E-commerce Commercial Analytics Dashboard with PostgreSQL + Power BI

Lauri Lehto

2. Abstract (5–8 lines) **1**
3. Business goal and questions **2**
4. Data description **2**
4.1 Data source 2
4.2 Tables used (short list) 2
5. Data pipeline and architecture **3**
6. Data modeling in Power BI (VERY IMPORTANT section) **4**
6.1 Star schema explanation 4
Fact Tables 4
Dimension Tables 4
6.2 Relationships (include screenshot of Model view) 5
6.3 Modeling problem 5
7. Measures (DAX KPIs) **6**
8. Dashboard design (pages + visuals) **7**
Commercial Overview 7
Page 2: Logistics & Payments 9
Delay Rate by Customer State (top delay rate states, not all states) 9
9. Findings and interpretation (most important narrative section) **11**
Commercial Side 11
Logistics and Payments 11
10. Limitations **12**
11. Future work (1 paragraph) **12**
12. Conclusion **13**
Appendix **13**
Measures and columns 13
Measures 13
Column 14
Tables 15

2. Abstract (5–8 lines)

This project demonstrates an end-to-end data analytics pipeline using a publicly available Brazilian e-commerce dataset by Olist. The objective is to transform raw transactional data into meaningful business insights through structured data modeling and visualization.

PostgreSQL is used for data storage, transformation, and view creation, while Power BI is applied for data modeling, DAX-based calculations, and interactive dashboard development. The analysis focuses on key commercial metrics such as revenue, order behavior, customer distribution, and delivery performance. The project highlights how business intelligence tools can support data-driven decision-making in e-commerce environments.

3. Business goal and questions

The primary business goal of this project is to analyze e-commerce performance and identify key factors that influence revenue, customer behavior, and operational efficiency. The analysis focuses on understanding delivery performance, regional demand patterns, purchasing behavior over time, and product category performance.

Specifically, the project seeks to answer the following questions:

- How frequently do delivery delays occur, and how do they impact overall performance?
- Which states generate the highest sales volume and revenue?
- At what times do customers most actively make purchases?
- What types of products are most commonly purchased?
- Which product categories contribute the most to total revenue?

By addressing these questions, the project aims to provide actionable insights that could support better inventory planning, logistics optimization, and strategic business decisions.

4. Data description

4.1 Data source

The dataset used in this project is the **Olist Brazilian E-Commerce Public Dataset**, which is publicly available on **Kaggle**. The dataset contains real transactional data from a Brazilian e-commerce platform and includes information on orders, customers, products, payments, sellers, and delivery details.

The data covers multiple years of operations and provides a comprehensive view of customer purchasing behavior, order processing, logistics performance, and product category distribution. As an open-source dataset, it is widely used for educational, analytical, and business intelligence projects.

4.2 Tables used (short list)

To support efficient analysis and clear relationships within the data model, the dataset was structured into several core tables in PostgreSQL and later connected in Power BI.

CustomersDim
A separate customer dimension table was created to improve relationships between customers and orders. The original public customers table contains multiple entries per customer due to unique customer_id values per order, even when the same person places

multiple purchases. To avoid duplication and enable a clean one-to-many relationship with the orders table, a distinct customer dimension (CustomersDim) was created using unique customer identifiers and selected attributes (e.g., state and city). This improves filtering, aggregation accuracy, and model clarity in Power BI.

public customers
This table is retained in its original form to preserve the raw source data. It contains detailed customer-level information, including geographic attributes and customer identifiers associated with each order.

public order_items
This table contains detailed information about each item within an order. Since one order may contain multiple products, this table enables product-level revenue analysis and category-based aggregation.

public order_payments
This table includes payment-related information for each order, such as payment type and payment value. It enables analysis of total payment amounts and payment method distribution.

