**Access Control(Roles & Users)**

**Access control** privileges determine who can access and perform operations on specific objects in **Snowflake****. There are 2 types of access control DAC & RBAC. actually** Snowflake uses Role-Based Access Control (RBAC). Users are people who log in, and roles define their Permissions (like creating tables, running queries, or accessing data) are granted to **roles**, not directly to users. A user can have multiple roles, but only one is active at a time. There are **two main types**:

1. **Discretionary Access Control (DAC):**
   * Object owners decide who can access their objects (like tables, schemas).
   * Example: If I create a table, I can grant SELECT privilege to a role.
2. **Role-Based Access Control (RBAC):**
   * Access is managed through **roles**.
   * Privileges are assigned to roles → roles assigned to users.
   * Example: ANALYST\_ROLE has read-only privileges, while DEVELOPER\_ROLE has create/update rights.

**🔹 Roles in Snowflake**

Snowflake provides some **system-defined roles**, and we can also create custom roles.

**Default/System Roles:**

* **ACCOUNTADMIN** → Super user, full access across account.
* **SECURITYADMIN** → Manages users and roles.
* **SYSADMIN** → Creates databases, schemas, and objects.
* **PUBLIC** → Every user automatically gets this (minimal access).
* USER ADMIN Role

**Custom Roles (common in projects):**

* **DEVELOPER\_ROLE** → Create tables, test data pipelines.
* **ANALYST\_ROLE** → Read-only access for reporting.
* **ETL\_ROLE** → Data loading, staging, and transformation.

**🔹**

**Users in Snowflake**

* A **user** is an account with login credentials (username, password, MFA).
* Each user is assigned one or more roles → but can only use **one active role at a time**.
* Example: In your Siemens project, you may have had multiple roles (like DEVELOPER\_ROLE and ANALYST\_ROLE), but you activate the one you need while working.

“In my projects, we used Snowflake’s RBAC model. For example, in Siemens, we created separate roles for developers, testers, and analysts. Developers had privileges to create tables and manage pipelines, while analysts only had SELECT rights for reporting. I also worked with the Security Admin to manage user accounts, reset roles, and ensure least-privilege access. This separation of duties helped us maintain security and compliance.”

// Dedicated Warehouse for Product Owners and Financial Analysts

// Product Owners

CREATE WAREHOUSE PRODUCT\_OWNERS\_WH

WITH WAREHOUSE\_SIZE = 'SMALL'

WAREHOUSE\_TYPE = 'STANDARD'

AUTO\_SUSPEND = 600

AUTO\_RESUME = TRUE

MIN\_CLUSTER\_COUNT = 1

MAX\_CLUSTER\_COUNT = 2

SCALING\_POLICY = 'ECONOMY';

// Financial analysts

CREATE WAREHOUSE FINANCIAL\_ANALYSTS\_WH

WITH WAREHOUSE\_SIZE = 'XSMALL'

WAREHOUSE\_TYPE = 'STANDARD'

AUTO\_SUSPEND = 600

AUTO\_RESUME = TRUE

MIN\_CLUSTER\_COUNT = 1

MAX\_CLUSTER\_COUNT = 2

SCALING\_POLICY = 'ECONOMY';

// Create roles for above user groups. Use Account Admin to handle roles

CREATE ROLE PRODUCT\_OWNERS\_ROLE;

GRANT USAGE ON WAREHOUSE PRODUCT\_OWNERS\_WH TO ROLE PRODUCT\_OWNERS\_ROLE;

CREATE ROLE FINANCIAL\_ANALYSTS\_ROLE;

GRANT USAGE ON WAREHOUSE FINANCIAL\_ANALYSTS\_WH TO ROLE FINANCIAL\_ANALYSTS\_ROLE;

// Create some users

// Product Owners - Adam, Smith, John, Alex

CREATE USER ADAM PASSWORD = 'ADAM'

LOGIN\_NAME = 'ADAM'

