
snowflake®
SUMMIT 2022
LIVE IN VEGAS
LAS VEGAS

LESSONS LEARNED BUILDING A LOW-LATENCY DATA WAREHOUSE

DAVID GARRISON | SENIOR DATA ENGINEER

DISCLAIMER

Other than statements of historical fact, all information contained in the presentations and accompanying oral commentary made available as part of this event (collectively, the "Materials"), including statements regarding (i) Snowflake's business strategy and plans, (ii) Snowflake's new or enhanced products, services, and technology offerings, including those that are under development, (iii) market size and growth, trends, and competitive considerations, and (iv) the integration, interoperability, and availability of our products with and on third-party platforms, are forward-looking statements. These forward-looking statements are subject to a number of risks, uncertainties and assumptions, including those described under the heading "Risk Factors" and elsewhere in the Quarterly Reports on Form 10-Q and Annual Reports on Form 10-K that Snowflake files with the Securities and Exchange Commission. In light of these risks, uncertainties, and assumptions, the future events and trends discussed in the Materials may not occur, and actual results could differ materially and adversely from those anticipated or implied in the forward-looking statements. As a result, you should not rely on any forwarding-looking statements as predictions of future events.

Any future product or roadmap information (collectively, the "Roadmap") is intended to outline general product direction; is not a commitment, promise, or legal obligation for Snowflake to deliver any future products, features, or functionality; and is not intended to be, and shall not be deemed to be, incorporated into any contract. The actual timing of any product, feature, or functionality that is ultimately made available may be different from what is presented in the Roadmap. The Roadmap information should not be used when making a purchasing decision. Further, note that Snowflake has made no determination as to whether separate fees will be charged for any future products, features, and/or functionality which may ultimately be made available.

The Materials may contain information provided by third-parties, including those participating in this event. Snowflake has not independently verified this information, and usage of this information does not mean or imply that Snowflake has adopted this information as its own or independently verified its accuracy.

David Garrison

Senior Data Engineer at QuoteWizard

Likes: Data, Math, Beer, Probably other things?

linkedin.com/in/**david-garrison**/
github.com/**DavidGarrison**/SnowflakeScripts

QuoteWizard® by lendingtree®

If only I had insurance for my new car

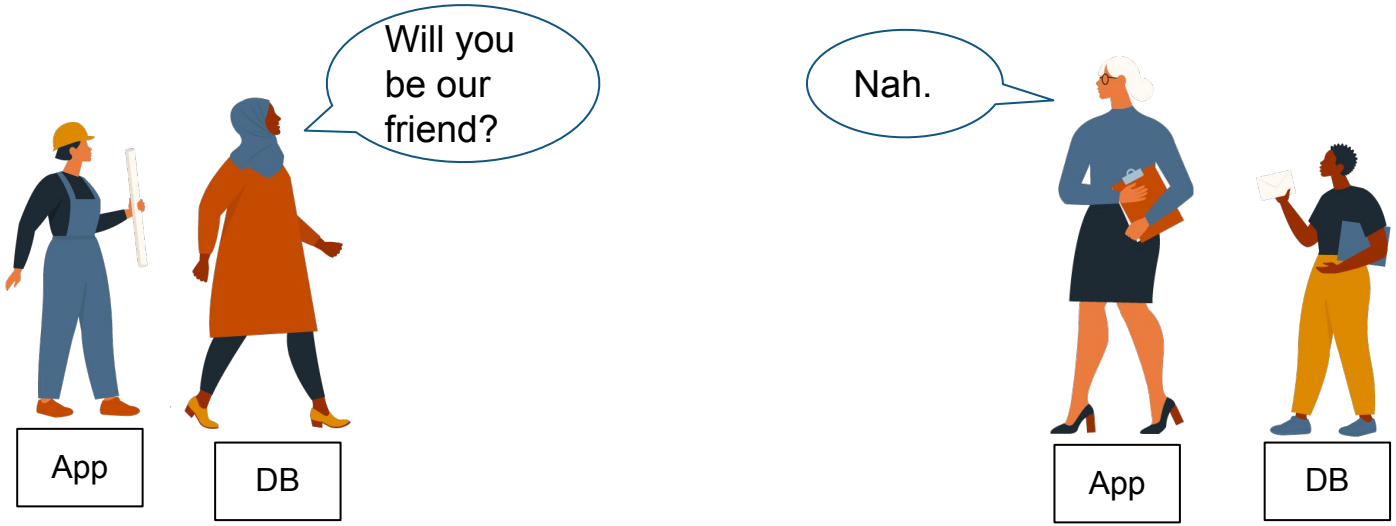
If only I didn't have to cold call for leads

