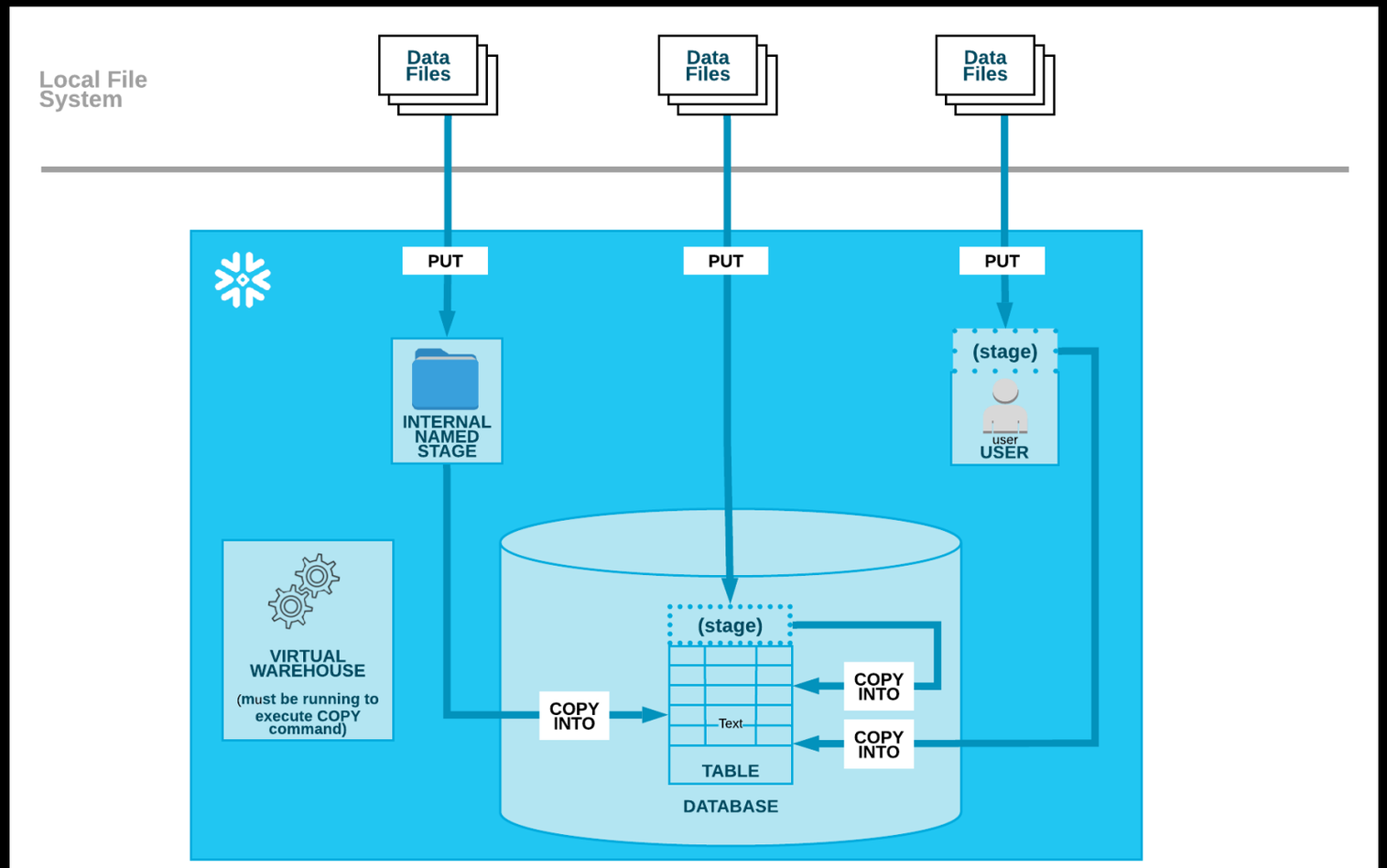
Snowflake Project

TR Raveendra

Snowflake Ingestion?

LinkedIn
@TRRaveendra

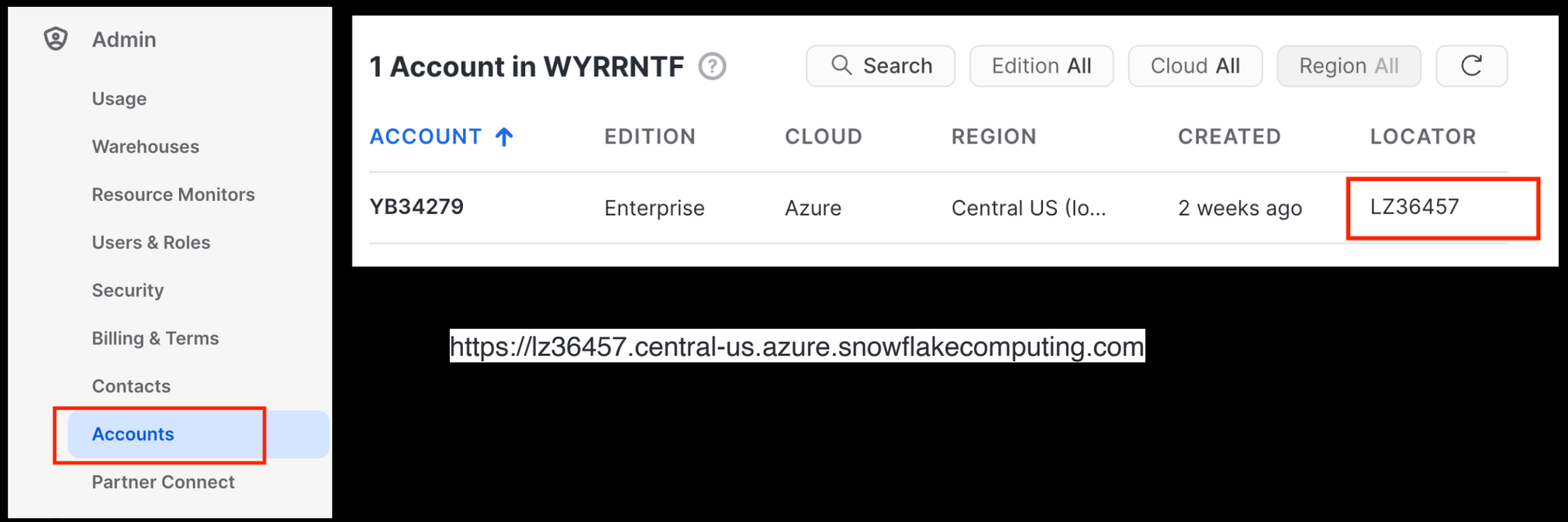
Step 1. Log into SnowSQL
Step 2. Create Snowflake Objects
Step 3. Stage the Data Files
Step 4. Copy Data into the Target Table
Step 5. Query the Loaded Data



Get Account Identifier to login Snowsql

snowsql -a <account_identifier> -u <user_name>

LinkedIn
@TRRaveendra



https://lz36457.central-us.azure.snowflakecomputing.com

Enter password in next input

```
C:\WINDOWS\system32>snowsql -a lz36457.central-us.azure -u pysparktraining11
Password:
* SnowSQL * v1.2.24
Type SQL statements or !help
pysparktraining11#(no warehouse)@(no database).(no schema)>
```

Creating a Database

Create the employee_db database using the CREATE DATABASE command:

