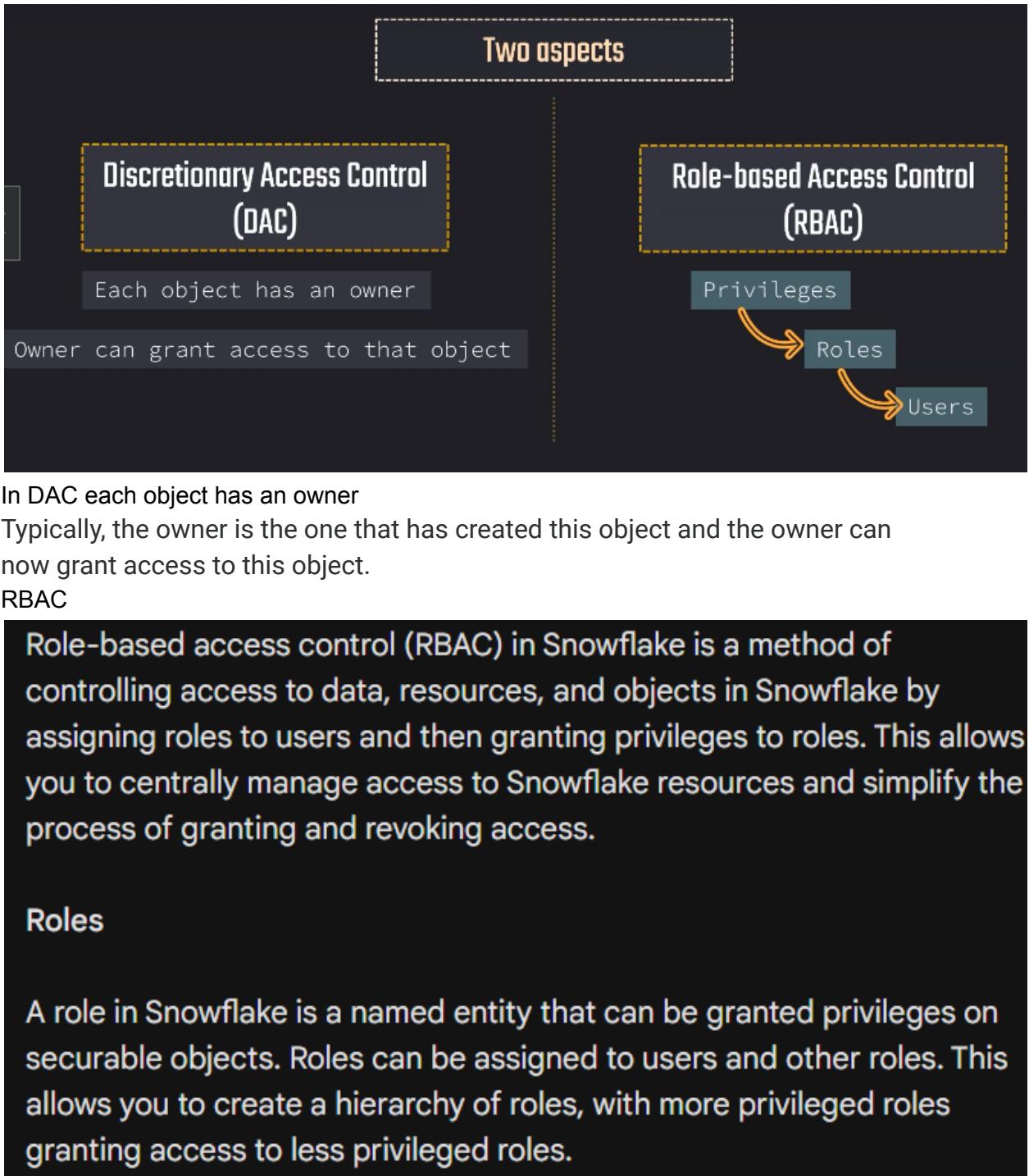


## Access Control in Snowflake



## Privileges

Privileges in Snowflake are the permissions that users need to perform specific actions on securable objects. For example, the `SELECT` privilege allows users to query data in a table, and the `INSERT` privilege allows users to insert new data into a table.

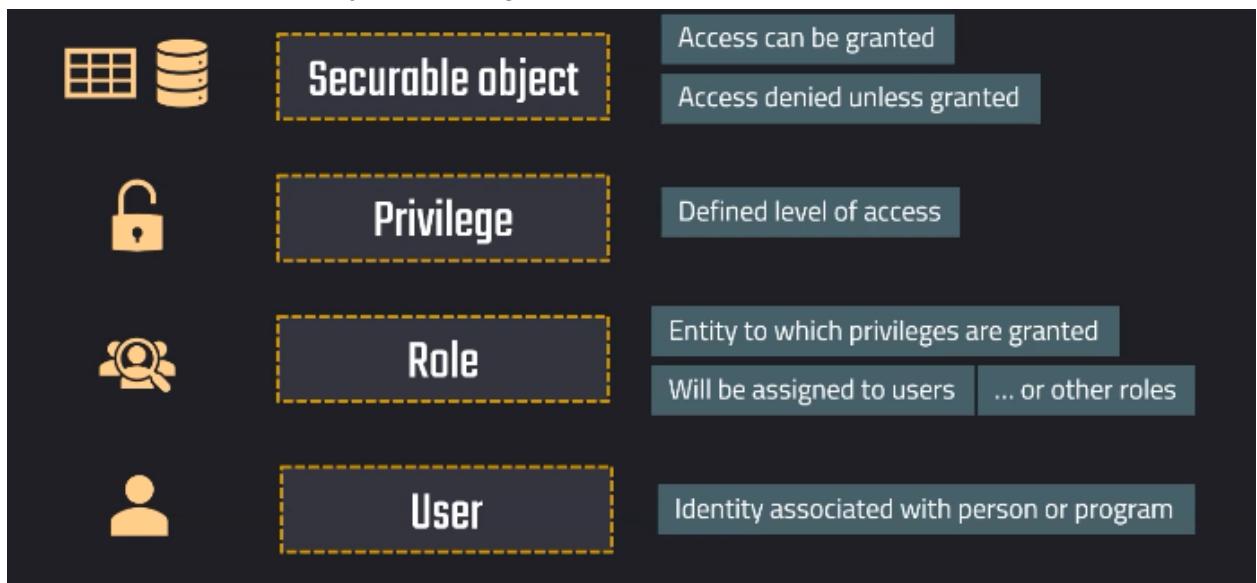
### Granting and revoking privileges

You can grant and revoke privileges on roles using the `GRANT` and `REVOKE` statements. For example, the following statement grants the `SELECT` privilege on the `my_table` table to the `my_role` role:

SQL

```
GRANT SELECT ON my_table TO my_role;
```

- 5.
6. Key concepts in Access Control
7. Here the user will be assigned some roles..
8. Based on this roles..a user will get some privileges...and based on this privileges ..the access to the securable object can be granted



- For example..when we are using worksheets...we need to choose a role to create or query tables etc

The image shows two side-by-side screenshots from the Google Cloud Platform interface.

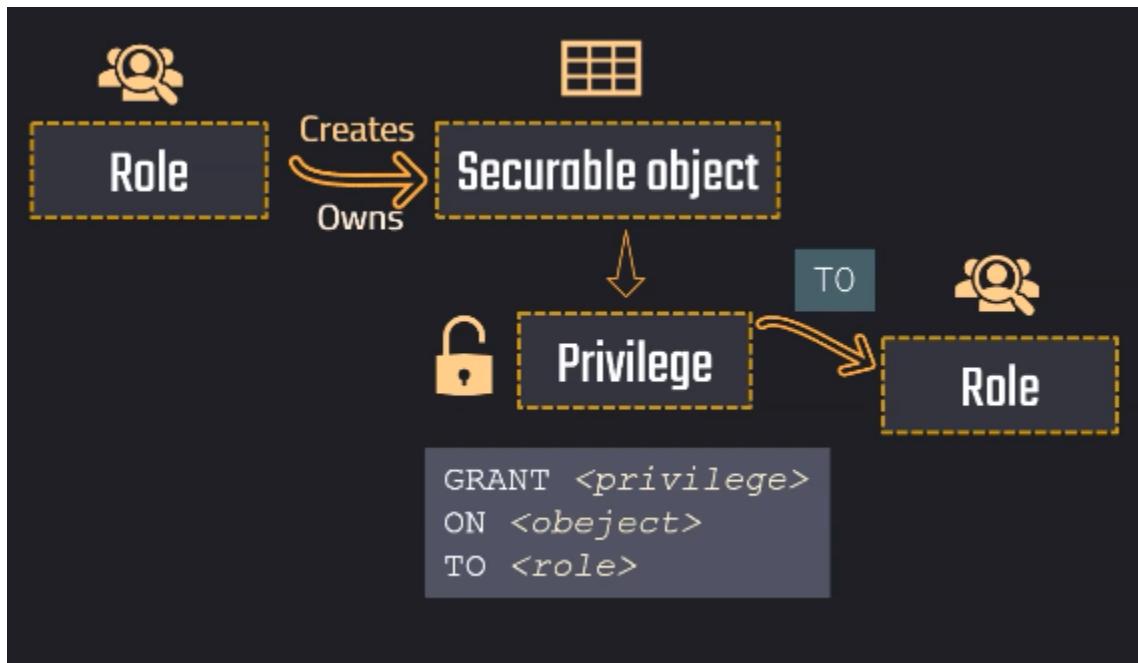
**Left Panel (Roles):**

- Search bar: Roles
- List of roles:
  - ACCOUNTADMIN (selected, marked with a checkmark)
  - ORGADMIN
  - PUBLIC
  - SECURITYADMIN
  - SYSADMIN
  - USERADMIN

**Right Panel (Warehouses):**

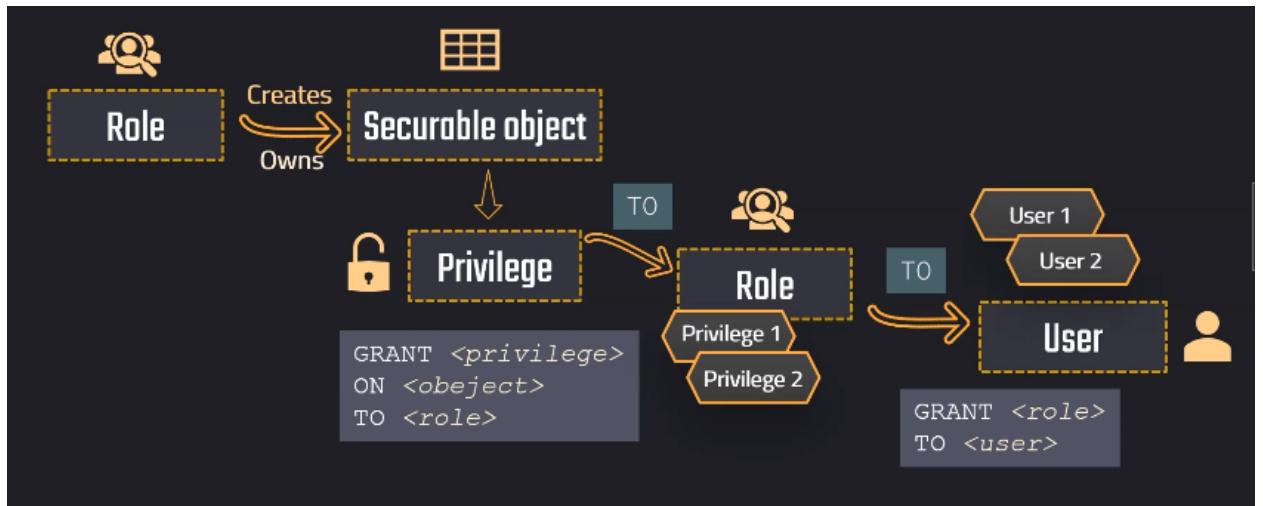
- Search bar: Warehouses
- List of warehouses:
  - COMPUTE\_WH (selected, marked with a checkmark)
- Size dropdown: X-Small (selected, marked with a checkmark)

- Now using this role here we have created a table(secured object)...so now this role is the owner of the this table which is created now
- And now this role can grant privileges to the other role

