

SECTION 1 – SNOWFLAKE INTRODUCTION

- INTRODUCTION
- CREATING SNOWFLAKE ACCOUNT
- TESTING SNOWFLAKE
- SNOWFLAKE ARCHITECTURE
- UNDERSTANDING THE WEB UI COMPONENTS

ABOUT INSTRUCTOR

- KASHISH GAKKAR
- POSSESS OVER 7 YEARS OF EXPERIENCE IN VARIOUS DOMAINS
- HAVE WORKED ON VARIOUS TECHNOLOGIES RELATED TO ETL, DATA WAREHOUSING, REPORTING, ANALYTICS & DATA SCIENCE.
- WORKING ON SNOWFLAKE FROM QUITE LONG TIME



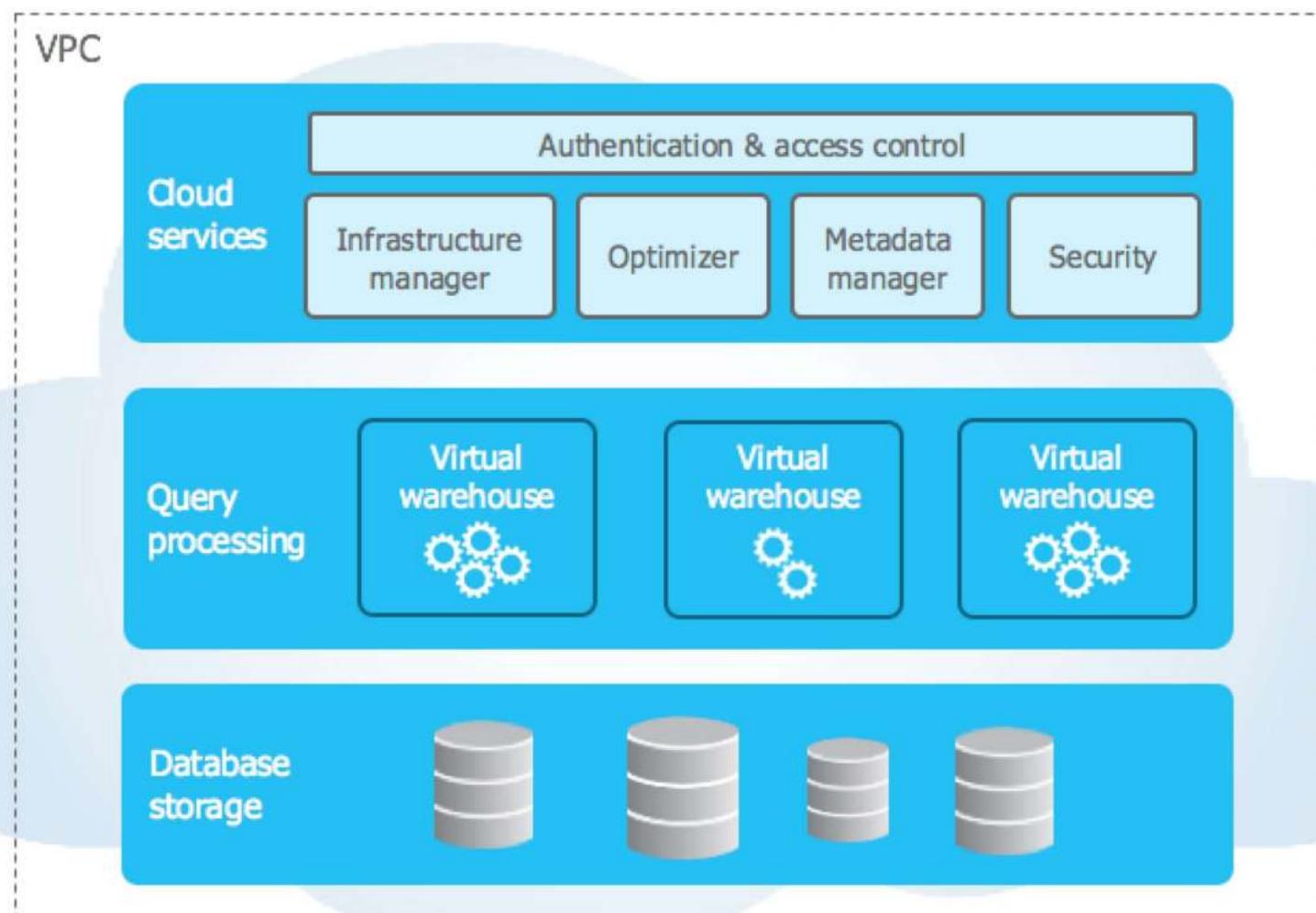
INTRO TO SNOWFLAKE

- CLOUD DATA WAREHOUSE
- IN HIGH DEMAND BECAUSE OF NICHE FEATURES
- PAY WHAT YOU USE
- MODERN ARCHITECTURE – SEPARATES COMPUTE AND STORAGE COST
- REMOVES SETUP OF INFRASTRUCTURE – BOTH HARDWARE AND SOFTWARE
- ELASTIC AND HIGHLY SCALABLE

SECTION 2 – DEEP DIVE INTO SNOWFLAKE

- INTRODUCTION
- SNOWFLAKE ARCHITECTURE
- UNDERSTANDING THE WEB UI COMPONENTS
- VIRTUAL WAREHOUSES

SNOWFLAKE ARCHITECTURE



STANDARD VS ECONOMY POLICY

Policy	Description	Cluster Starts...	Cluster Shuts Down...
Standard (default)	Prevents/minimizes queuing by favoring starting additional clusters over conserving credits.	The first cluster starts immediately when either a query is queued or the system detects that there's one more query than the currently-running clusters can execute. Each successive cluster waits to start 20 seconds after the prior one has started. For example, if your warehouse is configured with 10 max clusters, it can take a full 200+ seconds to start all 10 clusters.	After 2 to 3 consecutive successful checks (performed at 1 minute intervals), which determine whether the load on the least-loaded cluster could be redistributed to the other clusters without spinning up the cluster again.
Economy	Conserves credits by favoring keeping running clusters fully-loaded rather than starting additional clusters, which may result in queries being queued and taking longer to complete.	Only if the system estimates there's enough query load to keep the cluster busy for at least 6 minutes.	After 5 to 6 consecutive successful checks (performed at 1 minute intervals), which determine whether the load on the least-loaded cluster could be redistributed to the other clusters without spinning up the cluster again.

SECTION 3 – SNOWFLAKE PRICING

- PRICING INTRODUCTION
- WHAT IS SNOWFLAKE CREDIT
- SNOWFLAKE EDITIONS
- SERVERLESS FEATURES
- STORAGE COST
- CLOUD SERVICES AND DATA TRANSFER COST

SNOWFLAKE PRICING

- VERY IMPORTANT ASPECT WHILE WORKING ON SNOWFLAKE
- SEPARATES COMPUTE AND STORAGE COST
- CHARGES BASED ON CONSUMED SNOWFLAKE CREDITS
- VALUE OF SNOWFLAKE CREDITS IS BASED ON SNOWFLAKE EDITION
- WE WILL LEARN:

TYPES OF SNOWFLAKE EDITIONS AND THEIR CREDITS COST
SERVERLESS FEATURES

WHAT IS A SNOWFLAKE CREDIT?

STORAGE COST(ON-DEMAND/PRE-PURCHASED)

VIRTUAL WAREHOUSES/COMPUTE COST

DATA TRANSFER COST

CLOUD SERVICES COST

PRICING EXAMPLES



- A Snowflake credit is a unit of measure.
- Snowflake credits are used to pay for the consumption of resources on Snowflake.
- It is consumed only when a customer is using resources, such as when a virtual warehouse is running, the cloud services layer is performing work, or serverless features are used.



User Receives \$400 worth of free usage upon creation of Snowflake free trial account

WHAT IS A SNOWFLAKE CREDIT?

SNOWFLAKE EDITIONS

Standard



- Complete SQL Data Warehouse
- Secure Data Sharing across regions / clouds
- Premier Support 24 x 365
- 1 day of time travel
- Always-on enterprise grade encryption in transit and at rest
- Customer dedicated virtual warehouses
- Federated authentication
- Database Replication
- External Functions
- Snowsight analytics UI
- Create your own Data Exchange
- Data Marketplace access

Enterprise



- Standard +
- Multi-cluster warehouse
- Up to 90 days of Time Travel
- Annual rekey of all encrypted data
- Materialized Views
- Search Optimization Service
- Dynamic Data Masking
- External Data Tokenization

Business Critical



- Enterprise +
- HIPAA Support
- PCI Compliance
- Data encryption everywhere
- Tri-Secret Secure using customer managed keys
- AWS PrivateLink support
- Database failover and fallback for business continuity

Virtual Private Snowflake



- Business Critical +
- Customer dedicated virtual servers wherever the encryption key is in memory
- Customer dedicated metadata store

SERVERLESS FEATURES

SNOWFLAKE OFFERS THE FOLLOWING ADDITIONAL SERVERLESS FEATURES THAT USE SNOWFLAKE-MANAGED COMPUTE RESOURCES AND CONSUME SNOWFLAKE CREDITS WHEN THEY ARE USED.

Snowpipe

- Rapidly ingests streaming data
- Is an automated service
- Requires no virtual warehouses
- Uses Snowflake-managed compute resources, which are paid for with Snowflake credits and are billed per second, plus incurs a fixed Snowflake credit charge per file

Database Replication

- Seamlessly replicates data across regions and cloud platforms
- Requires no virtual warehouses
- Uses Snowflake-managed compute resources, which are paid for with Snowflake credits and are billed per second, plus incurs standard storage and data transfer costs

Materialized Views Maintenance

- Automatically syncs materialized views with underlying base tables
- Requires no virtual warehouses
- Uses Snowflake-managed compute resources, which are paid for with Snowflake credits and are billed per second

Automatic Clustering

- Maintains optimal clustering state using defined cluster keys
- Applies to tables and materialized views
- Uses Snowflake-managed compute resources, which are paid for with Snowflake credits and are billed per second

Search Optimization Service

- Greatly speed up point lookup queries in massive tables
- Pair SOS with smaller virtual warehouses to lower costs
- Uses Snowflake-managed compute resources, which are paid for with Snowflake credits and are billed per second

CHOOSE STORAGE TYPE WISELY

	ASSUMPTIONS	ACTUAL CONSUMPTION	COST PER TB	TOTAL COST PER MONTH
MARIA	<p>100GB OF STORAGE WOULD BE REQUIRED PER MONTH</p> <p>SELECTED PRE-PURCHASED STORAGE PLAN</p>	20GB	\$23/TB	$0.1 * 23 = \$2.3$
KAVITA	<p>10GB OF STORAGE WOULD BE REQUIRED PER MONTH</p> <p>SELECTED ON-DEMAND STORAGE PLAN</p>	20GB	\$40/TB	$0.02 * 40 = \$0.8$

CHOOSE STORAGE TYPE WISELY

	ASSUMPTIONS	ACTUAL CONSUMPTION	COST PER TB	TOTAL COST PER MONTH
 MARIA	<p>100GB OF STORAGE WOULD BE REQUIRED PER MONTH</p> <p>SELECTED PRE-PURCHASED STORAGE PLAN</p>	90GB	\$23/TB	$0.1 * 23 = \$2.3$
 KAVITA	<p>10GB OF STORAGE WOULD BE REQUIRED PER MONTH</p> <p>SELECTED ON-DEMAND STORAGE PLAN</p>	90GB	\$40/TB	$0.09 * 40 = \$3.6$

STORAGE COST(ON-DEMAND/PRE-PURCHASED)

ON-DEMAND

- THE EASIEST AND MOST FLEXIBLE WAY TO PURCHASE THE SNOWFLAKE SERVICE IS ON DEMAND
- CUSTOMERS ARE CHARGED A FIXED RATE FOR THE SERVICES THAT ARE CONSUMED AND ARE BILLED IN ARREARS EVERY MONTH
- COMMON PRICE ACROSS REGIONS: \$40 PER MONTH

PRE-PURCHASED CAPACITY

- SNOWFLAKE PROVIDES CUSTOMERS THE OPTION TO PRE-PURCHASE CAPACITY. A CAPACITY PURCHASE IS A SPECIFIC DOLLAR COMMITMENT TO SNOWFLAKE
- COMMON PRICE ACROSS REGIONS: \$23 PER MONTH

A POPULAR WAY OF GOING ABOUT THE PRICING STRATEGY, ESPECIALLY WHEN YOU ARE NEW AND UNSURE ABOUT THIS, IS TO FIRST OPT FOR THE ON-DEMAND, AND THEN SWITCH TO PRE-PURCHASED. ONCE THE ON-DEMAND CYCLE STARTS, MONITOR THE RESOURCE USAGE FOR A MONTH OR TWO, AND ONCE YOU HAVE A GOOD IDEA FOR YOUR MONTHLY DATA WAREHOUSING REQUIREMENTS, SWITCH TO A PRE-PURCHASED PLAN TO OPTIMIZE THE RECURRING MONTHLY CHARGES.

VIRTUAL WAREHOUSE/ COMPUTE COST

- SNOWFLAKE SUPPORTS A WIDE RANGE OF VIRTUAL WAREHOUSE SIZES: X-SMALL, SMALL, MEDIUM, LARGE, X-LARGE, 2X-LARGE, 3X-LARGE, AND 4X-LARGE.
- THE SIZE OF THE VIRTUAL WAREHOUSE DETERMINES HOW FAST QUERIES WILL RUN.
- WHEN A VIRTUAL WAREHOUSE IS NOT RUNNING (THAT IS, WHEN IT IS SET TO SLEEP MODE), IT DOES NOT CONSUME ANY SNOWFLAKE CREDITS.
- THE DIFFERENT SIZES OF VIRTUAL WAREHOUSES CONSUME SNOWFLAKE CREDITS AT THE FOLLOWING RATES, BILLED BY THE SECOND WITH A ONE-MINUTE MINIMUM.

	XS	S	M	L	XL	2XL	3XL	4XL
Credits consumed per hour	1	2	4	8	16	32	64	128

DATA TRANSFER COST

- CUSTOMERS WHO WISH TO MOVE OR COPY THEIR DATA BETWEEN REGIONS OR CLOUDS WILL INCUR DATA TRANSFER CHARGES.
- FEATURES SUCH AS EXTERNAL TABLES, EXTERNAL FUNCTIONS AND DATA LAKE EXPORT MAY INCUR DATA TRANSFER CHARGES.
- THE PRICING FOR THOSE CHARGES IS AS FOLLOWS FOR CUSTOMERS ON AMAZON WEB SERVICES OR MICROSOFT AZURE:

Cloud	Region where data is stored	Transfer to same region, same cloud (per TB)	Transfer to different region, same cloud (per TB)	Transfer to different cloud (per TB)
AWS	US East (Northern Virginia)	\$0	\$20	\$90
AWS	US East (Ohio)	\$0	\$20	\$90
AWS	US West (Oregon)	\$0	\$20	\$90
AWS	Canada (Central)	\$0	\$20	\$90
AWS	EU (Ireland)	\$0	\$20	\$90
AWS	EU (Frankfurt)	\$0	\$20	\$90
AWS	Asia Pacific (Sydney)	\$0	\$140	\$140
AWS	Asia Pacific (Singapore)	\$0	\$90	\$120
AWS	Asia Pacific (Tokyo)	\$0	\$90	\$114
AWS	Asia Pacific (Mumbai)	\$0	\$86	\$109.30
AWS	AWS US East (Commercial Gov - N. Virginia)	\$0	\$20	\$90
AZURE	East US 2 (Virginia)	\$0	\$87	\$87
AZURE	West US 2 (Washington)	\$0	\$87	\$87
AZURE	Canada Central (Toronto)	\$0	\$87	\$87
AZURE	West Europe (Netherlands)	\$0	\$87	\$87
AZURE	Switzerland North (Zurich)	\$0	\$87	\$87
AZURE	Australia East (Sydney)	\$0	\$120	\$120
AZURE	Southeast Asia (Singapore)	\$0	\$120	\$120
AZURE	US Gov Virginia	\$0	\$109	\$109

Cloud Provider	Data Transfer Source Region	Data Transfer To Same Cloud Provider				Data Transfer to Different Cloud Provider or Internet		
		Same Region	Different Region, Same Continent	Different Continents (excludes Oceania)	Oceania	Same Continent	Different Continent (excludes Oceania)	Oceania
GCP	US-Central1	\$0/TB	\$10/TB	\$80/TB	\$150/TB	\$105/TB	\$120/TB	\$190/TB
GCP	Europe-West2	\$0/TB	\$20/TB	\$80/TB	\$150/TB	\$105/TB	\$120/TB	\$190/TB
GCP	Europe-West4	\$0/TB	\$20/TB	\$80/TB	\$150/TB	\$105/TB	\$120/TB	\$190/TB

CLOUD SERVICES COST

- CLOUD SERVICES RESOURCES ARE AUTOMATICALLY ASSIGNED BY SNOWFLAKE BASED ON THE REQUIREMENTS OF THE WORKLOAD.
- TYPICAL UTILIZATION OF CLOUD SERVICES (UP TO 10% OF DAILY COMPUTE CREDITS) IS INCLUDED FOR FREE
- CUSTOMERS WILL NOT SEE INCREMENTAL CHARGES FOR CLOUD SERVICES USAGE.

