BUILDING AND SCALING ETL WITH AIRFLOW

MUHAMMAD RANDA YANDIKA

LinkedIn
https://www.linkedin.com/in/muhammad-randa-yandika/

Website
https://randayandika.github.io/

September 2024

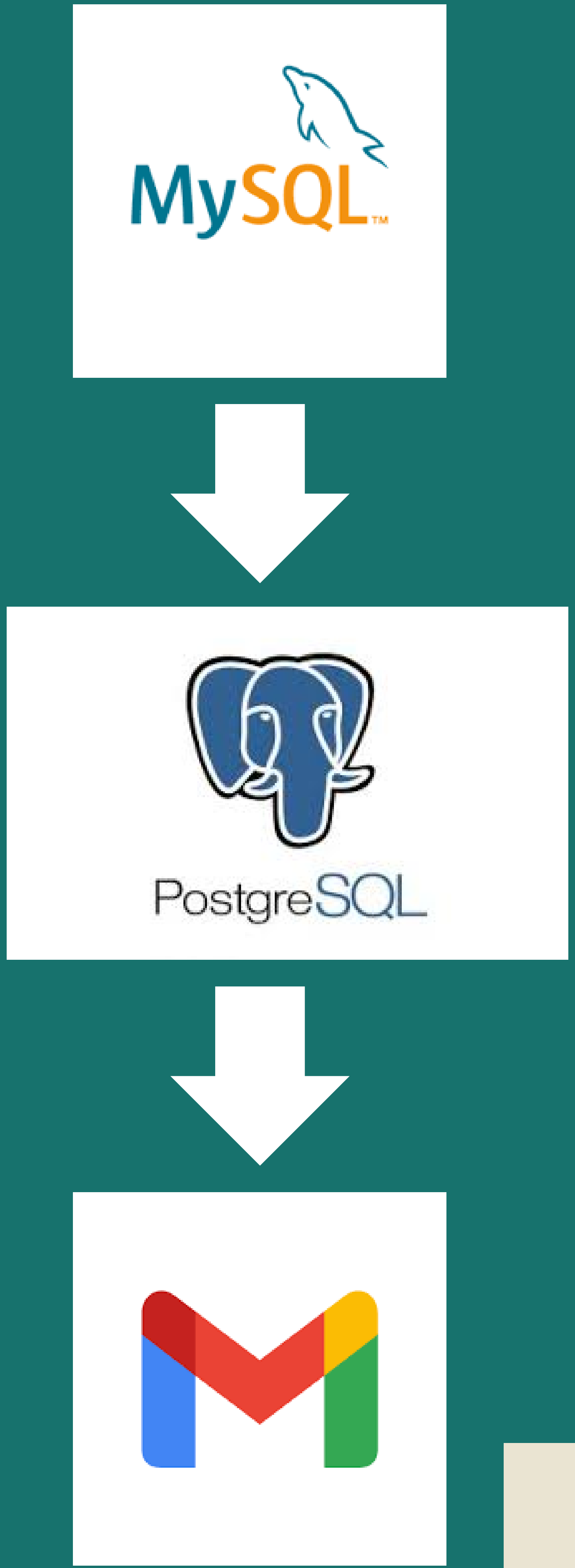
TABLE OF CONTENT

PROJECT OVERVIEW

ETL DATA AUTOMATION USING AIRFLOW, MYSQL, AND POSTGRESQL

This project implements a simple ETL process using Apache Airflow to extract data from MySQL and load it into PostgreSQL daily at 7 AM. The processed data is stored in a staging area in Parquet format. Once the ETL process is completed, a basic analysis job runs, and the results are sent via email at 9 AM.

The workflow is managed through three DAGs: one for ETL, one for data aggregation, and one for report delivery.



