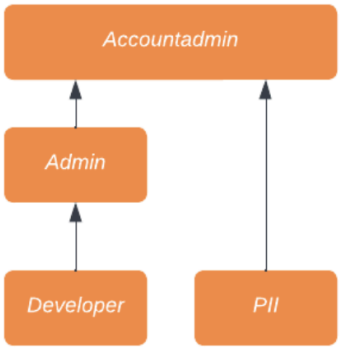
Assignment

Name: **Md. Owais Ashraf**
Employee ID: **TAS205**

PS 1: Create roles as per the below-mentioned hierarchy.



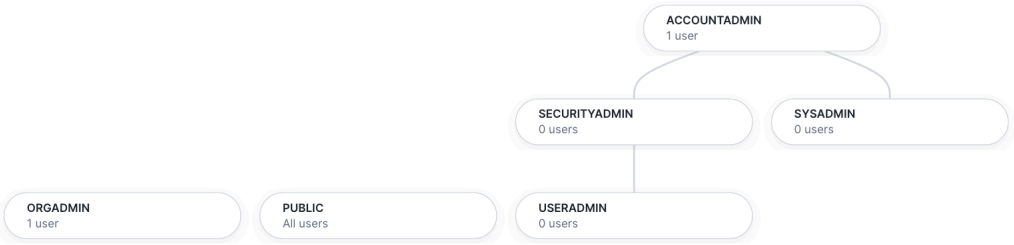
Also given that Accountadmin already exists in the hierarchy.

Solution :-

1. Entering SnowSQL CLI

```
owais@Mds-MacBook-Air ~ % snowsql -a uo09544.ap-southeast-1 -u OWAIS
Password:
* SnowSQL * v1.2.32
Type SQL statements or !help
OWAIS#COMPUTE_WH@(no database).(no schema)>
```

Looking into pre-existing architecture :-



ACCOUNTADMIN: This is the top-tier role that has complete control over all aspects of the Snowflake account, including user and role management, billing, and configuration settings.

SECURITYADMIN: This role specializes in the security aspects, such as managing access permissions and security settings. It's a focused role that does not cover broader administrative tasks.

SYSADMIN: The SysAdmin role is tasked with the administration of system-level objects like databases and warehouses. It does not encompass user or security administration.

USERADMIN: Dedicated to user management, this role can create and manage users and their roles but does not have access to the broader account or system administration.

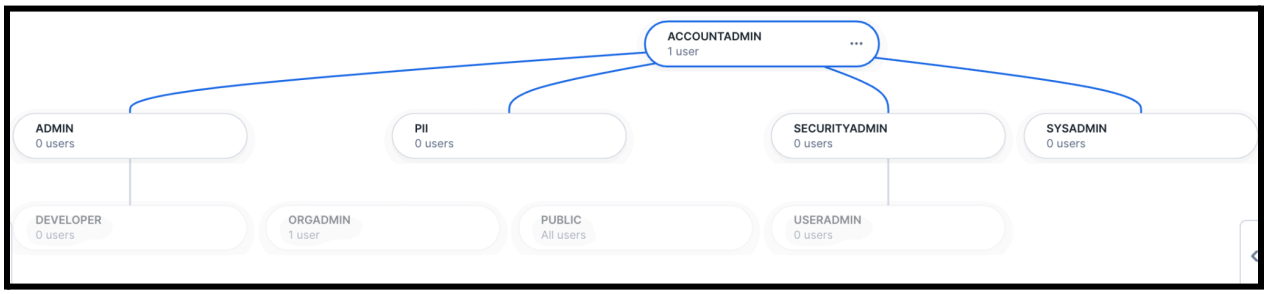
2. Adding roles as per problem statement ➖

```
OWAIS#COMPUTE_WH@(no database).(no schema)>CREATE ROLE Admin;
                                           GRANT ROLE Admin TO ROLE Accountadmin;

+----------------------------------+
| status                           |
|----------------------------------|
| Role ADMIN successfully created. |
+----------------------------------+
1 Row(s) produced. Time Elapsed: 0.194s
+----------------------------------+
| status                           |
|----------------------------------|
| Statement executed successfully. |
+----------------------------------+
1 Row(s) produced. Time Elapsed: 0.154s
```

```
OWAIS#COMPUTE_WH@(no database).(no schema)>CREATE ROLE Developer;
                                           GRANT ROLE Developer TO ROLE Admin;

+-------------------------------------+
| status                              |
|-------------------------------------|
| Role DEVELOPER successfully created. |
+-------------------------------------+
1 Row(s) produced. Time Elapsed: 0.274s
+----------------------------------+
| status                           |
|----------------------------------|
| Statement executed successfully. |
+----------------------------------+
1 Row(s) produced. Time Elapsed: 0.146s
```

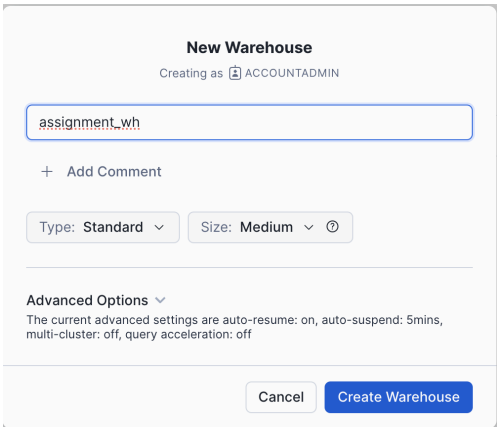
```
OWAIS#COMPUTE_WH@(no database).(no schema)>CREATE ROLE PII;
                                           GRANT ROLE PII TO ROLE Accountadmin;

+--------------------------------+
| status                         |
|--------------------------------|
| Role PII successfully created. |
+--------------------------------+
1 Row(s) produced. Time Elapsed: 0.169s
+----------------------------------+
| status                           |
|----------------------------------|
| Statement executed successfully. |
+----------------------------------+
1 Row(s) produced. Time Elapsed: 0.139s
```

