

Original Article

# Data Sharing, Disaster Management, and Security Capabilities of Snowflake a Cloud Datawarehouse

Ravi Kashyap

SAP Data Management & Analytics Lead, IBM Corporation, 600 Anton Blvd, Costa Mesa, CA, USA

Received: 11 January 2023

Revised: 12 February 2023

Accepted: 21 February 2023

Published: 28 February 2023

**Abstract** - Data is a valuable resource once provided in the proper context; data becomes information to make high-quality and quick business decisions. In today's world, an important factor that drives a successful organization is gathering data that can be analyzed to gain greater insights into the business and enable new opportunities allowing the business to innovate products/services based on consumer preference. Organizations leverage data to increase customer satisfaction, maximize profitability, and deliver products and services to customers. While information has always been an important asset for an organization, it has never been as critical earlier as it is in business today. Any interruption in data accessibility, lack of adequate data security practices and data breaches can lead to serious financial and legal implications. It is not always possible for organizations to avoid disasters; however, having disaster recovery plans and effective data security strategies can help in minimizing the potential damage and can also completely avoid it. The data warehouse, or enterprise data warehouse (EDW), is an organization's core analytics system that aggregates data from different sources system into a single data warehouse to support data analysis, data mining, machine learning, and artificial intelligence (AI). Choosing a data platform that can handle massive volumes of big data, high speeds, and reliability, not to mention ease of use, is a top priority for any business as they struggle to pull the right information from their data. Snowflake is the most popular solution which supports multi-cloud infrastructure environments such as Amazon Web Services (AWS), Microsoft Azure (Azure) and Google Cloud Platform (GCP). It is a highly scalable cloud data warehouse "as-a-service", enabling users to focus on analyzing data and providing a multilayered security architecture to protect customer data with replication and failover/failback functionality. This paper aims to evaluate Snowflake's capabilities for data sharing within and outside an organization, disaster management, data security, and handling of structured and semi-structured data loads and utilize them in cloud-based reporting solutions.

**Keywords** - Business intelligence, Enterprise reporting, Analytics cloud, Snowflake data sharing, Data warehousing, Disaster management, Data security.

## 1. Introduction

Organizations are handling vast amounts of data which is increasing every day. Various methods and technology have been built to handle these volumes of data at a low cost without worrying about underlying infrastructure costs, data security or maintenance.

The impact of cloud computing on industry and end users would be difficult to overstate: many aspects of everyday life have been transformed by the omnipresence of software that runs on cloud networks. Moving to the cloud means that the resources are owned and managed by a third-party provider instead of the end user. Organizations usually pay for cloud computing resources and storage on a pay-as-you-go basis. Many cloud computing analytics solutions are available in the market, e.g., RedShift by amazon web services, Synapse analytics by Microsoft Azure, and Big Query by Google cloud. Snowflake is also one such solution that provides cloud-based data storage, processing, and analytics.

The Snowflake company was founded in 2012 and publicly listed in 2014. It has supported Amazon S3 since 2014, Microsoft Azure since 2018, and Google Cloud Platform since 2019. Snowflake is a data warehouse provided as SaaS (software as a service) that provides a single platform for data lakes, data warehousing, data engineering, data science, application design, and secure sharing and consumption of real-time/shared data integrated with all widely used visualization tools like Power BI and Tableau. It provides high data security to a User account as data stored in Snowflake tables is highly encrypted using AES-256 strong encryption. The Snowflake architecture allows storage and computation to scale independently, so customers can use and pay for storage and computation separately. Organizations can leverage the sharing capabilities to quickly share governed and secure data in real-time.



## 2. Literature Review

The fact is that cloud computing has its root in other technologies, namely multi-core chips, parallel and distributed systems, virtualization, grid, and Internet technologies. Features that distinguish cloud computing from related on-premises technologies are on-demand self-service, broad network access, resource pooling, rapid elasticity, data storage and computing power which can be increased and decreased on demand without any significant investment in the resources.

With the advent of cloud computing, many data warehouse services are provided by cloud providers. The three biggest cloud providers in the market are Amazon Web Services (AWS), Microsoft Azure (Azure) and Google Cloud Platform (GCP). Snowflake is one such product, and its Cross-Cloud Deployment Capabilities support your multi-cloud strategy in the following cloud platforms: Amazon (AWS), Microsoft (Azure), and Google (GCP). Leveraging a multi-cloud strategy allows the organization to avoid vendor lock-in while taking advantage of the major cloud providers' different services, especially data security, disaster recovery, applications development, and analytics services.

Snowflake is distributed across availability zones of the platform on which it runs; either AWS, Azure, or GCP is designed to operate continuously and tolerate component and network failures with minimal impact on customers. It is SOC 2 Type II certified with additional levels of security, such as support for PHI data for HIPAA customers and encryption across all network communications available. Some organizations are concerned about not having physical server access to where the data gets saved in the event of a failure. Snowflake databases efficiently support disaster recovery scenarios and ensure that multiple data centers replicate and provide easy access to your data if disaster recovery is needed. Snowflake's architecture enables organizations to seamlessly share data with any data consumer within the organization by creating Snowflake users with appropriate roles. For outside the organization, it is possible through reader accounts that can be created directly from the user interface. Snowflake user interface option enables the provider to create and manage a consumer account.