PROJECT BACKRGOUND

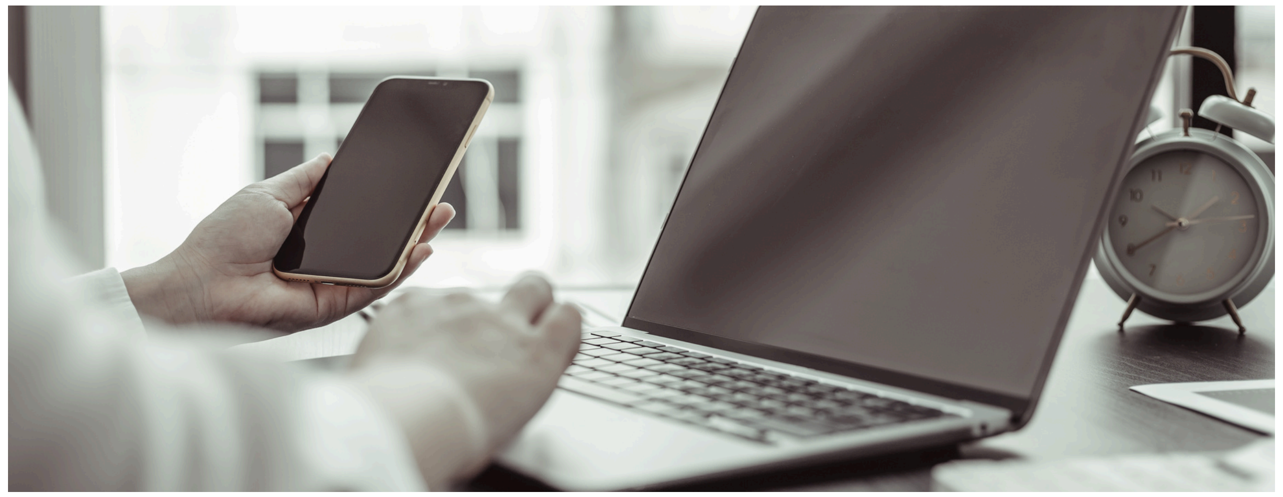
PROJECT BACKGROUND

This project aims to automate the data processing workflow using Apache Airflow, specifically focusing on the ETL process. The need for a streamlined approach to data extraction, transformation, and loading between databases was critical to ensure consistent, efficient, and error-free data movement.

WHO BENEFITS

The solution is particularly useful for organizations that handle large volumes of data across multiple platforms and require timely, accurate insights. Teams relying on frequent data updates for decision-making, reporting, or monitoring processes will greatly benefit from this automated pipeline, as it reduces manual workload and ensures consistent data availability.

PROBLEM STATEMENT



01 Problem Statement

The challenge addressed by this project is ensuring the reliable, automated transfer of data from MySQL to PostgreSQL on a daily basis. This requires building a robust ETL pipeline that guarantees data accuracy and timeliness, with minimal manual intervention. Without automation, delays in data processing could lead to outdated or incomplete information for analysis.

02 Project Objective

The project aims to design and implement a fully automated ETL pipeline using Apache Airflow, capable of extracting data at 7 AM daily, loading it into a PostgreSQL staging area, and performing necessary data aggregation. By 9 AM, the pipeline should deliver reports based on the latest data via email, ensuring that stakeholders receive timely and accurate insights.

