



Rogers Hands-On-Lab

Getting Started: Zero to Snowflake

June, 2024

Agenda:

9:00 am MT / 11:00 am ET | Introduction

- Snowflake 101
- Lab Setup and Explanation

9:30 am MT / 11:30 am ET | Part 1

- Financial Governance
- Data Transformation
- Semi-Structured Data

10:15 am MT / 12:15 pm ET | Break

10:30 am MT / 12:30 pm ET | Part 2

- Data Governance
- Collaboration
- Geospatial
- Optional: Snowsight Dashboard

Your Account Team



Duncan Sinclair
Account Director



Luis Villavicencio
Solutions Architect



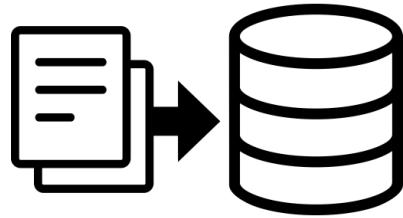
Nishikanta Bhadra
Solution Architect



Gurpreet Singh
Solution Architect



Mike Wies
Sales Engineer



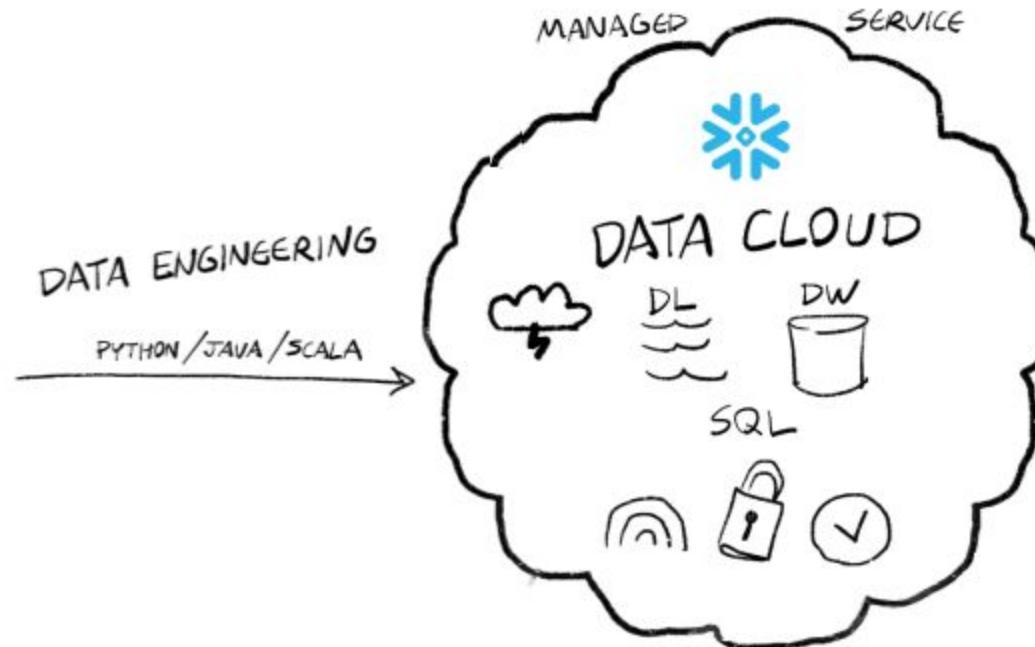
350 Monthly Active Users

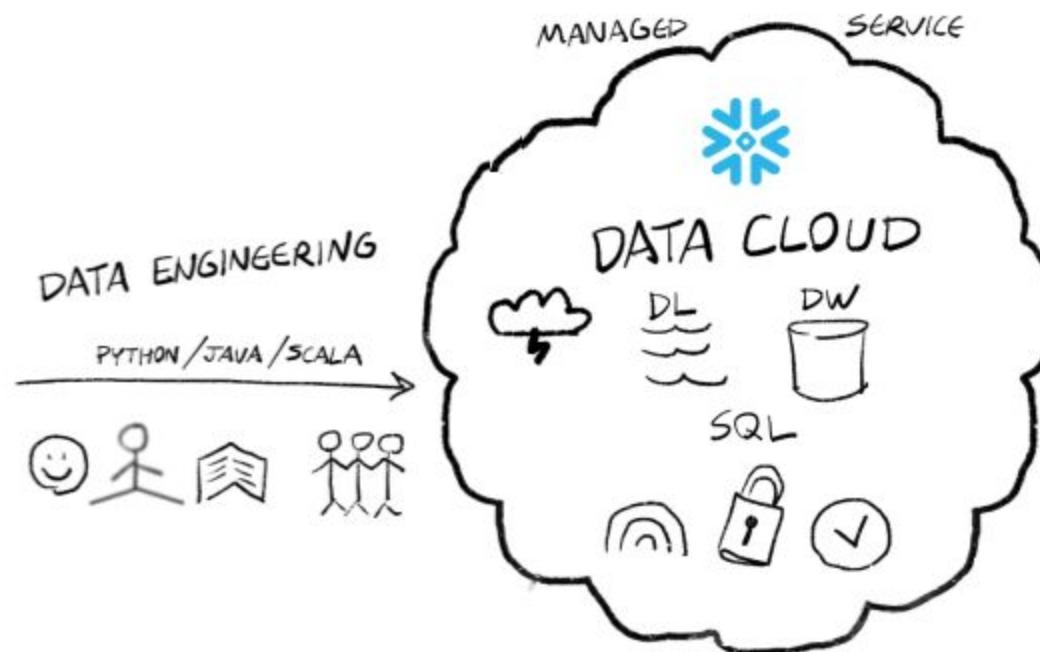
600k Queries per day

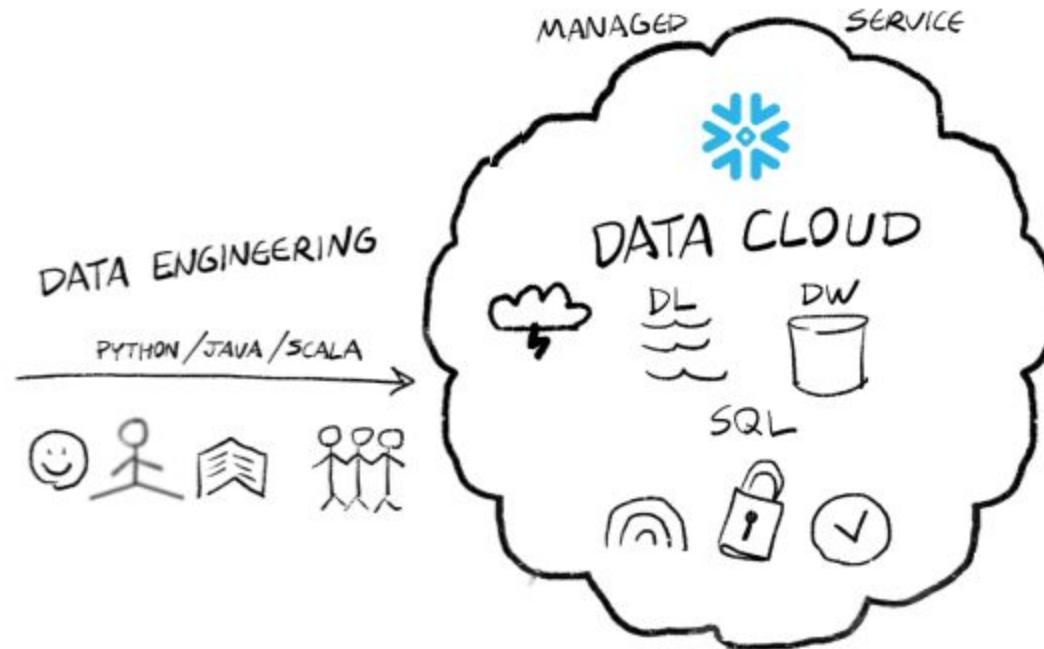
8PB of data
(Compressed!)













WHAT DO YOU ALREADY KNOW?





