



TIMESCALE



Diana Hsieh

Head of Product
Timescale



TIMESCALE



TIMESCALE



Cockroach LABS



Diana Hsieh

Head of Product
Timescale



TIMESCALE



Sven Klemm

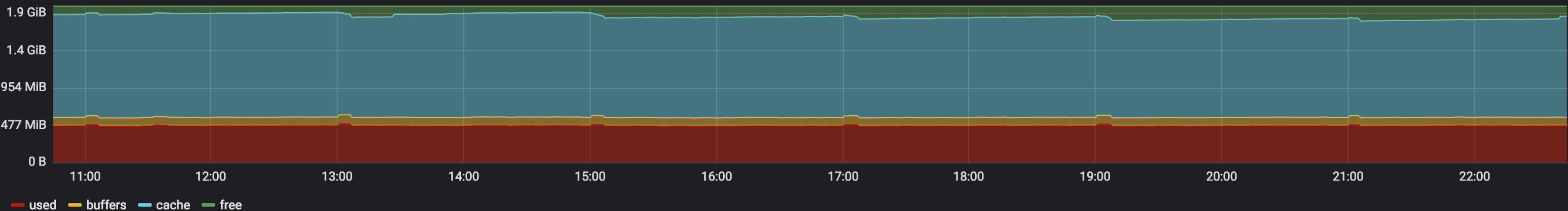
Software Engineering
Timescale



Host

grafana.timescaledb.rocks ▾

Memory



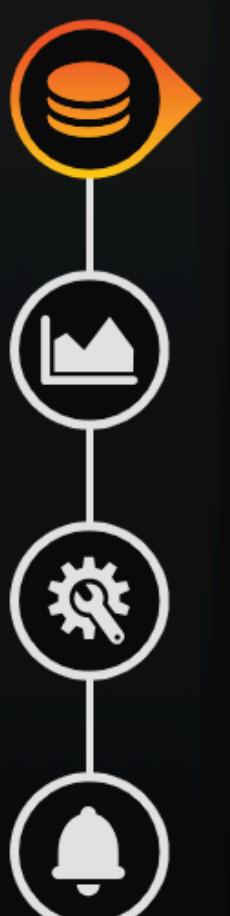
Queries to

default

Add Query

Query Inspector

?



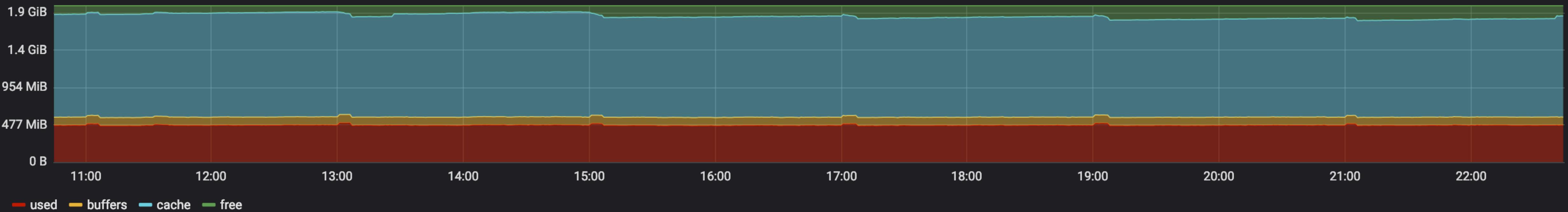
A

FROM		Time column	time	Metric column	i	none
SELECT	cpu	+				
	disk					
	diskio					
	docker					
	docker_container_blkio					
	docker_container_cpu					
WHERE	docker_container_mem					
	docker_container_net					
	docker_container_status					
GROUP BY	kernel					
Format as						

Show Help

Host grafana.timescaledb.rocks ▾

Memory



Queries to

default

Add Query

Query Inspector

?

A

FROM	mem	Time column	time	Metric column	none
------	-----	-------------	------	---------------	------

SELECT	Column: used	Alias: used	
--------	--------------	-------------	--

	Column: buffered	Alias: buffers	
--	------------------	----------------	--

	Column: cached	Alias: cache	
--	----------------	--------------	--

	Column: free	Alias: free	
--	--------------	-------------	--

WHERE	Expr: host = '\$host'	Macro: \${__time}	
-------	-----------------------	-------------------	--

GROUP BY	
----------	--

Format as

Time series

Edit SQL

Show Help ▾

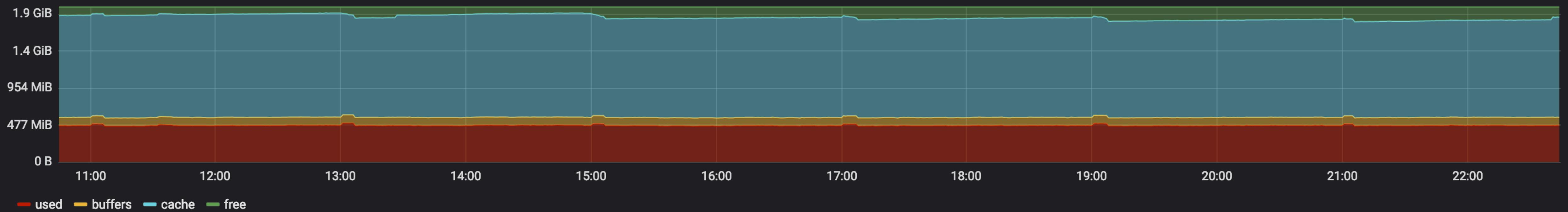
- Aggregate Functions
 - Average
 - Ordered-Set Aggregate Functions
 - Count
 - Maximum
 - Minimum
 - Sum
 - Standard deviation
 - Variance
 - First
 - Window Functions
 - Alias
 - Column



Host

grafana.timescaledb.rocks ▾

Memory



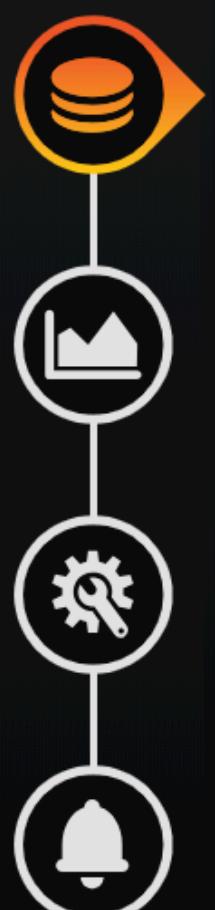
Queries to

default

Add Query

Query Inspector

?



▼ A



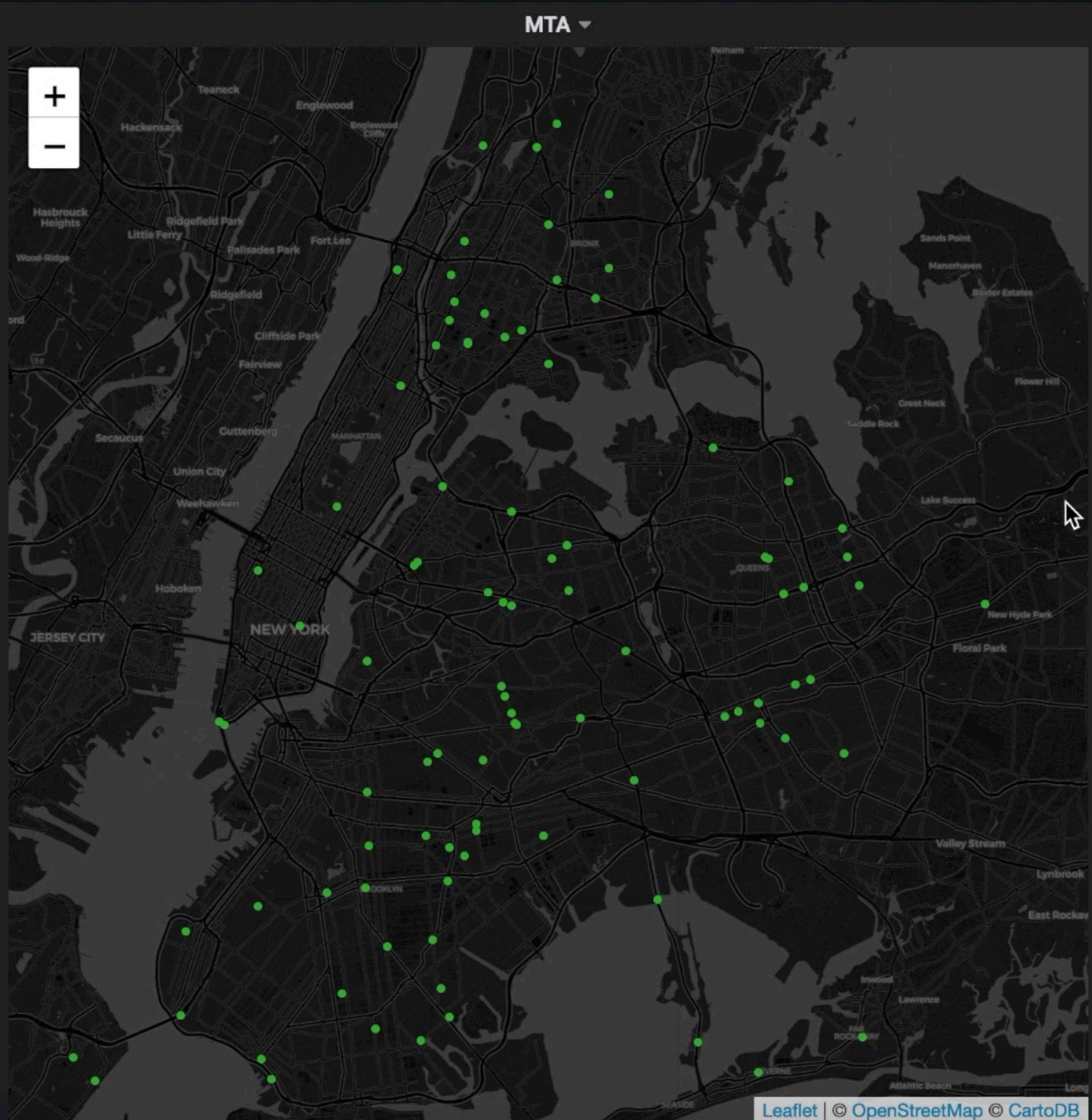
```
SELECT
  time AS "time",
  used AS "used",
  buffered AS "buffers",
  cached AS "cache",
  free AS "free"
FROM mem
WHERE
  host = '$host' AND
  $__timeFilter(time)
ORDER BY 1
```