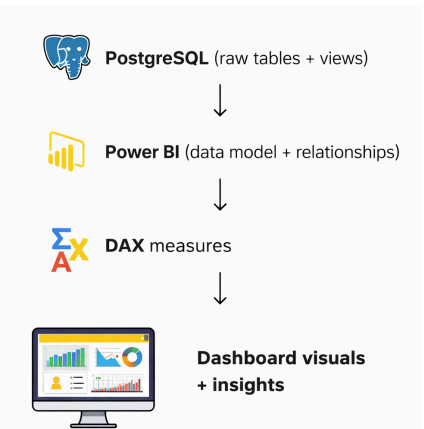
public orders
This is the central fact table of the model. It contains order-level information, including timestamps, order status, and delivery dates. Most key measures such as total orders, delay rate, and delivery performance are calculated based on this table.

public products
This table provides product-level metadata, including product category information. It enables analysis of which products and categories generate the most revenue and sales volume.

5. Data pipeline and architecture

The project follows a structured data pipeline where PostgreSQL is used as the primary data storage and transformation layer. It enables efficient handling of raw tables, relational joins, SQL-based filtering, and the creation of views to prepare clean, analysis-ready datasets. This ensures data consistency and reproducibility before visualization.

Power BI is used as the analytical and presentation layer. It provides a structured data model with defined relationships, supports advanced DAX measures for business calculations, and enables the creation of interactive dashboards. Together, PostgreSQL and Power BI form a scalable architecture that separates data engineering from business intelligence and insight generation.



6. Data modeling in Power BI (VERY IMPORTANT section)

6.1 Star schema explanation

The data model in Power BI follows a star schema structure. In this design, a central fact table (or multiple related fact tables) is connected to surrounding dimension tables through one-to-many relationships. This structure improves query performance, simplifies DAX calculations, and ensures clear filtering logic.

Fact Tables

Orders
The orders table acts as a central fact table at the order level. It contains transactional data such as order timestamps, status, and delivery dates. Measures such as total orders, delay rate, and delivery performance are primarily calculated from this table.

Order_Items
The order_items table represents product-level transactions. Since one order can contain multiple products, this table enables revenue analysis at a more detailed granularity (product and category level). Revenue measures are typically aggregated from this table.

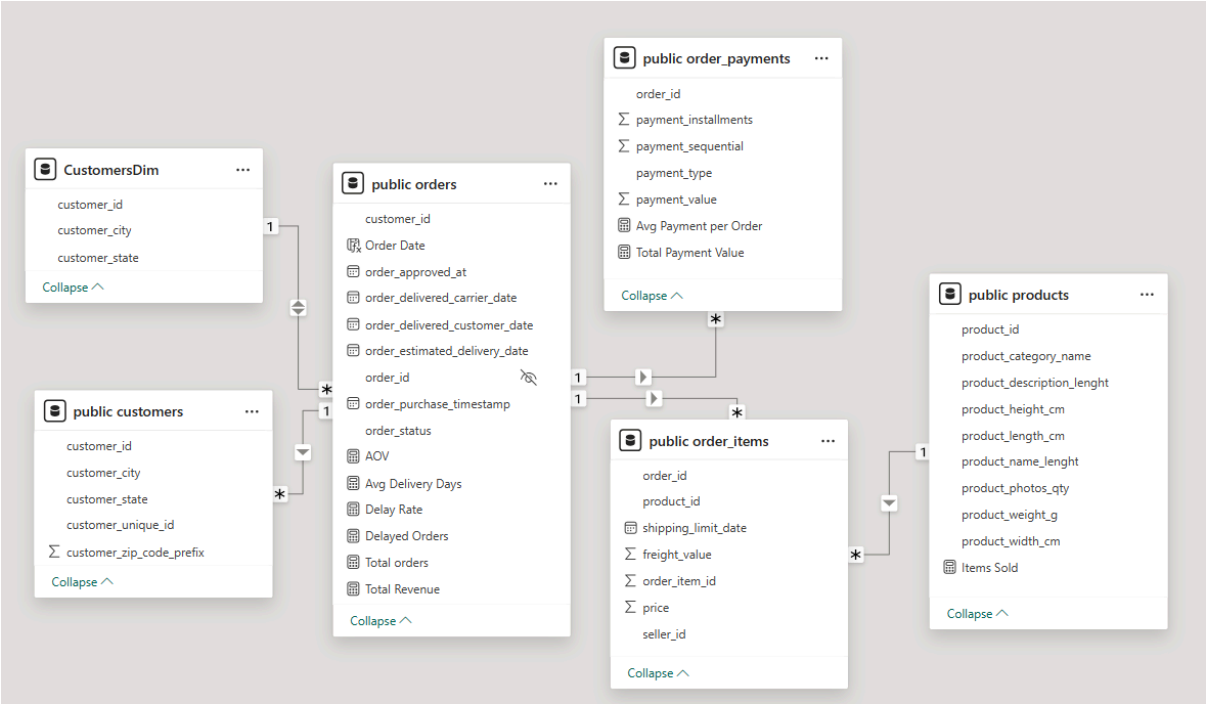
Order_Payments
The order_payments table contains payment-level data, including payment type and payment value. It allows analysis of total payment amounts and payment method distribution. One order may have multiple payment records.