Compute Warehouse is not required for any metadata activity.

```
Type SQL statements or !help
pysparktraining11#(no warehouse)@(no database).(no schema)>create database employee_db;
+-----------------------------------------+
| status                                  |
|-----------------------------------------|
| Database EMPLOYEE_DB successfully created. |
+-----------------------------------------+
1 Row(s) produced. Time Elapsed: 1.484s
pysparktraining11#(no warehouse)@EMPLOYEE_DB.PUBLIC>create schema employee_schema;
+-----------------------------------------+
| status                                  |
|-----------------------------------------|
| Schema EMPLOYEE_SCHEMA successfully created. |
+-----------------------------------------+
1 Row(s) produced. Time Elapsed: 0.359s
pysparktraining11#(no warehouse)@EMPLOYEE_DB.EMPLOYEE_SCHEMA>
```

Use WAREHOUSE compute_wh for any data load activities.

```
1 Row(s) produced. Time Elapsed: 0.359s
pysparktraining11#(no warehouse)@EMPLOYEE_DB.EMPLOYEE_SCHEMA>use WAREHOUSE COMPUTE_WH;
+----------------------------------+
| status                           |
|----------------------------------|
| Statement executed successfully. |
+----------------------------------+
1 Row(s) produced. Time Elapsed: 0.312s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>
```

Create below emp_basic table using DDL script https://github.com/raveendratal/snowflake/blob/main/emp_basic_script.sql

```
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>CREATE OR REPLACE TABLE emp_basic (
  first_name STRING ,
  last_name STRING ,
  email STRING ,
  streetaddress STRING ,
  city STRING ,
  start_date DATE
  );

+-------------------------------------+
| status                              |
|-------------------------------------|
| Table EMP_BASIC successfully created. |
+-------------------------------------+
1 Row(s) produced. Time Elapsed: 0.406s
```

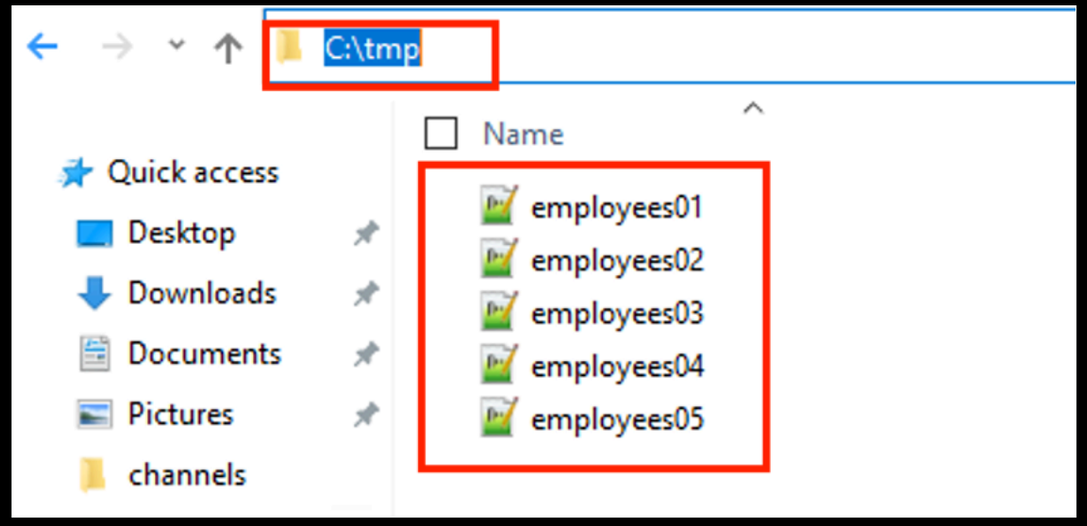
Verify current database, current schema, current schema using below query.

```
1 Row(s) produced. Time Elapsed: 0.641s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>select current_database(),current_schema(),current_warehouse();
+--------------------+--------------------+---------------------+
| CURRENT_DATABASE() | CURRENT_SCHEMA()   | CURRENT_WAREHOUSE() |
|--------------------+--------------------+---------------------|
| EMPLOYEE_DB        | EMPLOYEE_SCHEMA    | COMPUTE_WH          |
+--------------------+--------------------+---------------------+
1 Row(s) produced. Time Elapsed: 0.360s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>
```

Download sample datafile from below link and keep in your local system.

https://docs.snowflake.com/en/_downloads/34f4a66f56d00340f8f7a92acaccd977/getting-started.zip

I have extracted above downloaded file and files looks like as below.



Staging the Sample Data Files

Execute **PUT** to upload local data files to the **table stage** provided for the **emp_basic** table you created.

- Linux or macOS

```
PUT file:///tmp/employees0*.csv @sf_tuts.public.%emp_basic;
```

- Windows

```
PUT file://C:\temp\employees0*.csv @sf_tuts.public.%emp_basic;
```

PUT file://C:\tmp\employees*.csv @%EMP_BASIC;

```
0 Row(s) produced. Time Elapsed: 0.422s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>PUT file://C:\tmp\employees*.csv @%EMP_BASIC;
+---------------+--------------------+-------------+-------------+--------------------+--------------------+----------+---------+
| source        | target             | source_size | target_size | source_compression | target_compression | status   | message |
|---------------+--------------------+-------------+-------------+--------------------+--------------------+----------+---------|
| employees01.csv | employees01.csv.gz |         370 |         304 | NONE               | GZIP               | UPLOADED |         |
| employees02.csv | employees02.csv.gz |         364 |         288 | NONE               | GZIP               | UPLOADED |         |
| employees03.csv | employees03.csv.gz |         407 |         304 | NONE               | GZIP               | UPLOADED |         |
| employees04.csv | employees04.csv.gz |         375 |         304 | NONE               | GZIP               | UPLOADED |         |
| employees05.csv | employees05.csv.gz |         404 |         304 | NONE               | GZIP               | UPLOADED |         |
+---------------+--------------------+-------------+-------------+--------------------+--------------------+----------+---------+
5 Row(s) produced. Time Elapsed: 4.867s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>
```

Listing the Staged Files — just make sure files are loaded in stage.
You can list the staged files using the LIST command.

```
5 Row(s) produced. Time Elapsed: 4.867s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>LIST @%emp_basic;
+--------------------+-----+----------------------------------+-------------------------------+
| name               | size | md5                              | last_modified                 |
|--------------------+-----+----------------------------------+-------------------------------|
| employees01.csv.gz |  304 | 8bd8be18e33525c39155b522b5e18f6a | Thu, 16 Feb 2023 16:42:46 GMT |
| employees02.csv.gz |  288 | db14c44d15a7d96d66ab66bb1f0deafe | Thu, 16 Feb 2023 16:42:45 GMT |
| employees03.csv.gz |  304 | bf80ac50c437825c071f4a88e2f4e675 | Thu, 16 Feb 2023 16:42:45 GMT |
| employees04.csv.gz |  304 | a1f1ac59f7f4b2a64fa58e7da01b2a48 | Thu, 16 Feb 2023 16:42:45 GMT |
| employees05.csv.gz |  304 | 1be0ee3f933d0a9312708862a055b574 | Thu, 16 Feb 2023 16:42:46 GMT |
+--------------------+-----+----------------------------------+-------------------------------+
5 Row(s) produced. Time Elapsed: 1.688s
```

