Interactive exploration of complex relational data sets in a web interface

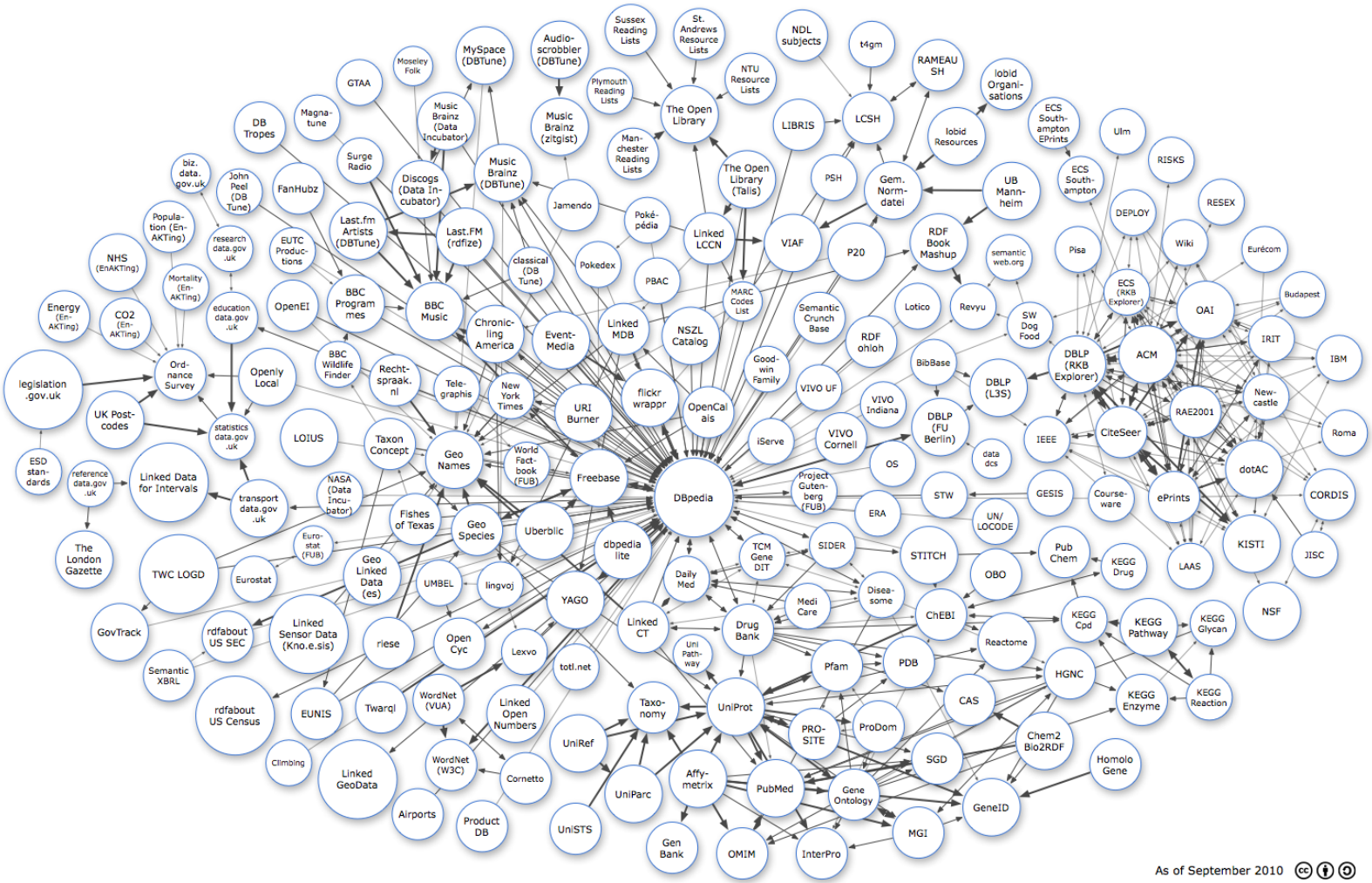
Vincent Michel - Logilab

SemWeb.pro 2012 - 2 mai 2012

Table des matières

Context - Number

Increasing number of databases



Context - Size

Increasing number of HUGE databases

Few examples

- **Data.bnf.fr** : 6.10^6 rdf triples / 65 Mo (April 2012).
- **Geonames** : 8.10^6 features / 150.10^6 rdf triples / 2Gb (March 2012).
- **Dbpedia** : 3.10^6 things / 1^9 rdf triples / > 30 Gb (July 2011).

Related issues

- **Storing data** : lots of space required.
- **Updating data** : massive amount of operations.
- **Querying/Visualization** : Scalable tools.

