**SNOWFLAKE**

ASSIGNMENT 3

A close-up of a logo

Description automatically generated

A white text with black text

Description automatically generated

A diagram of a snowflake

Description automatically generated

A white background with black text

Description automatically generated

User created with Access key ID and Access key: AKIA47CRXTAY3TSGQCHK – Active

|  |
| --- |
| Access key ID: |
| **AKIA47CRXTAY3TSGQCHK**   |  | | --- | | Secret access key: | | **53vJxO+V9pZ0y5erwx+UIpkF4SB5nvFBtlJQrXVW** | |
|  |

A screenshot of a computer

Description automatically generated

A close up of a text

Description automatically generated

**New S3 bucket created.**

A screenshot of a computer

Description automatically generated

A close up of words

Description automatically generated

use warehouse customer;

use customer;

USE ROLE ACCOUNTADMIN;

--creating external stage using Access Key and AWS Key ID

CREATE OR REPLACE STAGE customer.PUBLIC.S3\_JSON

URL='s3://datapipeline3/json\_data/'

CREDENTIALS=(AWS\_KEY\_ID='AKIA47CRXTAY3TSGQCHK' AWS\_SECRET\_KEY='53vJxO+V9pZ0y5erwx+UIpkF4SB5nvFBtlJQrXVW');

**EXECUTION:**

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A close up of a sign

Description automatically generated

--creating variant table

create table customer.public.person\_sourcedata(source\_data variant);

A screenshot of a computer

Description automatically generated

--create json file format

create or replace file format customer.public.json\_data

type = 'json';

A screenshot of a computer

Description automatically generated



**Before snowpipe creation, I have uploaded the person.json file into S3 bucket folder**

A screenshot of a computer

Description automatically generated

**--creating a snow pipe with auto ingest enabled**

create or replace pipe customer.public.person\_snow\_pipe auto\_ingest=true as

copy into person\_sourcedata

from @customer.PUBLIC.S3\_JSON

file\_format=(format\_name='customer.public.json\_data')

on\_error = CONTINUE;

A screenshot of a computer code

Description automatically generated

**SNOW PIPE** successfully created

A screenshot of a computer

Description automatically generated



--**First creating event notification in AWS console**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Selecting the destination as SQS Queue and copy the ARN from Snow pipe description:**

A screenshot of a computer

Description automatically generated

**Copy the above ARN into SQS queue in AWS console:**

A screenshot of a computer

Description automatically generated

**Successfully created event notification:**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

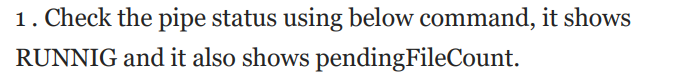
Description automatically generated

Black text on a white background

Description automatically generated

A black text on a white background

Description automatically generated



A screenshot of a computer

Description automatically generated

**Snow Pipe is successfully executing and in Running state**

A white background with black text

Description automatically generated

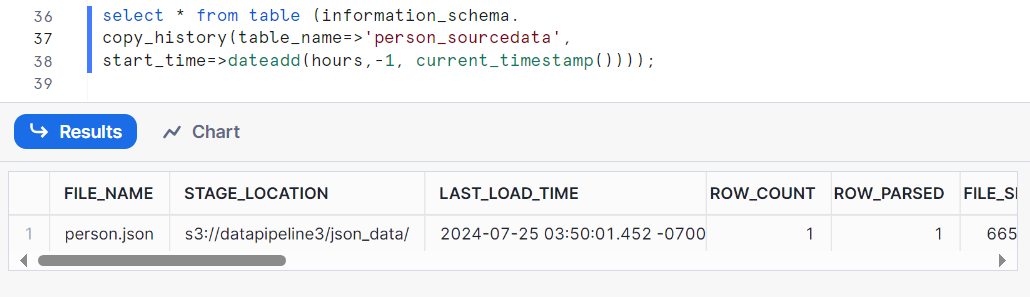
Before this I am deleting the existing JSON file in S3 bucket and uploading the person.json file again. As soon as I upload, **s3 SQS notification** will trigger the **Snowpipe** which will automatically ingest the data to Table.