SNOWFLAKE

101

June, 2024

THERE IS NO ENTERPRISE STRATEGY WITHOUT A DATA STRATEGY



Simplify Your Data Foundation

For all data and
workloads



Distribute Analytics

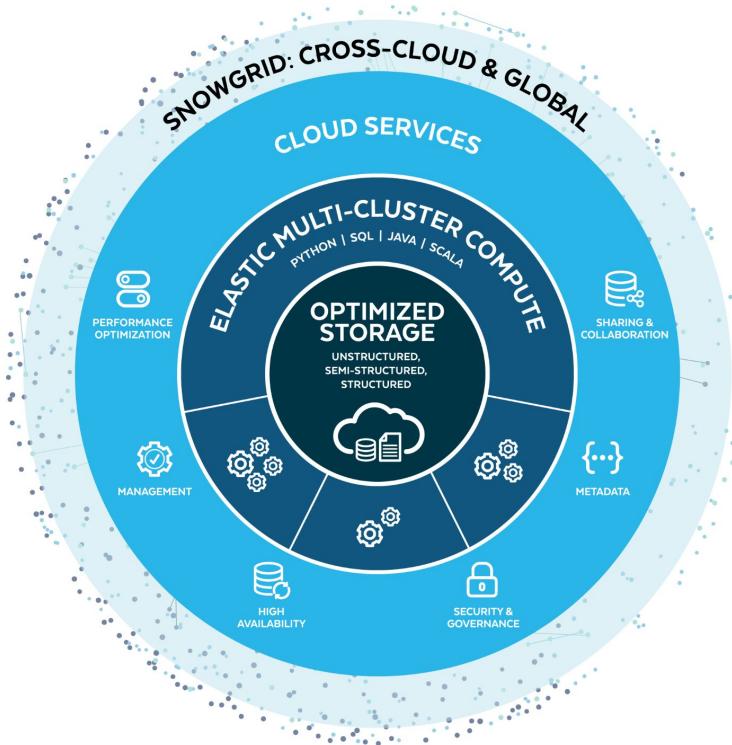
For all users against
enterprise data



Scale with Applications

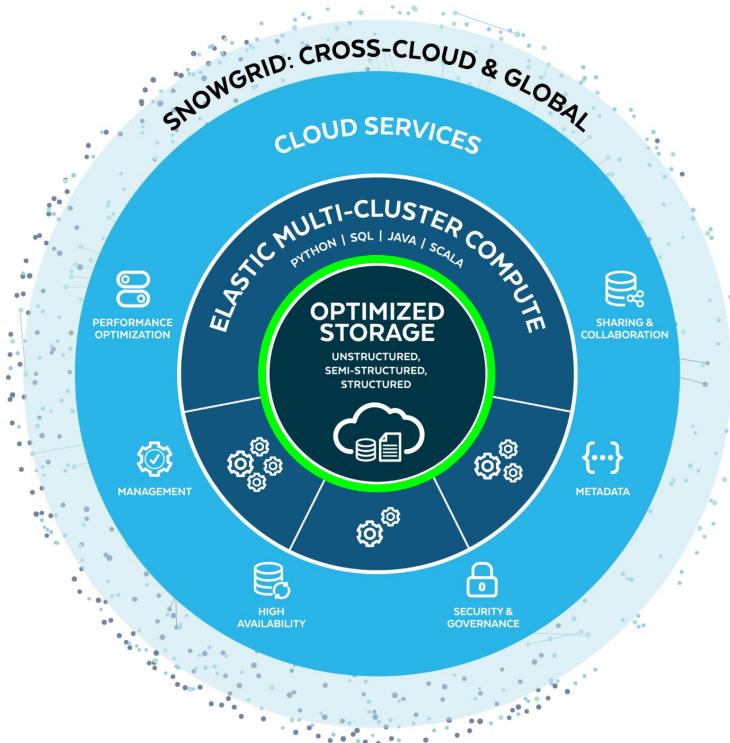
With streamlined development,
deployment, and productization

SNOWFLAKE PLATFORM ARCHITECTURE

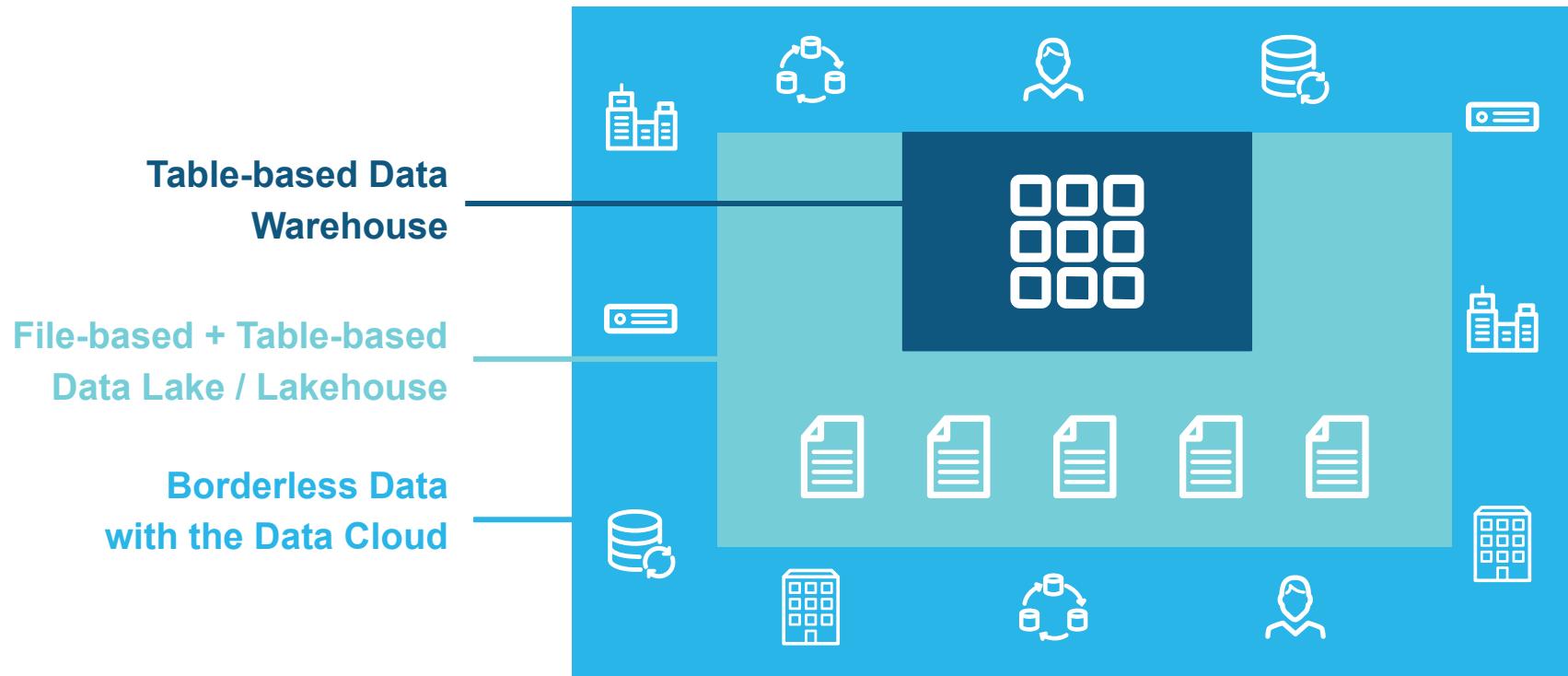


SNOWFLAKE PLATFORM ARCHITECTURE

STORAGE

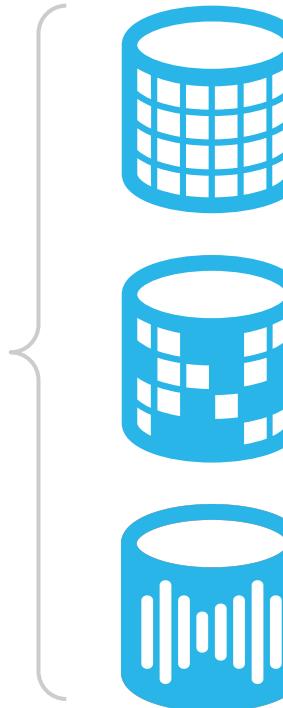


Evolution of Data Management



Supporting All Data Types

One place to store, govern, process, analyze, and share all data



Structured data

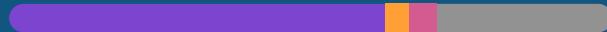
Semi-structured data

Unstructured data

Micro-partitions

Date	Product	Customer	Amount
Feb 14	Boots	Frank	\$150
Feb 14	Boots	Benoit	\$150
Feb 14	Skis	Thierry	\$300
Feb 14	Snowboard	Mike	\$250
Feb 15	Boots	Chris D	\$150
Feb 15	Skis	Denise	\$600
Feb 15	Snowboard	Shelly	\$250
Feb 16	Boots	Rob	\$150
Feb 16	Skis	Sunny	\$600
Feb 16	Snowboard	Chris K	\$250
Feb 16	Snowboard	Greg	\$750
Feb 16	Snowboard	Matt	\$750

Profile Overview (Finished)



- Processing (100%)
- Local Disk IO (63%)
- Synchronization (3%)
- Initialization (31%)

