Analytics and Visualization over Semantic RDF Graphs

Maria-Evangelia Papadaki, Yannis Tzitzikas, Nicolas Spyratos

ISIP, 9-10 May, 2019, Heraklion

FORTH-ICS
Information Systems Laboratory

University of Crete
Computer Science Department

OUTLINE

- Motivation
- HIFUN: A High Level Functional Query Language for Big Data Analytics
- Application of HIFUN to RDF Graphs
- Evaluating HIFUN Queries in SPARQL
- Concluding Remarks
- Future Work

MOTIVATION

Motivation (1/2)

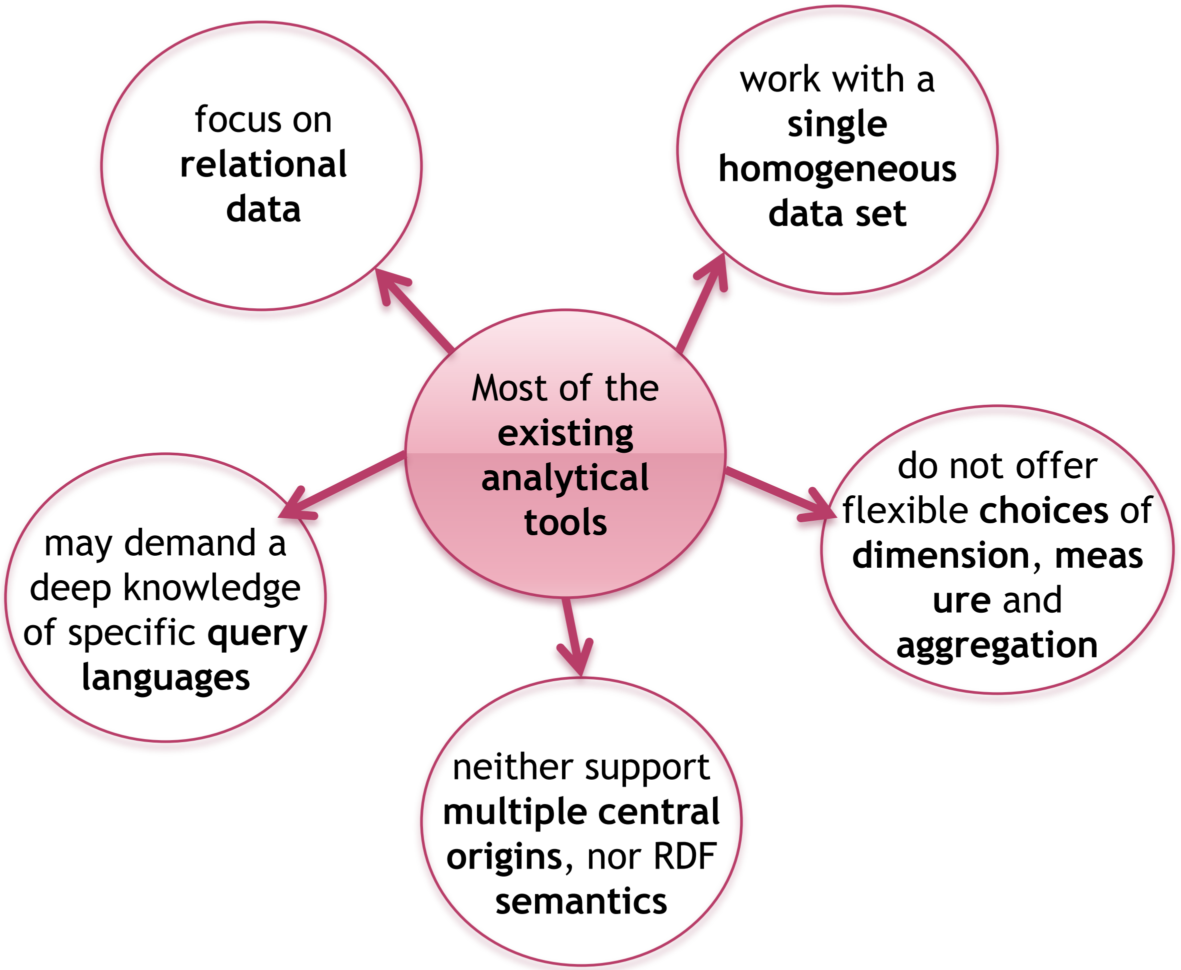
We live in the era of data **explosion**. At the same time, more and more data sources are being produced as **linked data**, using the Resource Description Format (RDF).

However, the exploitation and the application of **analytics** over RDF data, is not so straightforward, since its structure is not so simple:

- different **resources** may have different **sets of properties**,
- resources **may or may not have types,**
- **multi-value properties** may exist

Motivation (2/2)

- Even though, many analytical **tools** have been developed, they:



What do we **need**?

- An analytical **tool** that would:
  - be applied over **one** or **multiple linked data sets**,
  - **not** demand any **programming skill**,
  - **visualize** data **intuitively** and will support **collaborative** data exploration.

What are we **doing**?

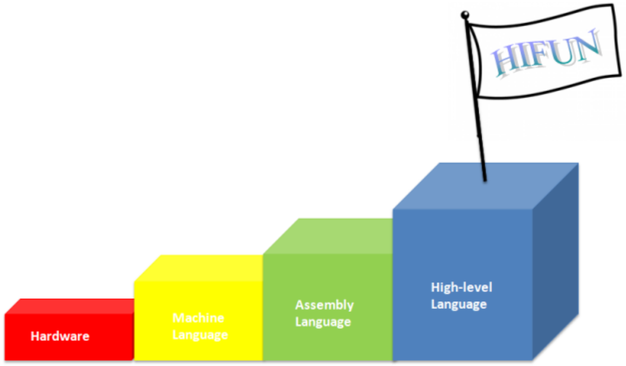
- Currently, we are investigating an approach based on:
  - **a high level query language**, for **analytics** over RDF graphs [N. Spyratos et al. 2018], and
  - an interactive 3D **system** [M.E. Papadaki et al. 2018] for **visualizing linked data**.

HIFUN: A HIGH LEVEL FUNCTIONAL QUERY LANGUAGE FOR BIG DATA ANALYTICS

Definition of HiFun (1/3)

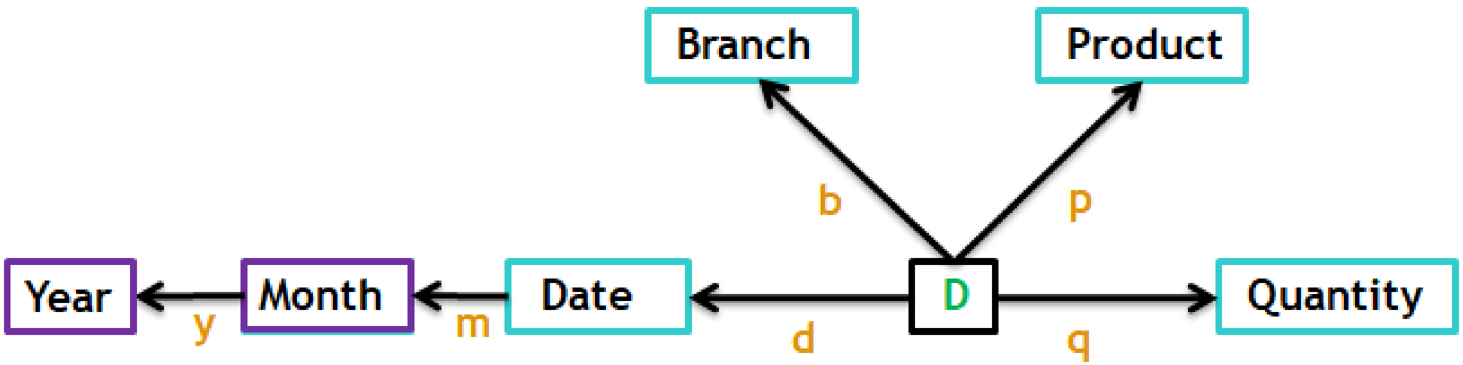
HIFUN [N. Spyratos et al. 2018], is a **high level functional query language** for defining **analytic** queries over **big data sets**, independently of how these queries are evaluated.

