Supported by

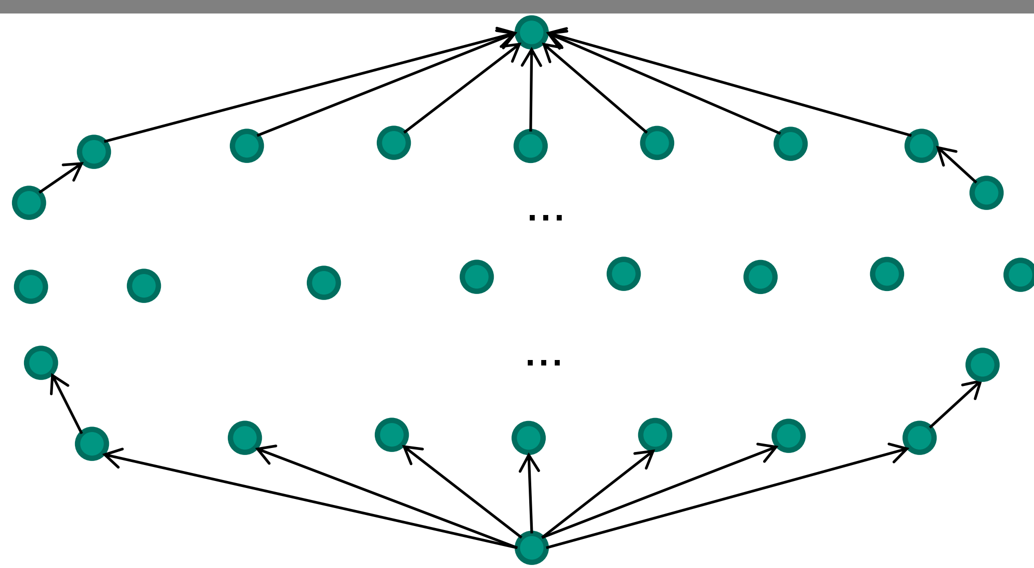
No Size Fits All – Running the Star Schema Benchmark with SPARQL and RDF Aggregate Views

Benedikt Kämpgen, Andreas Harth

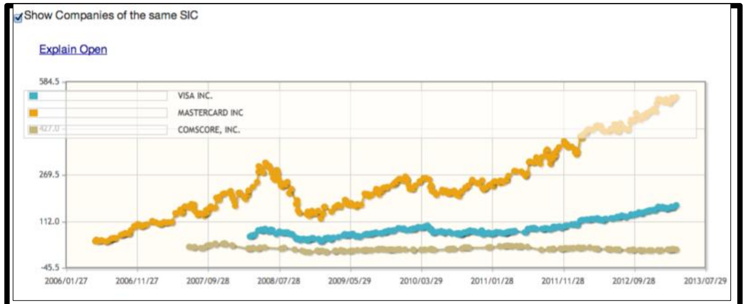
10th ESWC 2013 Semantics and Big Data

29 May 2013

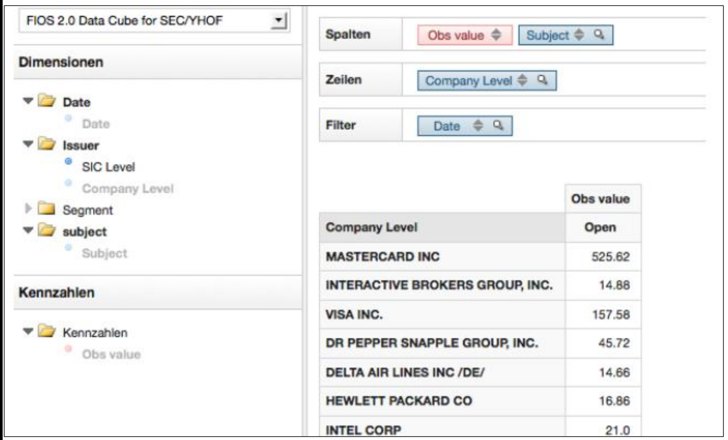
Institute of Applied Informatics and Formal Description Methods (AIFB)



Motivation



Tr-Price for selected companies



Company Level	Obs value Open
MASTERCARD INC	525.62
INTERACTIVE BROKERS GROUP, INC.	14.88
VISA INC.	157.58
DR PEPPER SNAPPLE GROUP, INC.	45.72
DELTA AIR LINES INC /DE/	14.66
HEWLETT PACKARD CO	16.86
INTEL CORP	21.0

Tr-Price aggregated over time

Data analytics

Analytical queries

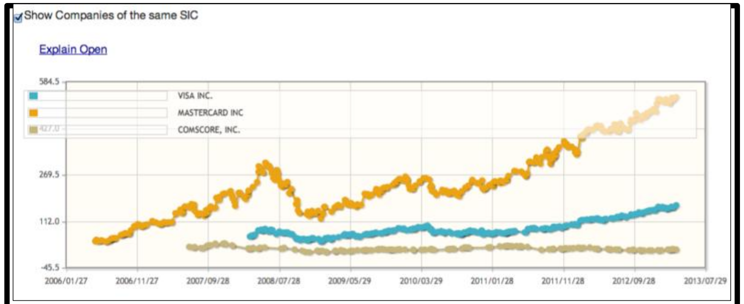
Query engine

Query results

Statistical Linked Data

http://wiki.planet-data.eu/web/Datasets

Motivation



Filter for Visa Inc.

Company	Dates	Average Trading Price
Visa Inc.	July 1, 2007	**80 $**
Visa Inc.	July 1, 2009	**90 $**
...	...	**...**

Data analytics

Analytical queries

Query engine

Query results

Metadata

RDF Data Cube Vocabulary [QB]

Data

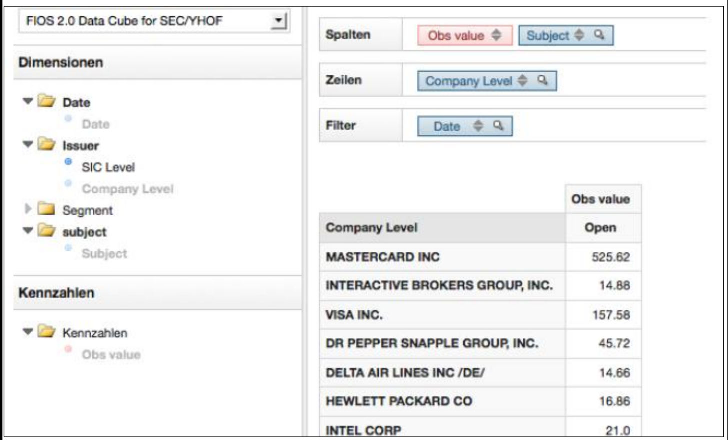
Company	Dates	Average Trading Price
Visa Inc.	July 1, 2008	**80 $**
Mastercard Inc.	July 1, 2008	**269 $**
Visa Inc.	July 1, 2009	**90 $**
...	...	**...**

Example Statistical Linked Data

Motivation

Aggregation over Dates

Company	Dates	Average Trading Price
Visa Inc.	ALL	**85 $**
Mastercard Inc.	ALL	**269 $**
...	...	**...**



Data analytics

Analytical queries

Query engine

Query results

Metadata

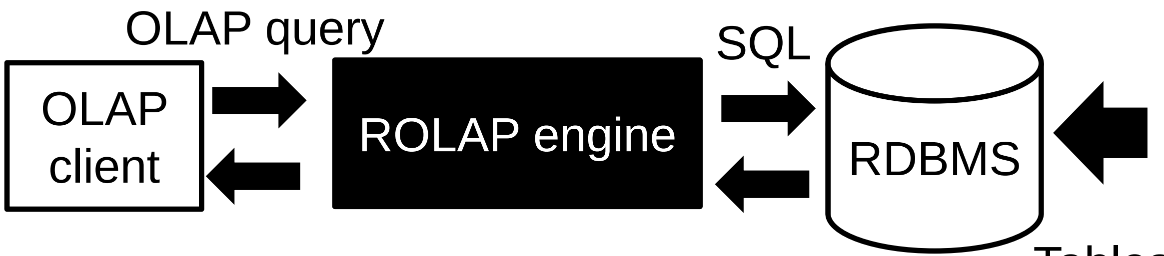
RDF Data Cube Vocabulary [QB]

Data

Company	Dates	Average Trading Price
Visa Inc.	July 1, 2008	**80 $**
Mastercard Inc.	July 1, 2008	**269 $**
Visa Inc.	July 1, 2009	**90 $**
...	...	**...**

Example Statistical Linked Data

Problem



- Fixed Star Schema
- Optimisations: aggregate tables

Metadata

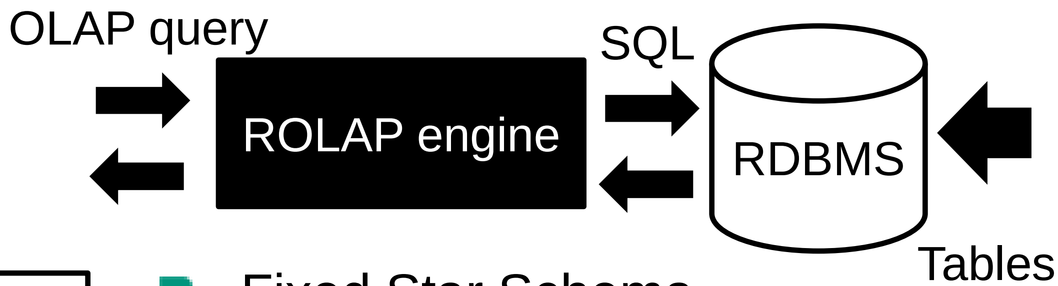
RDF Data Cube Vocabulary [QB]

Data

Company	**Dates**	**Average Trading Price**
Visa Inc.	July 1, 2008	**80 $**
Mastercard Inc.	July 1, 2008	**269 $**
...	...	**...**

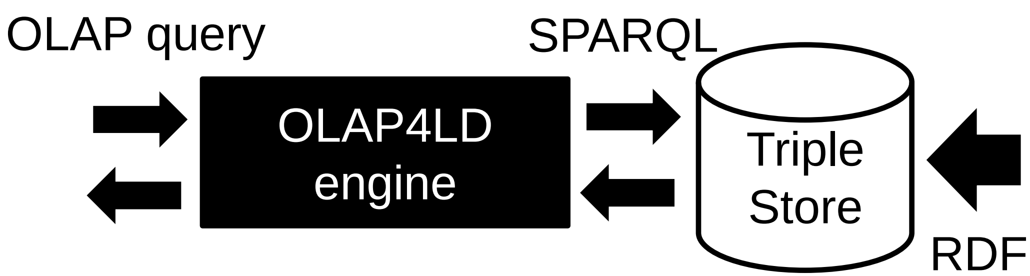
Statistical Linked Data

Problem



OLAP client

- Fixed Star Schema
- Optimisations: aggregate tables



Metadata

RDF Data Cube Vocabulary [QB]

Data

Company	**Dates**	**Average Trading Price**
Visa Inc.	July 1, 2008	**80 $**
Mastercard Inc.	July 1, 2008	**269 $**
...	...	**...**

- Motivation: Easier integration with other data sources
  - Filter for companies from cities with more than 100.000 inhabitants
  - Filter for companies that have a CEO who is younger than 20

▶ Problem: Performance? Materialised Aggregate Views?

