**Copy Into Command & Load History**

The COPY INTO command is used to **load data into Snowflake tables** from files in stages (internal or external like S3, Azure Blob, GCP). In my last (Siemens) project, I used COPY INTO to bulk load Oracle export files from Azure Blob into Snowflake tables. It was efficient for large initial loads before switching to Snowpipe for continuous ingestion.”

Snowflake automatically **records metadata** about all load operations (success, failure, errors). . Snowflake also maintains **load history**, which lets us track which files were loaded, when, and whether there were any errors — this was very useful for monitoring and reprocessing failed files.”

--Create Table

create or replace table CONTACTS ( id integer, last\_name string, first\_name string, company string, email string, workphone string, cellphone string, streetaddress string, city string, postalcode string);

-- copy command

COPY INTO contacts FROM @ext\_aws\_stage

file\_format = ( type = 'CSV'

field\_delimiter = '|'

skip\_header = 1)

files = ('contacts1.csv','contacts2.csv','contacts4.csv');

-- check data

select \* from contacts ;

-- Truncate Table

truncate table contacts ;

-- Handle Error using ON\_ERROR

COPY INTO contacts FROM @ext\_aws\_stage file\_format = (type = 'CSV'

field\_delimiter = '|' skip\_header = 1)

files = ('contacts1.csv','contacts2.csv','contacts3.csv')

on\_error = 'CONTINUE';

-- Validate data and count

select \* from contacts order by 1;

--truncate the table

truncate table contacts ;

-- By default in our first copy ABORT\_STATEMENT was applied. If we don’t give on\_error option it is abort statement, Lets see the

-- the result by specifying explicitly.

COPY INTO contacts FROM @ext\_aws\_stage file\_format = ( type = 'CSV'

field\_delimiter = '|' skip\_header = 1)

files = ('contacts1.csv','contacts2.csv','contacts3.csv')

on\_error = 'ABORT\_STATEMENT';

-- Validate data and count select \* from contacts order by 1;

--Truncate before re-running copy to clean up metadata information truncate table contacts ;

-- Handling Error Using = SKIP\_FILE

COPY INTO contacts FROM @ext\_aws\_stage file\_format = ( type = 'CSV'

field\_delimiter = '|' skip\_header = 1)

files = ('contacts1.csv','contacts2.csv','contacts3.csv')

on\_error = 'SKIP\_FILE';

-- Validate data

select \* from contacts order by 1;

--Truncate before re-running copy to clean up metadata information

truncate table contacts ;

-- Handling error using SKIP\_FILE\_<error\_number>

COPY INTO contacts FROM @ext\_aws\_stage file\_format = ( type = 'CSV'

field\_delimiter = '|' skip\_header = 1)

files = ('contacts1.csv','contacts2.csv','contacts3.csv')

on\_error = 'SKIP\_FILE\_2';

-- Validate data and count

select \* from contacts order by 1;

--Truncate before re-running copy to clean up metadata information

truncate table contacts ;

-- Error handling using the ON\_ERROR option = SKIP\_FILE\_<percent\_numbe>%

COPY INTO contacts FROM @ext\_aws\_stage file\_format = ( type = 'CSV'

field\_delimiter = '|' skip\_header = 1)

files = ('contacts1.csv','contacts2.csv','contacts3.csv') on\_error = 'SKIP\_FILE\_60%';

-- Validate data and count

select \* from contacts order by 1;

--Cleanup drop stage if exists ext\_aws\_satge ; drop table contacts ;

-------------------------------------------------------------------------

COPY INTO contacts FROM @ext\_aws\_stage file\_format = (type = 'CSV'

field\_delimiter = '|' skip\_header = 1)

files = ('contacts1.csv','contacts2.csv','contacts4.csv','contacts5.csv')

size\_limit = 1300 ; -- bytes

COPY INTO contacts FROM @ext\_aws\_stage file\_format = ( type = 'CSV'

field\_delimiter = '|' skip\_header = 1)

files = ('contacts1.csv','contacts4.csv')

ON\_ERROR = 'ABORT\_STATEMENT';

list @ext\_aws\_stage;

--------------------------------------------------------------------------

truncate table contacts ;

-- copy command with RETURN\_FAILED\_ONLY

COPY INTO contacts FROM @ext\_aws\_stage file\_format = ( type = 'CSV'

field\_delimiter = '|' skip\_header = 1)

files = ('contacts1.csv','contacts2.csv','contacts3.csv','contacts4.csv','contacts5.csv')

on\_error = 'CONTINUE'

