



## USING STORED PROCEDURE CONCEPT IN MATILLION LOAD TABLE FROM SRC TO DESTINATION

Use State\_Region zip files uploaded previously for all the data load into initial source table or click below link :

[https://drive.google.com/drive/folders/1KnNJ71p2M6x6kYc\\_jAlgzSKDyWqZbjCL?usp=sharing](https://drive.google.com/drive/folders/1KnNJ71p2M6x6kYc_jAlgzSKDyWqZbjCL?usp=sharing)

**--Execute the following commands in Snowflake by creating a respective source table**

```
CREATE OR REPLACE DATABASE TEST_SRC;  
CREATE OR REPLACE SCHEMA TEST_SCHEMA;  
  
USE SCHEMA TEST_SCHEMA;  
CREATE OR REPLACE SCHEMA DEST_SCHEMA;
```

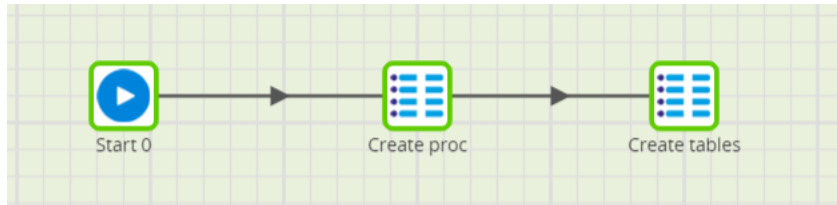
**--First Create a respective table in the desired source database & source schema**

```
CREATE OR REPLACE TABLE BROKER CLONE DEMO_DATABASE.DEMO_SCHEMA.AJ_BROKER;  
CREATE OR REPLACE TABLE CATEGORIES CLONE DEMO_DATABASE.DEMO_SCHEMA.AJ_CATAGORIES;  
CREATE OR REPLACE TABLE COMPLAIN CLONE DEMO_DATABASE.DEMO_SCHEMA.AJ_COMPLAIN;  
CREATE OR REPLACE TABLE CUSTOMER CLONE DEMO_DATABASE.DEMO_SCHEMA.AJ_CUSTOMER;  
CREATE OR REPLACE TABLE PRIORITIES CLONE DEMO_DATABASE.DEMO_SCHEMA.AJ_PRIORITIES;  
CREATE OR REPLACE TABLE PRODUCT CLONE DEMO_DATABASE.DEMO_SCHEMA.AJ_PRODUCT;  
CREATE OR REPLACE TABLE REGION CLONE DEMO_DATABASE.DEMO_SCHEMA.AJ_REGION;  
CREATE OR REPLACE TABLE SALES CLONE DEMO_DATABASE.DEMO_SCHEMA.AJ_SALES;  
CREATE OR REPLACE TABLE SOURCES CLONE DEMO_DATABASE.DEMO_SCHEMA.AJ_SOURCES;  
CREATE OR REPLACE TABLE STATE_REGION CLONE  
DEMO_DATABASE.DEMO_SCHEMA.AJ_STATE_REGION;  
CREATE OR REPLACE TABLE STATUSES CLONE DEMO_DATABASE.DEMO_SCHEMA.AJ_STATUSES;  
CREATE OR REPLACE TABLE TYPE CLONE DEMO_DATABASE.DEMO_SCHEMA.AJ_TYPE;
```

**--then using stored procedure we will create the table in destination database (respective dest schema) and load the records**

## USING STORED PROCEDURE CONCEPT IN MATILLION LOAD TABLE FROM SRC TO DESTINATION

--Copy the following Stored Procedure Code Under SQL SCRIPT COMPONENT in matillion



```

CREATE OR REPLACE PROCEDURE actiontobeperformed(src_db STRING,
                                                src_dbschema string,
                                                tablename string,
                                                tgt_db string,
                                                tgt_dbschema string,
                                                action_type string,
                                                table_view char(1))

return string

language javascript

execute as caller

as

$$
  
```

//VARIABLE DECLARATION

```

var action_type = ACTION_TYPE;
var src_db = SRC_DB;
var src_dbschema = SRC_DBSHEMA;
var tablename = TABLENAME;
var tgt_db = TGT_DB;
var tgt_dbschema = TGT_DBSHEMA;
var table_view = TABLE_VIEW;
var result = "";
  