Problem: ROLAP vs OLAP4LD – Performance? Materialised Aggregate Views?

	No Materialisation	Materialisation
RDBMS/ SQL (MySQL)	ROLAP (Mondrian)	ROLAP-M (Mondrian)
Triple Store/ SPARQL (Open Virtuoso)	OLAP4LD (olap4ld)	OLAP4LD-M (olap4ld)

Institute of Applied Informatics and Formal Description Methods (AIFB)

Related Work

We compare ROLAP and OLAP on a common triple store.

- Query optimisation on RDF: SPARQL over MapReduce [1], Column-Stores [2]

We use [QB] to represent aggregate views in RDF.

- Vocabularies for statistics: [SCOVO], [QB4OLAP], [QB]

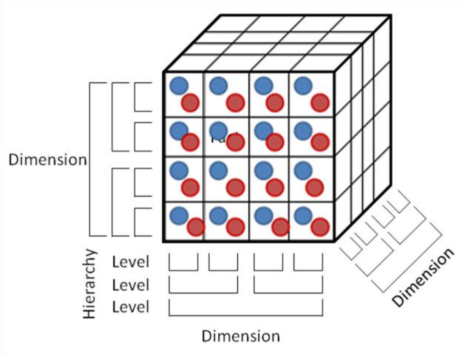
We materialise aggregate views with SPARQL INSERT queries and run a realistic benchmark with >100M triples.

- Views over RDF datasets [3]: Networked Graphs, vSPARQL

Outline

1. Data Analytics on Statistics from the Web
2. **Star Schema Benchmark (SSB)**
3. Comparing OLAP on RDBMS and on Triple Store
4. Evaluating Materialised Aggregate Views
5. Conclusions

Star Schema Benchmark [SSB] – Data



SSB Data Cube with Lineorders

Measures

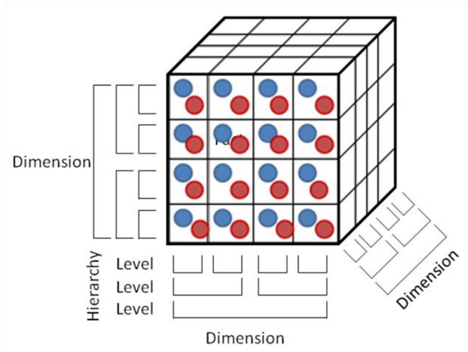
- Sum of revenues ...

Dimensions with Hierarchies

- Dates (→ Month → Year → ALL)
- Supplier (→ City → Nation → Region → ALL) ...

Metadata

Star Schema Benchmark [SSB] – Data



SSB Data Cube with Lineorders

Measures

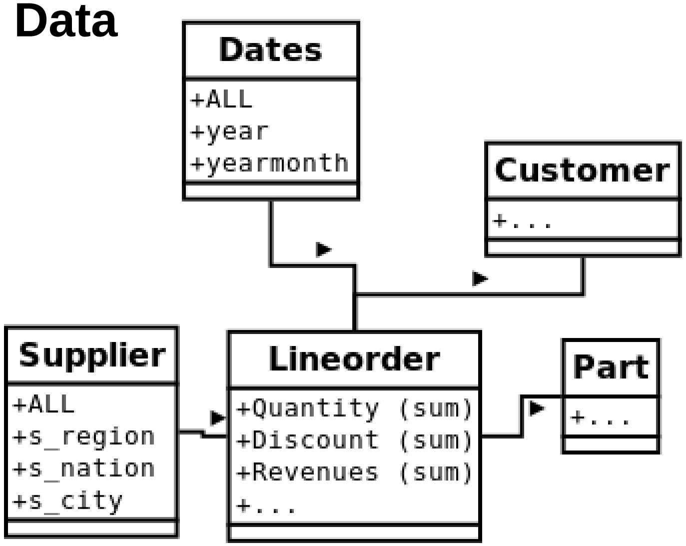
- Sum of revenues ...

Dimensions with Hierarchies

- Dates (→ Month → Year → ALL)
- Supplier (→ City → Nation → Region → ALL) ...

Metadata

Data



Dates	Month	Year
January 18, 1992	Jan, 199	1992
February 8, 1992	Feb 199	
...	...	

Dimension table with hierarchy

Dates	Customer	Part	Supplier	Revenue	...
January 18, 1992	Customer# 000009	olive drab	Supplier #0001	**950,915**	...
February 8, 1992	Customer# 000027	lace lemon	Supplier #0035	**593,625**	...
...	...	...	...	...	

Fact table with 6,000,000 lineorders at Scale 1 of SSB

Star Schema Benchmark (SSB) – Queries

Q2.1
*Revenues per **year***
*and per **product brands***
*of product category **MFGR#12***
*and of supplier from **America.***

Query

Aggregation (aka Rollup)

- Dates → Year
- Part → Brand
- Customer → ALL ...

Filter (aka Dice)

- Part.Category = MFGR#12
- Supplier.Region = AMERICA

Star Schema Benchmark (SSB) – Queries

Q2.1
*Revenues per **year***
*and per **product brands***
*of product category **MFGR#12***
*and of supplier from **America.***

Aggregation (aka Rollup)

- Dates → Year
- Part → Brand
- Customer → ALL ...

Filter (aka Dice)

- Part.Category = MFGR#12
- Supplier.Region = AMERICA

Query

Result

Year\Brand	MFGR#121	**...**
1992	667,692,830	...
...	...	...

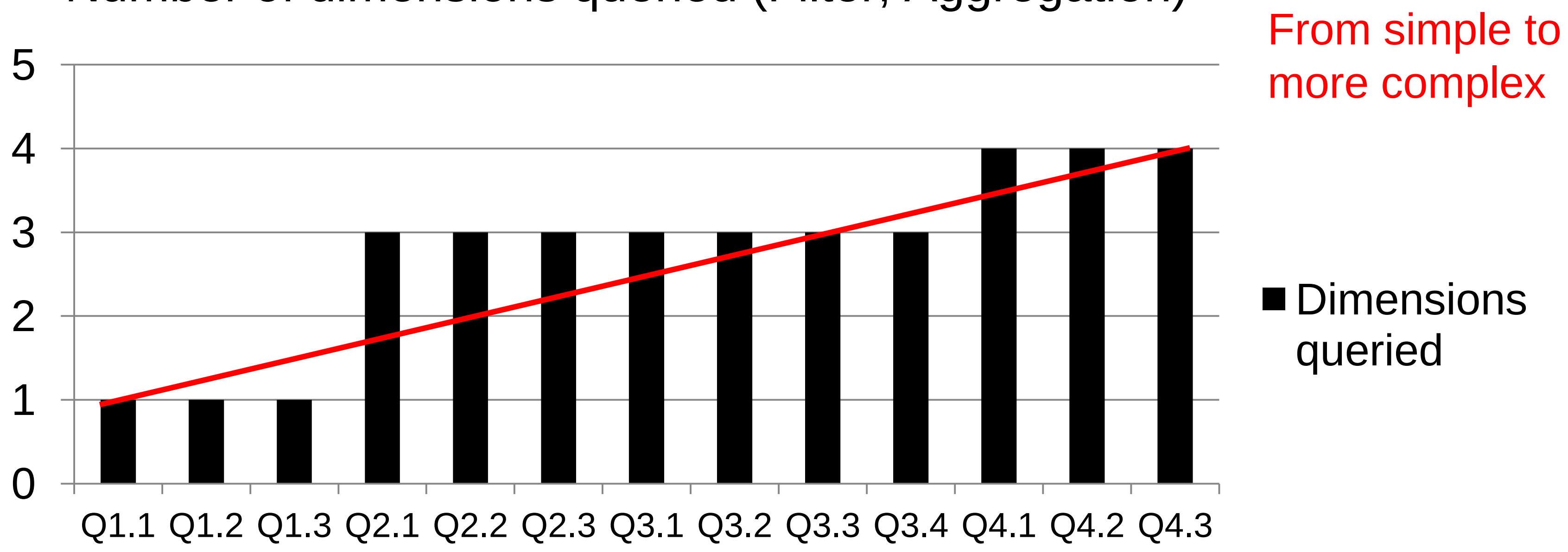
Result displayed in pivot table

Calculated from

Dates	Customer	Part	Supplier	Revenue	...
January 18, 1992	Customer# 000009	olive drab	Supplier #0001	**950,915**	**...**
February 8, 1992	Customer# 000027	lace lemon	Supplier #0035	**593,625**	**...**
...	...	...	...	...	

Star Schema Benchmark (SSB) – 4 Query Flights of 13 Queries

- Number of dimensions queried (Filter, Aggregation)



Experimental Setup

Hardware	
OS	Debian Linux 6.0.6
CPU	2 x 2.60GHz with 16 cores
Disk	900GB RAID10 on 15k SAS disks
RAM	128GB

Software	
DB setup	Limited memory (400MB for RDBMS, 600MB for Triple Store)
SSB data	Manually translated, loaded according to approach (ROLAP, OLAP4LD...)
SSB queries	Manually translated according to OLAP engine (Mondrian, olap4ld)
Results	Elapsed query times on hot runs with Business Intelligence Benchmark Framework (BIBM)

Paper website: http://people.aifb.kit.edu/bka/ssb-benchmark/

Outline

1. Data Analytics on Statistics from the Web
2. Star Schema Benchmark (SSB)
3. **Comparing OLAP on RDBMS and on Triple Store**
4. Evaluating Materialised Aggregate Views
5. Conclusions

Problem: ROLAP vs OLAP4LD – Performance? Materialised Aggregate Views?

