



# Data Apps: Data Warehouse as a Platform

Kashish Gupta - Co-Founder & CO-CEO, Hightouch

Data Council, Austin

March 2022



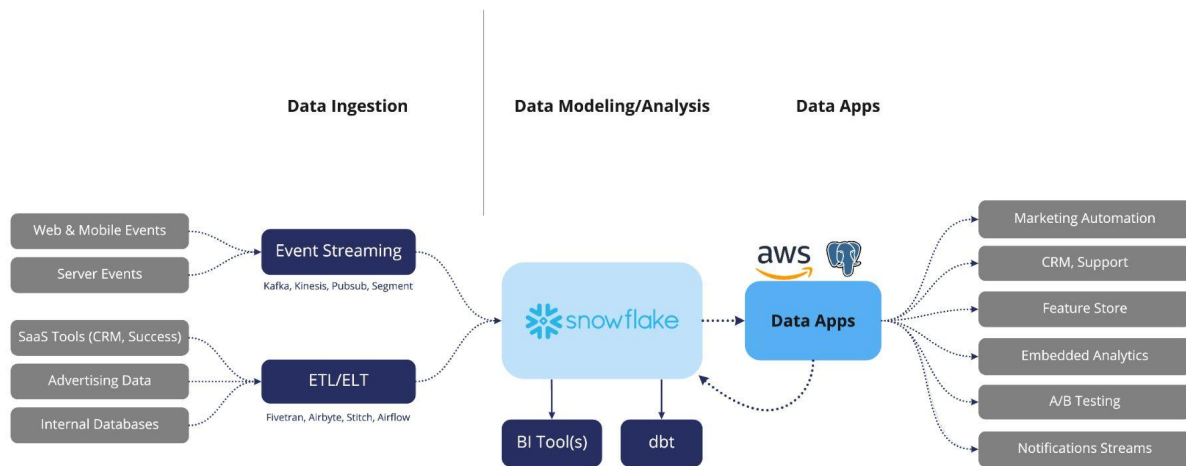
## Kashish Gupta

### Co-CEO and Co-Founder, Hightouch

Kashish Gupta is the co-founder and co-CEO of Hightouch.io, the leader in Data Activation. He spends every day helping companies realize the full value of their data and their data teams. He's been working on startups for the last 4 years, was part of Y-Combinator, and loves working on both product and go-to-market. A fan of racquet sports and karaoke, he's passionate about using Reverse ETL to make data warehouses actionable. Kashish attended the University of Pennsylvania where he earned an MSE in Robotics and a BS in Economics from the Wharton School.

# What is a Data App?

An app that is built on your analytics database rather than a transactional database.



Good primer: <https://pchase.substack.com/p/thenewbackend>

# Why do we care about data apps?

Why build a data app?

- Single source of truth
  - Use the database you already have
  - Consistency of information across all consumers
- Faster development
- Data team can have full control
- Better for reporting-esque use cases

Should I build these at my company?

- Maybe, but only if you have a strong use case. It's likely that you will inevitably build one without planning on it

When is the right time to think about this?

- After you have a warehouse with useful and clean data that you want to share with other parts of your company!

# When does this work well / not work well

## Pros:

- Consistency
- Less data silos (can access all application data)
- Faster to set up (data is already there and modeled)
- Works well with modern data stack

## Cons:

- Can't do transactions
- Performance (warehouses are slower)
- Can't do constraints (columnar data doesn't allow efficient searches by an ID or foreign keys)

# Common frameworks for data apps

- Data Warehouse is the primary store of the data
  - *Caching layer can help with realtime reads but should be a reflection of the warehouse*
  - The cache can update on a schedule (every hour) but not diverge from the warehouse
  - The data app can have its own transactional db for its control plane
- The app can be served from any server
- Data apps can be read-only
  - In fact read-only enforces consistency, making data apps more practical
  - If writes are allowed, the app may have to refresh the warehouse after each write.  
With materialization this becomes even more impractical

# Open Questions

How to reconcile that updates need to be fast enough to reflect in the UI?

