# Teradata to Snowflake Migration Procedure

## Overview

The P\_MIGRATE\_TERADATA\_TO\_SNOWFLAKE is a comprehensive stored procedure that automates the DMVA (Data Migration and Validation Automation) process for migrating tables from Teradata to Snowflake. This procedure implements the logic from sampleCode.sql with enhanced parameterization and conditional execution flows.

## Procedure Signature

CREATE OR REPLACE PROCEDURE NPD\_D12\_DMN\_GDWMIG.MIGRATION\_TRACKING\_V2.P\_MIGRATE\_TERADATA\_TO\_SNOWFLAKE(  
 "P\_SOURCE\_DATABASE\_NAME" VARCHAR, -- Source Teradata database/schema name  
 "P\_SOURCE\_TABLE\_NAME" VARCHAR, -- Source table name  
 "P\_TARGET\_DATABASE\_NAME" VARCHAR, -- Target Snowflake database name  
 "P\_TARGET\_SCHEMA\_NAME" VARCHAR, -- Target Snowflake schema name  
 "P\_TARGET\_TABLE\_NAME" VARCHAR, -- Target table name  
 "P\_WITH\_CHUNKING\_YN" VARCHAR DEFAULT 'N', -- Enable chunking (Y/N)  
 "P\_CHUNKING\_COLUMN" VARCHAR DEFAULT null, -- Column for chunking  
 "P\_CHUNKING\_VALUE" VARCHAR DEFAULT null, -- Chunking value  
 "P\_CHUNKING\_DATA\_TYPE" VARCHAR DEFAULT null, -- Data type for chunking  
 "P\_SKIP\_WAITING\_FOR\_MIGRATION\_TASKS" VARCHAR DEFAULT 'N', -- Skip task monitoring (Y/N)  
 "P\_LOAD\_TYPE" VARCHAR DEFAULT 'FULL', -- Load type: FULL or INCREMENTAL  
 "P\_REFRESH\_STRUCTURES\_YN" VARCHAR DEFAULT 'Y' -- Refresh structures (Y/N)  
)  
RETURNS VARCHAR

## New Parameters

### P\_LOAD\_TYPE (Default: ‘FULL’)

* **‘FULL’**: Performs complete data refresh, including cleanup of existing data
* **‘INCREMENTAL’**: Skips Step 4 (data cleanup), preserving existing data

### P\_REFRESH\_STRUCTURES\_YN (Default: ‘Y’)

* **‘Y’**: Executes all steps (1-6) including structure setup and data migration
* **‘N’**: Skips to Step 5 only (data migration), assumes structures are already configured

## Execution Flow

### Normal Flow (P\_REFRESH\_STRUCTURES\_YN = ‘Y’)

1. **Step 1: Configure Mapping Rules**
   * Inserts mapping rules into dmva\_mapping\_rules table
   * Checks for existing mappings to avoid duplicates
2. **Step 2: Get Metadata**
   * Calls DMVA\_GET\_METADATA\_TASKS to populate object and column metadata
   * Monitors metadata tasks until completion
   * Validates metadata was successfully created
3. **Step 3: Create Sync Tables**
   * Calls dmva\_create\_sync\_tables to create target structures and mappings
   * Verifies sync table creation
4. **Step 4: Fresh Migration Cleanup (Conditional)**
   * **If P\_LOAD\_TYPE = ‘FULL’**:
     + Deletes existing checksums
     + Clears target table data
   * **If P\_LOAD\_TYPE = ‘INCREMENTAL’**: Skips cleanup

### Data Migration Flow (Steps 5.1-5.3)

The following steps execute regardless of the P\_REFRESH\_STRUCTURES\_YN setting:

1. **Step 5.1: Configure Chunking (Optional)**
   * Configures chunking if P\_WITH\_CHUNKING\_YN = 'Y'
   * Validates chunking parameters
2. **Step 5.2: Data Migration**
   * Calls DMVA\_GET\_CHECKSUM\_TASKS to migrate data
3. **Step 5.3: Monitor Migration Tasks**
   * Monitors migration progress (unless skipped)
   * Provides final verification count

### Data-Only Flow (P\_REFRESH\_STRUCTURES\_YN = ‘N’)

When P\_REFRESH\_STRUCTURES\_YN = 'N', the procedure skips structure setup (Steps 1-4) and executes only the data migration flow (Steps 5.1-5.3), assuming all structures and mappings are already configured.