Updated hierarchy:-

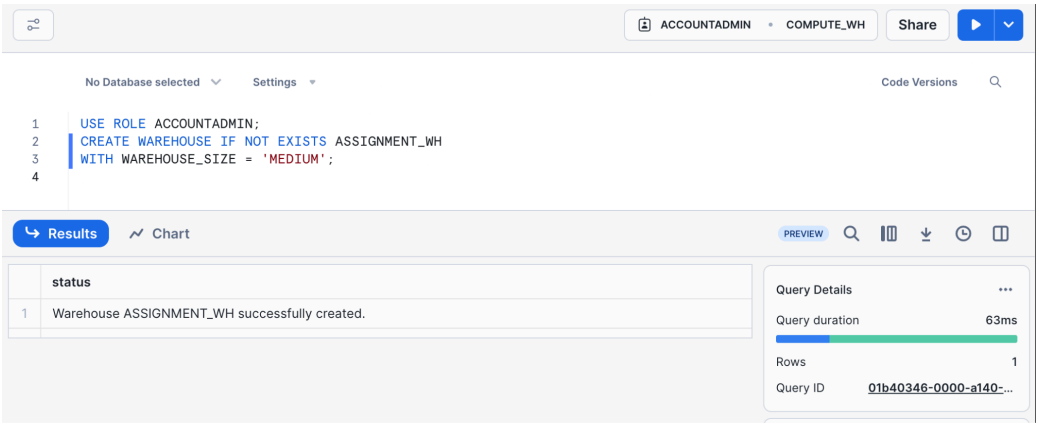


2. Create an M-sized warehouse using the accountadmin role, name -> assignment_wh and use it for all the queries.

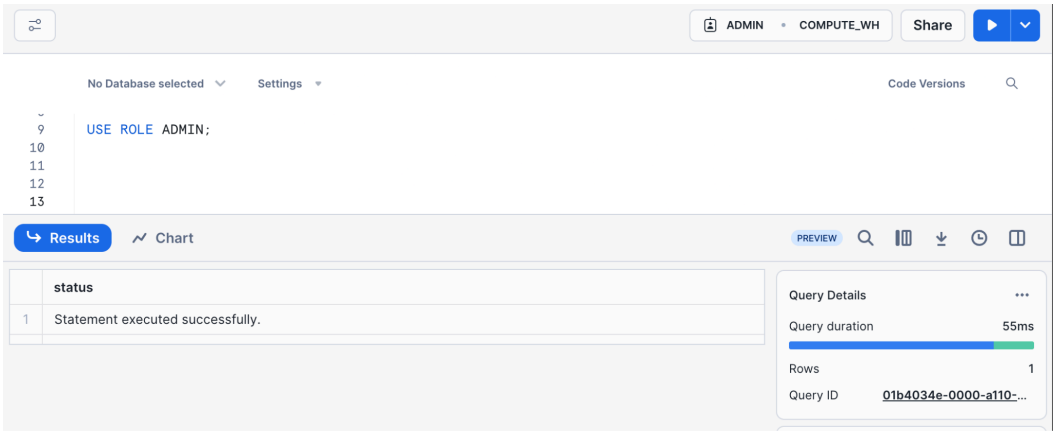
Using UI :-



Using Snowflake Worksheet / SnowSql CLI :-

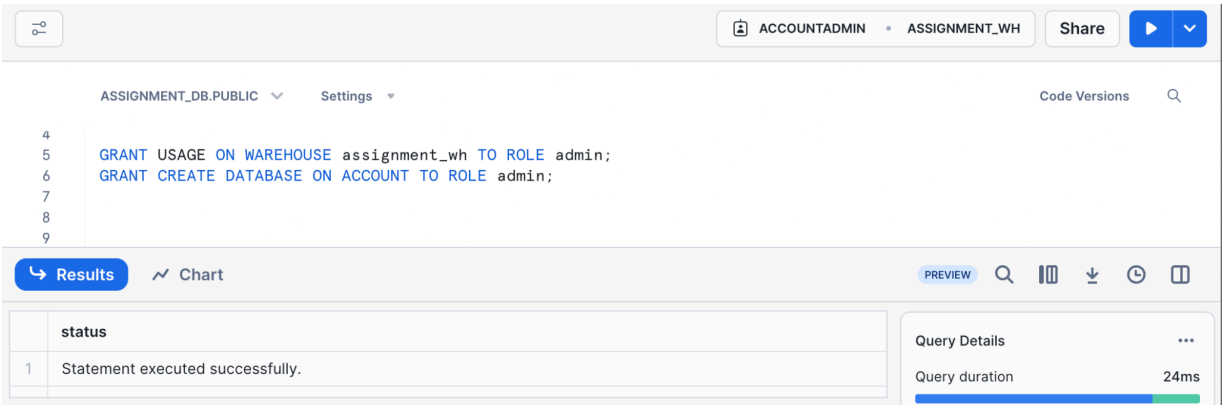


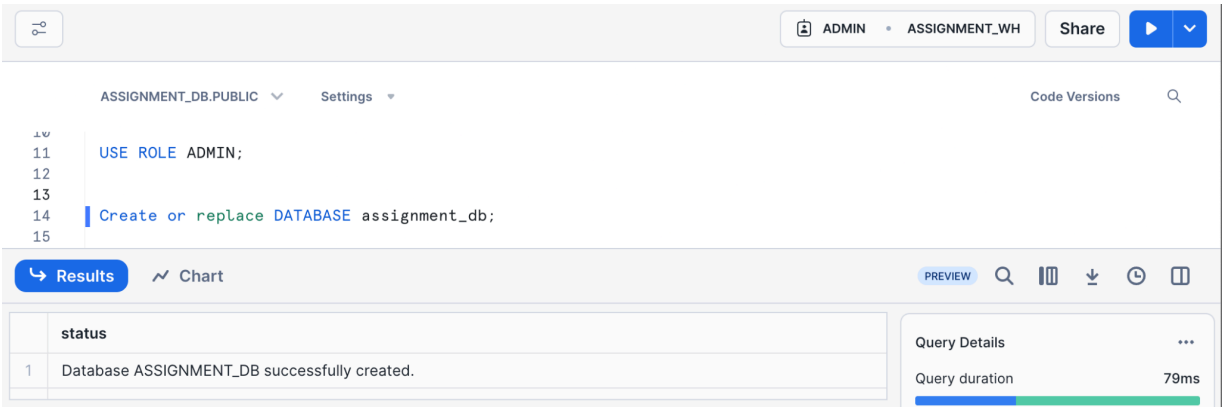
3. Switch to the admin role



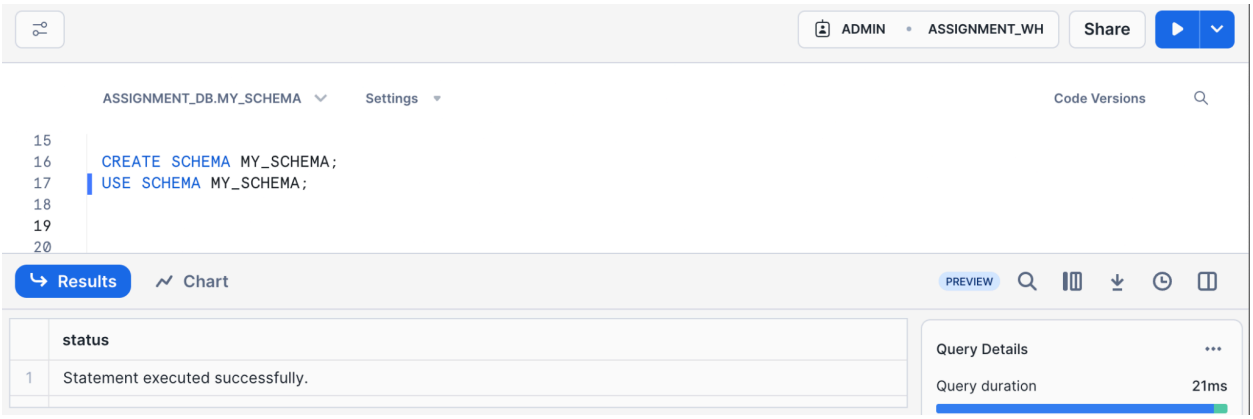
4. Create a database assignment_db

Before creation of the database we will have to grant permission to ADMIN role of creating database and warehouse.