- It can be applied over a data set that is **structured** or **unstructured**, **homogeneous** or **heterogeneous**, **centrally stored** or **distributed**.



Definition of HiFun (2/3)

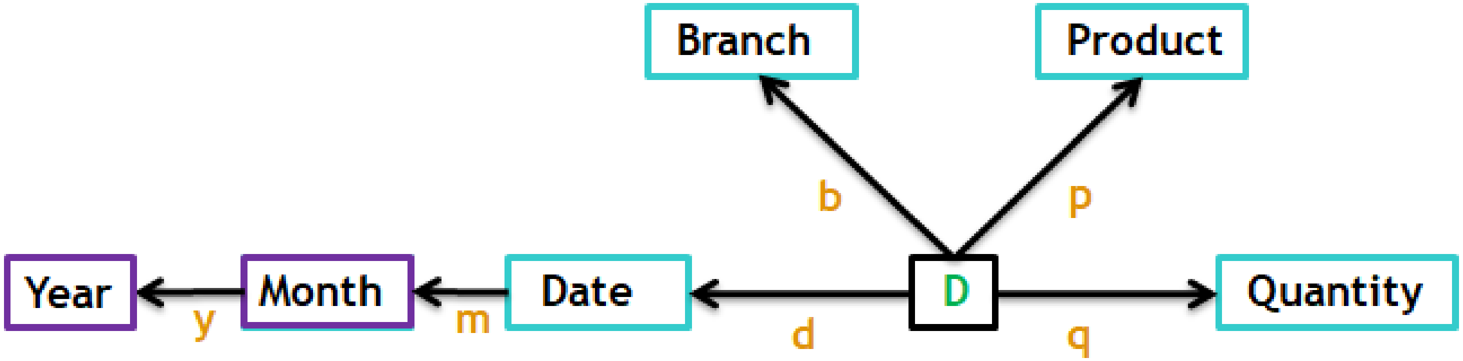
- **Data set assumptions**
  - consists of **uniquely indentified items**



  - has a set of **attributes,**
    - an attribute is **viewed as a function** from the data set to some domain of values: **nameOfAttribute: D → value**
      - e.g. date: D → String

Definition of HiFun (3/3)

- The set of **attributes** (direct or derived) that analysts are **interested** in is called **context** and D the **origin** of it.



The attributes that their values appear on the invoices are called **direct**, while those that can be computed from them are called **derived** (e.g. the attributes m and y can be extracted from d).

Definition of Hifun Query

A query Q in HiFun is viewed as an ordered triple, such that g and m are **attributes** of the data set D, and **op** is an **aggregate operation** applicable on m-values.

$$Q = \overbrace{(g, m, op)}^{\text{Ordered triple}}$$

Attributes

Aggregate Operation

HIFUN: Evaluation of Analytic Query (1/4)

a) **group the items** of the data set D using the values of g (i.e. items with the same g-value g_i are grouped together)

$$Q = \overbrace{(g, m, op)}^{\text{Ordered triple}}$$

grouping function

HIFUN: Evaluation of Analytic Query (2/4)

b) in each group of the items created, **extract** from D the **m-value of each item** in the group

$$Q = \overbrace{(g, m, op)}^{\text{Ordered triple}}$$

measuring function

HIFUN: Evaluation of Analytic Query (3/4)

c) **aggregate** the **m-values** obtained in each group to obtain a **single** value v_i

Ordered triple

$$Q = \overbrace{(g,\ m,\ op)}$$

aggregate operation

HIFUN: Evaluation of Analytic Query (4/4)

c) **aggregate** the **m-values** obtained in each group to obtain a **single** value v_i

Ordered triple

$$Q = \underbrace{(g,\ m,\ op)}$$

aggregate operation

Such a query can be **evaluated** easily using either **SQL** or **Map-Reduce**.

Application of HiFun to Relational Data (1/5)

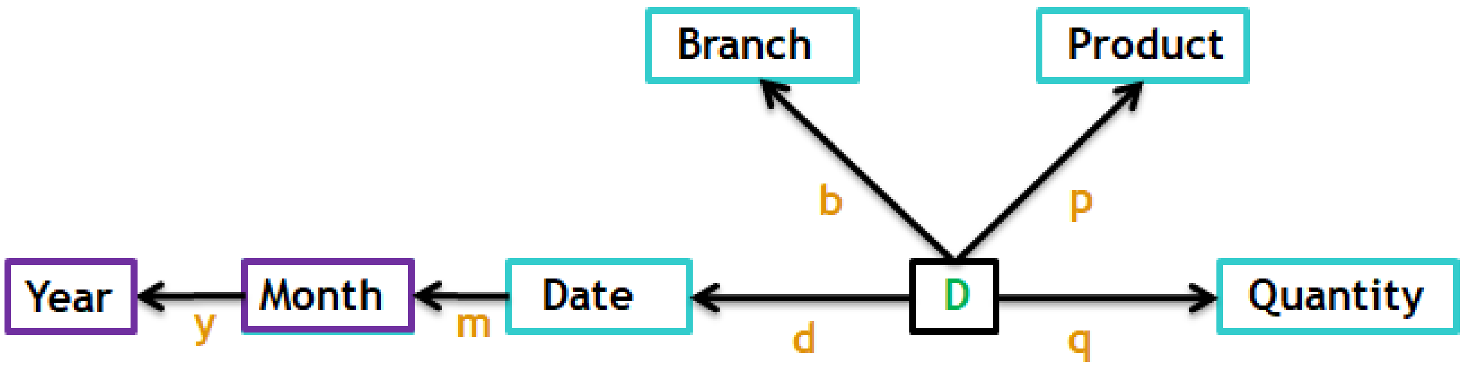
Example: suppose that, we would like to know the ***total*** *quantities of* ***products****, delivered to* ***each branch****, for the following* ***relational*** *data, using HiFun.*

D	Date	Month	Year	Branch	Product	Quantity
invoiceID1	date1	month1	year1	branch1	product1	100
invoiceID2	date2	month2	year2	branch1	product2	200
invoiceID3	date3	month3	year3	branch2	product3	300
invoiceID4	date4	month4	year4	branch3	product4	400

Application of HiFun to Relational Data (2/5)

Each invoice would be a uniquely identified item, that has a set of attributes, each of which could be seen as a Hifun function from dataset D to some domain of values.

