Capstone 3 – Building Data Pipeline

Rizky Fajar Aditya – JCDEOL001

December 2024

Executive Summary:

Project Highlights

As a growing biodiesel company, B40 Incorporation faces challenges in Data Access and Quality, hindering their decision-making and business agility – this urged them to create a robust data pipeline.

- The time to required to access critical data increases by 4% each month, which delays their ability to respond to market changes
- Inconsistent data quality makes it challenging to perform comprehensive data analysis

SDLC (Software Development Life Cycle) is applied to build a robust data pipeline. Tools like Airflow, PostgreSQL, BigQuery, Docker, Looker, and Python were utilized for end-to-end data pipeline

The result of this project were promising:

- Successfully generate data, load data, and build relationship between tables in PostgresSQL– all orchestrated using Airflow
- Successfully fetched data from PostgreSQL, incrementally load to staging stables in BigQuery, create sales_dashboard table as source for dashboard
- Created dashboard using Google Looker to generate business insights and informations: reducing reporting time through automation, increase data quality, and enhancing decision-making capabilities

Table of Contents:

Study Case: B40 Inc. faces growing challenges in Data Access and Quality, hindering decision-making and business agility.

Background and Study Case:

B40 Inc is a major energy company in Indonesia that distributes Biodiesel product to Mining Companies in every corner in Indonesia.

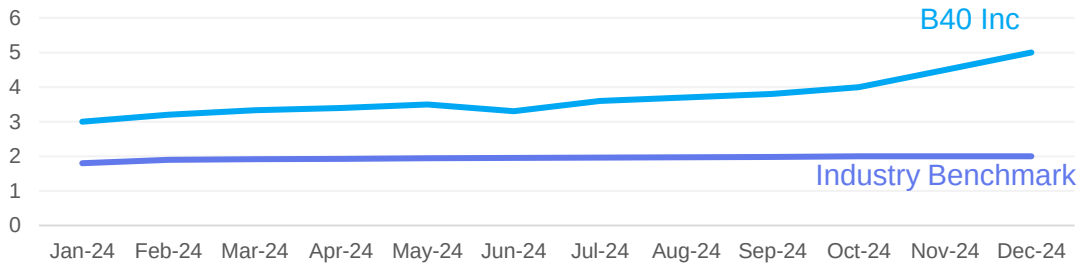
The Marketing and Sales Team at B40 Inc. is facing challenges in gathering data for their reports and insights, which is impacting their decision-making processes. To address this, they have enlisted the help of Data Engineering team to **design and implement a robust data pipeline** that will streamline data collection and processing, enabling faster reporting and more informed decision-making.

Please note: The case study and data presented in this slide are hypothetical and created for illustrative purposes only

Several challenges that makes B40 Inc is on the verge to develop a robust data pipeline:

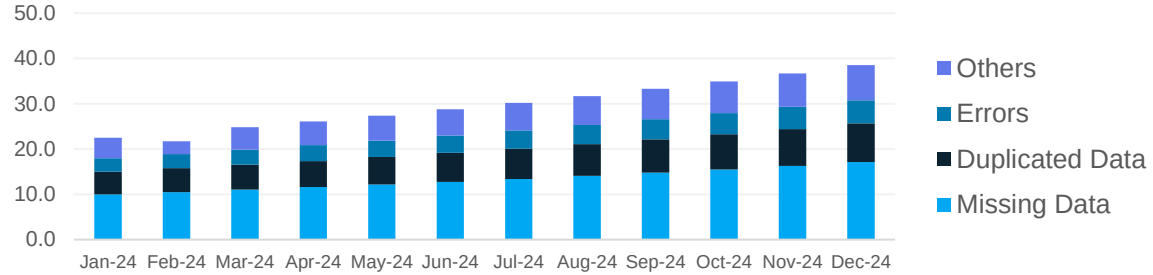
1. The time required to access critical data increases by 4% each month, as the duration from data collection to reporting grows, which delays their ability to respond to market changes

