

SINEW: A SQL SYSTEM FOR MULTI- STRUCTURED DATA

DANIEL TAHARA, YALE UNIVERSITY

THADDEUS DIAMOND, HADAPT

DANIEL J. ABADI, YALE UNIVERSITY

A MOTIVATING EXAMPLE

Relational Datastore

userID	name	email	preferred
1	Walter White	heisenberg@bb.com	false
2	Hank Schrader	mineral@rock.com	true
3	Jesse Pinkman	pink@thenewblack.ca	true
4	Gustavo Fring	gus@lospollos.com	true

Text Datastore

userId	text
1	What is your return policy?
2	Do you have bulk discounts?
4	I can't figure out your Website. You should fire your designer!

What if you want to send an email to all customers who indicate a potential desire to buy in bulk, with a deal on items they looked at?

JSON Datastore

JSON Documents
<pre>{ "userID": 1, "action": "view", "productName": "methylanine", "price": 12.50, "color": "blue", "tags": ["industrial", "barrel"] }</pre>
<pre>{ "userID": 3, "action": "purchase", "price": 57.12 }</pre>
<pre>{ "userID": 2, "action": "view", "productName": "sudafed", "color": "red", "price": 52.50, "tags": ["home", "medical"] }</pre>

OPTION 1: SCHEMA-ON-WRITE

Relational Datastore

userID	name	email	preferred
1	Walter White	heisenberg@bb.com	false
2	Hank Schrader	mineral@rock.com	true
3	Jesse Pinkman	pink@thenewblack.ca	true
4	Gustavo Fring	gus@lospollos.com	true

Text Datastore

userId	text
1	What is your return policy?
2	Do you have bulk discounts?
4	I can't figure out your Website. You should fire your designer!

Use ETL to combine data from separate backends into one unified datastore.



JSON Datastore

JSON Documents
<pre>{ "userID": 1, "action": "view", "productName": "methamphetamine", "price": 12.50, "color": "blue", "tags": ["industrial", "barrel"] }</pre>
<pre>{ "userID": 3, "action": "purchase", "price": 57.12 }</pre>
<pre>{ "userID": 2, "action": "view", "productName": "sulfadiazine", "color": "red", "price": 52.50, "tags": ["home", "medical"] }</pre>

OPTION 2: SCHEMA-ON-READ

JSON Documents
<pre>{ "userID": 1, "action": "view", "productName": "methamphetamine", "price": 12.50, "color": "blue", "tags": ["industrial", "barrel"] }</pre>
<pre>{ "userID": 3, "action": "purchase", "price": 57.12 }</pre>
<pre>{ "userID": 2, "action": "view", "productName": "sulfadiazine", "color": "red", "price": 52.50, "tags": ["home", "medical"] }</pre>



userID	key	value
1	action	"view"
1	price	12.50
1	tags	["industrial", "barrel"]
1	productName	"methamphetamine"
1	color	"blue"
2	action	"view"
2	productName	"sulfadiazine"
...

OPTION 2: SCHEMA-ON-READ

JSON Documents
<pre>{ "userID": 1, "action": "view", "productName": "methamphetamine", "price": 12.50, "color": "blue", "tags": ["industrial", "barrel"] }</pre>
<pre>{ "userID": 3, "action": "purchase", "price": 57.12 }</pre>
<pre>{ "userID": 2, "action": "view", "productName": "sudafed", "color": "red", "price": 52.50, "tags": ["home", "medical"] }</pre>



userID	preferred	product Name	price	tags	color	action
1	no	methamphetamine	12.50	industrial barrel	blue	view
2	yes	sudafed	52.50	home medical	red	view
3	yes		57.12			purchase
4	yes					

FEATURES

Automatic generation of a queryable schema

Full range of SQL primitives

Architecture extends an RDBMS but requires no modification to RDBMS code

OVERVIEW

Universal Logical Schema

Hybrid Physical Schema

Dynamic Column Materialization

Text Indexing and Search

Benchmarks

Conclusions

UNIVERSAL LOGICAL SCHEMA

userID	name	emailAddr	preferred	productName	price	tags	color	action	emails
1	Walter White	heisenberg@bb.com	no	methylamine	12.50	industrial barrel	blue	view	What is your return policy?
2	Hank Schrader	mineral@rock.com	yes	sudafed	52.50	home medical	red	view	Do you have bulk discounts?
3	Jesse Pinkman	pink@thenewblack.ca	yes		57.12			purchase	
4	Gustavo Fring	gus@lospollos.com	yes						I can't figure out your Website. You should fire your designer!


```
SELECT name,  
       emailAddr,  
       productName  
FROM T  
WHERE action = 'view'  
       AND emails like '%bulk%';
```


HYBRID PHYSICAL SCHEMA

userID		preferred	productName	price	tags	color	action	
1	...	no	methylanine	12.50	industrial barrel	blue	view	...
2	...	yes	sudafed	52.50	home medical	red	view	...
3	...	yes		57.12			purchase	...
4	...	yes						...