Format as

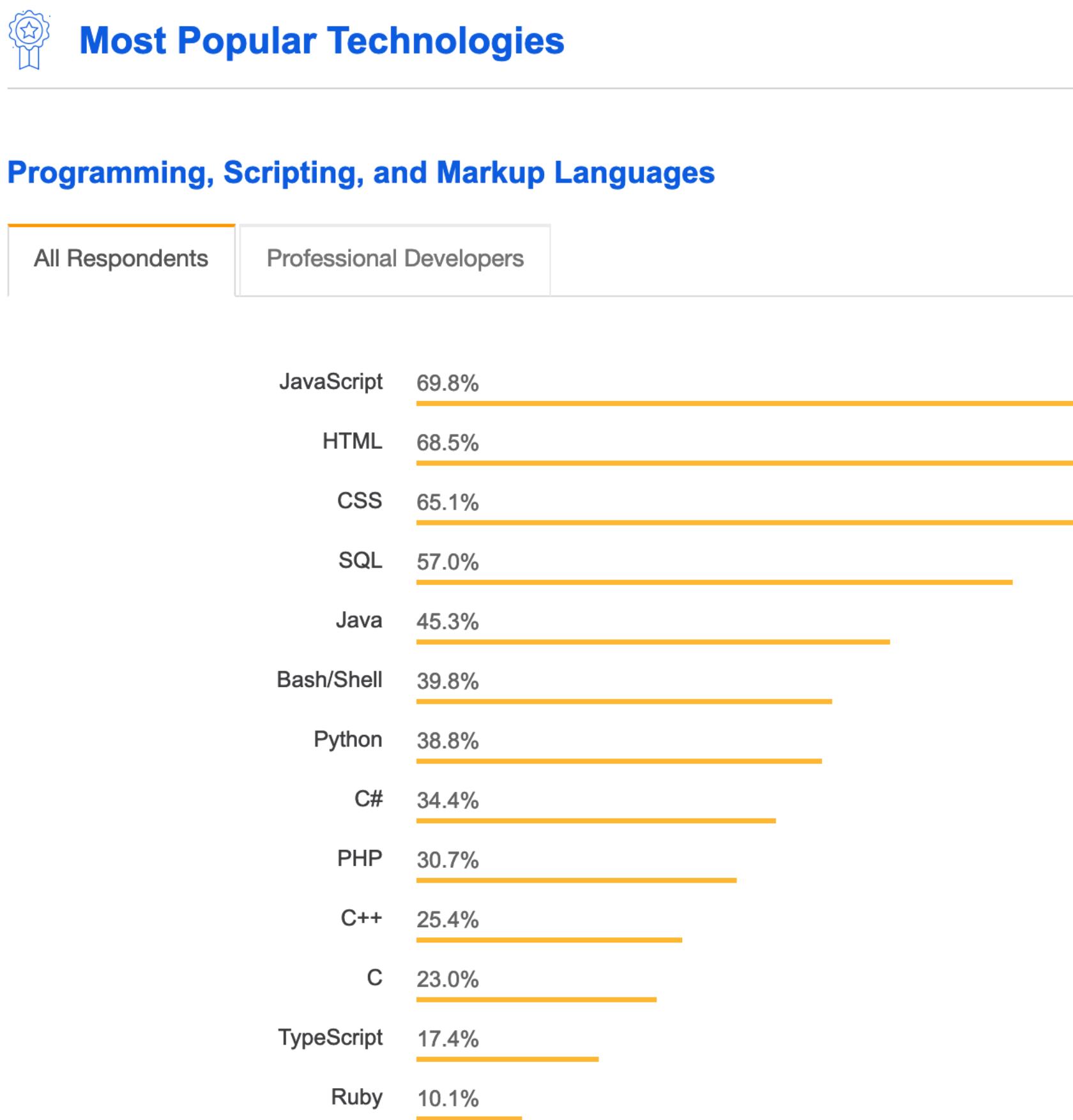
Time series ▾

Query Builder

Show Help ▾



SQL is a well-known querying language



 Q SQL 

People ▾ Connections ▾ Locations ▾ Current companies ▾ All Filters

 Q NoSQL 

Showing 9,518,302 results

 **sql advance** • 3rd Recruiter at Debian Systems United States

 **SQL Training** SQL Trainer at SQL Training Greater New York City Area

 **Jackson Andrew** • 3rd SQL Server DBA Greater New York City Area Summary: I am SQL DBA problems related...

 **Aitzaz Ashraf** • 3rd .NET/SQL Developer at AWS Greater New York City Area Past: SQL Server Developer

 **Gary Russo** • 2nd Multi-model NoSQL Architect and Developer at TD Ameritrade Greater New York City Area Past: NoSQL Architect and Developer at Broadridge 4 shared connections

 **Jason Laschewer** • 2nd Global Business Development Manager - NoSQL services at AWS Greater New York City Area Current: Global Business Development Manager - NoSQL at Amazon Web Services 5 shared connections

 **Nikunj Jain** • 2nd Big data, NoSQL Data Engineer Greater New York City Area Skill: NoSQL



SQL is a well-known querying language



 **SQL**

Home My Network Jobs Messaging

People Connections Locations Current companies All Filters

Get Data.  **NoSQL**

Showing 9,518,302 results

 **sql advance** • 3rd Recruiter at Debian Systems United States

 **SQL Training** SQL Trainer at SQL Training Greater New York City Area

 **Jackson Andrew** • 3rd SQL Server DBA Greater New York City Area Summary: I am SQL DBA problems related...

 **Aitzaz Ashraf** • 3rd .NET/SQL Developer at AWS Greater New York City Area Past: SQL Server Developer

 **Gary Russo** • 2nd Multi-model NoSQL Architect and Developer at TD Ameritrade Greater New York City Area Past: NoSQL Architect and Developer at Broadridge 4 shared connections

 **Jason Laschewer** • 2nd Global Business Development Manager - NoSQL services at AWS Greater New York City Area Current: Global Business Development Manager - NoSQL at Amazon Web Services 5 shared connections

 **Nikunj Jain** • 2nd Big data, NoSQL Data Engineer Greater New York City Area Skill: NoSQL

Fast, Reliable Internet. - Get 75 Mbps Internet for just \$49.95 per month when you add 2

Home My Network Jobs Messaging

People Connections Locations Current companies All Filters

Showing 186,638 results



SQL is a well-known querying language



 Q SQL  Q NoSQL

Home My Network Jobs Messaging

People Connections Locations Current companies All Filters

Showing 9,518,302 results

 **sql advance** • 3rd Recruiter at Debian Systems United States

 **SQL Training** SQL Trainer at SQL Training Greater New York City Area

 **Jackson Andrew** • 3rd SQL Server DBA Greater New York City Area Summary: I am SQL DBA problems related...

 **Aitzaz Ashraf** • 3rd .NET/SQL Developer at AWS Greater New York City Area Past: SQL Server Developer

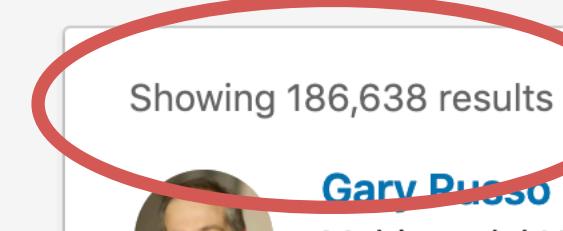
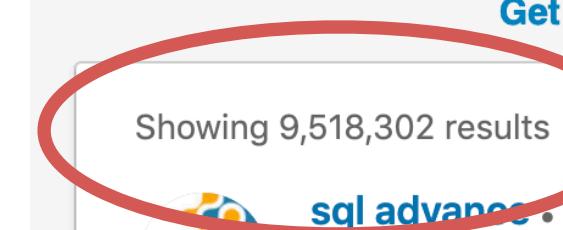
 **Gary Russo** • 2nd Multi-model NoSQL Architect and Developer at TD Ameritrade Greater New York City Area Past: NoSQL Architect and Developer at Broadridge 4 shared connections

 **Jason Laschewer** • 2nd Global Business Development Manager - NoSQL services at AWS Greater New York City Area Current: Global Business Development Manager - NoSQL at Amazon Web Services 5 shared connections

 **Nikunj Jain** • 2nd Big data, NoSQL Data Engineer Greater New York City Area Skill: NoSQL

Fast, Reliable Internet. - Get 75 Mbps Internet for just \$49.95 per month when you add 2

Connect Connect Connect



SQL is a well-known querying language

Most Popular Technologies

Programming, Scripting, and Markup Languages

All Respondents

Professional Developers

JavaScript 69.8%

HTML 68.5%

CSS 65.1%

SQL 57.0%

Java 45.5%

Bash/Shell 39.8%

Python 38.8%

C# 34.4%

PHP 30.9%

C++ 25.4%

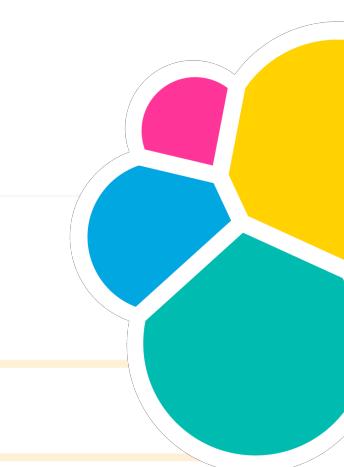
C 23.0%

TypeScript 17.4%

Ruby 10.1%



kafka



elastic

Showing 9,518,302 results

Get Data.

Showing 186,638 results

Fast, Reliable Internet. - Get 75 Mbps Internet for just \$49.95 per month when you add 2

Gary Pusso • 2nd
Multi-model NoSQL Architect and Developer at TD Ameritrade
Greater New York City Area

Jackson Andrew • 1st
SQL Server DBA
Greater New York City Area
Summary: I am SQL DBA problems related...

Aitzaz Ashraf • 3rd
.NET/SQL Developer at AWS
Greater New York City Area
Past: SQL Server Develop...