Time needed teed to collect data to reporting, in hours

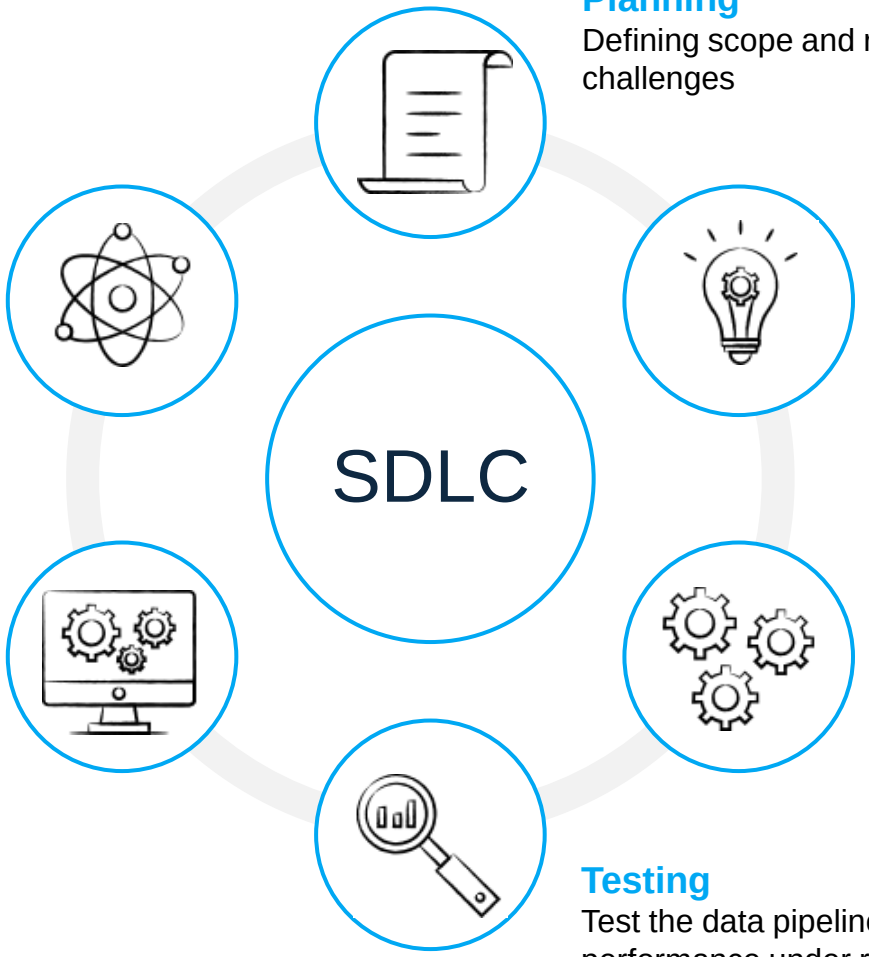


2. Inconsistent data quality makes it challenging to perform comprehensive data analysis

Types of data qualities issues, in #



To address the issue, The SDLC cycle is applied to develop a robust data pipeline tailored to meet B40 Inc.'s needs.



Planning
Defining scope and requirements of data pipeline to address current challenges

Design
Design the data pipeline architecture and outline the steps to ensure scalability, data quality, and efficient reporting

Development
Building data pipeline based on design specifications

Testing
Test the data pipeline for accuracy, reliability, and performance under real-world conditions

Deployment
Deploy the data pipeline into a production environment for ongoing use.

