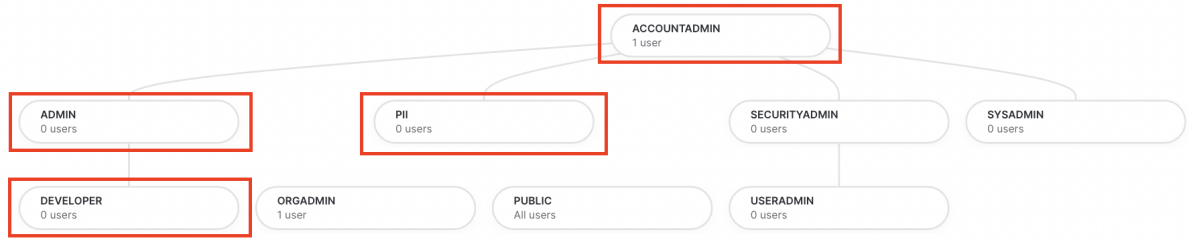
Snowflake Assignment

Name: Sahil Jaiswal

TASK 1: Create roles as per the below-mentioned hierarchy. ACCOUNTADMIN already exists in Snowflake.




```
/* ------------------------- STEP 1 ------------------------- */

-- Create a role "ADMIN" and Grant the role to "ACCOUNTADMIN"
create or replace ADMIN;
grant role ADMIN to role ACCOUNTADMIN;

-- Create a role "DEVELOPER" and Grant the role to "ADMIN"
create or replace role DEVELOPER;
grant role DEVELOPER to role ADMIN;

-- Create a role "PII" and Grant the role to "ACCOUNTADMIN"
create or replace PII;
grant role PII to role ACCOUNTADMIN;
```

TASK 2:Create an M-sized warehouse named assignment_wh and use it for all the queries.

```
/* ------------------------ STEP 2 ------------------------ */

-- Create a M-sized warehouse named assignment_wh

create or replace warehouse assignment_wh with
warehouse_size = MEDIUM
auto_suspend = 180
auto_resume = true
initially_suspended=true;
```

TASK 3: Switch to the ‘ADMIN’ role.

```
/* ------------------------ STEP 3 ------------------------ */

-- Switch to the ADMIN role.

grant create database on account to role ADMIN;
grant usage on warehouse assignment_wh to role admin;
show grants to role ADMIN;

use role ADMIN;
```

TASK 4: Create a database ‘assignment_db’.
TASK 5: Create a schema ‘my_schema’.

```
/* ------------------------- STEP 4 ------------------------- */

-- Create a database 'assignment_db'

create or replace database assignment_db;

/* ------------------------- STEP 5 ------------------------- */

-- Create a schema 'my_schema'

create or replace schema my_schema;
```

TASK 6:Create a table using any sample csv.

Using a transient table to prevent it from going into fail-safe every time because the table is modified many times.

```
-- Table for internal stage
    create or replace TRANSIENT table people_internal (
        s_num numeric,
        user_id text,
        first_name string,
        last_name string,
        sex varchar(10),
        email string,
        phone string,
        dob date,
        job_title string,
        elt_filename string,
        elt_ts_timestamp timestamp,
        elt_by_app string
    );

-- Table for external stage
    create or replace table people_external like people_internal;
```

TASK 8: Load the file into an external and internal stage.

- **Internal stage:**

  Use PUT command to load the files into the internal table stage.

  put file:///Users/sahil/Downloads/people-1000000.csv
  @assignment_db.my_schema.%people_internal;

```
sahiljais#COMPUTE_WH@ASSIGNMENT_DB.MY_SCHEMA>put file:///Users/sahil/Downloads/people-1000000.csv
                                             @assignment_db.my_schema.%people_internal;
+--------------------+----------------------+-------------+-------------+--------------------+--------------------+----------+---------+
| source             | target               | source_size | target_size | source_compression | target_compression | status   | message |
|--------------------+----------------------+-------------+-------------+--------------------+--------------------+----------+---------|
| people-1000000.csv | people-1000000.csv.gz |   116989639 |    50527984 | NONE               | GZIP               | UPLOADED |         |
+--------------------+----------------------+-------------+-------------+--------------------+--------------------+----------+---------+
1 Row(s) produced. Time Elapsed: 69.349s
```

- **External stage:**

  Using AWS S3 as the external staging area.

