RDF Analytics: Lenses over Semantic Graphs

Dario Colazzo, François Goasdoué, Ioana Manolescu, Alexandra Roatis.

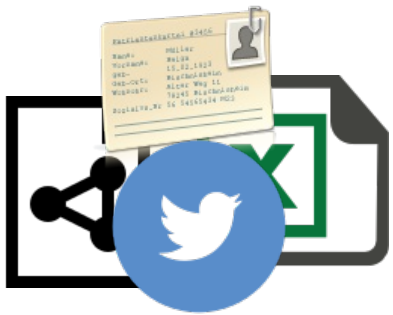
23rd International World Wide Web Conference,
Apr 2014, Seoul, South Korea

Presented by Kim A. Jakobsen (*kjakob09@cs.aau.dk*)
Department of Computer Science, Aalborg University
Study Circle

November 21, 2014

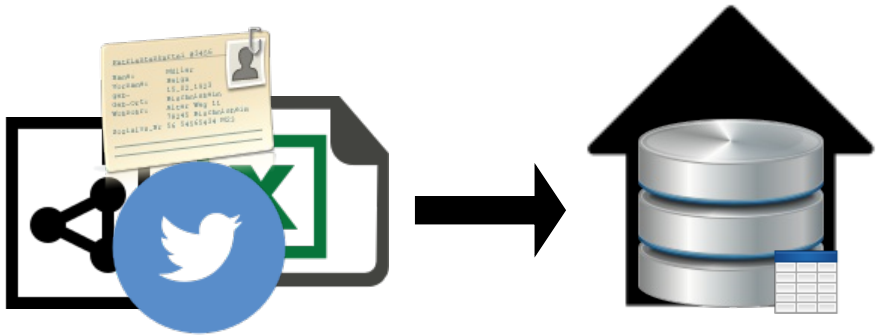
Motivation

Problem



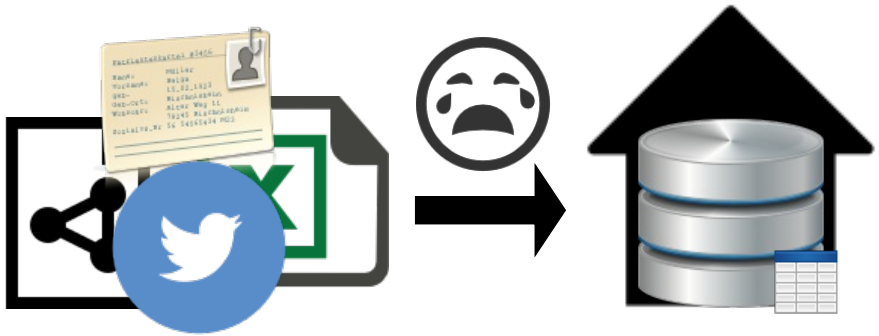
Motivation

Problem



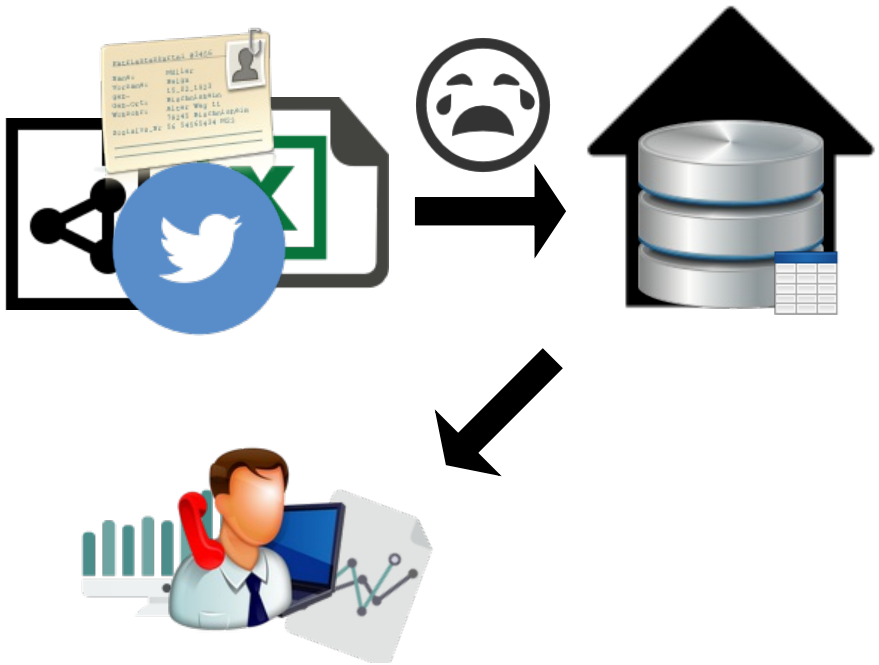
Motivation

Problem



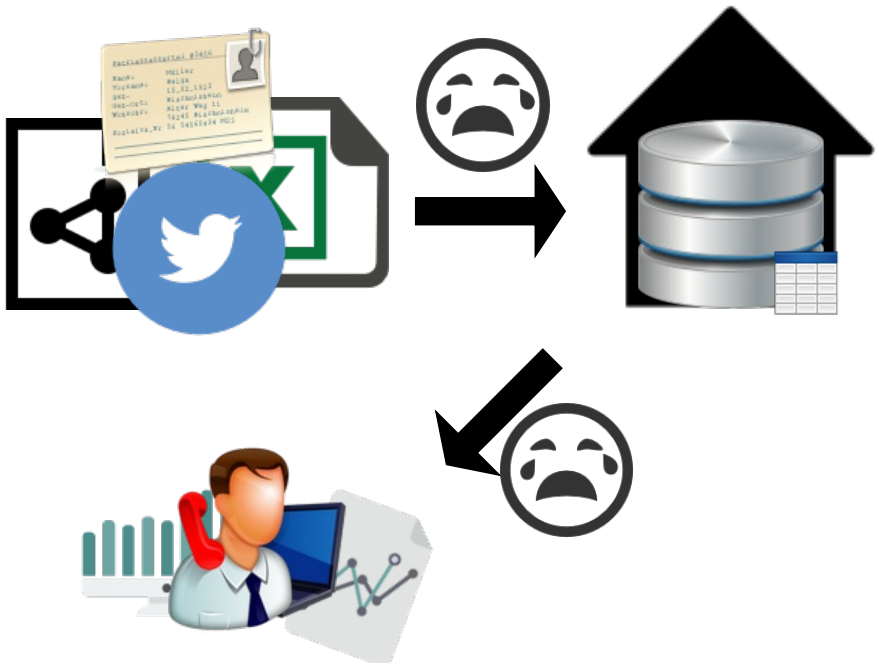
Motivation

Problem



Motivation

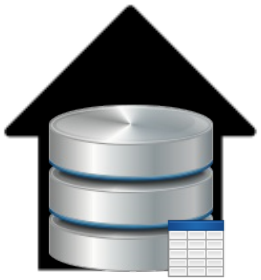
Problem



Motivation

WaRG: Warehousing RDF Graphs

- Full-RDF warehouse
- No single central concept
- Flexible choice of measures and classifiers
- Support cube analysis operations (slice, dice, roll-up ...)