Copy Data into the Target Table From Table Stage

@TRRaveendra

```
COPY INTO emp_basic
  FROM @%emp_basic
  FILE_FORMAT = (type = csv field_optionally_enclosed_by='"')
  PATTERN = '.*employees0[1-5].csv.gz'
  ON_ERROR = 'skip_file';
```

```
5 Row(s) produced. Time Elapsed: 1.688s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>COPY INTO emp_basic
  FROM @%emp_basic
  FILE_FORMAT = (type = csv field_optionally_enclosed_by='"')
  PATTERN = '.*employees0[1-5].csv.gz'
  ON_ERROR = 'skip_file';
+--------------------+--------+-------------+-------------+-------------+-------------+-------------+-----------------+-----------------------+--------------------------+
| file               | status | rows_parsed | rows_loaded | error_limit | errors_seen | first_error | first_error_line | first_error_character | first_error_column_name |
|--------------------+--------+-------------+-------------+-------------+-------------+-------------+-----------------+-----------------------+--------------------------|
| employees02.csv.gz | LOADED |           5 |           5 |           1 |           0 | NULL        |            NULL |                  NULL | NULL                     |
| employees05.csv.gz | LOADED |           5 |           5 |           1 |           0 | NULL        |            NULL |                  NULL | NULL                     |
| employees04.csv.gz | LOADED |           5 |           5 |           1 |           0 | NULL        |            NULL |                  NULL | NULL                     |
| employees03.csv.gz | LOADED |           5 |           5 |           1 |           0 | NULL        |            NULL |                  NULL | NULL                     |
| employees01.csv.gz | LOADED |           5 |           5 |           1 |           0 | NULL        |            NULL |                  NULL | NULL                     |
+--------------------+--------+-------------+-------------+-------------+-------------+-------------+-----------------+-----------------------+--------------------------+
5 Row(s) produced. Time Elapsed: 2.953s
```

Query the Loaded Data

You can query the data loaded in the **emp_basic** table using standard SQL and any supported functions and operators .

You can also manipulate the data, such as updating the loaded data or inserting more data, using standard DML commands.

```
5 Row(s) produced. Time Elapsed: 2.953s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>select * from EMP_BASIC;
+------------+-------------+---------------------------+-----------------------------+--------------------+------------+
| FIRST_NAME | LAST_NAME   | EMAIL                     | STREETADDRESS               | CITY               | START_DATE |
|------------+-------------+---------------------------+-----------------------------+--------------------+------------|
| Nyssa      | Dorgan      | ndorgan5@sf_tuts.com      | 7 Tomscot Way               | Pampas Chico       | 2017-04-13 |
| Catherin   | Devereu     | cdevereu6@sf_tuts.co.au   | 535 Basil Terrace           | Magapit            | 2016-12-17 |
| Grazia     | Glaserman   | gglaserman7@sf_tuts.com   | 162 Debra Lane              | Shiquanhe          | 2017-06-06 |
| Ivett      | Casemore    | icasemore8@sf_tuts.com    | 84 Holmberg Pass            | Campina Grande     | 2017-03-29 |
| Cesar      | Hovie       | chovie9@sf_tuts.com       | 5 7th Pass                  | Miami              | 2016-12-21 |
| Arlene     | Davidovits  | adavidovitsk@sf_tuts.com  | 7571 New Castle Circle      | Meniko             | 2017-05-03 |
| Violette   | Shermore    | vshermorel@sf_tuts.com    | 899 Merchant Center         | Troitsk            | 2017-01-19 |
| Ron        | Mattys      | rmattysm@sf_tuts.com      | 423 Lien Pass               | Bayaguana          | 2017-11-15 |
| Shurlocke  | Oluwatoyin  | soluwatoyinn@sf_tuts.com  | 40637 Portage Avenue        | Semënovskoye       | 2017-09-12 |
| Granger    | Bassford    | gbassfordo@sf_tuts.co.uk  | 6 American Ash Circle       | Kardítsa           | 2016-12-30 |
| Wallis     | Sizey       | wsizeyf@sf_tuts.com       | 36761 American Lane         | Taibao             | 2016-12-30 |
| Di         | McGowran    | dmcgowrang@sf_tuts.com    | 1856 Maple Lane             | Banjar Bengkelgede | 2017-04-22 |
| Carson     | Bedder      | cbedderh@sf_tuts.co.au    | 71 Clyde Gallagher Place    | Leninskoye         | 2017-03-29 |
| Dana       | Avory       | davoryi@sf_tuts.com       | 2 Holy Cross Pass           | Wenlin             | 2017-05-11 |
| Ronny      | Talmadge    | rtalmadgej@sf_tuts.co.uk  | 588 Chinook Street          | Yawata             | 2017-06-02 |
| Althea     | Featherstone| afeatherstona@sf_tuts.com | 8172 Browning Street, Apt B | Calatrava          | 2017-07-12 |
| Hollis     | Anneslie    | hanneslieb@sf_tuts.com    | 3248 Roth Park              | Aleysk             | 2017-11-16 |
| Betti      | Cicco       | bciccoc@sf_tuts.com       | 121 Victoria Junction       | Sinegor'ye         | 2017-06-22 |
| Brendon    | Durnall     | bdurnalld@sf_tuts.com     | 26814 Weeping Birch Place   | Sabadell           | 2017-11-14 |
| Kylila     | MacConnal   | kmacconnale@sf_tuts.com   | 04 Valley Edge Court        | Qingshu            | 2017-06-22 |
| Lem        | Boissier    | lboissier@sf_tuts.com     | 3002 Ruskin Trail           | Shikārpur          | 2017-08-25 |
| Iain       | Hanks       | ihanks1@sf_tuts.com       | 2 Pankratz Hill             | Monte-Carlo        | 2017-12-10 |
| Avo        | Laudham     | alaudham2@sf_tuts.com     | 6948 Debs Park              | Prażmów            | 2017-10-18 |
| Emili      | Cornner     | ecornner3@sf_tuts.com     | 177 Magdeline Avenue        | Norrköping         | 2017-08-13 |
| Harrietta  | Goolding    | hgoolding4@sf_tuts.com    | 450 Heath Trail             | Osielsko           | 2017-11-27 |
+------------+-------------+---------------------------+-----------------------------+--------------------+------------+
25 Row(s) produced. Time Elapsed: 2.031s
```

Insert data into emp_basic table using manual insert script

```
INSERT INTO emp_basic VALUES
   ('Clementine','Adamou','cadamou@sf_tuts.com','10510 Sachs Road','Klenak','2017-9-22') ,
   ('Marlowe','De Anesy','madamouc@sf_tuts.co.uk','36768 Northfield Plaza','Fangshan','2017-1-26');
```

```
25 Row(s) produced. Time Elapsed: 2.031s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>INSERT INTO emp_basic VALUES
                                                           ('Clementine','Adamou','cadamou@sf_tuts.com','10510 Sachs Road','Klenak','2017-9-22') ,
                                                           ('Marlowe','De Anesy','madamouc@sf_tuts.co.uk','36768 Northfield Plaza','Fangshan','2017-1-26');

+-------------------------+
| number of rows inserted |
|-------------------------|
|                       2 |
+-------------------------+
2 Row(s) produced. Time Elapsed: 0.750s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>
```

Query Rows Based on Start Date

Add 90 days to employee start dates using the DATEADD function to calculate when certain employee benefits might start. Filter the list by employees whose start date occurred earlier than January 1, 2017:

```
SELECT first_name, last_name, DATEADD('day',90,start_date) FROM emp_basic WHERE start_date <= '2017-01-01';
```

```
2 Row(s) produced. Time Elapsed: 0.750s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>SELECT first_name, last_name, DATEADD('day',90,start_date) FROM emp_basic WHERE start_date <= '2017-01-01';
+------------+-----------+-------------------------------+
| FIRST_NAME | LAST_NAME | DATEADD('DAY',90,START_DATE)  |
|------------+-----------+-------------------------------|
| Catherin   | Devereu   | 2017-03-17                    |
| Cesar      | Hovie     | 2017-03-21                    |
| Granger    | Bassford  | 2017-03-30                    |
| Wallis     | Sizey     | 2017-03-30                    |
+------------+-----------+-------------------------------+
4 Row(s) produced. Time Elapsed: 2.156s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>
```

Snowflake Ingestion?