Jason Laschewer • 2nd
Global Business Development Manager - NoSQL services at AWS
Greater New York City Area
Current: Global Business Development Manager - NoSQL at Amazon Web Services

Nikunj Jain • 2nd
Big data, NoSQL Data Engineer
Greater New York City Area
Skill: NoSQL

Connect

Connect

Connect

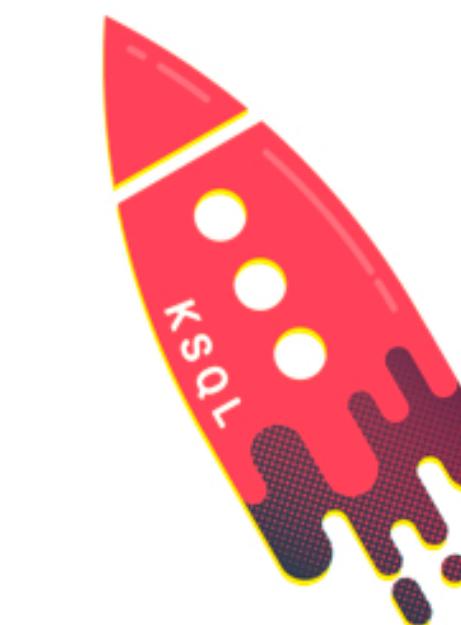
The image shows two LinkedIn search results pages. The top page is for 'SQL', showing 9,518,302 results. The bottom page is for 'NoSQL', showing 186,638 results. Both pages include filters for People, Connections, Locations, and Current companies, along with Home, My Network, Jobs, and Messaging tabs. A red oval highlights the 'SQL' search bar and results, while another red oval highlights the 'NoSQL' search bar and results. The elastic logo is overlaid on the center of the LinkedIn interface.



SQL is a well-known querying language



kafka

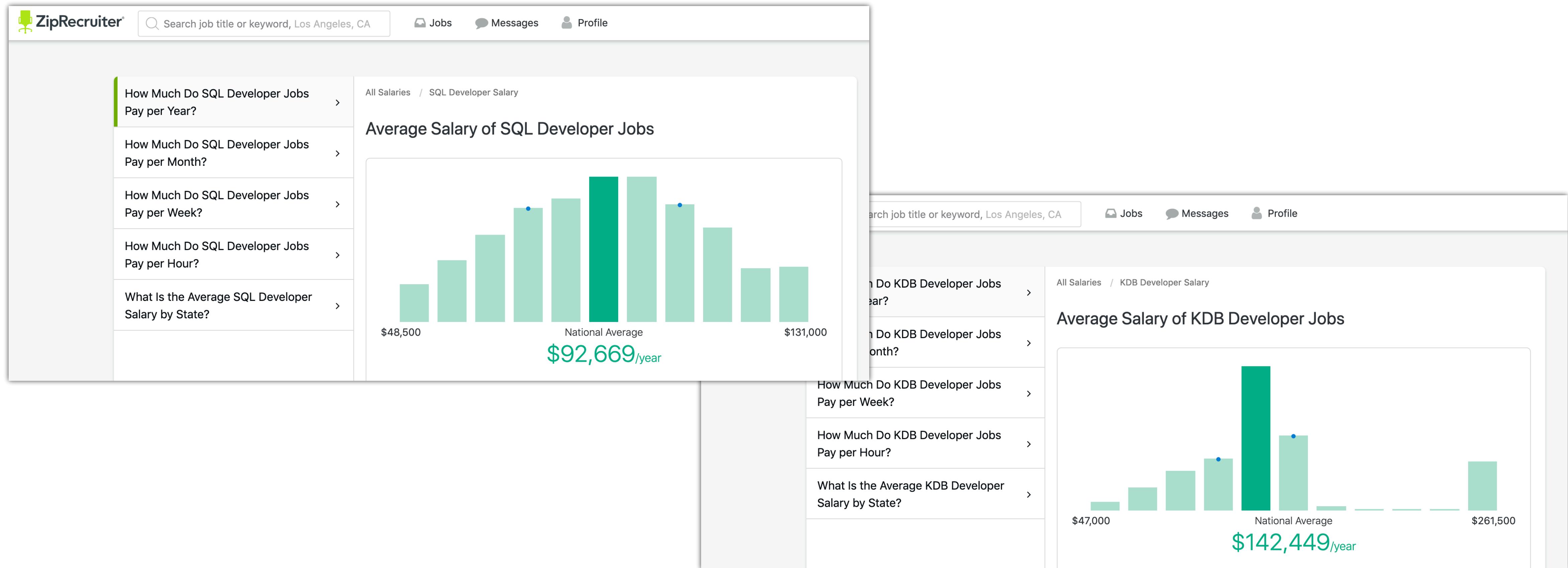


elastic

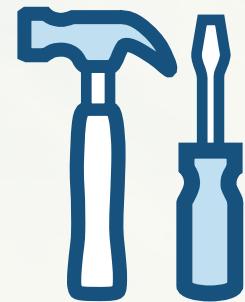
Two screenshots of the LinkedIn search interface. The top screenshot shows search results for 'SQL', with a red oval highlighting the search bar and the results count 'Showing 9,518,302 results'. The bottom screenshot shows search results for 'NoSQL', with a red oval highlighting the search bar and the results count 'Showing 186,638 results'. Both screenshots show profiles of professionals related to their respective fields.



SQL is a well-known querying language



Data must be accessible to provide value



New database, new integrations

Reinforced data silos due to lack of inter-operability between systems



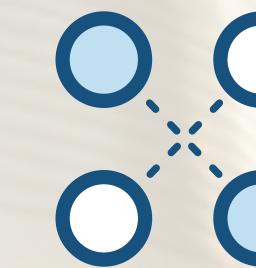
New database, new language

Increased training costs and talent gap to query disparate silos



Brittle architecture

Huge operational cost to manage multiple database solutions



De-centralized infrastructure

Hard to analyze data that spans across multiple silos

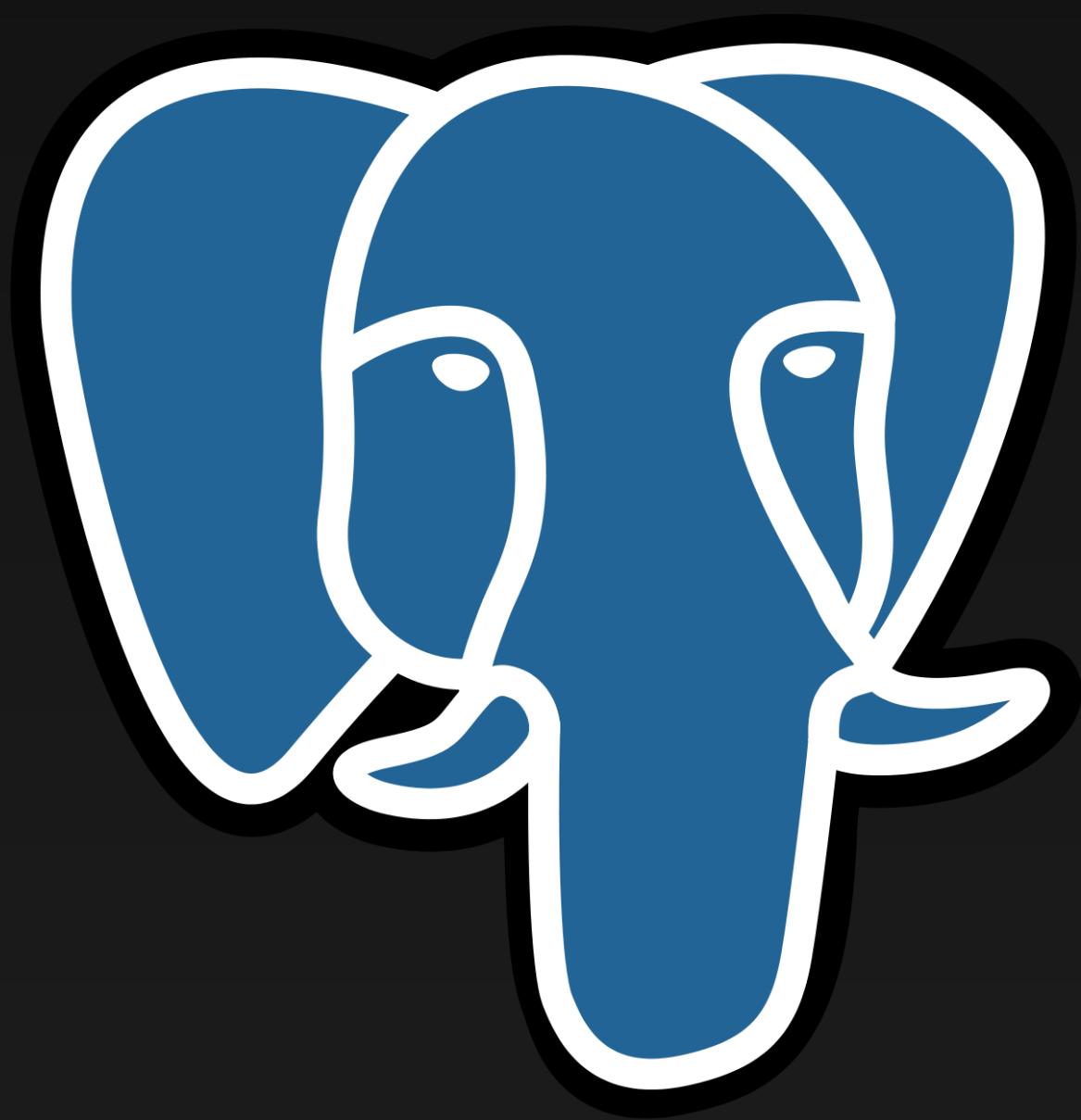


*“Initially my colleagues were skeptical when I suggested storing metrics for our **120 petabyte data center** in a relational database, but after replacing the prior NoSQL database with TimescaleDB **we couldn’t be more happy** with the performance.*

*Because TimescaleDB is an extension of PostgreSQL, we’re starting to expand the scope of the metrics storage to **power executive dashboards** and advanced analytical functions that our prior NoSQL solution couldn’t support.”*

- Chris Holcombe, Production Dev Engineer





But.. SQL doesn't scale....



Traditional SQL databases were built for transactional workloads

	State snapshot	Time-series
Financial Data	Account balance	Transaction history
DevOps	Capacity utilized	Usage over time
IoT	# of online devices	Devices over time
SaaS Applications	Total users	Logins over time

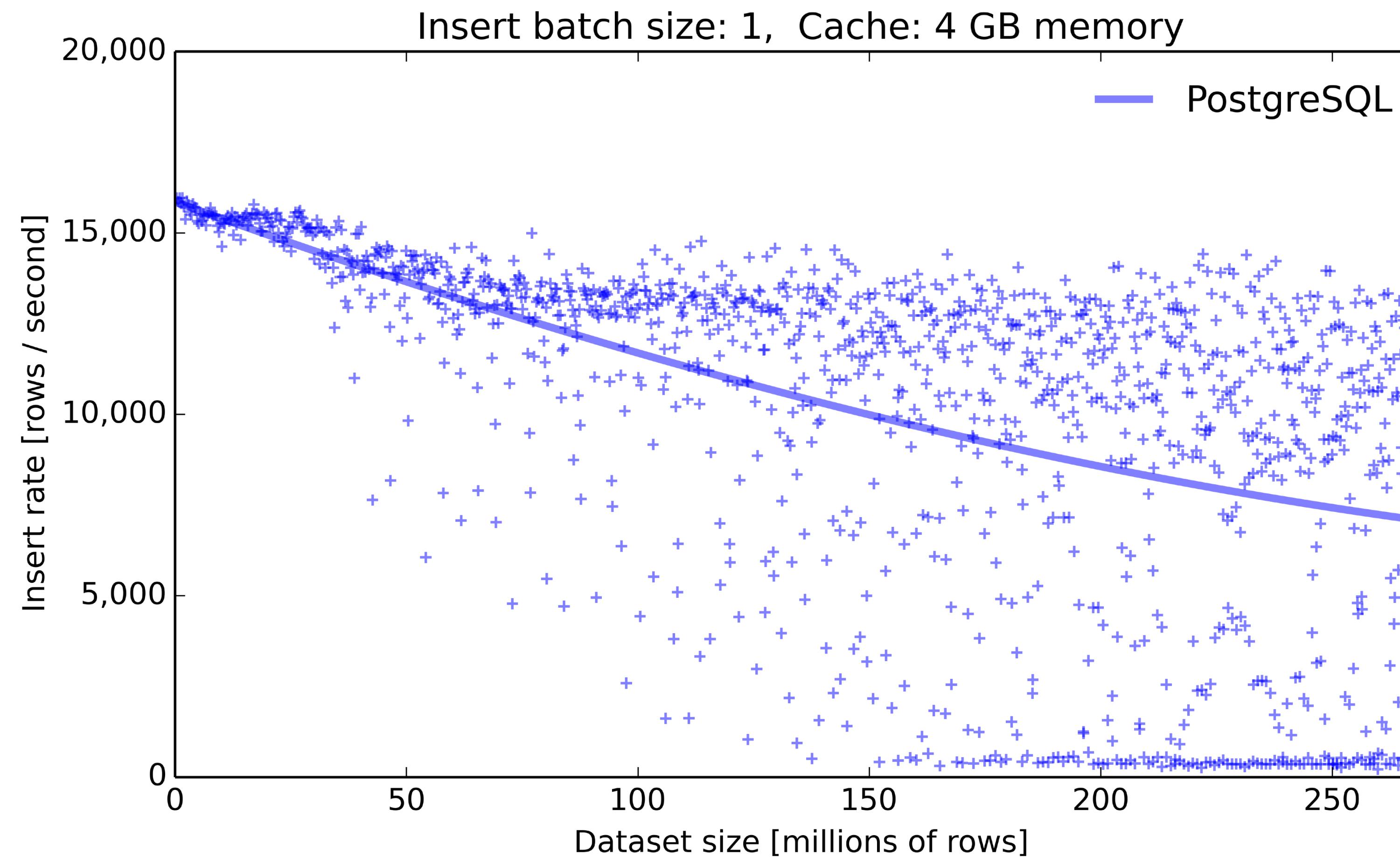


Time-series data requires a different kind of database

- Fast ingest of weakly ordered data
- Large storage requirements
- Time-series specific analytics
- Time-series specific data management



PostgreSQL is hard to scale



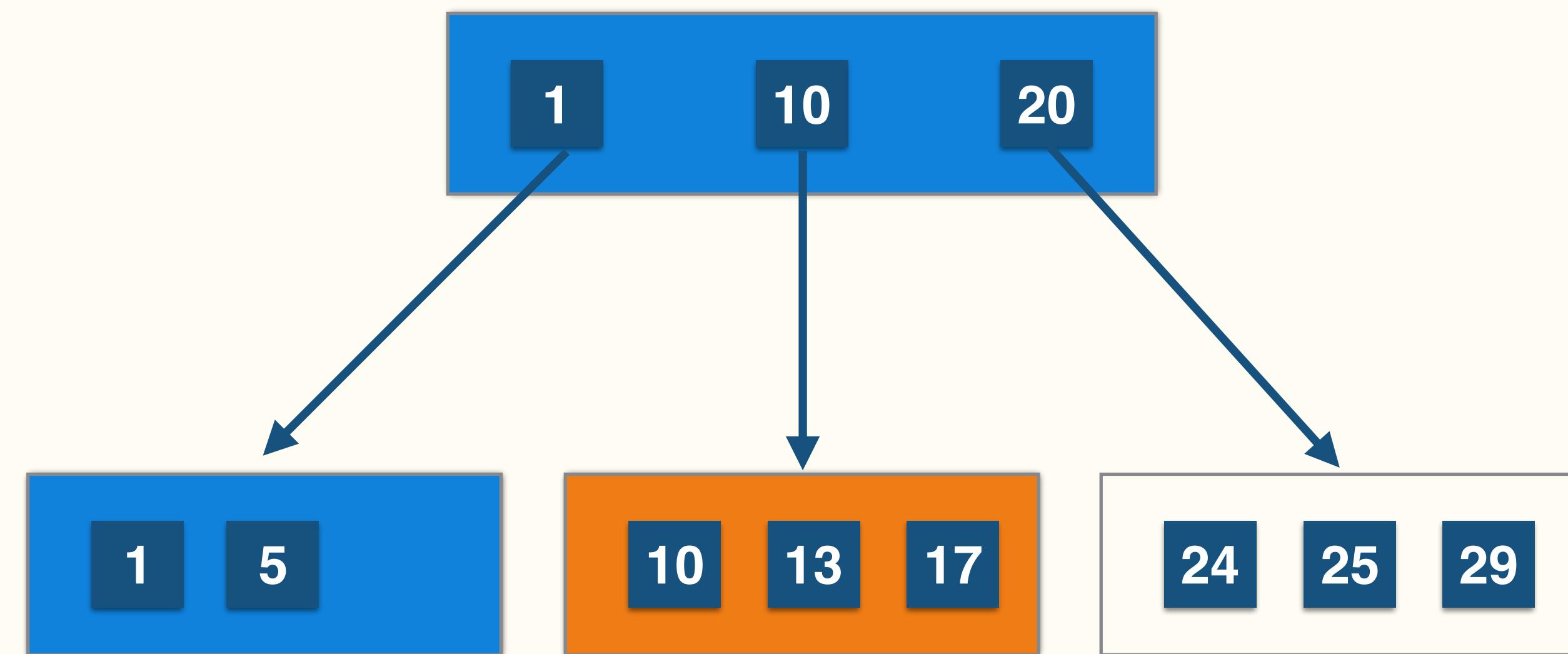
Postgres 9.6.2 on Azure standard DS4 v2 (8 cores), SSD (premium LRS storage)

Each row has 12 columns (1 timestamp, indexed 1 host ID, 10 metrics)