With that in mind, organizations evaluate these data warehousing options on key parameters such as cost to acquire, cost to maintain, data security, availability, skills required to implement and sustain the solution, performance of data warehouse, and ease of implementation.

## 3. Snowflake Architecture and Key concept for Data Warehouse

### 3.1. Architecture of Snowflake

Snowflake distinguishes the storage and compute power, which means businesses with high storage needs but low

CPU cycle requirements, or the opposite, do not have to pay for an integrated package that requires them to pay for both. Users are only charged for the resources they utilize and can modify their scale up or down as needed. Monthly fees for storage in terabytes are charged, whereas computation is billed on a per-second basis. Its architecture consists of three layers, each of which is independently scalable: storage, compute, and service.

#### 3.1.1. Database Storage

The database storage layer holds all data loaded into Snowflake, which can be structured and semi-structured data. Organizations have the option to choose the storage layer of AWS, Microsoft Azure, or Google Cloud Platform that's affordable, scalable, and user-friendly. Snowflake automatically organizes data into its internal optimized, compressed columnar format in the form of blobs, making it more efficient as the query fetches the blobs instead of rows. This storage layer runs independently of compute resources and provides flexibility to scale up or down as needed, and it is billed to the organization based on the resource used.

#### 3.1.2. Compute Layer

The compute layer consists of virtual warehouses that perform data processing tasks required for queries. Warehouses do not share computing resources because each virtual warehouse can access all the data in the storage layer and then operate independently. This allows automatic scaling, which means that compute resources can be scaled as queries are running without the need to redistribute or rebalance the data in the storage layer. These Virtual Warehouses can perform MPP (Massive Parallel Processing), including multiple nodes with CPU and Memory provided in the cloud by Snowflake. Depending on your workload, you can create multiple virtual warehouses on Snowflake for different needs. If your queries are complex, scaling your warehouse is beneficial but expensive. A larger warehouse does not always perform better than a smaller warehouse when dealing with simple queries.

Note: Warehouse billed in seconds with a minimum of 1 minute.

Instead of increasing the warehouse size, the organization also has the flexibility to use multi-clustering in the enterprise version.

Some of the benefits of a virtual warehouse are highlighted below:

- Virtual Warehouses can be started or suspended at any time and can be measured at any time.
- They can also be set to auto-stop or restart automatically once the query fetches the required result.
- It can also be set to default on a smaller and larger collection size. By default, a Snowflake warehouse consists of a single cluster of servers. With multi-cluster, a warehouse can control multiple resources of the predefined warehouse size.

- The cache in this layer is often referred to as Warehouse Cache. This cache contains the raw data queried recently and will disappear once a warehouse is suspended or dropped.

3.1.3. Cloud Services

The cloud services layer uses ANSI SQL and coordinates the entire system. It works as a brain in Snowflake and provides useful statistics like table size, the current schema you are in, or recent query results can be done without the requirement of having a running warehouse.

Services in this layer include:

- Authentication
- Infrastructure management
- Metadata management
- Query parsing and optimization
- Access control

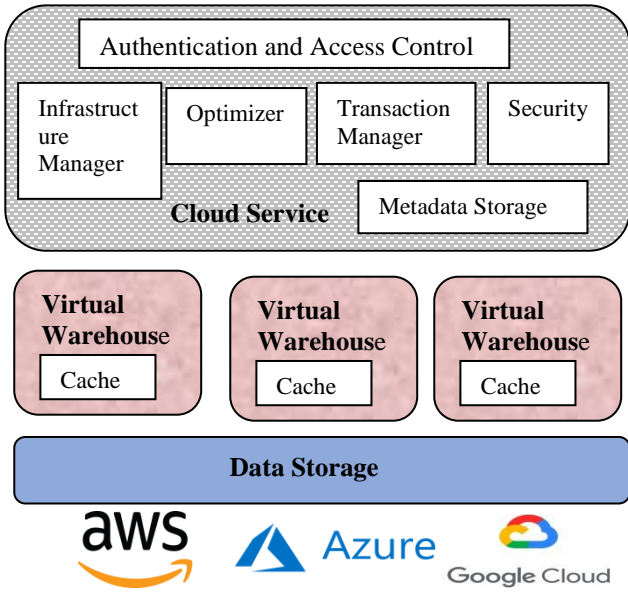


Fig. 1 Snowflake Architecture

Snowflake's performance also depends on the cache in the Services Layer. This layer manages two types of cache: Metadata Cache and Results Cache.

The Results Cache contains aggregated data that was queried recently, not later than 24 hours after the initial query. Snowflake does not need to hit the warehouse to retrieve this result.