Context - Format

Increasing number of HUGE databases with DIFFERENT formats

Few examples

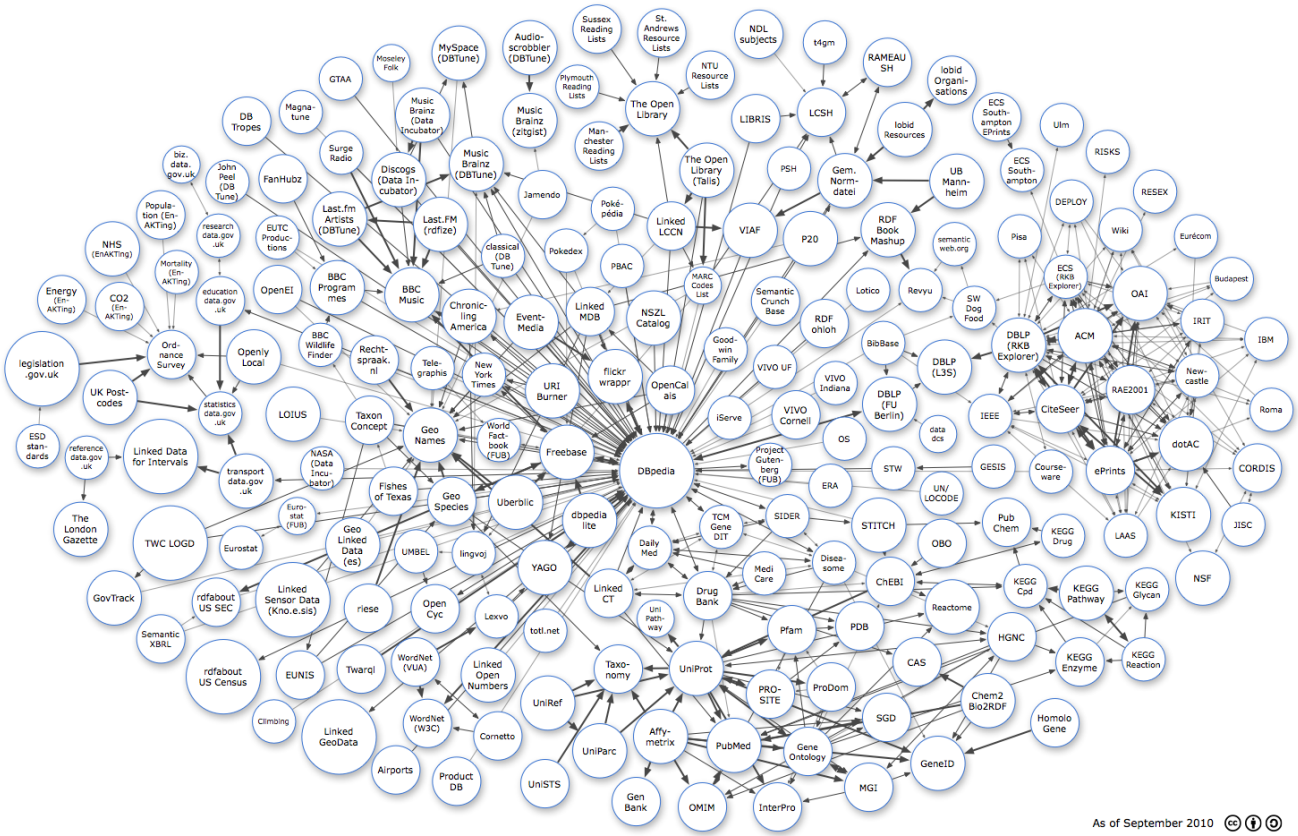
- **Geonames** : CSV file.
- **Data.bnf.fr** : separated nt/n3/rdf files.
- **Dbpedia** : (numerous) separated nt/n3 files.
- **NosDeputes** : SQL dumps.

Related issues

- **Storing data** : lots of conversions are required to push data within the same RDBMS.
- **Querying data** : some links have to be reconstructed from the dumps.

Context - THE Question

What can we do with THAT ? ? !



How can we query the data ? How can we visualize the data ?