DEFAULT\_ROLE='PRODUCT\_OWNERS\_ROLE'

DEFAULT\_WAREHOUSE = 'PRODUCT\_OWNERS\_WH'

MUST\_CHANGE\_PASSWORD = FALSE;

CREATE USER SMITH PASSWORD = 'SMITH'

LOGIN\_NAME = 'SMITH'

DEFAULT\_ROLE='PRODUCT\_OWNERS\_ROLE'

DEFAULT\_WAREHOUSE = 'PRODUCT\_OWNERS\_WH'

MUST\_CHANGE\_PASSWORD = FALSE;

CREATE USER JOHN PASSWORD = 'JOHN'

LOGIN\_NAME = 'JOHN'

DEFAULT\_ROLE='PRODUCT\_OWNERS\_ROLE'

DEFAULT\_WAREHOUSE = 'PRODUCT\_OWNERS\_WH'

MUST\_CHANGE\_PASSWORD = FALSE;

CREATE USER ALEX PASSWORD = 'ALEX'

LOGIN\_NAME = 'ALEX'

DEFAULT\_ROLE='PRODUCT\_OWNERS\_ROLE'

DEFAULT\_WAREHOUSE = 'PRODUCT\_OWNERS\_WH'

MUST\_CHANGE\_PASSWORD = FALSE;

GRANT ROLE PRODUCT\_OWNERS\_ROLE TO USER ADAM;

GRANT ROLE PRODUCT\_OWNERS\_ROLE TO USER SMITH;

GRANT ROLE PRODUCT\_OWNERS\_ROLE TO USER JOHN;

GRANT ROLE PRODUCT\_OWNERS\_ROLE TO USER ALEX;

SHOW USERS;

DESC USER ADAM;

//Reset Password generates a URL to share with User.

//Existing password remain valid until changed by user

// Generated URL expires in 4 hrs

// Disabling the user

//Terminates existing session

// User immediately locked out

ALTER USER ADAM RESET PASSWORD ;

// Financial Analysts - Ashley, Bob, Jessica

CREATE USER ASHLEY PASSWORD = 'ASHLEY'

LOGIN\_NAME = 'ASHLEY'

DEFAULT\_ROLE='FINANCIAL\_ANALYSTS\_ROLE'

DEFAULT\_WAREHOUSE = 'FINACIAL\_ANALYSTS\_WH'

MUST\_CHANGE\_PASSWORD = FALSE;

CREATE USER BOB PASSWORD = 'BOB'

LOGIN\_NAME = 'BOB'

DEFAULT\_ROLE='FINANCIAL\_ANALYSTS\_ROLE'

DEFAULT\_WAREHOUSE = 'FINACIAL\_ANALYSTS\_WH'

MUST\_CHANGE\_PASSWORD = FALSE;

CREATE USER JESSICA PASSWORD = 'JESSICA'

LOGIN\_NAME = 'JESSICA'

DEFAULT\_ROLE='FINANCIAL\_ANALYSTS\_ROLE'

DEFAULT\_WAREHOUSE = 'FINACIAL\_ANALYSTS\_WH'

MUST\_CHANGE\_PASSWORD = FALSE;

GRANT ROLE FINANCIAL\_ANALYSTS\_ROLE TO USER ASHLEY;

GRANT ROLE FINANCIAL\_ANALYSTS\_ROLE TO USER BOB;

GRANT ROLE FINANCIAL\_ANALYSTS\_ROLE TO USER JESSICA;

// Clean up

DROP USER ADAM;

DROP USER SMITH;

DROP USER JOHN;

DROP USER ALEX;

DROP USER ASHLEY;

DROP USER BOB;

DROP USER JESSICA;

DROP ROLE PRODUCT\_OWNERS\_ROLE;

DROP ROLE FINANCIAL\_ANALYSTS\_ROLE;

DROP WAREHOUSE PRODUCT\_OWNERS\_WH;

