Quick notes

Sunday, April 20, 2025 10:37 PM

GCP Data Engineering project - Retailer Domain

retailer - healthcare - production - banking - Engineering

any project you as a data Engineer:
- data ingestion
- data storage and management
- data processing
- data analysis
- data governance
- data orchestration
- data visualization

domain ==> retailer projects

retailer ==> amazon, nike, flipkart, adidas, target .....

Retailer project :
-- outsource their application ==> third party ==> paypal, baazarvoice, stripe...
-- understand more about their business
- customer behaviour
- sales trend
- inventory analysis

-- focus on the data ==> apps

-- data lake : ingestion, store, process, transform, analyse, visualize, automate

data sources :
- databases: mysql, sql server, postgre, cloud dbs...
- files : csv, json, avro, parquet...
- API

==========
create data sources:
1. retailer database - mysql - done
- products(full)
- categories(full)
- customers(incr)
- orders(incr)
- order_items(incr)
2. supplier database - mysql - done
- supplier(full)
- product_supplier(incr)

3. customer review api - done
    - api

gcs setup:
- create bucket
- create folders:
    1. configs:
        - metadata driven approach
        - retailer_config.csv
        - supplier_config.csv
    2. landing
        - ingestion
            - retailer
            - supplier
            - api
    3. temp
        - pipeline_logs

data ingestion:
- create dataproc cluster
- ingestion-1: mysql-retailer-dbs==>config,audit_tbl,archive,logs==>gcs_bucket/landing/retailer-db/*
    1. read config file for metadata
    2. archive the existing files in respective retailer-db folder in gcs
    3. extract the from mysql-retailer-db for respective table and load json to gcs landing
        - incr ==> get latest timestamp (audit table in bq) ==> compare with table watermark col ==> delta
        - full ==> total data from table will be fully loaded to gcs location
    4. store the logs to gcs and bigquery for future purposes

- ingestion-2: mysql-supplier-dbs==>config,audit_tbl,archive,logs==>gcs_bucket/landing/supplier-db/*
    1. read config file for metadata
    2. archive the existing files in respective supplier-db folder in gcs
    3. extract the from mysql-supplier-db for respective table and load json to gcs landing
        - incr ==> get latest timestamp (audit table in bq) ==> compare with table watermark col ==> delta
        - full ==> total data from table will be fully loaded to gcs location
    4. store the logs to gcs and bigquery for future purposes

- ingestion-3: review-api ===> gcs_bucket/landing/customer-reviews/*
    - fetch the data from api
    - convert api data to pd df
    - storing this df data locally
    - from local, writing to gcs landing

bronze:
- gcs_bucket/landing/retailer-db/* ===> external tables ===> bronze_dataset/5
- gcs_bucket/landing/supplier-db/* ===> external tables ===> bronze_dataset/2
- gcs_bucket/landing/customer-reviews/* ===> external tables ===> bronze_dataset/1

silver:
- bronze_dataset/* ==> nulls, duplicates, incr(scd2), full(truncate&load) ==> silver_dataset/*

gold:

- this is the main reason that so far we have done above steps
- what is that mean ?? valuable insights you are going to find in gold layer only
- example:
    - sales summary
    - customer engagement metrics
    - product performance
    - supplier performance
    - customer review summary
- silver_dataset/* ===> gold_dataset/{insight_tables} ====> BI (reports & dashboards)

workflow orchestration:
- create composer environment (20min+)
- create dags
    - pyspark_dag
    - bq_dag
    - parent_dag

CICD:
- setup composer environment
- setup github
- setup cloud build trigger
    ==> to run the cloudbuild.yaml as soon as there are changes in repository
    ==> all the dags and data gets updated in the composer bucket
    ==> test your cicd

-- read from config file ???

- what tables are scope for ingestion
- load type
  - full
  - incr (watermark)
- target path

pipeline - scheduled - at 5 am

archive_logic:
- gcs_bucket
  - landing
    - retailer-db
      - archive
        - 2025
          - 03
            - 29
              - products
                - products_29032025.json
              - customers
                - customers_29032025.json
      - products
      - customers

-- audit table: