Olist E-Commerce – A Case Study  
Portfolio Project | SQL-Based Business Analytics | Brazil E-Commerce (2016–2018)

**Dataset**: Brazilian E-Commerce Public Dataset (source: [Kaggle](https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce/data?select=olist_order_items_dataset.csv))  
**Tool Used**: PostgreSQL (via DBeaver)  
**Prepared by**: Kushagra Sudhir  
**Format**: SQL-only, Business analytics report  
**Objective**: Generate actionable insights across customer behavior, product sales, delivery performance, and operational gaps.

**Table of Contents**

1. Executive Summary
2. Dataset Overview
3. Methodology
4. Section 1: Monthly Order Volume & Cancellation Trends
5. Section 2: Top Product Categories by Revenue and AOV
6. Section 3: Customer Retention & Repeat Behavior
7. Section 4: Payment Method Trends & AOV Analysis
8. Section 5: Delivery Delays by Region
9. Section 6: Review Score Analysis vs Delivery Performance
10. Section 7: Revenue Lost Due to Order Cancellations
11. Section 8: Strategic Recommendations & Appendix
12. Appendix: SQL Queries Used

**Executive Summary**

This case study analyzes transactional and behavioral data from the Brazilian e-commerce platform Olist, using advanced SQL techniques. The goal is to derive data-driven insights around customer retention, payment trends, revenue loss, review performance, and delivery inefficiencies.

The project simulates a real-world analytics engagement, executed using only SQL, without the use of BI tools or scripting languages. Over 100,000 orders spanning 2016–2018 were analyzed through seven structured Questions, with insights that could drive strategic decisions in operations, customer retention, logistics, and seller management.

**Dataset Overview**

This dataset captures over 100,000 real-world e-commerce orders made between 2016 and 2018 across multiple online marketplaces in Brazil. It offers a comprehensive view of each order, spanning key dimensions such as order status, pricing, payment methods, freight logistics, customer geography, product attributes, and customer review behavior.

The data has been fully anonymized to ensure privacy. The dataset was made publicly available by **Olist**, a leading Brazilian e-commerce enabler that empowers small businesses across the country to sell online. Through the Olist Store, merchants can access major marketplaces with a single contract and rely on Olist’s logistics partners for streamlined fulfillment and delivery.

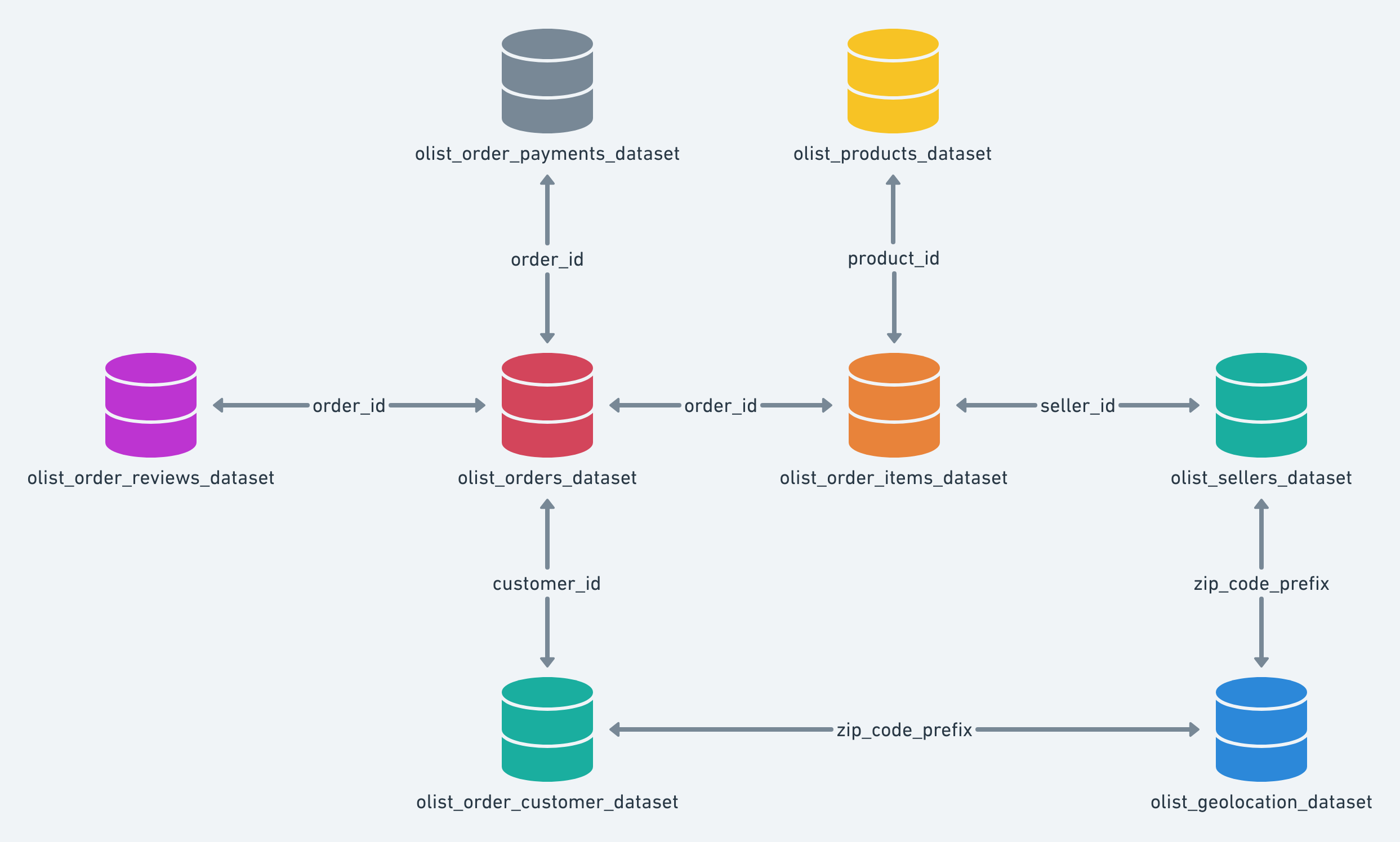
The dataset, includes:

* ~100K unique orders
* ~96K unique customers
* 8 key relational tables including orders, order items, customers, payments, reviews, sellers, and products

**Key Tables Used:**

* ***orders***: Order lifecycle details (timestamps, status)
* **order\_*items***: Line-level product and price info per order
* ***customers***: Customer identifiers and geographies
* ***order*\_*payments***: Payment methods and value
* ***order*\_*reviews***: Customer feedback with review scores
* ***products*** + ***product*\_*translation***: Product metadata and translated categories
* ***sellers***: Seller geography and ID

**Visual Schema:**



Note: Olist.geolocation\_dataset was not used in this project

**Methodology**

Each Question was treated as a standalone SQL case using:

* **CTEs** for stepwise clarity
* **Window functions** (ROW\_NUMBER, LAG) for trend and behavior tracking
* **FILTER and CASE** logic for conditional metrics
* **DATE\_TRUNC and arithmetic** for time-based aggregation
* **JOINs** across multiple tables for relational insights

Unlike BI dashboarding, this analysis emphasizes **clean code logic, thoughtful metric selection, and sharp business interpretation**, showcasing SQL as an end-to-end analysis tool.

