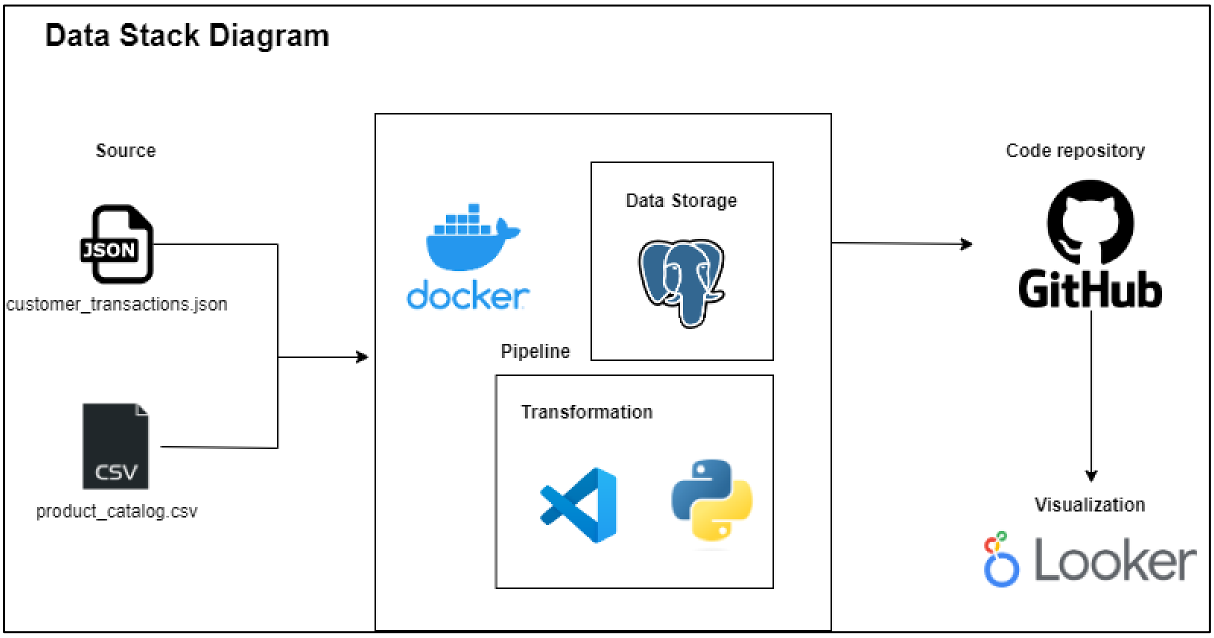


My Data Engineering Stack

In my data engineering workflow, I employ a comprehensive stack to efficiently manage and process data from its origin to visualization. Here's an overview of the stack I use:

Data Collection

To begin with, I gather data from various premises or origin sources. This data can come from multiple systems, applications, or databases, depending on the specific requirements of the project.

ETL Process

Once the data is collected, I use Python to perform the ETL (Extract, Transform, Load) process. This involves:

- **Extracting** the raw data from the origin sources.
- **Transforming** the data to fit the required format, which may include cleaning, aggregating, and enriching the data.
- **Loading** the transformed data into a data storage solution.

Data Storage

For data storage, I utilize PostgreSQL, which is set up on Docker. Docker allows me to containerize the PostgreSQL database, ensuring a consistent and portable environment. The entire pipeline and necessary libraries are also configured within this Docker setup, providing a robust and scalable solution for managing data.

Code Repository

To manage and store my code, I use GitHub. GitHub serves as my code repository, allowing for version control, collaboration, and efficient code management. By leveraging GitHub, I can ensure that my codebase is organized, accessible, and backed up.

Data Visualization

Finally, for the visualization part, I chose Looker Studio, a service from Google. Looker Studio enables me to build interactive and insightful dashboards, which help in analyzing and presenting the processed data effectively. With Looker Studio, I can create customized visualizations that provide valuable insights into the data, making it easier to interpret and act upon.