LinkedIn
@TRRaveendra

```
CREATE OR REPLACE DATABASE mydatabase;

/* Create target tables for CSV and JSON data. The tables are temporary, meaning they persist only for the duration of the user session and are not
visible to other users. */

CREATE OR REPLACE TEMPORARY TABLE mycsvtable (
  id INTEGER,
  last_name STRING,
  first_name STRING,
  company STRING,
  email STRING,
  workphone STRING,
  cellphone STRING,
  streetaddress STRING,
  city STRING,
  postalcode STRING);

CREATE OR REPLACE TEMPORARY TABLE myjsontable (
  json_data VARIANT);
```

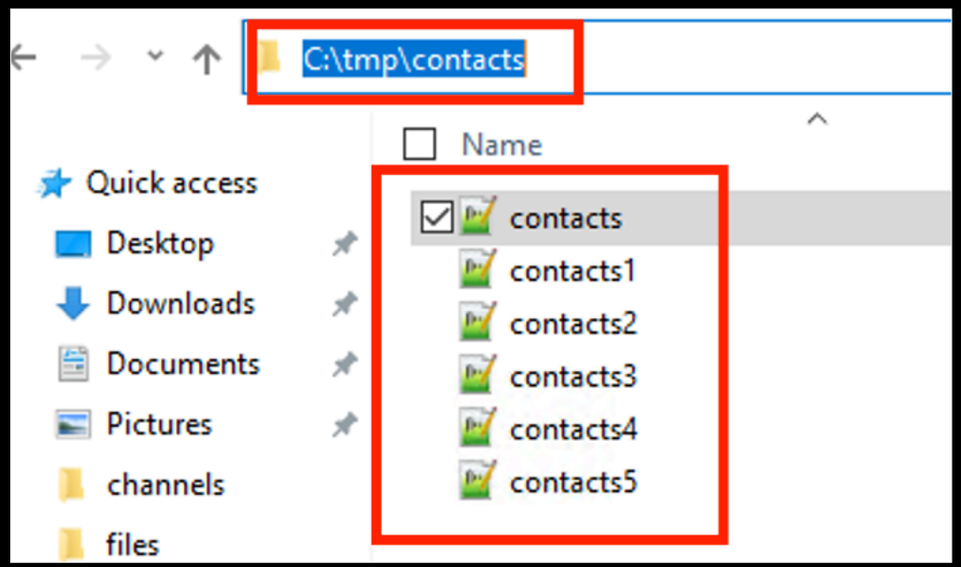
```
-- Create a warehouse

CREATE OR REPLACE WAREHOUSE mywarehouse WITH
  WAREHOUSE_SIZE='X-SMALL'
  AUTO_SUSPEND = 120
  AUTO_RESUME = TRUE
  INITIALLY_SUSPENDED=TRUE;
```

Contacts Data Files

Download below csv files and extract and create new folder in local system

https://docs.snowflake.com/en/_downloads/22c3a6290f5d1f4d97075282729f3859/data-load-internal.zip



Stage the Data Files

Execute PUT to upload (stage) sample data files from your local file system to the stages you created in

```
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>create stage my_contacts_stg;
+-------------------------------------------------+
| status                                          |
|-------------------------------------------------|
| Stage area MY_CONTACTS_STG successfully created. |
+-------------------------------------------------+
1 Row(s) produced. Time Elapsed: 1.585s
```

Put files into Named Stage

LinkedIn
@TRRaveendra

Staging the CSV Sample Data Files
Execute the PUT command to upload the CSV files from your local file system to the named stage.

PUT file://C:\tmp\contacts\contacts*.csv @my_contacts_stg AUTO_COMPRESS=TRUE;

```
1 Row(s) produced. Time Elapsed: 1.585s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>PUT file://C:\tmp\contacts\contacts*.csv @my_contacts_stg AUTO_COMPRESS=TRUE;
+---------------+------------------+-------------+-------------+--------------------+--------------------+----------+---------+
| source        | target           | source_size | target_size | source_compression | target_compression | status   | message |
|---------------+------------------+-------------+-------------+--------------------+--------------------+----------+---------|
| contacts1.csv | contacts1.csv.gz |         694 |         512 | NONE               | GZIP               | UPLOADED |         |
| contacts2.csv | contacts2.csv.gz |         763 |         576 | NONE               | GZIP               | UPLOADED |         |
| contacts3.csv | contacts3.csv.gz |         771 |         576 | NONE               | GZIP               | UPLOADED |         |
| contacts4.csv | contacts4.csv.gz |         750 |         576 | NONE               | GZIP               | UPLOADED |         |
| contacts5.csv | contacts5.csv.gz |         887 |         624 | NONE               | GZIP               | UPLOADED |         |
+---------------+------------------+-------------+-------------+--------------------+--------------------+----------+---------+
5 Row(s) produced. Time Elapsed: 6.454s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>
```

create stage my_contacts_json_stage;

```
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>create stage my_contacts_json_stage;
+-----------------------------------------------------------+
| status                                                    |
|-----------------------------------------------------------|
| Stage area MY_CONTACTS_JSON_STAGE successfully created.   |
+-----------------------------------------------------------+
1 Row(s) produced. Time Elapsed: 0.469s
```

Staging the JSON Sample Data Files
Execute the PUT command to upload the JSON file from your local file system to the named stage.