Consumer

Insurance Agent

WHAT WE HAD

- Applications using different standards
- Can’t all talk to each other effectively
- Multiple ways to generate and attribute revenue
- No single report for business-level analytics



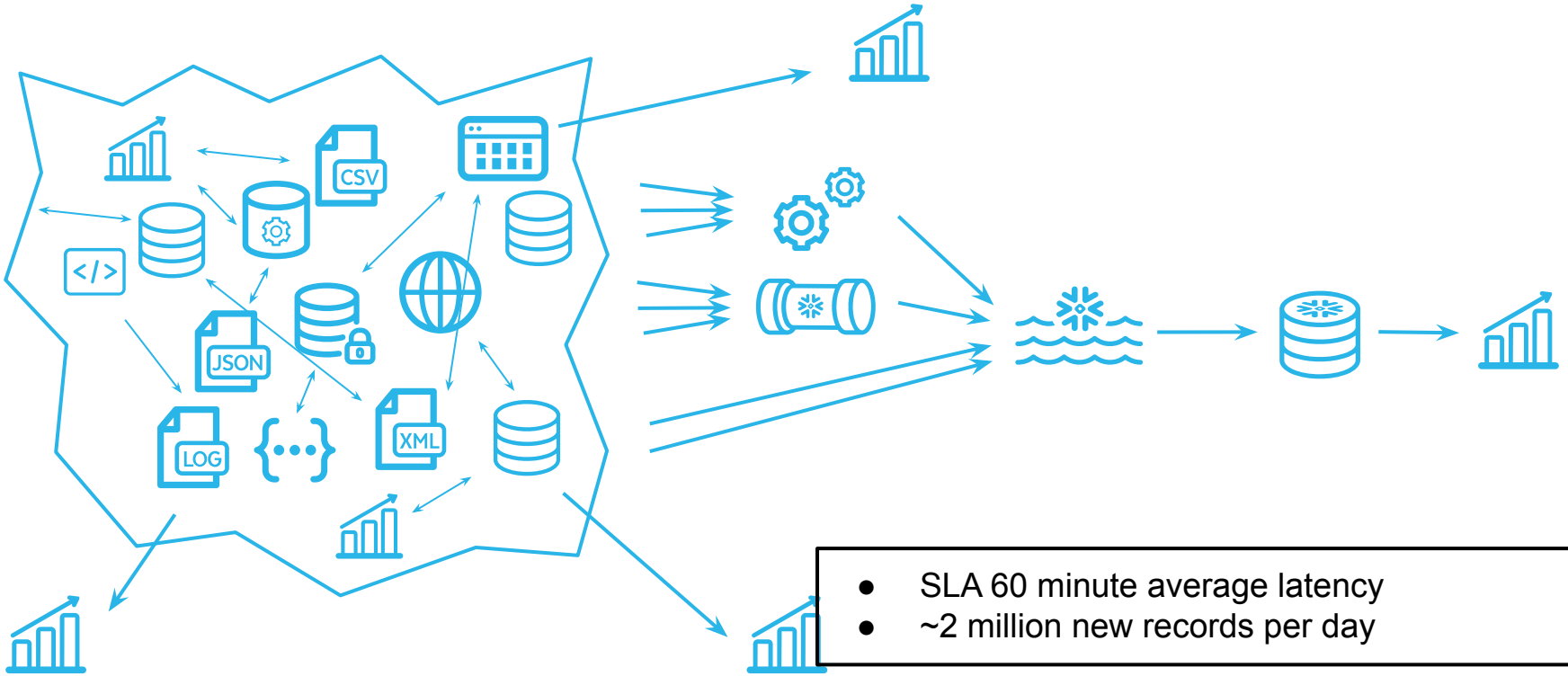
WHAT WE NEEDED

- A single reporting system built on a “source of truth”
- Combined data from across sources

The Number one demand from marketing/analysts/business:

- Up-to-date data
- Willing to sacrifice query speed, report features, historical data

WHERE WE ARE NOW



- SLA 60 minute average latency
- ~2 million new records per day

David Garrison

Senior Data Engineer at QuoteWizard

Likes: Data, Math, Beer, Probably other things?