100%

63%

3%

3%

31%

100.00%

179.58 MB

100.00%

25.33 MB

Scan progress

Bytes scanned

Percentage scanned from cache

Bytes written to result

Partitions scanned

13

Partitions total

73



Auto Clustering

Original Micro-Partitions																												
	Micro-Partition 1 (rows 1-6)	Micro-Partition 2 (rows 7-12)	Micro-Partition 3 (rows 13-18)	Micro-Partition 4 (rows 19-24)																								
2 Type	<table border="1"><tr><td>2</td><td>4</td><td>3</td></tr><tr><td>2</td><td>3</td><td>2</td></tr></table>	2	4	3	2	3	2	<table border="1"><tr><td>3</td><td>2</td><td>4</td></tr><tr><td>5</td><td>1</td><td>5</td></tr></table>	3	2	4	5	1	5	<table border="1"><tr><td>2</td><td>4</td><td>2</td></tr><tr><td>1</td><td>5</td><td>3</td></tr></table>	2	4	2	1	5	3	<table border="1"><tr><td>1</td><td>4</td><td>5</td></tr><tr><td>5</td><td>3</td><td>2</td></tr></table>	1	4	5	5	3	2
2	4	3																										
2	3	2																										
3	2	4																										
5	1	5																										
2	4	2																										
1	5	3																										
1	4	5																										
5	3	2																										
Name	<table border="1"><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>							<table border="1"><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>							<table border="1"><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>							<table border="1"><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>						
Country	<table border="1"><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>							<table border="1"><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>							<table border="1"><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>							<table border="1"><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>						
1 Date	<table border="1"><tr><td>11/2</td><td>11/2</td><td>11/2</td></tr><tr><td>11/2</td><td>11/2</td><td>11/2</td></tr></table>	11/2	11/2	11/2	11/2	11/2	11/2	<table border="1"><tr><td>11/2</td><td>11/2</td><td>11/2</td></tr><tr><td>11/3</td><td>11/3</td><td>11/3</td></tr></table>	11/2	11/2	11/2	11/3	11/3	11/3	<table border="1"><tr><td>11/2</td><td>11/2</td><td>11/2</td></tr><tr><td>11/3</td><td>11/3</td><td>11/4</td></tr></table>	11/2	11/2	11/2	11/3	11/3	11/4	<table border="1"><tr><td>11/3</td><td>11/4</td><td>11/4</td></tr><tr><td>11/5</td><td>11/5</td><td>11/5</td></tr></table>	11/3	11/4	11/4	11/5	11/5	11/5
11/2	11/2	11/2																										
11/2	11/2	11/2																										
11/2	11/2	11/2																										
11/3	11/3	11/3																										
11/2	11/2	11/2																										
11/3	11/3	11/4																										
11/3	11/4	11/4																										
11/5	11/5	11/5																										

New Micro-Partitions (after clustering by date, type)

	Micro-Partition 6 (rows 1, 4, 6, 8, 13, 15)	Micro-Partition 7 (rows 3, 5, 7, 2, 9, 14)	Micro-Partition 8 (rows 10, 12, 17, 11, 16, 19)	Micro-Partition 9 (rows 18, 20-24)																								
Type	<table border="1"><tr><td>2</td><td>2</td><td>2</td></tr><tr><td>2</td><td>2</td><td>2</td></tr></table>	2	2	2	2	2	2	<table border="1"><tr><td>3</td><td>3</td><td>3</td></tr><tr><td>4</td><td>4</td><td>4</td></tr></table>	3	3	3	4	4	4	<table border="1"><tr><td>5</td><td>5</td><td>5</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	5	5	5	1	1	1	<table border="1"><tr><td>3</td><td>4</td><td>5</td></tr><tr><td>5</td><td>3</td><td>2</td></tr></table>	3	4	5	5	3	2
2	2	2																										
2	2	2																										
3	3	3																										
4	4	4																										
5	5	5																										
1	1	1																										
3	4	5																										
5	3	2																										
Name	<table border="1"><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>							<table border="1"><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>							<table border="1"><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>							<table border="1"><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>						
Country	<table border="1"><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>							<table border="1"><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>							<table border="1"><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>							<table border="1"><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>						
Date	<table border="1"><tr><td>11/2</td><td>11/2</td><td>11/2</td></tr><tr><td>11/2</td><td>11/2</td><td>11/2</td></tr></table>	11/2	11/2	11/2	11/2	11/2	11/2	<table border="1"><tr><td>11/2</td><td>11/2</td><td>11/2</td></tr><tr><td>11/2</td><td>11/2</td><td>11/3</td></tr></table>	11/2	11/2	11/2	11/2	11/2	11/3	<table border="1"><tr><td>11/3</td><td>11/3</td><td>11/3</td></tr><tr><td>11/2</td><td>11/2</td><td>11/2</td></tr></table>	11/3	11/3	11/3	11/2	11/2	11/2	<table border="1"><tr><td>11/4</td><td>11/4</td><td>11/4</td></tr><tr><td>11/5</td><td>11/5</td><td>11/5</td></tr></table>	11/4	11/4	11/4	11/5	11/5	11/5
11/2	11/2	11/2																										
11/2	11/2	11/2																										
11/2	11/2	11/2																										
11/2	11/2	11/3																										
11/3	11/3	11/3																										
11/2	11/2	11/2																										
11/4	11/4	11/4																										
11/5	11/5	11/5																										

What is it?

Reorganize table data to align with query patterns

Value

Process only the relevant data from large tables

Faster queries and fewer compute credits

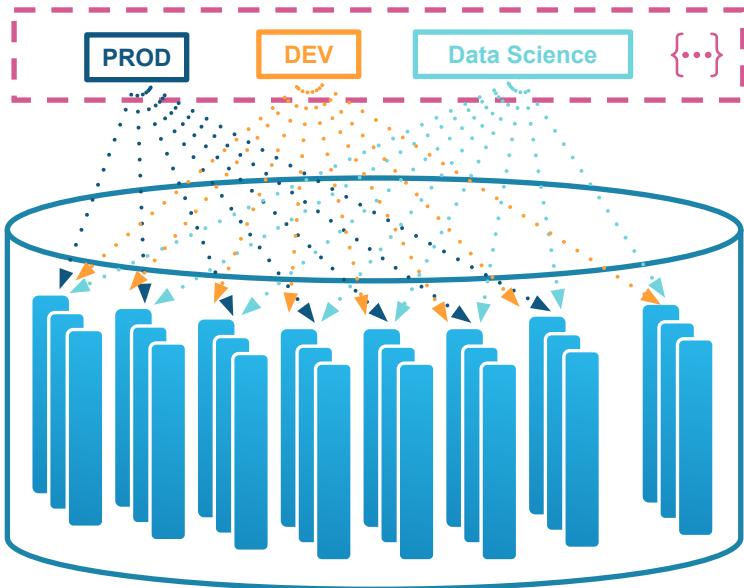
No blocking / impact on DML

Cost

Credits for background clustering maintenance



Zero-Copy Cloning



The Metadata layer keeps track of every micro-partition file in every customer database.

Creating a DEV environment usually means copying the PROD database

Limited to subset of full Prod

Up to 2x storage requirement

Periodic refreshes

Snowflake Zero-Copy Clones

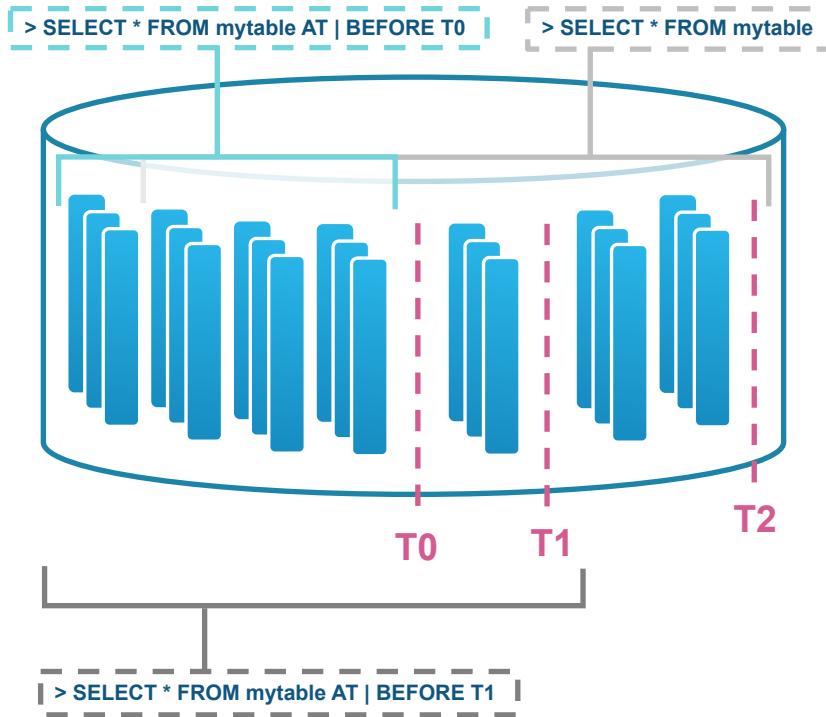
Simply “point” to the same files

Consumes zero additional storage

Changes to either DB are isolated



Time Travel



T0 – Initial state of database

T1 – update myTable set
colX = Y where...

