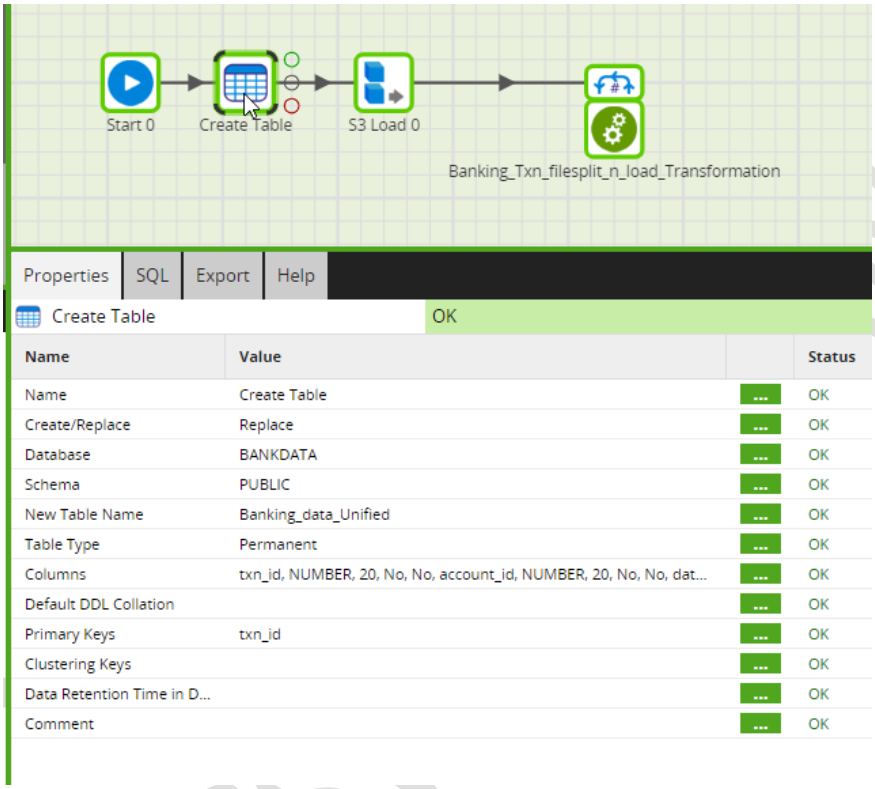
Splitting a Unified file with multiple years data and loading it in respective tables in Snowflake

This involves below main steps and sub steps:

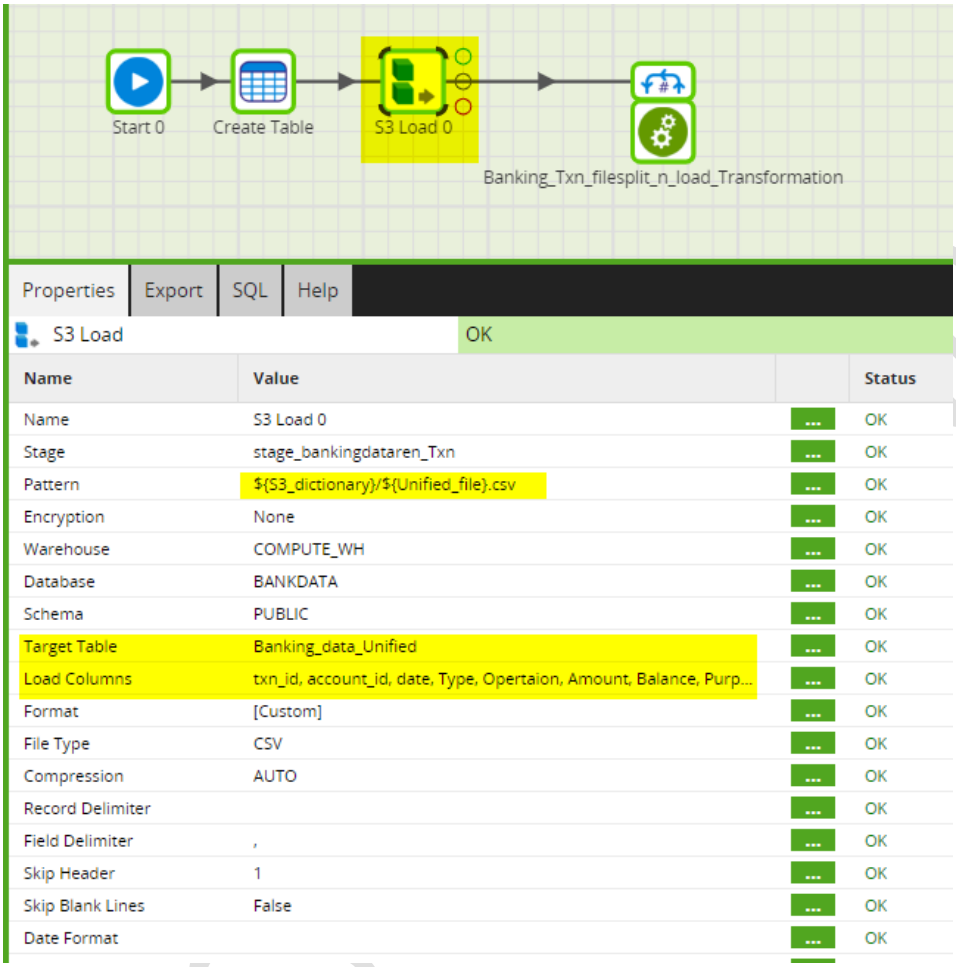
1. Load unified data from CSV in S3 to Snowflake table
    - Create a table structure in snowflake – using CREATE TABLE



