**What is snowflake?**

Snowflake is the most popular (Software-as-a-Service (SaaS) based) cloud data warehouse solution to storage and analytics of data. It works on all major cloud platforms, including AWS (Amazon Web Services), Microsoft Azure, and GCP (Google Cloud Platform). It works with structured and semi-structured data.

It supports different programing language including Go, C, .NET, Java, Python, Node.js, etc.

**Snowflake is OLAP?**

Snowflake is considered OLAP (Online Analytical Processing) technology. As online data retrieving and data analysis system using historical data.

**Why it is SAAS?**

Snowflake is considered as SAAS, means software as a service because it is data warehouse and analytics service over cloud. It allows user to access the platform via a web browser without configure any hardware or installing any software.

**What are the benefits of using Snowflake?**

There are many reasons why organizations are using Snowflake. Here’s some of the key benefits and advantages:

* **Fast data retrieval:** Snowflake works efficiently with massive data volumes with faster data retrieval.
* **Auto-scaling:** Snowflake supports auto-scaling that dynamically configures the compute resources. For example, adding more clusters if the workload increases and scaling down clusters if the workload decreases. This auto-scaling ensures high performance with cost optimization.
* **Concurrency:** It supports multiple users and workloads simultaneously without compromising performance.
* **Advanced security:** Snowflake has industry-leading security features for protecting data. These features include federated authentication and single sign-on (SSO), key-pair authentication, multi-factor authentication, network policies and rules, private endpoints, access controls, encryption, and an OAuth security interface.
* **Business Intelligence support:** it supports Various BI tools. For example, Adobe Analytics, Amazon QuickSight, Google Looker Studio, Microsoft Power BI, Oracle Analytics.
* **SQL queries:** It uses the SQL (SnowSQL) command line client tool to execute the SQL queries and perform DLL\DML operations.
* **Data sharing:** Snowflake's unique feature is, it shares data among partners without duplicating it.
* **Seamless integration:** Snowflake supports seamless integration with various data integration tools, allowing organizations to ingest and transform data from different sources into their Snowflake data warehouse.

**What is virtual warehouse?**

A virtual warehouse is basically a collection of computing resources (like CPU, memory, Solid state drive, etc.) customers can access it to run queries, load data, and perform other DML (Data Manipulation Language) and SQL (Structured Query Language) operations.

There are 3 different states of the Snowflake Virtual Warehouse:

* **Cold Virtual Warehouse:** If you are running a query whilst your virtual warehouses are not active, it will start a new instance of a ‘Cold’ virtual warehouse.
* **Warm Virtual Warehouse:** If your current virtual warehouse is active and has processed queries, this is called a ‘Warm’ virtual warehouse.
* **Hot Virtual Warehouse**: If the virtual warehouse is active and has processed queries, this is called a ‘Hot’ virtual warehouse.

A virtual warehouse can be created through the web interface or using SQL. These are the 3 different methods:

* Snowsight: Select Admin > Warehouses > Warehouse
* Classic Console: Select Warehouses > Create
* SQL: Execute a CREATE WAREHOUSE command,
* Snowflake high-level analytics functionalities −
* Data Transformation
* Supports for Business Application
* Business Analytics/Reporting/BI
* Data Science
* Data Sharing to other data systems
* Data Cloning

### **Explain Snowflake Architecture.**

The Snowflake architecture is a hybrid of shared-disk (A common disk or storage device is shared by all computing nodes) and shared-nothing (Each computing node has a private memory and storage space) database architecture.

As with shared-nothing architectures, Snowflake uses massively parallel computing (MPP) clusters for query processing, in which each node stores part of the whole data set locally.

The Snowflake architecture is divided into three layers:

* **Database Storage Layer:** In this layer, structured and semi-structured data is stored and automatically compressed, encrypted, and organized into micro-partitions. This is done to fully optimize storage and improve query performance.
* **Query processing:** Snowflake query execution occurs in the processing layer. It uses a concept known as a virtual warehouse or warehouse, which is a cluster of computing resources. Each warehouse comprises multiple compute nodes (MPP compute cluster) allocated by Snowflake. Each virtual warehouse is an independent compute cluster. It provides all the required resources for running queries in a Snowflake session. These queries include DML queries such as updating, loading, and unloading data.
* **Cloud services:** The cloud service layer collection of services that coordinating activities in Snowflake. It includes authentication, infrastructure management, metadata management, query parsing, optimization, and access control.