D	Date	Month	Year	Branch	Product	Quantity
invoiceID1	date1	month1	year1	branch1	product1	100
invoiceID2	date2	month2	year2	branch1	product2	200
invoiceID3	date3	month3	year3	branch2	product3	300
invoiceID4	date4	month4	year4	branch3	product4	400



Application of HiFun to Relational Data (3/5)

a) ***group*** *together all the* ***invoices*** *referring to the same* ***branch***

1. Grouping (b)

branch1: ID1, ID2
branch2: ID3
branch3: ID4

$Q = (\boldsymbol{b}, m, op)$

Application of HiFun to Relational Data (4/5)

*b) find the **quantity** corresponding to **each invoice** in the group*

1. Grouping (b)

branch1: ID1, ID2
branch2: ID3
branch3: ID4

2. Measuring (q)

branch1: 100, 200
branch2: 300
branch3: 400

$\boxed{Q = (\boldsymbol{b}, \boldsymbol{q}, op)}$

Application of HiFun to Relational Data (5/5)

*c) in **each group** of the previous step, we **sum up** the quantities found*

1. Grouping (b)	2. Measuring (q)	3. Reduction (sum)
branch1: ID1, ID2	branch1: 100, 200	branch1: 300
branch2: ID3	branch2: 300	branch2: 300
branch3: ID4	branch3: 400	branch3: 400

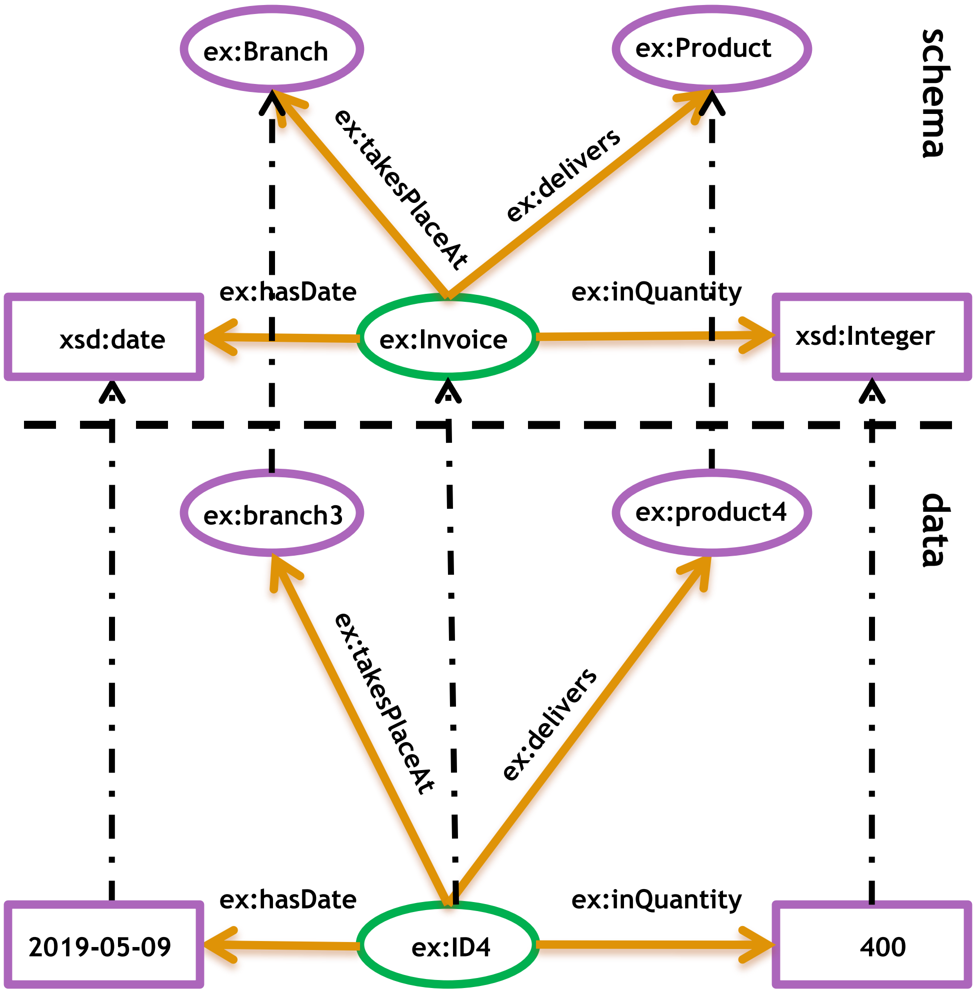
$\boxed{Q = (b, q, sum)}$

APPLICATION OF HIFUN TO SEMANTIC DATA

Application of HIFUN to semantic data (1/8)

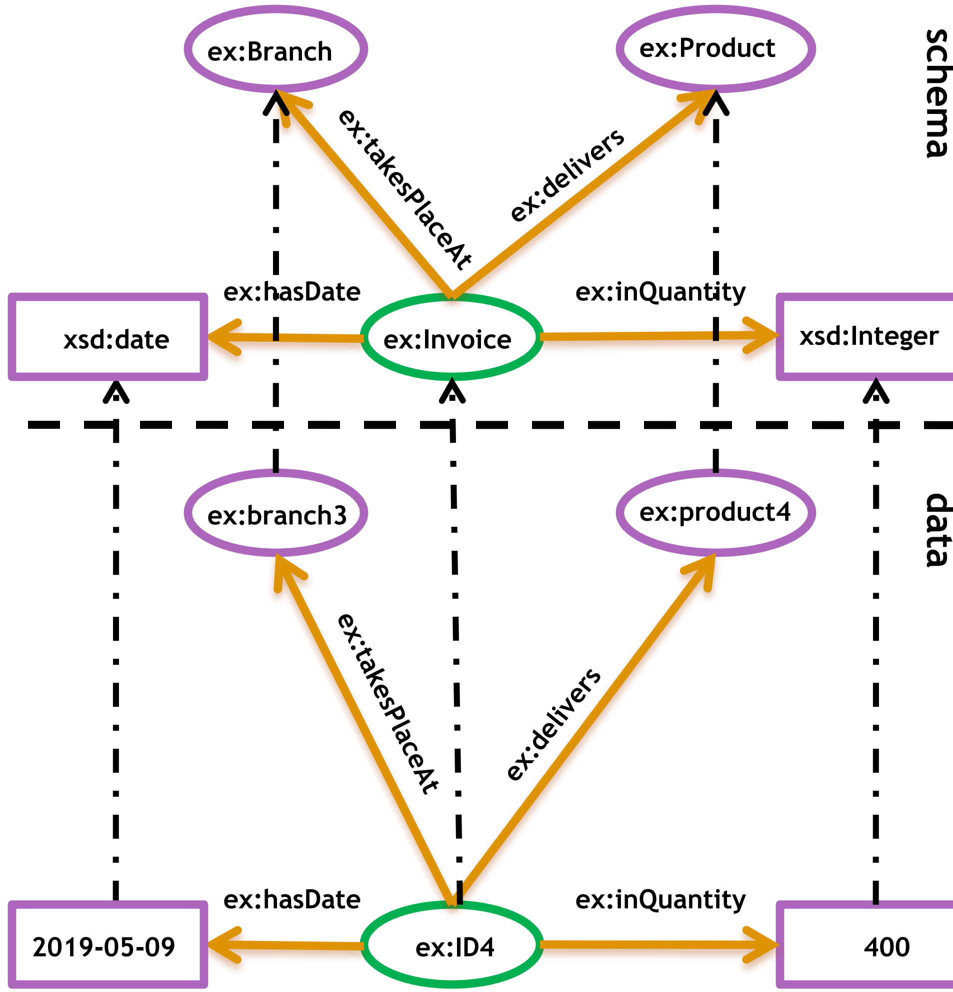
What if the data was **represented** as an **RDF** graph?

Each property would correspond to a Hifun attribute, having as source the domain of that property and target the range of it.