	No Materialisation	**Materialisation**
RDBMS/ SQL (MySQL)	ROLAP (Mondrian)	ROLAP-M (Mondrian)
Triple Store/ SPARQL (Open Virtuoso)	OLAP4LD (olap4ld)	OLAP4LD-M (olap4ld)

Institute of Applied Informatics and Formal Description Methods (AIFB)

ROLAP vs OLAP4LD – Results

- **Elapsed query time (eqt)**

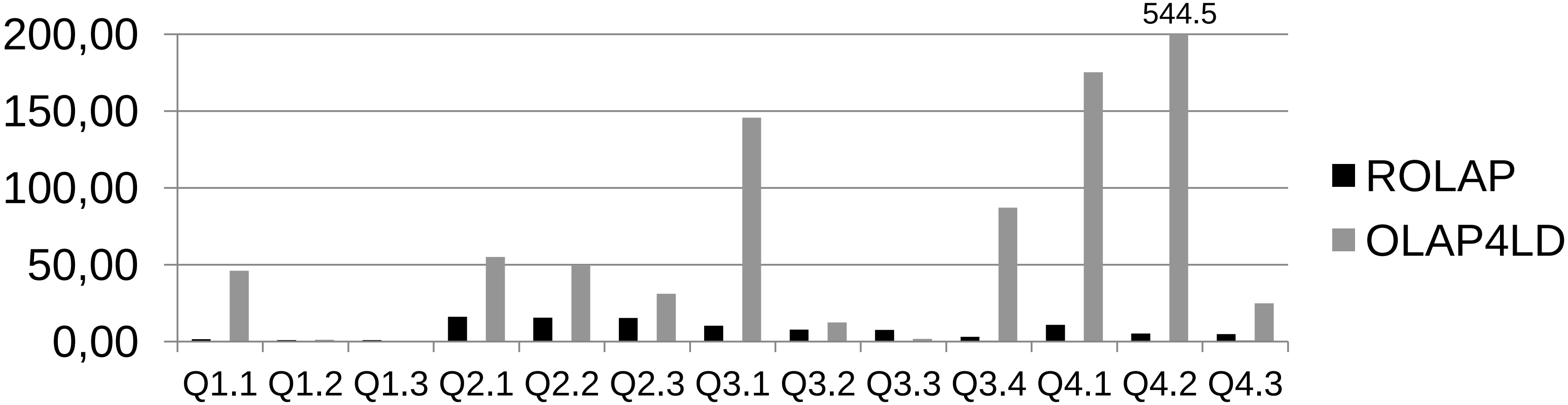
  OLAP4LD overall 12 times slower than ROLAP for executing all queries.

$$\frac{\sum eqt_{olap4ld}(q_i)}{\sum eqt_{rolap}(q_i)} \approx 12$$

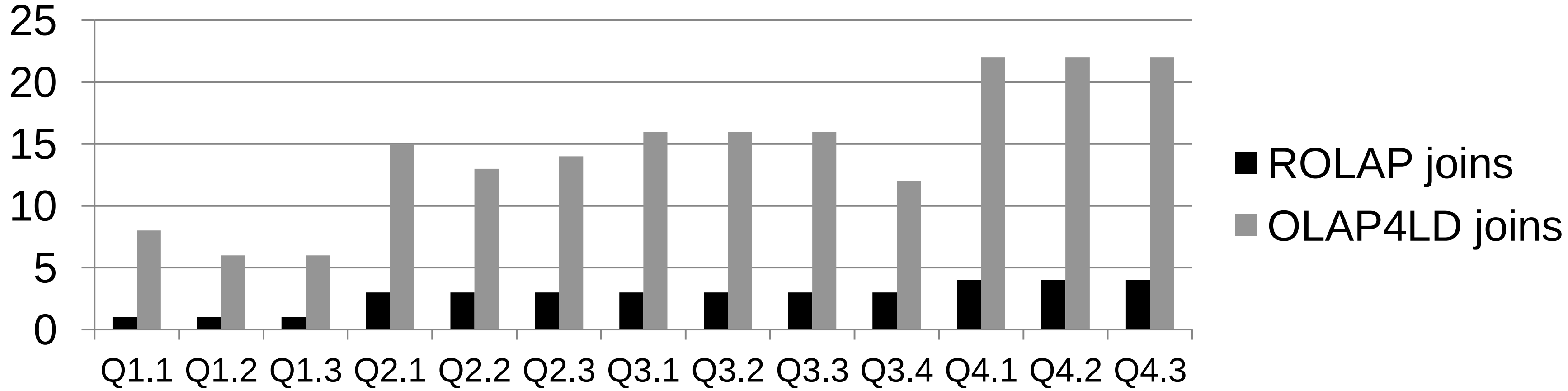
Institute of Applied Informatics and Formal Description Methods (AIFB)

ROLAP vs OLAP4LD – Query processing

OLAP4LD overall 12 times slower than ROLAP for executing all queries?



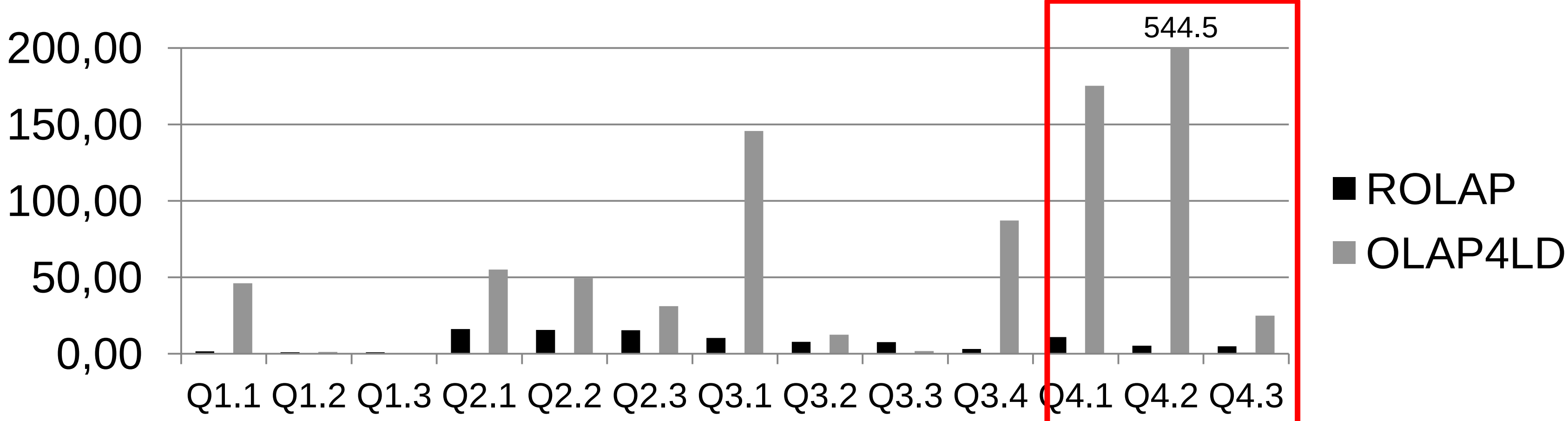
Hypothesis: Numerous joins for [QB] hierarchies



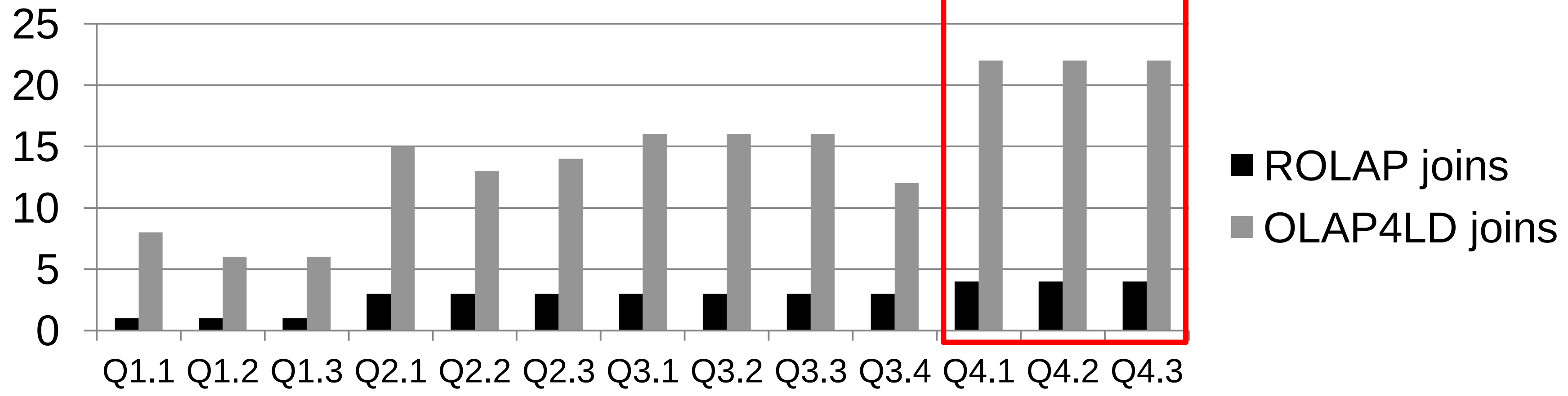
Institute of Applied Informatics and
Formal Description Methods (AIFB)

ROLAP vs OLAP4LD – Query processing

OLAP4LD overall 12 times slower than ROLAP for executing all queries?



Hypothesis: Numerous joins for [QB] hierarchies

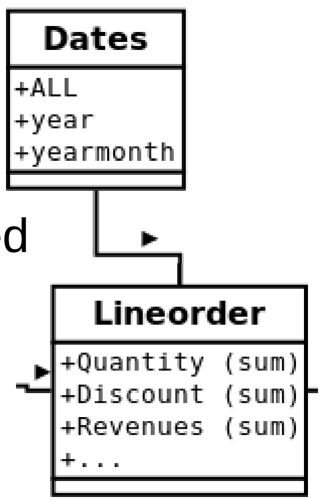


ROLAP vs OLAP4LD – Query processing

OLAP4LD overall 12 times slower than ROLAP for executing all queries?

ROLAP

Dates	Month	Year
January 18, 1992	Jan, 1992	1992
February 8, 1992	Feb, 1992	1992
...	...	



Denormalised Dimension Table

Aggregation (Rollup)
Dates → Year

ROLAP vs OLAP4LD – Query processing

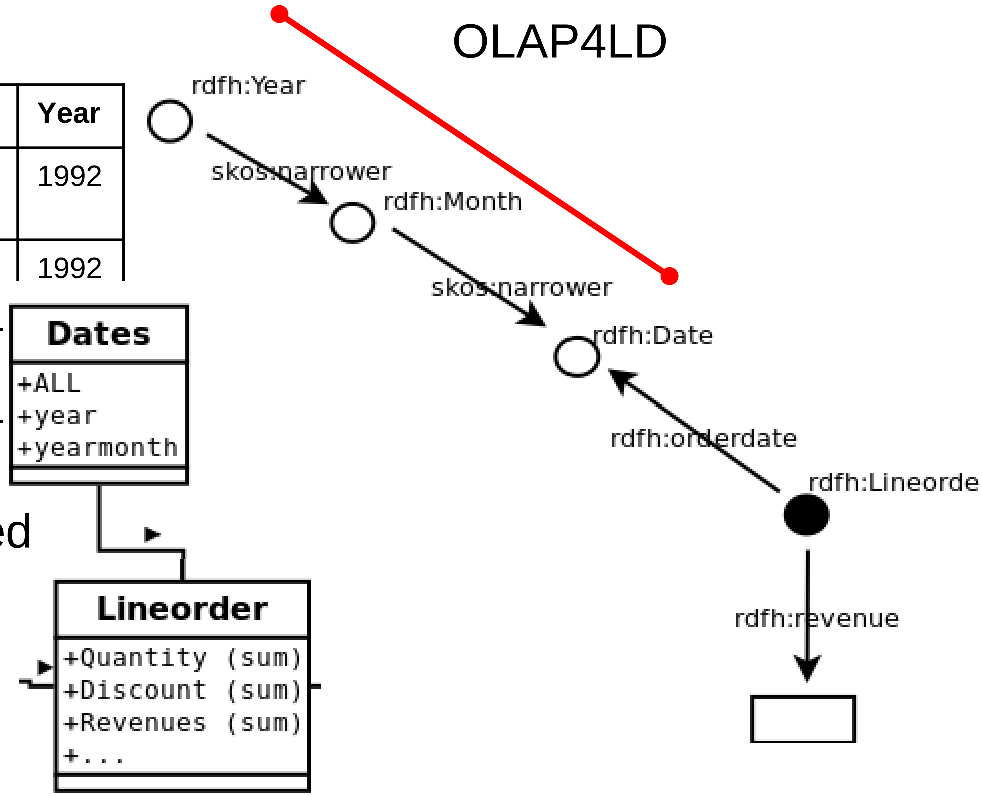
OLAP4LD overall 12 times slower than ROLAP for executing all queries?

