## **Snowflake RBAC Lab: Designing Role-Based Access Control with Least Privilege and Role Hierarchies**

*Implementing Secure, Scalable Access to HR and Finance Data in Snowflake*

For the best practices and the detailed understanding of RBAC and customized roles:

[Snowflake Accounts and assurance RBAC level 2](https://docs.google.com/document/d/1B_4P3fU29OpLhF4KRRBrlvtDlZyN4dkaIza9z94ZiIs/edit?usp=sharing)

Find the SQL script at the bottom of this lab manual

## **🔧 Lab Objective:**

Create custom roles and assign appropriate access privileges to them, then assign these roles to users.

## **🧱 Assumptions:**

* Two databases exist: fin and hr.
* Users exist (e.g., analyst\_user, accountant\_user).
* Your current role: USERADMIN.

## **✅ Step-by-Step Lab**

### **🔹 Step 1 & 2: Being Security admin create Custom Roles**

**Switch Role to SECURITYADMIN**

USERADMIN can manage users, but creating roles and granting privileges needs SECURITYADMIN.

USE ROLE SECURITYADMIN;

**Create the base roles:**

-- Read-only for HR

CREATE ROLE db\_hr\_r;

-- Read-only for FIN

CREATE ROLE db\_fin\_r;

-- Read-Write for FIN

CREATE ROLE db\_fin\_rw;

### **🔹 Step 3: Grant Privileges to Roles**

#### **📁 HR Database — Read-only**

-- Grant usage on database and schemas

GRANT USAGE ON DATABASE hr TO ROLE db\_hr\_r;

GRANT USAGE ON ALL SCHEMAS IN DATABASE hr TO ROLE db\_hr\_r;

-- Grant select on all tables

GRANT SELECT ON ALL TABLES IN DATABASE hr TO ROLE db\_hr\_r;

#### **📁 FIN Database — Read-only**

GRANT USAGE ON DATABASE fin TO ROLE db\_fin\_r;

GRANT USAGE ON ALL SCHEMAS IN DATABASE fin TO ROLE db\_fin\_r;

GRANT SELECT ON ALL TABLES IN DATABASE fin TO ROLE db\_fin\_r;

#### **📁 FIN Database — Read-Write**

GRANT USAGE ON DATABASE fin TO ROLE db\_fin\_rw;

GRANT USAGE ON ALL SCHEMAS IN DATABASE fin TO ROLE db\_fin\_rw;

GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN DATABASE fin TO ROLE db\_fin\_rw;

### **🔹 Step 4: Grant future privileges to the roles**

To cover future tables or schemas, consider setting up **future grants** .

-- HR Future Grants

GRANT USAGE ON FUTURE SCHEMAS IN DATABASE hr TO ROLE db\_hr\_r;

GRANT SELECT ON FUTURE TABLES IN DATABASE hr TO ROLE db\_hr\_r;

-- FIN Future Grants

GRANT USAGE ON FUTURE SCHEMAS IN DATABASE fin TO ROLE db\_fin\_r;

GRANT SELECT ON FUTURE TABLES IN DATABASE fin TO ROLE db\_fin\_r;

GRANT USAGE ON FUTURE SCHEMAS IN DATABASE fin TO ROLE db\_fin\_rw;

GRANT SELECT, INSERT, UPDATE, DELETE ON FUTURE TABLES IN DATABASE fin TO ROLE db\_fin\_rw;

### **Role Hierarchy and Activation**

**🔹Step 5: Create composite roles**

-- Composite role for analysts (read-only access to both HR and FIN)

CREATE ROLE analyst;

-- Composite role for accountants (read-write FIN + read-only HR)

CREATE ROLE accountant;

-- Assign base roles to composite roles

-- Analyst gets read-only HR and FIN

GRANT ROLE db\_hr\_r TO ROLE analyst;

GRANT ROLE db\_fin\_r TO ROLE analyst;

GRANT USAGE ON WAREHOUSE COMPUTE\_WH TO ROLE analyst;

-- Accountant gets read-write FIN and read-only HR

GRANT ROLE db\_fin\_rw TO ROLE accountant;

GRANT ROLE db\_hr\_r TO ROLE accountant;

GRANT USAGE ON WAREHOUSE COMPUTE\_WH TO ROLE accountant;

### **🔹 Step 6 & 7: Create/Assign Users and Attach Roles**

Switch back to USERADMIN to assign roles to users:

USE ROLE USERADMIN;

**Example users (if not already created):**

-- Optional: Create users (if needed)

CREATE OR REPLACE USER analyst\_user

PASSWORD = 'sn0wf@ll'

LOGIN\_NAME = 'snowflake17analyst'

FIRST\_NAME = 'Snow'

LAST\_NAME = 'Storm'

DEFAULT\_ROLE = analyst

EMAIL = ' snowflake17@azatech.co.in'

