



Presented by:

Sandro Frattura

B.I. Data Architect

A background image showing a group of people in a meeting. A man with glasses and a beard is smiling in the foreground, and a woman is partially visible behind him. The scene is set in a modern office environment with warm lighting.

*Data for Breakfast*

# About Hubspot



## Who is Hubspot?

Hubspot helps small-to-medium sized business *GROW BETTER*

- inbound marketing software and automation
- sales management software (CRM)
- website content management
- SEO optimization
- Blogging & social media integration

Our mission to help business attract  
leads and turn them into loyal customers

# About Hubspot



WE ARE NOT A GYM!



# Life Before Snowflake



## Scenario



Bring together disparate data sources into one unified data warehouse

## Pain Points



- Resource contention
- Downtime
- Requires Infrastructure Staffing
- Structured and semi-structured data
- Inflexible computing power

## Solution

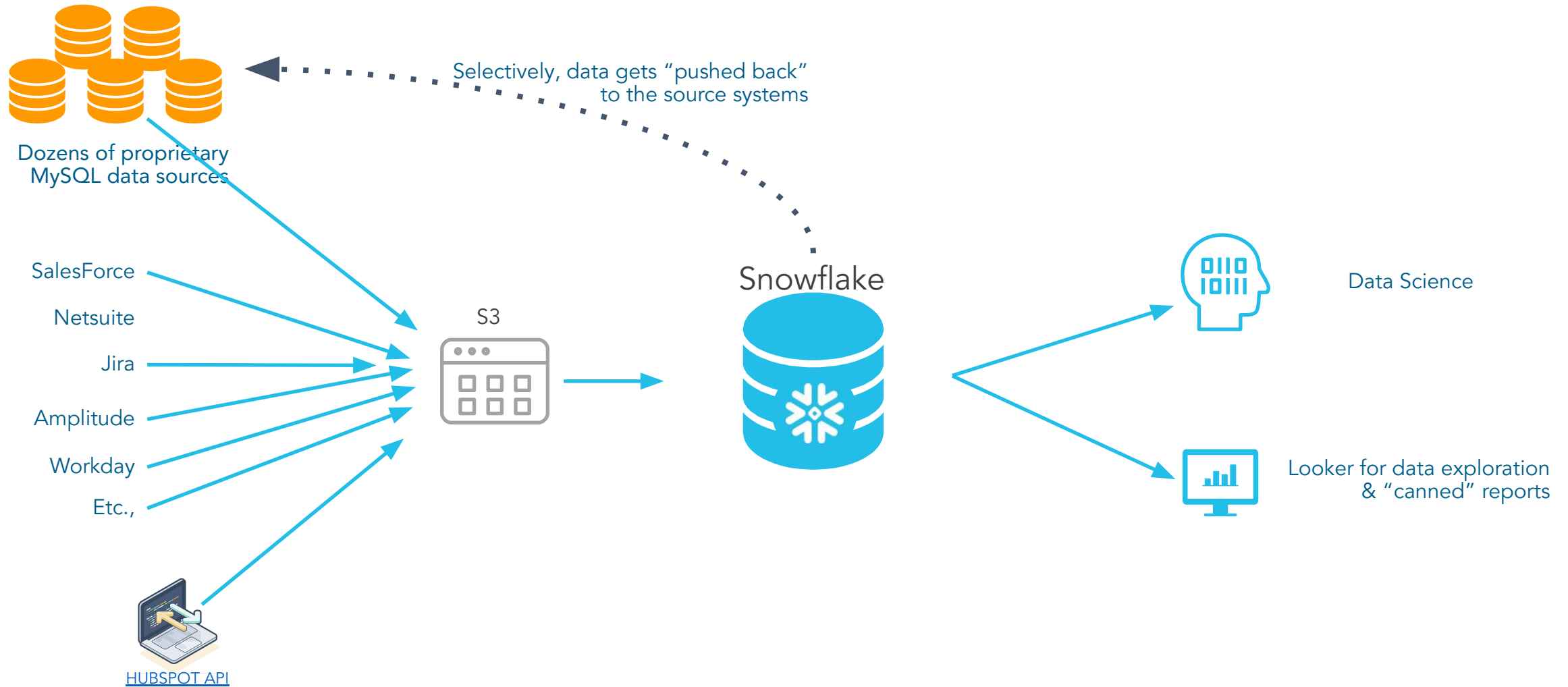


Replace existing columnar database solution with Snowflake

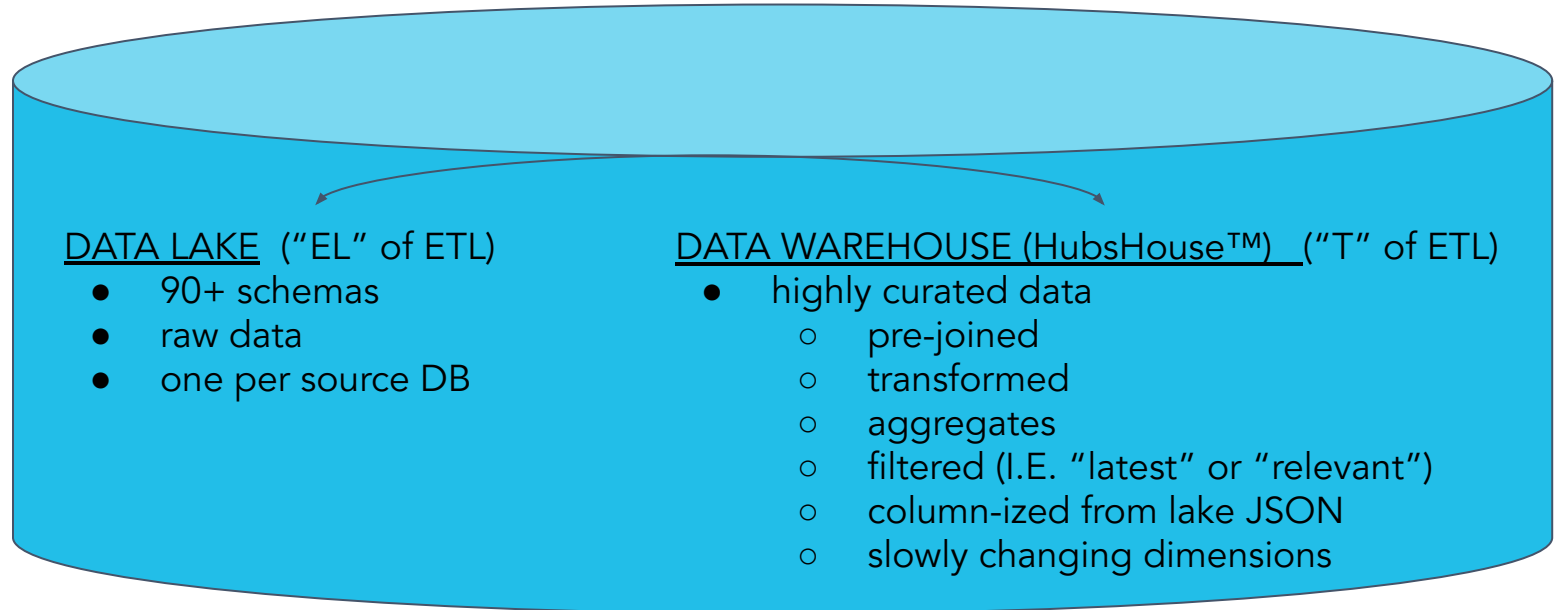
WAIT  
YOUR  
TURN!



# Data Flow From 30,000 Feet



# WHAT'S IN THE BOX???



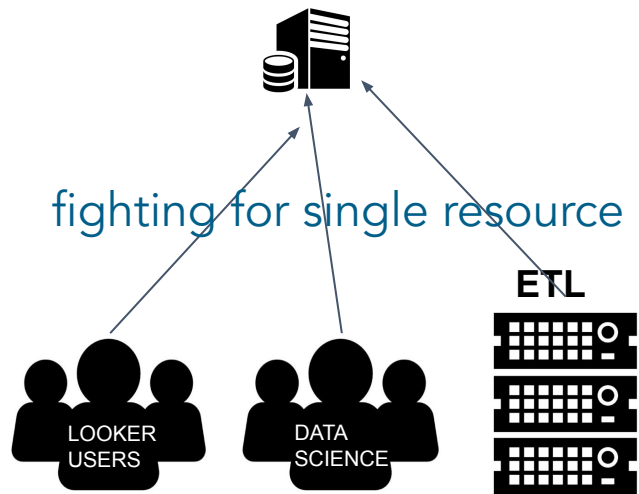
## Unified data

- raw data (Data Lake)
- curated (Data Warehouse)