ROLAP

Dates	Month	Year
January 18, 1992	Jan, 1992	1992
February 8, 1992	Feb, 1992	1992
...	...	

Denormalised Dimension Table

OLAP4LD



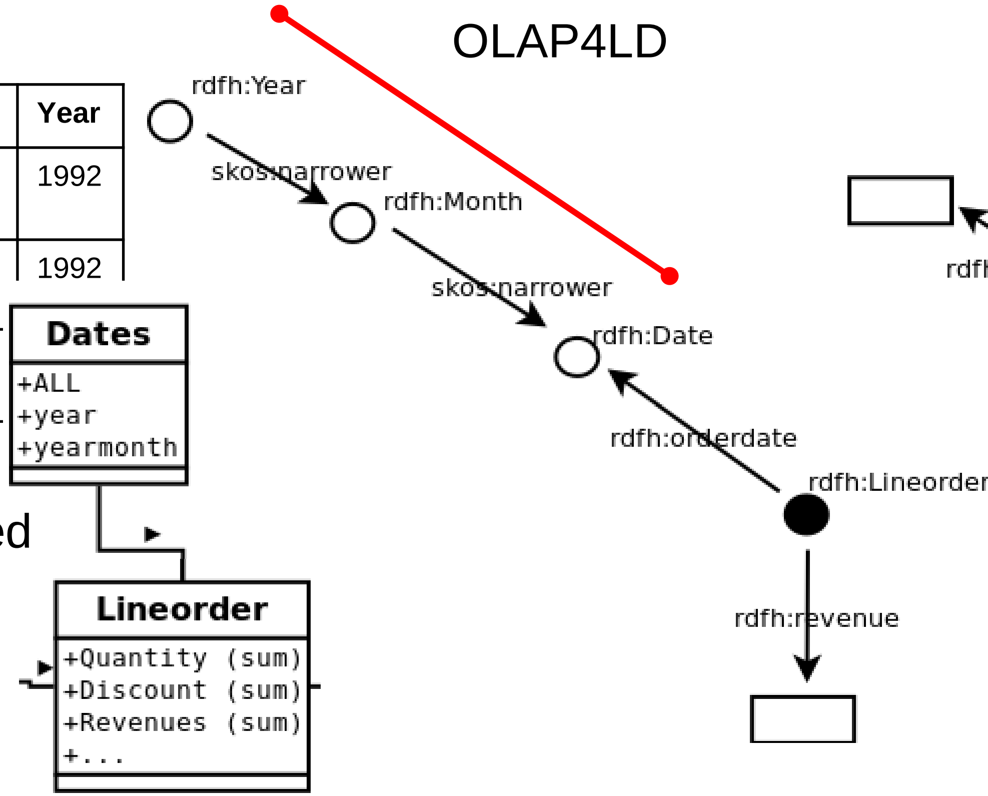
- Numerous joins **through skos:narrower paths** for [QB] hierarchies

ROLAP vs OLAP4LD – Query processing

OLAP4LD overall 12 times slower than ROLAP for executing all queries?

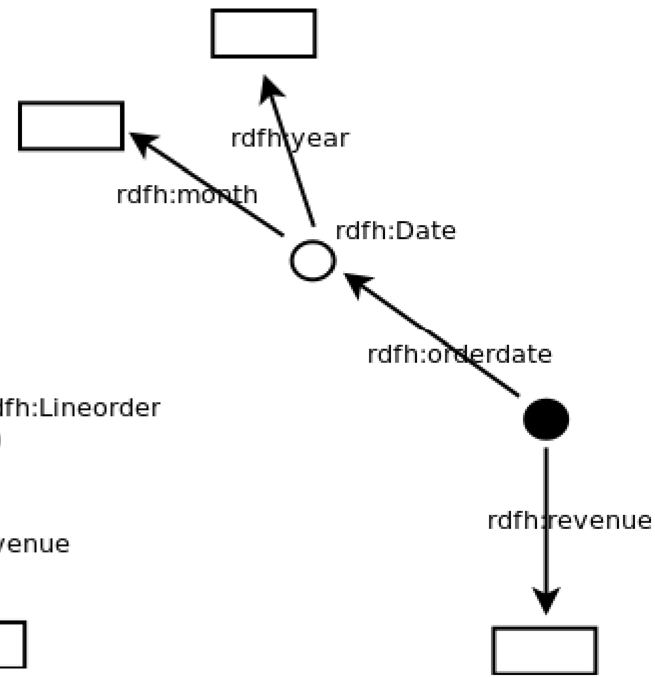
ROLAP

Dates	Month	Year
January 18, 1992	Jan, 1992	1992
February 8, 1992	Feb, 1992	1992
...	...	



Denormalised Dimension Table

OLAP4LD-SSB



- Numerous joins **through skos:narrower paths** for [QB] hierarchies
- Modeling more closely resembling Star Schema is faster (OLAP4LD-SSB)

Outline

1. Data Analytics on Statistics from the Web
2. Star Schema Benchmark (SSB)
3. Comparing OLAP on RDBMS and on Triple Store
4. Evaluating Materialised Aggregate Views
    1. **RDF Aggregate Views**
    2. Performance Gain of RDF Aggregate Views
5. Conclusions

Institute of Applied Informatics and Formal Description Methods (AIFB)

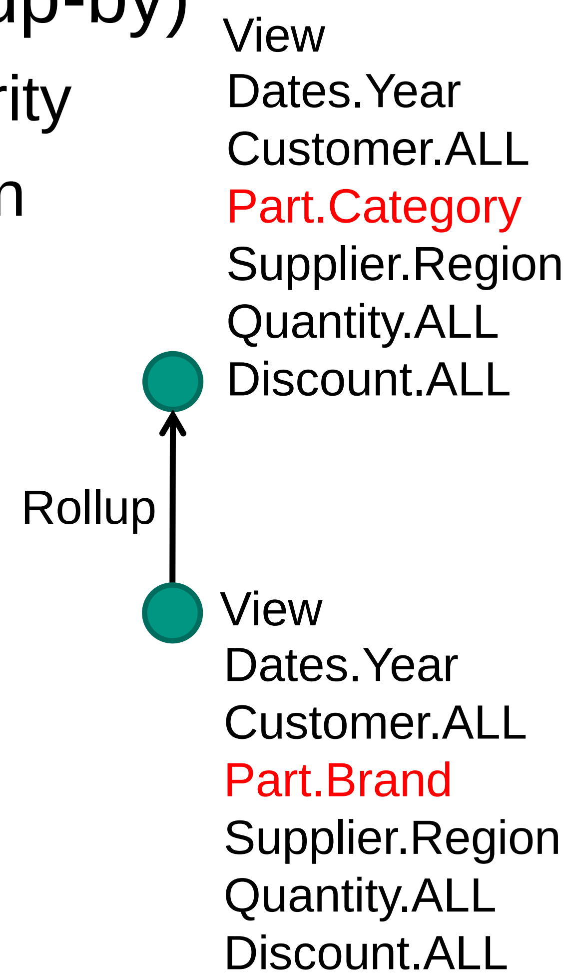
Evaluating Materialised Aggregate Views

- Aggregate View (aka cuboid, group-by)
  - Lineorders on same level of granularity

View
Dates.Year
Customer.ALL
Part.Brand
Supplier.Region
Quantity.ALL
Discount.ALL

Evaluating Materialised Aggregate Views

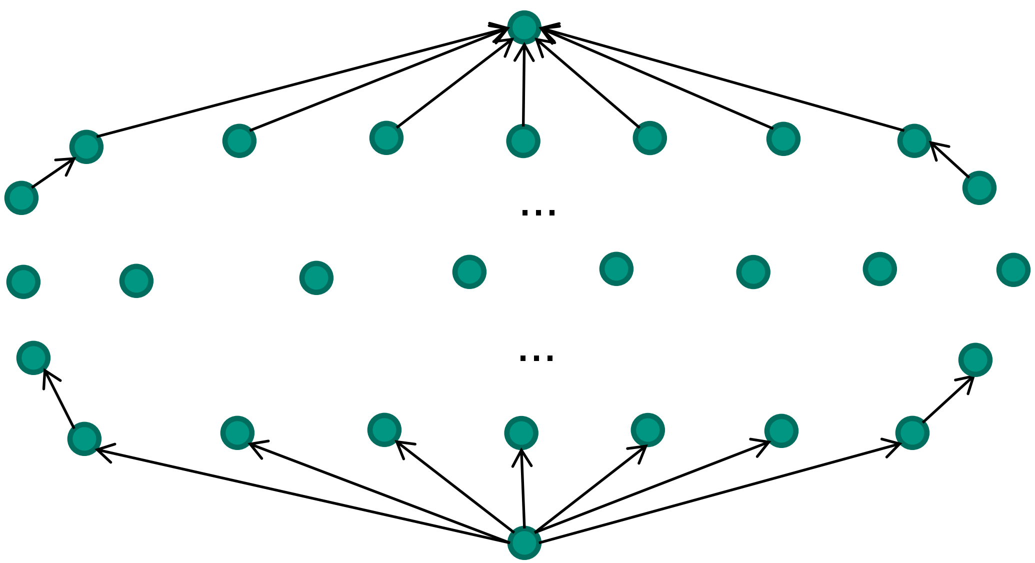
- Aggregate View (aka cuboid, group-by)
  - Lineorders on same level of granularity
  - Connected via Aggregation operation

