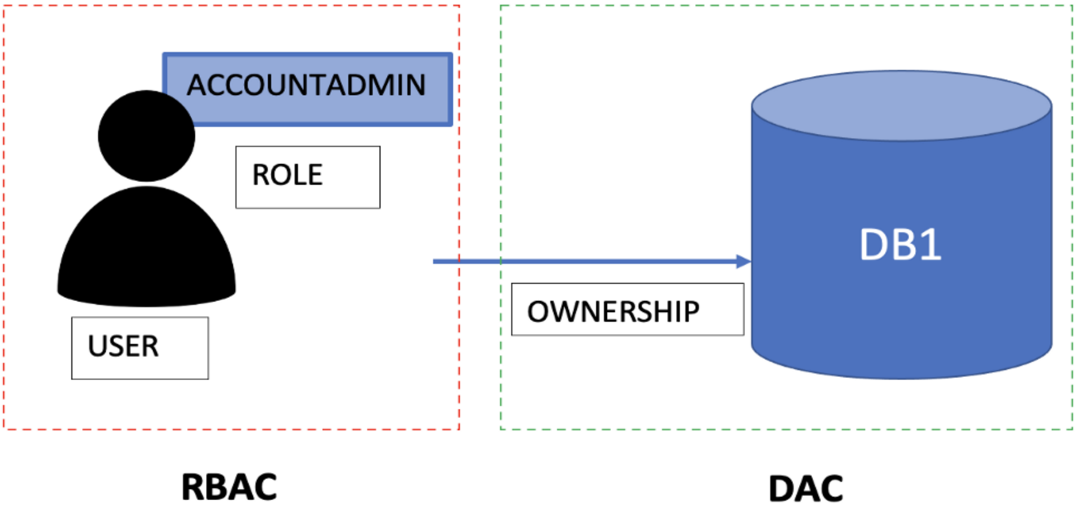
Chapter 14. Roles in SnowFlake

Access management is one of the most important security-related features in SnowFlake. Depending on the role we give to the users, we can control exactly which functions they can execute. Let's take a closer look!

ACCESS MANAGEMENT KEY CONCEPTS

Access control privileges determine who can access and perform operations on specific objects in SnowFlake. SnowFlake approach combines aspects from the following models:

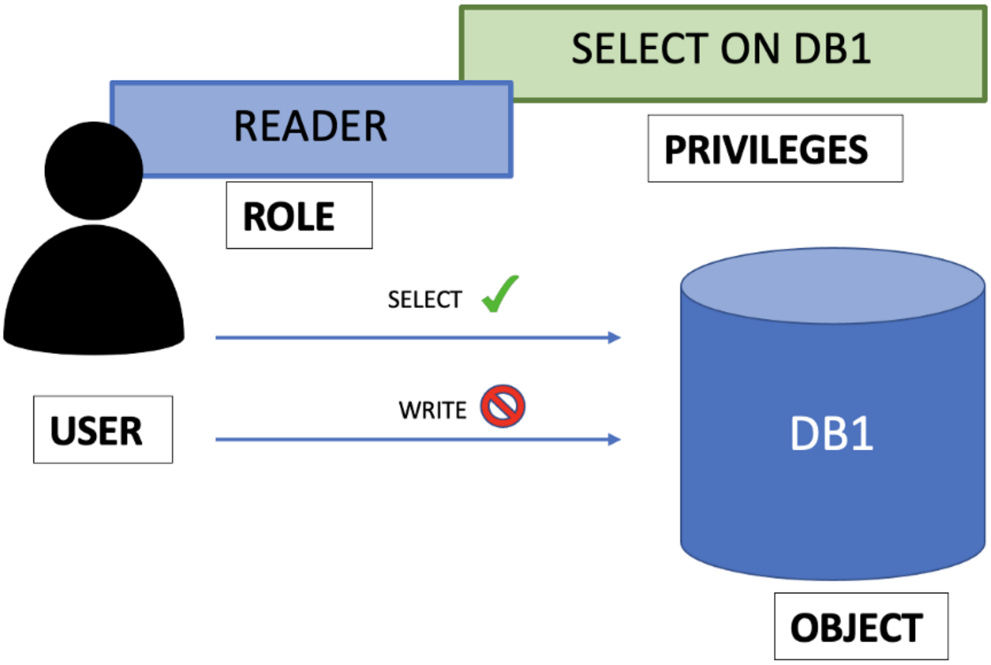
- **Discretionary Access Control (DAC)** → Each object has an owner, who can in turn grant access to that object.
- **Role-Based Access Control (RBAC)** → Access privileges are assigned to roles, which are in turn given to users.