```



## USING STORED PROCEDURE CONCEPT IN MATILLION LOAD TABLE FROM SRC TO DESTINATION

```
var sql_query_text = "";

try
{
    //CREATE TABLES
    if (action_type.toUpperCase() == 'CREATE' && table_view.toUpperCase() == 'T')
    {
        var sql_query_text = 'CREATE OR REPLACE TABLE ' + tgt_db + '.' + tgt_dbschema + '.' +
        tablename + ' like ' + src_db +
        '.' + src_dbschema + '.' + tablename + ';';
        snowflake.createStatement({sqlText:sql_query_text}).execute();
        result = "Table created successfully: " + tgt_db + '.' + tgt_dbschema + '.' + tablename
    }

    //INSERT RECORDS
    else if (action_type.toUpperCase() == 'INSERT' && table_view.toUpperCase() == 'I')
    {
        var sql_query_text = 'INSERT INTO ' + tgt_db + '.' + tgt_dbschema + '.' + tablename + '
select * from ' + src_db +
        '.' + src_dbschema + '.' + tablename + ';';
        snowflake.createStatement({sqlText:sql_query_text}).execute();
        result = "Insertion successfully completed: " + tgt_db + '.' + tgt_dbschema + '.' + tablename
    }

    else
    {
        result = "Fail: Please give proper parameter \n" + "1.If INSERTing the data then parameter
should be only Table(t/T)\n"
    }
}
```



Cancel

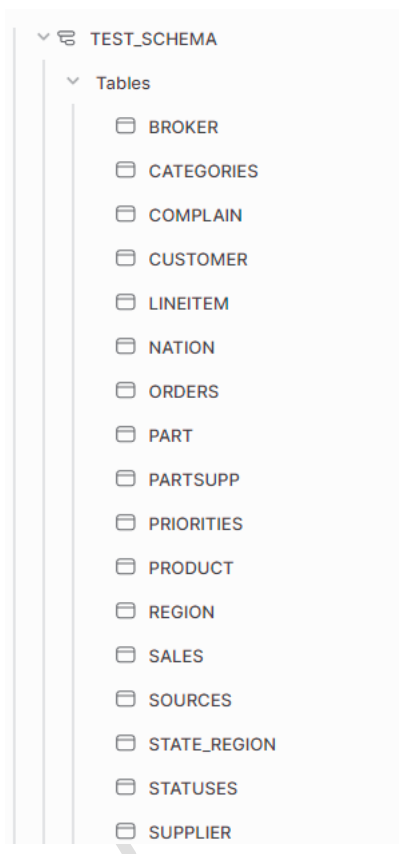


## USING STORED PROCEDURE CONCEPT IN MATILLION LOAD TABLE FROM SRC TO DESTINATION

--Then drag and place SQL SCRIPT component again where we will be calling stored procedure to create table first and then will change the script and instead of CREATE will replace INSERT along with alias as I as shown below

```
1 CALL actiontobeperformed('TEST_SRC','TEST_SCHEMA','CUSTOMER','TEST_DST','DEST_SCHEMA','CREATE','T');
2 CALL actiontobeperformed('TEST_SRC','TEST_SCHEMA','LINEITEM','TEST_DST','DEST_SCHEMA','CREATE','T');
3 CALL actiontobeperformed('TEST_SRC','TEST_SCHEMA','REGION','TEST_DST','DEST_SCHEMA','CREATE','T');
4 CALL actiontobeperformed('TEST_SRC','TEST_SCHEMA','NATION','TEST_DST','DEST_SCHEMA','CREATE','T');
5 CALL actiontobeperformed('TEST_SRC','TEST_SCHEMA','SUPPLIER','TEST_DST','DEST_SCHEMA','CREATE','T');
6 CALL actiontobeperformed('TEST_SRC','TEST_SCHEMA','PARTSUPP','TEST_DST','DEST_SCHEMA','CREATE','T');
7 CALL actiontobeperformed('TEST_SRC','TEST_SCHEMA','PART','TEST_DST','DEST_SCHEMA','CREATE','T');
8 CALL actiontobeperformed('TEST_SRC','TEST_SCHEMA','ORDERS','TEST_DST','DEST_SCHEMA','CREATE','T');
9
```

Once the table gets created in the destination database (resp schema) as shown below



Its time to load the records now by executing STORED PROCEDURE CALL by changing CREATE to INSERT with alias I