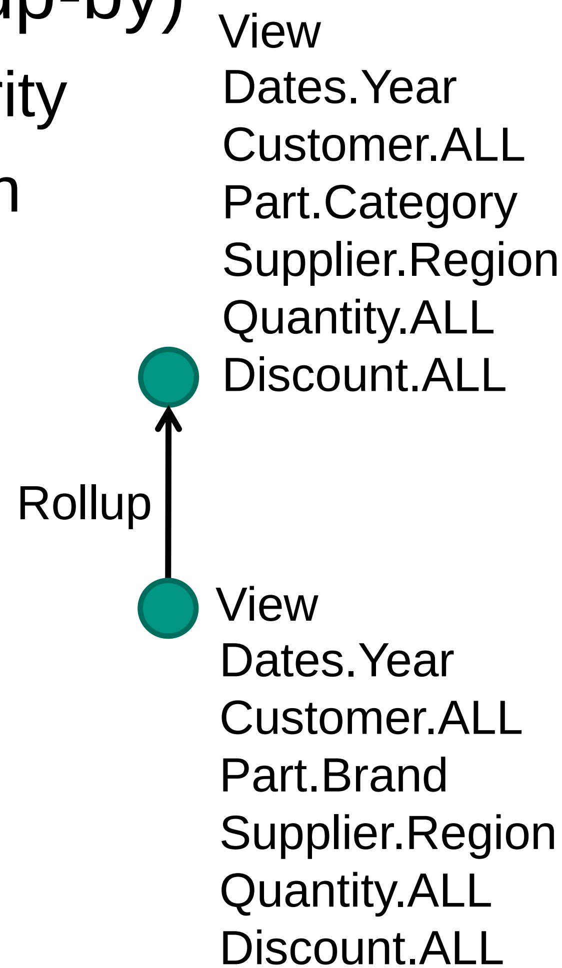


Institute of Applied Informatics and Formal Description Methods (AIFB)

Evaluating Materialised Aggregate Views

- Aggregate View (aka cuboid, group-by)
  - Lineorders on same level of granularity
  - Connected via Aggregation operation
  - Data Cube Lattice





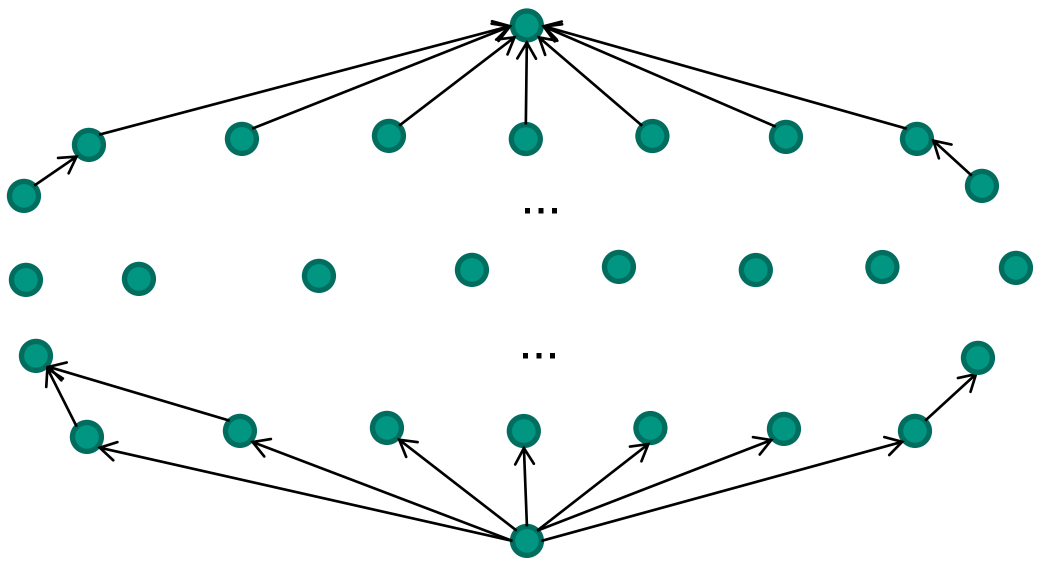
Data Cube Lattice – Features

- View computable from any view on lower level on Rollup path
- The higher the view in lattice the fewer lineorders
- Number of views:

$$\prod |L_i| \qquad (3{,}000)$$

- Number of possible lineorders in data cube:

$$\sum_{v \in View} \left(\prod_{l_i \in Level} |l_i(v)| \right) \qquad (2.6 * 10^{19})$$



▶ Problem: View Selection and Computation

View Selection and Computation

For each query, we **select** highest view from which query can be computed

- Cost of answering query **proportional to number of lineorders** that need to be scanned

Q2.1

*Revenues per **year** and per **product brands** of product category **MFGR#12** and of supplier from **America.***

Highest view to compute Q2.1

- View
  Dates.Year
  Customer.ALL
  Part.Brand
  Supplier.Region
  Quantity.ALL
  Discount.ALL

View Selection and Computation

For each query, we **select** highest view from which query can be computed

- Cost of answering query **proportional to number of lineorders** that need to be scanned

We **compute** views via SPARQL INSERT queries

- Views are **[QB] slices** of the lineorder data cube that fix dimensions to ALL value

Q2.1

*Revenues per **year** and per **product brands** of product category **MFGR#12** and of supplier from **America.***

Highest view to compute Q2.1

- View Dates.Year Customer.ALL Part.Brand Supplier.Region Quantity.ALL Discount.ALL

Slice of view Q2.1

- Slice Customer.ALL Quantity.ALL Discount.ALL

Outline

1. Data Analytics on Statistics from the Web
2. Star Schema Benchmark (SSB)
3. Comparing OLAP on RDBMS and on Triple Store
4. Evaluating Materialised Aggregate Views
    1. RDF Aggregate Views
    2. **Performance Gain of RDF Aggregate Views**
5. Conclusions

Problem: ROLAP vs OLAP4LD – Performance? Materialised Aggregate Views?

	No Materialisation	**Materialisation**
RDBMS/ SQL (MySQL)	ROLAP (Mondrian)	ROLAP-M (Mondrian)
Triple Store/ SPARQL (Open Virtuoso)	OLAP4LD (olap4ld) ⟷	OLAP4LD-M (olap4ld)

Institute of Applied Informatics and Formal Description Methods (AIFB)

OLAP4LD vs OLAP4LD-M – Results

- **Elapsed query time (eqt)**

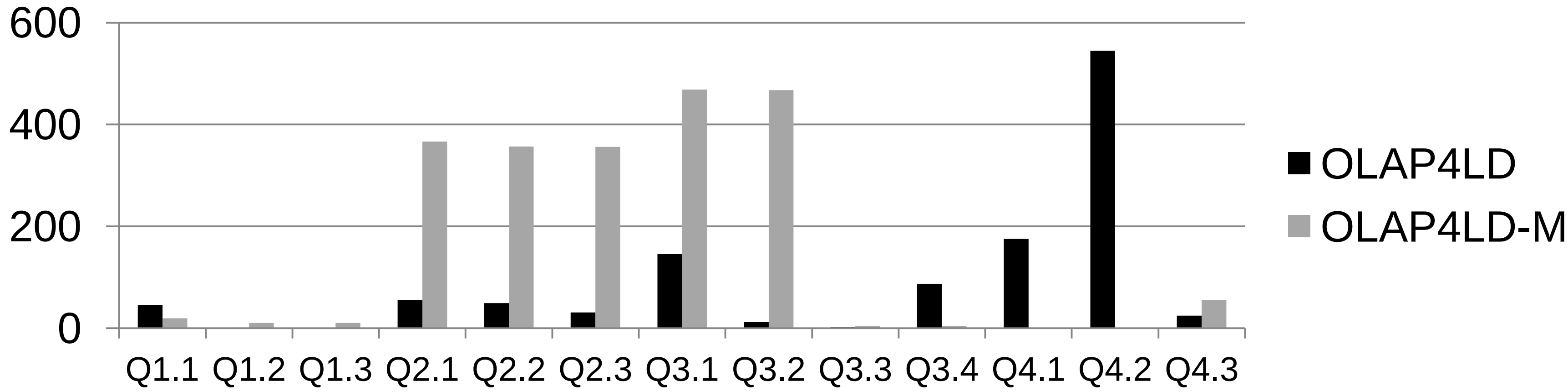
  OLAP4LD-M overall 2 times slower than OLAP4LD, but 13 times faster for query flight Q4

$$\frac{\sum eqt_{olap\ 4ld-m}(q_i)}{\sum eqt_{olap\ 4ld}(q_i)} \approx 2$$

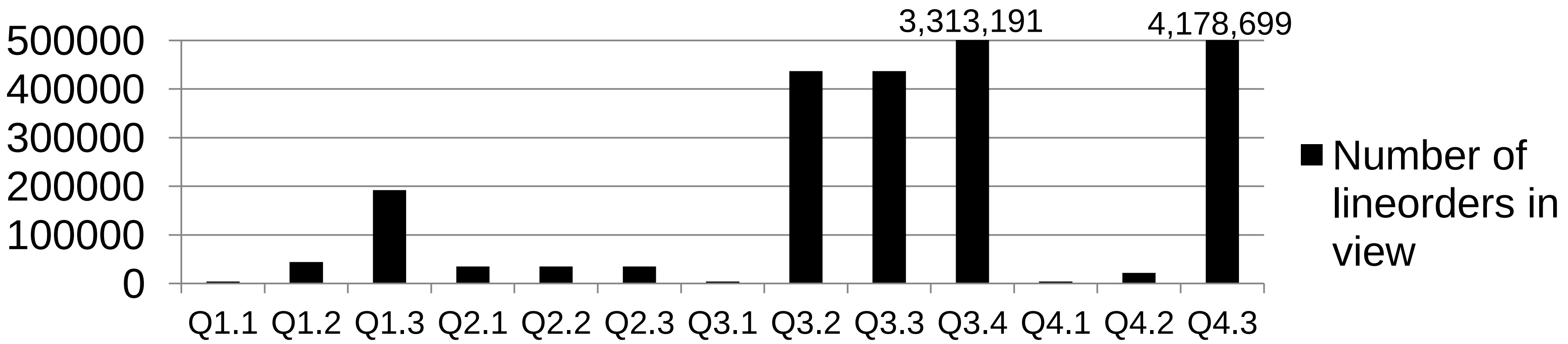
(ROLAP-M overall is 50 times faster than ROLAP)

OLAP4LD vs OLAP4LD-M – Query processing

OLAP4LD-M overall 2 times slower than OLAP4LD?

