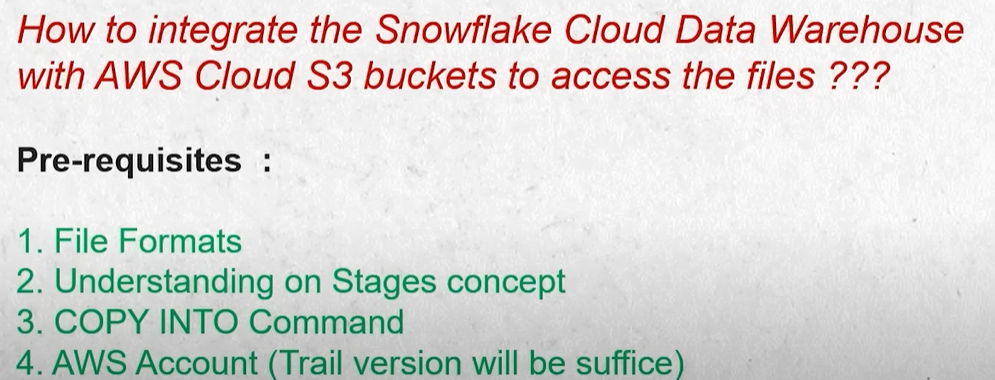
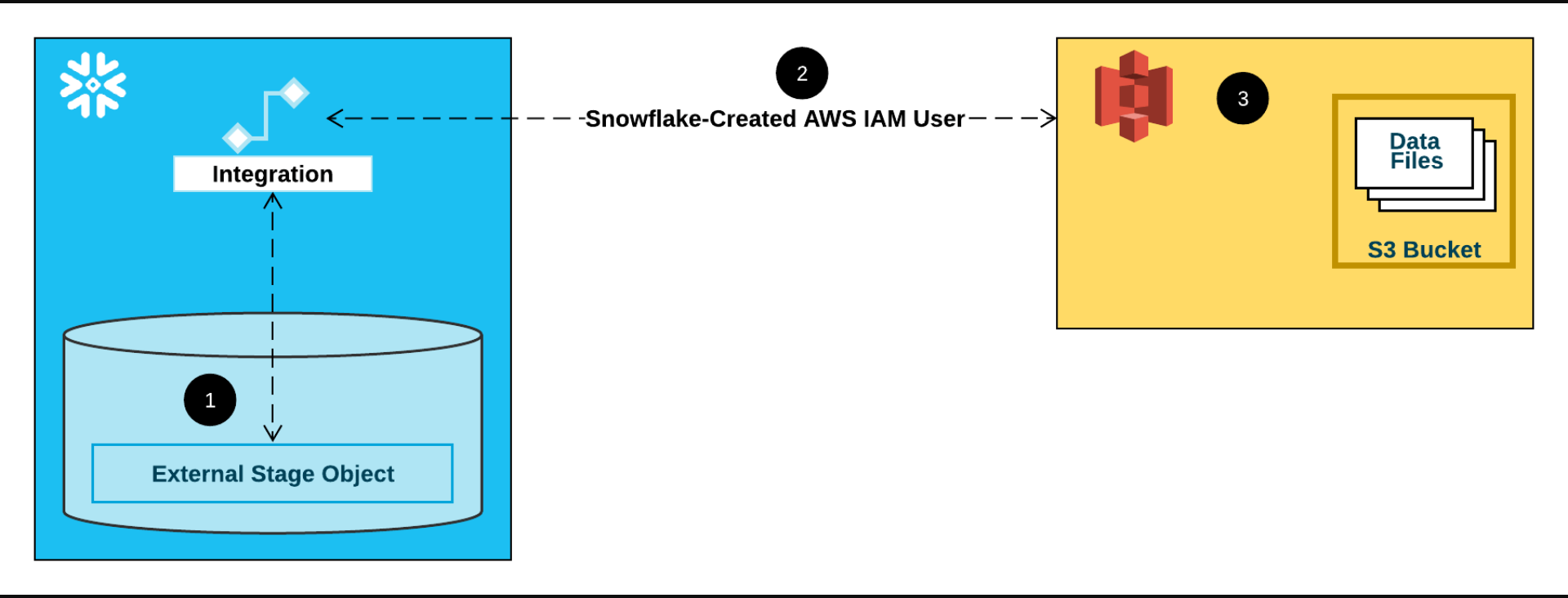
<https://docs.snowflake.com/en/user-guide/data-load-s3-config-storage-integration>







**Create S3 bucket:**

S3 URI: s3://demo-03012024/contacts1.csv

**Create IAM Role:**

Policy: JSON

{

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

"Statement": [

{

"Sid": "Statement1",

"Effect": "Allow",

"Action": [

"s3:PutObject",

"s3:GetObject",

"s3:GetObjectVersion",

"s3:DeleteObject",

"s3:DeleteObjectVersion"

],

"Resource": "arn:aws:s3:::demo-03012024/\*"

},

{

"Effect": "Allow",

"Action": [

"s3:ListBucket",

"s3:GetBucketLocation"

],

"Resource": "arn:aws:s3:::demo-03012024",

"Condition": {

"StringLike": {

"s3:prefix": [

"\*"

]

}

}

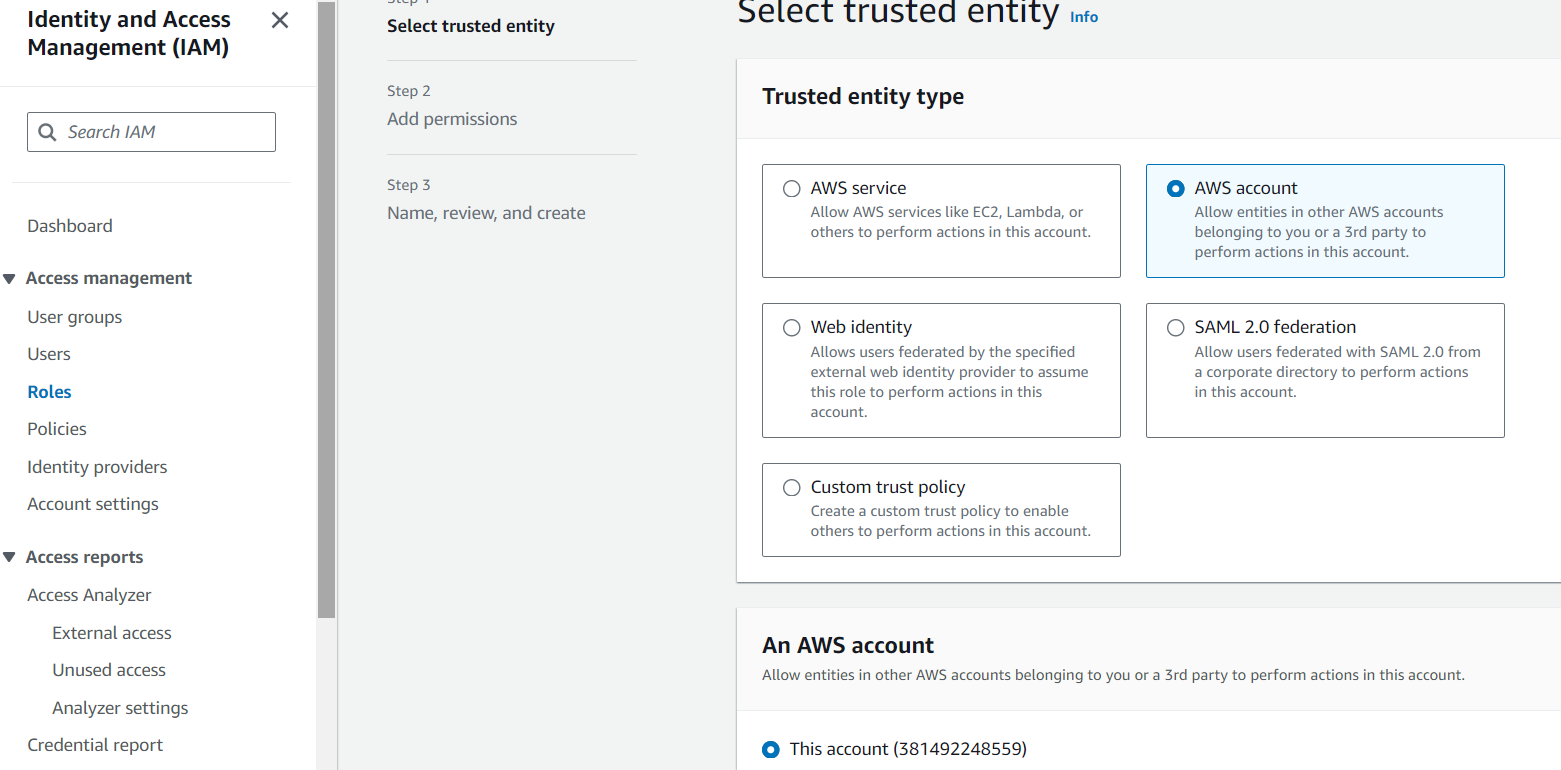
}

]