  Step 1: Create an integration object

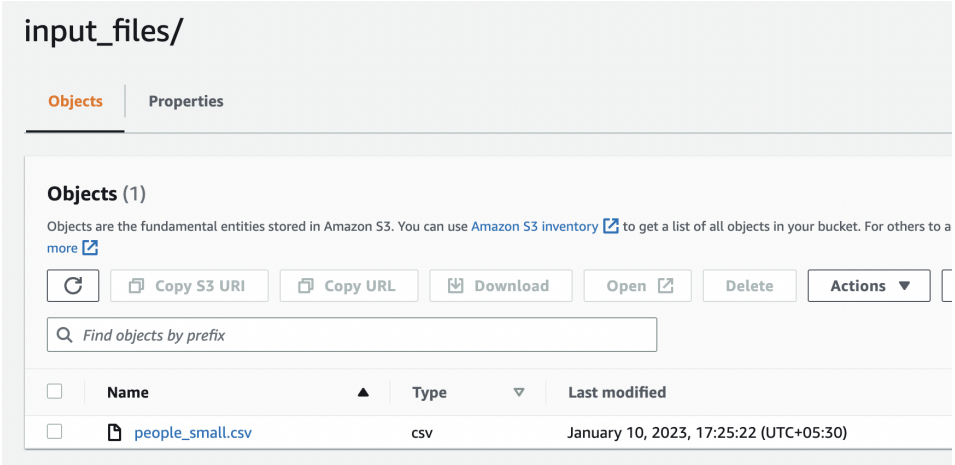
```
-- Grant privileges

    grant create integration on account to role ADMIN;

-- create an intergration object

    create or replace storage integration integrate_s3
    type = external_stage
    storage_provider = s3
    enabled = true
    storage_aws_role_arn = 'arn:aws:iam::969703917248:role/snowflake_role'
    storage_allowed_locations = ('s3://snowflakeassignment0901/input_files/');

    desc integration integrate_s3;
```



Step 2: Create an external stage for s3 bucket

```
-- Create a file format option

    create or replace file format assignment_db.my_schema.my_csv_format
        type = csv
        field_optionally_enclosed_by='"'
        record_delimiter = '\n'
        field_delimiter = ','
        skip_header = 1
        null_if = ('NULL', 'null')
        empty_field_as_null = true;

    desc file format assignment_db.my_schema.my_csv_format;

-- Create an external stage for s3 bucket

    create or replace stage s3_stage
    storage_integration = integrate_s3
    url = 's3://snowflakeassignment0901/input_files/'
    file_format = assignment_db.my_schema.my_csv_format;

    list @s3_stage;
```

	name	size	md5
1	s3://snowflakeassignment0901/input_files/people_small.csv	1,109	df60c82e682f54276a4619154ad66eac

TASK 9: Load data into the tables using copy into statements.

- Load data from INTERNAL STAGE

  Scenario 1: load data from a large single csv file.

```
// SCENARIO-1: load data from a large single csv file.

    list @assignment_db.my_schema.%people_internal;

    copy into assignment_db.my_schema.people_internal
    from (select t.$1, t.$2, t.$3, t.$4, t.$5, t.$6, t.$7, t.$8, t.$9,metadata$filename,
           current_timestamp(),'LOCAL' from @assignment_db.my_schema.%people_internal t)
    file_format = my_csv_format
    on_error = 'CONTINUE';

    select * from people_internal limit 10;
```

  Time taken to copy: **7.3 seconds** for 1 million rows

  Scenario 2: load data from multiple parts of the csv file.

  A large csv file can be split into multiple smaller units to load data by leveraging the capability of MPP compute clusters of snowflake.

