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

-- Accountadmin role already exists

-- Create roles
CREATE ROLE admin;
CREATE ROLE developer;
CREATE ROLE pii;

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 we have PII related columns e
CREATE or replace 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 varchar(10),
```

```
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
);

-- 7 Also, create a variant version of this dataset
CREATE or replace TABLE assignment_db.my_schema.employee_variant (
employee_id INT,
first_name VARCHAR(50),
last_name VARCHAR(50),
email VARCHAR(100),
phone_number VARCHAR(20),
hire_date VARCHAR(10),
salary DECIMAL(10,2),
inserted_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP(), -- Automatically records
insertion timestamp
elt_by VARCHAR(50) DEFAULT 'AWS', -- Default application name
file_name VARCHAR(255) -- File name used to insert data
);


--creating a new stage;
create stage my_emp_stage;
--granting all permissions to admin role
GRANT ALL PRIVILEGES ON STAGE my_emp_stage TO ROLE admin;

-- 8. LOAD DATA FROM INTERNAL STAGE TO TABLE

-- the following command was used in CLI for internal staging
put
file:///Users/jayantasudhani/Documents/SnowflakeAssignment/emp_assignment.csv
@my_emp_stage;

--creating csv file format to read file from external stage;
create or replace file format assingment_db.my_schema.my_csv_format
type = csv
field_delimiter = ','
skip_header = 1
null_if = ('NULL', 'null')
empty_field_as_null = true;



-- creating a storage integration  s3_int2 using role accountadmin
```

```
create or replace storage integration s3_int2 type = external_stage storage_provider= s3
enabled = true storage_aws_role_arn='arn:aws:iam::637423503468:role/newemprole'
storage_allowed_locations =('s3://assingmentbucket');


--using role accountadmin
grant ownership on integration s3_int2 to role admin;

--creating  external stage for s3  using role admin  and loading to external stage;
create stage my_emp_external_stage STORAGE_INTEGRATION =s3_int2
URL='s3://assingmentbucket/emp_assignment.csv' file_format=my_csv_format;


-- loading from external stage to employee_data_external;
copy into employee_data_external from @my_emp_external_stage;


--10) for staging the parquet file user1data.parquet; in terminal

--PUT
file:///Users/jayantasudhani/Documents/SnowflakeAssignment/userdata1.parquet
@my_emp_stage;

 --creating new file type
create file format my_parquet_format TYPE =parquet;

--(11)selecting using inferschema
select * from table (INFER_SCHEMA (LOCATION =>'@my_emp_stage',FILE
                                _FORMAT=>'my_parquet_format'));


CREATE MASKING POLICY PII_masking_policy
  AS (val STRING)
  RETURNS STRING ->
  CASE
    WHEN CURRENT_ROLE() = 'DEVELOPER' THEN '**MASKED**'
    ELSE val
  END;

-- Grant USAGE privilege on the masking policy to the developer role
ALTER TABLE employee_data_internal MODIFY COLUMN EMAIL SET MASKING POLICY
PII_masking_policy;
ALTER TABLE employee_data_internal MODIFY COLUMN EMPLOYEE_ID SET MASKING
POLICY PII_masking_policy;

-- Grant SELECT privileges on the masked columns to the 'developer' role
GRANT SELECT(EMAIL) ON employee_data_internal TO ROLE developer;
```

```
GRANT SELECT(EMPLOYEE_ID) ON employee_data_internal TO ROLE developer;
```