://dbpedia.org/sparql> {
    SELECT ?N WHERE { [ dbpedia2:born ?B; foaf:name ?N ]. }
  }
  FILTER ( Regex(Str(?B),str(?MyB)) )
}
```

## SPARQL 1.1 SERVICE

```
PREFIX dbpedia2: <http://dbpedia.org/property/>
```

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

```
SELECT ?N ?MyB
```

```
FROM <http://polleres.net/foaf.rdf>
```

```
WHERE { [ foaf:birthday ?MyB ].
```

```
SERVICE <http://dbpedia.org/sparql> {  
  SELECT ?N WHERE { [ dbpedia2:born ?B; foaf:name ?N ]. }  
}
```

```
FILTER ( Regex(Str(?B),str(?MyB)) )  
}
```

## SPARQL 1.1 SERVICE

```
PREFIX dbpedia2: <http://dbpedia.org/property/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?N ?MyB
FROM <http://polleres.net/foaf.rdf>
WHERE { [ foaf:birthday ?MyB ].

    SERVICE <http://dbpedia.org/sparql> {
        SELECT ?N WHERE { [ dbpedia2:born ?B; foaf:name ?N ]. }
    }
    FILTER ( Regex(Str(?B),str(?MyB)) )
}
```

## SPARQL 1.1 SERVICE

```
PREFIX dbpedia2: <http://dbpedia.org/property/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?N ?MyB
FROM <http://polleres.net/foaf.rdf>
WHERE { [ foaf:birthday ?MyB ].

  SERVICE <http://dbpedia.org/sparql> {
    SELECT ?N WHERE { [ dbpedia2:born ?B; foaf:name ?N ]. }
  }
  FILTER ( Regex(Str(?B), str(?MyB)) )
}
```

Result limits from SPARQL endpoints prevent this query from working!

## XSPARQL endpoint

```
prefix dbprop: <http://dbpedia.org/property/>
prefix foaf: <http://xmlns.com/foaf/0.1/>
prefix : <http://xsparql.deri.org/bday#>

let $MyB := for * from <http://polleres.net/foaf.rdf>
           where { [ foaf:birthday $B ]. }
           return $B

for * from <http://dbpedia.org/>
endpoint <http://dbpedia.org/sparql>
where { [ dbprop:born $B; foaf:name $N ].
        filter ( regex(str($B),str($MyB)) ) }
construct { :axel :sameBirthDayAs $N }
```

## XSPARQL endpoint

```
prefix dbprop: <http://dbpedia.org/property/>
prefix foaf: <http://xmlns.com/foaf/0.1/>
prefix : <http://xsparql.deri.org/bday#>

let $MyB := for * from <http://polleres.net/foaf.rdf>
  where { [ foaf:birthday $B ]. }
  return $B

for * from <http://dbpedia.org/>
endpoint <http://dbpedia.org/sparql>
where { [ dbprop:born $B; foaf:name $N ].
  filter ( regex(str($B),str($MyB)) ) }
construct { :axel :sameBirthDayAs $N }
```

- Query nesting is a much needed feature
- SPARQL 1.1's nesting is still quite limited (e.g. no injection of values)
- Other approaches for nesting can be considered
- *SPARQL<sub>Filter</sub>* is doable in XSPARQL but ugly! Possible (syntactic) extensions of nesting for XSPARQL