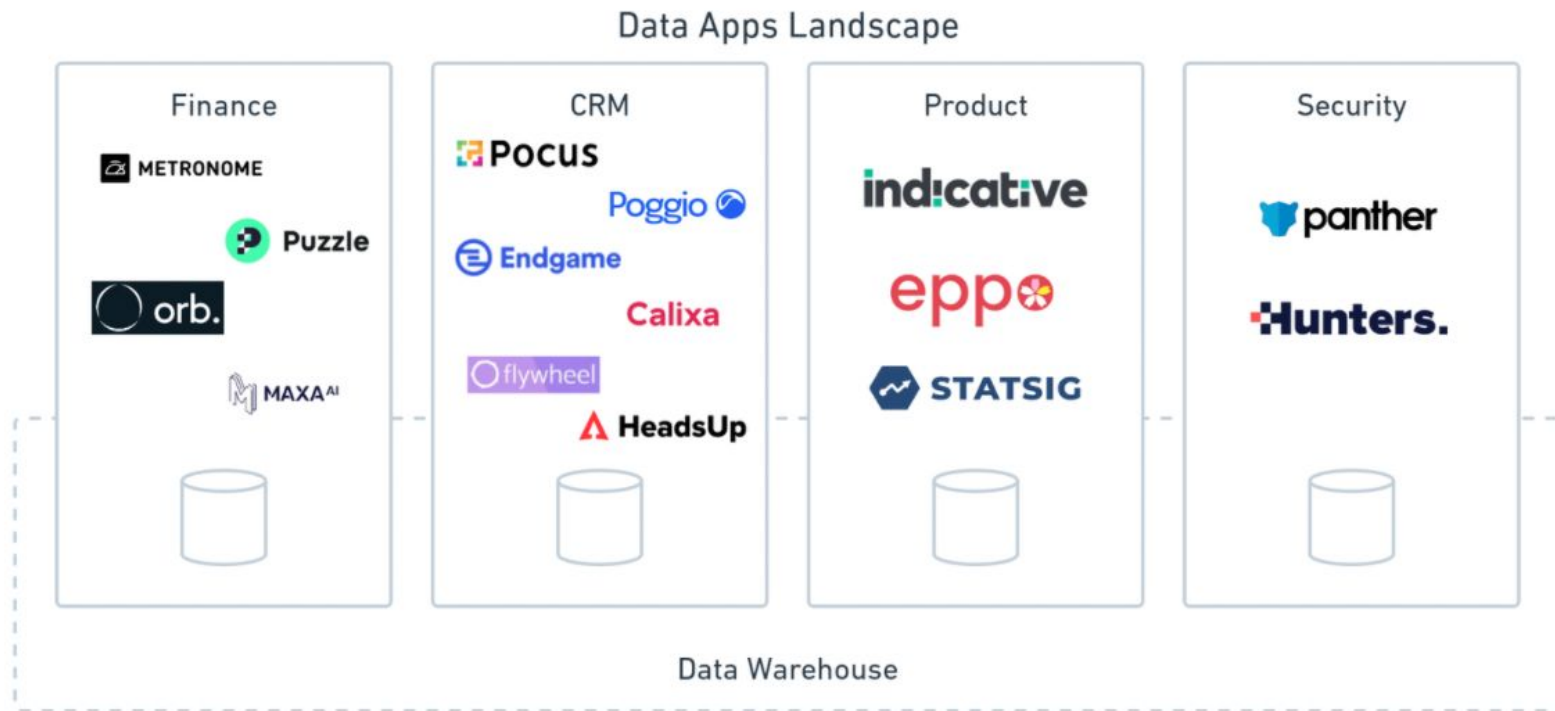
Which warehouses are fast enough to support data apps presently, if any?

## Why is this possible today?

- Centralized data all in one place
- Warehouse are FAST(er)
- Warehouses are easy to access, anyone can spin one up quickly without maintenance.  
Only pay when the query is run
- Materialize, Firebolt, and ClickHouse creating even better infra for reads