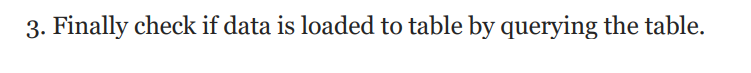
**Now, checking query history:**

select \* from table (information\_schema.

copy\_history(table\_name=>'person\_sourcedata',

start\_time=>dateadd(hours,-1, current\_timestamp())));





A white background with text

Description automatically generated

A black text on a white background

Description automatically generated

A black text on a white background

Description automatically generated

**-- Create a stream object**

create or replace stream customer.public.person\_stream on table customer.public.person\_sourcedata;

A screenshot of a computer

Description automatically generated

A black text on a white background

Description automatically generated

**--Creating a table to Load the unnested data from the table person\_sourcedata.**

create or replace table customer.public.person\_unnested(

id int,

name varchar,

age number,

location varchar,

zip number);

A screenshot of a computer

Description automatically generated

A white background with black text

Description automatically generated

**--Creating a task as per above mentioned specifications:**

CREATE OR REPLACE TASK customer.public.person\_task

WAREHOUSE = COMPUTE\_WH

SCHEDULE = '1 MINUTE'

when system$stream\_has\_data('customer.public.person\_stream')

AS

INSERT INTO customer.public.person\_unnested

select input\_data:id::int, input\_data:name::varchar, input\_data:age::number, input\_data:location::varchar,

input\_data:zip::number

from customer.public.person\_sourcedata;

A screenshot of a computer program

Description automatically generated





select \* from customer.public.person\_unnested;

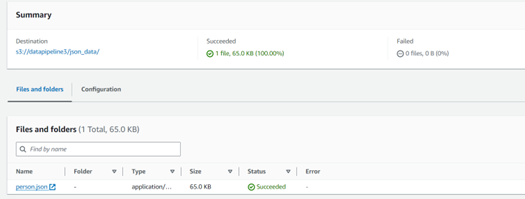
select \* from customer.public.person\_sourcedata;

select \* from customer.public.person\_stream;

A screenshot of a computer

Description automatically generated





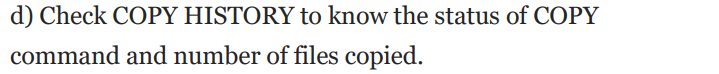
Uploaded person.json file from local folder to S3 bucket folder created.

A close up of a text

Description automatically generated

A screen shot of a computer

Description automatically generated



A screen shot of a computer code

Description automatically generated

A white text with black text

Description automatically generated

select \* from customer.public.person\_stream;

A screenshot of a computer

Description automatically generated

A white background with black text

Description automatically generated

A screenshot of a computer

Description automatically generated

A close up of text

Description automatically generated

A screen shot of a computer

Description automatically generated



A white background with black text

Description automatically generated

CREATE OR REPLACE PROCEDURE unnest\_and\_merge\_person\_data()

RETURNS STRING

LANGUAGE JAVASCRIPT

EXECUTE AS CALLER

AS

$$

try {

// Define the SQL command for the merge operation

var sql\_command = `

MERGE INTO person\_sourcedata AS target

USING (

SELECT

value:id::INTEGER AS id,

value:Name::STRING AS name,

value:age::INTEGER AS age,

value:location::STRING AS location,

value:zip::STRING AS zip

FROM PERSON\_NESTED,

LATERAL FLATTEN(INPUT => person\_unnested) AS flattened

) AS source

ON target.id = source.id

WHEN MATCHED THEN UPDATE SET

target.name = source.name,

target.age = source.age,

target.location = source.location,

target.zip = source.zip

WHEN NOT MATCHED THEN INSERT (id, name, age, location, zip)

VALUES (source.id, source.name, source.age, source.location, source.zip);

`;

// Create a statement object and execute the SQL command

var stmt = snowflake.createStatement({sqlText: sql\_command});

stmt.execute();

// Return success message

return "Data unnest and merge completed successfully.";

} catch (err) {

// Return error message if an exception occurs

return "Failed to unnest and merge data: " + err.message;

}

$$;

A screenshot of a computer

Description automatically generated