Report on Superstore Dataset Analysis

OBJECTIVE:

Analysis of a Superstore

ABSTRACT OF THE PROJECT:

The project is to develop Analysis of a Superstore.

TECHNOLOGY USED:

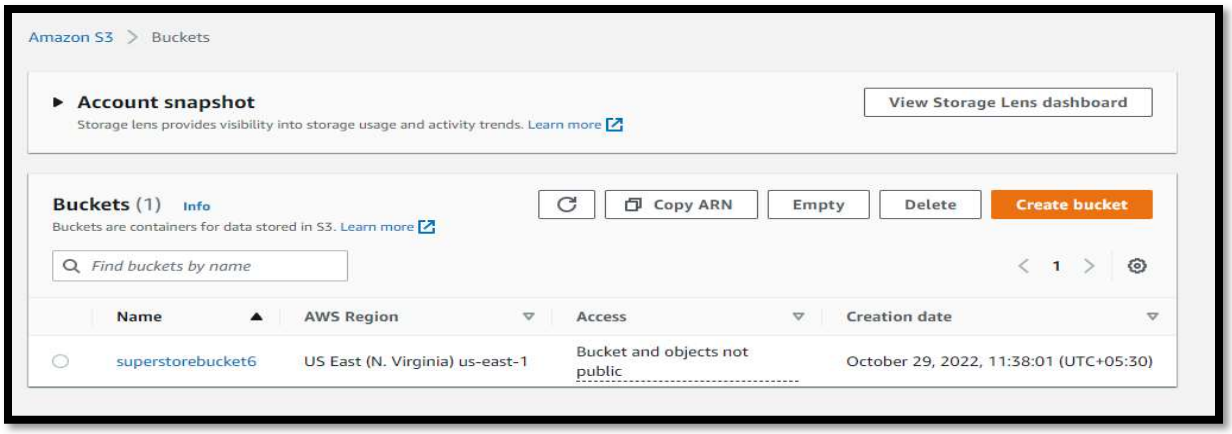
- Snowflake
- Cloud (AWS/Azure)

IMPLEMENTATION:

1. Create an External Stage to load the data continuously

In AWS,

Step 1: Create Bucket



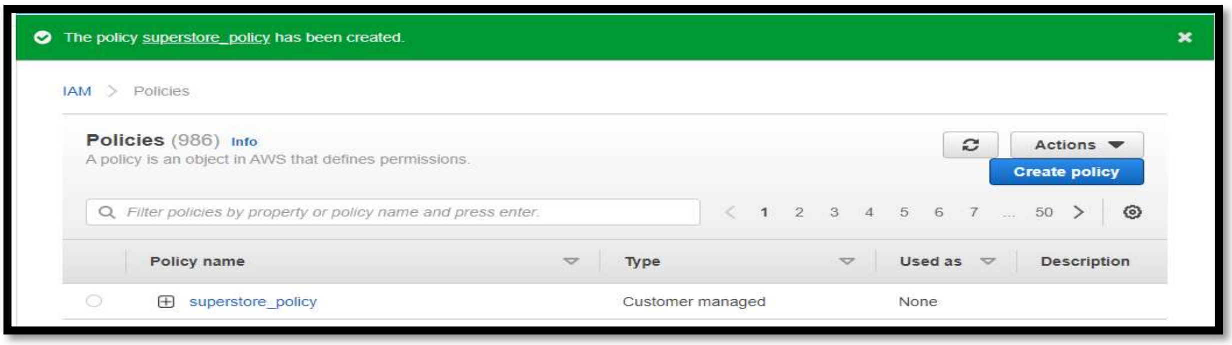
Storage lens provides visibility into storage usage and activity trends. Learn more

Buckets (1) Info

Buckets are containers for data stored in S3. Learn more

Name	AWS Region	Access	Creation date
superstorebucket6	US East (N. Virginia) us-east-1	Bucket and objects not public	October 29, 2022, 11:38:01 (UTC+05:30)

Step 2: Create Policy



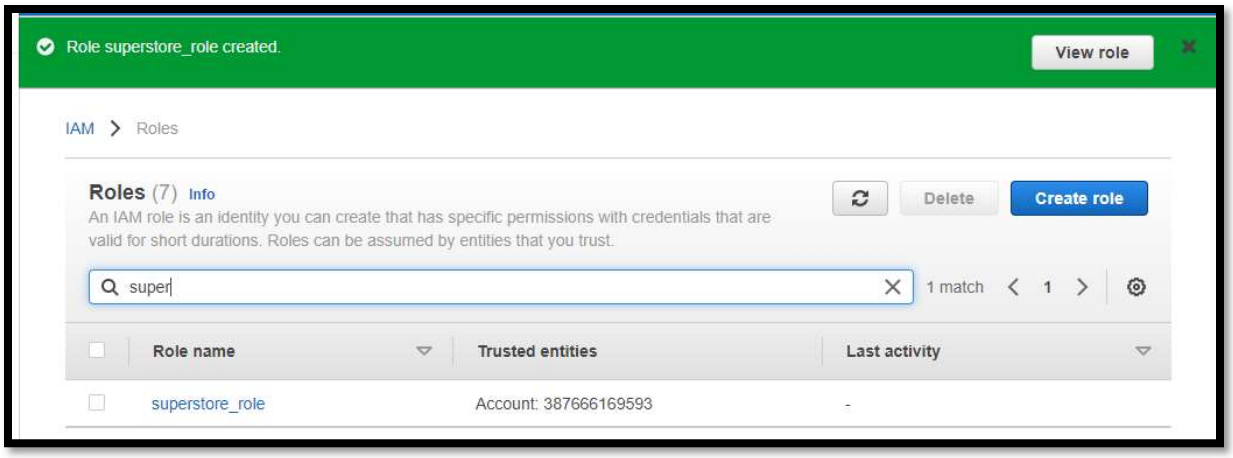
The policy superstore_policy has been created.

Policies (986) Info

A policy is an object in AWS that defines permissions.