Name	Value		Status
Name	Create Table	...	OK
Create/Replace	Replace	...	OK
Database	BANKDATA	...	OK
Schema	PUBLIC	...	OK
New Table Name	Banking_data_Unified	...	OK
Table Type	Permanent	...	OK
Columns	txn_id, NUMBER, 20, No, No, account_id, NUMBER, 20, No, No, dat...	...	OK
Default DDL Collation		...	OK
Primary Keys	txn_id	...	OK
Clustering Keys		...	OK
Data Retention Time in D...		...	OK
Comment		...	OK

    - Use S3 LOAD to load into snowflake

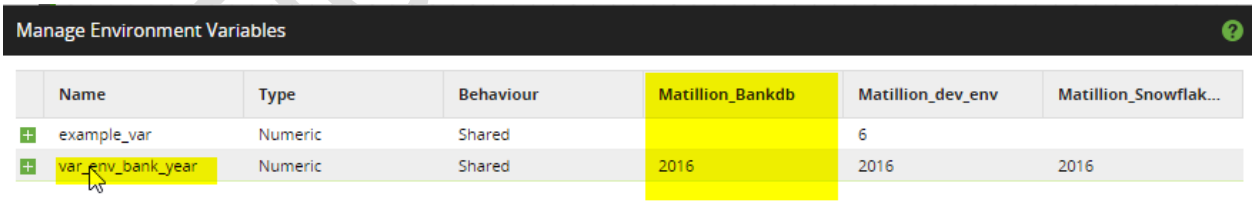
Splitting a Unified file with multiple years data and loading it in respective tables in Snowflake



Name	Value		Status
Name	S3 Load 0	...	OK
Stage	stage_bankingdataren_Txn	...	OK
Pattern	${S3_dictionary}/${Unified_file}.csv	...	OK
Encryption	None	...	OK
Warehouse	COMPUTE_WH	...	OK
Database	BANKDATA	...	OK
Schema	PUBLIC	...	OK
Target Table	Banking_data_Unified	...	OK
Load Columns	txn_id, account_id, date, Type, Opertaion, Amount, Balance, Purp...	...	OK
Format	[Custom]	...	OK
File Type	CSV	...	OK
Compression	AUTO	...	OK
Record Delimiter		...	OK
Field Delimiter	,	...	OK
Skip Header	1	...	OK
Skip Blank Lines	False	...	OK
Date Format		...	OK

-
-

2. Since we need to loop/iterarte YEAR ,create a variable for year

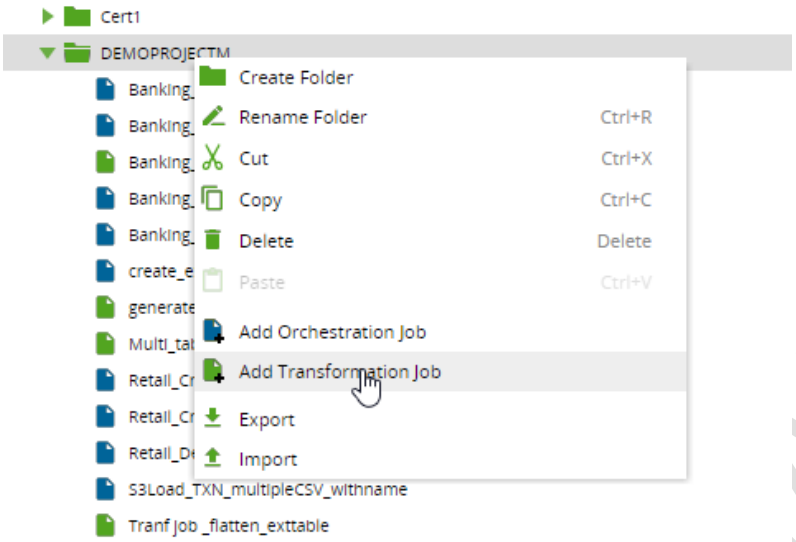


Name	Type	Behaviour	Matillion_Bankdb	Matillion_dev_env	Matillion_Snowflak...
example_var	Numeric	Shared		6	
var_env_bank_year	Numeric	Shared	2016	2016	2016