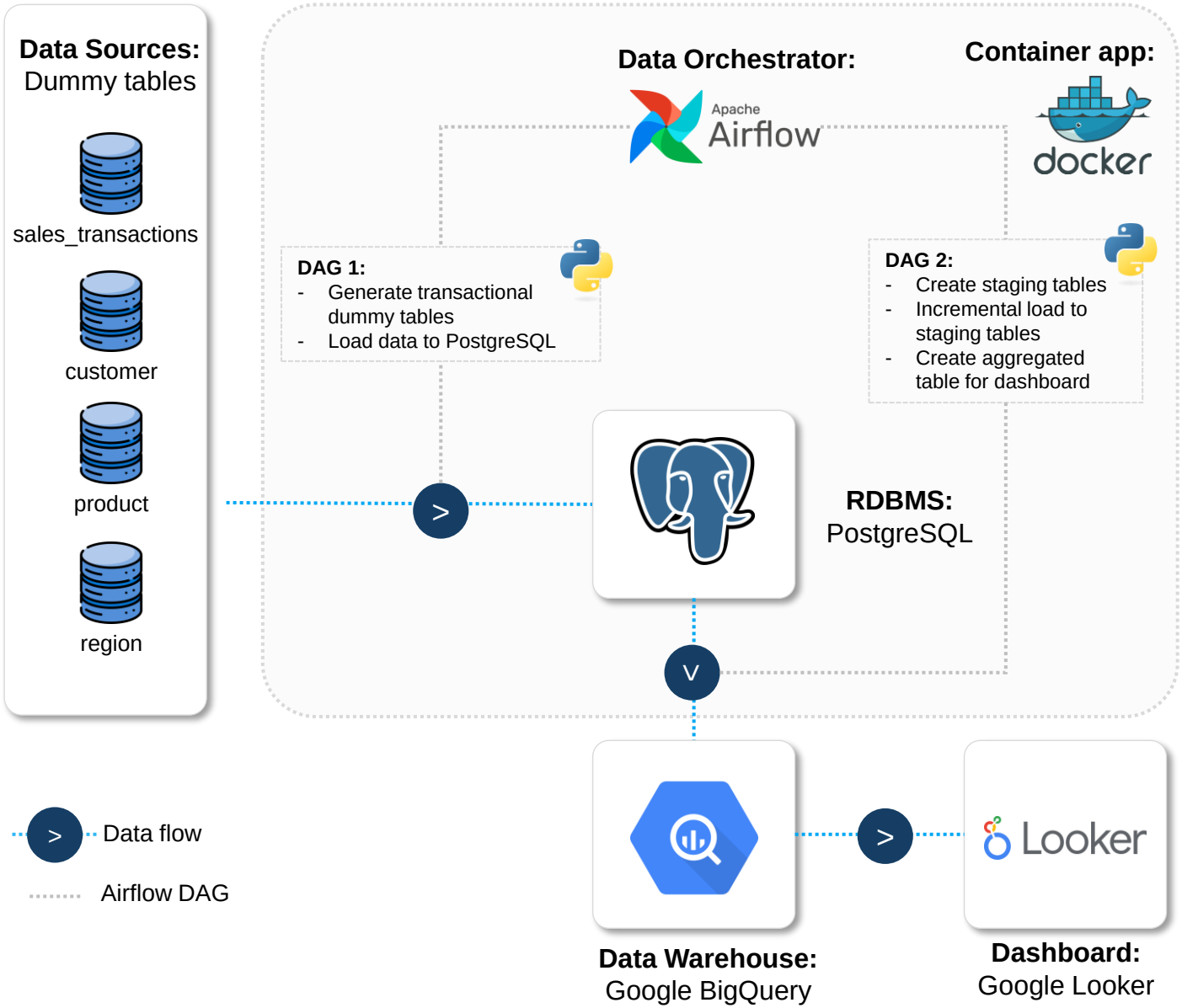
Maintenance
Continuously monitor and improve the data pipeline based on user feedback and evolving business needs

Stage 1: Planning and gathering design requirements

Defining scope and requirements of data pipeline to address current challenges**:**

Key Requirement	Description
Key Stakeholders	Marketing and Sales Team
Data Sources / Schema	Generate 4 dummy tables: 1. sales_transactions (fact) 2. product (dim) 3. customers (dim) 4. region (dim)
Transactional data storage	PostgreSQL Server
Data Warehouse	BigQuery
Batching Method	Data will be updated in daily basis at 10 AM
Incremental Load	Load only new or updated data to reduce processing time
Data Granularity	Daily
Reporting	Data and report can be accessed via Dashboard (Google Looker)

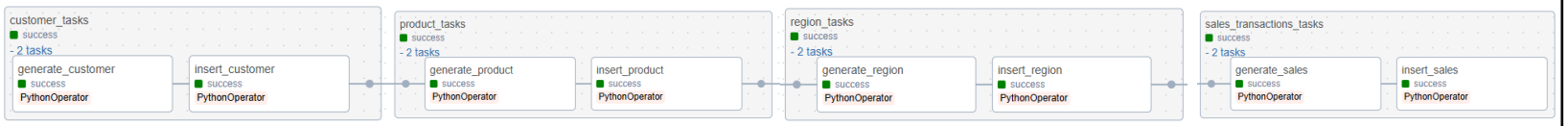
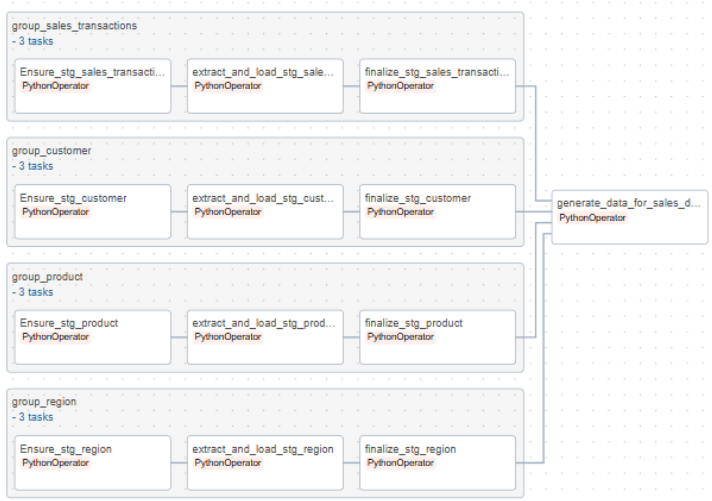
Stage 2: Data Pipeline Architecture Design