OPTIMIZE/REDUCE SNOWFLAKE COSTS

- DEPENDING ON YOUR LOCATION, IT IS IMPORTANT TO CHOOSE THE CLOUD REGION (LIKE US EAST, US WEST, ETC. DEPENDING ON THE CLOUD PROVIDER) WISELY, TO MINIMIZE LATENCY, TO HAVE ACCESS TO THE REQUIRED SET OF FEATURES, ETC. IF YOU ARE TO MOVE YOUR DATA TO A DIFFERENT REGION LATER, THERE ARE DATA TRANSFER COSTS ASSOCIATED WITH IT AT A PER TERABYTE SCALE. SO THE LARGER YOUR DATA STORE, THE MORE THE COSTS YOU INCUR.
- IT CAN MAKE QUITE A DIFFERENCE TO THE COSTS INCURRED BY OPTIMALLY MANAGING THE OPERATIONAL STATUS OF YOUR COMPUTE CLUSTERS. THE FEATURES SUCH AS 'AUTO SUSPENSION' AND 'AUTO RESUME' SHOULD BE MADE USE OF UNLESS THERE IS A BETTER STRATEGY TO ADDRESS THIS.
- THE WORKLOAD/DATA USAGE MONITORING AT AN ACCOUNT LEVEL, WAREHOUSE LEVEL, DATABASE OR TABLE LEVEL IS NECESSARY TO MAKE SURE THERE AREN'T UNNECESSARY QUERY OPERATIONS OR DATA STORAGE CONTRIBUTING TO THE OVERALL MONTHLY COSTS.
- MAKE SURE TO HAVE THE DATA COMPRESSED BEFORE STORAGE AS MUCH AS POSSIBLE. THERE ARE INSTANCES, SUCH AS STORING DATABASE TABLES, WHERE SNOWFLAKE AUTOMATICALLY DOES A DATA COMPRESSION, HOWEVER THIS IS NOT ALWAYS THE CASE, SO THIS IS SOMETHING TO BE MINDFUL OF AND TO BE MONITORED REGULARLY..
- SNOWFLAKE WORKS BETTER WITH DATE OR TIMESTAMP COLUMNS STORED AS SUCH RATHER THAN THEM BEING STORED AS TYPE VARCHAR.
- TRY TO MAKE MORE USE OF TRANSIENT TABLES AS THEY ARE NOT MAINTAINED IN THE HISTORY TABLES WHICH IN TURN REDUCES THE DATA STORAGE COSTS FOR HISTORY TABLES.

QUIZ TIME - PRICING

Q1. TOTAL COST IN DOLLARS/USD FOR 2TB OF DATA?

Q2. TOTAL COST IN DOLLARS/USD FOR 13TB OF DATA IN **AWS OHIO REGION**?

Q3. TOTAL CREDITS CONSUMED AND CORRESPONDING COST FOR RUNNING VIRTUAL WAREHOUSE (MEDIUM [4CREDITS/HOUR] - USED FOR ETL LOADS) RUNNING FOR **2HOURS PER DAY FOR 31DAYS** IN A MONTH, IN **AWS OHIO REGION**?

Q4. TOTAL CREDITS CONSUMED AND CORRESPONDING COST FOR RUNNING VIRTUAL WAREHOUSE (SMALL [2CREDITS/HOUR] USED BY USERS) RUNNING FOR **2HOURS PER DAY FOR 31DAYS** IN A MONTH, IN **AWS OHIO REGION**?

SECTION 4 – RESOURCE MONITORS

- RESOURCE MONITORS INTRODUCTION
- PROPERTIES OF RESOURCE MONITORS
- ACTIONS OR TRIGGERS OF RESOURCE MONITORS
- SUSPENSION OR RESUMPTION OF RESOURCE MONITORS
- CREATE RESOURCE MONITORS USING WEBUI
- CREATE RESOURCE MONITORS USING SQL

RESOURCE MONITORS

- TO CONTROL COSTS AND AVOID UNEXPECTED CREDIT USAGE CAUSED BY RUNNING WAREHOUSES
- CAN BE USED TO IMPOSE LIMITS ON THE NUMBER OF CREDITS THAT ARE CONSUMED BY VIRTUAL WAREHOUSES
- WHEN LIMITS ARE REACHED AND/OR ARE APPROACHING, THE RESOURCE MONITOR CAN TRIGGER VARIOUS ACTIONS, SUCH AS SENDING ALERT NOTIFICATIONS AND/OR SUSPENDING THE WAREHOUSES.
- CAN ONLY BE CREATED BY ACCOUNT ADMINISTRATORS



PROPERTIES OF RESOURCE MONITORS

CREDIT QUOTA

- SPECIFIES THE NUMBER OF SNOWFLAKE CREDITS ALLOCATED TO THE MONITOR FOR THE SPECIFIED FREQUENCY INTERVAL
- AT THE SPECIFIED INTERVAL, THIS NUMBER RESETS BACK TO ZERO
- CREDIT QUOTA ACCOUNTS FOR CREDITS CONSUMED BY BOTH USER-MANAGED VIRTUAL WAREHOUSES AND VIRTUAL WAREHOUSES USED BY CLOUD SERVICES
- FOR EXAMPLE, YOUR RESOURCE MONITOR LIMIT IS SET AT 500 CREDITS, IF YOUR WAREHOUSE CONSUMES 300 CREDITS, AND CLOUD SERVICES CONSUME 200 CREDITS WITHIN A SPECIFIED INTERVAL OR DATE RANGE, AN ALERT WILL BE TRIGGERED

SCHEDULE

- THE DEFAULT SCHEDULE FOR A RESOURCE MONITOR SPECIFIES THAT IT STARTS MONITORING CREDIT USAGE IMMEDIATELY AND THE USED CREDITS RESET BACK TO ZERO AT THE BEGINNING OF EACH CALENDAR MONTH (I.E. THE START OF THE STANDARD SNOWFLAKE BILLING CYCLE)
- CUSTOMIZE THE SCHEDULE FOR A RESOURCE MONITOR USING THE FOLLOWING PROPERTIES

FREQUENCY: THE INTERVAL AT WHICH THE USED CREDITS RESET RELATIVE TO THE SPECIFIED START DATE AND TIME.

START: DATE AND TIME (I.E. TIMESTAMP) WHEN THE RESOURCE MONITOR STARTS MONITORING THE ASSIGNED WAREHOUSES

END: DATE AND TIME (I.E. TIMESTAMP) WHEN SNOWFLAKE SUSPENDS THE WAREHOUSES ASSOCIATED WITH THE RESOURCE MONITOR, REGARDLESS OF WHETHER THE USED CREDITS REACHED ANY OF THE THRESHOLDS DEFINED FOR THE RESOURCE MONITOR'S ACTIONS

MONITOR LEVEL

- THIS PROPERTY SPECIFIES WHETHER THE RESOURCE MONITOR IS USED TO MONITOR THE CREDIT USAGE FOR YOUR ENTIRE ACCOUNT (I.E. ALL WAREHOUSES IN THE ACCOUNT) OR A SPECIFIC SET OF INDIVIDUAL WAREHOUSES
- IF THIS PROPERTY IS NOT SET, THE RESOURCE MONITOR DOESN'T MONITOR ANY CREDIT USAGE



RESOURCE MONITOR ACTIONS/TRIGGERS



- ACTION SPECIFIES A THRESHOLD, AS A PERCENTAGE OF THE CREDIT QUOTA FOR THE RESOURCE MONITOR
- ACTION TO PERFORM WHEN THE THRESHOLD IS REACHED WITHIN THE SPECIFIED INTERVAL
- A RESOURCE MONITOR MUST HAVE AT LEAST ONE ACTION DEFINED; IF NO ACTIONS HAVE BEEN DEFINED, NOTHING HAPPENS WHEN THE USED CREDITS REACH THE THRESHOLD

NOTIFY & SUSPEND

SEND A NOTIFICATION (TO ALL ACCOUNT ADMINISTRATORS WITH NOTIFICATIONS ENABLED) AND SUSPEND ALL ASSIGNED WAREHOUSES AFTER ALL STATEMENTS BEING EXECUTED BY THE WAREHOUSE(S) HAVE COMPLETED

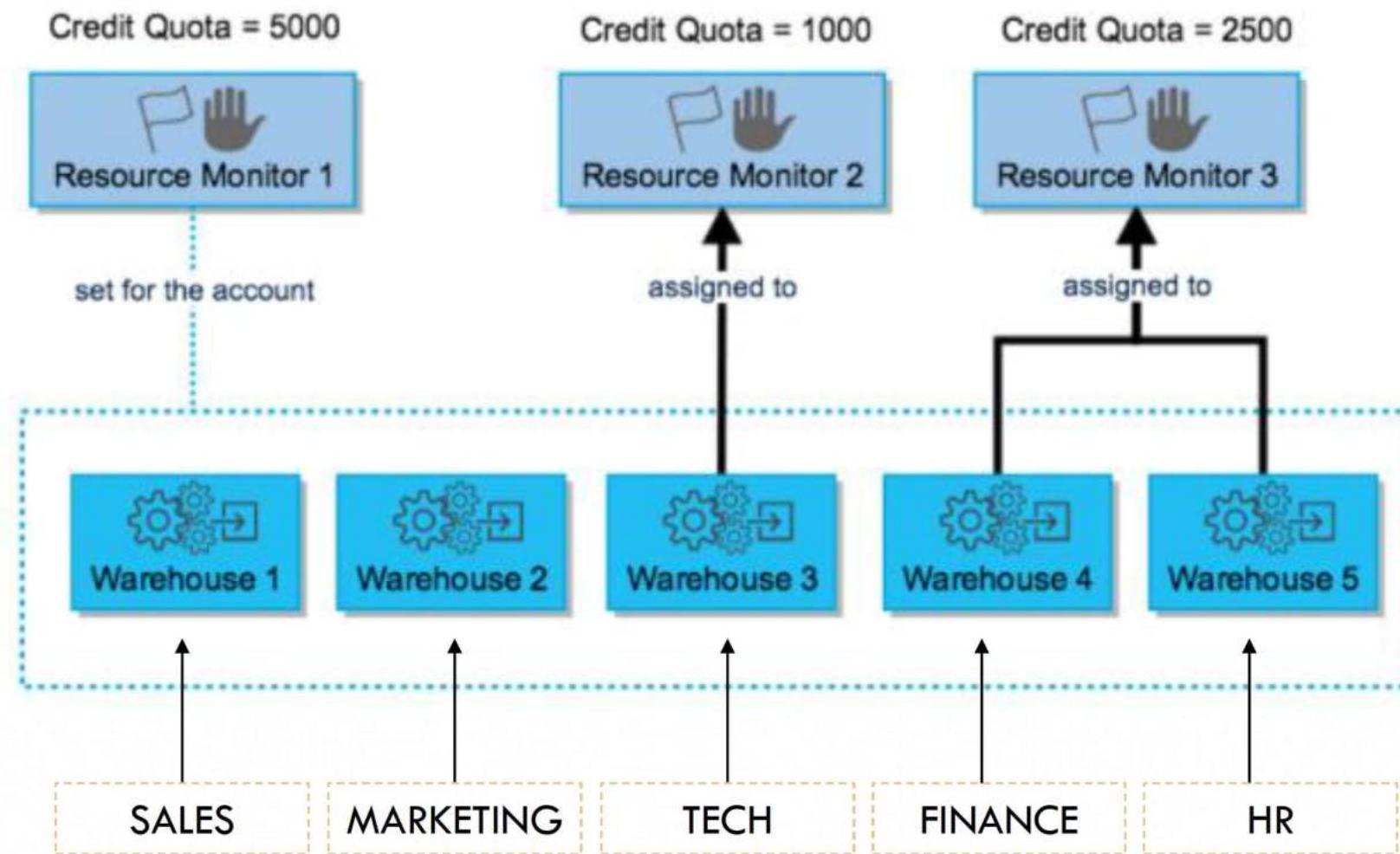
NOTIFY & SUSPEND IMMEDIATELY

SEND A NOTIFICATION (TO ALL ACCOUNT ADMINISTRATORS WITH NOTIFICATIONS ENABLED) AND SUSPEND ALL ASSIGNED WAREHOUSES IMMEDIATELY, WHICH CANCELS ANY STATEMENTS BEING EXECUTED BY THE WAREHOUSES AT THE TIME

NOTIFY

PERFORM NO ACTION, BUT SEND AN ALERT NOTIFICATION (TO ALL ACCOUNT ADMINISTRATORS WITH NOTIFICATIONS ENABLED)

EXAMPLE



An account-level resource monitor does not override resource monitor assignment for individual warehouses. If either the account resource monitor or the warehouse resource monitor reaches its defined threshold and a suspend action has been defined, the warehouse is suspended.

SUSPEND OR RESUME WAREHOUSE

- THE USED CREDITS FOR A RESOURCE MONITOR REFLECTS THE SUM OF ALL CREDITS CONSUMED BY ALL ASSIGNED WAREHOUSES WITHIN THE SPECIFIED INTERVAL
- IF A MONITOR HAS A **SUSPEND** OR **SUSPEND IMMEDIATELY** ACTION DEFINED AND ITS USED CREDITS REACH THE THRESHOLD FOR THE ACTION, ANY WAREHOUSES ASSIGNED TO THE MONITOR ARE SUSPENDED AND CANNOT BE RESUMED UNTIL ONE OF THE FOLLOWING CONDITIONS IS MET:
 - ❖ THE NEXT INTERVAL, IF ANY, STARTS, AS DICTATED BY THE START DATE FOR THE MONITOR.
 - ❖ THE CREDIT QUOTA FOR THE MONITOR IS INCREASED.
 - ❖ THE CREDIT THRESHOLD FOR THE SUSPEND ACTION IS INCREASED.
 - ❖ THE WAREHOUSES ARE NO LONGER ASSIGNED TO THE MONITOR.
 - ❖ THE MONITOR IS DROPPED.

NOTE: WHEN CREDIT QUOTA THRESHOLDS ARE REACHED FOR A RESOURCE MONITOR, THE ASSIGNED WAREHOUSES MAY TAKE SOME TIME TO SUSPEND, EVEN WHEN THE ACTION IS **SUSPEND IMMEDIATE**, THEREBY CONSUMING ADDITIONAL CREDITS

SECTION 5 – MICRO PARTITIONING

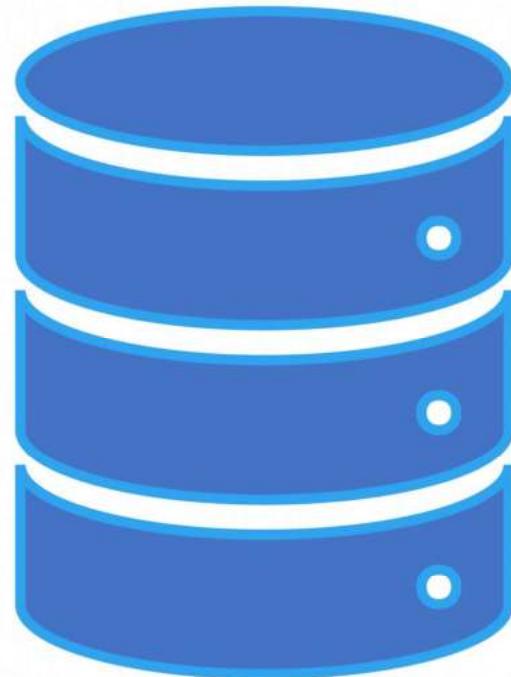
- PARTITIONING IN TRADITIONAL WAREHOUSES
- PARTITIONING IN SNOWFLAKE – MICRO PARTITIONING
- ADVANTAGES OF MICRO-PARTITIONING
- STRUCTURE OF MICRO-PARTITION
- QUERY PROCESSING IN SNOWFLAKE

PARTITIONING IN TRADITIONAL DATA WAREHOUSES

- PARTITION IS A UNIT OF MANAGEMENT THAT IS MANIPULATED INDEPENDENTLY
- STATIC PARTITIONING OF LARGE TABLES
- HELPS TO ACHIEVE ACCEPTABLE PERFORMANCE
- ENABLES BETTER SCALING
- **LIMITATIONS:**
 - MAINTENANCE OVERHEAD
 - DATA SKEWNESS
 - DISPROPORTIONATELY-SIZED PARTITIONS

PARTITIONING IN SNOWFLAKE

- UNIQUE WAY OF PARTITIONING, KNOWN AS MICRO-PARTITIONING
- HAVE ALL BENEFITS, OVERCOMES LIMITATIONS OF STATIC PARTITIONING
- MICRO-PARTITION CONTAINS BETWEEN 50 MB AND 500 MB OF UNCOMPRESSED DATA
- EACH MICRO PARTITION CONTAINS GROUP OF ROWS IN A COLUMNAR STORAGE
- SNOWFLAKE STORES METADATA ABOUT ALL ROWS STORED IN A MICRO-PARTITION
 - THE RANGE OF VALUES FOR EACH OF THE COLUMNS IN THE MICRO-PARTITION
 - THE NUMBER OF DISTINCT VALUES
 - ADDITIONAL PROPERTIES USED FOR BOTH OPTIMIZATION AND EFFICIENT QUERY PROCESSING



BENEFITS OF MICRO PARTITIONING

In contrast to traditional static partitioning, Snowflake micro-partitions are derived automatically; they don't need to be explicitly defined up-front or maintained by users.

As the name suggests, micro-partitions are small in size (50 to 500 MB, before compression), which enables extremely efficient DML and fine-grained pruning for faster queries.

Micro-partitions can overlap in their range of values, which, combined with their uniformly small size, helps prevent skew.

