

ELT PROCESS

with Snowflake Stored
Procedure and Task



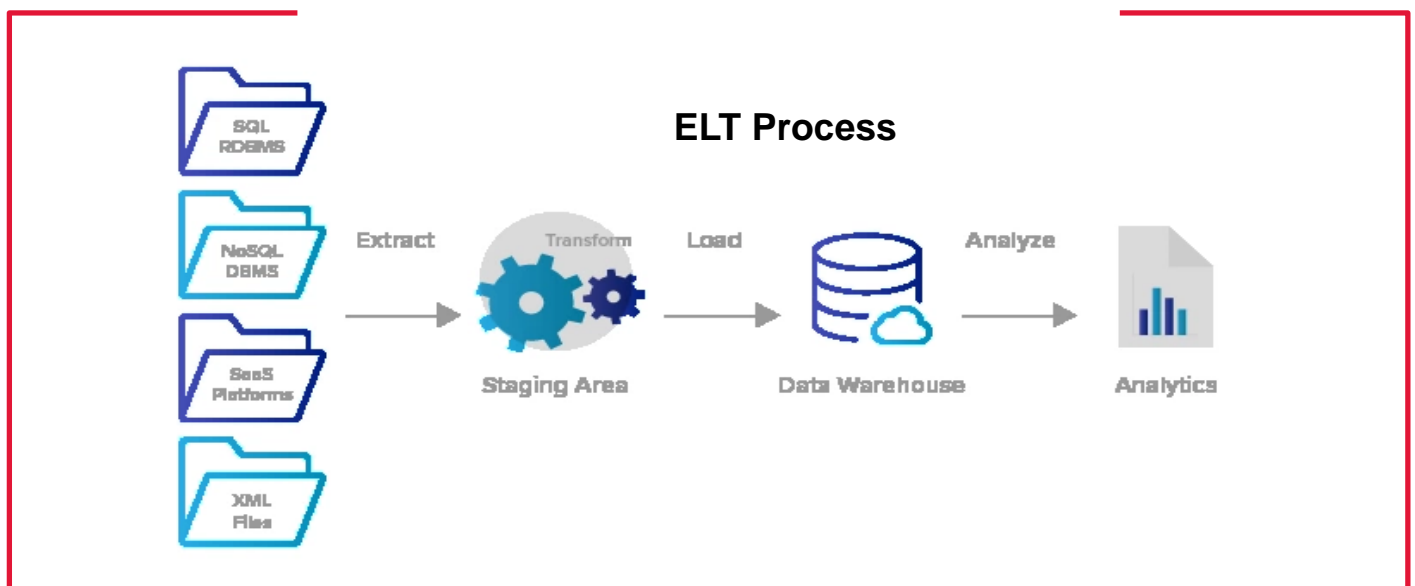
ELT

ELT is the process of extracting raw data from one or multiple sources and loading into the target table/data warehouse. On a high-level ELT has the following steps:

- Step I: Raw data will be extracted from various source systems.
- Step II: Raw data will be loaded into target tables/data warehouse without any transformation.
- Step III: Transformation will be applied to the target system to perform the data transformation.

ELT Highlights

- Cloud-based data warehouses offer unlimited storage as well as compute processing power. This make ELT more viable on cloud-based servers.
- In the ELT world, it transforms only the data required for business decision.
- ELT allows you to load all forms of data immediately once it is available.



This article attempts to discuss how one can perform data integration from snowflake staging area to final table, using snowflake stored procedure and schedule it using "Task".

Stored Procedure

A stored procedure is useful to perform one or more SQL, data transform and data validation. A stored procedure may contain one or many statements and even call additional stored procedures, passing parameters through as required.

Currently following UDF's are supported in snowflake:

- SQL
- Javascript

SQL UDF:

- A SQL UDF evaluates a random SQL expression and returns the result in the form of tabular or scalar format.

Javascript:

- A JavaScript UDF lets one use the JavaScript programming language to manipulate data and return either scalar or tabular results

ELT Process

At the end of this article, one would have preliminary information on,

- How to create a stored procedure in snowflake
- How to call one stored procedure from another procedure
- Variable concatenation / binding

Snowflake stored procedure will read the metadata table, execute respective SQL and return the status:

| ELT_SCHEMA_DETAILS | | | |
|--------------------|------------|------------|---------------|
| Column Name | Data Type | Constraint | Default Value |
| ELT_ID | NUMBER(10) | NOT NULL | NULL |
| ETL_SQL | VARCHAR | NOT NULL | NULL |

Note: Loading data from the stage table to the target table can be done using snowflake **Merge** statement. In order to process the data from the stage table to the target table using a merge statement, we need the **primary key**.

Snowflake merge syntax

```
MERGE
INTO <target_table>
using <source>
ON <join_expr> { matchedClause | notMatchedClause } [ ... ]
```

Snowflake Stored Procedure

```
-- Stored Procedure to generate merge statement
-- Proc Name : ELT_LOAD()
-- Input Parameters
--     A. ELT_ID
-- Author : Sriganesh Palani
-- Reachout @ sriganesh.palani@gmail.com
CREATE OR REPLACE PROCEDURE ELT_LOAD(elt_id float)
RETURNS TEXT NOT NULL
LANGUAGE JAVASCRIPT
COMMENT = 'SP reads schema table and '
EXECUTE AS CALLER
AS
$$
var sel_sql = "SELECT elt_sql FROM ELT_SCHEMA_DETAILS WHERE ELT_ID = "+ ELT_ID;
var sel_stmet = snowflake.createStatement({sqlText: sel_sql});
try{
    var sel_res = sel_stmet.execute();
    while (sel_res.next()){
        merge_sql = sel_res.getColumnValue(1);
        var merge_stmt = snowflake.createStatement({sqlText: merge_sql});

        try{
            var result = merge_stmt.execute();
            return "ELT Success";
        }
        catch(err){
            return 'Error in merge:' + err;
        }
    }
}
catch(err){
    return 'Error reading Schema table :' + err;
}
$$;
```