Policy name	Type	Used as	Description
superstore_policy	Customer managed	None	

Step 3: Create Role



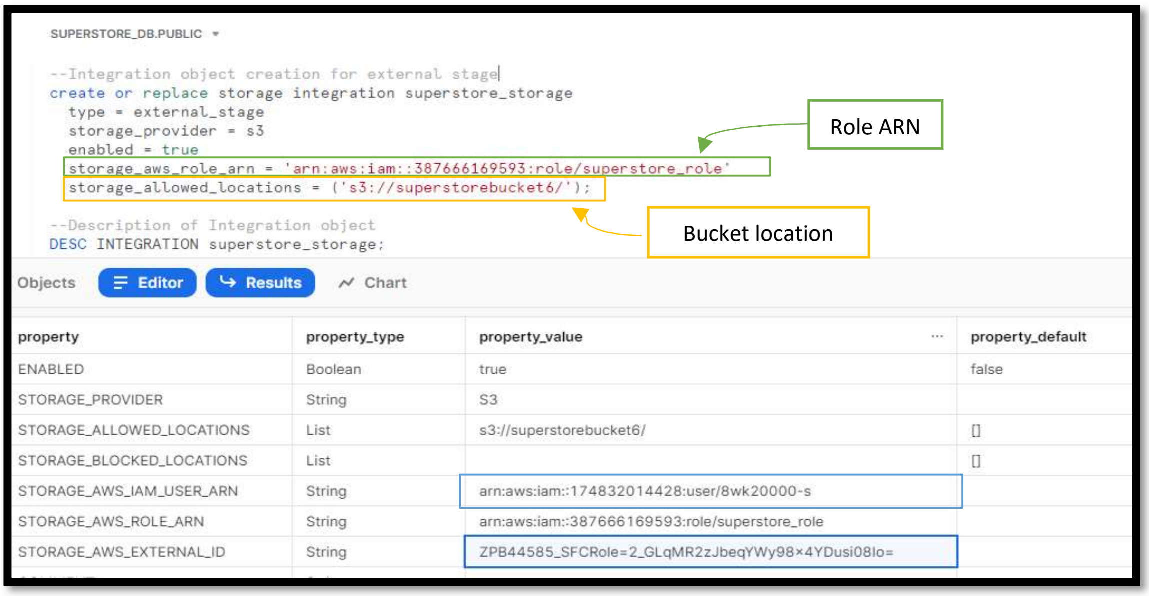
Roles (7) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Role name	Trusted entities	Last activity
superstore_role	Account: 387666169593	-

In Snowflake,

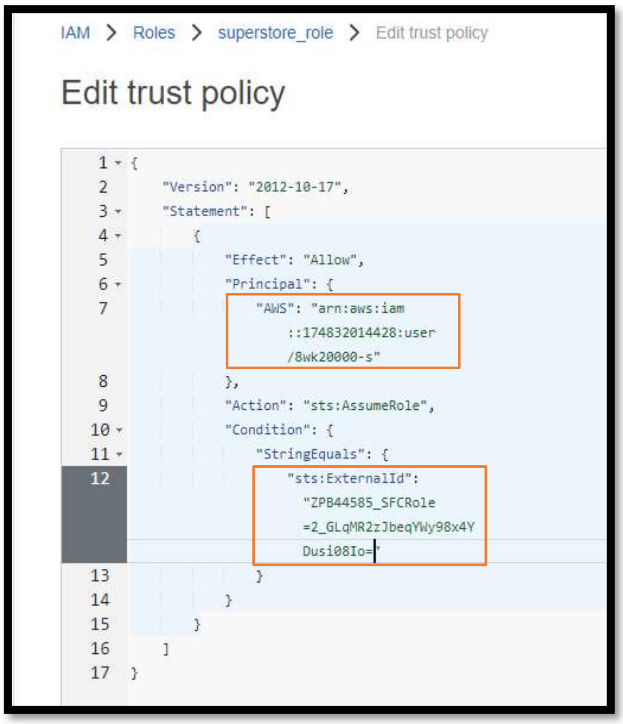
Step 4: Integration Creation (Integrating the bucket with storage)



```
--Integration object creation for external stage
create or replace storage integration superstore_storage
  type = external_stage
  storage_provider = s3
  enabled = true
  storage_aws_role_arn = 'arn:aws:iam::387666169593:role/superstore_role'
  storage_allowed_locations = ('s3://superstorebucket6/');

--Description of Integration object
DESC INTEGRATION superstore_storage;
```

property	property_type	property_value	property_default
ENABLED	Boolean	true	false
STORAGE_PROVIDER	String	S3	
STORAGE_ALLOWED_LOCATIONS	List	s3://superstorebucket6/	[]
STORAGE_BLOCKED_LOCATIONS	List		[]
STORAGE_AWS_IAM_USER_ARN	String	arn:aws:iam::174832014428:user/8wk20000-s	
STORAGE_AWS_ROLE_ARN	String	arn:aws:iam::387666169593:role/superstore_role	
STORAGE_AWS_EXTERNAL_ID	String	ZPB44585_SFCRole=2_GLqMR2zJbeqYWy98×4YDusi08lo=	

