ASSIGNMENT – 1 : QUESTIONS AND ANSWERS

**1.)** How will you use to change the warehouse for workload processing to a warehouse named ‘COMPUTE\_WH\_XL’?

*Solution*: We can use command:

**USE WAREHOUSE COMPUTE\_WH\_XL;**

To change the warehouse. This will transfer the workload to the said Virtual warehouse.

**2.)**  Consider a table vehicle\_inventory that stores vehicle information of all vehicles in your dealership. The table has only one VARIANT column called vehicle\_data which stores information in JSON format. The data is given below:

{

“date\_of\_arrival”: “2021-04-28”,

“supplier\_name”: “Hillside Honda”,

“contact\_person”: {

“name”: “Derek Larssen”,

“phone”: “8423459854”

},

“vehicle”: [

{

“make”: “Honda”,

“model”: “Civic”,

“variant”: “GLX”,

“year”: “2020”

}

]

}

What is the command to retrieve supplier\_name?

Solution:

**CREATE OR REPLACE TABLE vehicle\_inventory (vehicle\_data VARIANT);**

**INSERT INTO vehicle\_inventory (vehicle\_data)**

**SELECT parse\_json('{**

**""date\_of\_arrival"": ""2021-04-28"",**

**""supplier\_name"": ""Hillside Honda"",**

**""contact\_person"": {**

**""name"": ""Derek Larssen"",**

**""phone"": ""8423459854""**

**},**

**""vehicle"": [**

**{**

**""make"": ""Honda"",**

**""model"": ""Civic"",**

**""variant"": ""GLX"",**

**""year"": ""2020""**

**}**

**]**

**}');**

**SELECT \* from vehicle\_inventory;**

**-- Query to retrieve supplier\_name**

**SELECT**

**vehicle\_data:""supplier\_name""::STRING AS supplier\_name**

**FROM**

**vehicle\_inventory;**

Here I have created a table with single column of variant type and inserted the record over there. We have to use parse\_json function to enter the data. And using the column name and key ""supplier\_name"", we are able to retrive the value.

3.) From a terminal window, how to start SnowSQL from the command prompt ? And write the steps to load the data from local folder into a Snowflake table using three types of internal stages.

Step 1: Install SnowSQL CLI and open command prompt. Pass the below comment and enter password

**snowsql -a qjhtlxv-saurab -u mssaurabhp**

Step 2: Use PUT command in snowsql to upload files and use copy command to load the data from file to table.

Name Stage:

PUT file:///c:\users\saurabhp\Desktop\family.csv @MYSTAGE;

COPY INTO MYTABLE from @MYSTAGE FILE\_FORMAT = (FORMAT\_NAME = 'my\_csv\_format');

User Stage:

PUT file:///c:\users\saurabhp\Desktop\family.csv @ ~/STAGED;

COPY INTO MYTABLE from @~/STAGED FILE\_FORMAT = (FORMAT\_NAME = 'my\_csv\_format');

TABLE Stage:

PUT file:///c:\users\saurabhp\Desktop\family.csv @%MYTABLE;

COPY INTO mytable FILE\_FORMAT = (TYPE = CSV FIELD\_DELIMITER = '|' SKIP\_HEADER = 1);

4.) "Create an X-Small warehouse named xf\_tuts\_wh using the CREATE WAREHOUSE command with below options

a) Size with x-small

b) which can be automatically suspended after 10 mins

c) setup how to automatically resume the warehouse

d) Warehouse should be suspended once after created"

Solution:

CREATE OR REPLACE WAREHOUSE xf\_tuts\_wh

WAREHOUSE\_SIZE = 'X-Small'

AUTO\_SUSPEND = 600

AUTO\_RESUME = TRUE

INITIALLY\_SUSPENDED=true;

5.) "A CSV file ‘customer.csv’ consists of 1 or more records, with 1 or more fields in each record, and sometimes a header record. Records and fields in each file are separated by delimiters. How will Load the file into snowflake table ?"