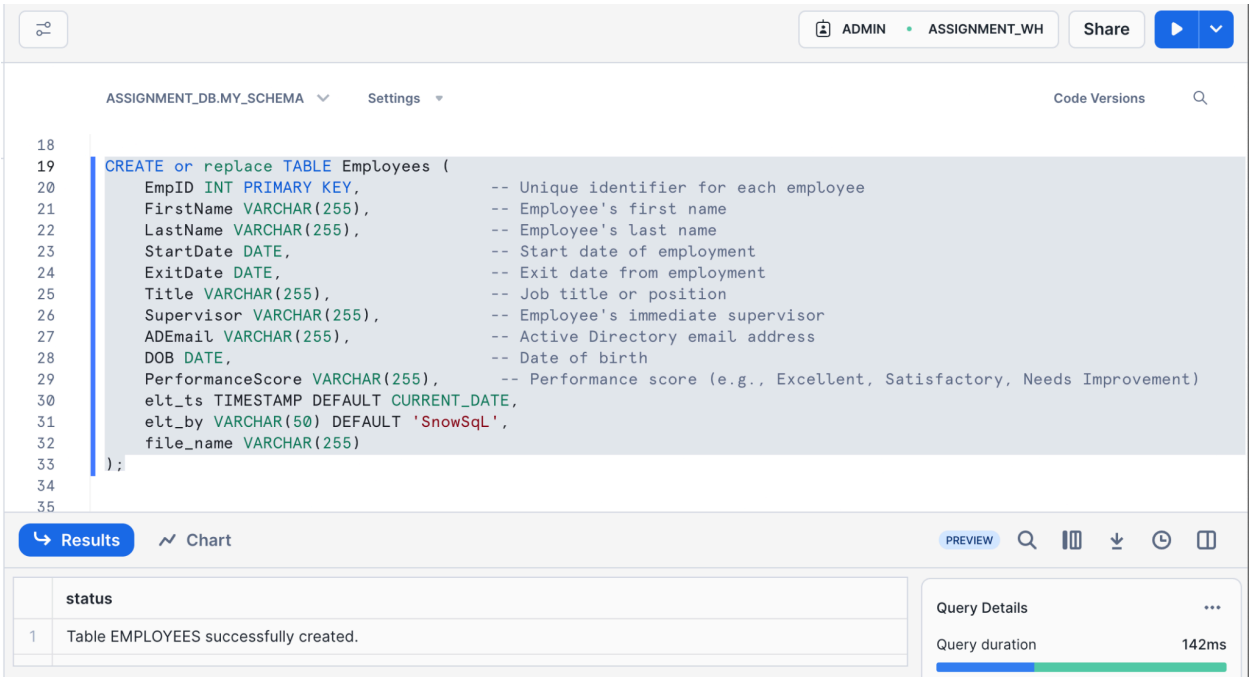




5. Create a schema my_schema

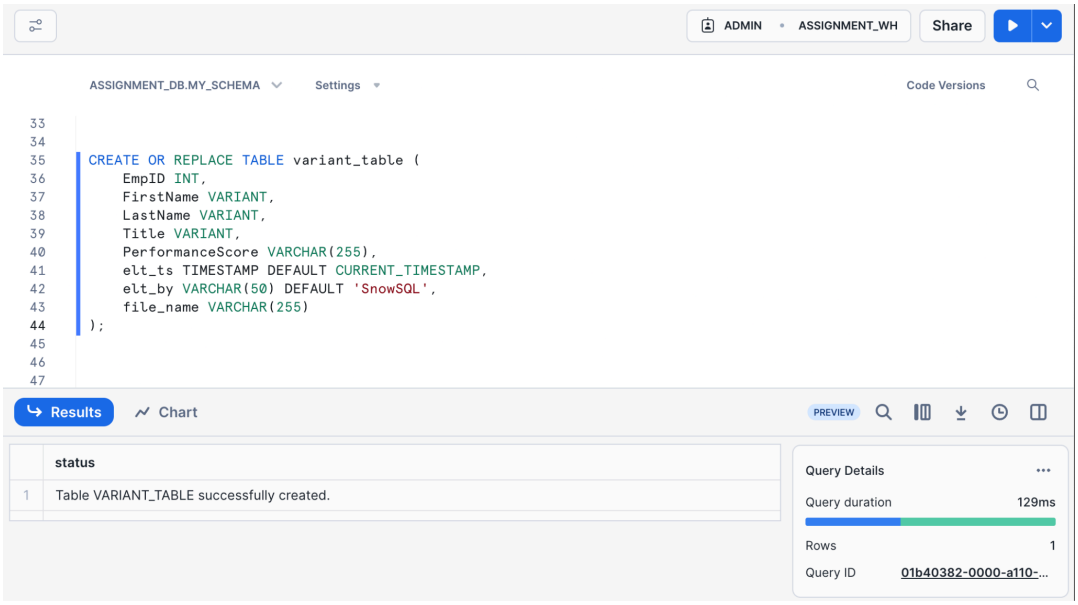


6. Create a table using any sample csv. You can get 1 by googling for sample csv's. Preferably search for a sample employee dataset so that you have PII related columns else you can consider any column as PII.

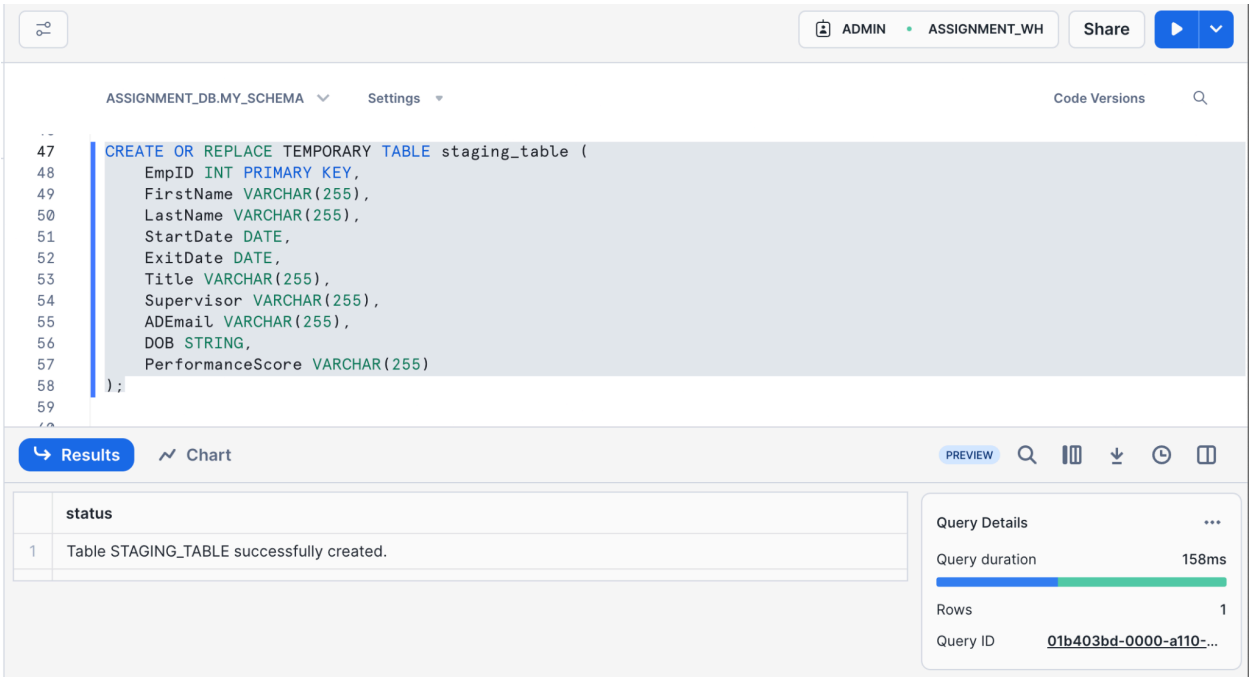


Dataset Link: Click

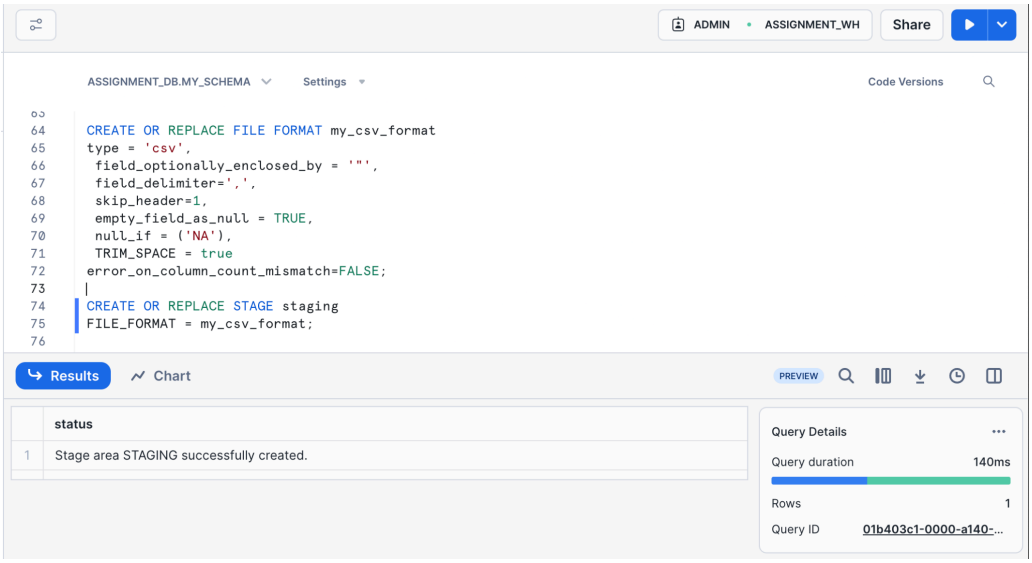
7. Create a Variant Table of the dataset.



In order to load data into the variant table, we need a staging table with the same schema as the original table.



