SQREAM

HOW TO ELIMINATE COMPLEX SQL QUERY TRAFFIC JAMS

David Leichner, CMO
Arnon Shimoni, Product Manager

SQREAM

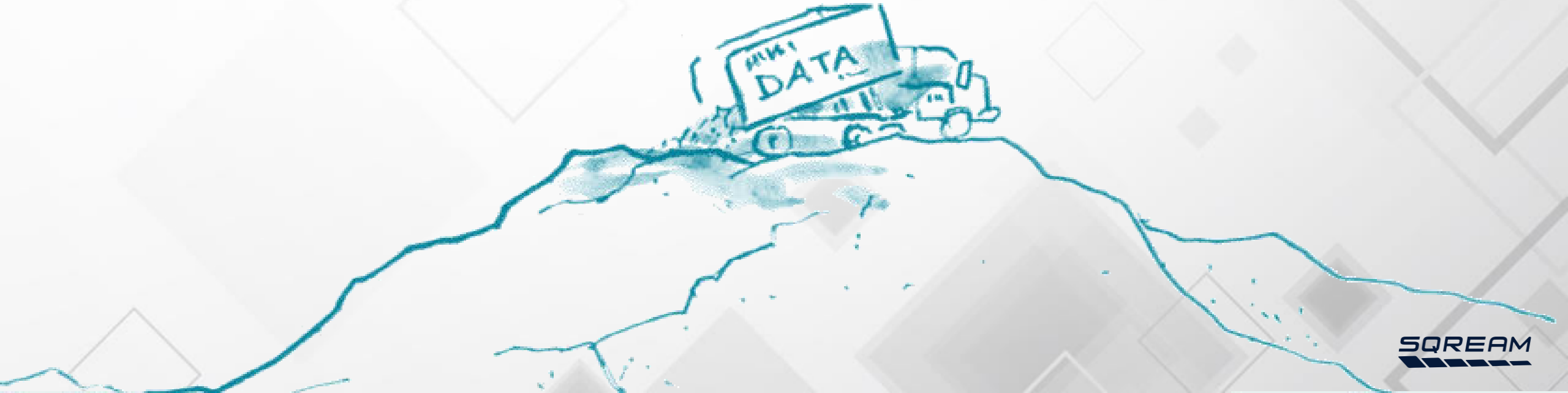

B+ Plus
Exclusive Sarees & Dress Materials
B+ Plus
Exclusive Sarees & Dress Materials

WHAT WE'LL COVER

- Problems with **lots of data**
- Problems with **data prep**
- Common causes of **bottlenecks**
- **Tips** for improving SQL performance
- The **elephant** in the room
- Q&A

SQREAM

Unfortunately, many organizations' data lakes and other data stores have turned into dumping grounds, with **no easy way to access and analyze their data**.