linkedin.com/in/**david-garrison**/
github.com/**DavidGarrison**/SnowflakeScripts

CUSTOM PIPELINES

- Most pipelines into Snowflake have the same basic pieces
- When I started this, Streams didn't exist
- So I built the whole pipeline myself

Lessons:

- Streams are great!
- Many tools do provide incremental loading. It's worth looking at and asking about
- Make extensive use of logging

LOG TABLE IN SNOWFLAKE

- Started with 10 tables. Worked great.
- Slowly built up to 120 tables

A **Compute warehouse queueing**

B **Too many concurrent connections to log table**

C **Snowflake connector throttling**

LOG TABLE IN SNOWFLAKE

- Started with 10 tables. Worked great.
- Slowly built up to 120 tables
- Started getting contention on log inserts
- Snowflake is built for batches (100MB)
- Needed to move logging to a different DBMS

Lessons:

- Use a tool how it's supposed to be used
- Take some time to learn about the boundaries, and why they're there, before pushing on them

SEMI-STRUCTURED DATA

- Tables with a single column of JSON (plus metadata)
- Querying semi-structured data directly worked great
- Added views directly on those tables
- We had performance issues with some queries, and not others

A	**Queries using Date Filters**
B	**Queries that SELECT all columns**
C	**Queries using ORDER BY**

SEMI-STRUCTURED DATA

- Tables with a single column of JSON (plus metadata)
- Querying semi-structured data directly worked great
- Added views directly on those tables
- We had performance issues with some queries, and not others
- “2022-01-01” and “10” aren’t native data types for JSON
- “Columnize” - Parsing into a separate column

Lesson:

- Semi-structured data is great, but sometimes it’s worth columnarizing
- Bonus tip: don’t parse JSON in Data Masks

DASHBOARD SOURCES

- Query a view that does business logic

Vs.

- Builds a set of staging tables
- Builds another table using the staging tables, which handles business logic and clustering
- Query the final table

Lesson:

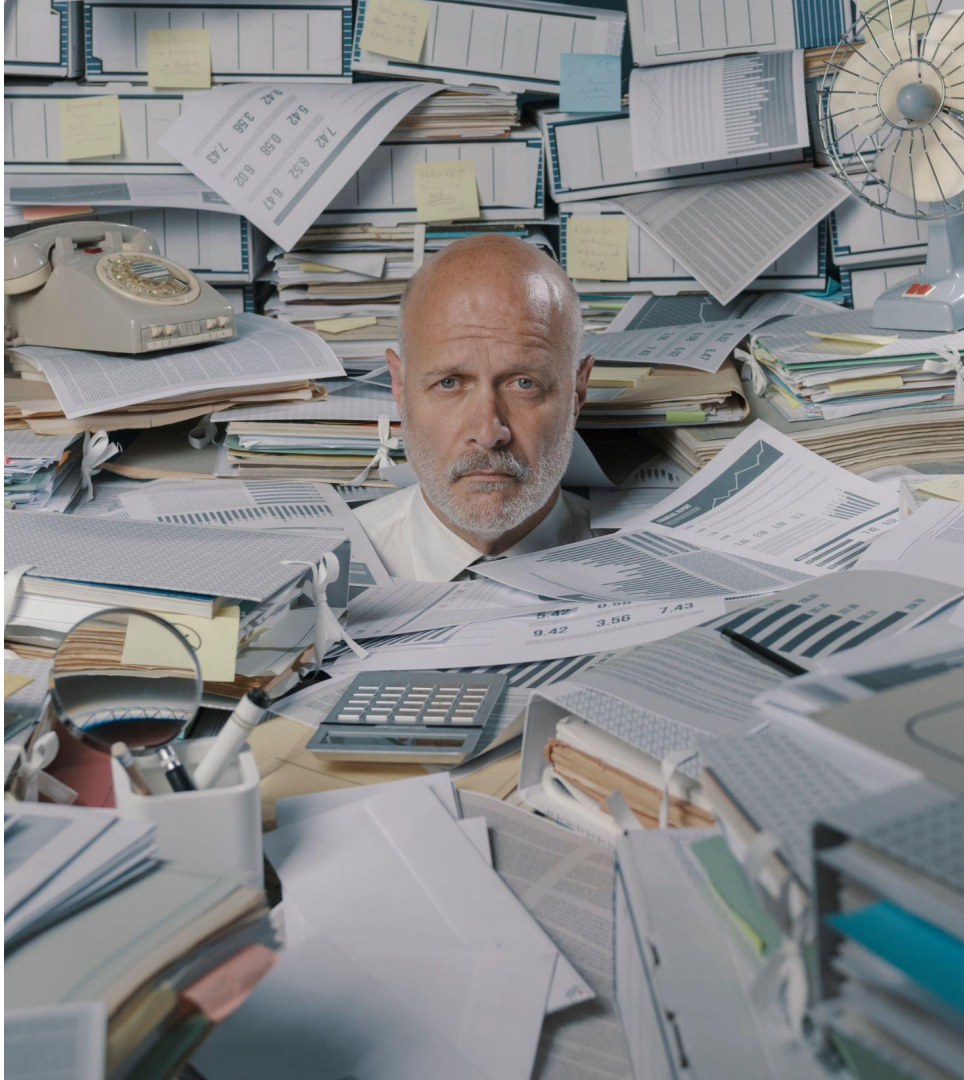
- Experiment with Queries vs. Views vs. Tables
- DBT is really good for experimenting quickly