  ➔ First split the file into multiple smaller units (here 5 mb files are used) using the below command:

  split -b 5000000 people-1000000.csv /Users/sahil/Downloads/people

➔ Load the files into the staging area and load into the snowflake table using COPY command.

```
sahiljais#COMPUTE_WH@ASSIGNMENT_DB.MY_SCHEMA>put file:///Users/sahil/Downloads/people/people**
                                             @assignment_db.my_schema.%people_internal;
```

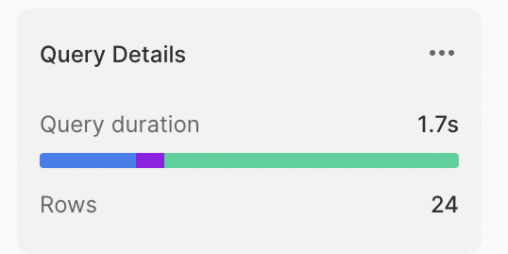
source	target	source_size	target_size	source_compression	target_compression	status	message
peopleaa	peopleaa.gz	5000000	2179504	NONE	GZIP	UPLOADED	
peopleab	peopleab.gz	5000000	2175312	NONE	GZIP	UPLOADED	
peopleac	peopleac.gz	5000000	2164880	NONE	GZIP	UPLOADED	
peoplead	peoplead.gz	5000000	2160208	NONE	GZIP	UPLOADED	
peopleae	peopleae.gz	5000000	2159568	NONE	GZIP	UPLOADED	
peopleaf	peopleaf.gz	5000000	2158720	NONE	GZIP	UPLOADED	
peopleag	peopleag.gz	5000000	2158160	NONE	GZIP	UPLOADED	
peopleah	peopleah.gz	5000000	2158464	NONE	GZIP	UPLOADED	
peopleai	peopleai.gz	5000000	2160080	NONE	GZIP	UPLOADED	
peopleaj	peopleaj.gz	5000000	2159600	NONE	GZIP	UPLOADED	
peopleak	peopleak.gz	5000000	2158784	NONE	GZIP	UPLOADED	
peopleal	peopleal.gz	5000000	2158160	NONE	GZIP	UPLOADED	
peopleam	peopleam.gz	5000000	2159328	NONE	GZIP	UPLOADED	
peoplean	peoplean.gz	5000000	2159344	NONE	GZIP	UPLOADED	
peopleao	peopleao.gz	5000000	2157712	NONE	GZIP	UPLOADED	
peopleap	peopleap.gz	5000000	2158816	NONE	GZIP	UPLOADED	
peopleaq	peopleaq.gz	5000000	2159792	NONE	GZIP	UPLOADED	
peoplear	peoplear.gz	5000000	2157376	NONE	GZIP	UPLOADED	
peopleas	peopleas.gz	5000000	2160928	NONE	GZIP	UPLOADED	
peopleat	peopleat.gz	5000000	2159488	NONE	GZIP	UPLOADED	
peopleau	peopleau.gz	5000000	2159264	NONE	GZIP	UPLOADED	
peopleav	peopleav.gz	5000000	2158896	NONE	GZIP	UPLOADED	
peopleaw	peopleaw.gz	5000000	2158688	NONE	GZIP	UPLOADED	
peopleax	peopleax.gz	1989639	860784	NONE	GZIP	UPLOADED	

24 Row(s) produced. Time Elapsed: 45.027s

```sql
list @assignment_db.my_schema.%people_internal;

copy into assignment_db.my_schema.people_internal
from (select t.$1, t.$2, t.$3, t.$4, t.$5, t.$6, t.$7, t.$8, t.$9,metadata$filename,
       current_timestamp(),'LOCAL' from @assignment_db.my_schema.%people_internal t)
file_format = my_csv_format
on_error = 'CONTINUE';

select * from people_internal limit 10;
```



Time taken to load 1 million rows only took **1.7 seconds**, this way bigger files can be loaded into snowflake efficiently.

