CSV dataset load into Snowflake

Neelesh Palaparthi

Login into Snowflake account:

#Login into snowflake account and create a new warehouse and perform the below steps.

Creation of Database:

#Create a new database with a appropriate name.
create or replace database insurance;
use database insurance;

	status
1	Database INSURANCE successfully created.

Creation of Schema:

#Create a schema so that the tables are store into it.
create or replace schema travel;
use schema travel;

	status
1	Schema TRAVEL successfully created.

Grant Permissions for Roles:

#before creating integration layer grant the permissions for different roles.
grant usage on database neelsf to role sysadmin;

Integration Layer:

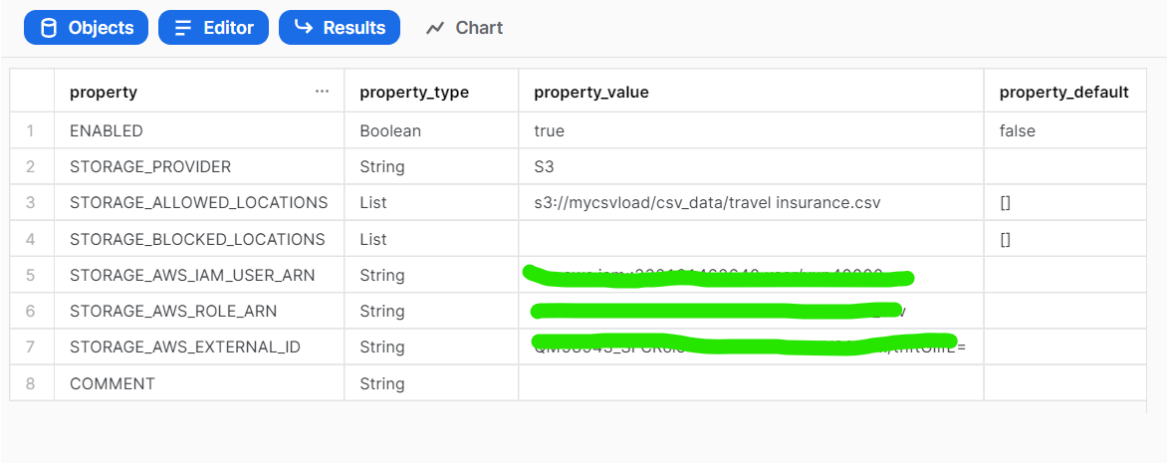
#integrating AWS with snowflake. For that you need to create S3 bucket and IAM role. Refer this link to create S3 bucket and IAM role. Then create a new integration with the following commands.

```
CREATE STORAGE INTEGRATION trin_integ
TYPE = EXTERNAL_STAGE
STORAGE_PROVIDER = S3
ENABLED = TRUE
```

STORAGE_AWS_ROLE_ARN ='arn:aws:iam::62[redacted]_load_csv'
STORAGE_ALLOWED_LOCATIONS = ('s3://[redacted]/travel insurance.csv');

Desribe the integration details:

#To show the integration details.
desc integration trin_integ



	property	property_type	property_value	property_default
1	ENABLED	Boolean	true	false
2	STORAGE_PROVIDER	String	S3	
3	STORAGE_ALLOWED_LOCATIONS	List	s3://mycsvload/csv_data/travel insurance.csv	[]
4	STORAGE_BLOCKED_LOCATIONS	List		[]
5	STORAGE_AWS_IAM_USER_ARN	String	[redacted]	
6	STORAGE_AWS_ROLE_ARN	String	[redacted]	
7	STORAGE_AWS_EXTERNAL_ID	String	[redacted]	
8	COMMENT	String		

File format:

#here we have to give the details of file format like csv, excel, parquet.

```
--file format object
create or replace file format mycsv_format
type = "csv"
field_delimiter = ','
skip_header = 1 -- skip the first row
field_optionally_enclosed_by = '"'
-- for multiple lines of data separated by comma engrossed inside''
empty_field_as_null = True;
```

Table creation:

you have to create table and write all the columns so that when loading data from aws the columns has to match with staging otherwise you will get errors.

```
create or replace table travel_insurance(
index int,
agency varchar,
agency_type varchar,
distribution_channel varchar,
product_name varchar,
```

```
claim varchar,
duration int,
destination varchar,
net_sales float,
commision float,
gender varchar,
age int
);
```

Stage creation:

```
#Creation of stage using integration.
create or replace stage awsf_stage
url = 's3://mycsvload/csv_data/travel insurance.csv'
storage_integration = trin_integ
file_format = mycsv_format;
```

Showing the stage creation:

```
list @awsf_stage;
```

	name	size	md5	last_modified
1	s3://mycsvload/csv_data/travel insurance.csv	4,939,772	f32f60253e0318c74eb3bfa3d96b3794	Fri, 24 Feb 2023 21:45:07 GM

Copying table from stage:

#Copying the data into target table which we have created In the above steps.

```
COPY INTO travel_insurance FROM @awsf_stage
ON_ERROR = 'skip_file';
```

Display the table:

```
select * from travel_insurance;
```

Objects | Editor | Results | Chart

	INDEX	AGENCY	AGENCY_TYPE	DISTRIBUTION_CHANNEL	PRODU
2	2	CBH	Travel Agency	Offline	Compr
3	3	CWT	Travel Agency	Online	Rental
4	4	CWT	Travel Agency	Online	Rental
5	5	CWT	Travel Agency	Online	Rental
6	6	JZI	Airlines	Online	Value F
7	7	CWT	Travel Agency	Online	Rental
8	8	CWT	Travel Agency	Online	Rental
9	9	CWT	Travel Agency	Online	Rental
10	10	CWT	Travel Agency	Online	Rental
11	11	JZI	Airlines	Online	Basic F
12	12	JZI	Airlines	Online	Basic F
13	13	KML	Travel Agency	Online	Premie
14	14	CWT	Travel Agency	Online	Rental
15	15	EPX	Travel Agency	Online	2 way