Database Flow

I decided to structure the database into two schemas: "pre_prd" and "prd".

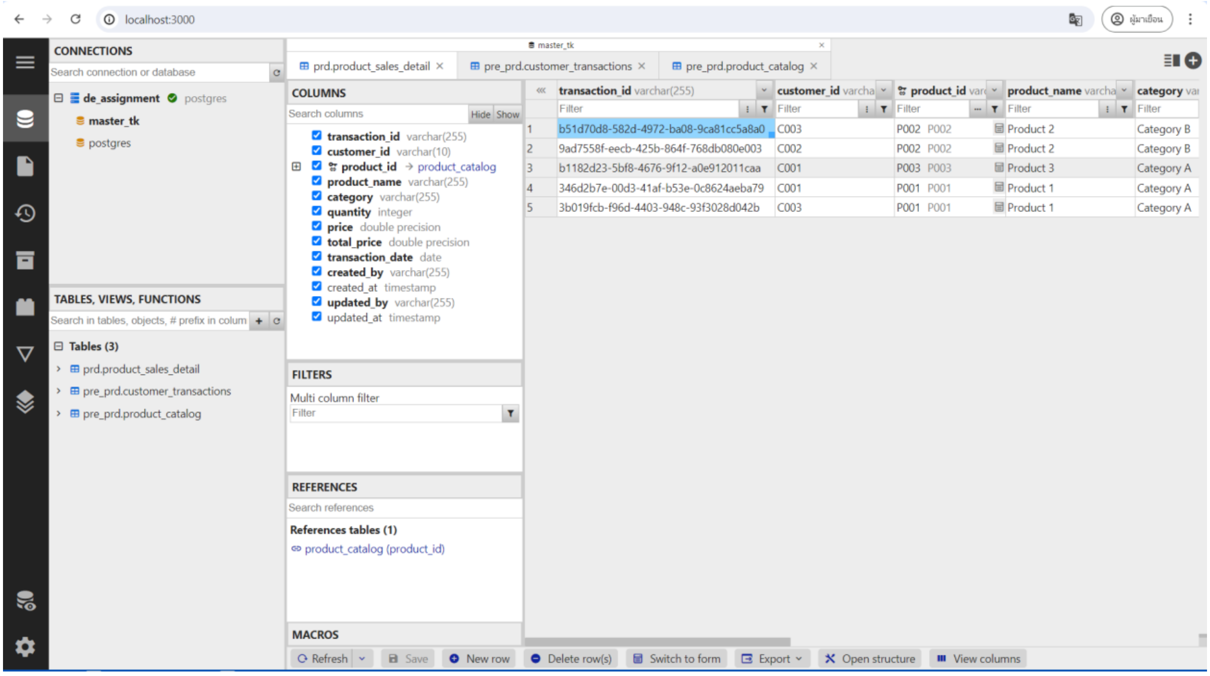
pre_prd Schema

The "pre_prd" schema is used to store raw data obtained from various sources. In this case, the sources are customer_transactions[2].json and product_catalog[1].csv. The data in this schema is cleaned and data types are defined, but the tables have not yet been joined to create a data mart.

prd Schema

The "prd" schema is used to collect data marts based on the data of interest. For this specific use case, we plan to join the two tables from the "pre_prd" schema to create a new table named "product_sales_detail". This table will represent the data mart that we are interested in and will be used to generate dashboards.

For SQL that I use to create table, you can see on file: database_source.sql



☑ transaction_id varchar(255)
☑ customer_id varchar(10)
☑ product_id → product_catalog
☑ product_name varchar(255)
☑ category varchar(255)
☑ quantity integer
☑ price double precision
☑ total_price double precision
☑ transaction_date date
☑ created_by varchar(255)
☑ created_at timestamp
☑ updated_by varchar(255)
☑ updated_at timestamp