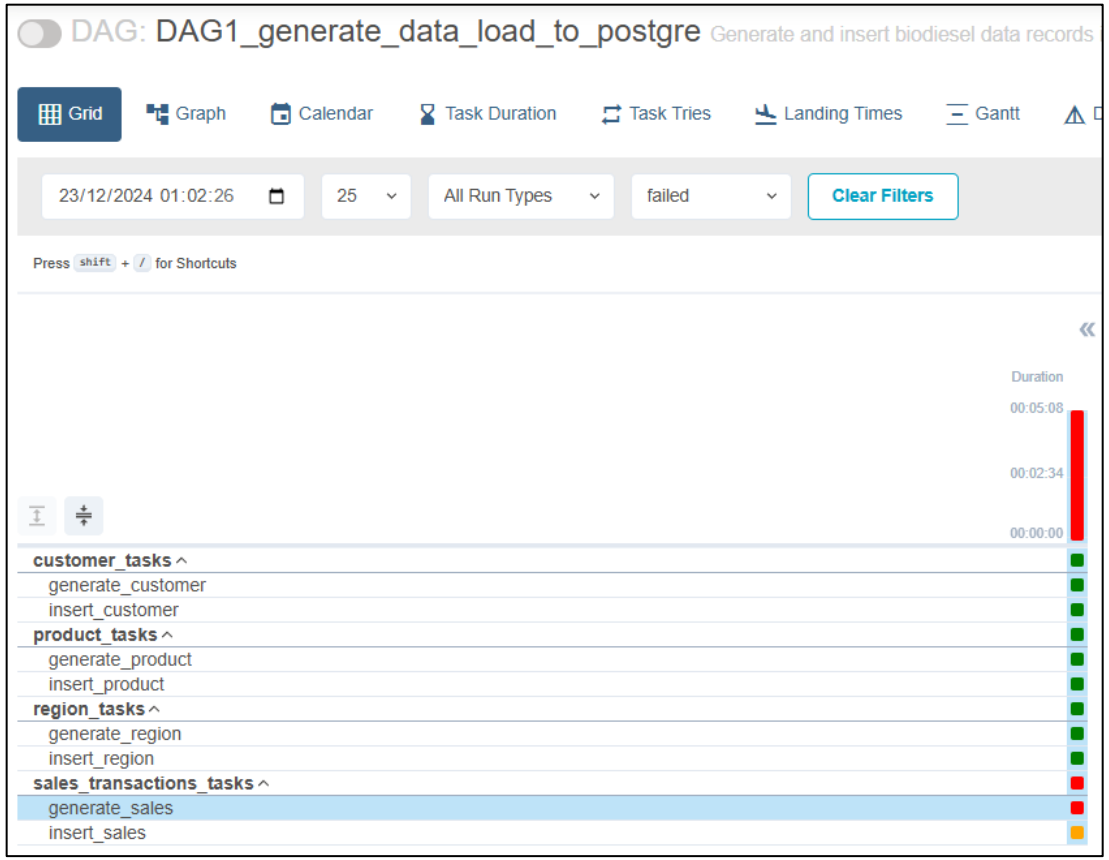
Tools utilized in this project:

Function	Tools	Explanation
Data Orchestrator	Apache Airflow	• Able to define task dependency • Suitable for batching pipeline
Program Script	Python	• Versatile programming language for building data pipeline and processing
Containerization	Docker	• Enables all apps run consistently for development
RDBMS	PostgreSQL	• The data sources are structured data, suitable for storing data with relationships.
Data Warehouse	Google BigQuery	• Partitioning enables faster querying, and improving performance • Good performance warehouse that's scalable and efficient
Dashboard	Google Looker	• Seamlessly integrates with Google BigQuery, allowing real-time data updates

