```
use role developer;

use database debadatta_mohanty_db;

use warehouse training_wh;


create transient schema training_schema_csvdataload;


use schema training_schema_csvdataload;


create table customer like training_schema_debadatta.customer;


select * from customer;


show file formats;


--1. Create File Format

create or replace FILE FORMAT CSV_INPUT_FORMAT TYPE = 'CSV' COMPRESSION = 'AUTO'
FIELD_DELIMITER = ',' RECORD_DELIMITER = '\n' SKIP_HEADER = 1

FIELD_OPTIONALLY_ENCLOSED_BY = '"' TRIM_SPACE = TRUE ERROR_ON_COLUMN_COUNT_MISMATCH
= FALSE ESCAPE = 'NONE' ESCAPE_UNENCLOSED_FIELD = '\134'

DATE_FORMAT = 'AUTO' TIMESTAMP_FORMAT = 'AUTO' NULL_IF = ('\\N');


-- create a stage with the file format

create or replace stage CSV_INPUT_stage file_format = CSV_INPUT_FORMAT;


-- put data into internal named stage

/*

--table stage

put
file://C:\Users\Snowflake1\Downloads\SnowflakeLabShare_PM\Data\CSVData\customer\none\*.csv
```

```
@%CUSTOMER/csv AUTO_COMPRESS = FALSE SOURCE_COMPRESSION = NONE OVERWRITE= FALSE
PARALLEL =10;

--nmed internal stage

put
file://C:\Users\Snowflake1\Downloads\SnowflakeLabShare_PM\Data\CSVData\customer\none\*.csv
@csv_input_stage/csv_input/customer/ AUTO_COMPRESS = TRUE SOURCE_COMPRESSION = NONE
OVERWRITE= FALSE PARALLEL =10;

*/


--put file://C:\SnowflakeLab\Data\CSV-Data-SnowflakeCOHORT\customer\none\*.csv
@csv_input_stage/csv_input/customer/ AUTO_COMPRESS = TRUE SOURCE_COMPRESSION = NONE
OVERWRITE= FALSE PARALLEL =10;


list @%CUSTOMER;


list @CSV_INPUT_stage;


--remove @CSV_INPUT_stage;


COPY INTO customer FROM '@csv_input_stage/csv_input/customer/' FILE_FORMAT =
CSV_INPUT_FORMAT

ON_ERROR = 'CONTINUE' PURGE = TRUE VALIDATION_MODE = RETURN_ALL_ERRORS;


COPY INTO customer FROM '@csv_input_stage/csv_input/customer/customer_0_1_0.csv.gz'
FILE_FORMAT =  CSV_INPUT_FORMAT

ON_ERROR = 'CONTINUE' PURGE = FALSE ;


select * from customer;


select * from table(validate(CUSTOMER, job_id => '019d1123-0000-32fb-0000-c7c90003f1a6'));
```

```
select cus.$1 as ROW_ID
, cus.$2 as "CUSTOMER ADDRESS"
,cus.$3  as"MARKET SEGMENT"
,cus.$4  as "PHONE NUMBER"
,cus.$5 as "COMMENTS"
,cus.$6 as "CUSTOMER NAME"
,cus.$7 as "NATKEY"
,cus.$8 as "CUSTOMER KEY"
,cus.$9 "ACCOUNT BALANCE"   from '@csv_input_stage/csv_input/customer/' cus;

select cus.$1 as ROW_ID
, cus.$2 as "CUSTOMER ADDRESS"
,cus.$3  as"MARKET SEGMENT"
,cus.$4  as "PHONE NUMBER"
,cus.$5 as "COMMENTS"
,cus.$6 as "CUSTOMER NAME"
,cus.$7 as "NATKEY"
,cus.$8 as "CUSTOMER KEY"
,cus.$9 "ACCOUNT BALANCE"   from
'@csv_input_stage/csv_input/customer_compressed/customer*.gz' cus;

select $1,$2,$3 from '@csv_input_stage/csv_input/customer_compressed/customer_0_0_0.csv.gz';

list @csv_input_stage;

select * from (
select cus.$1::number as ROW_ID
, cus.$2::VARCHAR as "CUSTOMER ADDRESS"
,cus.$3::VARCHAR  as"MARKET SEGMENT"
```

```
,cus.$4::VARCHAR  as "PHONE NUMBER"

,cus.$5::VARCHAR as "COMMENTS"

,cus.$6::VARCHAR as "CUSTOMER NAME"

,cus.$7::VARCHAR as "NATKEY"

,cus.$8::NUMBER as "CUSTOMER KEY"

,cus.$9::FLOAT "ACCOUNT BALANCE"   from
'@csv_input_stage/csv_input/customer/customer_0_1_0.csv.gz' cus

 )

order by "CUSTOMER KEY" ;



create or replace view CUS_FILE_VIEW

as

(select cus.$1::number as ROW_ID

, cus.$2::VARCHAR as "CUSTOMER ADDRESS"

,cus.$3::VARCHAR  as"MARKET SEGMENT"

,cus.$4::VARCHAR  as "PHONE NUMBER"

,cus.$5::VARCHAR as "COMMENTS"

,cus.$6::VARCHAR as "CUSTOMER NAME"

,cus.$7::VARCHAR as "NATKEY"

,cus.$8::NUMBER as "CUSTOMER KEY"

,cus.$9::FLOAT as "ACCOUNT BALANCE"   from '@CSV_INPUT_STAGE/csv_input/customer/' cus

 );



select * from CUS_FILE_VIEW;


COPY INTO
customer(C_CUSTKEY,C_NAME,C_ADDRESS,C_NATIONKEY,C_PHONE,C_ACCTBAL,C_MKTSEGMENT,