Metadata is stored for two things: tables and micro-partitions. Few Data Definition Language (DDL) commands can be run without an active warehouse like CREATE, ALTER, or DROP can be run without a warehouse; this command changes the underlying metadata of the table without processing any actual data. Also, some Data Manipulation Language (DML) commands can be run

without running a warehouse. The metadata cache keeps statistics like row count and table size in bytes.

3.1.4. Virtual Warehouse Scaling Policy

**Standard Policy:** This is the default Snowflake scaling policy. It prevents or minimizes queuing of queries by starting a new cluster. The system performs load checks every minute on the least loaded cluster; after 2 to 3 consecutive checks, if the system determines load can be handled by others or redistributed, the cluster is shut down.

**Economy Policy:** This policy aims to save cost by running the cluster fully loaded for at least 6 minutes rather than starting an additional cluster. A new cluster will start once the system determines that enough queries will have to be queued up initially to start a new cluster. However, to shut down the cluster, instead of 2 to 3 consecutive checks, the system performs 5 to 6 consecutive checks.

3.1.5. Snowflake Editions

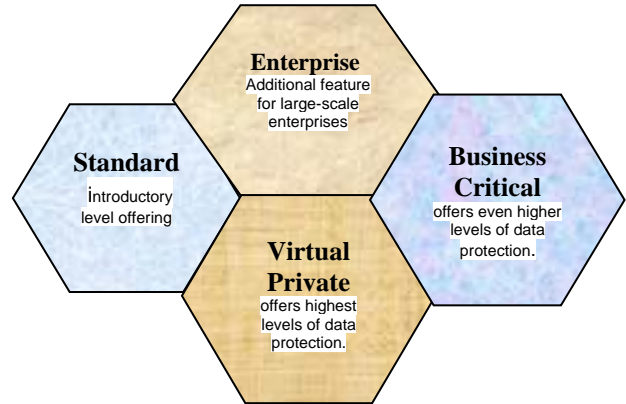


Fig. 2 Snowflake Editions

Standard Edition

The standard edition is an introductory level with full and unlimited access to all the Snowflake features. It provides a complete data warehouse with automatic data encryption. Modified and deleted data can be restored by time travel for up to 1 day, then disaster recovery for 7 days beyond the time travel through fail-safe. It allows secure data sharing with other multiple accounts and across regions with other cloud platforms. Snowflake provides premier customer support 24/7 to resolve any product issue.

Enterprise Edition

Enterprise edition provides all the standard edition features, with additional features of the multi-cluster virtual warehouse for scaling compute resources. Modified and deleted data can be restored by time travel up to 90 days, then disaster recovery for 7 days beyond the time travel through fail-safe. It supports the automatic maintenance of results from Materialized views with Column-level security in the tables or views.

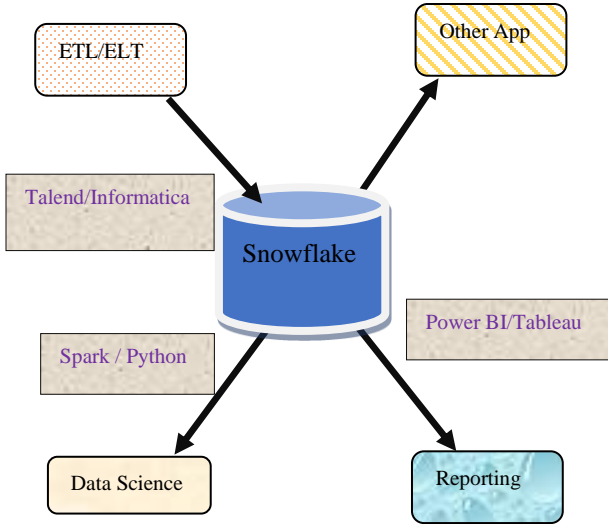


Fig. 3 Snowflake Ecosystem

*Business Critical Edition*

This edition provides all the features of the Enterprise edition. It offers a high level of data protection, such as data encryption everywhere for extremely sensitive data as per organizational needs. It supports database failover and disaster recovery.

*Virtual Private Edition*

This edition includes all the features of the Business-critical edition. It offers the highest level of security, which is useful for financial industries like banks or any organization dealing with highly sensitive data. It provides dedicated virtual servers and a separate Snowflake environment.

**3.2. Snowflake Ecosystem**

Snowflake can connect with third-party technologies and tools leading in today’s industry through a network of connectors, programming languages and drivers such as JDBC and ODBC. A range of Data integration tools like Ab inito, Azure Data factory, boomi, Dendod, Google cloud data fusion, Heap, Hevo, IBM Datastage, Informatica, Kafka, SAP Data service, Talend etc. can connect to Snowflake for ETL activities. The transformation could happen even during the data loading or after the load into Snowflake tables. Once data is loaded into Snowflake, BI tools can further consume it to deliver data visualization through dashboards, charts, and other graphical output. BI tools help in analyzing data deeper and then generate insights, recommendations, and reports accordingly, which will support customers in making better decisions. BI tools such as Adobe, TIBCO, IBM Cognos, Google Data Studio, Power BI and many more can connect with Snowflake.