**Section 1: Monthly Order Volume & Cancellation Trends**

**Question**

What does the monthly trend of total orders and cancellations reveal about Olist’s growth and operational efficiency? Are there any key inflection points?

**SQL Methodology**

* Used the orders table
* Cleaned timestamp using DATE\_TRUNC('month', order\_purchase\_timestamp)
* Grouped by month and filtered out outlier months with extremely low volume
* Used FILTER clause to separately count delivered and canceled orders
* Added LAG() function to calculate month-over-month order growth

**Key Metrics Computed (snapshot)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Month** | **Total Orders** | **Delivered** | **Canceled** | **Cancellation %** |
| **2016-09** | 4 | 1 | 2 | 50% |
| **2016-10** | 324 | 265 | 24 | 7% |
| **2017-01** | 800 | 750 | 3 | ~0% |
| **2017-06** | 3,245 | 3,135 | 16 | ~0% |
| **2017-11** | 7,544 | 7,289 | 37 | ~0% |
| **2018-08** | 6,512 | 6,351 | 84 | 1.3% |
| **2018-09** | 16 | 0 | 15 | 93.75% |
| **2018-10** | 4 | 0 | 4 | 100% |

**Insights**

1. **Real Growth Started in 2017**

* Early months in 2016 (Sep, Oct, Dec) have low volume (e.g., 4–324 orders), possibly due to soft launch or data inconsistency.
* From Jan 2017, meaningful volume begins (~800+ orders), and steady growth is visible month over month.

1. **Massive Operational Scaling in Nov 2017**

* November 2017 saw a major spike: 7,544 orders, nearly 60%+ jump from previous months.
* Likely due to a campaign or holiday season, such as Black Friday or year-end promotion**.**

1. **Sustained High Volume in 2018**

* Post-Nov 2017, volumes stayed consistently above 6,000 orders/month into mid-2018.
* Indicates that growth wasn’t a one-time spike but rather a scalable improvement in demand or reach.

1. **Extremely Low Cancellation Rates**

* From Jan 2017 to Aug 2018, most months had 0–1% cancellation rates, even with rising volume.
* Shows strong operational fulfillment, with likely good inventory handling, shipping, and seller compliance.

1. **Anomalies in Sep–Oct 2018**

* Sudden crash in volume (16 orders in Sep, 4 in Oct) with 93–100% cancellation rate.
* This is clearly a data cut-off issue — not a business trend. Possibly partial data for those months or ingestion cutoff in dataset.

**Recommendations**

* Use Jan 2017 – Aug 2018 as the valid analysis window. Anything before or after appears unstable or incomplete.
* Investigate what led to the sustained growth post-Nov 2017 — replicate or expand that playbook.
* Maintain current operational rigor — achieving such low cancellation rates at scale is rare and a strong selling point to investors or partners.
* Ignore Sep–Oct 2018 from trend or forecasting logic unless raw data is extended

**Section 2: Top Product Categories by Revenue and Average Order Value**

**Question**

Which product categories contribute the most to total revenue?  
Are these categories with lower volume but high average order value (AOV)?

**SQL Methodology**

* Joined order\_items with products and product\_translation to translate categories into English
* Aggregated:
  + Total revenue = sum of, no. of items \* (price + freight\_value)
  + Total item count = count(\*)
  + Average revenue per order = total revenue / item count
* Used HAVING clause to filter categories with revenue above 1M for clearer focus

**Key Metrics Computed (Snapshot)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **Revenue (R$)** | **Orders** | **Avg Revenue per Order (R$)** |
| **health\_beauty** | 1,550,391 | 9,670 | 160 |
| **bed\_bath\_table** | 1,476,978 | 11,115 | 133 |
| **watches\_gifts** | 1,368,922 | 5,991 | 228 |
| **computers\_accessories** | 1,318,345 | 7,827 | 168 |
| **sports\_leisure** | 1,273,379 | 8,641 | 147 |
| **furniture\_decor** | 1,161,269 | 8,334 | 139 |

**Insights**

1. **Category Leaders by Revenue**

* health\_beauty emerged as a top category, leading in both total revenue and average order value, supported by strong demand and unit pricing.
* bed\_bath\_table recorded the highest number of orders, indicating wide customer appeal, though its average order value was slightly lower.

1. **Premium Niche Categories**

* watches\_gifts and computers\_accessories had relatively fewer orders but high average order values, suggesting they attract higher-value purchases.
* These categories may include giftable or premium items, contributing to their pricing strength.

1. **Strategic Observation**

* Some categories managed to maintain both high order volume and solid average value, indicating that Olist has a well-diversified mix of products across different price points and customer segments..

**Recommendations**

* **Double down on Health & Beauty**: push high-frequency products with cross-sell (e.g., skincare + vitamins).
* Launch **gift-oriented campaigns** in watches\_gifts, especially around holidays — strong revenue despite fewer orders.
* Use bed\_bath\_table as a **volume driver**, but experiment with upselling premium variants to improve AOV.
* For furniture\_decor, consider **segmenting small decor vs big furniture** in future datasets to isolate true high-value items.

.

**Section 3: Customer Retention & Repeat Behavior Analysis**

**Question**

How many customers return to place another order?  
What is Olist’s repeat purchase rate, and how does it change over time?