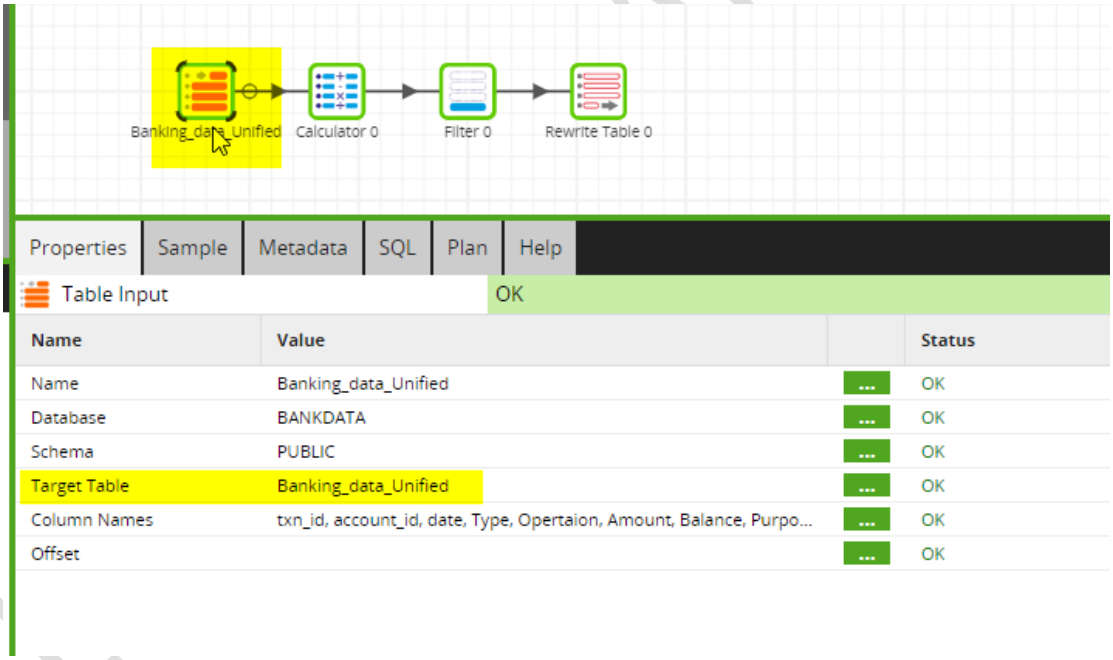
-

3. Creating transformations:
   - Create a transformation Job

Splitting a Unified file with multiple years data and loading it in respective tables in Snowflake

