E-Commerce Analysis

1. Project Overview

This project analyses real world Retail and e-commerce dataset has 100k orders from 2016 to 2018 made at multiple uncover trends in revenue, customer behaviour, product performance, seller performance, and delivery operations.

The main objective was to transform raw e-commerce data into actionable insights that can help management improve efficiency, reduce delivery delay, and enhance customer's experience.

2. Tools & Techniques

SQL

- Extracted data from PostgreSQL.
- Use JOINS for combining multiple tables

Python (Pandas, Matplotlib, Seaborn, NumPy)

- Performed EDA and data cleaning.
- Handled missing values and inconsistent data types.
- Generated exploratory insights and visualizations.
- Prepared datasets for Power BI.

Power BI

- Created an interactive dashboard showing:
  - Revenue month-wise
  - Order Delivery Delayed
  - Product-wise revenue performance
  - Payment Type trends
  - Repeated vs New Customers behaviour

3. Exploratory Data Analysis

We began with preparation and cleaning in Python:

- **Data Loading:** Importing the dataset from Database using Pandas.
- **Initial Exploration:** Used dfs.info() to check structure and dfs.discribe() summary of data.

	customer_zip_code_prefix	payment_installments	payment_value	order_item_id	price	freight_valu
count	119131.000000	119128.000000	119128.000000	118304.000000	118304.000000	118304.00000
mean	35033.713148	2.941366	172.750609	1.196553	120.643167	20.03150
std	29823.412428	2.777850	267.784834	0.699505	184.110764	15.83604
min	1003.000000	0.000000	0.010000	1.000000	0.850000	0.00000
25%	11250.000000	1.000000	60.850000	1.000000	39.900000	13.08000
50%	24240.000000	2.000000	108.165000	1.000000	74.900000	16.28000
75%	58475.000000	4.000000	189.250000	1.000000	134.900000	21.18000
max	99990.000000	24.000000	13664.080000	21.000000	6735.000000	409.68000

customer_unique_id	customer_zip_code_prefix	customer_city	customer_state	
ad7e2fa70d0db12bce950350ebc0e242	13177	sumare	SP	02e0c2efb6d9bb2a95c
ad7e2fa70d0db12bce950350ebc0e242	13177	sumare	SP	02e0c2efb6d9bb2a95c
ad7e2fa70d0db12bce950350ebc0e242	13177	sumare	SP	02e0c2efb6d9bb2a95c
ad7e2fa70d0db12bce950350ebc0e242	13177	sumare	SP	02e0c2efb6d9bb2a95c
ad7e2fa70d0db12bce950350ebc0e242	13177	sumare	SP	02e0c2efb6d9bb2a95c
6a05bcfd431dfae78ac8e99ea5e9c666	36415	congonhas	MG	02e151e3ad2aa3f8e99
6a05bcfd431dfae78ac8e99ea5e9c666	36415	congonhas	MG	02e151e3ad2aa3f8e99
3a3c4efd9d7b18393b6d774b9fc5a0da	3460	sao paulo	SP	02e15a348e5f7ae74cd2
3a3c4efd9d7b18393b6d774b9fc5a0da	3460	sao paulo	SP	02e15a348e5f7ae74cd2