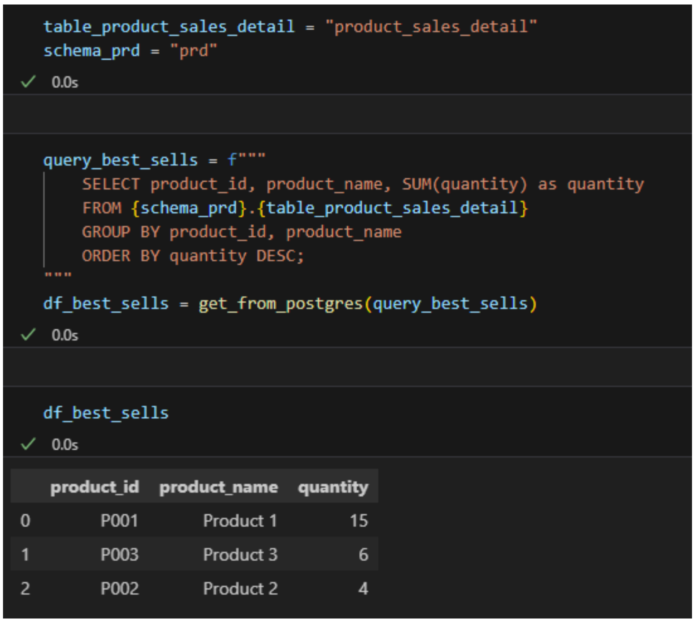
	transaction_id varchar(255)	customer_id varcha	product_id varc	product_name varcha	category var
1	b51d70d8-582d-4972-ba08-9ca81cc5a8a0	C003	P002 P002	Product 2	Category B
2	9ad7558f-eecb-425b-864f-768db080e003	C002	P002 P002	Product 2	Category B
3	b1182d23-5bf8-4676-9f12-a0e912011caa	C001	P003 P003	Product 3	Category A
4	346d2b7e-00d3-41af-b53e-0c8624aeba79	C001	P001 P001	Product 1	Category A
5	3b019fcb-f96d-4403-948c-93f3028d042b	C003	P001 P001	Product 1	Category A

Answer questions number 6:

SQL/NOSQL Queries: Write SQL/NOSQL queries to answer the following business questions:

i. What are the top 2 best-selling products?

Ans: The top 2 best-selling products are: first, Product 1, and second, Product 2.



```
table_product_sales_detail = "product_sales_detail"
schema_prd = "prd"
```

✓ 0.0s

```
query_best_sells = f"""
    SELECT product_id, product_name, SUM(quantity) as quantity
    FROM {schema_prd}.{table_product_sales_detail}
    GROUP BY product_id, product_name
    ORDER BY quantity DESC;
"""
df_best_sells = get_from_postgres(query_best_sells)
```

✓ 0.0s

```
df_best_sells
```

✓ 0.0s

	product_id	product_name	quantity
0	P001	Product 1	15
1	P003	Product 3	6
2	P002	Product 2	4

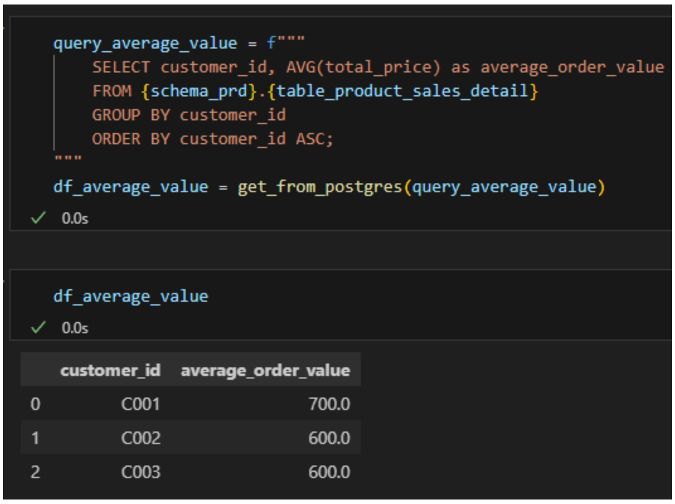
ii. What is the average order value per customer?

