Snowflake cloud data-warehouse

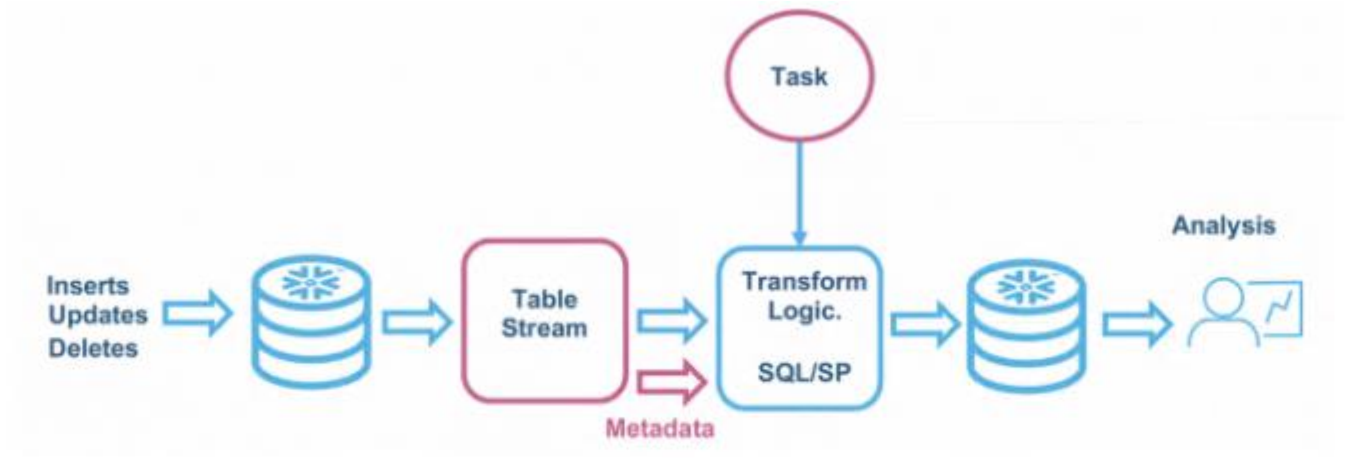
“Simplicity is the ultimate sophistication”

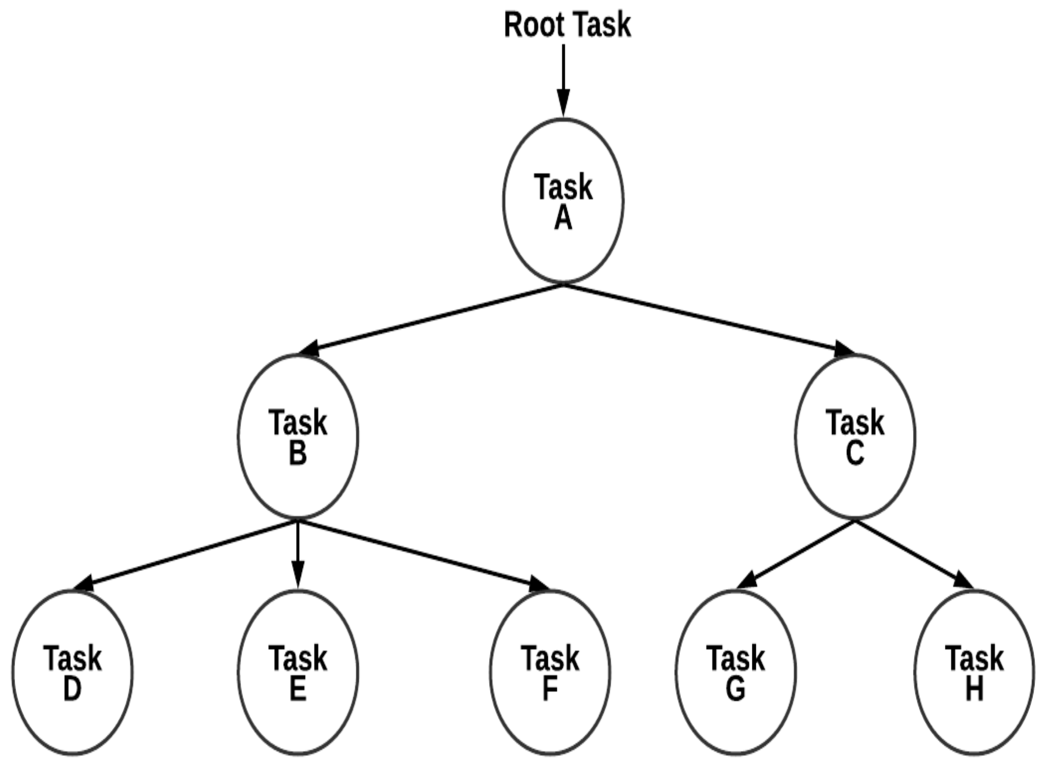
Leonardo da Vinci

Snowflake Cloud Data warehouse

Day-7

Snowflake TASK





- ❑ Executes DML Statement Or Stored Procedure
  - ✓ Runs on Schedule
  - ✓ Runs after Predecessor task completes
- ❑ Tasks can be used on stream having data

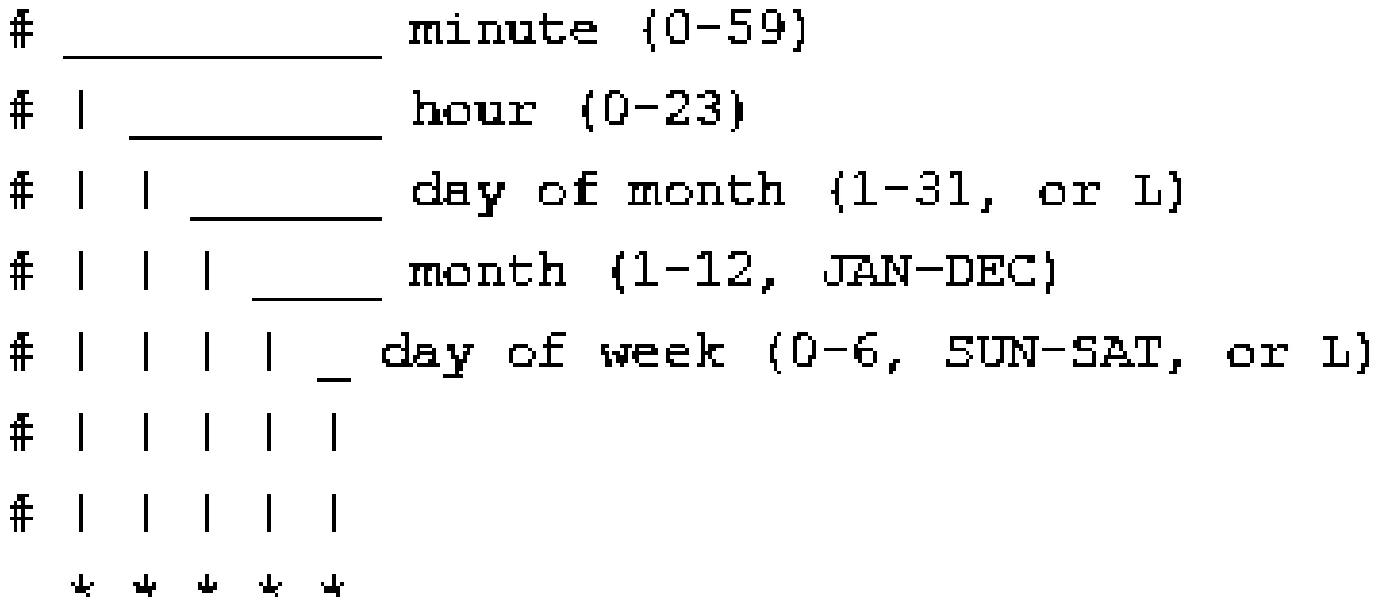
Snowflake TASK

```
 CREATE [ OR REPLACE ] TASK [ IF NOT EXISTS ] <name>
  WAREHOUSE = <string>
  [ SCHEDULE = '{ <num> MINUTE | USING CRON <expr> <time_zone> }' ]
  [ <session_parameter> = <value> [ , <session_parameter> = <value> ... ] ]
  [ USER_TASK_TIMEOUT_MS = <num> ]
  [ COMMENT = '<string_literal>' ]
  [ AFTER <string> ]
[ WHEN <boolean_expr> ]
AS
  <sql>
```