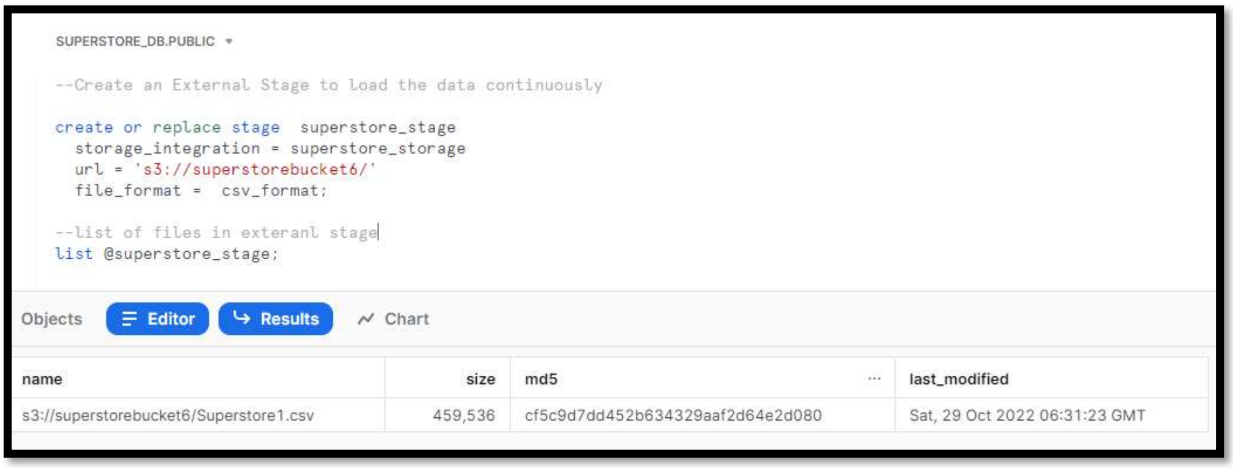
Step 5: Editing trust policy



Edit trust policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam
                    ::174832014428:user
                    /8wk20000-s"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "sts:ExternalId":
                      "ZPB44585_SFCRole
                      =2_GLqMR2zJbeqYWy98x4Y
                      Dusi08Io="
                }
            }
        }
    ]
}
```

Step 6: Create External stage and listing the files



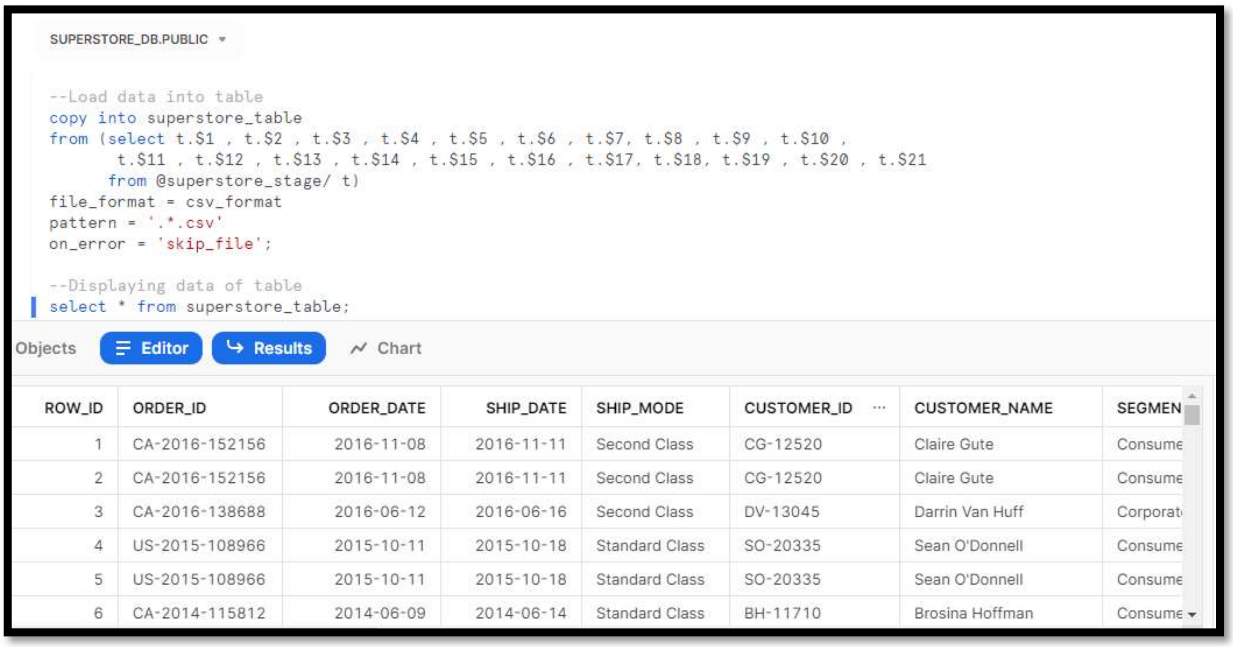
```
--Create an External Stage to load the data continuously

create or replace stage  superstore_stage
  storage_integration = superstore_storage
  url = 's3://superstorebucket6/'
  file_format =  csv_format;

--list of files in exteranl stage
list @superstore_stage;
```

name	size	md5	last_modified
s3://superstorebucket6/Superstore1.csv	459,536	cf5c9d7dd452b634329aaf2d64e2d080	Sat, 29 Oct 2022 06:31:23 GMT

Load the data from the External Stage to the respective Table

Step 1: Created Table superstore_table

Step 2: Loading data into table superstore_table from external stage



```
--Load data into table
copy into superstore_table
from (select t.$1 , t.$2 , t.$3 , t.$4 , t.$5 , t.$6 , t.$7, t.$8 , t.$9 , t.$10 ,
       t.$11 , t.$12 , t.$13 , t.$14 , t.$15 , t.$16 , t.$17, t.$18, t.$19 , t.$20 , t.$21
      from @superstore_stage/ t)
file_format = csv_format
pattern = '.*.csv'
on_error = 'skip_file';