# Some examples

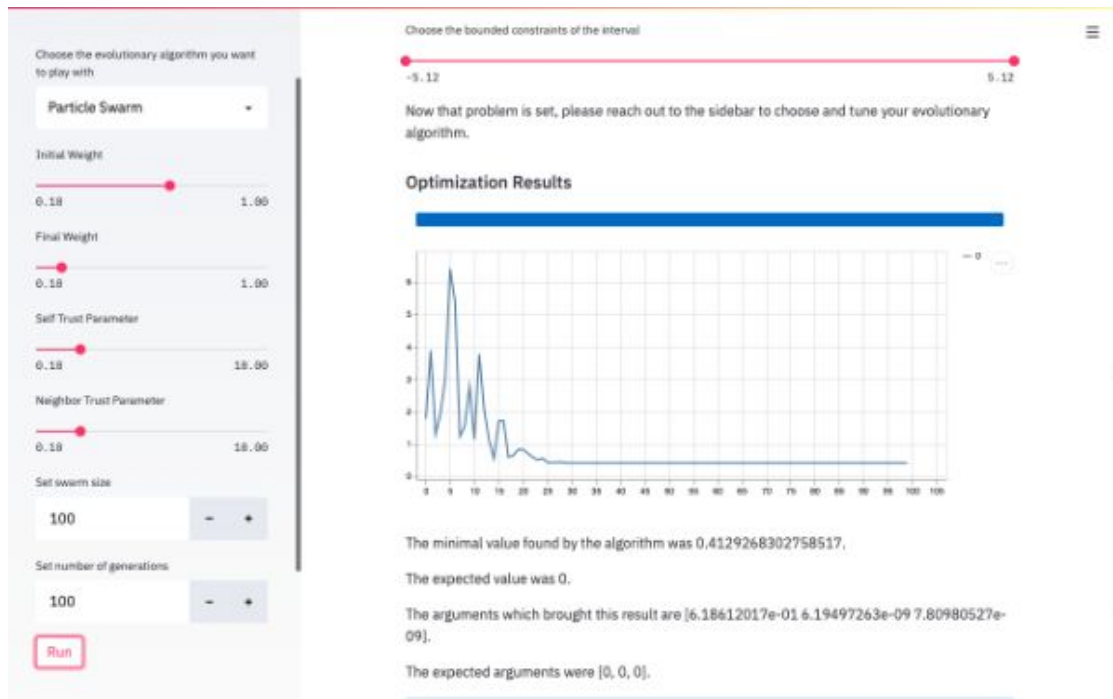


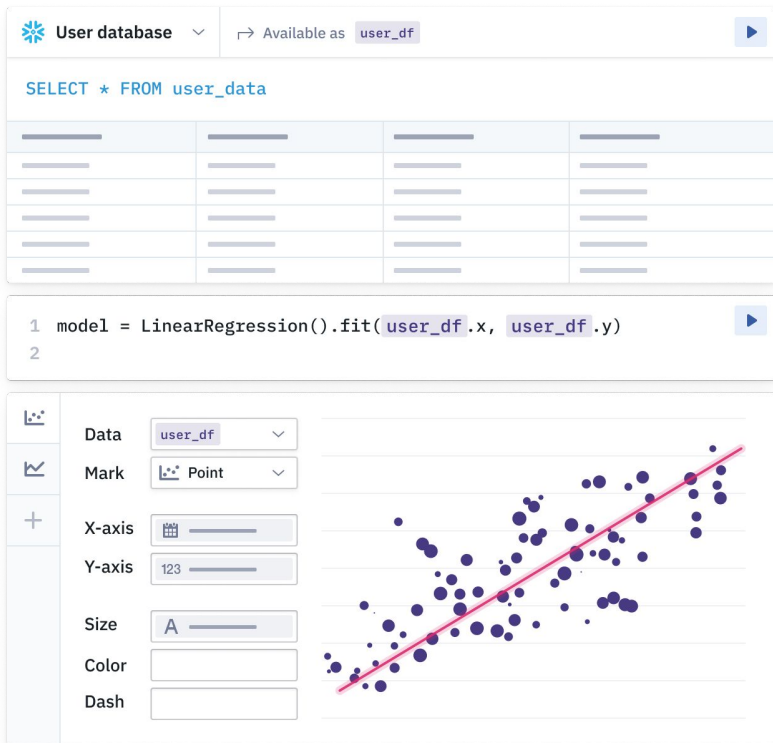


# Streamlit

## Streamlit

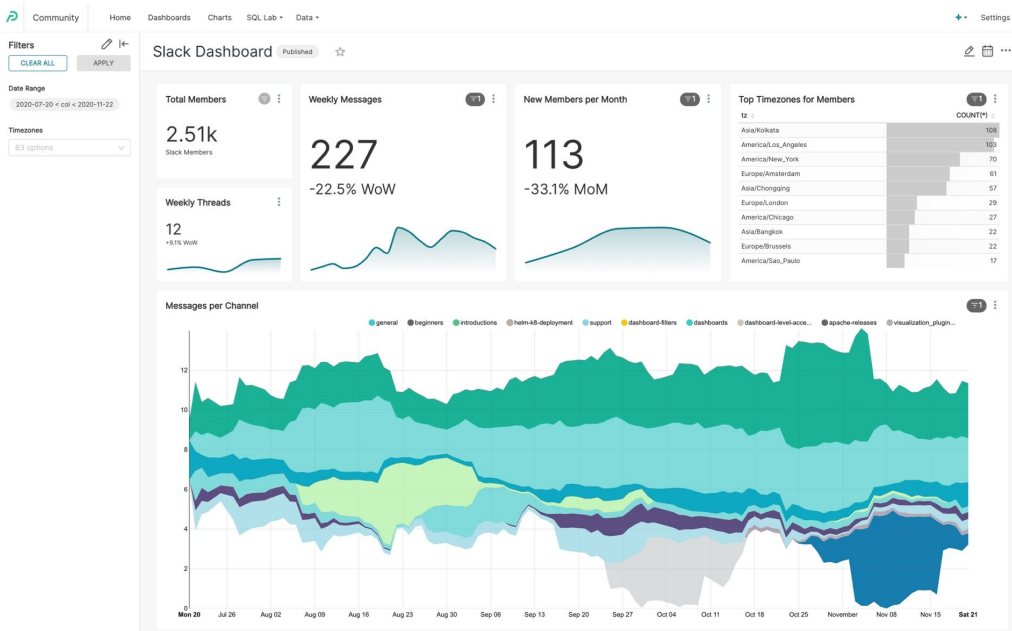
## Create data apps





Hex

## Collaborative notebooks



Preset

Embedded  
analytics



Build internal tools fast.

TEMPORARY STATE

Create new

QUERIES AND TRANSFORMERS

- button1ClickHandler {} 39 keys
- query1 {} 49 keys

COMPONENTS

- button1 {} 19 keys
- button2 {} 18 keys
- table1 {} 57 keys
- text1 {} 12 keys
- textInput1 {} 33 keys

GLOBALS

- current\_user {} 7 keys
- localStorage {} 1 key
- retoolContext {} 3 keys
- urlparams {} 2 keys
- viewport {} 2 keys

Header

Main

### Users

Name

id	email	first_name	last_name	active	created_at	updated_at	
3	droptablec...	Robert	DROP TAB	X	Dec 2, 20...	Jul 25	<div>Approve</div> <div>Reject</div>
4	louis@con...	Louis	Reasoner	X	Dec 3, 20...	Jul 25	
5	tenx@tryn...	Alyssa P.	Hacker	X	Dec 4, 20...	Jul 25	
6	cy@tryret...	Cy D.	Fect	X	Dec 5, 20...	Jul 25	

All queries completed.

Showing all 2

- query1
- button1ClickHandler

button1ClickHandler

Resource  Edit resource

Run query automatically when inputs change

Action type GET  (GET req) Will run automatically whenever a parameter changes.

