Linked EU Budget

Example queries explained

Document Metadata

Property	Value
Date	25 June 2018
Version	1.0
Authors	Brecht Wyns – PwC EU Services Emidio Stani – PwC EU Services
Reviewed by	Denis Navarre – Publications Office of the EU

This study was prepared for the ISA Programme by:

PwC EU Services

Disclaimer:

QUESTION 1

See https://joinup.ec.europa.eu/document/example-query-word-search-headings

The first thing in every query is to define the prefixes for the namespaces used.

```
PREFIX dcat: <http://www.w3.org/ns/dcat#>
PREFIX odp: <http://data.europa.eu/euodp/ontologies/ec-odp#>
PREFIX dc: <http://purl.org/dc/terms/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX bud: <http://data-test.opoce.cec.eu.int/euodp/bud#>
```

The next step is always to SELECT the variables of our query that we want displayed in the results.

In this case we:

1) use the str() function to select the string values of the variables;

2) use the replace() function in combination with regular expressions to clean out the values of some variables and format the correctly; and

3) use the xsd:string() function to transform certain variables to string types in order to apply the regular expressions.

```
SELECT

str(?TiAlias) AS ?TiAlias
str(?ChAlias) AS ?ChAlias
str(?ArAlias) AS ?ArAlias
str(?ItAlias) AS ?ItAlias
str(?ArHeading) AS ?ArHeading
str(?ItHeading) AS ?ItHeading


REPLACE(xsd:string(?A_Type), '.*[/#]', '') AS ?A_Type
REPLACE(REPLACE(xsd:string(?MV_Value), '.*[/#]', ''),' ','.') AS ?MV_Value
```

Optionally, we define which data set in a triple store we want to query. The data set is identified by a "graph IRI", as shown in the example below. The final graph IRI will be made public once the data is published on the EU Open Data Portal.

```
FROM <http://data-test.opoce.cec.eu.int/euodp/resource/4rx/ABB2018>
```

Finally we define the conditions of our query

Here we use the FILTER() and lang() functions to select only the headings that are in english.

```
WHERE {

 ?Title bud:heading ?TiHeading.
 FILTER (lang (?TiHeading) = 'en').
 ?Title bud:alias ?TiAlias.
 ?Title bud:hasChapter ?Chapter.
 ?Chapter bud:hasArticle ?Article.
 ?Chapter bud:hasArticle ?Article.
 ?Chapter bud:alias ?ChAlias.

 {
```

```
  ############ Articles #############

  {
       ?Article bud:alias ?ArAlias.
       ?Article bud:heading ?ArHeading.
       FILTER (lang (?ArHeading) = 'en').

       ?Article bud:hasAmount ?Amount.
       ?Amount bud:year ?A_Year.
       FILTER(str(?A_Year)="2018")

       ?Amount a ?A_Type.
       ?Amount bud:figure ?MonetaryValue.
       ?MonetaryValue bud:value ?MV_Value.
```

Here we use a FILTER() with the CONTAINS() and !CONTAINS() functions to select only the results whose headings contain the keywords "fisheries" and "aquacuture" and NOT the keyword "legal basis".

The lcase() function is used to transform the values to lowercase thus avoiding the need to check for every possible way they may be typed.

```
FILTER(CONTAINS(lcase(str(?ArHeading)),"fisheries") OR
CONTAINS(lcase(str(?ArHeading)),"aquaculture")).

FILTER(!CONTAINS(lcase(str(?ArHeading)),"legal basis") AND
!CONTAINS(lcase(str(?ArHeading)),"legal basis")).


  }
```

Here we use a UNION because we need to combine the results of 2 queries, for the articles and the items, under the same variables.

Notice that the folowing part of the query is the same as the previous, only adapted to return Items instead of Articles.

```
  UNION

  ############## ITEMS ###############
  {
       ?Article bud:alias ?ArAlias.
       ?Article bud:heading ?ArHeading.
       FILTER (lang (?ArHeading) = 'en').


       ?Article bud:hasItem ?Item.
       ?Item bud:alias ?ItAlias.
       ?Item bud:heading ?ItHeading.
       FILTER (lang (?ItHeading) = 'en').

       ?Item bud:hasAmount ?Amount.
       ?Amount bud:year ?A_Year.
       FILTER(str(?A_Year)="2018")


       ?Amount a ?A_Type.
       ?Amount bud:figure ?MonetaryValue.
       ?MonetaryValue bud:value ?MV_Value.


       FILTER(CONTAINS(lcase(str(?ItHeading)),"fisheries") OR
CONTAINS(lcase(str(?ItHeading)),"aquaculture") OR
CONTAINS(lcase(str(?ArHeading)),"fisheries") OR
CONTAINS(lcase(str(?ArHeading)),"aquaculture")).
```

```
       FILTER(!CONTAINS(lcase(str(?ItHeading)),"legal basis") AND
!CONTAINS(lcase(str(?ItHeading)),"legal basis") AND
!CONTAINS(lcase(str(?ArHeading)),"legal basis") AND
!CONTAINS(lcase(str(?ArHeading)),"legal basis")).
  }
 }
}
```