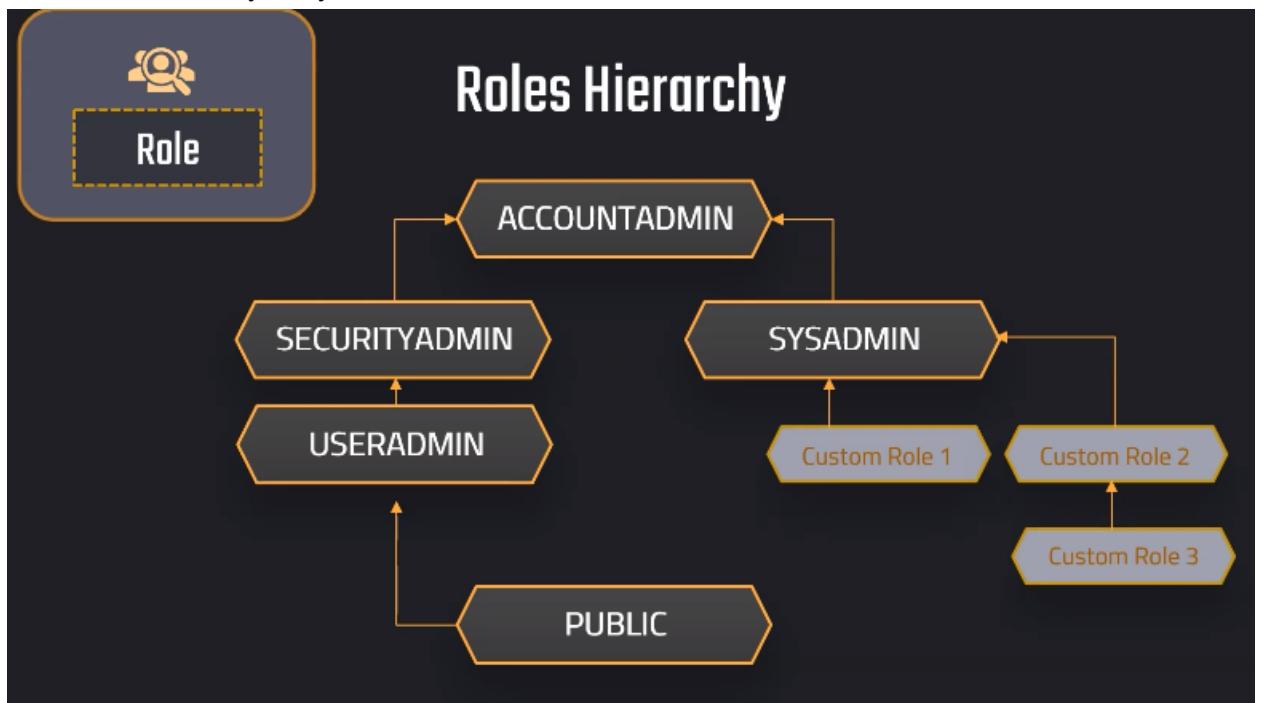


- Now this roles may have some privileges ..
- Privileges are nothing but...having rights on properties

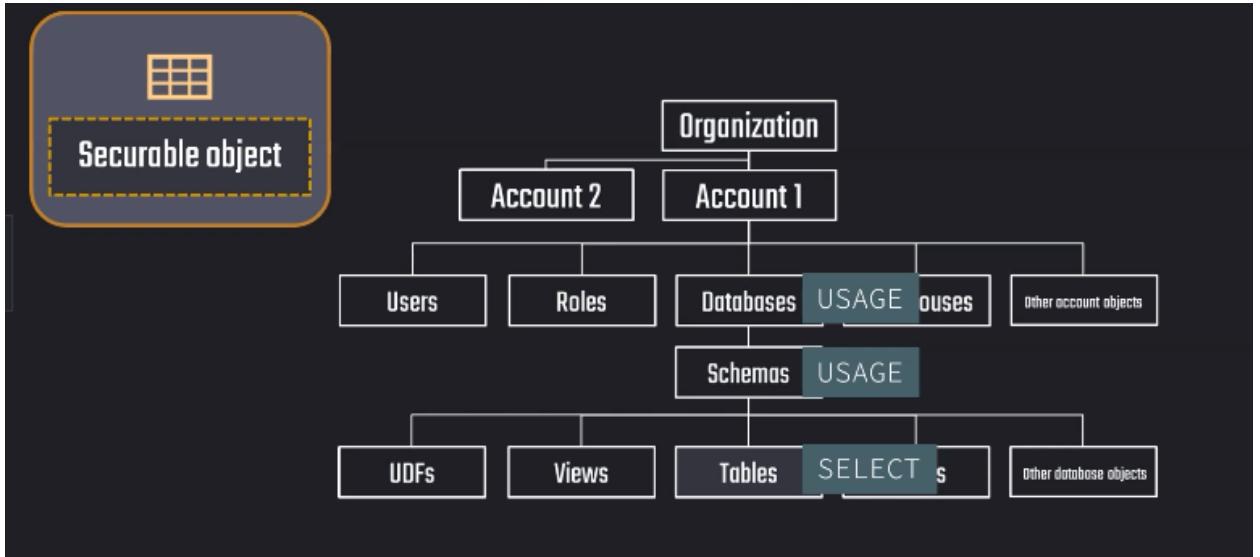
14. And now we can grant this role to some users..using grant command



15. There is a hierarchy of system defined roles

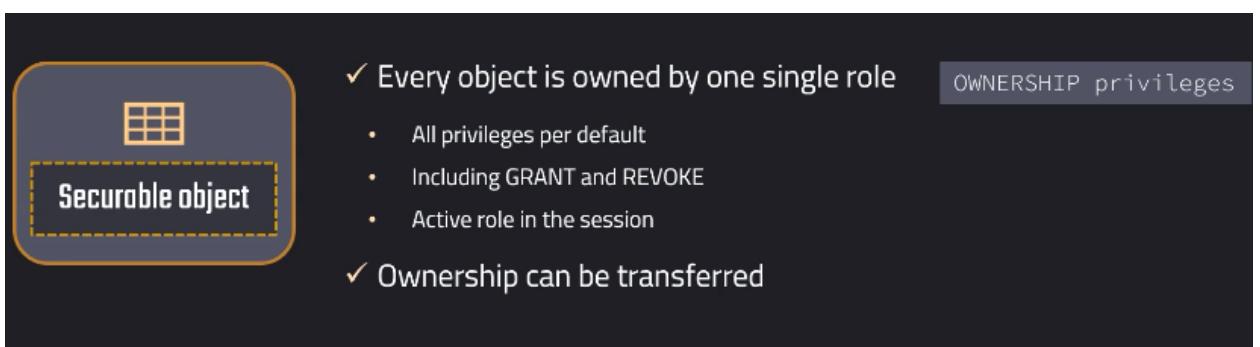


16. we just have to keep in mind that some privileges will be inherited through this hierarchy and we can also create custom roles.



17.

18. And in here we have to keep in mind that when we want to grant privileges to, for example, a child object, let's say a table, we also have to grant privileges to the parent object.
19. In this case, we would have to grant also usage to the schema and the database where this table is contained in.

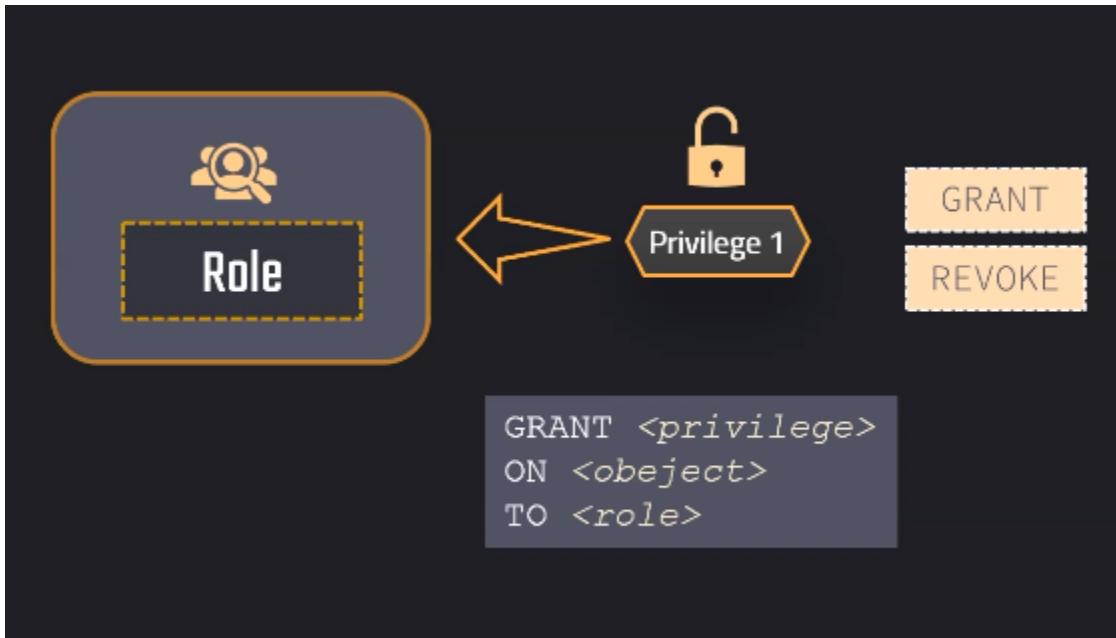


20.

21. So a securable object has always one single owner.
22. This is the one that has the ownership privilege. This cannot be assigned to multiple roles, but it can be transferred to another role.
23. So this ownership privilege also contains the grant and revoke privilege, which means the privileges on this object can be also granted or revoke to other roles.
24. When we create this object, this will be the active role that will be then the owner of this object.

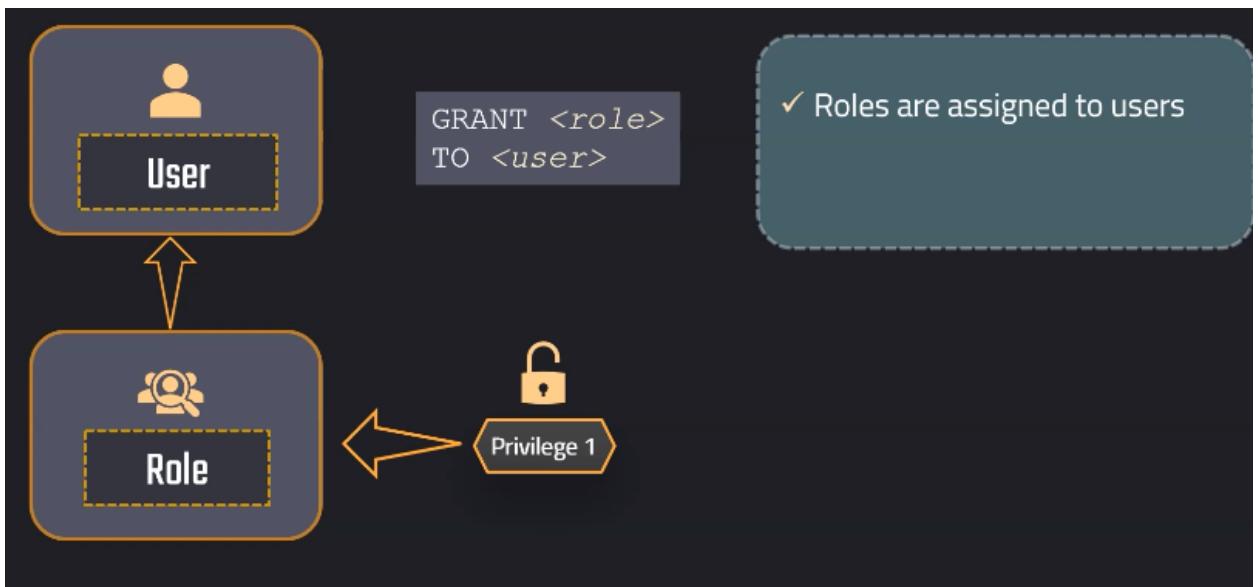
## Roles

1. This is the entity to which we are assigning privileges to.



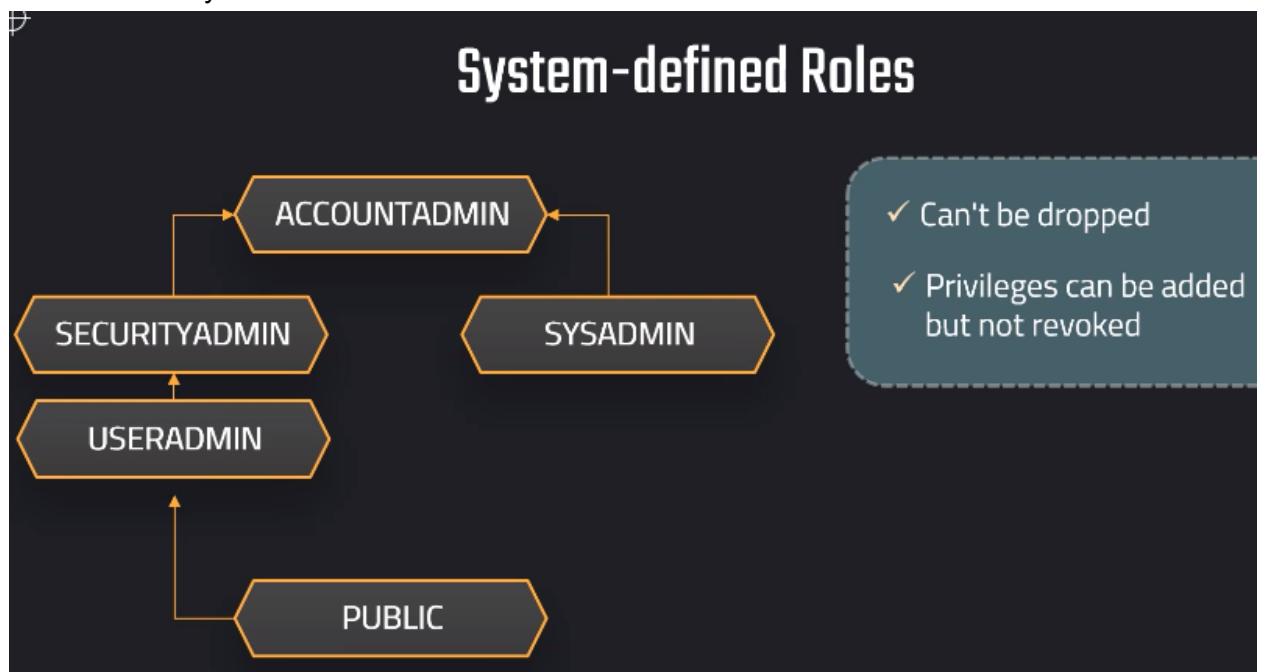
2.