Stage 3: Code Development

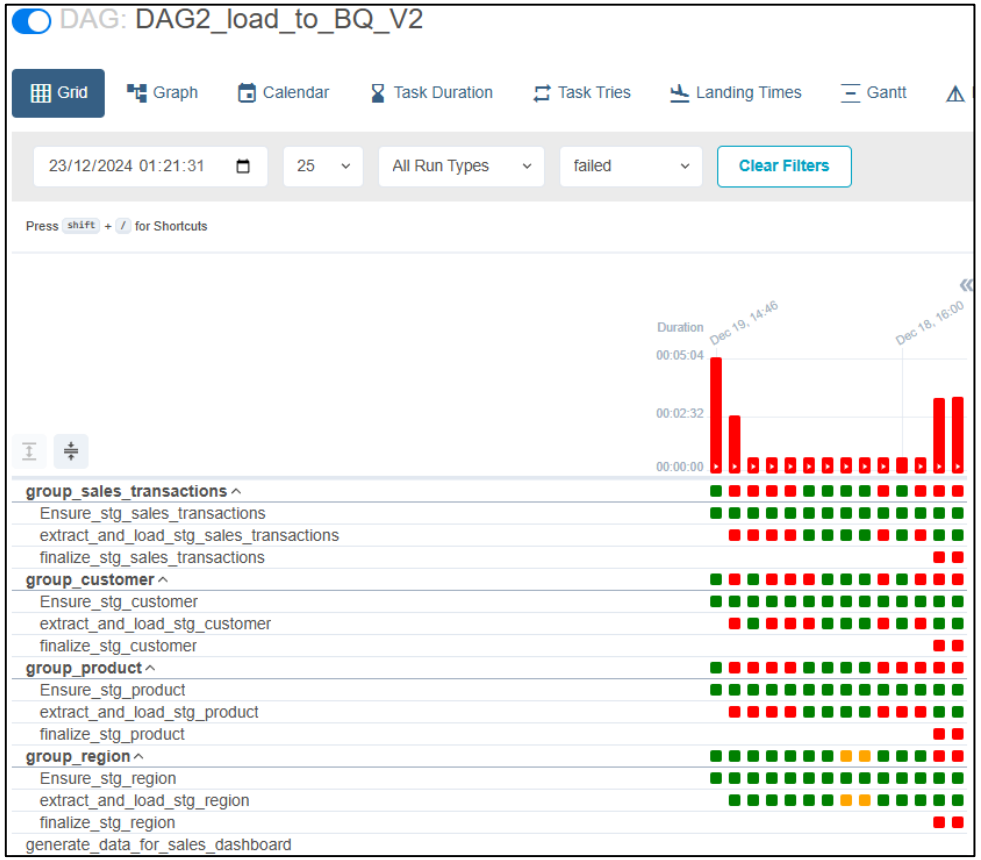
Code	Code Objective	File Name	Logical Flow/Task
Docker	• Build Airflow • Build PostgreSQL	• docker-compose.yaml (Airflow) << click here >> • docker-compose.yaml (postgresQL) << click here >>	N/A
DAG 1	• Generate dummy tables • Load to PostgreSQL	• DAG1_load_to_postgre.py as main DAG • Generate_dummy.py as helper • Insert_data.py as helper << click here >>	 **DAG's task dependency**: customer_group >> product_group >> region_group >> sales_group **Each group consists of 2 tasks**: generate_data >> insert_data
DAG 2	• Ensure staging table • Fetch data from PostgreSQL and incrementally load to staging tables in BQ • Create final data using replication • Create aggregated data for dashboard	• DAG2V2_load_to_BQ.py as main DAG • Helper_bigquery.py << click here >>	 **DAG's task dependency**: Ensure_table_task >> extract_and_load_task >> final_table task **Sub-task dependency** Final_table_task >> sales_dashboard_task

Stage 4: Testing



DAG1 Testing Log

During testing, **only one** error was recorded. The task failed because the function to connect to PostgreSQL was **not available** in the code. As a result, the code **was updated and successfully worked** at this stage.