Ans: Average per user is follow this:

Customer: C001, Average value is 700

Customer: C002, Average value is 600

Customer: C003, Average value is 600



```
query_average_value = f"""
    SELECT customer_id, AVG(total_price) as average_order_value
    FROM {schema_prd}.{table_product_sales_detail}
    GROUP BY customer_id
    ORDER BY customer_id ASC;
"""
df_average_value = get_from_postgres(query_average_value)
```

✓ 0.0s

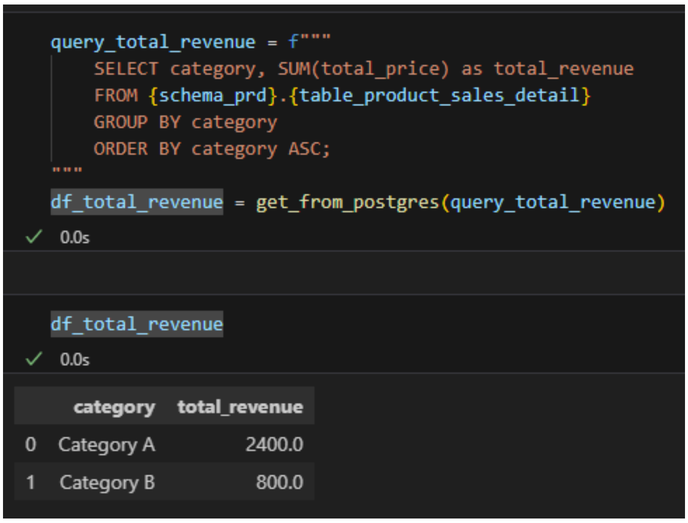
```
df_average_value
```

✓ 0.0s

	customer_id	average_order_value
0	C001	700.0
1	C002	600.0
2	C003	600.0

iii. What is the total revenue generated per product category?

Ans: Category A got revenue: 2400 and Category B got revenue: 800



```
query_total_revenue = f"""
    SELECT category, SUM(total_price) as total_revenue
    FROM {schema_prd}.{table_product_sales_detail}
    GROUP BY category
    ORDER BY category ASC;
"""
df_total_revenue = get_from_postgres(query_total_revenue)
```