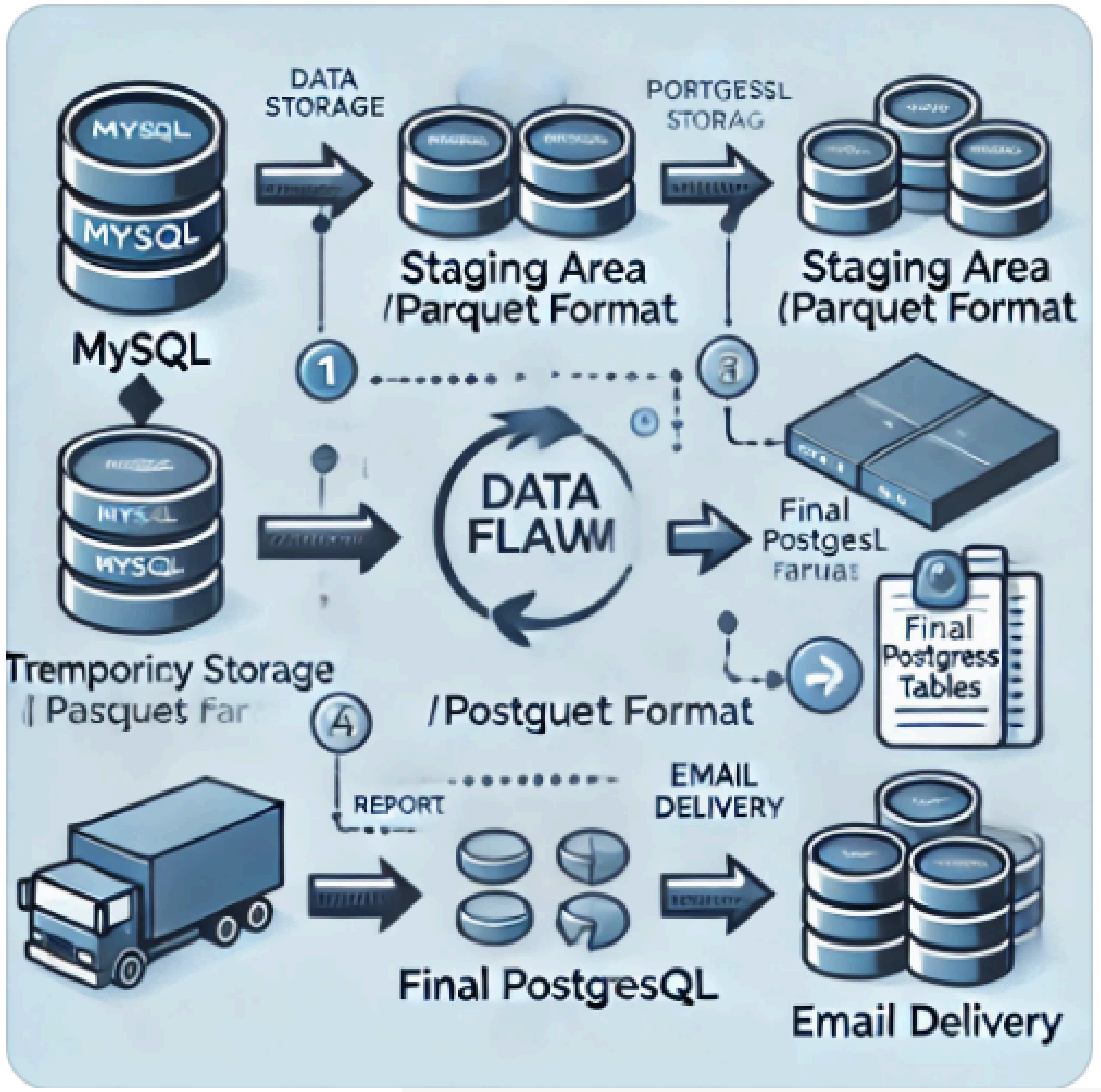
DATA PLATFORM UNDERSTANDING

01 Data Platform Overview

The data platform is designed to facilitate efficient extraction, transformation, and storage of data using modern database technologies and formats. In this project, data is extracted daily from a MySQL source and loaded into a PostgreSQL staging area using the Parquet format.

02 Data Flow Diagram

Data is extracted from MySQL and temporarily stored in Parquet format in PostgreSQL. After processing, it's loaded into final tables, with reports generated and emailed at scheduled intervals to provide timely insights.