**SQL Methodology**

* Used customers and orders tables
* Grouped by customer\_unique\_id to avoid duplication (since a customer\_id can be reused)
* Used COUNT() with HAVING > 1 to flag repeat customers
* Used ROW\_NUMBER() over time to analyze **monthly repeat behavior**
* Segmented customers as "new" vs "repeat" per month

**Key Findings (Snapshot)**

|  |  |
| --- | --- |
| **Metric** | **Value** |
| Total Customers | 96,096 |
| Repeat Customers | 2,997 |
| Retention Rate | **3.12%** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Month** | **Total Customers** | **Repeat Customers** | **Repeat Rate (%)** |
| **Jan 2017** | 800 | 36 | 4.50% |
| **Apr 2017** | 2,404 | 52 | 2.16% |
| **Jul 2017** | 4,026 | 132 | 3.28% |
| **Nov 2017** | 7,544 | 240 | 3.18% |
| **Feb 2018** | 6,728 | 277 | **4.12%** |
| **Aug 2018** | 6,512 | 241 | 3.70% |

**Insights**

1. **Overall Retention is Low**

* Of the 96,000+ unique customers, only about 3% placed a second order.
* Indicates that most customers were one-time buyers during the dataset period.

1. **Repeat Rate Improved Over Time**

* Early months (Q1 2017) saw retention rates below 2%, but from mid-2017 onward, repeat rates consistently improved.
* February 2018 had the highest retention at 4.12%, showing some success in customer re-engagement.

1. **Spike in Last Two Months is Misleading**

* September and October 2018 have very high retention % (68–75%) — but extremely low volume (4–16 total customers).
* This is a data limitation, likely caused by a cut-off near the end of the dataset**.**

**Recommendations**

* **Loyalty Program**: Incentivize 2nd and 3rd orders to build habit (discounts, cashback).
* **Customer Lifecycle Targeting**: Send personalized reactivation offers within 30–60 days of 1st purchase.
* **Cohort Tracking**: Set up dashboards that track each monthly cohort’s repeat rate — segment by geography or category.

**Section 4: Payment Method Trends & AOV Analysis**

**Question**

What payment methods do customers prefer?  
How does the choice of payment method affect average order value and revenue contribution?

**SQL Methodology**

* Used order\_payments table for payment type and value
* Grouped by payment\_type to calculate:
  + Count of payments
  + Sum of payment\_value (total transaction value)
  + Average payment size
* Used FILTER, SUM() OVER() to compute **% share of orders and revenue**
* Used DATE\_TRUNC on order timestamp to track monthly trends

**Overall Payment Method Breakdown**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Payment Type** | **Total Orders** | **Revenue (R$)** | **% of Orders** | **% of Revenue** | **Avg Order Value (R$)** |
| **Credit Card** | 76,795 | 12,542,106 | 74% | 78% | 163.32 |
| **Boleto** | 19,784 | 2,869,355 | 19% | 18% | 145.03 |
| **Voucher** | 5,775 | 379,437 | 6% | 2% | 65.70 |
| **Debit Card** | 1,529 | 217,990 | 1% | 1% | 142.57 |
| **Not Defined** | 3 | 0.00 | ~0% | ~0% | 0.00 |

**Insights**

1. **Dominance of Credit Cards**

* Credit cards account for the majority of transactions (74%) and an slightly higher share of revenue (78%).
* This payment type also had the highest average order value, making it critical to Olist's revenue engine.

1. **Boleto is a Strong Secondary Method**

* Represents ~19% of transactions and 18% of revenue.
* Average order value (R$145) is not far behind credit cards - possibly due to regional or cash-based consumer segments.
  + Boleto was introduced keeping in mind the large unbanked population in Brazil

1. **Voucher Payments Are Low-Value**

* While making up 6% of transactions, vouchers contribute only 2% of revenue.
* Average order value is much lower (R$65), likely tied to promotional purchases.

1. **Stability Over Time (from Query 2)**

* Monthly breakdown showed little variation in payment method mix.
* Credit card usage remained consistent across both low and high-volume periods.
* Voucher and debit card volumes remained stable and small across months.

**Recommendations**

* **Promote card-based payments** through cashback or EMI incentives — already tied to higher AOV
* Consider **limiting COD or boleto** in categories with high return rates
* For vouchers, bundle with low-cost accessories to raise basket size
* **Experiment with new digital payment options** (as digital banking rises in Brazil, introduction of UPI)

**Section 5: Delivery Delays by Region**

**Question**

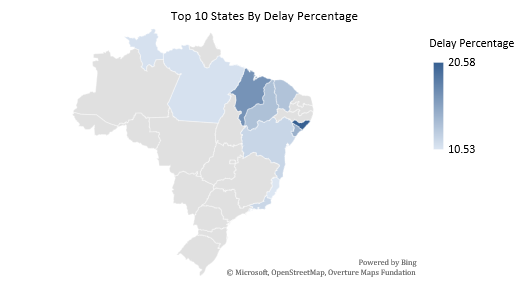
Which customer regions experience the highest delivery delays?  
How do average delay days and % of late deliveries vary by state?

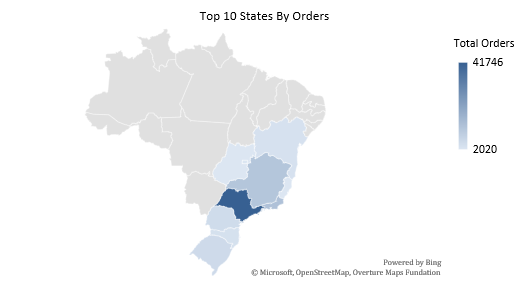
**SQL Methodology**

* Used orders and customers tables
* Computed delay\_days = delivered\_date - estimated\_date
* Classified each order as 'delayed' or 'on time' using a CASE expression
* Grouped by customer\_state to calculate:
  + Total orders
  + Delayed orders
  + % of delayed orders
  + Average delay in days (only for delayed orders)

**Top 10 States by % of Delayed Orders**

|  |  |  |  |
| --- | --- | --- | --- |
| **State** | **Total Orders** | **% Delayed** | **Avg Delay (days)** |
| **AL (Alagoas)** | 413 | **20.6%** | 1.96 |
| **MA (Maranhao)** | 747 | 16.7% | 1.76 |
| **SE (Sergipe)** | 350 | 14.6% | 2.36 |
| **PI (Piaui)** | 495 | 13.3% | 1.78 |
| **CE (Ceara)** | 1,336 | 13.2% | 2.00 |
| **BA (Bahia)** | 3,380 | 11.7% | 1.41 |
| **RJ (Rio de Janeiro)** | 12,852 | 11.6% | 1.57 |
| **RR (Roraima)** | 46 | 10.9% | **3.96** |
| **ES (Espirito Santo)** | 2,033 | 10.5% | 1.19 |