### **Can you tell me how to access the Snowflake Cloud data warehouse?**

Snowflake's data warehouse can be accessed using the following ways:

* ODBC Drivers (a driver for connecting to Snowflake).
* JDBC Drivers (a driver enabling a Java application to interact with a database).
* Python Libraries (for creating Python applications that connect to Snowflake and perform standard operations).
* Web User Interface (can be used for almost any task you can accomplish with SQL and the command line, such as: Creating and managing users and other account-level objects).
* SnowSQL Command-line Client (Python-based command-line interface to connect to Snowflake from Windows, Linux, and MacOS).

### 5. **What is the difference between snowflakes and redshift**

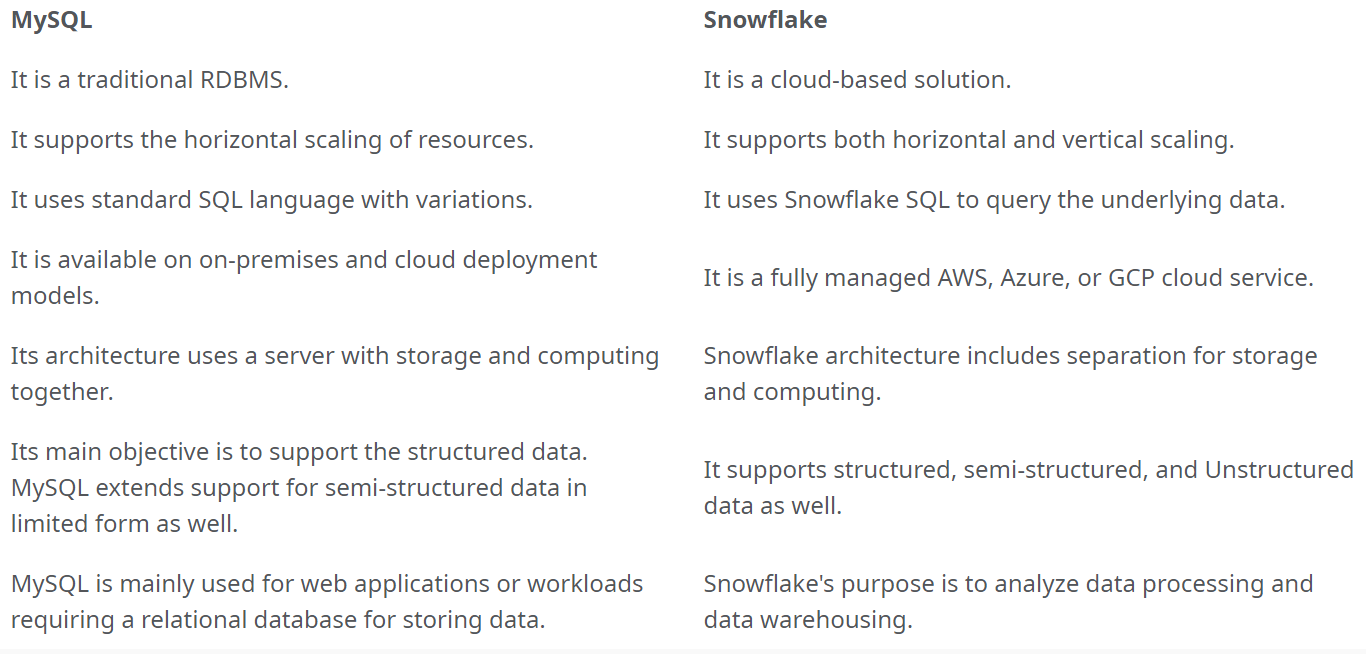
**Is Snowflake a database?**

Yes, Snowflake is a database warehouse solution built for cloud-native architecture. It can be considered a cloud-based relational database or SaaS data warehouse. Snowflake includes the features of the traditional database with cloud capabilities. A few characteristics are as follows:

* Snowflake stores data in compressed, optimized, and multiple micro partitions in the columnar format.
* Snowflake supports the standard SQL statements for querying data. These statements include SELECT, TOP, FROM, JOIN, PIVOT, UNPIVOT, GROUP BY, HAVING, ORDER BY, LIMIT, FETCH.
* Like a traditional database, it includes security features such as encryption, access controls, and multiple authentication mechanisms.
* It provides mechanisms to quickly scale up or down your Snowflake resources to meet workload-changing needs.

## Is Snowflake SQL or MySQL?

Snowflake does not have a direct relationship with SQL or MySQL.



### **Explain stages in Snowflake.**

Stages are locations in Snowflake where data is stored.

And staging is the process of uploading data into a stage.