RBAC vs. DAC

The key concepts about Access Management in SnowFlake are the following ones (we need to have this part clear for the exam, my best advice is to practice a little bit with SnowFlake to make sure we understand them):

- **User** → A person or a program. For example, Gonzalo as a user.
- **Role** → An entity to which we can grant privileges. We can grant roles to users. Roles are account-level objects, and they can also be granted to other roles, creating a hierarchy of roles. The privileges associated with a role are inherited by any roles above that role in the hierarchy. For example, Gonzalo will be associated with the AccountAdmin role.
- **Securable object** → An entity to which we can grant access. Unless allowed by a privilege, access will be denied. For example, a table, database, or schema are securable objects.
- **Privilege** → A defined level of access to an object. For example, the user Gonzalo, as AccountAdmin, will have the SELECT privilege on database "DB1".



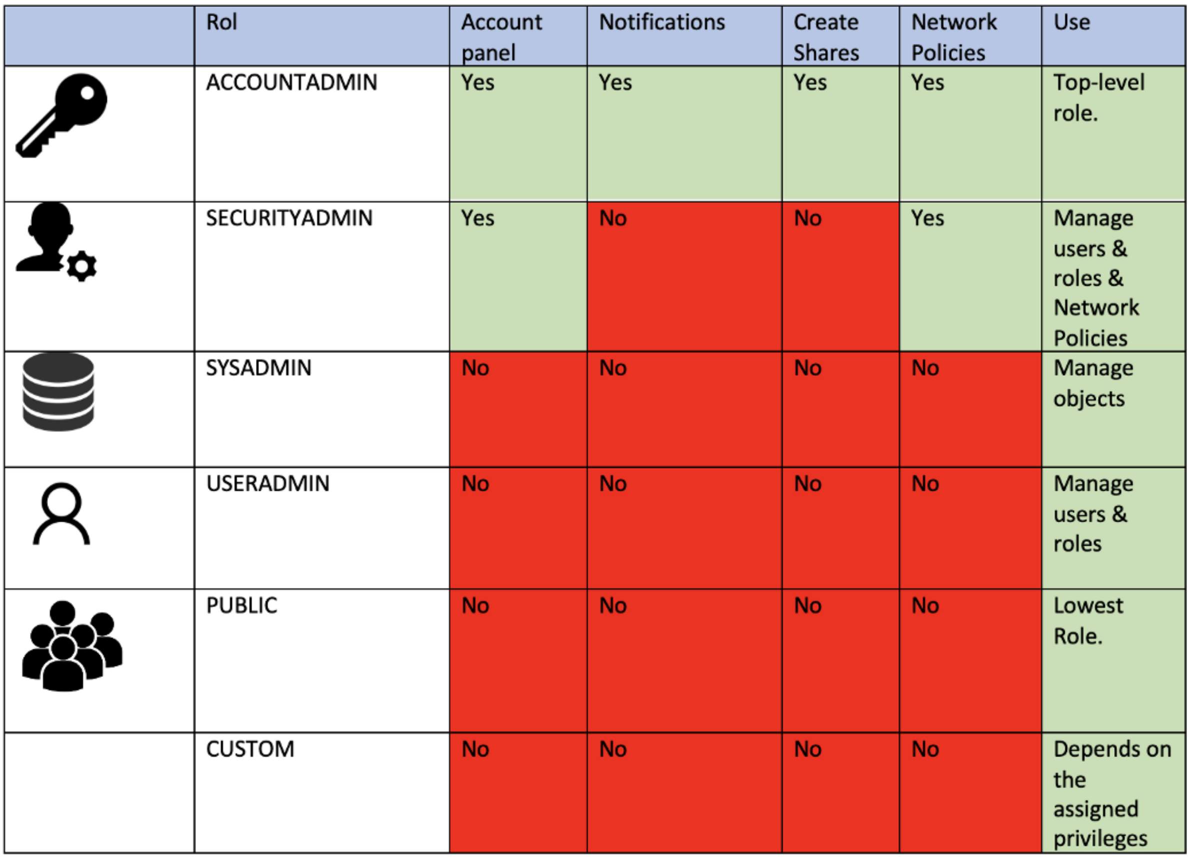
Access Management in SnowFlake

As we can see in the previous picture, a user will have a role that will contain privileges. As the select privilege is assigned on the role, we can perform select queries on the database, but we cannot write it as this privilege is not assigned to the role.