-- Step 1: Create the customer table

CREATE OR REPLACE TABLE customer (

id INT,

name STRING,

age INT,

email STRING

-- Add other columns as necessary

);

-- Step 2: Create the file format

CREATE OR REPLACE FILE FORMAT my\_csv\_format

TYPE = 'CSV'

FIELD\_OPTIONALLY\_ENCLOSED\_BY = '"'

SKIP\_HEADER = 1

FIELD\_DELIMITER = ','

EMPTY\_FIELD\_AS\_NULL = TRUE

NULL\_IF = ('NULL', 'null', '');

-- Step 3: Upload the CSV file to the internal or external stage

-- Step 4: Copy the data from the stage into the table

COPY INTO customer

FROM @~/customer.csv

FILE\_FORMAT = (FORMAT\_NAME = 'my\_csv\_format');

6.) Write the commands to disable < auto-suspend > option for a virtual warehouse

ALTER WAREHOUSE MY\_WAREHOUSE SET AUTO\_SUSPEND = NULL;

7.) What is the command to concat the column named 'EMPLOYEE' between two % signs ?

Solution:

SELECT CONCAT('%', EMPLOYEE, '%') AS EMPLOYEE\_WITH\_PERCENT

FROM EMPLOYEE\_TABLE;

8.) "You have stored the below JSON in a table named car\_sales as a variant column

{

""customer"": [

{

""address"": ""San Francisco, CA"",

""name"": ""Joyce Ridgely"",

""phone"": ""16504378889""

}

],

""date"": ""2017-04-28"",

""dealership"": ""Valley View Auto Sales"",

""salesperson"": {

""id"": ""55"",

""name"": ""Frank Beasley""

},

""vehicle"": [

{

""extras"": [

""ext warranty"",

""paint protection""

],

""make"": ""Honda"",

""model"": ""Civic"",

""price"": ""20275"",

""year"": ""2017""

}

]

}

How will you query the table to get the dealership data?"

CREATE OR REPLACE TABLE vehicle\_inventory (vehicle\_data VARIANT);

INSERT INTO vehicle\_inventory (vehicle\_data)

SELECT parse\_json('{

"customer": [

{

"address": "San Francisco, CA",

"name": "Joyce Ridgely",

"phone": "16504378889"

}

],

"date": "2017-04-28",

"dealership": "Valley View Auto Sales",

"salesperson": {

"id": "55",

"name": "Frank Beasley"

},

"vehicle": [

{

"extras": [

"ext warranty",

"paint protection"

],

"make": "Honda",

"model": "Civic",

"price": "20275",

"year": "2017"

}

]

}');

SELECT \* from vehicle\_inventory;

-- Query to retrieve supplier\_name

SELECT

vehicle\_data:"dealership"::STRING AS dealership

FROM

vehicle\_inventory;

9.) A medium size warehouse runs in Auto-scale mode for 3 hours with a resize from Medium (4 servers per cluster) to Large (8 servers per cluster). Warehouse is resized from Medium to Large at 1:30 hours, Cluster 1 runs continuously, Cluster 2 runs continuously for the 2nd and 3rd hours, Cluster 3 runs for 15 minutes in the 3rd hour. How many total credits will be consumed

Solution:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Cluster 1 | Cluster 2 | Cluster 3 | Total |
| 1st hour | 4 | 0 | 0 | 4 |
| 2nd hour | 4+2 | 4+2 | 0 | 12 |
| 3rd hour | 8 | 8 | 2 | 18 |
| Total Credits | 18 | 14 | 2 | **34** |

Total 34 Credits

10.) What is the command to check status of snowpipe?

SELECT SYSTEM$PIPE\_STATUS('mydb.myschema.mypipe');

11.) What are the different methods of getting/accessing/querying data from Time travel , Assume the table name is 'CUSTOMER' and please write the command for each method.