3. We can grant or revoke privileges to Role
4. A privilege can be a **select** command on a table....etc
5. Now we can assign these roles to the users...so the users get this privileges



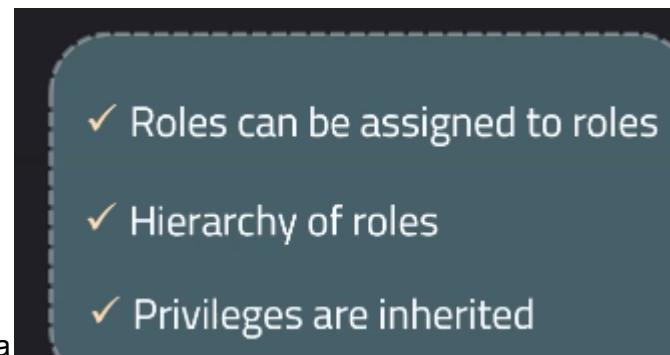
6. We always give privileges to roles...and assign these roles to the user...so the user gets privileges
7. One user can have multiple roles assigned
8. When we are using worksheets...then there must one role selected ..which is called a primary role

9. Lets come to system defined roles...



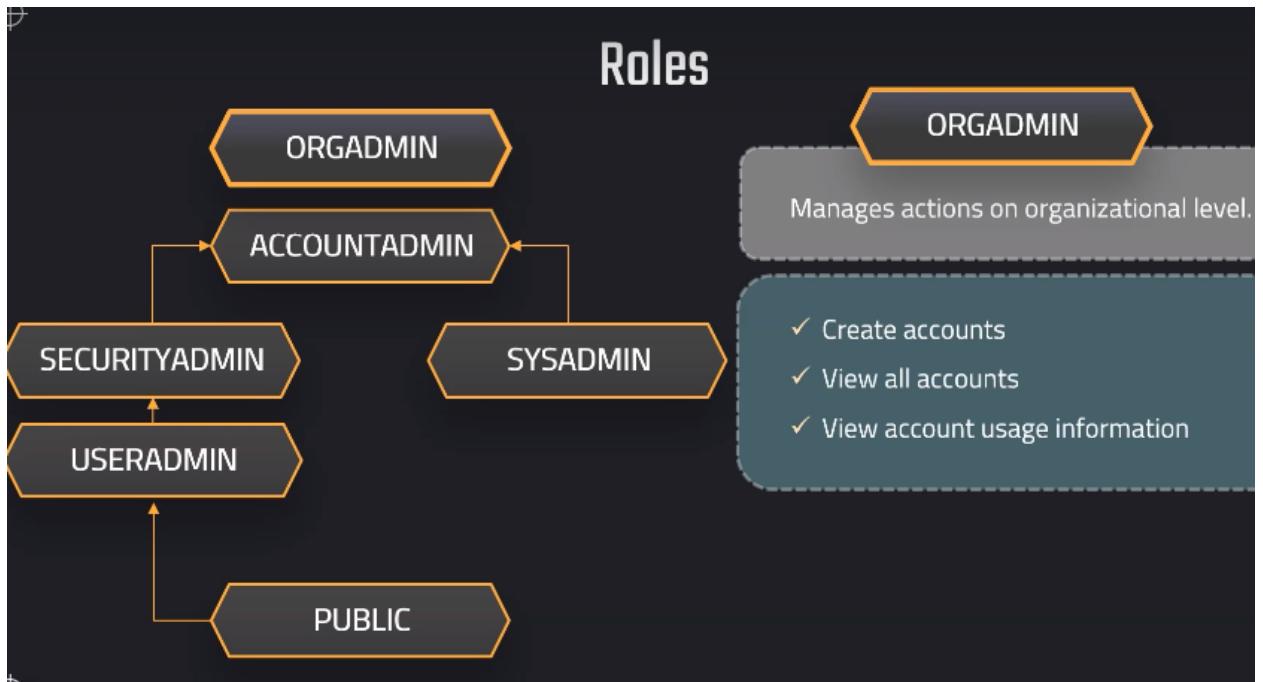
10. These roles cant be dropped..and also we give privileges ..but cannot revoke once added

11. Roles can be assigned to roles...like public role can be assigned to useradmin vice versa



12. We can also create custom roles..with required privileges in our org...and this should be assigned to sysadmin role

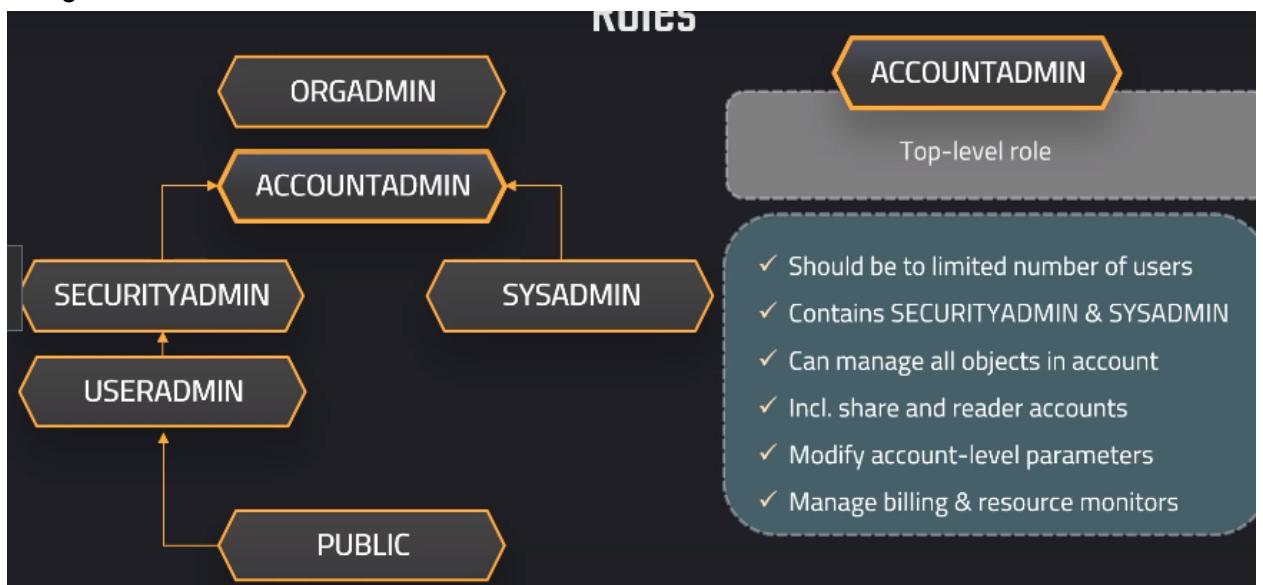
13. Outside of hierarchy ..we have special role which is ORGADMIN...it manages the system defined roles in the org



14. In the orgadmin....the top level role is ACCOUNT ADMININ....

15. Accountadmin has all privileges and can manage all objects in account....

16. Using accountadmin...we can create reader account or can create a share



17.

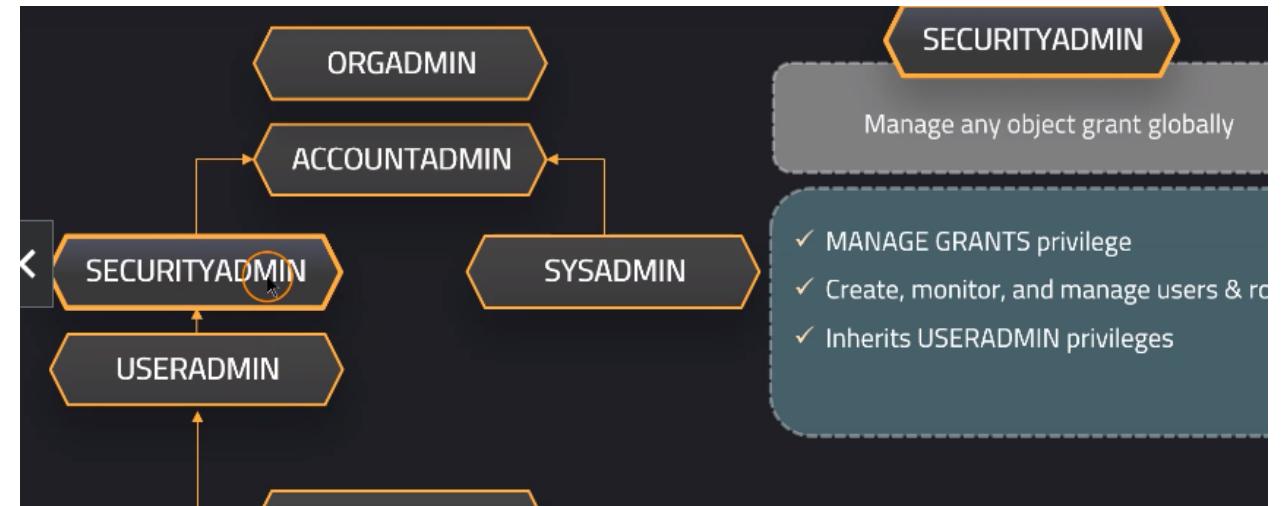
18. Coming to secutiryAdmin...it can have access to any object ,,that has granted globally(any where)

The SECURITYADMIN role in Snowflake is a system-defined role that has the most powerful privileges in the Snowflake account. It can manage any object grant globally, as well as create, monitor, and manage users and roles.

More specifically, the SECURITYADMIN role has the following privileges:

- CREATE USER
- ALTER USER
- DROP USER
- CREATE ROLE
- ALTER ROLE
- DROP ROLE
- GRANT ROLE
- REVOKE ROLE
- MANAGE GRANTS

The SECURITYADMIN role also inherits the privileges of the USERADMIN role, which include the ability to create, manage, and view users and roles.



- 19.
20. Coming to sysadmin role....it has abilities to create warehouse,databases and other objects

The SYSADMIN role in Snowflake is a system-defined role that has privileges to create warehouses, databases, and all database objects (schemas, tables, etc.), as well as grant those privileges to other roles. It is the second most powerful role in the Snowflake account, after the SECURITYADMIN role.

Here are some of the tasks that can be performed using the SYSADMIN role:

- Create and manage warehouses, databases, and other database objects
- Grant and revoke privileges on objects
- View all objects in the account
- Create, alter, and drop users and roles

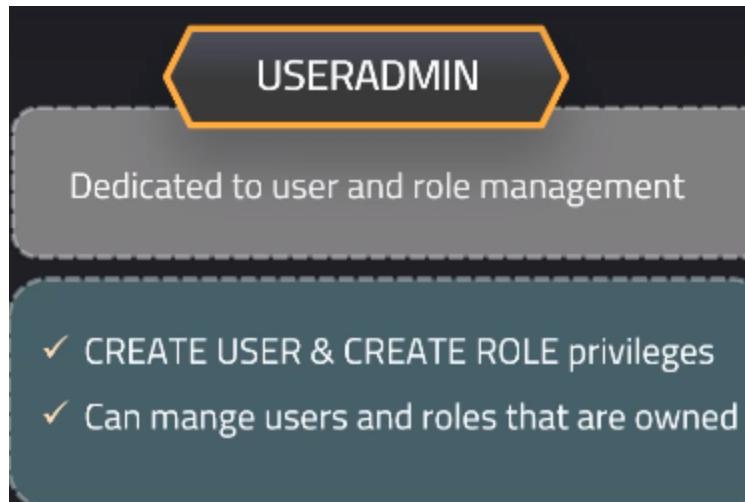
The SYSADMIN role is typically granted to a small number of users who are responsible for managing the Snowflake account and its resources. It is important to note that the SYSADMIN role can be used to grant any privilege to any user, except for the SECURITYADMIN role. Therefore, it is

## SYSADMIN

Create warehouses, databases & other objects

- ✓ All custom roles should be assigned to
- ✓ Can grant privileges on warehouses,  
databases, and other objects

21.

