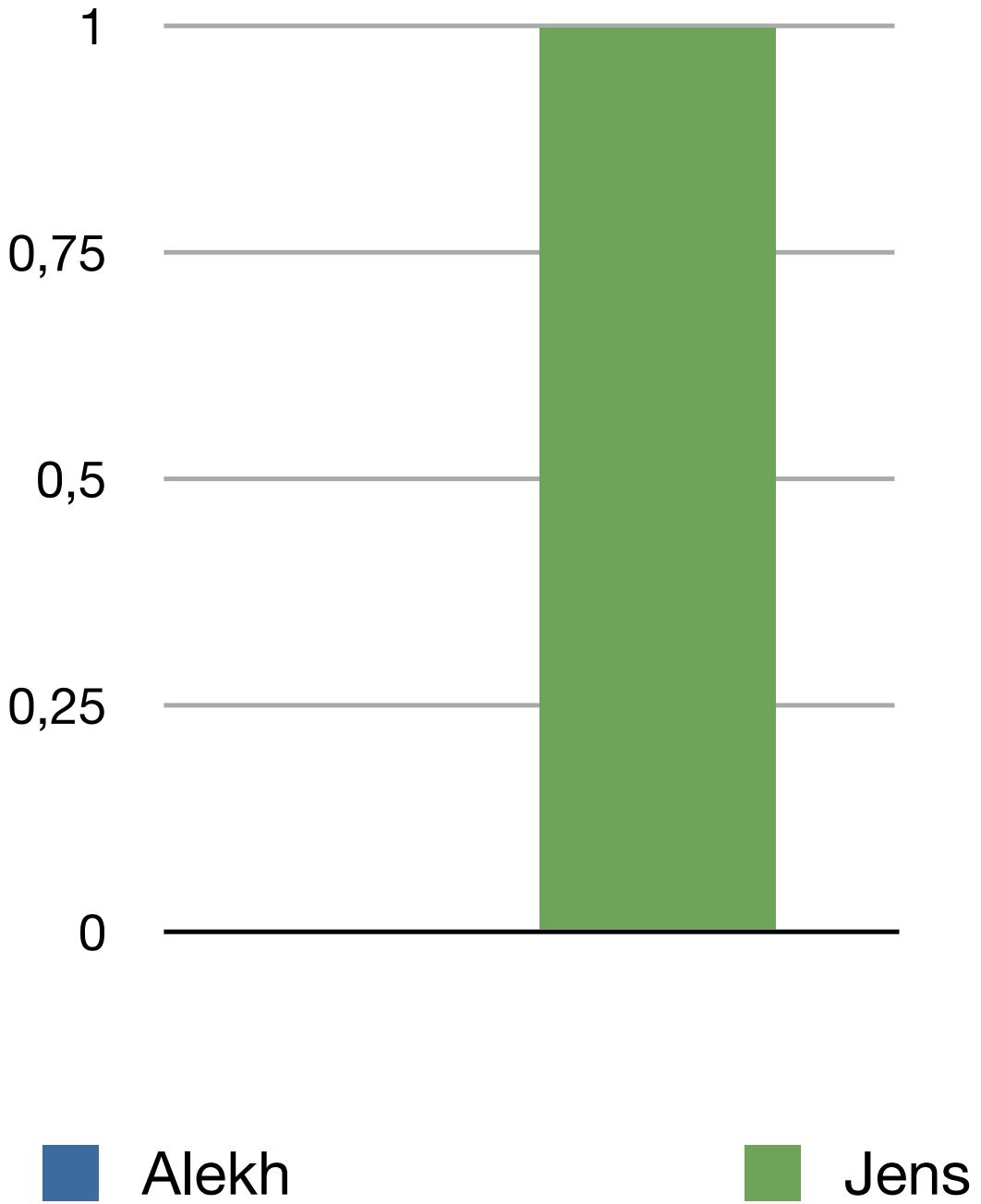
How Achaeans Would Construct Columns in Troy

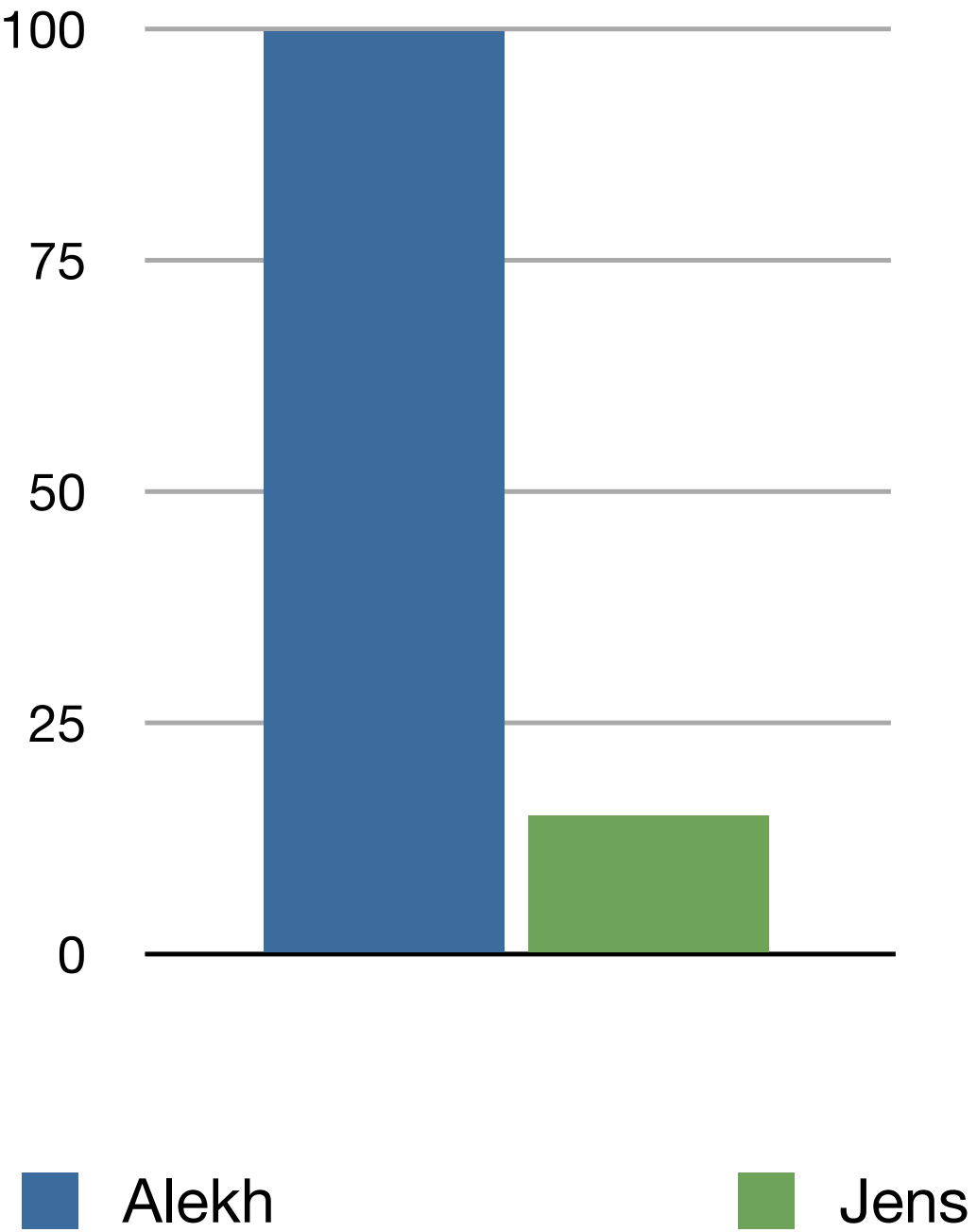
Alekh Jindal, Felix Martin Schuhknecht,
Jens Dittrich, Karen Khachatryan,
Alexander Bunte