DROP WAREHOUSE FINANCIAL\_ANALYSTS\_WH;

alter session set use\_cached\_result = false;

alter warehouse compute\_wh suspend;

select \* from SNOWFLAKE\_SAMPLE\_DATA.TPCDS\_SF100TCL.CATALOG\_RETURNS; -- 30m 39s s amf m

create database practice\_db;

create table customer as select \* from SNOWFLAKE\_SAMPLE\_DATA.TPCDS\_SF10TCL.CUSTOMER;

select count(\*) from CUSTOMER;

select min(C\_CUSTOMER\_ID) from CUSTOMER;

select max(C\_CUSTOMER\_ID) from CUSTOMER;

--Check that the warehouse in suspended state. The warehouse cashe is cleared if it is in suspended state.

show warehouses ;

-- Running First Time - Warehouse cache will not be used and result cache will not be used

alter session set use\_cached\_result = false ;

-- Lets run a little complicated query which comes back in few seconds

select ca\_state, ca\_city, count(\*) from SNOWFLAKE\_SAMPLE\_DATA.TPCDS\_SF10TCL.CUSTOMER\_ADDRESS join SNOWFLAKE\_SAMPLE\_DATA.TPCDS\_SF10TCL.CUSTOMER on c\_current\_addr\_sk = ca\_address\_sk where ca\_state in ('NC','OH','NY','TX','MT','VA','OK','GA','MN','KY') group by ca\_state, ca\_city;

-- Check the query profile, warehouse cache should not have been used

-- Run the query again

select ca\_state, ca\_city, count(\*) from SNOWFLAKE\_SAMPLE\_DATA.TPCDS\_SF10TCL.CUSTOMER\_ADDRESS join SNOWFLAKE\_SAMPLE\_DATA.TPCDS\_SF10TCL.CUSTOMER on c\_current\_addr\_sk = ca\_address\_sk where ca\_state in ('NC','OH','NY','TX','MT','VA','OK','GA','MN','KY') group by ca\_state, ca\_city;

-- check the query profile. This time Warehouse Cache must have been used

-- Now lets enable the result cache

alter session set use\_cached\_result = true ; -- Run the query again

select ca\_state, ca\_city, count(\*) from SNOWFLAKE\_SAMPLE\_DATA.TPCDS\_SF10TCL.CUSTOMER\_ADDRESS join SNOWFLAKE\_SAMPLE\_DATA.TPCDS\_SF10TCL.CUSTOMER on c\_current\_addr\_sk = ca\_address\_sk where ca\_state in ('NC','OH','NY','TX','MT','VA','OK','GA','MN','KY') group by ca\_state, ca\_city;

-- Check the query Profile it should use result cache

-- Now lets see if a different user sends the same query to the warehouse. It should use result cache

-- Note result cache is true by default, we have set it false to show the usage of warehouse cache

-- create role

CREATE ROLE PRODUCT\_OWNERS\_ROLE;

-- Product owner role

GRANT USAGE ON WAREHOUSE COMPUTE\_WH TO ROLE PRODUCT\_OWNERS\_ROLE;

-- create user

CREATE USER ADAM PASSWORD = 'ADAM'

LOGIN\_NAME = 'ADAM'

DEFAULT\_ROLE='PRODUCT\_OWNERS\_ROLE'

DEFAULT\_WAREHOUSE = 'COMPUTE\_WH'

MUST\_CHANGE\_PASSWORD = FALSE;

--grant role

GRANT ROLE PRODUCT\_OWNERS\_ROLE to USER ADAM;

-- Cleanup

-- drop role

drop role PRODUCT\_OWNERS\_ROLE ;

-- drop user

DROP USER ADAM ;

-- suspend warehouse alter warehouse compute\_wh resume; alter warehouse compute\_wh suspend;

-- show warehouses show warehouses ;

show users ;

-- snowflake and vksingh -- which got created at trial

show roles ; -- system defined roles.