- Load data from external stage.

```
list @assignment_db.my_schema.s3_stage;

copy into assignment_db.my_schema.people_external
from (select t.$1, t.$2, t.$3, t.$4, t.$5, t.$6, t.$7, t.$8, t.$9,metadata$filename,
current_timestamp(),'AWS_S3' from @assignment_db.my_schema.s3_stage t)
file_format = my_csv_format
on_error = 'CONTINUE';

select * from people_external limit 10;
```

Snapshot of the table is attached below:

S_NUM	USER_ID	FIRST_NAME	LAST_NAME	SEX	EMAIL	PHONE	DOB	JOB_TITLE	ELT_FILENAME	TS_TIMESTAMP	ELT_BY_APP
129,094	5e6cb465fB3	Derek	Moyer	Female	zimmermannina@example.org	+1-096-830-49(	2004-06-08	Music tutor	@PEOPLE_INTERNAL/p	0 04:38:35.645	LOCAL
129,095	DfCEaBFAf8B	Tyrone	Shields	Female	andresguzman@example.org	088.024.6651×6	1968-01-31	Tax inspector	@PEOPLE_INTERNAL/p	0 04:38:35.645	LOCAL
129,096	c62166f7a1E	Chase	Mueller	Female	shelbylarson@example.com	788-209-4956×	1950-02-13	Mining engineer	@PEOPLE_INTERNAL/p	0 04:38:35.645	LOCAL
129,097	A3aD112Fd5	Lonnie	Moon	Female	avillanueva@example.com	+1-128-950-05(	1933-08-21	Toxicologist	@PEOPLE_INTERNAL/p	0 04:38:35.645	LOCAL
129,098	6AAAaA0782	Bethany	Diaz	Female	cameronpreston@example.co	9759742195	1995-07-10	Market research	@PEOPLE_INTERNAL/p	0 04:38:35.645	LOCAL
129,099	c0CDB9Eb9C	Edgar	Leon	Male	pvasquez@example.org	921.003.7405×4	1961-12-03	Financial manage	@PEOPLE_INTERNAL/p	0 04:38:35.645	LOCAL
129,100	7E65bfD59Df	Latoya	Goodman	Male	deborahmack@example.com	(171)835-4603×	1934-07-16	Chartered loss a	@PEOPLE_INTERNAL/p	0 04:38:35.645	LOCAL
129,101	93e5fbfDD2e	Isaiah	Beck	Female	eflowers@example.net	+1-945-069-41	2016-07-22	Early years teach	@PEOPLE_INTERNAL/p	0 04:38:35.645	LOCAL
129,102	CFEAebc3C9	Bernard	Mathews	Male	choitaylor@example.net	962.139.2061	2002-07-17	Armed forces tec	@PEOPLE_INTERNAL/p	0 04:38:35.645	LOCAL
129,103	5c282a40a1(	Cory	Vazquez	Female	masonisaiah@example.org	0027135630	1922-12-27	Designer, televis	@PEOPLE_INTERNAL/p	0 04:38:35.645	LOCAL

TASK 7: create a variant version of this dataset.

Note: This is performed after the data load (TASK 8 & 9) to create a variant dataset.

➔ Create a simple table containing one variant type column and add the data from the existing table into the variant table (adding only 10 rows for the demo).

```
create or replace TRANSIENT table variant_data (
    data variant
);


insert into variant_data
    (select to_variant(object_construct(*)) as data from people_internal limit 10);

select * from variant_data;
```