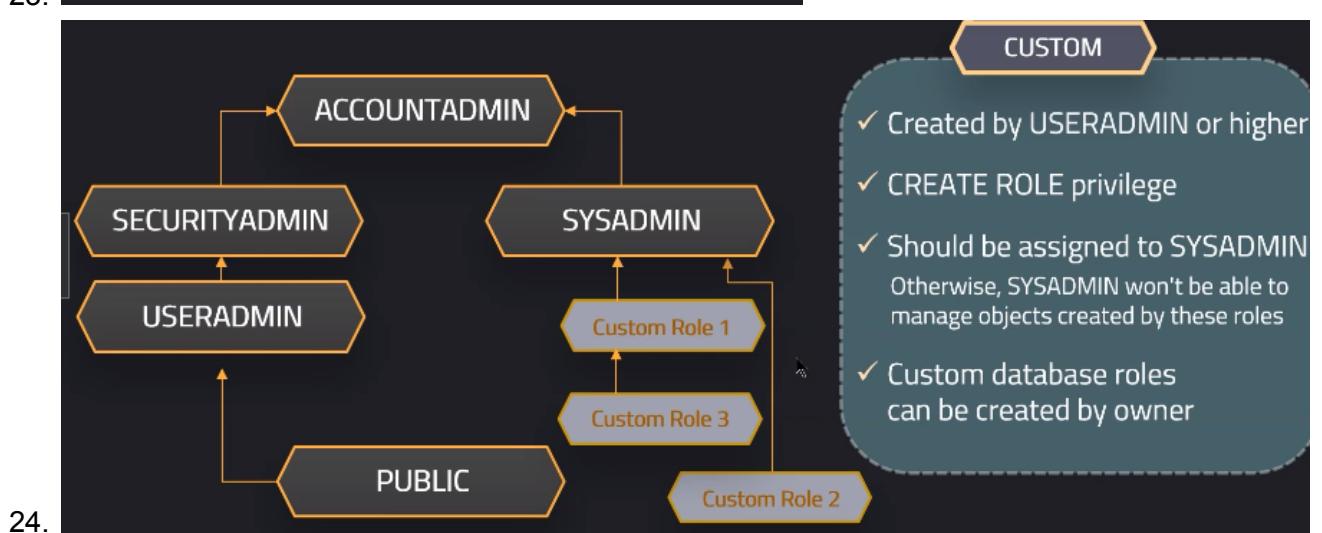
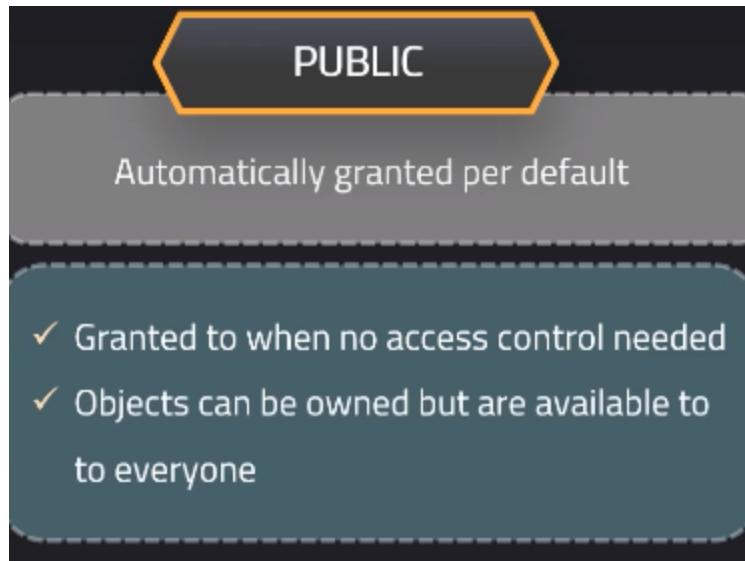


22.

The USERADMIN role in Snowflake is a system-defined role that has privileges to create and manage users and roles. It is less powerful than the SECURITYADMIN and SYSADMIN roles, but it is still a powerful role that should be used carefully.

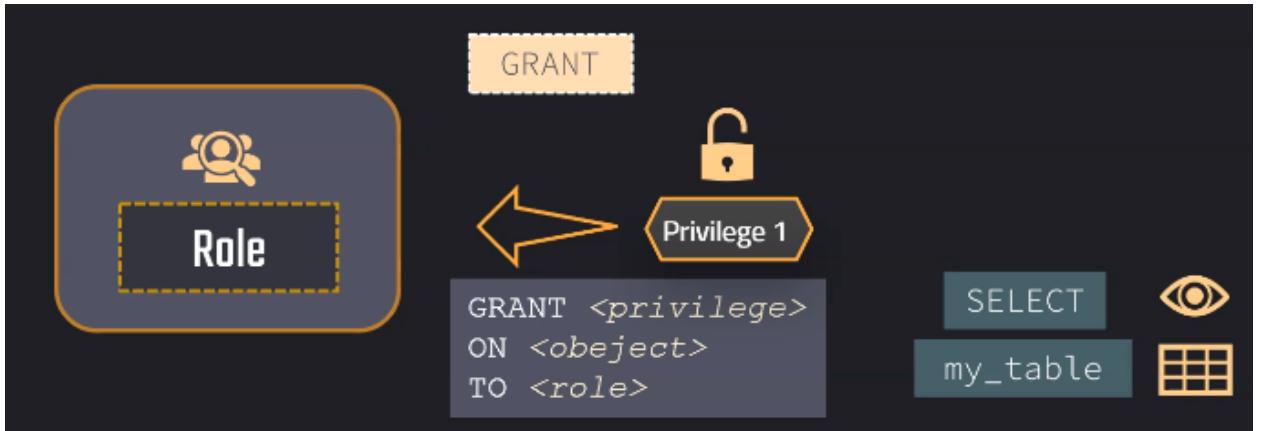
Here are some of the tasks that can be performed using the USERADMIN role:

- Create, alter, and drop users
- Create, alter, and drop roles
- Grant and revoke roles to users
- View all users and roles in the account



## Privileges

1. This is just the granular level of access to an object that will be granted to a role.  
So this is done by using the grant command.



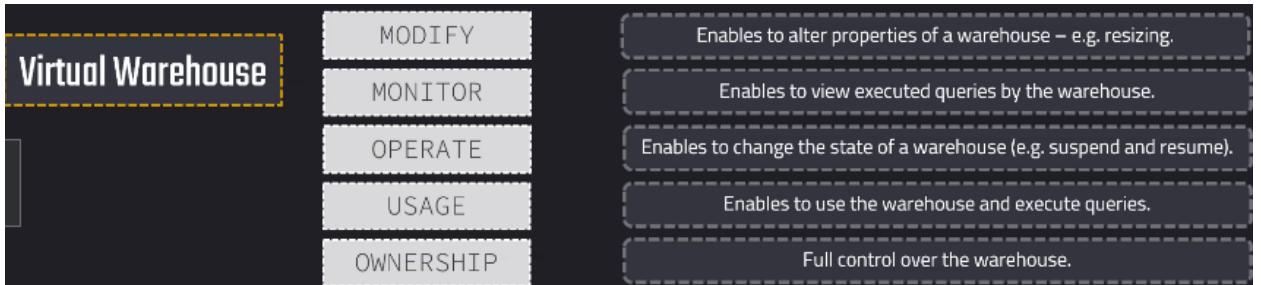
- 2.
3. So here we are granting select privilege on my\_table to role....we can also have objects like schemas etc
4. Also we can revoke a privilege .. So this will be usually possible for the ownership role. This is the role that can grant and revoke those privileges.



5. Then we have global manage grant privilege..which is handled by sysadmin role
6. Important privileges



- 7.
8. ANd for virtual warehouse we have these privil ges



- 9.
10. The ownership privilege can also be only transferred to another role because always only one single role can have the ownership of an object

11. Next we ALL

ALL	All privileges apart from OWNERSHIP.
-----	--------------------------------------

12. For databases we have these privileges

Databases	MODIFY	Enables to alter properties and settings of a database.
	MONITOR	Enables to perform DESCRIBE command.
	USAGE	Enables to use the database and execute SHOW DATABASES command.
	REFERENCE_USAGE	Enables using an object (shared secure view) to reference another object in a different database.
	ALL	All privileges apart from OWNERSHIP.
	OWNERSHIP	Full control over the database.
	CREATE SCHEMA	Enable creating a schema in the database.

13.

14. For stages we have

Stages	READ	Enables to perform operations that require reading (e.g. GET, LIST, COPY INTO table) from internal stages; not applicable to external stages
	USAGE	Enables to use an external stage; not applicable to internal stage.
	WRITE	Enables to perform writing to internal stage (PUT, REMOVE, COPY INTO location); not applicable to external stages
	ALL	All privileges apart from OWNERSHIP.
	OWNERSHIP	Full control over the stage.

Tables	SELECT	Using SELECT to query table.
	INSERT	Inserting values into the table and manually reclustering tables.
	UPDATE	Using UPDATE command on a table.
	TRUNCATE	Using TRUNCATE command on a table.
	DELETE	Using DELETE command on a table.
	ALL	All privileges apart from OWNERSHIP.
	OWNERSHIP	Full control over the database.

15.

16. For hands on refer practice

Multi-Factor Authentication

Authentication is proving that you are who you say you are.

Multi-Factor Authentication provides additional login security.

Standard Edition

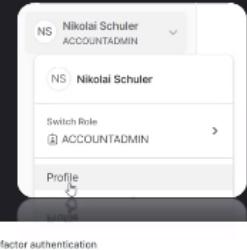
Powered by Duo Security but managed by Snowflake.

No sign-up,  
only installation.

- 1.
2. Starting from standard edition it is available to all edition

Per default enabled for accounts but requires user to [enroll](#).

Strongly recommended for  
ACCOUNTADMIN

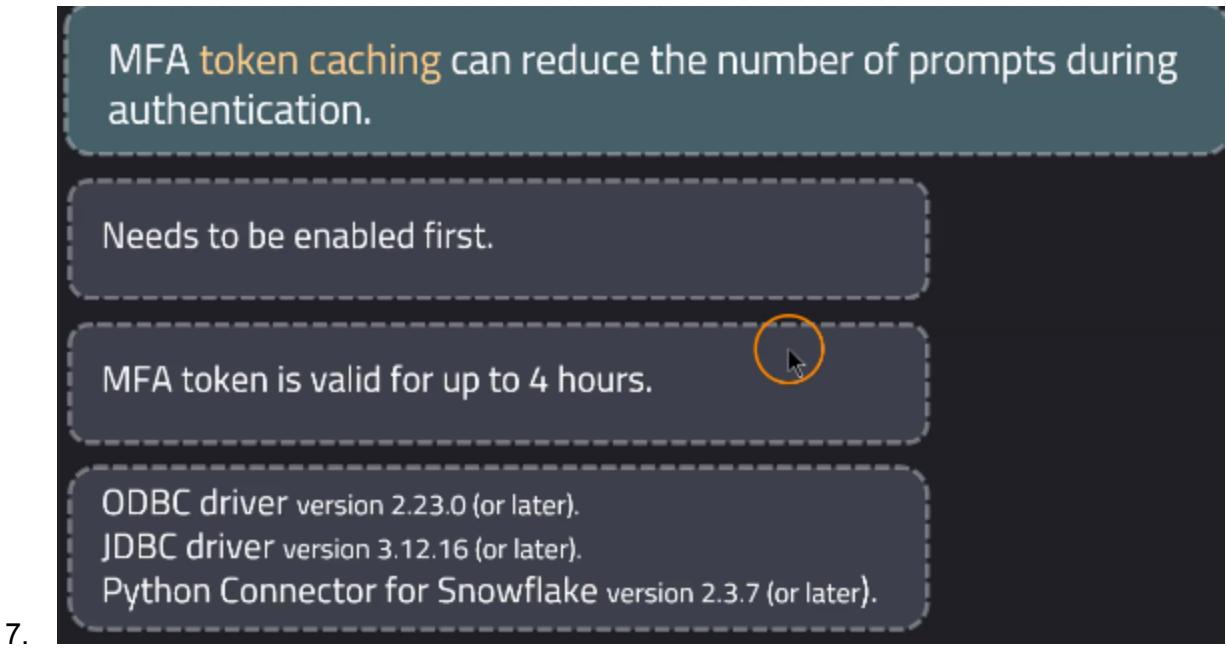


- 3.

SECURITYADMIN (or ACCOUNTADMIN) can disable MFA for user.