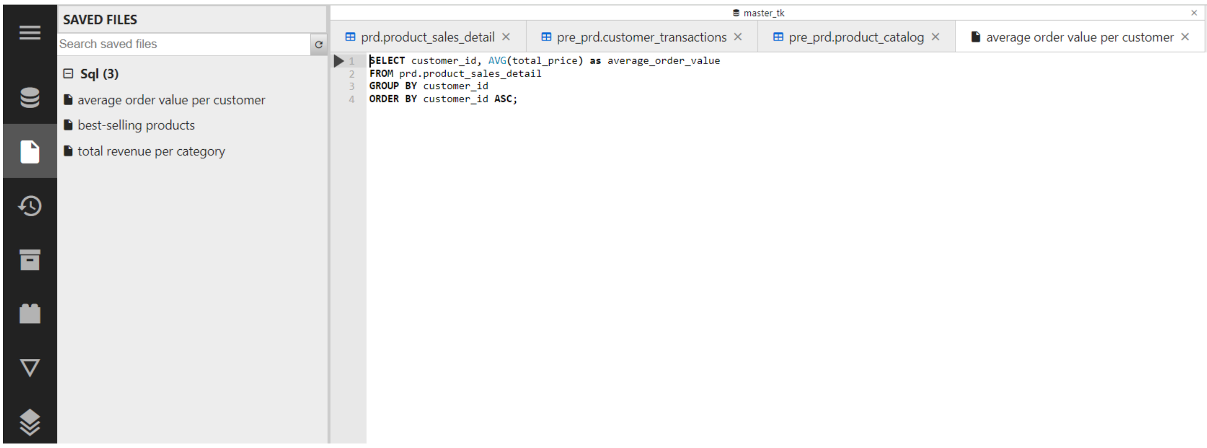
✓ 0.0s

```
df_total_revenue
```

✓ 0.0s

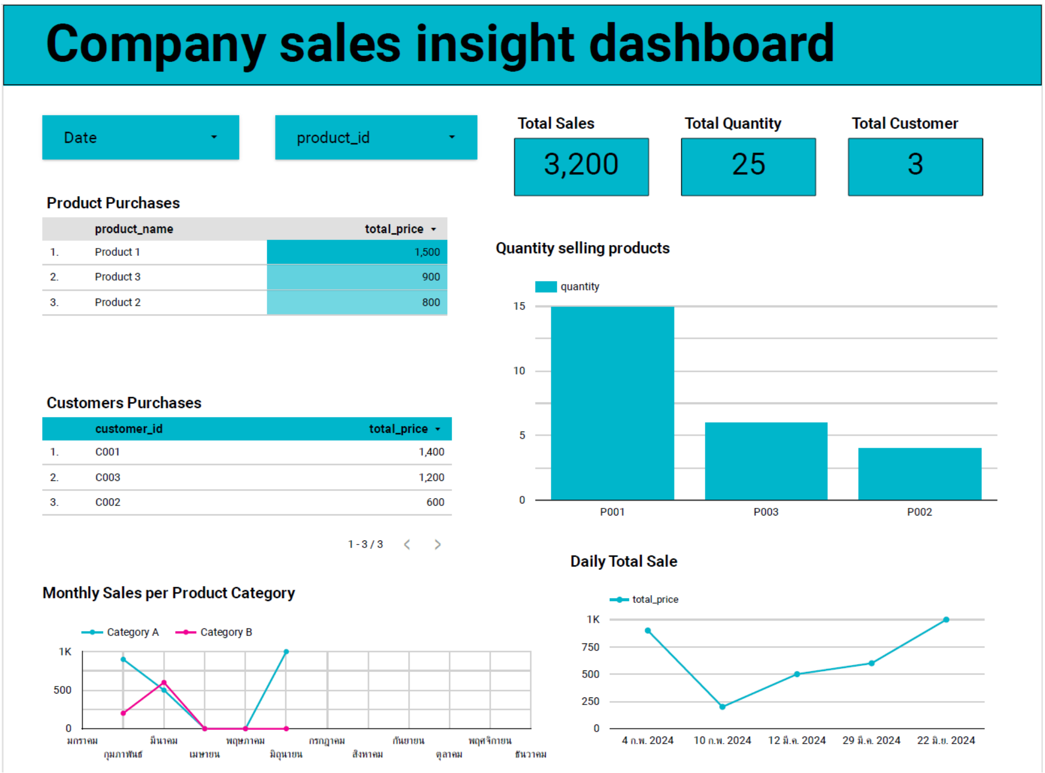
	category	total_revenue
0	Category A	2400.0
1	Category B	800.0

Incase if you want to see all of query you can see on: localhost:3000


SAVED FILES
Search saved files
Sql (3)
average order value per customer
best-selling products
total revenue per category
master_tk
prd.product_sales_detail
pre_prd.customer_transactions
pre_prd.product_catalog
average order value per customer

```
SELECT customer_id, AVG(total_price) as average_order_value
FROM prd.product_sales_detail
GROUP BY customer_id
ORDER BY customer_id ASC;
```

Company sales insight dashboard



Company sales insight dashboard

Date

product_id

Total Sales: 3,200

Total Quantity: 25

Total Customer: 3

Product Purchases

	product_name	total_price ▾
1.	Product 1	1,500
2.	Product 3	900
3.	Product 2	800

Customers Purchases

	customer_id	total_price ▾
1.	C001	1,400
2.	C003	1,200
3.	C002	600

1 - 3 / 3 < >

You can access this dashboard via this link:

https://lookerstudio.google.com/reporting/adcbbaf8-2b3c-43db-b475-3d19723aa170

Finally you can see all of my code on: https://github.com/MrSixTNine/DE_Assignment.git