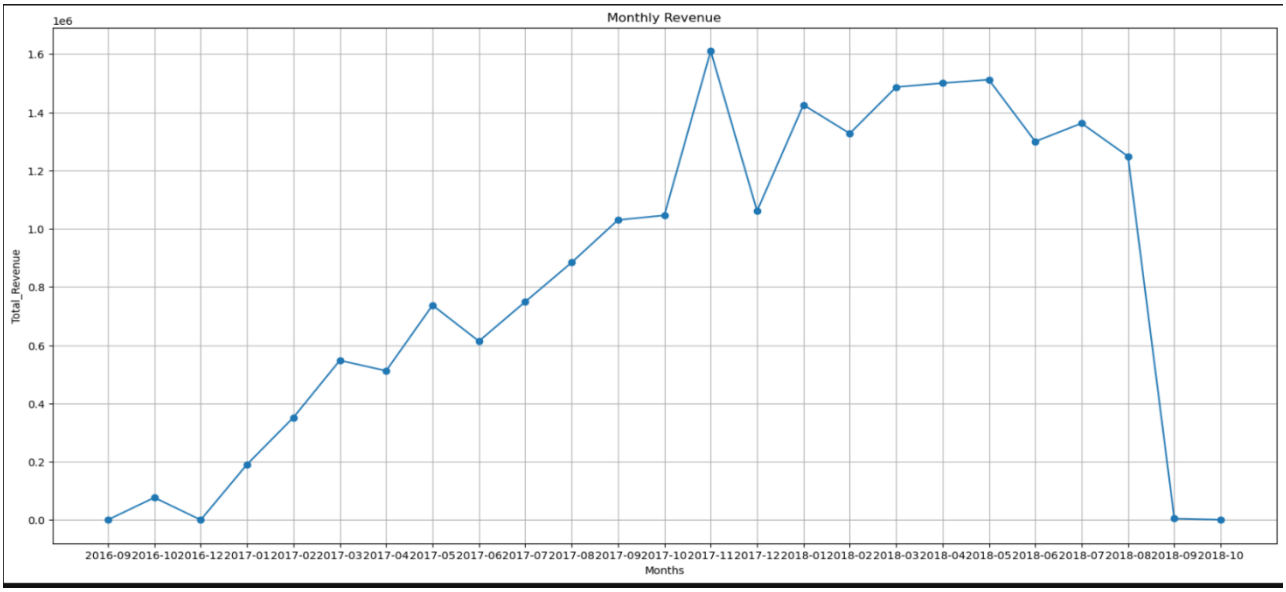
- **Missing Data Handling:** Checked for null values using dfs. isnull() to ensure smooth and uninterrupted in analysis.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 119131 entries, 0 to 119130
Data columns (total 27 columns):
 #   Column                         Non-Null Count   Dtype
---  ------                         --------------   -----
 0   customer_id                    119131 non-null  object
 1   customer_unique_id             119131 non-null  object
 2   customer_zip_code_prefix       119131 non-null  int64
 3   customer_city                  119131 non-null  object
 4   customer_state                 119131 non-null  object
 5   order_id                       119131 non-null  object