Snowflake TASK

USING CRON **<expr>** <time_zone> }'

0 15 * * 1-5	At 15:00 on every day from Monday through Friday
00 08-16 * * *	At every day, every hour, on the hour, from 8 AM through 4 PM
0 7 1-7 * 1	At first Monday of each month, at 7 a.m



```
# ___________ minute (0-59)
# | _________ hour (0-23)
# | | _______ day of month (1-31, or L)
# | | | _____ month (1-12, JAN-DEC)
# | | | | _ day of week (0-6, SUN-SAT, or L)
# | | | | |
# | | | | |
  * * * * *
```

Snowflake TASK

Points to consider while using TASK

- ❑ The SQL statement must be executable on its own
- ❑ The task owner must also have the global EXECUTE TASK
- ❑ Task can have a maximum of 100 child tasks
- ❑ Tree of tasks is limited to a maximum of 1000 tasks total
- ❑ All tasks in Tree must have same owner role
- ❑ Must execute ALTER TASK … RESUME before the task will run
- ❑ AUTOCOMMIT parameter must be set to TRUE
- ❑ Using CREATE OR REPLACE on is dropped and recreated using the specified definition
- ❑ SYSTEM$USER_TASK_CANCEL_ONGOING_EXECUTIONS('<task_name>')

Snowflake TASK

TASK Demo

Challenges in Data Copy

Many common tasks require copies

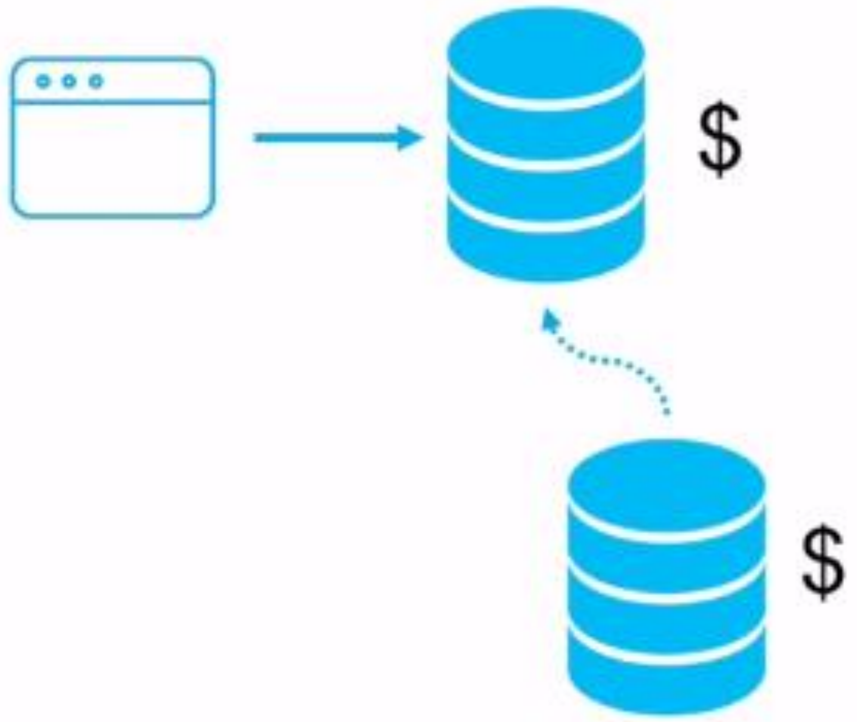
- Test/Development/QA environments

On going maintenance challenge

- Data continues to change in production
- Need to copy to update
- Promoting tests to production is challenging