	DATA
1	{ "DOB": "1914-03-25", "ELT_BY_APP": "LOCAL", "ELT_FILENAME": "@PEOPLE_INTERNAL/peopleax.gz", "ELT_TS_TIMESTAMP": "2023-01-10 04:38:35.645", "EN
2	{ "DOB": "1997-01-08", "ELT_BY_APP": "LOCAL", "ELT_FILENAME": "@PEOPLE_INTERNAL/peopleax.gz", "ELT_TS_TIMESTAMP": "2023-01-10 04:38:35.645", "EN
3	{ "DOB": "1982-11-05", "ELT_BY_APP": "LOCAL", "ELT_FILENAME": "@PEOPLE_INTERNAL/peopleax.gz", "ELT_TS_TIMESTAMP": "2023-01-10 04:38:35.645", "EN
4	{ "DOB": "1967-12-16", "ELT_BY_APP": "LOCAL", "ELT_FILENAME": "@PEOPLE_INTERNAL/peopleax.gz", "ELT_TS_TIMESTAMP": "2023-01-10 04:38:35.645", "EN
5	{ "DOB": "1967-02-12", "ELT_BY_APP": "LOCAL", "ELT_FILENAME": "@PEOPLE_INTERNAL/peopleax.gz", "ELT_TS_TIMESTAMP": "2023-01-10 04:38:35.645", "EN
6	{ "DOB": "1995-08-18", "ELT_BY_APP": "LOCAL", "ELT_FILENAME": "@PEOPLE_INTERNAL/peopleax.gz", "ELT_TS_TIMESTAMP": "2023-01-10 04:38:35.645", "EN
7	{ "DOB": "1948-09-22", "ELT_BY_APP": "LOCAL", "ELT_FILENAME": "@PEOPLE_INTERNAL/peopleax.gz", "ELT_TS_TIMESTAMP": "2023-01-10 04:38:35.645", "EN
8	{ "DOB": "2015-01-22", "ELT_BY_APP": "LOCAL", "ELT_FILENAME": "@PEOPLE_INTERNAL/peopleax.gz", "ELT_TS_TIMESTAMP": "2023-01-10 04:38:35.645", "EN
9	{ "DOB": "1923-04-21", "ELT_BY_APP": "LOCAL", "ELT_FILENAME": "@PEOPLE_INTERNAL/peopleax.gz", "ELT_TS_TIMESTAMP": "2023-01-10 04:38:35.645", "EN
10	{ "DOB": "1922-10-16", "ELT_BY_APP": "LOCAL", "ELT_FILENAME": "@PEOPLE_INTERNAL/peopleax.gz", "ELT_TS_TIMESTAMP": "2023-01-10 04:38:35.645", "EN

TASK 10: Upload any unrelated parquet file to the stage location and infer the schema of the file.

- ➔ Create a stage for parquet file format objects.
- ➔ Upload the file in the staging area.
- ➔ Infer the schema.

```
sahiljais#COMPUTE_WH@ASSIGNMENT_DB.MY_SCHEMA>put file:///Users/sahil/Downloads/userdata.parquet
                                             @assignment_db.my_schema.parquet_stage;

+------------------+------------------+-------------+-------------+--------------------+--------------------+----------+---------+
| source           | target           | source_size | target_size | source_compression | target_compression | status   | message |
|------------------+------------------+-------------+-------------+--------------------+--------------------+----------+---------|
| userdata.parquet | userdata.parquet |      113239 |      113248 | PARQUET            | PARQUET            | UPLOADED |         |
+------------------+------------------+-------------+-------------+--------------------+--------------------+----------+---------+
```

```
create or replace file format my_parquet_format
  type = 'parquet';

create or replace stage parquet_stage
file_format = my_parquet_format;

show stages;
list @parquet_stage;

select * from table(infer_schema(location=> '@parquet_stage', file_format=>'my_parquet_format'));
```

	COLUMN_NAME	TYPE	NULLABLE	EXPRESSION	FILENAMES	ORDER_ID
1	registration_dttm	TIMESTAMP_NTZ	TRUE	$1:registration_dttm::TIMESTAMP_NTZ	userdata.parquet	0
2	id	NUMBER(38, 0)	TRUE	$1:id::NUMBER(38, 0)	userdata.parquet	1
3	first_name	TEXT	TRUE	$1:first_name::TEXT	userdata.parquet	2
4	last_name	TEXT	TRUE	$1:last_name::TEXT	userdata.parquet	3
5	email	TEXT	TRUE	$1:email::TEXT	userdata.parquet	4
6	gender	TEXT	TRUE	$1:gender::TEXT	userdata.parquet	5
7	ip_address	TEXT	TRUE	$1:ip_address::TEXT	userdata.parquet	6
8	cc	TEXT	TRUE	$1:cc::TEXT	userdata.parquet	7
9	country	TEXT	TRUE	$1:country::TEXT	userdata.parquet	8
10	birthdate	TEXT	TRUE	$1:birthdate::TEXT	userdata.parquet	9
11	salary	REAL	TRUE	$1:salary::REAL	userdata.parquet	10
12	title	TEXT	TRUE	$1:title::TEXT	userdata.parquet	11
13	comments	TEXT	TRUE	$1:comments::TEXT	userdata.parquet	12

TASK 11: Run a select query on the staged parquet file without loading it to a snowflake table.