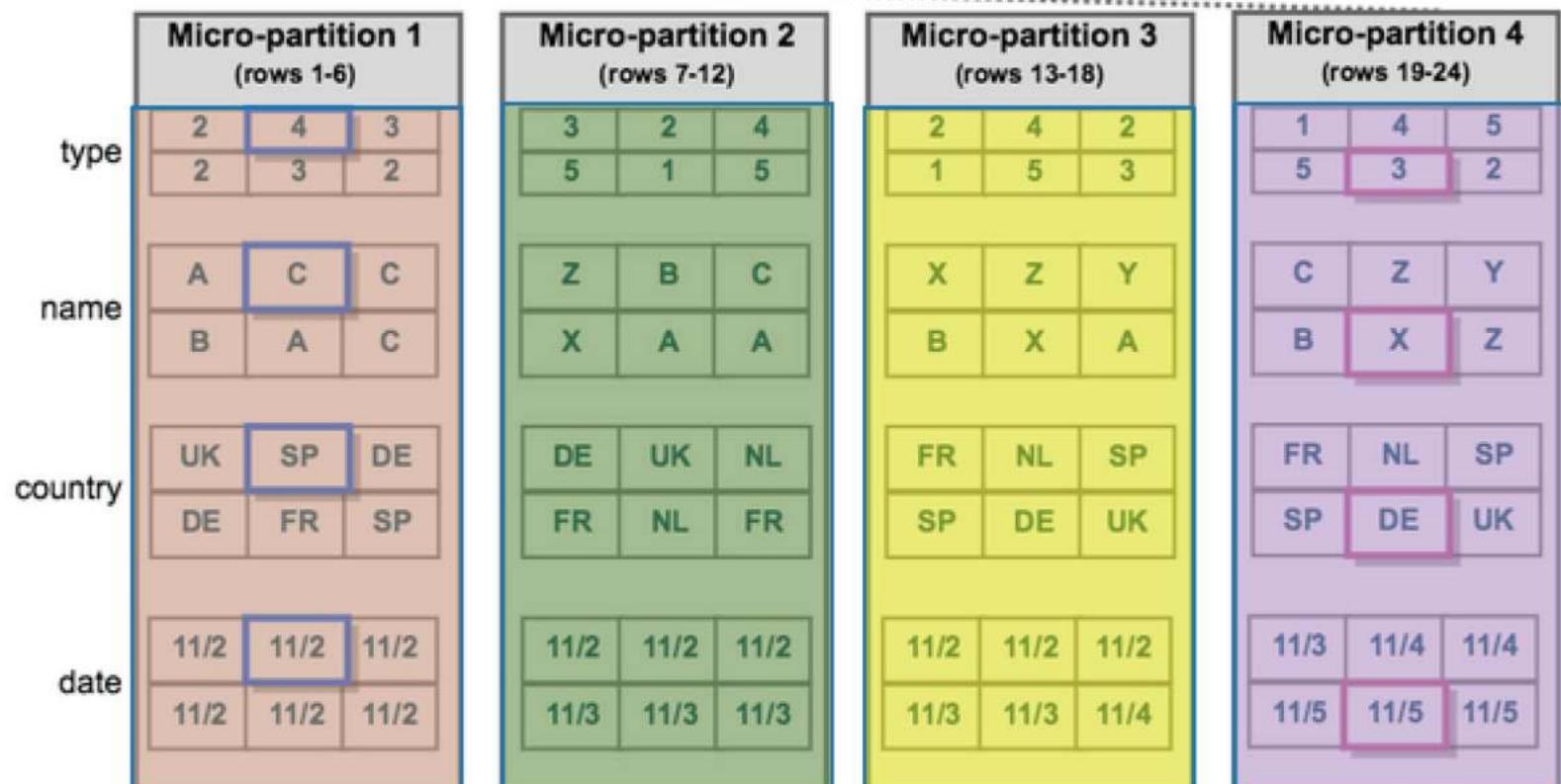
Columns are stored independently within micro-partitions, often referred to as *columnar storage*. This enables efficient scanning of individual columns; only the columns referenced by a query are scanned.

Columns are also compressed individually within micro-partitions. Snowflake automatically determines the most efficient compression algorithm for the columns in each micro-partition.

Logical Structure

type	name	country	date
2	A	UK	11/2
4	C	SP	11/2
3	C	DE	11/2
2	B	DE	11/2
3	A	FR	11/2
2	C	SP	11/2
3	Z	DE	11/2
2	B	UK	11/2
4	C	NL	11/2
5	X	FR	11/3
1	A	NL	11/3
5	A	FR	11/3
2	X	FR	11/2
4	Z	NL	11/2
2	Y	SP	11/2
1	B	SP	11/3
5	X	DE	11/3
3	A	UK	11/4
1	C	FR	11/3
4	Z	NL	11/4
5	Y	SP	11/4
5	B	SP	11/5
3	X	DE	11/5
2	Z	UK	11/5

Physical Structure



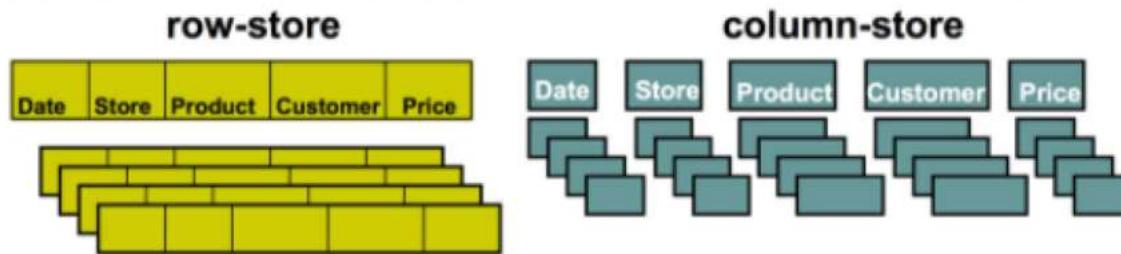
Logical Structure

type	name	country	date
2	A	UK	11/2
4	C	SP	11/2
3	C	DE	11/2
2	B	DE	11/2
3	A	FR	11/2
2	C	SP	11/2
3	Z	DE	11/2
2	B	UK	11/2
4	C	NL	11/2
5	X	FR	11/3
1	A	NL	11/3
5	A	FR	11/3
2	X	FR	11/2
4	Z	NL	11/2
2	Y	SP	11/2
1	B	SP	11/3
5	X	DE	11/3
3	A	UK	11/4
1	C	FR	11/3
4	Z	NL	11/4
5	Y	SP	11/4
5	B	SP	11/5
3	X	DE	11/5
2	Z	UK	11/5

Physical Structure

type	name	country	date
2	4	3	
2	3	2	
A	C	C	
B	A	C	
UK	SP	DE	
DE	FR	SP	
11/2	11/2	11/2	
11/2	11/2	11/2	
3	2	4	
5	1	5	
2	4	2	
1	5	3	
X	Z	Y	
B	X	A	
FR	NL	SP	
SP	DE	UK	
11/2	11/2	11/2	
11/3	11/3	11/4	
11/3	11/4	11/4	
11/5	11/5	11/5	

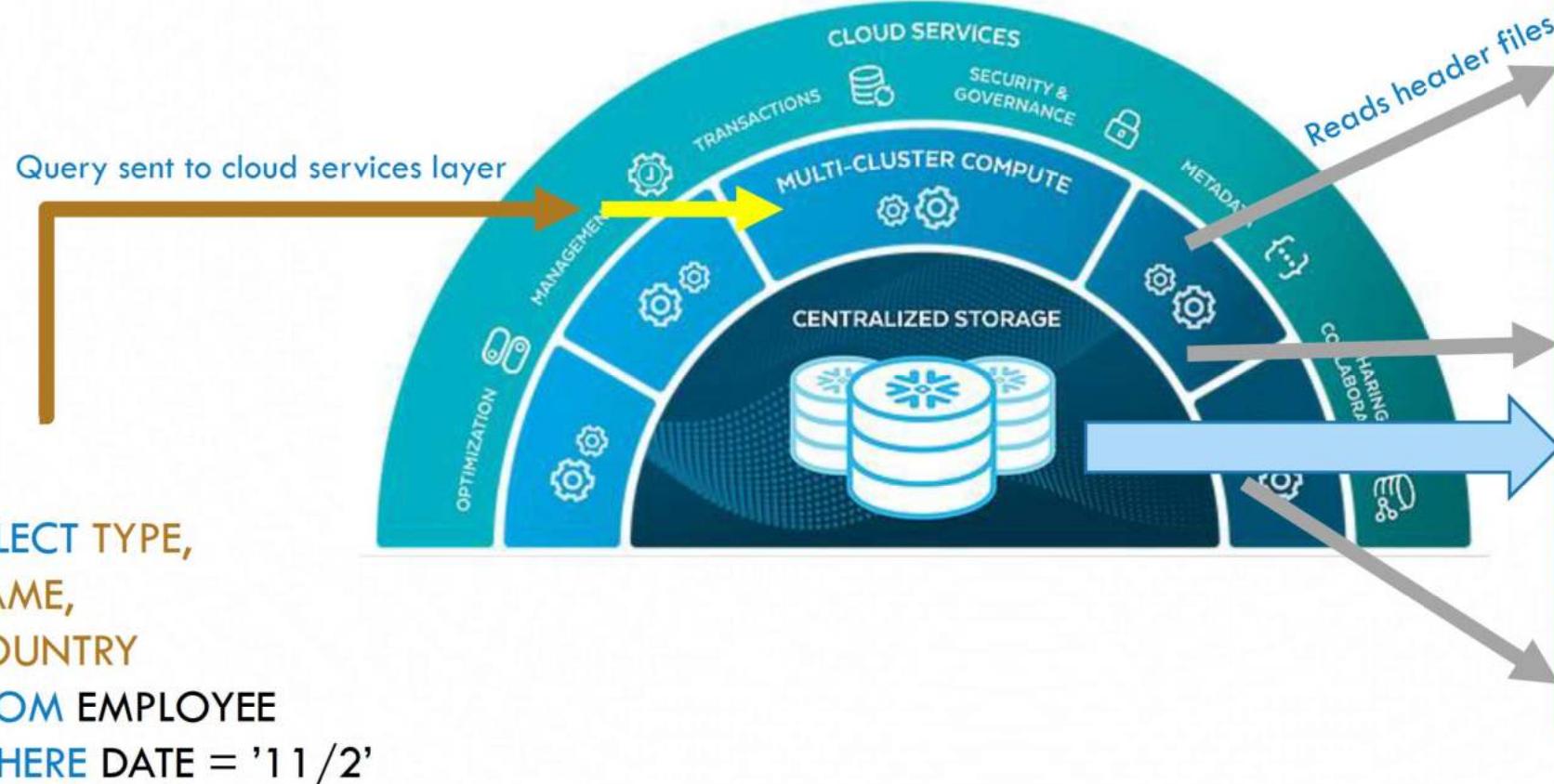
DATA FILES STORED IN COLUMNAR STORAGE



FILE HEADER (MIN, MAX, # OF DISTINCT VALUES)			
TYPE	NAME	COUNTRY	DATE
2	B	DE	11/2
3	A	FR	11/2
2	C	SP	11/2
3	X	DE	11/3
	Z	NL	11/3
	A		

Size of file will be between 50 MB and 500 MB of uncompressed data

QUERY PROCESSING ON SNOWFLAKE



MICRO PARTITIONS/TABLE FILES

FILE HEADER (MIN, MAX, # OF DISTINCT VALUES)			
TYPE	NAME	COUNTRY	DATE
2	2	DE	UK
3	4	FR	NL
2	5	SP	FR
3	1	DE	NL

FILE HEADER (MIN, MAX, # OF DISTINCT VALUES)			
TYPE	NAME	COUNTRY	DATE
2	2	DE	UK
3	4	FR	NL
2	5	SP	FR
3	1	DE	NL

FILE HEADER (MIN, MAX, # OF DISTINCT VALUES)			
TYPE	NAME	COUNTRY	DATE
2	2	DE	UK
3	4	FR	NL
2	5	SP	FR
3	1	DE	NL

MICRO-PARTITIONS SELECTION AND PRUNING

```
SELECT TYPE,  
      NAME,  
      COUNTRY  
   FROM EMPLOYEE  
 WHERE DATE = '11/2'
```

MICRO-PARTITIONS SELECTED

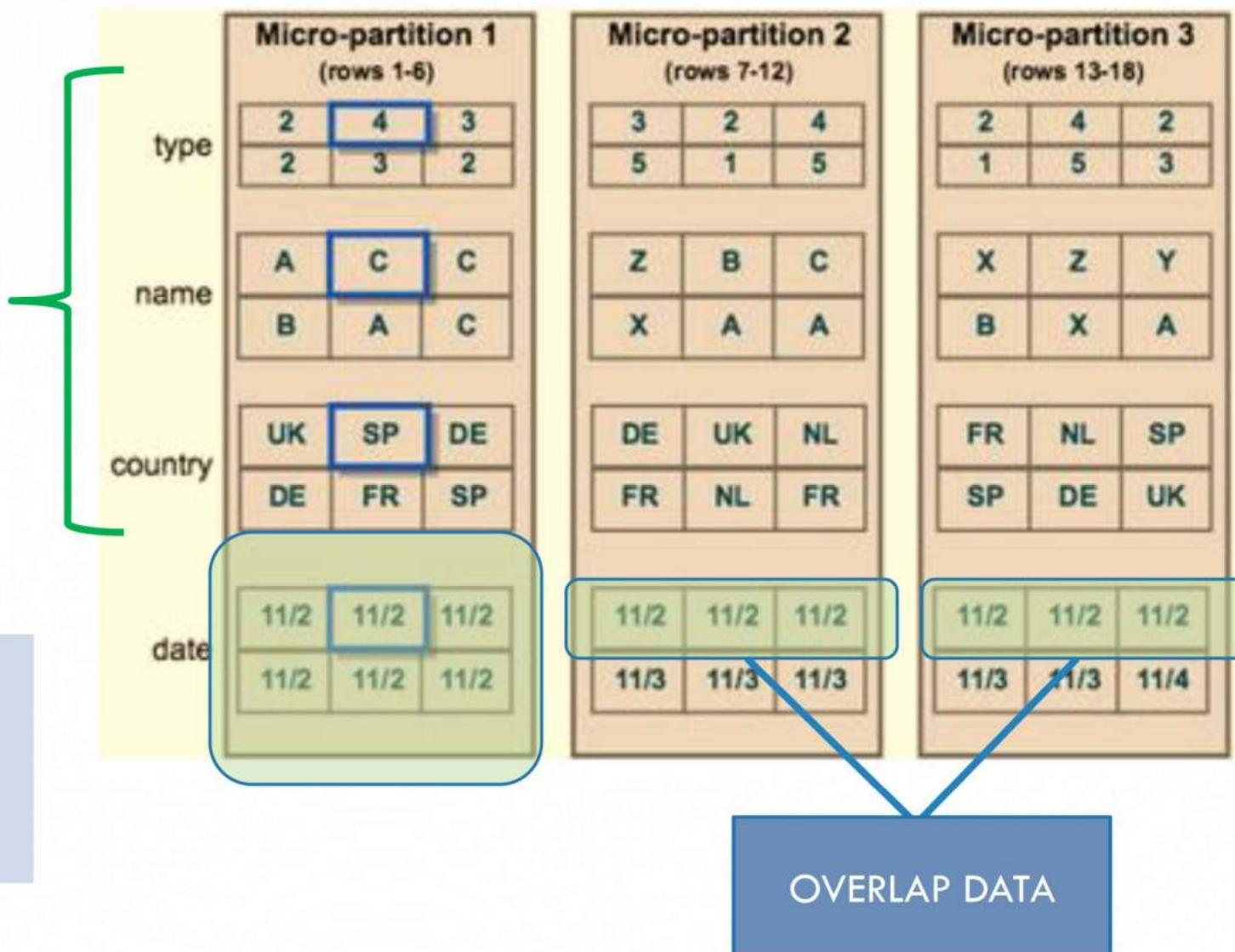
PRUNED

	Micro-partition 1 (rows 1-6)			Micro-partition 2 (rows 7-12)			Micro-partition 3 (rows 13-18)			Micro-partition 4 (rows 19-24)		
type	2	4	3	3	2	4	2	4	2	1	4	5
name	A	C	C	Z	B	C	1	5	3	5	3	2
country	UK	SP	DE	DE	UK	NL	FR	NL	SP	C	Z	Y
date	11/2	11/2	11/2	11/2	11/2	11/2	11/2	11/2	11/2	11/3	11/4	11/4
	11/2	11/2	11/2	11/3	11/3	11/3	11/3	11/3	11/4	11/5	11/5	11/5

COLUMNS SELECTION

```
SELECT TYPE,  
NAME,  
COUNTRY  
FROM EMPLOYEE  
WHERE DATE = '11/2'
```

SNOWFLAKE LOADS DATA INTO THE MICRO-PARTITIONS AS AND WHEN DATA IS INSERTED INTO THE TABLE



SINGLE MICRO-PARTITION CAN HOLD ALL THE DATA **ONLY IF MICRO-PARTITION SIZE IS BETWEEN 50-500MB**

	Micro-partition 1 (rows 1-6)			Micro-partition 2 (rows 7-12)			Micro-partition 3 (rows 13-18)		
type	2	4	3	3	2	4	2	4	2
	2	3	2	5	1	5	1	5	3
name	A	C	C	Z	B	C	X	Z	Y
	B	A	C	X	A	A	B	X	A
country	UK	SP	DE	DE	UK	NL	FR	NL	SP
	DE	FR	SP	FR	NL	FR	SP	DE	UK
date	11/2	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	11/3	11/4

SINGLE MICRO PARTITION

type	name	country	date
2	A	UK	11/2
4	C	SP	11/2
3	C	DE	11/2
2	B	DE	11/2
3	A	FR	11/2
2	C	SP	11/2
3	Z	DE	11/2
2	B	UK	11/2
4	C	NL	11/2
2	X	FR	11/2
4	Z	NL	11/2
2	Y	SP	11/2

SECTION 6 – CLUSTERING

- CLUSTERING INTRODUCTION
- CLUSTERING KEYS
- CLUSTERING METADATA
- CLUSTERING DEPTH
- AUTOMATIC RE-CLUSTERING

DATA CLUSTERING

- A PROCESS TO OPTIMIZE DATA RETRIEVAL
- DATA STORED IN TABLES IS SORTED/ORDERED ALONG NATURAL DIMENSIONS
- **CLUSTERING** IS A KEY FACTOR IN QUERIES TO SORT/ORDER THE DATA FOR FASTER DATA RETRIEVAL
- TABLE DATA THAT IS NOT SORTED OR IS ONLY PARTIALLY SORTED MAY IMPACT QUERY PERFORMANCE, PARTICULARLY ON VERY LARGE TABLES
- DATA IS INSERTED/LOADED INTO A TABLE AND CLUSTERING METADATA IS COLLECTED AND RECORDED FOR EACH MICRO-PARTITION
- USING CLUSTERING INFORMATION, SNOWFLAKE AVOIDS UNNECESSARY SCANNING OF MICRO-PARTITIONS DURING QUERYING WHICH SIGNIFICANTLY ACCELERATES THE PERFORMANCE OF QUERIES

CLUSTERING KEYS

- SUBSET OF COLUMNS IN A TABLE (OR EXPRESSIONS ON A TABLE)
- EXPLICITLY DESIGNATED TO CO-Locate THE DATA IN THE TABLE IN THE SAME MICRO-PARTITIONS
- USEFUL FOR VERY LARGE TABLES
 - IMPROVES SCAN EFFICIENCY IN QUERIES
 - NO ADDITIONAL ADMINISTRATION IS REQUIRED
 - FUTURE MAINTENANCE PERFORMED AUTOMATICALLY BY SNOWFLAKE
- WHEN TO DEFINE CLUSTERING KEY:
 - QUERIES ON THE TABLE ARE **RUNNING SLOWER THAN EXPECTED OR HAVE NOTICEABLY DEGRADED OVER TIME.**
 - THE **CLUSTERING DEPTH** FOR THE TABLE IS **LARGE**

NOTE: CLUSTERING CAN BE COMPUTATIONALLY EXPENSIVE SO ONLY CLUSTER WHEN QUERIES WILL BENEFIT SUBSTANTIALLY FROM THE CLUSTERING

