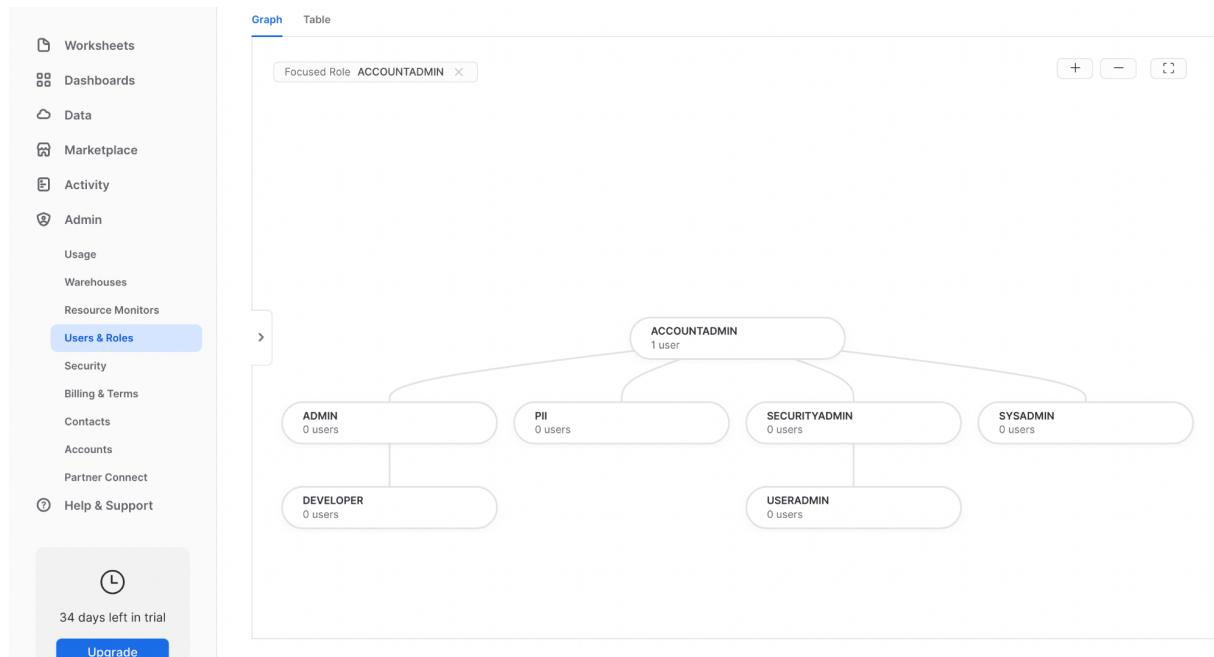


1. Created roles Admin , Developer , PII as per the mentioned hierarchy.(Account Admin already existed in Snowflake)



2. Created an Msized warehouse named assignment\_wh and used it for all the Queries.

FACTC	OWNER	CREATED
	ACCOUNTADMIN	2 weeks ago

The screenshot shows the Snowflake interface with the left sidebar expanded. The 'Warehouses' section is selected, highlighted with a blue background. The main content area displays a table titled '1 Warehouse' with one row. The table columns are: NAME, STATUS, SIZE, CLUSTERS, RUNNING, QUEUED, QAS (SCALE FACTC), OWNER, and CREATED. The single row shows: ASSIGN..., Started, Medium, 1 - 1 (1 active), 0, 0, Disabled, ACCOUNTADMIN, just now, and three dots. At the top right of the table are buttons for Search, Status All, Size All, and a trash can icon. Below the table, there's a message about a 17-day trial and a blue 'Upgrade' button.

NAME	STATUS	SIZE	CLUSTERS	RUNNING	QUEUED	QAS (SCALE FACTC)	OWNER	CREATED
ASSIGN...	Started	Medium	1 - 1 (1 active)	0	0	Disabled	ACCOUNTADMIN	just now

17 days left in trial

Upgrade

### 3. Switched to the admin role.

The screenshot shows the Snowflake interface with the left sidebar expanded. The 'Roles' section is selected, highlighted with a blue background. The main content area displays a table titled '1 Roles' with one row. The table columns are: NAME, STATUS, SIZE, CLUSTERS, RUNNING, QUEUED, QAS (SCALE FACTC), OWNER, and CREATED. The single row shows: roles, Enabled, Small, 1 - 1 (1 active), 0, 0, Disabled, ACCOUNTADMIN, 2 minutes ago, and three dots. A dropdown menu is open over the 'roles' entry, listing five options: ACCOUNTADMIN (selected, indicated by a checkmark), ORGADMIN, ADMIN, DEVELOPER, PII, and PUBLIC. At the top right of the table are buttons for Search, Status All, Size All, and a trash can icon. Below the table, there's a message about a 17-day trial and a blue 'Upgrade' button.