URL parameters

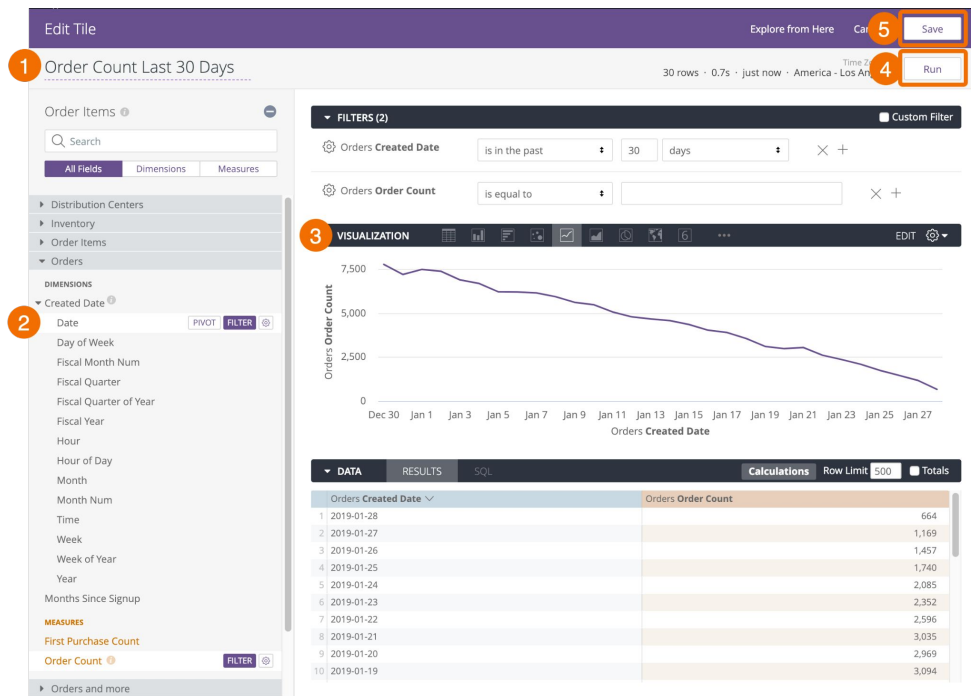
Retool

Internal tools  
that read from  
the warehouse



BI

(loosely counts)





Pocus

Dashboard Scores Automations

AG Alfredo Geidt  
alfredo@zapier.com

Self-Serve (Free)

Named Accounts

Expansion

+ Create

Search views

Full view

Alfredo's View

Filter

Fields

Actions

Sort

Save changes

Recent Usage Spike

Top Usage

Nearing Paywall

Alfredo's View

+ Create a view

Name	Customer fit	Product Engagement	Active Users
Gong > <span>Qualified</span> <span>...</span> <span>EXCELLENT</span> <span>EXCELLENT</span> 20			
Notion > <span>Qualified</span> <span>...</span> <span>EXCELLENT</span> <span>GOOD</span> 20			
Vanta > <span>Qualified</span> <span>...</span> <span>EXCELLENT</span> <span>EXCELLENT</span> 5			
Mailchimp > <span>Not qualified</span> <span>...</span> <span>EXCELLENT</span> <span>EXCELLENT</span> 50			
Pendo > <span>Not qualified</span> <span>...</span> <span>GOOD</span> <span>GOOD</span> 5			
Canva > <span>Qualified</span> <span>...</span> <span>FAIR</span> <span>POOR</span> 2			

Notion

Most important signals

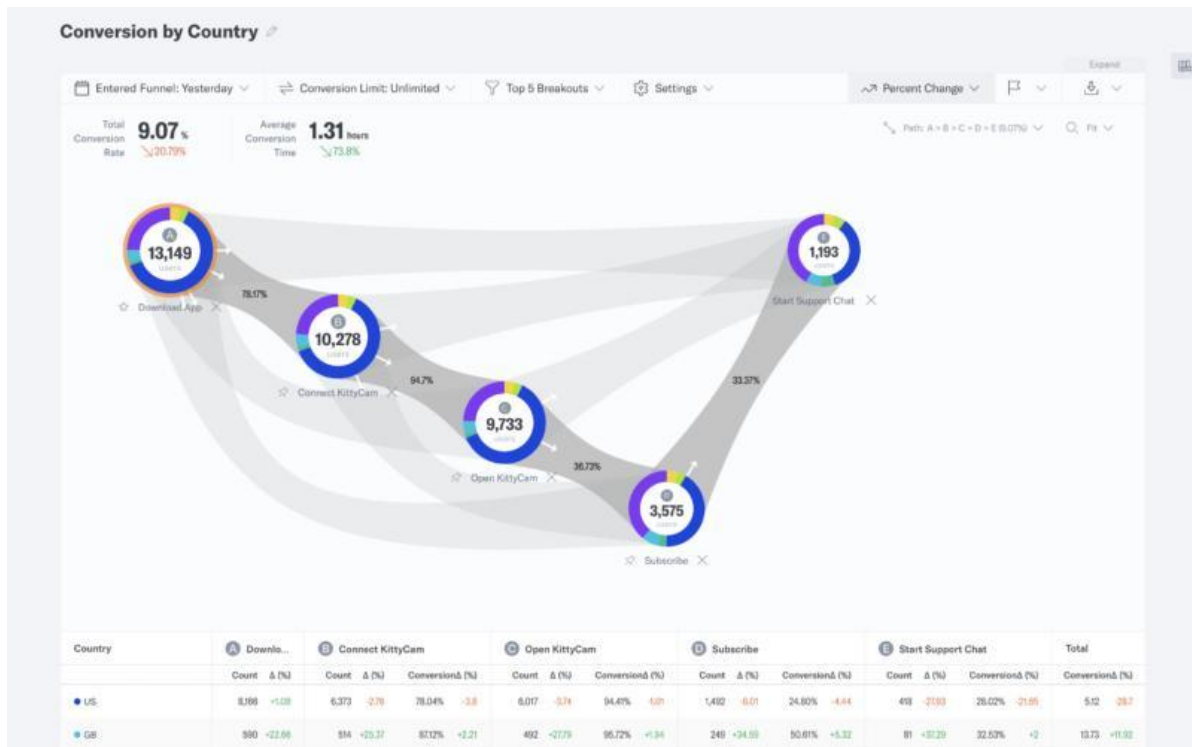
- 50-100 Zaps
- Usage spike
- Active users increased