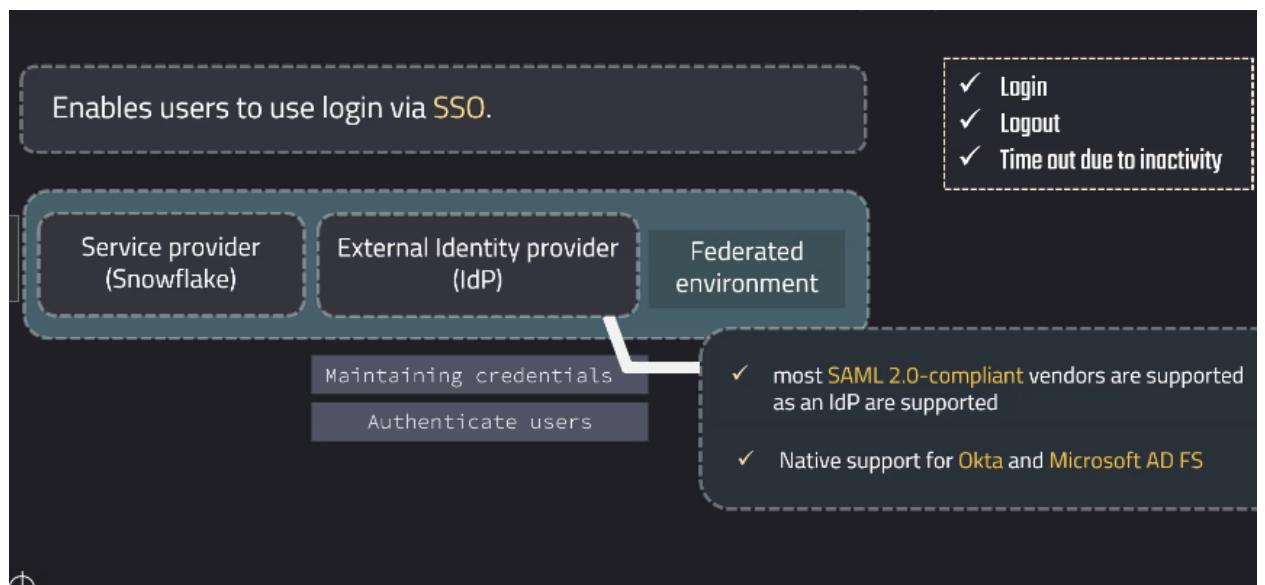
Fully-supported by web interface, SnowSQL, Snowflake ODBC and JDBC, and Python Connectors.

- 4.
5. Additionally, we can also enable MFA token caching.
6. This can reduce the number of prompts during authentication, especially if we are trying to authenticate in a short amount of period multiple times.



## Federated Authentication

- This means that they can just use one set of credentials and use this to log into Snowflake and also other applications.



Federated authentication in Snowflake allows users to log in to Snowflake using their credentials from an external identity provider (IdP). This can simplify the user login process and improve security, as users do not need to manage multiple passwords.

To enable federated authentication in Snowflake, you must create a security integration with your IdP. This involves providing Snowflake with the necessary information about your IdP, such as the IdP's URL and certificate.

Once you have created a security integration, you can configure Snowflake to use federated authentication for all users or for a specific set of users.

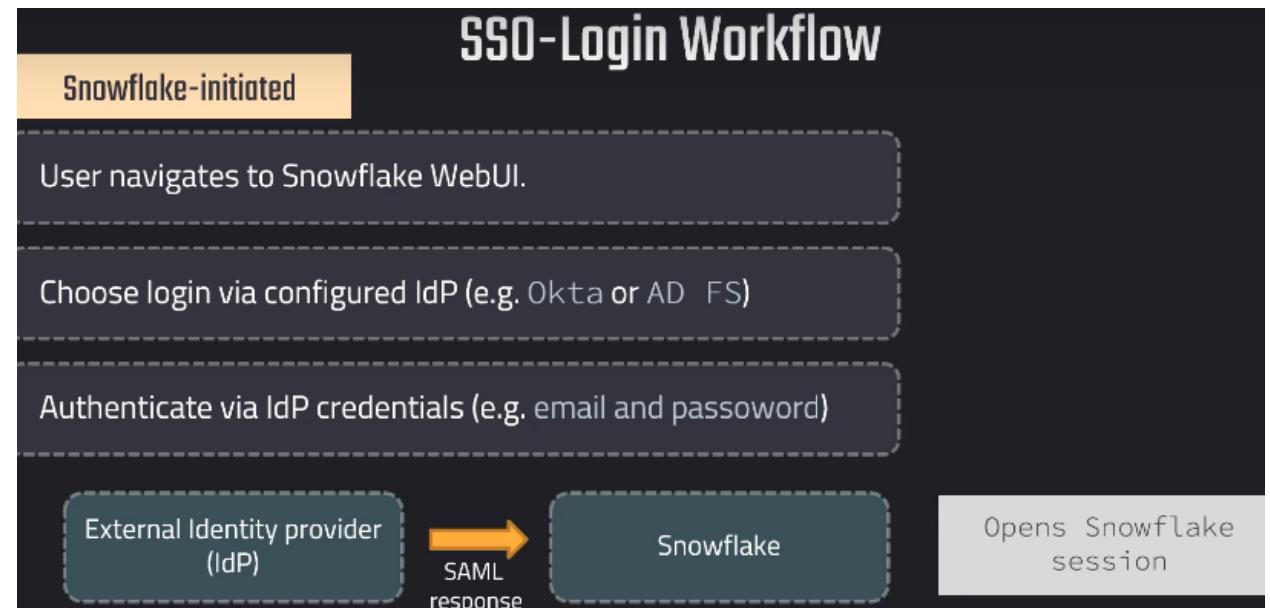
To log in to Snowflake using federated authentication, users simply need to visit the Snowflake login page and select the "Login with IdP" option. They will then be redirected to the IdP's login page, where they can enter their IdP credentials.

2.

Here is an example of how to use federated authentication in Snowflake with Azure Active Directory (AD):

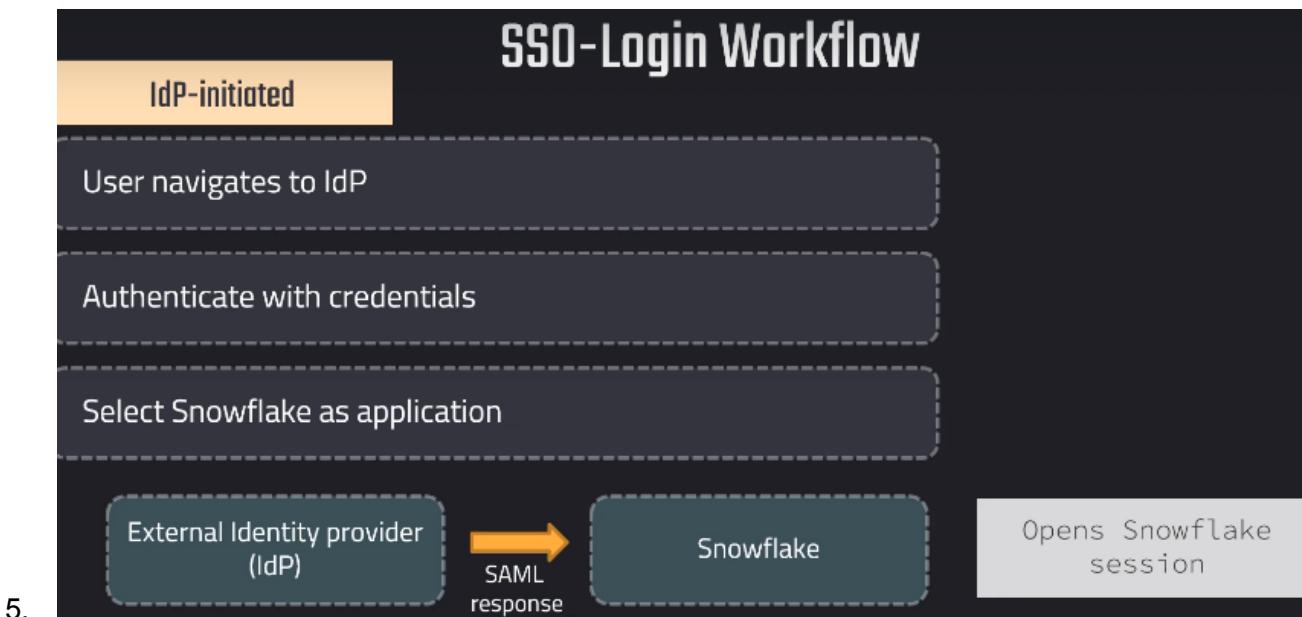
1. Create a security integration in Snowflake for Azure AD.
2. Configure Snowflake to use federated authentication for all users.
3. Users visit the Snowflake login page and select the "Login with IdP" option.
4. Users are redirected to the Azure AD login page, where they enter their Azure AD credentials.
5. Once the user has successfully authenticated with Azure AD, they are redirected back to Snowflake and logged in.

3.

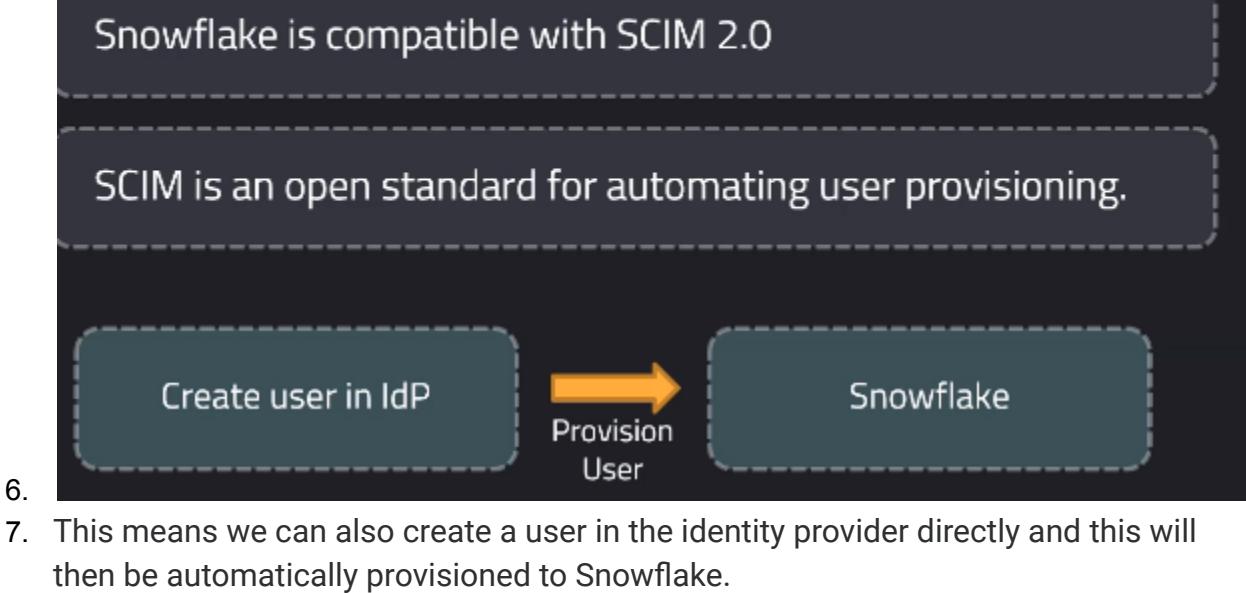


4.

## SSO-Login Workflow



## SCIM support

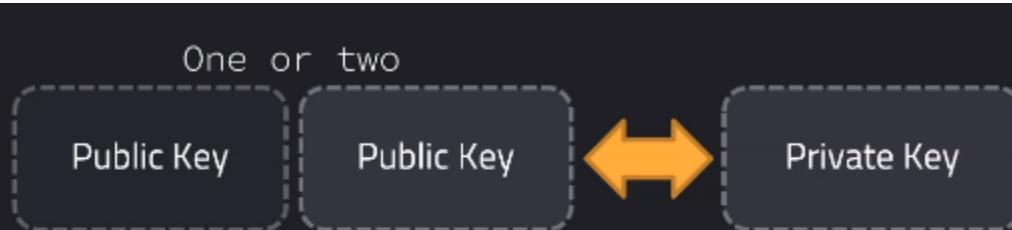


Key-pair Authentication

# Key Pair Authentication

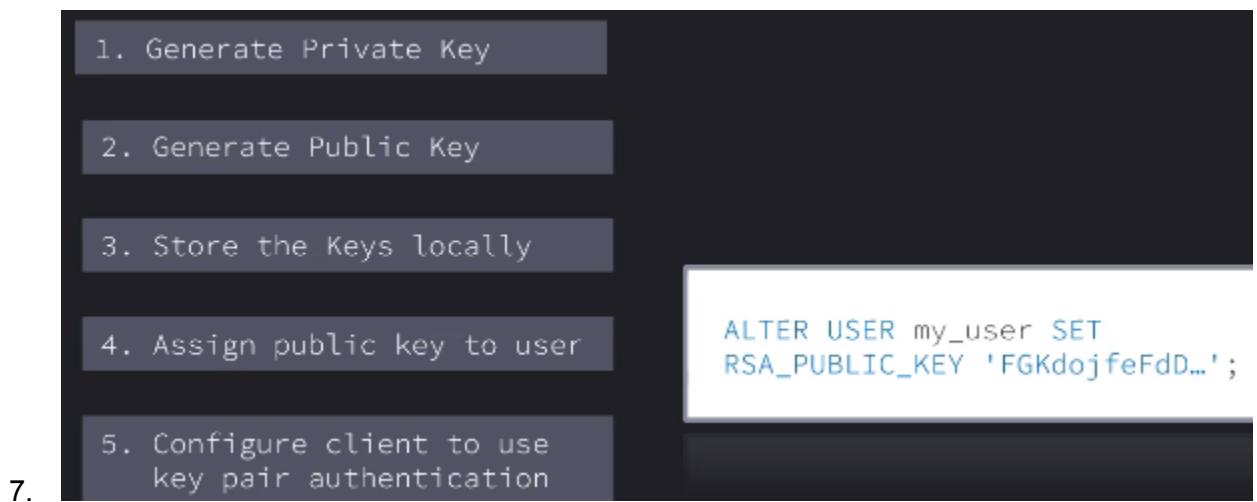
Enhanced security as an alternative to basic username/password.