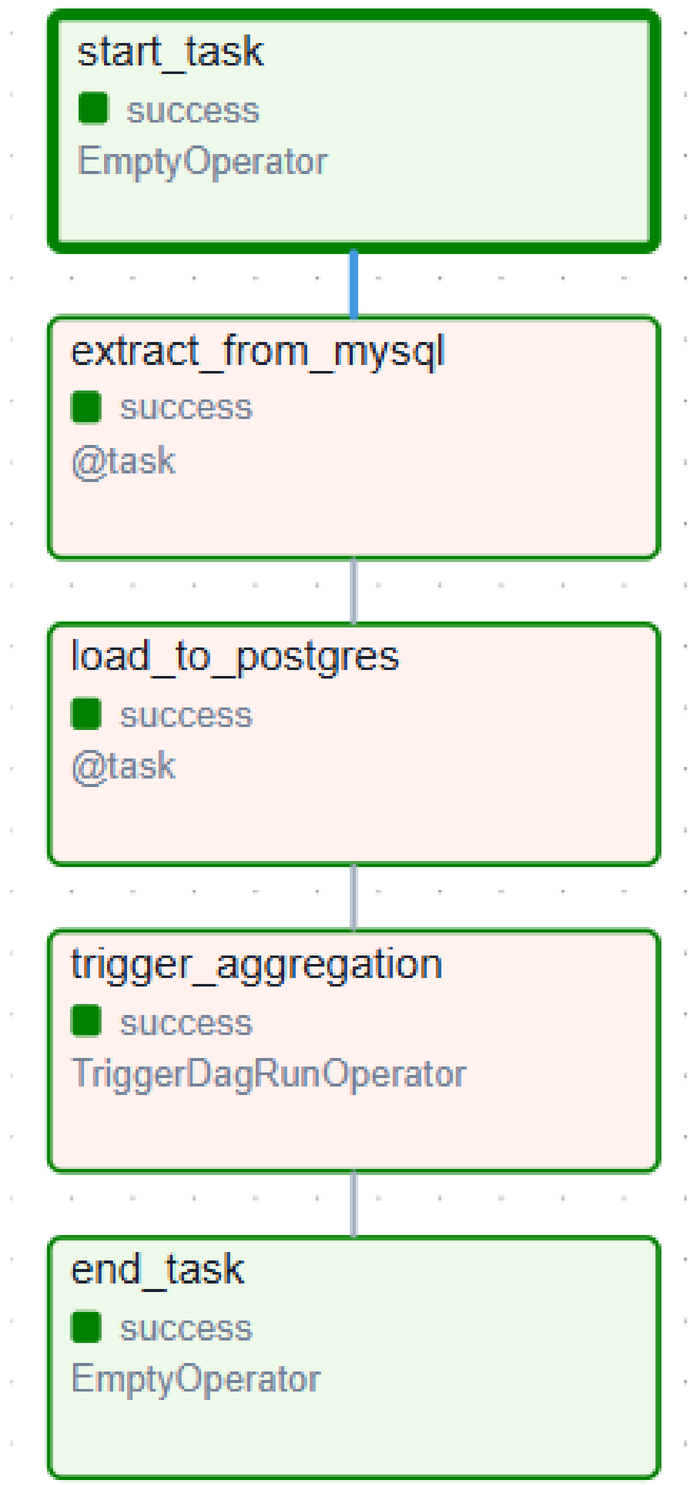


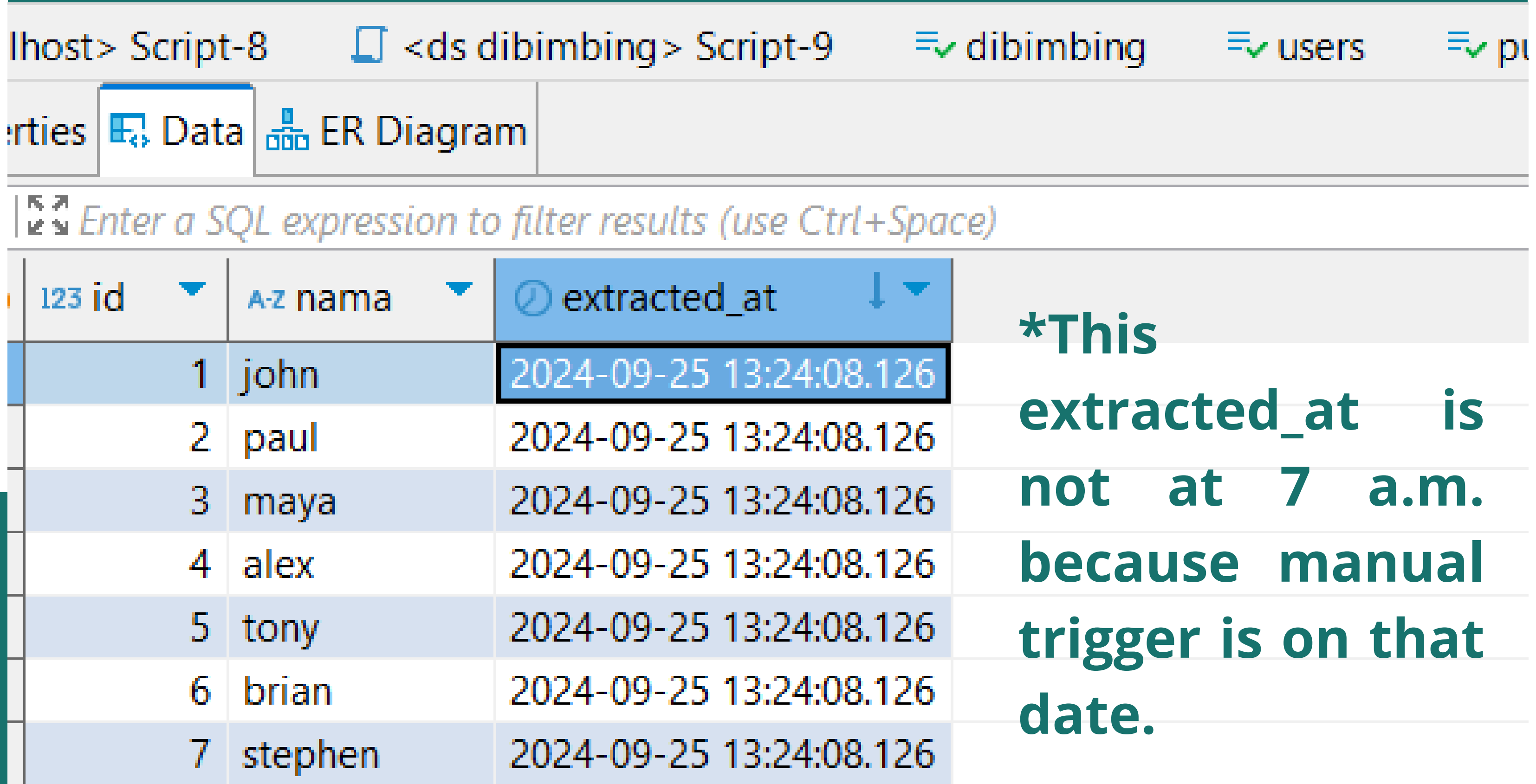
ETL PROCESS

● EXTRACT & LOAD -> AGGREGATION --> SEND TO EMAIL

EXTRACT & LOAD

The randa_dag_etl DAG is an ETL workflow that runs daily at 7 AM to extract user data from a MySQL database, specifically from the dibimbing.users table. It uses a MySQL hook to connect and retrieve the data, saving it as a Parquet file in the staging area with a timestamp. The extracted data is then loaded into a PostgreSQL database using a Postgres hook.





id	nama	extracted_at
1	john	2024-09-25 13:24:08.126
2	paul	2024-09-25 13:24:08.126
3	maya	2024-09-25 13:24:08.126
4	alex	2024-09-25 13:24:08.126
5	tony	2024-09-25 13:24:08.126
6	brian	2024-09-25 13:24:08.126
7	stephen	2024-09-25 13:24:08.126