 6   order_status                   119131 non-null  object
 7   order_purchase_timestamp       119131 non-null  object
 8   order_delivered_customer_date  115718 non-null  object
 9   payment_type                   119128 non-null  object
 10  payment_installments           119128 non-null  float64
 11  payment_value                  119128 non-null  float64
 12  order_item_id                  118304 non-null  float64
 13  product_id                     118304 non-null  object
 14  seller_id                      118304 non-null  object
 13  product_id                     118304 non-null  object
 14  seller_id                      118304 non-null  object
 15  price                          118304 non-null  float64
 16  freight_value                  118304 non-null  float64
 17  review_id                      117326 non-null  object
 18  review_score                   117326 non-null  float64
 19  review_creation_date           117326 non-null  object
 20  product_category_name_pt       116595 non-null  object
 21  product_weight_g               118284 non-null  float64
 22  seller_zip_code_prefix         118304 non-null  float64
 23  seller_city                    118304 non-null  object
 24  seller_state                   118304 non-null  object
 25  product_category_name_original 116570 non-null  object
 26  product_category_name_english  116570 non-null  object
dtypes: float64(8), int64(1), object(18)
memory usage: 24.5+ MB
```

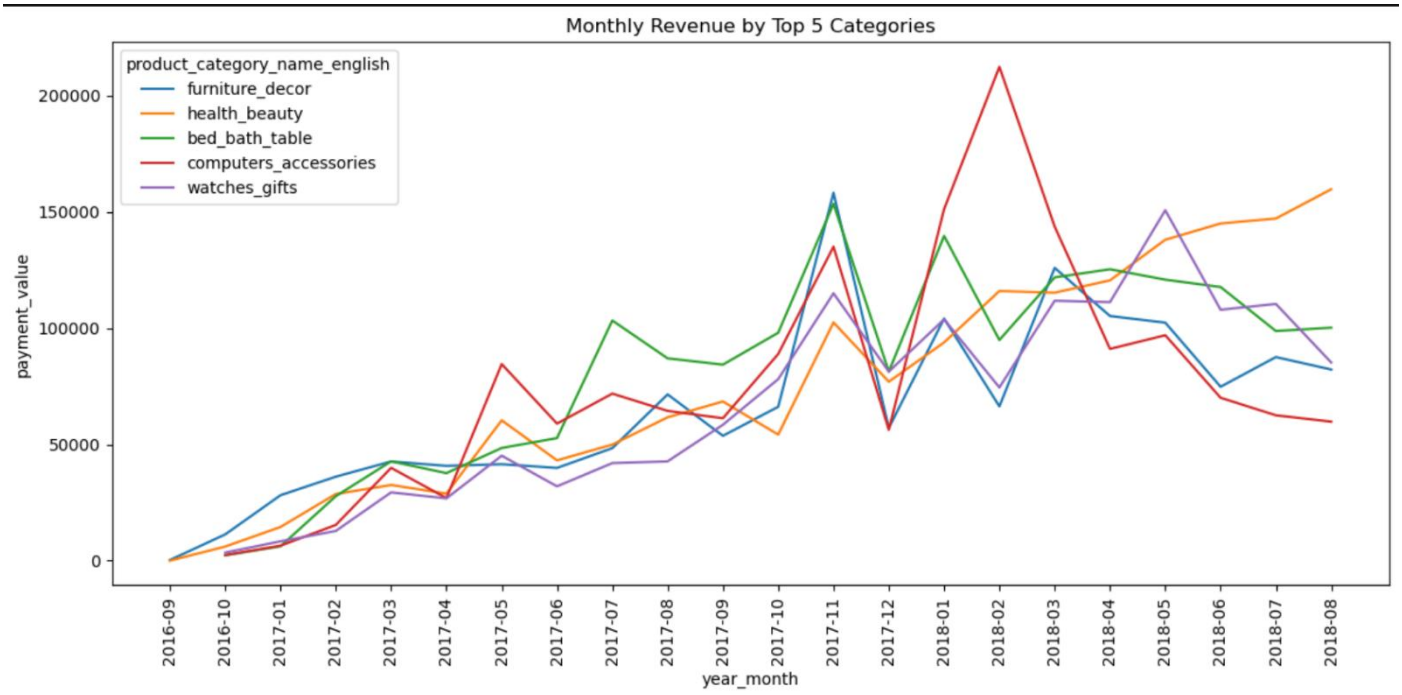