SQREAM

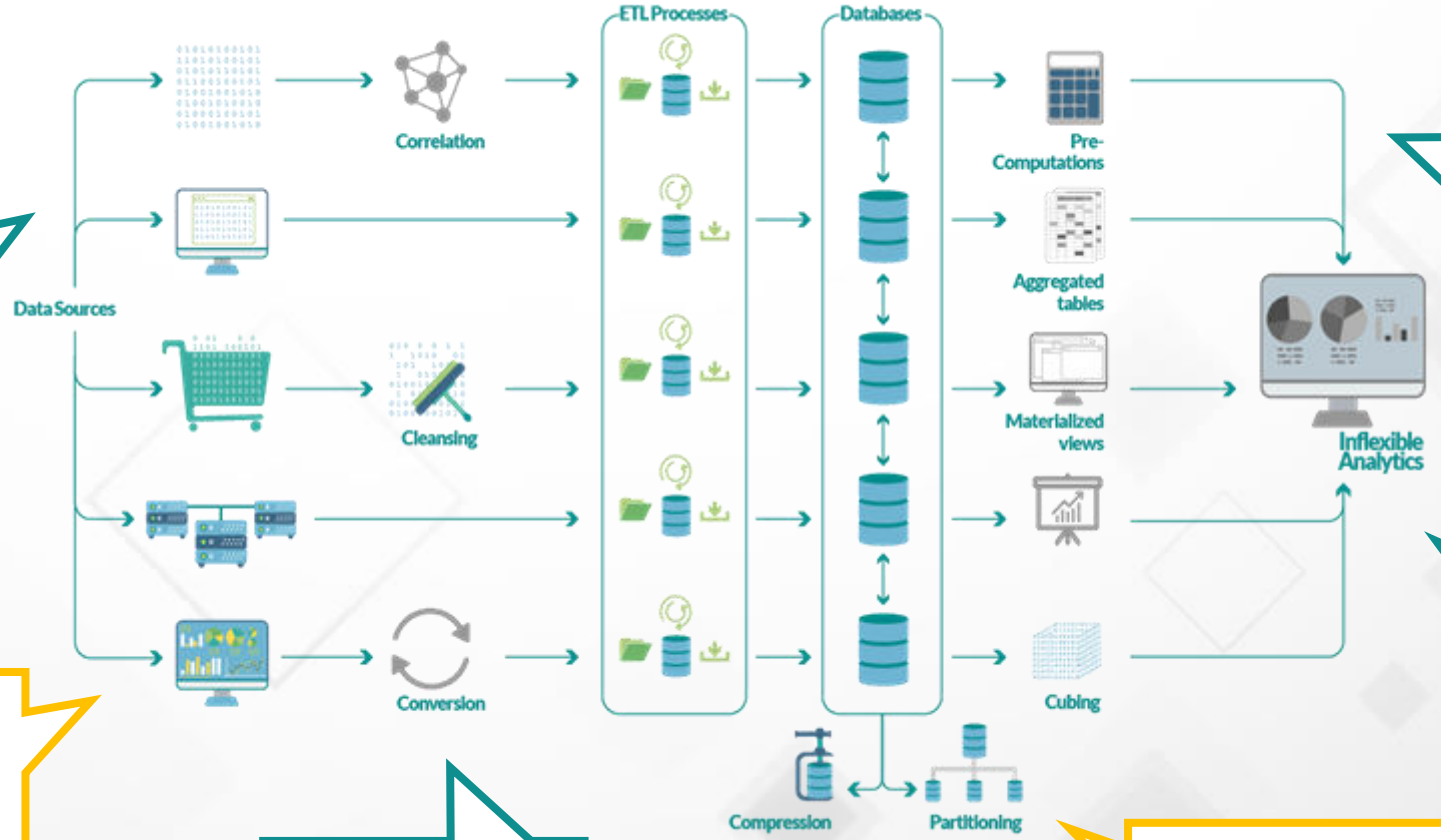
COMPANIES ARE DATA RICH, BUT INSIGHT POOR

Analytic Data	Data Lake	% of Data Analyzed
10 TB	100 TB	10 %
20 TB	500 TB	4 %
30 TB	1 PB	3 %
50 TB	10 PB	½ %

… VALUABLE INSIGHTS GO UNCOVERED

SQREAM

COMMON STRUGGLES FROM INGEST TO INSIGHTS



"Queries run **way too long** and **sometimes not at all**"

"We have lots of data, but I can analyze only **10%**"

"Lengthy data **preparation**"

"Some important queries are **too complex** to run"

Queries using too many system resources get **canceled**"

"Our analytic reports are **limited** in scope"

SQREAM

AVERAGE TIME SPENT ON DATA PREPARATION

Organizations report that they spend **more than 60%** of their time in data preparation, leaving little time for actual analysis.

