

Analyzing eCommerce Business Performance with SQL



Created by:

Muhammad Cikal Merdeka

Email : mcikalmerdeka@gmail.com

LinkedIn : [linkedin.com/in/mcikalmerdeka](https://www.linkedin.com/in/mcikalmerdeka)

Github : github.com/mcikalmerdeka

Dedicated entry-level data scientist with analytical and experimental background of Physics. My graduation 2023, a pivotal year marked by significant advancements in artificial intelligence with the introduction of GPT-4 and other generative AI models, has fueled my curiosity and excitement to delve into the field of data. I have comprehensive grasp of data science methodology from business understanding to modelling process with proficiency in **Python, SQL, Tableau, Power BI, Looker Studio and other tools** related to data analytics workflow from several coursework and bootcamps.

Overview :

In a company, measuring business performance is crucial for tracking, monitoring, and evaluating the success or failure of various business processes. Therefore, this project will analyze the business performance of an eCommerce company, taking into account several business metrics such as customer growth, product quality, and payment methods.

In this project, data analysis and querying will be done using **PostgreSQL**, and visualization will be performed using **Tableau**.



Datasets Information :

This dataset was generously provided by Olist, the largest department store in Brazilian marketplaces. The dataset has information of 100k orders from 2016 to 2018 made at multiple marketplaces in Brazil. Its features allows viewing an order from multiple dimensions: from order status, price, payment and freight performance to customer location, product attributes and finally reviews written by customers. We also released a geolocation dataset that relates Brazilian zip codes to lat/lng coordinates.

1. Create Database & Tables

In this case we can create database by using CREATE DATABASE or pgAdmin 4 GUI tools.

```
CREATE DATABASE
ecommerce_rakamin_miniproject;
```

Here are some of the tables created

- customers
- order_reviews
- geolocations
- orders
- order_items
- products
- order_payments
- sellers

```
-- Create Table Customers
CREATE TABLE IF NOT EXISTS customers(
    customer_id VARCHAR(50) NOT NULL,
    customer_unique_id VARCHAR(50) NULL,
    customer_zip_code_prefix VARCHAR(50) NULL,
    customer_city VARCHAR(50) NULL,
    customer_state VARCHAR(50) NULL,
    CONSTRAINT customers_pk PRIMARY KEY (customer_id)
);
```

2. Import Data

Then we import the data using COPY or pgAdmin 4 GUI tools

```
-- Import Data Sellers
COPY sellers(
    seller_id,
    seller_zip_code_prefix,
    seller_city,
    seller_state
)
FROM 'E:\Rakamin Academy\JAP Program\Mini P
DELIMITER ','
CSV HEADER;
```

3. Data Cleaning

Data cleaning are mostly done in geolocations table where we change the special characters in City, drop duplicate rows, and input new geolocations form customers and sellers.

```
-- Clean Original Geolocation
CREATE TABLE geolocations_cleaned_city_state AS
SELECT geolocation_zip_code_prefix, geolocation_lat, geolocation_lng,
REPLACE(REPLACE(REPLACE(
    TRANSLATE(TRANSLATE(TRANSLATE(TRANSLATE(
    geolocation_city, 'E',',',',', ' '), ' ', ' '),
    'é,ê, 'e,e'), 'á,â,ã, 'a,a,a'), 'ô,ó,õ, 'o,o,o'),
    'ç, 'c'), 'ú,û, 'u,u'), 'í, 'i'),
    '4o', '4o'), 'x', ' '), '%26apos%3b', ' ')
) AS geolocation_city, geolocation_state
from geolocations_original;
```

```
CREATE TABLE geolocations_final AS
WITH geolocations_cleaned_filtered AS (
    SELECT
        geolocation_zip_code_prefix,
        geolocation_lat,
        geolocation_lng,
        geolocation_city,
        geolocation_state
    FROM (
        SELECT *,
            ROW_NUMBER() OVER (PARTITION BY geolocation_zip_code_prefix) AS ROW_NUMBER
        FROM geolocations_cleaned_city_state
    ) AS TEMP
    WHERE ROW_NUMBER = 1
),
```

4. Entity Relationship Diagram

ERD is created utilizing the Primary Key and Foreign Key of each table that we specified in the table making process, making them have relationship of Parent and Child Table as shown in the diagram on the right.