It's really important to remember that privileges are assigned to Roles, which will be assigned to users. USER <- ROLE <- PRIVILEGE.

DEFAULT ROLES

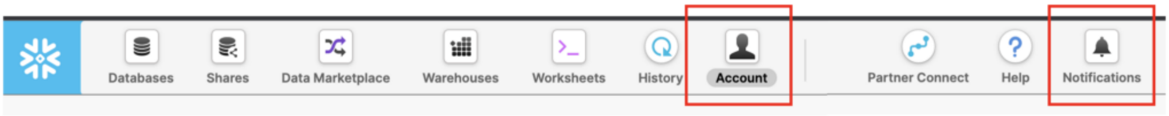
SnowFlake gives five different default roles plus the possibility to create custom ones. In this table, we can see some differences between them, and below that, we will see all of them in detail:



	Rol	Account panel	Notifications	Create Shares	Network Policies	Use
🔑	ACCOUNTADMIN	Yes	Yes	Yes	Yes	Top-level role.
👤⚙	SECURITYADMIN	Yes	No	No	Yes	Manage users & roles & Network Policies
🛢	SYSADMIN	No	No	No	No	Manage objects
👤	USERADMIN	No	No	No	No	Manage users & roles
👥	PUBLIC	No	No	No	No	Lowest Role.
	CUSTOM	No	No	No	No	Depends on the assigned privileges

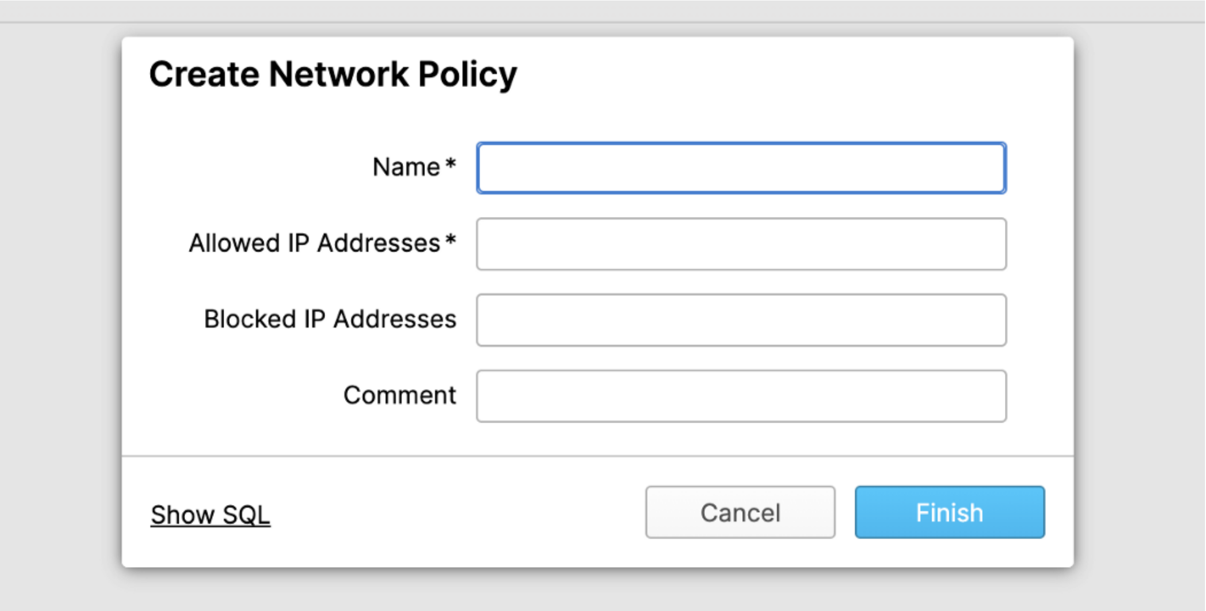
Default Roles in SnowFlake.

ACCOUNTADMIN → It encapsulates the SYSADMIN and SECURITYADMIN system-defined roles. It's the top-level role, and you shouldn't give this role to many users. AccountAdmin users can access the Account and Notification sections in the SnowFlake UI. They can also CREATE, ALTER, or DROP Network Policies and Shares (we will see this concept soon).



Account & Notification sections in the SnowFlake UI

SECURITYADMIN → It can manage users and roles. SecurityAdmins can access the Account tab in the interface, but they cannot see the Usage & Billing part. They cannot either create Reader sharing accounts or see the Notifications section. Users with this role can also CREATE, ALTER, or DROP Network Policies.