Indexes need to be updated on inserts

Insert batch: 8



IN MEMORY

WRITE TO DISK

Making SQL for time-series possible

STORAGE LAYER



Making SQL for time-series possible

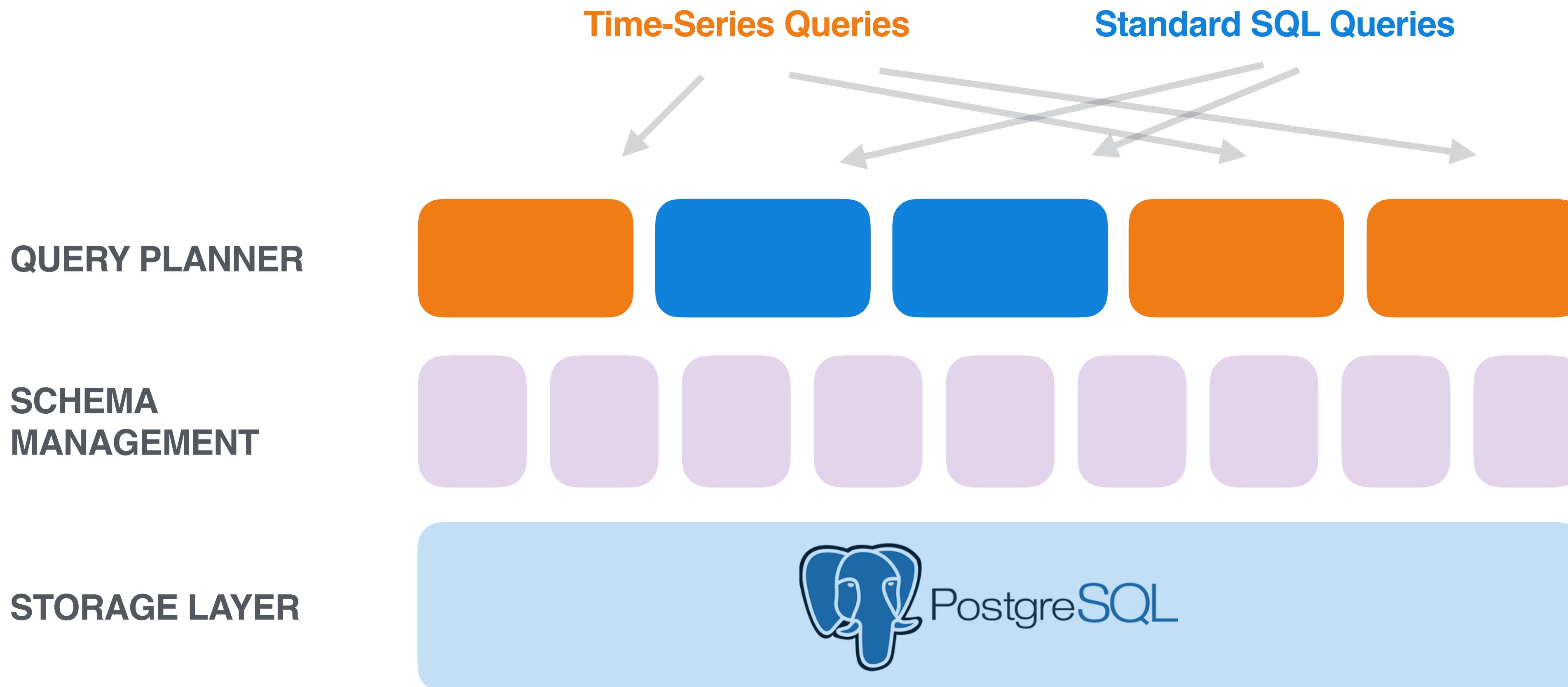
**SCHEMA
MANAGEMENT**



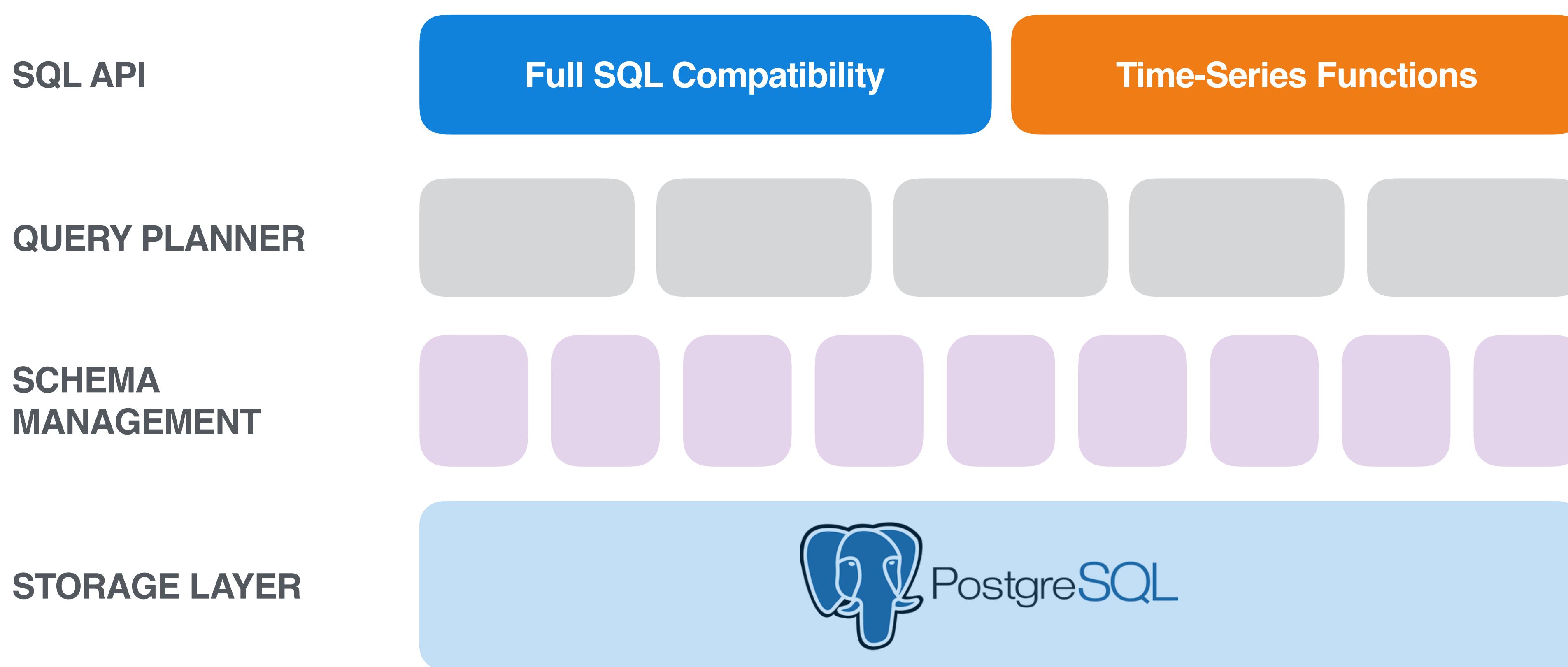
STORAGE LAYER



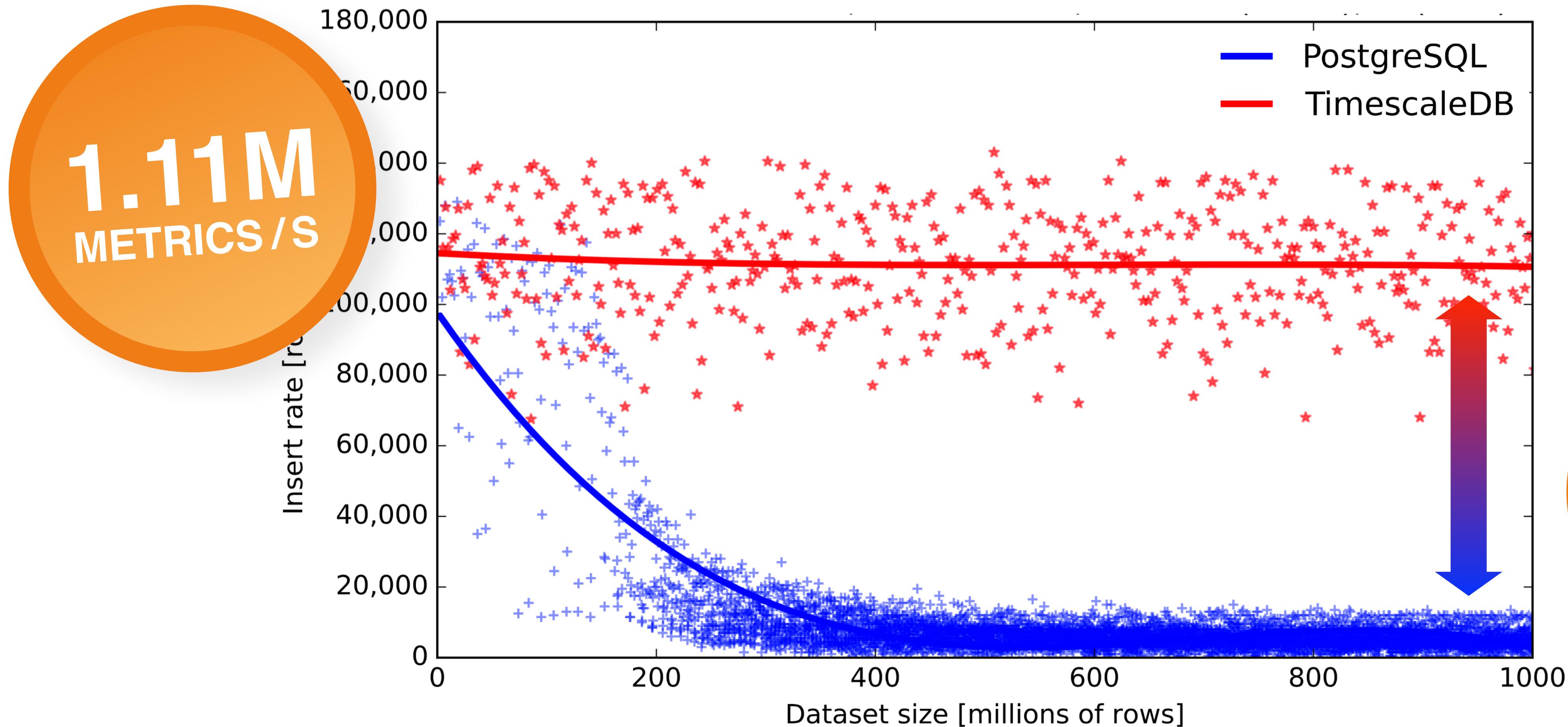
Making SQL for time-series possible



Making SQL for time-series possible



TimescaleDB vs PostgreSQL



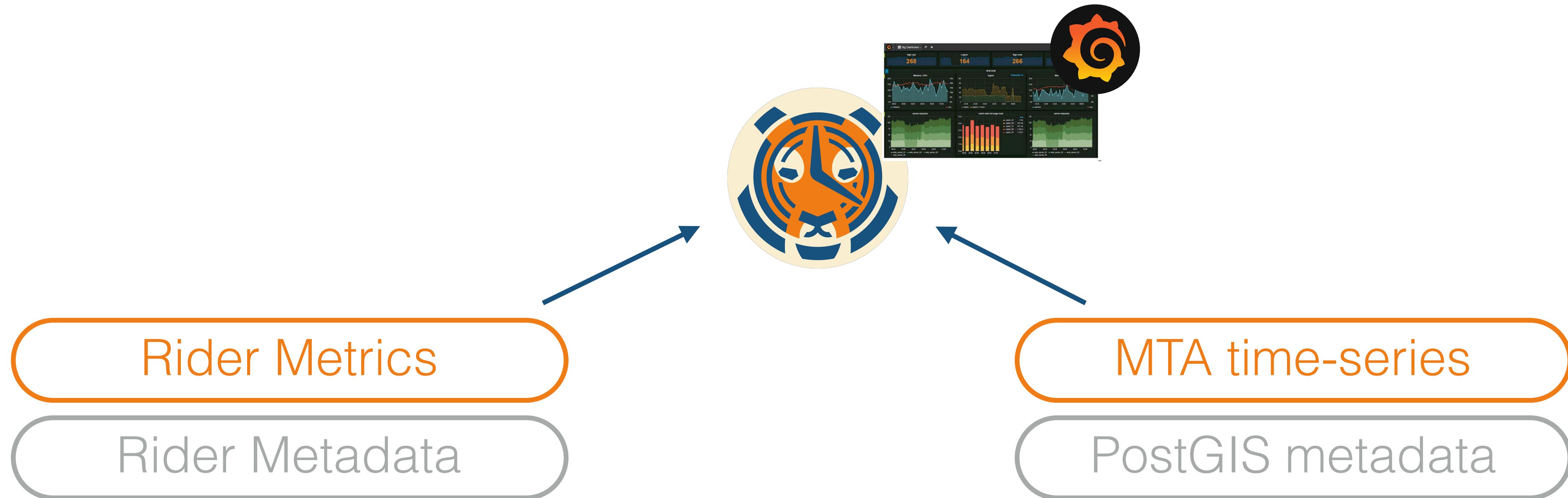
TimescaleDB 0.5, Postgres 9.6.2, Azure standard DS4 v2 (8 cores), SSD (LRS storage)
Each row has 12 columns (1 timestamp, indexed 1 host ID, 10 metrics)



How would I go about
modeling this?