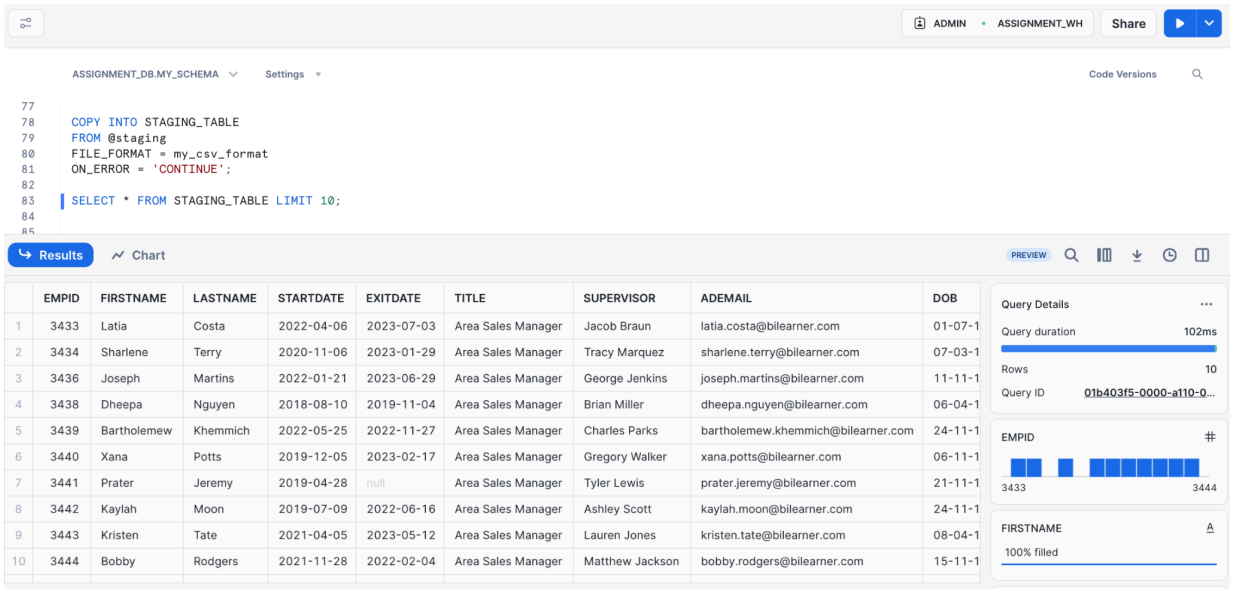
Creating staging area and file format



Using PUT command now to fetch data from local storage to snowflake schema (internal staging)

```
OWS#ASSIGNMENT_WH@ASSIGNMENT_DB.MY_SCHEMA>PUT file:///Users/owais/Downloads/employee_dataset.csv @staging;

+----------------------+-------------------------+-------------+-------------+--------------------+--------------------+----------+---------+
| source               | target                  | source_size | target_size | source_compression | target_compression | status   | message |
|----------------------+-------------------------+-------------+-------------+--------------------+--------------------+----------+---------|
| employee_dataset.csv | employee_dataset.csv.gz |      779998 |      193024 | NONE               | GZIP               | UPLOADED |         |
+----------------------+-------------------------+-------------+-------------+--------------------+--------------------+----------+---------+
1 Row(s) produced. Time Elapsed: 1.701s
```

Now copying data from staging area to our temporary table



Now, we have to bring data into our variant table

ASSIGNMENT_DB.MY_SCHEMA Settings Code Versions

```
88
89  INSERT INTO variant_table (
90      EmpID,
91      FullName,
92      Duration,
93      Title,
94      Supervisor,
95      Contact,
96      DOB,
97      PerformanceScore,
98      elt_by,
99      file_name
100 )
101 SELECT
102     EmpID,
103     OBJECT_CONSTRUCT(
104         'First Name', FirstName,
105         'Last Name', LastName
106     ) AS FullName,
107     OBJECT_CONSTRUCT(
108         'Start Date', StartDate,
109         'Exit Date', ExitDate
110     ) AS Duration,
111     Title,
112     Supervisor,
113     OBJECT_CONSTRUCT(
114         'AD Email', ADEmail
115     ) AS Contact,
116     DOB,
117     PerformanceScore,
118     'SnowSQL', -- Assuming the elt_by is hardcoded for this example
119     'employees_dataset.csv' -- Assuming a single source file for simplicity
120 FROM
121     STAGING_TABLE;
```

The columns we want to be paired together as one json file under one parent column, we will use OBJECT_CONSTRUCT method and alias it on the variant table column name.

	number of rows inserted
1	6024

*Select * from variant_table;*

Results Chart PREVIEW

	EMPID	FULLNAME	DURATION	TITLE	SUPERV	CONTACT	DOB	PERFO
1	3433	{ "First Name": "Latia", "Last Name": "Costa"}	{ "Exit Date": "0023-07-03", "Start Date": "0022-04-06"}	Area S	Jacob Br	{ "AD Email": "latia.costa@b	01-07-1942	Exceed
2	3434	{ "First Name": "Sharlene", "Last Name": "Terry"}	{ "Exit Date": "0023-01-29", "Start Date": "0020-11-06"}	Area S	Tracy Ma	{ "AD Email": "sharlene.terry	07-03-1957	Fully M
3	3436	{ "First Name": "Joseph", "Last Name": "Martins"}	{ "Exit Date": "0023-06-29", "Start Date": "0022-01-21"}	Area S	George .	{ "AD Email": "joseph.martin	11-11-1949	Fully M
4	3438	{ "First Name": "Dheepa", "Last Name": "Nguyen"}	{ "Exit Date": "0019-11-04", "Start Date": "0018-08-10"}	Area S	Brian Mil	{ "AD Email": "dheepa.nguye	06-04-1948	Fully M
5	3439	{ "First Name": "Bartholemew", "Last Name": "Khen	{ "Exit Date": "0022-11-27", "Start Date": "0022-05-25"}	Area S	Charles I	{ "AD Email": "bartholemew.	24-11-1981	Fully M
6	3440	{ "First Name": "Xana", "Last Name": "Potts"}	{ "Exit Date": "0023-02-17", "Start Date": "0019-12-05"}	Area S	Gregory	{ "AD Email": "xana.potts@b	06-11-1951	Fully M
7	3441	{ "First Name": "Prater", "Last Name": "Jeremy"}	{ "Start Date": "0019-04-28"}	Area S	Tyler Lev	{ "AD Email": "prater.jeremy	21-11-1989	Exceed
8	3442	{ "First Name": "Kaylah", "Last Name": "Moon"}	{ "Exit Date": "0022-06-16", "Start Date": "0019-07-09"}	Area S	Ashley S	{ "AD Email": "kaylah.moon@	24-11-1952	Exceed
9	3443	{ "First Name": "Kristen", "Last Name": "Tate"}	{ "Exit Date": "0023-05-12", "Start Date": "0021-04-05"}	Area S	Lauren J	{ "AD Email": "kristen.tate@	08-04-1994	Fully M
10	3444	{ "First Name": "Bobby", "Last Name": "Rodgers"}	{ "Exit Date": "0022-02-04", "Start Date": "0021-11-28"}	Area S	Matthew	{ "AD Email": "bobby.rodger	15-11-1983	Fully M
11	3446	{ "First Name": "Hector", "Last Name": "Dalton"}	{ "Start Date": "0021-08-24"}	Area S	Sydney I	{ "AD Email": "hector.dalton	01-05-1996	Exceed
12	3447	{ "First Name": "Mariela", "Last Name": "Schultz"}	{ "Exit Date": "0023-06-18", "Start Date": "0020-05-26"}	Area S	Michelle	{ "AD Email": "mariela.schult	17-02-1964	Fully M
13	3448	{ "First Name": "Angela", "Last Name": "Molina"}	{ "Exit Date": "0020-11-06", "Start Date": "0019-10-01"}	Area S	Patricia (	{ "AD Email": "angela.molina	12-05-1958	Fully M
14	3449	{ "First Name": "Gerald", "Last Name": "Preston"}	{ "Exit Date": "0023-05-27", "Start Date": "0023-05-10"}	Area S	Ashley R	{ "AD Email": "gerald.presto	18-09-1992	Fully M
15	3450	{ "First Name": "Reilly", "Last Name": "Moyer"}	{ "Exit Date": "0022-12-04", "Start Date": "0020-09-01"}	Area S	Stanley I	{ "AD Email": "reilly.moyer@	11-08-1994	Exceed
16	3451	{ "First Name": "Carlee", "Last Name": "French"}	{ "Exit Date": "0022-11-11", "Start Date": "0021-02-18"}	Area S	Michael	{ "AD Email": "carlee.french(	15-01-1968	Exceed
17	3455	{ "First Name": "Charity", "Last Name": "Miranda"}	{ "Exit Date": "0022-07-05", "Start Date": "0021-06-29"}	Area S	Daniel Re	{ "AD Email": "charity.miranc	18-01-1999	Exceed
18	3458	{ "First Name": "Cory", "Last Name": "Robinson"}	{ "Exit Date": "0023-05-24", "Start Date": "0022-04-28"}	Area S	Elizabeth	{ "AD Email": "cory.robinson	12-08-1996	Needs
19	3459	{ "First Name": "Saniya", "Last Name": "Yu"}	{ "Exit Date": "0022-06-21", "Start Date": "0021-04-18"}	Area S	Erin Baile	{ "AD Email": "saniya.yu@bil	09-02-1944	Exceed
20	3460	{ "First Name": "Alisa", "Last Name": "James"}	{ "Start Date": "0020-02-19"}	Area S	Dennis H	{ "AD Email": "alisa.james@t	10-02-1944	Fully M
21	3461	{ "First Name": "Lincoln", "Last Name": "Compton"}	{ "Exit Date": "0021-10-01", "Start Date": "0019-07-18"}	Area S	Tammy (	{ "AD Email": "lincoln.compt	29-12-1997	Needs
22	3462	{ "First Name": "Aliana", "Last Name": "Nolan"}	{ "Exit Date": "0021-09-17", "Start Date": "0018-09-13"}	Area S	Brianna I	{ "AD Email": "aliana.nolan@	09-08-1942	Needs
23	3463	{ "First Name": "Kayden", "Last Name": "Dodson"}	{ "Exit Date": "0021-07-18", "Start Date": "0020-04-30"}	Area S	Jessica I	{ "AD Email": "kayden.dodsc	21-06-1951	Fully M