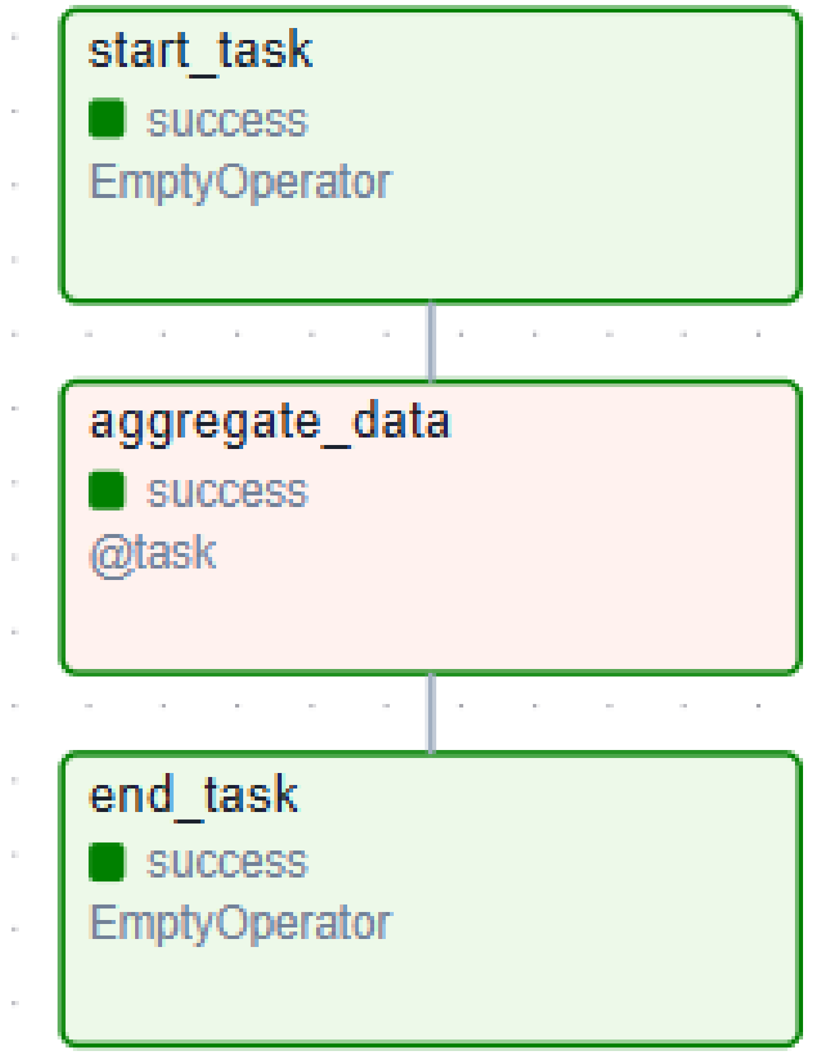
***This extracted_at is not at 7 a.m. because manual trigger is on that date.**

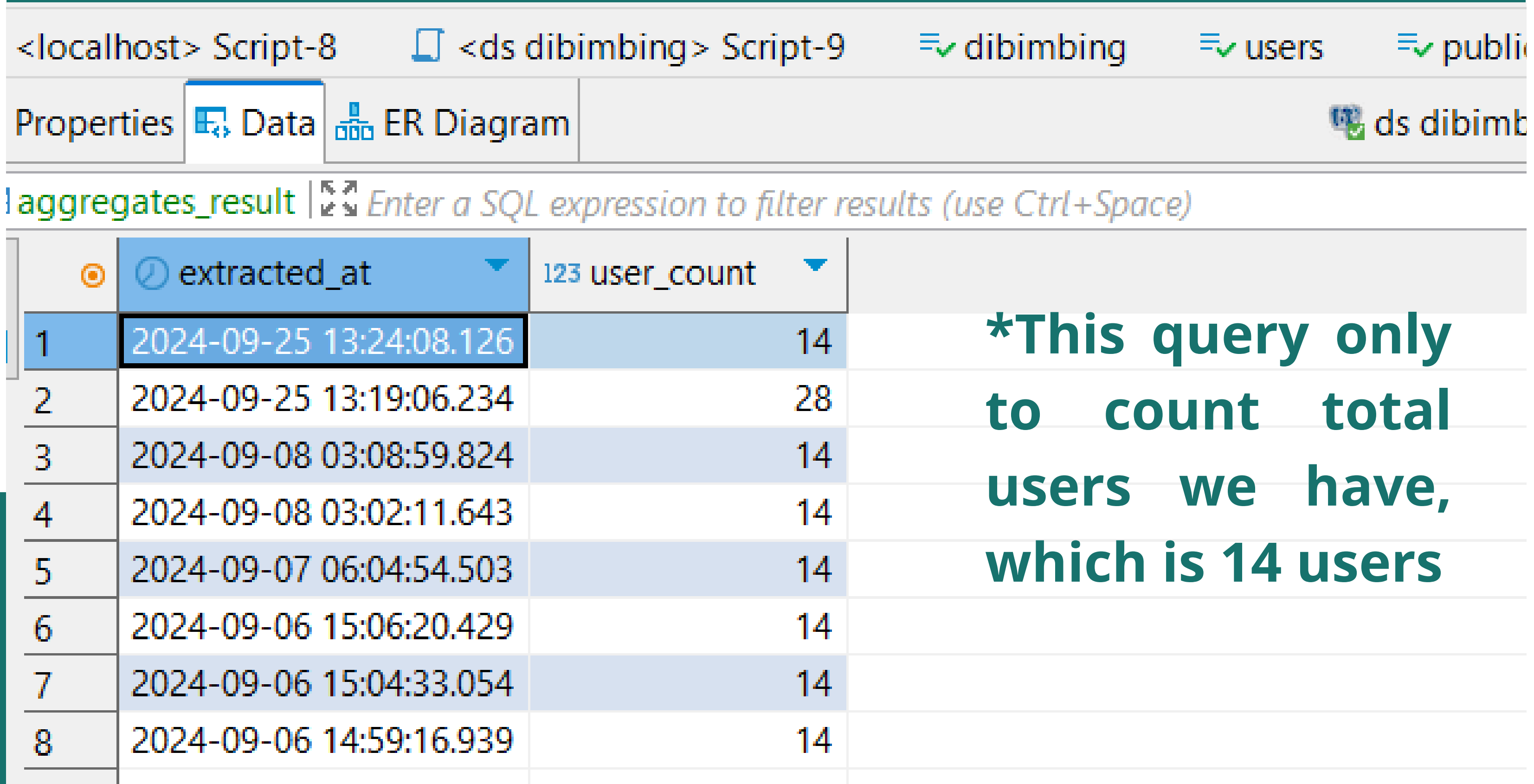
After successfully loading the data, the DAG triggers another DAG (randa_dag_aggregate) for further processing and concludes with an end task, ensuring a seamless data migration and processing pipeline.