Create Network Policy

Name *

Allowed IP Addresses *

Blocked IP Addresses

Comment

Network policies in SnowFlake

SYSADMIN → User with the SysAdmin role can create warehouses and databases (and other objects) in an account.

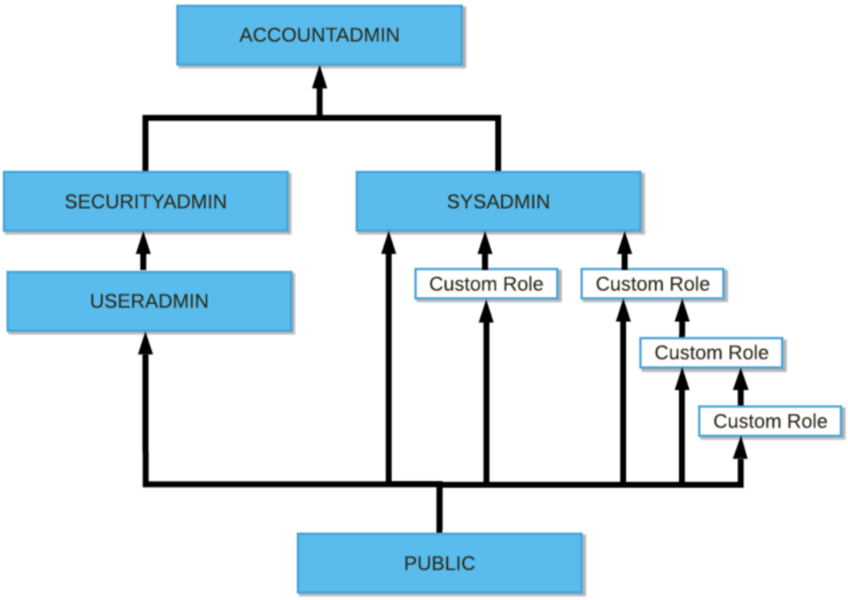
USERADMIN → This role is dedicated to user and role management only. It's like an admin Role, with fewer privileges than the SECURITYADMIN one.

PUBLIC → This role is automatically granted to every user and every role in your account. It can own its objects, but they'll be available for everybody as it is the lowest role in the SnowFlake hierarchy.

CUSTOM ROLES→ They can be created by the SECURITYADMIN role and any role to which the CREATE ROLE privilege has been granted. You can assign the privileges that you want. We will see an example later.

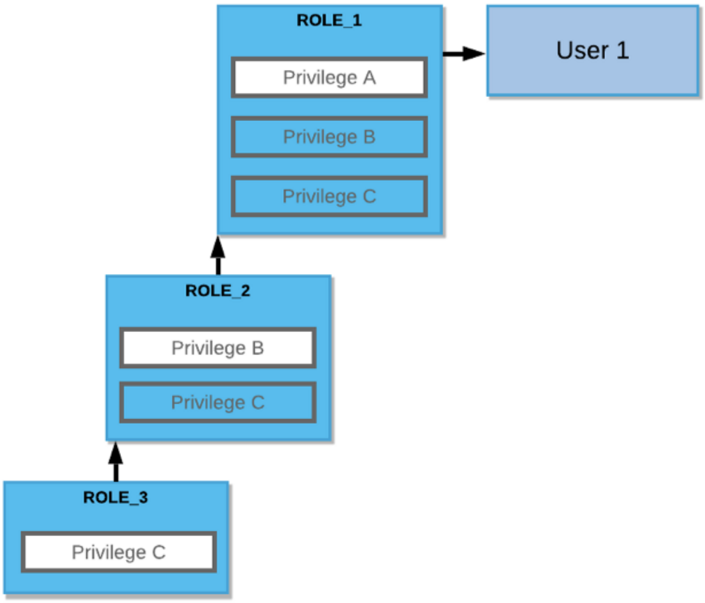
ROLES ENCAPSULATION

Let's start looking at the SnowFlake roles hierarchy:



Roles hierarchy in SnowFlake (via docs.snowflake.com)

What does it mean that ACCOUNTADMIN encapsulates SYSADMIN and SECURITY? It means that **it encapsulates their privileges**, let's see it in another example:



Roles encapsulation in SnowFlake (via docs.snowflake.com)

In this example, ROLE_2 inherits privilege C from the ROLE_3, and ROLE_1 inherits privilege B & C from Role_2. The same thing with the diagram above, SECURITYADMIN inherits privileges from USERADMIN, which inherits privileges from PUBLIC.