--Displaying data of table
select * from superstore_table;
```

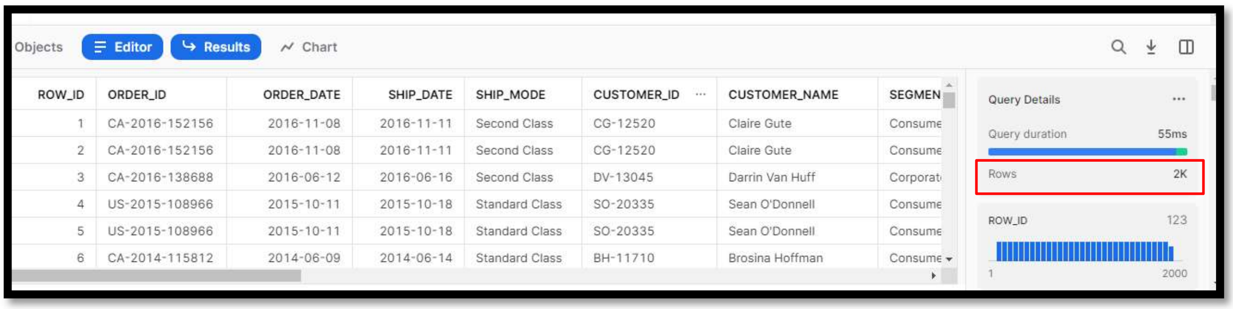
ROW_ID	ORDER_ID	ORDER_DATE	SHIP_DATE	SHIP_MODE	CUSTOMER_ID	CUSTOMER_NAME	SEGMEN
1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consume
2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consume
3	CA-2016-138688	2016-06-12	2016-06-16	Second Class	DV-13045	Darrin Van Huff	Corporat
4	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consume
5	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consume
6	CA-2014-115812	2014-06-09	2014-06-14	Standard Class	BH-11710	Brosina Hoffman	Consume

2. Create a scheduler to schedule the job at 12:00 AM IST hours every Thursday to perform previous step

Step 1: Created a task to schedule a job at 12:00 AM IST hours every Thursday

Step 2: The task will load the data from external stage to source table

Before task was executed:

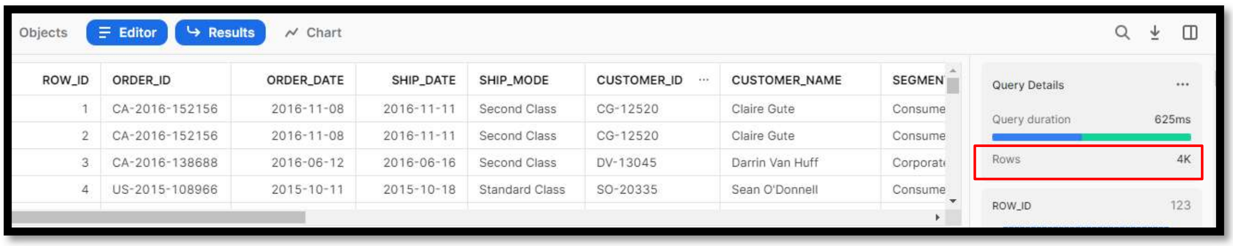


ROW_ID	ORDER_ID	ORDER_DATE	SHIP_DATE	SHIP_MODE	CUSTOMER_ID	CUSTOMER_NAME	SEGMEN
1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consume
2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consume
3	CA-2016-138688	2016-06-12	2016-06-16	Second Class	DV-13045	Darrin Van Huff	Corporat
4	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consume
5	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consume
6	CA-2014-115812	2014-06-09	2014-06-14	Standard Class	BH-11710	Brosina Hoffman	Consume

Query Details

Query duration	55ms
Rows	2K
ROW_ID	123
1	2000

After task was executed:



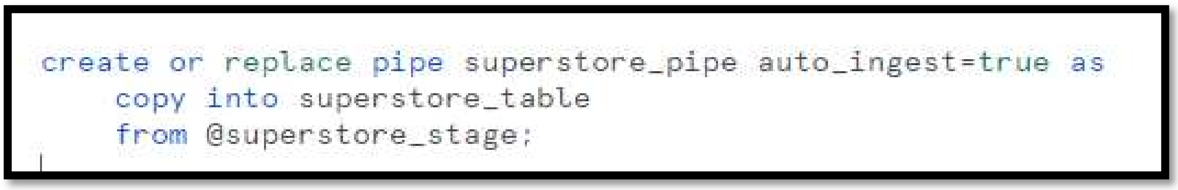
ROW_ID	ORDER_ID	ORDER_DATE	SHIP_DATE	SHIP_MODE	CUSTOMER_ID	CUSTOMER_NAME	SEGMEN
1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consume
2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consume
3	CA-2016-138688	2016-06-12	2016-06-16	Second Class	DV-13045	Darrin Van Huff	Corporat
4	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consume

Query Details

Query duration	625ms
Rows	4K
ROW_ID	123

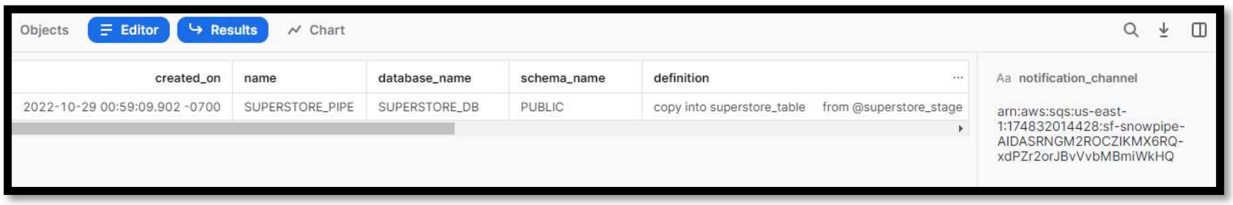
3. Create an alternative of the above step for auto ingestion of new record sets uploaded on the blob storage into respective snowflake table.

Step 1: Snowpipe created with auto-ingestion