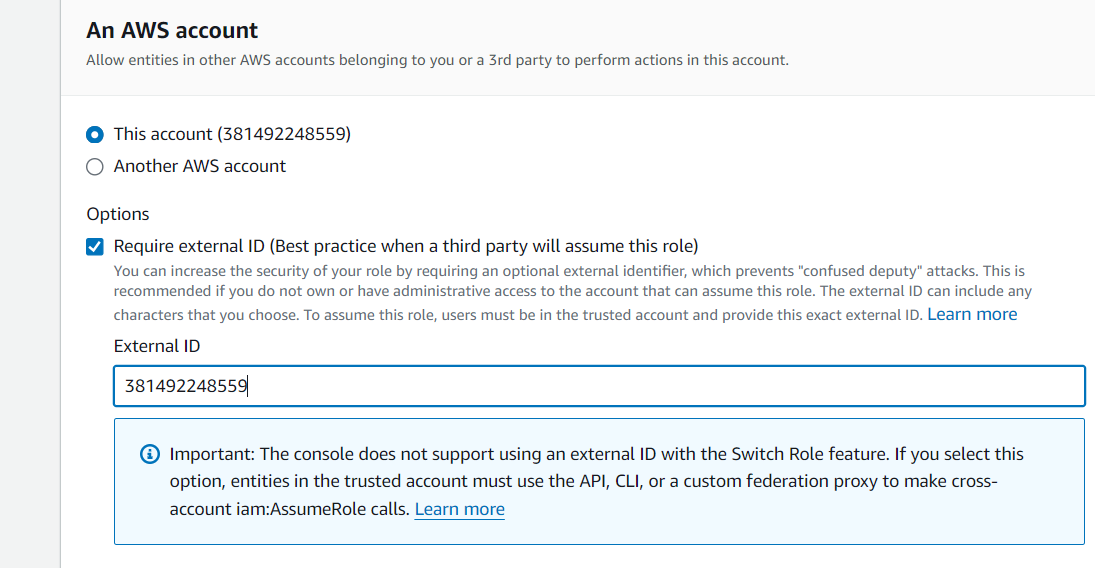
}

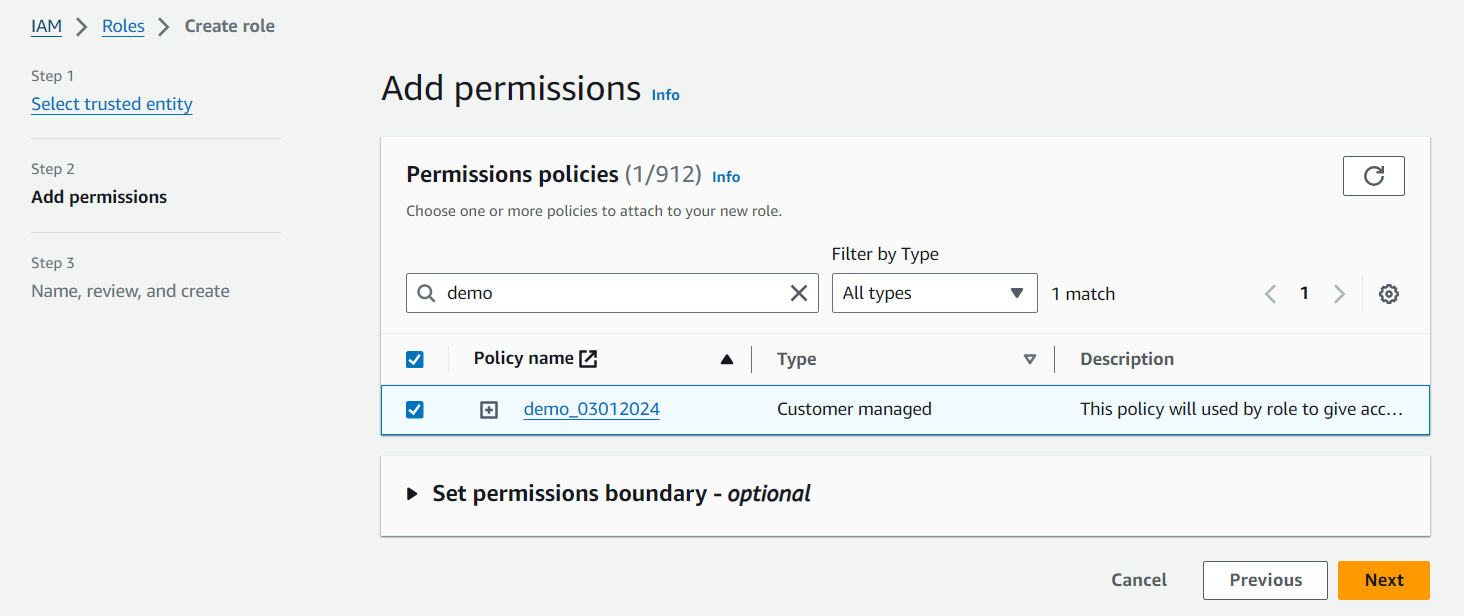
**Create Role:**

**Role ARN:** arn:aws:iam::381492248559:role/demo\_03012024



This account (381492248559)





**Go to Snowflake:**

create database DEMO;

create schema demo\_03012024;

**create integration**

CREATE or REPLACE STORAGE INTEGRATION aws\_s3\_integration

TYPE = EXTERNAL\_STAGE

STORAGE\_PROVIDER = 'S3'

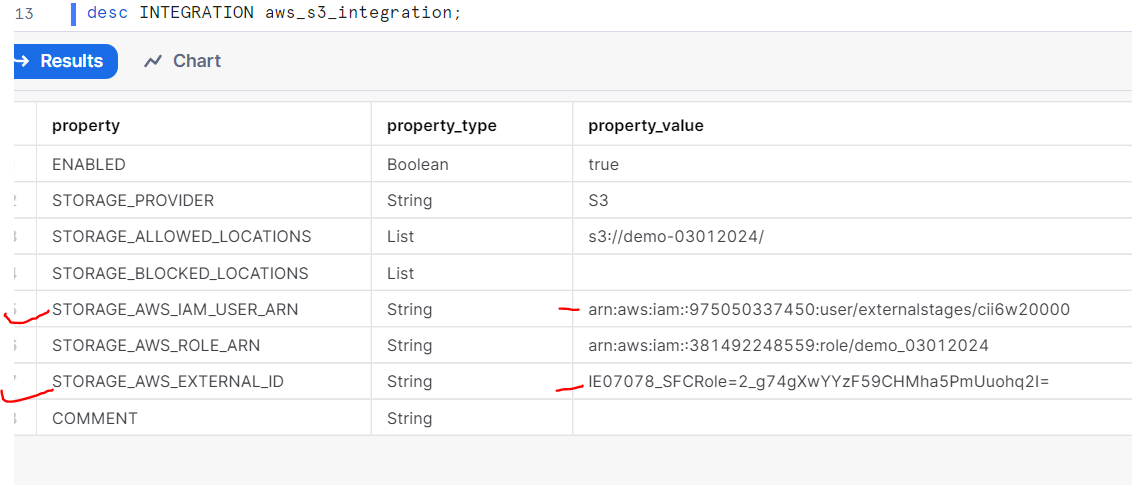
ENABLED = TRUE

STORAGE\_AWS\_ROLE\_ARN = 'arn:aws:iam::381492248559:role/demo\_03012024'

STORAGE\_ALLOWED\_LOCATIONS = ('s3://demo-03012024/');

show integrations;