RETURN\_FAILED\_ONLY = TRUE;

-------------------------------------------------------------------------

--Create Table

create or replace table CONTACTS ( id integer, last\_name varchar, first\_name string, company string, email string, workphone string, cellphone string, streetaddress string, city string, postalcode string);

--truncate table

truncate table contacts ;

-- copy with enforcelength

COPY INTO contacts FROM @ext\_aws\_stage file\_format = ( type = 'CSV'

field\_delimiter = '|' skip\_header = 1)

files = ('contacts1.csv','contacts2.csv','contacts4.csv','contacts5.csv')

on\_error = 'CONTINUE'

ENFORCE\_LENGTH = TRUE;

-- Truncate Table

truncate table contacts ;

-- Copy with truncatecolumns

COPY INTO contacts FROM @ext\_aws\_stage file\_format = ( type = 'CSV'

field\_delimiter = '|'

skip\_header = 1)

files = ('contacts1.csv','contacts2.csv','contacts4.csv','contacts5.csv')

on\_error = 'CONTINUE'

TRUNCATECOLUMNS = FALSE;

-- check data

select \* from contacts ;

-- copy with enforcelength

COPY INTO contacts FROM @ext\_aws\_stage file\_format = ( type = 'CSV'

field\_delimiter = '|' skip\_header = 1)

files = ('contacts1.csv','contacts2.csv','contacts4.csv','contacts5.csv') on\_error = 'CONTINUE'

ENFORCE\_LENGTH = FALSE;

-- Truncate Table

truncate table contacts ;

-- Copy with truncatecolumns

COPY INTO contacts FROM @ext\_aws\_stage file\_format = ( type = 'CSV'

field\_delimiter = '|' skip\_header = 1)

files = ('contacts1.csv','contacts2.csv','contacts4.csv','contacts5.csv') on\_error = 'CONTINUE'

TRUNCATECOLUMNS = TRUE;

-- Cleanup drop stage ext\_aws\_stage ; drop table contacts ;

-----------------------------------------------------------------------------------------

select \* from SNOWFLAKE.ACCOUNT\_USAGE.LOAD\_HISTORY;

-----------------------------------------------------------------------------------------

create or replace table VALIDATION\_MODE\_CONTACTS as

select \* from practice\_db.snowflake.contacts where 1 = 2;

select \* from validation\_mode\_contacts;

-- Use the validation mode, it will not load the data, just return the errors

COPY INTO VALIDATION\_MODE\_CONTACTS FROM @ext\_aws\_stage file\_format = ( type = 'CSV'

field\_delimiter = '|' skip\_header = 1)

files = ('contacts1.csv','contacts2.csv')

validation\_mode = RETURN\_ERRORS ;

COPY INTO VALIDATION\_MODE\_CONTACTS FROM @ext\_aws\_stage file\_format = ( type = 'CSV'

field\_delimiter = '|' skip\_header = 1)

files = ('contacts1.csv','contacts2.csv')

validation\_mode = RETURN\_ALL\_ERRORS ;

COPY INTO VALIDATION\_MODE\_CONTACTS FROM @ext\_aws\_stage file\_format = ( type = 'CSV'

field\_delimiter = '|' skip\_header = 1)

files = ('contacts1.csv','contacts2.csv')

validation\_mode = RETURN\_2\_ROWS ;

--------------------------------------------------------------------------------------------

create or replace table CONTACTS\_REJECTED (error string, file string, rejected\_record string);

insert into CONTACTS\_REJECTED select error, file, rejected\_record from table(result\_scan(last\_query\_id()));

select \* from CONTACTS\_REJECTED;

select \* from contacts\_rejected ; --Splitted Rejected record

select sp

select error, file, split\_part(rejected\_record,'|',1) as ID, split\_part(rejected\_record,'|',2) as LAST\_NAME, split\_part(rejected\_record,'|',3) as FIRST\_NAME, split\_part(rejected\_record,'|',4) as COMPANY, split\_part(rejected\_record,'|',5) as EMAIL, split\_part(rejected\_record,'|',6) as WORK\_PHONE, split\_part(rejected\_record,'|',7) as CELL\_PHONE, split\_part(rejected\_record,'|',8) as STREET\_ADDRESS, split\_part(rejected\_record,'|',9) as CITY, split\_part(rejected\_record,'|',10) as POSTAL\_CODE, split\_part(rejected\_record,'|',11) as OTHER1, split\_part(rejected\_record,'|',12) as OTHER2, split\_part(rejected\_record,'|',13) as OTHER3 from CONTACTS\_REJECTED ;