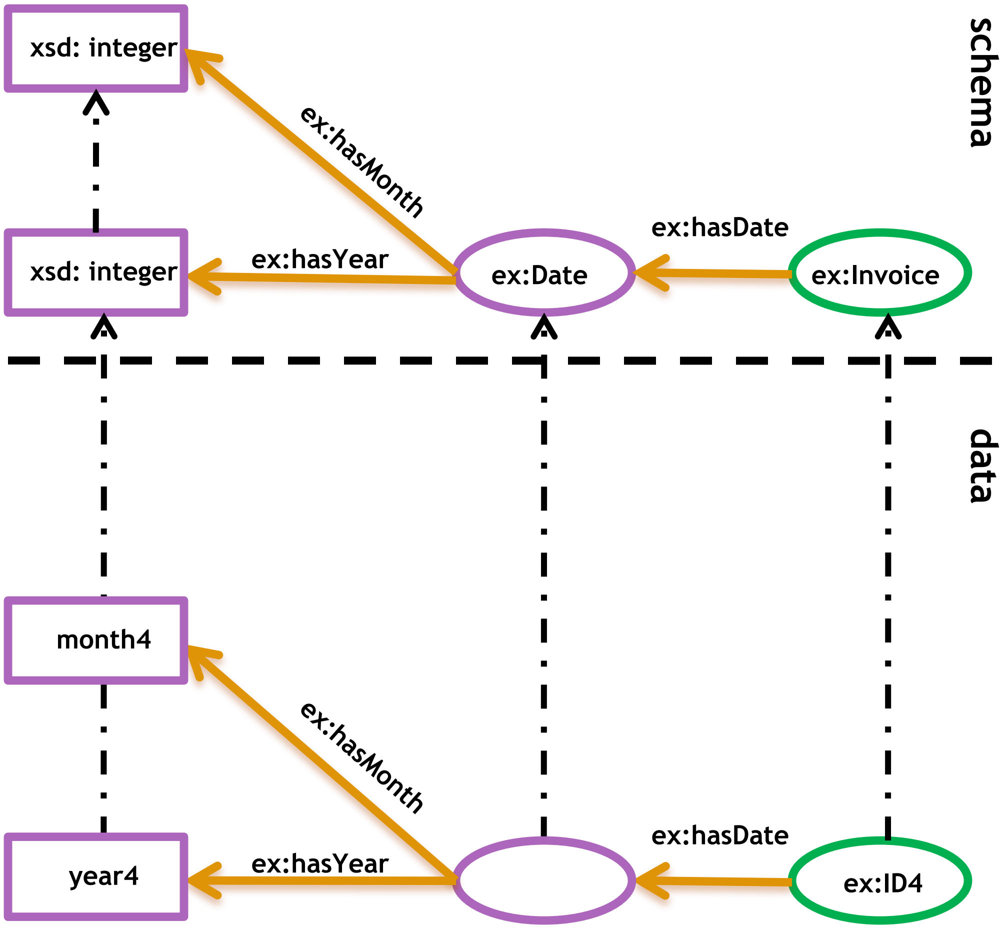
Application of HIFUN to semantic data (2/8)

In fact, one could also **derive** easily **attributes** from **literals** *(e.g. we could extract from date the attributes of "month" and "year").*



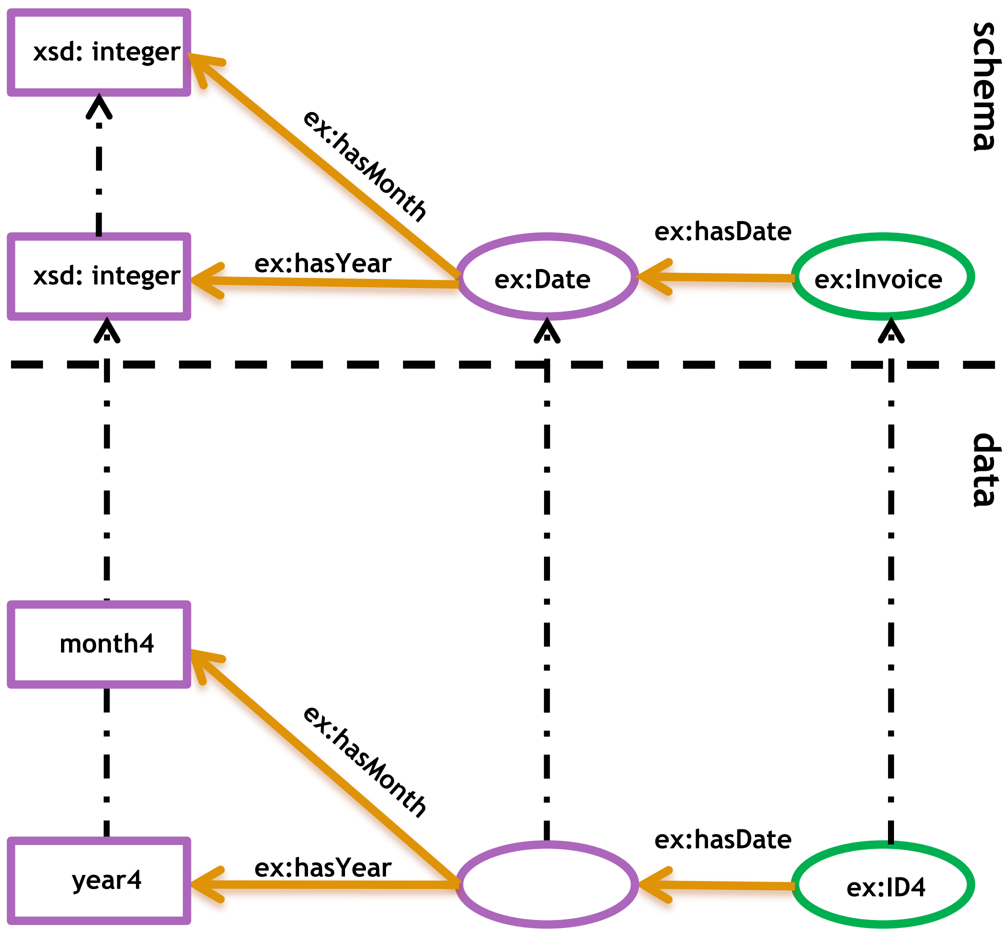
Application of HIFUN to semantic data (3/8)

Now, imagine that "date" was represented as a **blank node**. Could **HiFun** be **applied** over such data?