Table des matières

Cubicweb framework

A semantic open-source web framework written in Python

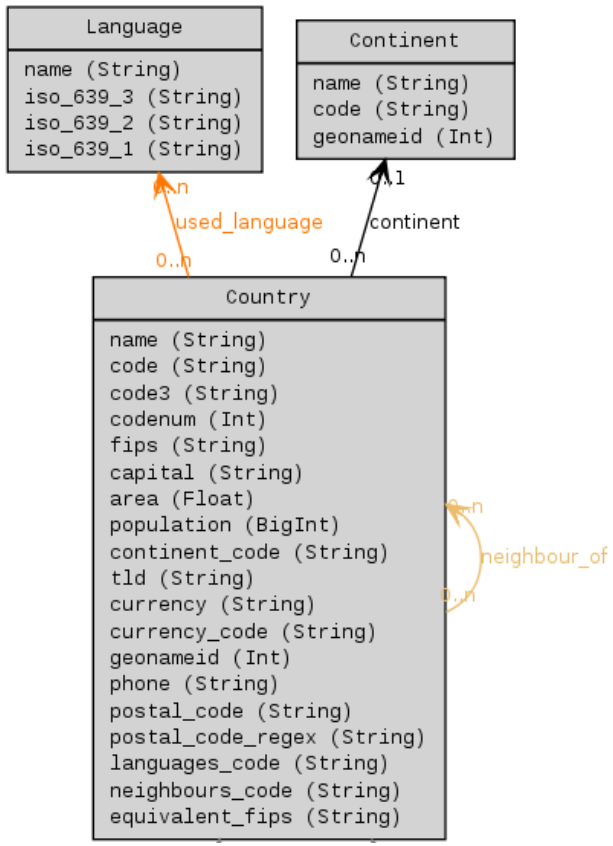
An efficient knowledge management system

- **Entity-relationship data-model** : query directly the *concepts* while abstracting the *physical structure* of the underlying database.
- **RQL (Relational Query Language)** : High-level query language operating over a relation database (PostgresSQL, MySQL, ...)
- **Separate query and display** : visualize the results in the standard web framework, download them in different formats (JSON, RDF, CSV,...)

Structured database : reconstruct the relations that may exist between different entities and entity classes, so that queries will be easier (and faster than a triple-store).

Data Model

The schema defines different entity classes with their attributes, as well as the relationships between the different entity classes



Querying data

RQL (Relational Query Language)

- **Similar to SPARQL**.
- Higher-level than SQL, **abstracts the details of the tables and the joins.**

Example :

Give me 10 locations that have a population greater than 1000000, and that are in a country named "France"

↓

```
Any X LIMIT 10 WHERE X is Location,
X population > 1000000, X country C, C name "France"
```

→ A query returns a **result set** (a list of results), that can be displayed using views.

Viewing data

Full process

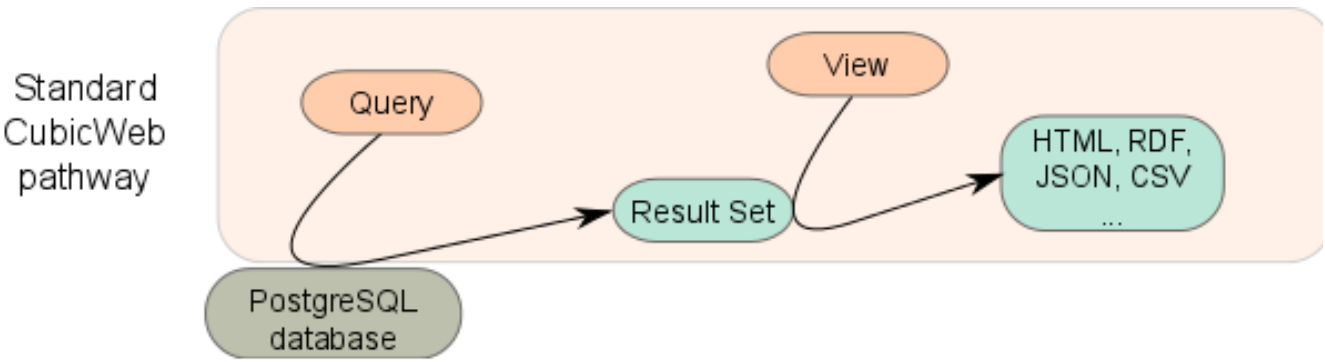
- **Request**

  Give me 10 locations that have a population greater than 1000000, and that are in a country named "France"

- **RQL**

```
Any X LIMIT 10 WHERE X is Location,
X population > 1000000, X country C, C name "France"
```

- **Standard views :** Web framework, JSON, RDF, CSV, ...




```
http://domain:8080/?rql=my rql&vid=json
```

Table des matières

Goals

We have databases with **complex structure**

→ **Building an interface to visualize queries/perform datamining over a relational database**

Why ?