-- ACCOUNTADMIN --

-- As mentioned earlier this is the top level role or most powerful role and should be granted to very limited people say 2.

-- It can configure parameters at account level

-- view billing and manage it

-- and can terminate the execution of SQL statements any time.

-- when your account get created the first user is assigned the ACCOUNTADMIN role

show users ;

-- So lets create one more accountadmin users - Nick. This will make two accountadmin users as recommended by snowflake

-- create another user having account admin role

create or replace user nick\_acct\_adm password = '123'

login\_name = 'nick\_acct\_adm'

first\_name = 'Nick'

email = 'info@alphaedgesolutions.com' -- important to give the email for urgent issues if snowflake wants to connect

default\_role = 'ACCOUNTADMIN'

must\_change\_password = false ;

grant role ACCOUNTADMIN to user nick\_acct\_adm ;

-- show users

show users ;

show grants to user nick\_acct\_adm ;

show grants on user nick\_acct\_adm ;

-- MFA authentication should be enabled for accountadmin users

-- Go to Profile and select on enroll MFA

-- if you want to disable you can run the following command

alter user <username> set DISABLE\_MFA = true;

--- SECURITYADMIN — Security administrator – manage object grant globally - modify or revoke any grant

-- Inherits useradmin

create or replace user dean\_sec\_adm password = '123'

login\_name = 'dean\_sec\_adm'

first\_name = 'Dean'

email = 'info@alphaedgesolutions.com' -- important to give the email for urgent issues if snowflake wants to connect

default\_role = 'SECURITYADMIN'

must\_change\_password = false ;

show grants to user dean\_sec\_adm ;

grant role SECURITYADMIN to user dean\_sec\_adm ;

-- check grants

show grants to user dean\_sec\_adm ;

---- USERADMIN ---- users and roles administrator -- dedicated to user and role management --

-- Now we need to create a user having user admin access who can create users and roles

create or replace user john\_usr\_adm password = '123'

login\_name = 'john\_usr\_adm'

first\_name = 'John'

email = 'info@alphaedgesolutions.com' -- important to give the email for urgent issues if snowflake wants to connect

default\_role = 'USERADMIN'

must\_change\_password = false ;

grant role USERADMIN to user john\_usr\_adm ;

show grants to user john\_usr\_adm ;

---- login --- as user admin to create users and roles --

show users ;

--- SYSADMIN -- create a sysadmin user – system administrator

-- role that has privs to create warehouse/databases and other objects

-- Now we need to create a user having user admin access who can create warehouse, databases and other objects

create or replace user jes\_sys\_adm password = '123'

login\_name = 'jes\_sys\_adm'

first\_name = 'Jess'

email = 'info@alphaedgesolutions.com' -- important to give the email for urgent issues if snowflake wants to connect

default\_role = 'SYSADMIN'

must\_change\_password = false ;

grant role SYSADMIN to user jes\_sys\_adm ;

-- show users

show users ;

show grants to user jes\_sys\_adm ;

-- so all the 4 users are created and we will see how these users control the system and give it to developers / qa and other users

-- Lets login as john\_usr\_adm to create some more users who will be using the systems and the related roles.

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

create or replace role developer\_role ;

create or replace user developer1 password = '123'

login\_name = 'developer1'

first\_name = 'Developer1'

email = 'info@alphaedgesolutions.com' -- important to give the email for urgent issues if snowflake wants to connect

default\_role = developer\_role

must\_change\_password = false ;

grant role developer\_role to user developer1 ;

show grants to user developer1 ;

create or replace user developer2 password = '123'

login\_name = 'developer2'

first\_name = 'Developer2'

email = 'info@alphaedgesolutions.com' -- important to give the email for urgent issues if snowflake wants to connect

default\_role = developer\_role

must\_change\_password = false ;

grant role developer\_role to user developer2 ;

show grants to user developer2 ;