```
create or replace pipe superstore_pipe auto_ingest=true as
    copy into superstore_table
    from @superstore_stage;
```

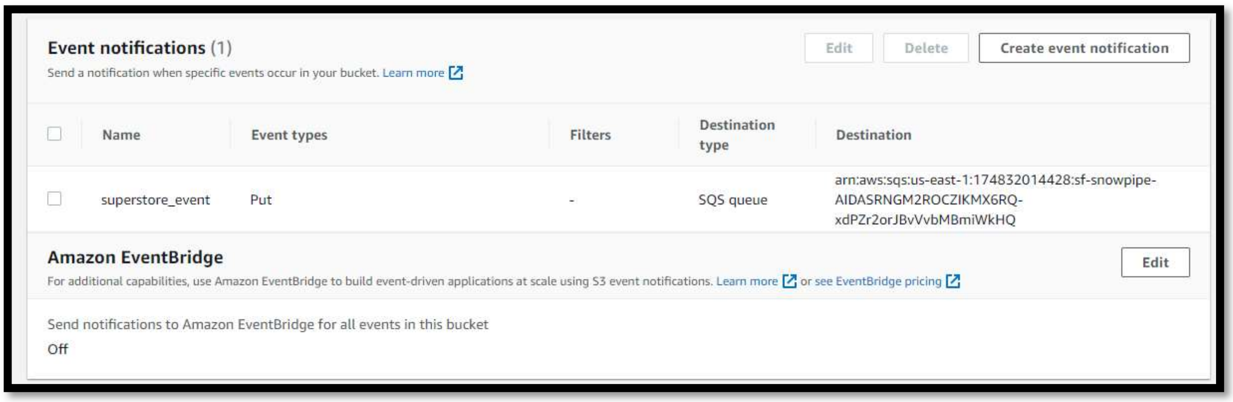
Step 2: Description of snowpipe



Objects | Editor | Results | Chart

created_on	name	database_name	schema_name	definition	notification_channel
2022-10-29 00:59:09.902 -0700	SUPERSTORE_PIPE	SUPERSTORE_DB	PUBLIC	copy into superstore_table from @superstore_stage	arn:aws:sqs:us-east-1:174832014428:sf-snowpipe-AIDASRNGM2ROCZIKMX6RQ-xdPZr2orJBvVvbMBmiWkHQ

Step 3: Event Notification created in AWS S3



Event notifications (1)

Send a notification when specific events occur in your bucket. Learn more

Edit | Delete | Create event notification

Name	Event types	Filters	Destination type	Destination
superstore_event	Put	-	SQS queue	arn:aws:sqs:us-east-1:174832014428:sf-snowpipe-AIDASRNGM2ROCZIKMX6RQ-xdPZr2orJBvVvbMBmiWkHQ

Amazon EventBridge

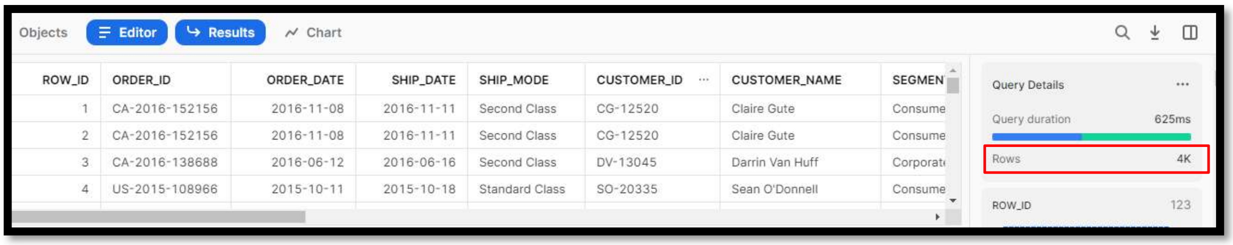
For additional capabilities, use Amazon EventBridge to build event-driven applications at scale using S3 event notifications. Learn more or see EventBridge pricing

Edit

Send notifications to Amazon EventBridge for all events in this bucket

Off

Before Upload Source table has 4k rows :



Objects | Editor | Results | Chart

ROW_ID	ORDER_ID	ORDER_DATE	SHIP_DATE	SHIP_MODE	CUSTOMER_ID	CUSTOMER_NAME	SEGMEN
1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consume
2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consume
3	CA-2016-138688	2016-06-12	2016-06-16	Second Class	DV-13045	Darrin Van Huff	Corporat
4	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consume

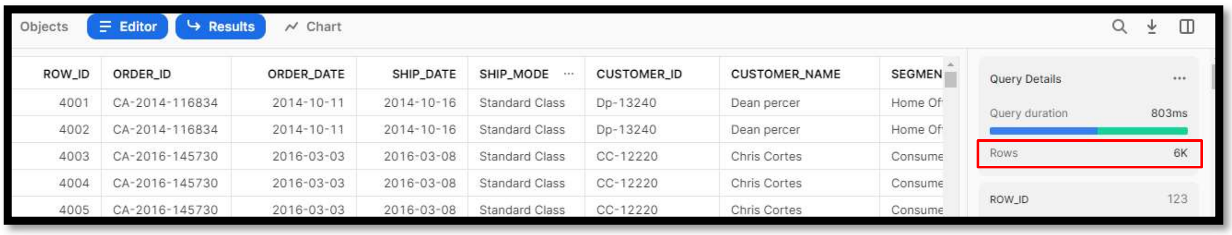
Query Details

Query duration 625ms

Rows 4K

ROW_ID 123

Step 4: After Uploading source file into S3, Fetching all data from source table



Objects | Editor | Results | Chart

ROW_ID	ORDER_ID	ORDER_DATE	SHIP_DATE	SHIP_MODE	CUSTOMER_ID	CUSTOMER_NAME	SEGMEN
4001	CA-2014-116834	2014-10-11	2014-10-16	Standard Class	Dp-13240	Dean percer	Home Of
4002	CA-2014-116834	2014-10-11	2014-10-16	Standard Class	Dp-13240	Dean percer	Home Of
4003	CA-2016-145730	2016-03-03	2016-03-08	Standard Class	CC-12220	Chris Cortes	Consume
4004	CA-2016-145730	2016-03-03	2016-03-08	Standard Class	CC-12220	Chris Cortes	Consume
4005	CA-2016-145730	2016-03-03	2016-03-08	Standard Class	CC-12220	Chris Cortes	Consume

Query Details

Query duration 803ms

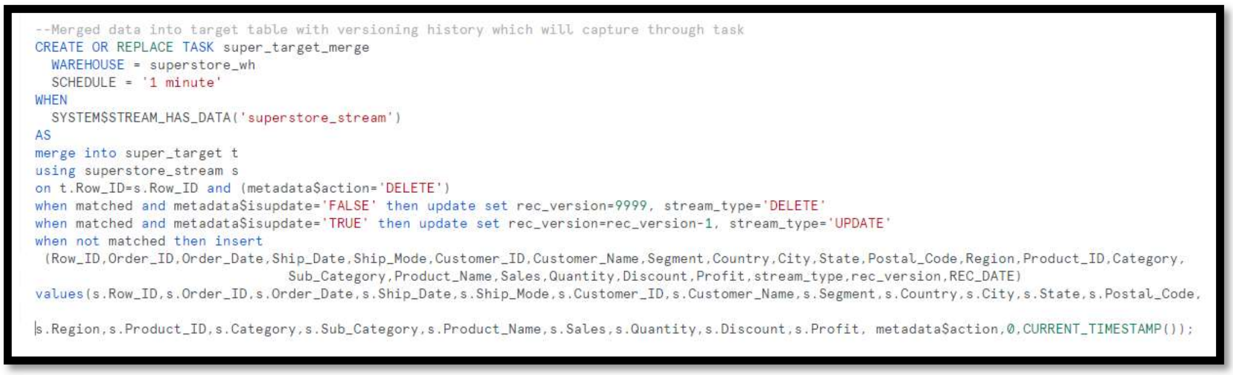
Rows 6K

ROW_ID 123

4. Implement SCD 2 in for the above data flow

Step 1: Created a stream to capture DML changes made to the source table

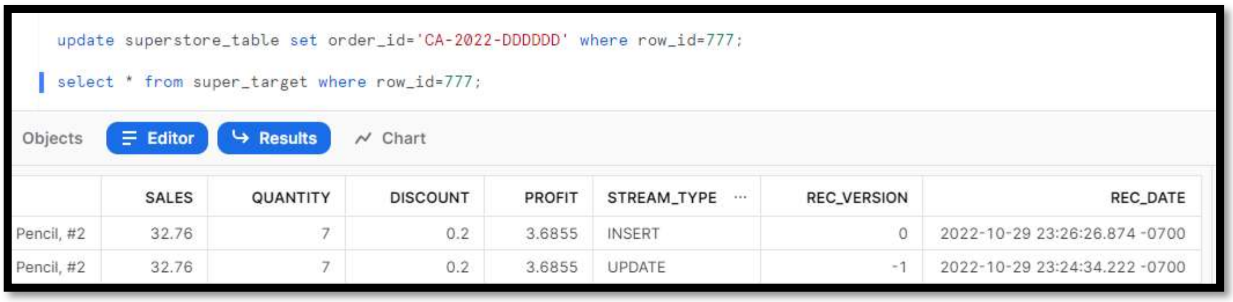
Step 2: Created a task to merge data into target table