- Each visualization has a corresponding url → **results are addressable, linkable and shareable.**
- Using **Javascript-based visualization** : d3.js, protovis, mapstraction, ...
- Provide more complex views, based on **datamining procedures.**

A two steps process

Pivot data structure : numerical arrays, JSON

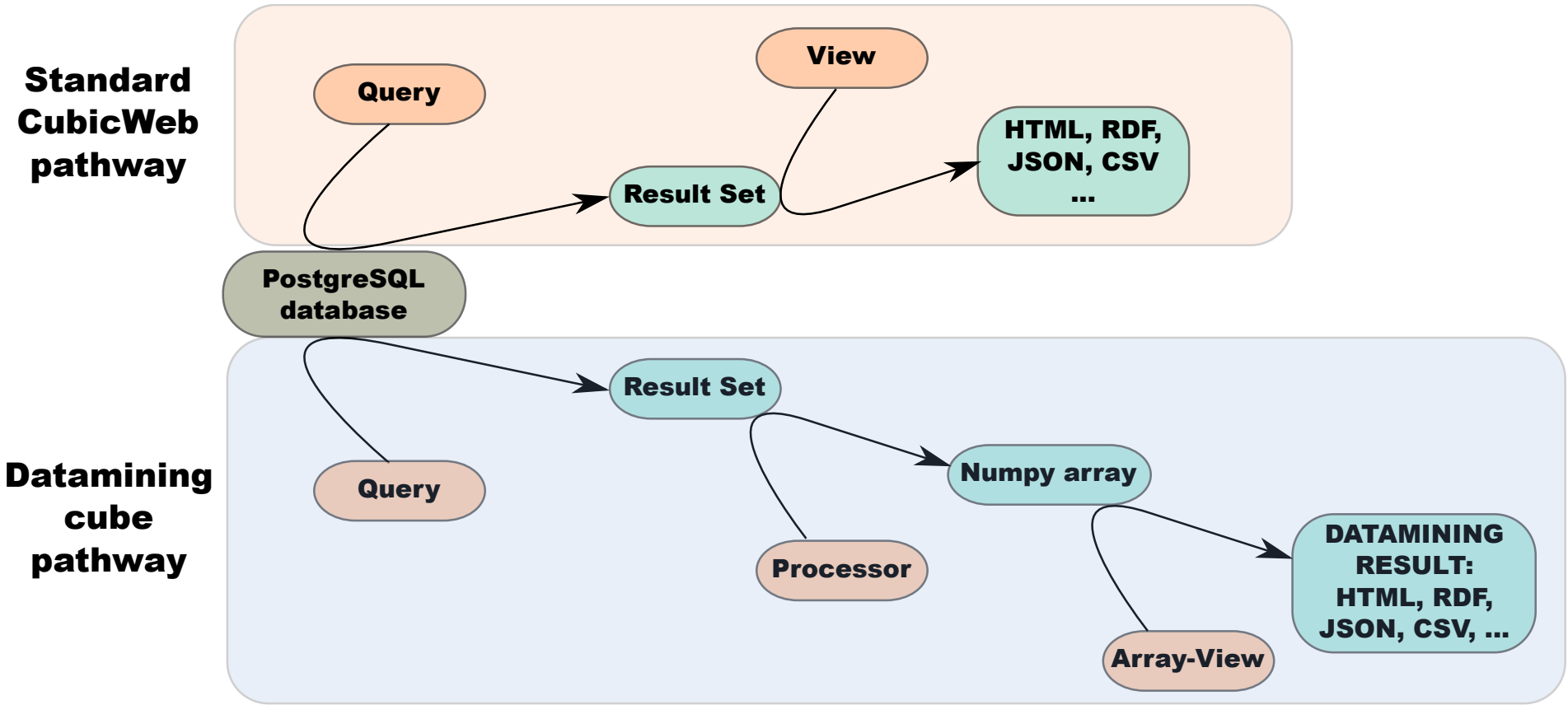
Processor

- **Convert result set $\rightarrow$ pivot structure**.
- The processor defines some criteria for the mathematical processing.

Array view

- For numerical array, we construct **Array views**, associated to some visualizations or datamining processes.
- Display the arrays as graphs, histograms, maps...

Visualization pathway



Overview of some possibilities

One can combine almost any *Processor* with any *Array view*

Processors

- *attr-asfloat* turns all Int, BigInt and Float attributes in the result set to floats, and returns the corresponding array.
- *undirected-rel* creates a binary numpy array that represents the relations (or corelations) between entities.
- *attr-astoken* turns all String attributes in the result set in a numpy array, based on a Word-n-gram analyze.

Array views

Histogram, scatterplot, matrix, force-directed graph, dendrogram, ... $\rightarrow$ each view has a default processor.