AGGREGATION

The randa_dag_aggregate DAG is designed to perform data aggregation on user records in a PostgreSQL database, executing after previous dags is success to run. It begins with a start task and proceeds to aggregate user data by counting the number of users grouped by their extraction date.

The aggregation is performed using a PostgreSQL hook, and the results are stored in the aggregates_result table, replacing any existing data.





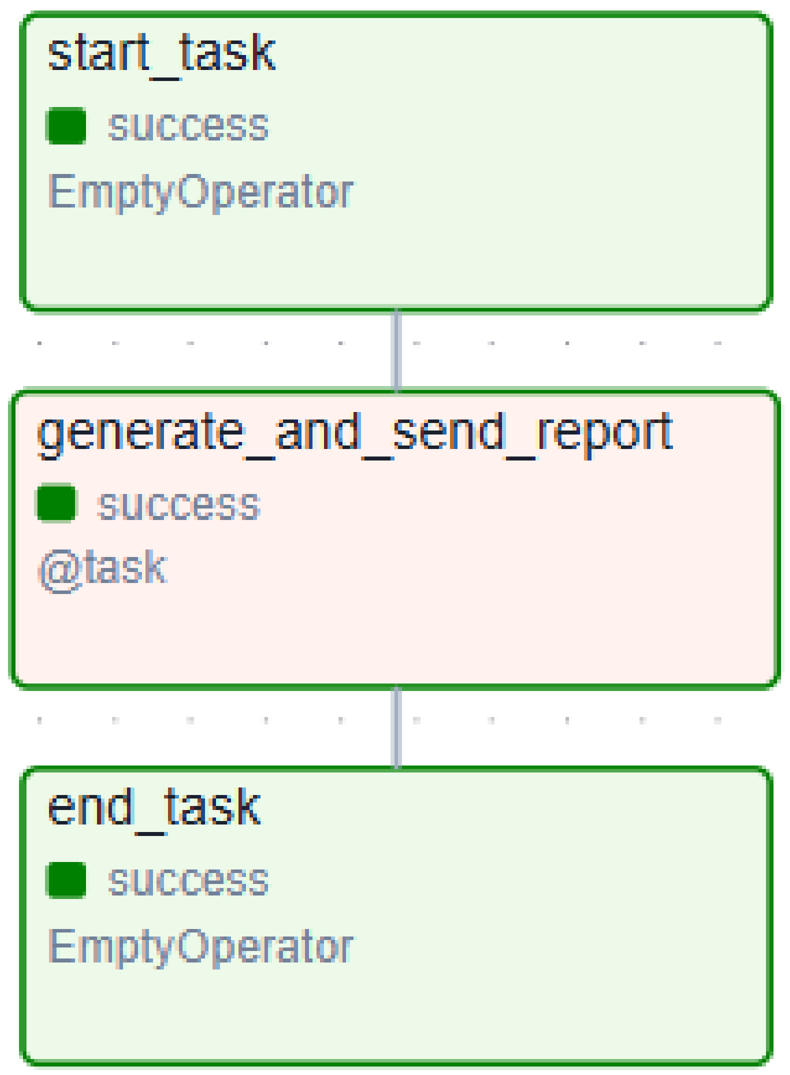
	extracted_at	user_count
1	2024-09-25 13:24:08.126	14
2	2024-09-25 13:19:06.234	28
3	2024-09-08 03:08:59.824	14
4	2024-09-08 03:02:11.643	14
5	2024-09-07 06:04:54.503	14
6	2024-09-06 15:06:20.429	14
7	2024-09-06 15:04:33.054	14
8	2024-09-06 14:59:16.939	14

