```sql
--MASKING OF ACCOUNT TABLE IN ACCOUNT SCHEMA-----

// ROLES
SHOW ROLES;

//CREATING MASKING ROLE
USE ROLE SECURITYADMIN;

CREATE ROLE MASKING_ADMIN;

//SHOWING ROLE
SHOW ROLES;

//GRANTING ACCOUNT SCHEMA PRIVELLAGE TO MASKING ADMIN
GRANT CREATE MASKING POLICY ON SCHEMA DEVELOPMENT.ACCOUNT TO ROLE
MASKING_ADMIN;
GRANT CREATE MASKING POLICY ON SCHEMA DEVELOPMENT.APEX TO ROLE
MASKING_ADMIN;
USE ROLE ACCOUNTADMIN;

GRANT APPLY MASKING POLICY on ACCOUNT to ROLE MASKING_ADMIN;

-- grant select privilege to analyst roles
GRANT SELECT ON TABLE DEVELOPMENT.ACCOUNT.ACCOUNT TO ROLE
DATA_ANALYST;
GRANT SELECT ON TABLE DEVELOPMENT.ACCOUNT.ACCOUNT TO ROLE
DATA_SCIENTIST;
GRANT SELECT ON TABLE DEVELOPMENT.ACCOUNT.ACCOUNT TO ROLE
SF_DEVELOPER;

//GET ALL USERS
SHOW USERS;

-- assign roles to a user
GRANT ROLE DATA_ANALYST TO USER MAHENDRA;
GRANT ROLE DATA_ANALYST TO USER ADITI;
GRANT ROLE DATA_ANALYST TO USER SATISH;
GRANT ROLE DATA_ANALYST TO USER PRADEEP;
GRANT ROLE DATA_ANALYST TO USER OSAMA;

GRANT ROLE DATA_SCIENTIST TO USER SATYAJIT;
GRANT ROLE DATA_SCIENTIST TO USER ANANT;
GRANT ROLE DATA_SCIENTIST TO USER SHAKSHAM;
GRANT ROLE DATA_SCIENTIST TO USER ANVESH;
```

```sql
GRANT ROLE DATA_SCIENTIST TO USER YUKTHA;

GRANT ROLE SF_DEVELOPER TO USER SATYAJIT;
GRANT ROLE SF_DEVELOPER TO USER SHAKSHAM;
GRANT ROLE SF_DEVELOPER TO USER SATISH;
GRANT ROLE SF_DEVELOPER TO USER YUKTHA;
GRANT ROLE SF_DEVELOPER TO USER ANANT;

//WAREHOUSE
SHOW WAREHOUSES;

-- grant warehouse access to roles
GRANT USAGE ON WAREHOUSE DATA_ANALYSIS_WH TO ROLE DATA_ANALYST;
GRANT USAGE ON WAREHOUSE DATA_SCIENCE_WH TO ROLE DATA_SCIENTIST;
GRANT USAGE ON WAREHOUSE SNOWFLAKE_WH TO ROLE SF_DEVELOPER;

-- grant schema access to roles
GRANT USAGE ON SCHEMA DEVELOPMENT.ACCOUNT TO ROLE DATA_ANALYST;
GRANT USAGE ON SCHEMA DEVELOPMENT.ACCOUNT TO ROLE DATA_SCIENTIST;
GRANT USAGE ON SCHEMA DEVELOPMENT.ACCOUNT TO ROLE SF_DEVELOPER;

-- CREATE MASKING POLICY
/*
Currently, Snowflake does not support different input and output data types in a masking policy,
such as defining the masking policy to target a timestamp and return a string (e.g.
***MASKED***);
the input and output data types must match.
*/

create or replace masking policy ACCOUNT_NUMBER_MASKING_POLICY as (val VARCHAR)
returns VARCHAR ->
  case
    when current_role() in ('DATA_ANALYST', 'ACCOUNTADMIN','SF_DEVELOPER') then val
    else 'XXXXXXXXXXXXXXX'
  end;

-- APPLY MASKING POLICY TO A TABLE's NUMBER COLUMN
ALTER TABLE IF EXISTS DEVELOPMENT.ACCOUNT.ACCOUNT MODIFY COLUMN
ACCOUNT_NUMBER SET MASKING POLICY ACCOUNT_NUMBER_MASKING_POLICY;

-- SEE MASKING IN ACTION
USE ROLE DATA_SCIENTIST;

SELECT * FROM ACCOUNT;
```

```sql
USE ROLE DATA_ANALYST;

SELECT * FROM ACCOUNT;
----------------------------------------------------------------DONE---------------------------------------------------
------------------

GRANT USAGE ON DATABASE DEVELOPMENT TO ROLE MASKING_ADMIN;
GRANT USAGE ON SCHEMA DEVELOPMENT.ACCOUNT TO ROLE MASKING_ADMIN;
GRANT USAGE ON SCHEMA DEVELOPMENT.APEX TO ROLE MASKING_ADMIN;

GRANT ROLE MASKING_ADMIN TO USER KIPIMINI;

USE ROLE MASKING_ADMIN;

create or replace masking policy ACCOUNT_NAME_MASKING_POLICY as (val VARCHAR)
returns VARCHAR ->
  case
    when current_role() in ('ACCOUNTADMIN') then val
    else 'XXXXXXXXXXXXXXX'
  end;

create or replace masking policy BILLING_CITY_MASKING_POLICY as (val VARCHAR)
returns VARCHAR ->
  case
    when current_role() in ('ACCOUNTADMIN') then val
    else 'XXXXXXXXXXXXXXX'
  end;

create or replace masking policy BILLING_STATE_MASKING_POLICY as (val VARCHAR)
returns VARCHAR ->
  case
    when current_role() in ('ACCOUNTADMIN') then val
    else 'XXXXXXXXXXXXXXX'
  end;

ALTER TABLE IF EXISTS DEVELOPMENT.ACCOUNT.ACCOUNT_2 MODIFY COLUMN
ACCOUNT_NAME SET MASKING POLICY ACCOUNT_NAME_MASKING_POLICY;
ALTER TABLE IF EXISTS DEVELOPMENT.ACCOUNT.ACCOUNT_2 MODIFY COLUMN
BILLING_STATE_PROVINCE SET MASKING POLICY BILLING_STATE_MASKING_POLICY;
ALTER TABLE IF EXISTS DEVELOPMENT.ACCOUNT.ACCOUNT_2 MODIFY COLUMN
BILLING_CITY SET MASKING POLICY BILLING_CITY_MASKING_POLICY;
//ACCOUNTADMIN
```

```
//GRANT SELECT ON TABLE

GRANT SELECT ON TABLE DEVELOPMENT.ACCOUNT.ACCOUNT_2 TO ROLE
DATA_SCIENTIST;

-- SEE MASKING IN ACTION
USE ROLE DATA_SCIENTIST;

SELECT * FROM ACCOUNT_2;

USE ROLE ACCOUNTADMIN;

SELECT * FROM ACCOUNT_2;

---------------------------------------------------------ACCOUNT_2 DONE
---------------------------------------------------------
```