Logical Structure

type	name	country	date
2	A	UK	11/2
4	C	SP	11/2
3	C	DE	11/2
2	B	DE	11/2
3	A	FR	11/2
2	C	SP	11/2
3	Z	DE	11/2
2	B	UK	11/2
4	C	NL	11/2
5	X	FR	11/3
1	A	NL	11/3
5	A	FR	11/3
2	X	FR	11/2
4	Z	NL	11/2
2	Y	SP	11/2
1	B	SP	11/3
5	X	DE	11/3
3	A	UK	11/4
1	C	FR	11/3
4	Z	NL	11/4
5	Y	SP	11/4
5	B	SP	11/5
3	X	DE	11/5
2	Z	UK	11/5

Physical Structure

type	Micro-partition 1 (rows 1-6)			Micro-partition 2 (rows 7-12)			Micro-partition 3 (rows 13-18)			Micro-partition 4 (rows 19-24)		
name	2	4	3	3	2	4	2	4	2	1	4	5
country	A	C	C	Z	B	C	X	Z	Y	C	Z	Y
date	UK	SP	DE	DE	UK	NL	FR	NL	SP	FR	NL	SP
	DE	FR	SP	FR	NL	FR	SP	DE	UK	SP	DE	UK
	11/2	11/2	11/2	11/2	11/2	11/2	11/2	11/2	11/2	11/3	11/4	11/4
	11/2	11/2	11/2	11/3	11/3	11/3	11/3	11/3	11/4	11/5	11/5	11/5

- 24 ROWS STORED ACROSS 4 MICRO-PARTITIONS
- ROWS DIVIDED EQUALLY BETWEEN EACH MICRO-PARTITION
- DATA IS SORTED AND STORED BY COLUMN
- SNOWFLAKE PERFORMS FOLLOWING ACTIONS:
 1. PRUNE MICRO-PARTITIONS THAT ARE NOT NEEDED FOR THE QUERY
 2. PRUNE BY COLUMN WITHIN THE REMAINING MICRO-PARTITIONS

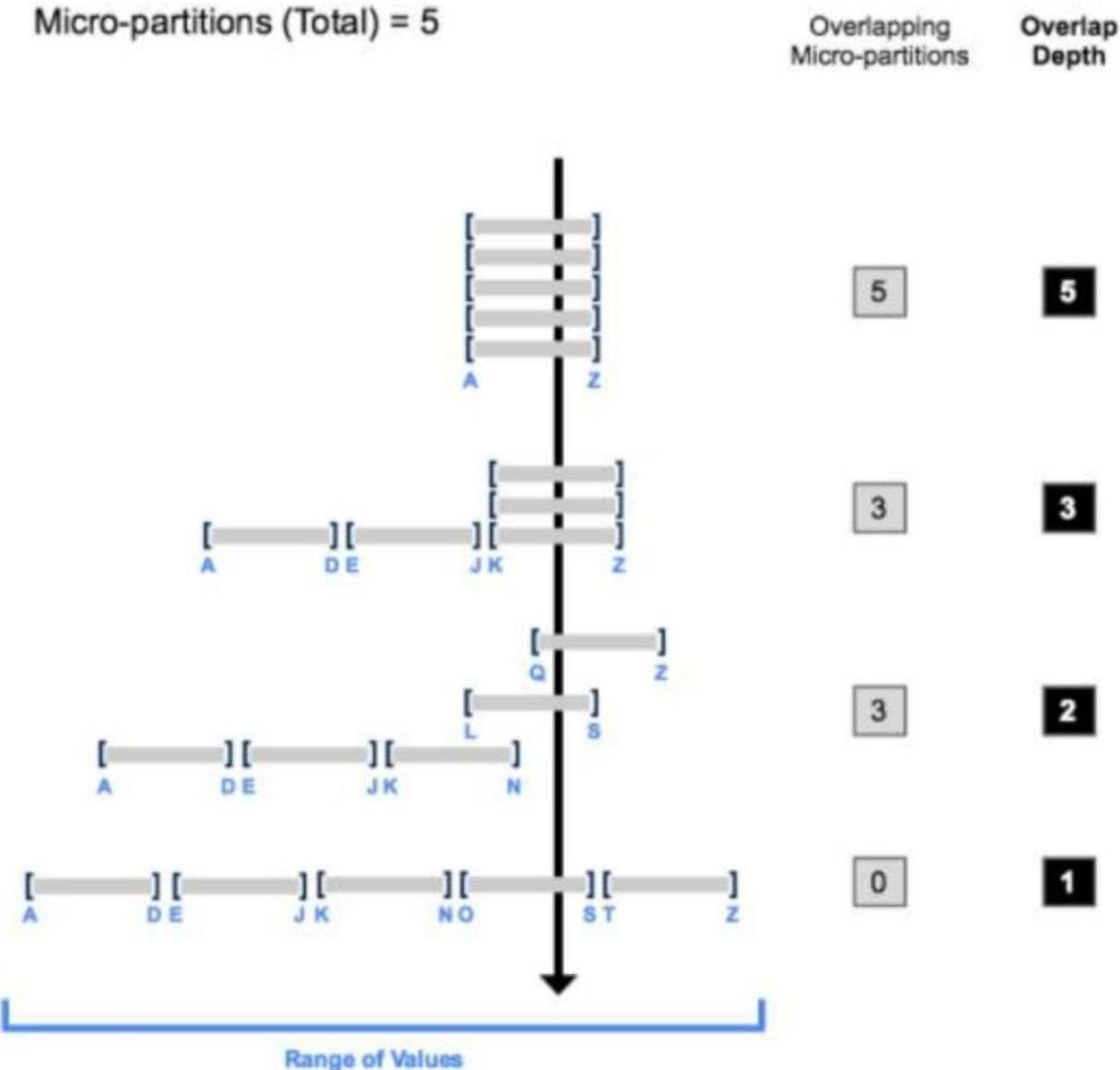
SNOWFLAKE MAINTAINS CLUSTERING METADATA FOR THE MICRO-PARTITIONS IN A TABLE

- THE TOTAL NUMBER OF MICRO-PARTITIONS THAT COMprise THE TABLE.
- THE NUMBER OF MICRO-PARTITIONS CONTAINING VALUES THAT OVERLAP WITH EACH OTHER.
- THE DEPTH OF THE OVERLAPPING MICRO-PARTITIONS.

CLUSTERING DEPTH

- MEASURES THE **AVERAGE DEPTH** (1 OR GREATER) OF THE OVERLAPPING MICRO-PARTITIONS FOR SPECIFIED COLUMNS IN A TABLE
- SMALLER THE AVERAGE DEPTH, THE BETTER CLUSTERED THE TABLE
- ADVANTAGES OF USING CLUSTERING DEPTH:
 - MONITORING THE CLUSTERING “HEALTH” OF A LARGE TABLE, PARTICULARLY OVER TIME AS DML IS PERFORMED ON THE TABLE.
 - DETERMINING WHETHER A LARGE TABLE WOULD BENEFIT FROM EXPLICITLY DEFINING A CLUSTERING KEY.
- TABLE WITH NO MICRO-PARTITIONS HAS A CLUSTERING DEPTH OF ZERO
- CLUSTERING DEPTH IS NOT A PERFECT MEASURE TO CHECK IF TABLE IS WELL-CLUSTERED. RATHER, CHECK QUERIES PERFORMANCE:
 - IF QUERIES ON A TABLE ARE PERFORMING AS NEEDED OR EXPECTED, THE TABLE IS LIKELY WELL-CLUSTERED.
 - IF QUERY PERFORMANCE DEGRADES OVER TIME, THE TABLE IS LIKELY NO LONGER WELL-CLUSTERED AND MAY BENEFIT FROM CLUSTERING.

Micro-partitions (Total) = 5



RE-CLUSTERING



AS DML OPERATIONS ([INSERT](#), [UPDATE](#), [DELETE](#), [MERGE](#), [COPY](#)) ARE PERFORMED ON A CLUSTERED TABLE, THE DATA IN THE TABLE MIGHT BECOME [LESS CLUSTERED](#). PERIODIC/REGULAR RECLUSTERING OF THE TABLE IS REQUIRED TO MAINTAIN OPTIMAL CLUSTERING.



DURING RECLUSTERING, SNOWFLAKE [USES THE CLUSTERING KEY](#) FOR A CLUSTERED TABLE TO REORGANIZE THE COLUMN DATA, SO THAT RELATED RECORDS ARE RELOCATED TO THE SAME MICRO-PARTITION.

Logical Structure

type	name	country	date
2	A	UK	11/2
4	C	SP	11/2
3	C	DE	11/2
2	B	DE	11/2
3	A	FR	11/2
2	C	SP	11/2
3	Z	DE	11/2
2	B	UK	11/2
4	C	NL	11/2
5	X	FR	11/3
1	A	NL	11/3
5	A	FR	11/3
2	X	FR	11/2
4	Z	NL	11/2
2	Y	SP	11/2
1	B	SP	11/3
5	X	DE	11/3
3	A	UK	11/4
1	C	FR	11/3
4	Z	NL	11/4
5	Y	SP	11/4
5	B	SP	11/5
3	X	DE	11/5
2	Z	UK	11/5

```
SELECT name, country FROM t1  
WHERE type = 2  
AND date = '11/2';
```

Physical Structure

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	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				
country				
1	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

ALTER TABLE t1
CLUSTER BY (date, type);

New Micro-partitions (After Reclustering)

	Micro-partition 5 (rows 1, 4, 6, 8, 13, 15)	Micro-partition 6 (rows 3, 5, 7, 2, 9, 14)	Micro-partition 7 (rows 10, 12, 17, 11, 16, 19)	Micro-partition 8 (rows 18, 20-24)
type	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				
country				
date	11/2 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 11/3 11/3	11/4 11/4 11/4 11/5 11/5 11/5

SECTION 7 – QUERY HISTORY AND CACHING

- QUERY HISTORY
- SQL QUERIES TO FETCH QUERY HISTORY
- CACHING INTRODUCTION
- CACHING – DETAILED EXAMPLES
- TYPES OF CACHES

QUERY HISTORY

AVAILABLE FROM THE RIBBON ITEM – **HISTORY**

ALLOWS VIEWING OF 14 DAYS OF QUERY HISTORY

VARIOUS OPTIONS TO FILTER – BY USER,
WAREHOUSE, ETC.

ALLOWS TO DOWNLOAD PREVIOUS 24 HOURS
RESULT SETS

OTHER PEOPLE QUERIES CAN BE VIEWED BUT NOT
THE RESULTS

CACHING

3 TYPES OF CACHES EXISTS ON SNOWFLAKE

CACHES PLAYS VITAL ROLE IN SAVING COSTS AND SPEEDING UP RESULTS

METADATA CACHE – HOLDS OBJECT INFORMATION AND UPDATED STATISTICS

RESULTS CACHE – HOLDS RESULTS UPTO 24HOURS

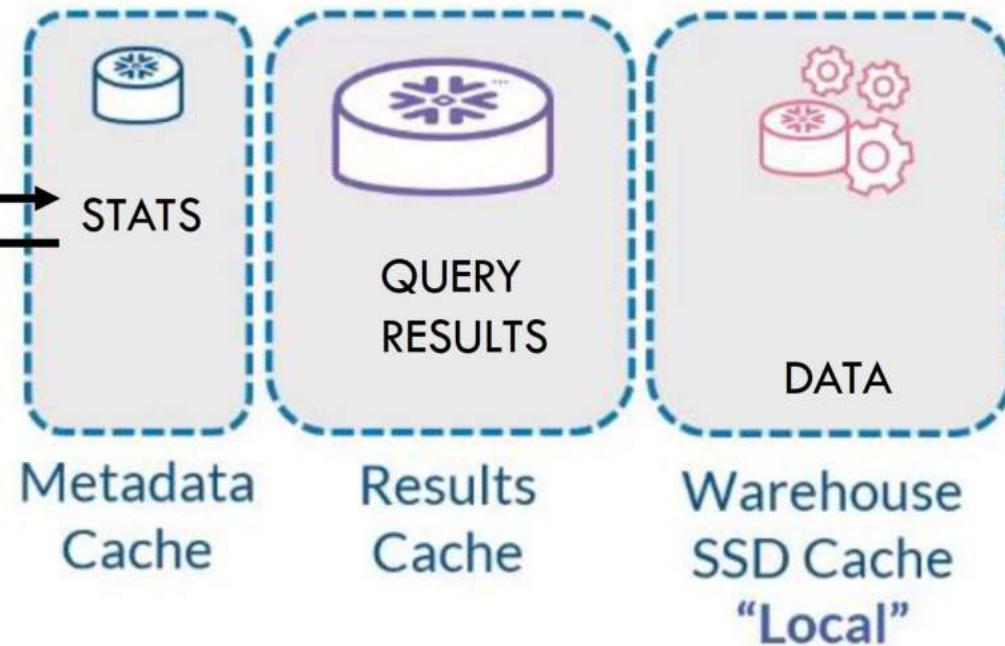
WAREHOUSE CACHE – HOLDS DATA IN LOCAL AS LONG AS WAREHOUSE IS RUNNING

HISTORY AREA OF WEBUI HELPS TO VIEW USE OF CACHES

USERS CANNOT SEE OTHER USER'S RESULTS, HOWEVER RESULTS CACHE CAN RESUSE ONE'S RESULT CACHE TO PRESENT IT TO ANOTHER USER

CACHING

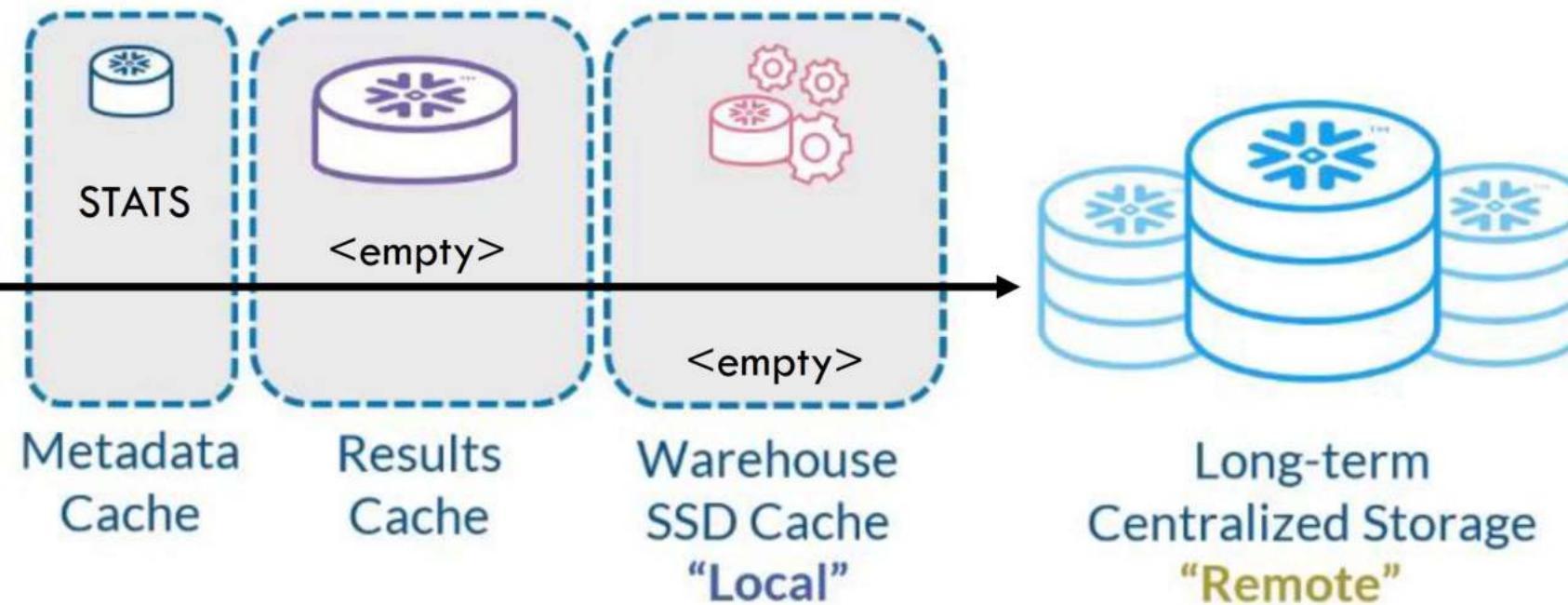
```
SELECT COUNT(*)  
FROM  
"SNOWFLAKE_SAMPLE_DATA"."TPCDS_SF100TCL".  
"CALL_CENTER"
```



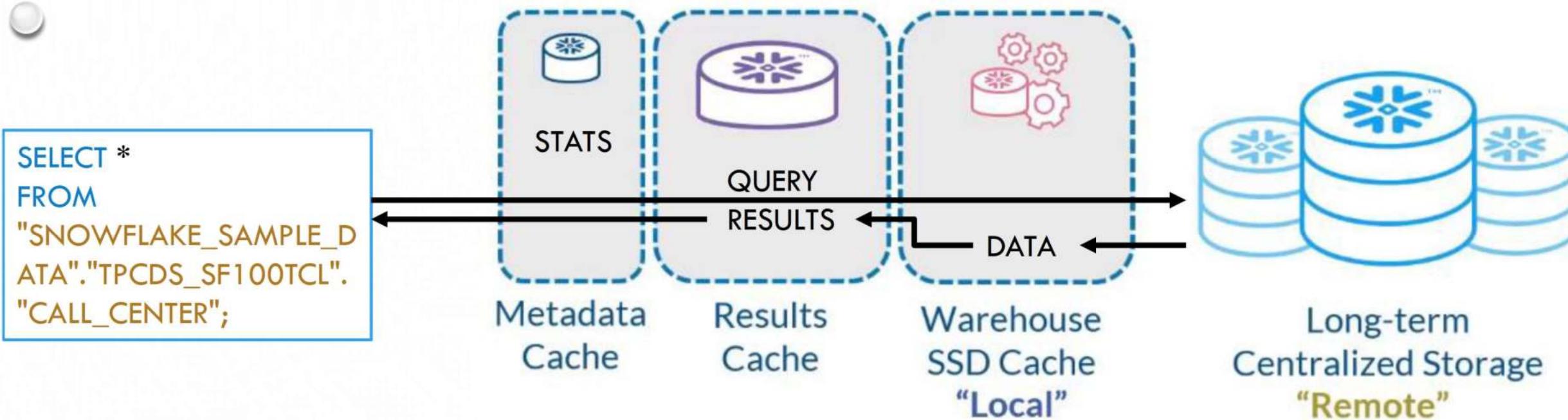
Long-term
Centralized Storage
“Remote”