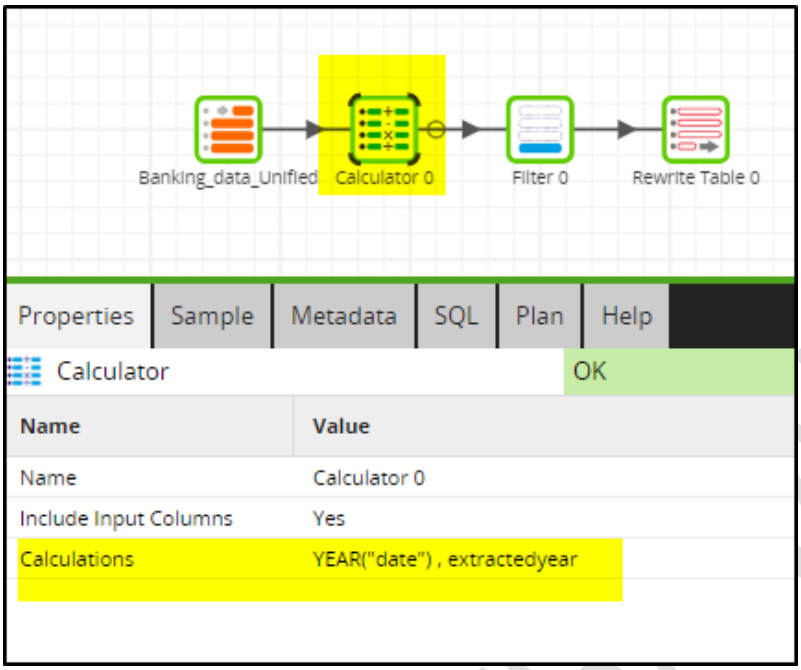


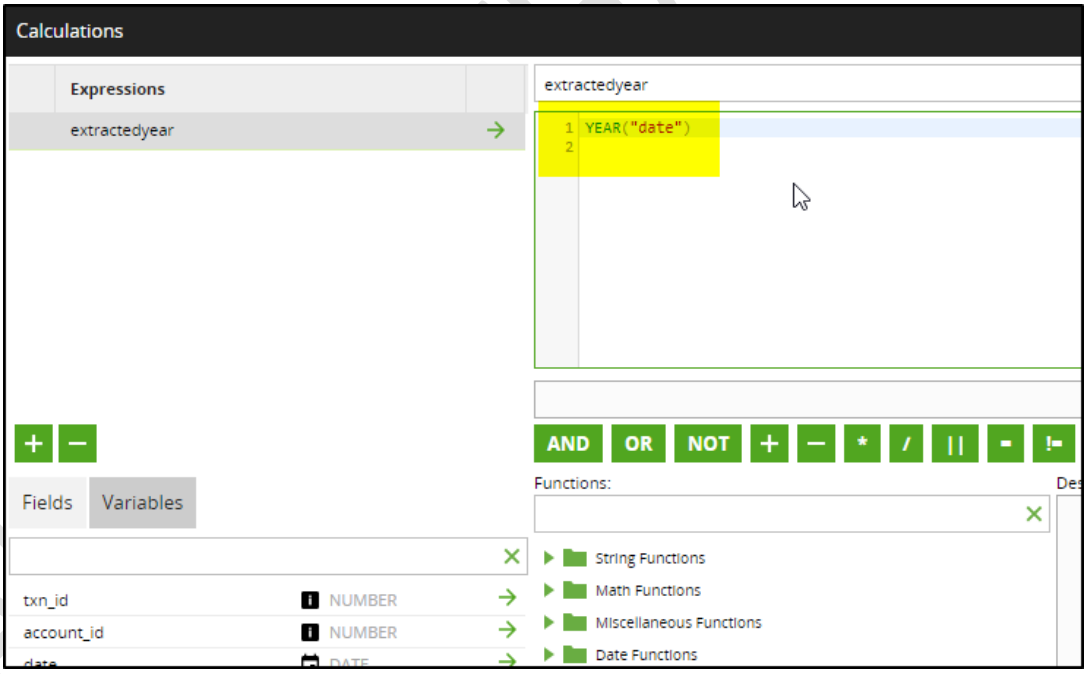
- Use TABLE INPUT , to bring in data to transform



- Use CALCULATER to add a YEAR column

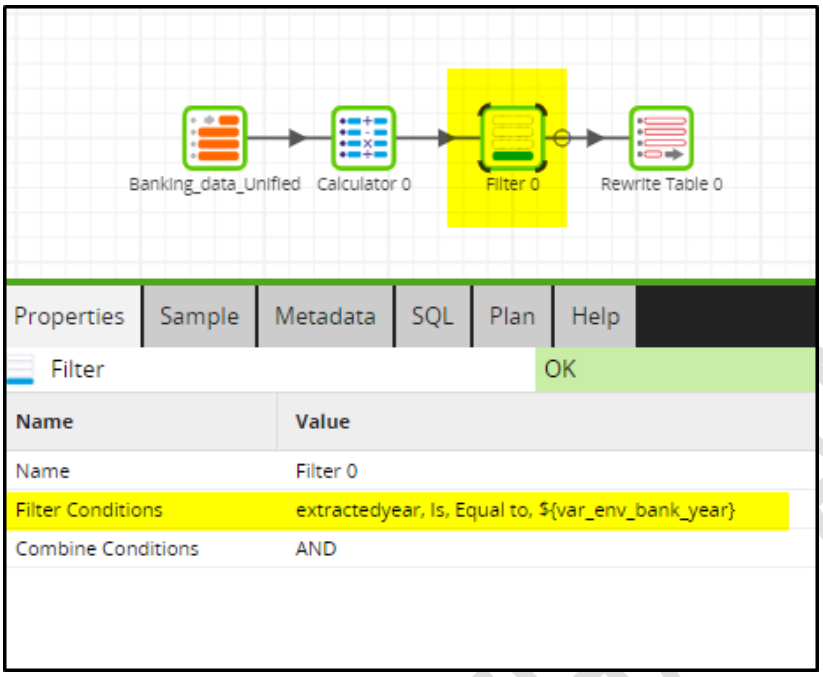
Splitting a Unified file with multiple years data and loading it in respective tables in Snowflake