- 1.
2. Here we will have one or two public key and one private key(pair)



Minimum:  
2048-bit RSA  
key pair

- 3.
4. Min key size is 2048bit
5. This can be used when connecting with the snowflake clients ...like snowSQL etc
6. Overview



## Column Level Security

1. We can assign some security policies on columns and it supports enterprise edition or higher

### Column-level Security

Column-level security masks data in tables and views enforced on columns

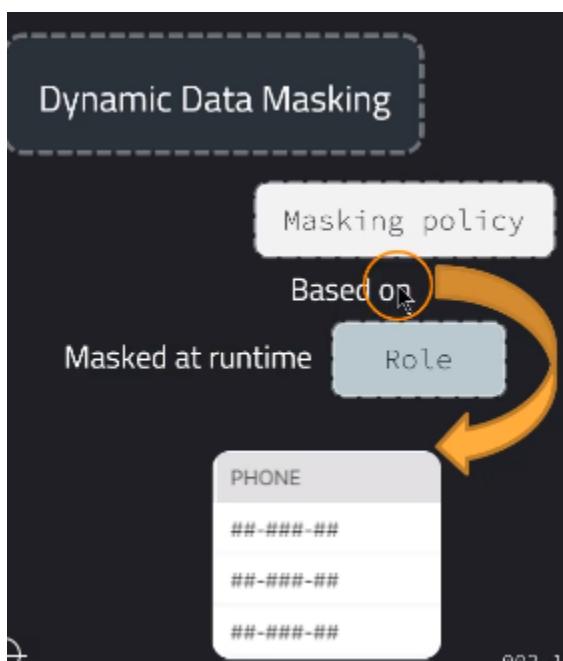
Enterprise Edition

FULL_NAME	EMAIL	PHONE
Lewis MacDwyer	lmacdwyer0@un.org	262-665-9168
Ty Pettingall	tpettingall1@mayoclinic.com	734-987-7120
Marlee Spadazzi	mspadazzi2@txnews.com	867-946-3659
Isabella Schaefer	mspa...@txnews.com	801-810-3020

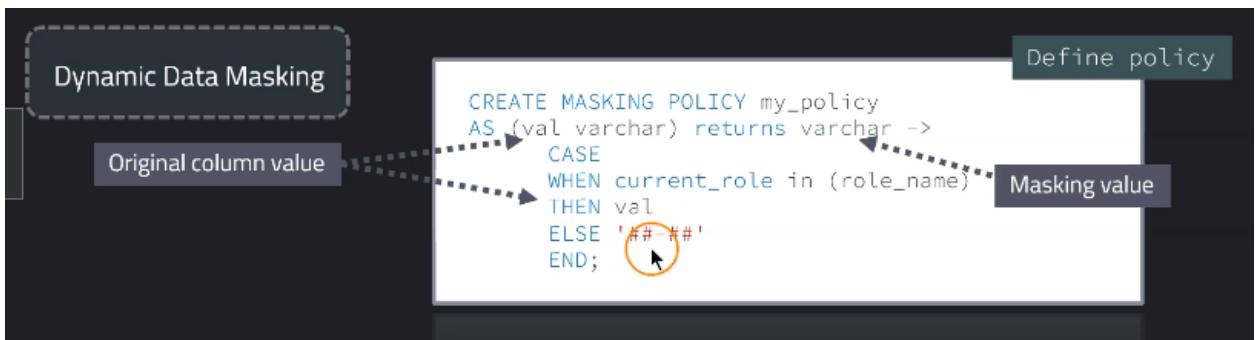
  

FULL_NAME	EMAIL	PHONE
L*****	l*****	##-###-##
T*****	t*****	##-###-##
M*****	m*****	##-###-##
W*****	w*****	##-###-##

- 2.
3. We can mask the sensitive data on columns
4. So the data is not masked in the database itself, but just based on the currently selected role at query runtime, this is when the data will be masked and then the data will not be revealed to unauthorized users.



## 5. Syntax



6.

7. Here we are masking the columns to specific role('when current\_role)..

8. Apply policy

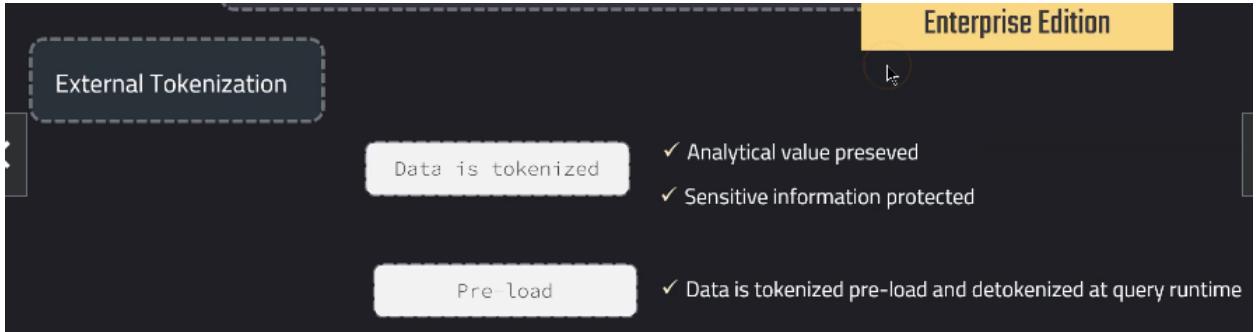
9.

10. We can also unset this by using unset

11. External tokenization

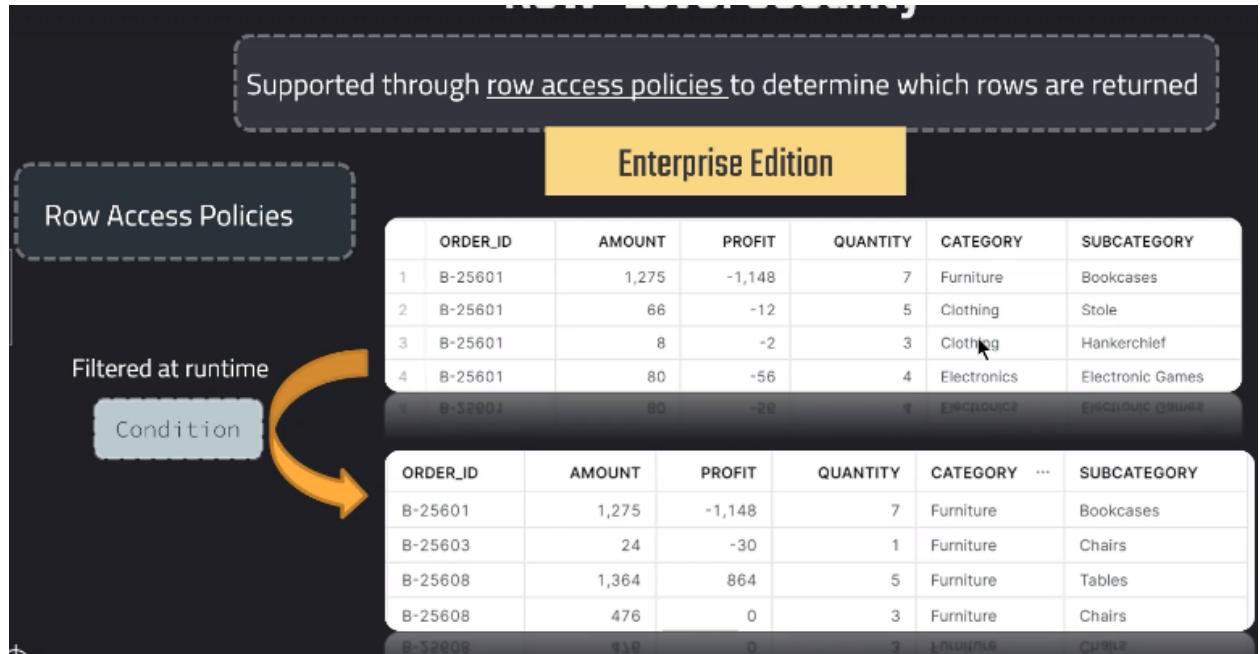
External tokenization in Snowflake is a security feature that allows you to tokenize sensitive data before loading it into Snowflake and detokenize it at query runtime. This helps to protect your sensitive data from unauthorized access, even if the Snowflake database is compromised.

12.

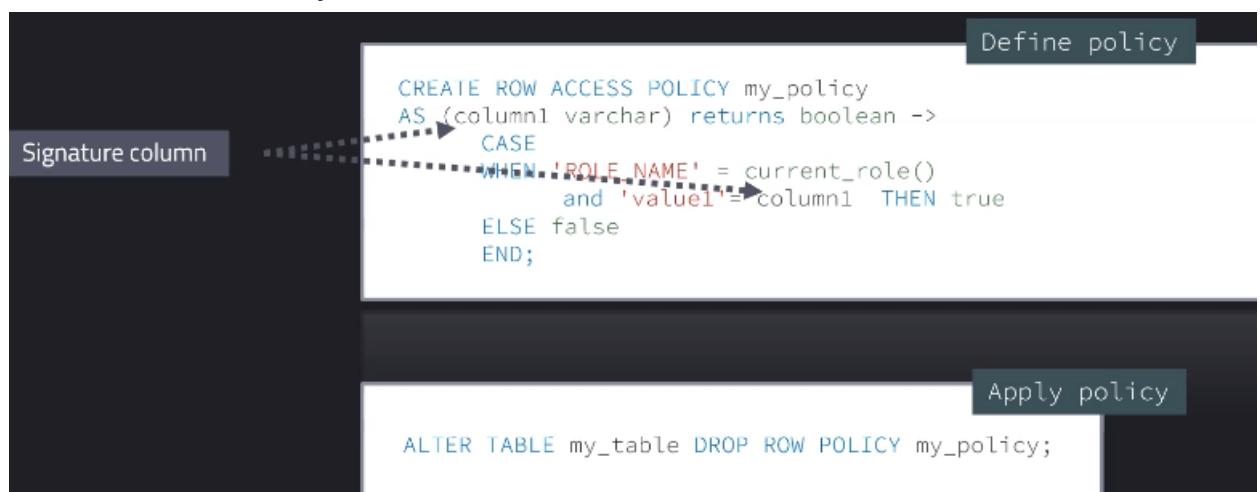


## Row-level security

1. By using row Access policies..we can provide security to particular rows



- 2.
3. How to setup
4. It is a schema level object



- 5.
6. Row access policies in Snowflake are a way to control which rows in a table or view can be accessed by a given user. They can be used to implement row-level security, which allows you to grant users access to specific rows of data based on their roles, permissions, and other criteria.

```
-- Policy that role can see everything
USE ROLE accountadmin;
CREATE OR REPLACE ROW ACCESS POLICY category_policy
AS (category varchar) RETURNS BOOLEAN ->
CASE WHEN 'HOME_MANAGER' = current_role() and 'Furniture'=category then true
else false
end;
```

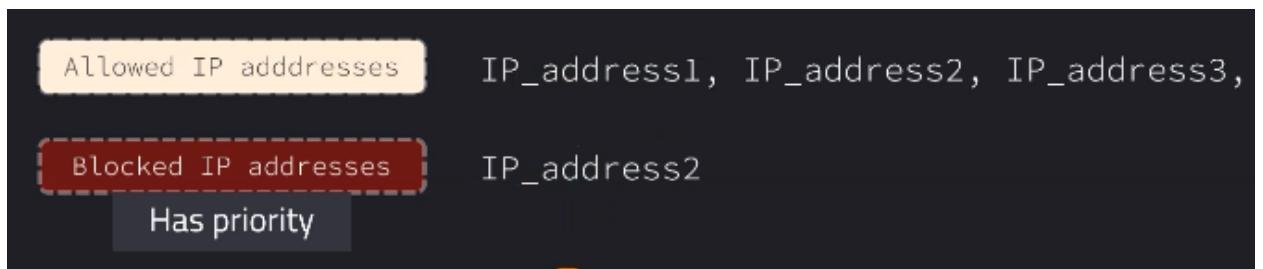
7.