CACHING

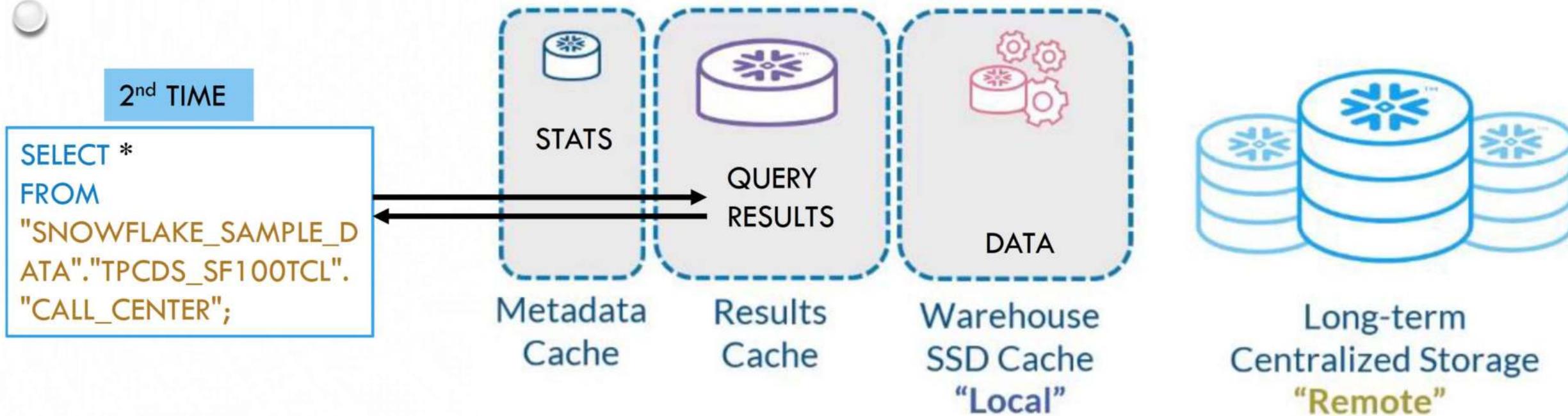
```
SELECT *\nFROM\n"SNOWFLAKE_SAMPLE_DATA"."TPCDS_SF100TCL".\n"CALL_CENTER";
```



CACHING



CACHING

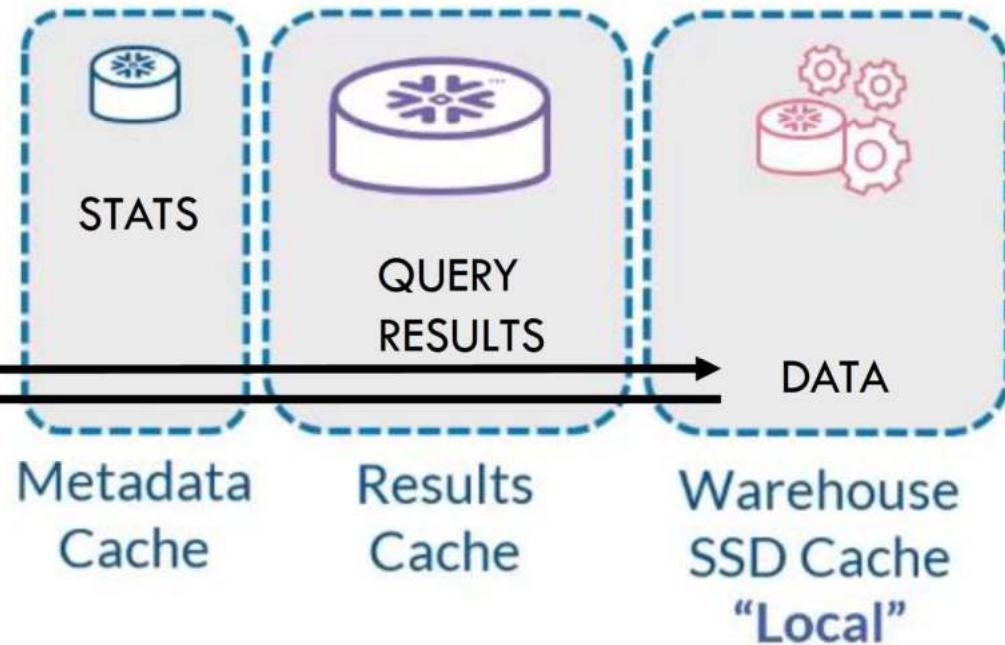


QUERY RESULTS ARE REUSED IF ALL OF THE FOLLOWING CONDITIONS ARE MET:

- THE NEW QUERY SYNTACTICALLY MATCHES THE PREVIOUSLY-EXECUTED QUERY.
- THE TABLE DATA CONTRIBUTING TO THE QUERY RESULT HAS NOT CHANGED.
- THE PERSISTED RESULT FOR THE PREVIOUS QUERY IS STILL AVAILABLE.
- THE ROLE ACCESSING THE CACHED RESULTS HAS THE REQUIRED PRIVILEGES.
 - IF THE QUERY WAS A SELECT QUERY, THE ROLE EXECUTING THE QUERY MUST HAVE THE NECESSARY ACCESS PRIVILEGES FOR ALL THE TABLES USED IN THE CACHED QUERY.
 - IF THE QUERY WAS A SHOW QUERY, THE ROLE EXECUTING THE QUERY MUST MATCH THE ROLE THAT GENERATED THE CACHED RESULTS.
- THE QUERY DOES NOT INCLUDE FUNCTIONS THAT MUST BE EVALUATED AT EXECUTION TIME, E.G. CURRENT_TIMESTAMP AND UUID_STRING. (NOTE THAT THE CURRENT_DATE() FUNCTION IS AN EXCEPTION TO THIS RULE; EVEN THOUGH CURRENT_DATE() IS EVALUATED AT EXECUTION TIME, QUERIES THAT USE CURRENT_DATE() CAN STILL USE THE QUERY REUSE FEATURE.)
- THE TABLE'S MICRO-PARTITIONS HAVE NOT CHANGED (E.G. BEEN RECLUSTERED OR CONSOLIDATED) DUE TO CHANGES TO OTHER DATA IN THE TABLE.

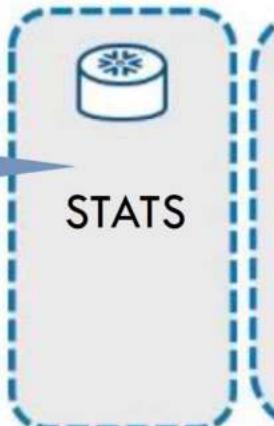
CACHING

```
SELECT  
CC_CALL_CENTER_SK AS  
CC_ID  
FROM  
"SNOWFLAKE_SAMPLE_D  
ATA"."TPCDS_SF100TCL".  
"CALL_CENTER";
```



DATA RETENTION IN CACHES

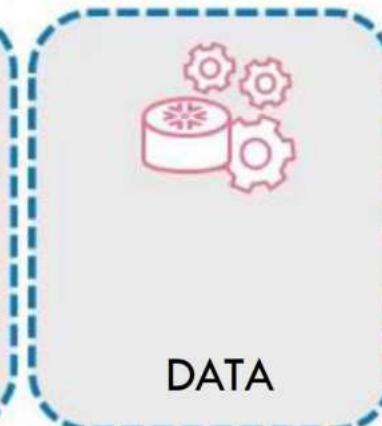
NEVER DROPPED.
STORES UP-TO-DATE
STATISTICS ALWAYS



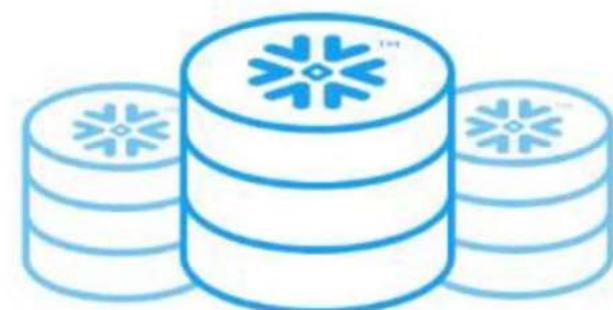
Metadata Cache



Results Cache



Warehouse
SSD Cache
“Local”



Long-term
Centralized Storage
“Remote”

RESULT SET EXPIRES
AFTER 24HOURS OF
NON-USAGE OR IF
UNDERLYING DATA
HAS CHANGED

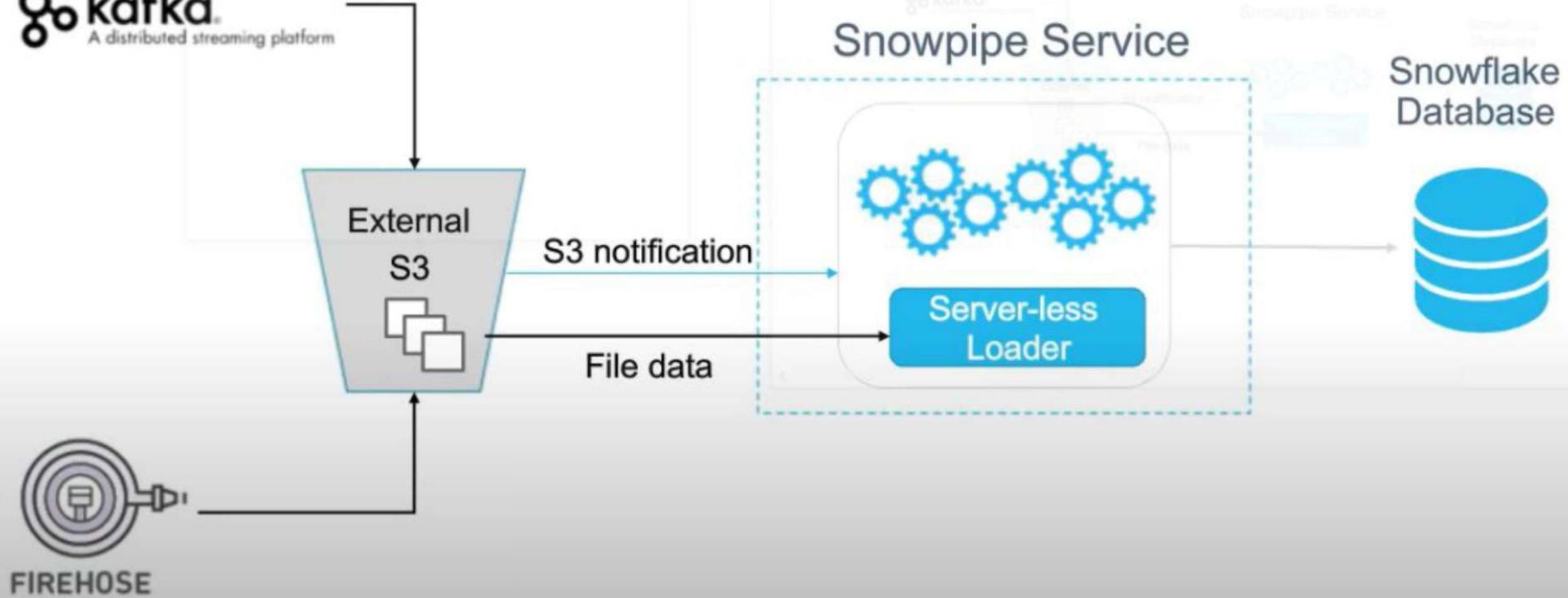
DROPPED AS SOON AS
WAREHOUSE IS
SUSPENDED

SECTION 10 – SNOWPIPE

- SNOWPIPE INTRODUCTION



Apache
kafka
A distributed streaming platform



SNOWPIPE A CONTINUOUS
INGESTION SERVICE

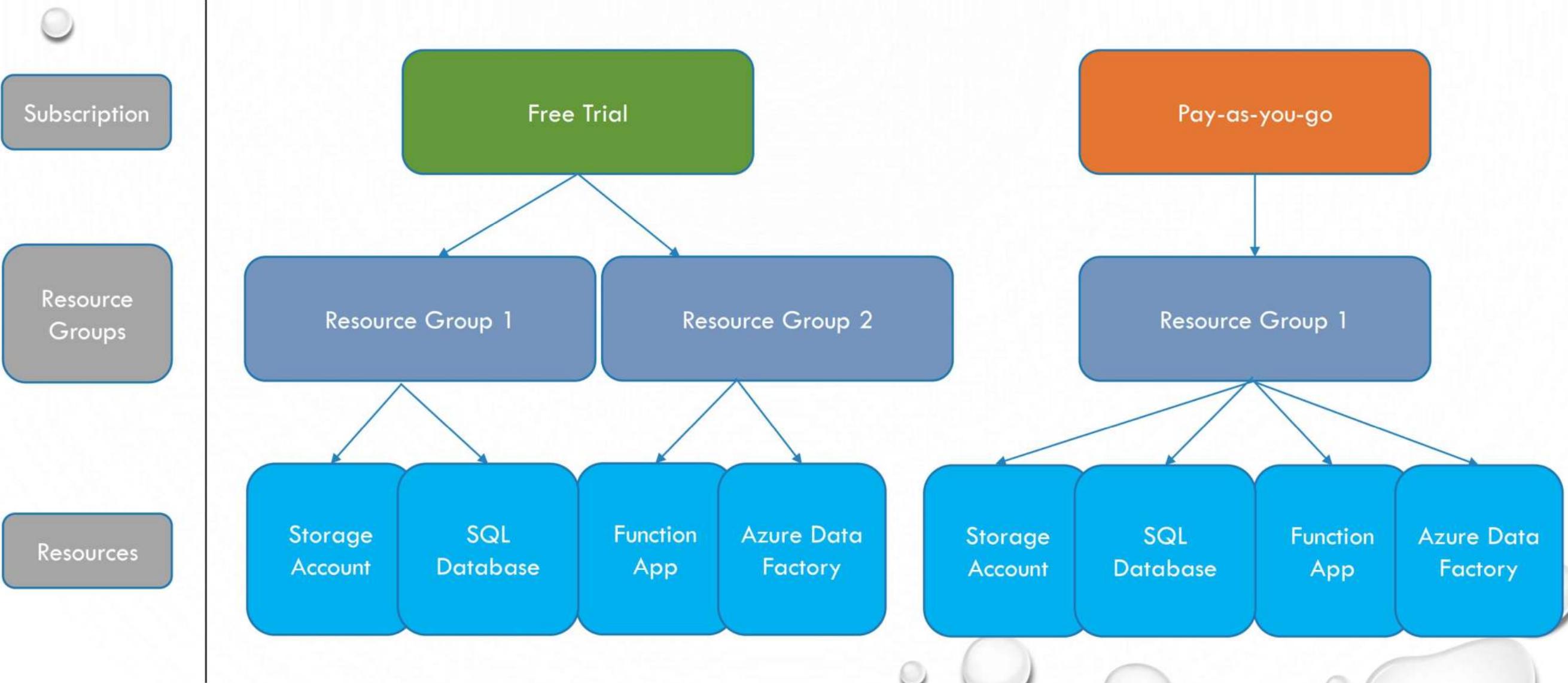
SECTION 11 – LOAD DATA FROM AZURE

- AZURE COMPONENTS
- AZURE INFRASTRUCTURE HIERARCHY

LOAD DATA FROM AZURE

- CREATE AZURE FREE TRIAL ACCOUNT
- SETUP DATA LAKE AND ITS OBJECTS
- CONFIGURE SNOWFLAKE TO INJECT DATA FROM AZURE INTO TABLES
- LOAD CSV DATA FROM AZURE
- LOAD PARQUET DATA FROM AZURE

AZURE INFRASTRUCTURE HIERARCHY



SECTION 12 – SNOWFLAKE TABLES

- TYPES OF TABLES AVAILABLE IN SNOWFLAKE
- PERMANENT TABLES
- TRANSIENT TABLES
- TEMPORARY TABLES

TYPES OF OBJECTS IN SNOWFLAKE

PERMANENT

TEMPORARY

TRANSIENT

PERMANENT TABLES



default table type
when creating any
object on snowflake



created for longer
period of time



level of data
protection and
recovery mechanisms
are implemented



have high number of
time travel retention
days in comparison to
transient & temporary
tables



have a fail-safe
period of 7 days

TEMPORARY TABLES



stores non-permanent data



temporary tables only exist within the session in which they were created and persist only for the remainder of the session



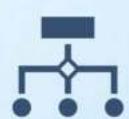
not visible to other users or sessions



do not support some standard features such as cloning



once the session ends, data stored in the table is purged completely and is not recoverable, either by the user who created the table or by snowflake



can create temporary and non-temporary tables with the same name within the same schema however temporary table takes precedence over any other table

TRANSIENT TABLES



persists until explicitly dropped



transient tables are similar to permanent tables with the key difference that they do not have a fail-safe period



transient tables are specifically designed for data that do not need the same level of data protection and recovery as provided by permanent tables



transient tables contribute to the overall storage charges that snowflake bills, similar to permanent tables



there are no fail-safe costs



can create transient databases and schemas as well



all tables created in a transient schema, as well as all schemas created in a transient database, are transient by definition

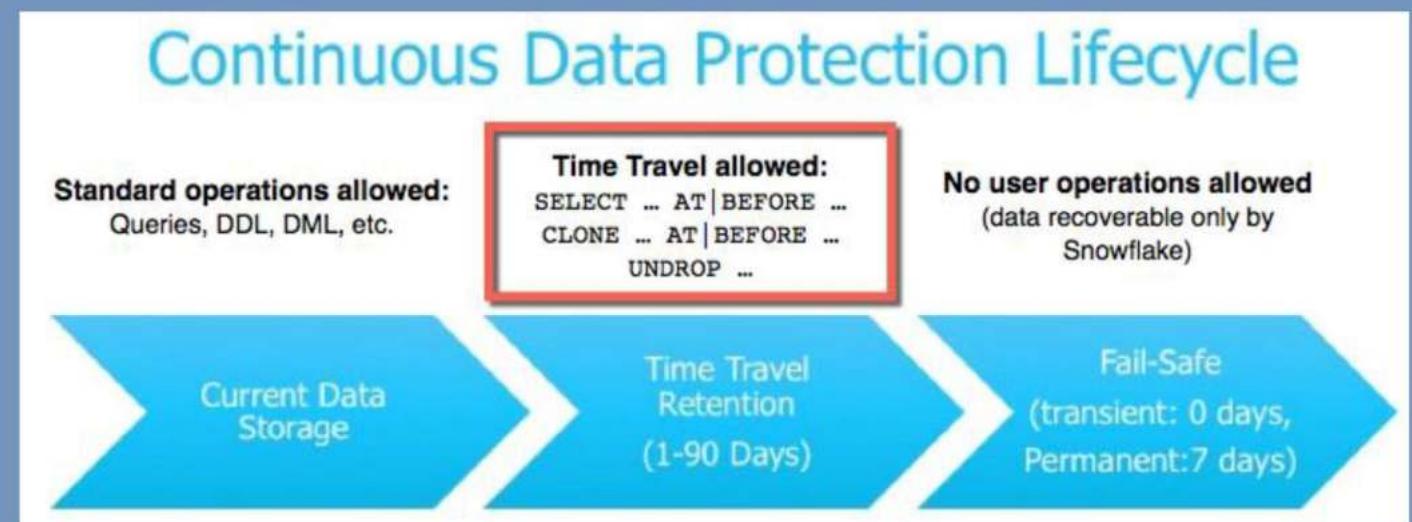
SECTION 13 - TIME-TRAVEL

- TIME TRAVEL UNDERSTANDING
- SET DATA RETENTION TIMINGS
- QUERY HISTORICAL DATA
- CLONING HISTORICAL OBJECTS
- TIME TRAVEL FOR DATA RECOVERY

TIME TRAVEL

USING TIME TRAVEL, FOLLOWING ACTIONS CAN BE PERFORMED WITHIN A DEFINED PERIOD OF TIME :

- RUN QUERIES ON DATA THAT HAS SINCE BEEN UPDATED OR DELETED IN THE PAST.
- CREATE CLONES OF ENTIRE TABLES, SCHEMAS, AND DATABASES AT OR BEFORE SPECIFIC POINTS IN THE PAST.
- RESTORE TABLES, SCHEMAS, AND DATABASES THAT HAVE BEEN DROPPED.