UNIVERSITÄT
DES
SAARLANDES

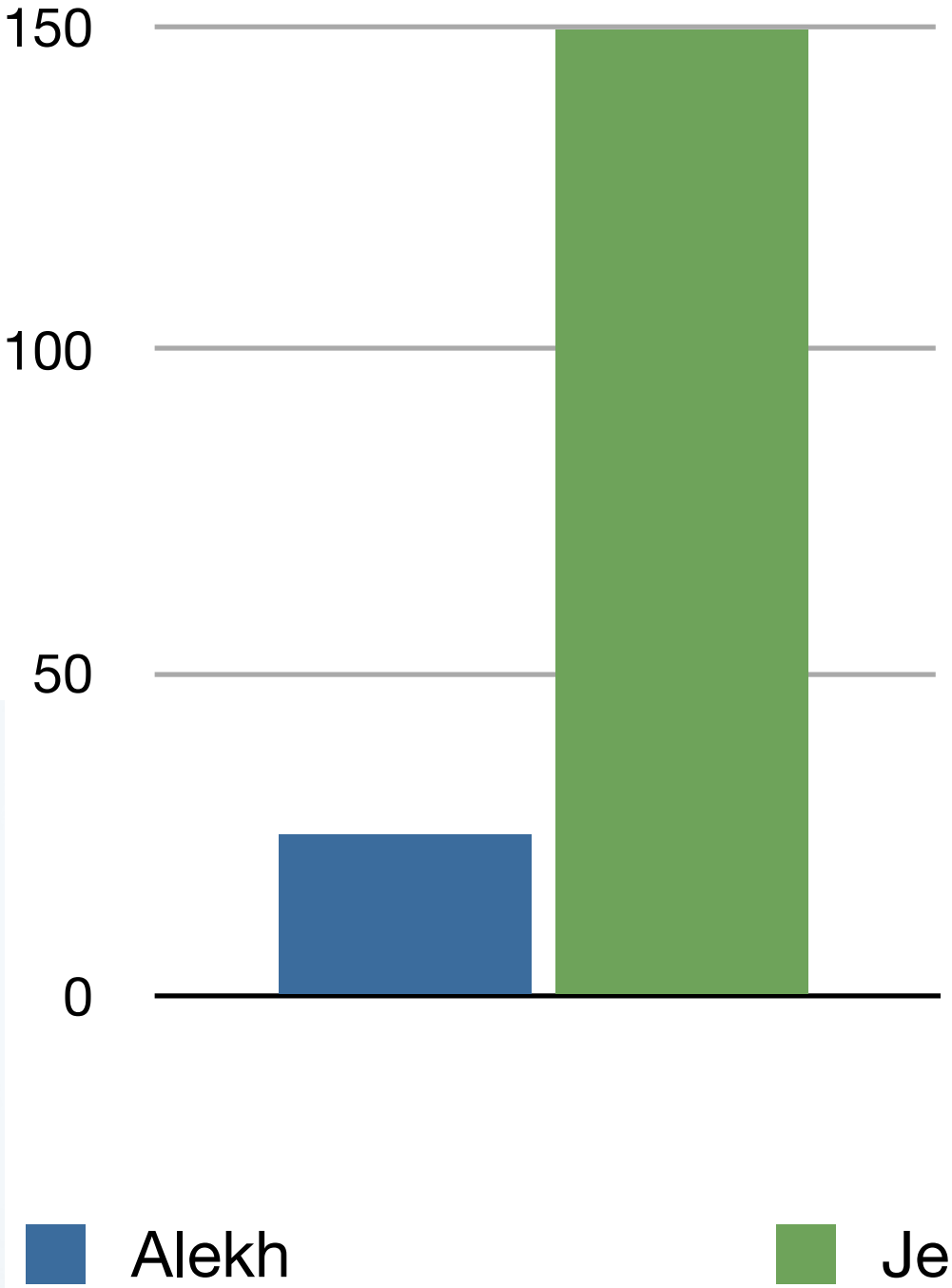
Number of Visas Received



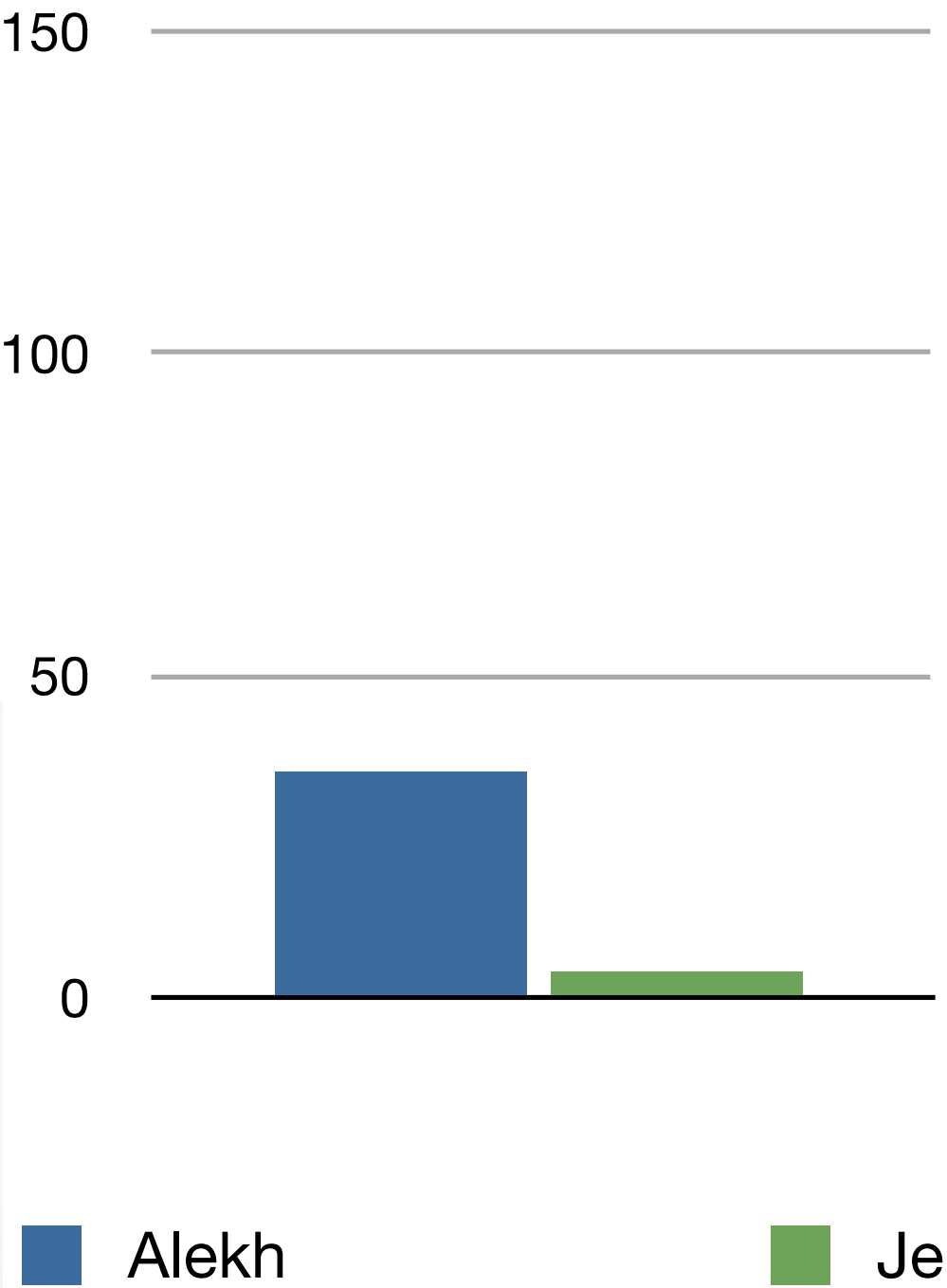
Health Level 5 days before CIDR



Average Number of Slides per 20min talk



Number of Slides Actually Prepared



“

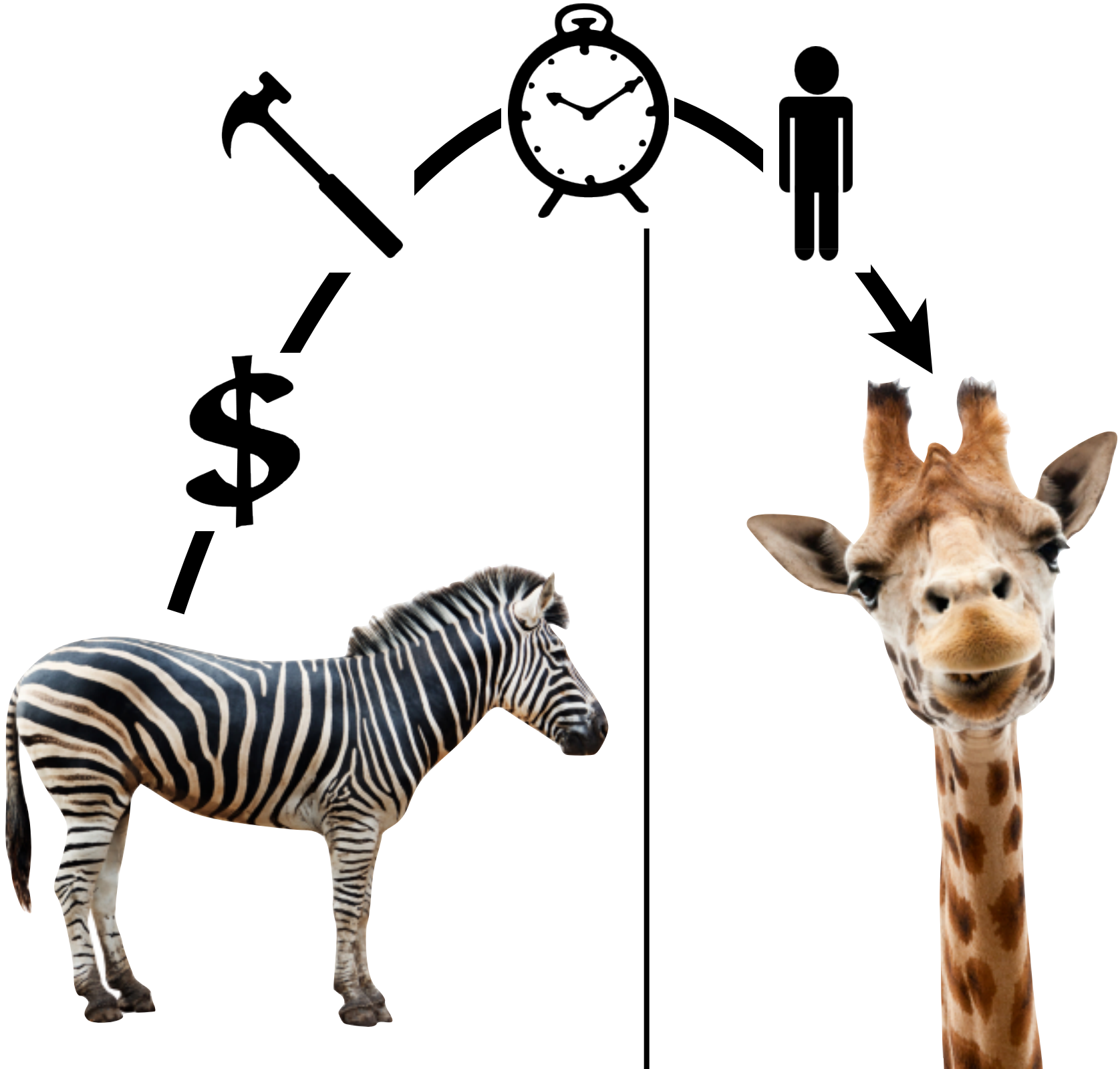
What is the problem?

Row-stores

Column-stores

OLTP

OLAP

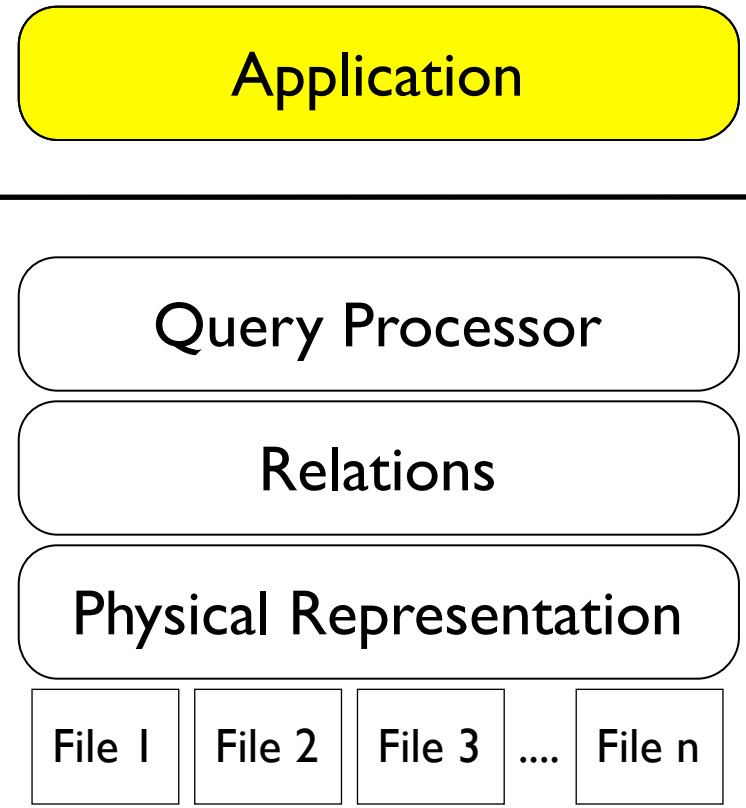


OLTP OLAP?

Can we do efficient
OLAP in Row-stores?

Any solutions out there?

C-Tables *



* Nicolas Bruno. Teaching an Old Elephant New Tricks. CIDR 2009

C-Tables *

Relation

Customer		
name	phone	market_segment
smith	2134	automobile
john	3425	household
kim	6756	furniture
joe	9878	building
mark	4312	building
steve	2435	automobile
jim	5766	household
ian	8789	household

* Nicolas Bruno. Teaching an Old Elephant New Tricks. CIDR 2009

C-Tables *

Sorted Relation

Customer		
name	phone	market_segment
smith	2134	automobile
steve	2435	automobile
mark	4312	building
joe	9878	building
kim	6756	furniture
john	3425	household
jim	5766	household
ian	8789	household

Physical Table

T_market_segment		
f	v	c
1	automobile	2
3	building	2
5	furniture	1
6	household	3

* Nicolas Bruno. Teaching an Old Elephant New Tricks. CIDR 2009

C-Tables *

Sorted Relation

Customer		
name	phone	market_segment
smith	2134	automobile
steve	2435	automobile
mark	4312	building
joe	9878	building
kim	6756	furniture
john	3425	household
jim	5766	household
ian	8789	household

Physical Table

T_market_segment		
f	v	c
1	automobile	2
3	building	2
5	furniture	1
6	household	3

* Nicolas Bruno. Teaching an Old Elephant New Tricks. CIDR 2009

C-Tables *

Sorted Relation

Customer		
name	phone	market_segment
smith	2134	automobile
steve	2435	automobile
mark	4312	building
joe	9878	building
kim	6756	furniture
john	3425	household
jim	5766	household
ian	8789	household

Physical Table

T_market_segment		
f	v	c
1	automobile	2
3	building	2
5	furniture	1
6	household	3

* Nicolas Bruno. Teaching an Old Elephant New Tricks. CIDR 2009

C-Tables *

Sorted Relation

Customer		
name	phone	market_segment
smith	2134	automobile
steve	2435	automobile
mark	4312	building
joe	9878	building
kim	6756	furniture
john	3425	household
jim	5766	household
ian	8789	household

Physical Table

T_market_segment		
f	v	c
1	automobile	2
3	building	2
5	furniture	1
6	household	3

* Nicolas Bruno. Teaching an Old Elephant New Tricks. CIDR 2009

C-Tables *

Sorted Relation

Customer		
name	phone	market_segment
smith	2134	automobile
steve	2435	automobile
mark	4312	building
joe	9878	building
kim	6756	furniture
john	3425	household
jim	5766	household
ian	8789	household

Physical Table

T_market_segment		
f	v	c
1	automobile	2
3	building	2
5	furniture	1
6	household	3

⋈

T_phone	
f	v
1	2134
2	2435
3	4312
4	9878
5	6756
6	3425
7	5766
8	8789

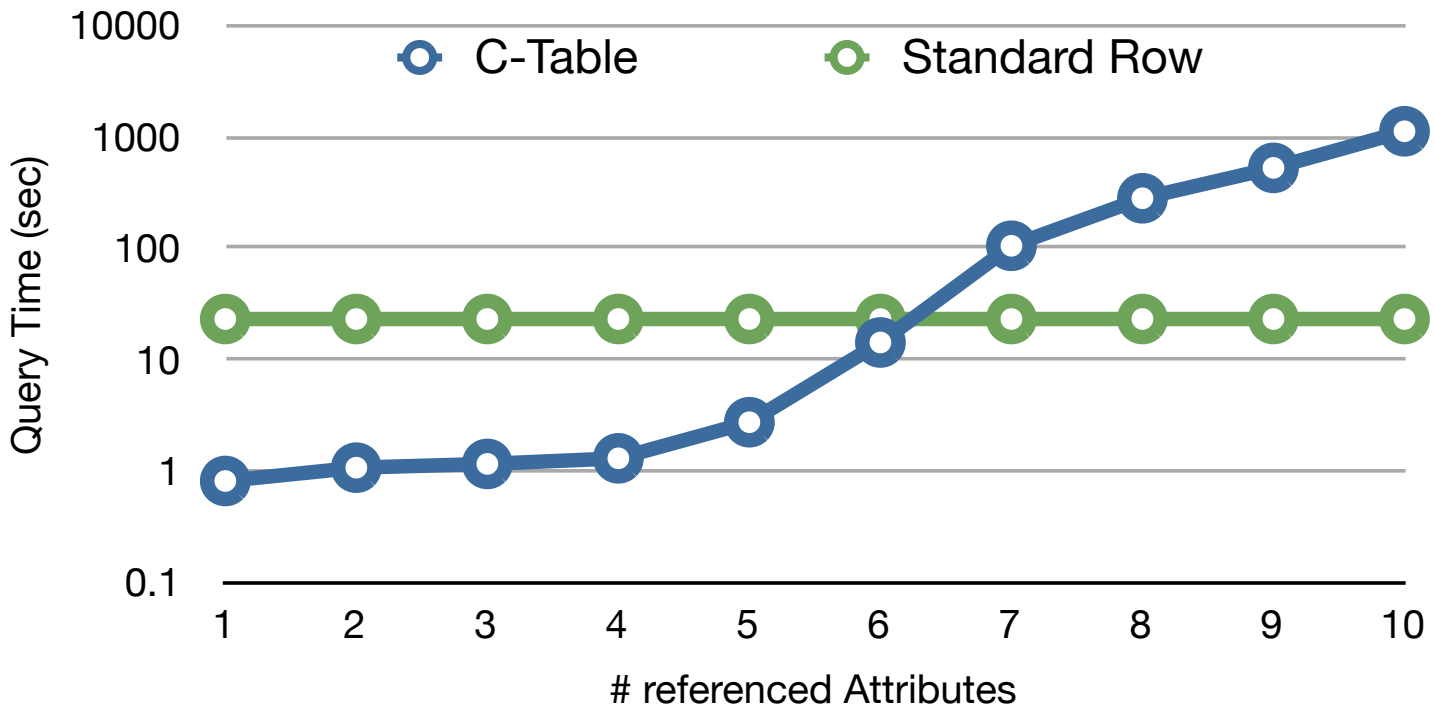
⋈

T_name	
f	v
1	smith
2	steve
3	mark
4	joe
5	kim
6	john
7	jim
8	ian

JOINS !