Motivation

WaRG: Warehousing RDF Graphs

- Full-RDF warehouse
- No single central concept
- Flexible choice of measures and classifiers
- Support cube analysis operations (slice, dice, roll-up ...)

Outline

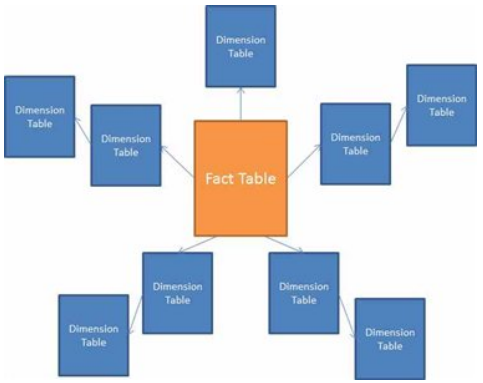
Outline

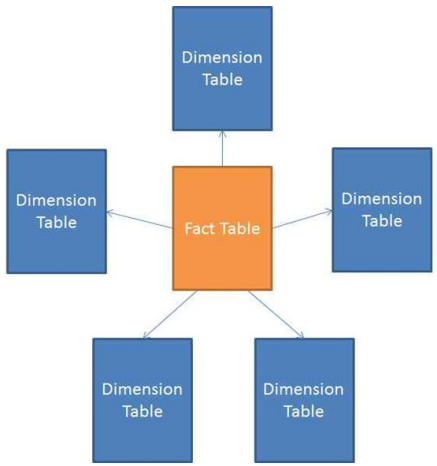
Analytical Schema

- The lense through which the underlying data is observed
- Abstraction over the instance data
- Corresponds to the relation schema
- Flexible
- RDF Graph

Analytical Schema

- The lense through which the underlying data is observed
- Abstraction over the instance data
- Corresponds to the relation schema
- Flexible
- RDF Graph





Images borrowed from `http://social.technet.microsoft.com/`

Analytical Schema

Example

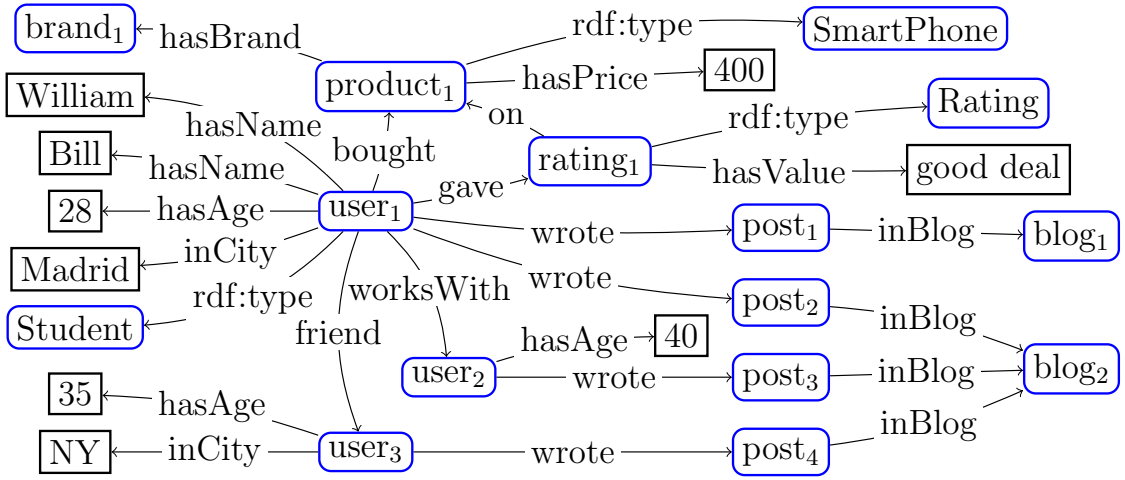


Figure 2: Running example: RDF graph.

Analytical Schema

Example

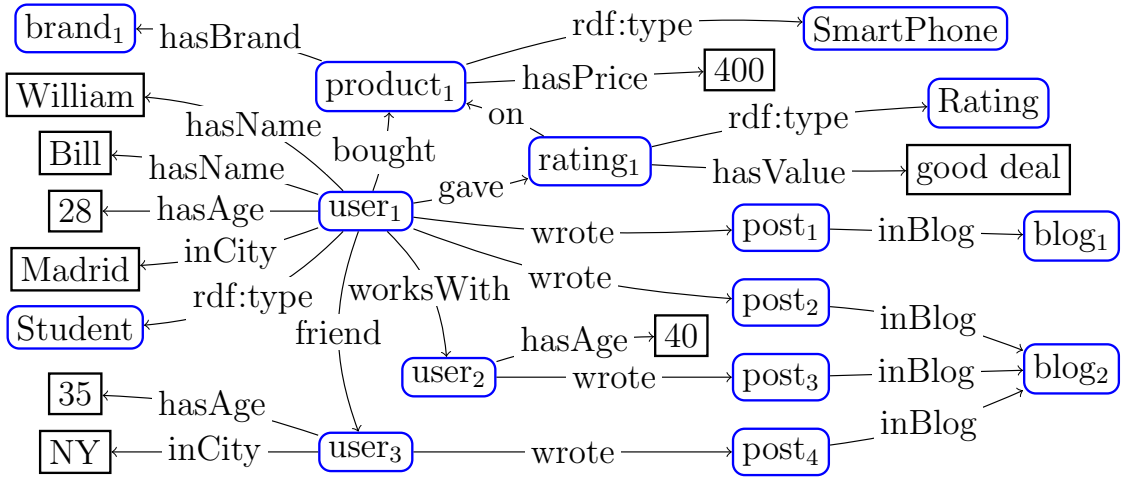


Figure 2: Running example: RDF graph.