T2 – ELT job loads new data

**Previous versions of data
automatically retained**

AT | BEFORE [timestamp | statement | offset]

CLONE AT | BEFORE to recreate a prior version

UNDROP recovers from accidental deletion

Accessed via SQL extensions

AT | BEFORE [timestamp | statement | offset]

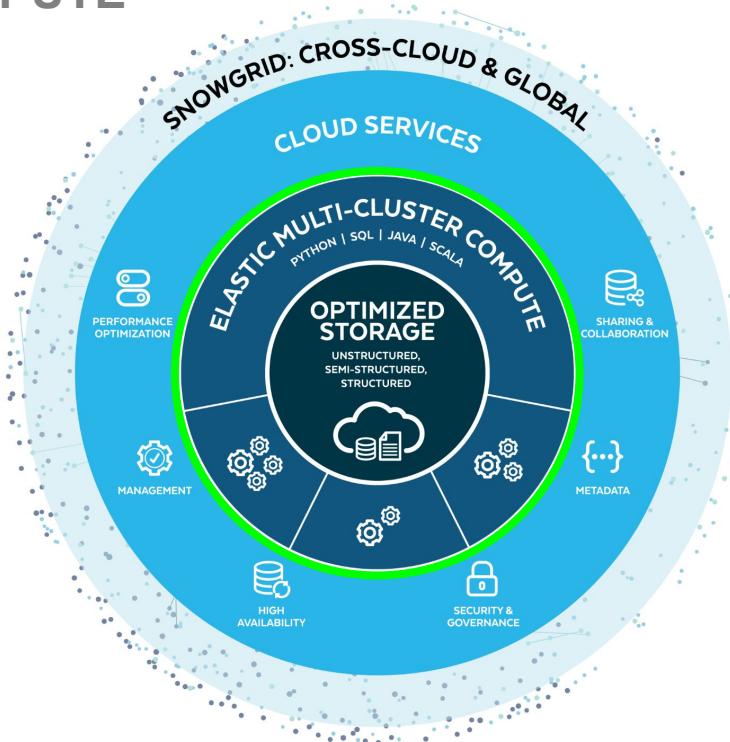
CLONE AT | BEFORE to recreate a prior version

UNDROP recovers from accidental deletion



SNOWFLAKE PLATFORM ARCHITECTURE

ELASTIC COMPUTE



THE POWER OF SNOWFLAKE'S ELASTIC SCALABILITY

4.2B

Average daily queries
from January 1, 2024
to
January 31, 2024

82T+*

The number of rows
in the largest
customer table

205K+*

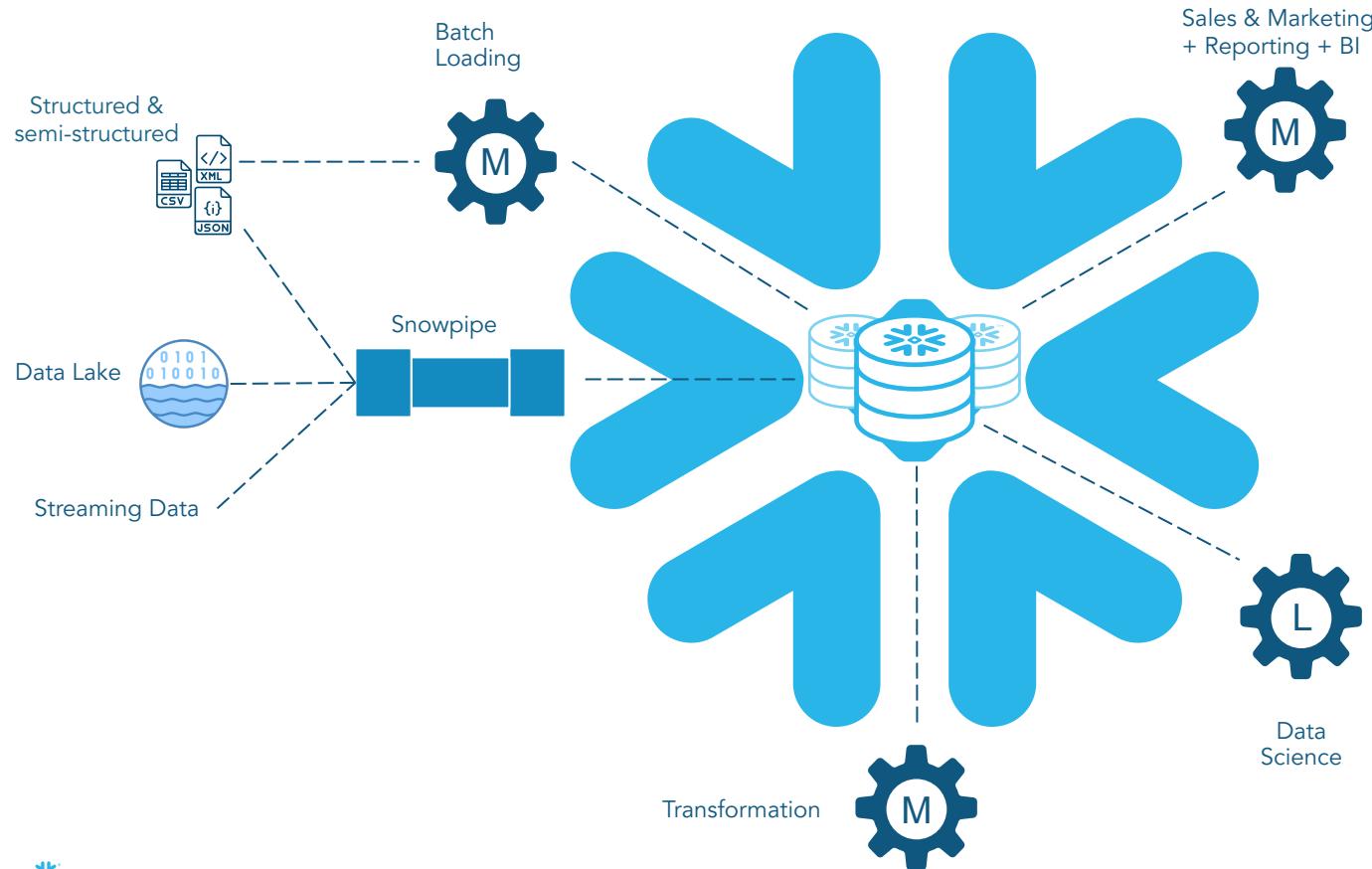
Largest number of
queries being executed
in a one minute interval
by a single customer

