create warehouse assign13;

USE warehouse assign13;

create database assign13\_db;

use database assign13\_db;

CREATE OR REPLACE TRANSIENT TABLE assign13\_db.PUBLIC.CUSTOMER\_TEST

AS

SELECT \* FROM "SNOWFLAKE\_SAMPLE\_DATA"."TPCDS\_SF100TCL"."CUSTOMER";

select \* from assign13\_db.PUBLIC.CUSTOMER\_TEST;

CREATE OR REPLACE STORAGE INTEGRATION s3\_csv\_int

TYPE = EXTERNAL\_STAGE

STORAGE\_PROVIDER = S3

ENABLED = TRUE

STORAGE\_AWS\_ROLE\_ARN ='arn:aws:iam::339713157319:role/Role13'

STORAGE\_ALLOWED\_LOCATIONS =('s3://snowflakenew13/folder13/');

create or replace file format my\_csv\_unload\_format

type = csv field\_delimiter = ',' skip\_header = 1 null\_if = ('NULL', 'null') empty\_field\_as\_null = true compression = gzip;

create or replace stage my\_s3\_assignment

storage\_integration = s3\_csv\_int

url = ' s3://snowflakenew13/folder13//';

desc integration s3\_csv\_int;

COPY INTO @assign13\_db.PUBLIC.my\_s3\_assignment/Customer\_folder/

from

assign13\_db.PUBLIC.CUSTOMER\_TEST;

list @assign13\_db.PUBLIC.my\_s3\_assignment/Customer\_data/;

create or replace file format MY\_CSV\_FORMAT

type = 'csv';

SELECT $1 C\_CUSTOMER\_SK,

$2 C\_CUSTOMER\_ID ,

$3 C\_CURRENT\_CDEMO\_SK ,

$4 C\_CURRENT\_HDEMO\_SK ,

$5 C\_CURRENT\_ADDR\_SK,

$6 C\_FIRST\_SHIPTO\_DATE\_SK ,

$7 C\_FIRST\_SALES\_DATE\_SK ,

$8 C\_SALUTATION ,

$9 C\_FIRST\_NAME ,

$10 C\_LAST\_NAME,

$11 C\_PREFERRED\_CUST\_FLAG ,

$12 C\_BIRTH\_DAY ,

$13 C\_BIRTH\_MONTH ,

$14 C\_BIRTH\_YEAR,

$16 C\_LOGIN ,

$17 C\_EMAIL\_ADDRESS ,

$18 C\_LAST\_REVIEW\_DATE

FROM @assign13\_db.PUBLIC.my\_s3\_assignment/Customer\_data/

(file\_format => MY\_CSV\_FORMAT);

SELECT $1 C\_CUSTOMER\_SK,

$2 C\_CUSTOMER\_ID ,

$3 C\_CURRENT\_CDEMO\_SK ,

$4 C\_CURRENT\_HDEMO\_SK ,

$5 C\_CURRENT\_ADDR\_SK,

$6 C\_FIRST\_SHIPTO\_DATE\_SK ,

$7 C\_FIRST\_SALES\_DATE\_SK ,

$8 C\_SALUTATION ,

$9 C\_FIRST\_NAME ,

$10 C\_LAST\_NAME,

$11 C\_PREFERRED\_CUST\_FLAG ,

$12 C\_BIRTH\_DAY ,

$13 C\_BIRTH\_MONTH ,

$14 C\_BIRTH\_YEAR,

$16 C\_LOGIN ,

$17 C\_EMAIL\_ADDRESS ,

$18 C\_LAST\_REVIEW\_DATE

FROM @assign13\_db.PUBLIC.my\_s3\_assignment/Customer\_folder/

(file\_format => assign13\_db.PUBLIC.MY\_CSV\_FORMAT)

WHERE C\_CUSTOMER\_SK ='64596949';

SELECT $9 C\_FIRST\_NAME,$10 C\_LAST\_NAME,COUNT(\*)

FROM @assign13\_db.PUBLIC.my\_s3\_assignment/Customer\_folder/

(file\_format => assign13\_db.PUBLIC.MY\_CSV\_FORMAT)

GROUP BY $9,$10;