Example of combination

Query : *20 couples (name, id) of parliamentarians, in alphabetical order*

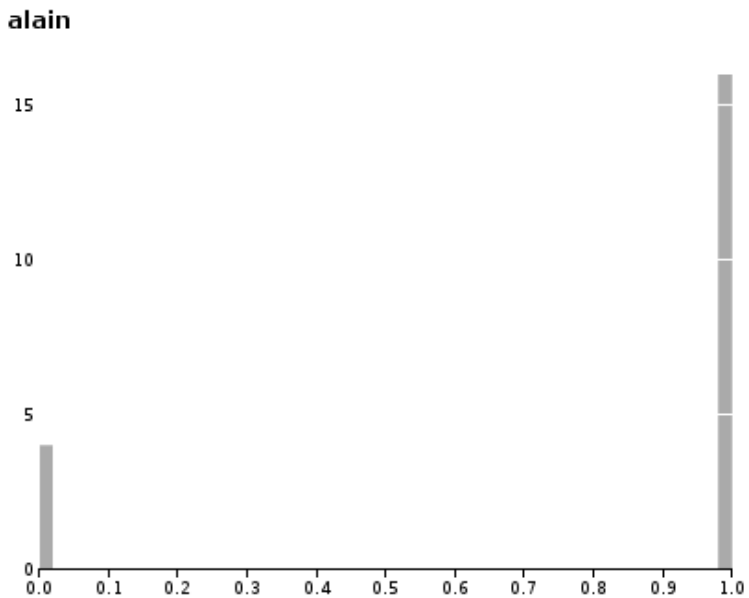
Create tokens from names (processor : **attr-astoken**), and view tokens in table (array-view : **array-table**)

	rousset	facon	neri	aly	almont	marc	claeys	joyandet	alfred	abdoulatifou	cacheux	alain
0	0	0	0	1	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	0	0	0	0	1	1

```
http://mondomain.vm:port/?rql= Any  X, N ORDERBY N ASC
LIMIT 20 WHERE X is Parlementaire, X nom N
&arid=attr-astoken&vid=array-table
```

Changing the view...

Create tokens from names (processor : **attr-astoken**), and now, view the number of occurence of tokens using a histogram (array-view : **protovis-hist**)

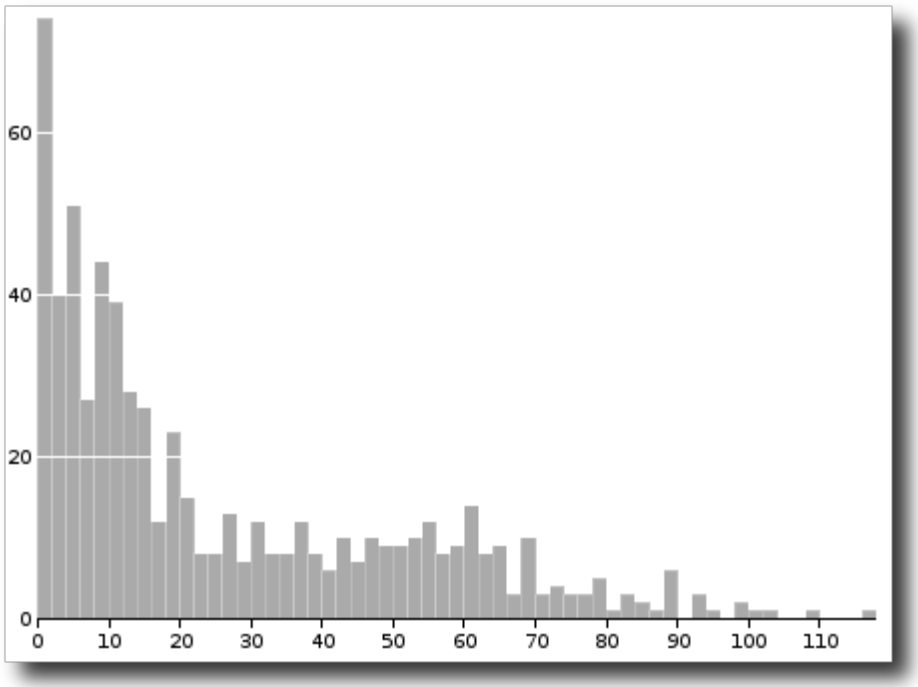



```
http://mondomain.vm:port/?rql= Any  X, N ORDERBY N ASC
LIMIT 20 WHERE X is Parlementaire, X nom N
&arid=attr-astoken&vid=protovis-hist
```