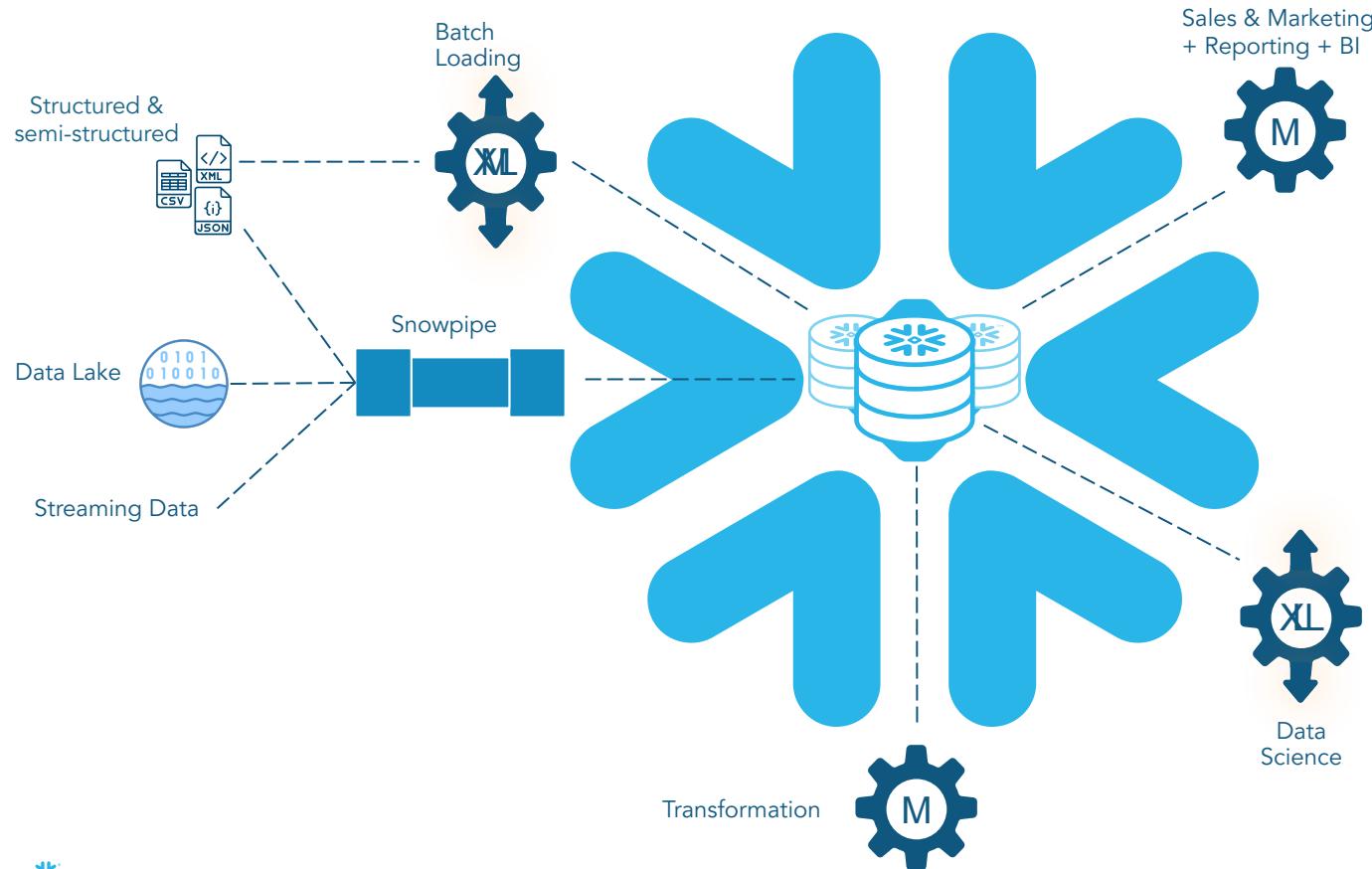
186PB*

The aggregate amount
of compressed data
stored in Snowflake for
the five largest
customers by data
volume

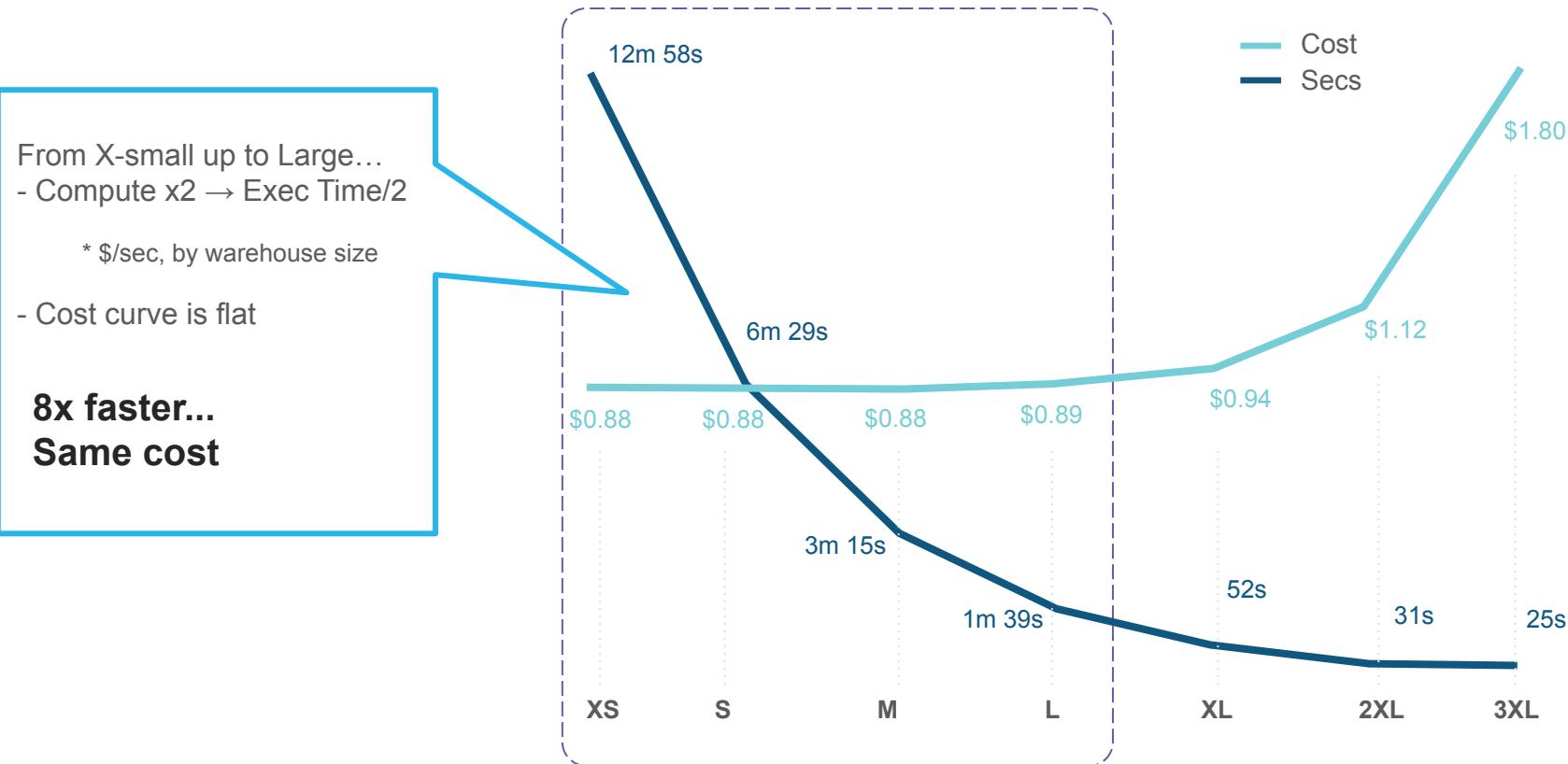
*Figures as of [January 31 2024](#)