Machine learning and data science are advanced analytic tools that provide capabilities for statistical and predictive modeling to analyze datasets deeply. Advanced analytics

tools like Alteryx and Apache Spark can be connected with Snowflake, which can scan large data sets to analyze the pattern and find hidden information to predict future trends.

**4. Data Sharing and Accessibility**

Data availability is important for any organization to allow its IT operations to work without any halt, and it is essential to secure data sharing with end-users and partners. Snowflake provides many ways to share the data with another Snowflake account or non-Snowflake account. It allows management who has access to the data and makes data identical across different people and groups.

Before understanding different data-sharing options, first, let us have a quick overview of the loading method in Snowflake - Bulk or Batch loading and Continuous loading. Bulk loading is the most frequent method, which uses the computing power of virtual warehouses; in this method, loading has done using stages. The COPY command has been used to load the data from stages to the table, allowing many transformations to simplify the warehouse. In contrast, the continuous method is suitable for loading a small volume of data which needs to be updated immediately, which means the latest result available for analysis. This method uses a snowpipe (Serverless feature) to load the data.

**4.1. Zero Copy**

Creating backup or cloning a new system environment is always a challenge for any organization; it usually takes hours, days, or weeks to create a test or development environment, which puts an extra cost on the project, but Snowflake zero-copy cloning enables us to create copies of a database, a schema, metadata, primary keys or a table using simple clone command. This feature is useful for taking quick backup that does not cost anything until copied object changes. This can help developers during code build and testing. This feature allows you to make many copies of data without additional storage. It allows users to treat a clone object independently of the original object to perform insert, delete or update.

Note: We cannot clone temporary tables or temporary objects.

SQL statement syntax to clone any object:  
 “CREATE <object\_type> <object\_name>  
 CLONE <source\_object\_name>”

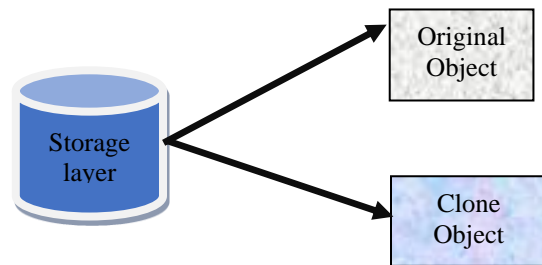


Fig. 4 Zero-copy Cloning

Snowflake provides another similar feature called swapping tables. This option can be used where development table data is required in the production table.

SQL statement syntax to swap table:  
 “ALTER TABLE <table\_name>  
 SWAP WITH <target\_table\_name>”

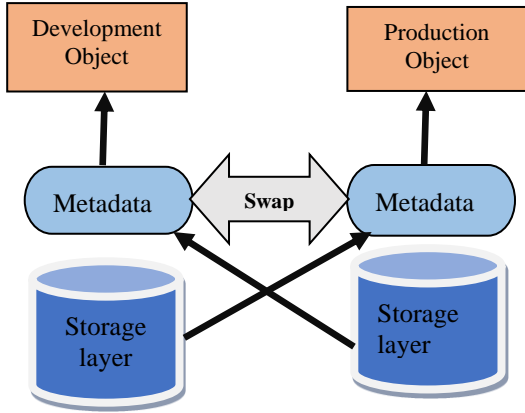


Fig. 5 Swapping Tables

**4.2. Data Sharing**

Generally, in a data warehouse, sharing data is a complicated process as it requires a physical copy of data. If any changes come, then we need to extract the data again.

In Snowflake, data sharing is very easy as it does not require an actual copy of data and is up to date in real time. Data sharing is possible with other Snowflake user accounts or non-Snowflake users through a reader account.  
 Note: Objects shared between accounts are read-only and cannot be modified or deleted.

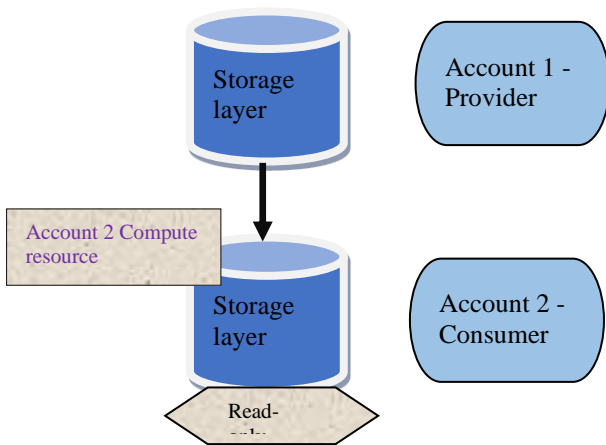


Fig. 6 Data sharing within the Snowflake account

When sharing data with another Snowflake account, the provider (the account that shares the data) uses grants to provide access to selected objects in the database (providing grant access to multiple objects is possible). As a consumer, once you add a shared database to your account, querying any object is like working in your own database. Shared data

can be consumed by its own compute resource, which means the consumer will get charged only for the compute resource used to query the shared data and does not charge for any data storage.