Cost

- Need to pay for storing multiple copies of data



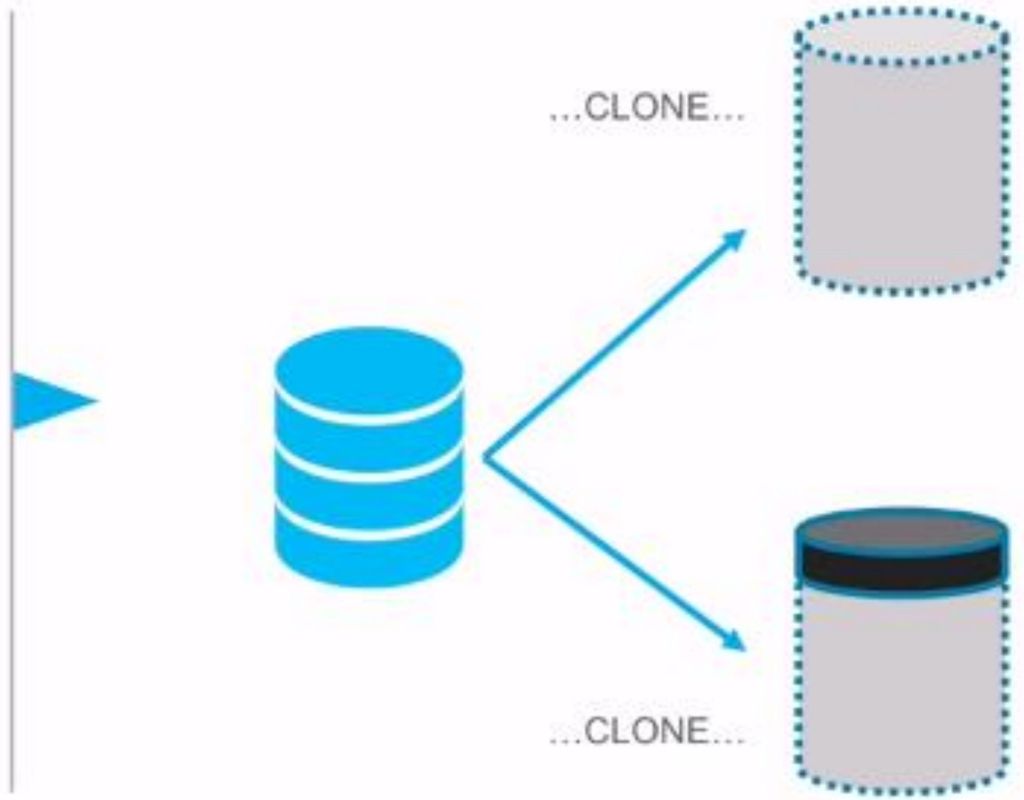
Snowflake Cloning

Fast data cloning operations

- Databases and tables
- Metadata-only operation
- No data copying required

Modified data stored as new blocks

- Unmodified data stored only once



Snowflake Cloning

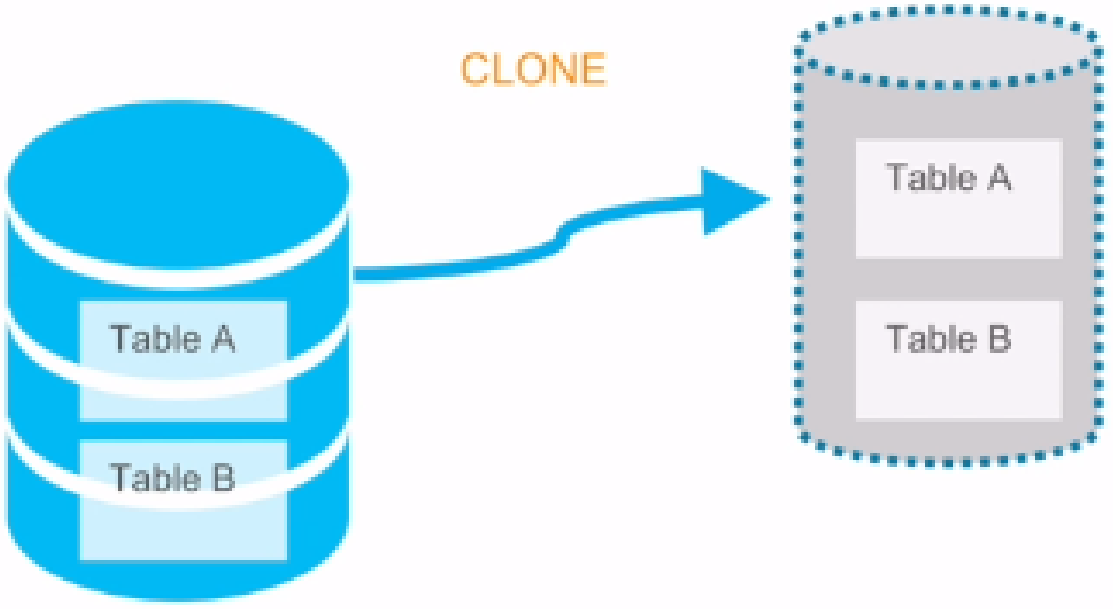
SNAPSHOT in point of time

Preserve a frozen copy of the data at critical milestones

Useful for audits and compliance

Useful as a basic backup

Cost Saving



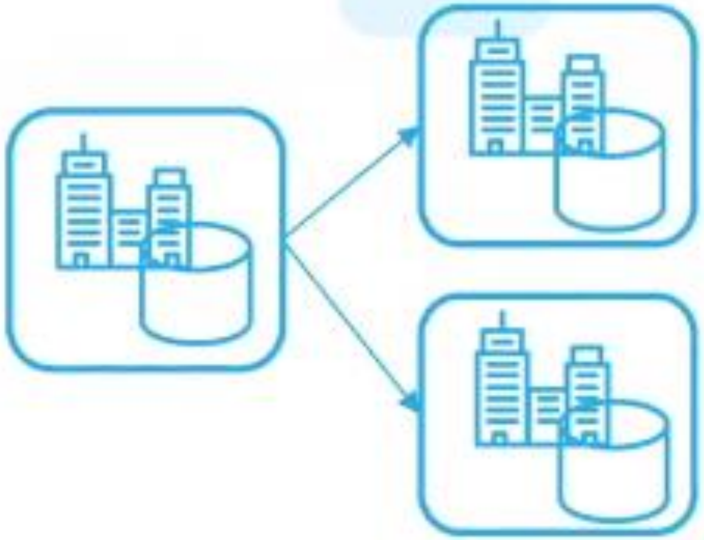
Snowflake Cloning

Points to consider while using CLONING

- ❑ **Cloning operations is not instantaneous**
- ❑ **Cloning operations are set of independent DDL statement**
- ❑ **Cloning table sets AT (Time travel) timestamp clause by default**
- ❑ **Individual external named stages can be cloned NOT internal named stages**
- ❑ **Tables are cloned so the internal Table stages are also cloned**
- ❑ **Pipes that reference an external stage are cloned**
- ❑ **Tasks are cloned but with suspended state**