Application of HIFUN to semantic data (4/8)

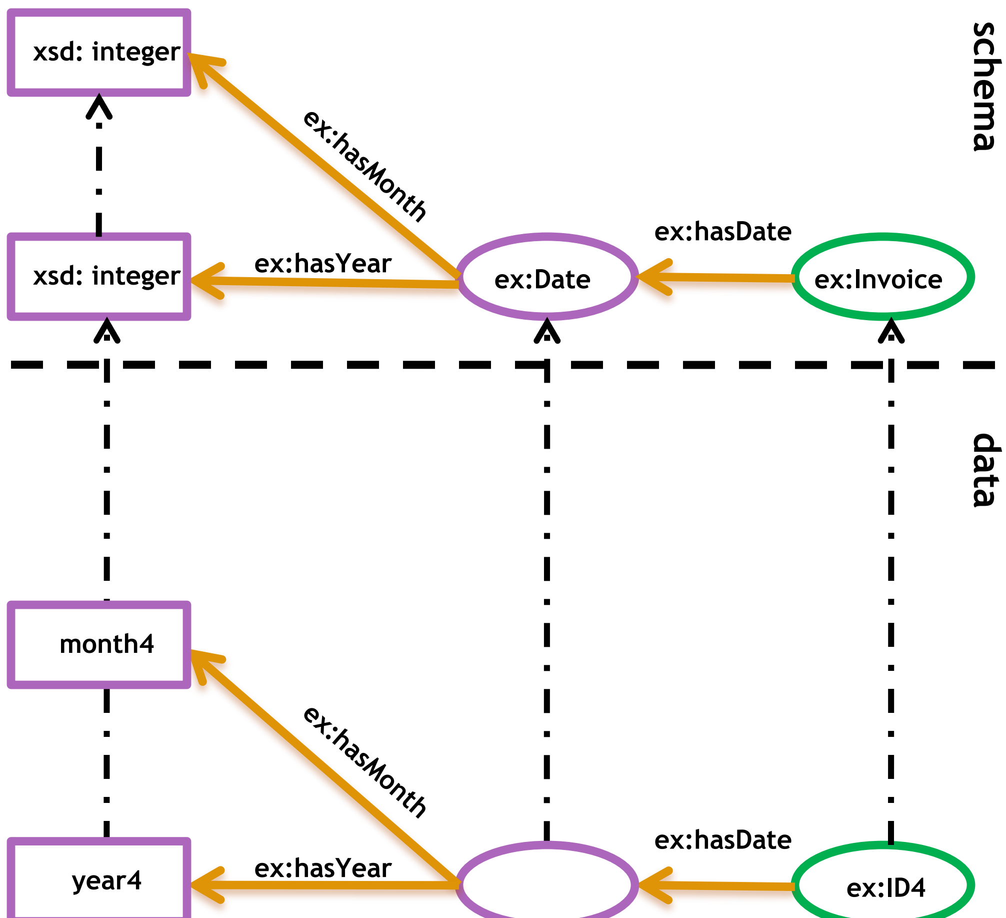
Yes, since the **attributes** of it would correspond to **1-1 functions**.



Application of HIFUN to semantic data (5/8)

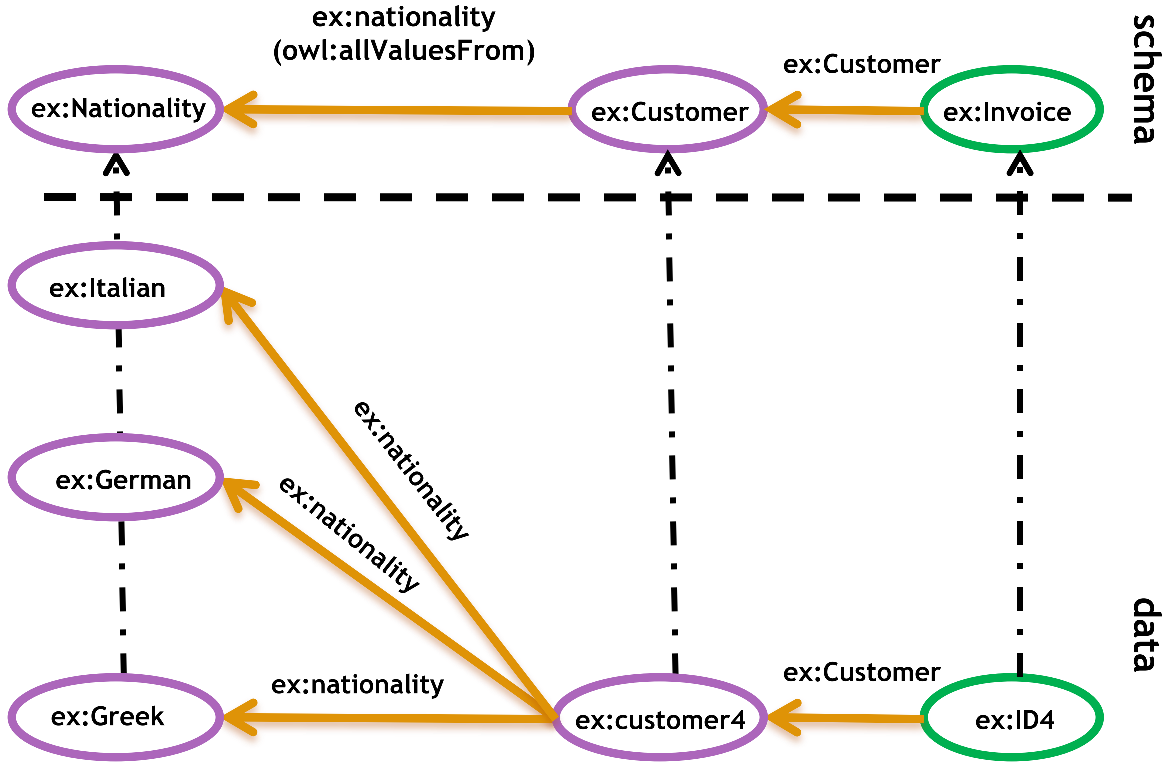
For example, date's functions would be:

hasDate(ID4) = _:b1
hasMonth(_:b1) = month4
hasYear(_:b1) = year4



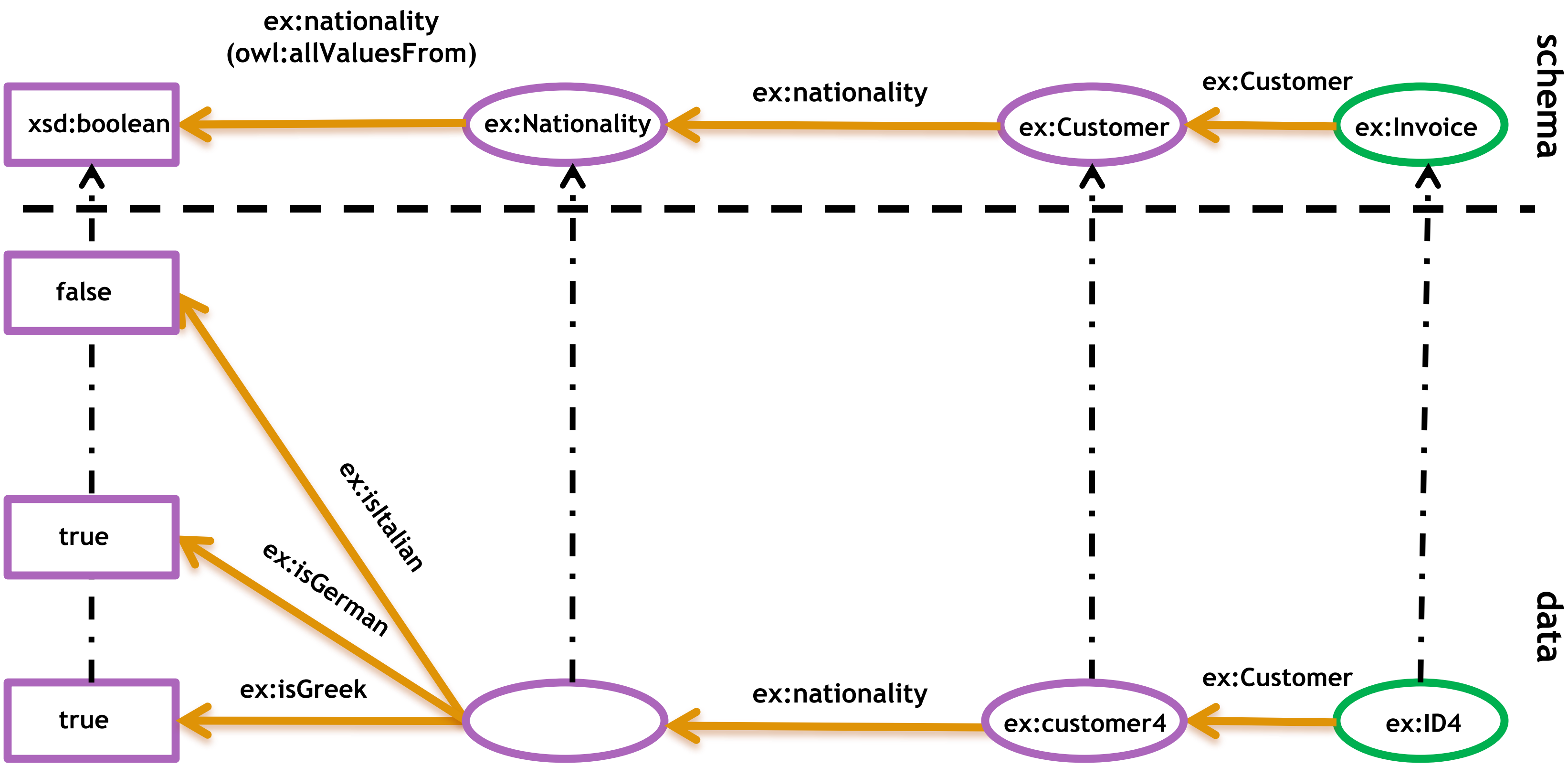
Application of HIFUN to semantic data (6/8)