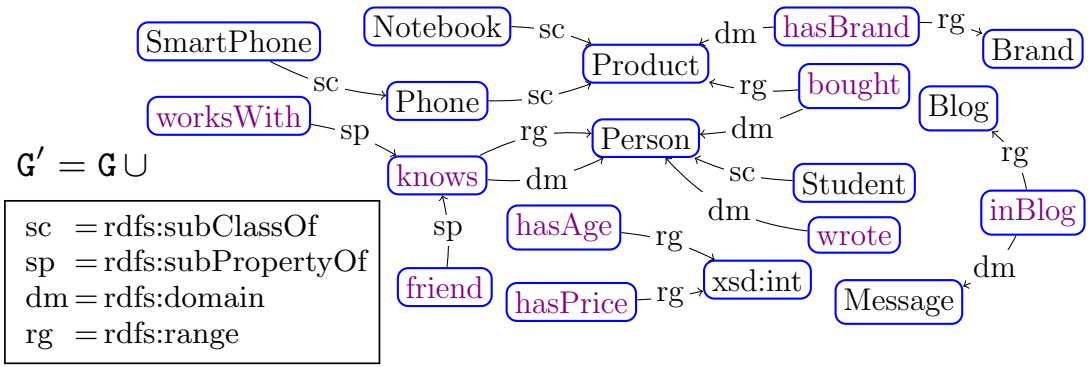
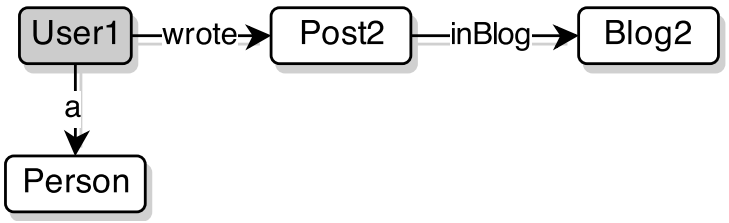


Figure 3: Running example: RDF Schema triples.

Analytical Schema

Nodes and Edges

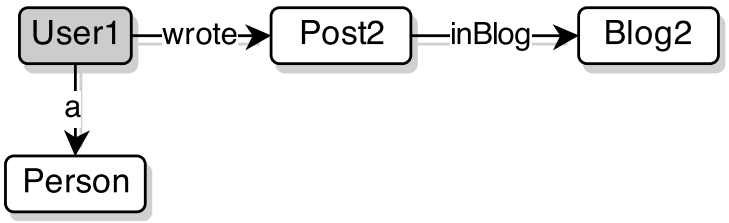
Blogger $= q(x)$:- x *rdf*:*type* *Person*, x *wrote* y, y *inBlog* z



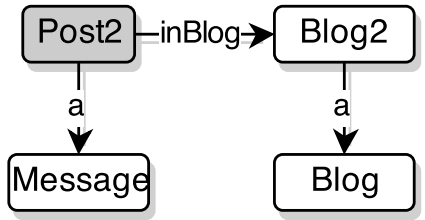
Analytical Schema

Nodes and Edges

Blogger = $q(x)$:- x *rdf*:*type* *Person*, x *wrote* y, y *inBlog* z



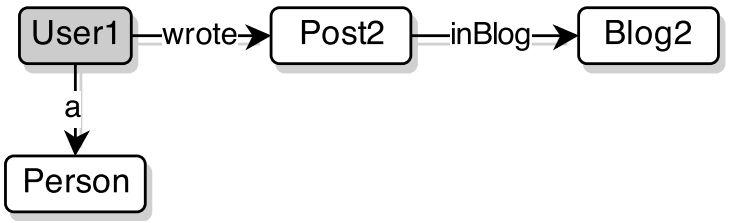
BlogPost = $q(x)$:- x *rdf*:*type* *Message*, x *inBlog* z, z *rdf*:*type* *Blog*



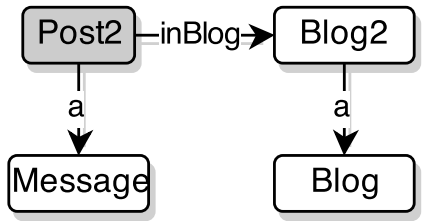
Analytical Schema

Nodes and Edges

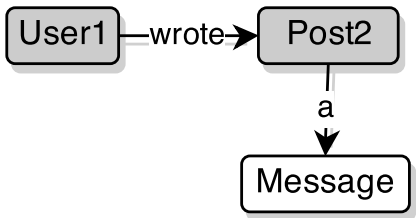
Blogger = $q(x)$:- x *rdf*:*type* *Person*, x *wrote* y, y *inBlog* z



BlogPost = $q(x)$:- x *rdf*:*type* *Message*, x *inBlog* z, z *rdf*:*type* *Blog*



wrotePost = $q(x, y)$:- x *wrote* y, y *rdf*:*type* *Message*



Analytical Schema

AnS

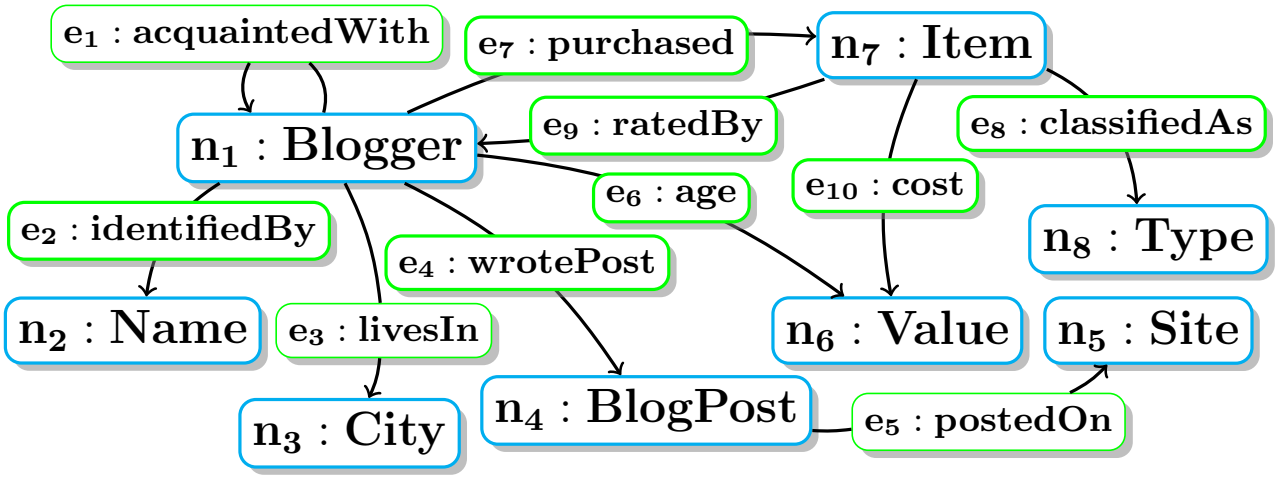


Figure 4: Sample Analytical Schema (AnS).