## PLG CRMs

# Using product data to identify PQLs



# Product analytics on the warehouse







panther

Home > Data > Explorer

Overview  
Alerts & Errors  
Detections  
Analysis  
Data  
Data Explorer  
Indicator Search  
Query History  
Saved Queries  
Cloud Resources  
Data Models  
Integrations  
Settings

Select Database  
panther\_cloudsecuritypublic

Columns  
Filter

resource\_history  
accountId (text)  
am (text)  
changes (variant)  
changeType (text)  
id (text)  
integrationId (text)  
integrationLabel (text)  
lastUpdated (timestamp\_ntz)  
name (text)

New Query  
1 select  
2 id, accountId, am, name  
3 from resource\_history

Run Query Save as

Results  
Elapsed Time 00m 05s Data Scanned 85.82 MB Download CSV

	id	accountId	am	name
View JSON	arn:aws:logs:us-east-1:567780804328:log-group:/aws/lambda/panther-policy-engine*	567780804328	arn:aws:logs:us-east-1:567780804328:log-group:/aws/lambda/panther-policy-engine*	/aws/lambda/panther-policy-engine
View JSON	arn:aws:logs:us-east-1:567780804328:log-group:/aws/lambda/panther-log-router*	567780804328	arn:aws:logs:us-east-1:567780804328:log-group:/aws/lambda/panther-log-router*	/aws/lambda/panther-log-router
View JSON	arn:aws:logs:us-east-1:567780804328:log-group:/aws/lambda/panther-log-router*	567780804328	arn:aws:logs:us-east-1:567780804328:log-group:/aws/lambda/panther-log-router*	/aws/lambda/panther-log-router
View JSON	arn:aws:logs:us-east-1:567780804328:log-group:/aws/lambda/panther-athena-api*	567780804328	arn:aws:logs:us-east-1:567780804328:log-group:/aws/lambda/panther-athena-api*	/aws/lambda/panther-athena-api
View JSON	arn:aws:logs:us-east-1:567780804328:log-group:/aws/lambda/panther-athena-api*	567780804328	arn:aws:logs:us-east-1:567780804328:log-group:/aws/lambda/panther-athena-api*	/aws/lambda/panther-athena-api
View JSON	arn:aws:logs:us-east-1:567780804328:log-group:/aws/lambda/panther-athena-api*	567780804328	arn:aws:logs:us-east-1:567780804328:log-group:/aws/lambda/panther-athena-api*	/aws/lambda/panther-athena-api
View JSON	arn:aws:iam:567780804328:role/panther-LogAnalysis-16NX-AthenaAdminFunctionRole-IXWOTWLTWO	567780804328	arn:aws:iam:567780804328:role/panther-LogAnalysis-16NX-AthenaAdminFunctionRole-IXWOTWLTWO	panther-LogAnalysis-16NX-AthenaAd
View JSON	arn:aws:lambda:us-east-1:567780804328:function:panther-datalog-compact	567780804328	arn:aws:lambda:us-east-1:567780804328:function:panther-datalog-compact	panther-datalog-compact
View JSON	arn:aws:iam:567780804328:role/panther-LogAnalysis-16NXQ2B-AthenaApiFunctionRole-YGVWURMAMWHD	567780804328	arn:aws:iam:567780804328:role/panther-LogAnalysis-16NXQ2B-AthenaApiFunctionRole-YGVWURMAMWHD	panther-LogAnalysis-16NXQ2B-Athen
View JSON	arn:aws:lambda:us-east-1:567780804328:function:panther-log-processor	567780804328	arn:aws:lambda:us-east-1:567780804328:function:panther-log-processor	panther-log-processor
View JSON	arn:aws:lambda:us-east-1:567780804328:function:panther-datalog-compact	567780804328	arn:aws:lambda:us-east-1:567780804328:function:panther-datalog-compact	panther-datalog-compact
View JSON	arn:aws:logs:us-east-1:567780804328:log-group:/aws/lambda/panther-rules-engine*	567780804328	arn:aws:logs:us-east-1:567780804328:log-group:/aws/lambda/panther-rules-engine*	/aws/lambda/panther-rules-engine
View JSON	arn:aws:logs:us-east-1:567780804328:log-group:/aws/lambda/panther-remediation-api*	567780804328	arn:aws:logs:us-east-1:567780804328:log-group:/aws/lambda/panther-remediation-api*	/aws/lambda/panther-remediation-ap
View JSON	arn:aws:logs:us-east-1:567780804328:log-group:/aws/lambda/panther-policy-engine*	567780804328	arn:aws:logs:us-east-1:567780804328:log-group:/aws/lambda/panther-policy-engine*	/aws/lambda/panther-policy-engine
View JSON	arn:aws:logs:us-east-1:567780804328:log-group:/aws/lambda/panther-rules-engine*	567780804328	arn:aws:logs:us-east-1:567780804328:log-group:/aws/lambda/panther-rules-engine*	/aws/lambda/panther-rules-engine
View JSON	arn:aws:logs:us-east-1:567780804328:log-group:/aws/lambda/panther-policy-engine*	567780804328	arn:aws:logs:us-east-1:567780804328:log-group:/aws/lambda/panther-policy-engine*	/aws/lambda/panther-policy-engine