Dimension Tables

CustomersDim
This dimension table contains unique customer-level attributes such as state and city. It is connected to the orders table through a one-to-many relationship (one customer → many orders). It enables geographic and customer-based analysis without duplication issues.

Products
The products table acts as a product dimension. It contains product metadata such as product category. It connects to order_items, enabling category-level revenue and product performance analysis.

6.2 Relationships (include screenshot of Model view)



The final Power BI data model follows a one-to-many structure consistent with a star schema design. The relationships are defined as follows:

- **CustomersDim (1) → orders (*)**
  One customer can place multiple orders.

- **orders (1) → order_items (*)**
  One order can contain multiple ordered items.

- **products (1) → order_items (*)**
  One product can appear in multiple order items.

- **orders (1) → order_payments (*)**
  One order can have multiple payment records.

These relationships ensure correct filtering behavior, prevent double counting, and allow measures to aggregate properly across customer, product, and payment dimensions.

6.3 Modeling problem

Initially, Power BI detected an incorrect cardinality in the relationship between customers and orders because the original customers.customer_id column contained duplicate values. This caused filtering issues in the model, where visualizations (such as state-level bar charts) appeared identical due to improper filter propagation.
To resolve this, a separate CustomersDim table was created using unique customer_id values, ensuring a proper one-to-many relationship with the orders table. After this adjustment, geographic slicing and filtering worked correctly across all related visuals.

The original customer table was unsuitable as a dimension due to duplicate IDs, so a dedicated CustomersDim table was created to enforce one-to-many relationships and correct filter propagation.

7. Measures (DAX KPIs)

Category	Measure	DAX / Logic
Revenue	Total Revenue	`SUMX(RELATEDTABLE(order_items), order_items[price])`
	Total Orders	`DISTINCTCOUNT(orders[order_id])`
	AOV	`DIVIDE([Total Revenue], [Total Orders])`
Delivery	Delayed Orders	Delivered > Estimated date
	Delay Rate	`DIVIDE([Delayed Orders], [Total Orders])`
	Avg Delivery Days	`DATEDIFF(purchase, delivered, DAY)` (average)
Payments	Total Payment Value	`SUM(order_payments[payment_value])`
	Avg Payment per Order	`DIVIDE([Total Payment Value], [Total Orders])`

These measures represent core business KPIs and are used consistently across all visuals (cards, charts, tables).

