SNOWFLAKE ASSIGNMENT : Aman Pandey

TAS 208

-- 1. Create roles as per the below-mentioned hierarchy. Accountadmin already exists in Snowflake

CREATE ROLE Admin;

CREATE ROLE PII;

CREATE ROLE Developer;

GRANT ROLE Admin TO ROLE Accountadmin;

GRANT ROLE PII TO ROLE Accountadmin;

GRANT ROLE Developer TO ROLE admin;

-- 2. Create an M-sized warehouse using the accountadmin role, name -> assignment\_wh and use it for all the queries

CREATE WAREHOUSE IF NOT EXISTS assignment\_wh

WAREHOUSE\_SIZE = 'Medium';

GRANT CREATE DATABASE ON ACCOUNT TO ROLE admin;

GRANT CREATE WAREHOUSE ON ACCOUNT TO ROLE admin;

GRANT CREATE WAREHOUSE ON ACCOUNT TO ROLE pii;

GRANT CREATE WAREHOUSE ON ACCOUNT TO ROLE developer;

GRANT USAGE ON WAREHOUSE assignment\_wh TO ROLE admin;

GRANT USAGE ON WAREHOUSE assignment\_wh TO ROLE PII;

GRANT USAGE ON WAREHOUSE assignment\_wh TO ROLE DEVELOPER;

-- 3 .Switch to the admin role (

USE ROLE admin;

-- 4 Create a database assignment\_db

CREATE DATABASE assignment\_db;

-- 5 Create a schema my\_schema

CREATE SCHEMA my\_schema;

use database ASSIGNMENT\_DB;

use SCHEMA MY\_SCHEMA;

-- 6. Create a table Preferably search for a sample employee dataset so that you have PII related columns e

CREATE TABLE assignment\_db.my\_schema.employee (

employee\_id INT,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

email VARCHAR(100),

phone\_number VARCHAR(20),

hire\_date DATE,

salary DECIMAL(10,2),

inserted\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP(), -- Automatically records insertion timestamp

elt\_by VARCHAR(50) DEFAULT 'SnowSQL CLI', -- Default application name

file\_name VARCHAR(255) -- File name used to insert data

);

-- 8. Load the file into an external and internal stage

-- 8.1 Internal Staged file here :

-- Internal Staging commands only work in CLI

-- CREATE OR REPLACE STAGE mystage;

-- USE ROLE ACCOUNTADMIN;

-- GRANT USAGE ON STAGE mystage TO ROLE admin;

-- GRANT CREATE ON STAGE mystage TO ROLE admin;

-- GRANT READ ON STAGE ASSIGNMENT\_DB.MY\_SCHEMA.mystage TO ROLE ADMIN;

-- GRANT WRITE ON STAGE ASSIGNMENT\_DB.MY\_SCHEMA.mystage TO ROLE ADMIN;

-- USE ROLE ADMIN;

-- PUT file:///users/amanpandey/Downloads/EMPP.csv @mystage;

-- 9.1 LOAD DATA FROM INTERNAL STAGE TO TABLE

COPY INTO EMPLOYEE FROM @mystage/EMPP.csv FILE\_FORMAT = (TYPE = CSV SKIP\_HEADER = 1);

select \* from employee;

-- 7. Also, create a variant version of this dataset

CREATE TABLE assignment\_db.my\_schema.employee\_variant (

employee\_data VARIANT

);

INSERT INTO assignment\_db.my\_schema.employee\_variant (employee\_data)

SELECT OBJECT\_CONSTRUCT(

'employee\_id', employee\_id,

'first\_name', first\_name,

'last\_name', last\_name,

'email', email,

'phone\_number', phone\_number,

'hire\_date', hire\_date,

'salary', salary,

'inserted\_at', inserted\_at,

'elt\_by', elt\_by,

'file\_name', file\_name

) AS employee\_data

FROM assignment\_db.my\_schema.employee;

select \* from employee\_variant;