Gartner - Market Guide for Data Preparation

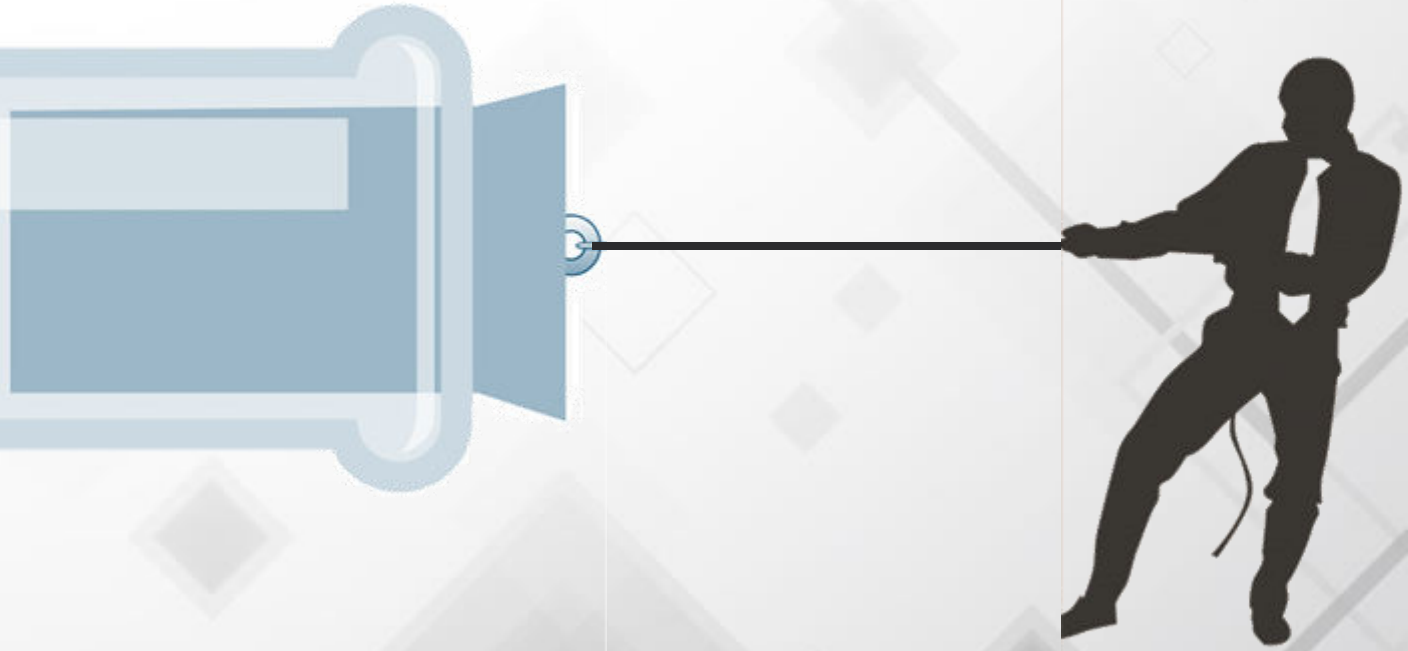
Most data scientists spend **only 20%** of their time on actual data analysis and 80% of their time finding, cleaning, and reorganizing huge amounts of data, which is an inefficient data strategy.

Forrester

SQREAM

MAIN REASONS FOR QUERY BOTTLENECKS

1. Size of the data stores
2. Data ingest
3. Data preparation
4. Varied location and storage of data
5. Query coding and complexity



SQREAM

COMMON PITFALLS HINDERING QUERY PERFORMANCE

- Bad (or outdated) schemas
- Legacy data warehousing strategies
- Poor indexing strategies
- Incorrectly configured servers
- Underpowered hardware

SO HOW DO I MAKE MY SQL QUERIES RUN FASTER?

IMPROVE PERFORMANCE OF YOUR EXISTING SYSTEMS

- Let the **database engine** do as much of the work as possible
- **Minimize I/O** of data to and from the database
- Break complex queries into **smaller queries**
- Prefer **optimized functions**, limit wildcards
- Use **views** carefully
- Check your **indexing strategy**
- Follow **best practices** for your DBMS

SQREAM

LET THE DATABASE ENGINE DO THE WORK

- Most DBMSs handle data very efficiently
- Wealth of functionality in text processing, math, summarization, sorting are best left for DBMSs, rather than in applications
- Moving some logic from the application to the DBMS ensures re-usability

SQREAM

MINIMIZE I/O OF DATA TO AND FROM THE DBMS

With very big queries, strain can come from transmitting result sets to the client. Do you really need *all the data?*

Reduce the result set size in the DBMS!