NAME	STATUS	SIZE	CLUSTERS	RUNNING	QUEUED	QAS (SCALE FACTC)	OWNER	CREATED
roles	Enabled	Small	1 - 1 (1 active)	0	0	Disabled	ACCOUNTADMIN	2 minutes ago

### 4. Created a database assignment\_db

The screenshot shows the Snowflake UI interface. On the left, there's a sidebar with 'Worksheets' and 'Databases' tabs, where 'Databases' is selected. Below it, there are sections for 'Pinned (0)', 'No pinned objects', and a search bar for 'All Objects'. A tree view shows databases: 'ASSIGNMENT\_DB', 'SAMPLE\_DATABASE', 'SNOWFLAKE', and 'SNOWFLAKE\_SAMPLE\_DATA'. The main area is titled 'ASSIGNMENT\_DB.PUBLIC'. It contains a code editor with the following SQL script:

```

1 create or replace database assignment_db;
2 grant usage on database assignment_db to role admin;
3

```

Below the editor is a results panel with tabs for 'Objects', 'Editor', 'Results', and 'Chart'. The 'Results' tab is selected, showing a single row with status 'Statement executed successfully.'.

On the right, there's a 'Query Details' panel with metrics: Query duration (43ms), Rows (1), and a status bar indicating 100% filled.

## 5. Loading data through external staging

The screenshot shows the AWS Lambda function configuration page. In the 'Function Policy' section, there is a large JSON block representing a policy document:

```

{
    "Id": "Policy1665674245003",
    "Statement": [
        {
            "Sid": "Stmt1665674243203",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::snowflake-ass/*"
        }
    ]
}

```

At the bottom of the page, there's an 'Object Ownership' section with a 'Info' link. It says: 'Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLS). Object ownership determines who can specify access to objects.' There are 'Edit' and 'Feedback' buttons, and a note about finding language selection in Unified Settings.

Access control list (ACL)		
Grant basic read/write permissions to other AWS accounts. <a href="#">Learn more</a>		
<p><span style="color: #337ab7;">i</span> <b>The console displays combined access grants for duplicate grantees</b> To see the full list of ACLs, use the Amazon S3 REST API, AWS CLI, or AWS SDKs.</p>		
<p><span style="color: red;">⚠</span> <b>AWS doesn't recommend granting access to the Everyone grantee</b> Anyone in the world can access the objects in this bucket. <a href="#">Learn more</a></p>		
Grantee	Objects	Bucket ACL
Bucket owner (your AWS account) Canonical ID: <a href="#">c124f76c70ed711f474ee584e87c733a11487bf36e1925bb08cb697e11c1b557</a>	List, Write	Read, Write
Everyone (public access) Group: <a href="#">http://acs.amazonaws.com/groups/global/AllUsers</a>	<span style="color: red;">⚠</span> List	<span style="color: red;">⚠</span> Read
Authenticated users group (anyone with an AWS account) Group: <a href="#">http://acs.amazonaws.com/groups/global/AuthenticatedUsers</a>	-	-
S3 log delivery group Group: <a href="#">http://acs.amazonaws.com/groups/s3LogDelivery</a>	-	-

## Code :

----- Task 1 : Creating roles

```
use role securityadmin;
create or replace role admin;
grant role admin to role accountadmin;
create or replace role PII;
grant role PII to role accountadmin;
create or replace role developer;
grant role developer to role admin;
```

----- Granting permissions to admin role for creating database and schemas

```
use role securityadmin;
```

```
grant role admin to user naresh;
```

```
grant usage on warehouse assignment_wh to role admin;
```

```
grant operate on warehouse assignment_wh to role admin;
--alter warehouse "assignment_wh" resume;
```

----- Task 3 , 4 ,5 Creating database and schema

```
use role sysadmin;
grant create database on account to role admin;
```

```
use role admin;
use warehouse assignment_wh;
create or replace database assignment_db;
use database assignment_db;
create or replace schema my_schema;
use schema my_schema;
```