Documentation  
Support

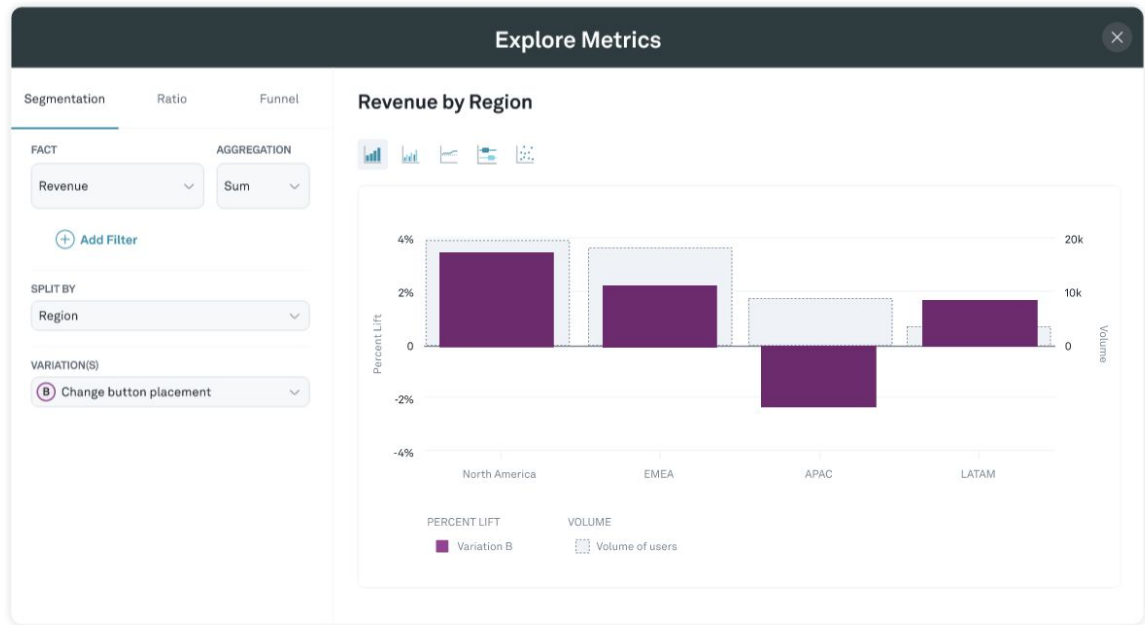
Panther

Security



Eppo

## A/B Testing



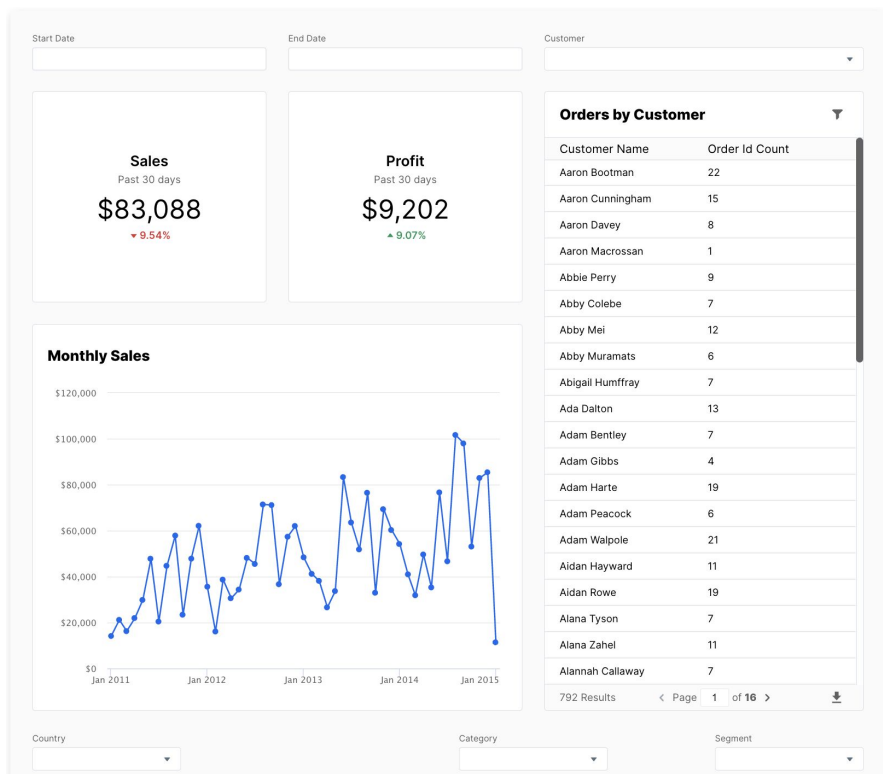


# EXPLO

# > hyperquery

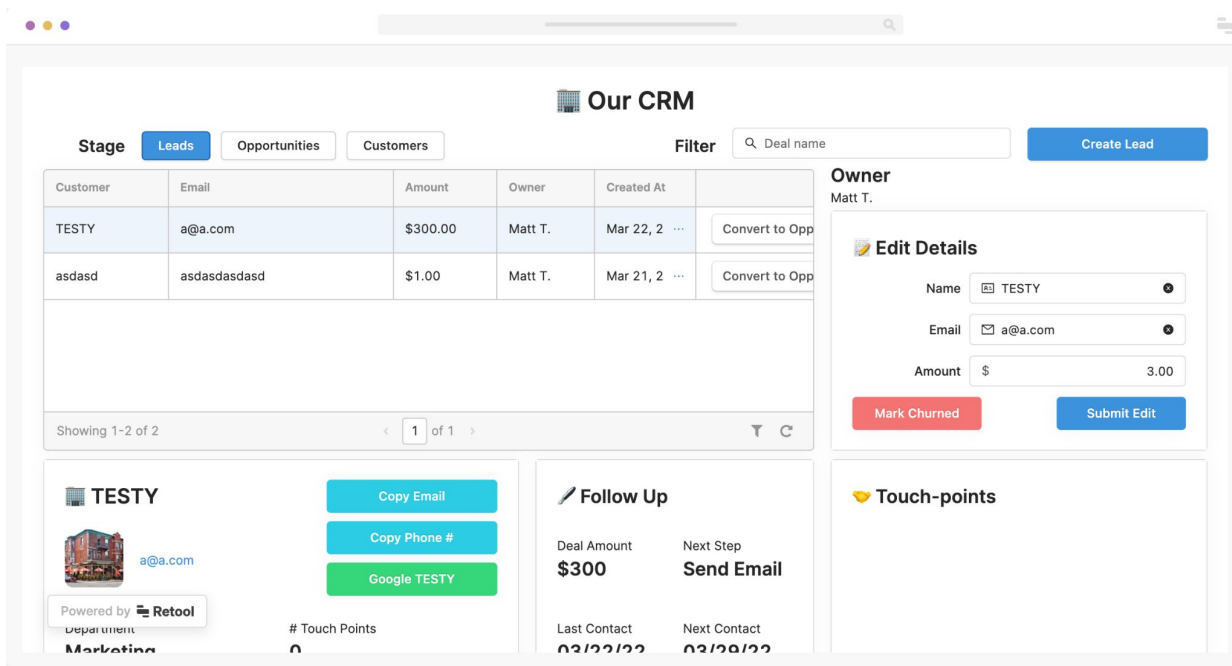
## Explo & Hyperquery

## Documents from BI



# Where we could go with data apps

- CRM on the data warehouse
- Customer support or success on the data warehouse