ROLES COMMANDS

Create Role:

```
CREATE ROLE <NEW_ROLE>;
```

Encapsulate Role:

```
GRANT ROLE <CHILD_ROLE> TO ROLE <FATHER_ROLE>
```

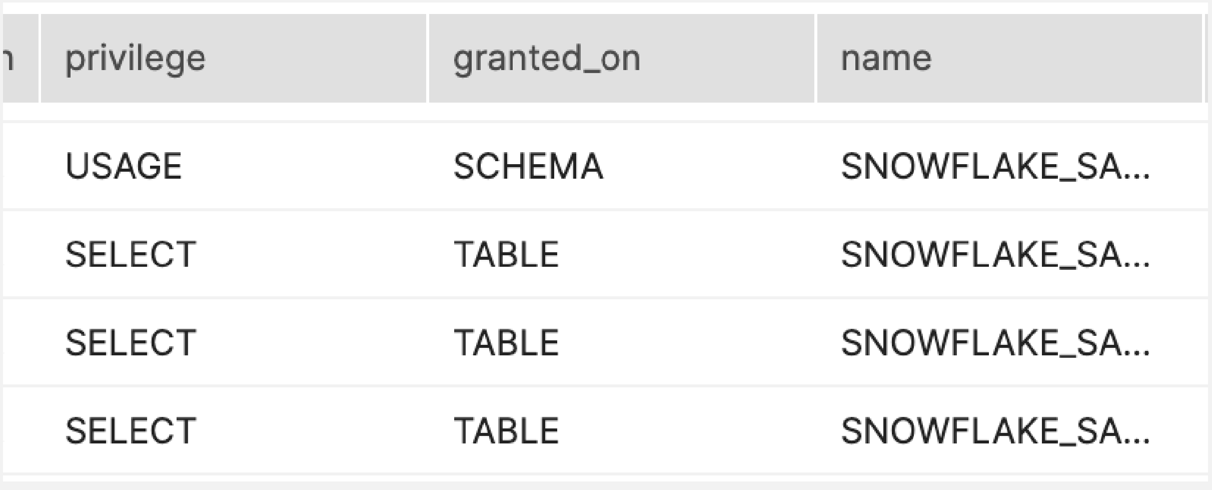
Assign Role to User → *It's essential to GRANT the ROLE to the user. Setting it as a DEFAULT_ROLE when creating the user won't work.*

```
CREATE USER <NEW_USER> PASSWORD = <PASSWORD> DEFAULT_ROLE =
<ROLE> MUST_CHANGE_PASSWORD = TRUE;
```

```
GRANT ROLE <ROLE> TO USER <NEW_USER>;
```

See privileges of the role:

```
SHOW GRANTS TO ROLE <ROLE>
```



privilege	granted_on	name
USAGE	SCHEMA	SNOWFLAKE_SA...
SELECT	TABLE	SNOWFLAKE_SA...
SELECT	TABLE	SNOWFLAKE_SA...
SELECT	TABLE	SNOWFLAKE_SA...

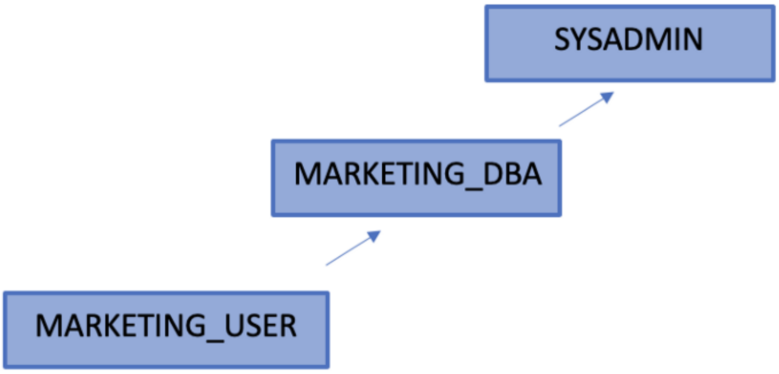
Example of the output of the command

See to whom the role is assigned: For instance, to three different users.

```
SHOW GRANTS OF ROLE <ROLE>
```