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

create or replace role qa\_role ;

create or replace user qa1 password = '123'

login\_name = 'qa1'

first\_name = 'Qa1'

email = 'info@alphaedgesolutions.com' -- important to give the email for urgent issues if snowflake wants to connect

default\_role = qa\_role

must\_change\_password = false ;

grant role qa\_role to user qa1 ;

show grants to user qa1;

create or replace user qa2 password = '123'

login\_name = 'qa2'

first\_name = 'Qa2'

email = 'info@alphaedgesolutions.com' -- important to give the email for urgent issues if snowflake wants to connect

default\_role = qa\_role

must\_change\_password = false ;

grant role qa\_role to user qa2 ;

show grants to user qa2 ;

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

create or replace role read\_role ;

create or replace user reader1 password = '123'

login\_name = 'reader1'

first\_name = 'Reader1'

email = 'info@alphaedgesolutions.com' -- important to give the email for urgent issues if snowflake wants to connect

default\_role = read\_role

must\_change\_password = false ;

grant role read\_role to user reader1 ;

create or replace user reader2 password = '123'

login\_name = 'reader2'

first\_name = 'Reader2'

email = 'info@alphaedgesolutions.com' -- important to give the email for urgent issues if snowflake wants to connect

default\_role = read\_role

must\_change\_password = false ;

grant role read\_role to user reader2 ;

show grants to user reader1;

show grants to user reader2;

show roles ;

------------------—------SYSADMIN—-----CREATE—OBJECTS –DBs–WAREHOUSE —

Login as jes\_sys\_adm/123

create or replace database db\_dev ;

create or replace schema stg ;

create or replace schema tgt ;

--

show databases;

use database db\_dev ;

use schema stg ;

show schemas ;

--- create warehouse

create or replace warehouse wh\_dev ;

-- show

show warehouses ;

show roles ;

-- Create table client\_stg

use schema stg ;

create or replace table client\_stg

( id NUMBER(38,0), first\_name VARCHAR(16), last\_name VARCHAR(50),

sex VARCHAR(1), ethinicity VARCHAR(30), ssn VARCHAR(15),

street\_address VARCHAR(90),status VARCHAR(10)

);

delete from client\_stg ;

insert into client\_stg values (111111, 'James', 'Schwartz', 'M', 'American','342-76-9087','5676 Washington Street','ACTIVE') ;

insert into client\_stg values (222222, 'Jessica', 'Escobar', 'F', 'Hispanic','456-93-5629','3234 WateringCan Drive','INACTIVE') ;

insert into client\_stg values (333333, 'Ben', 'Hardy', 'M', 'American','876-98-3245','6578 Historic Circle','INACTIVE') ;

insert into client\_stg values (444444, 'Anjali', 'Singh', 'F', 'Indian American','435-87-6532','8978 Autumn Day Drive','ACTIVE') ;

insert into client\_stg values (555555, 'Dean', 'Tracy', 'M', 'African','767-34-7656','2343 India Street','ACTIVE') ;

select \* from client\_stg ;

-- create a target table

create or replace table client\_tgt

( id NUMBER(38,0), first\_name VARCHAR(16), last\_name VARCHAR(50),

sex VARCHAR(1), ethinicity VARCHAR(30), ssn VARCHAR(15),

street\_address VARCHAR(90),status VARCHAR(10)

);

-- create a procedure

create or replace procedure prc\_load\_client\_tgt()

returns varchar

language sql

as

$$

begin

insert into client\_tgt

select \* from client\_stg ;

return 'Record Inserted' ;

end;

$$

;

-- test the procedure

select \* from client\_stg ;

select \* from client\_tgt ; -- empty

call prc\_load\_client\_tgt() ;

select \* from client\_tgt ; -- procedure works

-- create a view

create or replace view active\_client as select \* from client\_tgt where status = 'ACTIVE' ;

--

select \* from active\_client ;