* Nicolas Bruno. Teaching an Old Elephant New Tricks. CIDR 2009

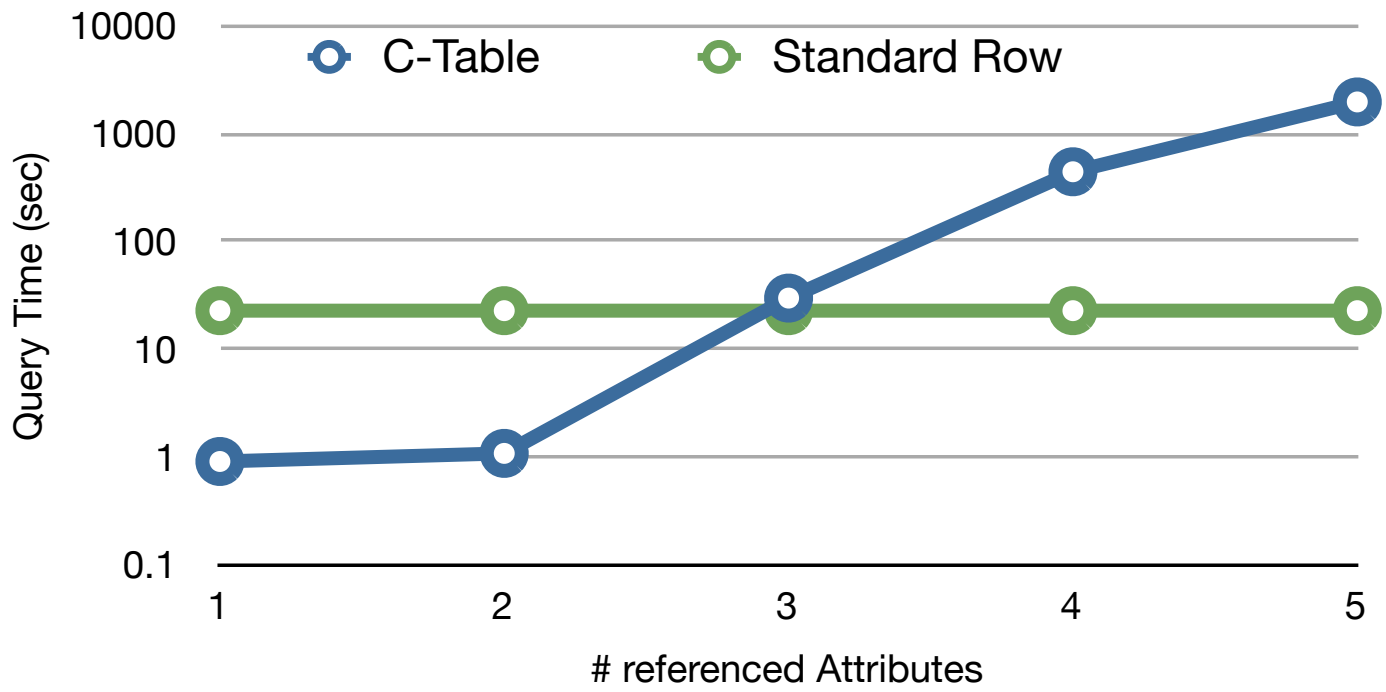
C-Tables *



(a) Cardinality = 10

* Nicolas Bruno. Teaching an Old Elephant New Tricks. CIDR 2009

C-Tables *



(b) Cardinality = 100

* Nicolas Bruno. Teaching an Old Elephant New Tricks. CIDR 2009

C-Tables *

Application

User

Database

Query Processor

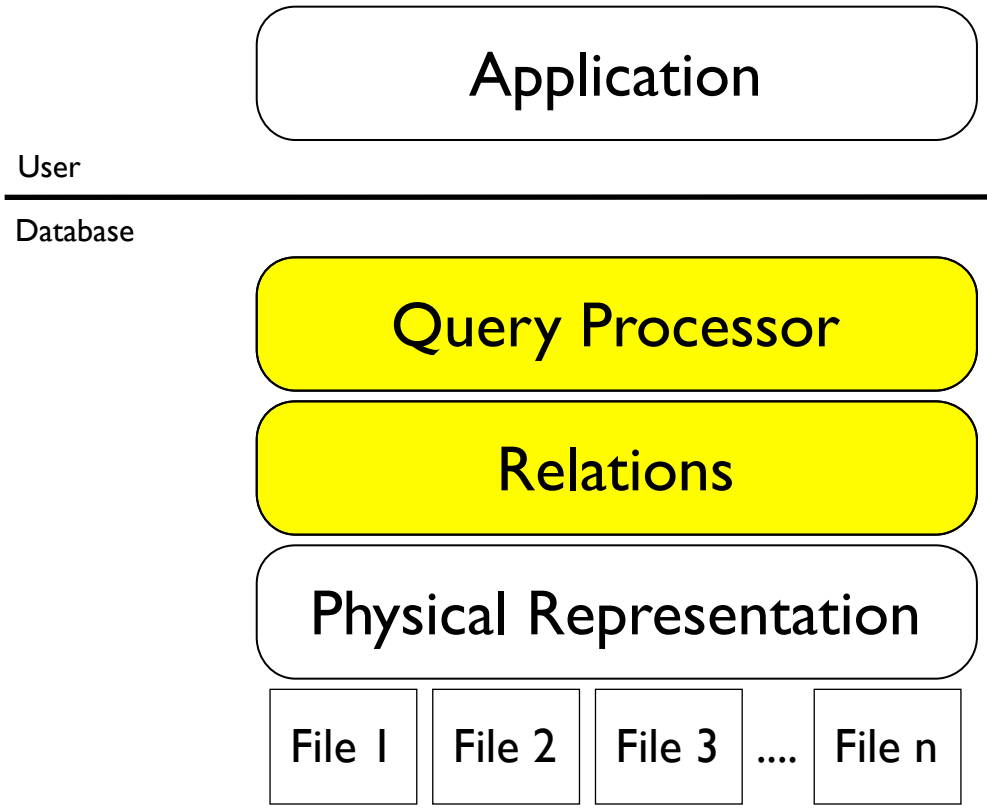
Relations

Physical Representation

File 1 | File 2 | File 3 | .... | File n

* Nicolas Bruno. Teaching an Old Elephant New Tricks. CIDR 2009

Column Index*



* P. Larson et. al. SQL Server Column Store Indexes. SIGMOD 2011

Column Index *

Relation

Customer		
name	phone	market_segment
smith	2134	automobile
john	3425	household
kim	6756	furniture
joe	9878	building
mark	4312	building
steve	2435	automobile
jim	5766	household
ian	8789	household

Physical Table

Customer_trojan		
segment_ID	attribute_ID	blob_data
1	name	smith, john, kim, joe
1	phone	2134, 3425, 6756, 9878
1	market_segment	automobile, household, furniture, building
2	name	mark, steve, jim, ian
2	phone	4312, 2435, 5766, 8789
2	market_segment	building, automobile, household, household

segment size = 4

* P. Larson et. al. SQL Server Column Store Indexes. SIGMOD 2011

Column Index *

Relation

Customer		
name	phone	market_segment
smith	2134	automobile
john	3425	household
kim	6756	furniture
joe	9878	building
mark	4312	building
steve	2435	automobile
jim	5766	household
ian	8789	household

Physical Table

Customer_trojan		
segment_ID	attribute_ID	blob_data
1	name	smith, john, kim, joe
1	phone	2134, 3425, 6756, 9878
1	market_segment	automobile, household, furniture, building
2	name	mark, steve, jim, ian
2	phone	4312, 2435, 5766, 8789
2	market_segment	building, automobile, household, household

segment size = 4

* P. Larson et. al. SQL Server Column Store Indexes. SIGMOD 2011

Column Index *

Relation

Customer		
name	phone	market_segment
smith	2134	automobile
john	3425	household
kim	6756	furniture
joe	9878	building
mark	4312	building
steve	2435	automobile
jim	5766	household
ian	8789	household

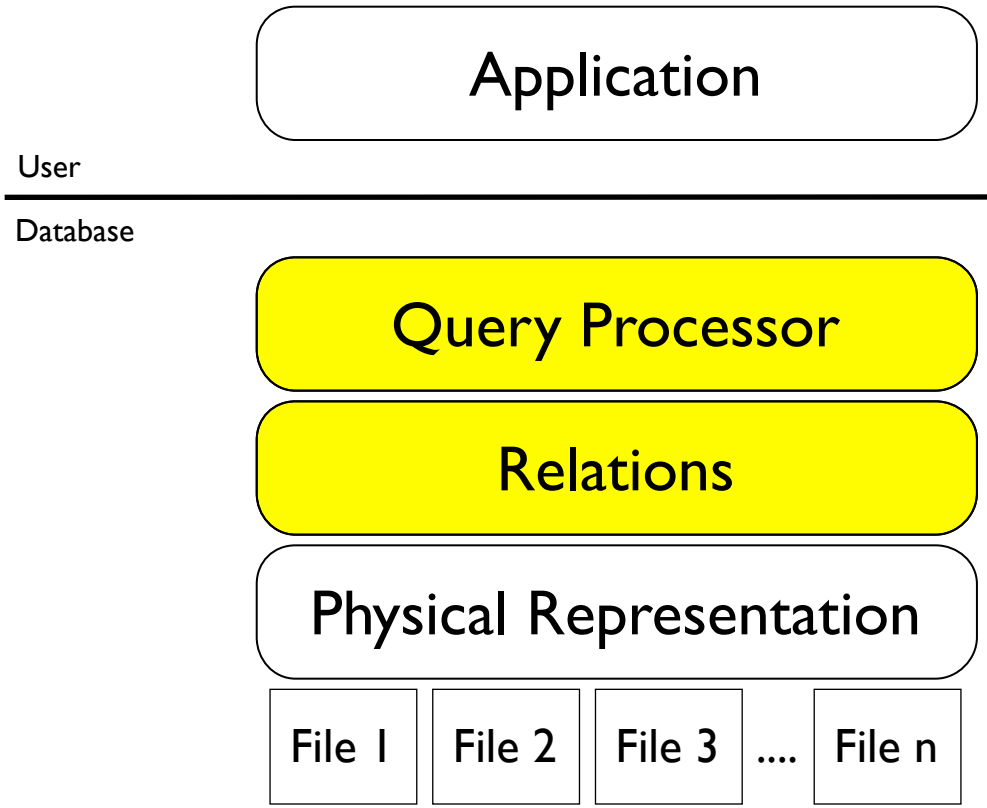
Physical Table

Customer_trojan		
segment_ID	attribute_ID	blob_data
1	name	smith, john, kim, joe
1	phone	2134, 3425, 6756, 9878
1	market_segment	automobile, household, furniture, building
2	name	mark, steve, jim, ian
2	phone	4312, 2435, 5766, 8789
2	market_segment	building, automobile, household, household

segment size = 4

* P. Larson et. al. SQL Server Column Store Indexes. SIGMOD 2011

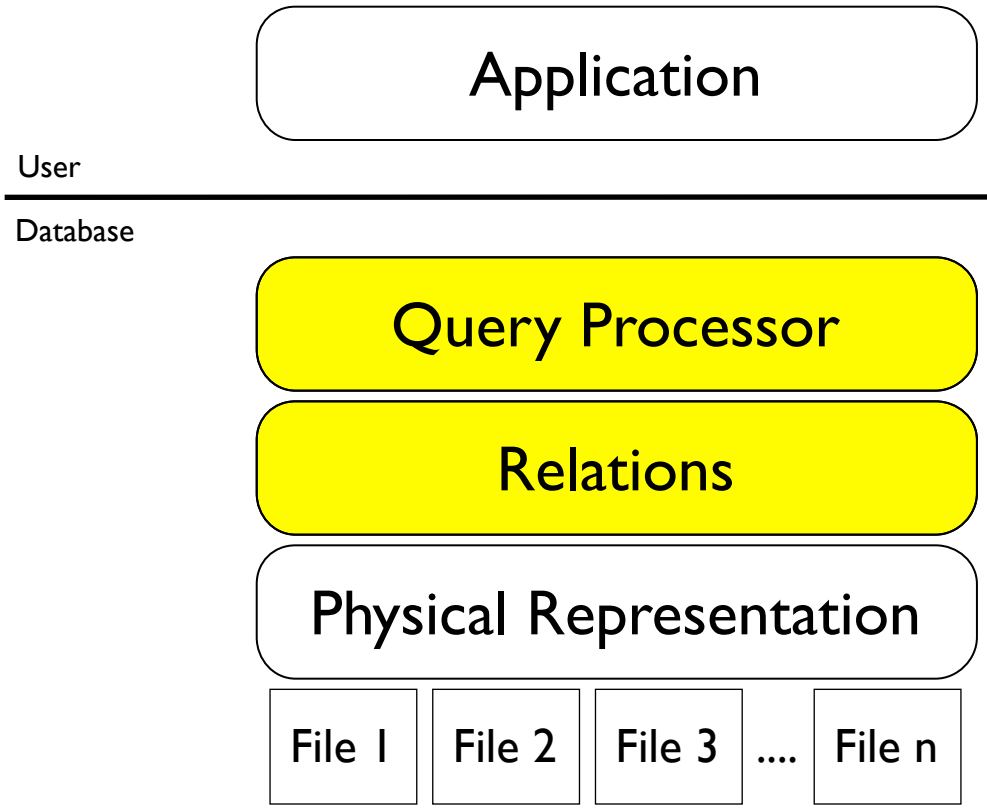
Column Index*



DEEP CHANGES !