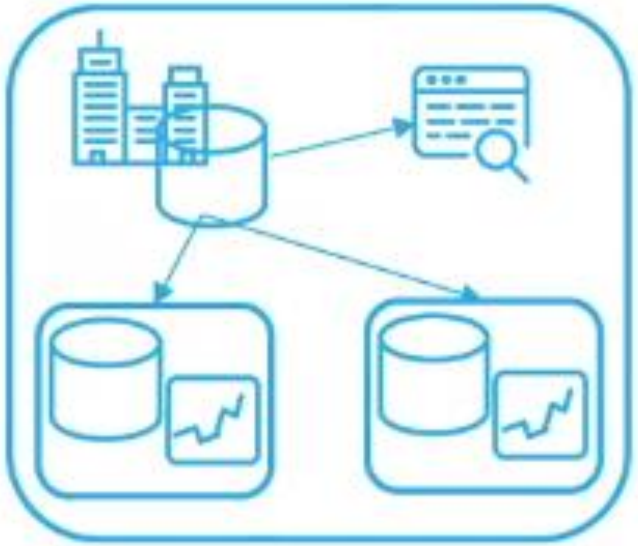
Snowflake Sharing

Enterprise-to-Enterprise



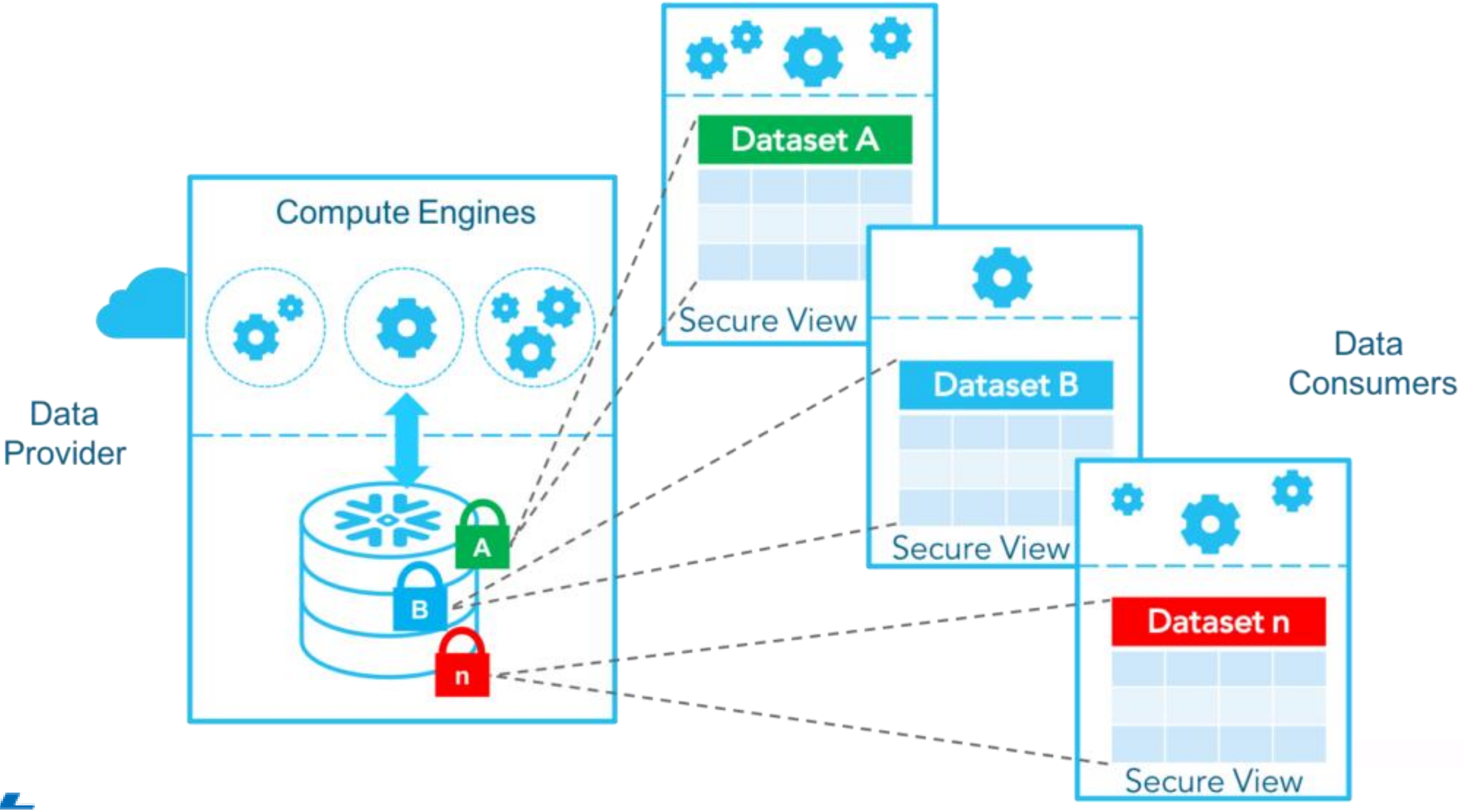
Outside the enterprise,
organizations share data to help