Analytical Schema

AnS

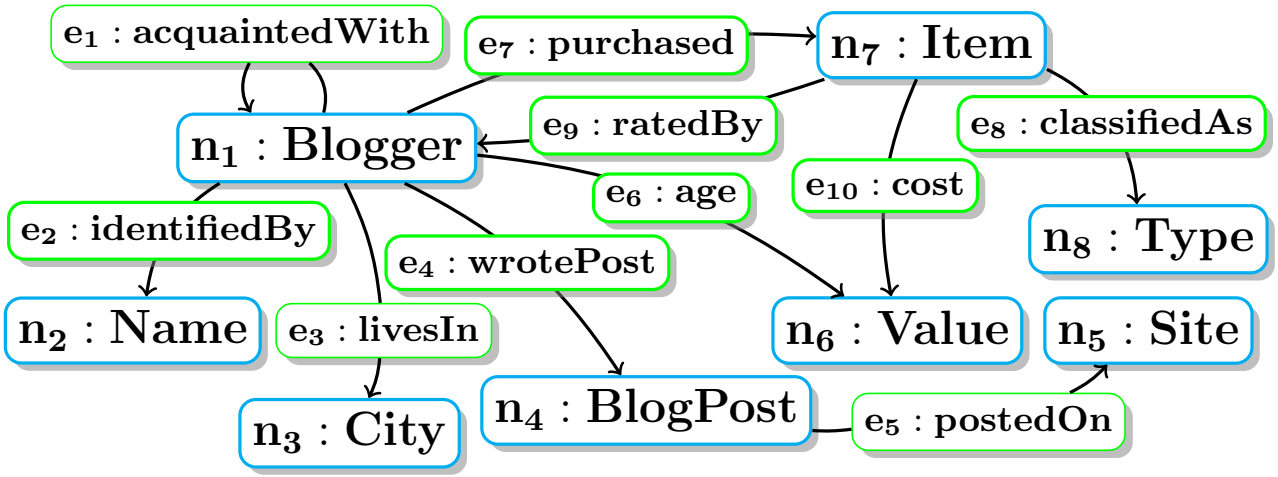


Figure 4: Sample Analytical Schema (AnS).

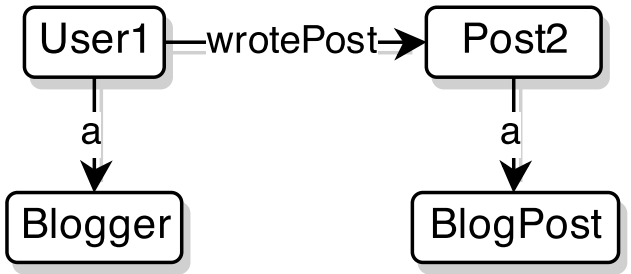


Figure: Analytical Schema Instance Data sample

Outline

Analytical Queries / Cubes

- Analytical Query (AnQ):
  - Classifiers / Dimensions
  - Measures
  - Aggregation function

Analytical Queries / Cubes

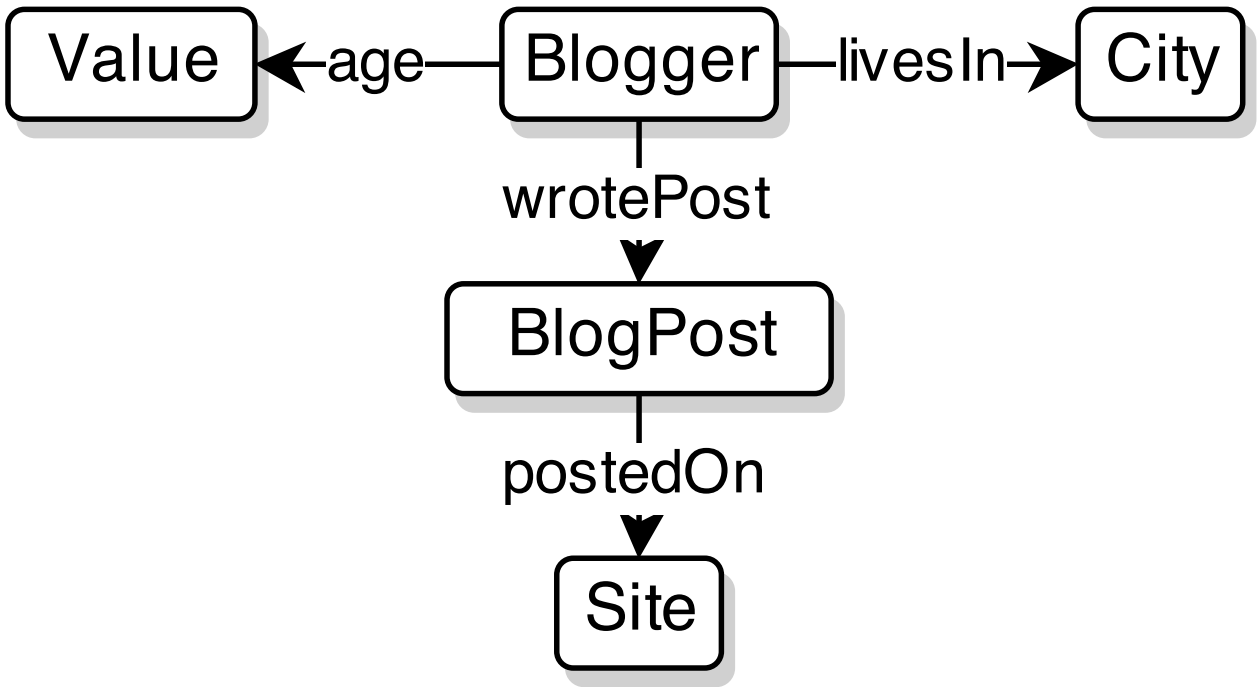
- Analytical Query (AnQ):
  - Classifiers / Dimensions
  - Measures
  - Aggregation function