4. Data Analysis using Python

We performed structured analysis in python tp answer kye business questions:

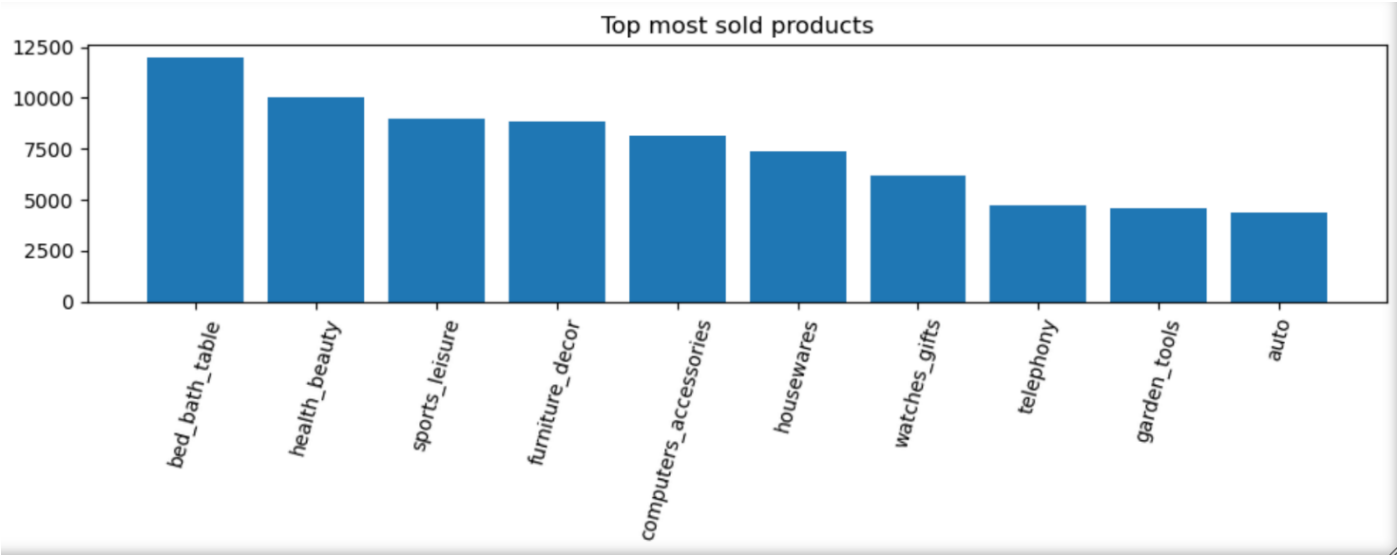
(1) Monthly Revenue -: Analysed month-over-month revenue increases and decreases.



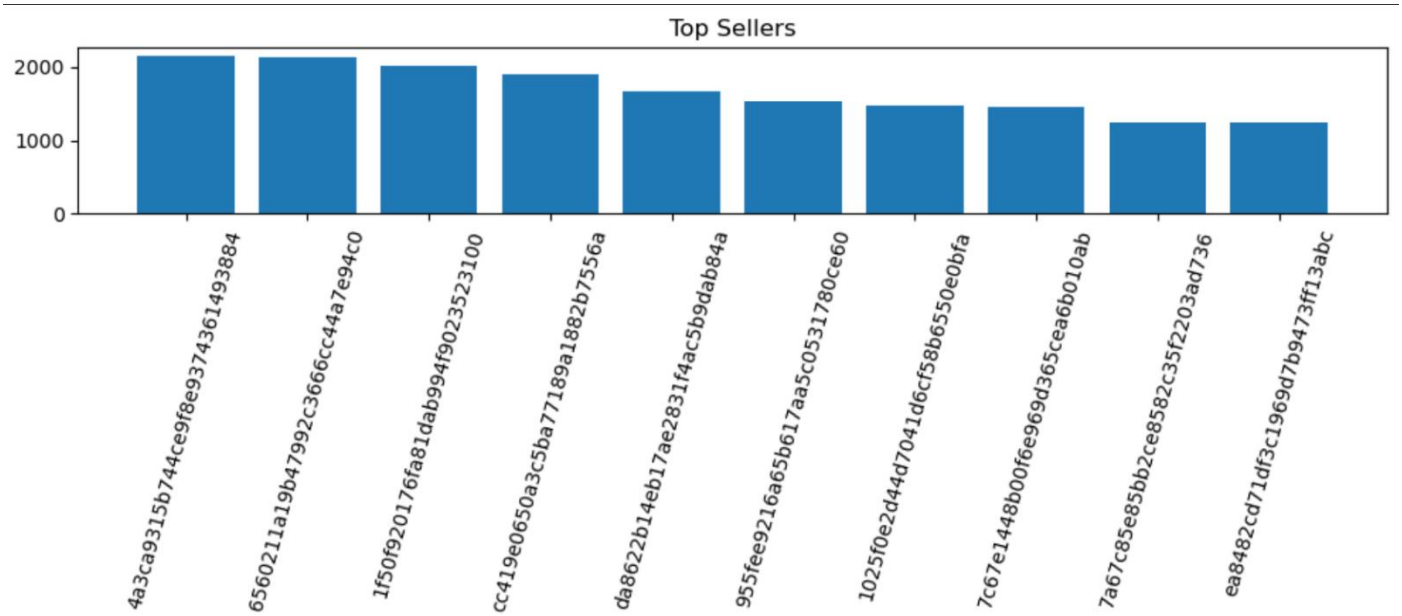
(2) Revenue by Top Products: - Identified products that played an important role in monthly revenue growth.



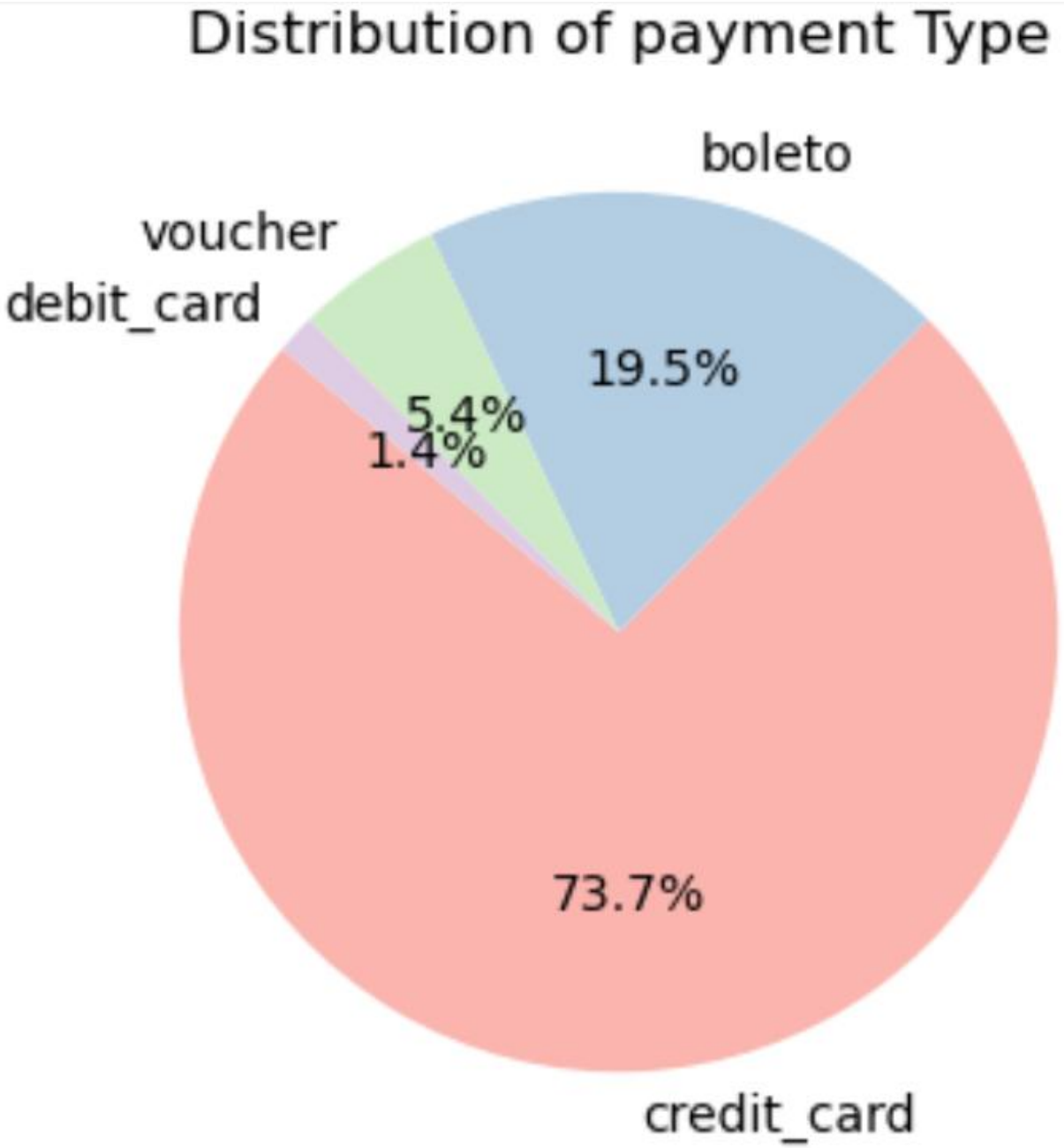
(3) Product Sold:- Identified the highest-performing product categories by sales.



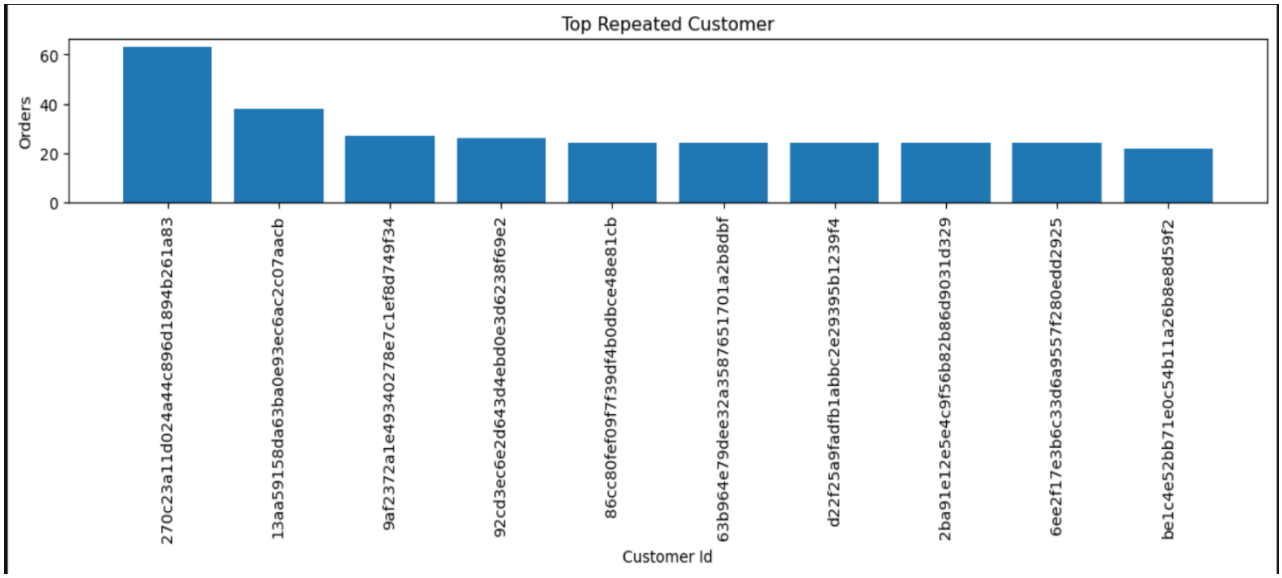
(4) Top Sellers: Analysed top-performing sellers across peak and off-peak seasons.



(5) Payment Types: Analyzed which payment types were most popular among customers.