Steps for Data sharing within the Snowflake account:

1. As a provider, first, create a shared object.  
 “CREATE SHARE <SHARE\_OBJECT\_NAME>”
2. Grant usage on the database to the shared object.  
 “GRANT USAGE ON <DATABASE\_NAME> TO SHARE <SHARE\_OBJECT\_NAME>”
3. Grant usage on the schema to the shared object.  
 “GRANT USAGE ON <SCHEMA\_NAME> TO SHARE <SHARE\_OBJECT\_NAME>”
4. Grant selects schema objects.  
 “GRANT SELECT ON <OBJECT> TO SHARE <SHARE\_OBJECT\_NAME>”
5. Validate Grants on Shared objects.  
 SHOW GRANT TO SHARE <SHARE\_OBJECT\_NAME>
6. Add a consumer account to the shared object  
 “ALTER SHARE <SHARE\_OBJECT\_NAME> ADD ACCOUNT= <CONSUMER\_ACCOUNT\_NUMBER>”
7. As a consumer, create a database on the shared object.  
 “CREATE DATABASE <DATABASE\_NAME> FROM SHARE <PRODUCRE\_ACCOUNT>.<SHARED\_OBJECT>”

In many scenarios, organizations need to share the data with third-party or consumers who do not have Snowflake direct access to the database. To facilitate sharing data with users who do not have a Snowflake account can go for a reader account to access the data easily in a cost-effective way.

The reader account belongs to the provider account that created it. Users in the reader account can read the data shared by the provider account but cannot perform any DML operation to manipulate the data. Providers need to pay for the computing resource for the reader account.

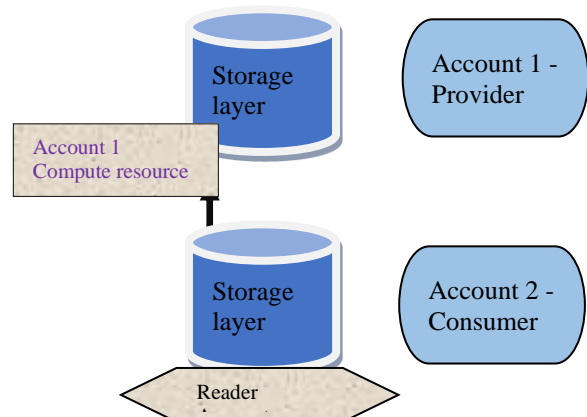


Fig. 7 Data sharing with Non-Snowflake User



Steps for Data sharing within the Snowflake account:

1. As a provider, first create a Reader account, an independent instance with its own url and Compute resource.
2. Then, the provider needs to create the shared objects.
3. In the Reader account, we must create the database based on a Shared object.
4. Multiple users can be created in a reader account to provide access to shared objects.

Note: Only secure views can be shared with another user.

SQL statement syntax to create a secure view  
 “CREATE SECURE VIEW <VIEW\_NAME>  
 AS SELECT <COLUMN\_NAME> FROM <TABLE>”

Users who dislike coding can use the interface option to set up data sharing.

### 5. Fail-safe and Disaster Management

Data is very important for any organization to perform their daily business operation, and a disaster recovery plan must be simple with a minimal amount of manual effort. Snowflake provides the below services to handle the problem of disaster recovery.

1. Time Travel
2. Fail-Safe
3. Data Replication and failover.

#### 5.1. Time Travel

Snowflake time travel feature enables the data restoration within a database as it was at a previous point in time. Users can travel back in time and see the data as it was before any update or how data has looked at a certain time in the past. If a user accidentally drops a table which is very critical for business, then with time travel, it can easily be tackled by un-dropping. Users can revert the data in case of an invalid insert, update or delete. The maximum retention period varies between editions of Snowflake. Organizations should choose the retention period that fits their requirement and budget. The greater the retention period, the greater the storage cost of Time Travel. Standard edition allows up to one-day time travel, and enterprise or higher editions are increased up to 90 days’ time travel.

SQL statement syntax for time travel.  
 “SELECT \* FROM <TABLE\_NAME> AT (OFFSET =>  
 <NO. OF SECOND>)”

Note: Time travel does not work if a table has been replaced. Another SQL statement syntax for time travel.