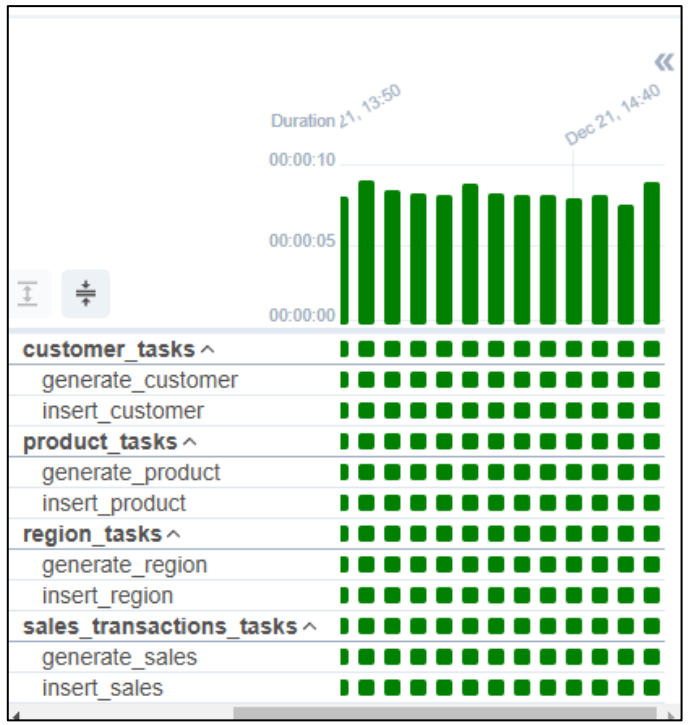


DAG2 Testing Log

During testing, **13 errors** was recorded. The task failed due to various reasons: failed fetching, failed ensuring staging tables, etc. Thus, every error was thoroughly learned and revised. As a result, the code **was updated and successfully worked** at this stage.

Stage 5 & 6: Deployment and Maintenance

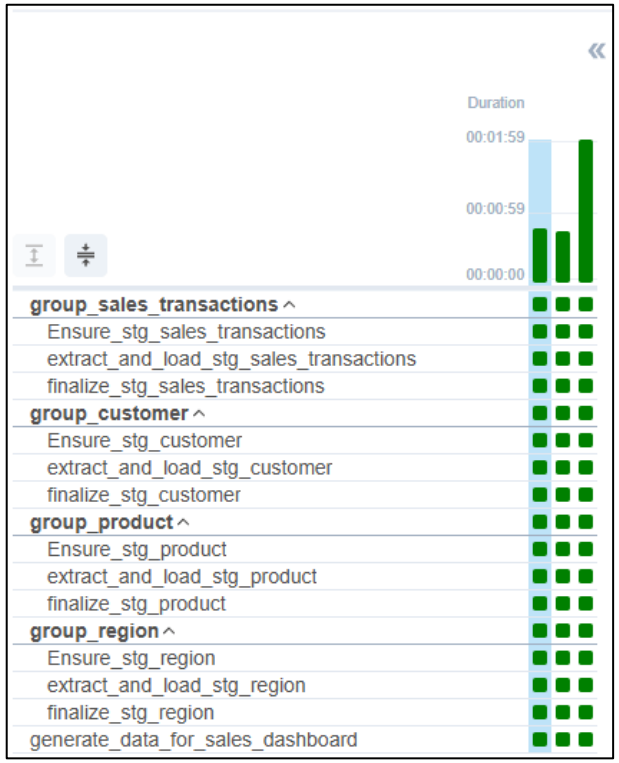
DAG: DAG1_generate_data_load_to_postgre



DAG1 Testing Log

DAG1 was deployed with a scheduled run on December 19, 2024. The task will execute automatically every 5 minutes to generate data and load to PostgreSQL. The scheduled tasks have been successfully performed and are running smoothly. Monitoring and maintenance will be closely overseen.