PUT file://C:\tmp\contacts\contacts*.json @my_contacts_json_stage AUTO_COMPRESS=TRUE;

```
 Row(s) produced. Time Elapsed: 0.469s
ysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>PUT file://C:\tmp\contacts\contacts*.json @my_contacts_json_stage AUTO_COMPRESS=TRUE;
---------------+------------------+-------------+-------------+--------------------+--------------------+----------+---------+
 source        | target           | source_size | target_size | source_compression | target_compression | status   | message |
---------------+------------------+-------------+-------------+--------------------+--------------------+----------+---------|
 contacts.json | contacts.json.gz |         965 |         448 | NONE               | GZIP               | UPLOADED |         |
---------------+------------------+-------------+-------------+--------------------+--------------------+----------+---------+
 Row(s) produced. Time Elapsed: 2.890s
ysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>
```

Contacts Data Files

LinkedIn
@TRRaveendra

Creating a File Format Object for CSV Data
Execute the CREATE FILE FORMAT command to create the mycsvformat file format.

```
CREATE OR REPLACE FILE FORMAT mycsvformat
  TYPE = 'CSV'
  FIELD_DELIMITER = '|'
  SKIP_HEADER = 1;
```

Creating a File Format Object for JSON Data
Execute the CREATE FILE FORMAT command to create the myjsonformat file format.

```
CREATE OR REPLACE FILE FORMAT myjsonformat
  TYPE = 'JSON'
  STRIP_OUTER_ARRAY = TRUE;
```

```
CREATE OR REPLACE FILE FORMAT mycsvformat
  TYPE = 'CSV'
  FIELD_DELIMITER = '|'
  SKIP_HEADER = 1;
```

```
CREATE OR REPLACE FILE FORMAT myjsonformat
  TYPE = 'JSON'
  STRIP_OUTER_ARRAY = TRUE;
```

Contacts Data Files

LinkedIn
@TRRaveendra

Start by loading the data from one of the files (contacts1.csv.gz). Execute the following:

```
COPY INTO mycsvtable  FROM @my_contacts_stg/contacts1.csv.gz
FILE_FORMAT = (FORMAT_NAME = mycsvformat)
ON_ERROR = 'skip_file';
```

```
1 Row(s) produced. Time Elapsed: 0.509s
pysparktraining11#MYWAREHOUSE@EMPLOYEE_DB.EMPLOYEE_SCHEMA>COPY INTO mycsvtable
    FROM @my_contacts_stg/contacts1.csv.gz
    FILE_FORMAT = (FORMAT_NAME = mycsvformat)
    ON_ERROR = 'skip_file';

+---------------------------------+--------+-------------+-------------+-------------+-------------+-------------+
| file                            | status | rows_parsed | rows_loaded | error_limit | errors_seen | first_error |
|---------------------------------+--------+-------------+-------------+-------------+-------------+-------------+
| my_contacts_stg/contacts1.csv.gz | LOADED |           5 |           5 |           1 |           0 | NULL        |
+---------------------------------+--------+-------------+-------------+-------------+-------------+-------------+
1 Row(s) produced. Time Elapsed: 2.547s
pysparktraining11#MYWAREHOUSE@EMPLOYEE_DB.EMPLOYEE_SCHEMA>
```

```
pysparktraining11#MYWAREHOUSE@EMPLOYEE_DB.EMPLOYEE_SCHEMA>COPY INTO mycsvtable
    FROM @my_contacts_stg
    FILE_FORMAT = (FORMAT_NAME = mycsvformat)
    PATTERN='.*contacts[1-5].csv.gz'
    ON_ERROR = 'skip_file';

+---------------------------------+-------------+-------------+-------------+-------------+-------------+
| file                            | status      | rows_parsed | rows_loaded | error_limit | errors_seen |
|                                 | first_error_line | first_error_character | first_error_column_name |
|---------------------------------+-------------+-------------+-------------+-------------+-------------+
| my_contacts_stg/contacts4.csv.gz | LOADED      |           5 |           5 |           1 |           0 |
|                                 |        NULL |                NULL | NULL                    |
| my_contacts_stg/contacts5.csv.gz | LOADED      |           6 |           6 |           1 |           0 |
|                                 |        NULL |                NULL | NULL                    |
| my_contacts_stg/contacts2.csv.gz | LOADED      |           5 |           5 |           1 |           0 |
|                                 |        NULL |                NULL | NULL                    |
| my_contacts_stg/contacts3.csv.gz | LOAD_FAILED |           5 |           0 |           1 |           2 |
ismatch=false to ignore this error |           3 |                 1 | "MYCSVTABLE"[11]        |
+---------------------------------+-------------+-------------+-------------+-------------+-------------+
4 Row(s) produced. Time Elapsed: 0.922s
```

Contacts Data Files

LinkedIn
@TRRaveendra

JSON
Load the contacts.json.gz staged data file into the myjsontable table.

```
COPY INTO myjsontable FROM @my_contacts_json_stage/contacts.json.gz
 FILE_FORMAT = (FORMAT_NAME = myjsonformat)
ON_ERROR = 'skip_file';
```

```
pysparktraining11#MYWAREHOUSE@EMPLOYEE_DB.EMPLOYEE_SCHEMA>COPY INTO myjsontable
                                                          FROM @my_contacts_json_stage/contacts.json.gz
                                                          FILE_FORMAT = (FORMAT_NAME = myjsonformat)
                                                          ON_ERROR = 'skip_file';
+---------------------------------------+--------+-------------+-------------+-------------+-------------+-------------+------------------+------------------------+--------------------------+
| file                                  | status | rows_parsed | rows_loaded | error_limit | errors_seen | first_error | first_error_line | first_error_character | first_error_column_name |
|---------------------------------------+--------+-------------+-------------+-------------+-------------+-------------+------------------+------------------------+--------------------------|
| my_contacts_json_stage/contacts.json.gz | LOADED |           3 |           3 |           1 |           0 | NULL        |             NULL |                   NULL | NULL                     |
+---------------------------------------+--------+-------------+-------------+-------------+-------------+-------------+------------------+------------------------+--------------------------+
1 Row(s) produced. Time Elapsed: 1.068s
```

```
pysparktraining11#MYWAREHOUSE@EMPLOYEE_DB.EMPLOYEE_SCHEMA>
```

Contacts Error Data

Validate the Sample Data Files and Retrieve Any Errors

You first need the query ID associated with the COPY INTO command that you previously executed. You then call the VALIDATE function, specifying the query ID.

Query ID

01aa5fda-0000-3c70-0000-73c50004f196

CREATE OR REPLACE TABLE save_copy_errors AS SELECT * FROM TABLE(VALIDATE(mycsvtable, JOB_ID=>'<query_id>'));

```
 Row(s) produced. Time Elapsed: 1.068s
ysparktraining11#MYWAREHOUSE@EMPLOYEE_DB.EMPLOYEE_SCHEMA>CREATE OR REPLACE TABLE save_copy_errors AS SELECT * FROM TABLE(VALIDATE(mycsvtable, JOB_ID=>'01aa5fda-0000-3c70-0000-73c50004f196'));
-------------------------------------------+
 status                                    |
-------------------------------------------|
 Table SAVE_COPY_ERRORS successfully created. |
-------------------------------------------+
 Row(s) produced. Time Elapsed: 2.905s
ysparktraining11#MYWAREHOUSE@EMPLOYEE_DB.EMPLOYEE_SCHEMA>select * from SAVE_COPY_ERRORS;
 ERROR                                                                                                                              | FILE              | LINE | CHARACTER
 | SQL_STATE | COLUMN_NAME                    | ROW_NUMBER | ROW_START_LINE | REJECTED_RECORD
 Number of columns in file (11) does not match that of the corresponding table (10), use file format option error_on_column_count_mismatch=false to ignore this error | contacts3.csv.gz |    3 |         1
 | 22000     | "MYCSVTABLE"[11]               |          1 |              2 | 11|Ishmael|Burnett|Dolor Elit Pellentesque Ltd|vitae.erat@necmollisvitae.ca|1-872|600-7301|1-513-592-6779|P.O. Box 975, 553 Odi
 Field delimiter '|' found while expecting record delimiter '\n'                                                                    | contacts3.csv.gz |    5 |       125
 | 22000     | "MYCSVTABLE"["POSTALCODE":10] |          4 |              5 | 14|Sophia|Christian|Turpis Ltd|lectus.pede@non.ca|1-962-503-3253|1-157-|850-3602|P.O. Box 824, 7971 Sagittis Rd.|Chattanooga|56
 Row(s) produced. Time Elapsed: 0.390s
ysparktraining11#MYWAREHOUSE@EMPLOYEE_DB.EMPLOYEE_SCHEMA>
```

Snowflake Error Checking?

