ASSIGNMENT-Saumya Rani (TAS_207)

```
–1. Creating various roles
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';

–3. Granting various permissions
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 Creating table
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
);
```

Internal staging done on 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/saumya/Downloads/EMPP.csv @mystage;
```

```
show stages;
```

```
COPY INTO EMPLOYEE  FROM @mystage/EMPP.csv FILE_FORMAT = (TYPE = CSV
SKIP_HEADER = 1);
```

```
–Creating variant of table
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;


–External Staging file format
create file format my_csv_format
type = csv
field_delimiter = ','
skip_header = 1
null_if = ('NULL', 'null')
empty_field_as_null = true;


use role accountadmin;

–Integrating AWS bucket
create or replace storage integration s3_int2
type=external_stage storage_provider=s3 enabled=true
storage_aws_role_arn='arn:aws:iam::891377274149:role/saumyarole'
storage_allowed_locations=('s3://saumyabucket/');


create or replace stage mystage2 STORAGE_INTEGRATION=s3_int2
url='s3://saumyabucket/EMPP.csv' file_format=my_csv_format;

desc integration s3_int2;

-- iam-user arn=arn:aws:iam::058264400890:user/eifl0000-s
-- ext-id =JQ60134_SFCRole=2_sWy1KfDbCgnYK/W03twGcbxI3yQ=

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;


use role admin;

–Parquet file

CREATE OR REPLACE FILE FORMAT my_parquet_format TYPE = parquet;

SELECT *
        FROM TABLE(
             INFER_SCHEMA(
             LOCATION=>'@mystage3/titanic.parquet' ,FILE_FORMAT=>'my_parquet_format')
                   );


SELECT t.* FROM @mystage3 (file_format => 'my_parquet_format') t;


-- Masking data

-- 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;
```