Histogram visualization - *NosDeputes*

Number of comments by parliamentary

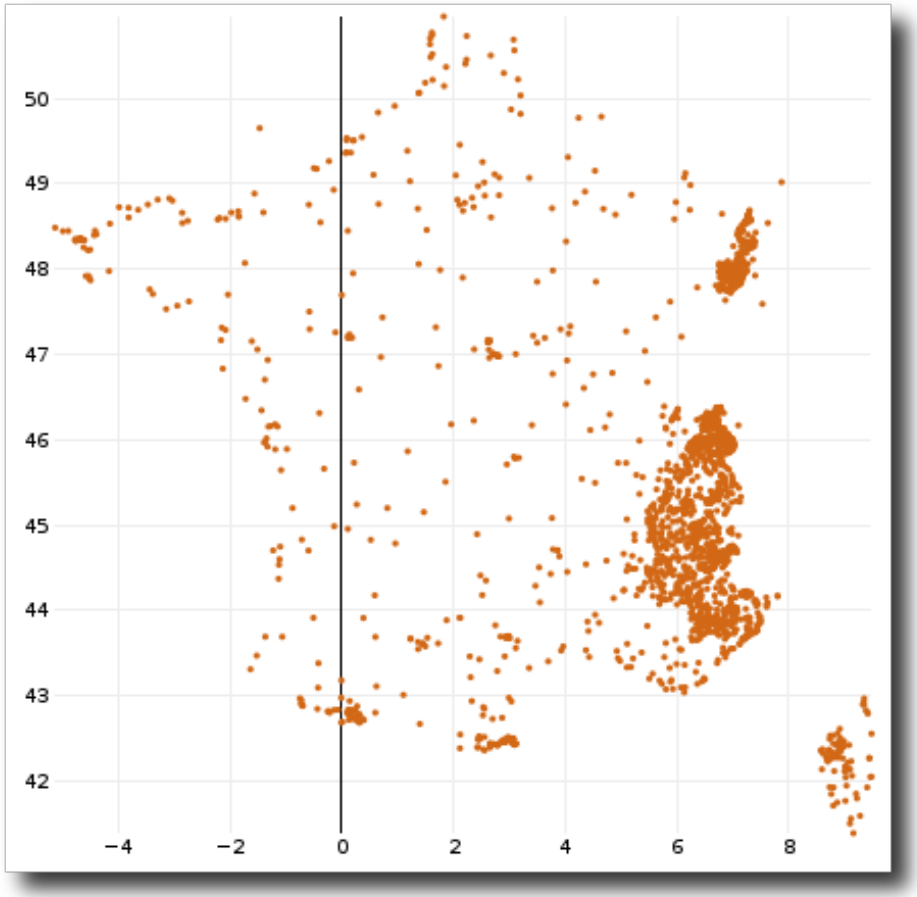
Processor : **attr-asfloat**, Array view : **protovis-hist**



Scatterplot visualization 1 - Geonames

All couples (latitude, longitude) of the locations in France, with an elevation not null

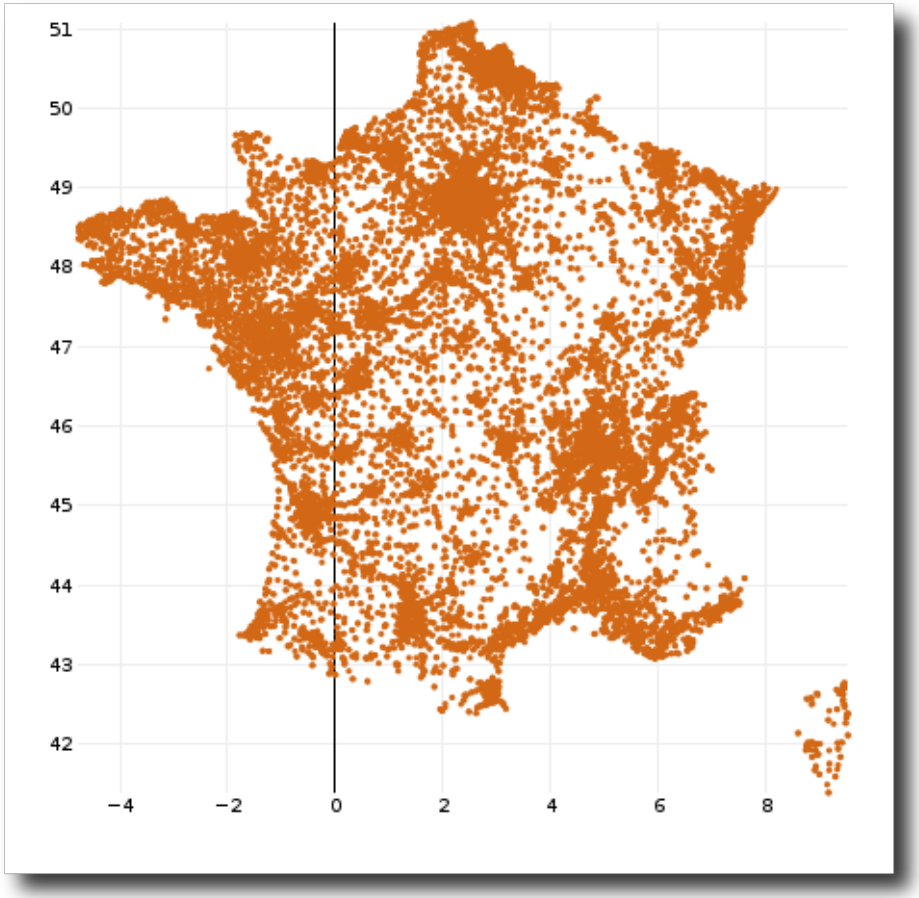
Processor : **attr-asfloat**, Array view : **protovis-scatterplot**