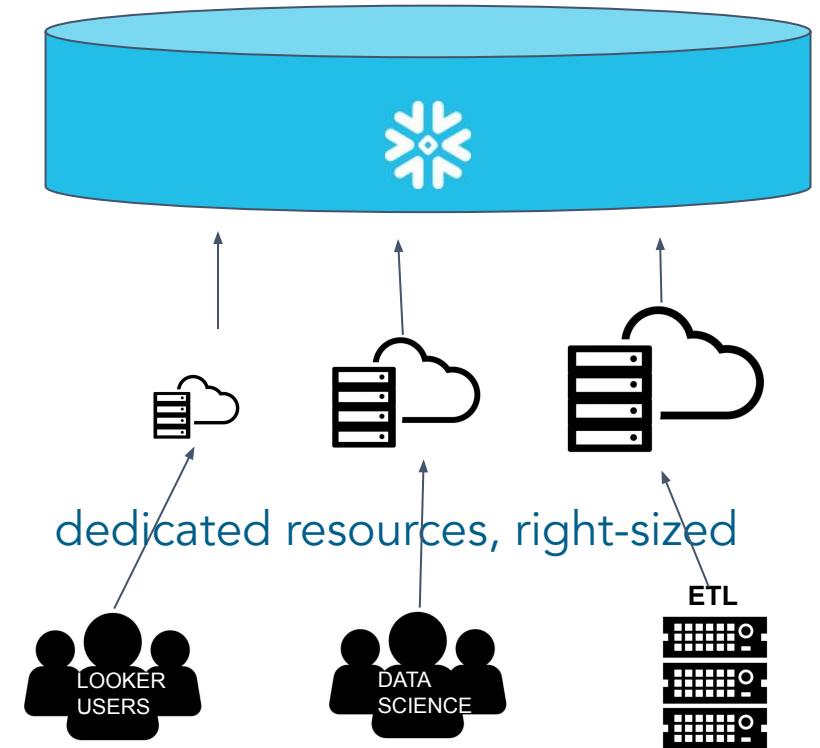
# Isolatable/Scalable Computing



Before



After



Assign computing power purposefully

Eliminate bottlenecks

Scale dynamically in real time

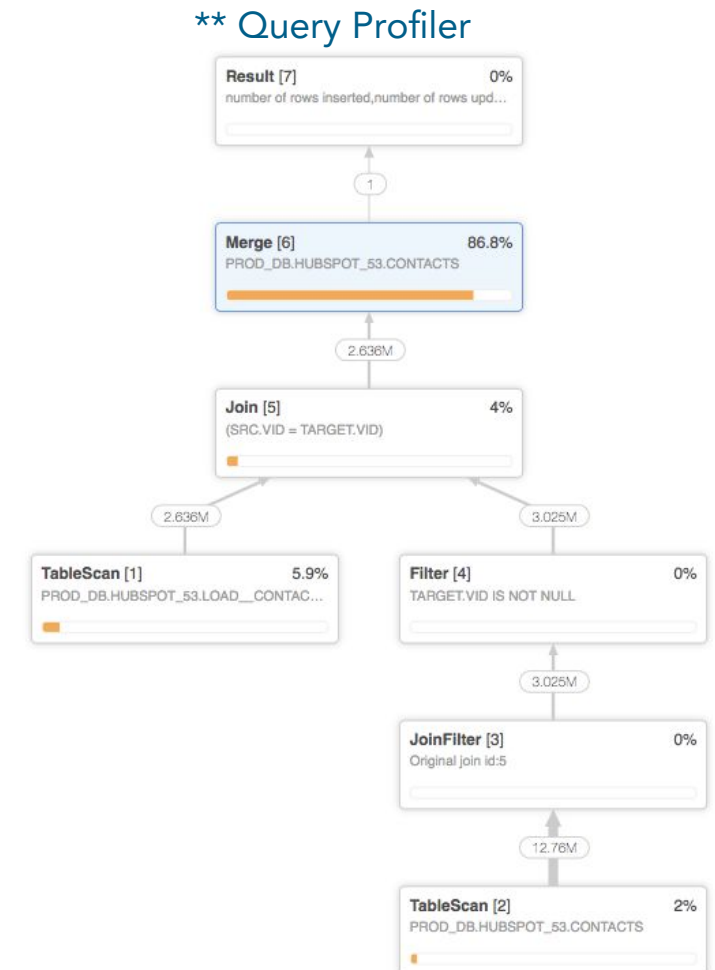
Measure usage and spend by function/team/dept