-- 8.2 External Staging

-- 8.2 ALL EXTERNAL COMMAND WILL WORK WITH ACCOUNTADMIN ROLE

create file format my\_csv\_format

type = csv

field\_delimiter = ','

skip\_header = 1

null\_if = ('NULL', 'null')

empty\_field\_as\_null = true;

GRANT ALL PRIVILEGES ON INTEGRATION s3\_int2 TO ROLE admin;

create or replace storage integration s3\_int2

type=external\_stage storage\_provider=s3 enabled=true storage\_aws\_role\_arn='arn:aws:iam::058264207227:role/amanrole' storage\_allowed\_locations=('s3://assignmentbucket7/');

create or replace stage mystage2 STORAGE\_INTEGRATION=s3\_int2 url='s3://assignmentbucket7/EMPP.csv' file\_format=my\_csv\_format;

desc integration s3\_int2;

--WO86653\_SFCRole=2\_mQ32CjVVx07lQadUgfMzo/a5hCU=

--arn:aws:iam::339712966940:user/umol0000-s

-- 9.2 LOAD DATA FROM EXTERNAL STAGE TO TABLE

CREATE TABLE assignment\_db.my\_schema.employeeExternal (

employee\_id INT,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

email VARCHAR(100),

phone\_number VARCHAR(20),

hire\_date DATE,

salary DECIMAL(10,2),

inserted\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP(), -- Automatically records insertion timestamp

elt\_by VARCHAR(50) DEFAULT 'SnowSQL CLI', -- Default application name

file\_name VARCHAR(255) -- File name used to insert data

);

show stages;

copy into employeeExternal from @mystage2;

select \* from employeeexternal;

-- 10. parquet file staging and infer schema without uploading it to snowflake.

-- INTERNAL STAGE PARQUET FILE FIRST

-- create or replace stage mystage3;

-- PUT file:///users/amanpandey/Downloads/titanic.parquet @mystage3;

-- after internal staging

CREATE OR REPLACE FILE FORMAT my\_parquet\_format TYPE = parquet;

SELECT \*

FROM TABLE(

INFER\_SCHEMA(

LOCATION=>'@mystage3/titanic.parquet' ,FILE\_FORMAT=>'my\_parquet\_format')

);

-- 11. Run a select query on the staged parquet file without loading it to a snowflake table

SELECT t.\* FROM @mystage3 (file\_format => 'my\_parquet\_format') t;

-- 12. Add masking policy to the PII columns such that fields like email,

-- phone number, etc. show as \*\*masked\*\* to a user with the developer role.

-- If the role is PII the value of these columns should be visible

-- Masking Policy

CREATE OR REPLACE MASKING POLICY pii\_masking\_policy AS (val STRING) RETURNS STRING ->

CASE

WHEN current\_role() IN ('PII','ADMIN','ACCOUNTADMIN') THEN VAL

ELSE '\*\*masked\*\*'

END;

-- add masking policy to specified columns

ALTER TABLE employee

MODIFY COLUMN email set masking policy pii\_masking\_policy ;

ALTER TABLE employee

MODIFY COLUMN phone\_number set masking policy pii\_masking\_policy ;

GRANT USAGE ON DATABASE ASSIGNMENT\_DB TO ROLE PII;

GRANT USAGE ON DATABASE ASSIGNMENT\_DB TO ROLE DEVELOPER;

GRANT USAGE ON SCHEMA assignment\_db.my\_schema TO ROLE PII;

GRANT USAGE ON SCHEMA assignment\_db.my\_schema TO ROLE DEVELOPER;

GRANT SELECT ON ALL TABLES IN SCHEMA assignment\_db.my\_schema TO ROLE PII;

GRANT SELECT ON ALL TABLES IN SCHEMA assignment\_db.my\_schema TO ROLE DEVELOPER;

USE ROLE PII;

SELECT \* FROM ASSIGNMENT\_DB.MY\_SCHEMA.EMPLOYEE;

USE ROLE DEVELOPER;

SELECT \* FROM ASSIGNMENT\_DB.MY\_SCHEMA.EMPLOYEE;