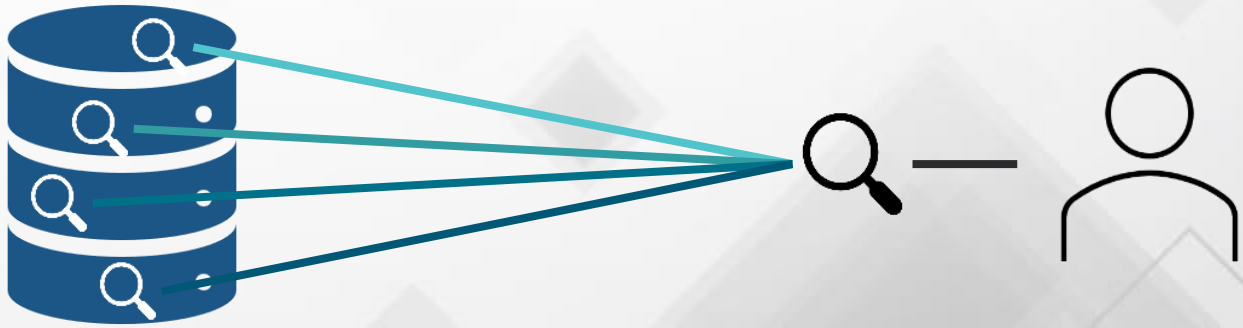
- Pull out the names of only the columns you need instead of using `SELECT *`. Otherwise, if you have a very wide table, the *client* may struggle putting it all together.
- Unless you need to see every row, limit the result set size with `LIMIT`.

SQREAM

BREAK COMPLEX QUERIES INTO SMALLER QUERIES

Large queries can be hard to work out. Breaking queries into small bits and using temporary tables can make the query much more understandable.

- Temporary tables and small queries are easier to debug - often eliminate the need for exotic syntax
- Less sensitive to failing because the query optimizer decides to do things differently
- Gives you a chance to optimize yourself. Sometimes this is better than stuff the DBMS optimizer hasn't figured out itself



SQREAM

PREFER OPTIMIZED FUNCTIONS, LIMIT WILDCARDS

Vendor-written functions often outperform SQL functions chained together.

Some examples:

- With lots of Unicode text, using pattern matching (`LIKE '%foo%'`) can result in very inefficient plans. A function like `ISPREFIXOF(x,'foo')` will likely perform better.
- Some DBMSs (like SQream DB and Postgres) are case-sensitive. If you need to match strings, try doing it on one side only or use a case-insensitive match (`ILIKE`)

USE VIEWS CAREFULLY

Views are virtual tables that are created from a query.
A view is materialized when you run a query that accesses them.

If your query uses a view, or even your view has another view, you're running many queries without knowing it (This isn't a problem in SQream DB, but with some other DBMSs could surprise you).

- Use temporary tables to materialize a view every hour/day/week or as required
- Consider using a CTE instead of a view.