Scatterplot visualization 2 - Geonames

All couples (latitude, longitude) of 50000 locations in France, with a population higher than 100 (inhabitants)

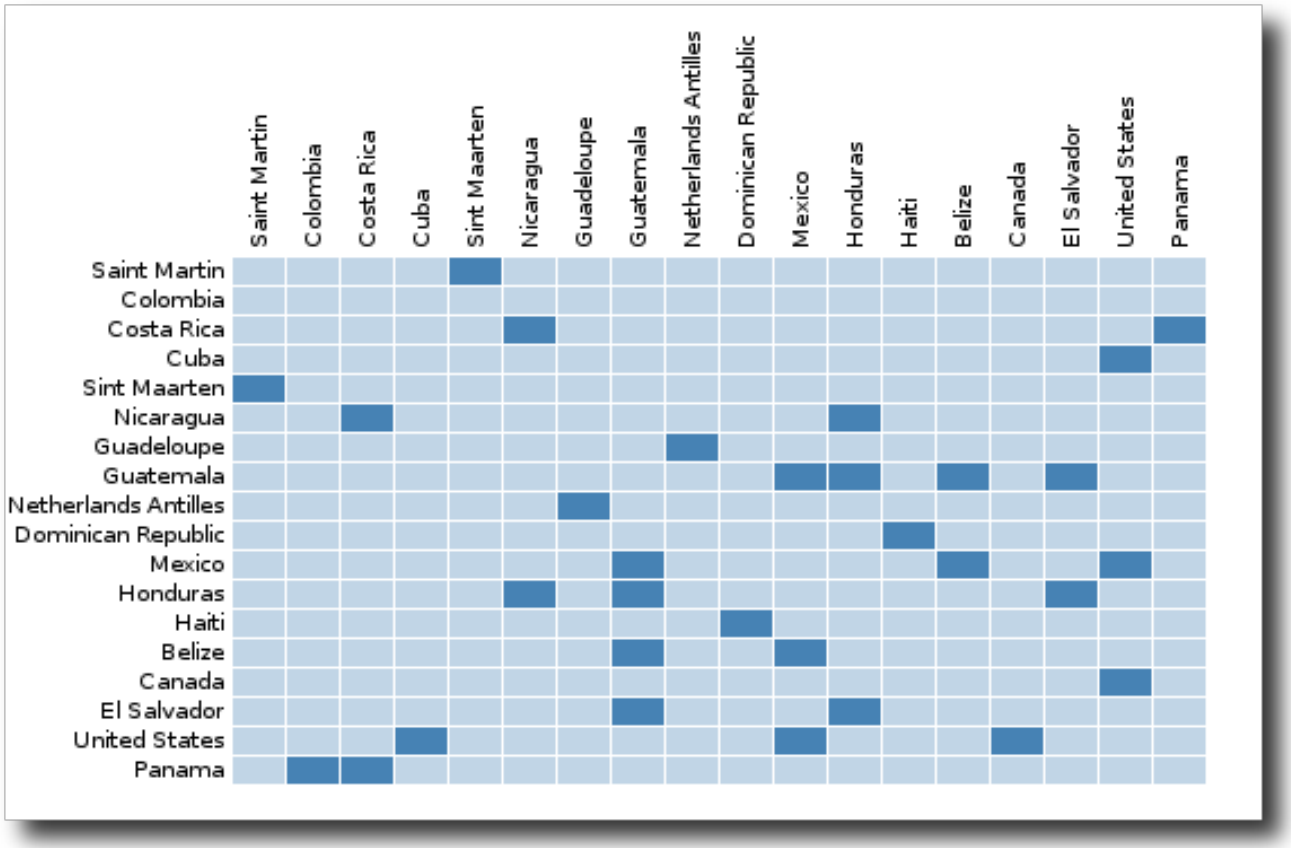
Processor : **attr-asfloat**, Array view : **protovis-scatterplot**