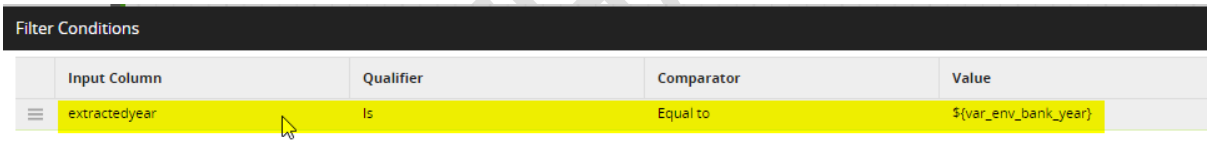
- 

- 

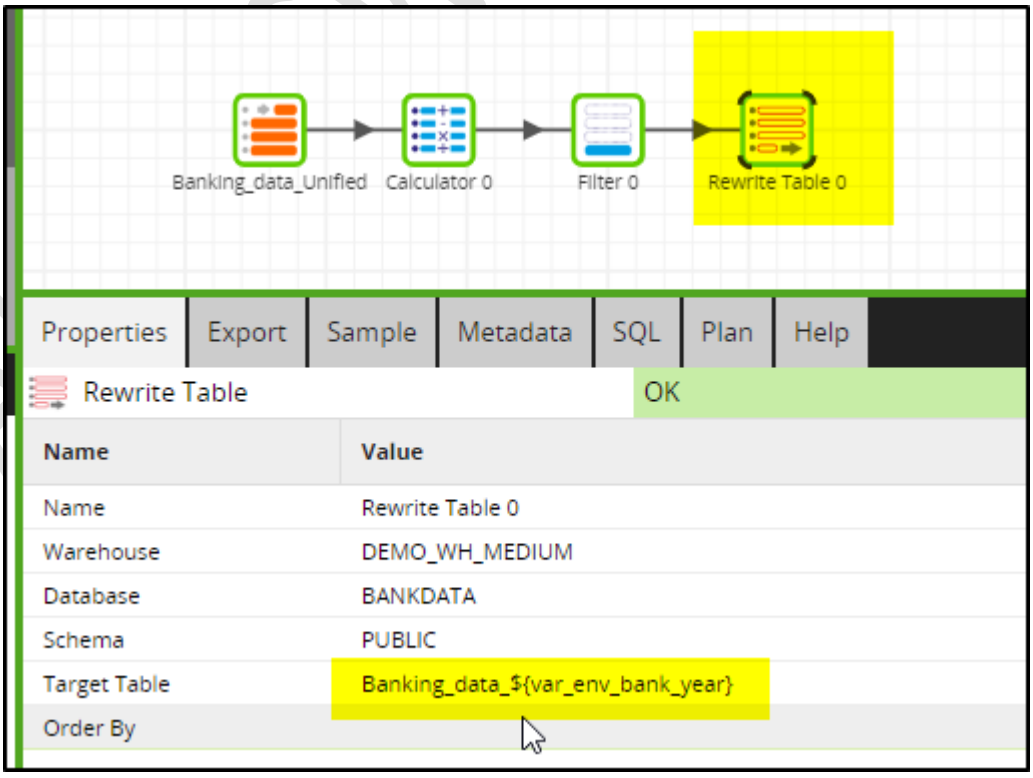
  - Use FILTER to filter out the year

Splitting a Unified file with multiple years data and loading it in respective tables in Snowflake



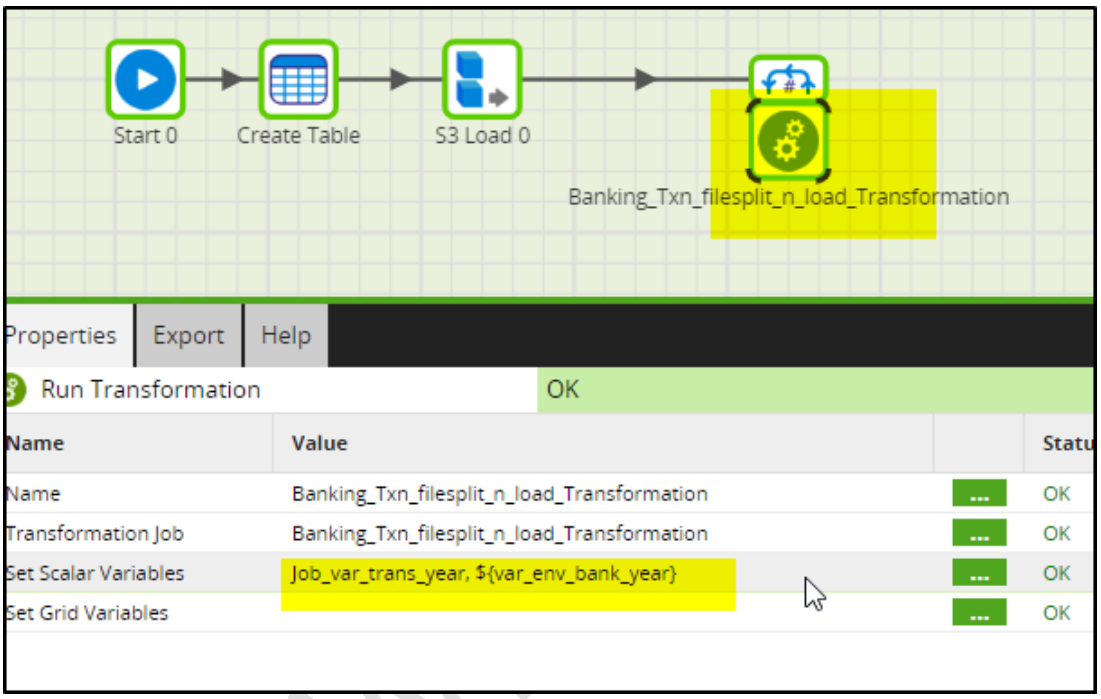


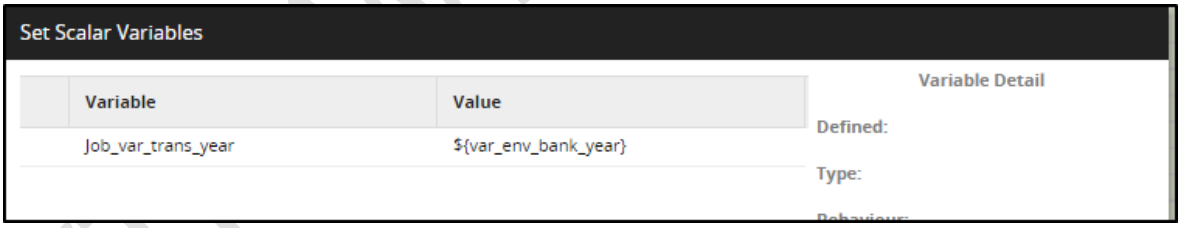
- Write into the respective tables



4. To tie the Orchestration and Iteration Job together we use 'Run transformation' component. But for the variable to loop when transforming we need a LOOP ITERATOR