But, what if an **attribute** was **not functional** (i.e. nationality)?



Application of HIFUN to semantic data (7/8)

Then, such an attribute could still be expressed in Hifun, **if** the **number of** its **values** was **finite**.



Possible transformation of multi-valued property to boolean-valued properties.

Application of HIFUN to semantic data (8/8)

- So far we,
  - have **examined** the **basics** for applying **HIFUN** over **RDF**
  - have **designed** and **implemented** a **HIFUN2SPARQL** converter and have **applied** it over RDF data expressed using the **RDF Data Cube vocabulary**
- Issues that worth further research:
  - **multi-valued** attributes
  - the interplay with **inference**
  - **complex** dimension **hierarchies**
  - **heterogeneous** graphs

EVALUATING HIFUN QUERIES IN SPARQL

Evaluating HIFUN Queries in SPARQL (1/7)

We could encode each Hifun query Q as a **SPARQL group-by query** over a triple store, as follows:

HIFUN Query
Q = (**e**, **e’**, **op**)

SPARQL Query
Select ?target(e), op(?target(e’)) As ?Result
WHERE {..}
GROUP BY ?target(**e**)

Evaluating HIFUN Queries in SPARQL (2/7)

***Query**: find the total quantities of products, group by branch.*

HIFUN Query
Q = (b, q, SUM)

```
SELECT ?branch SUM(?quantity) AS ?TOTALS
WHERE{
    ?ex:ID ex:branch ?branch .
    ?ex:ID xsd:quantity ?quantity.
}
GROUP BY ? branch
```

Results:

branch	TOTALS
"branch1"	300
"branch3"	400
"branch2"	300

Evaluating HIFUN Queries in SPARQL (3/7)

Result restricted-query

Suppose now that, we would like to set **restrictions** to the **final results**. The query would be formulated, as follows:

HIFUN Query
Q = (e, e', op)/F

SPARQL Query
Select ?target(e), op(?target(e')) As ?Result
WHERE {..}
Group by ?target(e)
HAVING(op(?target(e'))op')

Evaluating HIFUN Queries in SPARQL (4/7)

***Query**: find all the branches that received more than 300 products, group by branch.*

HIFUN Query
Q =
(b, q, SUM)/300

```
SELECT ?branch SUM(?quantity) AS
?TOTALS
WHERE{
    ?ex:ID ex:branch ?branch.
    ?ex:ID ex:quantity ?quantity.
}
GROUP BY ?branch
HAVING(SUM(?quantity) > 300)
```

Results:

branch	TOTALS
"branch3"	400

Evaluating HIFUN Queries in SPARQL (5/7)

Attribute restricted-query

Suppose now that, we would like to apply **restrictions** at the **level** of the **attributes** and filter the results, internally. The query would be formulated, as follows:

HIFUN Query
Q = (e/E, e’, op)

SPARQL Query

```
Select ?target(e), op(?target(e’)) As ?Result
WHERE {
FILTER(op(?target(e’ )) op’)
}
Group by ?target(e)
```

Evaluating HIFUN Queries in SPARQL (6/7)

Query: *find the total quantities of products that received by **branch "branch1"**, group by branch.*

HIFUN Query
Q = (b/ "branch1", q, SUM)

```
SELECT ?branch SUM(?quantity) AS ?TOTALS
WHERE{
    ?ex:ID ex:branch ?branch.
    ?ex:ID ex:quantity ?quantity.

FILTER regex((?branch), "branch1", "i")}
GROUP BY ?branch
```

Results:

branch	TOTALS
"branch1"	300

Evaluating HIFUN Queries in SPARQL (7/7)

***Query**: find the total quantities of products that received per branch, group by month.*

HIFUN Query
Q = (bom, q, SUM)

```
SELECT    ?month ?branch (SUM(?quantity)
AS ?TOTALS )

WHERE {
?ex:ID ex:hasDate ?date .
?date ex:month ?month .
?ex:ID ex:branch ?branch .
?ex:ID ex:quantity ?quantity .}

GROUP BY   ?month ?branch
```

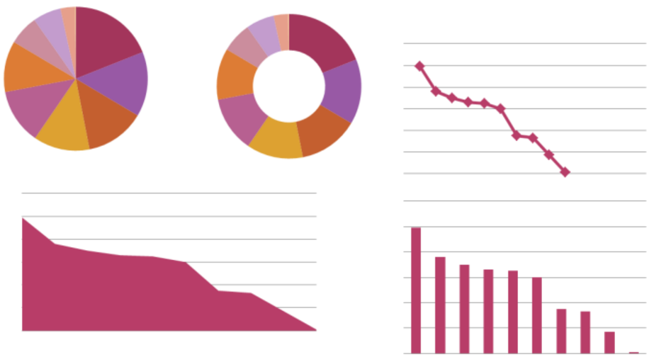
Results:

month	branch	TOTALS
"month4"	"branch3"	"400"
"month2"	"branch1"	"200"
"month3"	"branch2"	"300"
"month1"	"branch1"	"100"

VISUALIZATION FOR ANALYTICS

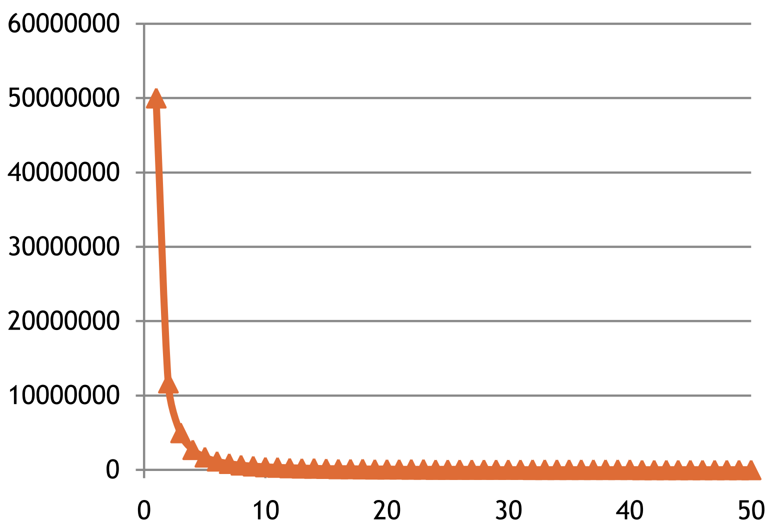
VISUALIZATION FOR ANALYTICS (1/4)