8. Load the file into an external and internal stage

1. Internal Staging: *Stores data files internally within Snowflake*

- **Firstly, we will be creating a file format(although we have already created it before too), to configure our data, like its type, delimiter and the first line is header.**

```
[OWAIS#ASSIGNMENT_WH@ASSIGNMENT_DB.MY_SCHEMA>CREATE OR REPLACE FILE FORMAT my_format
                                                TYPE = 'CSV'
                                                FIELD_DELIMITER = ','
                                                SKIP_HEADER= 1;

+-----------------------------------------+
| status                                  |
|-----------------------------------------|
| File format MY_FORMAT successfully created. |
+-----------------------------------------+
1 Row(s) produced. Time Elapsed: 0.168s
```

- **Creating Staging area, configured with our created file format, to load the data**

```
[OWAIS#ASSIGNMENT_WH@ASSIGNMENT_DB.MY_SCHEMA>CREATE OR REPLACE STAGE staging_area FILE_FORMAT=my_format;
+-----------------------------------------+
| status                                  |
|-----------------------------------------|
| Stage area STAGING_AREA successfully created. |
+-----------------------------------------+
1 Row(s) produced. Time Elapsed: 0.159s
```

- **PUT Command used to locate the data file in our system and put/load it on the stage we created.**

```
[OWAIS#ASSIGNMENT_WH@ASSIGNMENT_DB.MY_SCHEMA>PUT file:///Users/owais/Documents/employee_dataset.csv  @staging_area;
+--------------------+-----------------------+-------------+-------------+--------------------+--------------------+----------+---------+
| source             | target                | source_size | target_size | source_compression | target_compression | status   | message |
|--------------------+-----------------------+-------------+-------------+--------------------+--------------------+----------+---------|
| employee_dataset.csv | employee_dataset.csv.gz |      366918 |      104320 | NONE               | GZIP               | UPLOADED |         |
+--------------------+-----------------------+-------------+-------------+--------------------+--------------------+----------+---------+
1 Row(s) produced. Time Elapsed: 1.058s
```

9. Loading data to our stage using the “copy into” command.

1. **Loading via internal Staging**

Now, since we already have data loaded into our stage, we will need a table where we will be storing our data, providing column names & structure to the csv file.

```
OWAIS#ASSIGNMENT_WH@ASSIGNMENT_DB.MY_SCHEMA>CREATE OR REPLACE TABLE emp_table (
                                                EmpID INT PRIMARY KEY,
                                                FirstName VARCHAR(255),
                                                LastName VARCHAR(255),
                                                StartDate DATE,
                                                ExitDate DATE,
                                                Title VARCHAR(255),
                                                Supervisor VARCHAR(255),
                                                ADEmail VARCHAR(255),
                                                DOB STRING,
                                                PerformanceScore VARCHAR(255),
                                                elt_ts TIMESTAMP,
                                                elt_by VARCHAR(255),
                                                file_name VARCHAR(255)
                                            );

+-------------------------------------+
| status                              |
|-------------------------------------|
| Table EMP_TABLE successfully created. |
+-------------------------------------+
1 Row(s) produced. Time Elapsed: 0.454s
```

- **After we have the table, we will use the “COPY INTO” command which serves as a pipeline to bring staged data to our existing table in the warehouse.**

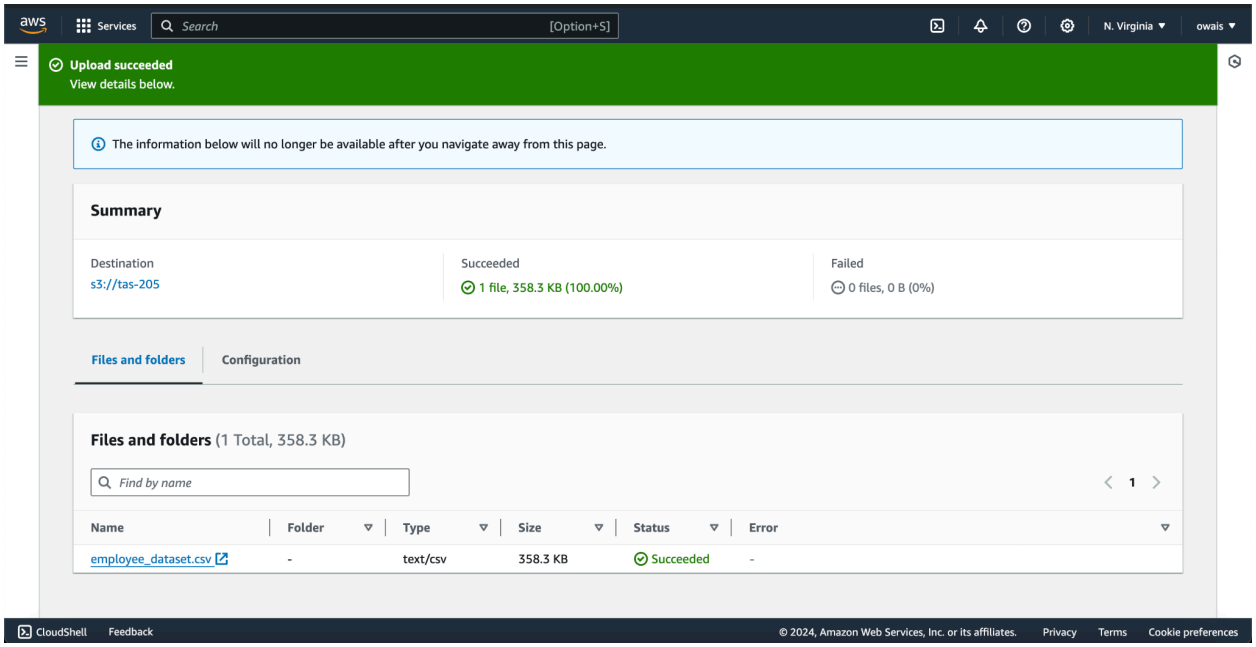
```
OWAIS#ASSIGNMENT_WH@ASSIGNMENT_DB.MY_SCHEMA>COPY INTO emp_table
                                            FROM (
                                                SELECT
                                                    $1,
                                                    $2,
                                                    $3,
                                                    TO_DATE($4, 'DD-Mon-YYYY'),
                                                    TO_DATE($5, 'DD-Mon-YYYY'),
                                                    $6,
                                                    $7,
                                                    $8,
                                                    $9,
                                                    $10,
                                                    CURRENT_TIMESTAMP(),
                                                    'AppName',
                                                    METADATA$FILENAME
                                                FROM @staging_area/employee_dataset.csv
                                            )
                                            FILE_FORMAT = (FORMAT_NAME = 'my_format');

+-------------------------------------+--------+-------------+-------------+-------------+-------------+-
-----+
| file                                | status | rows_parsed | rows_loaded | error_limit | errors_seen |
name |
|-------------------------------------+--------+-------------+-------------+-------------+-------------+-
-----|
| staging_area/employee_dataset.csv.gz | LOADED |        3000 |        3000 |           1 |           0 |
     |
+-------------------------------------+--------+-------------+-------------+-------------+-------------+-
-----+
1 Row(s) produced. Time Elapsed: 0.794s
```

Here we can see data being pipelined to our Snowflake.