The result shows two data errors in mycsvtable/contacts3.csv.gz:

Number of columns in file (11) does not match that of the corresponding table (10)

In Row 1, a hyphen was mistakenly replaced with the pipe (|) character, the data file delimiter, effectively creating an additional column in the record.

```
14|Sophia|Christian|Turpis Ltd|lectus.pede@non.ca|1-962-503-3253|1-157-|350-3602|P.O. Box 824, 7971 Sagittis Rd.|Chattanooga|56188
```

Field delimiter '|' found while expecting record delimiter '\n'

In Row 5, an additional pipe (|) character was introduced after a hyphen, breaking the record.

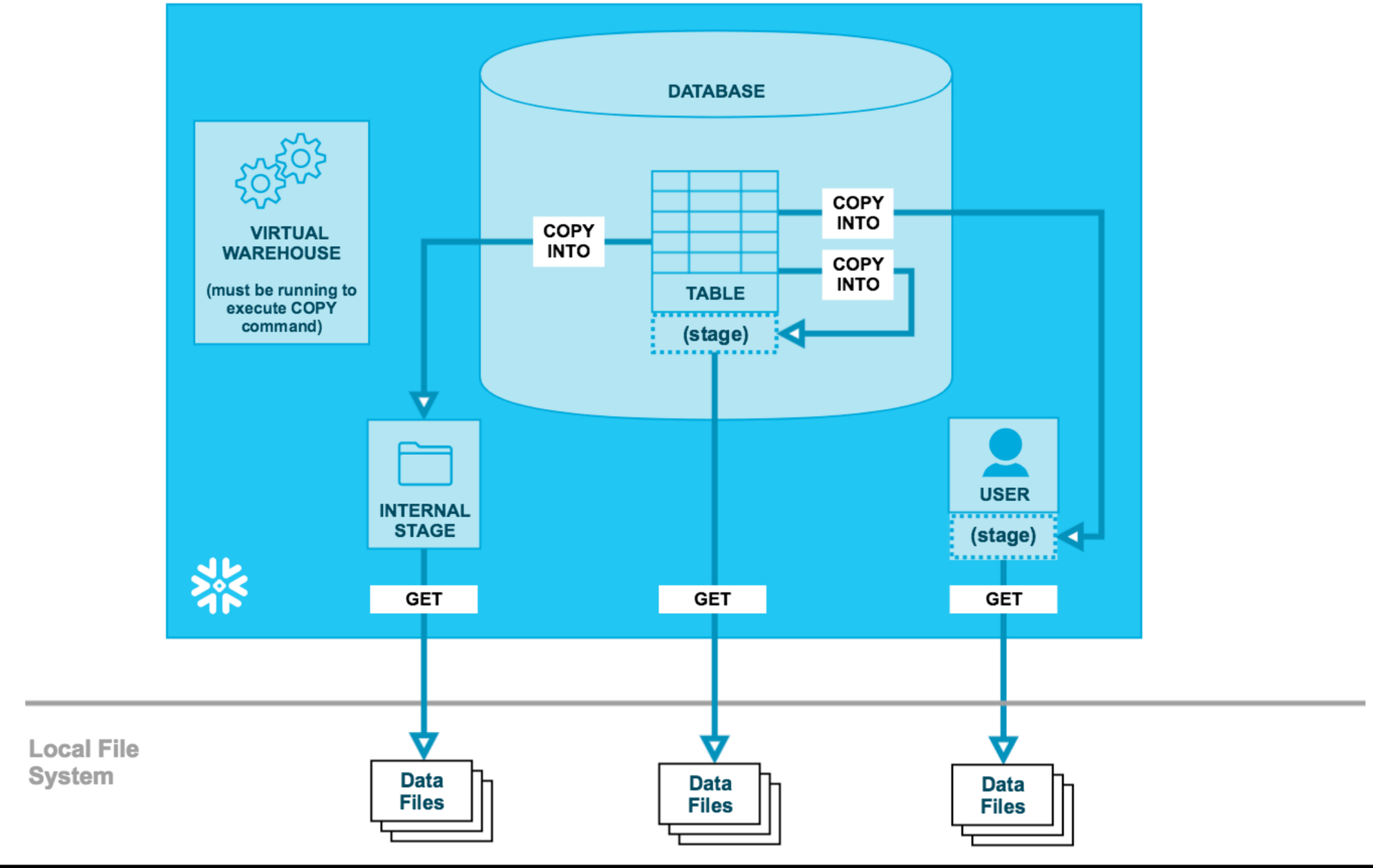
```
11|Ishmael|Burnett|Dolor Elit Pellentesque Ltd|vitae.erat@necmollisvitae.ca|1-872||600-7301|1-513-592-6779|P.O. Box 975, 553 Odio, Road|Hulste|63345
```

```
PUT file://C:\tmp\contacts\contacts3.csv @my_contacts_stg AUTO_COMPRESS=TRUE OVERWRITE=TRUE;
1 Row(s) produced. Time Elapsed: 1.953s
pysparktraining11#MYWAREHOUSE@EMPLOYEE_DB.EMPLOYEE_SCHEMA>PUT file://C:\tmp\contacts\contacts3.csv @my_contacts_stg AUTO_COMPRESS=TRUE OVERWRITE=TRUE;
+--------------+-----------------+-------------+-------------+--------------------+--------------------+----------+---------+
| source       | target          | source_size | target_size | source_compression | target_compression | status   | message |
|--------------+-----------------+-------------+-------------+--------------------+--------------------+----------+---------|
| contacts3.csv | contacts3.csv.gz |         771 |         576 | NONE               | GZIP               | UPLOADED |         |
+--------------+-----------------+-------------+-------------+--------------------+--------------------+----------+---------+
1 Row(s) produced. Time Elapsed: 0.765s
pysparktraining11#MYWAREHOUSE@EMPLOYEE_DB.EMPLOYEE_SCHEMA>
```

```
COPY INTO mycsvtable
  FROM @my_contacts_stg/contacts3.csv.gz
  FILE_FORMAT = (FORMAT_NAME = mycsvformat)
  ON_ERROR = 'skip_file';
pysparktraining11#MYWAREHOUSE@EMPLOYEE_DB.EMPLOYEE_SCHEMA>COPY INTO mycsvtable
+-------------------------------+--------+-------------+-------------+-------------+-------------+-------------+--
| file                          | status | rows_parsed | rows_loaded | error_limit | errors_seen | first_error | f
|-------------------------------+--------+-------------+-------------+-------------+-------------+-------------+--
| my_contacts_stg/contacts3.csv.gz | LOADED |           5 |           5 |           1 |           0 | NULL        |
+-------------------------------+--------+-------------+-------------+-------------+-------------+-------------+--
1 Row(s) produced. Time Elapsed: 1.484s
pysparktraining11#MYWAREHOUSE@EMPLOYEE_DB.EMPLOYEE_SCHEMA>
```

Snowflake Unloading?

