# Example of RSP-QL query

From RDF Stream Processing Community Group

This query continuously look for bars where people are falling in love like Paolo and Francesca in Dante's Divine Comedy (http://en.wikipedia.org/wiki/Francesca_da_Rimini#In_Inferno) because of a book by Gallehault (http://en.wikipedia.org/wiki/Galehaut).

The query checks :

- over the default graph containing the points of interest (POIs) of http://somesocialnetwork.org/ that the POI is a bar.
- over the entire stream from http://someinvasivesensornetwork.org, that pairs of people entered in the poi in different moments within 4 hours.
- over the same stream, with a long lasting time window of 1 hour, that pairs of those people have been staying close by for at least 30 minutes. Note: that this may require some resoning being the property isCloseBy symmetric.
- over the same stream but with a short time window of 10 minutes, that the same pairs exit together.

As output, for each bar, it streams out an RDF graph with the list of pairs and the total number of pairs that felt in love.

Note that this example query covers features of C-SPARQL, CQELS, and SPARQL-Stream as well as new features missing in all RSP languages:

- From C-SPARQL it takes the REGISTER clause, the FROM STREAM clause as dataset clause, the AT clause to access the timestamp (in C-SPARQL, AT is implemented with the timestamp() function) and the aggregates (which are computed in parallel without shrinking the result set, but extending it).
- From CQELS it takes the idea of the STREAM keyword in the WHERE clause, herein defined as WINDOW.
- From SPARQL-Stream it takes the ISTREAM clasue, that ask the RSP engine to use the R2S operator, and the notion of windows in the past.

Differently from a previous version of this query, it no longer covers features of EP-SPARQL such as SEQ or the getDuration() function. This reflects the decision to layer the complex event processing language on a continuous querying one.

The new features are:

- the usage of an IRI to identify the query (and its stream of results)
- the optional UNDER ENTAILMENT REGIME clause
- the FROM NAMED WINDOW ON STREAM <<stream iri>> <<window>> AS << window name>> clause in the dataset declaration
- the WINDOW keyword in the WHERE clause

```
PREFIX e: <http://somevocabulary.org/>
PREFIX s: <http://someinvasivesensornetwork.org/streams#>
PREFIX g: <http://somesocialnetwork.org/graphs#>

PREFIX : <http://acrasycompany.org/rsp>
REGISTER STREAM :GallehaultWasTheBar
UNDER ENTAILMENT REGIME <http://www.w3.org/ns/entailment/RIF>
AS
FROM NAMED WINDOW ON STREAM s:1 [RANGE PT4H STEP PT1H] AS :veryLongWindow
FROM NAMED WINDOW ON STREAM s:1 [FROM NOW-PT35M TO NOW-PT5M STEP PT5M] AS :longWindow
FROM NAMED WINDOW ON STREAM s:1 [RANGE PT10M STEP PT5M] AS :shortWindow
FROM NAMED GRAPH g:SocialGraph
FROM GRAPH g:POIs
CONSTRUCT ISTREAM {
 ?poi rdf:type :Gallehault ;
      :count ?howmanycouples ;
      :for (?somebody ?someoneelse)
}
WHERE {
 ?poi rdf:type e:bar .
 WINDOW :veryLongWindow {
       ?somebody e:enters ?poi
       ?someoneelse e:enters ?poi
 }
 WINDOW :longWindow {
     ?somebody e:isCloseTo ?someoneelse
     MINUS { ?somebody e:isCloseTo ?yetanotherone . FILTER (?yetanotherone != ?someoneelse) }
 }
 WINDOW :shortWindow {
     { ?somebody e:exits ?bar} BEGIN AT ?t1
     { ?someoneelse e:exits ?bar } BEGIN AT ?t2
     FILTER (abs(?t2-?t1)<"PT1M"^^xsd:duration )
 }
 GRAPH g:1 {
     FILTER NOT EXIST { ?somebody e:knows ?someoneelse }
 }
 FILTER (?somebody != ?someoneelse)
}
AGGREGATE {
 GROUP BY ?bar
 COUNT(?somebody) AS ?howmanycouples
}
```

Retrieved from "http://www.w3.org/community/rsp/wiki/index.php?title=Example_of_RSP-QL_query&oldid=474"

- This page was last modified on 11 December 2014, at 10:50.
- This page has been accessed 50 times.