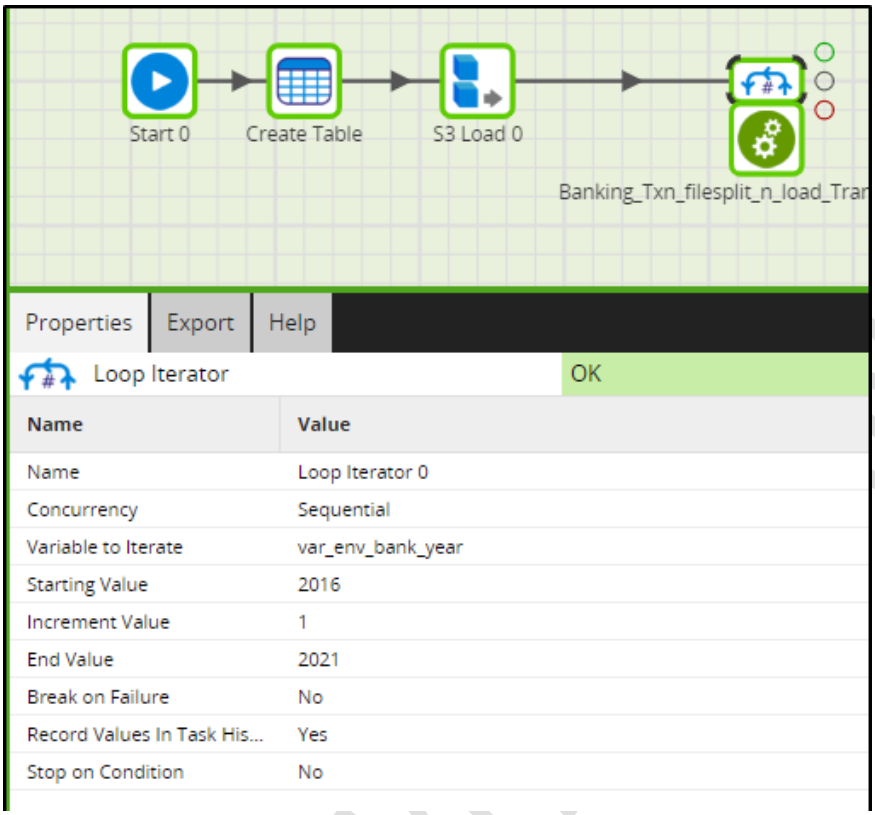
Splitting a Unified file with multiple years data and loading it in respective tables in Snowflake

- CREATE JOB VARIABLE IN TRANSFORMATION JOB –> Job_var_trans_year
  - This will be used to pass variable (here year) from Orchestration job to Transformation job , thru 'run transformation' componenets
- RUN TRANSFORMATION
  - 