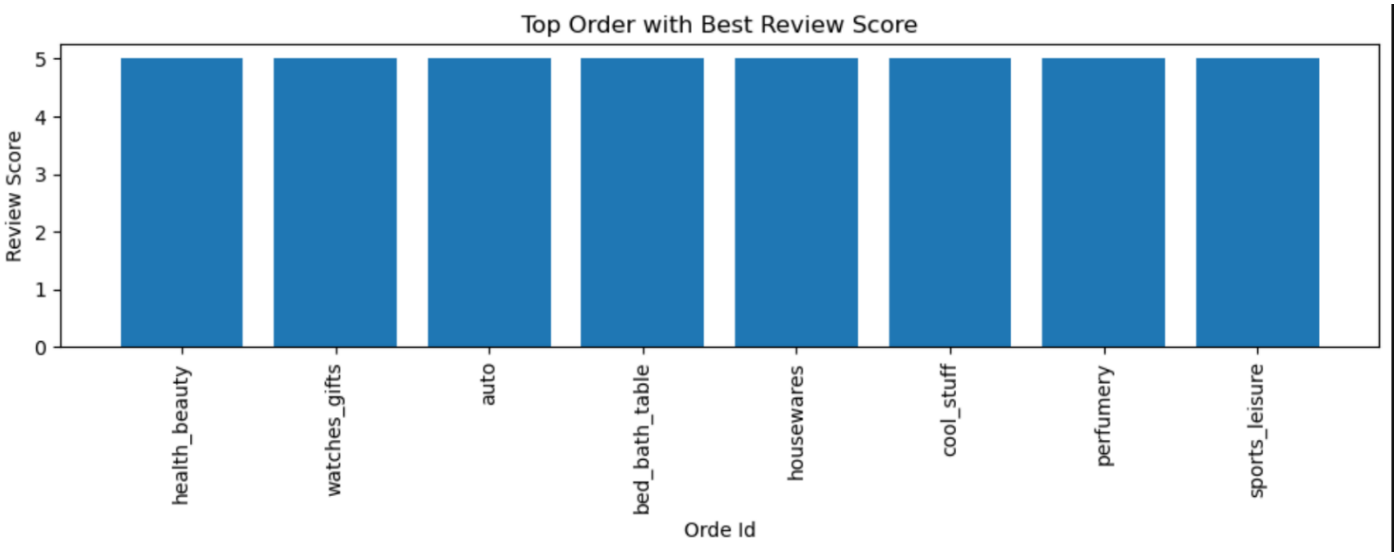


(6) Repeated Customers: Whether bills are paid, pending or failed as given by a patient.

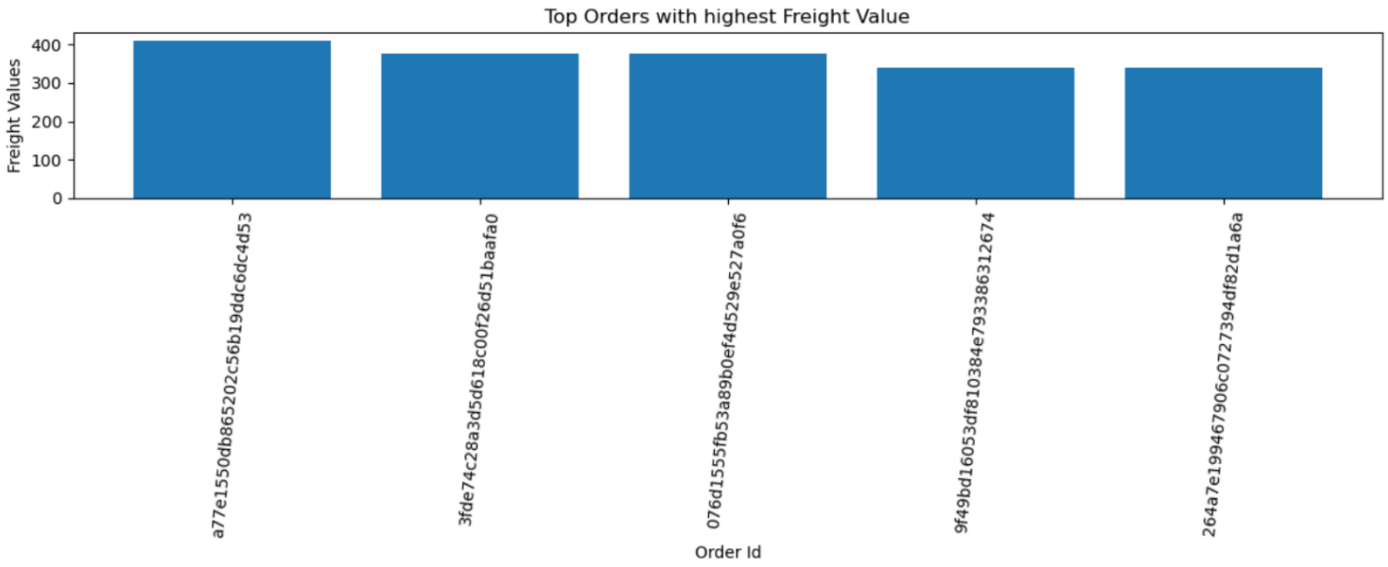


(7) Orders by Delivery Time: Identified orders with the longest delivery times.

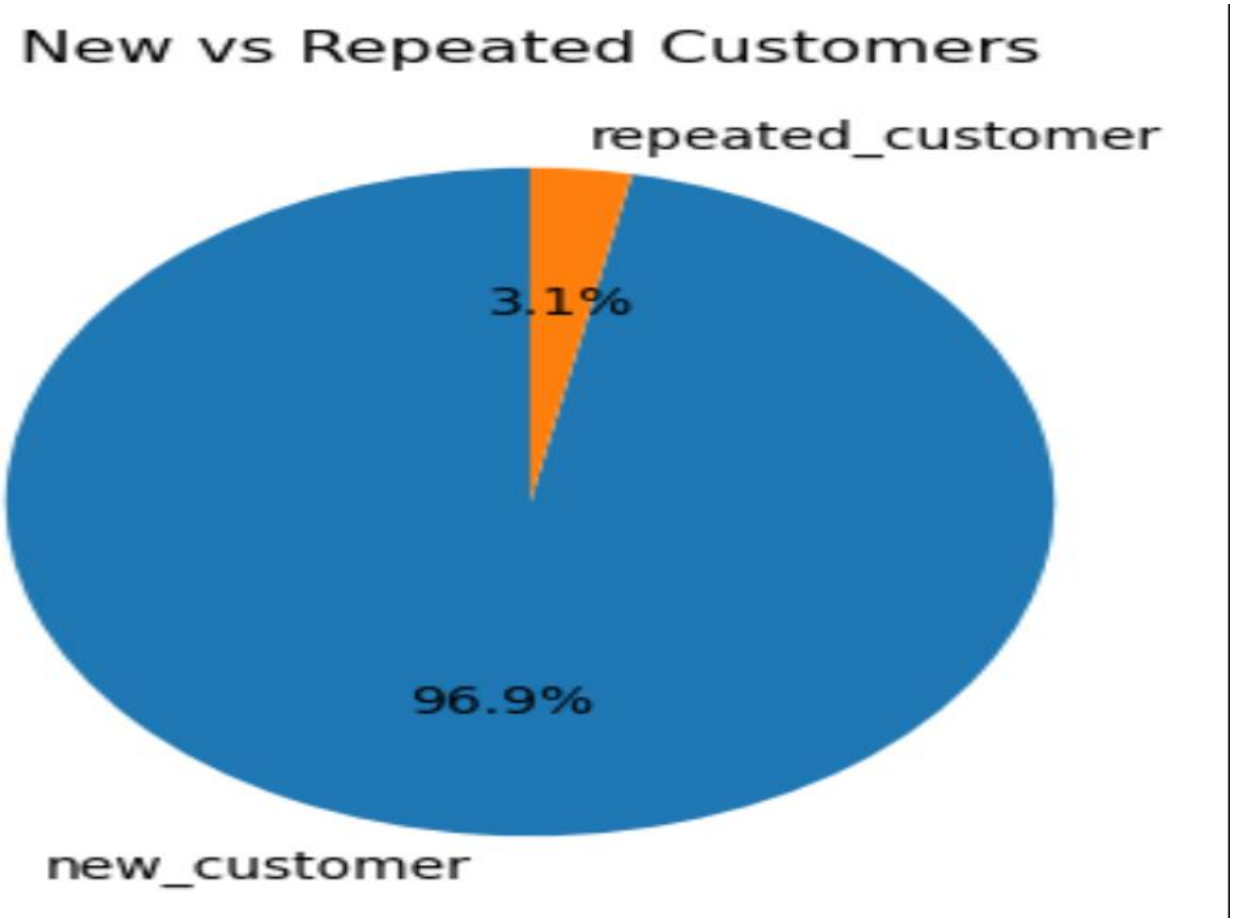
(8) Orders by Review Scores: Identified the order with the highest review score.



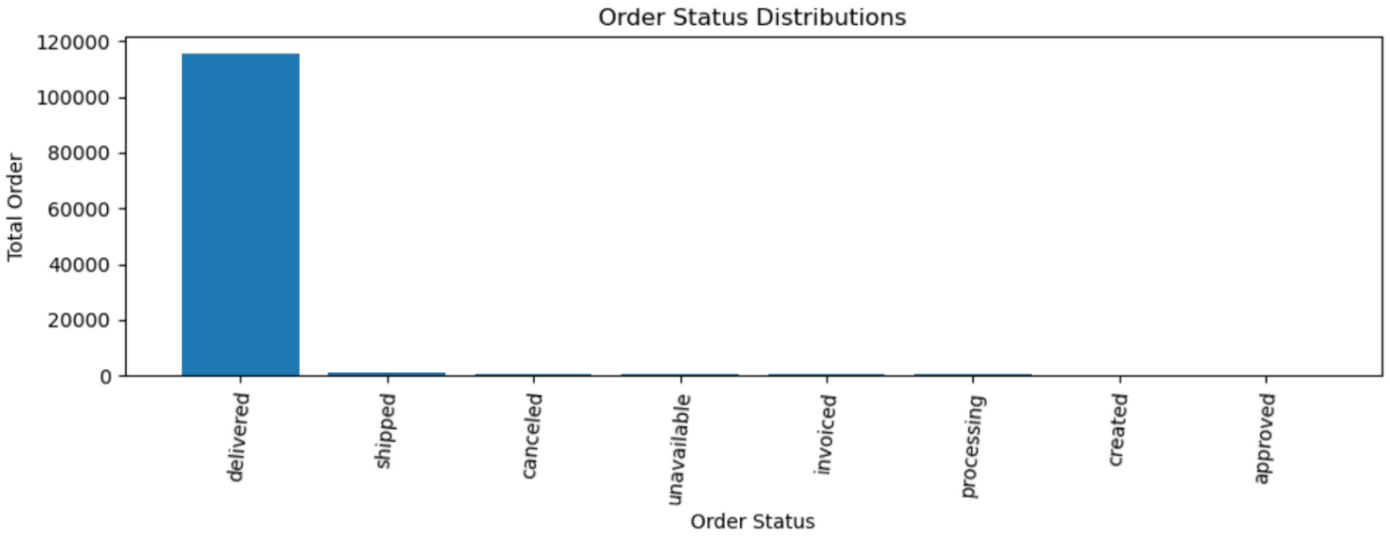
(9) Freight Value: Analysed Orders with highest Freight Cost.



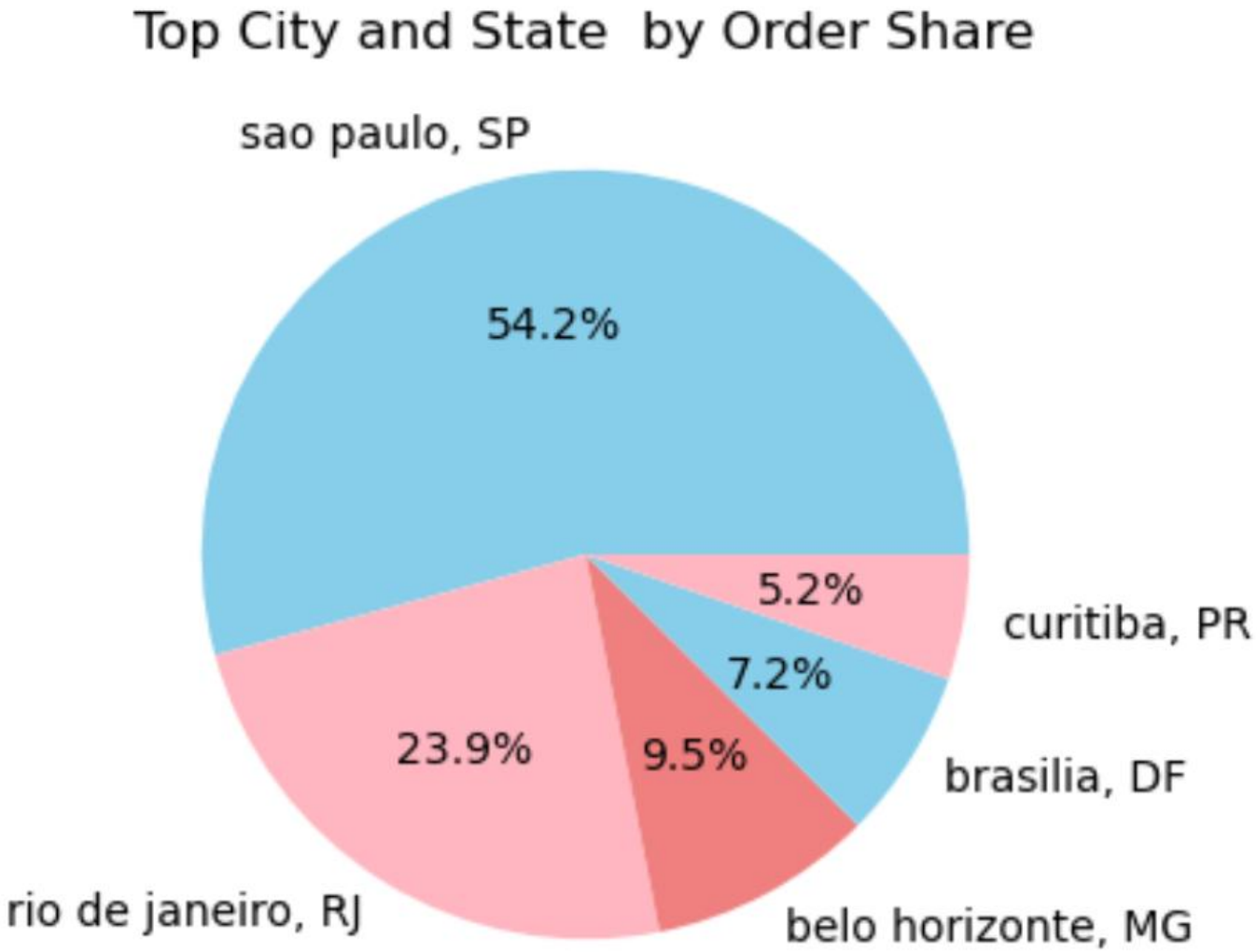
(10) New Customers vs Repeated Customer –: Find the ratio of new customer and repeated customer.



(11) Order Status: Analysed the distribution of Order Status(Delivered, Shipped, Cancelled, Unavailable, Invoiced, Processing, Created, Approved).