DAG: DAG2_load_to_BQ_V2

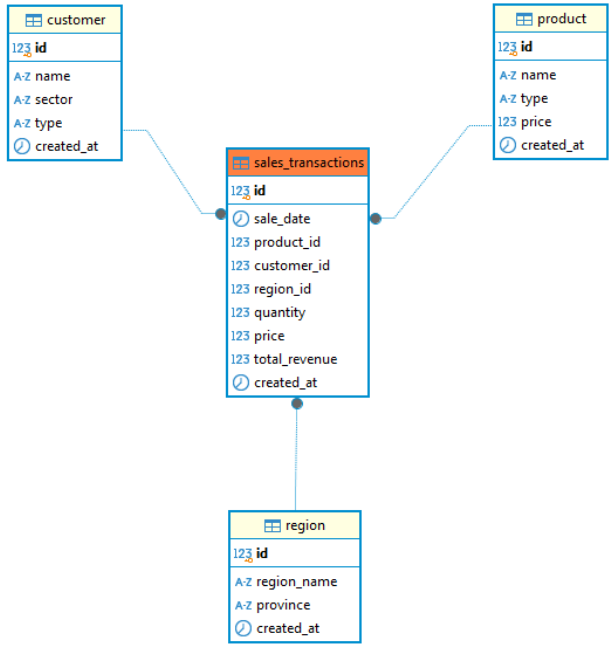


DAG2 Testing Log

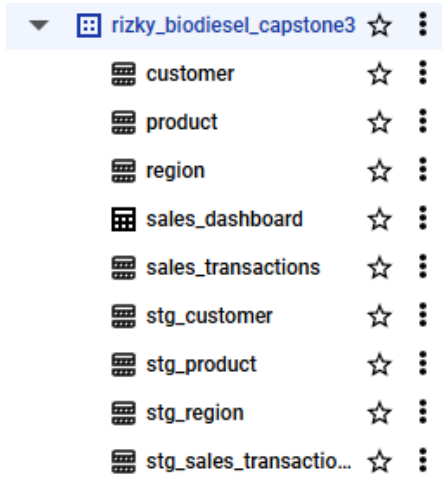
DAG2 were deployed using scheduled run in 20 December 2024. The task will run automatically daily at 10.00 AM local Jakarta time. The scheduled tasks have been successfully performed and are running smoothly. Monitoring and maintenance will be closely overseen.

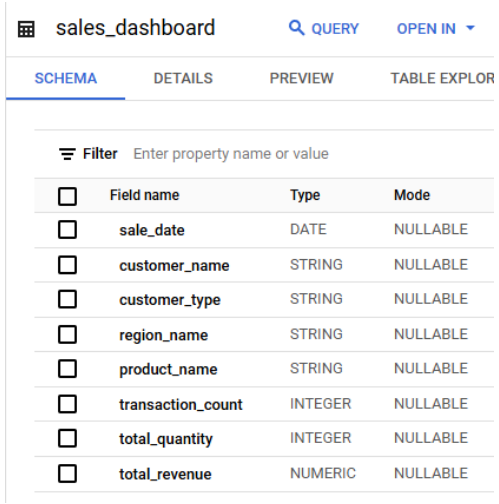
Project Results

Successfully generate data, load data, and build relationship between tables in PostgresSQL– all orchestrated using Airflow

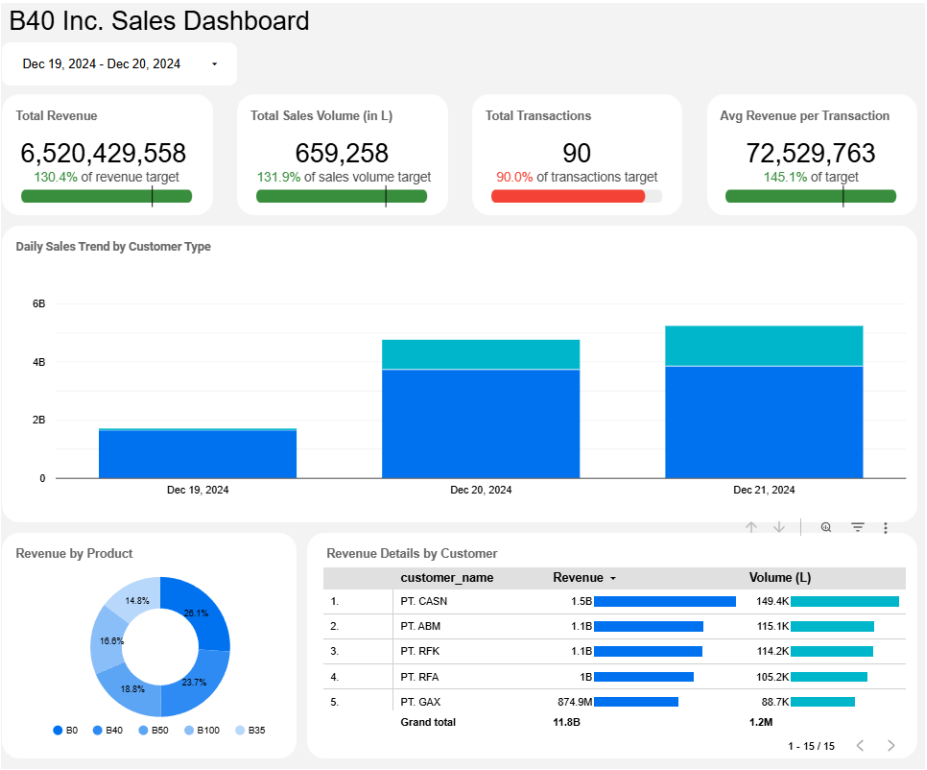


Successfully fetched data from Postgre, incrementally load to staging stables in BigQuery, create sales_dashboard table as source for dashboard