---- Task 6 Creating tables for internal\_staging and external\_staging

```
create or replace table employees_internal(
```

```
    EMPLOYEE_ID      integer PRIMARY KEY,
    FIRST_NAME string,
    LAST_NAME string,
    EMAIL string,
    PHONE_NUMBER string,
    HIRE_DATE string,
    JOB_ID string,
    SALARY integer,
    MANAGER_ID integer,
    DEPARTMENT_ID integer,
    file_name string,
    elt_by string,
    elt_ts timestamp
);
```

```
create or replace table employees_external(
```

```
    EMPLOYEE_ID      integer PRIMARY KEY,
    FIRST_NAME string,
    LAST_NAME string,
    EMAIL string,
    PHONE_NUMBER string,
    HIRE_DATE date,
    JOB_ID string,
    SALARY integer,
    MANAGER_ID integer,
    DEPARTMENT_ID integer,
    file_name string,
    elt_by string,
    elt_ts timestamp
);
```

----- Task 7 Creating stages for internal and external loading

```
create or replace stage internal_stage file_format='r_csv';
```

```
create or replace stage external_stage
```

----- Task 8 and 9

--i)Loading data from employee.csv through internal staging

--1.Downloaded employee.csv file into local

```
put file:///Users/naresh/Desktop/python/employee.csv @internal_stage;
```

```
create or replace file format r_csv
```

```
TYPE = "CSV"
```

```
COMPRESSION = "GZIP"
```

```
FILE_EXTENSION= 'csv.gz'
```

```
SKIP_HEADER = 1
```

```
ERROR_ON_COLUMN_COUNT_MISMATCH=FALSE
```

```
EMPTY_FIELD_AS_NULL = TRUE;
```

```
copy into employees_internal from
```

```
(SELECT $1,$2,$3,$4,$5,$6,$7,$8,$9,$10,
```

```
to_varchar('employee.csv'),
```

```
to_varchar('local'),
```

```
CURRENT_TIMESTAMP from @internal_stage)
```

```
file_format = (format_name = 'r_csv');
```

```
select * from employees_internal;
```

--ii)Loading data from employee.csv through external staging

--1.Uploaded employee.csv file to aws S3 Bucket

--2.Edited bucket policy

```
-- {
```

```
--   "Version": "2012-10-17",
```

```
--   "Id": "Policy1665674245003",
```

```
--   "Statement": [
```

```
--     {
```

```
--       "Sid": "Stmt1665674243203",
```

```
--       "Effect": "Allow",
```

```
--       "Principal": "*",
```

```
--       "Action": "s3:GetObject",
```

```
--       "Resource": "arn:aws:s3:::snowflake-ass/*"
```

```
--     }
```

```
--   ]
```

```
-- }
```

```
create or replace stage external_stage url='s3://snowflake-ass/';
```

```
create or replace table employees_external(
```

```
    EMPLOYEE_ID      integer PRIMARY KEY,
```

```
    FIRST_NAME string,
```

```
    LAST_NAME string,
```

```
    EMAIL string,
```

```

PHONE_NUMBER string,
HIRE_DATE string,
JOB_ID string,
SALARY integer,
MANAGER_ID integer,
DEPARTMENT_ID integer
);

copy into employees_external from
(SELECT $1,$2,$3,$4,$5,$6,$7,$8,$9,$10,
to_varchar('employees.csv'),
to_varchar('AWS S3'),
CURRENT_TIMESTAMP from @external_stage)
pattern='.*.csv'
file_format = (type = csv field_delimiter = ',' skip_header = 1);

```

----- Task 10 Upload any unrelated parquet file to the stage location and infer the schema of the file.

```

create or replace file format parquet_format type = 'parquet';
create or replace stage parquet_stage file_format = parquet_format;
put file:///Users/naresh/Downloads/snowflake_ass.parquet @parquet_stage;

```

----- Task 11 Select query

```
select $1:birthdate from @parquet_stage/snowflake_ass.parquet;
```

----- Task 12 Creating Masking policy

--i) Granting privileges for developer and PII roles

```

use role securityadmin;
grant usage on database assignment_db to role developer;
grant usage on schema assignment_db.my_schema to role developer;

```

```
grant usage on database assignment_db to role PII;
```

```
grant usage on schema assignment_db.my_schema to role PII;
```

```
grant select on all tables in schema assignment_db.my_schema to role developer;
grant select on all tables in schema assignment_db.my_schema to role PII;
```

```
grant usage on warehouse assignment_wh to role PII;
```

```
grant operate on warehouse assignment_wh to role PII;
```