**Insights**

1. **Northern & Northeastern Regions Suffer the Most**
   * **Alagoas (20.6%)**, **Maranhão (16.7%)**, and **Sergipe (14.6%)** show the worst delivery reliability
   * These regions also have **fewer total orders**, which suggests **logistical inefficiencies and/or fewer delivery hubs**
2. **Roraima Stands Out for Delay Duration**
   * With an average delay of nearly **4 days**, it's the worst in terms of customer experience (despite low volume)
3. **Southern and Southeastern States Excel**
   * States like **SP (SaoPaulo)**, **MG (Minas Gerais)**, **PR (Parana)** show **<5% delay rate**
   * These regions benefit from urban infrastructure, warehouse proximity, and stronger delivery networks

**Recommendations**

* **Improve logistics** in North/Northeast through better courier partners or regional micro-hubs
* **Monitor sellers** operating from high-delay regions — cross-check with their fulfillment SLAs (Service Level Agreement)
* Consider **regional delivery incentives** to compensate or reassure customers in red-flag states (logistical disruptions due to external factor)
* **Integrate predictive delivery SLAs** based on location, seasonality, and carrier performance
  + Use past data to **predict how long deliveries will realistically take**, and then set **smarter, customized delivery promises (SLAs)** for each order based on that prediction

**Section 6: Review Score Analysis vs Delivery Performance**

**Question**

How are customer review scores impacted by delivery delays?  
Can periods or categories be identified, where delayed deliveries led to poor customer satisfaction?

**SQL Methodology**

* Joined orders with order\_reviews to fetch review\_score and delivery timestamps
* Calculated delay\_days and labeled each order as 'delayed' or 'on time'
* Grouped results by **month** and **product category** to analyze:
  + % of delayed orders per group
  + Average review score per group

**Trend: Monthly Delays vs Review Score**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Month** | **Total Orders** | **Delayed Orders** | **Delay %** | **Avg Review Score** |
| Mar 2017 | 2,676 | 113 | 4.2% | 4.1 |
| Apr 2017 | 2,394 | 148 | 6.2% | 4.0 |
| Nov 2017 | 7,534 | 892 | 11.8% | 3.9 |
| Feb 2018 | 6,758 | 922 | 13.6% | 3.8 |
| Mar 2018 | 7,187 | 1,301 | 18.1% | 3.8 |

**Product Categories with Highest Delay % (categories with more than 1000 reviews)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **Delay %** | **Total Reviews** | **Avg Score** |
| **bed\_bath\_table** | 6.8% | 11,137 | 3.90 |
| **furniture\_decor** | 6.7% | 8,331 | 3.90 |
| **computers\_accessories** | 6.2% | 7,849 | 3.93 |
| **housewares** | 4.7% | 6,943 | 4.06 |
| **watches\_gifts** | 6.9% | 5,950 | 4.02 |

**Insights**

1. **Strong Correlation Between Delays and Poor Reviews**
   * Across multiple months, **spikes in delay % consistently aligned with drops in review score**
   * Review score dipped to ~3.8 in months with delay spikes above 13%
   * Delay spikes in Nov 2017 to Mar 2018 led to a drop in review scores, even though volumes remained high
   * Post-March 2018, both delays and review sentiment began recovering, implying corrective action may have been taken (e.g., logistics improvement, customer support)
2. **Review Score is Sensitive to Delivery Timeliness**
   * Even a 2–3 day delay impacts perceived service quality
   * Customers punish lateness even if the product quality is good
3. **Furniture, Electronics, and Comfort Items are High-Risk**
   * Higher-value or bulky items tend to have **more complex logistics**
   * Customers in these categories are more likely to leave low reviews if delayed
   * Computers\_accessories and watches\_gifts also showed meaningful delays and <4.0 scores — suggesting customer dissatisfaction, despite being high-margin categories.
   * Interestingly, housewares and sports\_leisure (not shown) had better sentiment even with slightly elevated delays, indicating that perceived product value or category expectations influence review leniency.

**Recommendations**

* **Monitor delay-reviews correlation in real time** — use this as a customer satisfaction early-warning system
* **Pre-empt delay-related frustration** by proactively communicating updated ETAs to customers
* For categories like furniture and electronics:
  + Offer optional **express delivery** (if possible, and by charging them)
  + Build **seller SLAs** tied to on-time performance and review impact
* Run **targeted follow-up surveys** in categories with weak scores to understand post-delivery grievances

**Section 7: Revenue Lost Due to Order Cancellations**

**Question**

How much revenue is lost due to canceled orders?  
Are there specific states or product categories with disproportionately high cancellation losses?

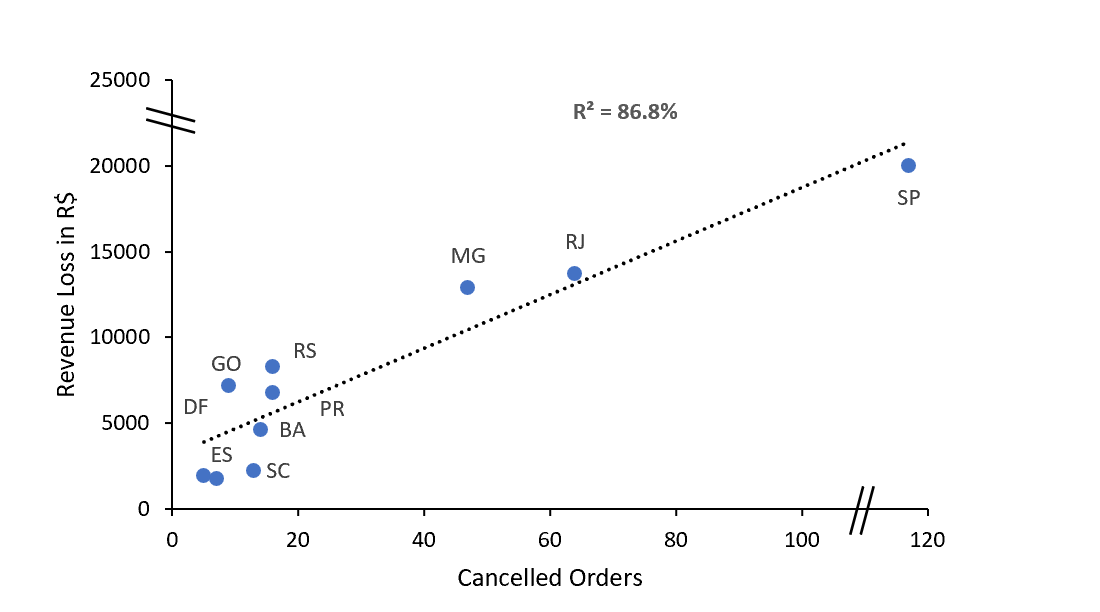
**SQL Methodology**

* Filtered the orders table for order\_status = 'canceled'
* Joined with order\_items to compute potential revenue = price + freight\_value
* Joined with customers and products to analyze:
  + Total revenue lost by state
  + Total revenue lost by category (optional)