```
--Merged data into target table with versioning history which will capture through task
CREATE OR REPLACE TASK super_target_merge
  WAREHOUSE = superstore_wh
  SCHEDULE = '1 minute'
WHEN
  SYSTEM$STREAM_HAS_DATA('superstore_stream')
AS
merge into super_target t
using superstore_stream s
on t.Row_ID=s.Row_ID and (metadata$action='DELETE')
when matched and metadata$isupdate='FALSE' then update set rec_version=9999, stream_type='DELETE'
when matched and metadata$isupdate='TRUE' then update set rec_version=rec_version-1, stream_type='UPDATE'
when not matched then insert
 (Row_ID,Order_ID,Order_Date,Ship_Date,Ship_Mode,Customer_ID,Customer_Name,Segment,Country,City,State,Postal_Code,Region,Product_ID,Category,
                              Sub_Category,Product_Name,Sales,Quantity,Discount,Profit,stream_type,rec_version,REC_DATE)
values(s.Row_ID,s.Order_ID,s.Order_Date,s.Ship_Date,s.Ship_Mode,s.Customer_ID,s.Customer_Name,s.Segment,s.Country,s.City,s.State,s.Postal_Code,

s.Region,s.Product_ID,s.Category,s.Sub_Category,s.Product_Name,s.Sales,s.Quantity,s.Discount,s.Profit, metadata$action,0,CURRENT_TIMESTAMP());
```

Step 3: Displayed the target table that will return all the records with versioning history

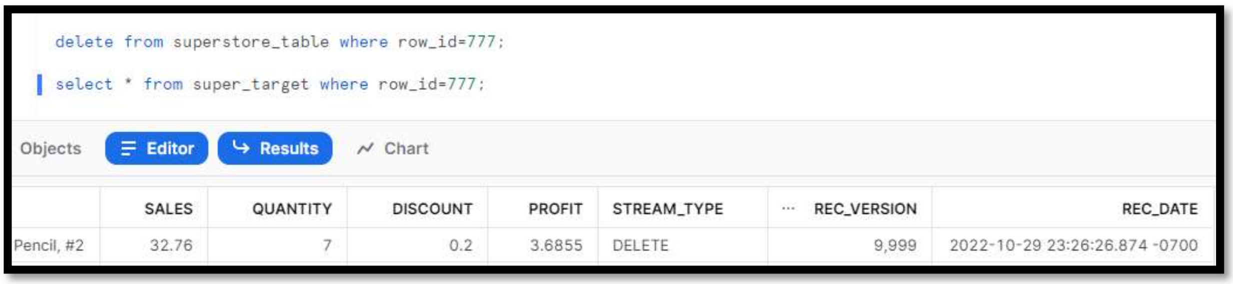
Update:



```
update superstore_table set order_id='CA-2022-DDDDDD' where row_id=777;

select * from super_target where row_id=777;
```

	SALES	QUANTITY	DISCOUNT	PROFIT	STREAM_TYPE	REC_VERSION	REC_DATE
Pencil, #2	32.76	7	0.2	3.6855	INSERT	0	2022-10-29 23:26:26.874 -0700
Pencil, #2	32.76	7	0.2	3.6855	UPDATE	-1	2022-10-29 23:24:34.222 -0700

Delete:



```
delete from superstore_table where row_id=777;

select * from super_target where row_id=777;
```

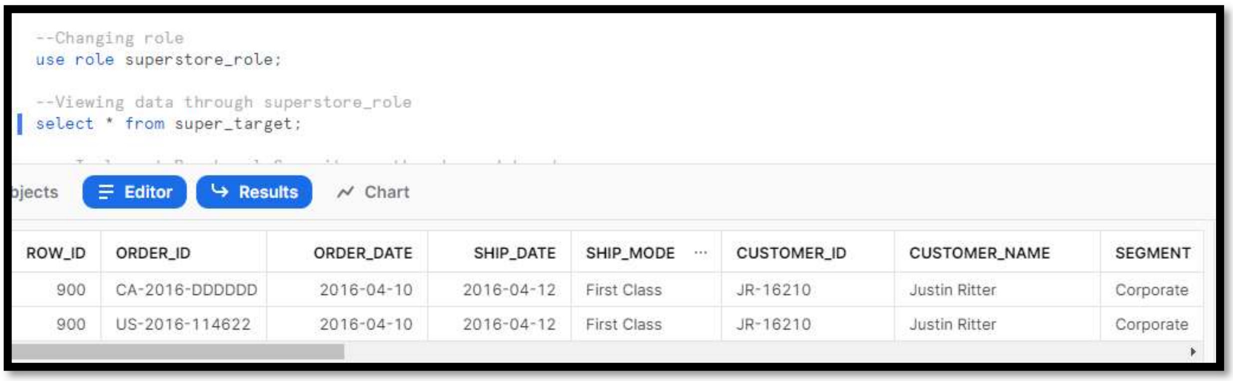
	SALES	QUANTITY	DISCOUNT	PROFIT	STREAM_TYPE	...	REC_VERSION	REC_DATE
Pencil, #2	32.76	7	0.2	3.6855	DELETE		9,999	2022-10-29 23:26:26.874 -0700

5. Implement Row Level and Column Level Security on the above dataset.

1. Column level Security

- Created a new role (superstore_role)
- Created masking policy
- Masked sensitive columns from target table when viewed from any other role.

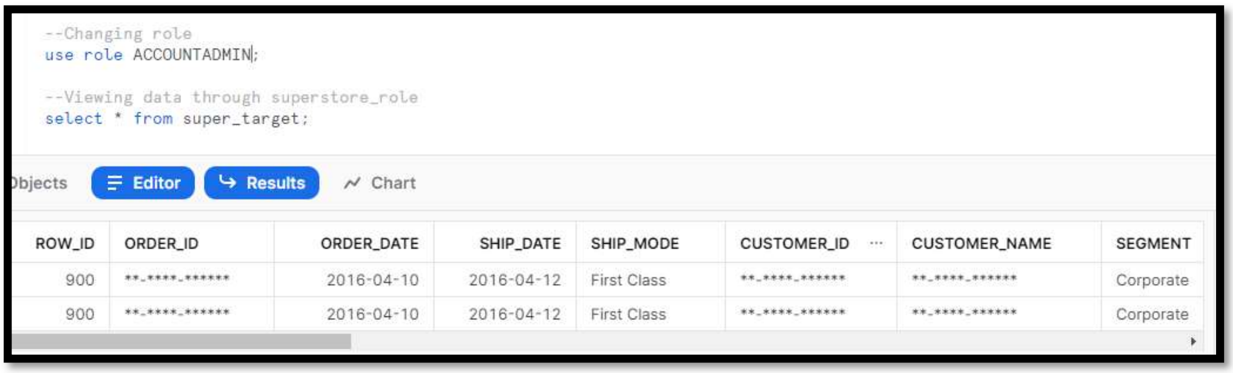
Viewing target table through superstore_role:



```
--Changing role
use role superstore_role;

--Viewing data through superstore_role
select * from super_target;
```

ROW_ID	ORDER_ID	ORDER_DATE	SHIP_DATE	SHIP_MODE	CUSTOMER_ID	CUSTOMER_NAME	SEGMENT
900	CA-2016-DDDDDD	2016-04-10	2016-04-12	First Class	JR-16210	Justin Ritter	Corporate
900	US-2016-114622	2016-04-10	2016-04-12	First Class	JR-16210	Justin Ritter	Corporate

Viewing target table through other role:



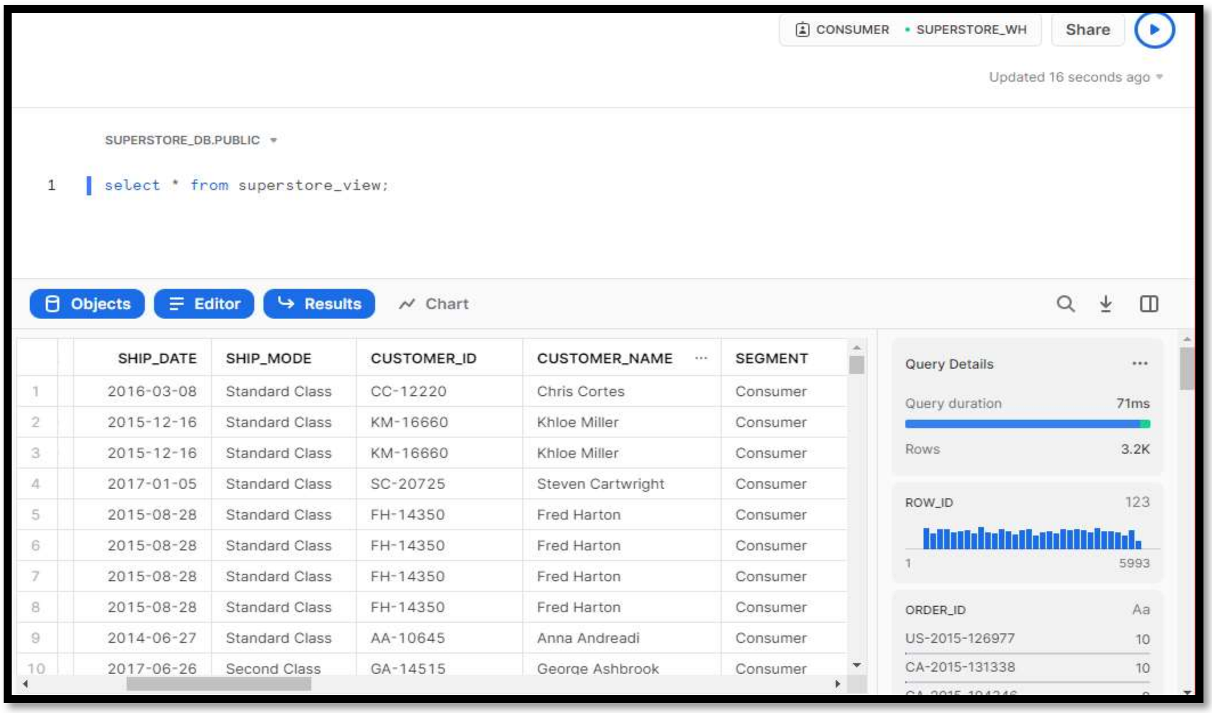
```
--Changing role
use role ACCOUNTADMIN;