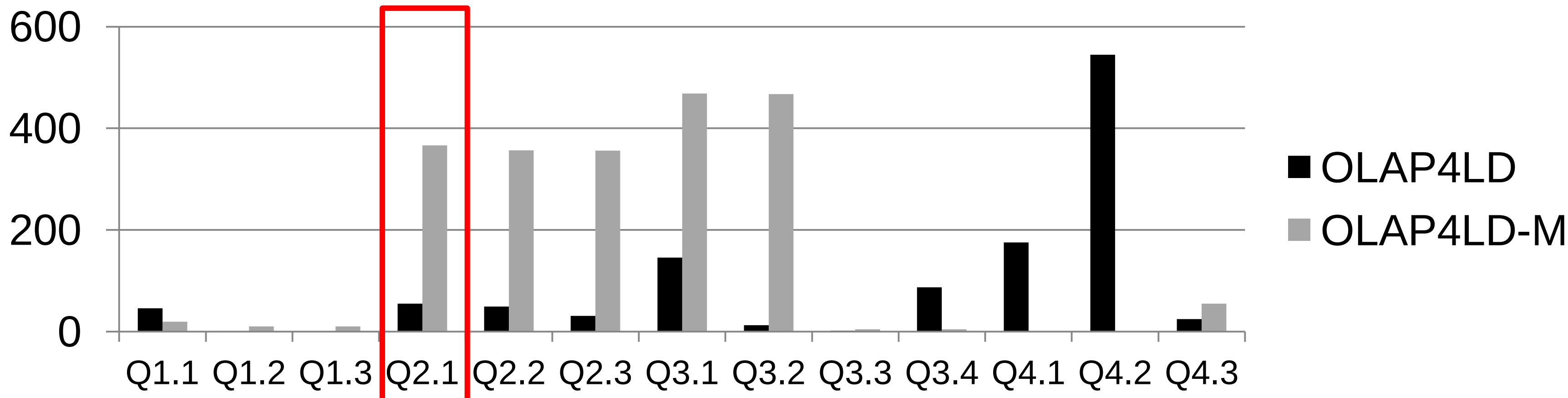


Hypothesis: OLAP4LD-M scans all lineorders although view contains fraction

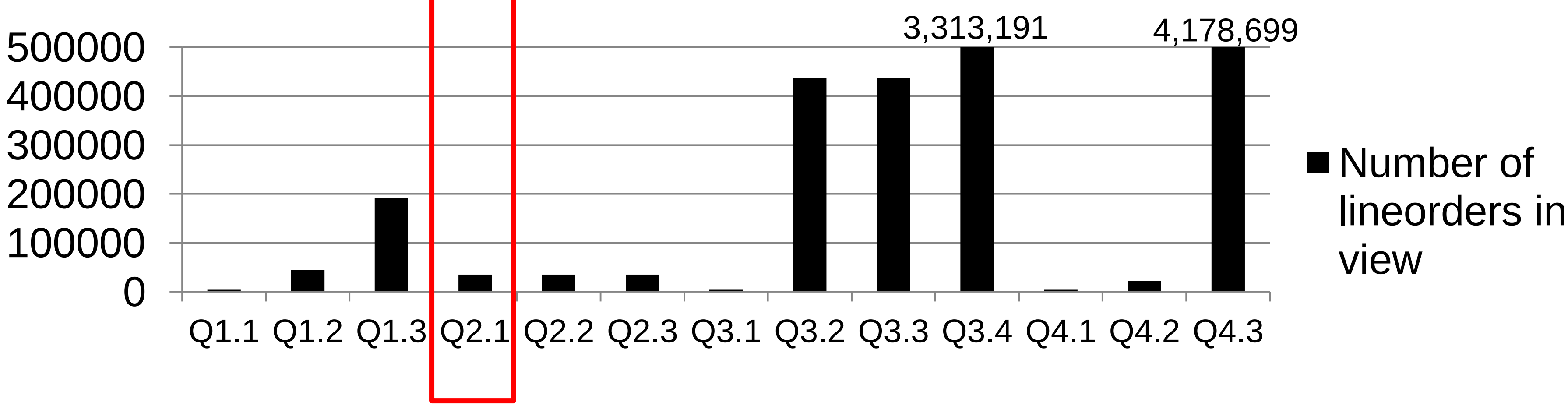


OLAP4LD vs OLAP4LD-M – Query processing

OLAP4LD-M overall 2 times slower than OLAP4LD?



Hypothesis: OLAP4LD-M scans all lineorders although view contains fraction



OLAP4LD vs OLAP4LD-M – Query processing

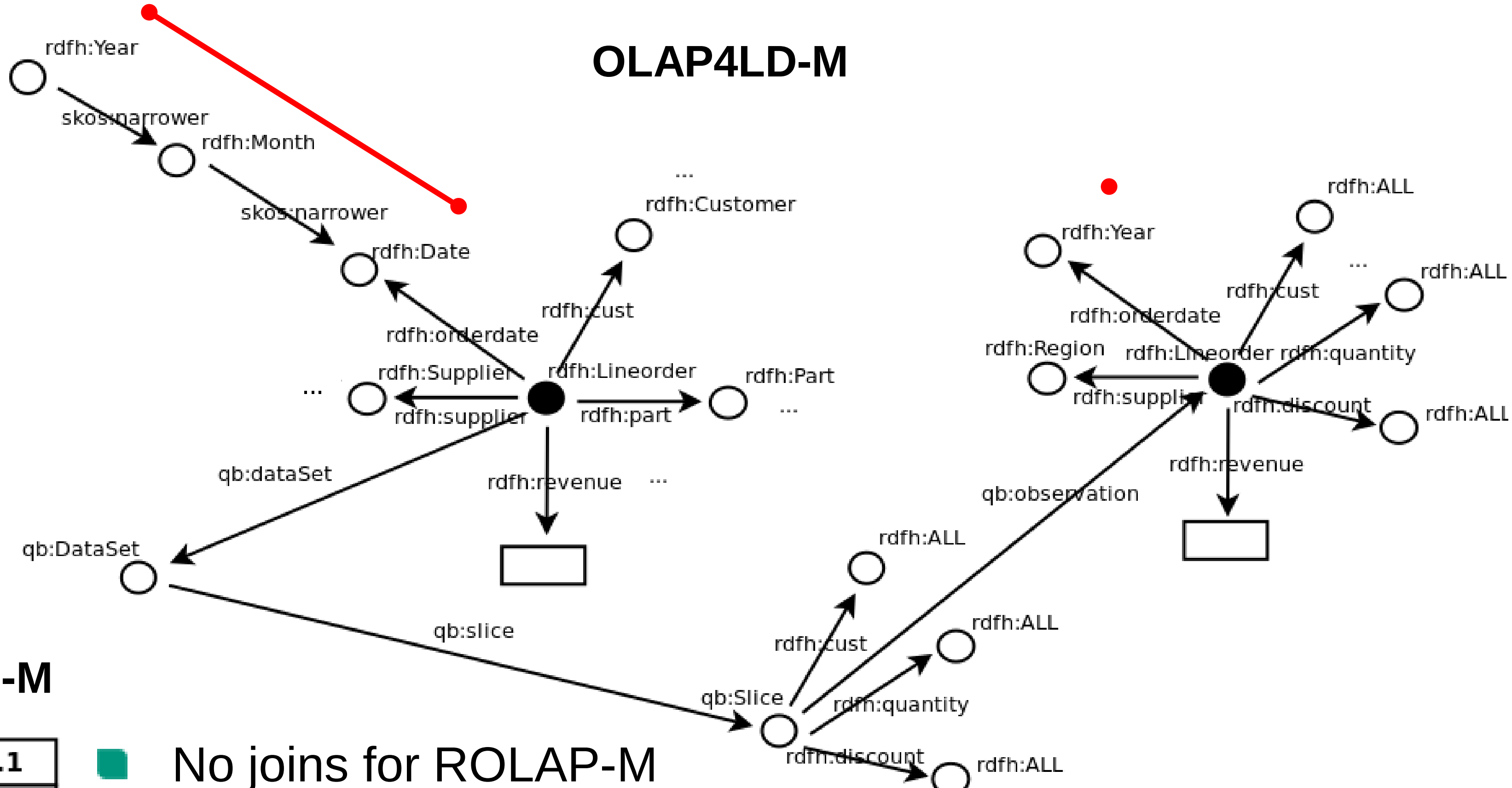
OLAP4LD-M overall 2 times slower than OLAP4LD, but 13 times faster for Q4?

ROLAP-M

- No joins needed for ROLAP-M

OLAP4LD vs OLAP4LD-M – Query processing

OLAP4LD-M overall 2 times slower than OLAP4LD, but 13 times faster for Q4?



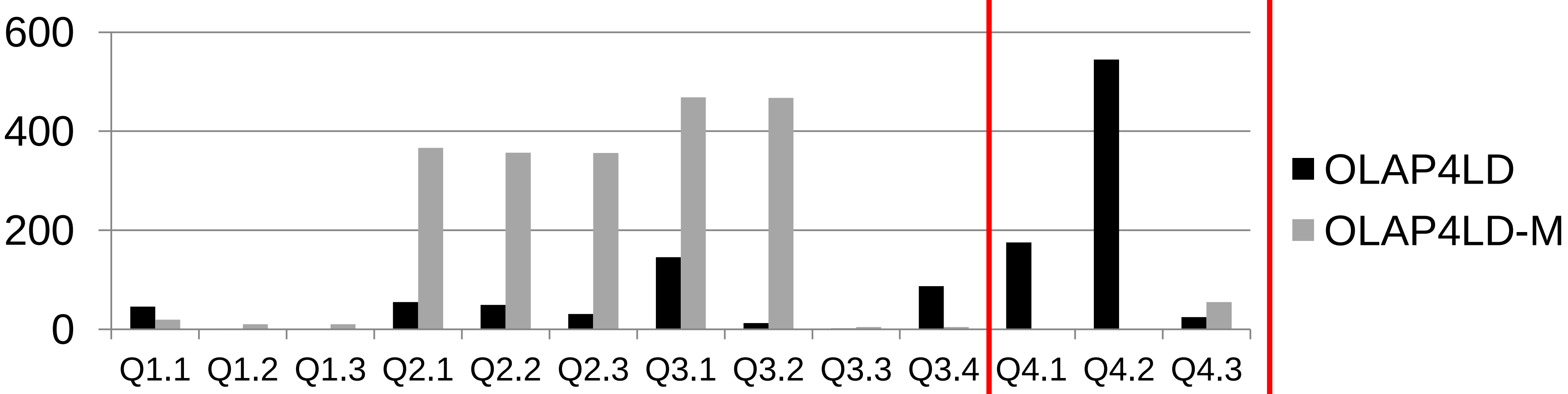
ROLAP-M

View_Q2.1
+Dates.Year
+Supplier.Region
+Part.Brand1
+Revenues (sum)

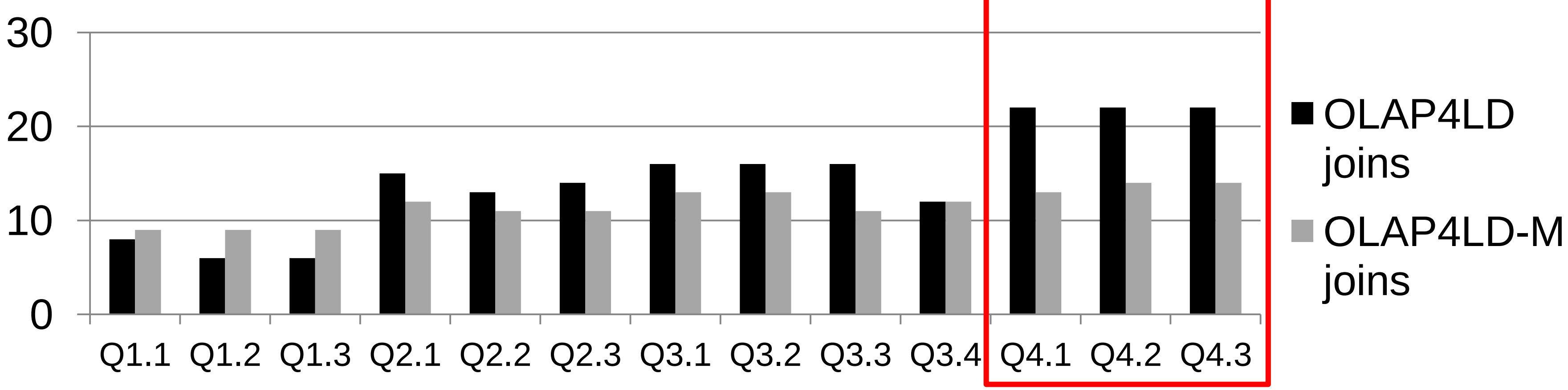
- No joins for ROLAP-M
- Reduced number of joins through skos:narrower paths for OLAP4LD-M hierarchies, but triples are stored in same graph.

