SNOWFLAKE PRODUCT ROADMAP

DTCI

Last updated: September 2020

Grace Johnson, Account Executive
Rachel Blum, Sales Engineering
Joe Cramer, Sales Engineering

BE A PART OF THE SNOWFLAKE COMMUNITY!

Slack Channel: #disney-snowflakeinc

Q&A Forum

Join experts on StackOverflow to ask questions, discuss projects, and help each other.

Documentation

Access the latest blogs, tutorials, and product documentation to get the most out of Snowflake.

Support

Need to contact Support? Submit a new case using your Snowflake Community account.

Education & Certification

Join our instructor-led, virtual hands-on labs or get SnowPro certified and stand out in the data community.

[Snowflake Webinars & Hands-on Labs]

[BUILD: Data Applications Summit for App Dev]

MODERN DATA ARCHITECTURE WITH SNOWFLAKE CLOUD DATA PLATFORM

DATA SOURCES

OLTP DATABASES

ENTERPRISE APPLICATIONS

THIRD-PARTY

WEB/LOG DATA

IoT

Google Cloud aws Azure

DATA CONSUMERS

DATA MONETIZATION

OPERATIONAL REPORTING

AD HOC ANALYSIS

REAL-TIME ANALYTICS

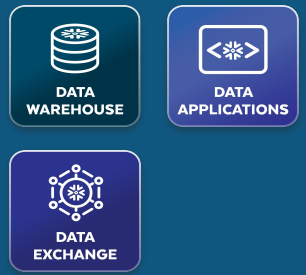
ROADMAP PILLARS

GLOBAL SNOWFLAKE

CORE PLATFORM LEADERSHIP

EXTENSIBLE DATA PIPELINES

DATA CLOUD CONTENT



GLOBAL SNOWFLAKE

GLOBAL SNOWFLAKE

Available on customer's cloud & region of choice

Generally Available

US West (Oregon)
US East (N. Virginia)
Europe (Frankfurt)
Europe (Ireland)
Asia Pacific (Sydney)
Asia Pacific (Singapore)
Canada Central (Montreal)
US East (Ohio)
Japan (Tokyo)
AWS India (Mumbai)
FIPS-ready region in US East (N. Virgina)

Generally Available

East US 2 (Virginia)
West Europe (Netherlands)
Australia East (New South Wales)
US Government (Virginia)
Canada Central (Toronto)
Southeast Asia (Singapore)
West US 2 (Washington)
Switzerland North (Zurich)

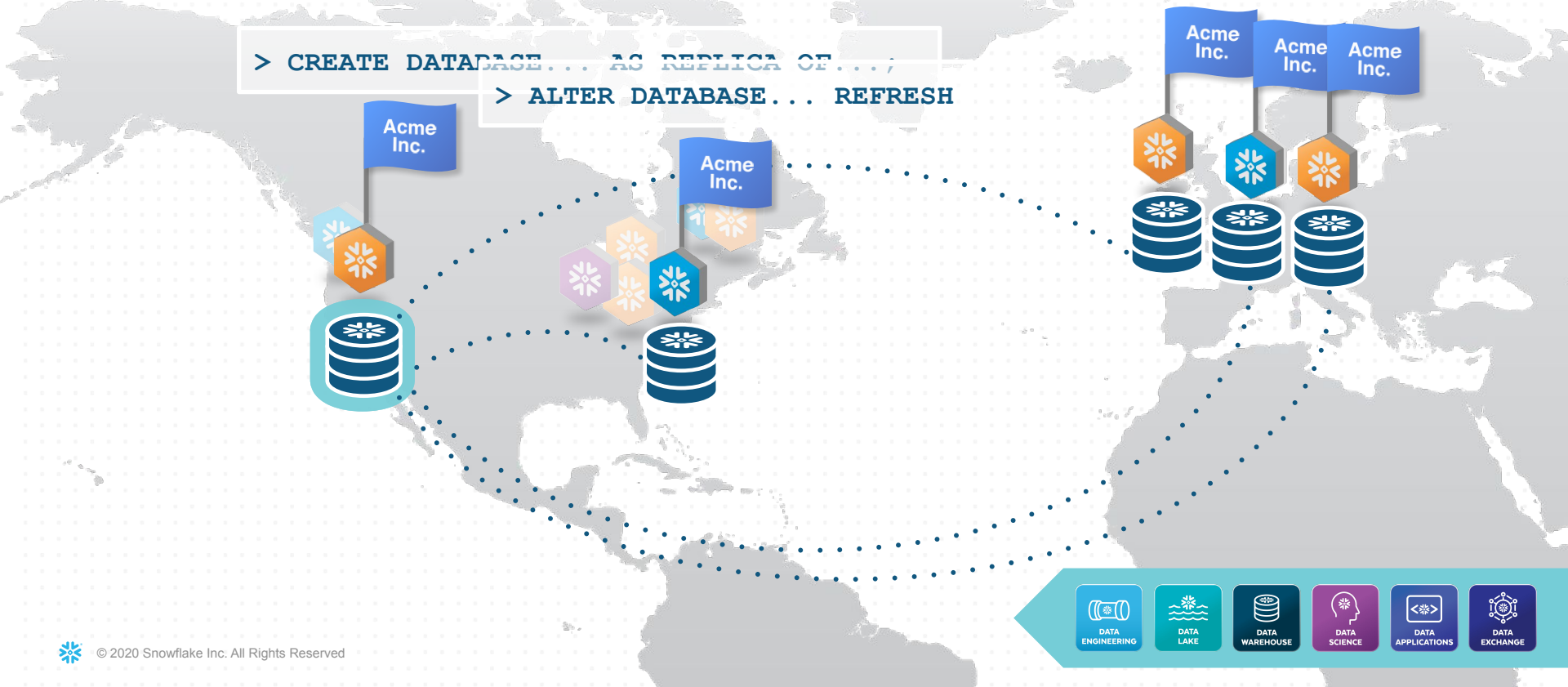
Generally Available

US Central 1 (Iowa)
Europe West 4 (Netherlands)
Europe West 2 (London)

REPLICATION

For business continuity during regional or cloud-provider outages

```
> CREATE DATABASE... AS REPLICA OF...;
> ALTER DATABASE... REFRESH
```



REPLICATION

Roadmap and differentiation

Competitive Alternatives

aws

Azure

Geo-backups with time-consuming recovery

Google Cloud Platform

Managed DR experience, no customer control over recovery point or time

Snowflake Differentiation

- Cross-Cloud Replication
- Active Secondary Databases for read-only workloads
- Instant Failover & Recovery

Roadmap

H2 2020
Account replication & Client failover

- Replicate & failover all account objects
- Failover client connections transparently

ACCOUNT MANAGEMENT

Self service account creation and management (GA in H2)

Single place to manage all accounts, across all regions and clouds:

- View all your company's accounts in one place
- Create and provision new accounts in any region
- Query consolidated usage information for all accounts

Use for:

- Separation of BUs
- Separate test / prod accounts
- Replication across different regions / clouds

CORE PLATFORM LEADERSHIP

SNOWSIGHT

Understand and analyze your data directly from the Snowflake UI

Fast and Responsive Querying
A fast, desktop-quality editor in the web

Smart Autocomplete
Contextual suggestions based on your query and SQL dialect, such as aliases and functions

Write Less SQL
Use a date picker to simplify date selection

Interactive Results
Preview data fast no matter how many rows a query returns

Automatic Stats
Interactive stats for all columns help you catch errors and spot trends without follow-up queries

Beautiful Charts
Create charts that look great on any device at any size

Modern Dashboards
Simple drag and drop interface for creating dashboards

Dynamic Data Filters
Use parameters to set dates, customers and more

Share Your Work
Private Links
Send colleagues queries or dashboards with a link to view, run, or edit the contents

