### **What Are Roles in Snowflake?**

**Roles control what users can see and do.** Think of roles as **permission sets**: each role defines what **objects (tables, warehouses, etc.)** a user can **access** and **what actions** they can perform (like read, write, create).

### **Types of Roles in Snowflake:**

| **Role** | **Purpose / Access** |
| --- | --- |
| **ACCOUNTADMIN** | Superuser – full access to everything in the account. Can manage billing, users, roles. |
| **SYSADMIN** | Manages all database objects (tables, views, warehouses, etc.). |
| **SECURITYADMIN** | Manages users and roles (but not objects like tables). |
| **USERADMIN** | Creates and manages users and assigns roles to them. |
| **PUBLIC** | Default role every user gets — very limited access. |
| **Custom Roles** | You can create your own roles with specific permissions (e.g., analyst, dev, etc.). |

### **Role Hierarchy (Simplified View)**

ACCOUNTADMIN

└── SECURITYADMIN

└── USERADMIN

└── SYSADMIN

└── Custom roles (e.g., ANALYST\_ROLE, DEVELOPER\_ROLE)

Higher roles can grant/revoke roles beneath them.

### **Best Practices**

* Grant **minimum access** required (principle of least privilege).
* Use **custom roles** for specific team needs.
* Avoid giving **ACCOUNTADMIN** unless absolutely necessary.

**Example: Create and Grant a Role**

-- Create a custom role

CREATE ROLE analyst\_role;

-- Grant access to a schema

GRANT USAGE ON DATABASE my\_db TO ROLE analyst\_role;

GRANT USAGE ON SCHEMA my\_db.public TO ROLE analyst\_role;

GRANT SELECT ON ALL TABLES IN SCHEMA my\_db.public TO ROLE analyst\_role;

-- Assign role to user

GRANT ROLE analyst\_role TO USER john\_doe;