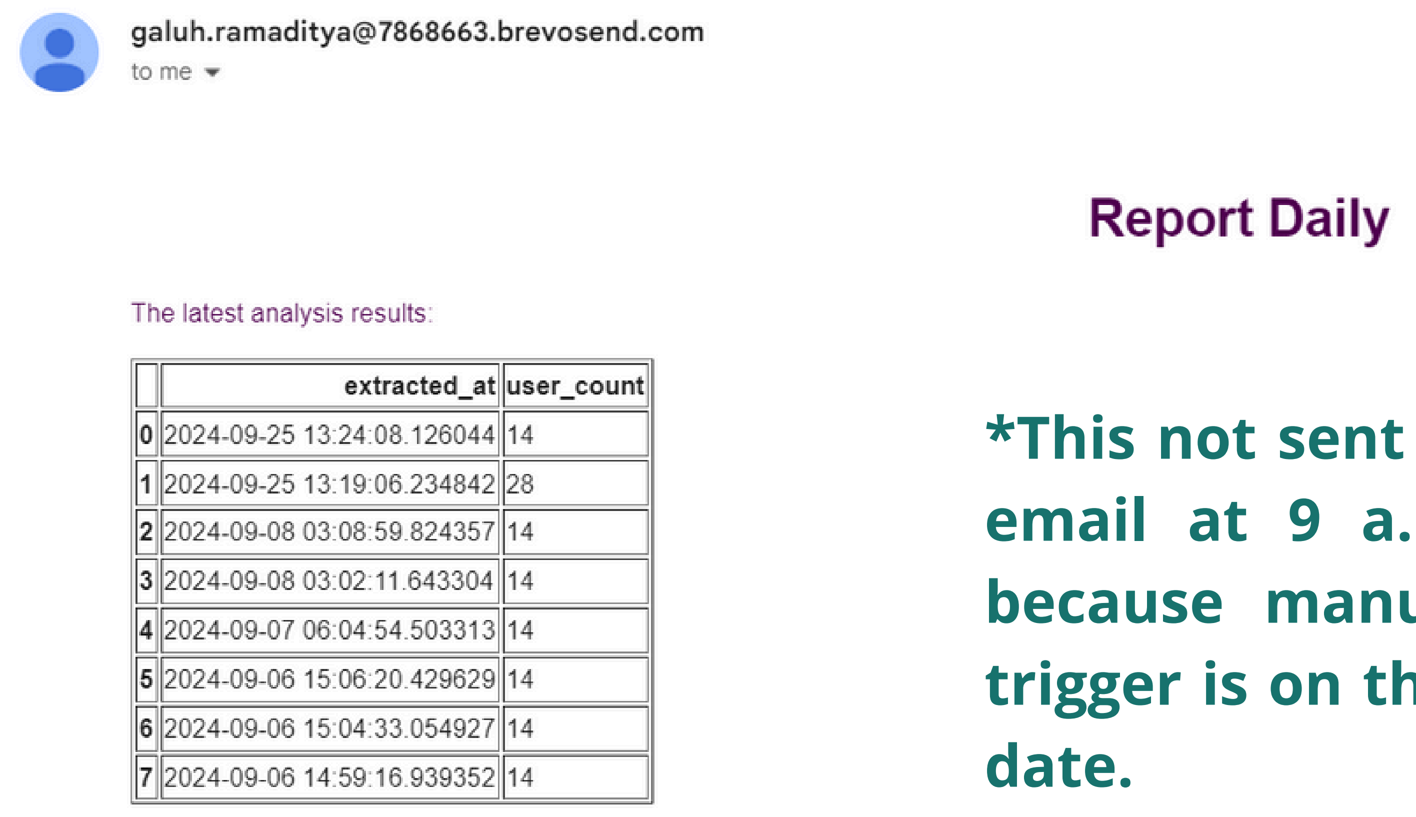
***This query only to count total users we have, which is 14 users**

Finally, the workflow concludes with an end task, ensuring a streamlined process for generating and storing aggregated user statistics.