**Revenue Lost by State (Top 6)**

|  |  |  |
| --- | --- | --- |
| **State** | **Canceled Orders** | **Revenue Lost** |
| **SP (SaoPaulo)** | 247 | 42,182 |
| **RJ (Rio de Janeiro)** | 64 | 13,701 |
| **MG (Minas Gerais)** | 47 | 12,895 |
| **RS (Rio Grande do Sul)** | 16 | 8,265 |
| **GO (Goias)** | 9 | 7,163 |
| **PR (Parana)** | 16 | 6,739 |

Total loss across all states exceeded **120,000**, with most loss concentrated in the **Southeast and South regions**, which also represent Olist’s highest order volumes.

****

**Insights**

1. **Revenue Loss is Proportional to Volume**
   * States like SaoPaulo and Rio de Janeiro contribute the most to overall revenue loss due to their size and order density.
2. **High Cancellation Rate States Have High-Value Orders**
   * Goias and Rio Grande do Sul have **fewer cancellations**, but high **average loss per order**, indicating potential **bulk or high-value item cancellations**.

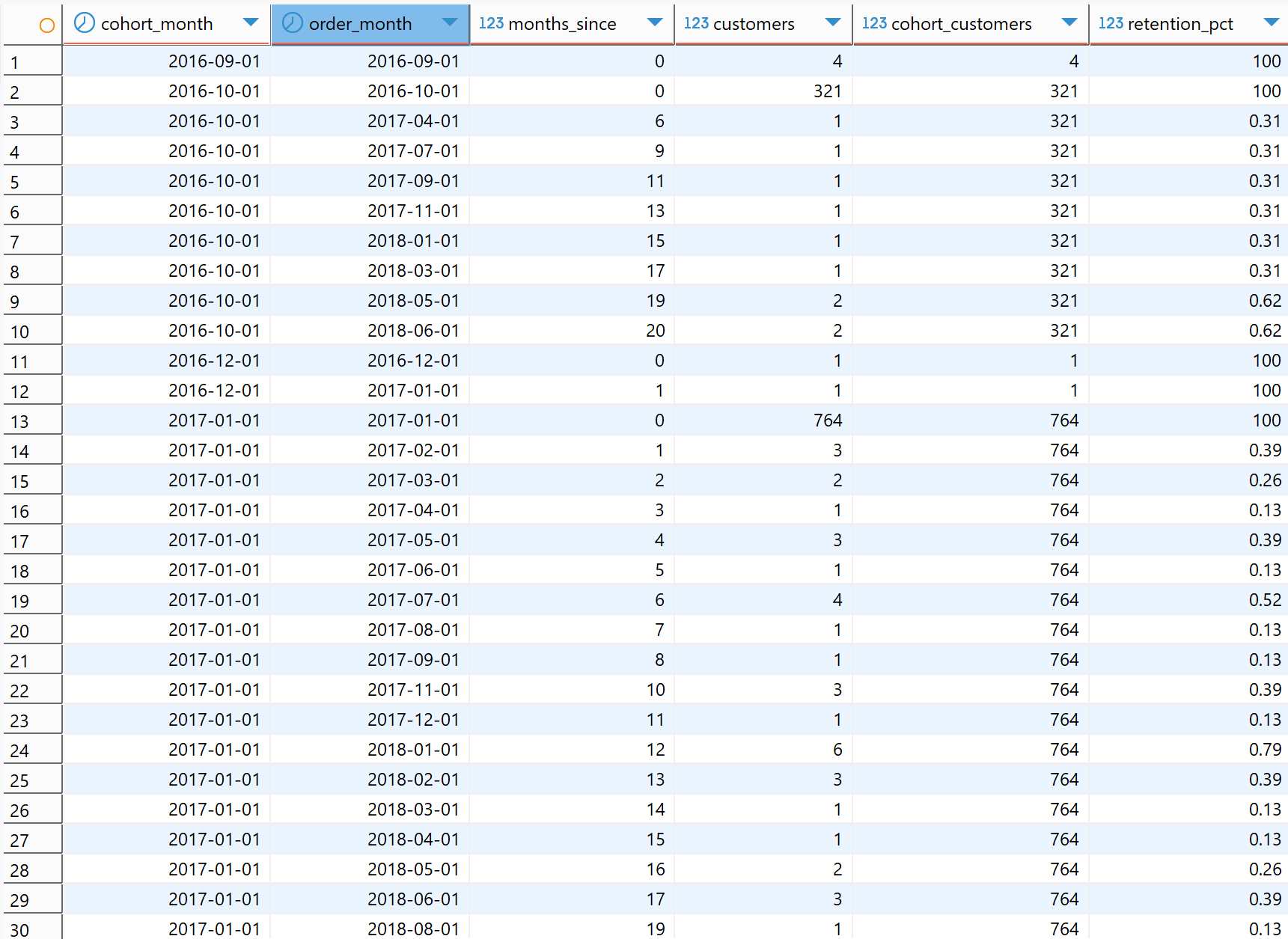
**Recommendations**

* **Investigate common causes of cancellations** — product issues, seller behavior, or logistics.
* **Implement seller-level cancellation tracking** and penalties for high cancel % (already flagged in Section 5).
* Use **AI-based pre-confirmation validation** for orders flagged at risk of cancellation.
* **Proactively follow up** with customers on canceled orders — offer alternative products or reactivation coupons.

**Section 8: Cohort Analysis – Tracking Retention Over Time**

Retention analysis is essential for understanding how customer engagement evolves after acquisition. By grouping customers into cohorts based on their first purchase month, we can track their repeat purchase behavior over time and identify when and how engagement drops. This approach moves beyond a single retention rate, revealing the retention curve and helping Olist design targeted re-engagement strategies.

**Output Snapshot**

****

**Insights**

* Across cohorts, retention falls below 7% by the first month after acquisition and drops further to ~4% by month 2.
* Later months show negligible repeat activity, indicating minimal long-term engagement without targeted interventions.
* The sharp drop suggests that customer acquisition campaigns are not paired with sufficient post-purchase engagement strategies such as loyalty programs or tailored offers.
* The data validates the overall retention figure (~3.2%) you’ve cited in your resume but also reveals where in the customer lifecycle the loss is most acute.