HYBRID PHYSICAL SCHEMA

userID		preferred	productName	price	tags	color	action	
1	...	no	methyamine	12.50	industrial barrel	blue	view	...
2	...	yes	sudafed	52.50	home medical	red	view	...
3	...	yes		57.12			purchase	...
4	...	yes						...



userID		preferred	productName	price	column reservoir	
1	...	no	methyamine	12.50	tags: [industrial, barrel] color: blue action: view	...
2	...	yes	sudafed	52.50	tags: [home, medical] color: red action: view	...
3	...	yes		57.12	action: purchase	...
4	...	yes				...

DYNAMIC COLUMN MATERIALIZATION

userID		preferred	productName	price	column reservoir	
1	...	no	methylamine	12.50	tags: [industrial, barrel] color: blue action: view	...
2	...	yes	sudafed	52.50	tags: [home, medical] color: red action: view	...
3	...	yes		57.12	action: purchase	...
4	...	yes				...

DYNAMIC COLUMN MATERIALIZATION

userID		preferred	productName	price	action	column reservoir	
1	...	no	methylamine	12.50		tags: [industrial, barrel] color: blue action: view	...
2	...	yes	sudafed	52.50		tags: [home, medical] color: red action: view	...
3	...	yes		57.12		action: purchase	...
4	...	yes					...

DYNAMIC COLUMN MATERIALIZATION: EXAMPLE

userID		preferred	productName	price	action	column reservoir	
1	...	no	methylamine	12.50	view	tags: [industrial, barrel] color: blue action: view	...
2	...	yes	sudafed	52.50		tags: [home, medical] color: red action: view	...
3	...	yes		57.12		action: purchase	...
4	...	yes					...

DYNAMIC COLUMN MATERIALIZATION

userID		preferred	productName	price	action	column reservoir	
1	...	no	methylamine	12.50	view	tags: [industrial, barrel] color: blue	...
2	...	yes	sudafed	52.50	view	tags: [home, medical] color: red action: view	...
3	...	yes		57.12		action: purchase	...
4	...	yes					...

TEXT SEARCH



TEXT SEARCH

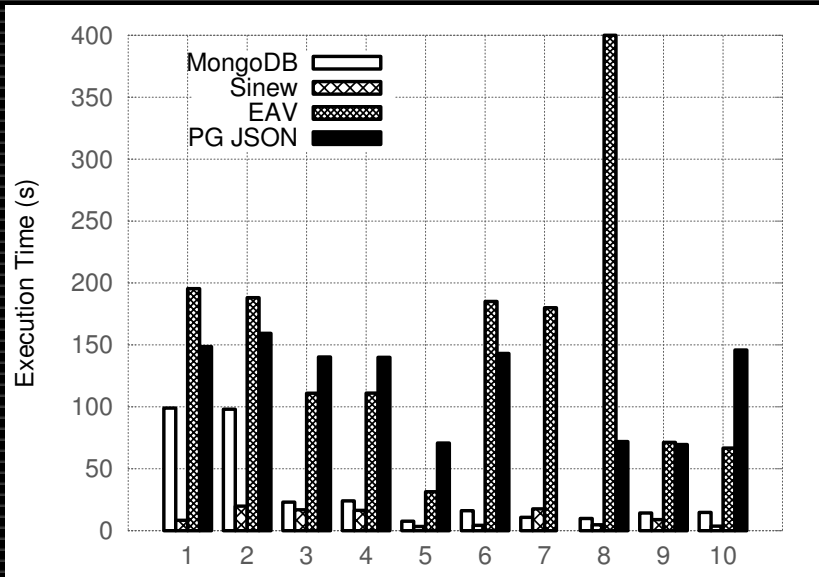
Add an external text index

- Space efficient
- Supports typed (e.g. numerical) primitives and operations
- UDF that performs text query and returns row IDs for matching records