Calling Stored Procedure

```
104 call ELT_LOAD(1);
```

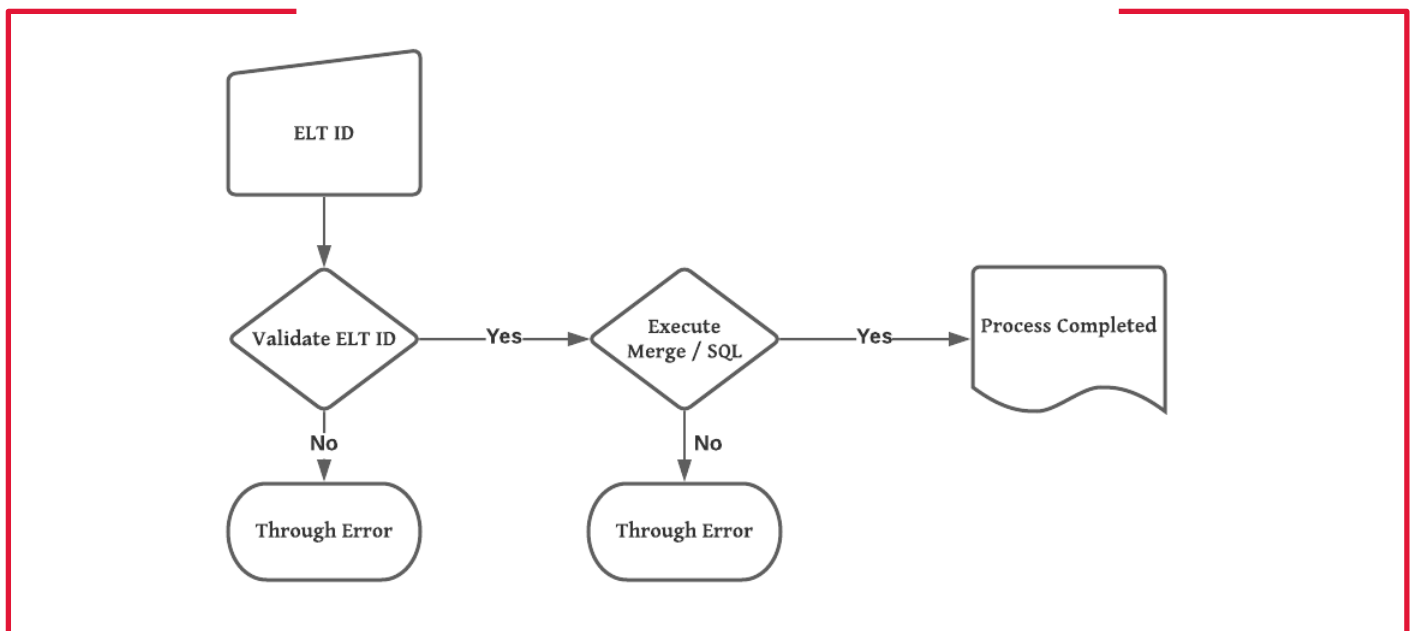
Results Data Preview

✓ Query ID SQL 9.47s 1 rows

Filter result... [Download] Copy

| Row | ELT_LOAD |
|-----|-------------|
| 1 | ELT Success |

How the procedure works



- Main stored procedure accepts the following input parameter:
 - ELT_ID
- Procedure reads respective SQL associate with ELT_ID from snowflake metadata table.
- Execute the merge / SQL statement to catch an error, if any.

Scheduling stored procedure using Task

Snowflake task allows to schedule a SQL script or stored procedure on snowflake instance. Snowflake support CRON based job schedule. CRON is the Linux version of windows task schedule, and it has a simpler mechanism regards to run a job.

| Min | Hour | Day | Month | Week Day | Description |
|-----|------|-----|-------|----------|-----------------------------|
| * | * | * | * | * | Run every min, hour and day |
| 10 | 16 | * | * | Mon | Every Mon day 4:10 PM |
| 10 | * | * | * | * | Every 10 Min |

Create Task

Following task will run the stored procedure every minute.

```
-- Task Creation: ACCOUNTADMIN PRIVILEGES
CREATE OR REPLACE TASK task_min
  WAREHOUSE = COMPUTE_WH
  SCHEDULE = '1 minute'
AS
  CALL ELT_LOAD(1);
```

Assign privileges to named role

```
-- Assign privilege to sysadmin role:
grant all privileges on task task_10min to role SYSADMIN;
```

Validate the status of task

```
-- Check status of task:
SHOW TASKS;
```

Alter the status of task

```
-- Alter the task status to resume
ALTER TASK IF EXISTS TASK_10MIN resume;
```

Task execution history

```
-- Monitoring task execution history:
select
  *
from
  table(information_schema.task_history())
order by
  scheduled_time;
```



**Tech
Mahindra**



www.youtube.com/user/techmahindra09
www.facebook.com/techmahindra
www.twitter.com/tech_mahindra
www.linkedin.com/company/tech-mahindra
www.techmahindra.com

Copyright © Tech Mahindra 2021. All Rights Reserved.
Disclaimer. Brand names, logos and trademarks used herein remain the property of their respective owners.