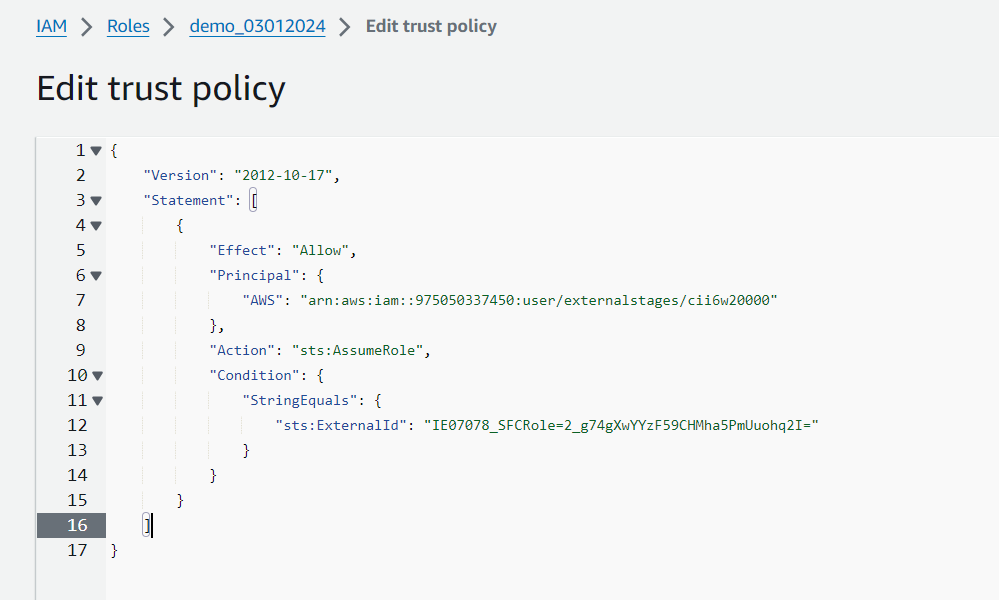
desc INTEGRATION aws\_s3\_integration;



**STORAGE\_AWS\_IAM\_USER\_ARN: arn:aws:iam::975050337450:user/externalstages/cii6w20000**

**STORAGE\_AWS\_ROLE\_ARN: arn:aws:iam::381492248559:role/demo\_03012024**

**Change trust policy**



**In snowflake:**

GRANT USAGE ON INTEGRATION aws\_s3\_integration TO ROLE accountadmin;

**File Format:**

create or replace file format demo\_format

TYPE = 'CSV'

FIELD\_DELIMITER = '|'

SKIP\_HEADER = 1;

**Create Stage:**

CREATE or REPLACE STAGE my\_s3\_stage

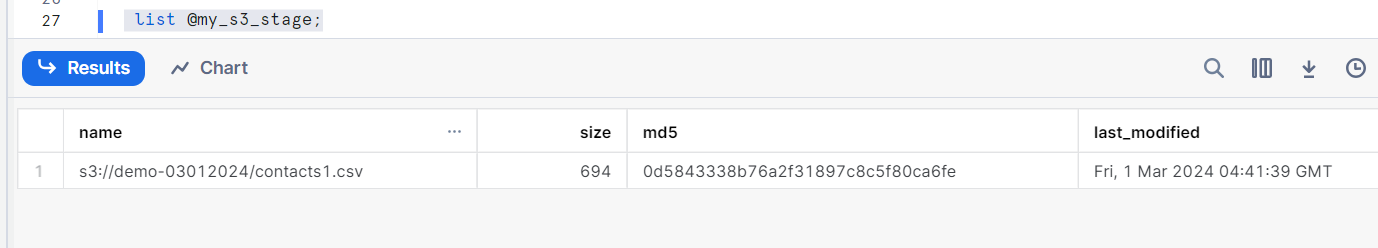
STORAGE\_INTEGRATION = aws\_s3\_integration

URL = 's3://demo-03012024/'

FILE\_FORMAT = demo\_format;

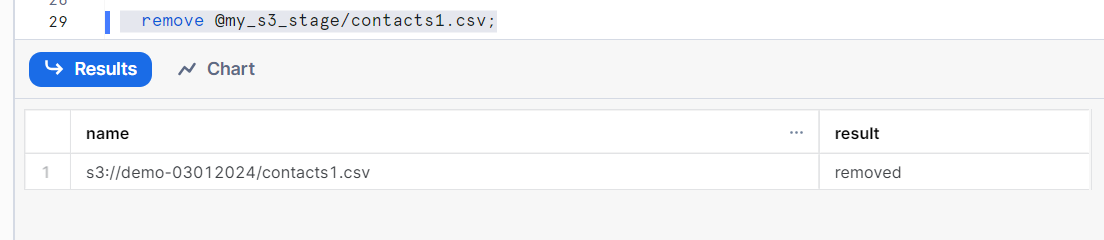
**To Check file available or not in s3**