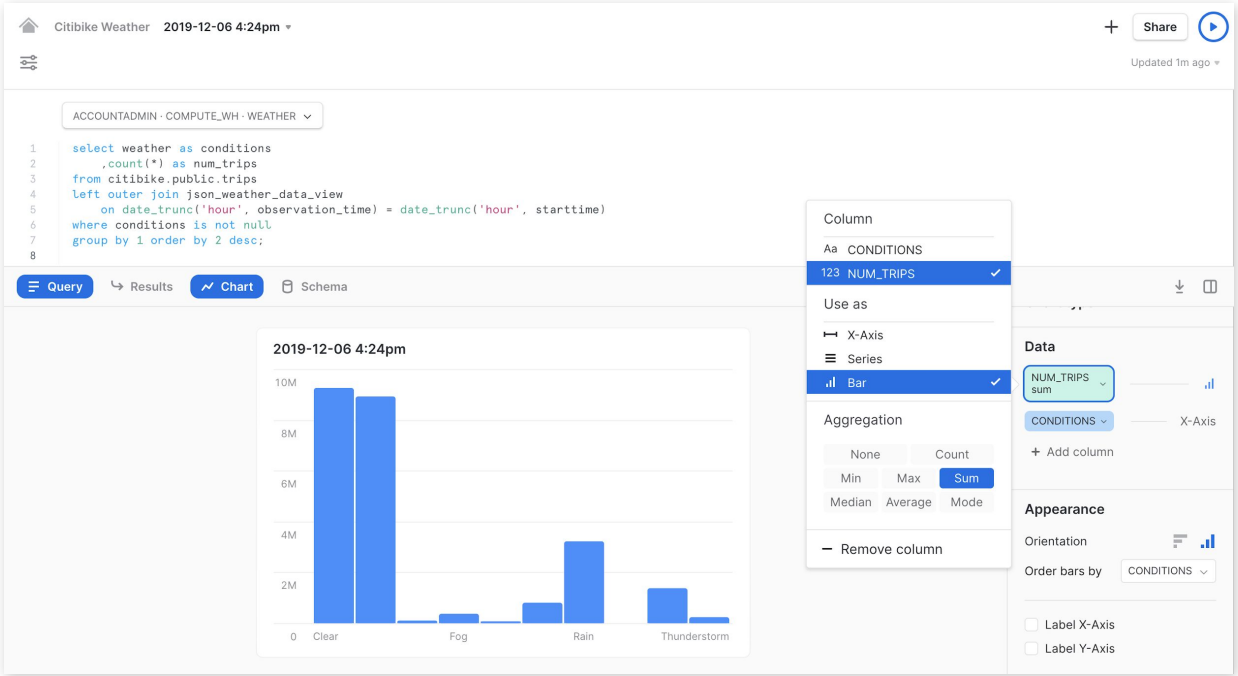
SNOWSIGHT

Interactive SQL editor with charts

Productivity

Collaboration

Visualization



DATA ENGINEERING | DATA LAKE | DATA WAREHOUSE | DATA SCIENCE

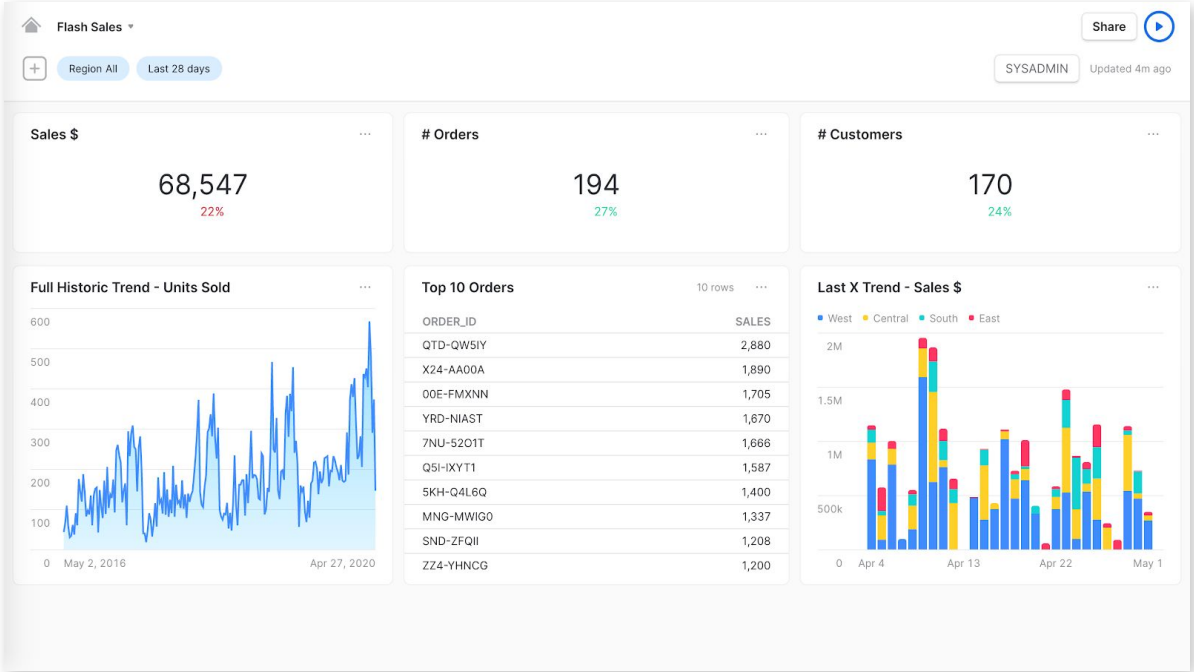
SNOWSIGHT

Dashboards

Productivity

Collaboration

Visualization



DATA ENGINEERING | DATA LAKE | DATA WAREHOUSE | DATA SCIENCE

NEW ADMIN UI

Upgraded experience for configuring, administering, and monitoring Snowflake

Databases and Shares

Manage everything about your data from databases down to individual securable objects

Warehouses

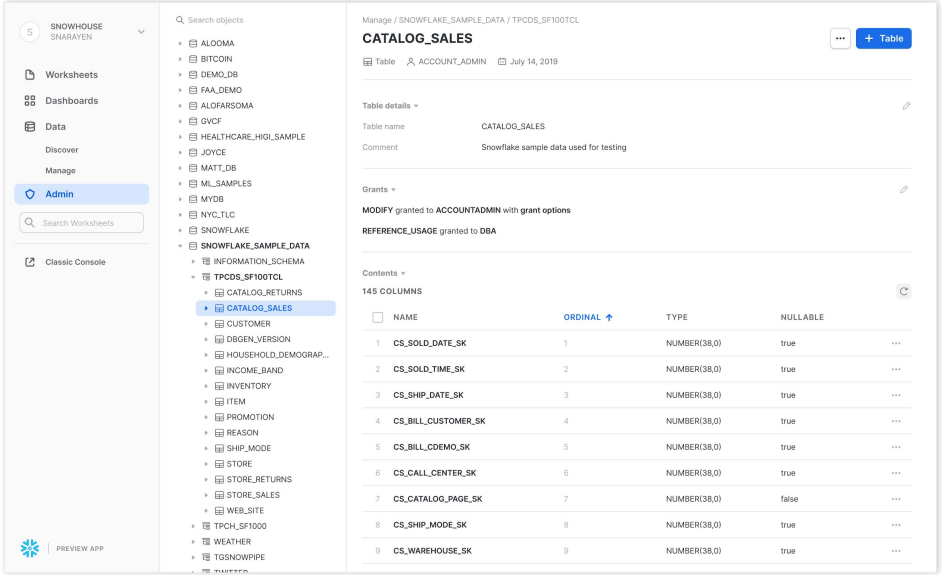
Easily configure your warehouses and monitor performance

Query History

Visualize, search, and analyze your full query history and performance

Account

Monitor your usage, administer users, and manage your RBAC configuration



DATA ENGINEERING | DATA LAKE | DATA WAREHOUSE | DATA SCIENCE

RICHER SQL LANGUAGE

Features

Data Types

- Geography
- Interval

Functions

- Row-based pattern matching through MATCH_RECOGNIZE function
- Migration compatibility

Complex Queries

- Enhanced support for subqueries

SQL-Based Procedural Language

(preview in Q3)

```
CREATE OR REPLACE PROCEDURE PROC (inArg NUMBER)
RETURNS variant
LANGUAGE SQL
DECLARE
  Var_a number;
BEGIN
  SELECT c1 INTO var_a FROM T1;
...
```

SQL

Geospatial support

Geography Type (Public Preview Q2)

- Store and analyze POINTs, LINESTRINGs, POLYGONs
- Input/Output as GeoJSON, Well-Known Text (WKT), and Well-Known Binary (WKB)
- Spherical model of earth

OGC-Compliant Functions

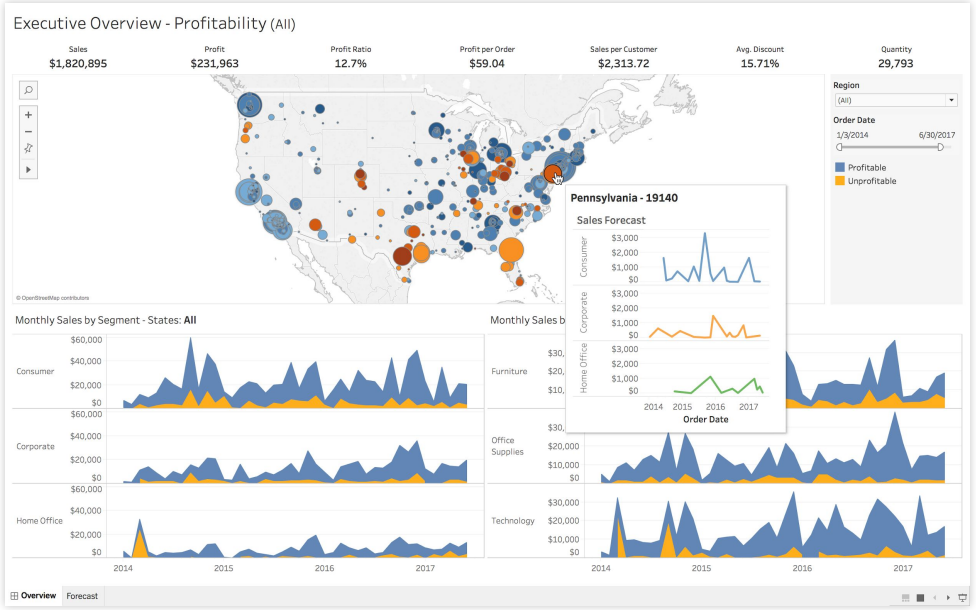
- CONTAINS, INTERSECTS, DISTANCE, DWITHIN, and more

Performance

- Pruning and joins on geospatial predicates (Q3)

Ecosystem

- Integrates with BI tools for visualization
- Works with spatial ETL tools for data integration



DATA ENGINEERING | DATA LAKE | DATA WAREHOUSE | DATA SCIENCE | DATA APPLICATIONS

PERFORMANCE

SQL

Performance

- Improved join order selection
- Cutoff for expensive phases of compilation
- New version of internal file format to improve compression and reduce IO
- Support for 5XL and 6XL warehouses
- Improved query initialization time

16% Reduction year / year

42% Reduction year / year

4,400 Reduction compute hours / year

COMPILATION

CLOUD SERVICES

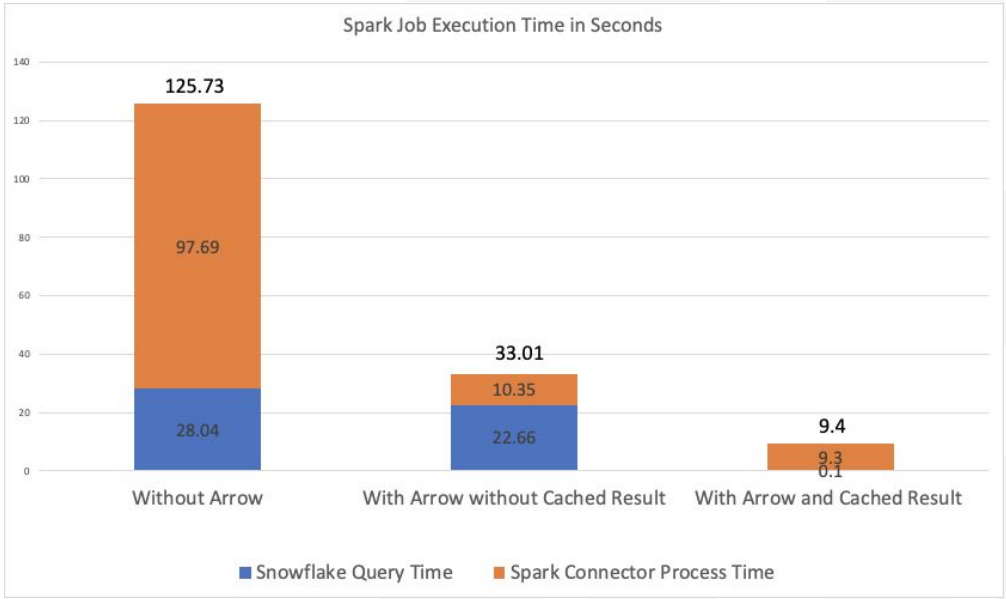
EXECUTION

PERFORMANCE

Spark & Python Connector Improvements

SPARK VERSION 2.6 TURBOCHARGES READS WITH APACHE ARROW

- Apache Arrow columnar result format to dramatically improve query read performance from Snowflake
- Previously, the Spark Connector would first execute a query and copy the result set to a stage in either CSV or JSON format
- With this 2.6.0 release, the Snowflake Spark Connector executes the query directly via JDBC and (de)serializes the data using Arrow
- Benchmark shows massive improvement on end-to-end processing with the Snowflake Spark Connector



https://www.snowflake.com/blog/snowflake-connector-for-spark-version-2-6-turbocharges-reads-with-apache-arrow/

PERFORMANCE

Spark & Python Connector Improvements

If you work with Pandas DataFrames, the performance is even better with the introduction of our new Python APIs, which d**ownload result sets directly into a Pandas DataFrame**. Internal tests show an improvement of up to 5x for fetching result sets over these clients, and up to a 10x improvement if you download directly into a Pandas DataFrame using the new Python client APIs.