* P. Larson et. al. SQL Server Column Store Indexes. SIGMOD 2011

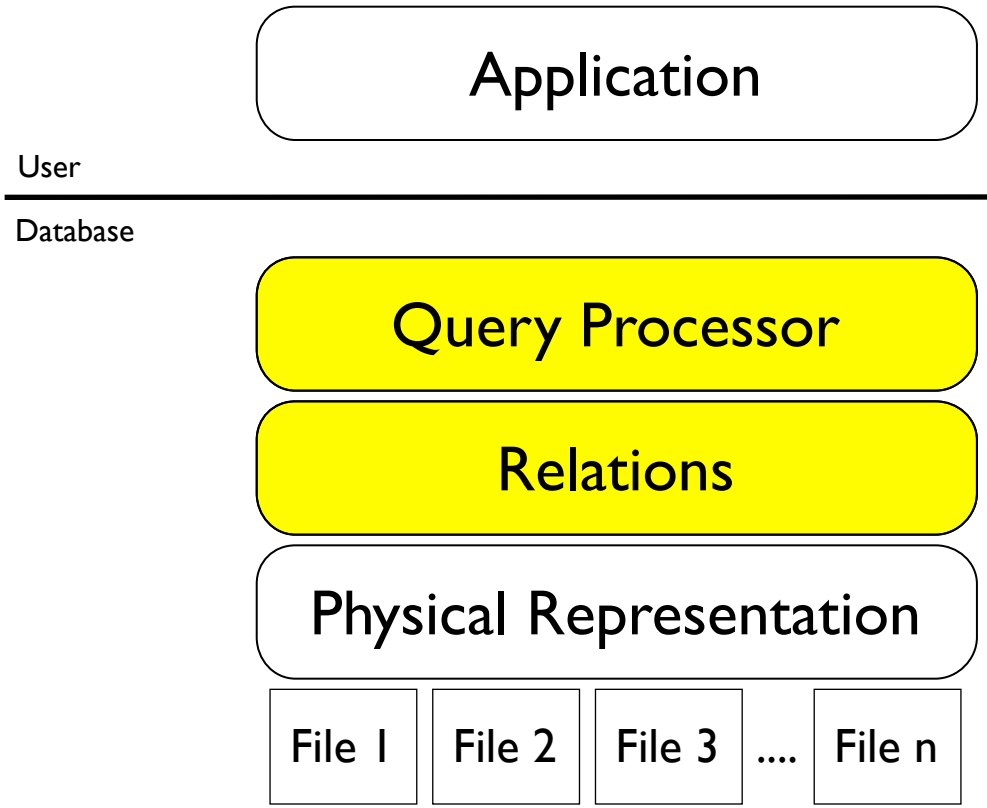
Column Index*



LONG TIME !

* P. Larson et. al. SQL Server Column Store Indexes. SIGMOD 2011

Column Index*



SOURCE CODE !

* P. Larson et. al. SQL Server Column Store Indexes. SIGMOD 2011

Column Index*

Application

User

Database

Query Processor

Relations

Physical Representation

File 1 | File 2 | File 3 | .... | File n

* P. Larson et. al. SQL Server Column Store Indexes. SIGMOD 2011

What do we propose?

Trojan Columns

Application

User

Database

Query Processor

Relations

UDF Storage Layer

Physical Representation

File 1 | File 2 | File 3 | .... | File n

Trojan Columns

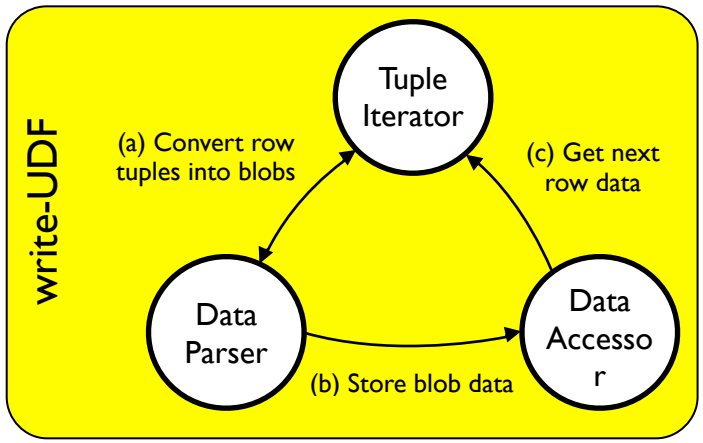
Relation

Customer		
name	phone	market_segment
smith	2134	automobile
john	3425	household
kim	6756	furniture
joe	9878	building
mark	4312	building
steve	2435	automobile
jim	5766	household
ian	8789	household

Physical Table

Customer_trojan		
segment_ID	attribute_ID	blob_data
1	name	smith, john, kim, joe
1	phone	2134, 3425, 6756, 9878
1	market_segment	automobile, household, furniture, building
2	name	mark, steve, jim, ian
2	phone	4312, 2435, 5766, 8789
2	market_segment	building, automobile, household, household

Trojan Columns



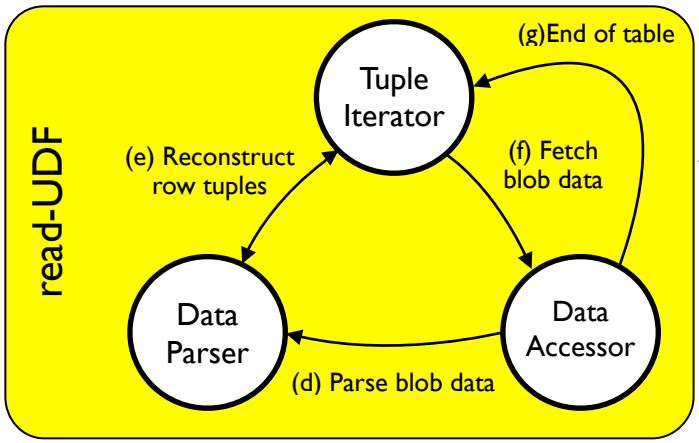
Relation

Customer		
name	phone	market_segment
smith	2134	automobile
john	3425	household
kim	6756	furniture
joe	9878	building
mark	4312	building
steve	2435	automobile
jim	5766	household
ian	8789	household

Physical Table

Customer_trojan		
segment_ID	attribute_ID	blob_data
1	name	smith, john, kim, joe
1	phone	2134, 3425, 6756, 9878
1	market_segment	automobile, household, furniture, building
2	name	mark, steve, jim, ian
2	phone	4312, 2435, 5766, 8789
2	market_segment	building, automobile, household, household

Trojan Columns



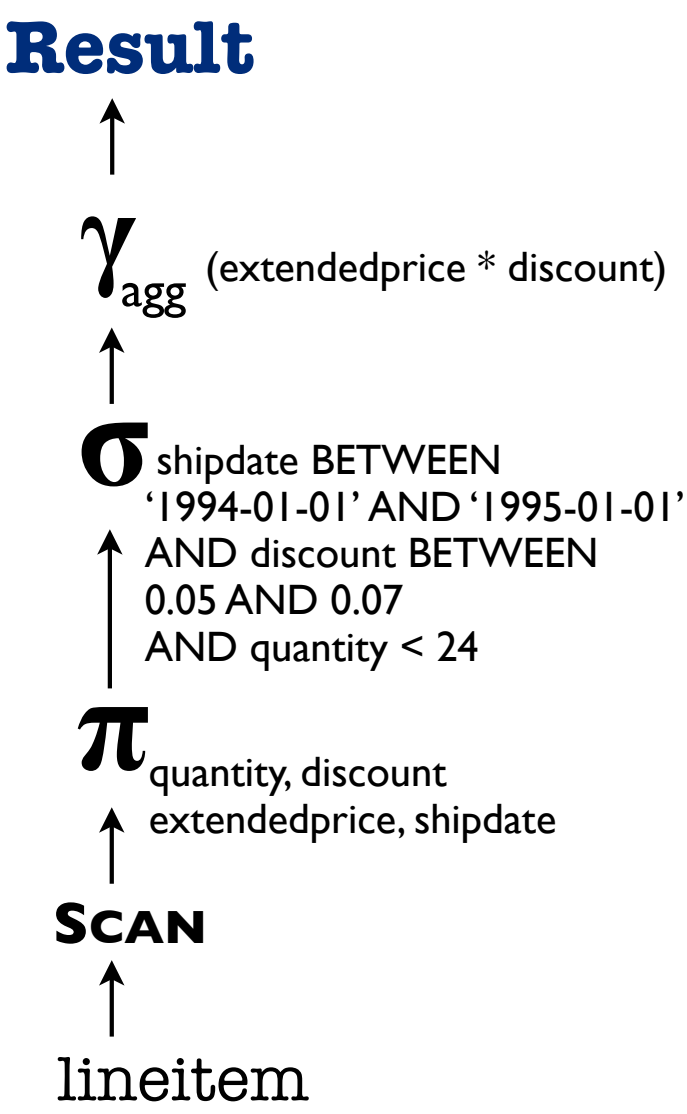
Relation

Customer		
name	phone	market_segment
smith	2134	automobile
john	3425	household
kim	6756	furniture
joe	9878	building
mark	4312	building
steve	2435	automobile
jim	5766	household
ian	8789	household

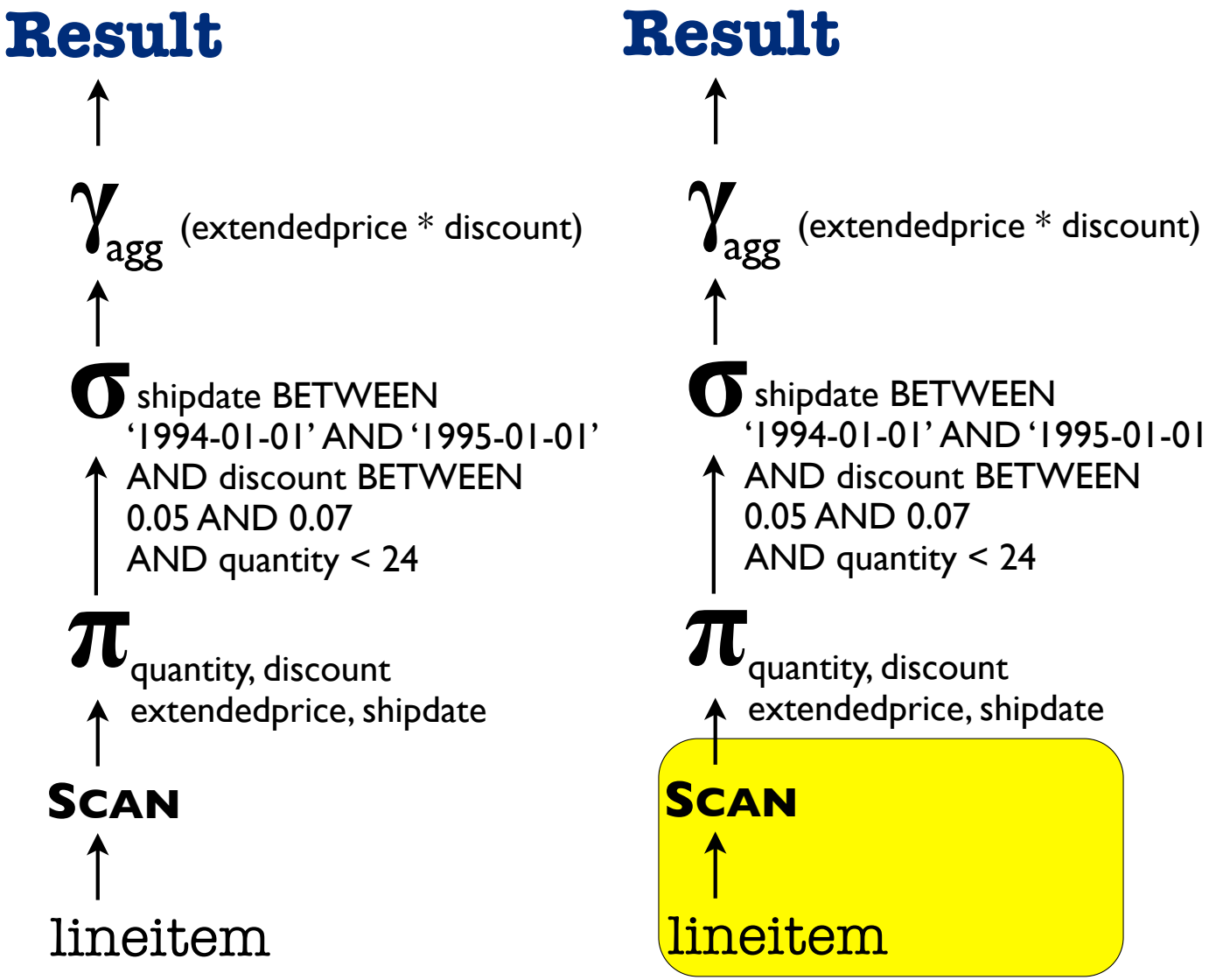
Physical Table

Customer_trojan		
segment_ID	attribute_ID	blob_data
1	name	smith, john, kim, joe
1	phone	2134, 3425, 6756, 9878
1	market_segment	automobile, household, furniture, building
2	name	mark, steve, jim, ian
2	phone	4312, 2435, 5766, 8789
2	market_segment	building, automobile, household, household