---

show roles ;

show tables ;

show procedures ;

show views ;

grant usage on warehouse wh\_dev to developer\_role ;

grant usage on database db\_dev to developer\_role ;

grant usage on schema db\_dev.stg to developer\_role ;

grant all privileges on database db\_dev to developer\_role ;

grant all privileges on schema db\_dev.stg to developer\_role ;

grant select, insert, update, delete on client\_stg to developer\_role ;

grant select, insert, update, delete on client\_tgt to developer\_role ;

grant usage on procedure prc\_load\_client\_tgt() to developer\_role ;

grant select on view active\_client to developer\_role ;

--- qa role

grant usage on warehouse wh\_dev to qa\_role ;

grant usage on database db\_dev to qa\_role ;

grant usage on schema db\_dev.stg to qa\_role ;

-- No insert and delete on stg to QA, No insert in tgt

grant select, update on client\_stg to qa\_role ;

grant select, update, delete on client\_tgt to qa\_role ;

grant usage on procedure prc\_load\_client\_tgt() to qa\_role ;

grant select on view active\_client to qa\_role ;

--- read role

grant usage on warehouse wh\_dev to read\_role ;

grant usage on database db\_dev to read\_role ;

grant usage on schema db\_dev.stg to read\_role ;

-- only select on table

grant select on client\_stg to read\_role ;

grant select on client\_tgt to read\_role ;

grant select on view active\_client to read\_role ;

– developer role should have all access in db\_dev on all objects

– qa role should have read and update and delete access in db\_qa

– should have read access only

------------------- qa role

grant usage on warehouse wh\_dev to qa\_role ;

grant usage on database db\_dev to qa\_role ;

grant usage on schema db\_dev.stg to qa\_role ;

-- No insert and delete on stg to QA, No insert in tgt

grant select, update on client\_stg to qa\_role ;

grant select, update, delete on client\_tgt to qa\_role ;

grant usage on procedure prc\_load\_client\_tgt() to qa\_role ;

grant select on view active\_client to qa\_role ;

--- read role

grant usage on warehouse wh\_dev to read\_role ;

grant usage on database db\_dev to read\_role ;

grant usage on schema db\_dev.stg to read\_role ;

-- only select on table

grant select on client\_stg to read\_role ;

grant select on client\_tgt to read\_role ;

grant select on view active\_client to read\_role ;

– developer role should have all access in db\_dev on all objects

– qa role should have read and update and delete access in db\_qa

– should have read access only

-- drop database (cleanup if needed)

drop database db\_dev ;

drop warehouse wh\_dev

developer1

-- check create table and insert access

use schema tgt ;

create or replace table developer1\_test\_table

( id NUMBER(38,0), first\_name VARCHAR(16), last\_name VARCHAR(50),

sex VARCHAR(1), ethinicity VARCHAR(30), ssn VARCHAR(15),

street\_address VARCHAR(90),status VARCHAR(10) );

insert into developer1\_test\_table values (111111, 'James', 'Schwartz', 'M', 'American','342-76-9087','5676 Washington Street','ACTIVE') ;

insert into developer1\_test\_table values (222222, 'Jessica', 'Escobar', 'F', 'Hispanic','456-93-5629','3234 WateringCan Drive','INACTIVE') ;

select \* from developer1\_test\_table ;

-- existing table insert/update/delete

select \* from stg.client\_stg ;

select \* from client\_tgt ;

delete from client\_tgt ;

-- execute the procedure

call prc\_load\_client\_tgt() ;

select \* from client\_tgt ;

select \* from active\_client ;

developer2

-- check create table

Use schema tgt ;

show tables ;

create or replace table developer2\_test\_table

( id NUMBER(38,0), first\_name VARCHAR(16), last\_name VARCHAR(50),

sex VARCHAR(1), ethinicity VARCHAR(30), ssn VARCHAR(15),

street\_address VARCHAR(90),status VARCHAR(10)

