

SNOWFLAKE ASSIGNMENT : Dev Juneja TAS 197

-- 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

```
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/devjuneja/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::058264234227:role/devrole'
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_SFCSRole=2_mQ32CjVVx07lQadUgfMzo/a5hCU=
--arn:aws:iam::339714366940: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/devjuneja/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;
```