SQREAM

CHECK YOUR INDEXING STRATEGY

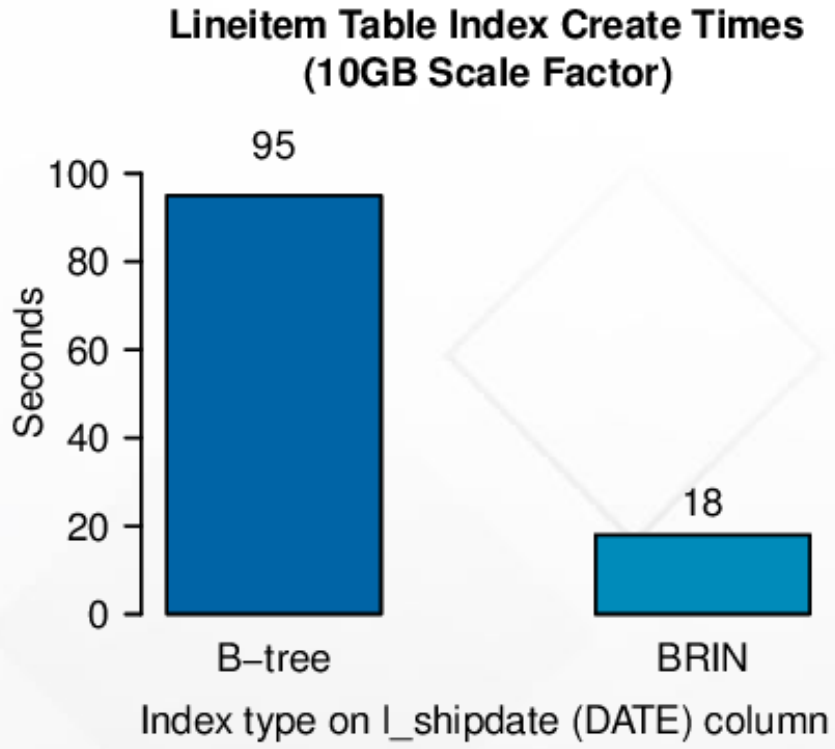
In many DBMSs, indexes speed up your query, letting the DBMS know where to look for data.

- Apply indices selectively – focus on columns with high cardinality or columns you use often.
- Too many indices decrease write performance (and take up lots of space):
  - Remove indices that aren't used or rarely used.
  - Remove indices that are placed on columns that have random data and/or are frequently updated

SQREAM

INDEXING STRATEGY AFFECTS DATA SIZES, PERFORMANCE

Postgres BRIN index vs. B-Tree



2ndQuadrant
Professional PostgreSQL

SQREAM

Data by "2ndQuadrant PostgreSQL", tested on Postgres 9.5

FOLLOW BEST PRACTICES FOR YOUR DBMS

Each DBMS is different, so be sure to apply best practices for the DBMS you use. However, there are some things that'll always be true:

1. If one query is slow, check the best practices for query optimization
2. If your whole database is slow, check your system:
   - Cluster status – all nodes are up, open for statements
   - Query distribution – are all statements ending up on the same node?
   - Data distribution – is data not distributed correctly? Could a schema change do some good?
   - Deleted rows – many big data DBMSs including SQream DB, Vertica, Postgres benefit from a periodic maintenance of deleted rows (also called delete predicates, delete vectors)
   - Locks – check if any statements are waiting due to locks, like when a big table is being updated
   - Counters – If your DBMS supports it, set up counters to track memory and compute usage