OLAP4LD vs OLAP4LD-M – Query processing

OLAP4LD-M 13 times faster than OLAP4LD for Q4?



Hypothesis: Reduced number of joins for hierarchies



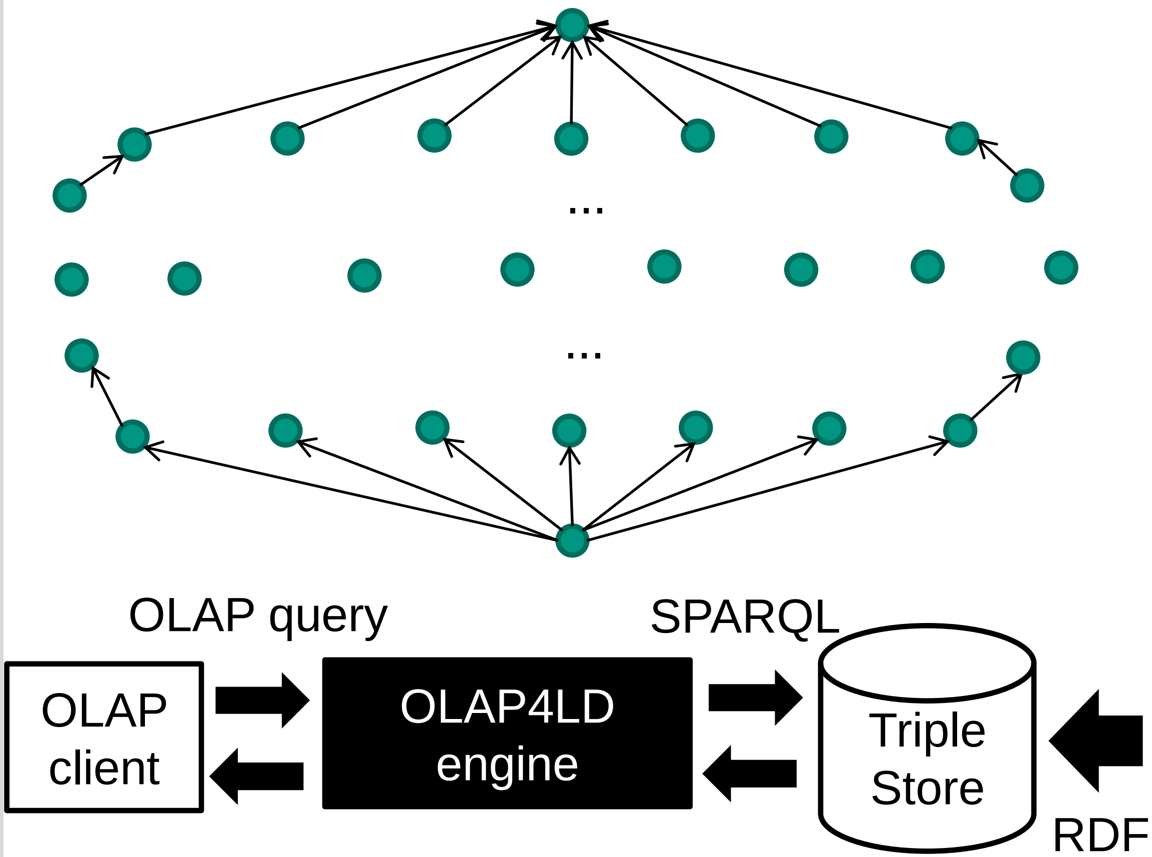
Conclusion: No Size Fits All – Efficiency versus Modeling Trade-off

Karlsruhe Institute of Technology

- **Efficiency:** Triple Stores per se not worse for analytical queries than RDBMS
  - If modelled **similar to Star Schema**, query processing just as fast as on RDBMS
- **Modeling:** If modeled using standard vocabulary [QB], statistics can more easily be integrated with other data sources, but
  - Require numerous **joins for hierarchies**
  - **RDF aggregate views** can be represented but increase number of triples in graph
  - Some queries **successfully optimised but** others much slower
- Analytical queries **on Statistical Linked Data** call for more optimised query engines

Feedback & Questions

- Thank you!



Metadata

RDF Data Cube Vocabulary [QB]

Company	Dates	Average Trading Price
Visa Inc.	July 1, 2008	**80 $**
Mastercard Inc.	July 1, 2008	**269 $**
...	...	**...**

Paper website: http://people.aifb.kit.edu/bka/ssb-benchmark/

References

[1] Spyros Kotoulas, Jacopo Urbani, Peter A. Boncz, Peter Mika: Robust Runtime Optimization and Skew-Resistant Execution of Analytical SPARQL Queries on Pig. ISWC. 2012.

[2] Lefteris Sidirourgos, Romulo Goncalves, Martin Kersten, Niels Nes, and Stefan Manegold. 2008. Column-store support for RDF data management: not all swans are white. Proc. VLDB Endow. 2008.

[SCOVO] Michael Hausenblas, Wolfgang Halb, Yves Raimond, Lee Feigenbaum, and Danny Ayers. SCOVO: Using Statistics on the Web of Data. ESWC 2009 Heraklion. 2009.

[QB4OLAP] Lorena Etcheverry, Alejandro A. Vaisman: QB4OLAP: A Vocabulary for OLAP Cubes on the Semantic Web. ISWC Workshop on Consuming Linked Data. 2012.

[QB] http://www.w3.org/TR/vocab-data-cube/

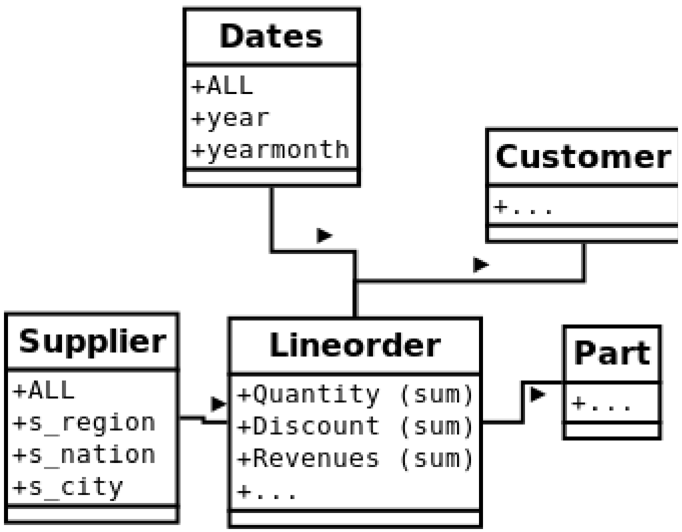
[3] Lorena Etcheverry, Alejandro Vaisman. Views over RDF Datasets: A Survey. Reasoning Web Summer School. 2011.

[SSB] O'Neil, P., O'Neil, E., Chen, X.: Star Schema Benchmark - Revision 3. Tech. rep., UMass/Boston (2009), http://www.cs.umb.edu/~poneil/StarSchemaB.pdf

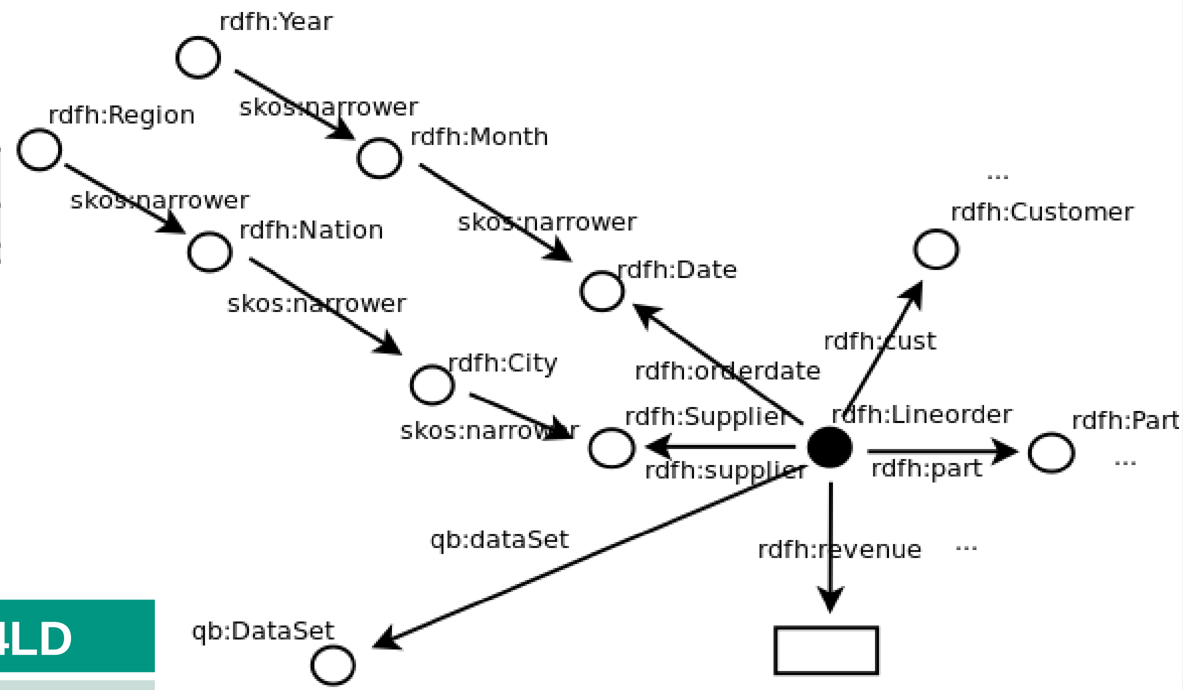
Backup: ROLAP vs OLAP4LD – Pre-processing

OLAP4LD takes 261 times longer than ROLAP for transforming and loading the data?