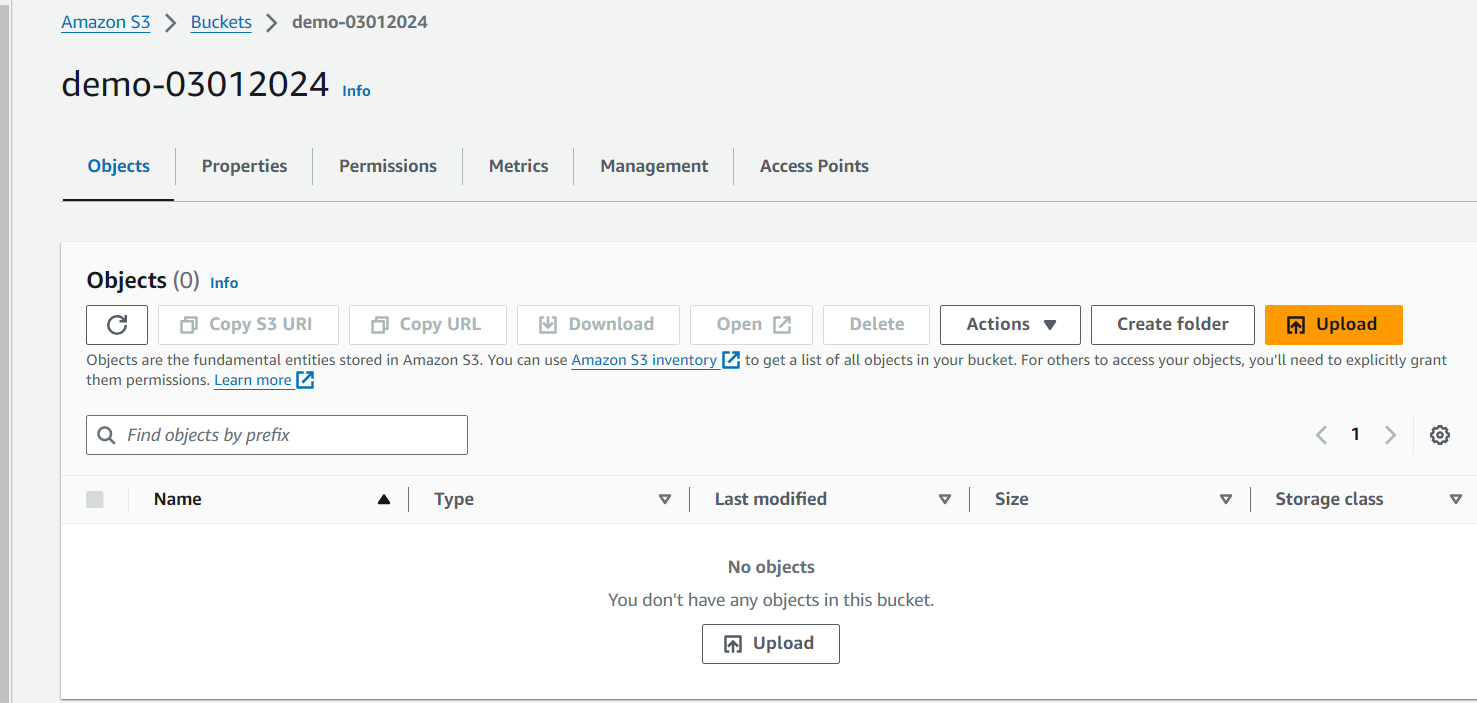
list @my\_s3\_stage;



**To remove file from aws s3**

remove @my\_s3\_stage/contacts1.csv;





**Create a temporary table in Snowflake:**

create or replace table contacts\_csv

(

ID integer,

Last\_Name String,

First\_Name String,

Company String,

Email String,

Workplace String,

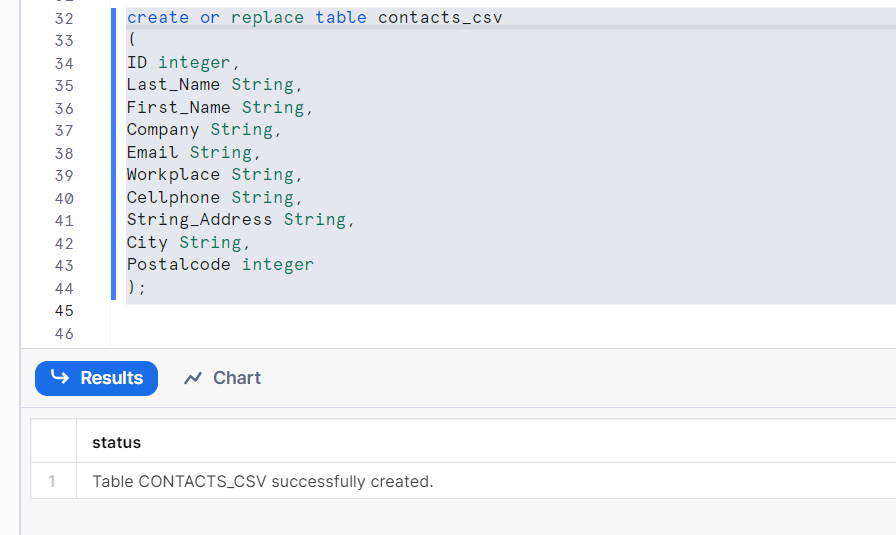
Cellphone String,

String\_Address String,

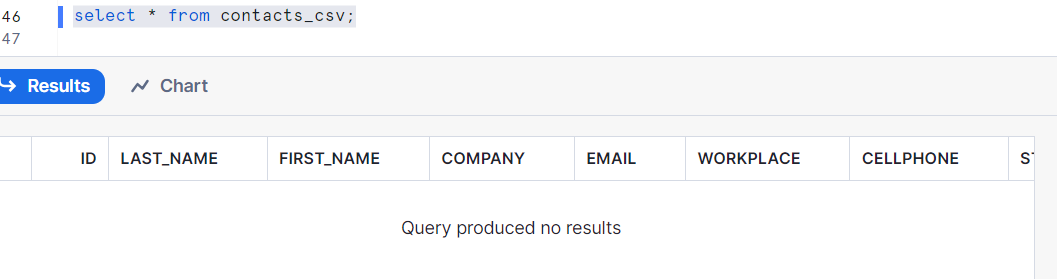
City String,

Postalcode integer

);