Finally we use the ORDER BY() function to order our results by the value of the ?TiAlias variable, and the LIMIT function to limit the results to 1000

```
ORDER BY(?TiAlias) (?ChAlias) (?ArAlias) (?ItAlias) (?A_Type) (?MV_Value)
LIMIT 1000
```

QUESTION 2

See https://joinup.ec.europa.eu/document/example-query-eu-programmes-related-budget-lines

```
PREFIX dcat: <http://www.w3.org/ns/dcat#>
PREFIX odp:  <http://data.europa.eu/euodp/ontologies/ec-odp#>
PREFIX dc: <http://purl.org/dc/terms/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX bud: <http://data-test.opoce.cec.eu.int/euodp/bud#>

SELECT
```

Here we create a dummy variable "?Results" with the text below to make the result more user friendly.

Then we use the sum() function to sum up all the values of the variable ?Amount in a single result.

Then, we ask to display the ?Programme variable to break down the sum according to it. Commenting out this line will result in a single number being returned.

The xsd:decimal() function is used in order to format the values of ?MV_Value as necessary to perform the calculations.

```
' Amounts spent on ' AS ?Total_of
REPLACE(REPLACE(xsd:string(?EU_Prog), '.*[/#]', ''),' ','.') AS ?Programme
sum(xsd:decimal(?MV_Value)) AS ?Amount
FROM <http://data-test.opoce.cec.eu.int/euodp/resource/4rx/ABB2018>
WHERE {
{
  ############ Articles #############
```

Here we FILTER our results for those associated with EURATOM, H2020 and ITER. This is done through the value of the bud:hasEuProgramme property.

```
  {
       ?Article a bud:Article.
       ?Article bud:hasEuProgramme ?EU_Prog.
       FILTER(CONTAINS(str(?EU_Prog),"EURATOM")   OR   CONTAINS(str(?EU_Prog),"ITER")   OR
CONTAINS(str(?EU_Prog),"H2020")).

       ?Article bud:hasAmount ?Amount.
```

Here we FILTER our results for results in 2016. This is done through the value of the bud:year property Amount class.

```
       ?Amount bud:year ?A_Year.
       FILTER(str(?A_Year)="2018")

       ?Amount a ?A_Type.
       FILTER(CONTAINS(str(?A_Type),"Commitment")                                    OR
CONTAINS(str(?A_Type),"NonDifferenciated")).

       ?Amount bud:figure ?MonetaryValue.
       ?MonetaryValue bud:value ?MV_Value.
```

Here we FILTER our results to remove the budget lines that have closed. This is done by filtering out the values containing "p.m."

```
       FILTER(str(?MV_Value)!='p.m.').
  }
  UNION
  ############## ITEMS ###############
```

```
  {
       ?Article bud:hasItem ?Item.
       ?Item bud:hasEuProgramme ?EU_Prog.
       FILTER(CONTAINS(str(?EU_Prog),"EURATOM")   OR   CONTAINS(str(?EU_Prog),"ITER")   OR
CONTAINS(str(?EU_Prog),"H2020")).

       ?Item bud:hasAmount ?Amount.
       ?Amount bud:year ?A_Year.
       FILTER(str(?A_Year)="2018")

       ?Amount a ?A_Type.
       FILTER(CONTAINS(str(?A_Type),"Commitment")                                          OR
CONTAINS(str(?A_Type),"NonDifferenciated")).

       ?Amount bud:figure ?MonetaryValue.
       ?MonetaryValue bud:value ?MV_Value.
       FILTER(str(?MV_Value)!='p.m.').
  }
}
}
ORDER BY (?Programme)
LIMIT 10
```

QUESTION 3

See https://joinup.ec.europa.eu/node/701321

```
PREFIX dcat: <http://www.w3.org/ns/dcat#>
PREFIX odp:  <http://data.europa.eu/euodp/ontologies/ec-odp#>
PREFIX dc: <http://purl.org/dc/terms/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX bud: <http://data-test.opoce.cec.eu.int/euodp/bud#>
```

For this query we need to sum all the Commitments and Non-Differantiated Payments under each Political Category. In this case we can conveniently use the Political Category code of each amount, which has its own property, to break down the sum.

