TAP - ETL Pipeline

Usman Malik
Dated: 2025-01-12

Table of contents

Environment Setup

Following tools were allowed and utilized to complete the assignment

- dockerized mongo
- dockerized postgres
- dockerized airflow
- powerBI

Assumptions and Decisions

- awaiting decision if duplicate record is to be considered "duplicate_data" OR "quantity of same product sold"
    - I have assumed it to be duplicate data
- since we are going with hr/hour granularity, we can compute the date_key from actual date, we will do that during loading process, eliminating the need of roundtrips to DB for insertion/retrieval

Limitations

- using airflow CLI ONLY, as my GUI was not working (the reason why I did not used MongoHook, PostgresHook)
- we will allow the DAG/tasks to stop with an exception and reply on the user to view the DAG logs, fix the issue and perform data cleaning before manually running the DAG

DataSet

- Dataset seems to contain incorrect/irrelevant data with date '1970-01-01 00:33:40', will be **ignoring** these records
- Dataset is missing sales for some hours a day BUT we will generate the date_dimension in **sequence** using generators available in postgres
- Since the business requirements don't mention reporting "on per user" basis, I will **not**
    - Create the user_dimension
    - Use user_id in any report generation process
- As per data analysis, product_id can uniquely identify a product and is the natural key for products dimension

Processing Sequence

- Dump all [recent] data from mongo.sales to postgres.raw_sales (identify recent data by maintaining a meta_table)
- During Transform, encrypt product_id and use it as Identity
- Remove duplicates from raw_sales[id>last_processed_id] and dump to temporary table postgres.non_duplicate_sales
- Update dimension tables
- Update fact table
- Generate reports

Optimizations

- Rather than creating an autoincrement ID for date_dimension, I will be using yyyymmddHH extracted from event_time as KEY to save on processing
- I will be generating hash for unique products and saving it in [character varying(50)]. The same hash will be the primary key of the products dimension table. character length 50 can accommodate product_ids upto 10^100 OR 1e100 (A googol)
- I can use the same hashing mechanism for user_id, category_id BUT since business requirements don't mention reporting on these facts, I will safely ignore them