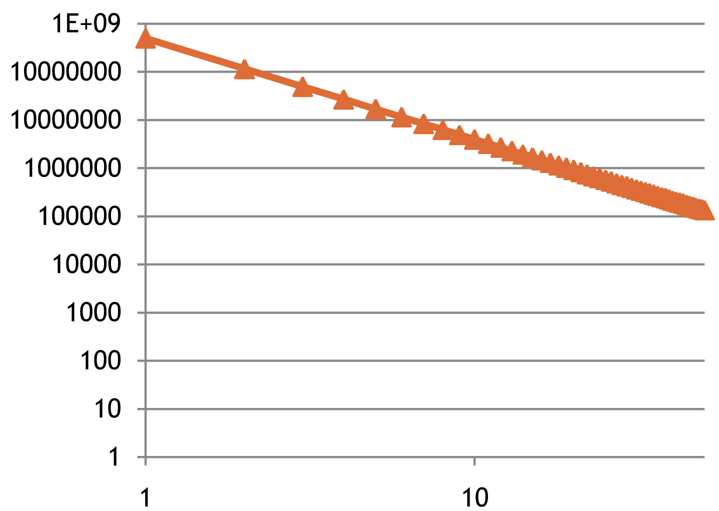
- Generally, the results of OLAP usually are visualized using:
  - 2D plots (in normal and/or log scale)
  - Pie charts
  - Histograms and bar charts
  - Scatter plots etc.



VISUALIZATION FOR ANALYTICS (2/4)

In our work, we plan to investigate visualizations appropriate also for **power law distributions.**



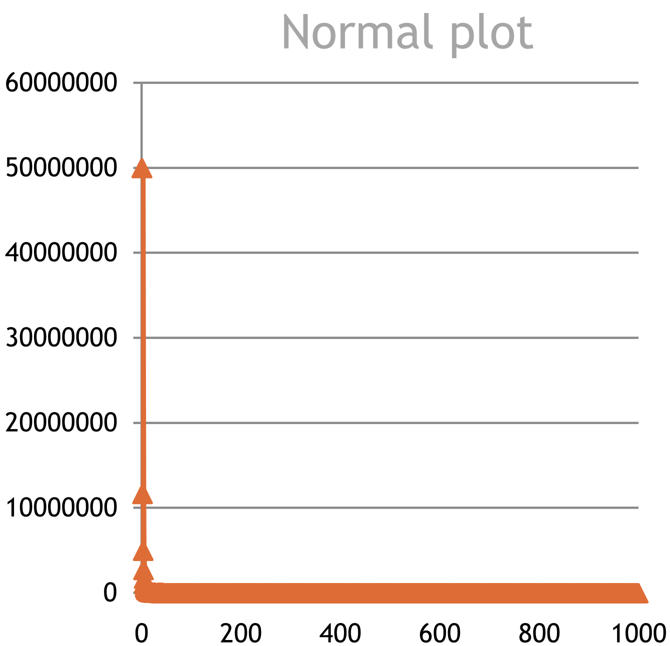


Total quantities of products, which were sold per branch, group by product (for 50 branches).

When the **data** that is visualized is **not** too **many**, then the existing **visualizations** are **adequate**.

VISUALIZATION FOR ANALYTICS (3/4)

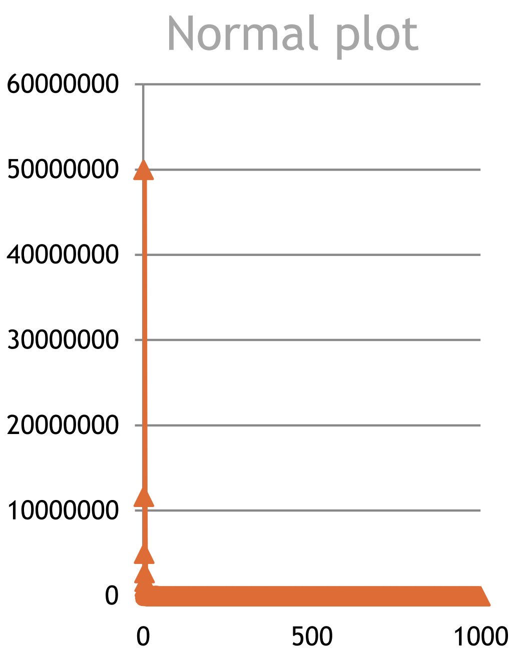
When the **number** of data is **big**, the result of is **not** so **informative**.

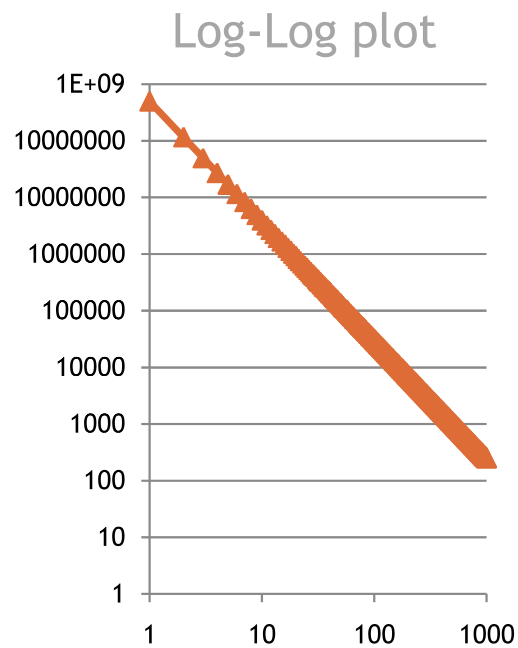


Total quantities of products, which were sold per branch, group by product (for 1000 branches).

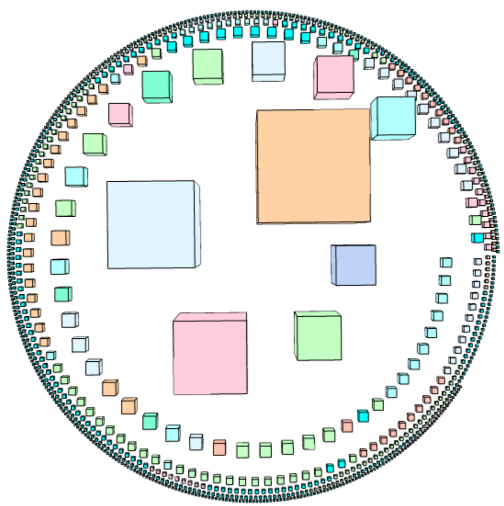
VISUALIZATION FOR ANALYTICS (4/4)

Traditional plots **vs**. the **proposed** method:



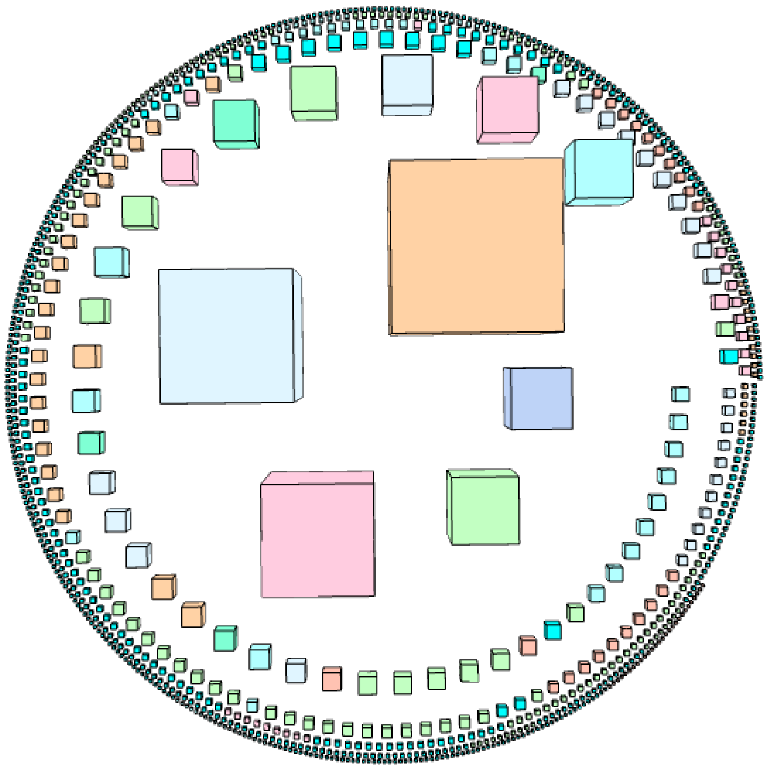


Spiral-layout



Total quantities of products, which were sold per branch, group by product (for 1000 branches).

MORE ON THE SPIRAL LAYOUT



Ref: [Papadaki et. Al. 2018]

- The relative sizes are more clear
- The number of values is more evident
- It is like “coiling” the long tail of the normal plot
- Its complexity is linear

- Moreover
  - The 3rd dimension can be exploited for visualizing an additional function
  - An interactive environment allows the user to zoom in any area and explore the space

An application that visualizes
Only the datasets and their connections
Is accessible at www.ics.forth.gr/isl/3DLod/

CONCLUDING REMARKS

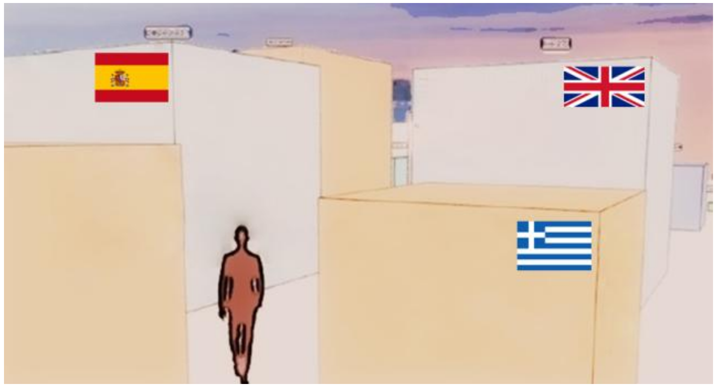
Concluding Remarks

- Several models, languages and **tools** have been developed for data **analysis**.
  - **however**, these tools are not adequate for applying analytics over RDF data.
- Also, the **existing** visualization **tools** are not so informative, when the number of the results is too big.
- So, we are investigating an approach based on:
  - a **high level query language**, for analytics **over RDF graphs** and
  - a 3D **interactive system** for **visualizing** linked data.

FUTURE WORK

Future Work

- We could **extend** our application to support:
  - **more complex analytical queries** over RDF Graphs
  - **incremental algorithms**
- Also, we could design **more layout algorithms** appropriate **for analytics**
  - and perhaps provide **visualizations** with **immersion**, for the **intuitive** and **collaborative** interpretation of the results.



References (1/2)

[1] Spyratos, Nicolas, and Tsuyoshi Sugibuchi. "HiFun-A High Level Functional Query Language for Big Data Analytics."

[2] Papadaki, M. E., Papadakos, P., Mountantonakis, M., & Tzitzikas, Y. (2018). An Interactive 3D Visualization for the LOD Cloud.

[3] Ravindra, Padmashree, Vikas V. Deshpande, and KemaforAnyanwu. "Towards scalable RDF graph analytics on MapReduce." Proceedings of the 2010 Workshop on Massive Data Analytics on the Cloud.ACM, 2010.

[4] Zapilko, Benjamin, and Brigitte Mathiak. "Performing statistical methods on linked data." International conference on dublin core and metadata applications. 2011. – sparql statistical.

[5] Etcheverry, L., Vaisman, A.A.: Enhancing OLAP Analysis with Web Cubes. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 469–483. Springer, Heidelberg (2012).

[6] Etcheverry, Lorena, and Alejandro A. Vaisman. "Efficient Analytical Queries on Semantic Web Data Cubes." Journal on Data Semantics 6.4 (2017): 199-219.

[7] Zhao, Peixiang, et al. "Graph cube: on warehousing and OLAP multidimensional networks." Proceedings of the 2011 ACM SIGMOD International Conference on Management of data.ACM, 2011.

[8] P. Zhao, X. Li, D. Xin, and J. Han. Graph cube: on warehousing and OLAP multidimensional networks. In SIGMOD, pages 853–864, 2011.

References (2/2)

[9] Colazzo, D., Goasdoué, F., Manolescu, I., &Roatiş, A. (2014, April). RDF analytics: lenses over semantic graphs. In Proceedings of the 23rd international conference on World wide web (pp. 467-478). ACM.

[10] Mutlu, Belgin, et al. "Automated Visualization Support for Linked Research Data." I-SEMANTICS (Posters & Demos) 1026 (2013): 40-44.

[11] Bikakis, Nikos, Melina Skourla, and George Papastefanatos. "rdf: SynopsViz–a framework for hierarchical linked data visual exploration and analysis." European Semantic Web Conference.Springer, Cham, 2014.

[12] Wong, Pak Chung, et al. "Have Green-a visual analytics framework for large semantic graphs." Visual Analytics Science And Technology, 2006 IEEE Symposium On. IEEE, 2006.

[14] Agrawal, Rajeev, et al. ”Challenges and opportunities with big data visualization.” Proceedings of the 7th International Conference on Management of computational and collective intElligence in Digital EcoSystems.ACM, 2015.

[15] Wang, Lidong, Guanghui Wang, and Cheryl Ann Alexander. "Big data and visualization: methods, challenges and technology progress." Digital Technologies 1.1 (2015): 33-38.

[16] Vinnik, Svetlana, and Florian Mansmann. "From analysis to interactive exploration: Building visual hierarchies from OLAP cubes." International Conference on Extending Database Technology.Springer, Berlin, Heidelberg, 2006.

[17] Bayerl, Sebastian, and Michael Granitzer. "Discovering, Ranking and Merging RDF Data Cubes." Semantic Computing (ICSC), 2017 IEEE 11th International Conference on. IEEE, 2017.

[18] Köpp, Cornelius, Hans-Jörg von Mettenheim, and Michael H. Breitner. "Decision analytics with heatmap visualization for multi-step ensemble data." *Business & Information Systems Engineering* 6.3 (2014): 131-140.

Links to tools

3DLod: www.ics.forth.gr/isl/3DLod/

THANKS!

QUESTIONS?