“SELECT \* FROM <TABLE\_NAME> BEFORE  
 (TIMESTAMP => <PASS TIME STAMP>)”

SQL statement syntax for time travel by certain Query ID to get the result, how it has looked before.  
 “SELECT \* FROM <TABLE\_NAME> BEFORE  
 (STATEMENT => QUERY ID)”

SQL statement syntax for time travel by certain Query ID to get the result, how it has looked before.  
 “SELECT \* FROM <TABLE\_NAME> BEFORE  
 (STATEMENT => QUERY ID)”

SQL statement syntax to un-drop the table or schema, or database.  
 “UNDROPTED TABLE <TABLE\_NAME>”  
 “UNDROPTED SCHEMA <SCHEMA\_NAME>”

SQL statement syntax to check retention policy on any table.  
 “SHOW TABLE LIKE <TABLE\_NAME>”

SQL statement syntax to alter the retention policy of any table.  
 “SET DATA\_RETENTION\_TIME\_IN\_DAYS =NUMBER  
 OF DAYS”

#### 5.2. Fail Safe

Fail-safe feature to provide protection of historical data in case of a disaster or if data gets corrupted in which other methods can no longer recover the data. It provides non-configurable 7 days storage for permanent tables and contributes additional storage costs. This period starts immediately after the time travel period ends; only the Snowflake team can conduct the recovery process through Fail-Safe storage. It should be the last option to recover the data.

Note: After a 7-day period, data cannot be recovered (even by Snowflake).

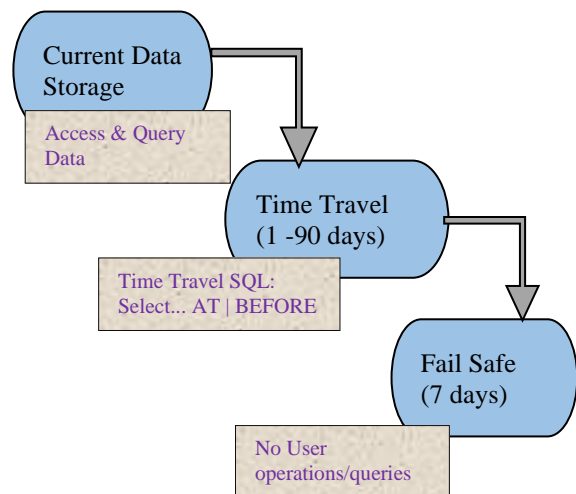


Fig. 8 Data Protection Lifecycle

SQL statement syntax to check storage usage at the account level.

```
“SELECT * FROM
<SNOWFLAKE>.<ACCOUNT_USAGE>.<STORAGE_US
AGE>”
```

SQL statement syntax to check storage usage at the table level.

```
“SELECT * FROM
<SNOWFLAKE>.<ACCOUNT_USAGE>.<TABLE_STOR
AGE_METRICS>”
```

Note: To run the above query, use the ACCOUNTADMIN role.

### 5.3. Snowflake Data Replication and Failover

To avoid disaster, Snowflake provides a data replication process that periodically enables primary database replication to a secondary database in a different availability region, zone and even across cloud providers. All DDL operations performed on the primary database (read-write database) are copied to the secondary database (read-only). Replication feature available for the existing permanent or transient databases.

Snowflake has another feature called failover which can be implemented on the secondary database. In the event of disaster in the primary database, it supports making the secondary database primary to perform database operations to avoid downtime. Snowflake, by default, replicates the data across multiple zones within the region and performs failover in case of a disaster scenario at no additional cost. It is recommended to have database replication with failover across regions to avoid a disaster at a regional scale; this is also supported by Snowflake and is best for the organization to avoid downtime for the database.

Note: Primary and secondary databases would incur storage costs, and additional changes would be associated with data transfer, which depends on the frequency of copying data from the primary database to the secondary database. Business Critical or higher editions of Snowflake are required for cross-region failover.

All the data is stored in underlying database tables in the Snowflake and logically structured as a collection of rows and columns. It supports three types of tables.

- Permanent table
- Transient table
- Temporary table

#### Permanent Table

This is a default type and regular database table in a Snowflake, which can be created easily without additional syntax. It supports all the standard capabilities like time

travel up to 90 days and fail safely with an additional cost associated with data storage.

#### Transient Table

This table behaves like a permanent table but does not support fail-safe periods and provides a 1-day time travel limit. It is ideal where the data is not critical and saves costs associated with fail-safe storage.

SQL statement syntax to create a Transient table

```
“CREATE TRANSIENT TABLE <TABLE_NAME>.”
```

#### Temporary Table

This table does not have a fail-safe period and has limited time travel like a transient table. However, this table exists only in the current session, which is not visible to other users and unavailable once the session is closed. It is recommended to drop these tables once they are no longer required to avoid any storage charges.

SQL statement syntax to create a Temporary table

```
“CREATE TEMPORARY TABLE <TABLE_NAME>.”
```

## 6. Data Security

Organizations invest a lot of money in infrastructure to protect their digital information from unauthorized access or accidental loss. Data breaches can impact the business financially and are a major concern for any organization or industry.

Snowflake provides a full suite of security features and multiple data security techniques to avoid potential risks. All the data stored in Snowflake tables is encrypted using AES-256 strong encryption. Three security layers Snowflake uses to secure the data: Network security, Identity and access management, and data encryption.