Our MTA demo



Riders Schema

Rider Metadata

```
mta=> \d riders
      Table "public.riders"
 Column | Type   | Collation | Nullable | Default
-----+-----+-----+-----+
rider_id | text   |           | not null |
ny_resident | boolean |           |           |
Indexes:
"riders_pkey" PRIMARY KEY, btree (rider_id)
```

Rider Metrics

```
Table "public.riders_metrics"
Column | Type           | Collation | Nullable | Default
-----+-----+-----+-----+
time   | timestamp with time zone |           | not null |
route_id | text           |           | not null |
rider_id | integer        |           | not null |
```



MTA Schema

PostGIS metadata

```
mta=> \d route_geofences
      Table "public.route_geofences"
 Column |          Type          | Collation | Nullable | Default
-----+----------------+-----+-----+-----+
route_id | character varying(254) |          |          |
geom     | geometry           |          |          |
Indexes:
"route_geofences_geom_idx" gist (geom)
"route_geofences_route_id_idx" btree (route_id)
```

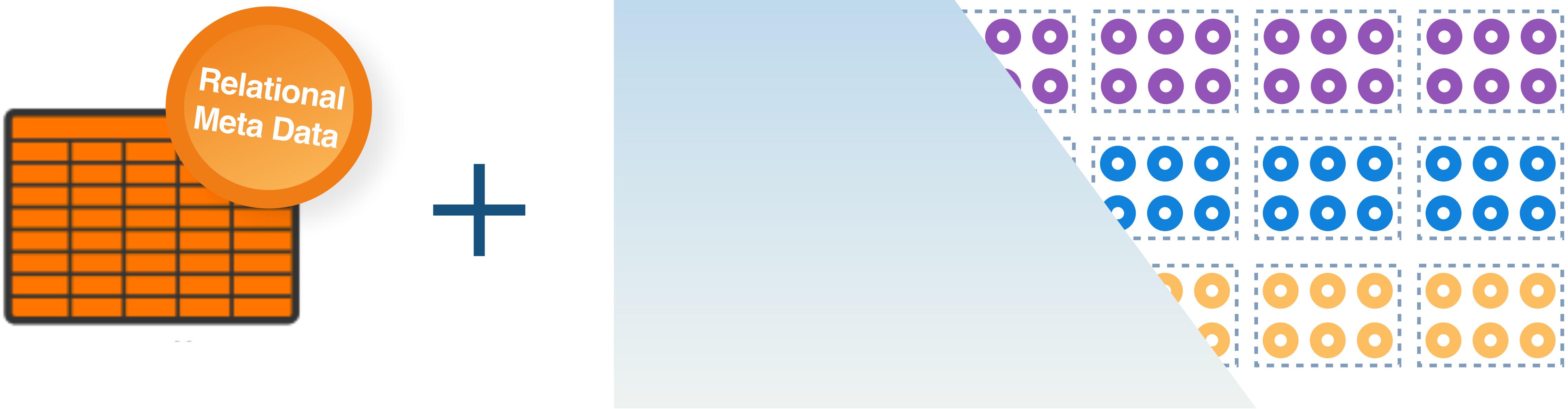
Table "public.mta"					
Column	Type		Collation	Nullable	Default
vid	text				
time	timestamp with time zone			not null	
route_id	text				
bearing	numeric				
geom	geometry(Point,4326)				
Indexes:					
"idx_mta_geom" gist (geom)					
"idx_mta_route_id" btree (route_id)					
"mta_time_idx" btree ("time" DESC)					
Triggers:					
ts_insert_blocker BEFORE INSERT ON mta FOR EACH ROW EXECUTE PROCEDURE _timescaledb_internal.insert_blocker()					
Number of child tables: 31 (Use \d+ to list them.)					

MTA time-series



Separate meta data from time series

Speed up queries and inserts by storing values that describe time series metrics in relational tables

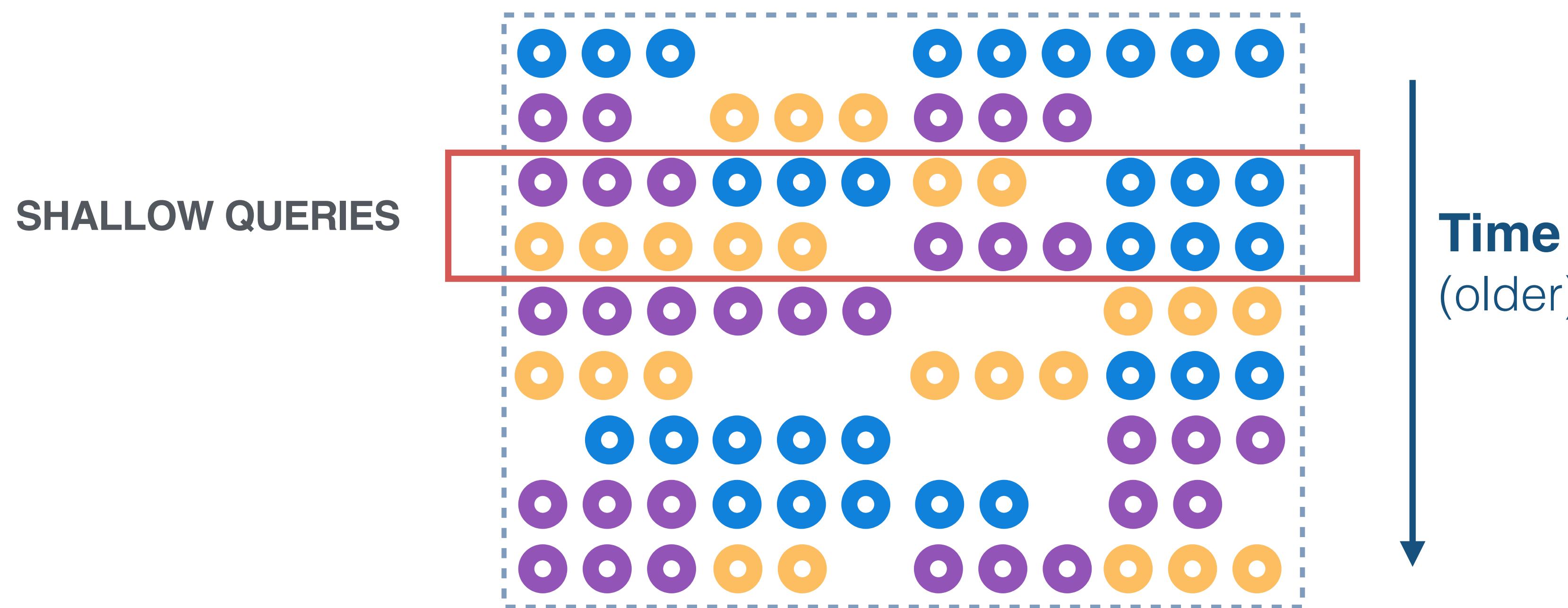


Store metrics more efficiently with metadata tables

```
webinar=# select * from hardware_metadata limit 5;
server_id | location | start_date | description
-----+-----+-----+-----+
 1 | {DE} | 2015-01-22 00:00:00+00 | server1: some random and long text about this server that no one needs, except when they need it
 2 | {FR} | 2010-08-08 00:00:00+00 | server2: some random and long text about this server that no one needs, except when they need it
 3 | {CN} | 2012-05-14 00:00:00+00 | server3: some random and long text about this server that no one needs, except when they need it
 4 | {CN} | 2010-10-14 00:00:00+00 | server4: some random and long text about this server that no one needs, except when they need it
 5 | {FR} | 2016-10-15 00:00:00+00 | server5: some random and long text about this server that no one needs, except when they need it
(5 rows)
```