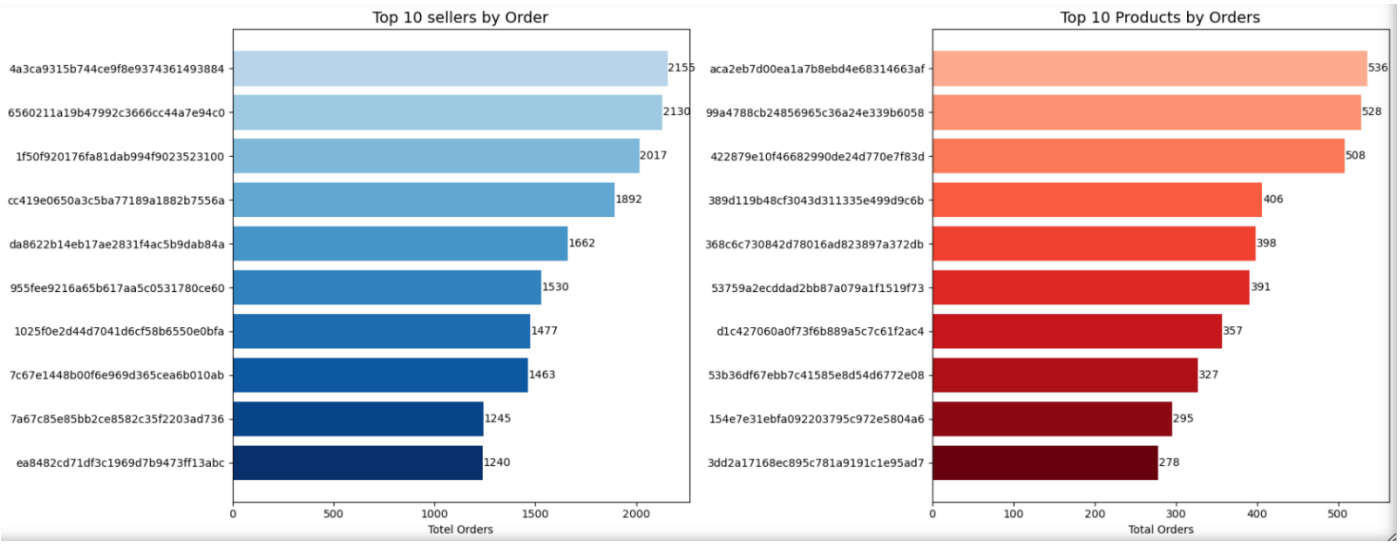


(12) Customers by City & State: Analysed the cities and states with the highest number of customer orders.



(13) Product weight vs Freight value: Analysed whether higher or lower product weights increase or decrease freight value.

(14) Best Seller vs Best Product: Compared products and sellers to determine top performers by order count.



5. **Using JOINS for Combining multiple tables with PostgreSQL:** Use LEFT and RIGHT JOIN for joining multiple tables and create a unified Table for performing Exploratory Data Analysis.

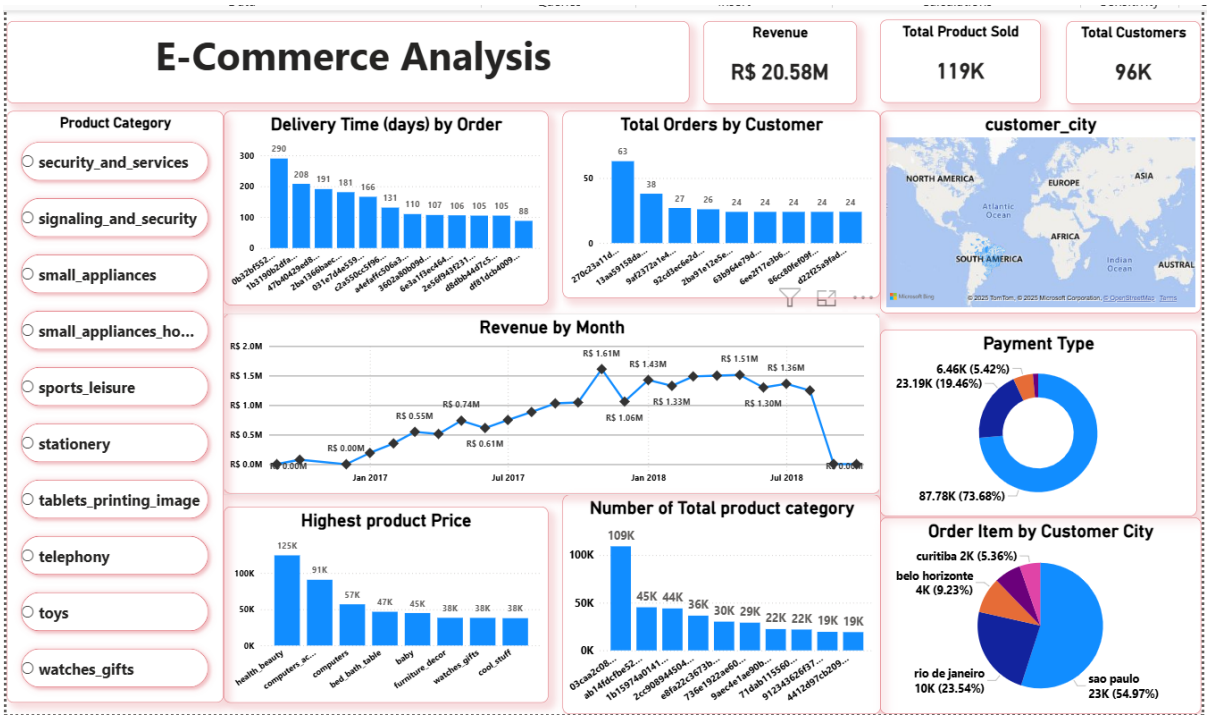
e_commerce/postgres@PostgreSQL 17

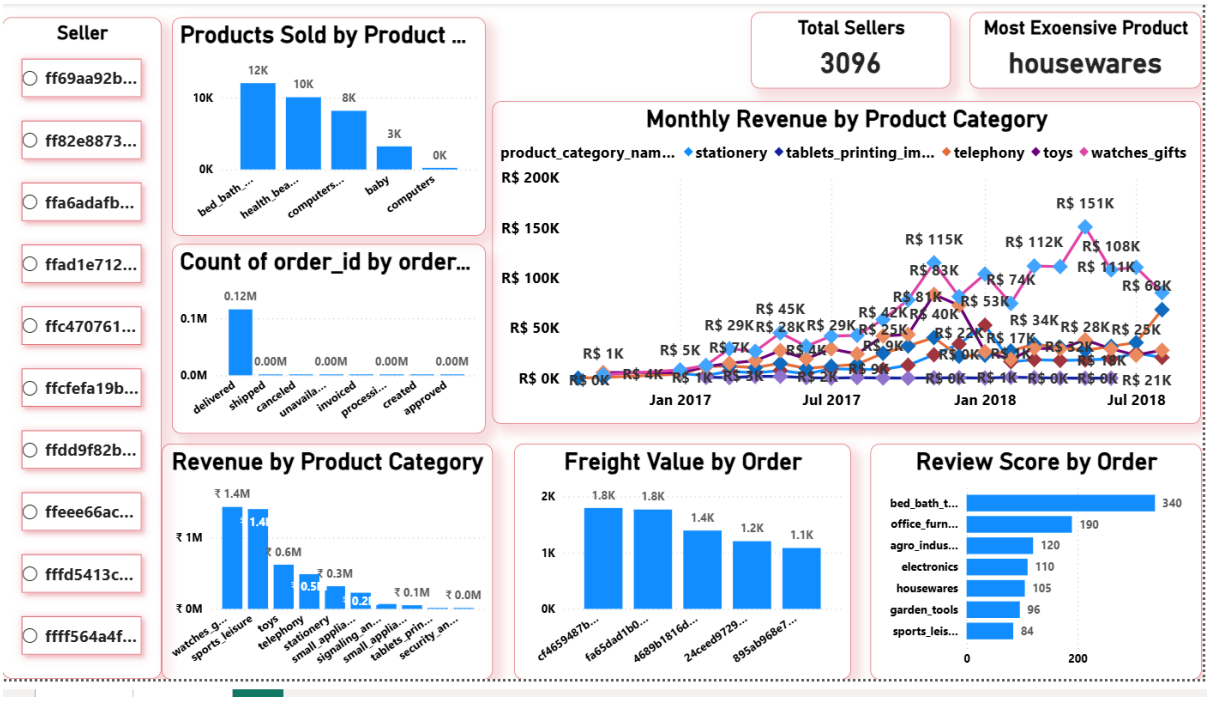
No limit

Query | Query History