Command: select $1::varchar data from @parquet_stage/userdata.parquet;

	DATA
1	{"birthdate":"","cc":"5610608195667267","comments":"","country":"Israel","email":"efuller0@examiner.com","first_name":"Ernest","gender":"Male","id":
2	{"birthdate":"1/16/1998","cc":"4508242795214771","comments":"🍆 🙇 🤦 🤦 🧓 🙋 🤦 🤦 ","country":"Indonesia","email":"afoster1@weibo.com","firs
3	{"birthdate":"11/21/1978","cc":"","comments":"ъ","country":"Colombia","email":"rmontgomery2@mozilla.org","first_name":"Ryan","gender":"Male","id":3,
4	{"birthdate":"10/29/1998","cc":"","comments":"","country":"Guatemala","email":"bnelson3@photobucket.com","first_name":"Brenda","gender":"Female",
5	{"birthdate":"7/12/1959","cc":"","comments":"","country":"Russia","email":"jellis4@amazon.com","first_name":"Jacqueline","gender":"Female","id":5,"ip_a
6	{"birthdate":"","cc":"30501574577558","comments":"","country":"Thailand","email":"pferguson5@gmpg.org","first_name":"Paul","gender":"Male","id":6,

TASK 12: Add masking policy to the PII columns such that fields like email, phone number, etc. show as **masked to a user with the developer role. If the role is PII the value of these columns should be visible.**

➔ Create a masking policy. Such that the masking only applies for the "DEVELOPER" role and visible to other roles (only considering ADMIN, PII, and DEVELOPER roles).

```
create or replace masking policy assignment_db.my_schema.pii_masked as (val string) returns string ->
    case
        when current_role() in ('PII', 'ADMIN') then val
        else '**masked**'
    end;
```

If the role is "ADMIN" or "PII" then the personal details will be visible.
If the role is "DEVELOPER" then the personal details will show be shown as **masked**.

➔ Attach the masking policy to the personal details (pii) columns.

```
alter table assignment_db.my_schema.people_internal modify
    column email set masking policy assignment_db.my_schema.pii_masked,
    column phone set masking policy assignment_db.my_schema.pii_masked;
```

NOTE: "Date" is not selected under masking policy because 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.

➔ Check the masking policy using the "DEVELOPER" role.

```
-- grant necessary privileges to DEVELOPER role

grant usage on database assignment_db to role DEVELOPER;
grant usage on schema my_schema to role DEVELOPER;
grant select on table assignment_db.my_schema.people_internal to role DEVELOPER;

use role DEVELOPER;

select * from people_internal limit 10;
```

	S_NUM	USER_ID	FIRST_NAME	LAST_NAME	SEX	EMAIL	PHONE	DOB	JOB_TITLE	ELT_FILENAME	ELT_TS_TIMESTAMP	ELT_BY_APP
1	983,007	8F8cDF0257a	Tamara	Burns	Female	**masked**	**masked**	1914-03-25	Office manager	@PEOPLE_INTERNAL3	-01-10 04:38:35.645	LOCAL
2	983,008	6C28F02A4De	Tricia	Bowen	Female	**masked**	**masked**	1997-01-08	Pension scheme mana	@PEOPLE_INTERNAL3	-01-10 04:38:35.645	LOCAL
3	983,009	d977B9956aA	Kristin	Christian	Male	**masked**	**masked**	1982-11-05	Psychologist, counsel	@PEOPLE_INTERNAL3	-01-10 04:38:35.645	LOCAL
4	983,010	20659cC078C	Brian	Greer	Female	**masked**	**masked**	1967-12-16	Theatre director	@PEOPLE_INTERNAL3	-01-10 04:38:35.645	LOCAL
5	983,011	F2ABa08Fcd1E	Steven	Conley	Female	**masked**	**masked**	1967-02-12	Programmer, applicati	@PEOPLE_INTERNAL3	-01-10 04:38:35.645	LOCAL
6	983,012	0eCae61b70d	Jillian	Jacobson	Female	**masked**	**masked**	1995-08-18	Public relations accou	@PEOPLE_INTERNAL3	-01-10 04:38:35.645	LOCAL
7	983,013	aa0be77e3fd2	Darryl	Whitney	Female	**masked**	**masked**	1948-09-22	Analytical chemist	@PEOPLE_INTERNAL3	-01-10 04:38:35.645	LOCAL
8	983,014	be3B6A04DF5	Jim	Walter	Female	**masked**	**masked**	2015-01-22	Radiographer, diagno	@PEOPLE_INTERNAL3	-01-10 04:38:35.645	LOCAL
9	983,015	8D445fd7b4dc	Mckenzie	Carroll	Female	**masked**	**masked**	1923-04-21	Research officer, trad	@PEOPLE_INTERNAL3	-01-10 04:38:35.645	LOCAL
10	983,016	bcAACED87E6	Sheila	Sims	Female	**masked**	**masked**	1922-10-16	Dentist	@PEOPLE_INTERNAL3	-01-10 04:38:35.645	LOCAL