### 6.1. Network Security

The first line of defense is network security which protects Snowflake from outside attacks and unauthorized access and can be used to set up secure access to cloud storage for loading/unloading data and allow the firewall to connect a client application to Snowflake. Use Snowflake network policies to connect with known client IP address range while blocking others. Snowflake allows setting up policies for specific scenarios to override the account-level policies. It also gives the feature to set up DNS to resolve Snowflake's private URL. Using private DNS for a customer who is using both on-premises and cloud provider network resources is a best practice.

### 6.2. Identify and Access Management

Providing access and authorization to the right users is very important to secure the data. Snowflake provides features to manage user access and roles, authentication and

single sign-on, and access control at the object level and session. To provision and externally manage users and roles in Snowflake, it is recommended to use SCIM where supported by your Identity Provider

Snowflake has two aspects of access control: DAC (Discretionary Access Control) and RBAC (Role-based Access Control). In DAC, each object has an owner who has the authority to grant access to others on that object, whereas in RBAC, access privileges are assigned to roles, and then these roles are assigned to users.

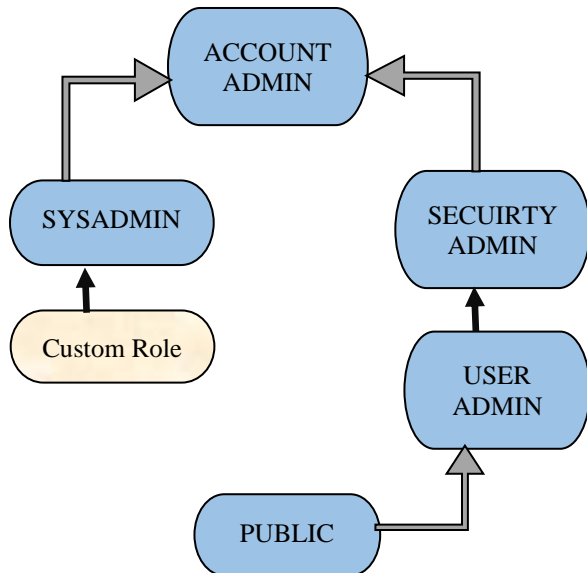


Fig. 9 Snowflake Roles

*Account Admin*

Its top-level role in the system and inherited all other roles should be granted only to limited users in the organization. This role manages and views all objects and does all of the account operations activities, such as adding and monitoring billing information. This role is important for setup the initial environment and managing account-level objects. It is recommended that users who use this role should use multifactor authentication, and at least 2 users should be assigned to that role.

*Security Admin*

It can create and manage users and roles. It has limited access to the account admin tab in Snowflake and can manage any object grant globally.

*Sysadmin:*

The role's main purpose is to manage data warehouses, databases, and other objects. All customer roles should be assigned under this role as best practice.

*Useradmin:*

Dedicated to user and role management only.

*Public*

This role is granted automatically to every user. Users with public roles can create objects and then, as owners, can assign privileges to other users.

Note: Every object is owned by a single role; by default, Owner has all privileges.

**6.2. Data Encryption**

All the data in Snowflake has been encrypted using transparent encryption and a key hierarchy that encrypts individual data pieces using different keys. Snowflake rotates the keys every 30 days to use a new key hierarchy for new data. Organizations can use the built-in encryption function of a Snowflake in addition to transparent encryption to encrypt/decrypt specific columns.

In addition to these security layers, Snowflake provides another advanced feature called dynamic data masking at the column level. It is a policy-based security feature that keeps the data in the database unchanged while hiding sensitive data and provides full control to the data owner to authorize how they can see the data. Masking policies are schema-level objects that can be applied to one or more columns in a table or a view.

Note: This policy-based approach with RBAC prevents sensitive data from being revealed by even table/view owners.

SQL statement syntax to set up a masking policy.

```

"CREATE MASKING POLICY <POLICY_NAME>
AS ( VAL DATATYPE OF ORIGINAL COLUMN)
RETURNS DATATYPE →
CASE
WHEN
<MASKING CONDITION>
THEN '*****'
ELSE
VAL
END;"
    
```

SQL statement syntax to apply policy on a specific column.

```

"ALTER TABLE IF EXISTS <TABLE_NAME> MODIFY
COLUMN <COLUMN_NAME>
SET MASKING POLICY <POLICY_NAME>"
    
```

**7. Conclusion**

Snowflake data warehouses are one of the most popular architectures allowing storage and computing to scale independently, so customers can use and pay for storage and computation separately. It is offered as a software as a service (SaaS) that can be quickly implemented without any expensive software or hardware configuration. It resolves many of the problems found in older hardware-based data



warehouses, such as limited scalability, data transformation issues, and delays or failures due to high query volumes. Snowflake cross-cloud and cross-region connectivity features make it easy for organizations to connect their data in real-time, regardless of where it is located. It is the most scalable,

cost-effective, and secure cloud solution for data warehousing with features like auto-scaling warehouse size, auto suspend, and data sharing without any administrative cost.