Spark Connector Libraries

Scala

- Snowflake Connector for Spark Version 2.6 Turbocharges Reads with Apache Arrow

SparkSQL

- Spark Release

Python/Pandas

- Fetching Query Results From Snowflake Just Got a Lot Faster With Apache Arrow
- Using Pandas DataFrames with the Python Connector — Snowflake Documentation

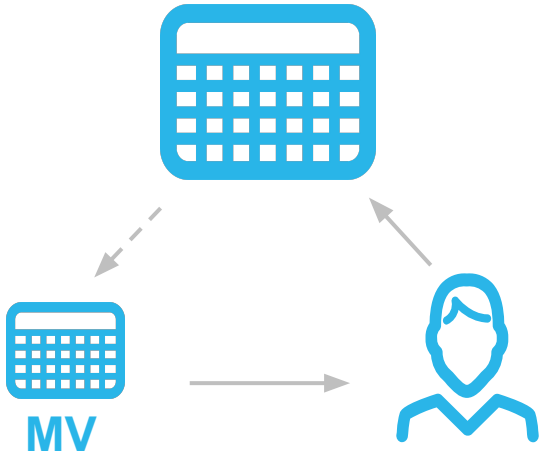
User Guide

PERFORMANCE

Query Rewrites to MVs

Materialized Views

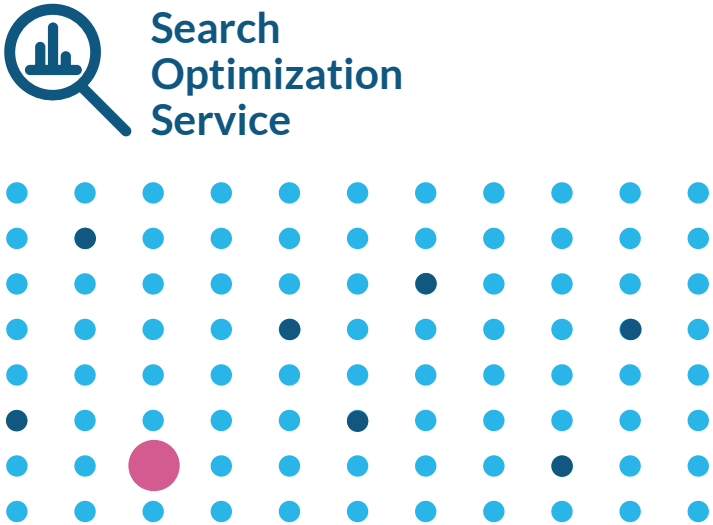
- **Ease:** Automatic query rewrite/view matching
  (In Public Preview)
- **Efficiency** improvements for MV maintenance
  (decoupling MV refresh from garbage collection)
- Highlight **benefits** seen by the workload via MV use
  (planned for Q3)



Transparent Materialized View Usage

PERFORMANCE

Search optimization service



SELECT * **FROM** T **WHERE** C_1 **=** <constant> **AND** … C_n **=** <constant>

Optimize **performance** for selective point lookups(c_{1-n}= constant) on large (TB+) tables

Efficient data structure (search access path in the background) managed by Snowflake in a **serverless fashion**

Equality predicates / equality searches supported in V1 version

Support of fixed data types

- Numbers
- Date, Time
- Strings (exact match)

Future work: Support for text search (LIKE|ILIKE) with wildcard and JSON / Variant data type

SEARCH OPTIMIZATION

Account usage & metadata

Account Usage Views (serverless features)

- AUTOMATIC_CLUSTERING_HISTORY
- MATERIALIZED_VIEW_REFRESH_HISTORY
- PIPE_USAGE_HISTORY
- REPLICATION_USAGE_HISTORY
- → **SEARCH_OPTIMIZATION_HISTORY**
  (new view available for PuPr)

Information Schema & Show Statement

- PrPr: New column added **SEARCH_OPTIMIZATION** to existing INFORMATION_SCHEMA.TABLES
- PrPr: New information schema **SEARCH_OPTIMIZATION_HISTORY**
- Expose additional storage used in corresponding table information schema

SEARCH_OPTIMIZATION_HISTORY (for PuPr)

Column Name	Data Type	Description
START_TIME	TIMESTAMP_LTZ	Start of specified time range
END_TIME	TIMESTAMP_LTZ	End of specified time range
CREDITS_USED	TEXT	Number of credits billed during window above
TABLE_ID	NUMBER	Internal system generated identifier for table
TABLE_NAME	TEXT	Name of the table
SCHEMA_ID	NUMBER	Internal system generated identifier for schema
SCHEMA_NAME	TEXT	Name of the schema
DATABASE_ID	NUMBER	Internal system generated identifier for database
DATABASE_NAME	TEXT	Name of the database

OPTIMIZATION TECHNIQUES

Clustering Metrics

snowflake

How Snowflake Handles Reclustering

- In general there are two ways to recluster: full and incremental

- Full reclustering - Reload Order By
  - Fully sorted - expensive
  - Impractical for large tables
  - Cannot keep up with DML

- Incremental approximate reclustering
  - Only recluster the part that's most needed
  - Can keep up with DML
  - Good enough for pruning

- ***Snowflake's clustering service performs incremental clustering***
  - Trade-off between pruning benefit and cost

Clustering Recap

- Snowflake allows you to define clustering keys, which indicate the intent of how the table should be clustered
- Clustering keys facilitate reorganizing table from natural clustering (ingest) to a specific clustering order
- The Automatic Clustering Service clusters data according to the clustering key
- Automatic Clustering is the Snowflake service that seamlessly and continually manages all reclustering, as needed, of clustered tables.
- After a clustered table is defined, reclustering does not necessarily start immediately. Snowflake only reclusters a clustered table if it will benefit from the operation.

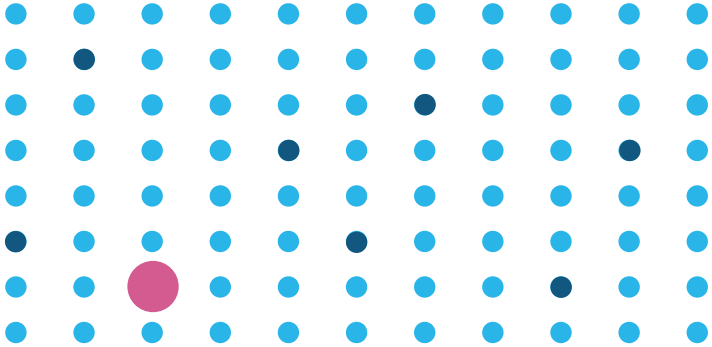
Search Optimization Service

snowflake

Search Optimization

- Aims to significantly improve the performance of selective point lookup queries on large tables. A point lookup query returns only one or a small number of distinct rows.
- Use case examples include:
  - Business users who need fast response times for critical dashboards with highly selective filters.
  - Data scientists who are exploring large data volumes and looking for specific subsets of data.
- A user can register one or more tables to the search optimization service.
- Search optimization is a table-level property and applies to all columns with supported data types (see the list of supported data types further below).

Search Optimization Service



SELECT * **FROM** T **WHERE** C_1 **=** <constant> **AND** … C_n **=** <constant>

Optimize **performance** for selective point lookups(c_{1-n}= constant) on large (TB+) tables

Efficient data structure (search access path in the background) managed by Snowflake in a **serverless fashion**

Equality predicates / equality searches supported in V1 version

Support of fixed data types

- Numbers
- Date, Time
- Strings (exact match)

Support for JSON / Variant for GA

DATA LAKE | DATA WAREHOUSE | DATA APPLICATIONS

When to use *what* optimization feature?

Feature	Use cases to be considered	Add'l notes
Automatic (Data) Clustering	If you want to speed up range and equality searches on single column or composite key	Table is primarily clustered by first column in the clustering scheme
Search Optimization Service	If you want to speed up equality searches on column(s) other than the primary key	At least one of the equality predicates on a column with high number of distinct values (NDVs), 'rule of thumb: > 100-200k' (counter example: boolean)

Materialized Views

snowflake

When to Use MVs

- Query results typically contain a small number of rows and/or columns relative to the base table
- Query results contain results that require significant processing, including:
    - **Analysis of semi-structured data.**
    - **Aggregates that take a long time to calculate.**
- Materialized views can be used to improve the performance of queries that use the same subquery results repeatedly.
- Because materialized views are automatically and transparently maintained/updated by Snowflake, they provide fast OLAP analysis even on **near real time** ingested data.
- Data accessed through materialized views is always current.

Improvements from Legacy MVs

- They do not need to be actively refreshed to stay current - **updates are automatic**
- There is **no query performance degradation** when the MV is updated as new data is added, changed or deleted in the underlying table
- The Snowflake optimizer can **automatically rewrite queries** to use materialized views, even if those materialized views are not directly referenced in the query.

When to use *what* optimization feature?