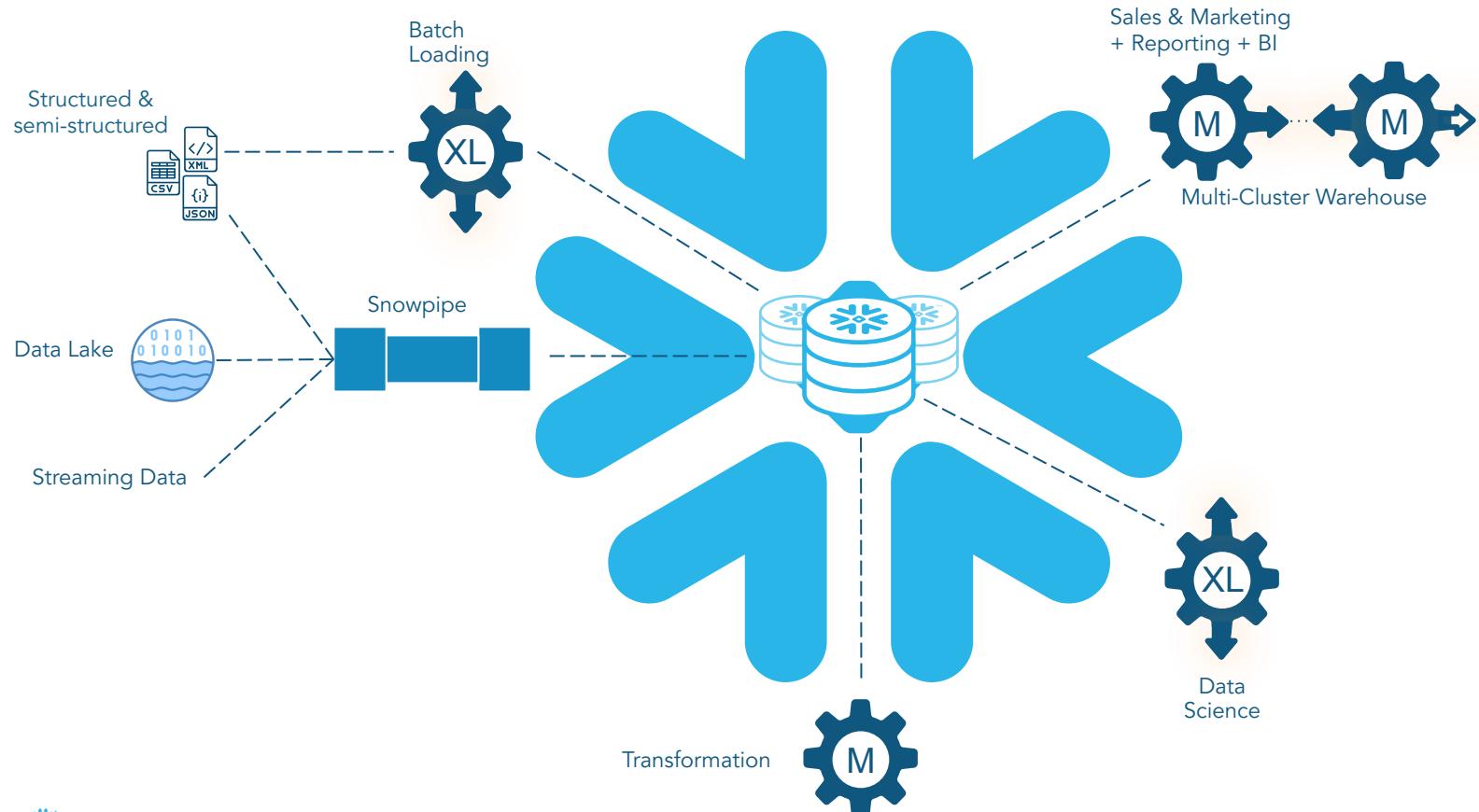




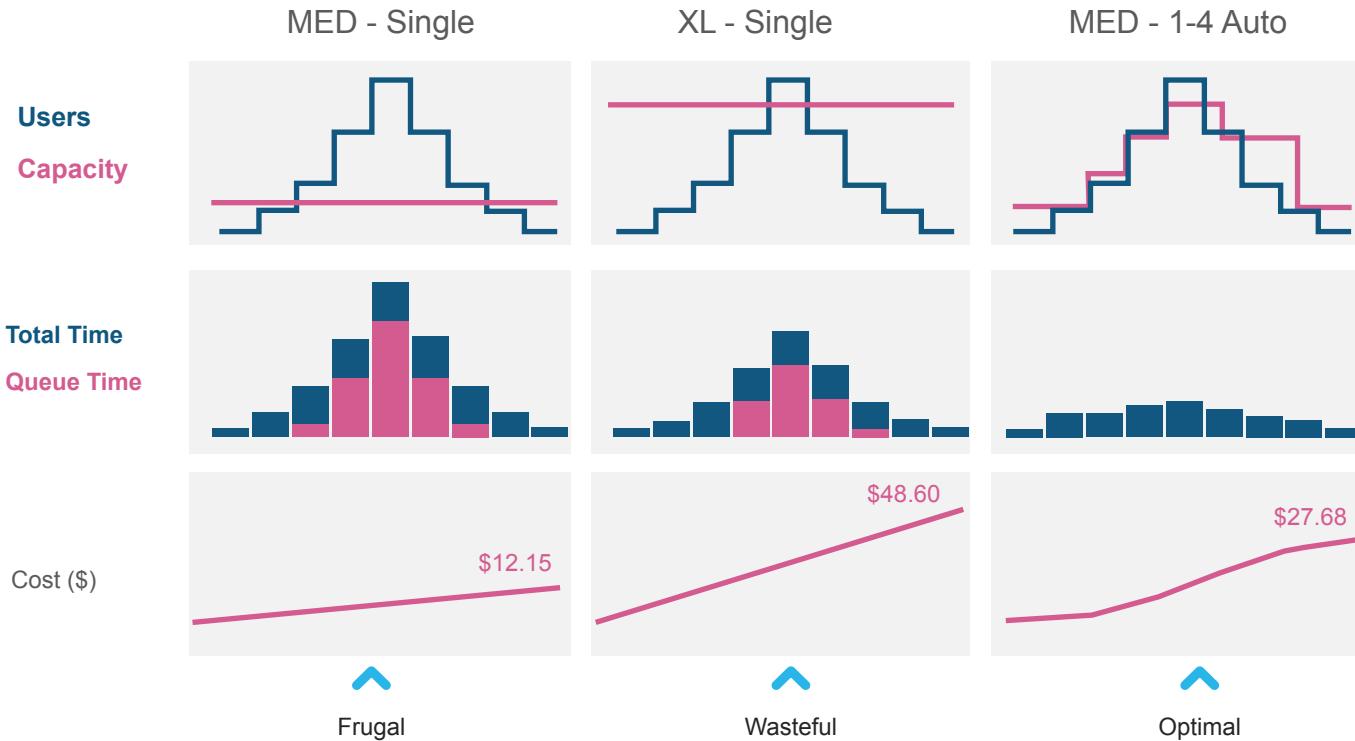


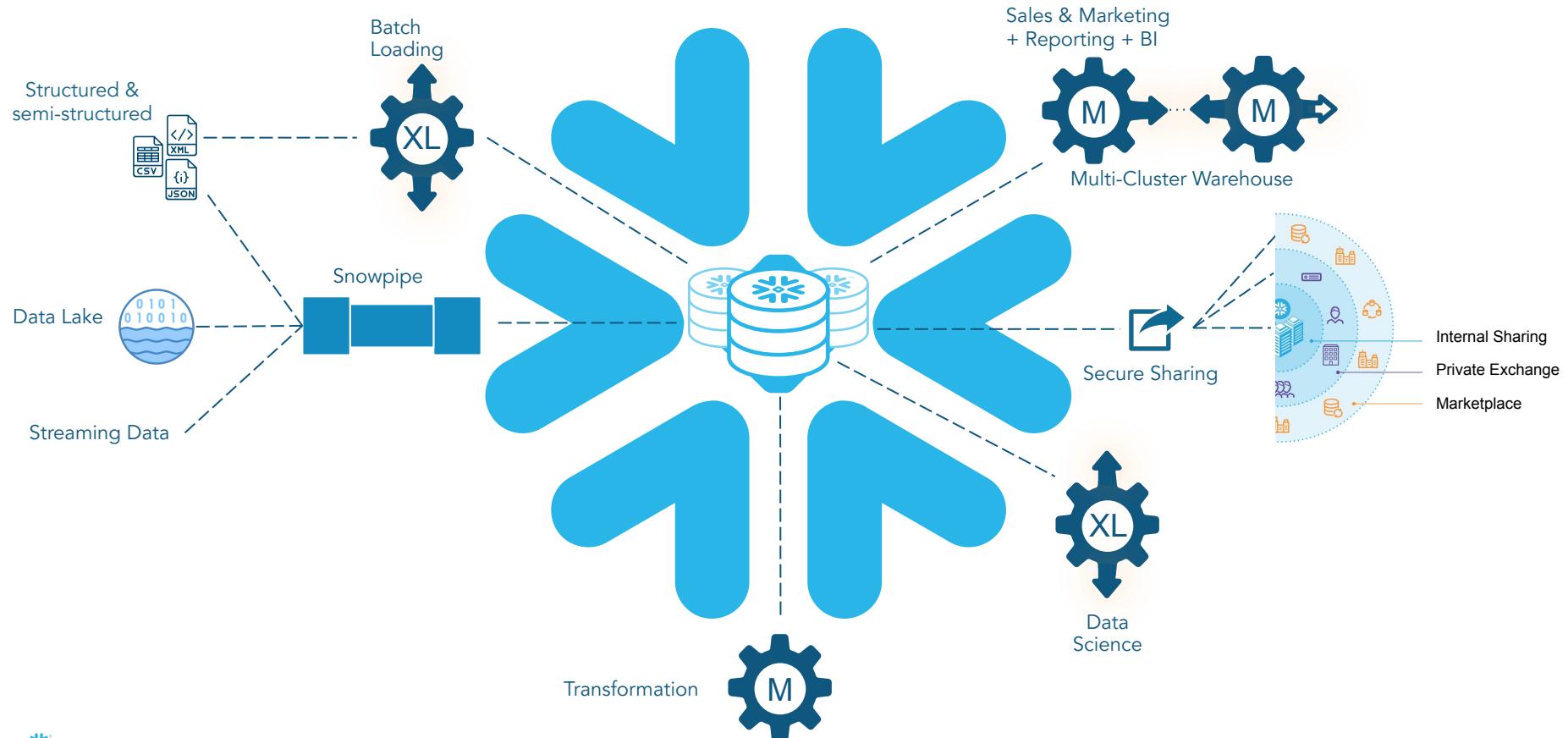
SCALE UP - LOADING 1BN RECORDS





SCALE OUT - MULTI-CLUSTER WAREHOUSES





Adaptive Caching

Metadata

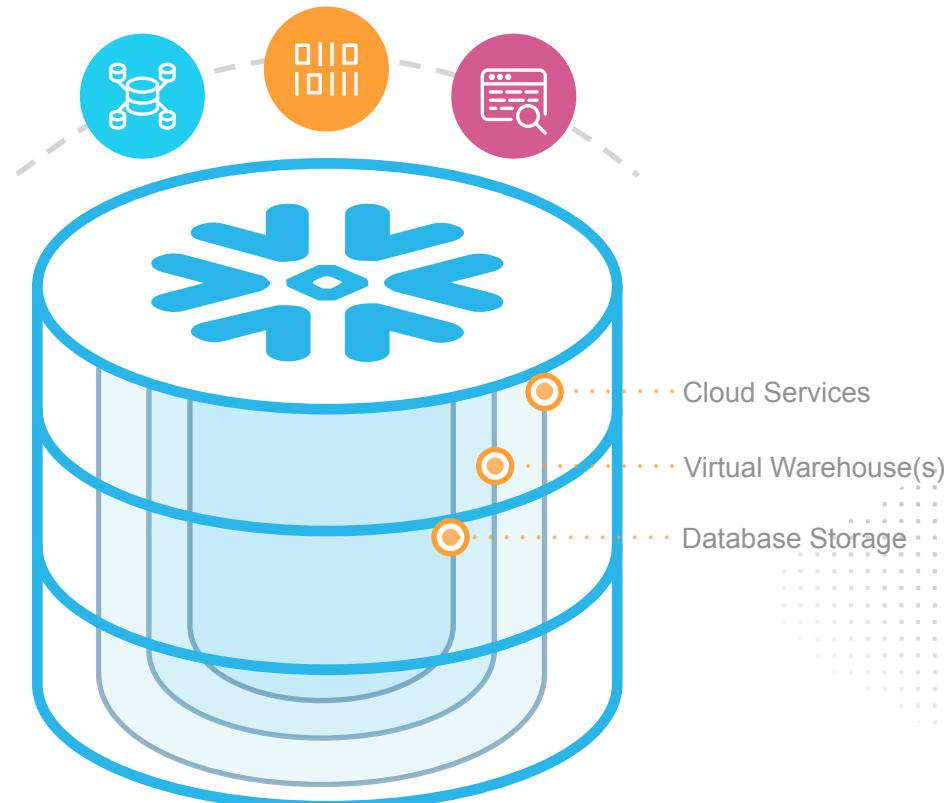
Metadata cached for fast access during query planning

Data

Active working set transparently cached on virtual warehouse SSD

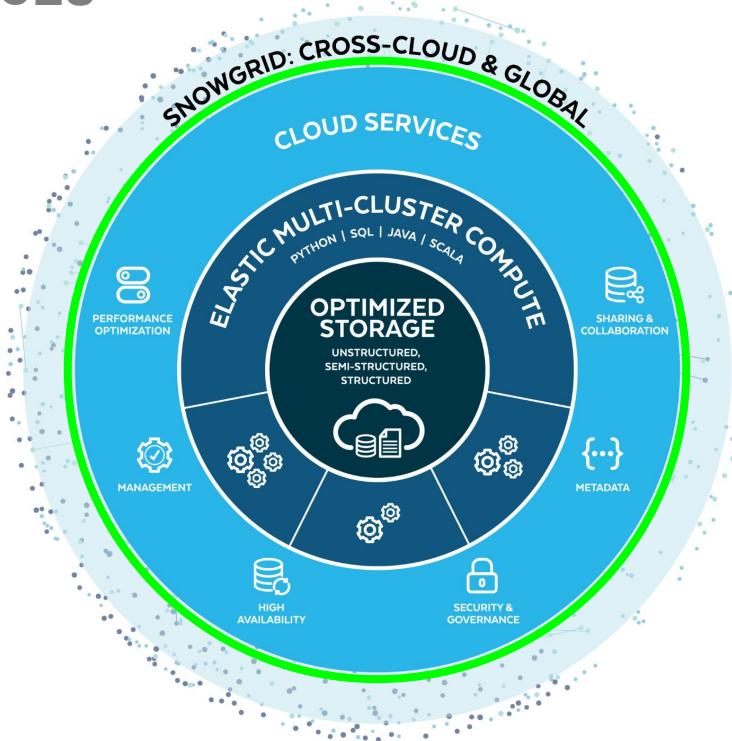