Example: TPC-H Query 6

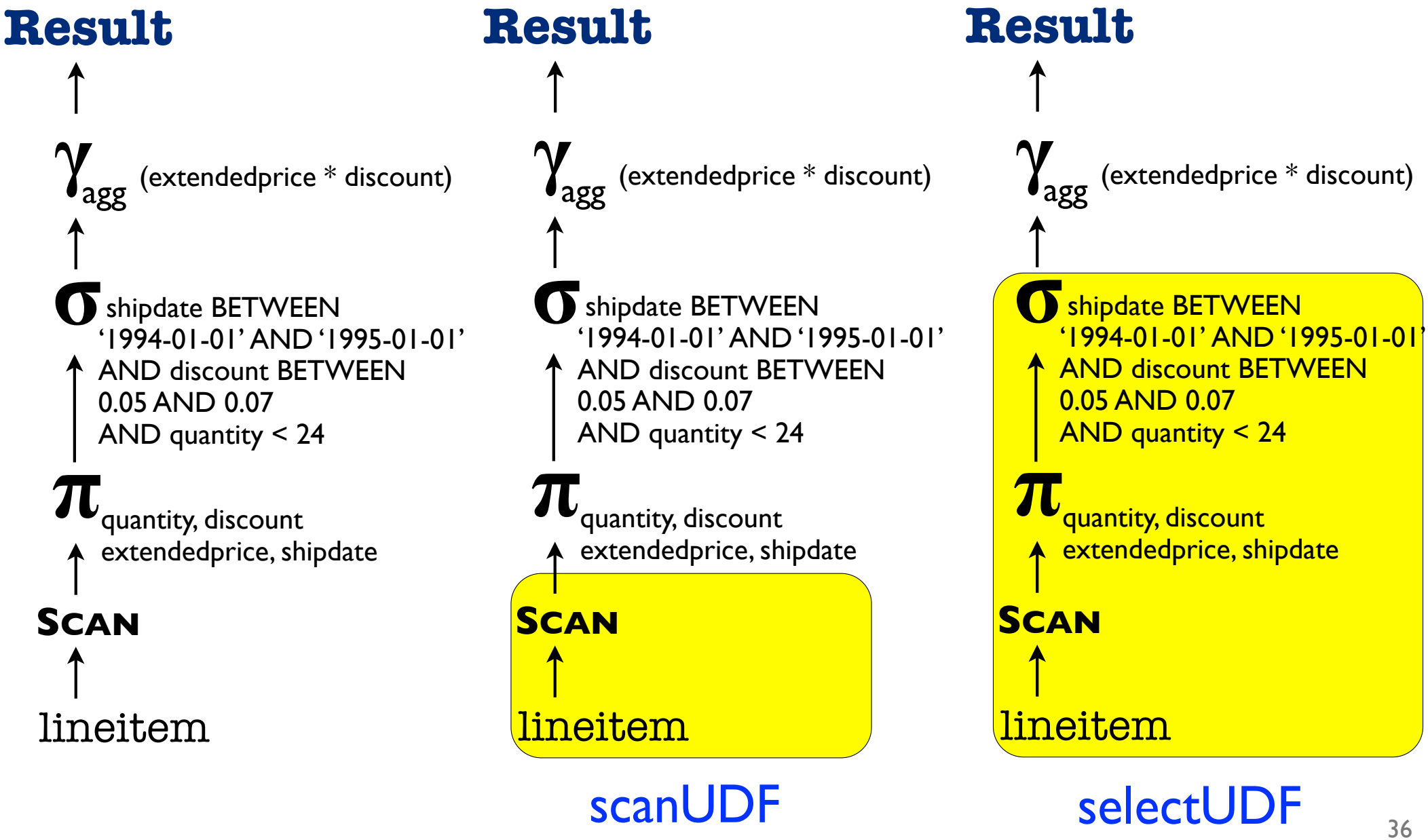


Example: TPC-H Query 6

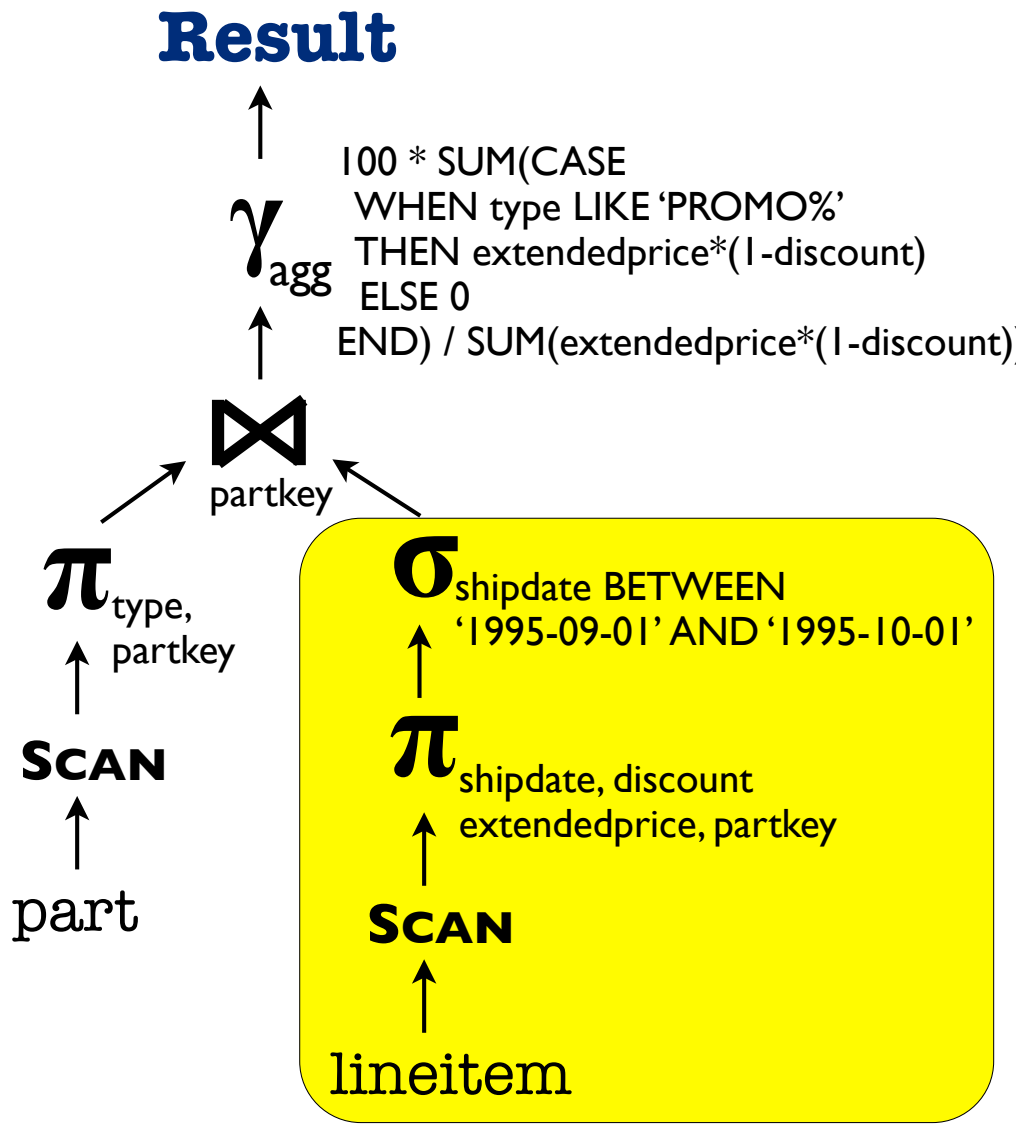


scanUDF

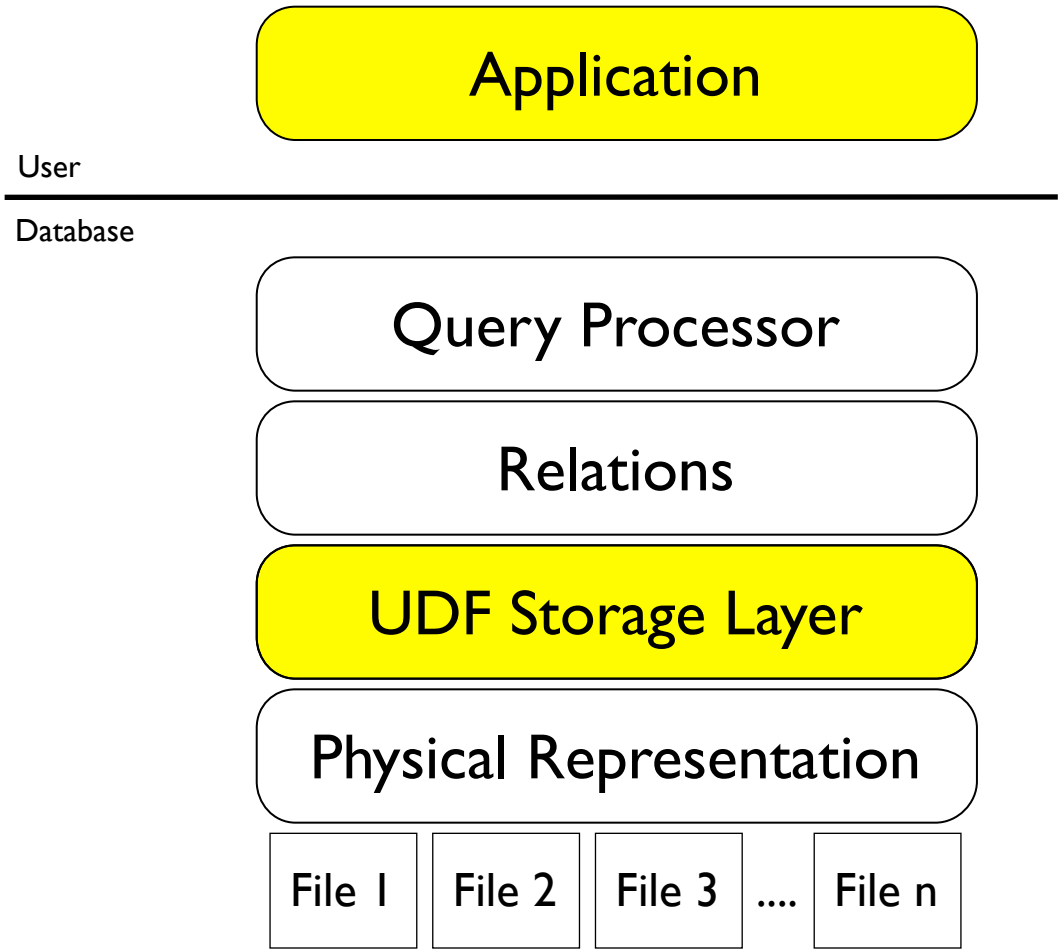
Example: TPC-H Query 6



Example: TPC-H Query 14



Trojan Columns



Plug-and-play

Trojan Columns

Application

User

Database

Query Processor

Relations

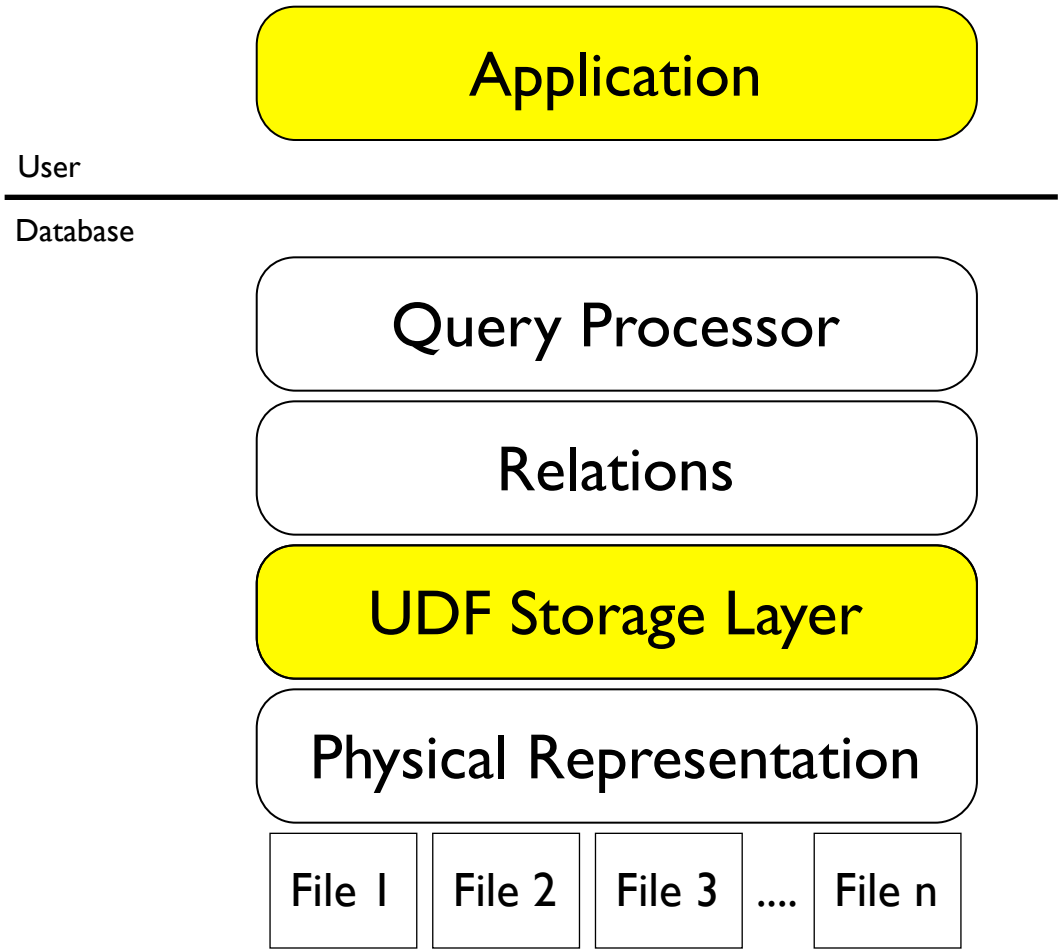
UDF Storage Layer

Physical Representation

File 1 | File 2 | File 3 | .... | File n

Quick Deployment

Trojan Columns



Closed-source

Trojan Columns

Application

User

Database

Query Processor

Relations

UDF Storage Layer

Physical Representation

File 1 | File 2 | File 3 | .... | File n

Will this work?