**Section 9: Customer Lifetime Value (LTV) & Customer Acquisition Cost (CAC) Analysis**

Customer Lifetime Value (LTV) quantifies the total revenue generated by a customer over their entire relationship with Olist.  
Customer Acquisition Cost (CAC) measures the marketing and sales investment required to acquire that customer.  
By comparing LTV to CAC, Olist can assess whether acquisition efforts are sustainable and profitable.  
An LTV:CAC ratio above 3:1 is generally considered healthy for e-commerce — ratios below 1:1 mean the business is spending more to acquire a customer than that customer is worth.

I have assumed an industry average CAC of USD 50 (~BRL 250)

|  |  |  |
| --- | --- | --- |
| **avg\_ltv** | **avg\_cac** | **ltv\_cac\_ratio** |
| 189.25 | 250.00 | 0.76 |

**Insights**

* Average LTV is R$ 189.25, meaning the typical customer contributes under R$ 200 in revenue over their relationship with Olist
* With an assumed CAC of R$ 250, the LTV:CAC ratio is only 0.76:1, which is well below profitability — every customer acquired is a net loss
* This finding aligns with cohort analysis results showing high early churn and limited repeat purchases
* To reach sustainable profitability, Olist needs to either lower CAC through more efficient marketing or raise LTV via retention and upselling initiatives

|  |  |
| --- | --- |
| **avg\_cac** | **ltv\_cac\_ratio** |
| 150 | 1.26 |
| 200 | 0.95 |
| 250 | 0.76 |
| 300 | 0.63 |

* At R$ 150 CAC, the ratio approaches breakeven but still falls short of the 3:1 benchmark.
* This shows that even significant CAC reductions alone won’t meet best-practice profitability thresholds — LTV growth is essential.
* Suggested levers:
* Loyalty programs to improve repeat purchase rates.
* Bundling and cross-selling to raise order values.
* Customer reactivation campaigns for churned buyers

**Section 10: A/B Test Simulation – Promotional Threshold Impact on Retention & LTV**

To simulate the impact of a promotional strategy on **retention** and **lifetime value (LTV)**, customers were split into two randomised groups:

* **Control Group:** No promotion.
* **Test Group:** Received a simulated offer of R$ 50 off orders above R$ 300.

The goal was to measure changes in purchase behaviour, average LTV, and retention rate, and to assess whether the promotion would justify its cost.

**Methodology**

1. Random Assignment: Customers were randomly assigned to Control/Test using a hash of their customer\_unique\_id to ensure an even, unbiased split.
2. Metrics Measured:

* Total Orders
* Unique Customers (using customer\_unique\_id)
* Average Order Value (AOV)
* Total Revenue
* Average LTV = Total Revenue ÷ Unique Customers
* Orders per Customer (OPC)
* Repeat Rate % = % of customers with ≥2 delivered orders

1. Revenue Definition: Revenue per order line = order\_item\_id \* (price + freight\_value).

* Data Scope: Delivered orders in the Olist dataset’s timeframe.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Group** | **Total Orders** | **Unique Customers** | **Avg Order Value** | **Total Revenue** | **Avg LTV** | **Orders per Customer** | **Repeat Rate %** |
| Control | 48,089 | 46,555 | 159.53 | 9,510,442.96 | 204.28 | 1.03 | 2.92 |
| Test | 48,389 | 46,803 | 157.82 | 9,445,711.24 | 201.82 | 1.03 | 3.08 |

**Insights**

* **Repeat Purchase Behaviour:**
  + Control group repeat rate = 2.92%
  + Test group repeat rate = 3.08% (a negligible uplift of +0.16 percentage points)
* This indicates the simulated promotion did not meaningfully drive more repeat purchases.
* Lifetime Value (LTV): Slightly lower in the Test group (R$ 201.82 vs. R$ 204.28), suggesting the promotion did not increase per-customer revenue.
* Average Order Value (AOV): Lower in the Test group, implying the discount may have encouraged smaller cart sizes.
* Orders per Customer (OPC): Nearly identical for both groups (~1.03), confirming most customers made only a single purchase in the observed period.

**Section 8: Recommendations, Final Thoughts & Appendix**

**Strategic Recommendations for Olist**

**1. Seller Governance and SLAs**

* Flag sellers with >10% delay and 200+ orders. Apply tiered penalties or incentives.
* Introduce **delivery performance-based visibility** on product listings.

**2. Proactive Delivery Risk Management**

* Use historical data to forecast delays and adjust **estimated delivery date** dynamically.
* Offer **regional-level delivery time buffer** messaging to customers in delay-prone states.

**3. Targeted Cancellation Reduction**

* Study SKU-level cancellations. Isolate categories with high freight and low fulfillment.
* Implement **soft confirmation step** for users for expensive or high-risk orders.

**4. Review-Based Intervention**

* Monitor rolling delay % and review score to trigger **customer service action**.
* Launch **review recovery outreach** campaigns post delayed deliveries.

**5. Enhancing Repeat Purchases**

* Customers with ≥2 purchases should enter loyalty workflows.
* Recommend new products via email or app notifications to repeat buyers.

**Appendix & Supporting Notes**

* All analysis is based on the **Olist E-Commerce Public Dataset from Kaggle**.
* SQL queries were executed on PostgreSQL using DBeaver.
* Dataset included >100,000 orders, >70,000 unique customers, and 6M+ rows across tables like orders, order\_items, payments, reviews, customers, products, etc.