Feature	Use cases to be considered	Notes
Automatic (Data) Clustering	If you want to speed up range and equality searches on single column or composite key	Table is primarily clustered by first column in the clustering scheme
Search Optimization Service	If you want to speed up equality searches on column(s) other than the primary key	At least one of the equality predicates on a column with high number of distinct values (NDVs), 'rule of thumb: > 100-200k' (counter example: boolean)
Materialized Views	Speeds up range & equality searches, but for a subset of rows / columns (Speeds up aggregation, variant flatten & more)	Combination MVs with clustering to support different clustering keys

GOVERNANCE AND SECURITY

GOVERNANCE AND SECURITY

Dynamic data masking

Dynamically Mask Protected (PII, PHI) Column Data at Query Time

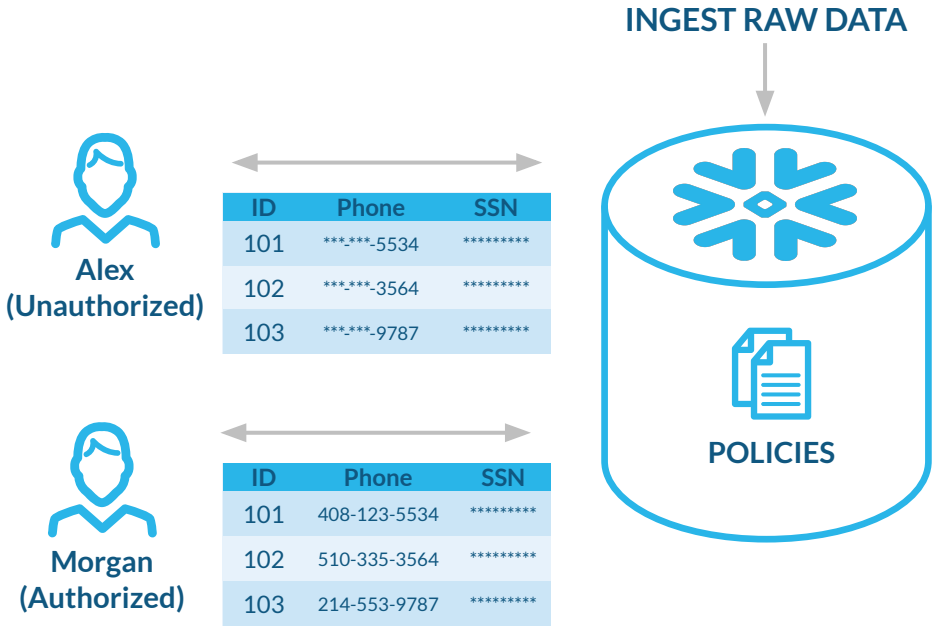
- No change to the stored data
- Mask or partial mask using constant value, hash, and custom functions
- Unmask for authorized users only

Policy Based Control

- Table/View owners and privileged users (such as accountadmin) unauthorized by default
- Centralized policy mgt

Ease of Management

- Apply single policy to multiple columns
- Prevent secure view explosion



DATA ENGINEERING | DATA LAKE | DATA WAREHOUSE | DATA APPLICATIONS

GOVERNANCE AND SECURITY

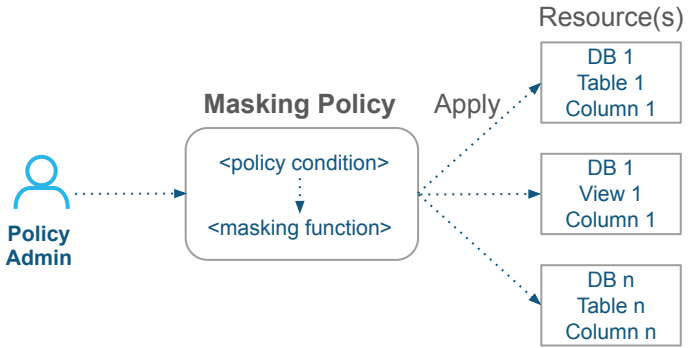
Dynamic data masking policies

Masking Policy

- Policy contains condition(s) and masking function to apply under those conditions
- Policy is applied to one or more table, view, or external table columns in an account
- Nested policy execution for views - policy on table executed before policy on view(s)

Supports

- All data types
- Data sharing
- Streams
- Clone carries over policy associations



Masking Policy Example

Unmask
Partial mask
Mask

```
CASE
  WHEN invoker_role() IN ('pii_role') THEN val
  WHEN invoker_role() IN ('support') THEN
regexp_replace(val,'.+\@','*****@')
  ELSE '********'
END;
```

GOVERNANCE AND SECURITY

External tokenization using third party

Ingest Protected (PII/PHI) Data as Externally Tokenized

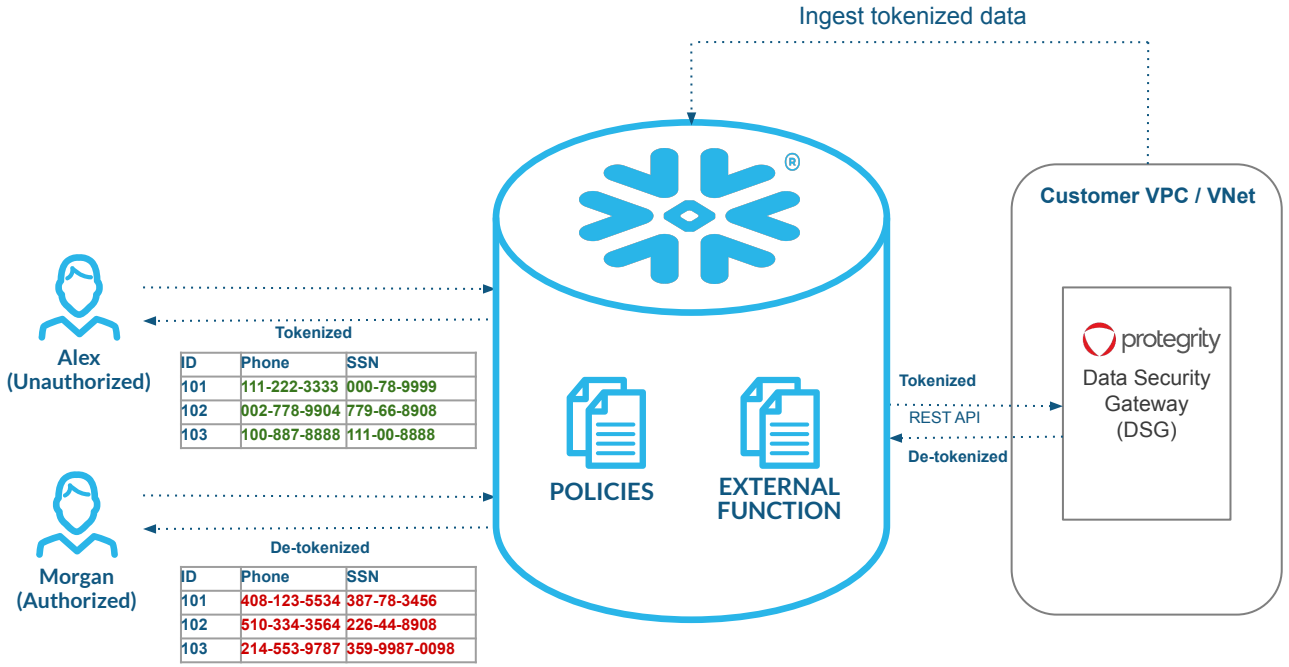
- Using Protegrity agents on ETL tools

De-tokenize for Authorized Users at Query Time

- Protegrity DSG called using external functions to de-tokenize data
- For unauthorized users, Protegrity DSG is not called

Policy Based Control

- Table/View owners and privileged users (such as accountadmin) unauthorized by default
- Centralized policy mgt



DATA ENGINEERING | DATA LAKE | DATA WAREHOUSE | DATA APPLICATIONS

COST MANAGEMENT

Cost Transparency

- **Warehouse Load History:** Improvements to warehouse_load_history function to help you measure per-cluster load (for multi-cluster warehouses) and identify idle clusters (Q2)
- **Warehouse Event Logs:** Logs of warehouse operations and state changes (Q2)
- **Query History:** New stats to help you better understand query performance and behavior such as spilling, pruning, and IO statistics (Available Now)
- **In-App Dashboards:** Faster time-to-insight with pre-built dashboards based on Account Usage data (H2)

Cost Control

- **Resource Monitors:** Integration with serverless features (Snowpipe, Automatic Clustering Service, MV refresh service) (Q2)
- **Alerts Service:** Monitor usage data and trigger alerts using flexible Alert Service. Define the condition and frequency of your alerts, as well as how you want to be notified (e.g. E-mail, Slack, etc.) (H2)

RELEASE PROCESS

Updating Snowflake's Continuous Release Process

We are constantly improving how Snowflake delivers our service in a continuous, uninterrupted fashion. Some of the improvements we're making in 2020:

- Staged Releases - access to releases in test accounts 24 hours ahead of time - Recently GA
- Version information in account usage's query history - monitor for issues by seeing which version of snowflake's software executed each query - Recently GA
- Breaking Changes - updates to how breaking changes are communicated and rolled out (with ability to delay rollout of some changes) - H2

ECOSYSTEM

Client drivers

Feature	Description	Current State	Target
Arrow Format - ODBC, Go	Improved fetch performance	NA	Q3 2020 (GA)
MFA Connection Cache - ODBC, JDBC	Suppress extra MFA requests	NA	Q3 2020 (GA)
Async APIs - Python	Submit statements asynchronously	Private (JDBC)	Q3 2020 (GA)
Multi-statement support - Go	Submit multiple SQL statements in a single request	NA	Q3 2020 (GA)
PHP Driver	Moving from PrPr to GA	Private Preview	Q4 2020 (GA)
SQL API	New lightweight SQL REST API	NA	Q4 2020 (PrPr)

Supported Drivers: JDBC, ODBC, Python, SnowSQL, .NET, Golang

ECOSYSTEM

Partner connectors*

Coming Soon (Highlights)

- Azure: PowerBI SSO Custom Roles
- GCP: Data Flow, Data Fusion, Data Proc, Data Studio
- AWS: Glue crawler

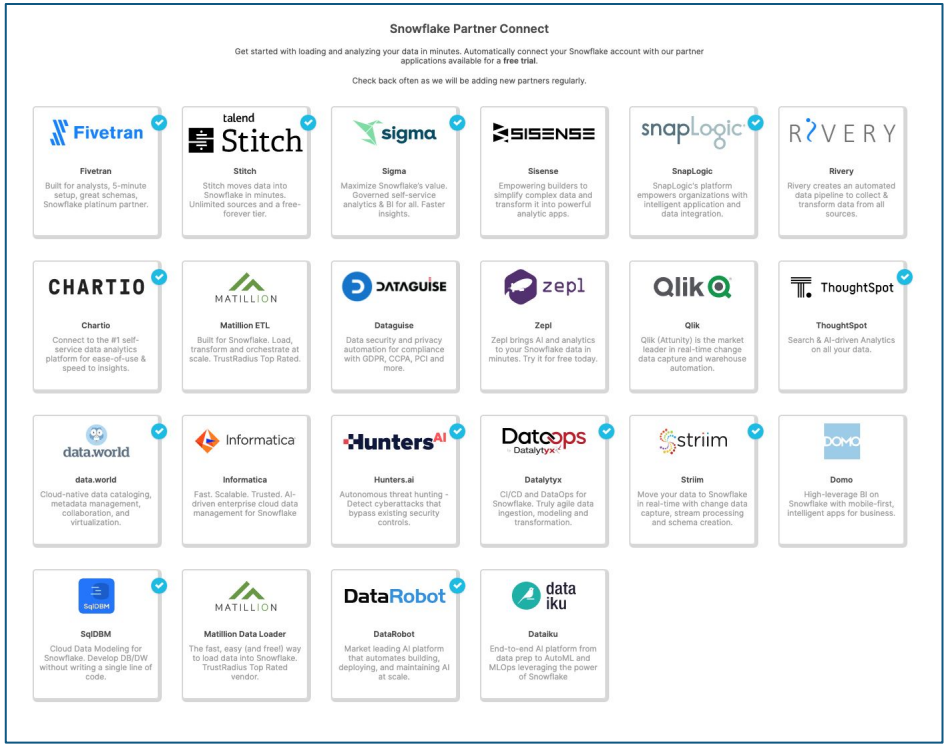
Partner landscape

These connectors are owned and maintained by the Partner. Contact the partner for dates and details