LinkedIn
@TRRaveendra



Supported File Formats

Structured/Semi-structured	Type	Notes
Structured	Delimited (CSV, TSV, etc.)	Any valid singlebyte delimiter is supported; default is comma (i.e. CSV).
Semi-structured	JSON, Parquet	

Snowflake Unloading?

LinkedIn
@TRRaveendra

Create File Formats For UnLoad

```
CREATE OR REPLACE FILE FORMAT my_csv_unload_format
  TYPE = 'CSV'
  FIELD_DELIMITER = '|';

  CREATE OR REPLACE FILE FORMAT my_json_unload_format
  TYPE = 'JSON';

  CREATE OR REPLACE FILE FORMAT my_parquet_format
  TYPE = PARQUET
  COMPRESSION = SNAPPY;
```

```
4 Row(s) produced. Time Elapsed: 2.156s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>CREATE OR REPLACE FILE FORMAT my_csv_unload_format
                                                           TYPE = 'CSV'
                                                           FIELD_DELIMITER = '|';
+----------------------------------------------------+
| status                                             |
|----------------------------------------------------|
| File format MY_CSV_UNLOAD_FORMAT successfully created. |
+----------------------------------------------------+
1 Row(s) produced. Time Elapsed: 2.156s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA> CREATE OR REPLACE FILE FORMAT my_json_unload_format
                                                           TYPE = 'JSON';
+----------------------------------------------------+
| status                                             |
|----------------------------------------------------|
| File format MY_JSON_UNLOAD_FORMAT successfully created. |
+----------------------------------------------------+
1 Row(s) produced. Time Elapsed: 0.453s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>CREATE OR REPLACE FILE FORMAT my_parquet_format
                                                           TYPE = PARQUET
                                                           COMPRESSION = SNAPPY;
+----------------------------------------------------+
| status                                             |
|----------------------------------------------------|
| File format MY_PARQUET_FORMAT successfully created. |
+----------------------------------------------------+
1 Row(s) produced. Time Elapsed: 0.359s
```

Snowflake Unloading?

LinkedIn
@TRRaveendra

Creating a Named Stage For Unloading

The following example creates an internal stage that references the named file format object called my_csv_unload_format that was created in Preparing to Unload Data:

```
CREATE OR REPLACE STAGE my_unload_stage
  FILE_FORMAT = my_csv_unload_format;
```

```
1 Row(s) produced. Time Elapsed: 0.559s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>CREATE OR REPLACE STAGE my_unload_stage
                                                           FILE_FORMAT = my_csv_unload_format;
+-----------------------------------------------+
| status                                        |
|-----------------------------------------------|
| Stage area MY_UNLOAD_STAGE successfully created. |
+-----------------------------------------------+
1 Row(s) produced. Time Elapsed: 0.516s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>
```

COPY INTO @my_unload_stage/unload/ from EMP_BASIC;

list @my_unload_stage/unload;

```
1 Row(s) produced. Time Elapsed: 0.516s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>COPY INTO @my_unload_stage/unload/ from EMP_BASIC;
+---------------+-------------+--------------+
| rows_unloaded | input_bytes | output_bytes |
|---------------+-------------+--------------|
|            27 |        2093 |         1117 |
+---------------+-------------+--------------+
1 Row(s) produced. Time Elapsed: 0.594s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>list @my_unload_stage/unload
                                                         ;
+----------------------------------------+------+----------------------------------+------------------------------+
| name                                   | size | md5                              | last_modified                |
|----------------------------------------+------+----------------------------------+------------------------------|
| my_unload_stage/unload/data_0_0_0.csv.gz | 1120 | e34fc8cc6870938a18d30c1f23a0e8e7 | Thu, 16 Feb 2023 17:05:39 GMT |
+----------------------------------------+------+----------------------------------+------------------------------+
1 Row(s) produced. Time Elapsed: 0.344s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>
```

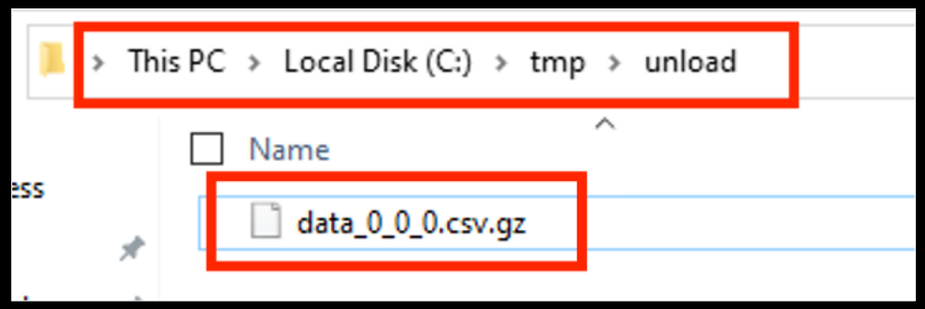
Snowflake Unloading?

LinkedIn
@TRRaveendra

Use the **GET** command to download the generated file(s) from the table stage to your local machine. The following example downloads the files to the data/unload directory:

GET @my_unload_stage/unload/data_0_0_0.csv.gz file:///tmp/unload;

```
1 Row(s) produced. Time Elapsed: 0.953s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>GET @my_unload_stage/unload/ file:///tmp/unload;
+------------------+------+------------+---------+
| file             | size | status     | message |
|------------------+------+------------+---------|
| data_0_0_0.csv.gz | 1117 | DOWNLOADED |         |
+------------------+------+------------+---------+
1 Row(s) produced. Time Elapsed: 0.891s
```



Snowflake Unloading?

LinkedIn
@TRRaveendra

Unloading Data to a Table Stage

Use the COPY INTO <location> command to unload all the rows from a table into one or more files in the stage for the table. The following example unloads data files to the stage using the named my_csv_unload_format file format created in Preparing to Unload Data. The statement prefixes the unloaded file(s) with unload/ to organize the files in the stage:

```
COPY INTO @%emp_basic/unload/ from emp_basic FILE_FORMAT = (FORMAT_NAME = 'my_csv_unload_format' COMPRESSION = NONE);
```

```
1 Row(s) produced. Time Elapsed: 0.891s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>COPY INTO @%emp_basic/unload/ from emp_basic FILE_FORMAT = (FORMAT_NAME = 'my_csv_unload_format' COMPRESSION = NONE);
+---------------+-------------+--------------+
| rows_unloaded | input_bytes | output_bytes |
|---------------+-------------+--------------|
|            27 |        2093 |         2093 |
+---------------+-------------+--------------+
1 Row(s) produced. Time Elapsed: 1.984s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>
```

Snowflake Unloading?