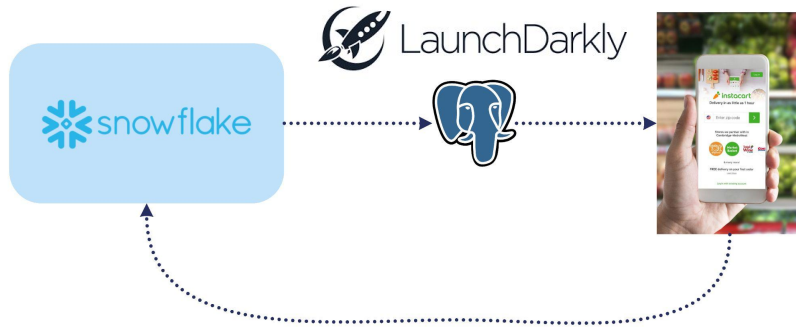


# Where we could go with data apps

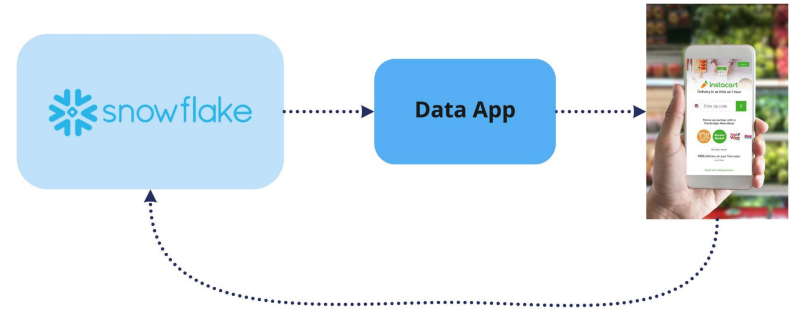
- Metrics Layer on the data warehouse (served in real time)  
(LookML but single place to define aggregations)
- Feature Stores
- Serving data science models for things like in-app personalization
- ML related applications like Data Robot