DATA ENGINEERING | DATA LAKE | DATA WAREHOUSE | DATA SCIENCE | DATA APPLICATIONS

ECOSYSTEM

Partner Connect

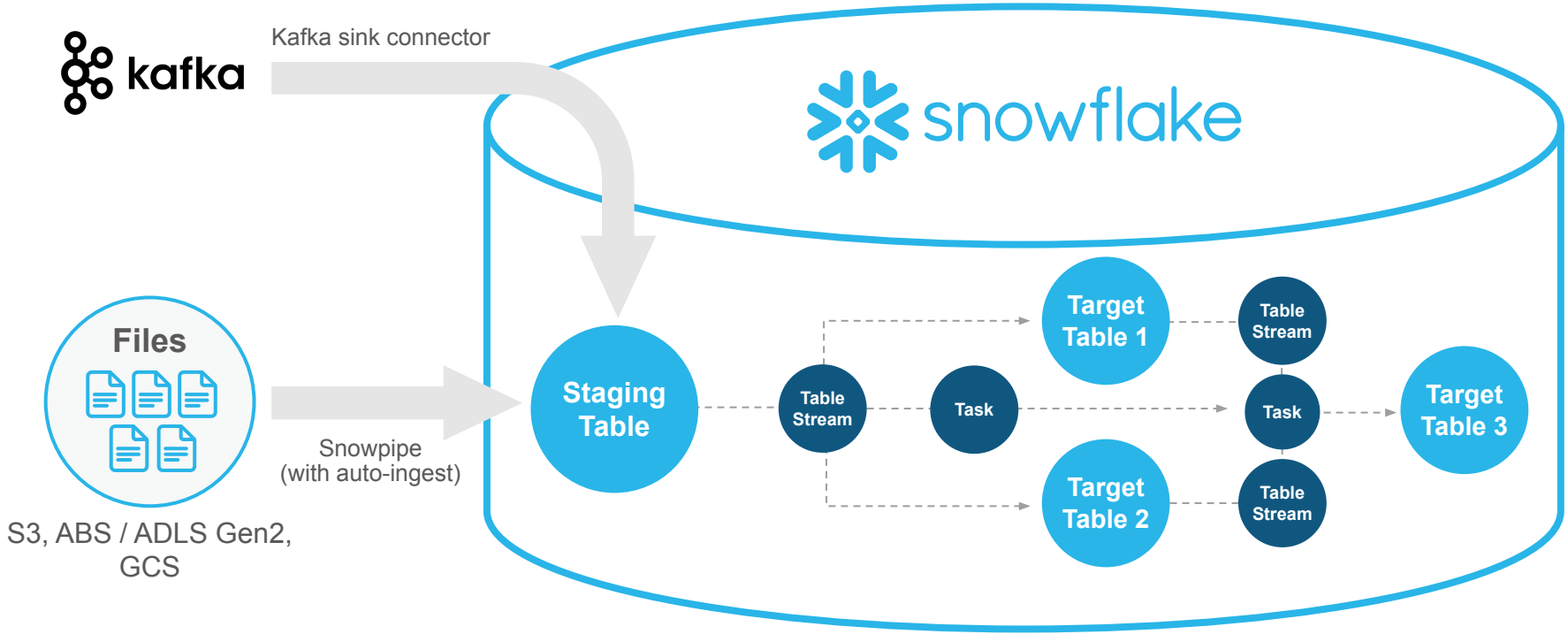


Pipeline:

- H2O
- Tableau Online
- Talend Pipeline Designer
- Trifacta
- Qlik Sense
- DBT

EXTENSIBLE DATA PIPELINES

CONTINUOUS DATA PIPELINES



DATA ENGINEERING | DATA LAKE | DATA WAREHOUSE | DATA SCIENCE | DATA APPLICATIONS | DATA EXCHANGE

RELEASE PLAN (ESTIMATES)

Capability	Release	ETA
Multi-statement, scoped transaction in sprocs (for tasks)	PuPr	Q2 CY 20
Auto-ingest Snowpipe on GCP	PuPr	Q3 CY 20
Cross-cloud auto-ingest to AWS	PuPr	Q3 CY 20
Streams on external tables	PuPr	Q3 CY20
ADLS Gen2 Support (COPY, Snowpipe, Auto-ingest, Auto-refresh)	GA	Q4 CY20

RELEASE PLAN (ESTIMATES)

Capability	Release	ETA
Error notifications for Snowpipe	PrPr (AWS)	Q2 CY 20
Schema detection for Avro, Parquet, ORC (COPY/Snowpipe)	PrPr	Q3 CY 20
Direct invocation of a task w/o schedule	PrPr	Q3 CY 20
Task replication (does NOT include stream replication)	PrPr	Q3 CY 20
Journal Tables	PrPr	Q3 CY 20
COPY validations support for non-csv formats, errors in files	PrPr	Q4 CY 20
DAG support in tasks (i.e. multiple parents)	PrPr	Q4 CY 20
Error notifications for tasks	PrPr	Q4 CY 20
Pipe replication	PrPr	Q4 CY 20

EXTENSIBILITY

External Functions

Bind SQL Functions to Implementations Outside Snowflake

- V1: Batched scalar functions
- Mediated via API Gateway
- Usable wherever functions are used

Example Scenarios:

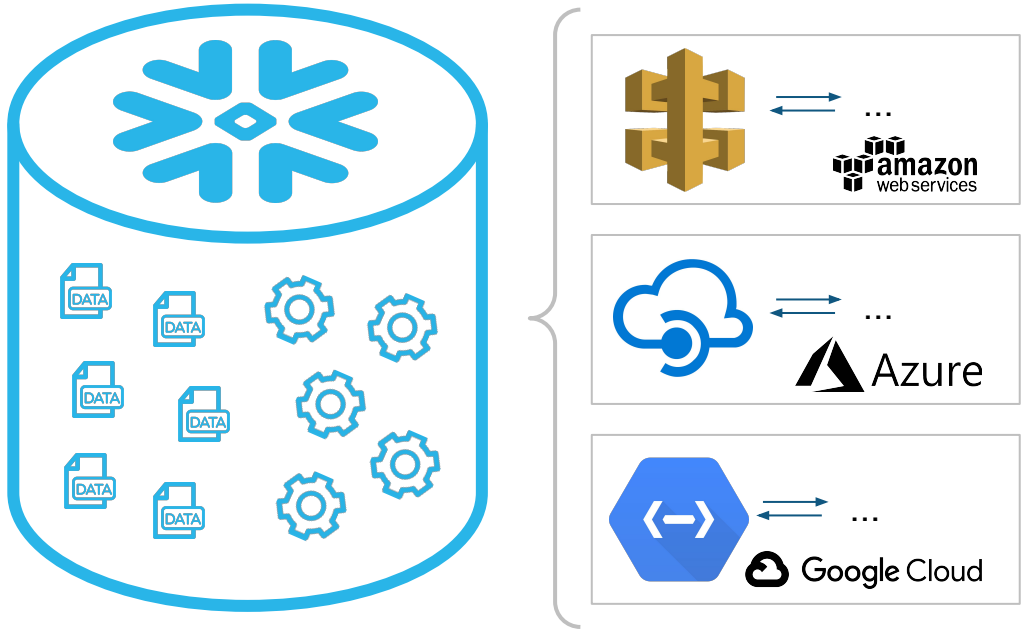
- Custom Lambda
- ML Scoring
- Geocoding

ACCOUNTADMIN in Control of Security

- Must explicitly whitelist endpoints

Public Preview

- AWS only
- Azure and GCP coming soon



EXTENSIBILITY

Java Functions

Bind SQL Functions to Implementations *Inside* Snowflake

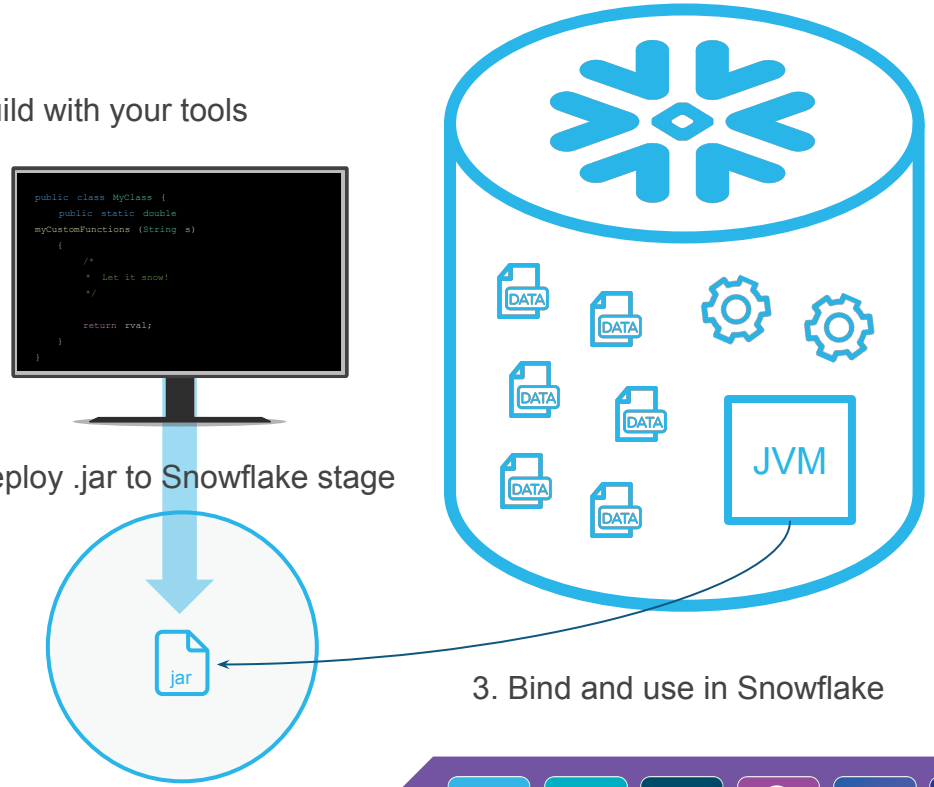
- V1: scalar functions
- Usable wherever functions are used