There are two types of staging 1. Internal and 2. External

**External Stages**: When data is stored in another cloud region (like in AWS), this is known as an external stage.

**Internal Stages:** when it is stored inside a snowflake, it is known as an internal stage.

Internal stages can be further categorized as follows:

**user:** Each user has a Snowflake stage allocated to them by default for storing files and these cannot be altered or dropped. These stages are unique to the user, meaning no other user can access the stage. User Stages are not suitable option if files need to be accessed by multiple users.

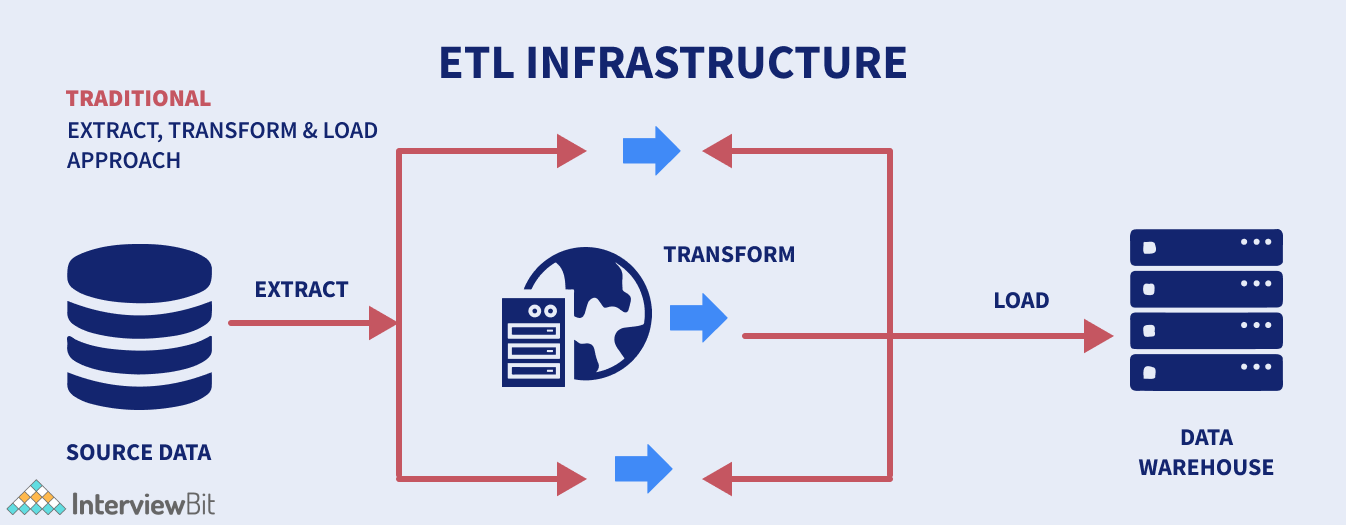
**Table**: Each table has a Snowflake stage allocated to it by default for storing files and these cannot be altered or dropped. Table stages can be accessed by multiple users, but we cannot load data into the multiple tables. Table stages are not suitable if the data needs to be loaded into multiple tables.

**Named:** Named stages are database objects that provide the greatest degree of flexibility for data loading. They overcome the limitations of both User and Table stages.

* Named stages are accessible by all the users with appropriate privileges.
* The data from Named stages can be loaded into multiple tables