# Where we could go with data apps

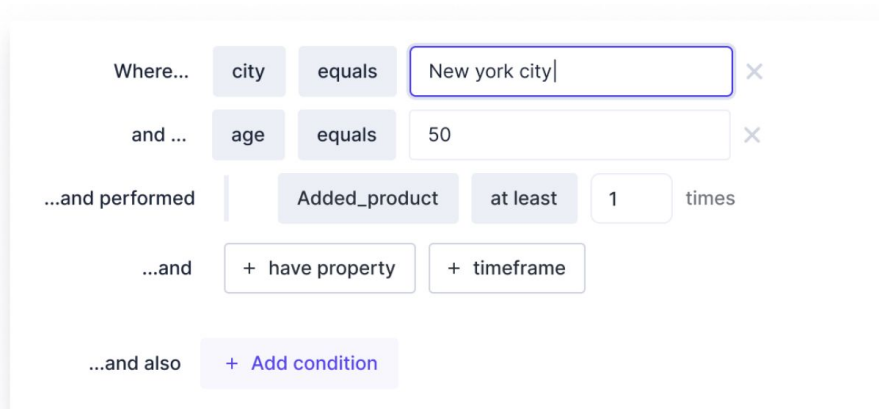
**Old Way:**



**New Way:**



# Hightouch Audiences + Metrics Layer



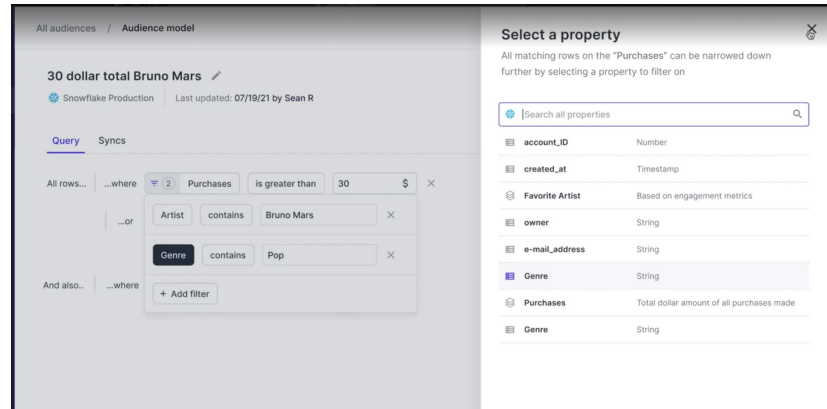
Where... city equals New york city

and ... age equals 50

...and performed Added\_product at least 1 times

...and + have property + timeframe

...and also + Add condition



All audiences / Audience model

30 dollar total Bruno Mars

Snowflake Production Last updated: 07/19/21 by Sean R

Query Syncs

All rows... where Purchases is greater than 30 \$

Artist contains Bruno Mars

Genre contains Pop

And also... where + Add filter

Select a property

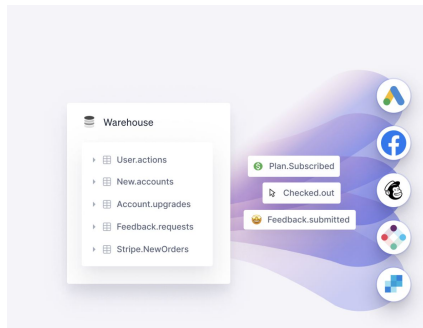
All matching rows on the "Purchases" can be narrowed down further by selecting a property to filter on

Search all properties

account_ID	Number
created_at	Timestamp
Favorite Artist	Based on engagement metrics
owner	String
e-mail_address	String
Genre	String
Purchases	Total dollar amount of all purchases made
Genre	String

## Hightouch Audiences:

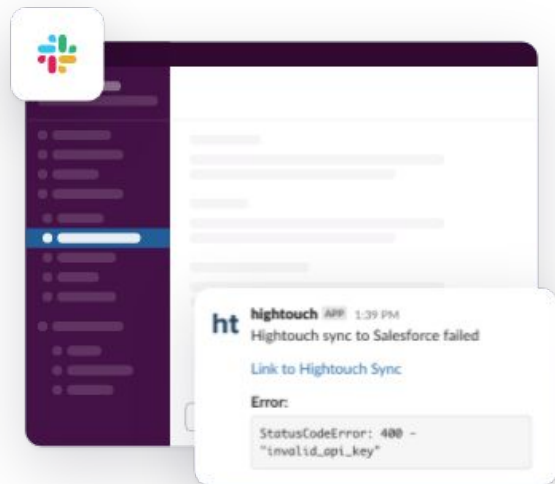
Visual UI for marketers to create marketing audiences on top of warehouse (ex: users who abandoned cart)



## Hightouch Traits:

Expose common metrics/traits for business stakeholders to use (ex: Sum of items bought)

# Hightouch Audiences + Metrics Layer



## Hightouch Notify:

Use data in your warehouse to notify customers or internal stakeholders (ex: Slack channel for new customer signups)



# Ecosystem enablers



# When will this be possible? Is it even practical?

- We recognize that data apps without constraint are quite idealistic
- There will never be a full transition to a data apps framework - thats why reverse ETL is needed
- In some instances, data apps should work. Especially when the user cannot write new data quickly
- Today, data apps are more of a “I wish I had this” rather than “I can have this” but their scope increase over time with streamlit, streaming reads from snowflake, and with tools like Hex and Hightouch.



**Thank you!**

kashish@hightouch.io

ht