Delight your users/stakeholders

# Miscellaneous Goodness



- ❖ Natively support of JSON (et al): You can query a single-level JSON object as efficiently as transforming it into rows/columns. *For more complex JSON (nested, arrays, lists, time-series), you should FLATTEN the data in your transformation layer*
- ❖ Merge: The merge functionality (sometimes referred to in our industry as “upsert”) allows you to join data sets, and apply complex post-join logic to decide if you want to:
  - update rows (in any number of ways - very flexible)
  - insert rows
  - delete rows
- ❖ Time-travel: Query at the data as it was at a previous moment in time
- ❖ Web-based UI: While you are able use 3rd party tools (like DataGrip, Workbench etc), Snowflake offers a nice webUI to work with your data
- ❖ Query Profiler\*\*: Snowflake’s query profiler gives you a VISUAL way to understand what is happening in your queries.



# Tip #1: Migrate Immediately



- ❖ DATA ENGINEERS: Changing all of your pipelines to send data to Snowflake will *take some* engineering effort
  - you might want to address some tech debt while you rewrite your ETL code
- ❖ DATA ANALYSTS: some functions from legacy DB need to be rewritten. This *takes time*
  - you might find query rewrite opportunities
- ❖ END-USERS: Connecting downstream tools (Tableau, Looker, ML/AI scripts etc) at Snowflake will *take some time*
  - users will need some time connecting to, and getting used to, a new database

To achieve these goals, create a tool to blindly “bulk copy” your data from the legacy DB to Snowflake (maybe even nightly)



\*\*write speed: ~20MB/node/sec.

Large Warehouse (8 nodes)  
~160MB/second  
~10 GB/minute  
~1 TB in 90 minutes

# Tip #2: Cluster Large Data Sets



Regardless of #nodes, CPU speed, amount of RAM etc., the most important aspect of query performance on large data sets is data organization. In Snowflake, this means applying cluster keys on your large tables and then maintaining the organization by reclustering regularly.



By analyzing your data query patterns, you can decide which column(s) to cluster by. Typically a DATE column is a good candidate, plus another mid-cardinality column that is frequently found in "where" clauses (such as "client\_tier" or "product\_category" or "marketing\_cohort")

# Tip #3: Manage Warehouse Sizes



- SHUT IT DOWN: make sure warehouses auto-suspend after periods of inactivity
- SUPERSIZE ME: programatically upsize (on-the-fly) your warehouse for large ETL operations
- NO "BORED" WAREHOUSES: check & monitor usage. If you have very few queries, scattered throughout an hour -- your warehouse might never shut down but will actually also be highly unused. Rather than have 2 warehouses doing very little, you are better off having 1 warehouse doing a lot.
- POWER-UP YOUR USERS: don't be penny-wise but pound-foolish. Make sure your end users have a warehouse that is large enough to return queries in an acceptable time-frame. Making users wait for MINUTES to get a response when they could be waiting SECONDS instead is not only frustrating to the user, it is costing you money. *Trust me, your employee costs a lot more per hour than your warehouse.*
- MEMORY MATTERS: Warehouse performance is not always linear. A LARGE warehouse is 8x bigger than an XSMALL. In some cases, you get 8x better performance. But sometimes, you will get 100x+ better performance. Why? Because on the XSMALL warehouse, you might be spilling to disk during some operations (typically joins). Monitor your slowest queries closely and use Snowflake's **\*\*AMAZING\*\*** query profile to see if your warehouse does not have enough RAM.

# Closing Thoughts



How we brought our data nation together with Snowflake

- Have all of our data (raw and curated) in one place
- Give everyone access to data, eliminating resource conflict
- Leverage the cloud

Thank you!