C_COMMENT)
```

```
FROM (
  select
cus.$8::NUMBER,cus.$6::VARCHAR,cus.$2::VARCHAR,cus.$7::VARCHAR,cus.$4::VARCHAR,cus.$9::FLOAT,
cus.$3::VARCHAR,cus.$5::VARCHAR
  from '@csv_input_stage/csv_input/customer/' cus)
FILE_FORMAT = 'CSV_INPUT_FORMAT' ON_ERROR = 'CONTINUE' PURGE = FALSE;


select
cus.$8::NUMBER,cus.$6::VARCHAR,cus.$2::VARCHAR,cus.$7::VARCHAR,cus.$4::VARCHAR,cus.$9::FLOAT,
cus.$3::VARCHAR,cus.$5::VARCHAR
  from '@csv_input_stage/csv_input/customer/' cus;


truncate table customer;


select count(*) from customer;


--working example
COPY INTO
customer(C_CUSTKEY,C_NAME,C_ADDRESS,C_NATIONKEY,C_PHONE,C_ACCTBAL,C_MKTSEGMENT,
C_COMMENT)
FROM ( select
cus.$8::NUMBER,cus.$6::VARCHAR,cus.$2::VARCHAR,cus.$7::VARCHAR,cus.$4::VARCHAR,cus.$9::FLOAT,
cus.$3::VARCHAR,cus.$5::VARCHAR from '@csv_input_stage/csv_input/customer/' cus)
FILE_FORMAT = 'CSV_INPUT_FORMAT' ON_ERROR = 'CONTINUE' PURGE = FALSE;


select cus.$8::NUMBER
--
,cus.$6::VARCHAR,cus.$2::VARCHAR,cus.$7::VARCHAR,cus.$4::VARCHAR,cus.$9::FLOAT,cus.$3::VARCHA
R,cus.$5::VARCHAR
from '@csv_input_stage/csv_input/customer/' cus;


list @csv_input_stage;
```

```
/* ***********this does not work as validation mode does not work for transformaton
******************************/

COPY INTO
customer(C_CUSTKEY,C_NAME,C_ADDRESS,C_NATIONKEY,C_PHONE,C_ACCTBAL,C_MKTSEGMENT,
C_COMMENT)

FROM ( select
cus.$8::NUMBER,cus.$6::VARCHAR,cus.$2::VARCHAR,cus.$7::VARCHAR,cus.$4::VARCHAR,cus.$9::FLOAT,

    cus.$3::VARCHAR,cus.$5::VARCHAR

    from '@csv_input_stage/csv_input/customer/' cus)

FILE_FORMAT = 'CSV_INPUT_FORMAT' ON_ERROR = 'CONTINUE' PURGE = FALSE VALIDATION_MODE =
RETURN_ALL_ERRORS;




/*   you can use merge command on the next set of files, which might have duplicates  */



merge into customer cc using CUS_FILE_VIEW cv

  on cc.C_CUSTKEY = cv."CUSTOMER KEY"

  when not matched then

     insert (C_CUSTKEY,C_NAME,C_ADDRESS,C_NATIONKEY,C_PHONE,C_ACCTBAL,C_MKTSEGMENT,
C_COMMENT)

    values ( cv."CUSTOMER KEY",cv."CUSTOMER NAME",cv."CUSTOMER
ADDRESS",cv."NATKEY",cv."PHONE NUMBER",cv."ACCOUNT BALANCE",cv."MARKET
SEGMENT",cv."COMMENTS" )

  when matched then

  update set cc.C_ACCTBAL = cv."ACCOUNT BALANCE";


select * from table(information_schema.copy_history(table_name=>'CUSTOMER', start_time=>
dateadd(hours, -1, current_timestamp())));
```

```
-- Load Data from

\\snowflake3-0\Users\Snowflake43\C:\SnowflakeLab\Data\CSV-Data-
SnowflakeCOHORT\customer\none

;


create or replace FILE FORMAT CSV_INPUT_FORMAT_SUPPLIER TYPE = 'CSV'

COMPRESSION = 'AUTO'

FIELD_DELIMITER = '@~'

RECORD_DELIMITER = '\n'

SKIP_HEADER = 1

FIELD_OPTIONALLY_ENCLOSED_BY = '""'

TRIM_SPACE = TRUE

ERROR_ON_COLUMN_COUNT_MISMATCH = FALSE

ESCAPE = 'NONE'

ESCAPE_UNENCLOSED_FIELD = '\134'

DATE_FORMAT = 'AUTO'

TIMESTAMP_FORMAT = 'AUTO'

NULL_IF = ('\\N');


create or replace stage CSV_INPUT_stage_supplier file_format = CSV_INPUT_FORMAT_SUPPLIER;


select count(*) from supplier;


DESC TABLE supplier;


list @csv_input_stage_supplier;
```

```
COPY INTO supplier(S_SUPPKEY,S_NAME,S_ADDRESS,S_NATIONKEY,S_PHONE,S_ACCTBAL,
S_COMMENT)
FROM ( select
    sup.$7::NUMBER,
    sup.$2::VARCHAR,
    sup.$5::VARCHAR,
    sup.$8::NUMBER,
    sup.$4::VARCHAR,
    sup.$3::FLOAT,
    sup.$6::VARCHAR
    from '@csv_input_stage_supplier/csv_input/supplier/' sup)
FILE_FORMAT = 'CSV_INPUT_FORMAT_SUPPLIER' ON_ERROR = 'CONTINUE' PURGE = FALSE ;


 Select sup.$2::VARCHAR  from '@csv_input_stage_supplier/csv_input/supplier/supplier_0_0_0.csv.gz'
sup;


 list @csv_input_stage_supplier;


 select count(*) from supplier;


 select count(*) from training_schema_debadatta.supplier;
```