ETL Validation Framework: Teradata to Snowflake using Python

Document generated on: 2025-05-05 04:34:20

# Overview

This document outlines a Python-based ETL validation framework designed to verify data integrity during migration from Teradata to Snowflake. It includes row count checks, null checks, and duplicate checks, along with logging for audit and automation.

# Components

- Python with Pandas, teradatasql, and snowflake-connector-python  
- JSON-based config for database credentials  
- Log file for validation results  
- Modular structure for extensibility

# Validation Logic

1. Row Count Validation: Ensures source and target tables have the same number of rows.

2. Null Check Validation: Ensures specified columns have no NULL values.

3. Duplicate Check Validation: Ensures key columns are unique.

# Sample Script

import json  
from datetime import datetime  
import teradatasql  
import snowflake.connector  
  
def log\_to\_file(message):  
 with open("logs/validation.log", "a") as f:  
 f.write(f"[{datetime.now()}] {message}\n")  
  
def validate\_row\_count(td\_conn, sf\_conn):  
 with td\_conn.cursor() as cur:  
 cur.execute("SELECT COUNT(\*) FROM source.customer")  
 td\_count = cur.fetchone()[0]  
 with sf\_conn.cursor() as cur:  
 cur.execute("SELECT COUNT(\*) FROM customer")  
 sf\_count = cur.fetchone()[0]  
 assert td\_count == sf\_count, f"Row count mismatch: Teradata({td\_count}) vs Snowflake({sf\_count})"  
  
def validate\_null\_check(sf\_conn, table, columns):  
 with sf\_conn.cursor() as cur:  
 for col in columns:  
 cur.execute(f"SELECT COUNT(\*) FROM {table} WHERE {col} IS NULL")  
 null\_count = cur.fetchone()[0]  
 assert null\_count == 0, f"Nulls found in column {col}"  
  
def validate\_duplicate\_check(sf\_conn, table, keys):  
 key\_expr = ", ".join(keys)  
 query = f"SELECT {key\_expr}, COUNT(\*) FROM {table} GROUP BY {key\_expr} HAVING COUNT(\*) > 1"  
 with sf\_conn.cursor() as cur:  
 cur.execute(query)  
 rows = cur.fetchall()  
 assert not rows, f"Duplicates found on keys {keys}: {rows}"  
  
def run\_validations():  
 with open("config/connections.json") as f:  
 config = json.load(f)  
 td\_conn = teradatasql.connect(\*\*config['teradata'])  
 sf\_conn = snowflake.connector.connect(\*\*config['snowflake'])  
 try:  
 validate\_row\_count(td\_conn, sf\_conn)  
 validate\_null\_check(sf\_conn, "customer", ["email"])  
 validate\_duplicate\_check(sf\_conn, "customer", ["customer\_id"])  
 log\_to\_file("✅ All validations passed")  
 except Exception as e:  
 log\_to\_file(f"❌ Validation failed: {str(e)}")  
 raise  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 run\_validations()