➔ Check the masking policy using the “PII” role.

```
-- grant necessary privileges to PII role

grant usage on database assignment_db to role PII;
grant usage on schema my_schema to role PII;
grant select on table assignment_db.my_schema.people_internal to role PII;

use role PII;

select * from people_internal limit 10;
```

	S_NUM	USER_ID	FIRST_NAME	LAST_NAME	SEX	EMAIL	PHONE	DOB	JOB_TITLE	ELT_FILENAME	ELT_TS_TIMESTAMP	ELT_BY_APP
1	854,859	E99bebB0C8B	Brenda	Harmon	Female	horneann@exan	478.643.3656	1931-05-29	Secretary/administrat	@PEOPLE_INTERNAL3	-01-10 04:38:35.645	LOCAL
2	854,860	b81ee08cFFf1	Katrina	Mahoney	Male	stanleypaul@ex	210-554-553	1956-08-06	Engineer, electronics	@PEOPLE_INTERNAL3	-01-10 04:38:35.645	LOCAL
3	854,861	db9eA3bb1FD	Rickey	Poole	Male	christy24@exan	+1-785-988-	1927-01-06	Magazine features ed	@PEOPLE_INTERNAL3	-01-10 04:38:35.645	LOCAL
4	854,862	c1bE5Fa2c736	Marc	Molina	Female	hpotts@example	001-174-311	1999-02-19	Garment/textile techn	@PEOPLE_INTERNAL3	-01-10 04:38:35.645	LOCAL
5	854,863	f106CC59b7C	Noah	White	Male	rodney90@exan	(887)639-986	1989-01-28	Health service manag	@PEOPLE_INTERNAL3	-01-10 04:38:35.645	LOCAL
6	854,864	c7E789777aA	Eric	Austin	Female	mike75@examp	936-820-382	2011-06-22	Chartered public finar	@PEOPLE_INTERNAL3	-01-10 04:38:35.645	LOCAL
7	854,865	C70de8A2817	Steve	Sanchez	Male	bowmanterri@e	001-153-849	1998-12-23	Horticultural therapist	@PEOPLE_INTERNAL3	-01-10 04:38:35.645	LOCAL
8	854,866	5112fFCeF6cE	Krystal	English	Female	garrettroth@exa	(602)551-065	1967-02-15	Hotel manager	@PEOPLE_INTERNAL3	-01-10 04:38:35.645	LOCAL
9	854,867	7ef4CbAff941	Joann	Mcknight	Female	griffinjaime@exa	001-591-171	1971-02-19	Phytotherapist	@PEOPLE_INTERNAL3	-01-10 04:38:35.645	LOCAL
10	854,868	03D938AfBBf4	Jonathan	Huff	Male	vegabrett@exan	(047)703-182	1913-09-07	Engineer, civil (consul	@PEOPLE_INTERNAL3	-01-10 04:38:35.645	LOCAL