- Number of sites where each blogger posts, classified by the bloggers's age and city
- $< c(x, y_1, y_2), m(x, z), \oplus >$
  - $c(x, y_1, y_2)$:- x *age* y_1, x *livesIn* y_2
  - $m(x, z)$:- x *wrotePost* y, y *postedOn* z
  - $\oplus$:- *count*

Analytical Queries / Cubes

Example

- Number of sites where each blogger posts, classified by the bloggers's age and city



Outline

Analytical Queries Answering

A Query

- Number of sites where each blogger posts, classified by the bloggers's age and city

```
SELECT ?Blogger ?Age ?City COUNT(*)
WHERE {
      ?Blogger a Blogger .
      ?Blogger wrotePost ?BlogPost .
      ?BlogPost postedOn ?Site .
      ?Blogger age ?Age .
      ?Blogger livesIn ?City .
}
GROUP BY ?Site
```

Analytical Queries Answering

A Query

- Number of sites where each blogger posts, classified by the bloggers's age and city

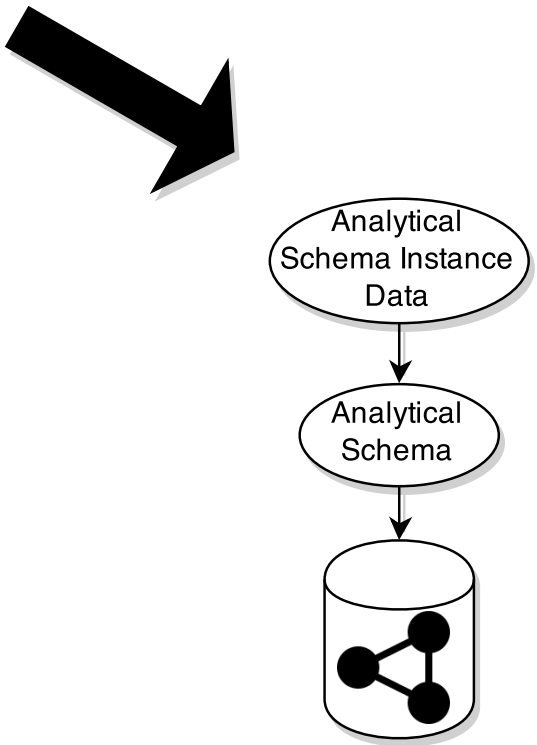
```
SELECT ?Blogger ?Age ?City COUNT(*)
WHERE {
      ?Blogger a Blogger .
      ?Blogger wrotePost ?BlogPost .
      ?BlogPost postedOn ?Site .
      ?Blogger age ?Age .
      ?Blogger livesIn ?City .
}
GROUP BY ?Site
```

- They use the language q not SPARQL

Analytical Queries Answering

Materialization and Query Reformulation

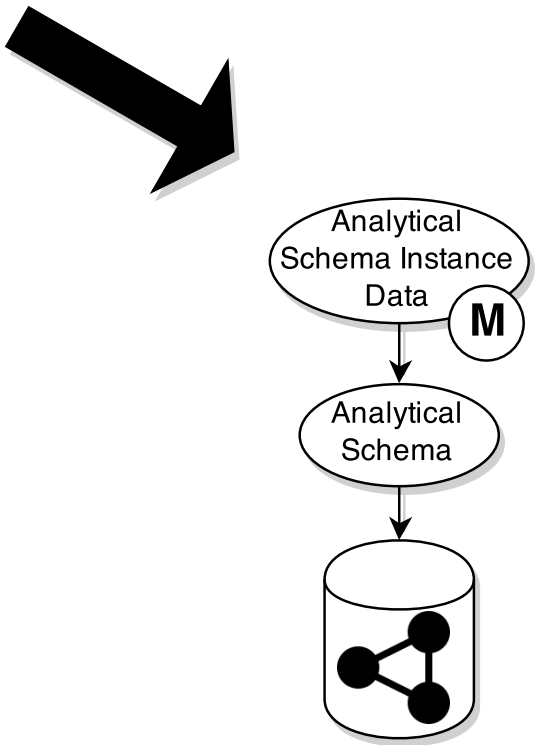
```
SELECT ?Blogger ?Age ?City COUNT(*)
WHERE {
    ?Blogger a Blogger .
    ?Blogger wrotePost ?BlogPost .
    ?BlogPost postedOn ?Site .
    ?Blogger age ?Age .
    ?Blogger livesIn ?City .
}
GROUP BY ?Site
```



Analytical Queries Answering

Materialization and Query Reformulation

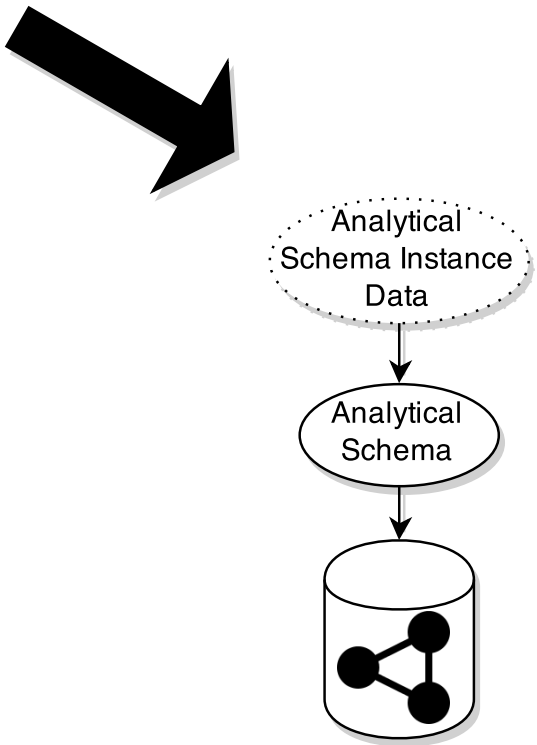
```
SELECT ?Blogger ?Age ?City COUNT(*)
WHERE {
    ?Blogger a Blogger .
    ?Blogger wrotePost ?BlogPost .
    ?BlogPost postedOn ?Site .
    ?Blogger age ?Age .
    ?Blogger livesIn ?City .
}
GROUP BY ?Site
```