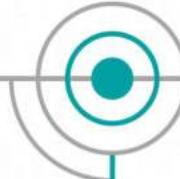


TIMELINE

INSERTED 5 RECORDS

DELETED 1 RECORD

DAY 1



CREATED EMPLOYEES TABLE
&
INSERTED 3 RECORDS

DAY 5



DAY 10



UPDATED 2 RECORDS



DAY 12

DAY 16



INSERTED 2 NEW RECORDS
&
UPDATED 1 RECORD

TIME TRAVEL – DATA RETENTION PERIOD

- KEY COMPONENT OF SNOWFLAKE TIME TRAVEL
- SPECIFIES THE NUMBER OF DAYS FOR WHICH HISTORICAL DATA IS PRESERVED
- ENABLES TIME TRAVEL MECHANISM TO PERFORM OPERATIONS (SELECT, CREATE, CLONE, UNDROP) ON THE DATA
- STANDARD RETENTION PERIOD IS 1 DAY (24 HOURS) AND IS AUTOMATICALLY ENABLED FOR ALL SNOWFLAKE ACCOUNTS
- FOR SNOWFLAKE STANDARD EDITION, THE RETENTION PERIOD CAN BE SET TO 0 (OR UNSET BACK TO THE DEFAULT OF 1 DAY) AT THE ACCOUNT AND OBJECTS* LEVEL
- FOR SNOWFLAKE ENTERPRISE EDITION (AND HIGHER):
 - FOR TRANSIENT OBJECTS*, THE RETENTION PERIOD CAN BE SET TO 0 (OR UNSET BACK TO THE DEFAULT OF 1 DAY). THE SAME IS ALSO TRUE FOR TEMPORARY TABLES.
 - FOR PERMANENT OBJECTS*, THE RETENTION PERIOD CAN BE SET TO ANY VALUE FROM 0 UP TO 90 DAYS.
- USERS WITH THE ACCOUNTADMIN ROLE ONLY CAN SET DATA_RETENTION_TIME_IN_DAYS VALUE
- ONCE THE DEFINED PERIOD HAS ELAPSED, THE DATA IS MOVED INTO SNOWFLAKE FAIL-SAFE

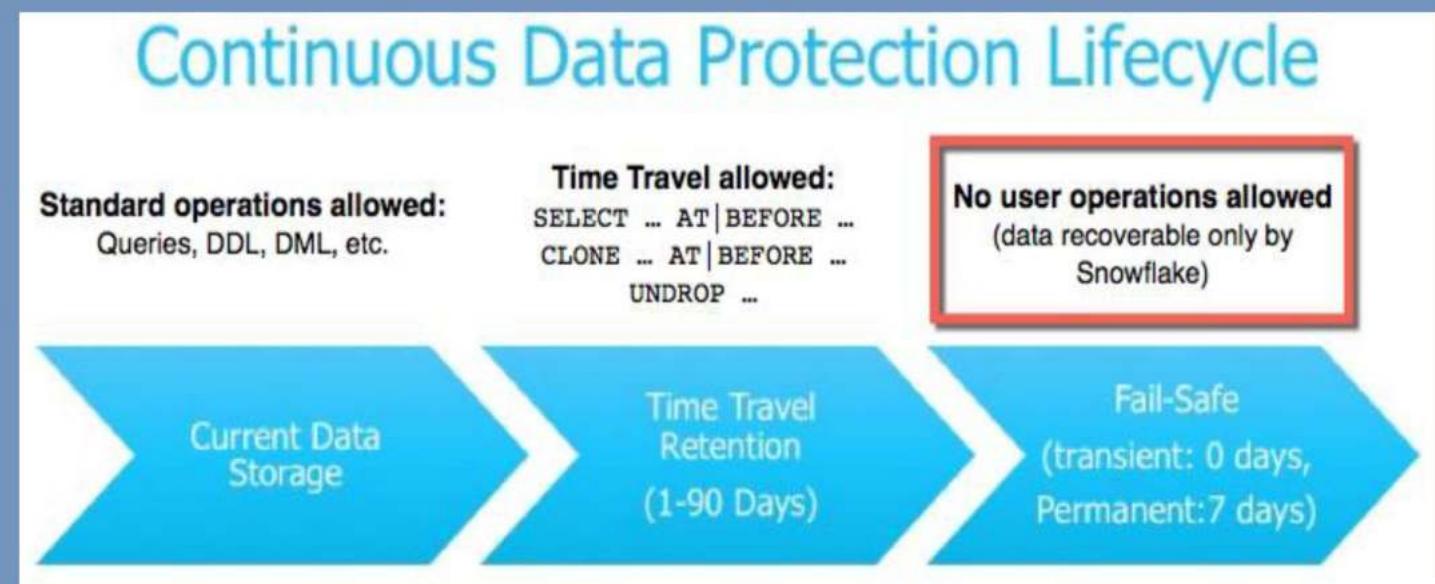
*NOTE: OBJECTS REFERRED TO AS DATABASES, SCHEMAS, AND TABLES

SECTION 14 - FAILSAFE

- DETAILED OVERVIEW
- FEATURES BY SNOWFLAKE

FAIL SAFE

- FAIL-SAFE ENSURES HISTORICAL DATA IS PROTECTED IN THE EVENT OF A SYSTEM FAILURE OR ANY DISASTER
- PROVIDES A (NON-CONFIGURABLE) 7-DAY PERIOD DURING WHICH HISTORICAL DATA IS RECOVERABLE BY SNOWFLAKE
- PERIOD STARTS IMMEDIATELY AFTER THE TIME TRAVEL RETENTION PERIOD ENDS.
- FAIL-SAFE IS **NOT** PROVIDED AS A MEANS FOR ACCESSING HISTORICAL DATA AFTER THE TIME TRAVEL RETENTION PERIOD HAS ENDED.
- IT IS FOR USE **ONLY** BY SNOWFLAKE TO RECOVER DATA THAT MAY HAVE BEEN LOST OR DAMAGED DUE TO EXTREME OPERATIONAL FAILURES.

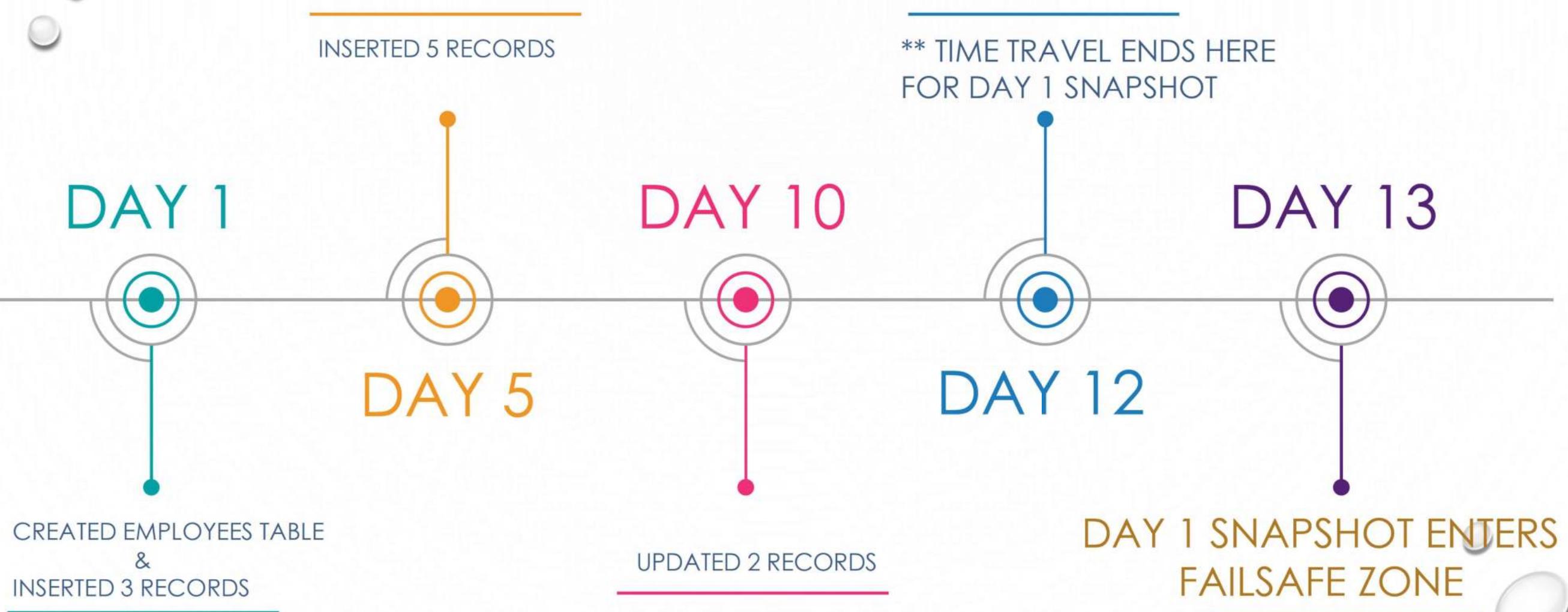


WHY FAIL SAFE OVER BACKUP?

“DATA CORRUPTION OR LOSS CAN HAPPEN WITH ANY DATABASE MANAGEMENT SYSTEM”

- TO MITIGATE THE RISK, DBA USED TO PERFORM FULL AND INCREMENTAL BACKUPS, BUT THIS CAN DOUBLE OR EVEN TRIPLE OVERALL DATA STORAGE
- DATA RECOVERY CAN BE PAINFUL AND COSTLY DUE TO VARIOUS FACTORS:
 - TIME REQUIRED TO RELOAD LOST DATA.
 - BUSINESS DOWNTIME DURING RECOVERY.
 - LOSS OF DATA SINCE THE LAST BACKUP.
- FAIL-SAFE PROVIDES AN EFFICIENT AND COST-EFFECTIVE ALTERNATIVE TO BACKUP THAT ELIMINATES THE REMAINING RISK AND SCALES WITH YOUR DATA

TIMELINE

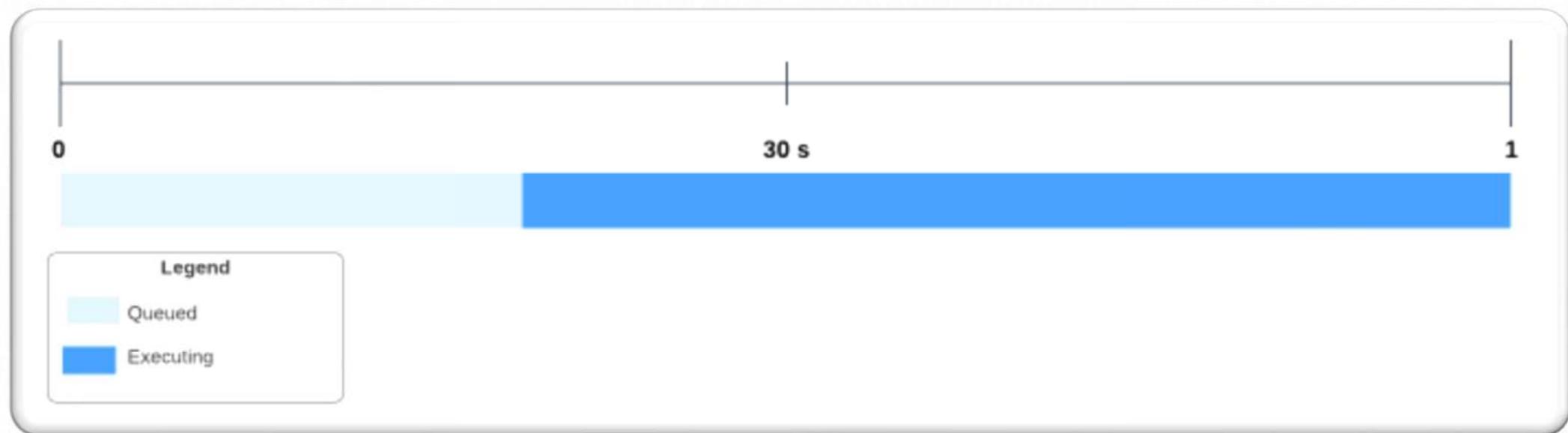


SECTION 15 – TASKS IN SNOWFLAKE

- WHAT ARE TASKS
- TASKS EXECUTION
- TREE OF TASKS

TASKS IN SNOWFLAKE

- A TASK IS A KIND OF TRIGGER WHICH GETS EXECUTED AT A SPECIFIC TIME/PERIOD.
- IT CAN EXECUTE A SINGLE SQL STATEMENT, OR CAN MAKE A CALL TO A STORED PROCEDURE
- THERE IS NO EVENT SOURCE THAT CAN TRIGGER A TASK; INSTEAD, A TASK RUNS ON A SCHEDULE, WHICH CAN BE DEFINED WHEN CREATING A TASK
- SNOWFLAKE ENSURES ONLY ONE INSTANCE OF A TASK AT A SCHEDULE TIME I.E. A STANDALONE TASK OR THE ROOT TASK IN A TREE OF TASKS