LinkedIn
@TRRaveendra

Unloading Data to Your User Stage

Use the COPY INTO <location> command to unload all the rows from a table into one or more files in your stage. The following example unloads data files to your user stage using the named my_csv_unload_format file format created in Preparing to Unload Data. The statement prefixes the unloaded file(s) with unload/ to organize the files in the stage:

```
COPY INTO @~/unload/ from emp_basic FILE_FORMAT = (FORMAT_NAME = 'my_csv_unload_format' COMPRESSION = NONE);
```

```
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>COPY INTO @~/unload/ from emp_basic FILE_FORMAT = (FORMAT_NAME = 'my_csv_unload_format' COMPRESSION = NONE);
+---------------+-------------+--------------+
| rows_unloaded | input_bytes | output_bytes |
|---------------+-------------+--------------|
|            27 |        2093 |         2093 |
+---------------+-------------+--------------+
1 Row(s) produced. Time Elapsed: 1.641s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>
```

Get Files to Local system from user Stage using @~

```
GET @~/unload/data_0_0_0.csv file:///tmp/unload;
```

```
1 Row(s) produced. Time Elapsed: 1.641s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>GET @~/unload/data_0_0_0.csv file:///tmp/unload;
+----------------+------+------------+---------+
| file           | size | status     | message |
|----------------+------+------------+---------|
| data_0_0_0.csv | 2093 | DOWNLOADED |         |
+----------------+------+------------+---------+
1 Row(s) produced. Time Elapsed: 2.109s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>
```

Semi-Structured Data

LinkedIn
@TRRaveendra

Load Semi Structured data from External Stage

Create Another External Stage
In the CITIBIKE_ZERO_TO_SNOWFLAKE worksheet, use the following command to create a stage that points to the bucket where the semi-structured JSON data is stored on AWS S3:

```
create stage nyc_weather url = 's3://snowflake-workshop-lab/zero-weather-nyc';
```

```
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>create stage nyc_weather
                                                         url = 's3://snowflake-workshop-lab/zero-weather-nyc';

+-----------------------------------------+
| status                                  |
|-----------------------------------------|
| Stage area NYC_WEATHER successfully created. |
+-----------------------------------------+
1 Row(s) produced. Time Elapsed: 2.547s
```

```
1 Row(s) produced. Time Elapsed: 2.547s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>list @nyc_weather;

+-----------------------------------------------------------------------+------+----------------------------------+-------------------------------+
| name                                                                  | size | md5                              | last_modified                 |
|-----------------------------------------------------------------------+------+----------------------------------+-------------------------------|
| s3://snowflake-workshop-lab/zero-weather-nyc/hourlyData-2016-10.json.gz | 27300 | 632312c472022bedd43f902efa189798 | Mon, 25 Jul 2022 10:50:43 GMT |
| s3://snowflake-workshop-lab/zero-weather-nyc/hourlyData-2016-11.json.gz | 28643 | 81c19c899223bd7c26d49e41c2402dbb | Mon, 25 Jul 2022 10:50:44 GMT |
| s3://snowflake-workshop-lab/zero-weather-nyc/hourlyData-2016-12.json.gz | 32969 | 93827a2d30d5301d8d45f566a798dd3c | Mon, 25 Jul 2022 10:50:45 GMT |
| s3://snowflake-workshop-lab/zero-weather-nyc/hourlyData-2016-7.json.gz  | 22592 | cf06bf9515c40e59171f3f3cc671a7ae | Mon, 25 Jul 2022 10:50:40 GMT |
| s3://snowflake-workshop-lab/zero-weather-nyc/hourlyData-2016-8.json.gz  | 29101 | 9d370a4bd315a99daf6177081b8926f0 | Mon, 25 Jul 2022 10:50:41 GMT |
| s3://snowflake-workshop-lab/zero-weather-nyc/hourlyData-2016-9.json.gz  | 29195 | 29fdc003be228442c4a1405ac07ccdaf | Mon, 25 Jul 2022 10:50:42 GMT |
| s3://snowflake-workshop-lab/zero-weather-nyc/hourlyData-2017-1.json.gz  | 32418 | 978f50c17f42c61f479581f5af03d3a4 | Mon, 25 Jul 2022 10:50:45 GMT |
| s3://snowflake-workshop-lab/zero-weather-nyc/hourlyData-2017-10.json.gz | 32310 | 1d4bb572de88e56d2904f90b9de20bdc | Mon, 25 Jul 2022 10:50:55 GMT |
| s3://snowflake-workshop-lab/zero-weather-nyc/hourlyData-2017-11.json.gz | 30581 | 24771215944eae2e95f63ba0e51be92c | Mon, 25 Jul 2022 10:50:56 GMT |
| s3://snowflake-workshop-lab/zero-weather-nyc/hourlyData-2017-12.json.gz | 31497 | a7bd32720435183446b02f0be91eaef2 | Mon, 25 Jul 2022 10:50:57 GMT |
| s3://snowflake-workshop-lab/zero-weather-nyc/hourlyData-2017-2.json.gz  | 29656 | 3fc24dbb9c1fea37947665554a8d413a | Mon, 25 Jul 2022 10:50:46 GMT |
```

Semi-Structured Data

LinkedIn
@TRRaveendra

Create Weather Table

create table json_weather_data(v variant);

```
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>create table json_weather_data(v variant);
+-------------------------------------------+
| status                                    |
|-------------------------------------------|
| Table JSON_WEATHER_DATA successfully created. |
+-------------------------------------------+
1 Row(s) produced. Time Elapsed: 1.672s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>
```

copy into json_weather_data from @nyc_weather file_format = (type = json strip_outer_array = true);

```
1 Row(s) produced. Time Elapsed: 1.672s
pysparktraining11#COMPUTE_WH@EMPLOYEE_DB.EMPLOYEE_SCHEMA>copy into json_weather_data
                                                          from @nyc_weather
                                                              file_format = (type = json strip_outer_array = true);
+-----------------------------------------------------------------------------+--------+-------------+-------------+-------------+------
| file                                                                        | status | rows_parsed | rows_loaded | error_limit | error
|-----------------------------------------------------------------------------+--------+-------------+-------------+-------------+------
| s3://snowflake-workshop-lab/zero-weather-nyc/hourlyData-2016-10.json.gz     | LOADED |        1960 |        1960 |           1 |
| s3://snowflake-workshop-lab/zero-weather-nyc/hourlyData-2017-5.json.gz      | LOADED |        2231 |        2231 |           1 |
| s3://snowflake-workshop-lab/zero-weather-nyc/hourlyData-2018-12.json.gz     | LOADED |        2226 |        2226 |           1 |
| s3://snowflake-workshop-lab/zero-weather-nyc/hourlyData-2018-9.json.gz      | LOADED |        2100 |        2100 |           1 |
| s3://snowflake-workshop-lab/zero-weather-nyc/hourlyData-2016-11.json.gz     | LOADED |        1936 |        1936 |           1 |
| s3://snowflake-workshop-lab/zero-weather-nyc/hourlyData-2017-2.json.gz      | LOADED |        2004 |        2004 |           1 |
```

Semi-Structured Data

LinkedIn
@TRRaveendra

Create Weather View

```
// create a view that will put structure onto the semi-structured data
create or replace view json_weather_data_view as
select
    v:obsTime::timestamp as observation_time,
    v:station::string as station_id,
    v:name::string as city_name,
    v:country::string as country,
    v:latitude::float as city_lat,
    v:longitude::float as city_lon,
    v:weatherCondition::string as weather_conditions,
    v:coco::int as weather_conditions_code,
    v:temp::float as temp,
    v:prcp::float as rain,
    v:tsun::float as tsun,
    v:wdir::float as wind_dir,
    v:wspd::float as wind_speed,
    v:dwpt::float as dew_point,
    v:rhum::float as relative_humidity,
    v:pres::float as pressure
from
    json_weather_data
where
    station_id = '72502';
```

https://github.com/raveendratal/snowflake/blob/main/json_weather_data.sql

LinkedIn
@TRRaveendra

Learn 📖 & Lead ♗

TR Raveendra