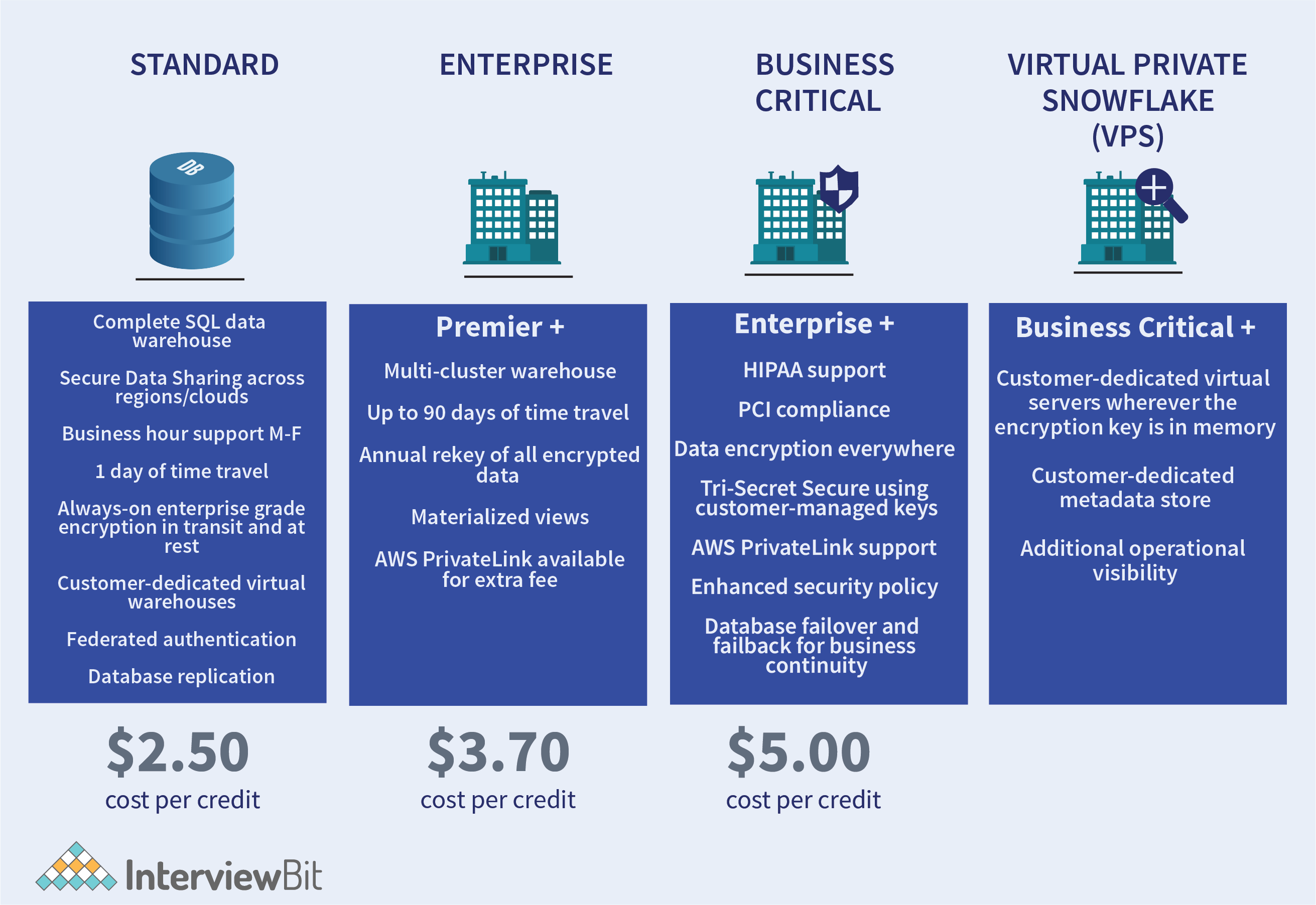
### **Is Snowflake an ETL (Extract, Transform, and Load) tool?**

Yes, Snowflake is an [**ETL**](https://www.interviewbit.com/etl-testing-interview-questions/) (Extract, Transform, and Load) tool, since it is performed in three steps, including:



* The data is extracted from the source and saved in data files in a variety of formats including JSON, CSV, XML, and more.
* Loads data into a stage, either internal (Snowflake managed location) or external (Microsoft Azure, Amazon S3 bucket, Google Cloud).
* The COPY INTO command is used to copy data into the Snowflake database.

### **What are different snowflake editions?**



### 

### **What do you mean by zero-copy cloning in Snowflake?**

Zero-copy cloning is one of the great features of Snowflake. It basically allows you to duplicate the source object without making a physical copy of it or adding additional storage costs to it

**What is data object?**

### In Snowflake, a data object refers to any entity that contains or represents data, such as tables, views, or external stages. Essentially, it's anything within Snowflake where you can store or manipulate data.

## Framework to Access Control in Snowflake Roles

In Snowflake, there are two approaches/methods to establish access control:

1. **Discretionary Access Control (DAC)**:  Each object has an owner, who can in turn grant access to that object.
2. **Role-based Access Control (RBAC):** access privileges are assigned to roles, which are in turn assigned to users.

Some key elements to Access Control in Snowflake Roles are as follows:

* Securable object: A secure object is one to which permission can be granted. Access will be refused unless a grant by the admin allows it.
* Role: A role is a type of entity to which privileges can be assigned. It’s worth noting that roles can be given to other objects, forming a hierarchy.
* Privilege: Privilege is a level of access to an object that is defined. To manage the granularity of access allowed, multiple separate privileges might be employed.
* User: Snowflake recognizes a user’s identity, whether it’s affiliated with a person or a program.