Example Scenarios:

- ML Scoring
- Apply custom code
- Use third-party libraries

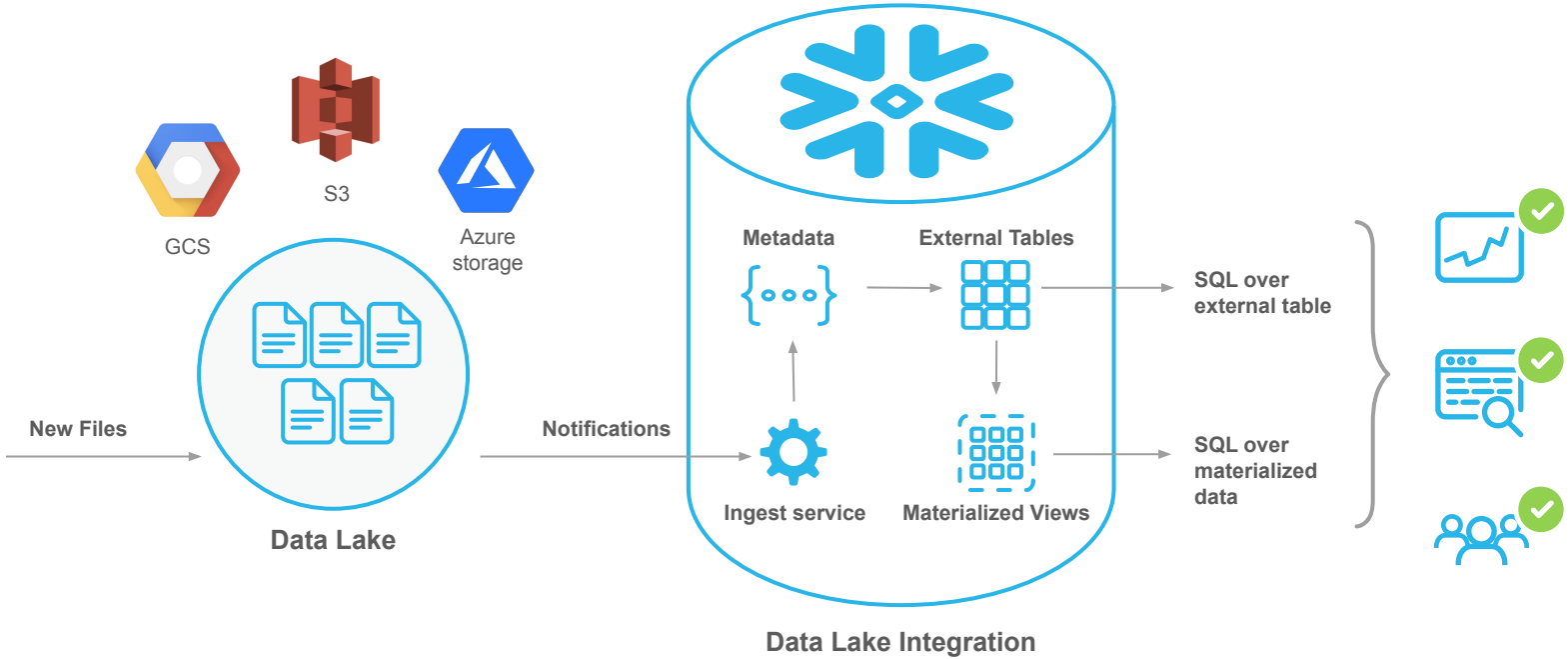
Coming Soon:

- Not yet in preview



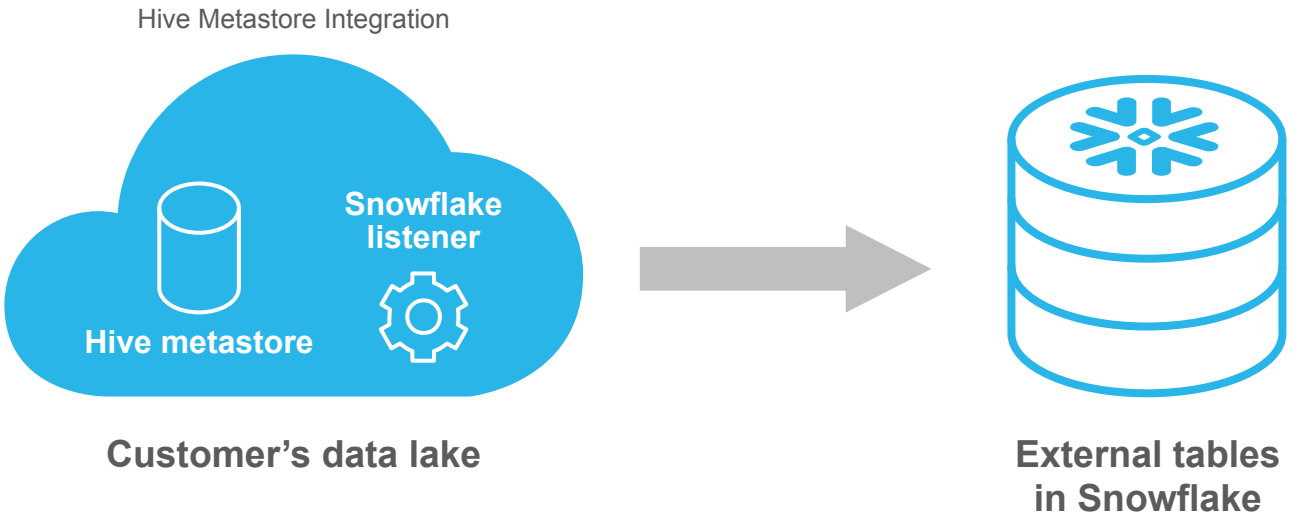
DATA LAKE INTEGRATION

Use external tables + MVs to query data directly from data lake



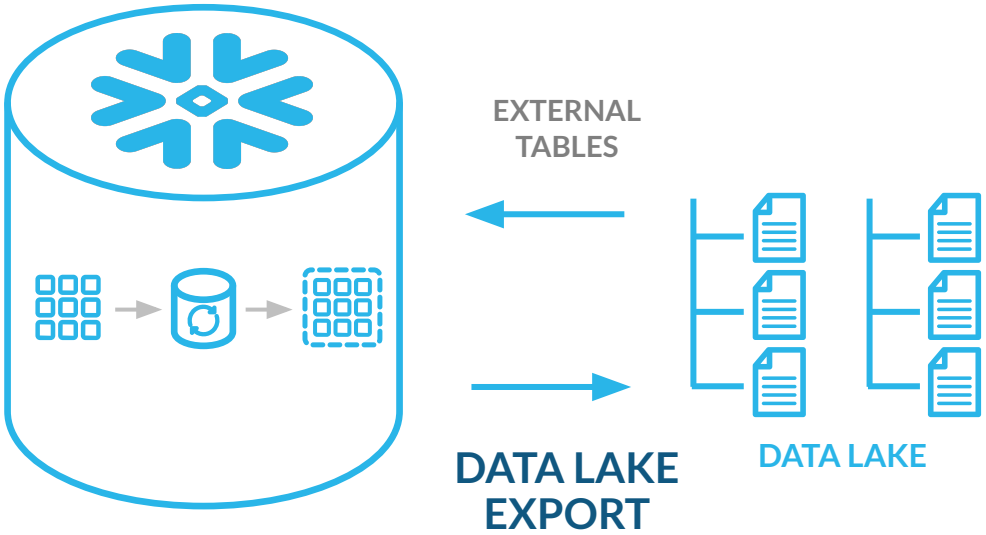
HIVE METASTORE INTEGRATION

Automatically Synchronize Metadata in Hive Metastore with External Tables Schema in Snowflake



DATA LAKE EXPORT

Export data from Snowflake table to files on data lake in partitioned folders

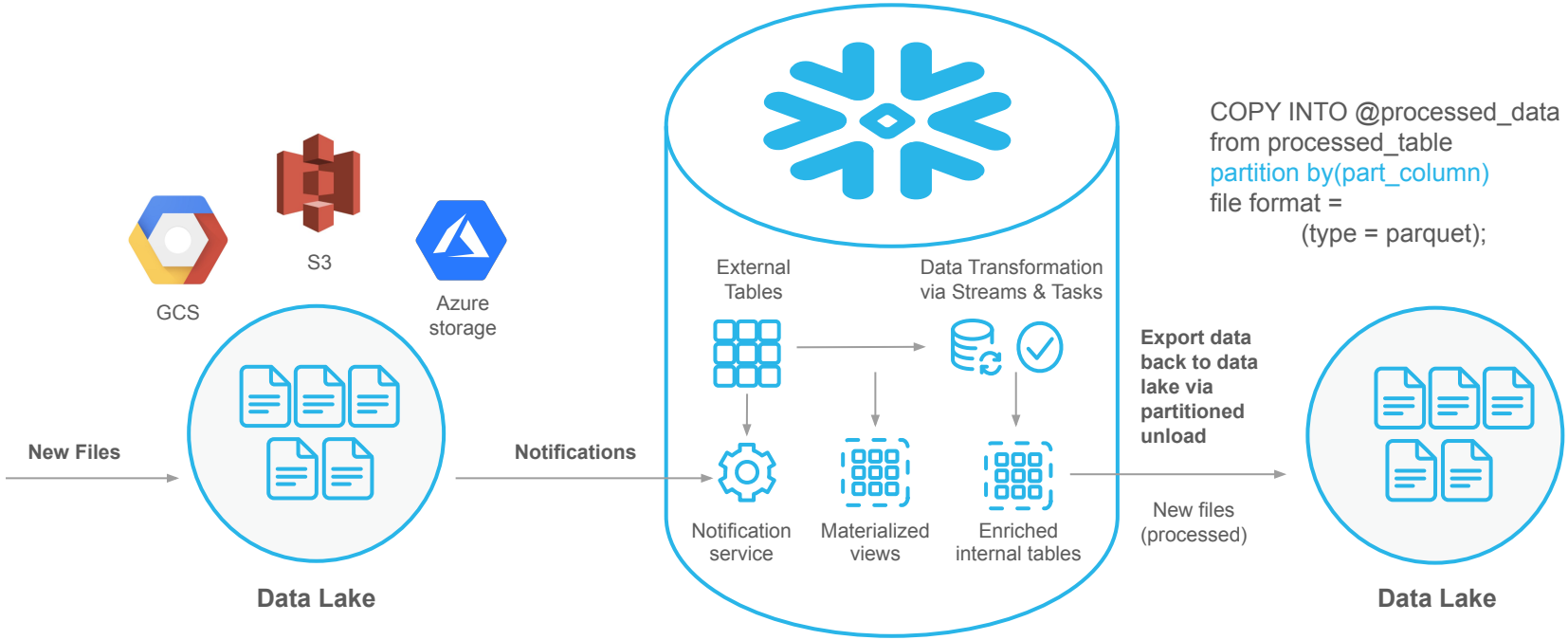



```
COPY INTO @processed_data from
processed_table
partition by('dt='|| date_col)
file format = (type = parquet)
max_file_size = 512000000;
```