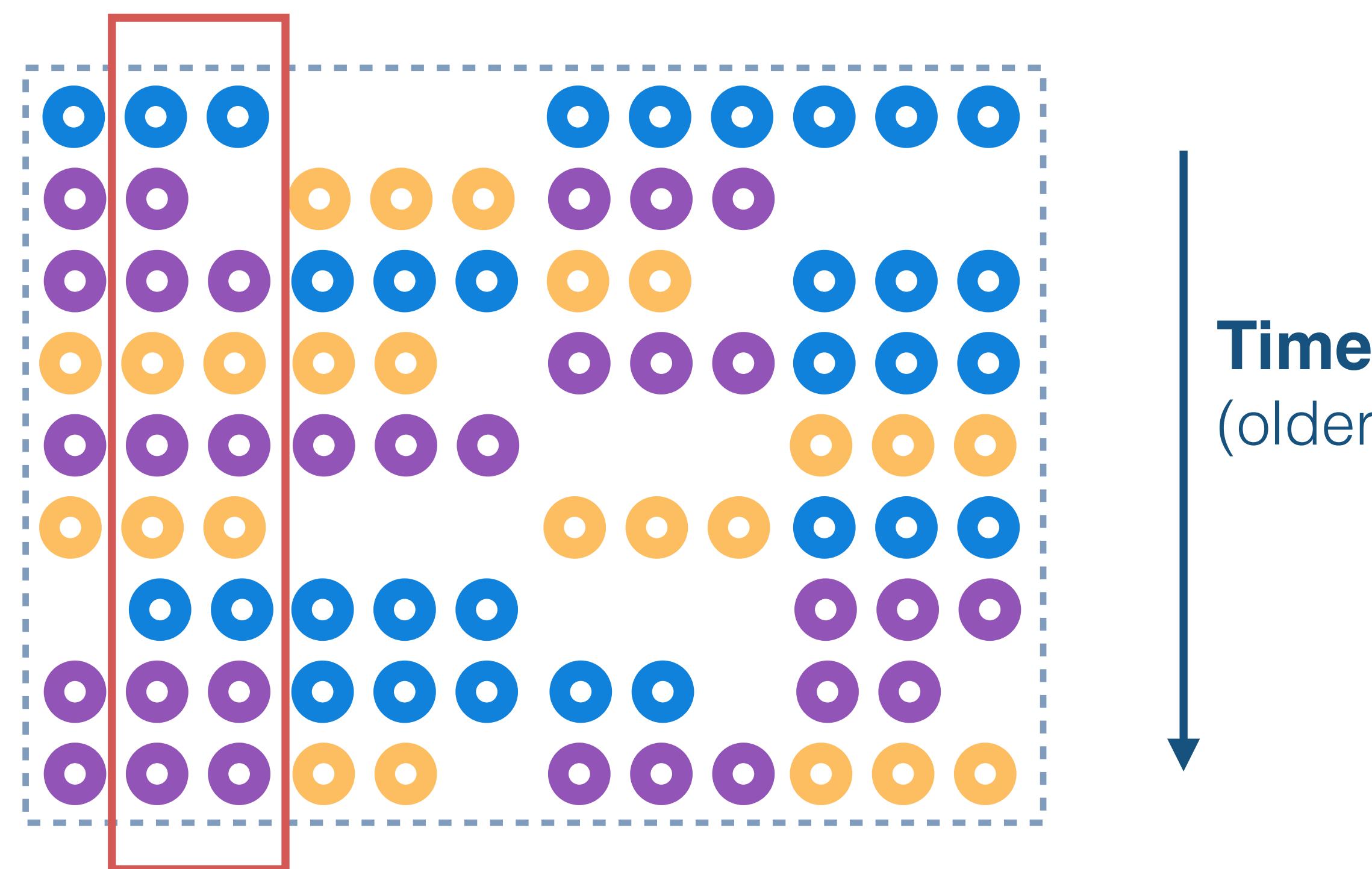


Optimize for your query patterns

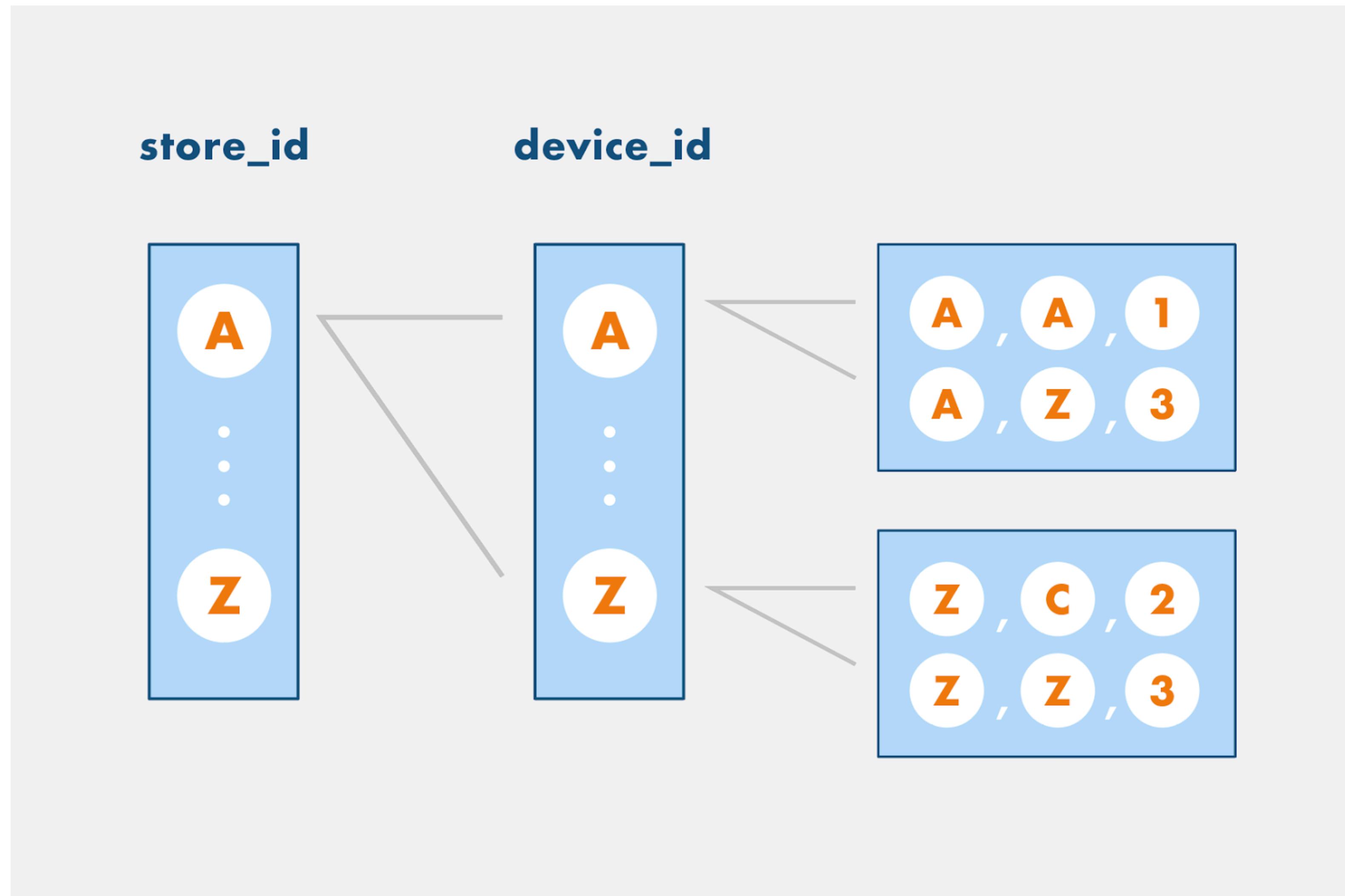


Optimize for your query patterns

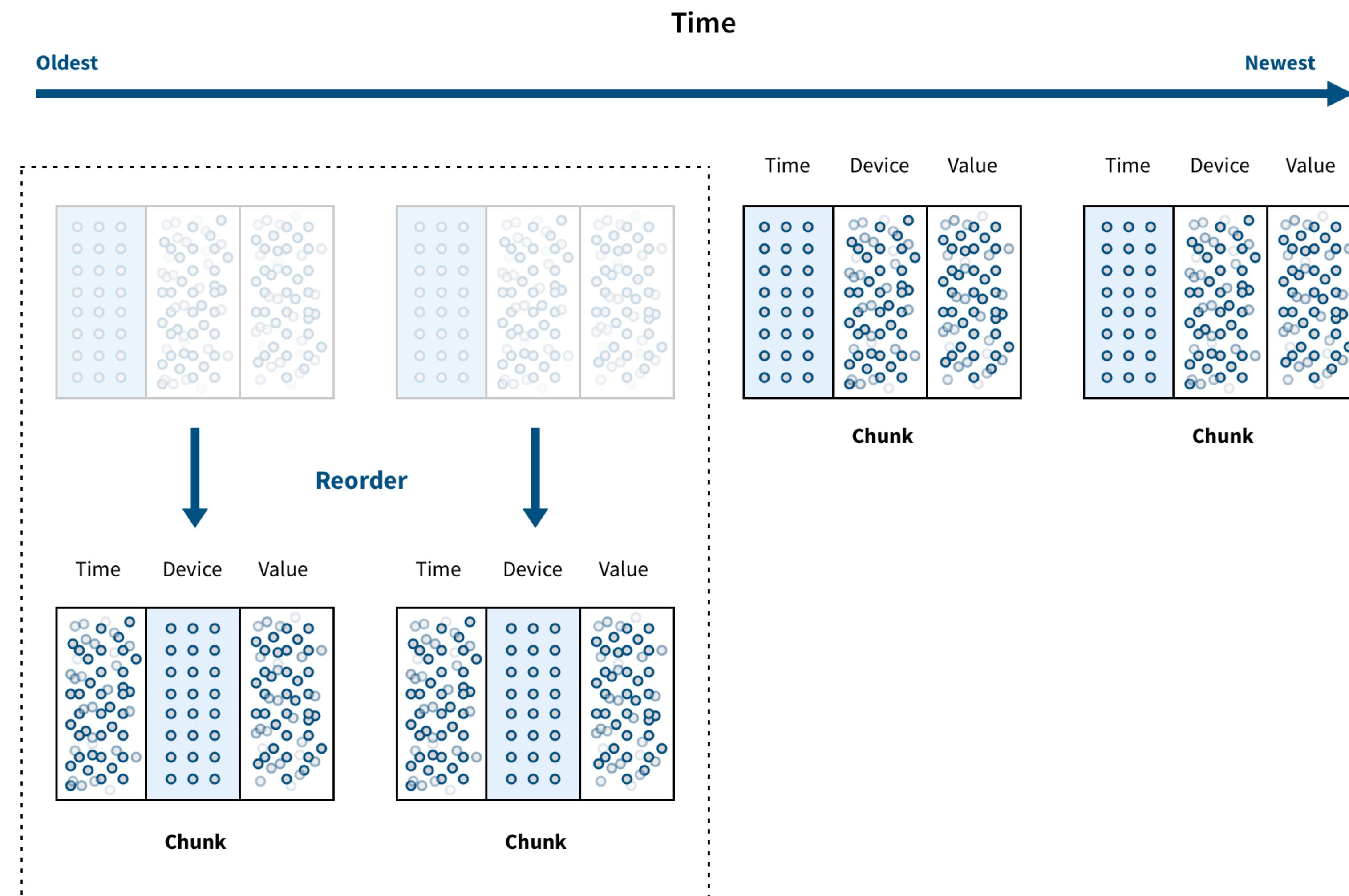
DEEP QUERIES



Leverage indexes...