Intra-Enterprise

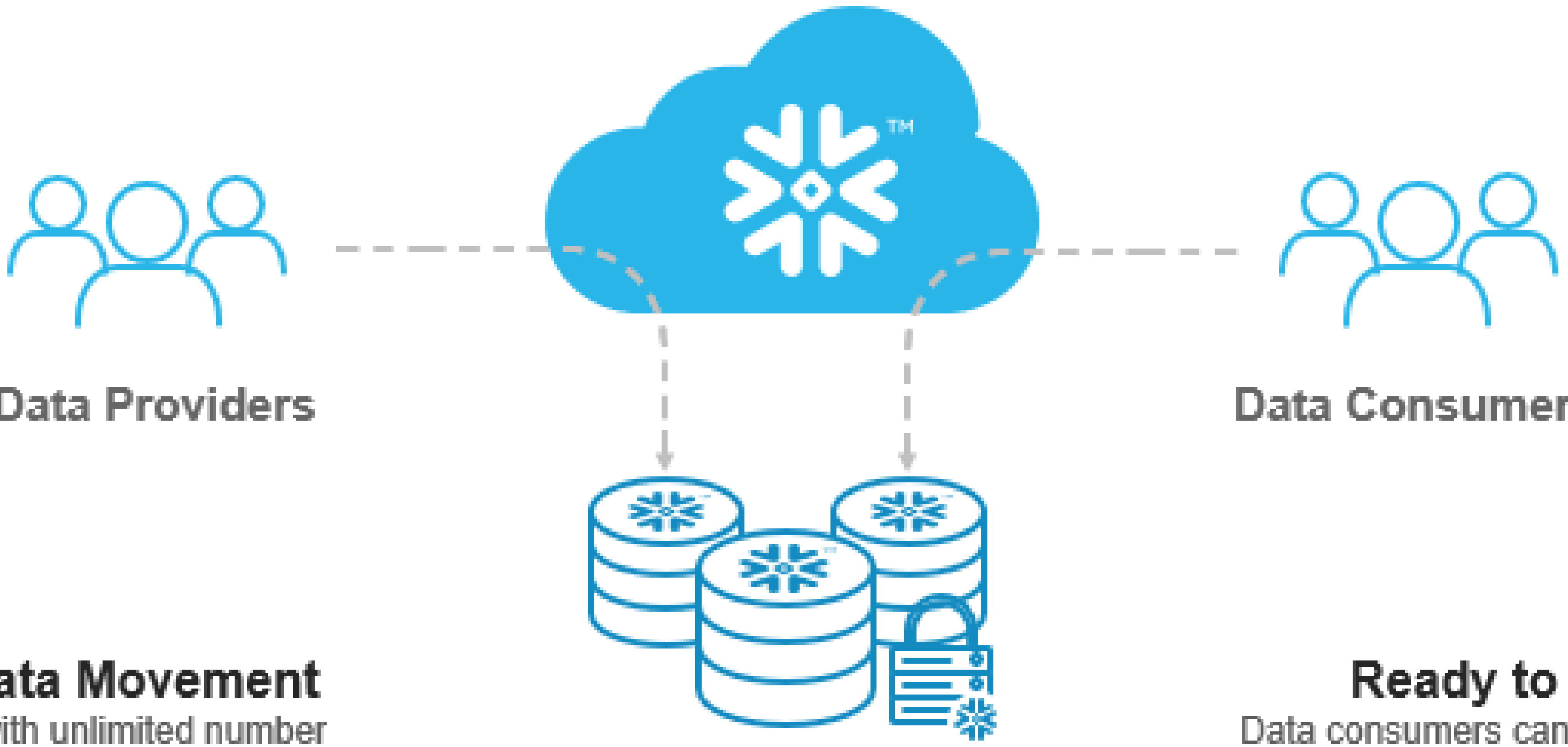


Organizations inside an
enterprise acquire crucial insight

Snowflake Sharing



Snowflake Sharing

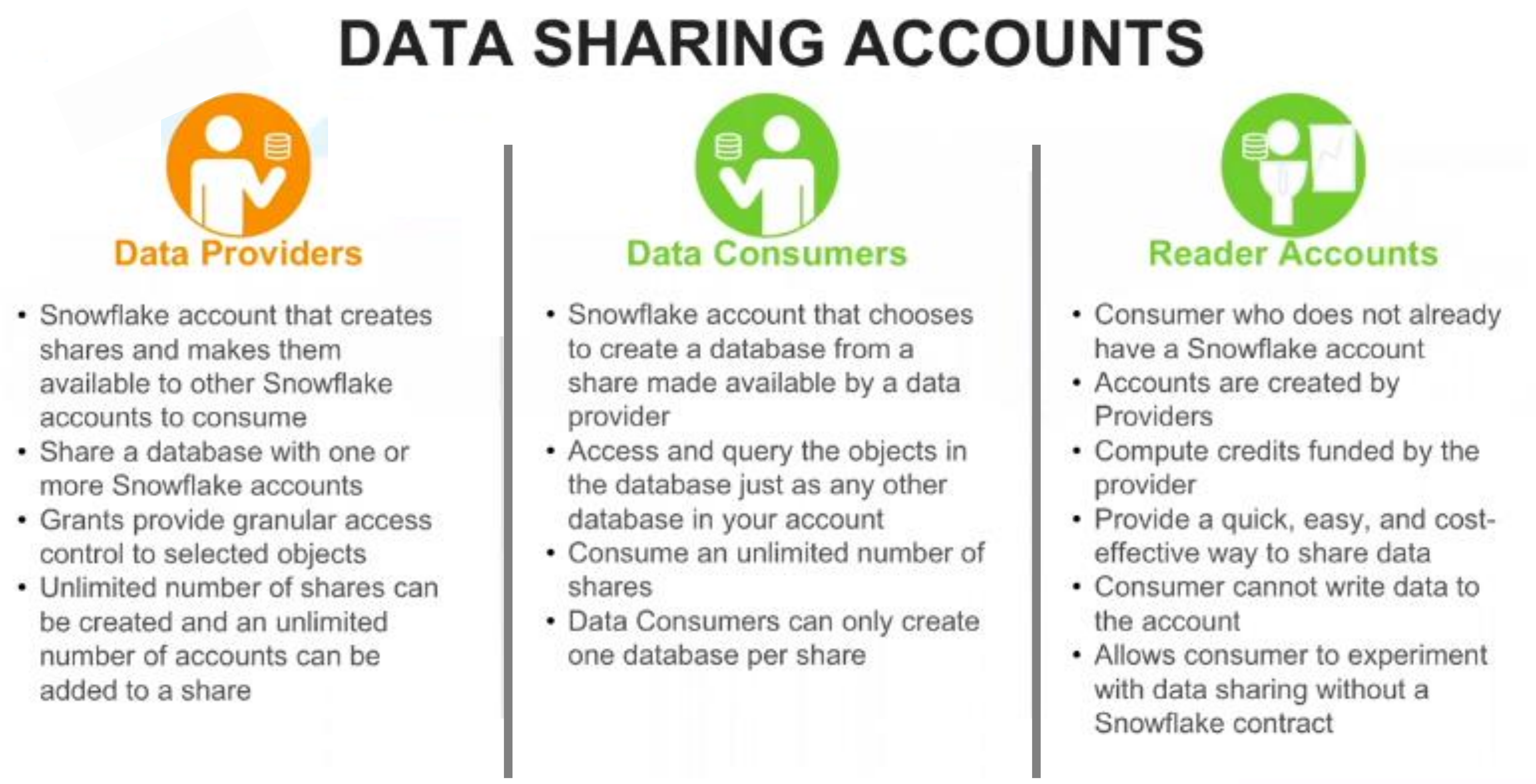


No Data Movement
Share with unlimited number of consumers without having to copy or move data

Live and Secure Access
You control who, how, and what data can be accessed, protecting PII and enforcing security policies

Ready to Use
Data consumers can immediately start querying and see all the latest updates

Snowflake Sharing



DATA SHARING ACCOUNTS

Data Providers

- Snowflake account that creates shares and makes them available to other Snowflake accounts to consume
- Share a database with one or more Snowflake accounts
- Grants provide granular access control to selected objects
- Unlimited number of shares can be created and an unlimited number of accounts can be added to a share

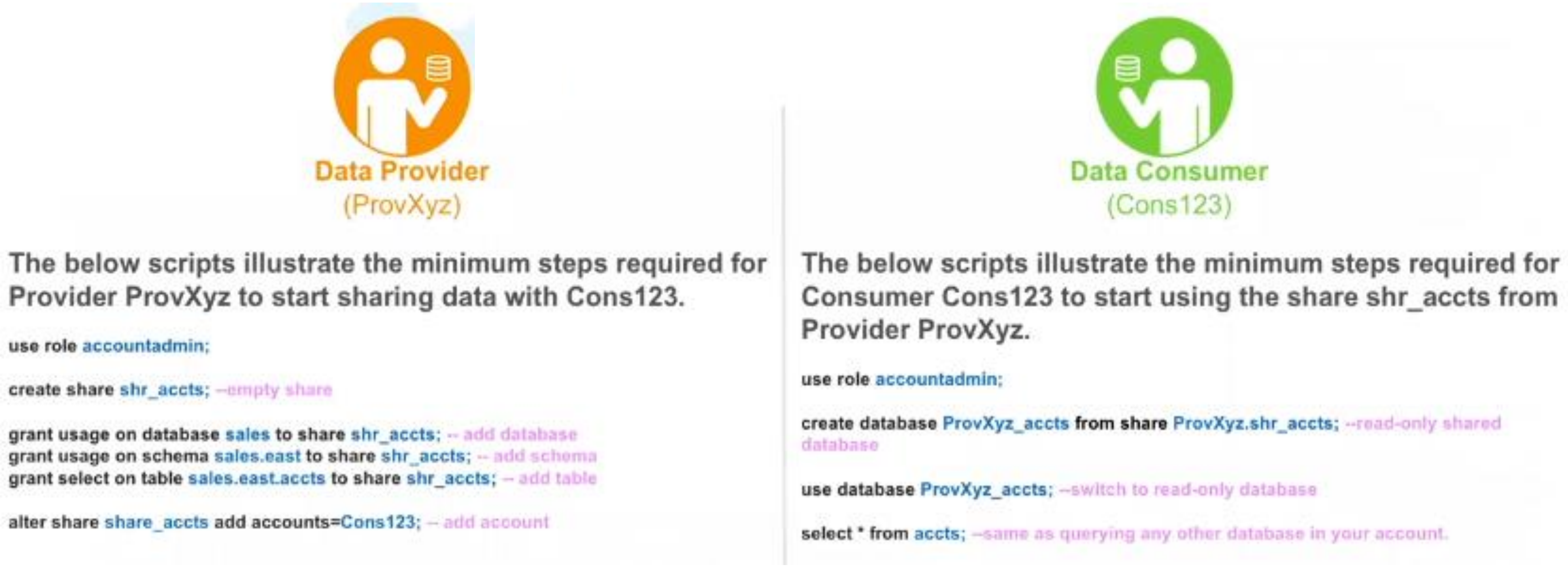
Data Consumers

- Snowflake account that chooses to create a database from a share made available by a data provider
- Access and query the objects in the database just as any other database in your account
- Consume an unlimited number of shares
- Data Consumers can only create one database per share

Reader Accounts

- Consumer who does not already have a Snowflake account
- Accounts are created by Providers
- Compute credits funded by the provider
- Provide a quick, easy, and cost-effective way to share data
- Consumer cannot write data to the account
- Allows consumer to experiment with data sharing without a Snowflake contract

Snowflake Sharing



Data Provider
(ProvXyz)

The below scripts illustrate the minimum steps required for Provider ProvXyz to start sharing data with Cons123.

```
use role accountadmin;

create share shr_accts; --empty share

grant usage on database sales to share shr_accts; -- add database
grant usage on schema sales.east to share shr_accts; -- add schema
grant select on table sales.east.accts to share shr_accts; -- add table

alter share share_accts add accounts=Cons123; -- add account
```

Data Consumer
(Cons123)

The below scripts illustrate the minimum steps required for Consumer Cons123 to start using the share shr_accts from Provider ProvXyz.