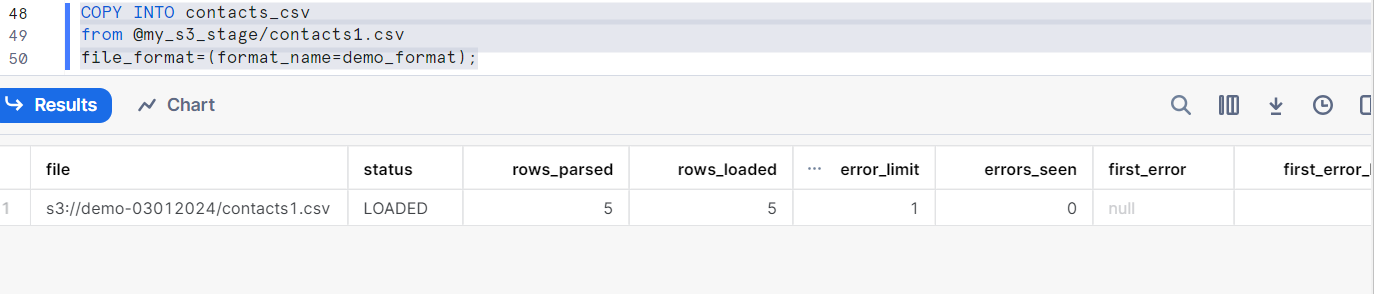
**Data is not present in this table. So, now we can upload data into this table from AWS s3 where we have uploaded file. But we have already integrate to snowflake staging. We can leverage snowflake stage to copy data into that table.**



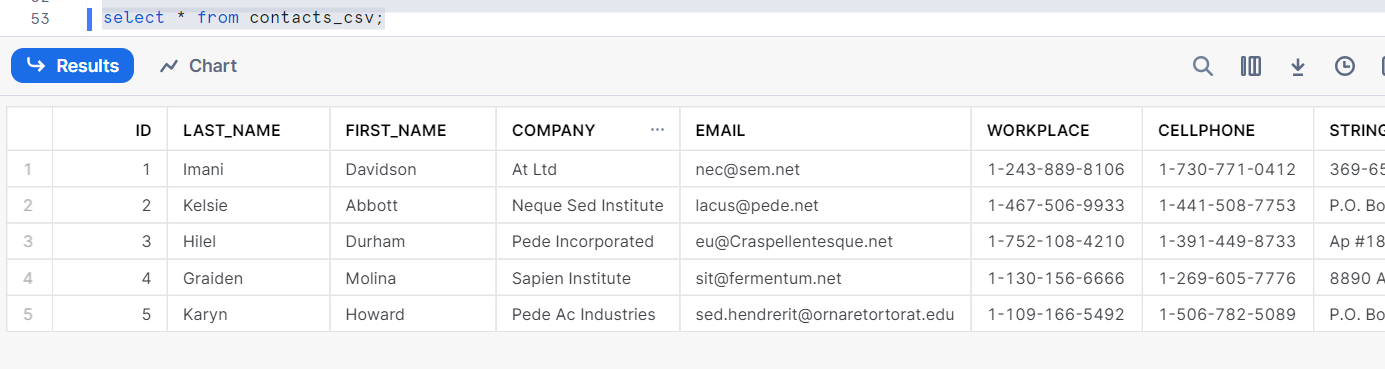
COPY INTO contacts\_csv

from @my\_s3\_stage/contacts1.csv

file\_format=(format\_name=demo\_format);



**Now we can check data is present in our table or not.**



**-- Scenario:1**

1]

COPY INTO contacts\_csv

from @my\_s3\_stage/

file\_format=(format\_name=demo\_format)

on\_error='skip\_file'; -- skip the whole file

2]

COPY INTO contacts\_csv

from @my\_s3\_stage/

file\_format=(format\_name=demo\_format)

on\_error='Continue'; -- skip only the bad records and load rest of the record.

3]

COPY INTO contacts\_csv

from @my\_s3\_stage/

file\_format=(format\_name=demo\_format)

on\_error='abort'; -- stope loading data when first error occured.

**--** **Scenario:2**

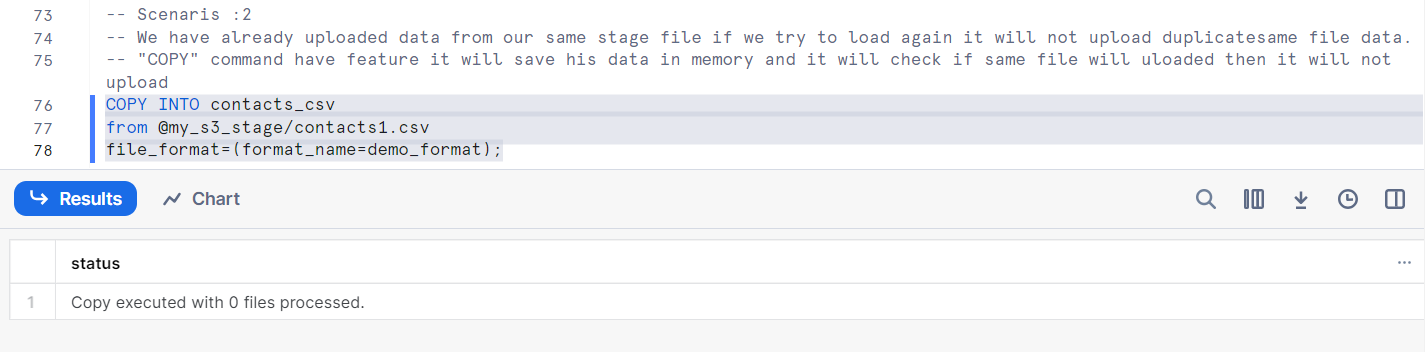
-- We have already uploaded data from our same stage file if we try to load again it will not upload duplicate same file data. "COPY" command have feature it will save his data in memory and it will check if same file will uploaded then it will not upload. "COPY" command maintain history for certain time.