SEND REPORT TO EMAIL

The randa_dag_report DAG is designed to generate and send a daily report via email at 9 AM after aggregation task is success to run. It begins with a start task and proceeds to connect to a PostgreSQL database to fetch the latest analysis results from the aggregates_result table.
The results are converted into an HTML table format using pandas, and an email is sent to the specified recipient with the report content.




galuh.ramaditya@7868663.brevosend.com
to me

Report Daily

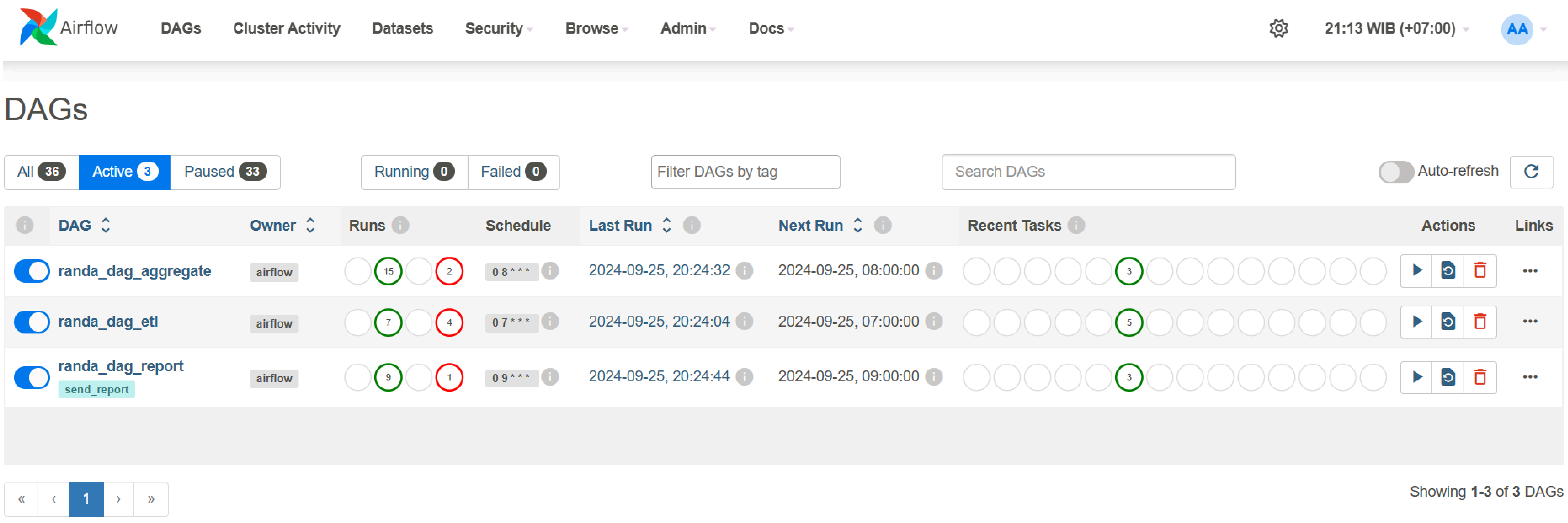
The latest analysis results:

	extracted_at	user_count
0	2024-09-25 13:24:08.126044	14
1	2024-09-25 13:19:06.234842	28
2	2024-09-08 03:08:59.824357	14
3	2024-09-08 03:02:11.643304	14
4	2024-09-07 06:04:54.503313	14
5	2024-09-06 15:06:20.429629	14
6	2024-09-06 15:04:33.054927	14
7	2024-09-06 14:59:16.939352	14

***This not sent to email at 9 a.m. because manual trigger is on that date.**

The workflow concludes with an end task, ensuring that the email is only sent if the report generation task is successful, thereby facilitating timely delivery of daily analysis insights.

DAGS VIEW IN APACHE AIRFLOW



Airflow | DAGs | Cluster Activity | Datasets | Security | Browse | Admin | Docs | 21:13 WIB (+07:00) | AA

DAGs

All 36 | Active 3 | Paused 33 | Running 0 | Failed 0 | Filter DAGs by tag | Search DAGs | Auto-refresh

DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks	Actions	Links
randa_dag_aggregate	airflow	15, 2	0 8 * * *	2024-09-25, 20:24:32	2024-09-25, 08:00:00	3		...
randa_dag_etl	airflow	7, 4	0 7 * * *	2024-09-25, 20:24:04	2024-09-25, 07:00:00	5		...
randa_dag_report send_report	airflow	9, 1	0 9 * * *	2024-09-25, 20:24:44	2024-09-25, 09:00:00	3		...

« ‹ 1 › »

Showing **1-3** of **3** DAGs

CONCLUSION

WHAT WE COULD DO

Conclusion

The Airflow DAGs automate the data pipeline by first performing ETL, where data is extracted from MySQL, stored in Parquet, and loaded into PostgreSQL. After the ETL process, the aggregation DAG groups counting the number of users and saves the result in PostgreSQL. Finally, the report DAG sends a daily email with the latest analysis results, completing the process from data extraction to reporting.

Recommendation

To improve the system, it's advisable to enhance error handling across all DAGs, especially for database connections and data processing tasks. Adding data validation steps before loading to ensure quality would also be beneficial. Monitoring and alerting features, such as task failure notifications, should be integrated for real-time tracking.

THANK YOU

FOR YOUR NICE ATTENTION

LinkedIn
https://www.linkedin.com/in/muhammad-randa-yandika/

Website
https://randayandika.github.io/

September 2024