CREATE OR REPLACE VIEW CUSTOMER\_DATA

AS

SELECT $1 C\_CUSTOMER\_SK,

$2 C\_CUSTOMER\_ID ,

$3 C\_CURRENT\_CDEMO\_SK ,

$4 C\_CURRENT\_HDEMO\_SK ,

$5 C\_CURRENT\_ADDR\_SK,

$6 C\_FIRST\_SHIPTO\_DATE\_SK ,

$7 C\_FIRST\_SALES\_DATE\_SK ,

$8 C\_SALUTATION ,

$9 C\_FIRST\_NAME ,

$10 C\_LAST\_NAME,

$11 C\_PREFERRED\_CUST\_FLAG ,

$12 C\_BIRTH\_DAY ,

$13 C\_BIRTH\_MONTH ,

$14 C\_BIRTH\_YEAR,

$16 C\_LOGIN ,

$17 C\_EMAIL\_ADDRESS ,

$18 C\_LAST\_REVIEW\_DATE

FROM @assign13\_db.PUBLIC.my\_s3\_assignment/Customer\_folder/

(file\_format => assign13\_db.PUBLIC.MY\_CSV\_FORMAT);

SELECT \* FROM CUSTOMER\_DATA;

Create or replace transient table CUSTOMER\_SNOWFLAKE\_TABLE

AS

SELECT \* FROM CUSTOMER\_TEST limit 10000;

SELECT B.\*

FROM CUSTOMER\_SNOWFLAKE\_TABLE B

LEFT OUTER JOIN

CUSTOMER\_DATA A

ON

A.C\_CUSTOMER\_SK = B.C\_CUSTOMER\_SK;

COPY INTO @assign13\_db.PUBLIC.my\_s3\_assignment/Customer\_folder2/

from(

SELECT B.\*

FROM CUSTOMER\_SNOWFLAKE\_TABLE B

LEFT OUTER JOIN

CUSTOMER\_DATA A

ON

A.C\_CUSTOMER\_SK = B.C\_CUSTOMER\_SK);

COPY INTO @assign13\_db.PUBLIC.my\_s3\_assignment/Customer\_joined\_data/

from(

SELECT B.\*

FROM CUSTOMER\_SNOWFLAKE\_TABLE B

LEFT OUTER JOIN

CUSTOMER\_DATA A

ON

A.C\_CUSTOMER\_SK = B.C\_CUSTOMER\_SK

);

Now we can directly query data from s3 through view. What is the disadvantage of using   
this approach ? Can you see partitions being scanned in the backend ?  
ANswer:  
DISADVANTAGE is this view created without partitions hence it will scan through all the data and will take time to return the results.

Now we successfully joined data in s3 with snowflake table. It may look simple but this   
approach has lot of potential. Can you mention few below,  
Answer:  
Once set up, storing data in S3 with Snowflake doesn't cost anything extra. S3 stores data in compressed form, which saves storage space and reduces costs. This setup only needs to be configured once for ongoing benefits.

How many partitions got scanned from snowflake table: 355 partitions..

 -- 6.--------------------------ADVANTAGES AND DISADVANTAGES----------------------------------------

-- PROS  
Storage Capacity:Snowflake runs on Microsoft Azure, offering scalable and user-friendly cloud storage solutions ideal for handling large amounts of data.

Multi-Cloud Support:You can deploy Snowflake on Azure, Google Cloud, and AWS, allowing seamless integration across different cloud platforms to meet diverse business needs.

Server Scalability:Unlike traditional data warehouses, Snowflake operates in the cloud, enabling flexible scaling of computing resources based on demand without needing to invest in physical servers.

Security Features: Snowflake provides robust security measures such as IP whitelisting, two-factor authentication, and AES 256 encryption to protect sensitive data during storage and transmission.

Performance Optimization:Snowflake databases are designed for optimal performance without constant monitoring, allowing users to organize and process data efficiently.

Disaster Recovery:Snowflake ensures data availability through replication across multiple data centers, providing reliable access to data in case of emergencies.

Scalable Performance:Snowflake clusters adjust dynamically to handle varying workloads, ensuring consistent performance even during peak usage periods.

-- CONS  
Unstructured Data Support:Snowflake currently supports semi-structured and structured data types, with plans for future integration of unstructured data support.

Bulk Data Load Challenges:Migrating data to Snowflake can be complex. While Snowpipe offers continuous loading, alternative solutions like Zuar Runner may be more effective for managing large-scale data migration and automation.

No Data Constraints:Snowflake's scalability and pay-as-you-go model have no predefined limits, which can lead to unexpected costs if usage exceeds expectations, particularly in computing and storage