Analytical Queries Answering

Materialization and Query Reformulation

```
SELECT ?Blogger ?Age ?City COUNT(*)
WHERE {
    ?Blogger a Blogger .
    ?Blogger wrotePost ?BlogPost .
    ?BlogPost postedOn ?Site .
    ?Blogger age ?Age .
    ?Blogger livesIn ?City .
}
GROUP BY ?Site
```



```
SELECT ?Blogger ?Age ?City COUNT(*)
WHERE {
      ?Blogger a Blogger .
      ?Blogger wrotePost ?BlogPost .
      ?BlogPost postedOn ?Site .
      ?Blogger age ?Age .
      ?Blogger livesIn ?City .
}
GROUP BY ?Site
```

Blogger $= q(x)$:- x *rdf*:*type Person*, x *wrote* y, y *inBlog* z

BlogPost $= q(x)$:- x *rdf*:*type Message*, x *inBlog* z, z *rdf*:*type Blog*

wrotePost $= q(x, y)$:- x *wrote* y, y *rdf*:*type Message*

Analytical Queries Answering

Materialization vs Query Reformulation

- Materilization
  - Fast query times
- Query Reformulation
  - Low Storage Cost
  - No maintanance at update

Outline

Experiments

Setting

- Database
  - kdb+ v3.0
  - Commercial software
  - Query language q
  - In-memory column store
  - Light weight
- Hardware
  - 8-core 2.13 GHz
  - 16 GB RAM
  - Linux Kernal 2.6.31.14
- Data
  - Ontology and Ontology Infobox from DBpedia
  - 34,000,000 triples (4.4 GB)
  - Scale by replicate data

Experiments

Materialization (1)

Blogger(3) = $q(x)$:-x *rdf*:*type Person*, x *wrote* y, y *inBlog* z

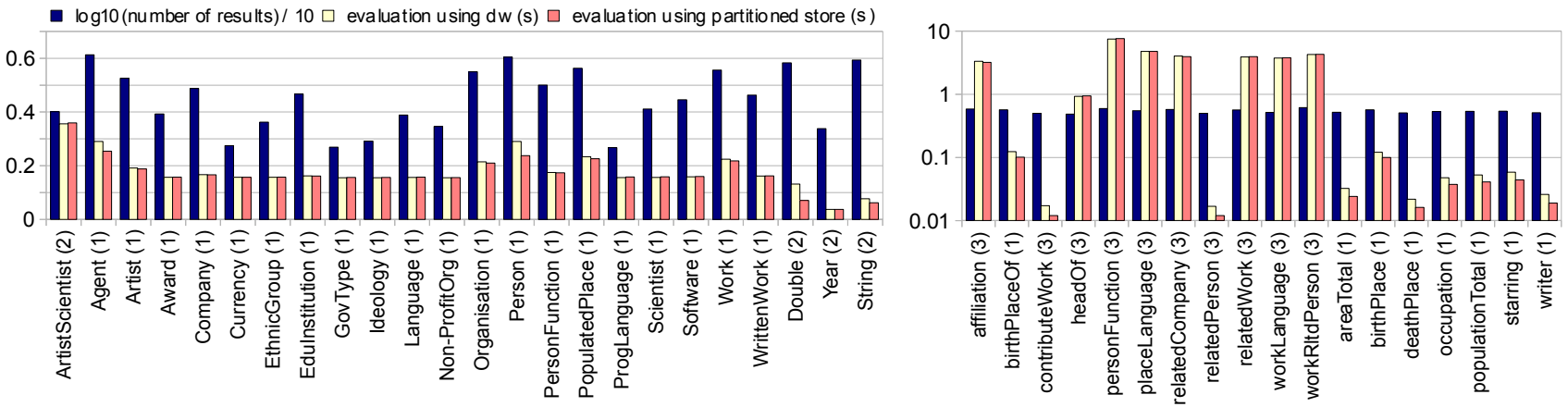


Figure 6: Evaluation time (s) and number of results for AnS node queries (left) and edge queries (right).

Experiments

Materialization (1)

Blogger(3) = $q(x)$:-x *rdf*:*type Person*, x *wrote* y, y *inBlog* z

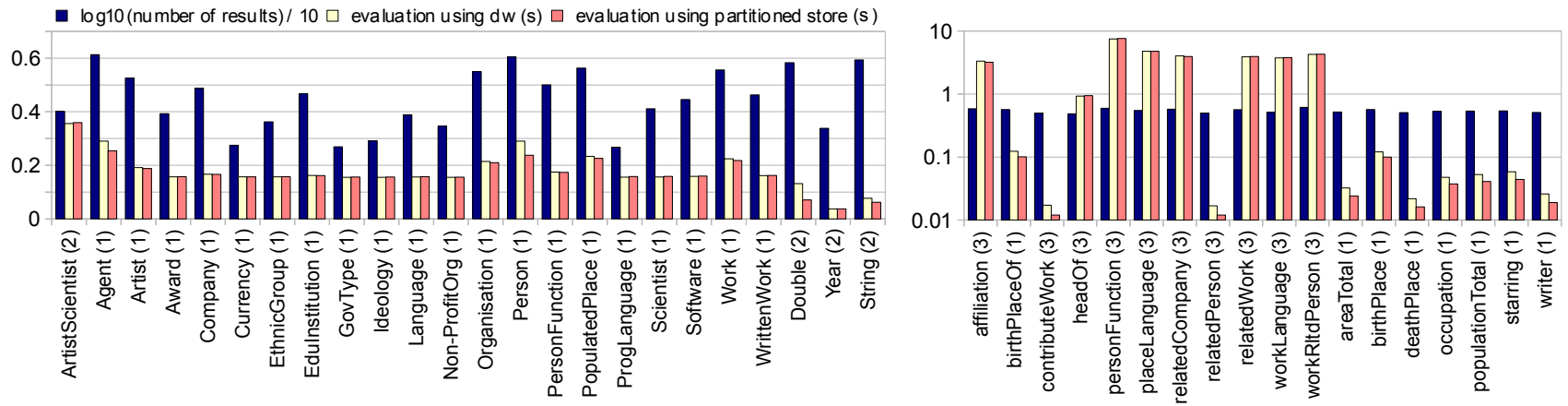


Figure 6: Evaluation time (s) and number of results for AnS node queries (left) and edge queries (right).