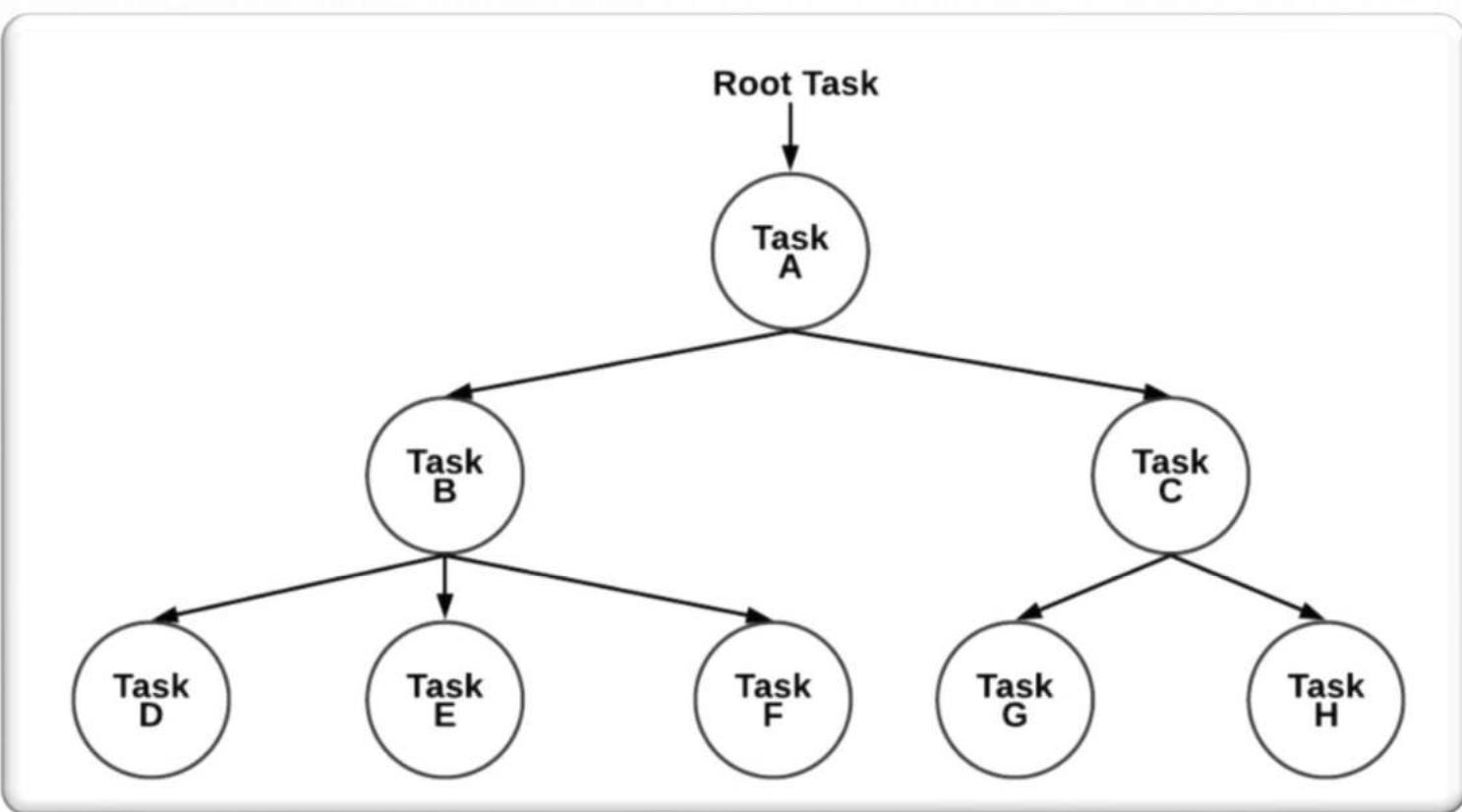


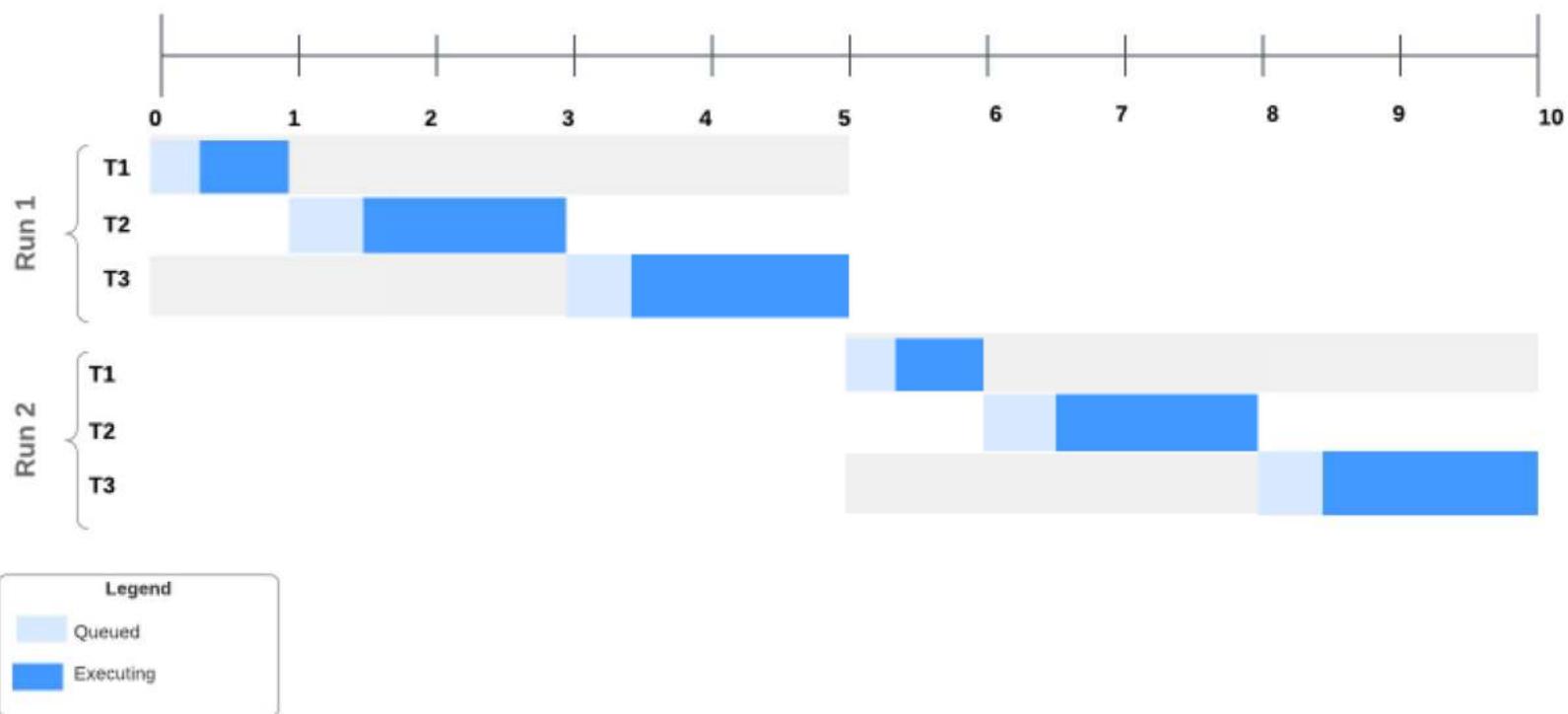
EXECUTION OF A TASK

- A WINDOW OF 1 MINUTE IN WHICH A SINGLE TASK QUEUED FOR 20 SECONDS AND THEN RAN FOR 40 SECONDS.
- THIS HAPPENED BECAUSE WAREHOUSE MUST BE BUSY EXECUTING OTHER QUERIES/TASKS
- WAREHOUSE SIZE MUST BE DETERMINED CONSIDERING THE NUMBER/VOLUME OF TASKS YOU WILL BE EXECUTING

TREE OF TASKS

- SINGLE PATH BETWEEN ANY TWO NODES CAN BE ESTABLISHED; I.E. AN INDIVIDUAL TASK CAN HAVE ONLY A SINGLE PREDECESSOR (PARENT) TASK
- THE ROOT TASK IN THE TREE SHOULD HAVE A DEFINED SCHEDULE, WHILE EACH OF THE OTHER TASKS IN THE TREE HAVE A DEFINED PREDECESSOR TO LINK THEM TOGETHER
- TREE OF TASKS IS LIMITED TO A MAXIMUM OF 1000 TASKS TOTAL (INCLUDING THE ROOT TASK) IN A RESUMED STATE
- A TASK CAN HAVE A MAXIMUM OF 100 CHILD TASKS





- IT SHOWS A TREE OF TASKS THAT REQUIRES 5 MINUTES ON AVERAGE TO COMPLETE FOR EACH RUN
- WINDOW IS CALCULATED FROM THE TIME THE ROOT TASK IS SCHEDULED TO START UNTIL THE LAST CHILD TASK IN THE TREE HAS COMPLETED RUNNING
- IF THIS TREE OF TASKS RAN ON A DEDICATED WAREHOUSE, A BRIEF LAG WOULD BE EXPECTED AFTER A PARENT TASK FINISHES RUNNING AND ANY CHILD TASK IS EXECUTED; HOWEVER, NO QUEUEING FOR SHARED RESOURCES WITH OTHER OPERATIONS WOULD OCCUR

SECTION 16 – STREAMS IN SNOWFLAKE

- INTRODUCTION TO STREAMS
- HOW STREAMS WORK UNDER THE HOOD
- CAPTURE AND PROCESS INSERT OPERATIONS
- CAPTURE AND PROCESS UPDATE OPERATIONS
- CAPTURE AND PROCESS DELETE OPERATIONS

STREAMS IN SNOWFLAKE

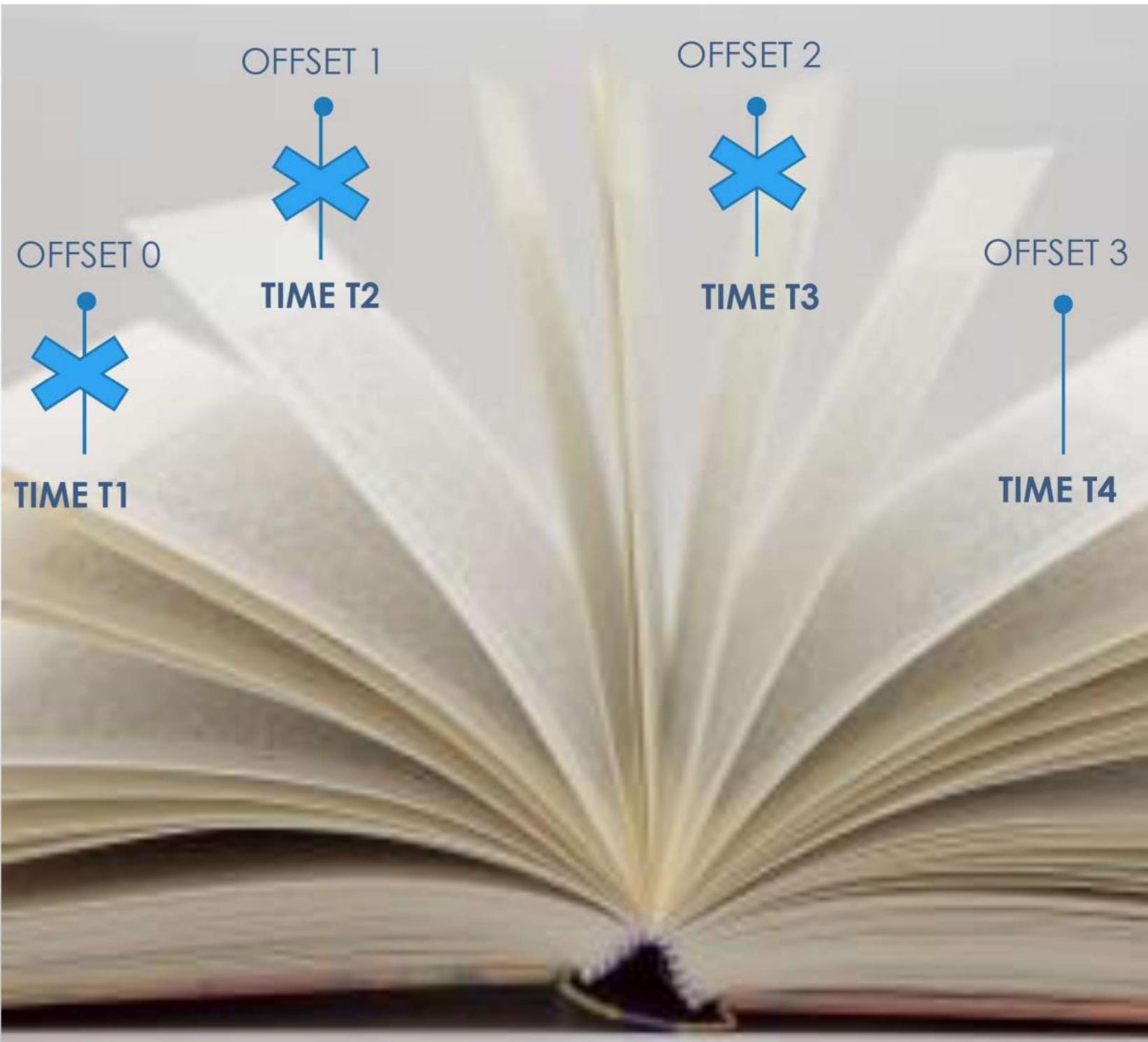
- STREAM OBJECT RECORDS DATA MANIPULATION LANGUAGE (DML) CHANGES MADE TO TABLES, INCLUDING INSERTS, UPDATES, AND DELETES, AS WELL AS METADATA ABOUT EACH CHANGE (CDC PROCESS)

- INDIVIDUAL STREAM TRACKS THE CHANGES MADE TO ROWS IN A SOURCE TABLE
- TRACKS CHANGES AT THE ROW LEVEL, BETWEEN TWO TRANSACTIONAL POINTS OF TIME IN A TABLE(FOR EXAMPLE, TIME T1 & T2)

HOW STREAMS WORK UNDER THE HOOD

- STREAM LOGICALLY TAKES AN INITIAL SNAPSHOT OF EVERY ROW IN THE SOURCE TABLE
- THEN, CHANGE TRACKING SYSTEM UTILIZED BY THE STREAM THEN RECORDS INFORMATION ABOUT THE DML CHANGES (INSERT, UPDATE, DELETE) COMMITTED AFTER THIS SNAPSHOT WAS TAKEN
- STREAM ITSELF DOES **NOT** CONTAIN ANY TABLE DATA.
- A STREAM ONLY STORES THE OFFSET FOR THE SOURCE TABLE AND RETURNS CDC RECORDS BY LEVERAGING THE VERSIONING HISTORY FOR THE SOURCE TABLE

- THINK OF A STREAM AS A BOOKMARK, WHICH INDICATES A POINT IN TIME IN THE PAGES OF A BOOK
- STREAM MAINTAINS A POINT OF TIME INTO THE TRANSACTIONAL VERSIONED TIMELINE OF THE SOURCE TABLE, CALLED AN OFFSET
- TRANSACTIONAL POINT WHEN THE STREAM CONTENTS WERE LAST CONSUMED USING A DML STATEMENT
- IF MULTIPLE DML STATEMENTS CHANGE A ROW, THE STREAM CONTAINS ONLY THE LATEST ACTION TAKEN ON THAT ROW
- STREAMS RECORD THE DIFFERENCES BETWEEN TWO OFFSETS



SECTION 17 – ZERO-COPY CLONING

- CLONE DATABASE, PIPES, STREAMS, TASKS& STAGES
- CLONE SCHEMA & TABLES
- CLONING TABLE STORAGE METRICS
- CLONING USING TIME TRAVEL
- SWAP TABLES

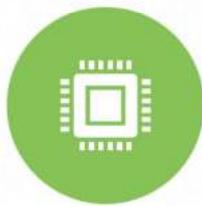
ZERO-COPY CLONING



Creates a copy of existing databases, schemas, and non-temporary tables in the system



useful for creating instant backups that do not incur any additional costs (until changes are made to the cloned object)



zero-copy cloning feature provides a convenient way to quickly take a “snapshot” of any table, schema, or database



clones can be cloned, with no limitations on the number or iterations of clones that can be created

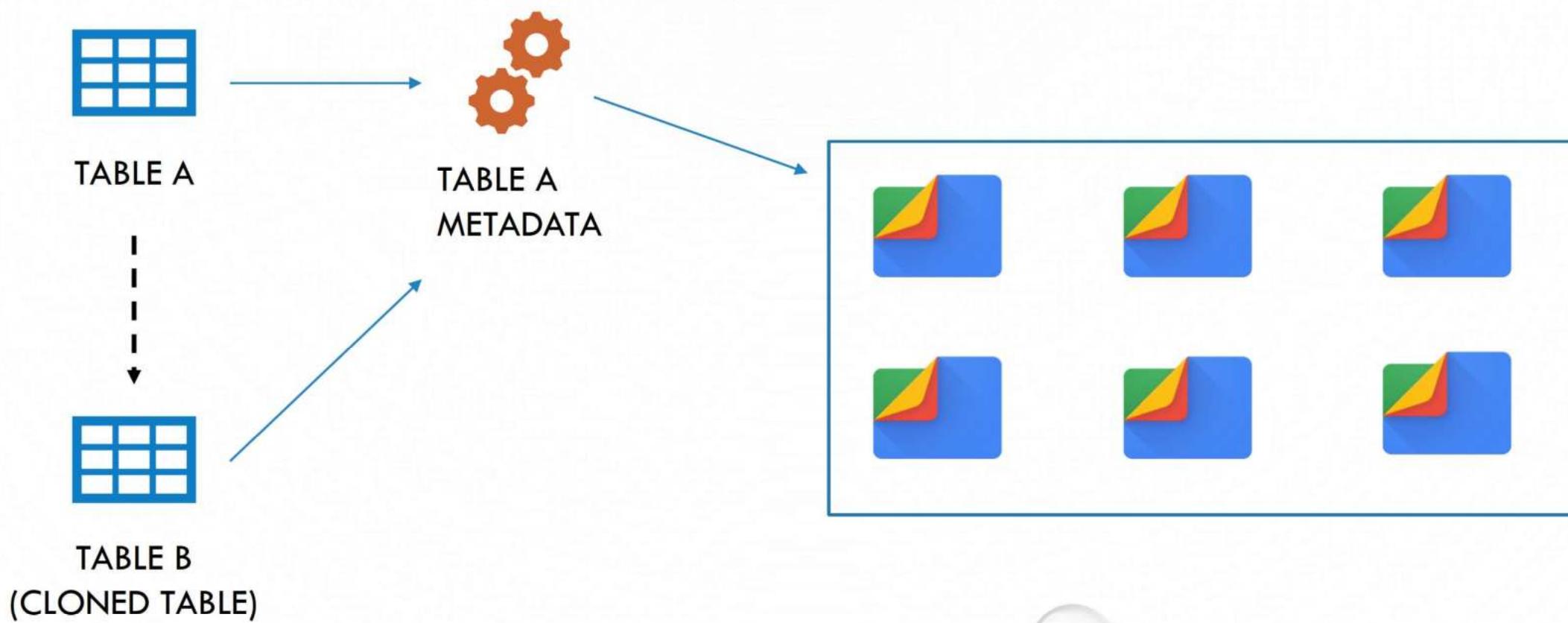


creates a derived copy of that object which initially shares the underlying storage



DDL operation on the cloned table will not affect original table

HOW CLONING WORKS UNDER THE HOOD



CLONING POSSIBLE ON LIST OF OBJECTS

- DATA STORAGE OBJECTS
 - DATABASES
 - SCHEMAS
 - TABLES (ONLY PERMANENT AND TRANSIENT)
 - STREAMS
- DATA CONFIGURATION AND TRANSFORMATION OBJECTS
 - STAGES
 - FILE FORMATS
 - SEQUENCES
 - TASKS



SWAP TABLES

TABLE A
METADATA



TABLE A

TABLE B
METADATA

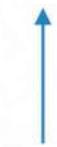


TABLE B

SWAP TABLES

TABLE A
METADATA

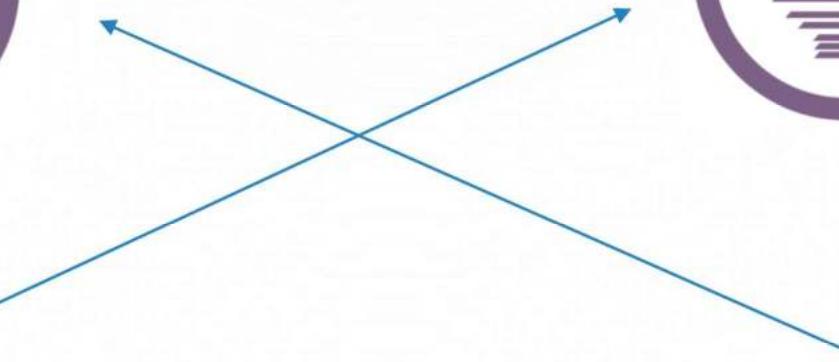


TABLE A

TABLE B
METADATA



TABLE B



SECTION 18 – SECURE DATA SHARING

- WHAT IS DATA SHARING
- HOW TO CREATE SHARES
- WHAT ARE PROVIDER AND CONSUMER ACCOUNTS
- SHARE DATA WITH CONSUMER ACCOUNTS
- CONSUME DATA ON CONSUMER ACCOUNTS
- LIMITATIONS OF SHARED DATABASES
- REMOVE OR ADD ACCOUNTS FROM EXISTING SHARES

DATA SHARING DEFINITION



- THE ABILITY TO SHARE THE SAME DATA RESOURCE WITH MULTIPLE APPLICATIONS OR USERS
- DATA SHARING IS THE WAY TO OPTIMIZE HIGHER-RELEVANT DATA, GENERATING MORE ROBUST DATA AND ANALYTICS TO SOLVE BUSINESS CHALLENGES AND MEET ENTERPRISE GOALS