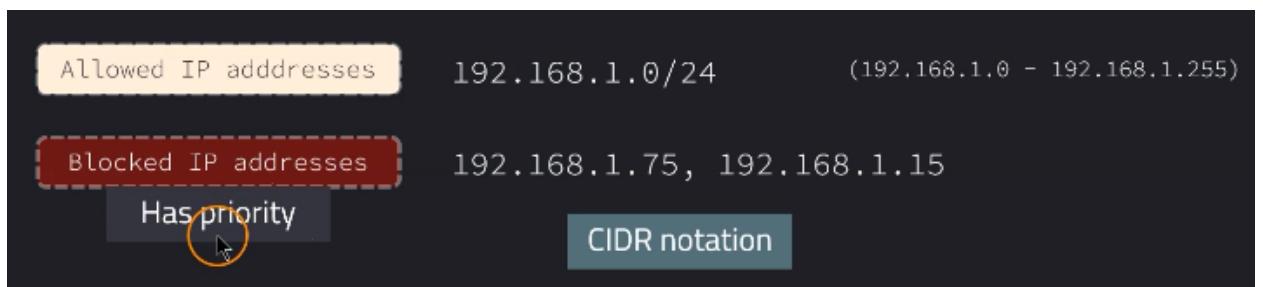
8. For hands on refer file

## Network policies

1. Here we can define..who has access and to whom we restrict access to
2. It maintains two lists



3. If an ip address is present in two lists..then blocked list will be given priority
4. We can have a range of ip address that are allowed



5. Here we have a range 0-255...and two ip address are present in both Allowed and blocked...blocked will get priority
6. We need to have security admin role...to create network policies



7.

Create Network Policy

SECURITYADMIN

global CREATE NETWORK POLICY privilege

```
CREATE NETWORK POLICY my_network_policy  
ALLOWED_IP_LIST = ('192.168.1.95', '192.168.1.113'),  
BLOCKED_IP_LIST = ('192.168.1.95');
```

- 8.
9. To set policies to an account

```
ALTER ACCOUNT SET NETWORK_POLICY = mynetwork_policy;
```

ALTER ACCOUNT UNSET NETWORK\_POLICY;

10. To unset we use
11. If we have ownership role...then we can set policies to the user

## Data Encryption

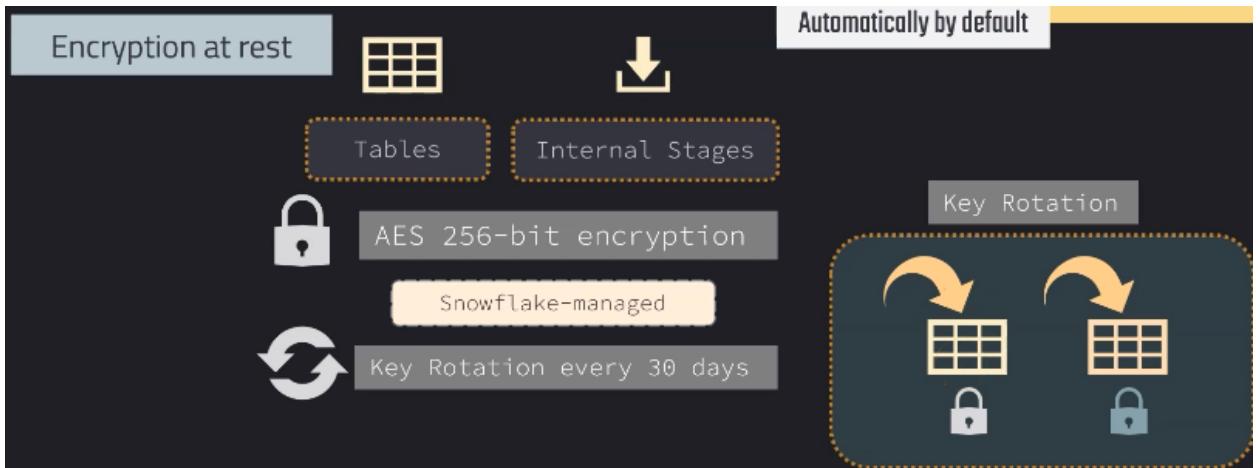
# Data Encryption

All data is encrypted at rest and in transit

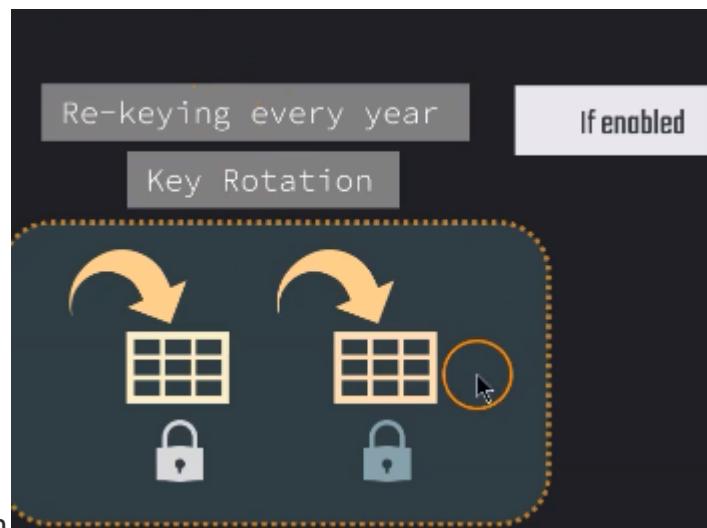
Standard Edition

Automatically by default

- 1.
2. For the data which is at rest...like tables and files in internal stages...snowflake uses 256bit encryption

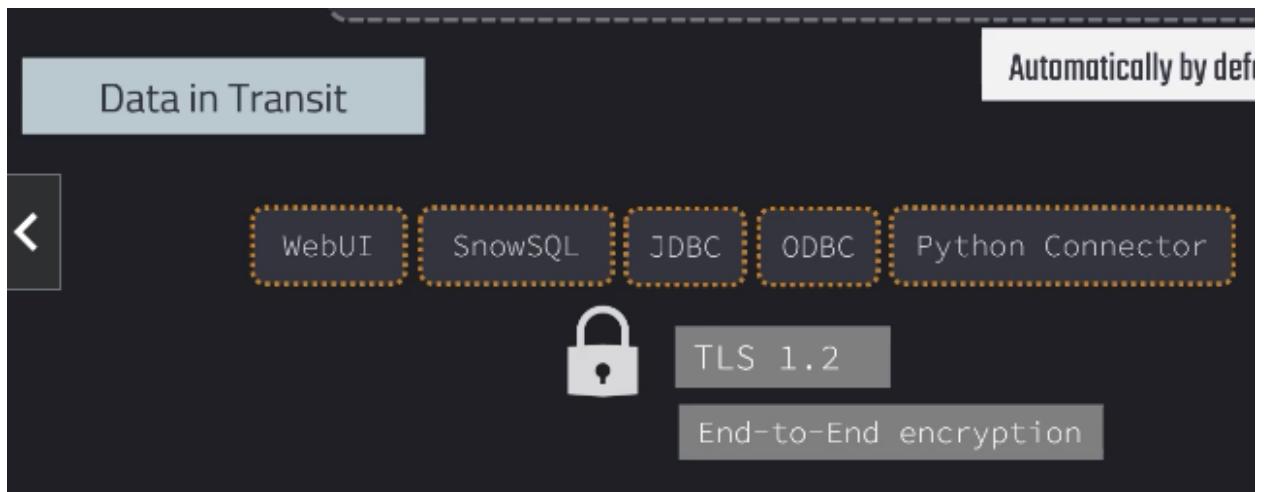


- 3.
4. When we create a table...snowflake generates a key for the table...till 30 days it uses one key
5. And after 30 days ..when we create a new table...it generates new key....and old tables..will have old keys
6. So if we have no more tables where a certain key is used, this key will be then destroyed
7. We can also add re-keying every year....which re-keys...every old key..it needs



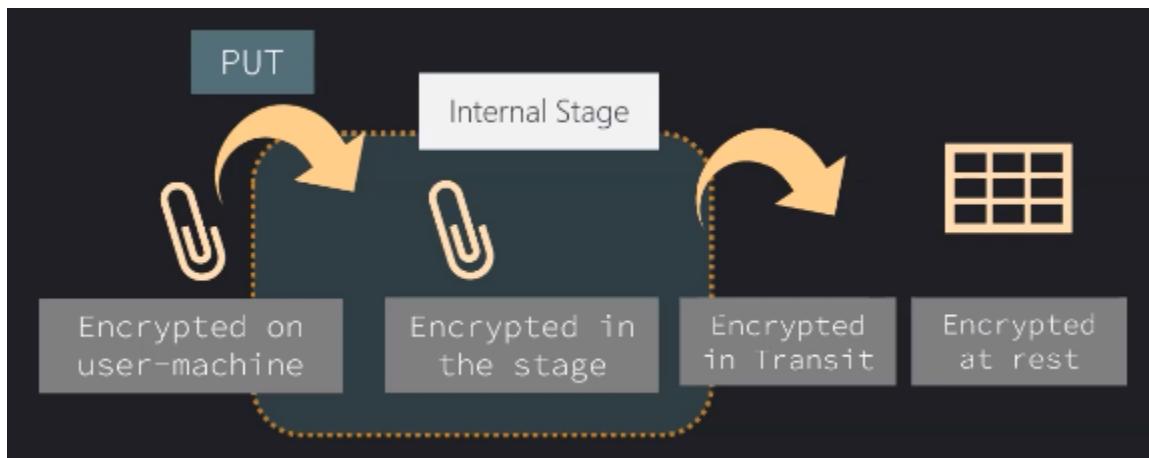
8. ENcryption for data in transit

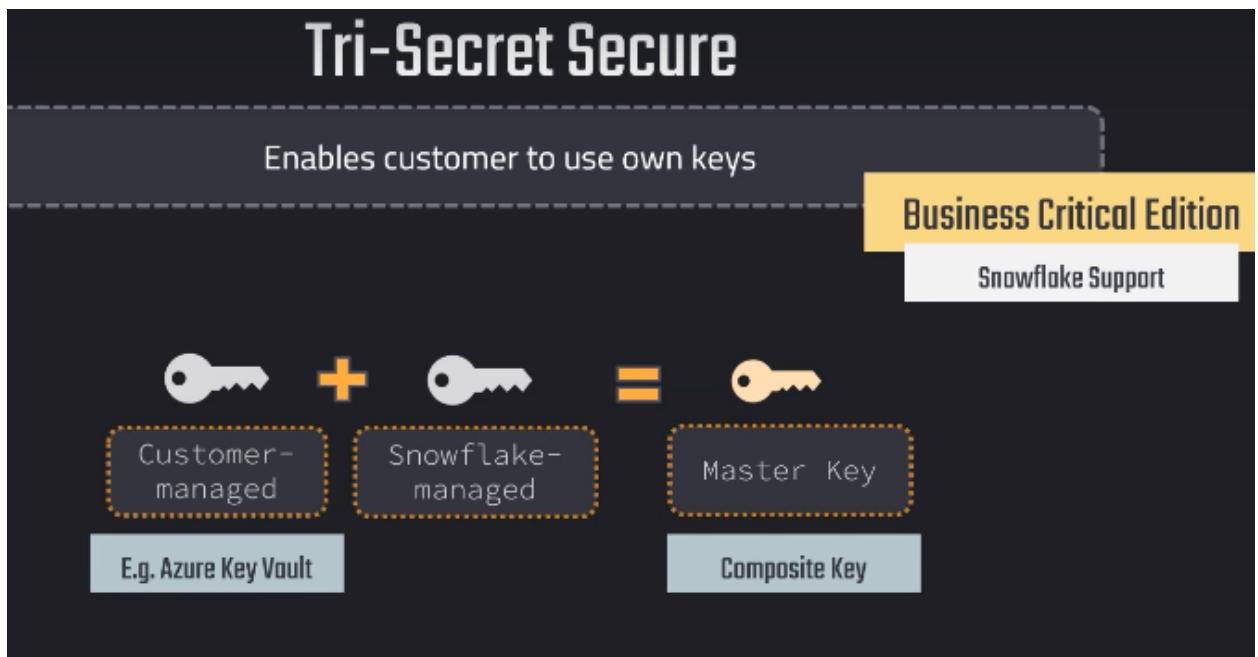
9. Snowflake uses TLS1.2 encryption for data at transit



10. For example...if put a file in a stage..then it is encrypted on user-machine and as well as by 256bit encryption(for data at rest)

11. When we move this data to table..then it will be encrypted by TLS1.2...and after going into table...it will be in rest