Experimental Setup

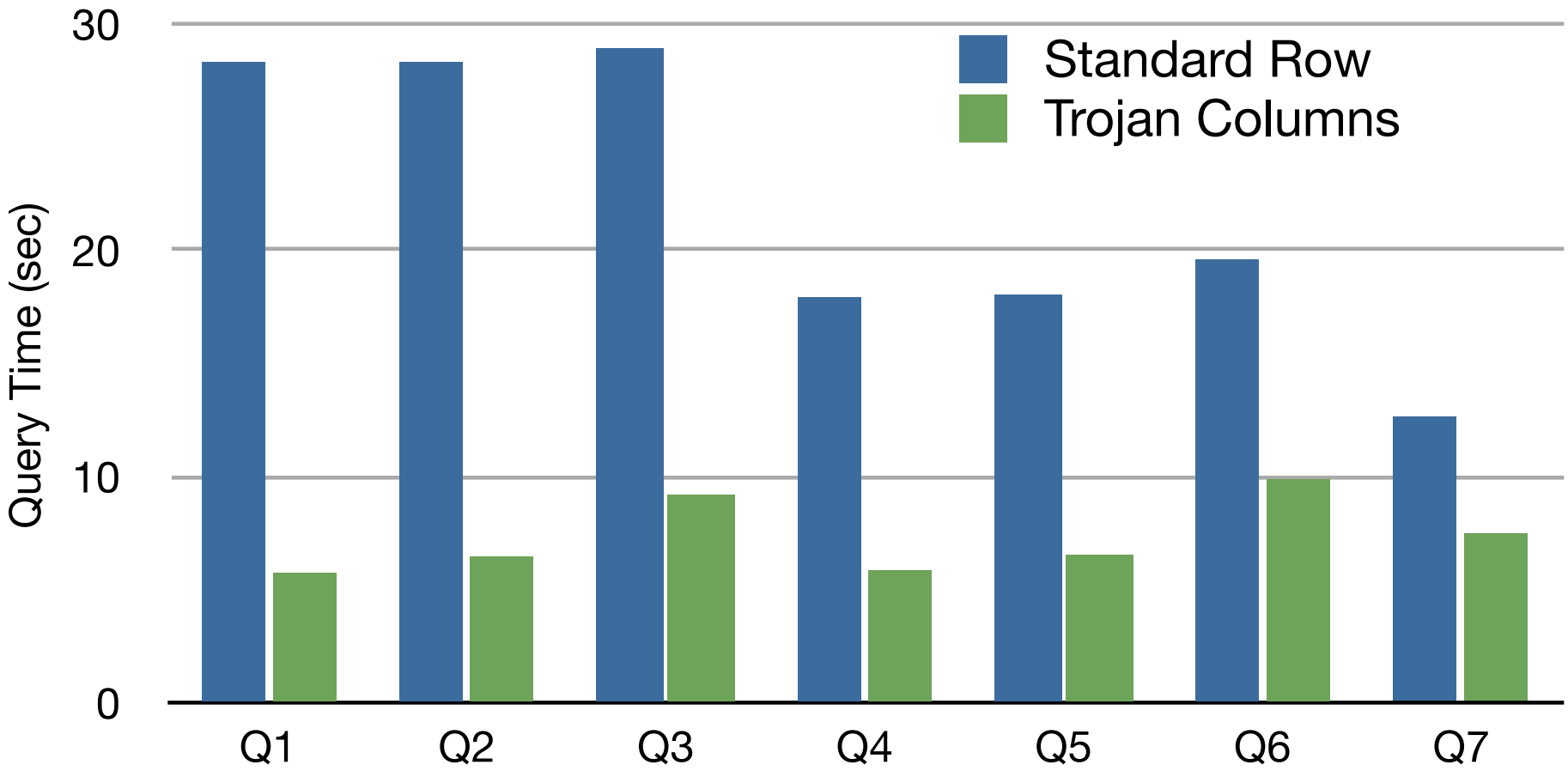
Commercial closed-source Row-store (Standard Row)

Trojan Columns in commercial closed-source Row-store (Trojan Columns)

Three variants of TPC-H benchmark:

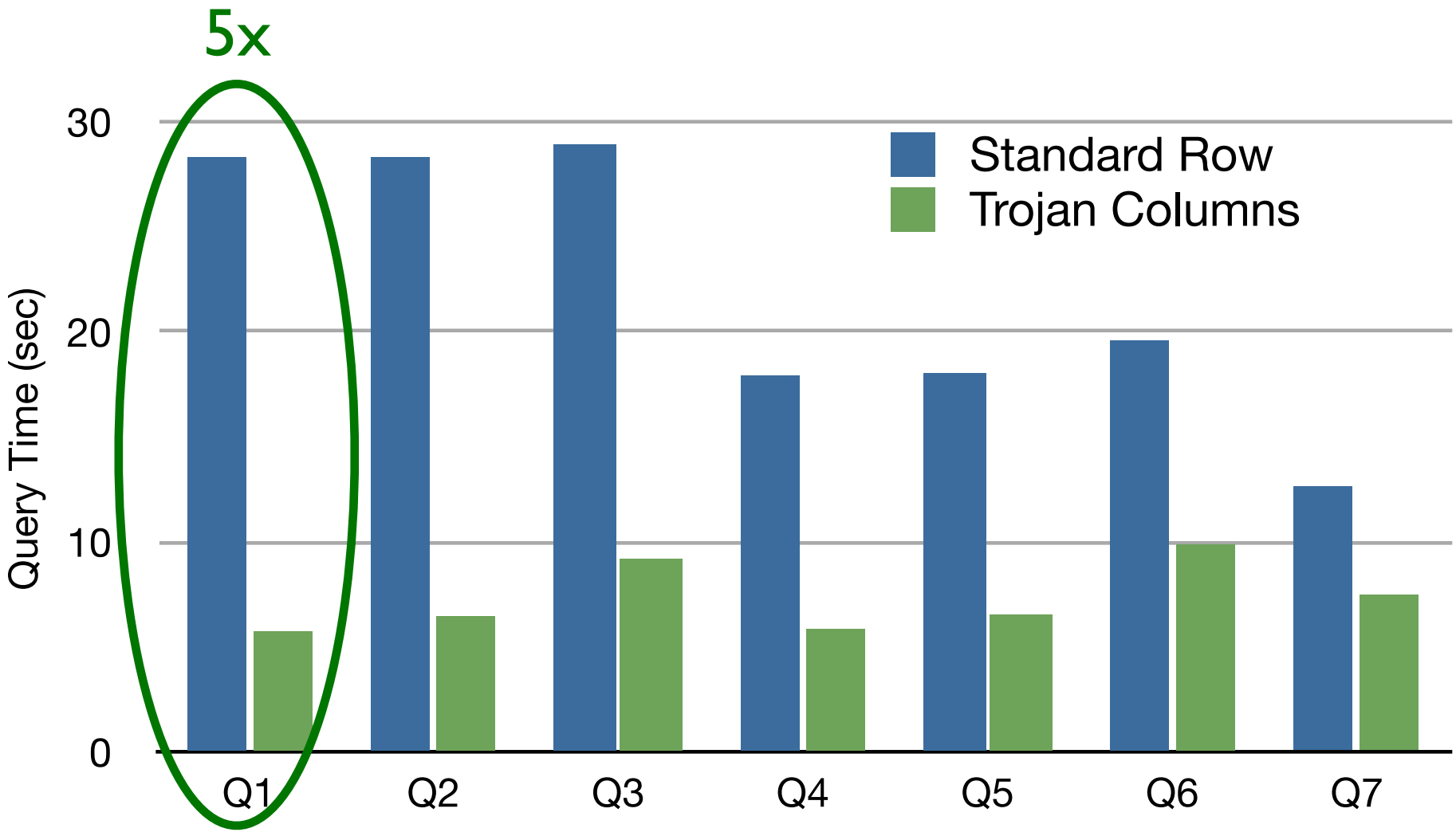
1. simplified queries , simplified dataset
2. simplified queries , original dataset
3. original queries , original dataset

Simplified Queries, Simplified Dataset *



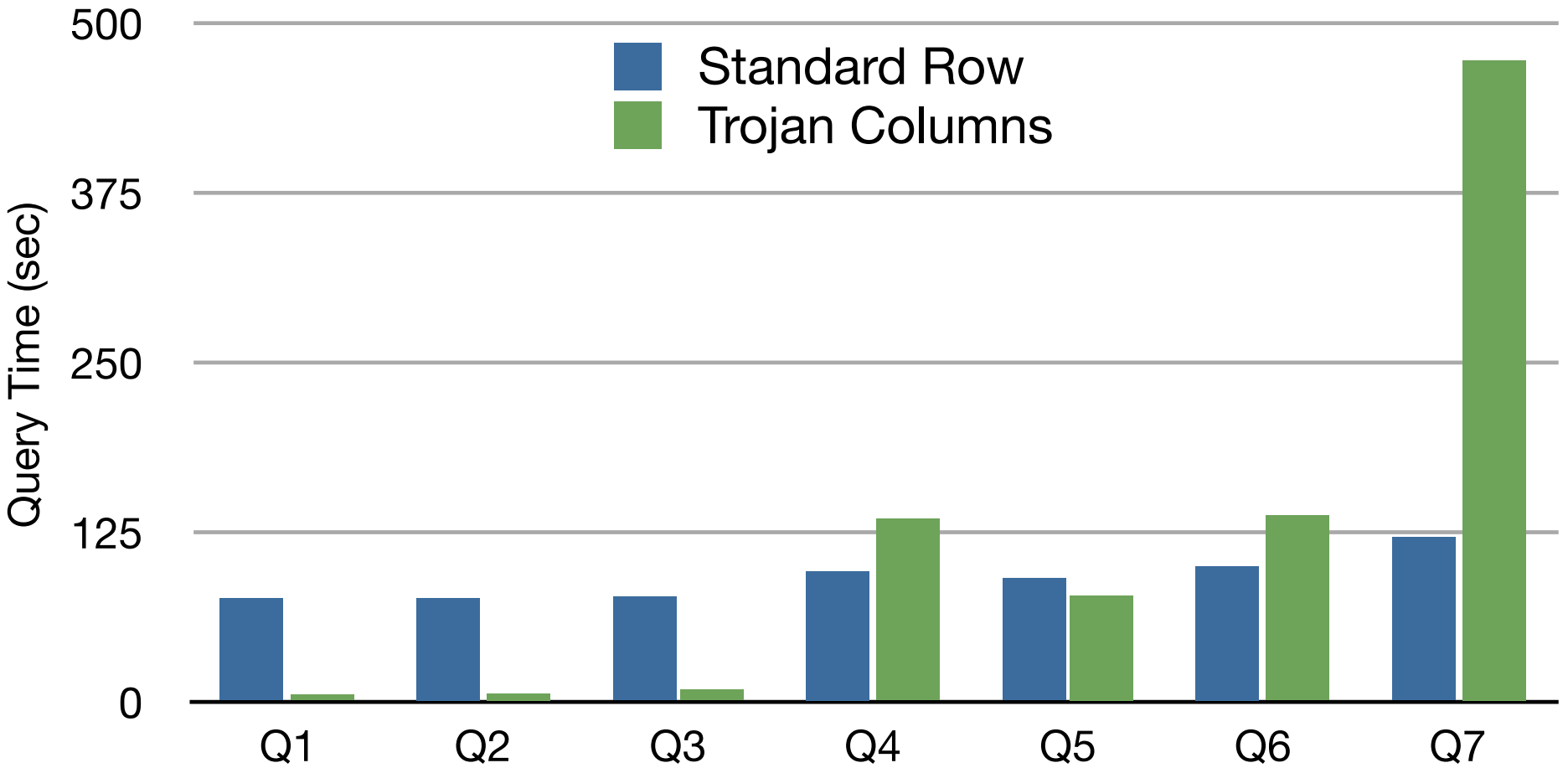
* Mike Stonebraker et. al. C-Store: A Column Oriented DBMS. VLDB 2005