TYPES OF ACCOUNTS WHEN SHARING DATA ON SNOWFLAKE

SNOWFLAKE ENABLES THE SHARING OF DATABASES THROUGH SHARES

PROVIDER ACCOUNTS

ANY SNOWFLAKE ACCOUNT WHICH PRODUCE OR CREATE THE DATA AND SHARES WITH OTHER SNOWFLAKE ACCOUNTS (USING **OUTBOUND SHARES**)

NO HARD LIMITS ON THE NUMBER OF SHARES YOU CAN CREATE OR THE NUMBER OF ACCOUNTS YOU CAN ADD

CONSUMER ACCOUNTS

ANY SNOWFLAKE ACCOUNT WHICH CONSUMES THE DATA (USING **INBOUND SHARES**)

NO HARD LIMITS ON THE NUMBER OF SHARES YOU CAN CONSUME FROM DATA PROVIDERS

INBOUND SHARES ARE CREATED AUTOMATICALLY AS SOON AS PROVIDERS CREATE OUTBOUND SHARES AND ADD THE CONSUMER ACCOUNT ID

LIST OF OBJECTS ALLOWED TO SHARE ON SNOWFLAKE

- TABLES
- EXTERNAL TABLES
- SECURE VIEWS
- SECURE MATERIALIZED VIEWS
- SECURE UDFS

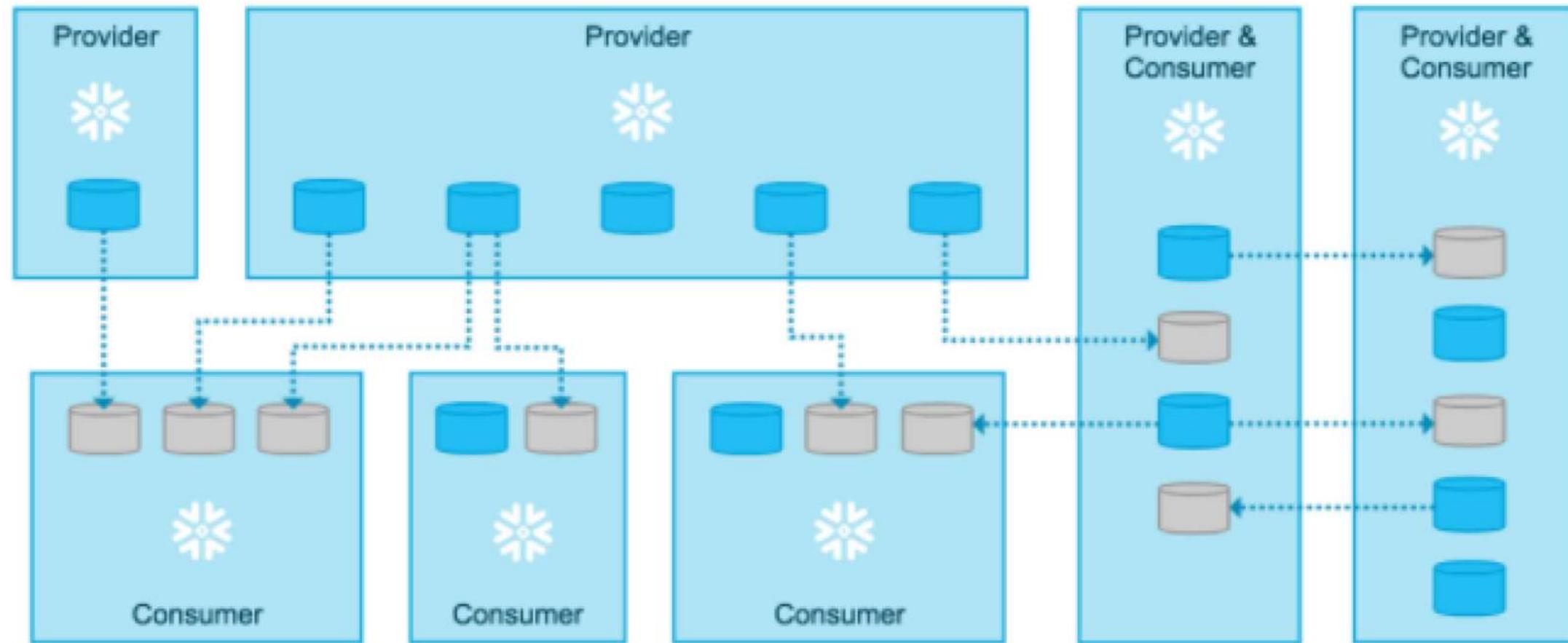
ALL DATABASE OBJECTS SHARED BETWEEN ACCOUNTS ARE READ-ONLY (I.E. THE OBJECTS CANNOT BE MODIFIED OR DELETED, INCLUDING ADDING OR MODIFYING TABLE DATA)



HOW SECURE DATA SHARING WORK?

- NO ACTUAL DATA IS COPIED OR TRANSFERRED BETWEEN ACCOUNTS
- SHARING IS DONE USING CLOUD SERVICES LAYER AND METADATA STORE
- DOES NOT TAKE UP ANY STORAGE IN A CONSUMER ACCOUNT
- CONSUMERS PAYS FOR ONLY COMPUTE RESOURCES

DATA SHARING ARCHITECTURE



- DATABASE
- SHARED DATABASE

SQL VIEWS

VIEWS IN SQL ARE KIND OF VIRTUAL TABLES.

A VIEW ALSO HAS ROWS AND COLUMNS AS THEY ARE IN A REAL TABLE IN THE DATABASE.

SPECIAL TYPES OF VIEWS ON SNOWFLAKE

- MATERIALIZED VIEWS
- SECURE VIEWS

SECTION 19 – MATERIALIZED VIEWS

- INTRODUCTION
- WHEN TO CREATE MV
- ADVANTAGES
- LIMITATIONS
- HANDS-ON
- IMPORTANT METADATA COLUMNS
- REFRESH HISTORY
- MV CLUSTERING AND RECLUSTERING

MATERIALIZED VIEWS | INTRODUCTION

- PRE-COMPUTED DATA SET DERIVED FROM A QUERY SPECIFICATION
- QUERYING A MATERIALIZED VIEW IS FASTER THAN EXECUTING A QUERY AGAINST THE BASE TABLE OF THE VIEW
- DESIGNED TO IMPROVE QUERY PERFORMANCE FOR WORKLOADS COMPOSED OF COMMON, REPEATED QUERY PATTERNS
- WORKS ON ENTERPRISE EDITION OR HIGHER
- MAINTENANCE OF MATERIALIZED VIEWS INCURS ADDITIONAL COSTS

MATERIALIZED VIEWS | WHEN TO CREATE

- QUERY RESULTS CONTAIN A SMALL NUMBER OF ROWS AND/OR COLUMNS
- RESULTS CONTAIN RESULTS THAT REQUIRE SIGNIFICANT PROCESSING
- VIEW'S BASE TABLE DOES NOT CHANGE FREQUENTLY
- THE RESULTS OF THE VIEW ARE USED OFTEN

MATERIALIZED VIEWS | ADVANTAGES

- AUTOMATICALLY AND TRANSPARENTLY MAINTAINED BY SNOWFLAKE
- DATA ACCESSED THROUGH MATERIALIZED VIEWS IS ALWAYS CURRENT
- PROVIDES QUERY RESULTS FASTER THAN QUERY ON BASE TABLES

MATERIALIZED VIEWS | LIMITATIONS

- CAN QUERY ONLY A SINGLE TABLE
- JOINS, INCLUDING SELF-JOINS, ARE NOT SUPPORTED
- MATERIALIZED VIEW CANNOT QUERY:
 - A MATERIALIZED VIEW.
 - A NON-MATERIALIZED VIEW.
 - A UDTF (USER-DEFINED TABLE FUNCTION)
- MANY AGGREGATE FUNCTIONS ARE NOT ALLOWED IN A MATERIALIZED VIEW DEFINITION
- DOES NOT SUPPORT HAVING, ORDER BY, LIMIT, WINDOW FUNCTIONS, UDF, ETC.

SECTION 20 – SECURE VIEWS

- INTRODUCTION
- HANDS-ON

SECURE VIEWS | INTRODUCTION

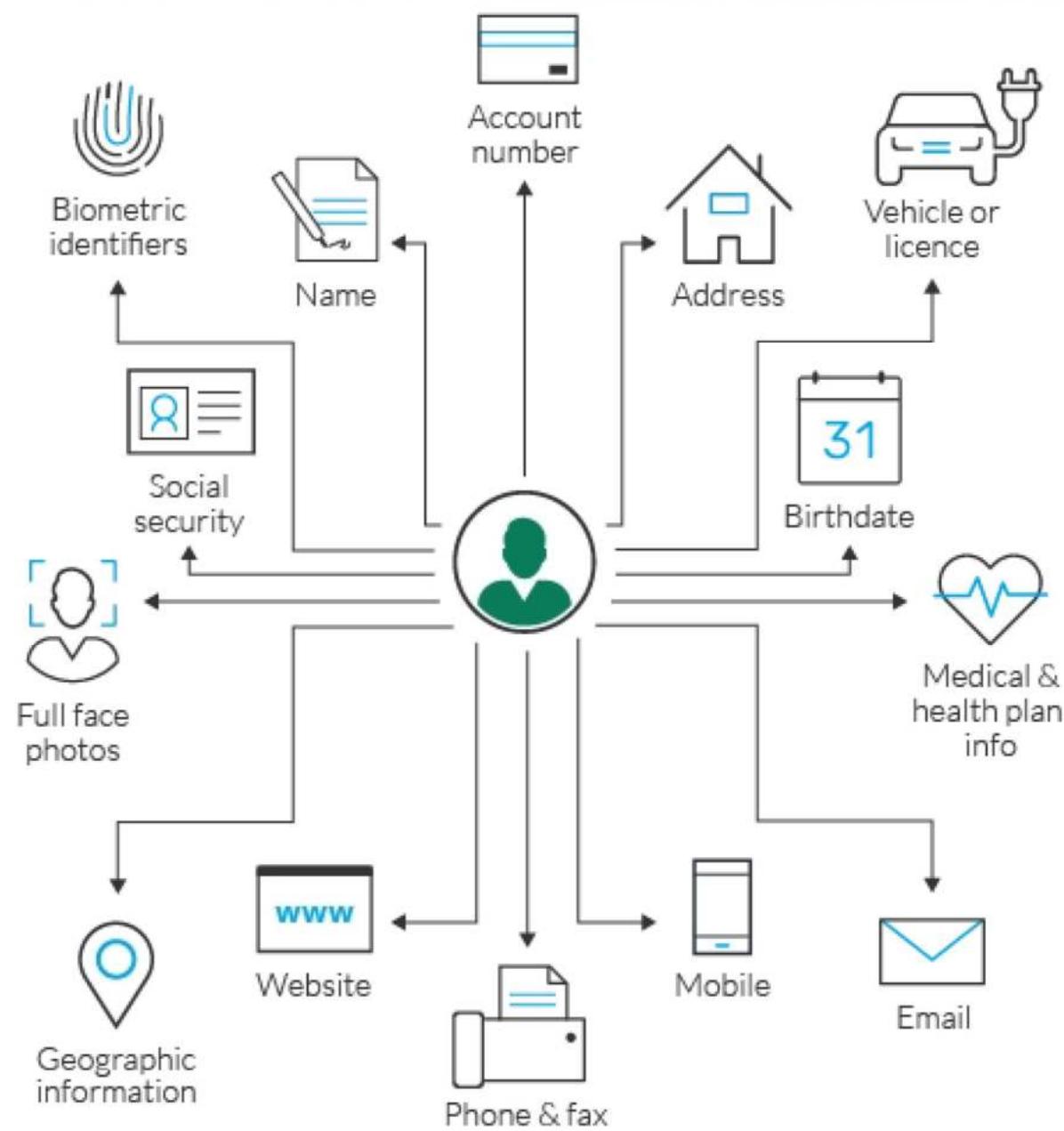
- SECURE VIEWS ARE DEFINED USING THE SECURE KEYWORD WITH THE STANDARD DDL FOR VIEWS
- UNAUTHORIZED USERS WILL NOT BE ABLE TO SEE THE VIEW DEFINITION USING GET_DDL OR DESC COMMANDS
- BOTH REGULAR AND MATERIALIZED VIEWS CAN BECOME SECURE VIEWS
- SHOULD BE CREATED WHEN CONSIDERING SECURITY OR DATA PRIVACY ON THE UNDERLYING TABLE USED IN VIEW DEFINITION
- CAN BE USED TO APPLY ROW-LEVEL SECURITY
- WHEN EVALUATING SECURE VIEWS, SNOWFLAKE BYPASSES CERTAIN OPTIMIZATIONS WHICH ARE USED FOR REGULAR VIEWS

SECTION 22 – DYNAMIC DATA MASKING

- INTRODUCTION
- DETAILED OVERVIEW
- HANDS-ON
- RECREATE AND UNSET MASKING POLICY
- APPLY MASKING POLICY ON VIEWS
- ALTER MASKING POLICY

DATA MASKING

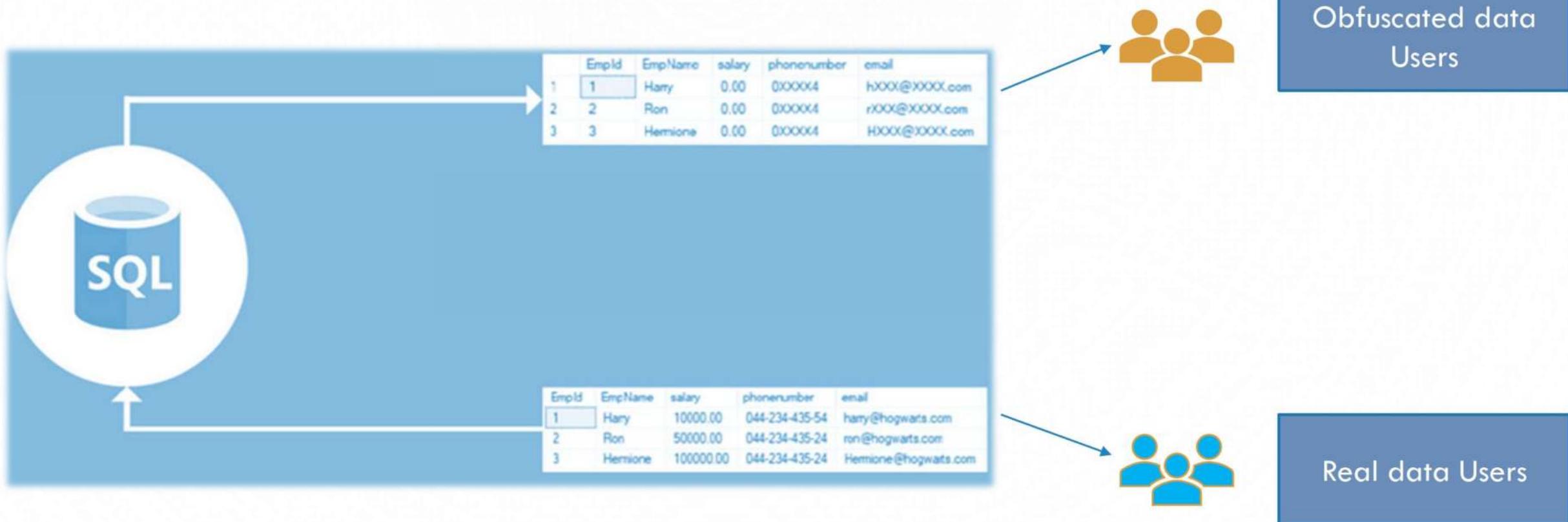
- PROCESS OF HIDING ORIGINAL DATA WITH MODIFIED CONTENT
- TO PROTECT DATA THAT IS CLASSIFIED AS PERSONALLY IDENTIFIABLE INFORMATION, SENSITIVE PERSONAL DATA, OR COMMERCIALLY SENSITIVE DATA
- ALSO KNOWN AS DATA OBFUSCATION
- USED FOR COLUMN-LEVEL SECURITY



PII DATA

PERSONAL IDENTIFIABLE INFORMATION

DATA MASKING EXAMPLE



DATA MASKING REQUIREMENTS

- WORKS ON ENTERPRISE EDITION (OR HIGHER)
- DATABASE AND SCHEMA MUST EXIST

MORE ON DATA MASKING

- SNOWFLAKE SUPPORTS USING DYNAMIC DATA MASKING ON TABLES AND VIEWS.
- MASKING POLICIES ARE APPLIED ON INDIVIDUAL COLUMNS
- AUTOMATICALLY APPLIED AT EVERY LOCATION WHERE THE COLUMN APPEARS
- WORKS ON SQL QUERY EXECUTION

LIMITATIONS OF DATA MASKING

- PRIOR TO DROPPING A POLICY, UNSET IS REQUIRED
- ONLY ONE COLUMN PER ALTER TABLE OR ALTER VIEW STATEMENT.
- MASKING POLICY DEFINITION MUST HAVE THE SAME DATA TYPE FOR THE INPUT AND OUTPUT.

SECTION 23 – SNOWFLAKE PARTNER CONNECT

- INTRODUCTION
- KNOW ABOUT AVAILABLE TOOLS/PARTNERS
- START FREE TRIAL

SNOWFLAKE PARTNER CONNECT

SNOWFLAKE PARTNER CONNECT ALLOWS ITS USERS TO ACCESS PARTNERS' PLATFORMS

USERS CAN EASILY CREATE TRIAL ACCOUNT WITH ANY OF THE PARTNERS

INTEGRATE PARTNER TOOLS WITH SNOWFLAKE

CONVENIENT OPTION FOR TRYING ADDITIONAL TOOLS AND SERVICES

ADOPT THE ONES THAT BEST MEET YOUR BUSINESS NEEDS

PARTNER CONNECT USE CASES

- ETL - MOVE DATA IN OR OUT OF SNOWFLAKE
- ETL - PERFORM TRANSFORMATIONS
- BUSINESS INTELLIGENCE - INTERACTIVE DASHBOARDS & REPORTS
- CI/CD & DATAOPS - BUILD PIPELINES IN AGILE WAY
- DATA SECURITY & PRIVACY
- AI-DRIVEN ANALYTICS

AND MANY MORE...

