Assignment -7

1. Create and Manage Users and Roles

Step 1: Create Roles

CREATE ROLE admin;
CREATE ROLE analyst;
CREATE ROLE viewer;

ADMIN	0	0	0
ANALYST	0	0	0
VIEWER	0	0	0

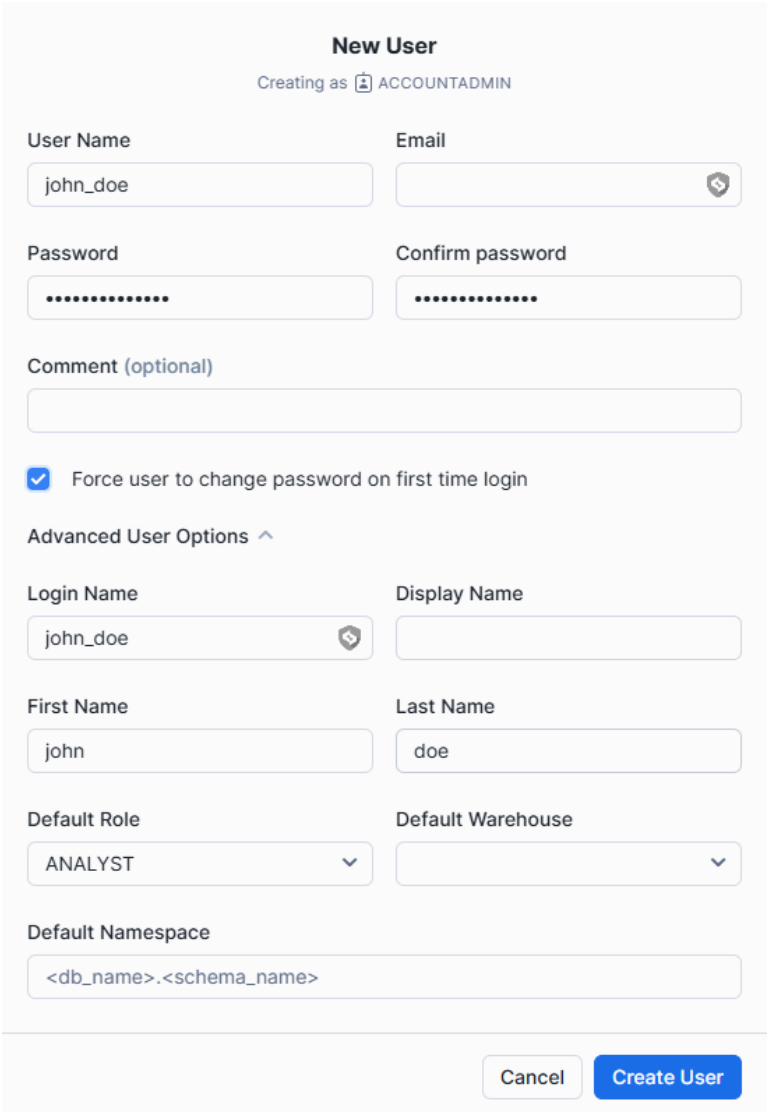
Step 2: Assign Privileges to Roles

GRANT ALL PRIVILEGES ON DATABASE sales_db TO ROLE admin;
GRANT SELECT ON ALL TABLES IN SCHEMA sales_db.public TO ROLE analyst;
GRANT USAGE ON WAREHOUSE compute_wh TO ROLE viewer;

```
--privileges
GRANT ALL PRIVILEGES ON DATABASE SALES_DB TO ROLE ADMIN;
GRANT SELECT ON ALL TABLES IN SCHEMA SALES_DB.PUBLIC TO ROLE ANALYST;
GRANT USAGE ON WAREHOUSE COMPUTE_WH TO ROLE VIEWER;
```

Step 3: Create Users

CREATE USER john_doe
PASSWORD = 'Password123!'
DEFAULT_ROLE = analyst
MUST_CHANGE_PASSWORD = TRUE;



Step 4: Assign Roles to Users

GRANT ROLE admin TO USER admin_user;
GRANT ROLE analyst TO USER john_doe;
GRANT ROLE viewer TO USER jane_doe;

```
GRANT ROLE ADMIN TO USER SHAKSHILIKHIA;
GRANT ROLE ANALYST TO USER JOHN_DOE;
GRANT ROLE VIEWER TO USER JOHN_DOE;
```

Granted Roles

SHAKSHILIKHIA has been granted 3 roles

NAME ↑
ACCOUNTADMIN
ADMIN
ORGADMIN

Granted Roles

JOHN_DOE has been granted 2 roles

NAME ↑
ANALYST
VIEWER

2. Configure Network Policies

Step 1: Create a Network Policy

CREATE NETWORK POLICY secure_policy
ALLOWED_IP_LIST = ('192.168.1.0/24', '203.0.113.0/32')
BLOCKED_IP_LIST = ('10.0.0.0/8');

```
CREATE NETWORK POLICY secure_policy
ALLOWED_IP_LIST = ('192.168.1.0/24', '203.0.113.0/32')
BLOCKED_IP_LIST = ('10.0.0.0/8');
```

Results Chart

	status
1	Network policy SECURE_POLICY is created.

Step 2: Apply Network Policy

ALTER ACCOUNT SET NETWORK_POLICY = secure_policy;

```
ALTER ACCOUNT SET NETWORK_POLICY=SECURE_POLICY;
```

3. Implement Data Masking on Sensitive Data

Step 1: Create a Table with Sensitive Data

CREATE TABLE employees (
id INT,
name STRING,
ssn STRING
);
INSERT INTO employees VALUES (1, 'Alice', '123-45-6789');

```
CREATE TABLE employees (
    id INT,
    name STRING,
    ssn STRING
);
INSERT INTO employees VALUES (1, 'Alice', '123-45-6789');
```

Step 2: Define a Masking Policy

CREATE MASKING POLICY ssn_mask AS (val STRING) RETURNS STRING -
>
CASE
WHEN CURRENT_ROLE() = 'admin' THEN val
ELSE 'XXX-XX-XXXX'
END;

```
CREATE MASKING POLICY ssn_mask AS (val STRING) RETURNS STRING ->
CASE
    WHEN CURRENT_ROLE() = 'ADMIN' THEN val
    ELSE 'XXX-XX-XXXX'
END;
```

Results | Chart

	status
1	Masking policy SSN_MASK successfully created.

Step 3: Apply the Masking Policy

ALTER TABLE employees MODIFY COLUMN ssn SET MASKING POLICY ssn_mask;

```
ALTER TABLE employees MODIFY COLUMN ssn SET MASKING POLICY ssn_mask;
```

Step 4: Test the Masking Policy

-- With admin role:
SET ROLE admin;
*SELECT * FROM employees;*

```
USE ROLE admin;
SELECT * FROM employees;
```

# ID	A NAME	A SSN
1	Alice	123-45-6789

-- With analyst role:
SET ROLE analyst;
*SELECT * FROM employees;*

```
USE ROLE analyst;
SELECT * FROM employees;
```

# ID	A NAME	A SSN
1	Alice	XXX-XX-XXXX

But for both of these we have to grant usage on database and schema for both respective roles as without it the employees table won't be accessible.

And for analyst role, we first need to even grant it to the user.