--Viewing data through superstore_role
select * from super_target;
```

ROW_ID	ORDER_ID	ORDER_DATE	SHIP_DATE	SHIP_MODE	CUSTOMER_ID	CUSTOMER_NAME	SEGMENT
900	**-****-******	2016-04-10	2016-04-12	First Class	**-****-******	**-****-******	Corporate
900	**-****-******	2016-04-10	2016-04-12	First Class	**-****-******	**-****-******	Corporate

2. Row level Security:

- Created roles based on a relevant column
- Created users with their default roles
- Granted secure view of respective entries/rows to the roles.

Viewing superstore data for Consumer role:

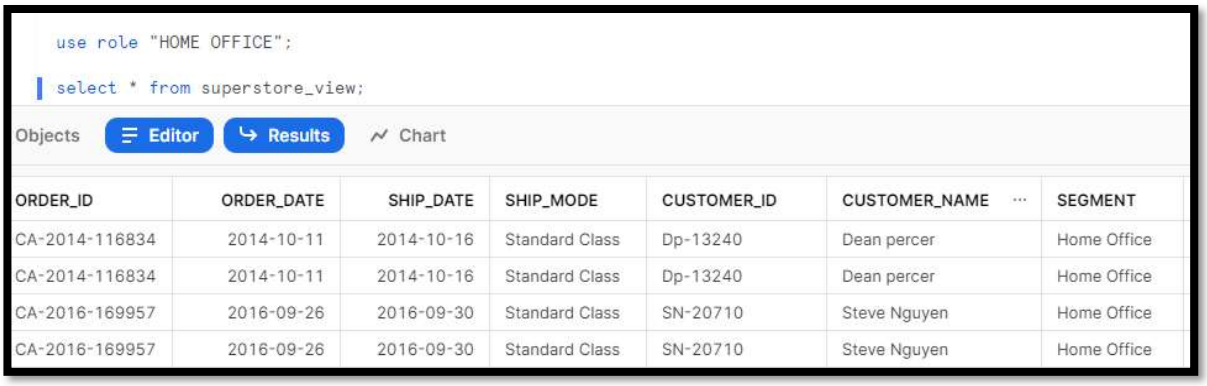


```
select * from superstore_view;
```

	SHIP_DATE	SHIP_MODE	CUSTOMER_ID	CUSTOMER_NAME	SEGMENT
1	2016-03-08	Standard Class	CC-12220	Chris Cortes	Consumer
2	2015-12-16	Standard Class	KM-16660	Khloe Miller	Consumer
3	2015-12-16	Standard Class	KM-16660	Khloe Miller	Consumer
4	2017-01-05	Standard Class	SC-20725	Steven Cartwright	Consumer
5	2015-08-28	Standard Class	FH-14350	Fred Harton	Consumer
6	2015-08-28	Standard Class	FH-14350	Fred Harton	Consumer
7	2015-08-28	Standard Class	FH-14350	Fred Harton	Consumer
8	2015-08-28	Standard Class	FH-14350	Fred Harton	Consumer
9	2014-06-27	Standard Class	AA-10645	Anna Andreadi	Consumer
10	2017-06-26	Second Class	GA-14515	George Ashbrook	Consumer

Query Details
Query duration 71ms
Rows 3.2K
ROW_ID 123
1 5993
ORDER_ID Aa
US-2015-126977 10
CA-2015-131338 10

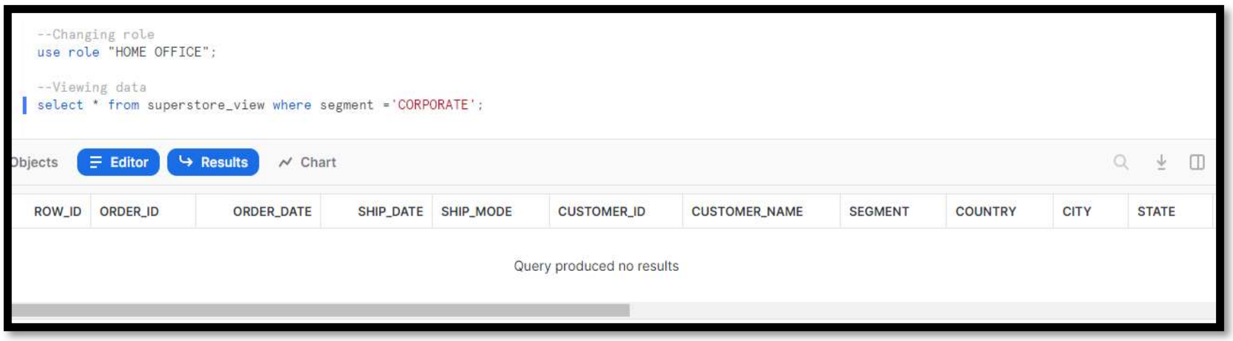
Viewing superstore data for Home Office user:



```
use role "HOME OFFICE";

select * from superstore_view;
```

ORDER_ID	ORDER_DATE	SHIP_DATE	SHIP_MODE	CUSTOMER_ID	CUSTOMER_NAME	SEGMENT
CA-2014-116834	2014-10-11	2014-10-16	Standard Class	Dp-13240	Dean percer	Home Office
CA-2014-116834	2014-10-11	2014-10-16	Standard Class	Dp-13240	Dean percer	Home Office
CA-2016-169957	2016-09-26	2016-09-30	Standard Class	SN-20710	Steve Nguyen	Home Office
CA-2016-169957	2016-09-26	2016-09-30	Standard Class	SN-20710	Steve Nguyen	Home Office

Negative testing :



```
--Changing role
use role "HOME OFFICE";

--Viewing data
select * from superstore_view where segment ='CORPORATE';
```

ROW_ID	ORDER_ID	ORDER_DATE	SHIP_DATE	SHIP_MODE	CUSTOMER_ID	CUSTOMER_NAME	SEGMENT	COUNTRY	CITY	STATE

Query produced no results