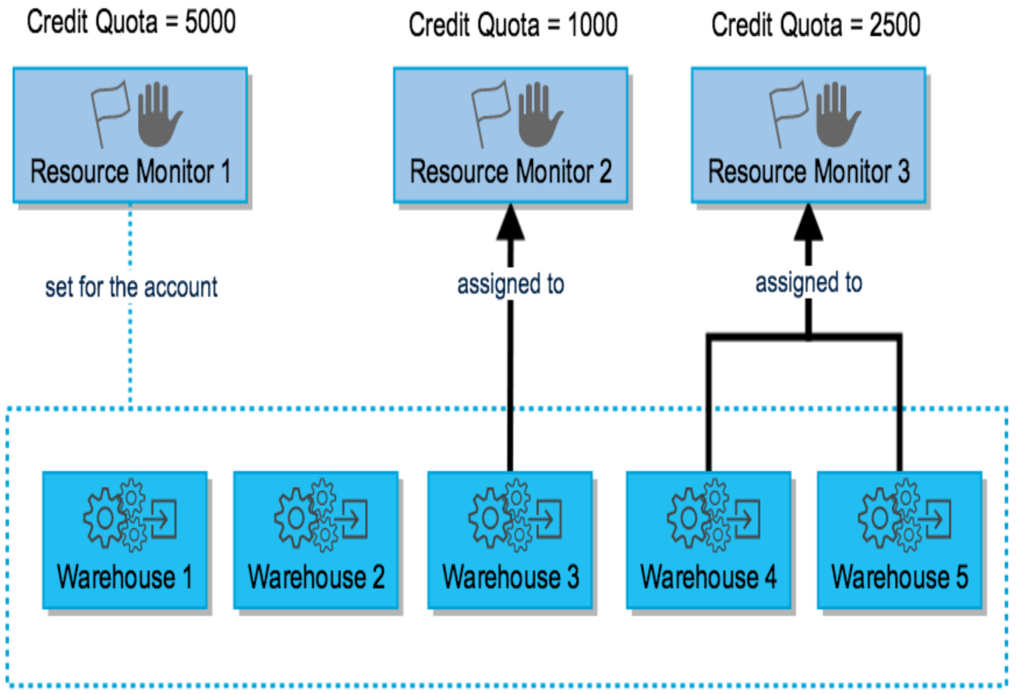
```
use role accountadmin;

create database ProvXyz_accts from share ProvXyz.shr_accts; --read-only shared
database

use database ProvXyz_accts; --switch to read-only database

select * from accts; --same as querying any other database in your account.
```

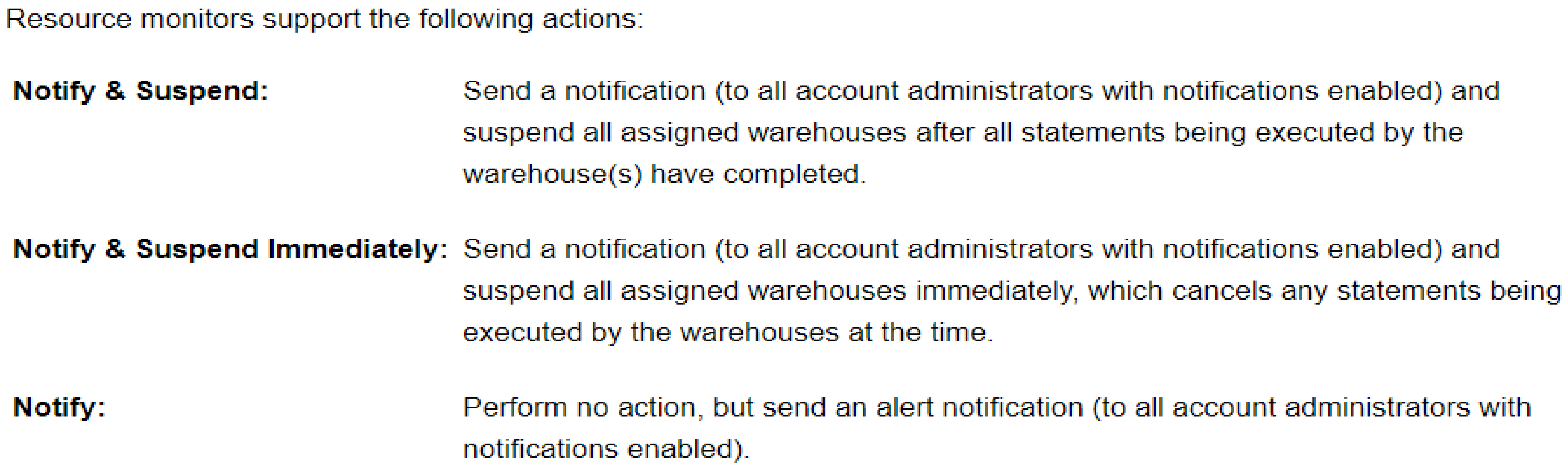
Resource Monitor



Resource monitors can be used to impose limits on the number of credits that are consumed by:

•User-managed virtual warehouses
•Virtual warehouses used by cloud services

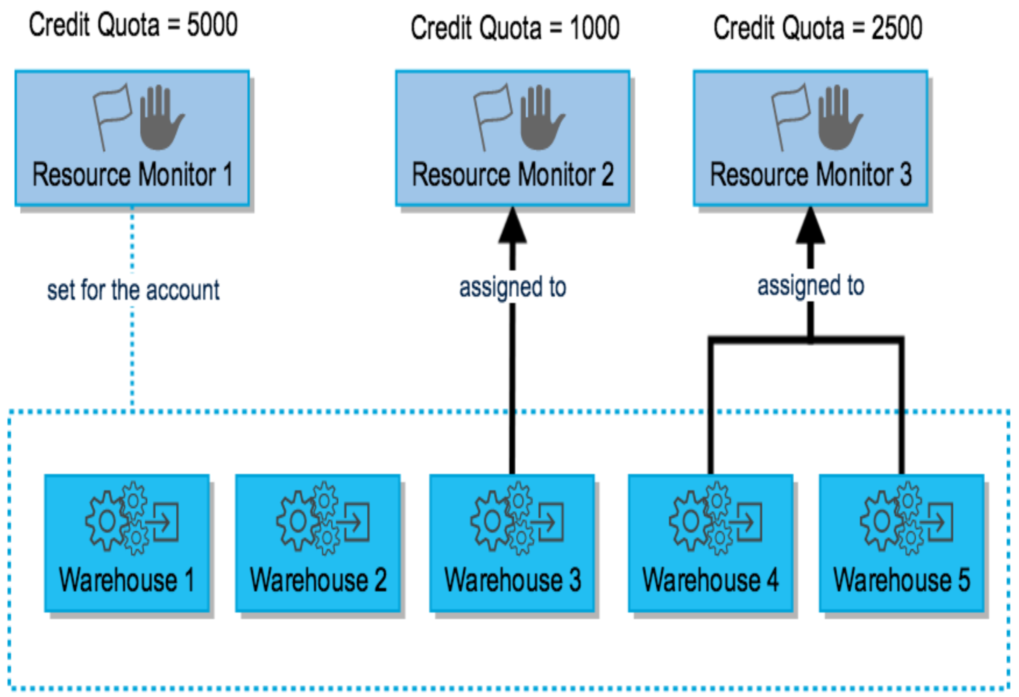
Resource Monitor



Resource monitors support the following actions:

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).

Resource Monitor

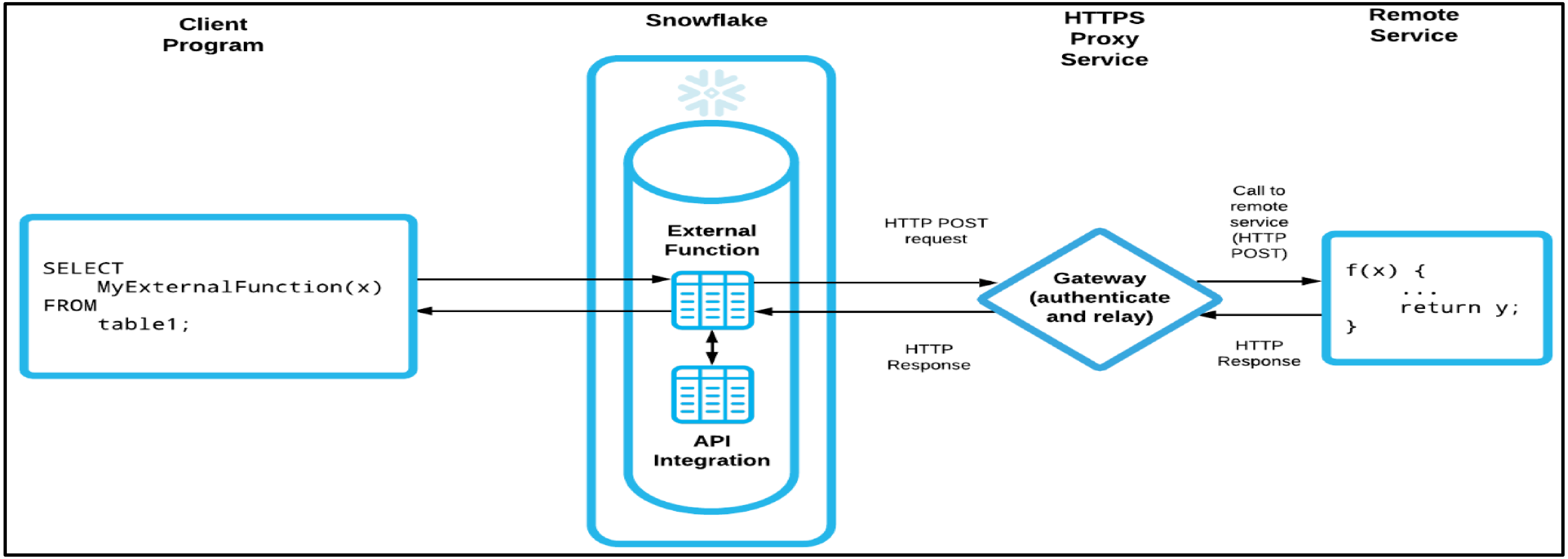


Resource monitors can be used to impose limits on the number of credits that are consumed by:

•User-managed virtual warehouses
•Virtual warehouses used by cloud services

External Function

- An external functions calls an executable code that is developed, maintained, stored, and executed outside Snowflake.
- The remotely executed code is known as a remote service.
- Information sent to a remote service is usually relayed through a proxy service.
- Snowflake stores security-related external function information in an API integration.



External Function

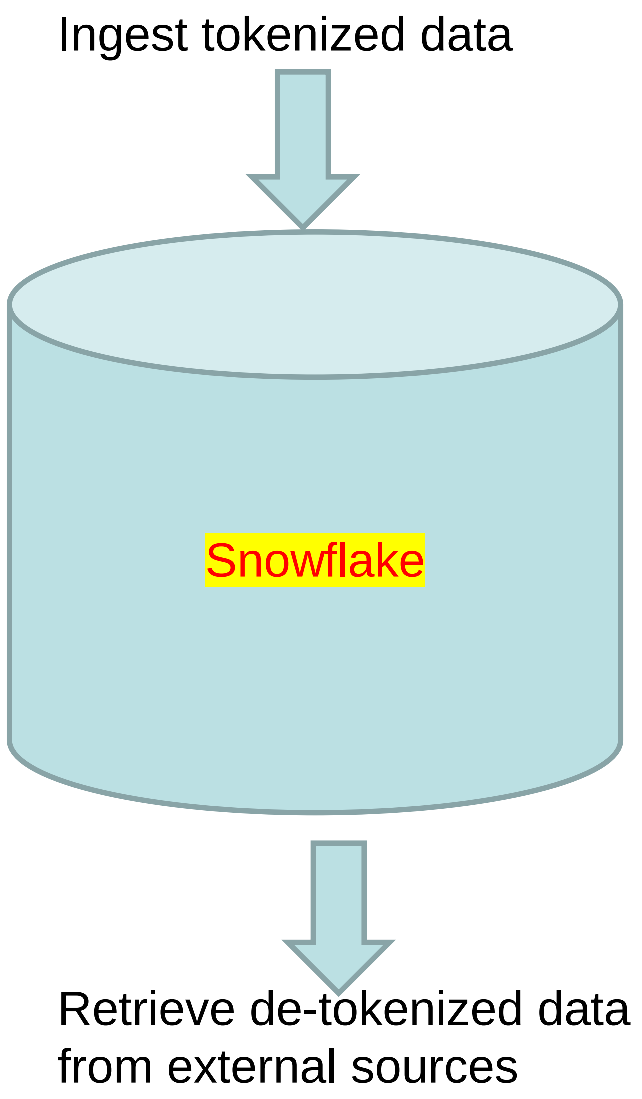
- CREATE [OR REPLACE] [SECURE] EXTERNAL FUNCTION <name> ([<arg_name> <arg_data_type>] [,..])
- RETURNS <result_data_type>
- [[NOT] NULL]
- [{ CALLED ON NULL INPUT | { RETURNS NULL ON NULL INPUT | STRICT } }]
- [VOLATILE | IMMUTABLE]
- [COMMENT = '<string_literal>']
- API_INTEGRATION = <api_integration_name>
- [HEADERS = ('<header_1>' = '<value_1>' [, '<header_2>' = '<value_2>' ...])]
- [CONTEXT_HEADERS = (<context_function_1> [, context_function_2> ...])]
- [MAX_BATCH_ROWS = <integer>]
- [COMPRESSION = <compression_type>]
- [REQUEST_TRANSLATOR = <request_translator_udf_name>]
- [RESPONSE_TRANSLATOR = <response_translator_udf_name>]
- AS <url_of_proxy_and_resource>;