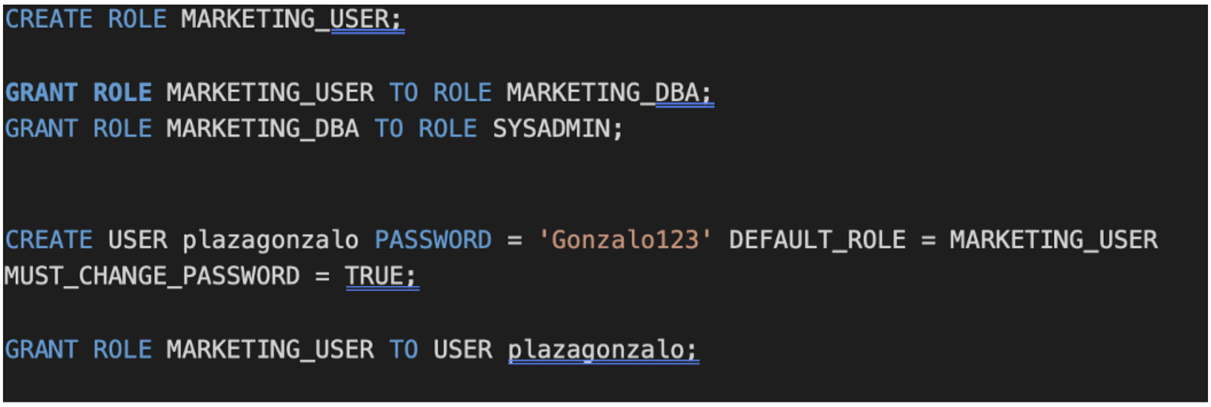
Real example:

Let's see all these commands together with a real example. In this case, the role hierarchy that we want to create will be "MARKETING_USER→ MARKETING_DBA → SYSADMIN". This last role will have privileges from both MARKETING_DBA & MARKETING_USER. This is shown in the following diagram:



Expected hierarchy to create

To create that hierarchy, we should execute the following commands:



```
CREATE ROLE MARKETING_USER;

GRANT ROLE MARKETING_USER TO ROLE MARKETING_DBA;
GRANT ROLE MARKETING_DBA TO ROLE SYSADMIN;


CREATE USER plazagonzalo PASSWORD = 'Gonzalo123' DEFAULT_ROLE = MARKETING_USER
MUST_CHANGE_PASSWORD = TRUE;

GRANT ROLE MARKETING_USER TO USER plazagonzalo;
```

Create New Roles Example SnowFlake.

This is an advanced exam question, but we could also make that SYSADMIN couldn't see the databases from the Marketing users. This might be useful in case we want independent departments with sensitive information. In that case, **SYSADMIN should GRANT the OWNERSHIP of the DB to MARKETING_DBA**

```
CREATE DATABASE MARKETING_DB;GRANT OWNERSHIP ON SCHEMA
MARKETING_DB.PUBLIC TO ROLE MARKETING_DBA;GRANT OWNERSHIP ON
DATABASE MARKETING_DB TO ROLE MARKETING_DBA;
```

TYPICAL EXAM QUESTIONS

1. Which answers are true about roles in SnowFlake?

1. SnowFlake users have a limit on the number of roles that they can assume
2. SnowFlake users can have one or more roles
3. Only a role can be active for a particular session
4. Privileges can be directly assigned to users

Solution: 2, 3

2. Which of the following object types are stored within a schema?

1. Tables
2. Views
3. File Formats
4. Roles

Solution: 1, 2, 3. Roles are account-level objects.

3.Which of the following roles are the default ones in SnowFlake?

1. ACCOUNTADMIN
2. SECURITYADMIN
3. VIEWER
4. USERADMIN
5. SYSADMIN
6. NETWORKADMIN

Solution: 1, 2, 4, 5

4.Which command will you run to list all users and roles to which a role has been granted?

1. SHOW GRANTS TO ROLE <ROLE>
2. SHOW GRANTS OF ROLE <ROLE>
3. SHOW GRANTS IN ROLE <ROLE>

Solution: 2. "SHOW GRANTS OF ROLE" will list the users, whereas "SHOW GRANTS TO ROLE" will list the privileges that this role has access to.

5.**Which roles can create, alter or drop network policies?**

1. ACCOUNTADMIN
2. SECURITYADMIN
3. SYSADMIN
4. USERADMIN

Solution: 1, 2

6.Which roles can create shares and resource monitors?

1. ACCOUNTADMIN
2. SECURITYADMIN
3. SYSADMIN
4. USERADMIN

Solution: 1. AccountAdmins can only create Shares and Resource Monitors by default.

7.Can worksheets of the SnowFlake UI have a different role, warehouse, and database?

1. True
2. False

Solution: 1

Thanks for Reading!