);

insert into developer2\_test\_table values (111111, 'James', 'Schwartz', 'M', 'American','342-76-9087','5676 Washington Street','ACTIVE') ;

insert into developer2\_test\_table values (222222, 'Jessica', 'Escobar', 'F', 'Hispanic','456-93-5629','3234 WateringCan Drive','INACTIVE') ;

select \* from developer2\_test\_table ;

-- existing table insert/update/delete

select \* from stg.client\_stg ;

select \* from client\_tgt ;

delete from client\_tgt ;

-- execute the procedure

call prc\_load\_client\_tgt() ;

select \* from client\_tgt ;

select \* from active\_client ;

QA1

-- check create table

show tables ;

create or replace table qa1\_test\_table

( id NUMBER(38,0), first\_name VARCHAR(16), last\_name VARCHAR(50),

sex VARCHAR(1), ethinicity VARCHAR(30), ssn VARCHAR(15),

street\_address VARCHAR(90),status VARCHAR(10)

);

-- Table creation not allowed

-- existing table delete not allowed

select \* from client\_stg ;

delete from client\_stg ; -- not allowed

insert into client\_stg values (6666666, 'James', 'Schwartz', 'M', 'American','342-76-9087','5676 Washington Street','ACTIVE') ; -- not allowed

-- update

select \* from client\_stg ;

update client\_stg set ethinicity = 'African' where id = 111111 ;

select \* from client\_stg where id = 111111 ;

delete from client\_tgt ;

select \* from client\_tgt ;

show procedures

-- execute the procedure

call prc\_load\_client\_tgt() ; -- not allowed

select \* from client\_tgt ;

show views

select \* from active\_client ; --

show roles ;

show grants to role qa\_role ;

— reader1 —

-- check create table

show tables ;

create or replace table reader1\_test\_table

( id NUMBER(38,0), first\_name VARCHAR(16), last\_name VARCHAR(50),

sex VARCHAR(1), ethinicity VARCHAR(30), ssn VARCHAR(15),

street\_address VARCHAR(90),status VARCHAR(10)

);

-- Table creation not allowed

-- existing table delete not allowed

select \* from stg.client\_stg ;

delete from stg.client\_stg ; -- not allowed

insert into stg.client\_stg values (6666666, 'James', 'Schwartz', 'M', 'American','342-76-9087','5676 Washington Street','ACTIVE') ; -- not allowed

-- update

select \* from stg.client\_stg ;

update stg.client\_stg set ethinicity = 'African' where id = 111111 ; -- not allowed

select \* from client\_stg where id = 111111 ;

delete from client\_tgt ; -- not allowed

select \* from client\_tgt ;

show procedures

-- execute the procedure

call prc\_load\_client\_tgt() ; -- not allowed

select \* from client\_tgt ;

show views

select \* from active\_client ; --

show roles ;

show grants to role reader\_role ;

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

JOHNUSRADM

-- create a public user

create or replace user tony\_usr\_pub password = '123'

login\_name = 'tony\_usr\_pub'

first\_name = 'Tony'

email = 'info@alphaedgesolutions.com' -- important to give the email for urgent issues if snowflake wants to connect

default\_role = public

must\_change\_password = false ;

-- cleanup

drop user tony\_usr\_pub ;

JESSYSADM

create or replace warehouse wh\_pub ;

create or replace database db\_pub ;

create or replace schema pub ;

grant usage on warehouse wh\_pub to public ;

grant usage on database db\_pub to public ;

grant usage on schema pub to public ;

grant all privileges on database db\_pub to public ;

grant all privileges on schema db\_pub.pub to public ;

grant role QA\_ROLE to role sysadmin;

-- cleanup

drop warehouse if exists wh\_pub ;

drop database if exists db\_pub ;

TONYUSRPUB

create or replace table client\_pub

( id NUMBER(38,0), first\_name VARCHAR(16), last\_name VARCHAR(50),

sex VARCHAR(1), ethinicity VARCHAR(30), ssn VARCHAR(15),

street\_address VARCHAR(90),status VARCHAR(10)