Simplified Queries, Simplified Dataset *

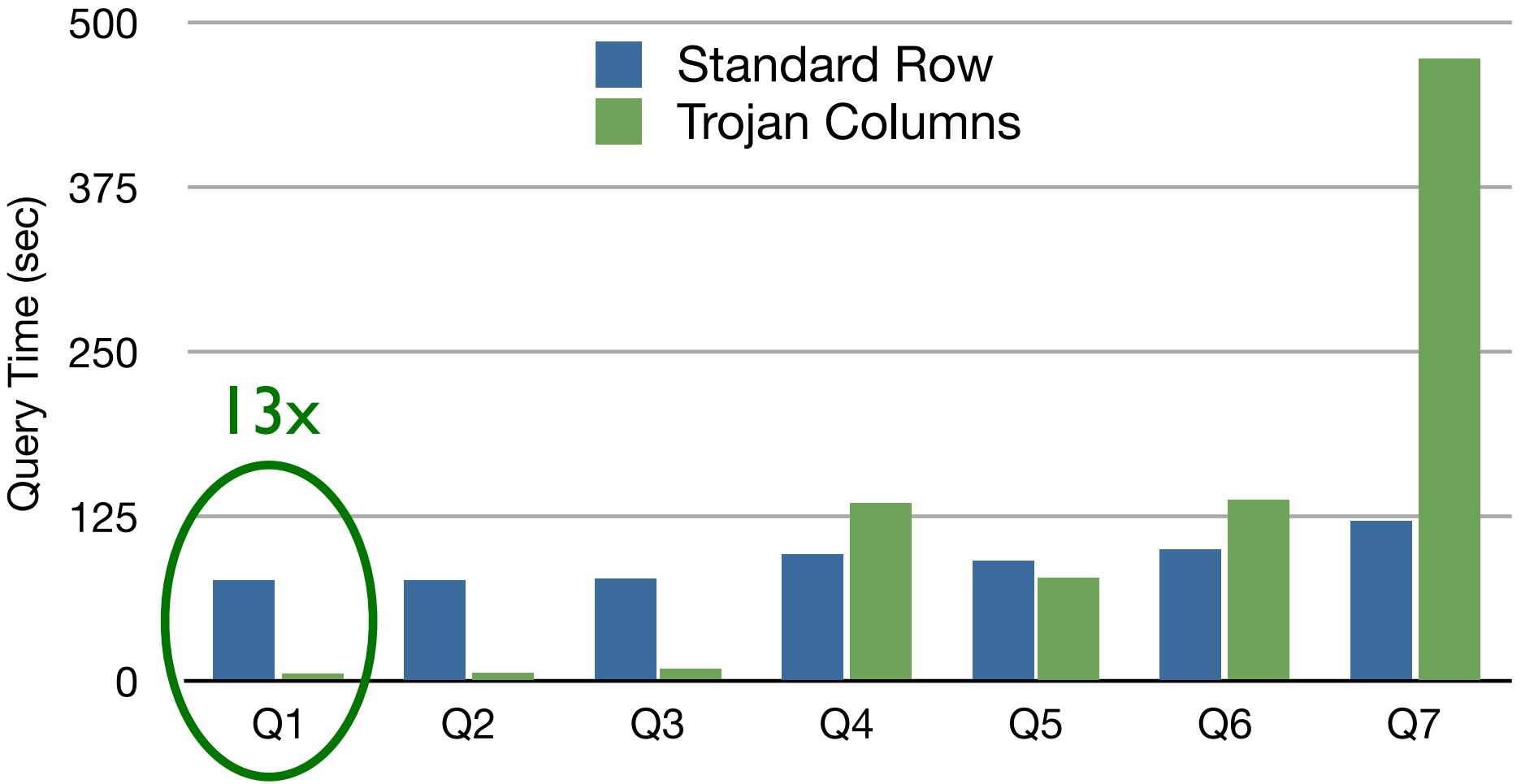


* Mike Stonebraker et. al. C-Store: A Column Oriented DBMS. VLDB 2005

Simplified Queries, Original Dataset

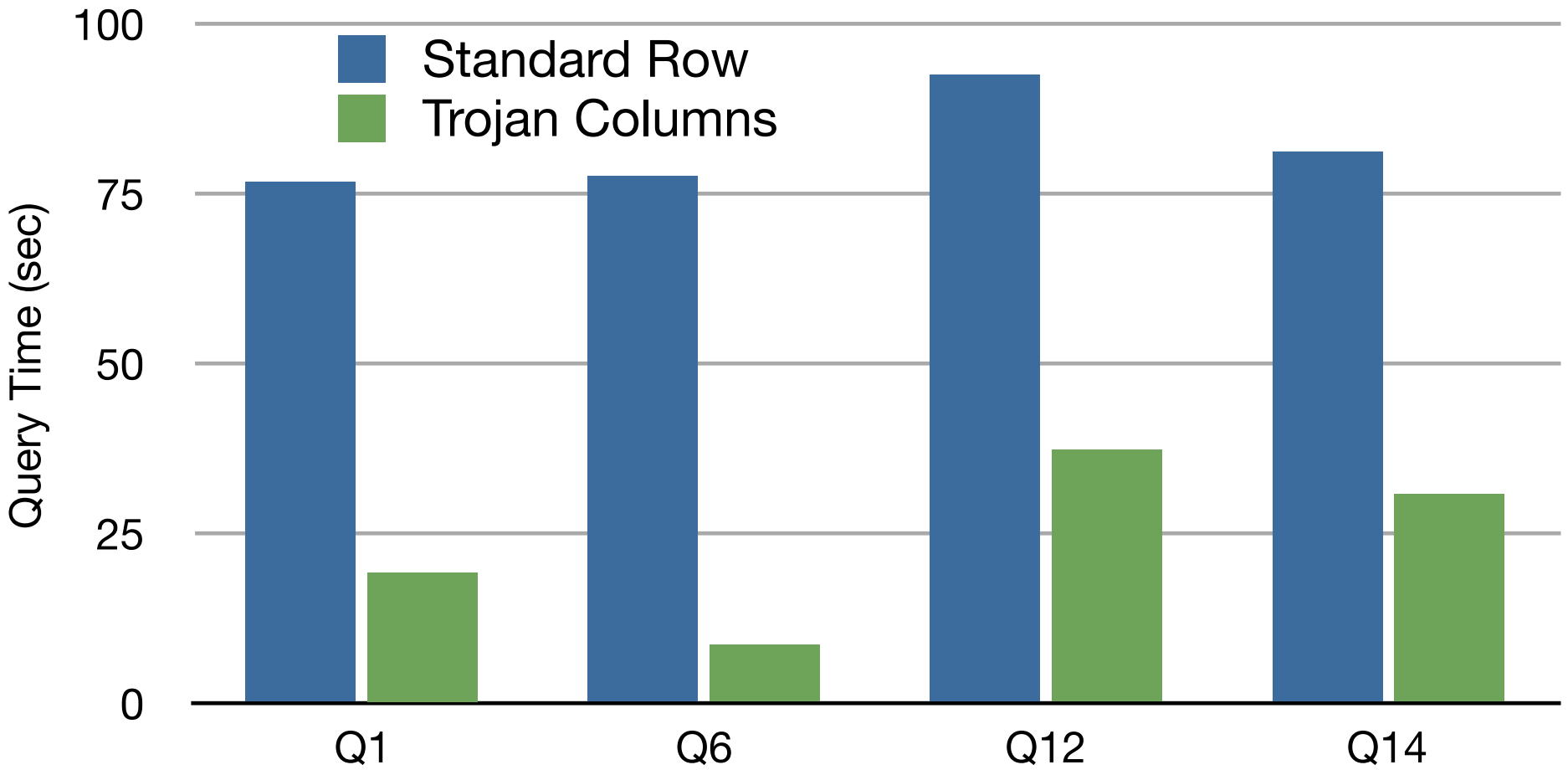


Simplified Queries, Original Dataset



Original Queries, Original Dataset *

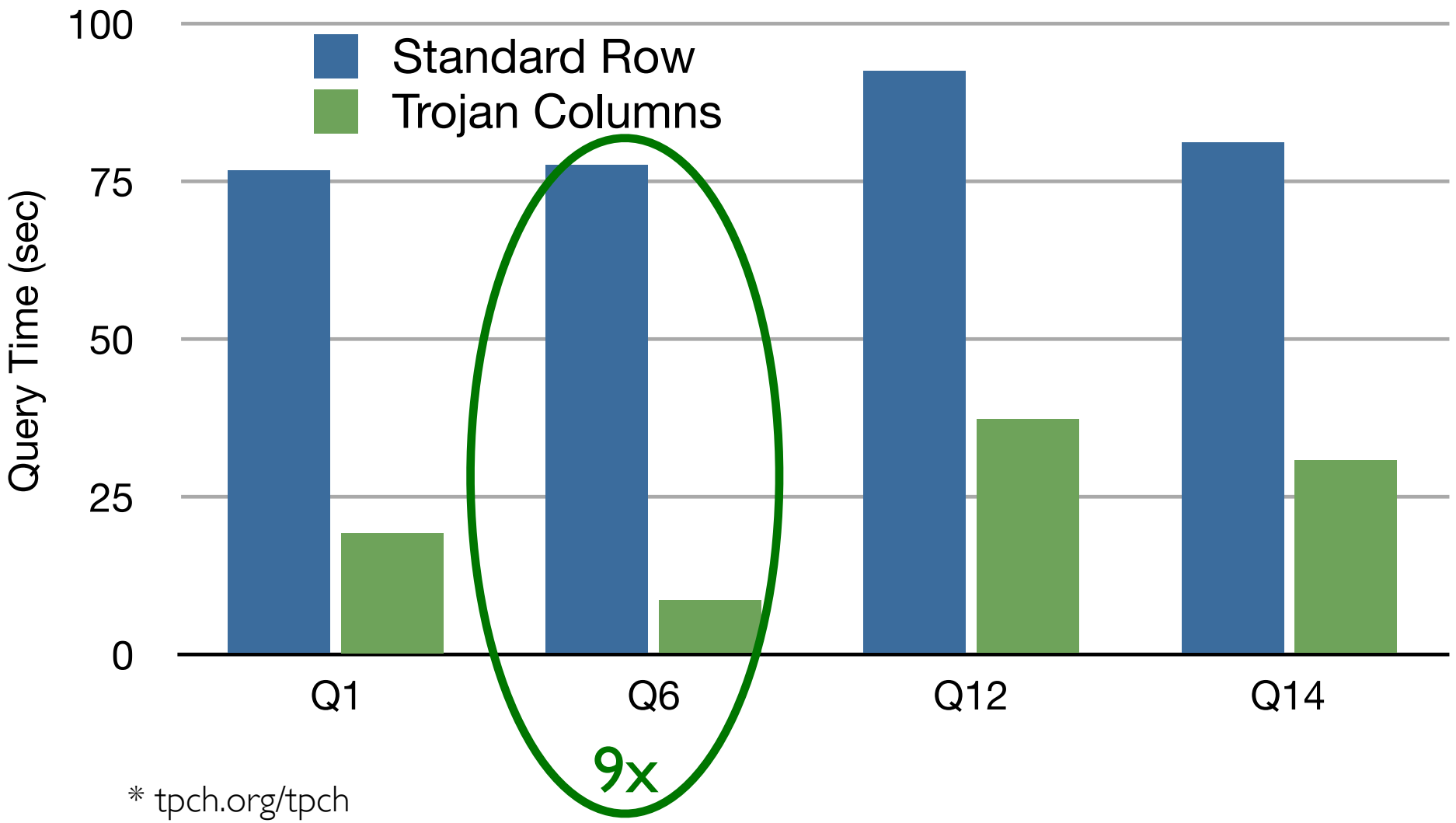
The "Good Queries"



* tpch.org/tpch

Original Queries, Original Dataset *

The "Good Queries"

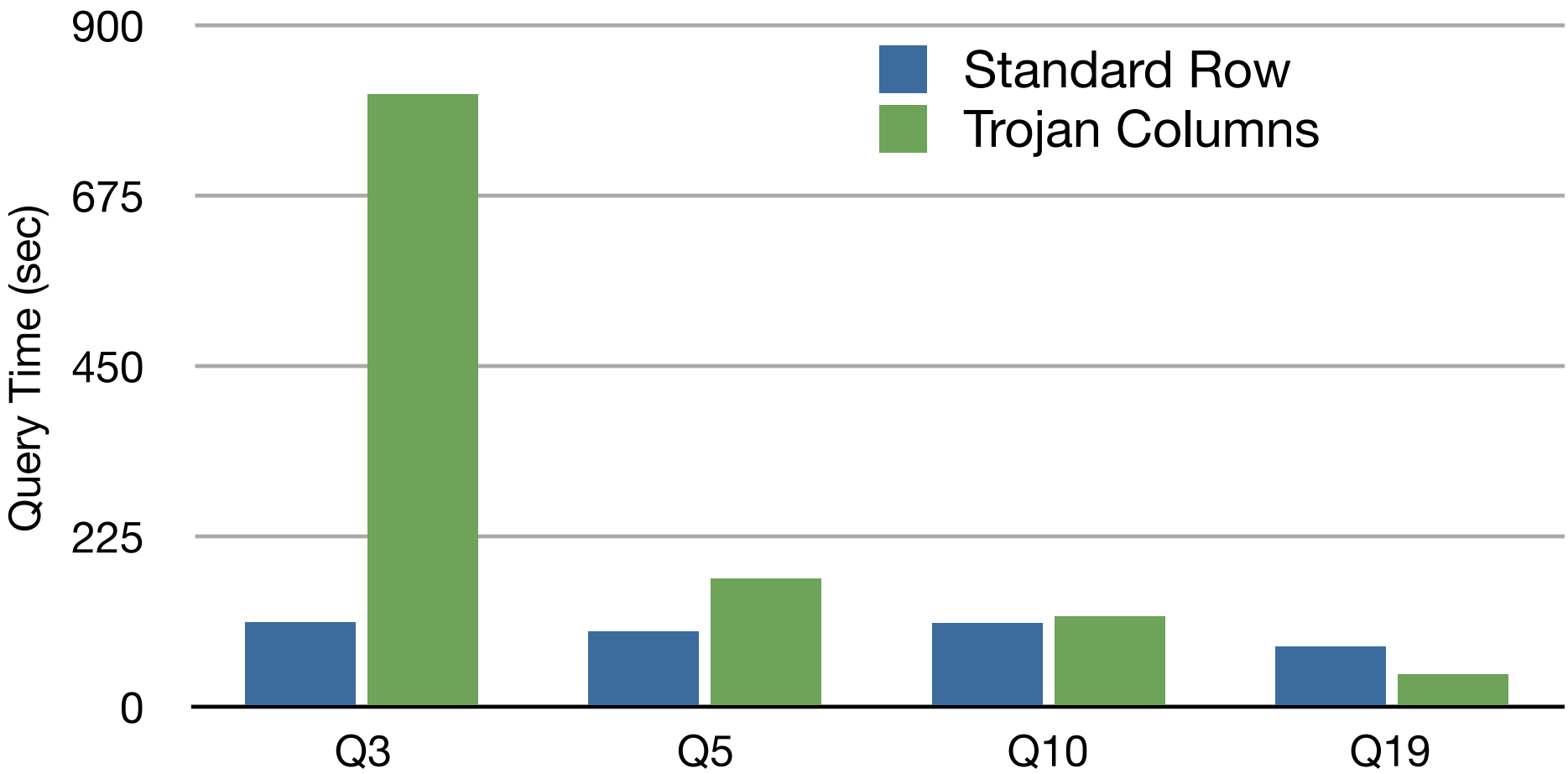


* tpch.org/tpch

What are the trade-offs?