ROLAP

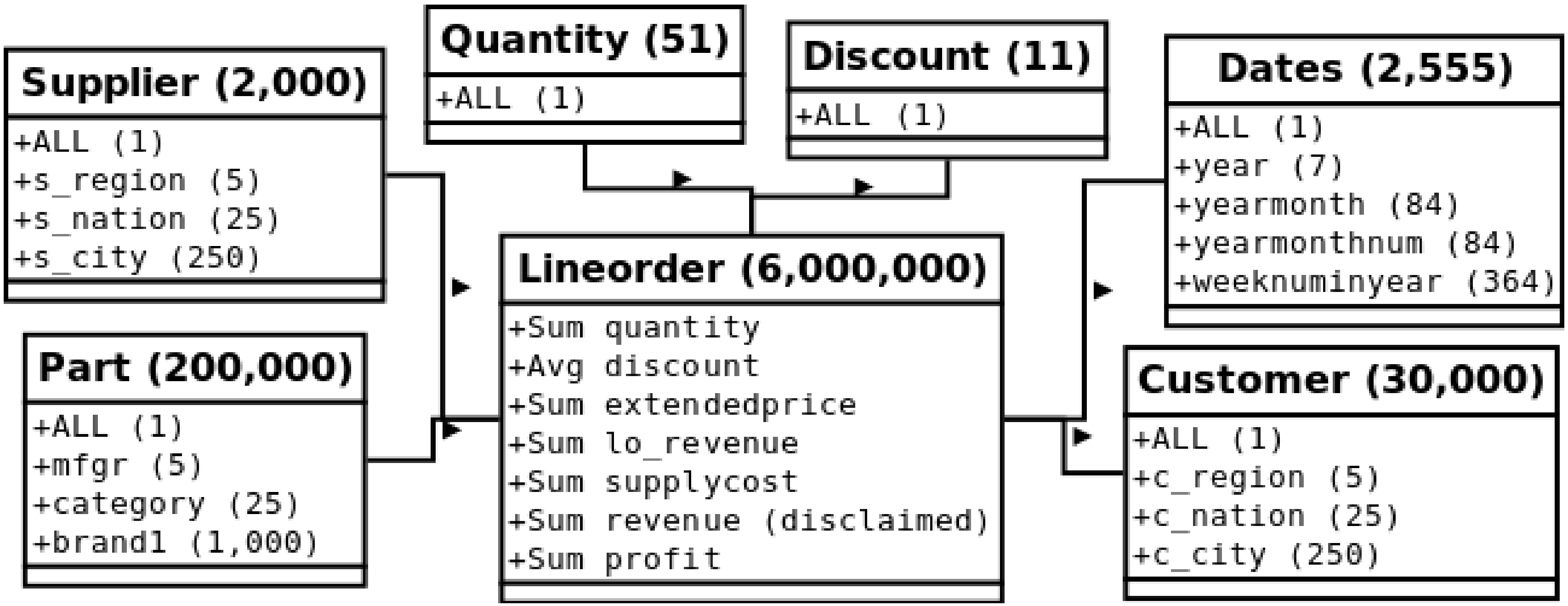


OLAP4LD



	ROLAP	OLAP4LD
Indices	No	RDF indexed
Hierarchies	Strings	Resources
Size of data	600MB	4.3GB

Backup: SSB Schema Detailed



Backup: SPARQL Query for Q2.1

```
SELECT ?rdfh_lo_orderdate ?rdfh_lo_partkey1 sum(?rdfh_lo_revenue) as ?lo_revenue
WHERE {
?obs qb:dataSet rdfh-inst:ds.
?obs rdfh:lo_orderdate  ?rdfh_lo_orderdate0.
?rdfh_lo_orderdate1 skos:narrower ?rdfh_lo_orderdate0.
?rdfh_lo_orderdate2 skos:narrower ?rdfh_lo_orderdate1.
?rdfh_lo_orderdate skos:narrower ?rdfh_lo_orderdate2.
rdfh:lo_orderdateYearLevel skos:member ?rdfh_lo_orderdate.
?obs rdfh:lo_partkey ?rdfh_lo_partkey0.
?rdfh_lo_partkey1 skos:narrower ?rdfh_lo_partkey0.
?rdfh_lo_partkey skos:narrower ?rdfh_lo_partkey1.
rdfh:lo_partkeyCategoryLevel skos:member ?rdfh_lo_partkey.
?obs rdfh:lo_suppkey ?rdfh_lo_suppkey0.
...
?obs rdfh:lo_revenue ?rdfh_lo_revenue.
FILTER(?rdfh_lo_partkey = rdfh:lo_partkeyCategoryMFGR-12 AND ?rdfh_lo_suppkey =
   rdfh:lo_suppkeyRegionAMERICA ).
} group by ?rdfh_lo_orderdate ?rdfh_lo_partkey1 order by ?rdfh_lo_orderdate
   ?rdfh_lo_partkey1
```

SPARQL Query for Q2.1 – Cube

```
SELECT ?rdfh_lo_orderdate ?rdfh_lo_partkey1 sum(?rdfh_lo_revenue) as ?lo_revenue
WHERE {
?obs qb:dataSet rdfh-inst:ds.
?obs rdfh:lo_orderdate  ?rdfh_lo_orderdate0.
?rdfh_lo_orderdate1 skos:narrower ?rdfh_lo_orderdate0.
?rdfh_lo_orderdate2 skos:narrower ?rdfh_lo_orderdate1.
?rdfh_lo_orderdate skos:narrower ?rdfh_lo_orderdate2.
rdfh:lo_orderdateYearLevel skos:member ?rdfh_lo_orderdate.
?obs rdfh:lo_partkey ?rdfh_lo_partkey0.
?rdfh_lo_partkey1 skos:narrower ?rdfh_lo_partkey0.
?rdfh_lo_partkey skos:narrower ?rdfh_lo_partkey1.
rdfh:lo_partkeyCategoryLevel skos:member ?rdfh_lo_partkey.
?obs rdfh:lo_suppkey ?rdfh_lo_suppkey0.
...
?obs rdfh:lo_revenue ?rdfh_lo_revenue.
FILTER(?rdfh_lo_partkey = rdfh:lo_partkeyCategoryMFGR-12 AND ?rdfh_lo_suppkey =
   rdfh:lo_suppkeyRegionAMERICA ).
} group by ?rdfh_lo_orderdate ?rdfh_lo_partkey1 order by ?rdfh_lo_orderdate
   ?rdfh_lo_partkey1
```

Lineorder Data Cube

SPARQL Query for Q2.1 – Slice/Roll-Up

```
SELECT ?rdfh_lo_orderdate ?rdfh_lo_partkey1 sum(?rdfh_lo_revenue) as ?lo_revenue
WHERE {
?obs qb:dataSet rdfh-inst:ds.
?obs rdfh:lo_orderdate  ?rdfh_lo_orderdate0.
?rdfh_lo_orderdate1 skos:narrower ?rdfh_lo_orderdate0.
?rdfh_lo_orderdate2 skos:narrower ?rdfh_lo_orderdate1.
?rdfh_lo_orderdate skos:narrower ?rdfh_lo_orderdate2.
rdfh:lo_orderdateYearLevel skos:member ?rdfh_lo_orderdate.
?obs rdfh:lo_partkey ?rdfh_lo_partkey0.
?rdfh_lo_partkey1 skos:narrower ?rdfh_lo_partkey0.
?rdfh_lo_partkey skos:narrower ?rdfh_lo_partkey1.
rdfh:lo_partkeyCategoryLevel skos:member ?rdfh_lo_partkey.
?obs rdfh:lo_suppkey ?rdfh_lo_suppkey0.
...
?obs rdfh:lo_revenue ?rdfh_lo_revenue.
FILTER(?rdfh_lo_partkey = rdfh:lo_partkeyCategoryMFGR-12 AND ?rdfh_lo_suppkey =
   rdfh:lo_suppkeyRegionAMERICA ).
} group by ?rdfh_lo_orderdate ?rdfh_lo_partkey1 order by ?rdfh_lo_orderdate
   ?rdfh_lo_partkey1
```

Roll-Up DatesYear

Roll-Up Part Brand1

Roll-Up Supplier Region

Slice Customer, Supplier, Quantity, Discount

SPARQL Query for Q2.1 – Projection

```
SELECT ?rdfh_lo_orderdate ?rdfh_lo_partkey1 sum(?rdfh_lo_revenue) as ?lo_revenue
WHERE {
?obs qb:dataSet rdfh-inst:ds.
?obs rdfh:lo_orderdate  ?rdfh_lo_orderdate0.
?rdfh_lo_orderdate1 skos:narrower ?rdfh_lo_orderdate0.
?rdfh_lo_orderdate2 skos:narrower ?rdfh_lo_orderdate1.
?rdfh_lo_orderdate skos:narrower ?rdfh_lo_orderdate2.
rdfh:lo_orderdateYearLevel skos:member ?rdfh_lo_orderdate.
?obs rdfh:lo_partkey ?rdfh_lo_partkey0.
?rdfh_lo_partkey1 skos:narrower ?rdfh_lo_partkey0.
?rdfh_lo_partkey skos:narrower ?rdfh_lo_partkey1.
rdfh:lo_partkeyCategoryLevel skos:member ?rdfh_lo_partkey.
?obs rdfh:lo_suppkey ?rdfh_lo_suppkey0.
...
?obs rdfh:lo_revenue ?rdfh_lo_revenue.
FILTER(?rdfh_lo_partkey = rdfh:lo_partkeyCategoryMFGR-12 AND ?rdfh_lo_suppkey =
   rdfh:lo_suppkeyRegionAMERICA ).
} group by ?rdfh_lo_orderdate ?rdfh_lo_partkey1 order by ?rdfh_lo_orderdate
   ?rdfh_lo_partkey1
```

Revenues (sum)

SPARQL Query for Q2.1 – Dice

```
SELECT ?rdfh_lo_orderdate ?rdfh_lo_partkey1 sum(?rdfh_lo_revenue) as ?lo_revenue
WHERE {
?obs qb:dataSet rdfh-inst:ds.
?obs rdfh:lo_orderdate  ?rdfh_lo_orderdate0.
?rdfh_lo_orderdate1 skos:narrower ?rdfh_lo_orderdate0.
?rdfh_lo_orderdate2 skos:narrower ?rdfh_lo_orderdate1.
?rdfh_lo_orderdate skos:narrower ?rdfh_lo_orderdate2.
rdfh:lo_orderdateYearLevel skos:member ?rdfh_lo_orderdate.
?obs rdfh:lo_partkey ?rdfh_lo_partkey0.
?rdfh_lo_partkey1 skos:narrower ?rdfh_lo_partkey0.
?rdfh_lo_partkey skos:narrower ?rdfh_lo_partkey1.
rdfh:lo_partkeyCategoryLevel skos:member ?rdfh_lo_partkey.
?obs rdfh:lo_suppkey ?rdfh_lo_suppkey0.
...
?obs rdfh:lo_revenue ?rdfh_lo_revenue.
FILTER(?rdfh_lo_partkey = rdfh:lo_partkeyCategoryMFGR-12 AND ?rdfh_lo_suppkey =
   rdfh:lo_suppkeyRegionAMERICA ).
} group by ?rdfh_lo_orderdate ?rdfh_lo_partkey1 order by ?rdfh_lo_orderdate
   ?rdfh_lo_partkey1
```

Supplier Region = AMERICA AND
Part Category = MFGR#12