LOTS OF SNOWPIPES

- Snowflake best practice: Use a single “Notification Integration”
- A new test environment queue with lots of messages, but no Snowpipe was set up
- The Integration quickly backed up and stopped receiving messages for **ALL pipes**
- Paused the flow, integration cleared up
- Ran **ALTER PIPE REFRESH** on all pipes

Lessons:

- “Best practices” are a good place to start. They are not always rigid
- Notification Integrations can handle a lot (when they’re wired up correctly)
- ALTER PIPE REFRESH



LESSON LEARNED?

- Created a separate notification integration for our test database
- Pointed it at the same Azure storage queue
- ALL messages started going to the new integration, and **ALL pipes** stopped again
- Had to recreate our integration and all of our pipes

Lessons:

- Be careful with things that affect your system at an account level
- Test in a separate account if possible

ALL GOOD NOW, RIGHT?

- Pipes back up and running with a new shiny integration for test environment
- The pipes had been down for a while and needed to get caught up
- Ran **ALTER PIPE REFRESH** on all pipes.
  (The exact same script from before)

ALTER PIPE REFRESH

What happened?

A

The pipes/tables refreshed normally

B

Nothing

C

Created a bunch of duplicate records

D

Test records went to Production tables

DUPLICATES

- All of our pipes reloaded 14 days of data and created a TON of duplicates
- Had to do manual data cleanup, but at least the pipes were working again

Lessons Learned:

- Include metadata in your Snowpipe tables (FILENAME, FILE_ROW_NUMBER, CURRENT_TIMESTAMP)
- Recreating objects tends to mess with things like copy history
- ALTER PIPE REFRESH does have parameters to prevent this issue
- Spare some extra time for testing, even if people are watching you

THE END?

FAST, FLEXIBLE

- 5-15 minute delay on everything in our data lake
- 30-60 minute delay for data warehouse
- Queries are fast
- We slow it down in the evenings to conserve on cost (mostly ADF cost)
- Snowflake cost: a lot less than the old system

RECAP

- Be aware of how a tool is supposed to be used
  - Read the documentation
  - Try to follow best practices (most of the time)
- Parse dates and important columns out of semi-structured data
- Use a separate account for testing account-level changes
- Streams are cool
- ALTER PIPE REFRESH is cool – Be careful with it
- Let Snowflake do the heavy lifting for report logic

ASSORTED LESSONS

- Read the documentation
- Don’t treat all data the same
  - Full reloads vs. incremental
  - Semi-structured vs. columnarized
  - Clustered index vs. natural clustering
- Cloning does weird things with pipes, masks, and future grants
- Use a generic XS warehouse for all the little jobs and pipelines. Isolate/scale as needed
- Why’d my table disappear? Because of ownership or usage grants, probably
- Is my Pipe/Task working? Look at the Copy/Task History
- Snowflake Quality of Life features: QUALIFY, DIV0, EQUAL_NULL
- (**Windows + V**) - the superior form of (**Ctrl + V**)
- Slides and Code Snippets: github.com/**DavidGarrison**/SnowflakeSummit

QUESTIONS

Please take a moment to rate your experience during this session.

INCLUDE THIS SESSION'S ID:

AA202

THANK YOU.