**What Is Snowflake Micro-Partitions?**

Micro- Partitions are contiguous units of storage. Each micro-partition contains between 50 MB and 500 MB of uncompressed data. Snowflake uses a unique architecture to store table data in columnar format across a large number of micro-partitions.

Key aspects:

* Small units of table storage automatically managed by Snowflake
* New micro-partitions are added as data grows
* Micro-partitions transparently pruned during query execution
* Columnar storage provides performance benefits
* Parallel queries can process multiple micro-partitions
* Provides scalability and query concurrency

## Table Types& views

Snowflake categorizes tables into different types based on its uses and nature. There are four types of tables –

**Permanent Table:** It is default table as well as Permanent tables are created in the database. These tables exist until deleted or dropped from database. It is Fail-safe and data can be recovered if lost due to fail.

**Temporary Table:** It exists for a shorter duration. It is mostly used for transitory data like ETL/ELT. It is not fail-safe, which means data cannot be recovered automatically.

**Transient Table:** Transient table is similar to permanent table, available to all users and session, exists until dropped but does not have fail-safe features. Time travel is possible in transient tables but only for 0 to 1 day.

### **External Table:** External tables are like outside of snowflake, and they can't be dropped or deleted. It should be removed. External tables are only meant for reading. Time travel is not possible for external tables.

It is not fail-safe inside Snowflake environment.

**Time travel:** Snowflake Time Travel enables accessing historical data (i.e. data that has been changed or deleted) at any point within a defined period. It serves as a powerful tool for performing the following tasks:

* Restoring data-related objects (tables, schemas, and databases) that might have been accidentally or intentionally deleted.
* Duplicating and backing up data from key points in the past.
* Analyzing data usage/manipulation over specified periods of time.

**Time Travel SQL Extensions**

To support Time Travel, the following SQL extensions have been implemented:

* [AT | BEFORE](https://docs.snowflake.com/en/sql-reference/constructs/at-before) clause which can be specified in SELECT statements and CREATE … CLONE commands (immediately after the object name). The clause uses one of the following parameters to pinpoint the exact historical data you wish to access:
* CLONE: To create a logical duplicate of the object at a specific point in its history.
* TIMESTAMP: From a given time (Data & Time) provided.
* OFFSET: Time difference from current time till offset provided in seconds.
* STATEMENT: Using a Statement ID from the point where the last DML query was fired.
* UNDROP: If a table is dropped accidentally, it can be restored using the UNDROP command.

**Data Retention Period**

A key component of Snowflake Time Travel is the data retention period.

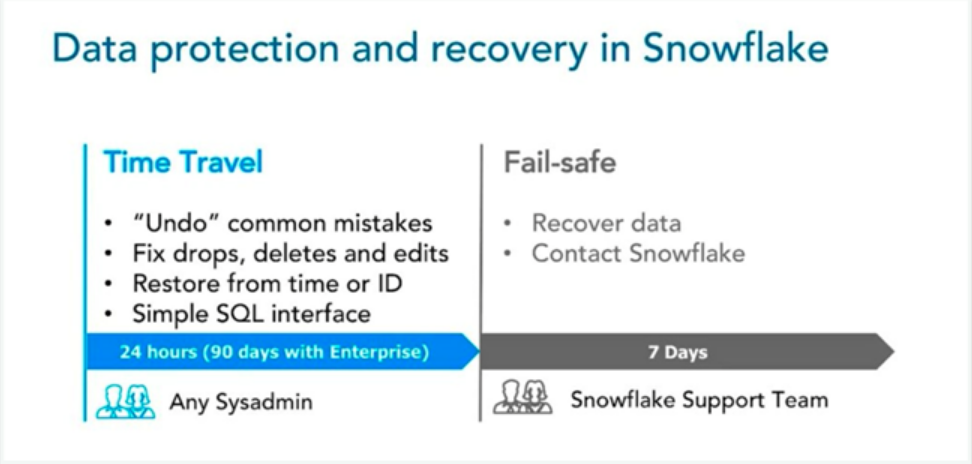
The data retention period specifies the number of days for which this historical data is preserved and, therefore, Time Travel operations (SELECT, CREATE … CLONE, UNDROP) can be performed on the data.

The standard retention period is 1 day (24 hours).

Standard and enterprise periods

**Fail-Safe:** Snowflake provides another exciting feature called Fail-safe where historical data can be protected in case of any failure. Fail-safe allows a maximum period of 7 days which begins after the Time Travel retention period ends wherein Historical data can be recovered.

No user operation is allowed. Data recover is possible by snowflake.



Time travel syntax