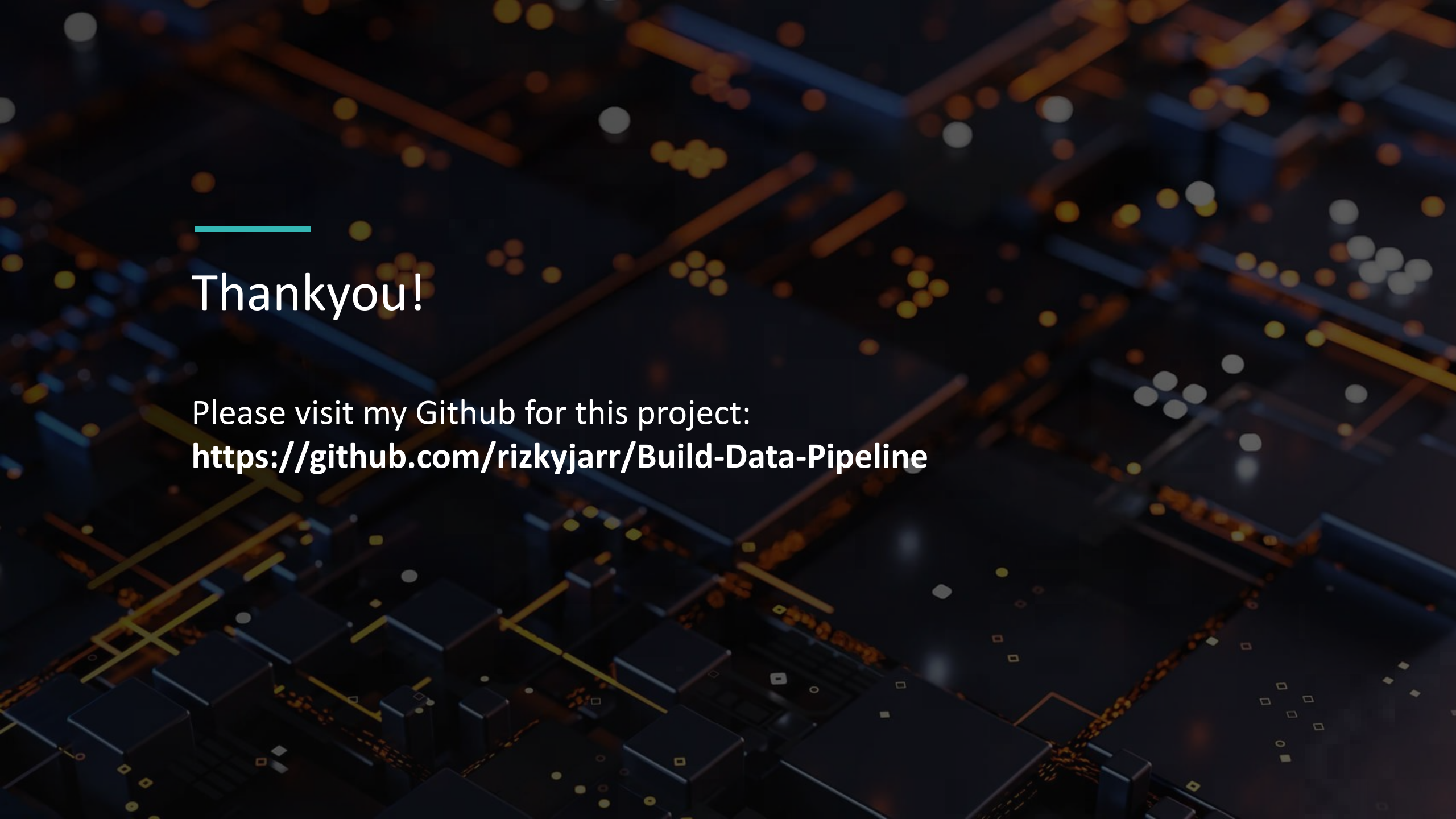




Created dashboard to generate business insights and informations: **reducing reporting time** through automation, **increase data quality**, and **enhancing decision-making capabilities**



<< Dashboard link click here >>


Thankyou!
Please visit my Github for this project:
https://github.com/rizkyjarr/Build-Data-Pipeline