So, we use the sum() function to sum all the amounts, and add to the results the Political Category code to break them down.

```
SELECT
' Amounts spent on ' AS ?Total_of
REPLACE(REPLACE(xsd:string(?Catpol), '.*[/#]', ''),' ','.') AS ?Political_Category
sum(xsd:decimal(?MV_Value)) AS ?Amount
FROM <http://data-test.opoce.cec.eu.int/euodp/resource/4rx/ABB2018>
WHERE {
 {
  ############ Articles #############
  {
       ?Article a bud:Article.
       ?Article bud:hasAmount ?Amount.
       ?Amount bud:year ?A_Year.
       FILTER(str(?A_Year)="2018")

       ?Amount a ?A_Type.
       FILTER(CONTAINS(str(?A_Type),"Commitment")                                    OR
CONTAINS(str(?A_Type),"NonDifferenciated")).

       ?Amount bud:figure ?MonetaryValue.
       ?MonetaryValue bud:value ?MV_Value.
       FILTER(str(?MV_Value)!='p.m.').

       ?Amount bud:hasPoliticalCategory ?Catpol.
  }
  UNION
  ############## ITEMS ###############
  {
       ?Item a bud:Item.
       ?Item bud:hasAmount ?Amount.
       ?Amount bud:year ?A_Year.
       FILTER(str(?A_Year)="2018").

       ?Amount a ?A_Type.
       FILTER(CONTAINS(str(?A_Type),"Commitment")                                    OR
CONTAINS(str(?A_Type),"NonDifferenciated")).

       ?Amount bud:figure ?MonetaryValue.
       ?MonetaryValue bud:value ?MV_Value.
       FILTER(str(?MV_Value)!='p.m.').

       ?Amount bud:hasPoliticalCategory ?Catpol.
  }
 }
}
ORDER BY ?Political_Category
```

```
LIMIT 10000
```

QUESTION 4

See https://joinup.ec.europa.eu/document/example-query-filtering-amounts

```
PREFIX dcat: <http://www.w3.org/ns/dcat#>
PREFIX odp:  <http://data.europa.eu/euodp/ontologies/ec-odp#>
PREFIX dc: <http://purl.org/dc/terms/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX bud: <http://data-test.opoce.cec.eu.int/euodp/bud#>
```

For this query we need to obtain all the Commitments and Non-Differentiated Payments, then sum them up to the level of Article, and after summing them up filter out the ones that amount to less than 100.000.000.

To do this a somewhat complex structure is required, which also helps illustrate some useful capabilities of SPARQL. We will use a Nested SPARQL Query.

This means including a whole SELECT-WHERE construct within the WHERE of another query. Essentially the inner construct works exactly like all we've seen before, and the outer WHERE can enact on, and only on, the results returned by the SELECT nested within it. This allows us to further work on the results of that SELECT. In a normal query the possiblities of doing that are very limited.

In our case we will use this to filter on the sums returned by the nested query. This would have been impossible with a different structure because FILTER functions are limited to use inside the WHERE and the sums we need to filter on are only created in the SELECT phase.

Furthermore this will allow us to format the numbers accordingly to calculate the sums, but at the same time format them in a human friendly way in another variable and display them only so.

==========

So!

Here in the outer SELECT we return the Article Alias, the Article Heading, the Amount Type, the Year, and the Amount format in two different ways, as "Amount" for human readability, and as "Amount_a" to perform calculations.

Here in the outer SELECT we return the Article Alias, the Article Heading, the Amount Type, the Year and finally the Amount, which is the one formatted for human consumption in the nested SELECT.

```
SELECT  ?ArAlias ?Article_Heading ?Amount ?Year ?A_Type
FROM <http://data-test.opoce.cec.eu.int/euodp/resource/4rx/ABB2018>
WHERE { {

SELECT
str(?ArAlias) AS ?ArAlias
str(?ArHeading) AS ?Article_Heading
REPLACE(xsd:string(?A_Type), '.*[/#]', '') AS ?A_Type
xsd:decimal(?MV_Value) AS ?Amount
sum(xsd:decimal(REPLACE(REPLACE(str(?MV_Value),' ',''),',','.'))) AS ?Amount_a
str(?A_Year) AS ?Year
WHERE
  {
            {
                 ?Article a bud:Article.
                 ?Article bud:alias ?ArAlias.
                 ?Article bud:heading ?ArHeading.
                 FILTER (lang (?ArHeading) = 'en').
```

```
                ?Article bud:hasAmount ?Amount.
                ?Amount bud:year ?A_Year.
                FILTER(str(?A_Year)="2018" OR str(?A_Year)="2017")

                ?Amount bud:hasPoliticalCategory ?A_CatPol.
                ?Amount a ?A_Type.
                ?Amount bud:figure ?MonetaryValue.
                ?MonetaryValue bud:value ?MV_Value.
                FILTER(REPLACE(xsd:string(?A_Type),   '.*[/#]',   '')="Commitment"   OR
REPLACE(xsd:string(?A_Type), '.*[/#]', '')="NonDifferenciated")
            }
            UNION
            {
                ?Article bud:alias ?ArAlias.
                ?Article bud:heading ?ArHeading.
                FILTER (lang (?ArHeading) = 'en').

                ?Article bud:hasItem ?Item.
                ?Item bud:alias ?ItAlias.
                ?Item bud:heading ?ItHeading.
                FILTER (lang (?ItHeading) = 'en').

                ?Item bud:hasAmount ?Amount.
                ?Amount bud:year ?A_Year.
                FILTER(str(?A_Year)="2018" OR str(?A_Year)="2017")

                ?Amount a ?A_Type.
                ?Amount bud:figure ?MonetaryValue.
                ?MonetaryValue bud:value ?MV_Value.
                FILTER(REPLACE(xsd:string(?A_Type),   '.*[/#]',   '')="Commitment"   OR
REPLACE(xsd:string(?A_Type), '.*[/#]', '')="NonDifferenciated")
            }
          }
} }
```