MUST\_CHANGE\_PASSWORD = true

DEFAULT\_WAREHOUSE = COMPUTE\_WH;

CREATE OR REPLACE USER accountant\_user

PASSWORD = 'sn0wf@ll'

LOGIN\_NAME = snowflake17accountant'

FIRST\_NAME = 'Snow06'

LAST\_NAME = 'Storm06'

DEFAULT\_ROLE = accountant

EMAIL = ' snowflake17@azatech.co.in'

MUST\_CHANGE\_PASSWORD = true

DEFAULT\_WAREHOUSE = COMPUTE\_WH;

**Assign roles:**

-- Analysts get read-only access to both HR and FIN

GRANT ROLE db\_hr\_r TO USER analyst\_user;

GRANT ROLE db\_fin\_r TO USER analyst\_user;

-- Accountants get read-write access to FIN, read-only to HR

GRANT ROLE db\_fin\_rw TO USER accountant\_user;

GRANT ROLE db\_hr\_r TO USER accountant\_user;

**🔹 Step 8:** Now share the url from the browser tab with the new employees or the users. If you're already logged into Snowflake using **Snowsight**, look at the address bar in your browser:

https://abc12345.us-east-1.snowflakecomputing.com/console

So your account URL is:

https://abc12345.us-east-1.snowflakecomputing.com

**Role activation**

Each user can activate their role in Snowsight or by:

-- Within session

USE ROLE db\_hr\_r;

## 

## **Step 9. Explore the users and roles in your account**

## **Now you can explore all the users and roles in your account by using the ACCOUNTADMIN role.**

## **To explore users and roles, do the following:**

## **In the open worksheet, place your cursor in the** [**USE ROLE**](https://docs.snowflake.com/en/sql-reference/sql/use-role) **line, then select Run. USE ROLE ACCOUNTADMIN;**

## **Place your cursor in the** [**SHOW USERS**](https://docs.snowflake.com/en/sql-reference/sql/show-users) **line, then select Run. SHOW USERS;**

## **Your output looks similar to the following image.**

## **Show all the users in the account. Table output with the following columns: name, created_on, login_name, display_name, first_name.Place your cursor in the** [**SHOW ROLES**](https://docs.snowflake.com/en/sql-reference/sql/show-roles) **line, then select Run. SHOW ROLES;**

## **Your output looks similar to the following image.**

## **Show all the roles in the account. Table output with the following columns: created_on, name, is_default, is_current, is_inherited.**

## 

## **📌 Summary:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | **Database** | **Privileges** | **Assigned To** |
| db\_hr\_r | hr | USAGE, SELECT | analyst\_user, accountant\_user |
| db\_fin\_r | fin | USAGE, SELECT | analyst\_user |
| db\_fin\_rw | fin | USAGE, SELECT, INSERT, UPDATE, DELETE | accountant\_user |

**SQL Script**

-- ========================================

-- LAB: Role-Based Access Control in Snowflake

-- OBJECTIVE:

-- Implement read/write roles for HR and FIN databases

-- Use least privilege and composite role best practices

-- ROLES:

-- db\_hr\_r - Read-only access to HR data

-- db\_fin\_r - Read-only access to FIN data

-- db\_fin\_rw - Read/Write access to FIN data

-- analyst - Composite role for analyst users

-- accountant - Composite role for accountant users

-- ========================================

-- ----------------------------------------

-- Step 1: Switch to SECURITYADMIN role

-- Required to create and manage roles

-- ----------------------------------------

USE ROLE SECURITYADMIN;

-- ----------------------------------------

-- Step 2: Create Base Roles

-- Each role has a narrow, focused purpose

-- ----------------------------------------

-- HR Read-Only

CREATE ROLE IF NOT EXISTS db\_hr\_r;

-- FIN Read-Only

CREATE ROLE IF NOT EXISTS db\_fin\_r;

-- FIN Read-Write

CREATE ROLE IF NOT EXISTS db\_fin\_rw;

-- ----------------------------------------

-- Step 3: Grant Privileges to Roles

-- Following the principle of least privilege

-- ----------------------------------------

-- HR Read-Only Privileges

GRANT USAGE ON DATABASE hr TO ROLE db\_hr\_r;

GRANT USAGE ON ALL SCHEMAS IN DATABASE hr TO ROLE db\_hr\_r;

GRANT SELECT ON ALL TABLES IN DATABASE hr TO ROLE db\_hr\_r;

-- FIN Read-Only Privileges

GRANT USAGE ON DATABASE fin TO ROLE db\_fin\_r;

GRANT USAGE ON ALL SCHEMAS IN DATABASE fin TO ROLE db\_fin\_r;

GRANT SELECT ON ALL TABLES IN DATABASE fin TO ROLE db\_fin\_r;

-- FIN Read-Write Privileges

GRANT USAGE ON DATABASE fin TO ROLE db\_fin\_rw;