## References

- [1] Baljit Singh, "Enterprise Reporting on SAP S/4HANA Using Snowflake As Cloud Datawarehouse," *International Journal of Computer Trends and Technology*, vol. 71, no. 1, pp. 45-49, 2023. *Crossref*, <https://doi.org/10.14445/22312803/IJCTT-V71I1P107>
- [2] Naga Mallika Gunturu, "Enterprise API Transformation: Driving Towards API Economy," *International Journal of Computer Trends and Technology*, vol. 70, no. 6, pp. 44-50, 2022. *Crossref*, <https://doi.org/10.14445/22312803/IJCTT-V70I6P105>
- [3] The Snowflake Website, 2020. [Online]. Available: [https://docs.snowflake.com/en/?\\_ga=2.131591436.2086032114.1673908722-1347278072.1673734864](https://docs.snowflake.com/en/?_ga=2.131591436.2086032114.1673908722-1347278072.1673734864).
- [4] The SAP Website, 2022. [Online]. Available: [https://help.sap.com/docs/sap\\_hana\\_platform/4505d0bdaf4948449b7f7379d24d0f0d/bc18c37df9154cb0bfacad4b045f7c49.html](https://help.sap.com/docs/sap_hana_platform/4505d0bdaf4948449b7f7379d24d0f0d/bc18c37df9154cb0bfacad4b045f7c49.html)
- [5] The SAP Website, 2022 [Online]. Available: <https://community.sap.com/topics/bw4-hana/hana-2-0>
- [6] Sikender Mohsienuddin Mohammad, "Cloud Computing in IT and How It's Going to Help United States Specifically," *International Journal of Computer Trends and Technology*, vol. 67, no. 10, pp. 103-109, 2019. *Crossref*, <https://doi.org/10.14445/22312803/IJCTT-V67I10P118>
- [7] Haleem Khan, and Yu Jiong, "Cloud Computing Effect on Enterprises in Terms of Cost," *International Journal of Computer Trends and Technology*, vol. 67, no. 5, pp. 14-19, 2019. *Crossref*, <https://doi.org/10.14445/22312803/IJCTT-V67I5P103>
- [8] Yonah Etene, and Josphat M. Karani, "Performance in Layered Software Architectures: the Case of Customized Organizational Software," *International Journal of Computer Trends and Technology*, vol. 67, no. 12, pp. 15-19, 2019.
- [9] Marcel Chibuzor Amaechi, Matthias Daniel, and Bennett. E. O, "Data Storage Management in Cloud Computing Using Deduplication Technique," *SSRG International Journal of Computer Science and Engineering*, vol. 7, no. 7, pp. 1-7, 2020. *Crossref*, <https://doi.org/10.14445/23488387/IJCSE-V7I7P101>
- [10] G. Anitha et al., "A Survey of Security Issues in IIOT and Fault Identification Using Predictive Analysis in Industry 4.0," *International Journal of Engineering Trends and Technology*, vol. 70, no. 12, pp. 99-108, 2022. *Crossref*, <https://doi.org/10.14445/22315381/IJETT-V70I12P211>
- [11] Twinkle Graf.F, and Mrs.Prema. P, "Secure Collaborative Privacy in Cloud Data with Advanced Symmetric Key Block Algorithm," *SSRG International Journal of Computer Science and Engineering*, vol. 2, no. 2, pp. 40-44, 2015. *Crossref*, <https://doi.org/10.14445/23488387/IJCSE-V2I2P109>
- [12] Sidath Weerasinghe, and Indika Perera, "Taxonomical Classification and Systematic Review on Microservices," *International Journal of Engineering Trends and Technology*, vol. 70, no. 3, pp. 222-233, 2022. *Crossref*, <https://doi.org/10.14445/22315381/IJETT-V70I3P225>
- [13] Manjunatha. S, and Dr. Suresh. L, "A Study on Consolidation of Data Servers in Virtualized Cloud Atmosphere," *SSRG International Journal of Computer Science and Engineering*, vol. 6, no. 11, pp. 47-50, 2019. *Crossref*, <https://doi.org/10.14445/23488387/IJCSE-V6I11P110>
- [14] Dr. P. K. Rai, and Rajesh Kumar Bunkar, "Architectural Data Security in Cloud Computing," *International Journal of Computer & Organization Trends*, vol. 4, no. 4, pp. 23-27, 2014. *Crossref*, <https://doi.org/10.14445/22492593/IJCOT-V4I4P306>
- [15] N. Madhavi Latha, and Y. Pavan Narasimha Rao, "A Novel Secured Data Transmission Model with Load Balancing for Cloud Computing," *International Journal of Computer & Organization Trends*, vol. 6, no. 1, pp. 45-50, 2016. *Crossref*, <https://doi.org/10.14445/22492593/IJCOT-V29P304>
- [16] Disaster Management. [Online]. Available: <https://docs.snowflake.com/en/user-guide/replication-intro.html>
- [17] Snowflake Security. [Online]. Available: <https://www.snowflake.com/trending/intro-to-data-security>
- [18] Statistics and Machine Learning at Scale.