Row Access Policy

In this experiment, we will learn to use Snowflake's data row access policies.

1. Create new database

```
USE ROLE ACCOUNTADMIN;

CREATE OR REPLACE DATABASE SNOWTEST;
CREATE OR REPLACE SCHEMA SNOWTEST.PUBLIC;
```

2. Create roles and apply to the user

```
CREATE OR REPLACE ROLE ROLE1;
CREATE OR REPLACE ROLE ROLE2;
CREATE OR REPLACE ROLE ROLE3;
CREATE OR REPLACE ROLE SUPERADMIN;

GRANT ROLE SUPERADMIN TO USER <user>;
GRANT ROLE ROLE1 TO USER <user>;
GRANT ROLE ROLE2 TO USER <user>;
GRANT ROLE ROLE3 TO USER <user>;
```

3. We will create a new table of start, social security, age, CC

```
CREATE OR REPLACE TABLE SNOWTEST.PUBLIC.SAMPLE_DATA_TBL(
    STATE VARCHAR(2)
    , SSN VARCHAR(11)
    , AGE NUMERIC
    , CC VARCHAR(19)
);

INSERT INTO SNOWTEST.PUBLIC.SAMPLE_DATA_TBL VALUES ('KS','234-45-6477',27,'4053 0495 0394 0494'), ('TX','234-85-6477',67,'4653 0495 0394 0494'), ('TX','235-45-6477',44,'4053 0755 0394 0494'), ('MD','234-85-6477',81,'4873 0495 0394 4094'), ('CA','234-85-0877',18,'4653 0495 0084 0494');
```

4. Create a mapping table for row-level access

```
CREATE OR REPLACE TABLE SNOWTEST.PUBLIC.MAPPING (
    ROLE_ENTITLED varchar, STATE varchar
);
```

5. Insert mapping values

```
INSERT INTO SNOWTEST.PUBLIC.MAPPING VALUES
('ROLE1','TX'),('ROLE1','KS'),('ROLE1','MD'),('ROLE1','CA'),('ROLE1','TX'),('ROLE1','
NA'),('ROLE3','TX');
```

6. Create row access policy

```
CREATE ROW ACCESS POLICY SNOWTEST.PUBLIC.TEST_POLICY AS
(state_filter varchar) RETURNS BOOLEAN ->  -- This will provide a true or false on
the row based on the mapping table
CURRENT_ROLE() = 'SUPERADMIN'
OR EXISTS (
   SELECT 1 FROM SNOWTEST.PUBLIC.MAPPING
   WHERE STATE = state_filter
   AND ROLE_ENTITLED = CURRENT_ROLE());
```

7. Apply the row access policy on the created table

```
ALTER TABLE SNOWTEST.PUBLIC.SAMPLE_DATA_TBL ADD ROW ACCESS
POLICY SNOWTEST.PUBLIC.TEST_POLICY ON (STATE);
```

8. Grant permissions to all of the roles

```
USE ROLE ACCOUNTADMIN;
GRANT SELECT ON SNOWTEST.PUBLIC.SAMPLE_DATA_TBL TO ROLE
ROLE1;
GRANT SELECT ON SNOWTEST.PUBLIC.SAMPLE_DATA_TBL TO ROLE
ROLE2;
GRANT SELECT ON SNOWTEST.PUBLIC.SAMPLE_DATA_TBL TO ROLE
ROLE3;
GRANT SELECT ON SNOWTEST.PUBLIC.SAMPLE_DATA_TBL TO ROLE
SUPERADMIN;

GRANT ALL ON WAREHOUSE COMPUTE_WH TO ROLE ROLE1;
GRANT ALL ON DATABASE SNOWTEST TO ROLE ROLE1;
GRANT ALL ON SCHEMA SNOWTEST.PUBLIC TO ROLE ROLE1;

GRANT ALL ON WAREHOUSE COMPUTE_WH TO ROLE ROLE2;
GRANT ALL ON DATABASE SNOWTEST TO ROLE ROLE2;
GRANT ALL ON SCHEMA SNOWTEST.PUBLIC TO ROLE ROLE2;

GRANT ALL ON WAREHOUSE COMPUTE_WH TO ROLE ROLE3;
GRANT ALL ON DATABASE SNOWTEST TO ROLE ROLE3;
```

```
GRANT ALL ON SCHEMA SNOWTEST.PUBLIC TO ROLE ROLE3;

GRANT ALL ON WAREHOUSE COMPUTE_WH TO ROLE SUPERADMIN;
GRANT ALL ON DATABASE SNOWTEST TO ROLE SUPERADMIN;
GRANT ALL ON SCHEMA SNOWTEST.PUBLIC TO ROLE SUPERADMIN;
```

9. Test the access by role

```
USE ROLE SUPERADMIN;
SELECT CURRENT_ROLE();
SELECT * FROM SNOWTEST.PUBLIC.SAMPLE_DATA_TBL;

USE ROLE ROLE1;
SELECT CURRENT_ROLE();
SELECT * FROM SNOWTEST.PUBLIC.SAMPLE_DATA_TBL;

USE ROLE ROLE2;
SELECT CURRENT_ROLE();
SELECT * FROM SNOWTEST.PUBLIC.SAMPLE_DATA_TBL;

USE ROLE ROLE3;
SELECT CURRENT_ROLE();
SELECT * FROM SNOWTEST.PUBLIC.SAMPLE_DATA_TBL;

USE ROLE ACCOUNTADMIN;
```

10 Clear resources

```
USE ROLE ACCOUNTADMIN;

DROP DATABASE SNOWTEST;
DROP ROLE ROLE1;
DROP ROLE ROLE2;
DROP ROLE ROLE3;
DROP ROLE SUPERADMIN;
```

Test Your Skills

Use the below

```
CREATE OR REPLACE DATABASE SNOWTEST;
CREATE OR REPLACE SCHEMA SNOWTEST.PUBLIC;

CREATE OR REPLACE TABLE SNOWTEST.PUBLIC.SAMPLE_DATA_TBL(
```

```
    ID VARCHAR(5)
    , ANIMAL VARCHAR(10)
    , PASSWORD VARCHAR(10)
    , REGION VARCHAR(2)
);

INSERT INTO SNOWTEST.PUBLIC.SAMPLE_DATA_TBL
VALUES
('A0001','Dog','DDKe43##@','N'),('A0002','Cat','24454##@','S'),('A0003','Mouse','334
452552@','N'),('A0004','Pig','JILL12345','N'),('A0005','Dog','PASS321','W'),('A0006','
Dog','LMONKEY','E'),('A0007','Horse','JILL12345','S'),('A0008','Cat','whisker@$','W'),
('A0009','Dog','Melon','N')
;

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

Using row access policies and masking, make it such that:

- The password is masked from everyone
- TeamA can only see regions 'N' and 'S'
- TeamB can see everything
- TeamC can only see the region W and has everything by ID masked
- TeamD can only see dogs