SQREAM

THE ELEPHANT IN THE ROOM…

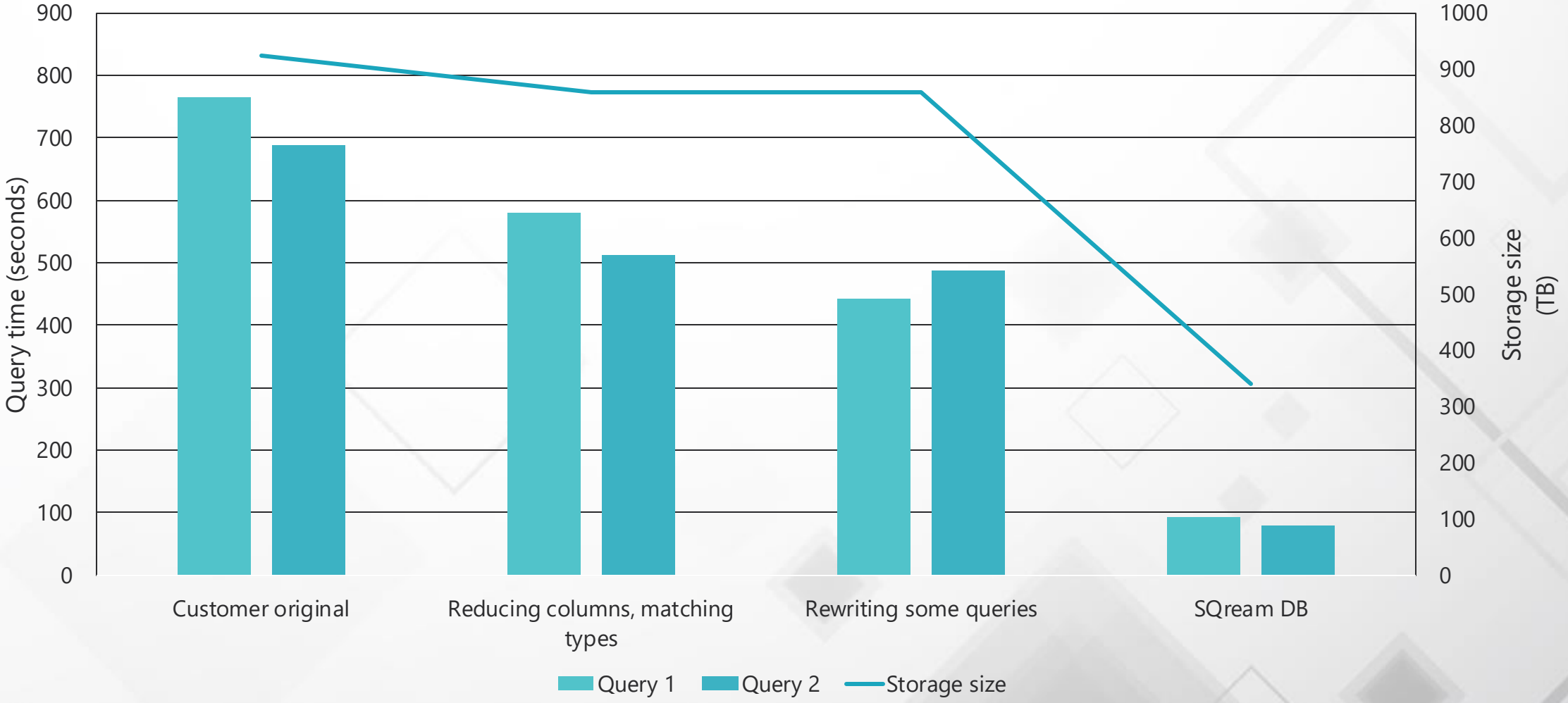
There's only so much you can tweak and tune. When you can't (or won't) tune any further, consider updating your DBMS.

New SQL DBMSs like SQream DB:

- ✓ Use compute resources to their fullest, and then some
- ✓ Can automate metadata collection
- ✓ Have new methodologies that scale better, less susceptible to "growing pains"
- ✓ Allow joining any amount of tables without significant performance impact

SQREAM

RESULTS AFTER FOLLOWING BEST PRACTICES



AIS is the largest mobile operator in Thailand, with over 41 million subscribers.
Faced with increased competition, AIS wants to improve customer service and reduce churn.

BILLIONS OF CDRs
weekly

DIFFICULTY SCALING
MPP system

EXTREMELY SLOW
Ad-hoc querying

SQREAM

AFTER THREE DAYS OF IMPLEMENTATION

Dashboard showing 3G/4G data throughput in Bangkok through the day (morning, lunch, evening, night…)

Larger circles represent better data throughput.