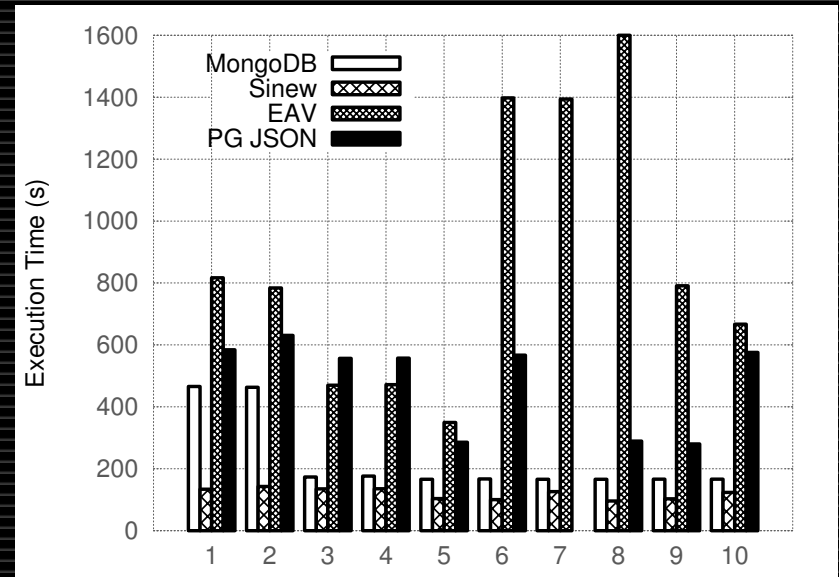
Two use cases:

1. Unstructured queries
2. References to sparse, virtual columns

BENCHMARKS

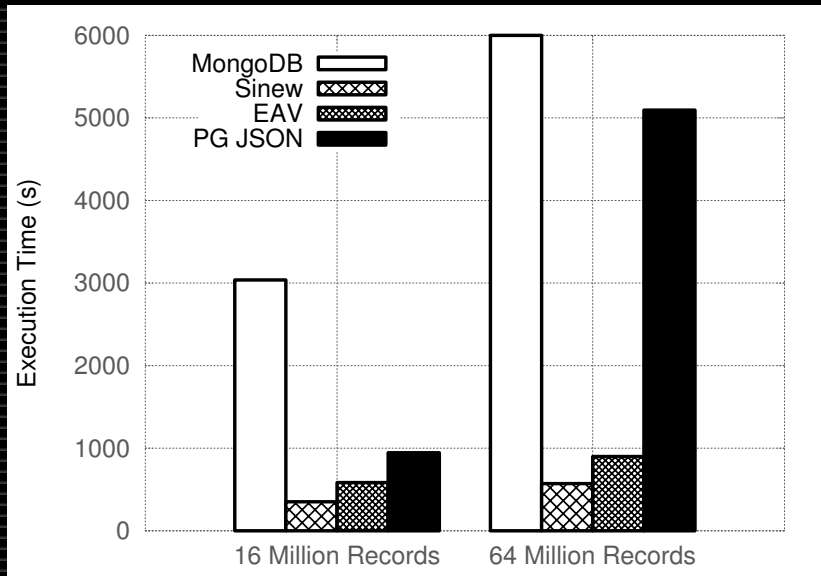


16 million records

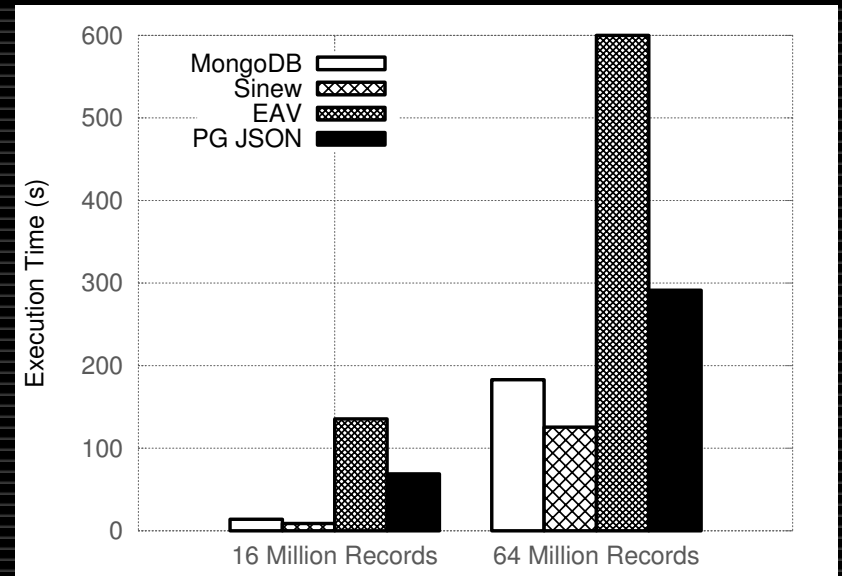


64 million records

BENCHMARKS: JOIN, UPDATE



Join (NoBench Q11)



Update

SUMMARY

SQL over relational, semi-structured, and unstructured data

Schema inferred at load and presented at query time

Extends an RDBMS without modifying RDBMS code

QUESTIONS?

Email: daniel.tahara@aya.yale.edu

Further Resources:

- Yale Database Group:
 - <http://db.cs.yale.edu/>
- My Website:
 - <http://danieltahara.com/>
- Daniel Abadi:
 - <http://www.cs.yale.edu/homes/dna/>