**USING STORED PROCEDURE CONCEPT IN MATILLION  
LOAD TABLE FROM SRC TO DESTINATION**

Variables

Name	Default value
------	---------------

Manage Variables

```
1 CALL actiontobeperformed('TEST_SRC','TEST_SCHEMA','CUSTOMER','TEST_DST','DEST_SCHEMA','INSERT','I');
2 CALL actiontobeperformed('TEST_SRC','TEST_SCHEMA','LINEITEM','TEST_DST','DEST_SCHEMA','INSERT','I');
3 CALL actiontobeperformed('TEST_SRC','TEST_SCHEMA','REGION','TEST_DST','DEST_SCHEMA','INSERT','I');
4 CALL actiontobeperformed('TEST_SRC','TEST_SCHEMA','NATION','TEST_DST','DEST_SCHEMA','INSERT','I');
5 CALL actiontobeperformed('TEST_SRC','TEST_SCHEMA','SUPPLIER','TEST_DST','DEST_SCHEMA','CREATE','I');
6 CALL actiontobeperformed('TEST_SRC','TEST_SCHEMA','PARTSUPP','TEST_DST','DEST_SCHEMA','CREATE','I');
7 CALL actiontobeperformed('TEST_SRC','TEST_SCHEMA','PART','TEST_DST','DEST_SCHEMA','CREATE','I');
8 CALL actiontobeperformed('TEST_SRC','TEST_SCHEMA','ORDERS','TEST_DST','DEST_SCHEMA','CREATE','I');
9
```

Run

Press "Run" to test script execution.

Update Component

OK

Cancel

Variables

Name	Default value
------	---------------

Manage Variables

```
1 CALL actiontobeperformed('TEST_SRC','TEST_SCHEMA','CUSTOMER','TEST_DST','DEST_SCHEMA','INSERT','I');
2 CALL actiontobeperformed('TEST_SRC','TEST_SCHEMA','LINEITEM','TEST_DST','DEST_SCHEMA','INSERT','I');
3 CALL actiontobeperformed('TEST_SRC','TEST_SCHEMA','REGION','TEST_DST','DEST_SCHEMA','INSERT','I');
4 CALL actiontobeperformed('TEST_SRC','TEST_SCHEMA','NATION','TEST_DST','DEST_SCHEMA','INSERT','I');
5 CALL actiontobeperformed('TEST_SRC','TEST_SCHEMA','SUPPLIER','TEST_DST','DEST_SCHEMA','INSERT','I');
6 CALL actiontobeperformed('TEST_SRC','TEST_SCHEMA','PARTSUPP','TEST_DST','DEST_SCHEMA','INSERT','I');
7 CALL actiontobeperformed('TEST_SRC','TEST_SCHEMA','PART','TEST_DST','DEST_SCHEMA','INSERT','I');
8 CALL actiontobeperformed('TEST_SRC','TEST_SCHEMA','ORDERS','TEST_DST','DEST_SCHEMA','INSERT','I');
9
```

Run

Press "Run" to test script execution.

Update Component

OK

Cancel

Job

03:0

03:0

03:0

03:0

02:5

02:5

**Finally Run the Jobs and we will be able to see the records in the destination**

▼	✓	load tables using proc		6.5s	02:54:10	02:54:10	02:54:17	02:54:17		
	✓	load tables using proc	Start 0	0.0s	02:54:10	02:54:10	02:54:10			
	✓	load tables using proc	Create proc	1.5s	02:54:10	02:54:10	02:54:12	0		Successfully executed [2] queries.
	✓	load tables using proc	Create tables	5.0s	02:54:12	02:54:12	02:54:17	0		Successfully executed [8] queries.



## USING STORED PROCEDURE CONCEPT IN MATILLION LOAD TABLE FROM SRC TO DESTINATION

practice\_matillion

load tables using proc

Task - load tables using proc

```
graph LR; Start0[Start 0] --> CreateProc[Create proc]; CreateProc --> CreateTable[Create tables];
```

Components

SQL Script

Name	Value	Status
Name	Create tables	OK
SQL Script	CALL actiontobeperf...	OK

Task	Envir...	Version	Job	Queu...	Job C...	Sched...
Validate	testin...	default	load t...	03:26...	03:26...	03:26...
Run	testin...	default	load t...	03:07...	03:08...	03:08...
Validate	testin...	default	load t...	03:07...	03:07...	03:07...
Sample	testin...	default	load t...	03:07...	03:07...	03:07...

\*\*\*\*\*HAPPY LEARNING\*\*\*\*\*