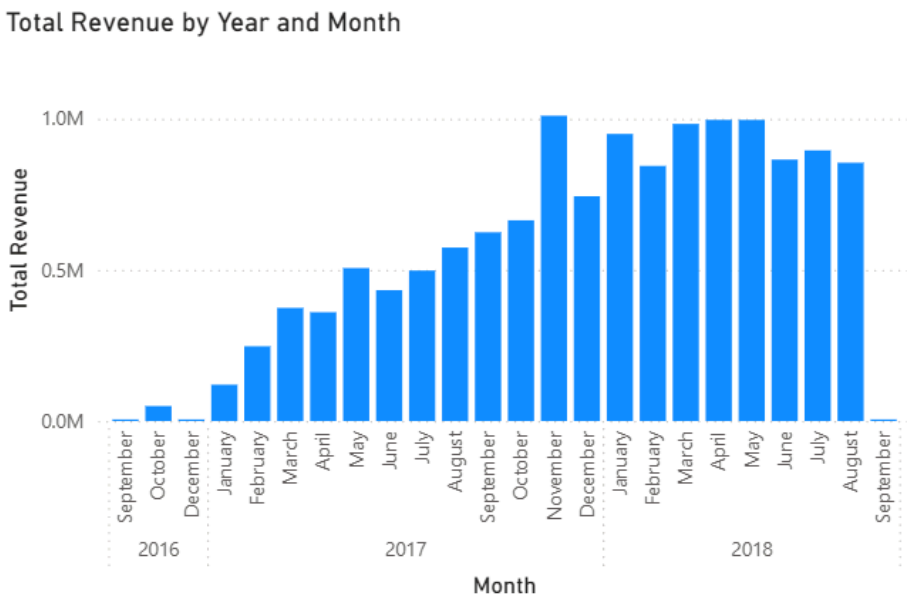
8. Dashboard design (pages + visuals)

The dashboard visuals were selected to provide a structured and business-oriented overview of performance. KPI cards (Revenue, Total Orders, AOV, and Delay Rate) are used to present the most important indicators, allowing immediate evaluation of overall business status.

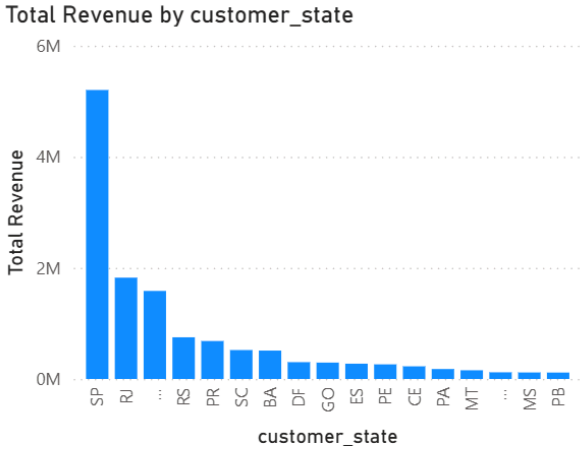
- Trend line charts are used to analyze revenue over time and detect seasonality or growth patterns.
- Bar charts support clear comparison and ranking between states and product categories, making high-performing regions and segments easy to identify.
- On the logistics and payments page, comparative charts evaluate operational performance and its relationship to revenue. Payment visuals show customer preferences and differences in average transaction values.

Overall, the visualizations prioritize clarity, comparability, and business relevance.