);

delete from client\_pub ;

insert into client\_pub values (111111, 'James', 'Schwartz', 'M', 'American','342-76-9087','5676 Washington Street','ACTIVE') ;

insert into client\_pub values (222222, 'Jessica', 'Escobar', 'F', 'Hispanic','456-93-5629','3234 WateringCan Drive','INACTIVE') ;

insert into client\_pub values (333333, 'Ben', 'Hardy', 'M', 'American','876-98-3245','6578 Historic Circle','INACTIVE') ;

insert into client\_pub values (444444, 'Anjali', 'Singh', 'F', 'Indian American','435-87-6532','8978 Autumn Day Drive','ACTIVE') ;

insert into client\_pub values (555555, 'Dean', 'Tracy', 'M', 'African','767-34-7656','2343 India Street','ACTIVE') ;

select \* from client\_pub ;

Check in other users if this public warehouse / database / schema / table is available to all users. We have not given any explicit grant to any role or user. It is only to public

reader1

select \* from client\_pub ;

delete from client\_pub ;

insert into client\_pub values (111111, 'James', 'Schwartz', 'M', 'American','342-76-9087','5676 Washington Street','ACTIVE') ;

insert into client\_pub values (222222, 'Jessica', 'Escobar', 'F', 'Hispanic','456-93-5629','3234 WateringCan Drive','INACTIVE') ;

insert into client\_pub values (333333, 'Ben', 'Hardy', 'M', 'American','876-98-3245','6578 Historic Circle','INACTIVE') ;

insert into client\_pub values (444444, 'Anjali', 'Singh', 'F', 'Indian American','435-87-6532','8978 Autumn Day Drive','ACTIVE') ;

insert into client\_pub values (555555, 'Dean', 'Tracy', 'M', 'African','767-34-7656','2343 India Street','ACTIVE') ;

select \* from client\_pub ;

update client\_pub set status = 'INACTIVE' where id = 111111;

-- all works

deansecadm

select \* from client\_pub ;

delete from client\_pub ;

insert into client\_pub values (111111, 'James', 'Schwartz', 'M', 'American','342-76-9087','5676 Washington Street','ACTIVE') ;

insert into client\_pub values (222222, 'Jessica', 'Escobar', 'F', 'Hispanic','456-93-5629','3234 WateringCan Drive','INACTIVE') ;

insert into client\_pub values (333333, 'Ben', 'Hardy', 'M', 'American','876-98-3245','6578 Historic Circle','INACTIVE') ;

insert into client\_pub values (444444, 'Anjali', 'Singh', 'F', 'Indian American','435-87-6532','8978 Autumn Day Drive','ACTIVE') ;

insert into client\_pub values (555555, 'Dean', 'Tracy', 'M', 'African','767-34-7656','2343 India Street','ACTIVE') ;

select \* from client\_pub ;

update client\_pub set status = 'INACTIVE' where id = 111111;

-- all works

nickacctadm

select \* from client\_pub ;

delete from client\_pub ;

insert into client\_pub values (111111, 'James', 'Schwartz', 'M', 'American','342-76-9087','5676 Washington Street','ACTIVE') ;

insert into client\_pub values (222222, 'Jessica', 'Escobar', 'F', 'Hispanic','456-93-5629','3234 WateringCan Drive','INACTIVE') ;

insert into client\_pub values (333333, 'Ben', 'Hardy', 'M', 'American','876-98-3245','6578 Historic Circle','INACTIVE') ;

insert into client\_pub values (444444, 'Anjali', 'Singh', 'F', 'Indian American','435-87-6532','8978 Autumn Day Drive','ACTIVE') ;

insert into client\_pub values (555555, 'Dean', 'Tracy', 'M', 'African','767-34-7656','2343 India Street','ACTIVE') ;

select \* from client\_pub ;

update client\_pub set status = 'INACTIVE' where id = 111111;