```sql
CREATE TABLE customer_order_payments as
SELECT c.*, o.order_id, o.order_status, o.order_purchase_timestamp,
o.order_delivered_customer_date,
op.payment_type,op.payment_value,op.payment_installments,
ot. order_item_id,ot.product_id,
ot.seller_id,ot.price,ot.freight_value,
r.review_id,r.review_score,r.review_creation_date,
p.product_category_name as product_category_name_pt,
p.product_weight_g,
s.seller_zip_code_prefix,s.seller_city,s.seller_state,
cn.product_category_name as product_category_name_original,
cn.product_category_name_english
FROM customers c
LEFT JOIN orders o
ON c.customer_id = o.customer_id
LEFT JOIN order_payments op
ON o.order_id = op.order_id
LEFT JOIN order_item ot
ON o.order_id = ot.order_id
LEFT JOIN order_reviews r
ON ot.order_id = r.order_id
LEFT JOIN products p
ON ot.product_id = p.product_id
LEFT JOIN sellers s
ON ot.seller_id = s.seller_id
LEFT JOIN product_category_name_translation cn
ON p.product_category_name = cn.product_category_name;
```

6. Dashboard:





7. Business Recommendation:

- Customer retention is critically low (only **3.1% repeat customers**); implement loyalty programs, repeat-order discounts, and personalized offers.
- Protect top-performing sellers with better visibility and operational support to maintain revenue stability.
- Rely on **median-based metrics** rather than averages for pricing and revenue decisions to avoid distortion from extreme orders.
- Prioritize monitoring and resolving **extreme delivery delays**, as a small number of late orders can disproportionately damage customer satisfaction.

Github Link:- [Rafia3023/e-commerce-retail-analysis-python-sql-powerbi]