

Masking / Tokenization

In this experiment, we will learn to use Snowflake's data masking functions.

1. Create a sample data and table with ID, Social Security number, Age and Credit Card

```
USE ROLE ACCOUNTADMIN;
```

```
CREATE OR REPLACE DATABASE SNOWTEST;  
CREATE OR REPLACE SCHEMA SNOWTEST.DATA_CLASS;  
CREATE OR REPLACE TABLE SNOWTEST.DATA_CLASS.SAMPLE_DATA_TBL(  
    ID VARCHAR(10)  
    , SSN VARCHAR(11)  
    , AGE NUMERIC  
    , CREDIT_CARD VARCHAR(19)  
);
```

2. Let's enter some fake sensitive data

```
INSERT INTO SNOWTEST.DATA_CLASS.SAMPLE_DATA_TBL VALUES  
( 'A0000001', '234-45-6747', 24, '4053-0495-0394-0494' ), ( 'A0000002', '284-85-  
6127', 22, '4653-0495-0394-0494' ), ( 'A0000003', '235-45-2967', 20, '4053-0755-0394-  
0494' ), ( 'A0000004', '284-85-0787', 22, '4653-0495-6885-0494' ), ( 'A0000005', '235-45-  
9857', 20, '5356-0755-0394-0494' );
```

3. Here is the table, but we have sensitive data like SSN and Credit Card information

```
SELECT * FROM SNOWTEST.DATA_CLASS.SAMPLE_DATA_TBL;
```

4. We want to mask the social security number so it only shows the last 4 digits

```
SELECT CONCAT('XXX-XX-', RIGHT(SSN, 4)) as SSN  
FROM SNOWTEST.DATA_CLASS.SAMPLE_DATA_TBL;
```

5. Create a masking role and analyst

```
CREATE OR REPLACE ROLE masking_admin;  
CREATE OR REPLACE ROLE analyst;
```

```
GRANT ROLE masking_admin TO USER <user>;
```

```
GRANT ROLE analyst TO USER <user>;
```

```
GRANT ALL ON WAREHOUSE COMPUTE_WH TO ROLE analyst;  
GRANT ALL ON DATABASE SNOWTEST TO ROLE analyst;  
GRANT ALL ON SCHEMA SNOWTEST.DATA_CLASS TO ROLE analyst;  
GRANT ALL ON TABLE SNOWTEST.DATA_CLASS.SAMPLE_DATA_TBL TO  
ROLE analyst;
```

```
GRANT ALL ON WAREHOUSE COMPUTE_WH TO ROLE masking_admin;  
GRANT ALL ON DATABASE SNOWTEST TO ROLE masking_admin;  
GRANT ALL ON SCHEMA SNOWTEST.DATA_CLASS TO ROLE masking_admin;  
GRANT ALL ON TABLE SNOWTEST.DATA_CLASS.SAMPLE_DATA_TBL TO  
ROLE masking_admin;
```

6. Create a marketing policy and apply to masking role

```
GRANT CREATE MASKING POLICY on SCHEMA SNOWTEST.DATA_CLASS to  
ROLE masking_admin;  
GRANT APPLY MASKING POLICY on ACCOUNT to ROLE masking_admin;
```

7. Create masking policy for SSN number

```
CREATE OR REPLACE MASKING POLICY ssn_mask AS (val string) RETURNS  
string ->  
CASE  
  WHEN CURRENT_ROLE() = 'ANALYST' THEN val  
  ELSE CONCAT('XXX-XX-',RIGHT(val,4))  
END;
```

8. Allow role to set and unset Social Security mask

```
GRANT APPLY ON MASKING POLICY ssn_mask to ROLE masking_admin;
```

9. Apply masking policy to the SSN column

```
USE ROLE masking_admin;
```

```
ALTER TABLE IF EXISTS SNOWTEST.DATA_CLASS.SAMPLE_DATA_TBL  
MODIFY COLUMN SSN SET MASKING POLICY ssn_mask;
```

10. Using the ANALYST role

```
USE ROLE analyst;  
SELECT CURRENT_ROLE();
```

```
SELECT * FROM SNOWTEST.DATA_CLASS.SAMPLE_DATA_TBL; -- should see  
plain text value
```

11. Using the ACCOUNTADMIN role

```
USE ROLE ACCOUNTADMIN;  
SELECT * FROM SNOWTEST.DATA_CLASS.SAMPLE_DATA_TBL; -- should see  
full data mask
```

12. Clear resources

```
USE ROLE ACCOUNTADMIN;  
DROP DATABASE SNOWTEST;  
DROP ROLE MASKING_ADMIN;  
DROP ROLE ANALYST;
```

Test Your Skills

Create three new roles, administration, enrollment and accounting
Create a masking policy so that SSN and Credit Card is masked for administration,
credit card is masked for enrollment and SSN is masked for finance