- Export large parquet files with 128MB row groups
- Export with hive style partitioning

DATA ENGINEERING WITH EXTERNAL DATA LAKE

Process data lake files using external tables, streams, tasks and export parquet files to data lake




```
COPY INTO @processed_data
from processed_table
partition by(part_column)
file format =
        (type = parquet);
```

DATA ENGINEERING | DATA LAKE | DATA WAREHOUSE

ACCESS UNSTRUCTURED FILES

Get pre-signed URLs of files + metadata from single query

Pre_signed_url	Img_height	Img_width	Img_size
https://my_bucket.aws.amazon.com/images/raw/2018-01-01/file_019229c7-025c-8b78-0000-00000121d295_0_0_0.jpg	768	1024	6.3 MB
https://my_bucket.aws.amazon.com/images/raw/2018-01-01/file_019229c7-025c-8b78-0000-00000121d295_0_0_1.jpg	1440	2010	9.7 MB
....	....		....

External stage

Internal stage

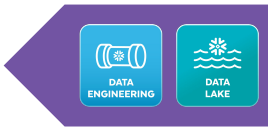
Table

GCS

S3

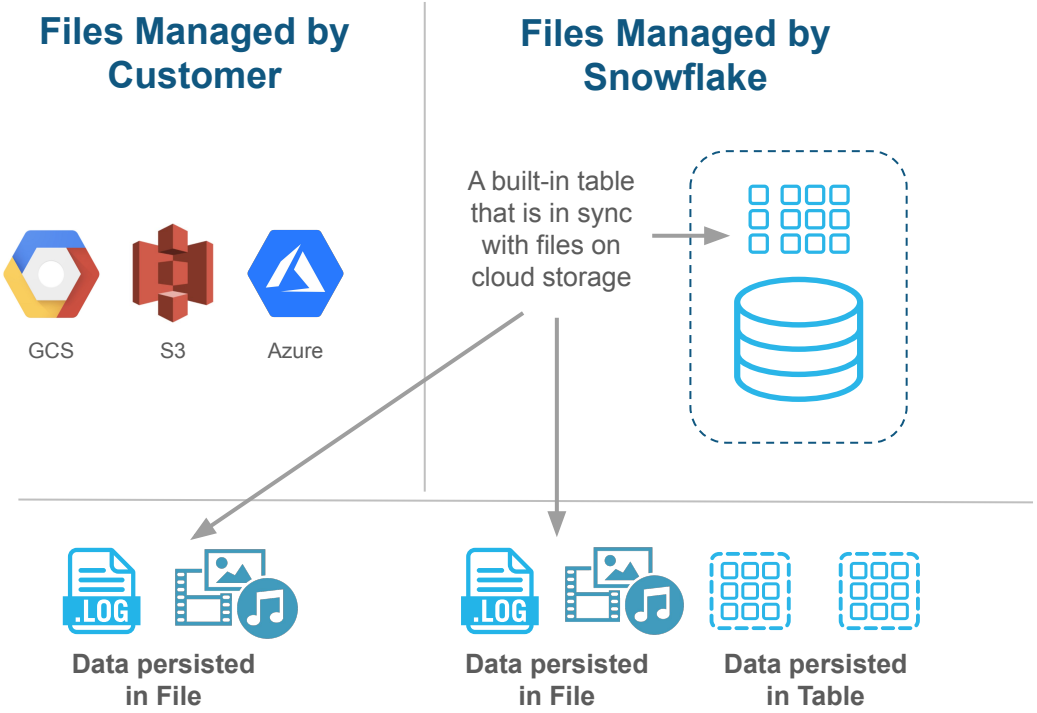
Azure

```
CREATE VIEW images_v AS
SELECT
        img_height,
        img_width,
        img_size,
        get_pre_signed_url(
                @images_stage,
                img_url)
        as pre_signed_url
FROM images_table;
```

BUILT-IN FILE CATALOG IN SNOWFLAKE

Automated sync of file catalog to a built-in table in Snowflake




```
-- internal stage
CREATE STAGE images_stage
store_file_catalog = true;

-- external stage
CREATE STAGE images_stage
url = 's3://my_bucket/images'
storage_integration = st_int
store_file_catalog = true;

-- select files from FILE table
select * from table (files(<stage_name>));
```


DATA LAKE

FILE CATALOG IS AUTOMATICALLY IN SYNC WITH STORAGE

Relative File URL	Last Modified	Size
/bugs/output/partitioned/test_L_3M_2/2018-01-01/merge1_019229c7-025c-8b78-0000-00000121d295_0_0_0.snappy.parquet	Feb 10, 2020 1:29:04 PM GMT-0800	86.3 MB
/bugs/output/partitioned/test_L_3M_2/2018-01-01/merge1_019229c7-025c-8b78-0000-00000121d295_0_0_1.snappy.parquet	Feb 10, 2020 1:29:49 PM GMT-0800	79.7 MB
...	....	....

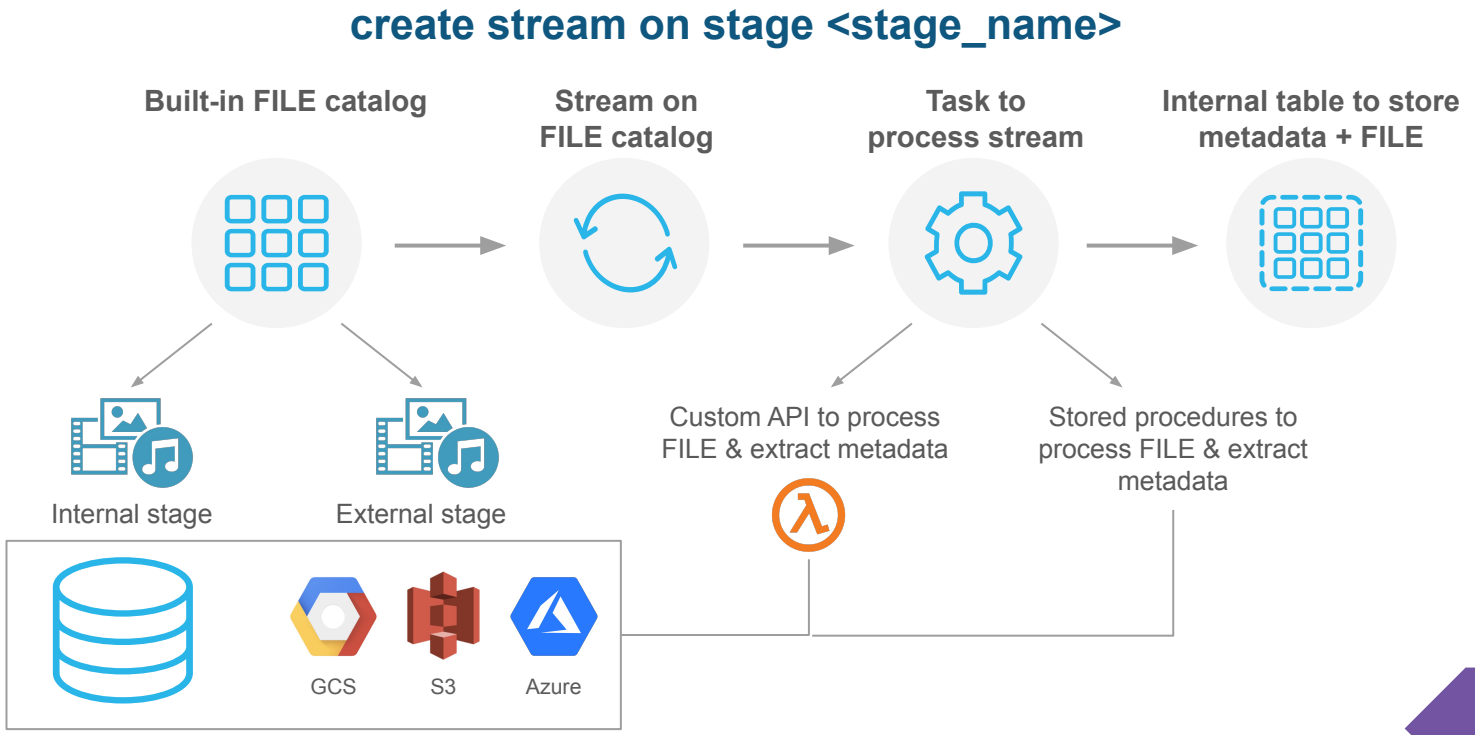
GCS

S3

Azure

Name	Last modified	Size
merge1_019229c7-025c-8b78-0000-00000121d295_0_0_0.snappy.parquet	Feb 10, 2020 1:29:04 PM GMT-0800	86.3 MB
merge1_019229c7-025c-8b78-0000-00000121d295_0_0_1.snappy.parquet	Feb 10, 2020 1:29:49 PM GMT-0800	79.7 MB
merge1_019229c7-025c-8b78-0000-00000121d295_0_0_10.snappy.parquet	Feb 10, 2020 9:51:14 PM GMT-0800	77.0 MB
merge1_019229c7-025c-8b78-0000-00000121d295_0_0_2.snappy.parquet	Feb 10, 2020 1:30:46 PM GMT-0800	85.7 MB
merge1_019229c7-025c-8b78-0000-00000121d295_0_0_3.snappy.parquet	Feb 10, 2020 1:31:23 PM GMT-0800	76.0 MB