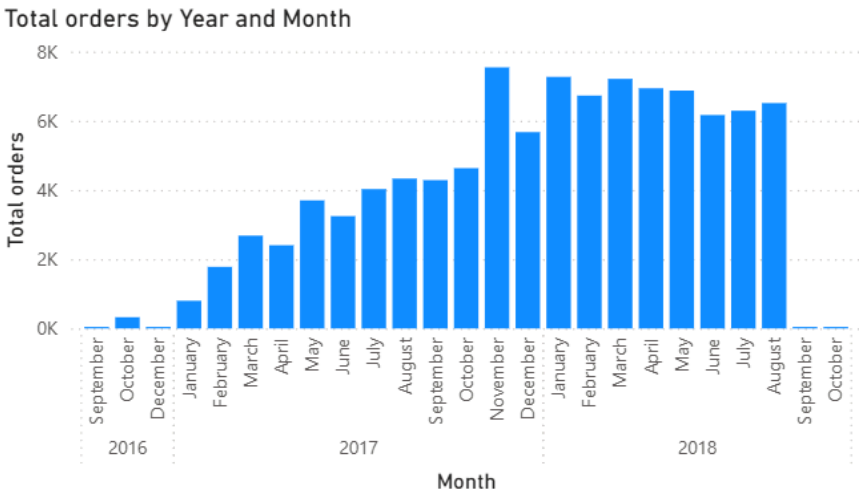
Commercial Overview



Total Revenue by Month

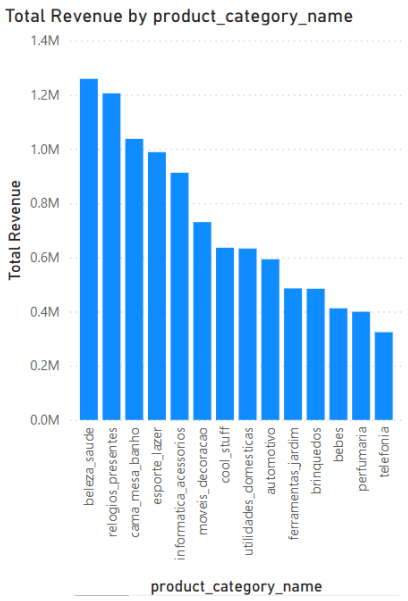


Total Revenue by Customers State



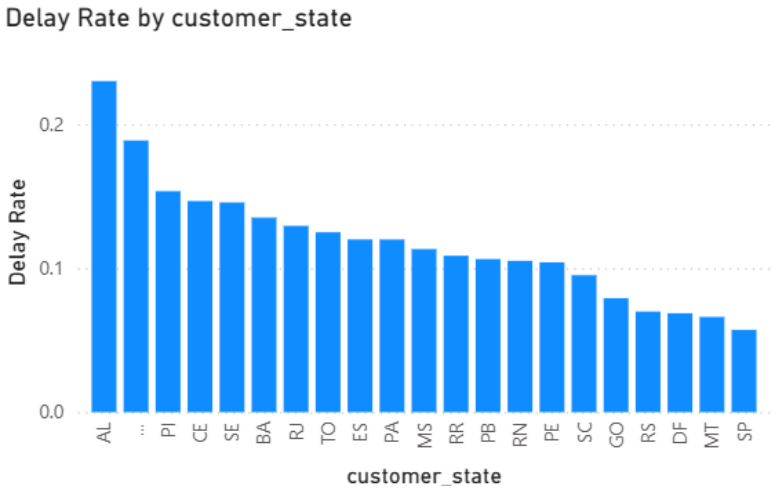
Total Orders by Month

Average Order Value (AOV) by Month

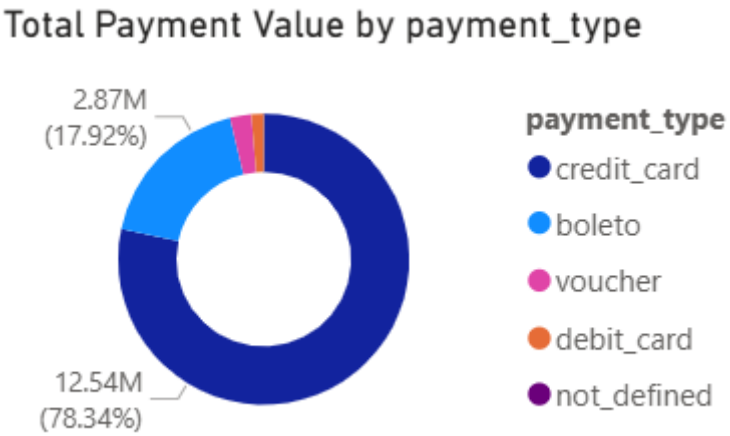


Total Revenue by Product Category Name

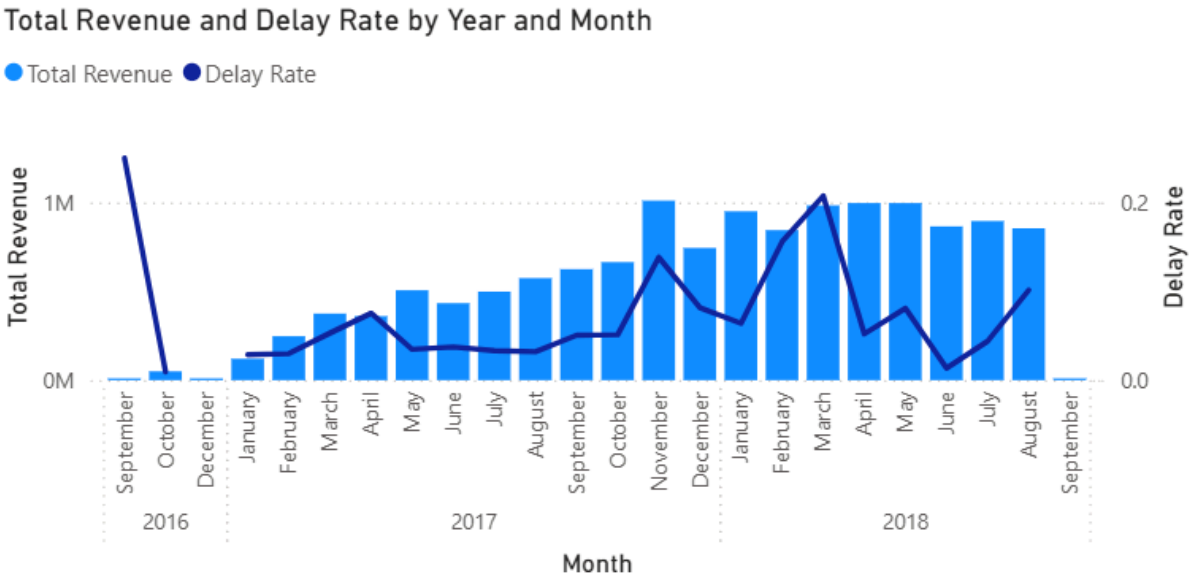
Page 2: Logistics & Payments



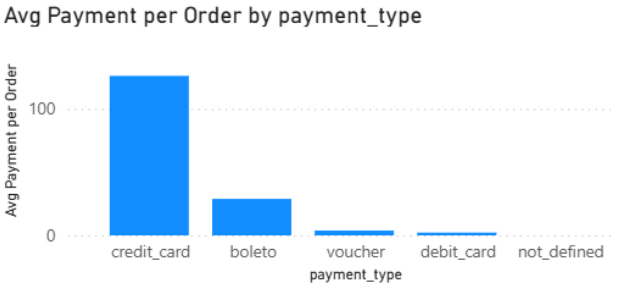
Delay Rate by Customer State (top delay rate states, not all states)



Total Payment Value by Payment Type



Total Revenue and Delay Rate by Month



Avg Payment perOrder by Payment Type

9. Findings and interpretation (most important narrative section)

Commercial Side

Total revenue ramps up strongly through 2017, reaching a peak around November 2017 about 1M. After that 2018 stays consistently high around 0.85 - 1M per month. It suggests that the business moved from a growth phase into a more stable, scaled plateau. The very small values at the end of the timeline are likely partial-month artifacts and therefore shouldn't be over-interpreted.

Total orders are similar: constant stable growth across 2017, a spike around November 2017 little under 8k. Then a mostly flat band in 2018 6–7k/month, which supports that performance in 2018 is driven by stable volume rather than fast expansion. AOV is steady during the "full-data" period roughly 130–150, implying that most revenue movement is explained by changes in order volume rather than customers spending more per order. The extreme dips at the dataset edges are consistent with incomplete months.

Revenue by customer state is highly concentrated: São Paulo (SP) dominates by a wide margin, followed by Rio de Janeiro (RJ) and Minas Gerais (MG), with a long tail after that this indicates both dependency on the Southeast and a clear target area where logistics and marketing improvements will have the biggest payoff.