## Usage Examples

### Example 1: Full Migration with Structure Setup

CALL NPD\_D12\_DMN\_GDWMIG.MIGRATION\_TRACKING\_V2.P\_MIGRATE\_TERADATA\_TO\_SNOWFLAKE(  
 'K\_PDDSTG', -- Source database/schema  
 'DERV\_ACCT\_PATY\_CHG', -- Source table  
 'NPD\_D12\_DMN\_GDWMIG', -- Target database  
 'TMP', -- Target schema  
 'PDDSTG\_DERV\_ACCT\_PATY\_CHG\_20250903', -- Target table  
 'N', -- No chunking  
 null, null, null, -- Chunking parameters (not used)  
 'N', -- Wait for migration tasks  
 'FULL', -- Full load  
 'Y' -- Refresh structures  
);

### Example 2: Incremental Load (No Data Cleanup)

CALL NPD\_D12\_DMN\_GDWMIG.MIGRATION\_TRACKING\_V2.P\_MIGRATE\_TERADATA\_TO\_SNOWFLAKE(  
 'K\_PDDSTG', 'DERV\_ACCT\_PATY\_CHG', 'NPD\_D12\_DMN\_GDWMIG', 'TMP',   
 'PDDSTG\_DERV\_ACCT\_PATY\_CHG\_20250903',  
 'N', null, null, null, 'N',  
 'INCREMENTAL', -- Incremental load (skips cleanup)  
 'Y'  
);

### Example 3: Data-Only Migration (Structures Already Exist)

CALL NPD\_D12\_DMN\_GDWMIG.MIGRATION\_TRACKING\_V2.P\_MIGRATE\_TERADATA\_TO\_SNOWFLAKE(  
 'K\_PDDSTG', 'DERV\_ACCT\_PATY\_CHG', 'NPD\_D12\_DMN\_GDWMIG', 'TMP',   
 'PDDSTG\_DERV\_ACCT\_PATY\_CHG\_20250903',  
 'N', null, null, null, 'N',  
 'FULL',  
 'N' -- Skip structure refresh (Step 5 only)  
);

### Example 4: With Chunking

CALL NPD\_D12\_DMN\_GDWMIG.MIGRATION\_TRACKING\_V2.P\_MIGRATE\_TERADATA\_TO\_SNOWFLAKE(  
 'K\_PDDSTG', 'LARGE\_TABLE', 'NPD\_D12\_DMN\_GDWMIG', 'TMP',   
 'MIGRATED\_LARGE\_TABLE',  
 'Y', -- Enable chunking  
 'DATE\_COLUMN', -- Chunking column  
 '2024-01-01', -- Chunking value  
 'DATE', -- Chunking data type  
 'N', 'FULL', 'Y'  
);

## Return Value

The procedure returns a detailed log string containing: - Timestamp for each step - Parameter values and validation results - Progress updates for metadata and migration tasks - Error messages (if any) - Final verification counts

## Error Handling

The procedure includes comprehensive error handling: - Parameter validation with descriptive error messages - Metadata validation to ensure prerequisites are met - Exception handling with detailed error reporting - Graceful handling of missing or invalid configurations

## Dependencies

* DMVA\_GET\_METADATA\_TASKS procedure
* dmva\_create\_sync\_tables procedure
* DMVA\_GET\_CHECKSUM\_TASKS procedure
* P\_SET\_CHUNKING\_KEY procedure (for chunking functionality)
* DMVA metadata tables: dmva\_mapping\_rules, dmva\_object\_info, dmva\_tasks, etc.

## Monitoring

The procedure provides real-time monitoring of: - Metadata task completion - Migration task progress - Row counts before and after migration - Task execution times and loop counters