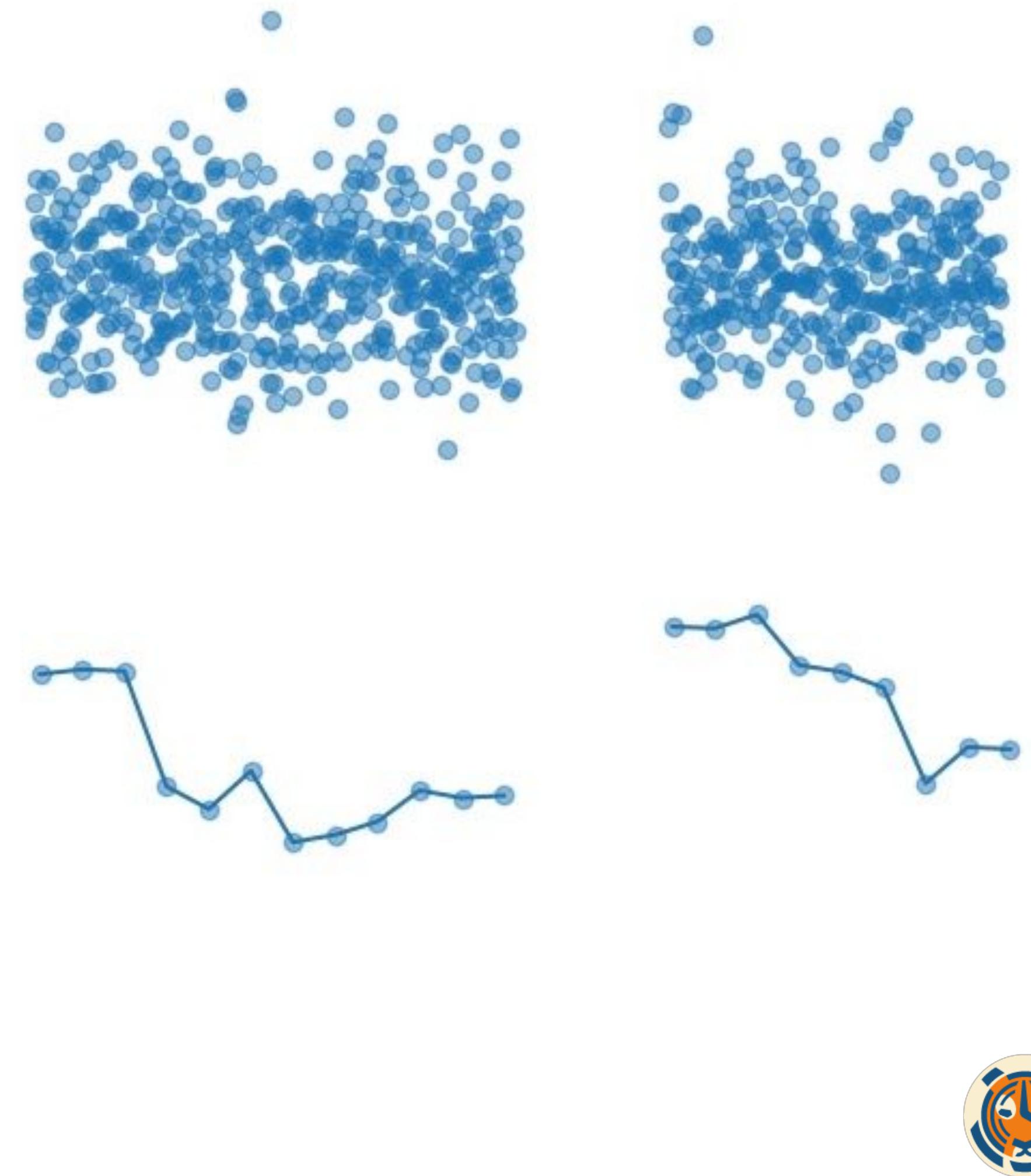


...or optimize how data is written on disk



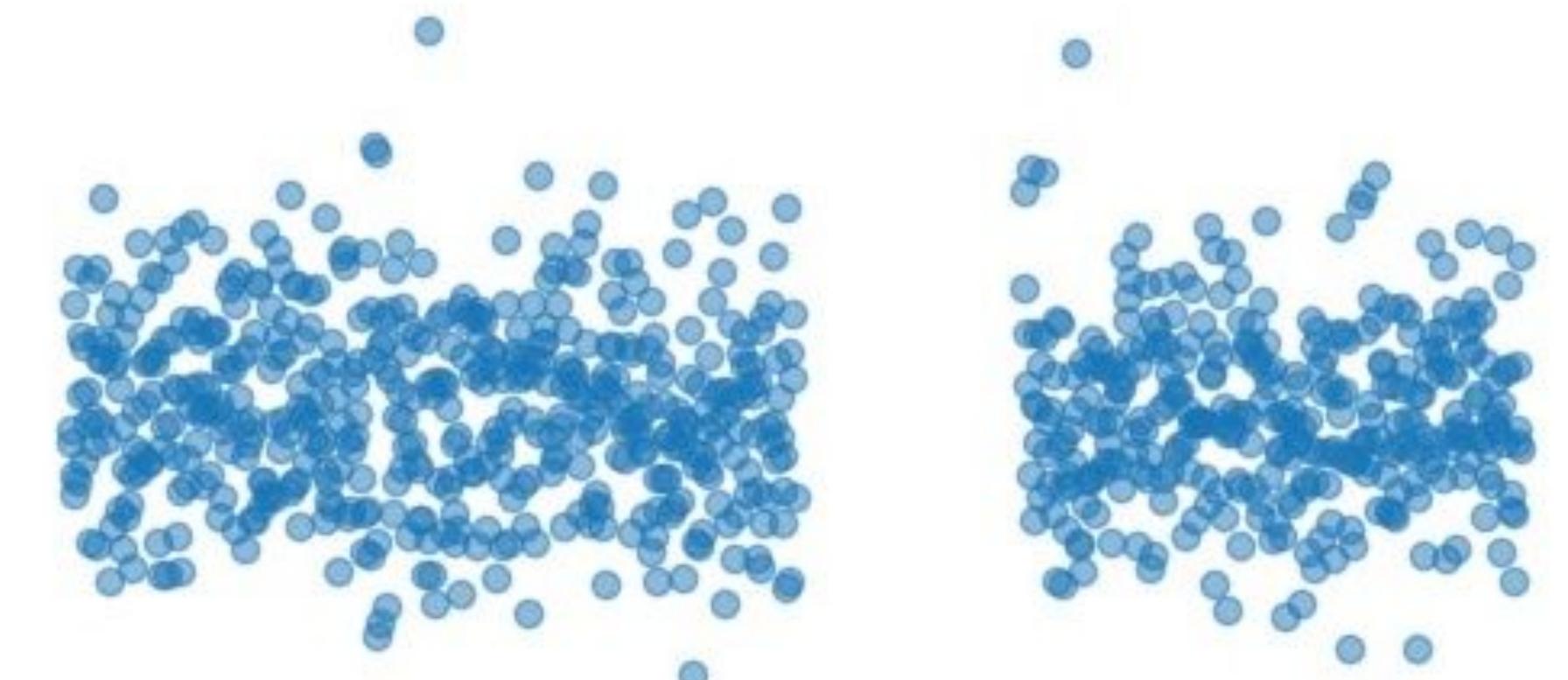
Leverage TimescaleDB functions

```
SELECT  
  time_bucket('1 minute', time) as minute,  
  avg(value)  
FROM observations  
GROUP BY minute  
ORDER BY minute;
```



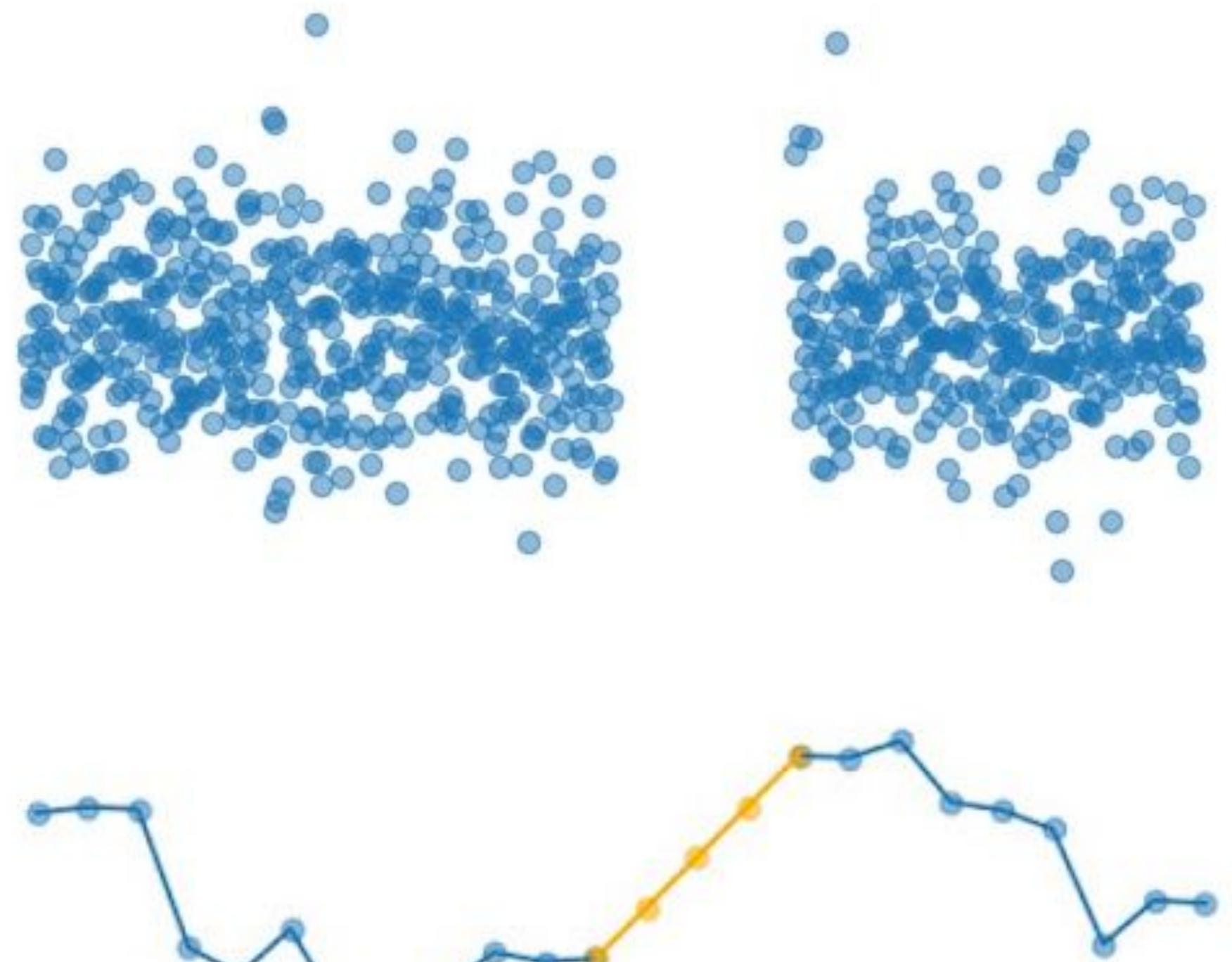
Leverage TimescaleDB functions

```
SELECT
  time_bucket_gapfill(
    '1 minute', time,
    start => '2019-01-21 9:00',
    finish => '2019-01-21 17:00') as minute
  locf(avg(value))
FROM observations
GROUP BY minute;
```



Leverage TimescaleDB functions

```
SELECT
  time_bucket_gapfill(
    '1 minute', time,
    start => '2019-01-21 9:00',
    finish => '2019-01-21 17:00') as minute
  interpolate(avg(value))
FROM observations
GROUP BY minute;
```





Open Source (Apache 2.0)

- github.com/timescale/timescaledb



Join the Community

- slack.timescale.com