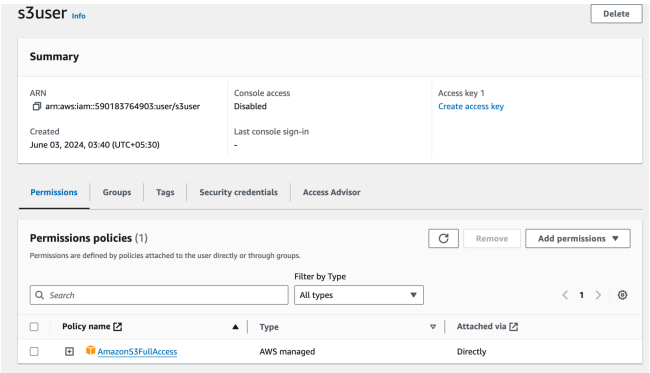
EMPID	FIRSTNAME	LASTNAME	STARTDATE	EXITDATE	TITLE	SUPERVISOR	ADEMAIL	DOB	PERFORMANCESCORE	ELT_TS	ELT_BY	FILE_NAME
3427	Uriah	Bridges	0019-09-20	null	Production Technician I	Peter Oneill	uriah.bridges@bilearner.com	07-10-1969	Fully Meets	2024-05-27 14:52:47.338	TAS205	employee_dataset.csv.gz
3428	Paula	Small	0023-02-11	null	Production Technician I	Renee Mccormick	paula.small@bilearner.com	30-08-1965	Fully Meets	2024-05-27 14:52:47.338	TAS205	employee_dataset.csv.gz
3429	Edward	Buck	0018-12-10	null	Area Sales Manager	Crystal Walker	edward.buck@bilearner.com	06-10-1991	Fully Meets	2024-05-27 14:52:47.338	TAS205	employee_dataset.csv.gz
3430	Michael	Riordan	0021-06-21	null	Area Sales Manager	Rebekah Wright	michael.riordan@bilearner.com	04-04-1998	Fully Meets	2024-05-27 14:52:47.338	TAS205	employee_dataset.csv.gz
3431	Jasmine	Onque	0019-06-29	null	Area Sales Manager	Jason Kim	jasmine.onque@bilearner.com	29-08-1969	Fully Meets	2024-05-27 14:52:47.338	TAS205	employee_dataset.csv.gz
3432	Maruk	Fraval	0020-01-17	null	Area Sales Manager	Sheri Campos	maruk.fraval@bilearner.com	03-04-1949	Fully Meets	2024-05-27 14:52:47.338	TAS205	employee_dataset.csv.gz
3433	Latia	Costa	0022-04-06	0023-07-03	Area Sales Manager	Jacob Braun	latia.costa@bilearner.com	01-07-1942	Exceeds	2024-05-27 14:52:47.338	TAS205	employee_dataset.csv.gz
3434	Sharlene	Terry	0020-11-06	0023-01-29	Area Sales Manager	Tracy Marquez	sharlene.terry@bilearner.com	07-03-1957	Fully Meets	2024-05-27 14:52:47.338	TAS205	employee_dataset.csv.gz
3435	Jac	McKinzie	0018-08-18	null	Area Sales Manager	Sharon Becker	jac.mckinzie@bilearner.com	15-05-1974	Exceeds	2024-05-27 14:52:47.338	TAS205	employee_dataset.csv.gz
3436	Joseph	Martins	0022-01-21	0023-06-29	Area Sales Manager	George Jenkins	joseph.martins@bilearner.com	11-11-1949	Fully Meets	2024-05-27 14:52:47.338	TAS205	employee_dataset.csv.gz
3437	Myriam	Givens	0023-08-04	null	Area Sales Manager	Troy White	myriam.givens@bilearner.com	26-01-1964	Fully Meets	2024-05-27 14:52:47.338	TAS205	employee_dataset.csv.gz
3438	Dheepa	Nguyen	0018-08-10	0019-11-04	Area Sales Manager	Brian Miller	dheepa.nguyen@bilearner.com	06-04-1948	Fully Meets	2024-05-27 14:52:47.338	TAS205	employee_dataset.csv.gz
3439	Bartholemew	Khemmich	0022-05-25	0022-11-27	Area Sales Manager	Charles Parks	bartholemew.khemmich@bilearner.com	24-11-1981	Fully Meets	2024-05-27 14:52:47.338	TAS205	employee_dataset.csv.gz
3440	Xana	Potts	0019-12-05	0023-02-17	Area Sales Manager	Gregory Walker	xana.potts@bilearner.com	06-11-1951	Fully Meets	2024-05-27 14:52:47.338	TAS205	employee_dataset.csv.gz
3441	Prater	Jeremy	0019-04-28	null	Area Sales Manager	Tyler Lewis	prater.jeremy@bilearner.com	21-11-1989	Exceeds	2024-05-27 14:52:47.338	TAS205	employee_dataset.csv.gz
3442	Kaylah	Moon	0019-07-09	0022-06-16	Area Sales Manager	Ashley Scott	kaylah.moon@bilearner.com	24-11-1952	Exceeds	2024-05-27 14:52:47.338	TAS205	employee_dataset.csv.gz
3443	Kristen	Tate	0021-04-05	0023-05-12	Area Sales Manager	Lauren Jones	kristen.tate@bilearner.com	08-04-1994	Fully Meets	2024-05-27 14:52:47.338	TAS205	employee_dataset.csv.gz
3444	Bobby	Rodgers	0021-11-28	0022-02-04	Area Sales Manager	Matthew Jackson	bobby.rodgers@bilearner.com	15-11-1983	Fully Meets	2024-05-27 14:52:47.338	TAS205	employee_dataset.csv.gz
3445	Reid	Park	0021-01-16	null	Area Sales Manager	Michelle Mitchell	reid.park@bilearner.com	07-12-1985	Exceeds	2024-05-27 14:52:47.338	TAS205	employee_dataset.csv.gz
3446	Hector	Dalton	0021-08-24	null	Area Sales Manager	Sydney French	hector.dalton@bilearner.com	01-05-1996	Exceeds	2024-05-27 14:52:47.338	TAS205	employee_dataset.csv.gz

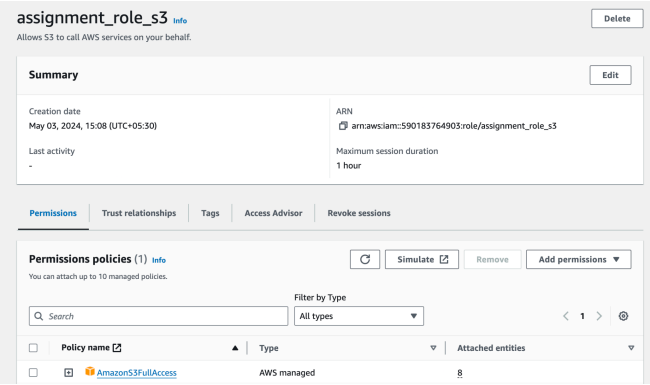
8. B. Loading via External Stage

Our dataset is currently in Amazon S3 Bucket.



But to access our bucket, we need to create an IAM role assigning to a user with access to read files from s3 bucket.