--SELECT \* FROM CUSTOMER AT (TIMESTAMP => '2024-07-16 10:00:00');

--SELECT \* FROM CUSTOMER AT (OFFSET => -60\*60); -- 1 hour ago

Using Query\_ID

SELECT \* FROM my\_CUSTOMER BEFORE(STATEMENT => '8e5d0ca9-005e-44e6-b858-a8f5b37c5726');

12.) If comma is defined as column delimiter in file "employee.csv" and if we get extra comma in the data how to handle this scenario?

We can use escape character in this scenario.

CREATE OR REPLACE FILE FORMAT my\_escape\_format

TYPE = 'CSV'

FIELD\_DELIMITER = ','

ESCAPE = '\\'

SKIP\_HEADER = 1;

OR we can use pipe as delimeter

CREATE OR REPLACE FILE FORMAT my\_pipe\_format

TYPE = 'CSV'

FIELD\_DELIMITER = '|'

SKIP\_HEADER = 1;

13.) What is the command to read data directly from S3 bucket/External/Internal Stage

* Define File Format.

-- Create a file format

CREATE OR REPLACE FILE FORMAT my\_json\_format TYPE = 'json';

* Create a Stage ( internal or external with file format)

-- Create an internal stage

CREATE OR REPLACE STAGE mystage2 FILE\_FORMAT = my\_json\_format;

* Upload file

-- Stage the data file

PUT file:///tmp/data1.json @mystage2;

* Query the data file using appropriate function

-- Query the repeating a.b element in the staged file

SELECT parse\_json($1):a.b FROM @mystage2/data1.json.gz;

* After this we can load the data in tables using COPY INTO command.

14.) Lets assume we have table with name 'products' which contains duplicate rows. How will delete the duplicate rows ?

Solution: We can using Common Table Expression(CTE) and apply ROW NUMBER window function and pass combination of column names which we want to be unique to PARTITION BY parameter. This will create unique row number for each row. Then we can keep the first row and delete rest.

WITH CTE AS (

SELECT \*,

ROW\_NUMBER() OVER (PARTITION BY column1, column2, column3 ORDER BY id) AS row\_num

FROM MYTABLE

)

DELETE FROM MYTABLE

USING CTE

WHERE your\_table.id = CTE.id

AND CTE.row\_num > 1;

15.) How is data unloaded out of Snowflake?

**CREATE** STAGE

DEFINE **FILE FORMAT**

**COPY INTO** STAGE FROM TABLE

Use **GET** command to push file from snowflake stage to other cloud or local environment

SELECT \* FROM emp\_basic\_local;

--copy into command to unload the data from table to internal stage:

copy into @%emp\_basic\_local

from emp\_basic\_local

file\_format = (type = csv field\_optionally\_enclosed\_by='"')

--on\_error = 'skip\_file';

--get command to download the data from internal stage to local file location:

get @emp\_stage file://C:\temp\Employee\unload

2. Steps to download the data from Snowflake table to external S3 bucket using external stage:

create or replace file format unload\_csv

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

alter storage integration s3\_int set storage\_allowed\_locations=('s3://snowflake069/employee/','s3://snowflake069/emp\_unload/','s3://snowflake069/zip\_folder/')

--describe the integration:

desc integration s3\_int

--create the externak stage:

create or replace stage my\_s3\_unload\_stage

storage\_integration = s3\_int

url = 's3://snowflake069/emp\_unload/'

file\_format = unload\_csv;

--copy into command to copy the data from table to external stage:

copy into @my\_s3\_unload\_stage

from

emp\_ext;

--If we need to change the file format as parquet, we can use the following command)

copy into @my\_s3\_unload\_stage/parquet\_

from

emp\_ext\_stage

FILE\_FORMAT=(TYPE='PARQUET' SNAPPY\_COMPRESSION=TRUE)