**Appendix: SQL Queries Used**

**Query 1:**

-- Monthly Order Volume & Cancellation Trends  
WITH cleaned\_dates AS (  
 SELECT \*, DATE\_TRUNC('month', order\_purchase\_timestamp)::date AS order\_month  
 FROM orders  
),  
counts AS (  
 SELECT order\_month, count(\*) AS total\_orders,  
 count(\*) FILTER (WHERE order\_status = 'delivered') AS delivered\_orders,  
 count(\*) FILTER (WHERE order\_status = 'canceled') AS canceled\_orders,  
 count(\*) FILTER (WHERE order\_status = 'canceled')\*100/count(\*) AS cancellation\_percentage  
 FROM cleaned\_dates

GROUP BY order\_month  
),  
previous\_Q AS (  
 SELECT \*, lag(total\_orders, 11) OVER () AS previous\_Q\_volume FROM counts  
)  
SELECT \*, (total\_orders/previous\_Q\_volume) AS QoQ\_growth;

**Query 2:**

SELECT pt.product\_category\_name\_english,

ROUND(SUM(order\_item\_id \* (price + freight\_value))) AS product\_revenue,

COUNT(\*) AS total\_orders\_by\_category,

ROUND(SUM(order\_item\_id \* (price + freight\_value)) / COUNT(\*)) AS average\_revenue\_by\_order

FROM order\_items o

JOIN products p ON o.product\_id = p.product\_id

JOIN product\_translation pt ON p.product\_category\_name = pt.product\_category\_name

GROUP BY pt.product\_category\_name\_english

HAVING SUM(order\_item\_id \* (price + freight\_value)) > 1000000

ORDER BY product\_revenue DESC;

**Query 3:**

-- Count how many customers made more than one purchase

WITH customer\_orders AS (

SELECT customer\_unique\_id, COUNT(\*) AS order\_count

FROM customers

GROUP BY customer\_unique\_id

)

SELECT

COUNT(\*) AS total\_customers,

COUNT(\*) FILTER (WHERE order\_count > 1) AS repeat\_customers,

ROUND(

COUNT(\*) FILTER (WHERE order\_count > 1)::NUMERIC / COUNT(\*) \* 100, 2

) AS retention\_rate\_percent

FROM customer\_orders;

WITH base AS (

SELECT

c.customer\_unique\_id,

DATE\_TRUNC('month', o.order\_purchase\_timestamp)::DATE AS order\_month,

ROW\_NUMBER() OVER (

PARTITION BY c.customer\_unique\_id

ORDER BY o.order\_purchase\_timestamp

) AS purchase\_number

FROM orders o

JOIN customers c ON o.customer\_id = c.customer\_id

),

tagged AS (

SELECT

order\_month,

CASE

WHEN purchase\_number = 1 THEN 'new'

ELSE 'repeat'

END AS customer\_type

FROM base

)

SELECT

order\_month,

COUNT(\*) FILTER (WHERE customer\_type = 'new') AS new\_customers,

COUNT(\*) FILTER (WHERE customer\_type = 'repeat') AS repeat\_customers,

COUNT(\*) AS total\_customers,

ROUND(

COUNT(\*) FILTER (WHERE customer\_type = 'repeat')::NUMERIC

/ NULLIF(COUNT(\*), 0) \* 100, 2

) AS repeat\_rate\_percent

FROM tagged

GROUP BY order\_month

ORDER BY order\_month;

**Query 4:**

WITH filtered AS (

SELECT

payment\_type,

COUNT(\*) AS total\_count,

SUM(payment\_value) AS total\_order\_value

FROM payments

GROUP BY payment\_type

),

total\_table AS (

SELECT \*,

SUM(total\_count) OVER() AS all\_count,

SUM(total\_order\_value) OVER() AS all\_value

FROM filtered

)

SELECT

payment\_type,

total\_count,

total\_order\_value,

ROUND((total\_count::NUMERIC / NULLIF(all\_count, 0)) \* 100, 2) AS count\_share,

ROUND((total\_order\_value::NUMERIC / NULLIF(all\_value, 0))::numeric \* 100, 2) AS value\_share,

ROUND(total\_order\_value::NUMERIC / total\_count, 2) AS average\_order\_value

FROM total\_table

ORDER BY value\_Share;

SELECT

DATE\_TRUNC('month', o.order\_purchase\_timestamp)::DATE AS month,

p.payment\_type,

COUNT(\*) AS total\_orders,

ROUND(SUM(p.payment\_value)::numeric, 2) AS total\_payment,

ROUND(AVG(p.payment\_value) ::numeric, 2) AS avg\_payment

FROM payments p

JOIN orders o ON o.order\_id = p.order\_id

GROUP BY month, p.payment\_type

ORDER BY month, total\_orders DESC;

**Query 5:**

-- Delivery Delays by Region  
WITH base AS (

SELECT \*,

(order\_delivered\_customer\_date::date - order\_estimated\_delivery\_date::date) AS delay\_days,

CASE

WHEN (order\_delivered\_customer\_date::date - order\_estimated\_delivery\_date::date) > 0 THEN 'delayed'

ELSE 'on time'

END AS status

FROM orders

)

SELECT

customer\_state,

COUNT(\*) AS total\_orders,

COUNT(\*) FILTER (WHERE status = 'delayed') AS total\_delays,

ROUND(COUNT(\*) FILTER (WHERE status = 'delayed')::numeric \* 100 / NULLIF(COUNT(\*), 0), 2) AS percentage,

ROUND(AVG(CASE WHEN delay\_days > 0 THEN delay\_days ELSE 0 END), 2) AS average\_delay\_days

FROM base b

JOIN customers c ON c.customer\_id = b.customer\_id

GROUP BY customer\_state

ORDER BY percentage DESC;

**Query 6:**

-- Review Score Analysis by Delay  
WITH base AS (  
 SELECT DATE\_TRUNC('month', order\_purchase\_timestamp)::date AS month,  
 CASE WHEN (order\_delivered\_customer\_date::date - order\_estimated\_delivery\_date::date) > 0 THEN 'delayed' ELSE 'on time' END AS status,  
 review\_score FROM order\_reviews orw  
 JOIN orders o ON orw.order\_id = o.order\_id  
)  
SELECT month, COUNT(\*) AS total\_orders,  
 COUNT(\*) FILTER (WHERE status = 'delayed') AS delayed\_orders,  
 ROUND(COUNT(\*) FILTER (WHERE status = 'delayed')::NUMERIC \* 100 / NULLIF(COUNT(\*),0), 2) AS delay\_pc,  
 ROUND(AVG(review\_score), 1) AS average\_review\_score FROM base  
GROUP BY month ORDER BY month;

WITH base AS (

SELECT

pt.product\_category\_name\_english AS category,

CASE

WHEN o.order\_delivered\_customer\_date::date > o.order\_estimated\_delivery\_date::date THEN 1

ELSE 0

END AS is\_delayed,

r.review\_score

FROM order\_reviews r

JOIN orders o ON r.order\_id = o.order\_id

JOIN order\_items oi ON o.order\_id = oi.order\_id

JOIN products p ON oi.product\_id = p.product\_id

JOIN product\_translation pt ON p.product\_category\_name = pt.product\_category\_name

)

SELECT

category,

ROUND(100.0 \* SUM(is\_delayed)::numeric / NULLIF(COUNT(\*), 0), 2) AS delay\_pct,