Query results

Results sets cached for reuse without requiring compute (e.g., static dashboard queries)

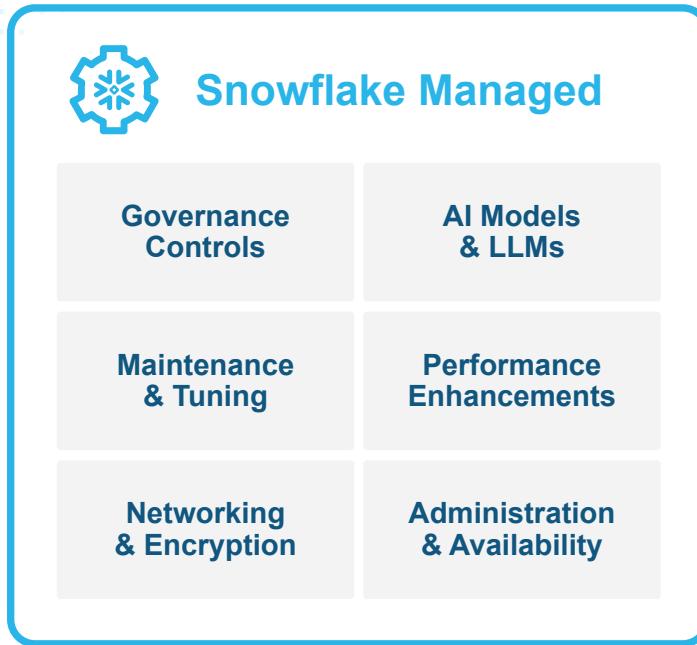


SNOWFLAKE PLATFORM ARCHITECTURE

CLOUD SERVICES



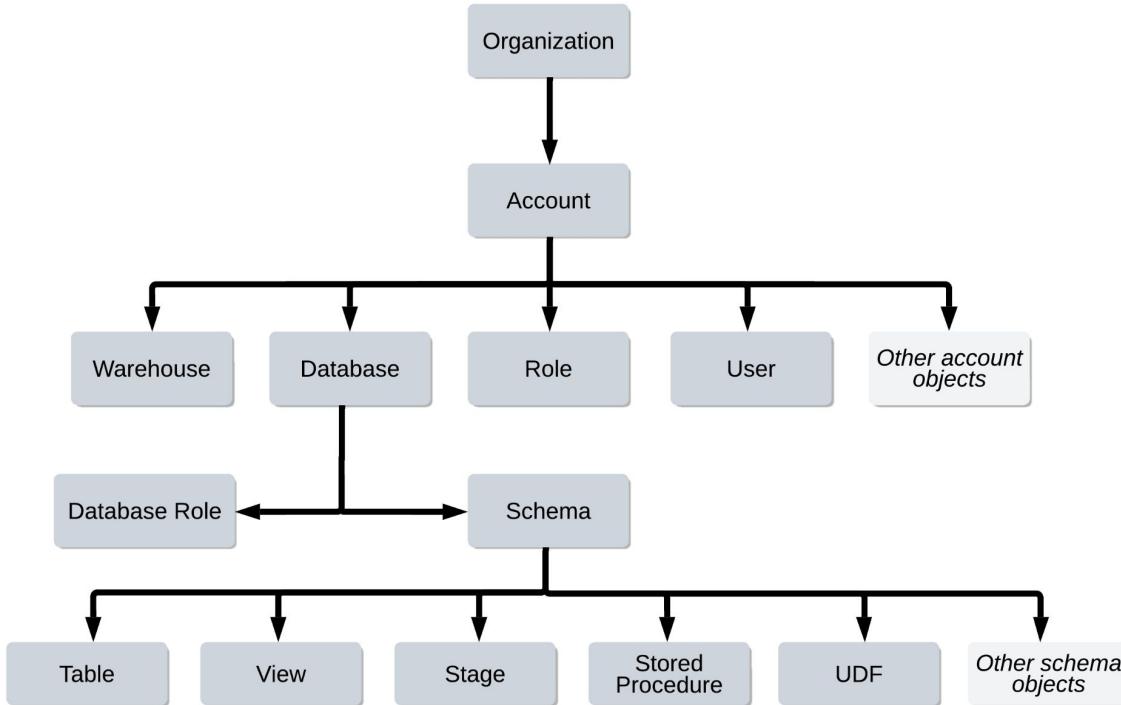
CLOUD SERVICES



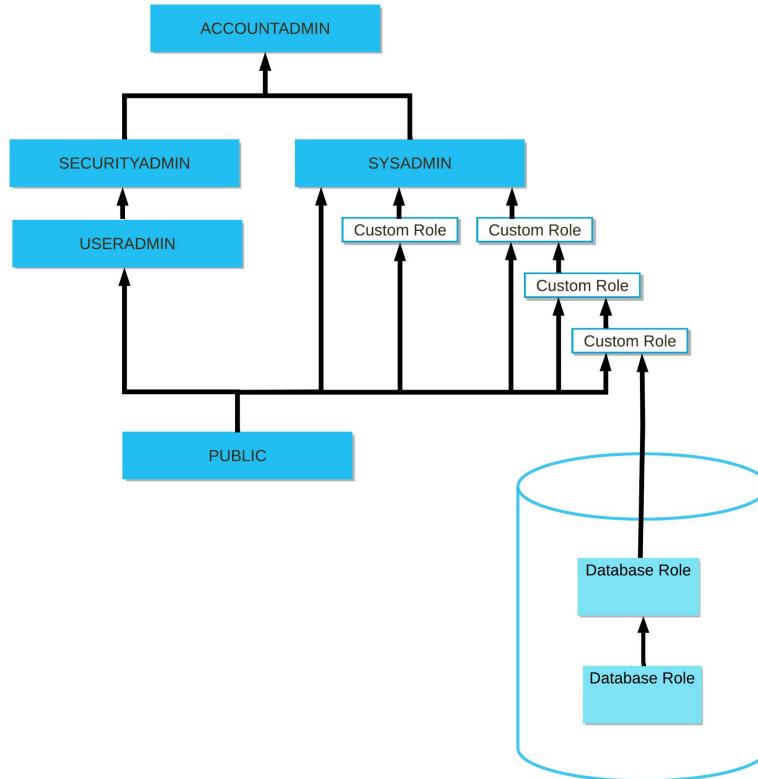
- Automate costly and complex operations
- Get unified, built-in governance
- Access built-in AI models, LLM, and vector search functionality