Matrix visualization - Geonames

All neighbour countries in North America

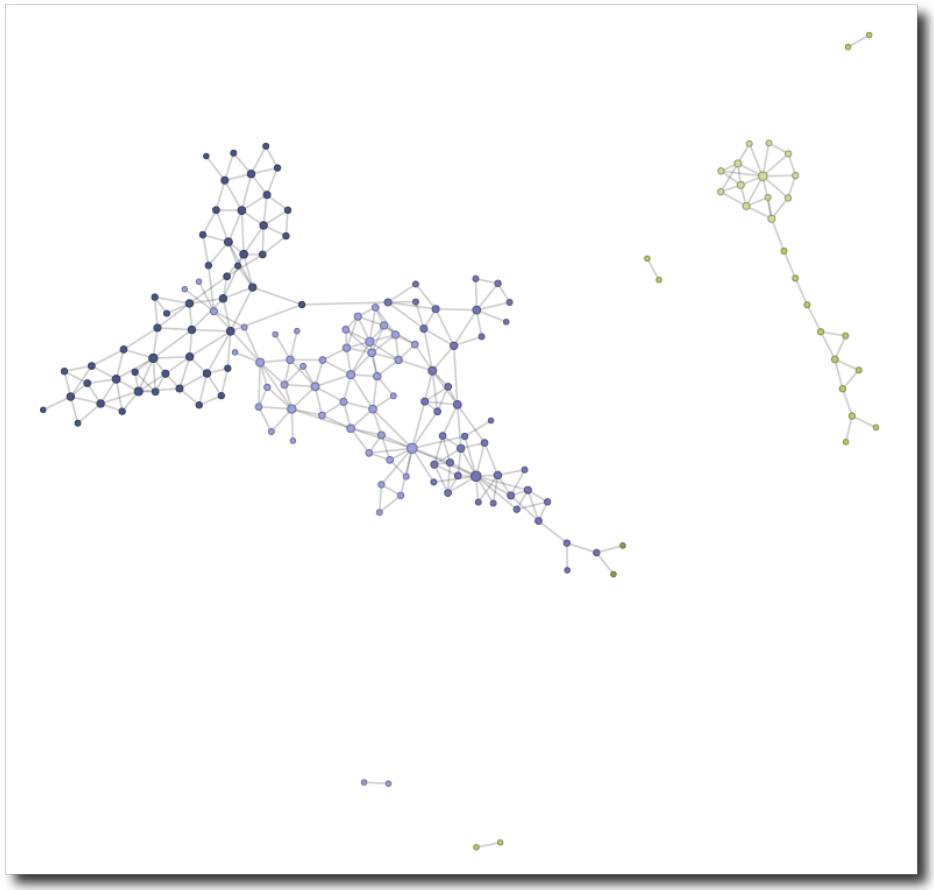
Processor : **undirected-rel**, Array view : **protovis-binarymap**



Graph visualization - Geonames

All neighbour countries in the World, and their corresponding continent

Processor : **undirected-rel**, Array view : **protovis-forcedirected**



Applying datamining processes

Another kind of processor based on the Numpy Array

All couples (latitude, longitude) of the locations in France, with an elevation higher than 1

Processor : **attr-asfloat**, Array view : **protovis-scatterplot**, Datamining process : **Kmeans**

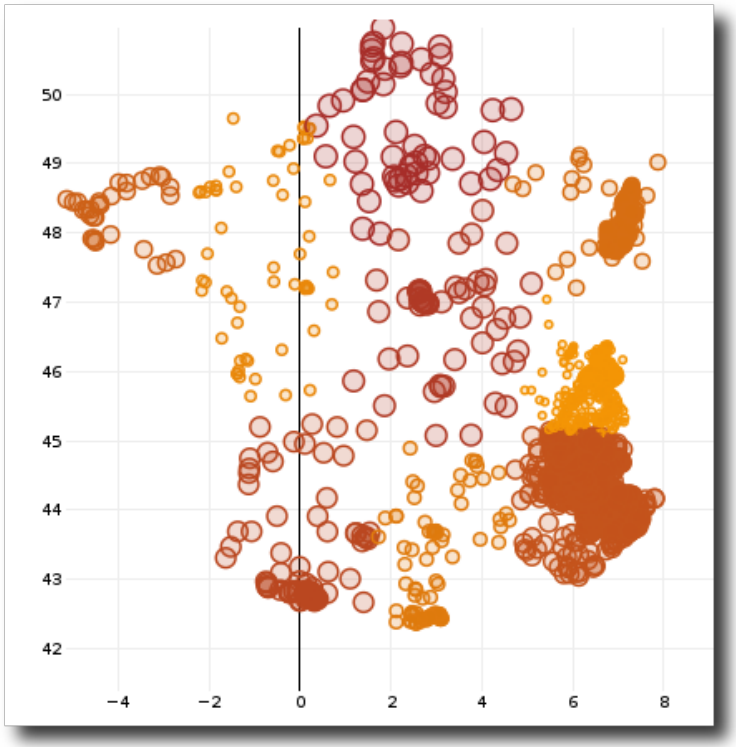


Table des matières

Visualization in CubicWeb

Easy to interactively process and visualize datasets.

Visualizations directly applied on queries.

All visualizations are defined by an unique URL !

Features

- quick interactive exploratory visualization/datamining processings.
- different *processors* and *views* for more flexibility.
- based on the pivto structure (JSON, numerical arrays).
- Use jointly with **facets** to filter results.

Conclusion

Future developments

- More datamining procedures (unsupervised learning, clustering, ...).
- Including more Javascript libraries.
- Use sparse arrays for better performance.

Questions ?