- 2 node queries and 57 edge queries are omitted because they are below 0.01 seconds

Experiments

Materialization (2)

- Instance table (Subject, Predicate, Object)
- Partitioned table Node(Subject) Edge(Subject, Object)

Experiments

Materialization (2)

- Instance table (Subject, Predicate, Object)
- Partitioned table Node(Subject) Edge(Subject, Object)

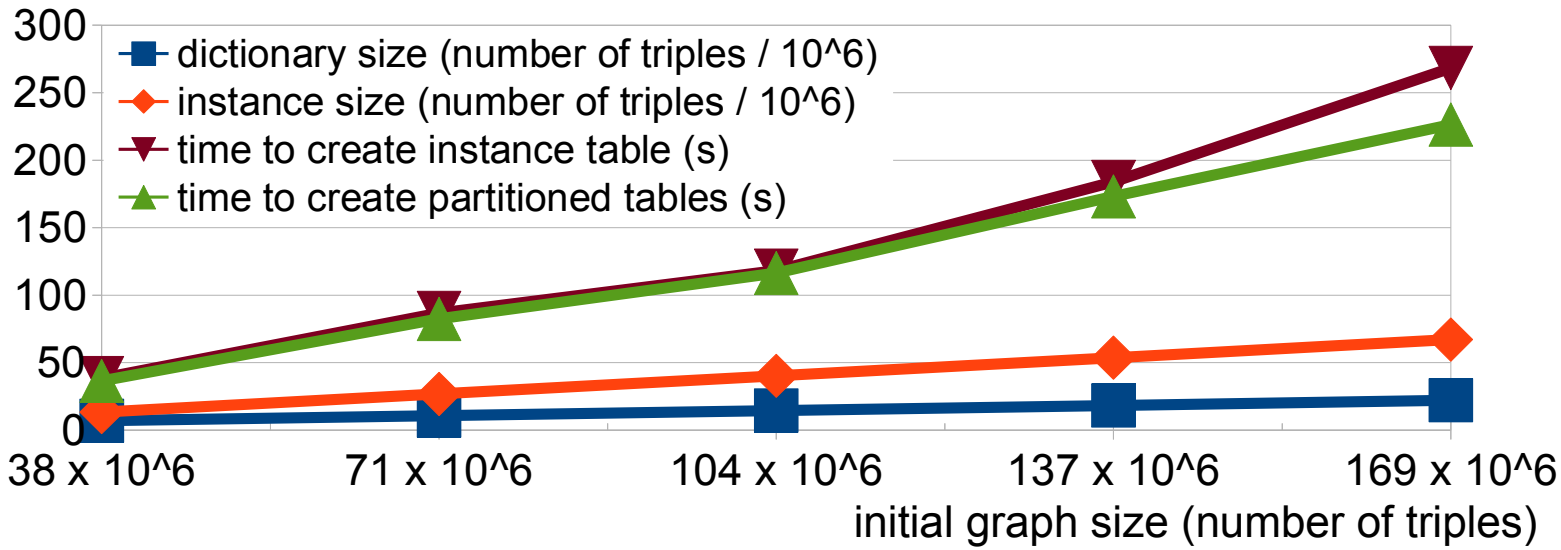


Figure 7: $\mathcal{I}$ materialization time vs. $\mathcal{I}$ size.

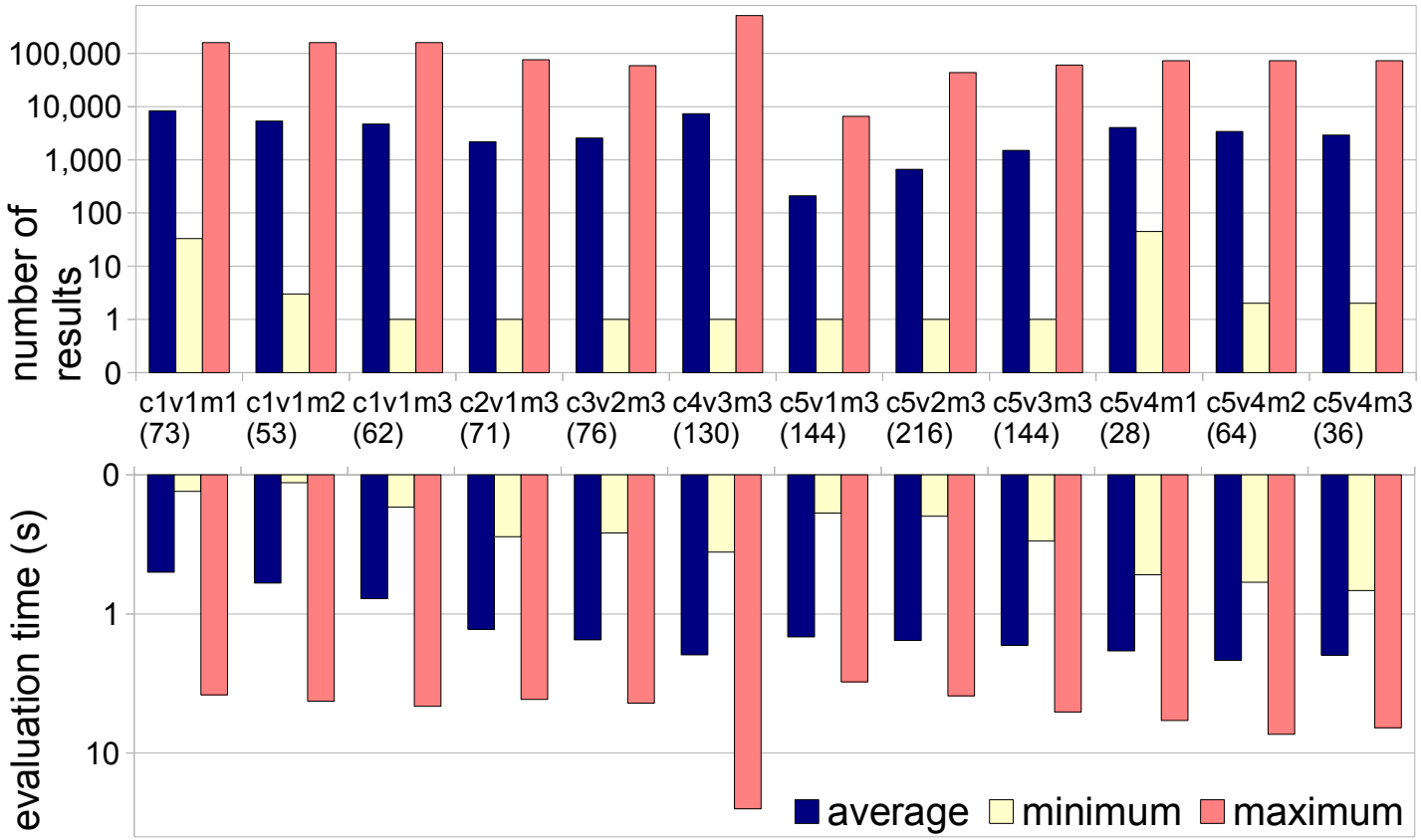


Figure 8: AnQ statistics for query patterns.