--ii) Displaying table output from developer role

```

use role PII;
show databases;
show tables;
select * from my_schema.employees_internal;
```

```
select * from my_schema.employees_external;
--iii) Creating masking policy for employees_internal table
use role admin;
use schema my_schema;
create or replace masking policy PII_masking_email as (email string) returns string ->
    case
        when current_role() in ('PII') then email
        else '*****'
    end;

create or replace masking policy PII_masking_phone_number as (phone_number string) returns string ->
    case
        when current_role() in ('PII') then phone_number
        else '*****'
    end;
--iv) set masking ploicy for columns of employees_internal table
alter table if exists my_schema.employees_internal modify column email set masking policy
PII_masking_email;
alter table if exists my_schema.employees_internal modify column phone_number set masking
policy PII_masking_phone_number;

--v ) Displaying the output from different roles
use role PII;
select * from employees_internal;
use role developer;
select * from employees_internal;
```

app.snowflake.com/us-east-1/zvb41915/w5CBjkeI43HS#query

snowsql\_lass

Worksheets Databases

Pinned (0)

No pinned objects

All Objects

ASSIGNMENT\_DB

SNOWFLAKE\_SAMPLE\_DB

```

127 --ii) Displaying table output from developer role
128 use role PII;
129 show databases;
130 show tables;
131
132 | select * from my_schema.employees_internal;
133 | select * from my_schema.employees_external;
134

```

Objects Editor Results Chart

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID
2	199	Douglas	Grant	DGRANT	650.507.9844	13-JAN-08
3	200	Jennifer	Whalen	JWHALEN	515.123.4444	17-SEP-03
4	201	Michael	Hartstein	MHARTSTE	515.123.5555	17-FEB-04
5	202	Pat	Fay	PFAY	603.123.6666	17-AUG-05
6	203	Susan	Mavris	SMAVRIS	515.123.7777	07-JUN-02
7	204	Hermann	Baer	HBAER	515.123.8888	07-JUN-02
8	205	Shelley	Higgins	SHIGGINS	515.123.8080	07-JUN-02
9	206	William	Gietz	WGIEWTZ	515.123.8181	07-JUN-02
10	100	Steven	King	SKING	515.123.4567	17-JUN-03
11	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-05
12	102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-01
13	103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-06
14	104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-07
15	105	David	Austin	DAUSTIN	590.423.4569	25-JUN-05

Query Details

Query duration 219ms

Rows 50

EMPLOYEE\_ID 123

FIRST\_NAME Aa

Michael 2

Steven 2

Alexander 2

+ 42 more

LAST\_NAME Aa

100% filled

app.snowflake.com/us-east-1/zvb41915/w5CBjkeI43HS#query

snowsql\_lass

Worksheets Databases

Pinned (0)

No pinned objects

All Objects

ASSIGNMENT\_DB

SNOWFLAKE\_SAMPLE\_DB

```

148 --iv) set masking policy for columns of employees_internal table
149 alter table if exists my_schema.employees_internal modify column email set masking policy PII_masking_email;
150 alter table if exists my_schema.employees_internal modify column phone_number set masking policy PII_masking_phone_number;
151
152 --v) Displaying the output from different roles
153 use role PII;
154 select * from employees_internal;
155 use role developer;
156 select * from employees_internal;
157

```

Objects Editor Results Chart

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID
1	198	Donald	OConnell	*****	21-JUN-07	SH_CLERK
2	199	Douglas	Grant	*****	13-JAN-08	SH_CLERK
3	200	Jennifer	Whalen	*****	17-SEP-03	AD_ASST
4	201	Michael	Hartstein	*****	17-FEB-04	MK_MAN
5	202	Pat	Fay	*****	17-AUG-05	MK_REP
6	203	Susan	Mavris	*****	07-JUN-02	HR REP
7	204	Hermann	Baer	*****	07-JUN-02	PR REP
8	205	Shelley	Higgins	*****	07-JUN-02	AC_MGR
9	206	William	Gietz	*****	07-JUN-02	AC_ACCO
10	100	Steven	King	*****	17-JUN-03	AD_PRES
11	101	Neena	Kochhar	*****	21-SEP-05	AD_VP
12	102	Lex	De Haan	*****	13-JAN-01	AD_VP
13	103	Alexander	Hunold	*****	03-JAN-06	IT_PROG

Query Details

Query duration 842ms

Rows 50

EMPLOYEE\_ID 123

FIRST\_NAME Aa

Michael 2

Steven 2

Alexander 2

+ 42 more

LAST\_NAME Aa