Here we GROUP our sums by the Article Alias, without this operation we would get a sum on the Article level for each Item below. So essentially we now sum the sums, which are already on an article level, because of the sum() function in the inner WHERE, and only then apply the HAVING function to filter out the Articles with less than 100.000.000.

```
GROUP BY(?ArAlias)
HAVING(?Amount_a >100000000)
ORDER BY(?ArAlias) (?Year) (str(?Amount))
```

QUESTION 4B

See https://joinup.ec.europa.eu/document/example-query-filtering-amounts

This query will be the same as before with the only change being modifying all instances of language filters (shown below) to a different language.

```
FILTER (lang (?ArHeading) = ‘nl’).

FILTER (lang (?ItHeading) = 'nl').
```

QUESTION 5

See https://joinup.ec.europa.eu/node/701323

This query is similar to question 4, though all amounts are added up and the results go back to 2016. In this case, they are added up at the level of Title, while in the previous question the level of Article was used.

```
PREFIX dcat: <http://www.w3.org/ns/dcat#>
PREFIX odp:  <http://data.europa.eu/euodp/ontologies/ec-odp#>
PREFIX dc: <http://purl.org/dc/terms/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX bud: <http://data-test.opoce.cec.eu.int/euodp/bud#>

SELECT   ?TiAlias CONCAT(?Title_Heading," - ", ?Year) AS ?Head  ?Amount_a


WHERE { {

SELECT
str(?TiAlias) AS ?TiAlias
str(?TiHeading) AS ?Title_Heading
sum(xsd:decimal(?MV_Value)) AS ?Amount_a
str(?A_Year) AS ?Year

WHERE
  {
             {
                  ?Title a bud:Title.
                  ?Title bud:alias ?TiAlias.
                  ?Title bud:heading ?TiHeading.
                  FILTER (lang (?TiHeading) = 'en').

                  ?Title bud:hasChapter ?Chapter.
                  ?Chapter bud:hasArticle ?Article.
                  ?Article bud:hasAmount ?Amount.
                  ?Amount bud:year ?A_Year.
                  #FILTER(str(?A_Year)="2018" OR str(?A_Year)="2017")

                  ?Amount bud:hasPoliticalCategory ?A_CatPol.
                  ?Amount a ?A_Type.
                  ?Amount bud:figure ?MonetaryValue.
                  ?MonetaryValue bud:value ?MV_Value.
                  FILTER(REPLACE(xsd:string(?A_Type),   '.*[/#]',   '')="Commitment"   OR
REPLACE(xsd:string(?A_Type), '.*[/#]', '')="NonDifferenciated")
             }
             UNION
             {
                  ?Title a bud:Title.
                  ?Title bud:alias ?TiAlias.
                  ?Title bud:heading ?TiHeading.
                  FILTER (lang (?TiHeading) = 'en').

                  ?Title bud:hasChapter ?Chapter.
                  ?Chapter bud:hasArticle ?Article.
                  ?Article bud:hasItem ?Item.
                  ?Item bud:hasAmount ?Amount.
                  ?Amount bud:year ?A_Year.
                  #FILTER(str(?A_Year)="2018" OR str(?A_Year)="2017")

                  ?Amount a ?A_Type.
                  ?Amount bud:figure ?MonetaryValue.
                  ?MonetaryValue bud:value ?MV_Value.
```

```
                  FILTER(REPLACE(xsd:string(?A_Type),   '.*[/#]',   '')="Commitment"   OR
REPLACE(xsd:string(?A_Type), '.*[/#]', '')="NonDifferenciated")
             }
           }
} }
GROUP BY(?TiAlias)
#HAVING(?Amount_a >100000000)
ORDER BY(?TiAlias) (?Head)
```