COPY INTO contacts\_csv

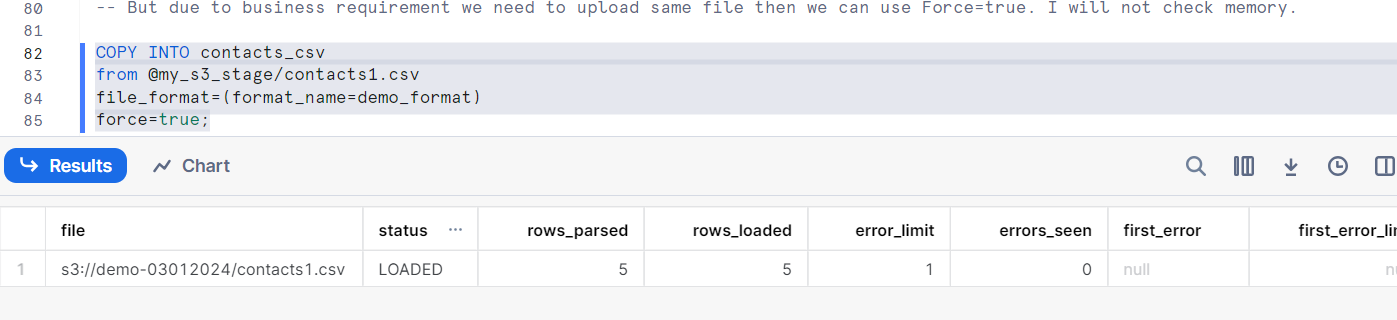
from @my\_s3\_stage/contacts1.csv

file\_format=(format\_name=demo\_format);

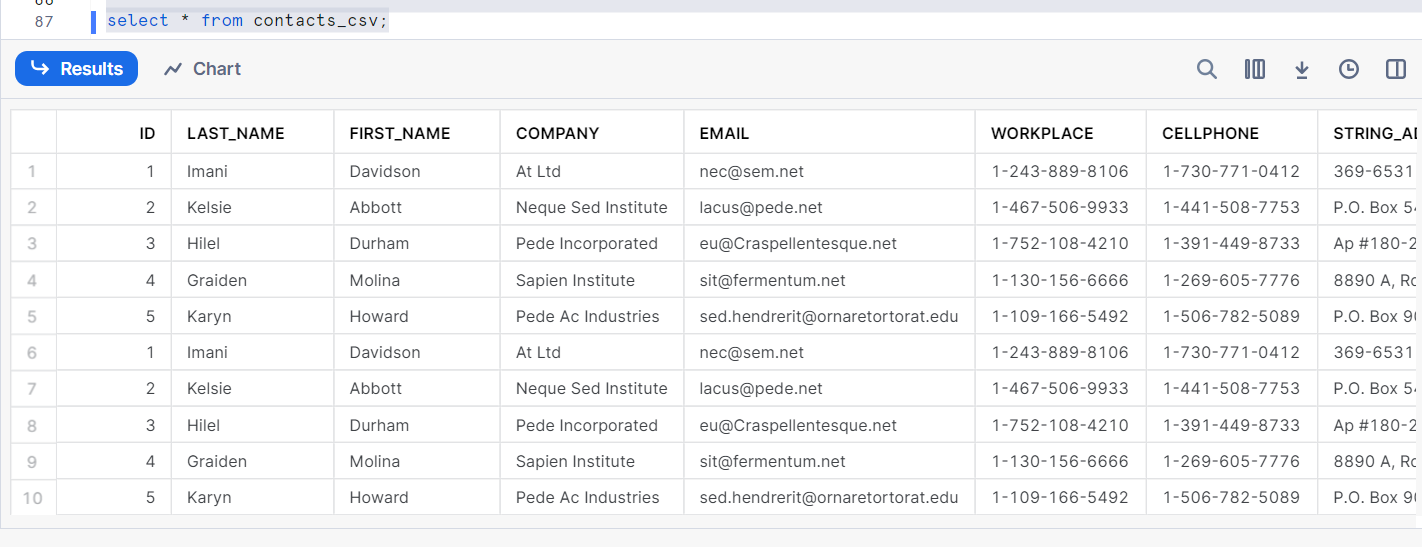
Output: Copy executed with 0 files processed.