Example:

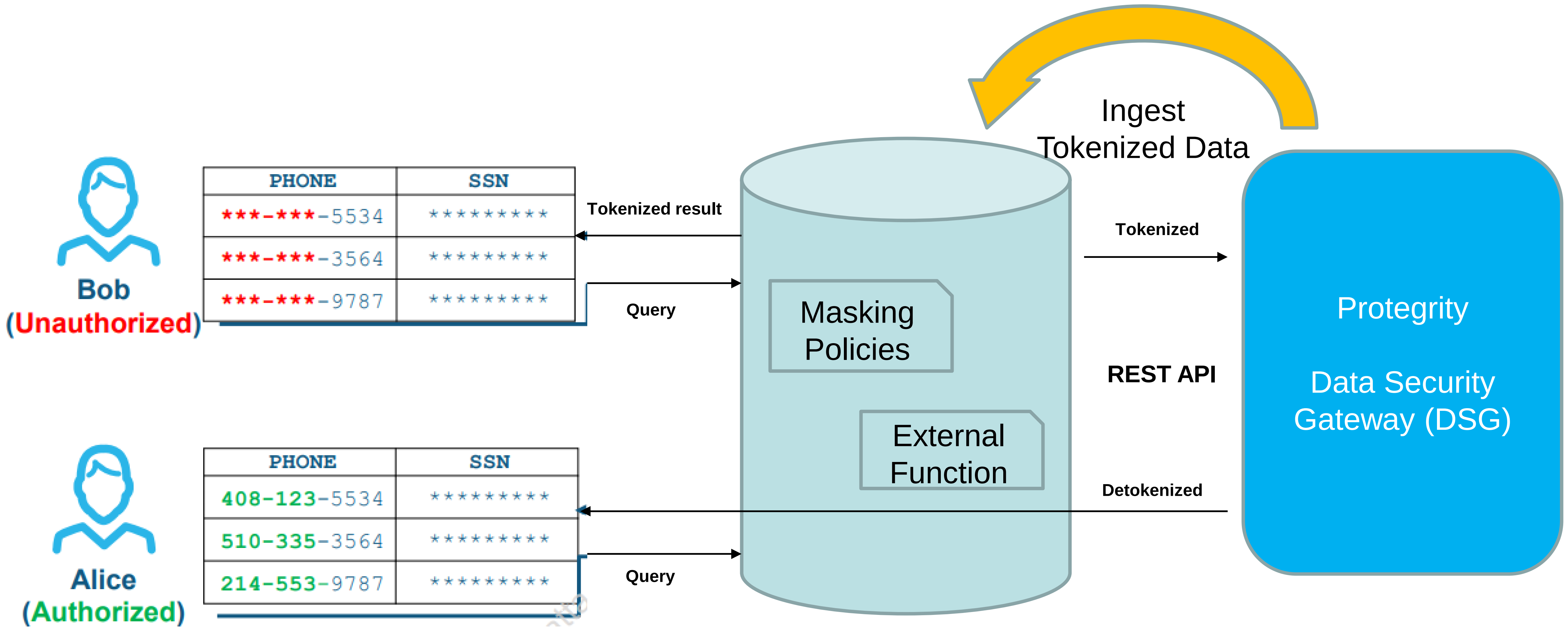
```
create or replace external function local_echo (string_col varchar)
returns variant
api_integration = demonstration_external_api_integration_01
as 'https://xyz.execute-api.us-west-2.amazonaws.com/prod/remote_echo';
```

External Tokenization

- External Tokenization allows organizations to tokenize sensitive data before loading that data into Snowflake and dynamically detokenize data at query runtime using masking policies with External Functions.
- Actual Data is never stored in Snowflake
- Policies determines who can view the de-tokenized data
- When needed, external function is used to retrieve de-tokenized data from provider



External Tokenization



PHONE	SSN
--5534	*********
--3564	*********
--9787	*********

PHONE	SSN
408-123-5534	*********
510-335-3564	*********
214-553-9787	*********

External Tokenization

External Tokenization Partner Integrations
The following partners facilitate external tokenization in Snowflake.
To use these partner integrations, follow the instructions in the partner documentation or contact the partner to begin the configuration process:

- ALTR
- Baffle
- Fortanix
- MicroFocus CyberRes Voltage
- Protegrity
- Privacera
- SecuPI

External Tokenization

Creating a Custom External Tokenization Integration

Step 1: Create an External Function

Step 2: Grant Masking Policy Privileges to Custom Role

Step 3: Create a Masking Policy

Step 4: Apply the Masking Policy to a Table or View Column

Step 5: Query Data in Snowflake

- **Step 1: Create an External Function**
  - Creating External Functions on AWS
  - Creating External Functions on Microsoft Azure
  - Creating External Functions on GCP

External Tokenization

- **Step 2: Grant Masking Policy Privileges to Custom Role**

```
-- create a masking policy administrator custom role

create role masking_admin;

-- grant privileges to masking_admin role.

grant create masking policy on schema <schema_name> to role masking_admin;

grant apply masking policy on account to role masking_admin;

-- allow table_owner role to set or unset the ssn_mask masking policy (optional)

grant apply on masking policy ssn_mask to role table_owner;
```

External Tokenization

- **Step 3: Create a Masking Policy**

```
-- create masking policy

create or replace masking policy email_de_token as (val string) returns string ->
  case
    when current_role() in ('ANALYST') then de_email(val)
    else val
  end;
```

- **Step 4: Apply the Masking Policy to a Table or View Column**

```
-- apply masking policy to a table column

alter table if exists user_info modify column email set masking policy email_de_token;

-- apply the masking policy to a view column

alter view user_info_v modify column email set masking policy email_de_token;
```

External Tokenization

- **Step 5: Query Data in Snowflake**

```
-- using the ANALYST custom role

use role analyst;
select email from user_info; -- should see plain text value

-- using the PUBLIC system role

use role public;
select email from user_info; -- should see tokenized value
```

Search Optimization

- The search optimization service aims to significantly improve the performance of selective point lookup queries on 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

- To add search optimization to a table, follow these steps:

  - Switch to a role that has the privileges to add search optimization to the table.

  - Run the following command:

```
ALTER TABLE [IF EXISTS] <table_name> ADD SEARCH OPTIMIZATION;

Example :
        alter table test_table add search optimization;
```

Thank you!!! ...