Experiments

Materialized Analytical Queries Scaling

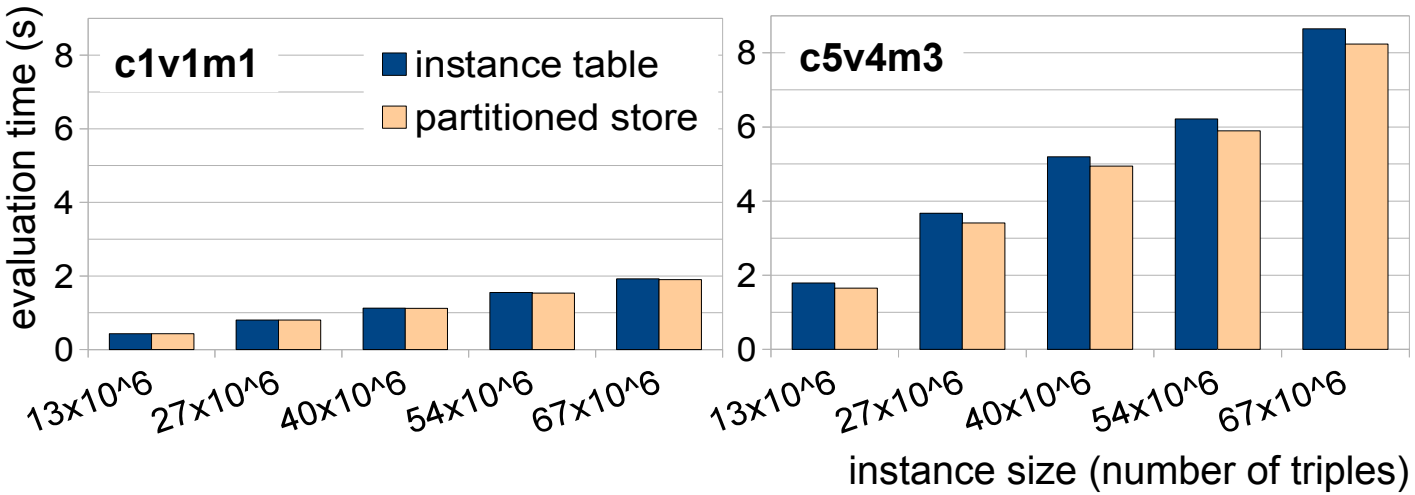


Figure 9: AnQ evaluation time over large datasets.

Outline

Conclusion

- RDF data warehouse
- Dynamic Analytical Schema to model data
- Make Analytical Queries (cubes)
- Linear scale of Analytical Queries (cubes)

Outline

Strengths

- The story about Alice (Context) :)

Strengths

- The story about Alice (Context) :)
- Well written (Art of complicated writing)

Strengths

- The story about Alice (Context) :)
- Well written (Art of complicated writing)
- Relational concepts get new Semantic Web names
  - Analytical Schema = (Snowflake/star) Schema
  - Analytical Queries = Cube
  - OLAP (Specified)
  - Dimensions are define by a classifier query

Strengths

- The story about Alice (Context) :)
- Well written (Art of complicated writing)
- Relational concepts get new Semantic Web names
  - Analytical Schema = (Snowflake/star) Schema
  - Analytical Queries = Cube
  - OLAP (Specified)
  - Dimensions are define by a classifier query
- Precise formulations

Strengths

- The story about Alice (Context) :)
- Well written (Art of complicated writing)
- Relational concepts get new Semantic Web names
  - Analytical Schema = (Snowflake/star) Schema
  - Analytical Queries = Cube
  - OLAP (Specified)
  - Dimensions are define by a classifier query
- Precise formulations
- Running example throughout the paper

Strengths

- The story about Alice (Context) :)
- Well written (Art of complicated writing)
- Relational concepts get new Semantic Web names
  - Analytical Schema = (Snowflake/star) Schema
  - Analytical Queries = Cube
  - OLAP (Specified)
  - Dimensions are define by a classifier query
- Precise formulations
- Running example throughout the paper
- Two materializations strategies
  - Instance table
  - Partitioned tables

Strengths

- The story about Alice (Context) :)
- Well written (Art of complicated writing)
- Relational concepts get new Semantic Web names
  - Analytical Schema = (Snowflake/star) Schema
  - Analytical Queries = Cube
  - OLAP (Specified)
  - Dimensions are define by a classifier query
- Precise formulations
- Running example throughout the paper
- Two materializations strategies
  - Instance table
  - Partitioned tables
- Basic OLAP support

Weaknesses

- No query reformulation results, this is the hard part :(
  - In another paper (informal French conference)

Weaknesses

- No query reformulation results, this is the hard part :(
  - In another paper (informal French conference)
- What kind of index do they use on their data?
  - PSOC?

Weaknesses

- No query reformulation results, this is the hard part :(
  - In another paper (informal French conference)
- What kind of index do they use on their data?
  - PSOC?
- Analytical Queries (Cubes) can only have one type of aggregation

Weaknesses

- No query reformulation results, this is the hard part :(
  - In another paper (informal French conference)
- What kind of index do they use on their data?
  - PSOC?
- Analytical Queries (Cubes) can only have one type of aggregation
- Simple cubes in experiments (not complex)
  - No "nextLevel" in the dataset
  - No aggregation
  - No unbalanced hierarchies

Weaknesses

- No query reformulation results, this is the hard part :(
  - In another paper (informal French conference)
- What kind of index do they use on their data?
  - PSOC?
- Analytical Queries (Cubes) can only have one type of aggregation
- Simple cubes in experiments (not complex)
  - No "nextLevel" in the dataset
  - No aggregation
  - No unbalanced hierarchies
- No BI queries :S They only have "setup" times!
  - Time to get Analytical Schema Nodes and Edges
  - Creation time of Analytical Queries (cubes) and how this scales

Questions?