  - 

- LOOP ITERATOR

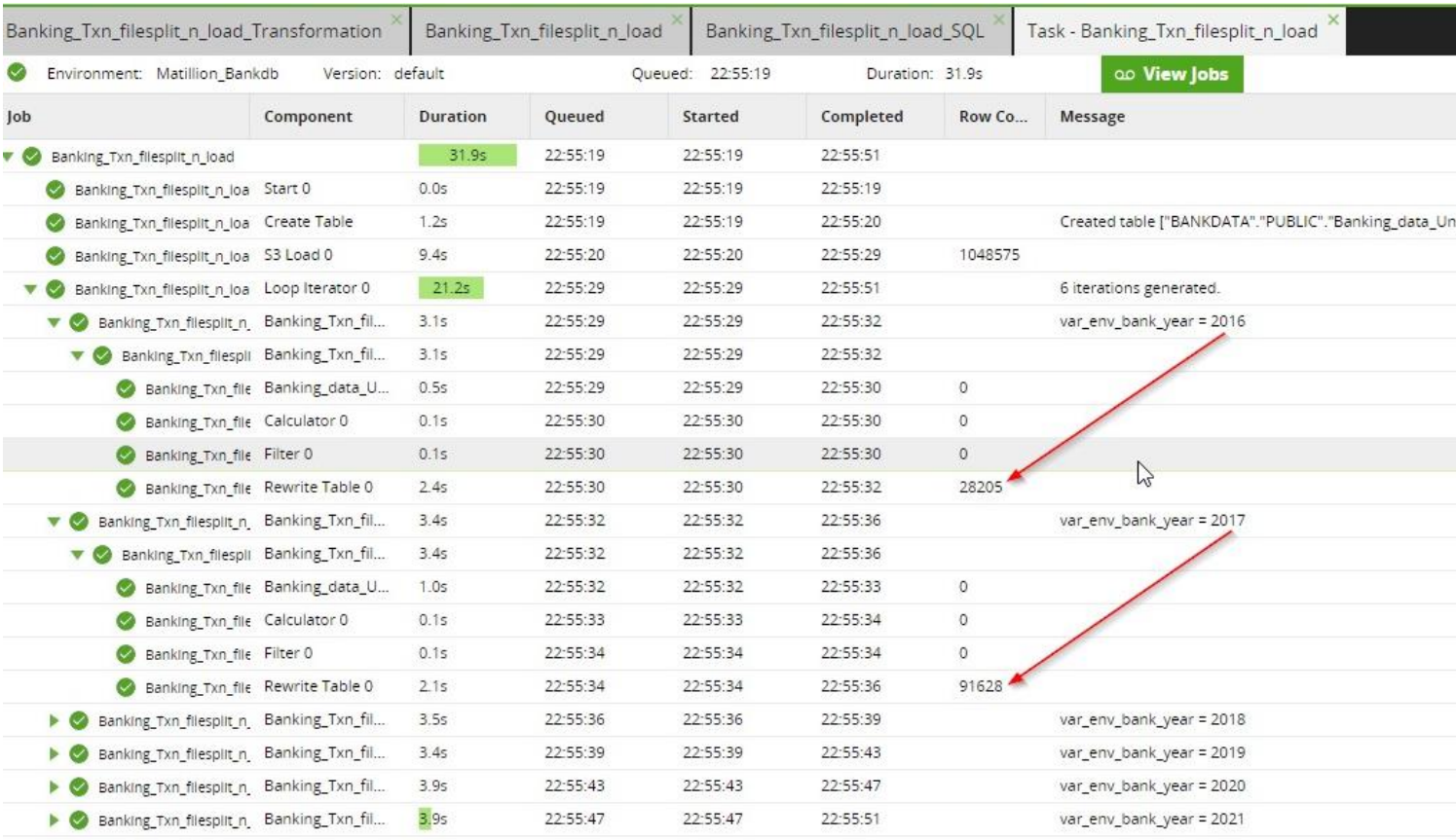
Splitting a Unified file with multiple years data and loading it in respective tables in Snowflake

- 

RESULTS:

In mattillion:

Splitting a Unified file with multiple years data and loading it in respective tables in Snowflake

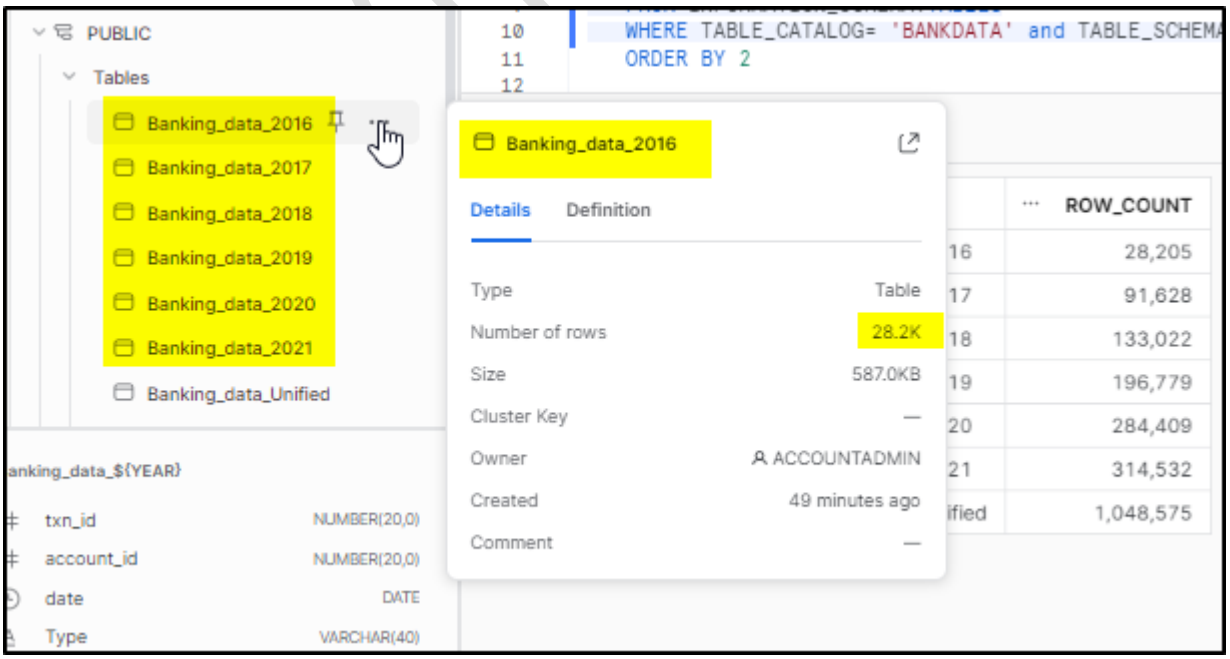


Banking_Txn_filesplit_n_load_Transformation | Banking_Txn_filesplit_n_load | Banking_Txn_filesplit_n_load_SQL | Task - Banking_Txn_filesplit_n_load