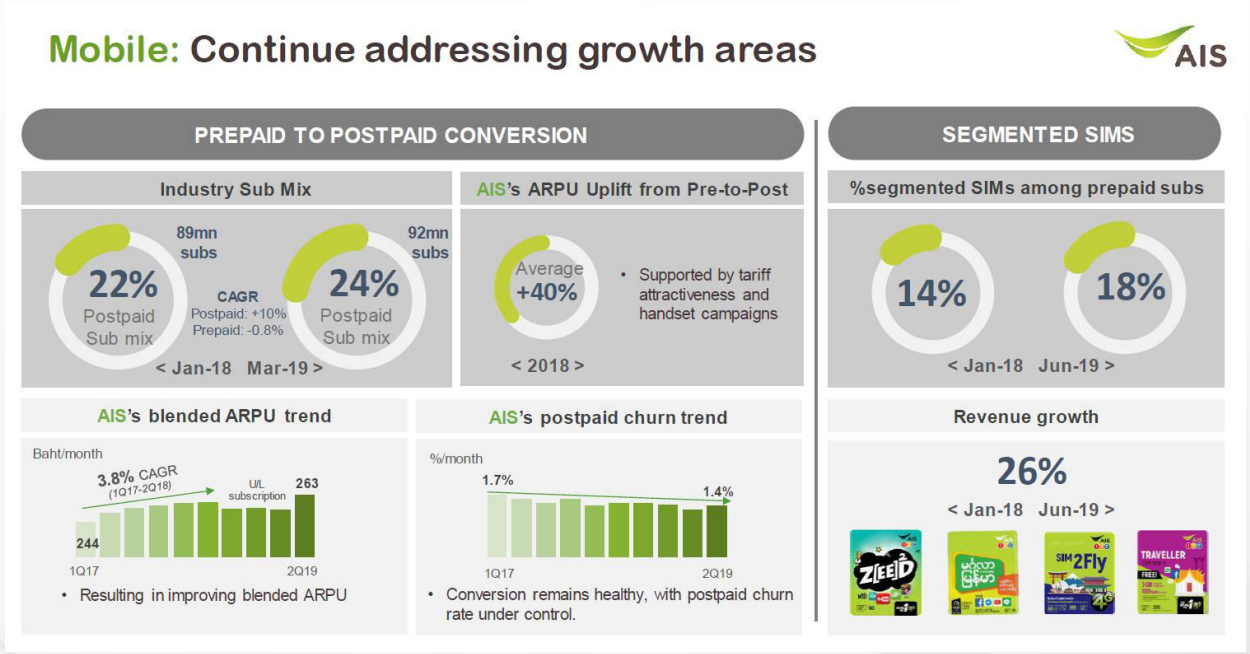
3 Table Join
(3.3B rows ⋈ 40M rows ⋈ 300K rows)

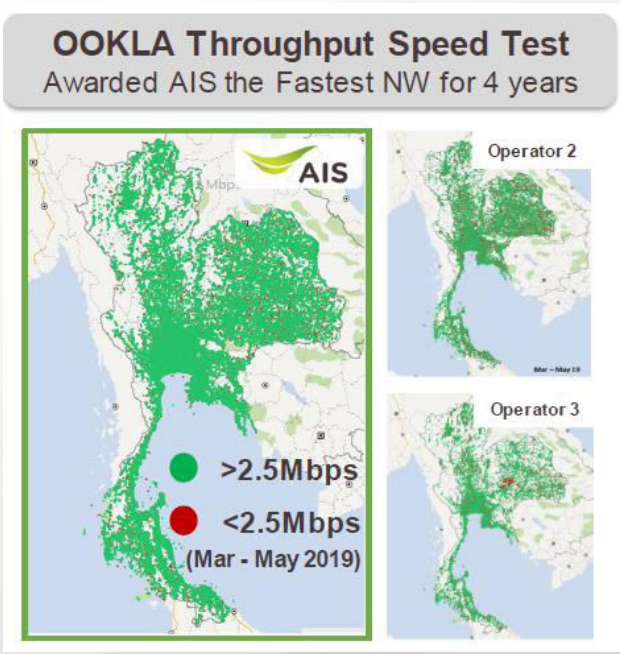
Dashboard aggregates directly off SQream DB, with no pre-aggregation steps.

HOW AIS USES ACCELERATED ANALYTICS

1. Segmenting the mobile market
2. Converting customers to better plans
3. Analyzing network performance to maintain superiority

} **Happier customers that feel "seen"**





SQREAM

WHAT IT MEANS FOR THEIR BUSINESS

Holistic view of network subscribers

Achieve deeper data analysis

Troubleshoot multiple locations

Save hours on queries and reports

Ability to drill down on massive raw data

Faster competitive analysis

Q&A

David Leichner, CMO

david@sqream.com | **sqream.com**

ADDRESS

Headquarters, 7 WTC
250 Greenwich Street
New York, New York

WE ARE SOCIAL

linkedin.com/company/sqream_/

@SQreamTech