Revenue by product category shows a pattern, where a small set of categories accounts for a disproportionate share of total revenue, with top 3 contributors including beleza_saude, relogios_presentes and cama_mesa_banho. meaning the fastest impact likely comes from prioritizing availability, seller quality, and delivery performance in these and other top categories in SP.

Logistics and Payments

Delay performance varies quite a lot by geography. The highest delay rates are in AL (around 0.23) and MA (around 0.19), while large-market states such as SP sit among the lowest around 0.06 (not all states are on the paragraph), which suggests delays are driven more by logistics distance constraints than by demand size. Practically, this highlights a clear "problem cluster" of states where delivery operations could benefit from targeted seller improvements.

Payment behavior is dominated by credit cards, which account for 12.54M, around 78% of total payment value, while boleto is the second major method at 2.87M and the rest are marginal, which means that the revenue reliability and customer experience are heavily tied to card-processing and card-friendly checkout. Average payment per order is much higher for credit card (around 130) than for boleto (around 35). Other methods are close to negligible, which indicates that higher-value baskets are disproportionately paid by card. This makes card UX and acceptance a direct lever for improving monetization.

When you compare total revenue vs delay rate over time, revenue rises and stays high while the delay rate climbs notably into early 2018, peaking around 0.2, implying that as volume scaled, logistics performance temporarily worsened. Then it partially recovered mid-2018 before rising up again. Growth periods correlate with operational pressure.

10. Limitations

From a technical perspective, some data transformations and cleaning steps were performed in Power BI instead of fully in PostgreSQL views. While functional for analysis, this reduces reproducibility, version control clarity, and database-level optimization. Additionally, revenue from order items and total payment values may not perfectly align due to refunds, freight costs, discounts, or split payments, which introduces reconciliation limitations.

From a data perspective, the dataset lacks important features such as customer reviews, detailed seller performance metrics, cost data, and precise geolocation variables. This restricts deeper operational diagnostics and profitability analysis. The timeframe of the dataset is also limited, making long-term trend and seasonality conclusions less robust. Furthermore, partial months at dataset edges may distort time-based metrics.

Overall, the model is suitable for business intelligence demonstration purposes, but it is not a fully production-ready analytical system.

11. Future work (1 paragraph)

If the project were expanded further, several improvements could have strengthened the analytical depth and scalability. Implementing a dedicated DateDim table would enable more advanced time intelligence features such as year-over-year comparison, rolling averages and seasonal decomposition. Also integrating a reviews table would allow analysis of customer satisfaction in relation to delivery delays and product categories, providing a stronger link between operational performance and customer experience. Finally, incorporating seller-level data would support marketplace performance analysis, enabling evaluation of seller quality, regional logistics impact, and revenue contribution by seller segment.

12. Conclusion

The end-to-end data pipeline from PostgreSQL to Power BI functioned effectively, enabling structured storage, transformation, and visualization of e-commerce data. The star schema model improved analytical clarity, and the creation of a dedicated CustomersDim table resolved cardinality issues, ensuring correct filter propagation and accurate aggregations.

The final dashboard successfully addresses both commercial and operational questions, including revenue trends, geographic concentration, product mix, delivery performance and payment behavior. The insights generated demonstrate how structured data modeling and business intelligence tools can support informed, data-driven decision-making.