We will have to create integration from AWS S3 to Snowflake but integration is allowed using ACCOUNTADMIN and SYSADMIN roles only. Hence :-

1. **First we shift to role ACCOUNTADMIN**
2. **Create Integration with S3 bucket using ARN of the role we made with s3 bucket access.**
3. **Grant usage permission of that integration to role ADMIN.**

```
OWAIS#ASSIGNMENT_WH@ASSIGNMENT_DB.MY_SCHEMA>use ROLE ACCOUNTADMIN;
+----------------------------------+
| status                           |
|----------------------------------|
| Statement executed successfully. |
+----------------------------------+
1 Row(s) produced. Time Elapsed: 0.196s
OWAIS#ASSIGNMENT_WH@ASSIGNMENT_DB.MY_SCHEMA>CREATE OR REPLACE STORAGE INTEGRATION s3_to_snowflake
                                            type = external_stage
                                            storage_provider=s3
                                            enabled = true
                                            storage_aws_role_arn = "arn:aws:iam::590183764903:role/assignment_role_s3"
                                            storage_allowed_locations = ('s3://tas-205/');
+-------------------------------------------------+
| status                                          |
|-------------------------------------------------|
| Integration S3_TO_SNOWFLAKE successfully created. |
+-------------------------------------------------+
1 Row(s) produced. Time Elapsed: 0.179s
OWAIS#ASSIGNMENT_WH@ASSIGNMENT_DB.MY_SCHEMA>GRANT USAGE ON INTEGRATION s3_to_snowflake TO ROLE ADMIN;
+----------------------------------+
| status                           |
|----------------------------------|
| Statement executed successfully. |
+----------------------------------+
1 Row(s) produced. Time Elapsed: 0.161s
OWAIS#ASSIGNMENT_WH@ASSIGNMENT_DB.MY_SCHEMA>use ROLE ADMIN;
+----------------------------------+
| status                           |
|----------------------------------|
| Statement executed successfully. |
+----------------------------------+
1 Row(s) produced. Time Elapsed: 0.146s
```

Next, we create our external stage from s3 data source URL with our file format over the created storage integration.

```
OWAIS#ASSIGNMENT_WH@ASSIGNMENT_DB.MY_SCHEMA>CREATE or REPLACE STAGE external_stage
                                            STORAGE_INTEGRATION = s3_to_snowflake
                                            URL = 's3://tas-205/employee_dataset.csv'
                                            file_format = new_format;
+-------------------------------------------------+
| status                                          |
|-------------------------------------------------|
| Stage area EXTERNAL_STAGE successfully created. |
+-------------------------------------------------+
1 Row(s) produced. Time Elapsed: 2.193s
```

9. B. Loading data via External Loading

To grant our stage to fetch data from s3, we will need to configure trust policy of our S3 role with STORAGE_AWS_IAM_USER_ARN & STORAGE_AWS_EXTERNAL_ID

```
[OWAIS#ASSIGNMENT_WH@ASSIGNMENT_DB.MY_SCHEMA>desc integration s3_to_snowflake;
+---------------------------+---------------+-----------------------------------------------------+------------------+
| property                  | property_type | property_value                                      | property_default |
|---------------------------+---------------+-----------------------------------------------------+------------------|
| ENABLED                   | Boolean       | true                                                | false            |
| STORAGE_PROVIDER          | String        | S3                                                  |                  |
| STORAGE_ALLOWED_LOCATIONS | List          | s3://tas-205/                                       | []               |
| STORAGE_BLOCKED_LOCATIONS | List          |                                                     | []               |
| STORAGE_AWS_IAM_USER_ARN  | String        | arn:aws:iam::058264540845:user/2r0n0000-s           |                  |
| STORAGE_AWS_ROLE_ARN      | String        | arn:aws:iam::590183764903:role/assignment_role_s3   |                  |
| STORAGE_AWS_EXTERNAL_ID   | String        | BL39191_SFCRole=2_J6d9G0w/vWnq/Glqpq8x5koocdc=      |                  |
| COMMENT                   | String        |                                                     |                  |
+---------------------------+---------------+-----------------------------------------------------+------------------+
8 Row(s) produced. Time Elapsed: 1.128s
```

Trusted entities

Entities that can assume this role under specified conditions.

Edit trust policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::058264540845:user/2r0n0000-s"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "sts:ExternalId": "BL39191_SFCRole=2_J6d9G0w/vWnq/Glqpq8x5koocdc="
                }
            }
        }
    ]
}
```

Next Step is to create the table we will be loading our data in

```
CREATE OR REPLACE TABLE ext_emp_table (
                        EmpID INT PRIMARY KEY,
                        FirstName VARCHAR(255),
                        LastName VARCHAR(255),
                        StartDate DATE,
                        ExitDate DATE,
                        Title VARCHAR(255),
                        Supervisor VARCHAR(255),
                        ADEmail VARCHAR(255),
                        DOB STRING,
                        PerformanceScore VARCHAR(255),
                        elt_ts TIMESTAMP,
                        elt_by VARCHAR(255),
                        file_name VARCHAR(255)
                                    );
```

Results | Chart

	status
1	Table EXT_EMP_TABLE successfully created.

Query Details

Query duration 185ms

Rows 1

Query ID 01b4bfe7-3201-1a21-0...

Similar COPY INTO command to load data in the table.

```
COPY INTO ext_emp_table
    FROM (
        SELECT
            $1,
            $2,
            $3,
            TO_DATE($4, 'DD-Mon-YYYY'),
            TO_DATE($5, 'DD-Mon-YYYY'),
            $6,
            $7,
            $8,
            $9,
            $10,
            CURRENT_TIMESTAMP(),
            'TAS205',
            METADATA$FILENAME
        FROM @stage_area/employee_dataset.csv
    )
    FILE_FORMAT = (FORMAT_NAME = 'new_format');
```

Results | Chart

	file	status	rows_parsed	rows_loaded	error_limit	errors_seen	first_error	first_error
1	stage_area/employee_dataset.csv.gz	LOADED	3000	3000	1	0	null	

Query Details
Query duration 876ms

	EMPID	FIRSTNAME	LASTNAME	STARTDATE	EXITDATE	TITLE	SUPERVISOR	ADEMAIL
1	3427	Uriah	Bridges	0019-09-20	null	Production Technician I	Peter Oneill	uriah.bridges@bil
2	3428	Paula	Small	0023-02-11	null	Production Technician I	Renee Mccormick	paula.small@bilea
3	3429	Edward	Buck	0018-12-10	null	Area Sales Manager	Crystal Walker	edward.buck@bil
4	3430	Michael	Riordan	0021-06-21	null	Area Sales Manager	Rebekah Wright	michael.riordan@
5	3431	Jasmine	Onque	0019-06-29	null	Area Sales Manager	Jason Kim	jasmine.onque@b
6	3432	Maruk	Fraval	0020-01-17	null	Area Sales Manager	Sheri Campos	maruk.fraval@bile
7	3433	Latia	Costa	0022-04-06	0023-07-03	Area Sales Manager	Jacob Braun	latia.costa@bilea
8	3434	Sharlene	Terry	0020-11-06	0023-01-29	Area Sales Manager	Tracy Marquez	sharlene.terry@b
9	3435	Jac	McKinzie	0018-08-18	null	Area Sales Manager	Sharon Becker	jac.mckinzie@bile
10	3436	Joseph	Martins	0022-01-21	0023-06-29	Area Sales Manager	George Jenkins	joseph.martins@t
11	3437	Myriam	Givens	0023-08-04	null	Area Sales Manager	Troy White	myriam.givens@b
12	3438	Dheepa	Nguyen	0018-08-10	0019-11-04	Area Sales Manager	Brian Miller	dheepa.nguyen@
13	3439	Bartholemew	Khemmich	0022-05-25	0022-11-27	Area Sales Manager	Charles Parks	bartholemew.khe
14	3440	Xana	Potts	0019-12-05	0023-02-17	Area Sales Manager	Gregory Walker	xana.potts@bilea
15	3441	Prater	Jeremy	0019-04-28	null	Area Sales Manager	Tyler Lewis	prater.jeremy@bi
16	3442	Kaylah	Moon	0019-07-09	0022-06-16	Area Sales Manager	Ashley Scott	kaylah.moon@bil
17	3443	Kristen	Tate	0021-04-05	0023-05-12	Area Sales Manager	Lauren Jones	kristen.tate@bilea
18	3444	Bobby	Rodgers	0021-11-28	0022-02-04	Area Sales Manager	Matthew Jackson	bobby.rodgers@t
19	3445	Reid	Park	0021-01-16	null	Area Sales Manager	Michelle Mitchell	reid.park@bilearn
20	3446	Hector	Dalton	0021-08-24	null	Area Sales Manager	Sydney French	hector.dalton@bi

Query Details
Query duration 380ms
Rows 20
Query ID 01b4bfed-3201-1a21-0...

EMPID
3427 3446

FIRSTNAME
100% filled

LASTNAME
100% filled

STARTDATE
100% filled

EXITDATE
45% filled 55% null

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

```
CREATE OR REPLACE FILE FORMAT my_parquet_format
  TYPE = 'PARQUET'
  SNAPPY_COMPRESSION = TRUE;

CREATE OR REPLACE STAGE PARQUET_STAGE FILE_FORMAT = my_parquet_format;