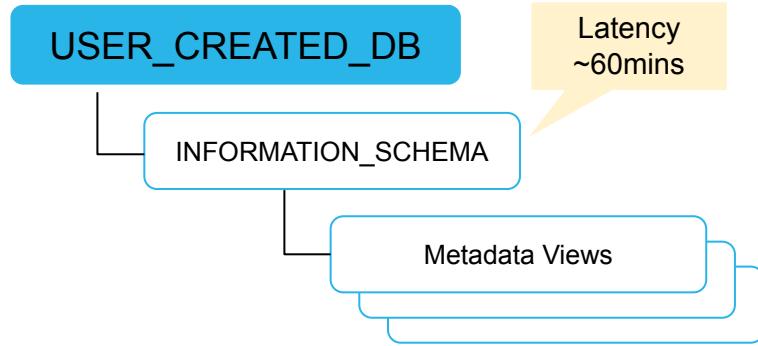
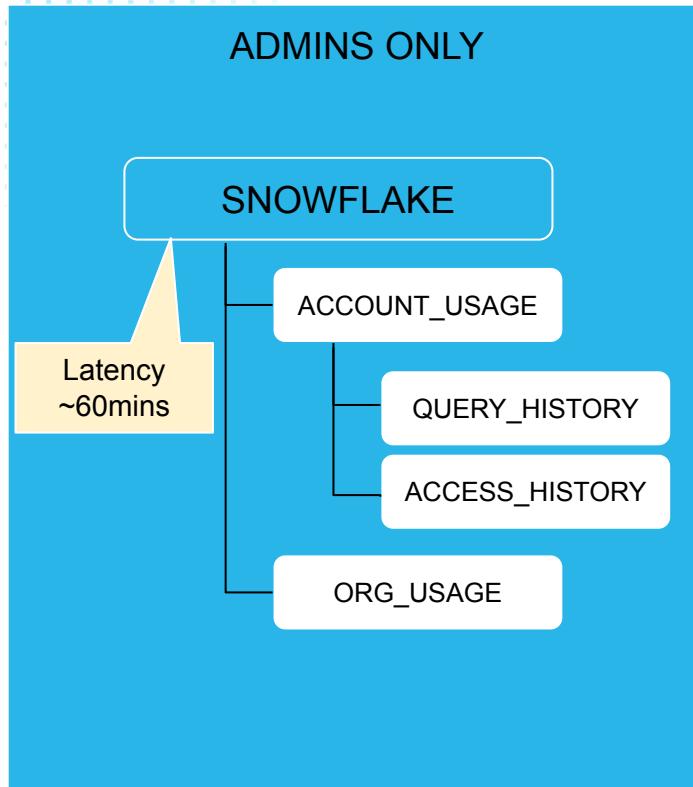
Securable Objects



Role-Based Access Control



Database Structure



SNOWPARK PYTHON INTRODUCTION



Snowpark for Python Features

Client API

DataFrame queries / transforms and submit UDFs / Stored Procs for execution.

UDFs

Execute custom Python code, **including OS packages**, in Snowflake secure Python sandbox.

Stored Procs

Host and operationalize Python code and/or Snowpark API calls. Single node bounded.

Vectorized UDFs

Pandas dataframe batch processing of vectorized functions (e.g. model inference).

UDTFs: Table Functions

Non 1:1 transformations with custom partitioning guaranteeing contiguous batches.



What's New: Snowpark for Python



Familiar Programming Constructs

Use familiar syntax
with DataFrame
abstraction



Rich Ecosystem

Easy access to hundreds of
packages with automated
dependency management

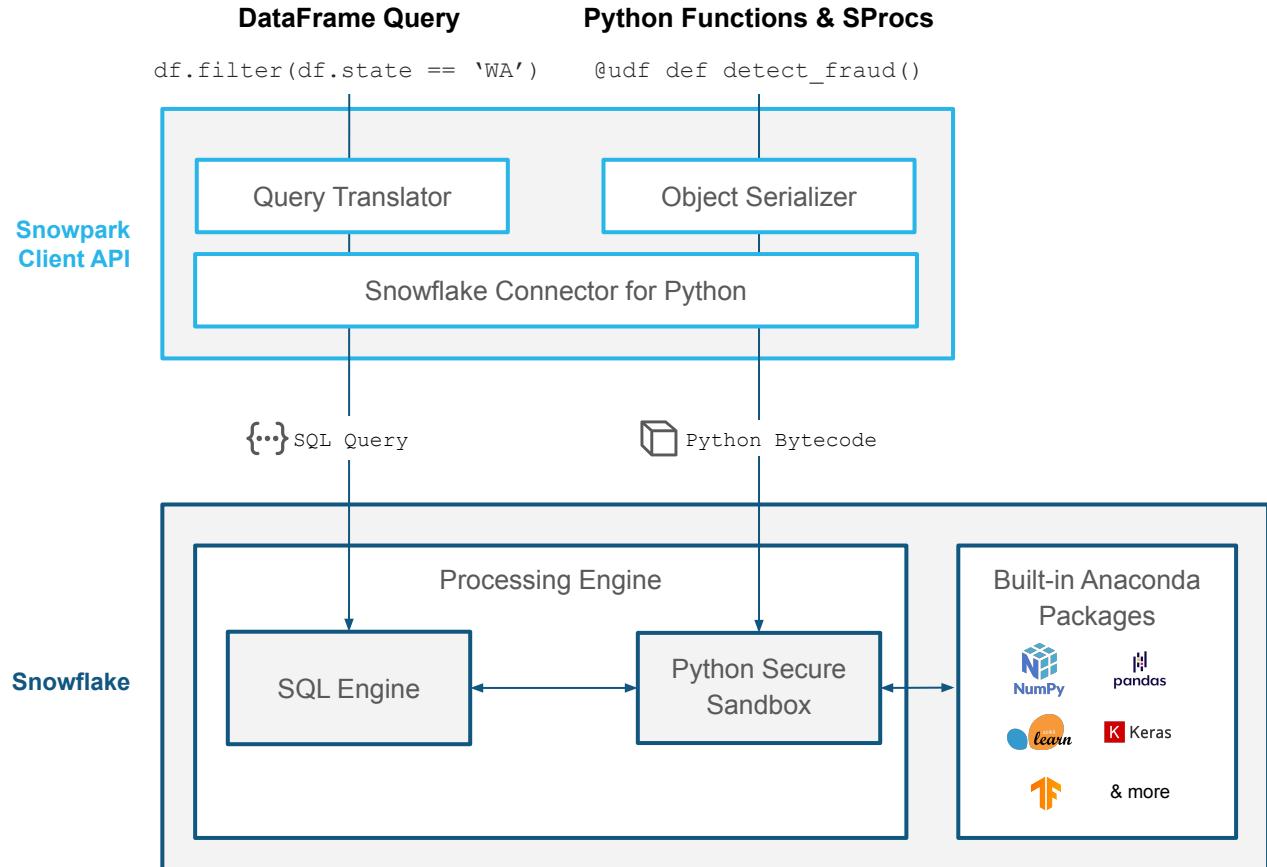


Secure Processing

Build with confidence
in a highly secure,
sandboxed environment



Snowpark for Python



WHAT IS CORTEX?

An fully managed service to host AI models, LLMs and vector functions.

Built-in solutions, Foundational, or Custom

WHY USE IT

Quickly and securely analyze data and build AI applications contextualized with your enterprise data.

HOW TO USE IT

Access the power of Snowflake Cortex via serverless SQL/Python functions or as part of an LLM-powered experience such as Document AI and Snowflake Copilot.