Appendix

Measures and columns

Measures

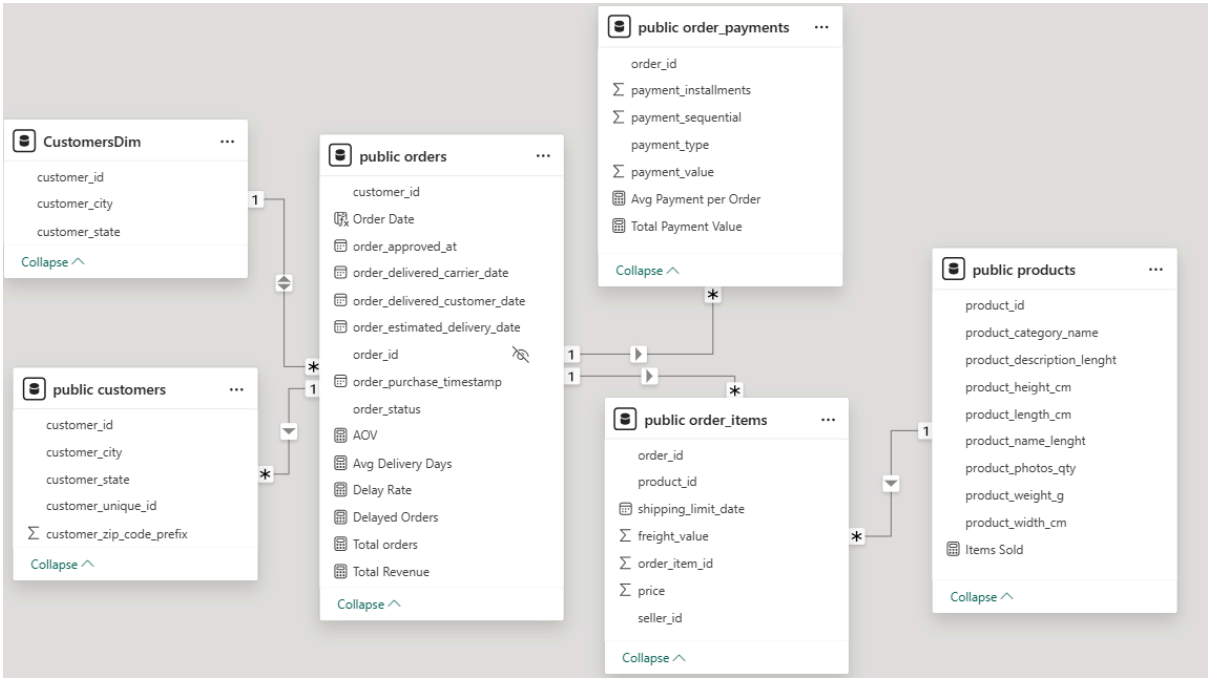
```
AOV = DIVIDE([Total Revenue],[Total orders])

Avg Delivery Days =
AVERAGEX(
   FILTER(
      'public orders',
      NOT ISBLANK('public orders'[order_delivered_customer_date])
   ),
   DATEDIFF(
      'public orders'[order_purchase_timestamp],
      'public orders'[order_delivered_customer_date],
      DAY
   )
)

Avg Payment per Order =
DIVIDE([Total Payment Value], [Total Orders])

Delay Rate =
DIVIDE([Delayed Orders], [Total Orders])

Delayed Orders =
CALCULATE(
   [Total Orders],
```

```
    FILTER(
        'public orders',
        NOT ISBLANK('public orders'[order_delivered_customer_date]) &&
        'public orders'[order_delivered_customer_date] > 'public
orders'[order_estimated_delivery_date]
    )
)

Items Sold =
COUNTROWS('public order_items')

Total orders = DISTINCTCOUNT('public orders'[order_id])

Total Payment Value =
SUM('public order_payments'[payment_value])

Total Revenue =
SUMX(
    RELATEDTABLE('public order_items'),
    'public order_items'[price]
)
```

Column

```
Order Date =
DATE(
    YEAR('public orders'[order_purchase_timestamp]),
    MONTH('public orders'[order_purchase_timestamp]),
    DAY('public orders'[order_purchase_timestamp])
)
```

Tables



Additionally Created table

```
CustomersDim =
DISTINCT(
    SELECTCOLUMNS(
        'public customers',
        "customer_id", 'public customers'[customer_id],
        "customer_state", 'public customers'[customer_state],
        "customer_city", 'public customers'[customer_city]
    )
)
```