SELECT * FROM TABLE(
infer_schema( location=>'@parquet_stage/test1.parquet',
         file_format=>'my_parquet_format')
);
```

Results | Chart

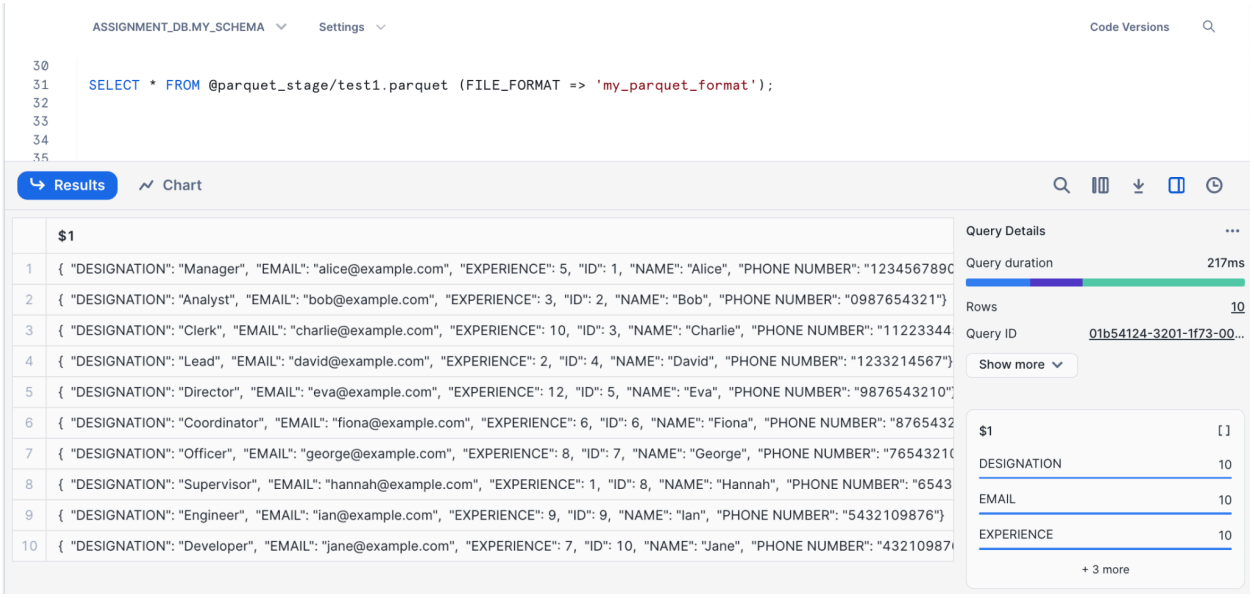
	COLUMN_NAME	TYPE	NULLABLE	EXPRESSION	FILENAMES	ORDER_ID
1	ID	NUMBER(38, 0)	TRUE	$1:ID::NUMBER(38, 0)	test1.parquet	0
2	NAME	TEXT	TRUE	$1:NAME::TEXT	test1.parquet	1
3	PHONE NUMBER	TEXT	TRUE	$1:PHONE NUMBER::TEXT	test1.parquet	2
4	EMAIL	TEXT	TRUE	$1:EMAIL::TEXT	test1.parquet	3
5	DESIGNATION	TEXT	TRUE	$1:DESIGNATION::TEXT	test1.parquet	4
6	EXPERIENCE	NUMBER(38, 0)	TRUE	$1:EXPERIENCE::NUMBER(38, 0)	test1.parquet	5

Parquet File Source**: Generated via python code**

PARQ.PY > ...

```python
import pandas as pd
import pyarrow as pa
import pyarrow.parquet as pq

# Create a DataFrame with the specified columns and 10 entries
data = {
    'ID': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    'NAME': ['Alice', 'Bob', 'Charlie', 'David', 'Eva', 'Fiona', 'George', 'Hannah', 'Ian', 'Jane'],
    'PHONE NUMBER': ['1234567890', '0987654321', '1122334455', '1233214567', '9876543210', '8765432109', '7654321098', '6543210987',
    'EMAIL': [
        'alice@example.com', 'bob@example.com', 'charlie@example.com',
        'david@example.com', 'eva@example.com', 'fiona@example.com',
        'george@example.com', 'hannah@example.com', 'ian@example.com', 'jane@example.com'
    ],
    'DESIGNATION': ['Manager', 'Analyst', 'Clerk', 'Lead', 'Director', 'Coordinator', 'Officer', 'Supervisor', 'Engineer', 'Develope
    'EXPERIENCE': [5, 3, 10, 2, 12, 6, 8, 1, 9, 7]
}
df = pd.DataFrame(data)

# Convert DataFrame to a Table and write to Parquet
table = pa.Table.from_pandas(df)
pq.write_table(table, 'test1.parquet')
```

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



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**

```
48
49  USE ROLE ACCOUNTADMIN;
50
51  GRANT USAGE ON DATABASE ASSIGNMENT_DB TO ROLE PII;
52  GRANT USAGE ON DATABASE ASSIGNMENT_DB TO ROLE DEVELOPER;
53  GRANT USAGE ON SCHEMA ASSIGNMENT_DB.MY_SCHEMA TO ROLE PII;
54  GRANT USAGE ON SCHEMA ASSIGNMENT_DB.MY_SCHEMA TO ROLE DEVELOPER;
55  GRANT SELECT ON ALL TABLES IN SCHEMA ASSIGNMENT_DB.MY_SCHEMA TO ROLE PII;
56  GRANT SELECT ON ALL TABLES IN SCHEMA ASSIGNMENT_DB.MY_SCHEMA TO ROLE DEVELOPER;
57  GRANT USAGE ON WAREHOUSE ASSIGNMENT_WH TO ROLE PII;
58  GRANT USAGE ON WAREHOUSE ASSIGNMENT_WH TO ROLE DEVELOPER;
59
60
61  -- Grants the ability to insert, update, delete, and select data from tables
62  GRANT INSERT, UPDATE, DELETE, SELECT ON ALL TABLES IN SCHEMA MY_SCHEMA TO ROLE developer;
63
64  use role developer;
65  -- Masking policy for email
66  CREATE OR REPLACE MASKING POLICY email_masking_policy AS (val STRING) RETURNS STRING ->
67      CASE
68          WHEN CURRENT_ROLE() IN ('PII') THEN val
69          ELSE '****masked****'
70      END;
71
72  -- Masking policy for phone number
73  CREATE OR REPLACE MASKING POLICY phone_number_masking_policy AS (val STRING) RETURNS STRING ->
74      CASE
75          WHEN CURRENT_ROLE() IN ('PII') THEN val
76          ELSE '****masked****'
77      END;
78
79  -- Apply masking policy to the email column
80  ALTER TABLE my_table MODIFY COLUMN email SET MASKING POLICY email_masking_policy;
81
82  -- Apply masking policy to the phone_number column
83  ALTER TABLE my_table MODIFY COLUMN phone_number SET MASKING POLICY phone_number_masking_policy;
84
85
86  select * from my_table;
87
```

Results | Chart

	ID	NAME	PHONE_NUMBER	EMAIL	DESIGNATION	EXPERIENCE
1	1	Alice	****masked****	****masked****	Manager	5
2	2	Bob	****masked****	****masked****	Analyst	3
3	3	Charlie	****masked****	****masked****	Clerk	10
4	4	David	****masked****	****masked****	Lead	2
5	5	Eva	****masked****	****masked****	Director	12
6	6	Fiona	****masked****	****masked****	Coordinator	6
7	7	George	****masked****	****masked****	Officer	8
8	8	Hannah	****masked****	****masked****	Supervisor	1
9	9	Ian	****masked****	****masked****	Engineer	9
10	10	Jane	****masked****	****masked****	Developer	7

Using Role PII:

```
89  use role pii;
90  select * from my_table;
91
92
```

Results | Chart

	ID	NAME	PHONE_NUMBER	EMAIL	DESIGNATION	EXPERIENCE
1	1	Alice	null	alice@example.com	Manager	5
2	2	Bob	null	bob@example.com	Analyst	3
3	3	Charlie	null	charlie@example.com	Clerk	10
4	4	David	null	david@example.com	Lead	2
5	5	Eva	null	eva@example.com	Director	12
6	6	Fiona	null	fiona@example.com	Coordinator	6
7	7	George	null	george@example.com	Officer	8
8	8	Hannah	null	hannah@example.com	Supervisor	1
9	9	Ian	null	ian@example.com	Engineer	9
10	10	Jane	null	jane@example.com	Developer	7

- The End -