GRANT USAGE ON ALL SCHEMAS IN DATABASE fin TO ROLE db\_fin\_rw;

GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN DATABASE fin TO ROLE db\_fin\_rw;

-- ----------------------------------------

-- Best Practice: Future Grants

-- Ensure access to future objects without manual updates

-- ----------------------------------------

-- HR Future Grants

GRANT USAGE ON FUTURE SCHEMAS IN DATABASE hr TO ROLE db\_hr\_r;

GRANT SELECT ON FUTURE TABLES IN DATABASE hr TO ROLE db\_hr\_r;

-- FIN Future Grants

GRANT USAGE ON FUTURE SCHEMAS IN DATABASE fin TO ROLE db\_fin\_r;

GRANT SELECT ON FUTURE TABLES IN DATABASE fin TO ROLE db\_fin\_r;

GRANT USAGE ON FUTURE SCHEMAS IN DATABASE fin TO ROLE db\_fin\_rw;

GRANT SELECT, INSERT, UPDATE, DELETE ON FUTURE TABLES IN DATABASE fin TO ROLE db\_fin\_rw;

-- ----------------------------------------

-- Step 4: Create Composite Roles (Role Hierarchy)

-- Improves manageability by grouping roles

-- ----------------------------------------

-- Composite role for Analysts: Read-only on HR + FIN & warehouse usage

CREATE ROLE IF NOT EXISTS analyst;

GRANT ROLE db\_hr\_r TO ROLE analyst;

GRANT ROLE db\_fin\_r TO ROLE analyst;

GRANT USAGE ON WAREHOUSE COMPUTE\_WH TO ROLE analyst;

-- Composite role for Accountants: Read/Write on FIN, Read-only on HR & warehouse usage

CREATE ROLE IF NOT EXISTS accountant;

GRANT ROLE db\_fin\_rw TO ROLE accountant;

GRANT ROLE db\_hr\_r TO ROLE accountant;

GRANT USAGE ON WAREHOUSE COMPUTE\_WH TO ROLE accountant;

-- ----------------------------------------

-- Step 5: Switch to USERADMIN to manage users

-- USERADMIN manages users and assigns roles

-- ----------------------------------------

USE ROLE USERADMIN;

-- ----------------------------------------

-- Step 6: Create Users (Optional)

-- For lab/demo purposes

-- ----------------------------------------

CREATE OR REPLACE USER analyst\_user

PASSWORD = 'sn0wf@ll'

LOGIN\_NAME = 'snowstorm'

FIRST\_NAME = 'Snow'

LAST\_NAME = 'Storm'

DEFAULT\_ROLE = analyst

EMAIL = 'snow.storm@snowflake.com'

MUST\_CHANGE\_PASSWORD = true

DEFAULT\_WAREHOUSE = COMPUTE\_WH;

CREATE OR REPLACE USER accountant\_user

PASSWORD = 'sn0wf@ll'

LOGIN\_NAME = 'snowstorm'

FIRST\_NAME = 'Snow'

LAST\_NAME = 'Storm'

DEFAULT\_ROLE = accountant

EMAIL = 'snow.storm@snowflake.com'

MUST\_CHANGE\_PASSWORD = true

DEFAULT\_WAREHOUSE = COMPUTE\_WH;

-- ----------------------------------------

-- Step 7: Assign Composite Roles to Users

-- Simplifies user privilege management

-- ----------------------------------------

GRANT ROLE analyst TO USER analyst\_user;

GRANT ROLE accountant TO USER accountant\_user;

-- ========================================

-- END OF THE SCRIPT

-- ========================================

**Step 8:** Now share the url from the browser tab with the new employees or the users. If you're already logged into Snowflake using **Snowsight**, look at the address bar in your browser:

https://abc12345.us-east-1.snowflakecomputing.com/console

So your account URL is:

https://abc12345.us-east-1.snowflakecomputing.com

**Role activation**

Each user can activate their role in Snowsight or by:

-- Within session

USE ROLE db\_hr\_r;

**✅ LAB COMPLETE**

BEST PRACTICES FOLLOWED:

1. Role hierarchy for simplified access control
2. Least privilege design
3. Future grants to support automation
4. Use of USERADMIN/SECURITYADMIN roles appropriately

|  |  |  |
| --- | --- | --- |
| **Best Practice** | **Included?** | **Notes** |
| **Least Privilege Principle** | ✅ | Roles are scoped to specific privileges only |
| **Future Grants** | ✅ | Ensures access to new tables/schemas |
| **Role Hierarchy** (Composite Roles) | ✅ | Improves scalability and clarity |
| **Separation of Duties** | ✅ | SECURITYADMIN for roles, USERADMIN for users |
| **Clear Role Naming Conventions** | ✅ | Descriptive, purpose-based role names |

Happy Learning

Regards

Saransh Jain