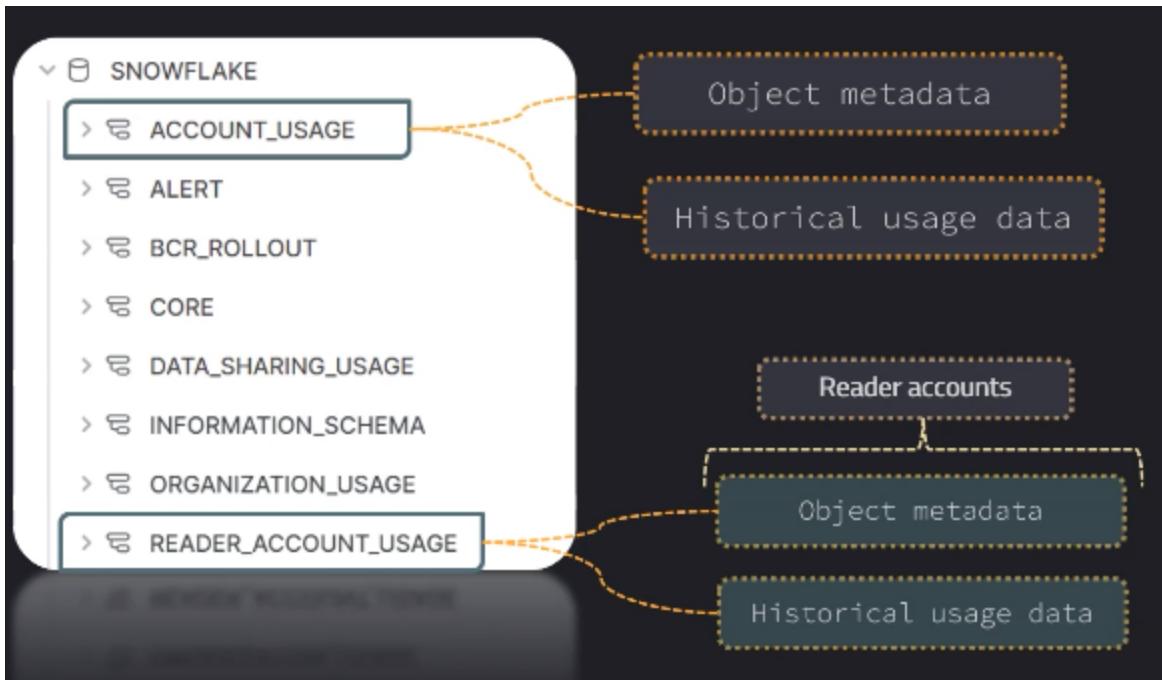


12.

13. For Tri secret secure we have contact snowflake support(refer bard for more info)

#### Account usage and information Schema

1. Account usage schema that's available in the Snowflake database, which is a shared database... that's per default available to all of the accounts.
2. Account admin can see everything in this schema and can view all data in schema



- 3.
4. Account usage schema gives details about object metadata and historical usage data
5. Then we have reader account usage..see pic
6. Information schema is available in all the databases

## 7. Lets look into account usage schema

The screenshot shows the Snowflake UI with the path `SNOWFLAKE > ACCOUNT_USAGE > Views`. The `COLUMNS` view is selected, indicated by a dashed orange arrow pointing to a callout box labeled "Long-term historical usage data". Another dashed orange arrow points from the `COPY_HISTORY` and `DATABASES` views to another callout box labeled "Object metadata". To the right of the UI are two tables:

COLUMN_ID	COLUMN_NAME	TABLE_ID	TABLE_NAME
1	16390	JOB	16386 TEST
2	18481	FIRST_NAME	18444 SEQUENCE_TEST
3	2061	ORDER_ID	2054 ORDERS

FILE_NAME	STAGE_LOCATION
1 /OrderDetails_error.csv	s3://bucketsnowflakes4
2 Orders2.csv	s3://snowflakebucket-copyoption/returnfailed/
3 OrderDetails_error2 - Copy.csv	s3://snowflakebucket-copyoption/returnfailed/

The screenshot shows the Snowflake UI with the path `SNOWFLAKE > READER_ACCOUNT_USAGE > Views`. A dashed orange arrow points from the `WAREHOUSE_METERING...` view to a callout box labeled "Reader accounts".

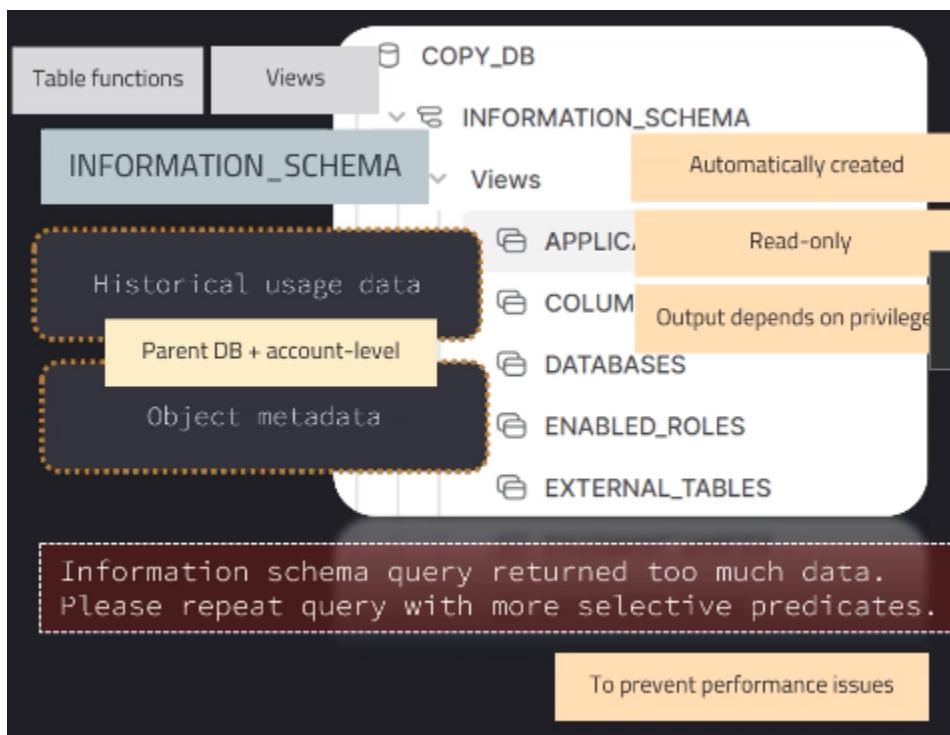
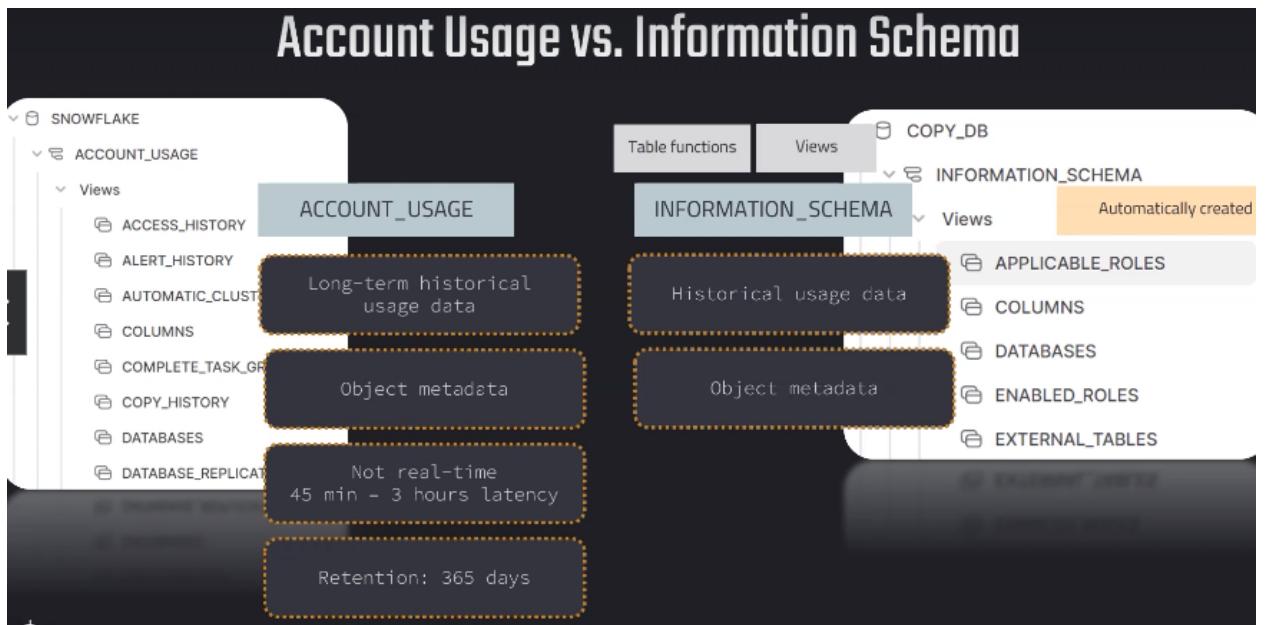
8. Most of the views have a latency of around two hours and also the data retention in here is, as mentioned, long term.

Data provided is not real-time  
45 min - 3 hours latency

Retention: 365 days

- 9.

## 10. Account vs information\_schema



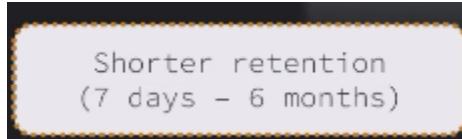
11.

The account usage schema provides information about the usage of your Snowflake account, such as the number of queries executed, the amount of data processed, and the resources consumed. This information can be used to optimize your Snowflake account performance and to track your Snowflake usage costs.

The information schema provides information about the metadata of your Snowflake account, such as the tables, views, and stored procedures that exist in your account. This information can be used to manage and query your Snowflake database objects.

12.

13. The main diff is in information schema..we have shorter retention



14. 2nd diff is Theres no latency in information schema ..wer as in account we have 45min-3hrs latency

15. 3rd diff is ...Account usage schem includes dropped table...

...	DELETED
	null
	2023-03-30 01:15:06.191 -0700

16. Information schema does not include dropped objects

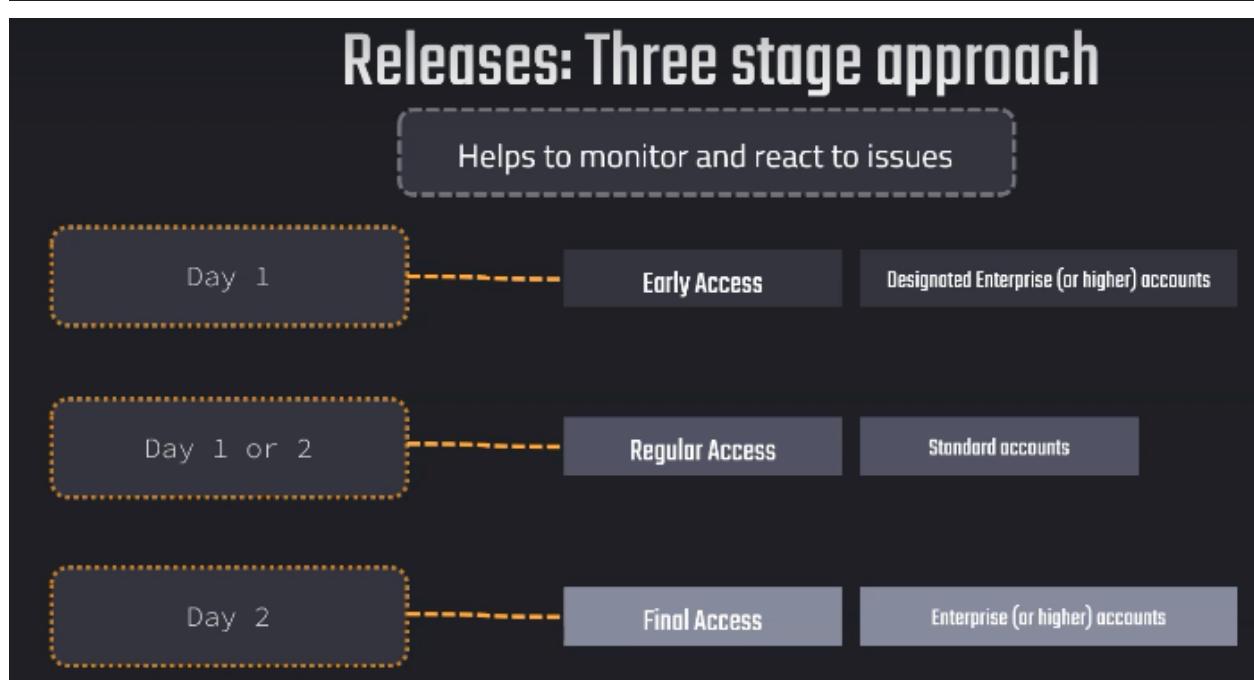
Releases

Snowflake releases new features and functionality on a monthly basis. These releases are typically announced in advance and are made available to all customers on a specific date.

1. Each Snowflake release includes a variety of new features and enhancements, such as new SQL functions, new database objects, and new performance and security features.



- 2.



- 3.

4. For early access we have to request snowflake support...if we dont have early access then we will get on final access