BUILD END-TO-END PIPELINE TO PROCESS FILES

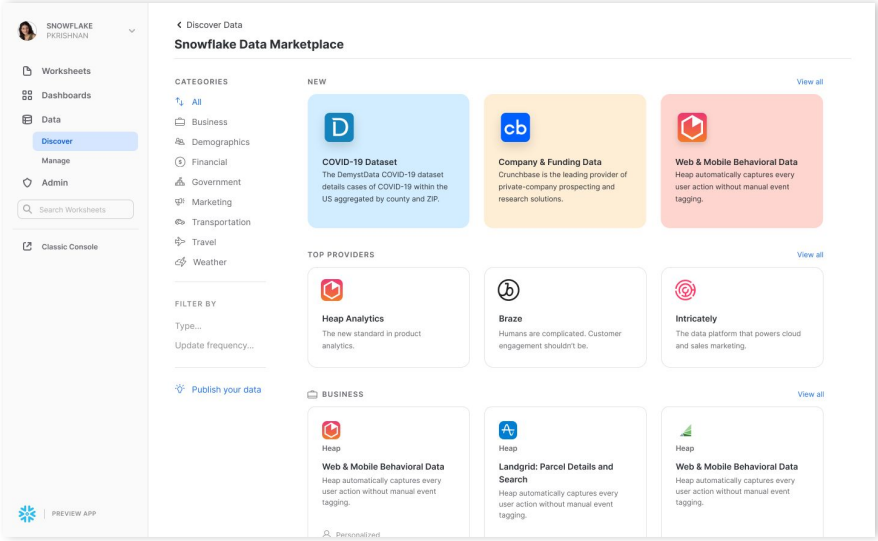


DATA ENGINEERING DATA LAKE

DATA CLOUD CONTENT

SNOWFLAKE DATA MARKETPLACE

Discover external data from ~100 Ddata and SaaS providers and access through Data Sharing

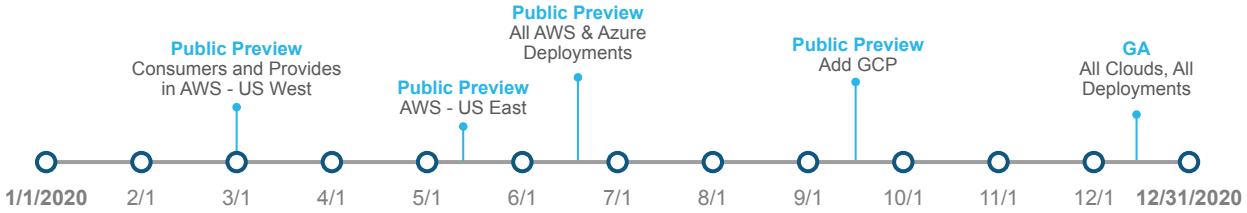


For Data Providers:

- Reach 3,000+ Snowflake customers
- Reduce COGS
- Improve customer experience as customers get access to live data, without ETL
- Provide personalized data feeds

For Data Consumers:

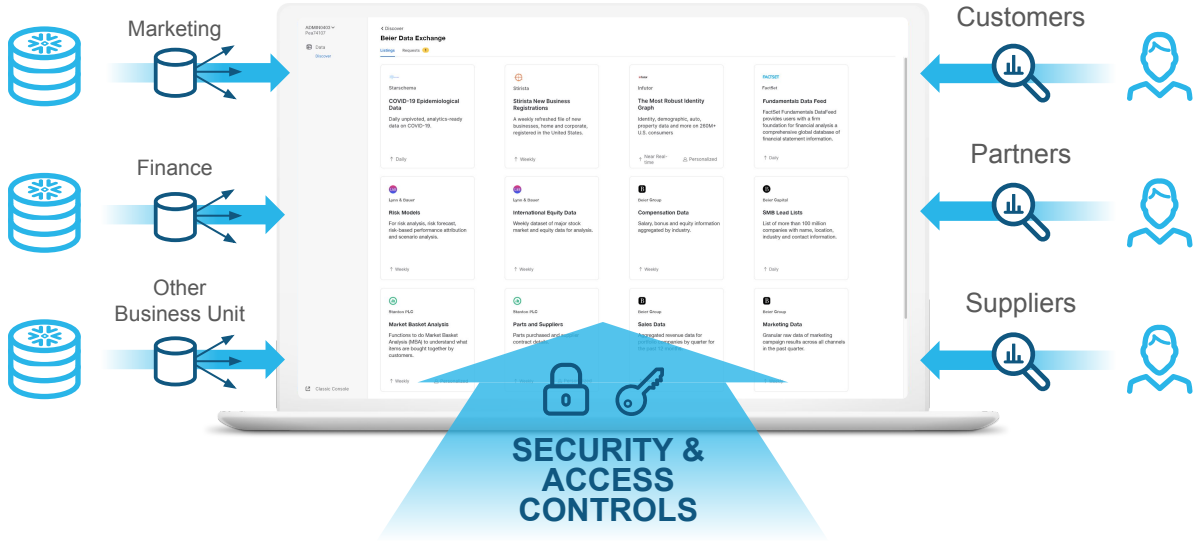
- Discover and access external data to drive business insights
- Get live, ready-to-query data
- Reduce data analytics costs
- Easily become a provider and monetize your data



DATA WAREHOUSE | DATA APPLICATIONS | DATA EXCHANGE

CREATE YOUR OWN DATA EXCHANGE

- Consumers can Discover shared datasets within your company or from outside
- Control who can publish/consume on your Data Exchange; request approval workflow
- Access metrics on who's using the listings
- Live data sharing facilitates compliance with privacy laws



Private Preview
AWS US - West

Private Preview
AWS US - East

Public Preview
AWS & Azure Deployments (non VPS)

Public Preview
Add GCP

GA
All Clouds, All Deployments

1/1/2020 | 2/1 | 3/1 | 4/1 | 5/1 | 6/1 | 7/1 | 8/1 | 9/1 | 10/1 | 11/1 | 12/1 | 12/31/2020

DATA WAREHOUSE | DATA APPLICATIONS | DATA EXCHANGE

GLOBAL DATA SHARING

Share with consumers on any region and any cloud

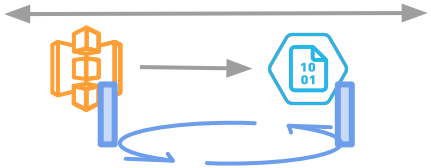
Q3 2020 Theme

Improve Provider Experience

Q4 2020 Theme

Reduce cost and operational overhead for providers to serve remote consumers

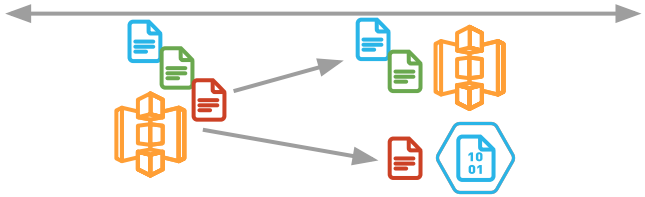
Q3 2020



Replicate Share to remote deployment and sync grants

Link two accounts and execute commands in remote deployment via Link without logging to remote deployment

Q4 2020



Sub-database replication to remote deployments to save replication cost - only replicate the tables that are shared

Automated fulfilment of Standard Listing for consumer in remote regions

DATA WAREHOUSE | DATA APPLICATIONS | DATA EXCHANGE

AVAILABLE LATE 2020

EINSTEIN ANALYTICS OUTPUT CONNECTOR

Access Salesforce data on Snowflake

Use Salesforce Data Everywhere

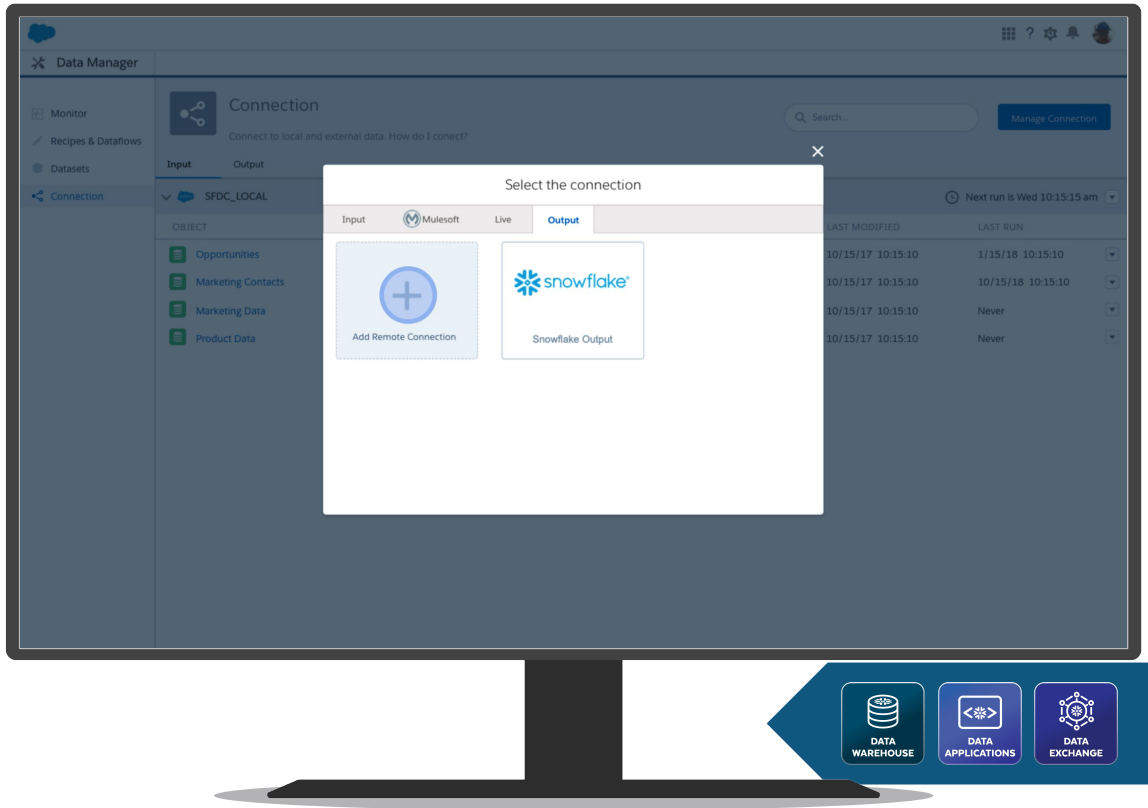
Extend access to your CRM data and get more valuable customer insights

Simplify Your Stack

Use clicks not code to send CRM data to Snowflake

Smart, Curated Datasets

Build your business logic once and use curated Einstein Analytics datasets to augment your data workloads in Snowflake



THANK YOU