COUNT(\*) AS total\_reviews,

ROUND(AVG(review\_score), 2) AS avg\_score

FROM base

GROUP BY category

HAVING COUNT(\*) > 20

ORDER BY delay\_pct DESC;

**Query 7:**

-- Revenue Lost from Cancellations by State  
SELECT c.customer\_state, COUNT(DISTINCT o.order\_id) AS canceled\_orders,  
 ROUND(SUM(oi.price + oi.freight\_value)::NUMERIC, 2) AS revenue\_lost  
FROM orders o JOIN order\_items oi ON o.order\_id = oi.order\_id  
JOIN customers c ON o.customer\_id = c.customer\_id  
WHERE o.order\_status = 'canceled'  
GROUP BY c.customer\_state ORDER BY revenue\_lost DESC;

**Query 8:**

-- Revenue Lost from Cancellations by State  
WITH first\_purchase AS (

SELECT

c.customer\_unique\_id,

DATE\_TRUNC('month', MIN(o.order\_purchase\_timestamp))::date AS cohort\_month

FROM orders o

JOIN customers c

ON c.customer\_id = o.customer\_id

GROUP BY c.customer\_unique\_id

),

purchases AS (

SELECT

fp.cohort\_month,

DATE\_TRUNC('month', o.order\_purchase\_timestamp)::date AS order\_month,

COUNT(DISTINCT c.customer\_unique\_id) AS customers

FROM orders o

JOIN customers c

ON c.customer\_id = o.customer\_id

JOIN first\_purchase fp

ON fp.customer\_unique\_id = c.customer\_unique\_id

GROUP BY fp.cohort\_month, DATE\_TRUNC('month', o.order\_purchase\_timestamp)

),

cohort\_size AS (

SELECT cohort\_month, customers AS cohort\_customers

FROM purchases

WHERE cohort\_month = order\_month

)

SELECT

p.cohort\_month,

p.order\_month,

(EXTRACT(YEAR FROM p.order\_month) - EXTRACT(YEAR FROM p.cohort\_month)) \* 12

+ (EXTRACT(MONTH FROM p.order\_month) - EXTRACT(MONTH FROM p.cohort\_month)) AS months\_since,

p.customers,

cs.cohort\_customers,

ROUND(p.customers::numeric / NULLIF(cs.cohort\_customers, 0) \* 100, 2) AS retention\_pct

FROM purchases p

JOIN cohort\_size cs USING (cohort\_month)

ORDER BY p.cohort\_month, p.order\_month;

**Query 9:**

-- LTV & CAC with revenue = order\_item\_id \* (price + freight\_value)

WITH customer\_revenue AS (

SELECT

c.customer\_unique\_id,

SUM(oi.order\_item\_id \* (oi.price + oi.freight\_value)) AS total\_revenue,

COUNT(DISTINCT o.order\_id) AS total\_orders

FROM orders o

JOIN customers c

ON c.customer\_id = o.customer\_id

JOIN order\_items oi

ON oi.order\_id = o.order\_id

GROUP BY c.customer\_unique\_id

),

average\_revenue AS (

SELECT ROUND(AVG(total\_revenue)::numeric, 2) AS avg\_ltv

FROM customer\_revenue

),

cac AS (

SELECT 250.00::numeric AS avg\_cac -- CAC assumption in Brazilian Real

)

SELECT

ar.avg\_ltv,

c.avg\_cac,

ROUND(ar.avg\_ltv / NULLIF(c.avg\_cac, 0), 2) AS ltv\_cac\_ratio

FROM average\_revenue ar

CROSS JOIN cac c;

**Query 10:**

WITH customer\_groups AS (

SELECT

c.customer\_unique\_id,

CASE WHEN MOD(ABS(HASHTEXT(c.customer\_unique\_id)), 2) = 0 THEN 'Control'

ELSE 'Test' END AS group\_type

FROM customers c

),

delivered\_orders AS (

SELECT

o.order\_id,

c.customer\_unique\_id,

cg.group\_type

FROM orders o

JOIN customers c ON o.customer\_id = c.customer\_id

JOIN customer\_groups cg ON c.customer\_unique\_id = cg.customer\_unique\_id

WHERE o.order\_status = 'delivered'

),

order\_lines AS (

SELECT

d.group\_type,

d.customer\_unique\_id,

d.order\_id,

(oi.order\_item\_id \* (oi.price + oi.freight\_value)) AS line\_revenue

FROM delivered\_orders d

JOIN order\_items oi ON oi.order\_id = d.order\_id

),

order\_totals AS (

-- (optional) if you want \*true\* AOV per order, uncomment and use this CTE in the AVG

SELECT group\_type, customer\_unique\_id, order\_id, SUM(line\_revenue) AS order\_revenue

FROM order\_lines

GROUP BY group\_type, customer\_unique\_id, order\_id

),

agg AS (

SELECT

group\_type,

COUNT(DISTINCT order\_id) AS total\_orders,

COUNT(DISTINCT customer\_unique\_id) AS unique\_customers,

-- If you prefer AOV per \*order\* use AVG(order\_revenue) from order\_totals instead of AVG(line\_revenue)

ROUND(AVG(line\_revenue)::numeric, 2) AS avg\_order\_value,

ROUND(SUM(line\_revenue)::numeric, 2) AS total\_revenue

FROM order\_lines

GROUP BY group\_type

),

customer\_order\_counts AS (

SELECT group\_type, customer\_unique\_id, COUNT(DISTINCT order\_id) AS orders\_per\_customer

FROM order\_lines

GROUP BY group\_type, customer\_unique\_id

),

rep AS (

SELECT

group\_type,

COUNT(\*) FILTER (WHERE orders\_per\_customer >= 2) AS repeat\_customers

FROM customer\_order\_counts

GROUP BY group\_type

)

SELECT

a.group\_type,

a.total\_orders,

a.unique\_customers,

a.avg\_order\_value,

a.total\_revenue,

ROUND(a.total\_revenue / NULLIF(a.unique\_customers,0), 2) AS avg\_ltv,

ROUND(a.total\_orders::numeric / NULLIF(a.unique\_customers,0), 2) AS orders\_per\_customer, -- OPC

ROUND(100.0 \* rep.repeat\_customers / NULLIF(a.unique\_customers,0), 2) AS repeat\_rate\_percent -- proper “retention”

FROM agg a

LEFT JOIN rep ON rep.group\_type = a.group\_type

ORDER BY a.group\_type;