Environment: Matillion_Bankdb Version: default Queued: 22:55:19 Duration: 31.9s View Jobs

Job	Component	Duration	Queued	Started	Completed	Row Co...	Message
Banking_Txn_filesplit_n_load		31.9s	22:55:19	22:55:19	22:55:51		
Banking_Txn_filesplit_n_loa	Start 0	0.0s	22:55:19	22:55:19	22:55:19		
Banking_Txn_filesplit_n_loa	Create Table	1.2s	22:55:19	22:55:19	22:55:20		Created table ["BANKDATA"."PUBLIC"."Banking_data_Un
Banking_Txn_filesplit_n_loa	S3 Load 0	9.4s	22:55:20	22:55:20	22:55:29	1048575	
Banking_Txn_filesplit_n_loa	Loop Iterator 0	21.2s	22:55:29	22:55:29	22:55:51		6 iterations generated.
Banking_Txn_filesplit_n_	Banking_Txn_fil...	3.1s	22:55:29	22:55:29	22:55:32		var_env_bank_year = 2016
Banking_Txn_filesplit	Banking_Txn_fil...	3.1s	22:55:29	22:55:29	22:55:32		
Banking_Txn_file	Banking_data_U...	0.5s	22:55:29	22:55:29	22:55:30	0	
Banking_Txn_file	Calculator 0	0.1s	22:55:30	22:55:30	22:55:30	0	
Banking_Txn_file	Filter 0	0.1s	22:55:30	22:55:30	22:55:30	0	
Banking_Txn_file	Rewrite Table 0	2.4s	22:55:30	22:55:30	22:55:32	28205	
Banking_Txn_filesplit_n_	Banking_Txn_fil...	3.4s	22:55:32	22:55:32	22:55:36		var_env_bank_year = 2017
Banking_Txn_filesplit	Banking_Txn_fil...	3.4s	22:55:32	22:55:32	22:55:36		
Banking_Txn_file	Banking_data_U...	1.0s	22:55:32	22:55:32	22:55:33	0	
Banking_Txn_file	Calculator 0	0.1s	22:55:33	22:55:33	22:55:34	0	
Banking_Txn_file	Filter 0	0.1s	22:55:34	22:55:34	22:55:34	0	
Banking_Txn_file	Rewrite Table 0	2.1s	22:55:34	22:55:34	22:55:36	91628	
Banking_Txn_filesplit_n_	Banking_Txn_fil...	3.5s	22:55:36	22:55:36	22:55:39		var_env_bank_year = 2018
Banking_Txn_filesplit_n_	Banking_Txn_fil...	3.4s	22:55:39	22:55:39	22:55:43		var_env_bank_year = 2019
Banking_Txn_filesplit_n_	Banking_Txn_fil...	3.9s	22:55:43	22:55:43	22:55:47		var_env_bank_year = 2020
Banking_Txn_filesplit_n_	Banking_Txn_fil...	3.9s	22:55:47	22:55:47	22:55:51		var_env_bank_year = 2021

In Snowflake new tables:



Splitting a Unified file with multiple years data and loading it in respective tables in Snowflake

Total rows in each table

```
SELECT TABLE_CATALOG,TABLE_NAME,ROW_COUNT
   FROM INFORMATION_SCHEMA.TABLES
   WHERE TABLE_CATALOG= 'BANKDATA' and TABLE_SCHEMA='PUBLIC' and table_name LIKE 'Banking_data_%';
   ORDER BY 2
```

Results | Chart

	TABLE_CATALOG	TABLE_NAME	ROW_COUNT
1	BANKDATA	Banking_data_2016	28,205
2	BANKDATA	Banking_data_2017	91,628
3	BANKDATA	Banking_data_2018	133,022
4	BANKDATA	Banking_data_2019	196,779
5	BANKDATA	Banking_data_2020	284,409
6	BANKDATA	Banking_data_2021	314,532
7	BANKDATA	Banking_data_Unified	1,048,575

Cross verifying counts in the unified table:

Splitting a Unified file with multiple years data and loading it in respective tables in Snowflake

BANKDATA.PUBLIC Settings

```
SELECT YEAR("date") as YEAR, COUNT(*) FROM BANKDATA.PUBLIC."Banking_data_Unified"
GROUP BY YEAR("date")
ORDER BY 1;



SELECT TABLE_CATALOG,TABLE_NAME,ROW_COUNT
   FROM INFORMATION_SCHEMA.TABLES
   WHERE TABLE_CATALOG= 'BANKDATA' and TABLE_SCHEMA='PUBLIC' and table_name LIKE 'Ba
   ORDER BY 2
```

Results Chart

YEAR	COUNT(*)
2,016	28,205
2,017	91,628
2,018	133,022
2,019	196,779
2,020	284,409
2,021	314,532