Original Queries, Original Dataset *

The "Bad Queries"



* tpch.org/tpch

Micro-Benchmark: Improvement over Row-store

# referenced attributes (r)	1E-06	1E-05	1E-04	1E-03	1E-02	1E-01	1E+00
16	2.13	2.13	2.11	2.06	1.55	0.47	0.06
15	4.64	4.62	4.55	4.27	2.57	0.55	0.06
13	5.00	5.00	4.94	4.61	2.70	0.57	0.06
11	5.79	5.82	5.75	5.24	2.87	0.56	0.06
9	6.39	6.38	6.25	5.79	3.11	0.54	0.06
7	7.00	6.96	6.80	6.23	3.17	0.56	0.06
5	10.96	10.94	10.55	9.27	3.75	0.57	0.06
3	12.86	13.57	13.22	11.03	4.16	0.56	0.06
1	17.43	17.61	16.61	13.57	4.39	0.57	0.06

selectivity (fraction of tuples accessed)

Micro-Benchmark: Improvement over Row-store

Not Affected

referenced attributes (r)

	1E-06	1E-05	1E-04	1E-03	1E-02	1E-01	1E+00
16	2.13	2.13	2.11	2.06	1.55	0.47	0.06
15	4.64	4.62	4.55	4.27	2.57	0.55	0.06
13	5.00	5.00	4.94	4.61	2.70	0.57	0.06
11	5.79	5.82	5.75	5.24	2.87	0.56	0.06
9	6.39	6.38	6.25	5.79	3.11	0.54	0.06
7	7.00	6.96	6.80	6.23	3.17	0.56	0.06
5	10.96	10.94	10.55	9.27	3.75	0.57	0.06
3	12.86	13.57	13.22	11.03	4.16	0.56	0.06
1	17.43	17.61	16.61	13.57	4.39	0.57	0.06

selectivity (fraction of tuples accessed)

Micro-Benchmark: Improvement over Row-store

Not Affected

referenced attributes (r)

	1E-06	1E-05	1E-04	1E-03	1E-02	1E-01	1E+00
16	2.13	2.13	2.11	2.06	1.55	0.47	0.06
15	4.64	4.62	4.55	4.27	2.57	0.55	0.06
13	5.00	5.00	4.94	4.61	2.70	0.57	0.06
11	5.79	5.82	5.75	5.24	2.87	0.56	0.06
9	6.39	6.38	6.25	5.79	3.11	0.54	0.06
7	7.00	6.96	6.80	6.23	3.17	0.56	0.06
5	10.96	10.94	10.55	9.27	3.75	0.57	0.06
3	12.86	13.57	13.22	11.03	4.16	0.56	0.06
1	17.43	17.61	16.61	13.57	4.39	0.57	0.06

selectivity (fraction of tuples accessed)

Affected

How far are we?

Four Systems

Commercial Row-store (Standard Row)

Trojan Columns in commercial Row-store

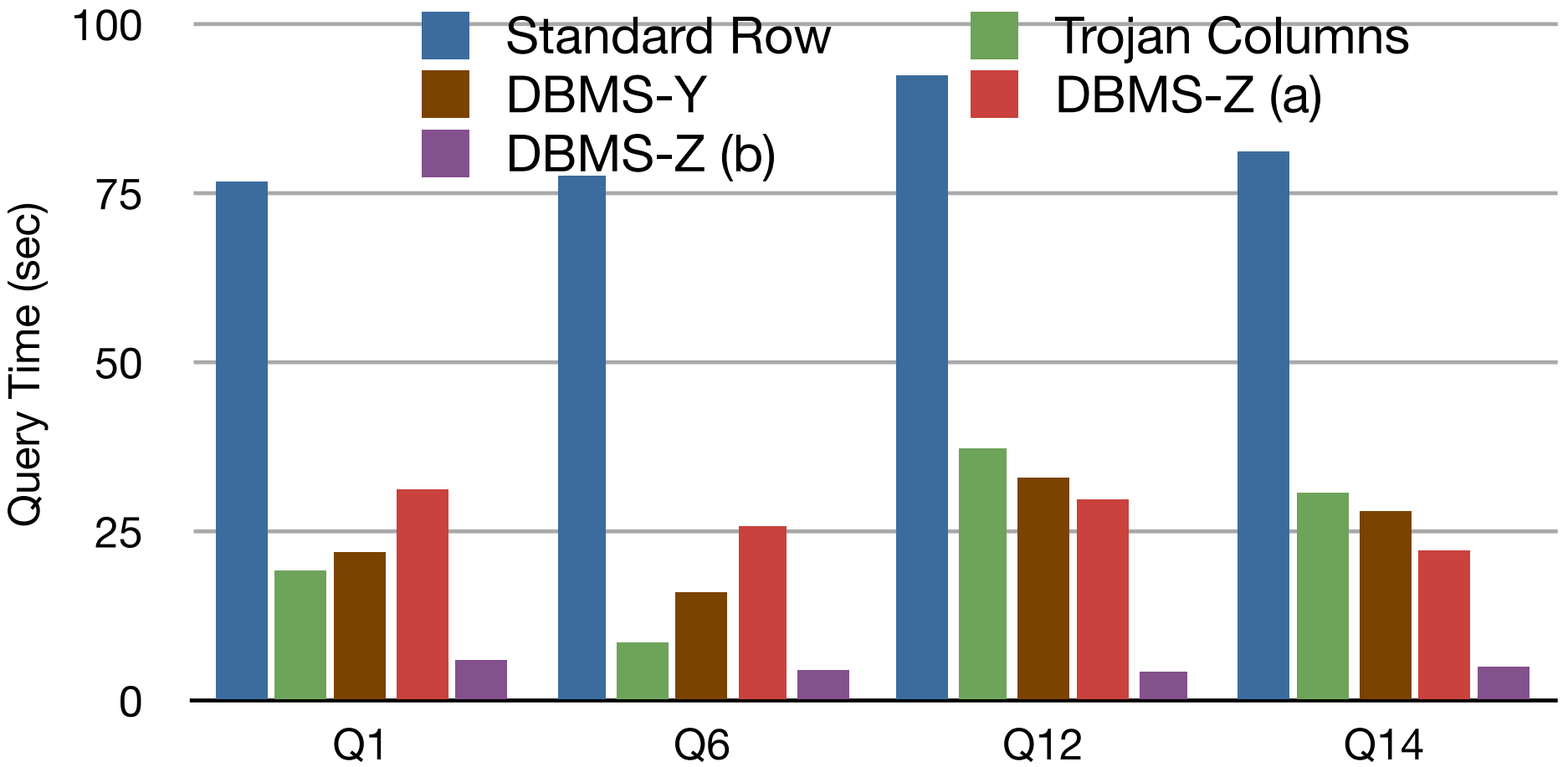
Commercial Row-store with vendor support for column technology (DBMS-Y)

Commercial Column-store (DBMS-Z)

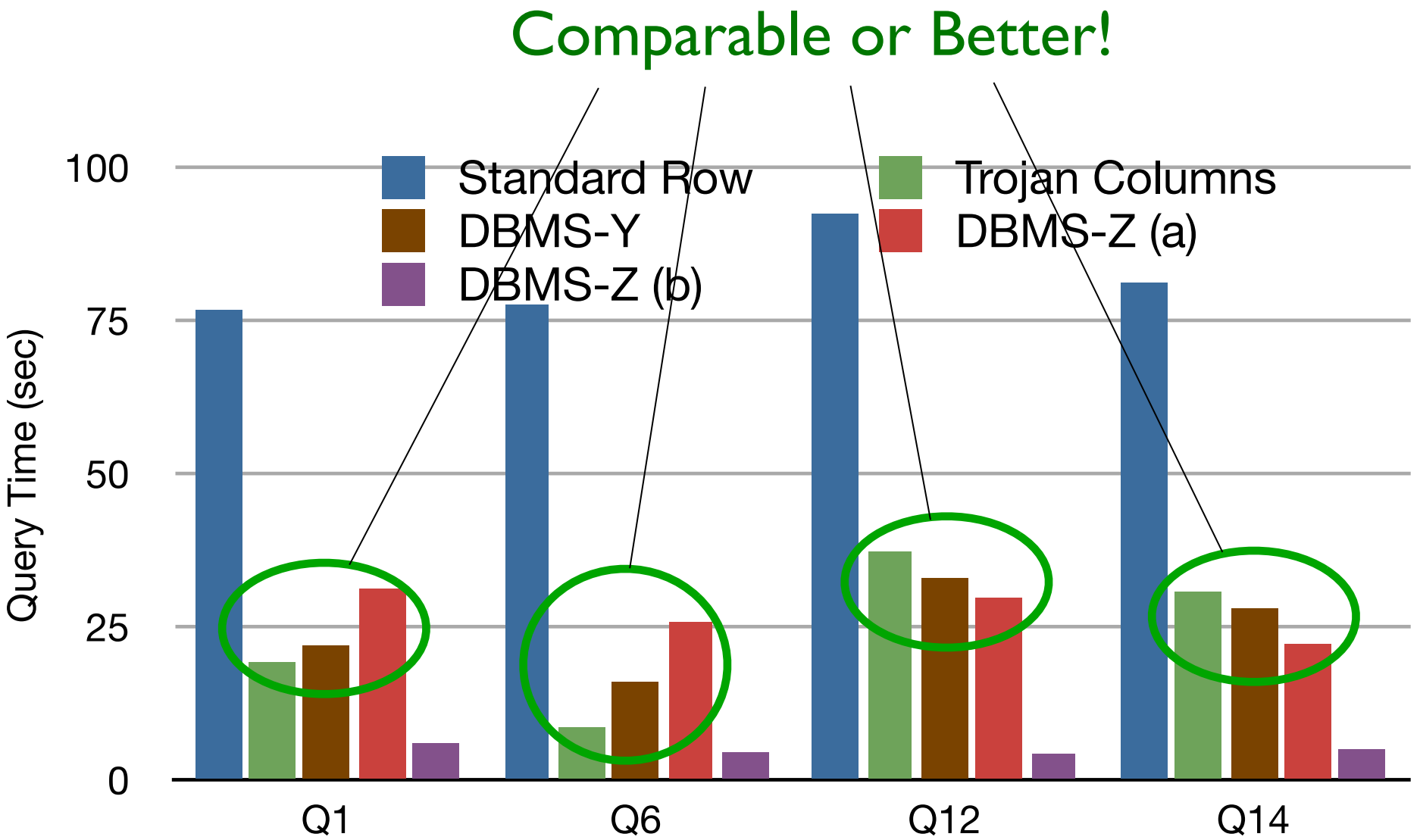
(a) default TPC-H schema

(b) tuned schema

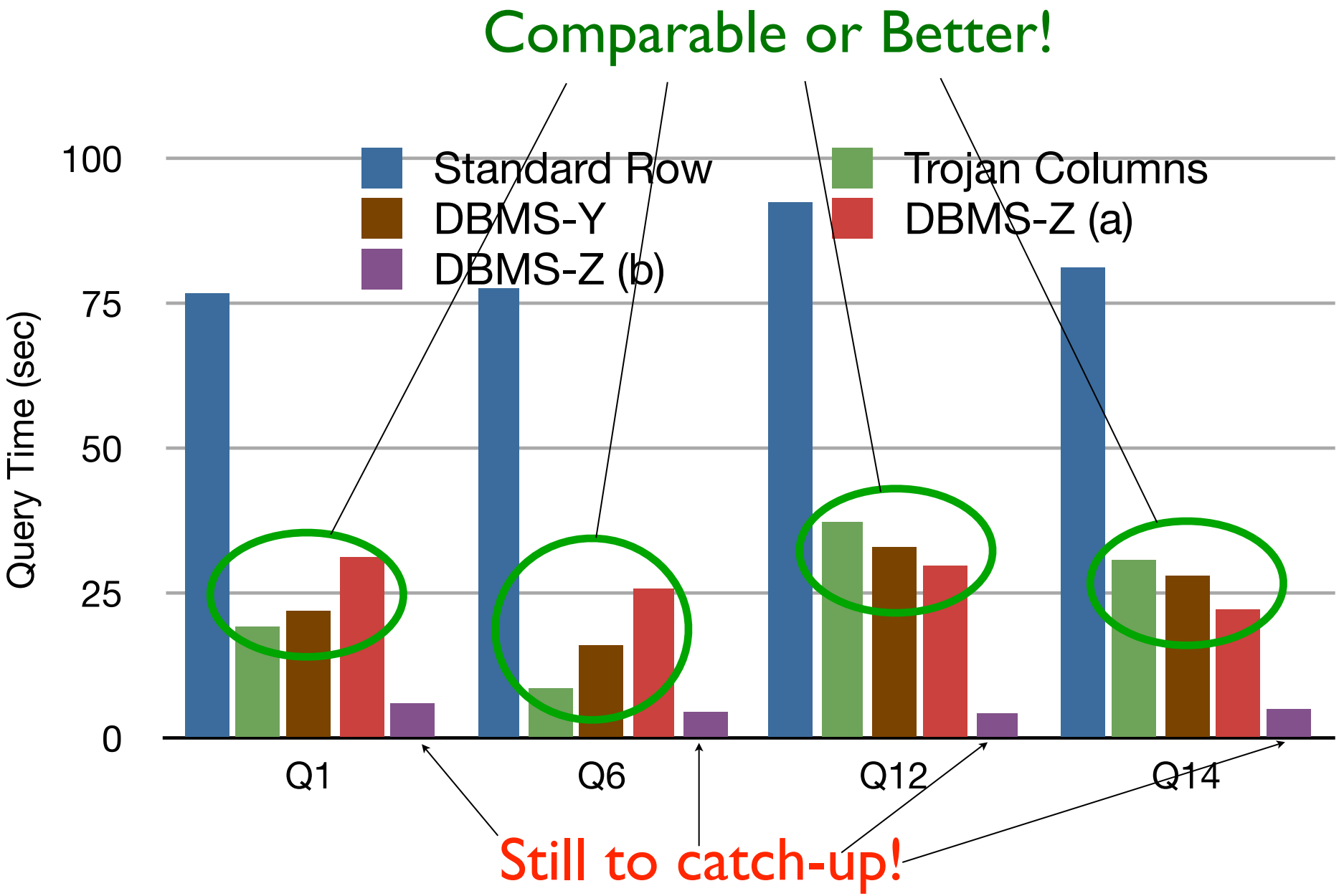
TPC-H Benchmark



TPC-H Benchmark



TPC-H Benchmark



What about query optimization?

Rules out query optimization?

Rules out query optimization?

NO!

Rules out query optimization?

NO!

QO with aggregate UDFs [SIGMOD'06]

Manimal [WebDB'10]

HadoopToSQL [EuroSys'10]

Black box QO [VLDB'12]

,,

The UDF Business Model

UDFs

Not just for application-specific code

Integrate **core database functionality after the fact**

Column layouts are **just one** example!

Meet customer demands quickly

Provide quick feedback before new product release

Summary

row-store

slow good-enough fast

performance

Summary

row-store

native
column-store

slow good-enough fast

performance

Summary

row-store

Trojan Columns

native
column-store

slow

good-enough

fast

performance