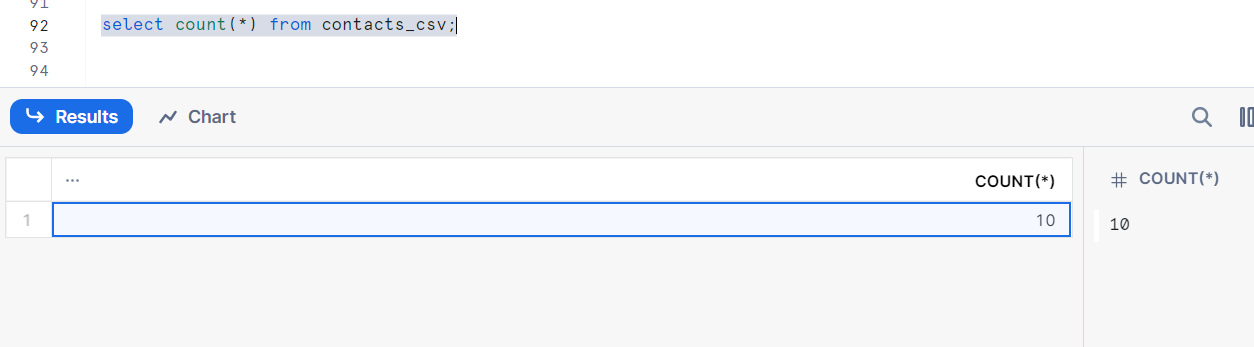
But due to business requirement we need to upload same file then we can use “ Force=true”. I will not check memory.



select \* from contacts\_csv; -- duplicate record uploaded into the table.



select count(\*) from contacts\_csv;



**-- Scenario :3**

-- After succesfully uploading the file we want to remove the that file then we can leverage below feature and file also will purge from AWS S3.

COPY INTO contacts\_csv

from @my\_s3\_stage/contacts1.csv

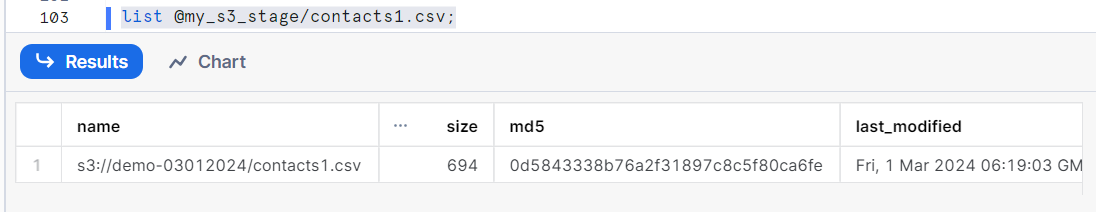
file\_format=(format\_name=demo\_format)

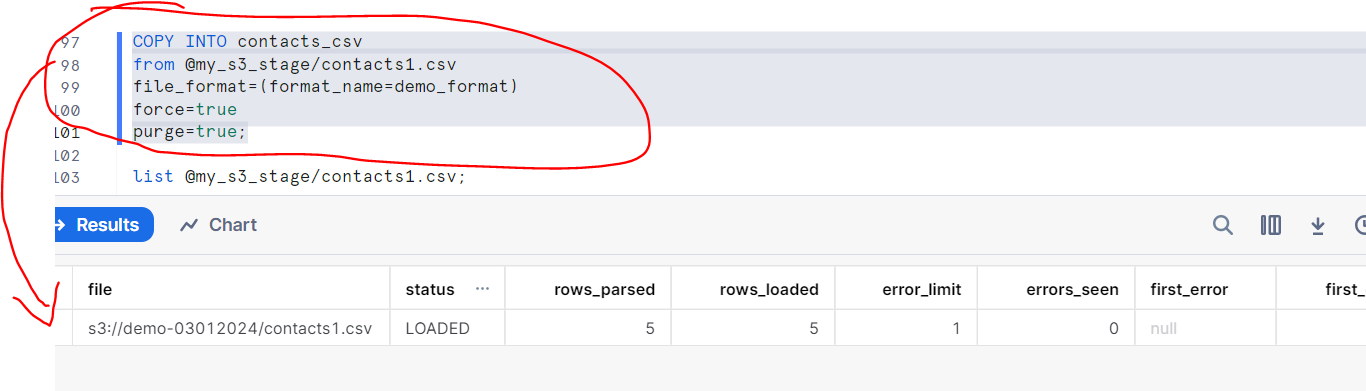
force=true

purge=true;

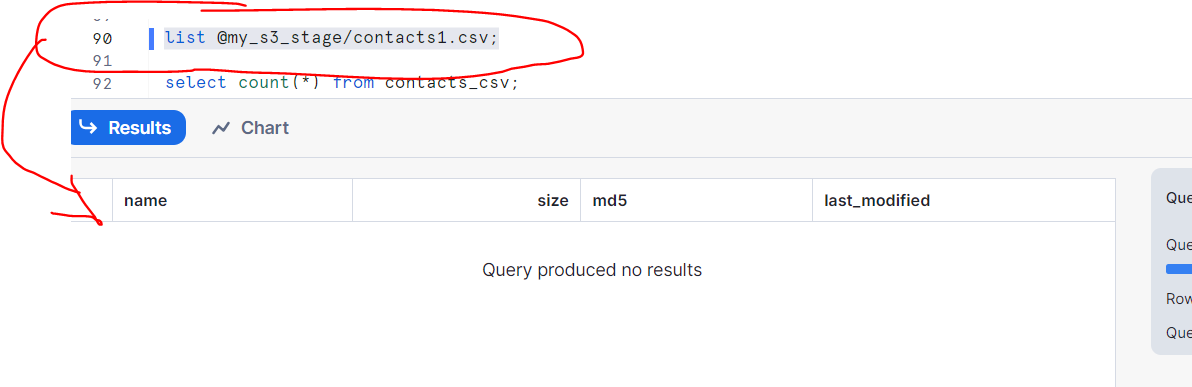
**Now check file still available or not.**

list @my\_s3\_stage/contacts1.csv;





**File is not present.**



**File will also deleted from AWS S3.**

