

Advanced Data Analysis using SQL

Amazon Brazil's Data Analysis using SQL



By: Ansh Jain

Program: NextLeap **Data Analyst** Fellowship

Date: June 2025

Table of Contents

1. Introduction

- 1.1 Leveraging Data-Driven Insights from Amazon Brazil's Data
- 1.2 Objective
- 1.3 Data Source
- 1.4 Data Preparation

2. Analysis – I

- Q 2.1 Standardizing Payment Values
- Q 2.2 Payment Type Distribution
- Q 2.3 Smart Products in Price Range
- Q 2.4 Top 3 Sales Months
- Q 2.5 Categories with High Price Variations
- Q 2.6 Payment Type Consistency Analysis
- Q 2.7 Products with Incomplete Category Names

3. Analysis – II

- Q 3.1 Payment Type Popularity Across Order Value Segments
- Q 3.2 Product Category Price Analysis
- Q 3.3 Multi-Order Customers
- Q 3.4 Customer Categorization
- Q 3.5 Top Revenue Generating Categories

4. Analysis – III

- Q 4.1 Seasonal Sales Comparison
- Q 4.2 Above-Average Sales Volume Products
- Q 4.3 Monthly Revenue Trends for 2018
- Q 4.4 Customer Segmentation Using CTE
- Q 4.5 Top 20 High-Value Customers
- Q 4.6 Monthly Cumulative Sales Using Recursive CTE
- Q 4.7 Month-over-Month Growth Rate Analysis

1. Introduction

1.1 Leveraging Data-Driven Insights from Amazon Brazil's Data:

Amazon, a global e-commerce leader, has demonstrated tremendous success in emerging markets like Brazil by connecting local businesses with millions of customers. With similar market characteristics—large population, diverse consumer behavior, and a growing digital economy—India represents a strong opportunity for replicating this success.

By analyzing transactional data from Amazon Brazil, we can uncover customer behaviors, product trends, and operational metrics that can be adapted for strategic growth in the Indian market. This project leverages the comprehensive dataset from Amazon Brazil to inform Amazon India's strategic initiatives using advanced SQL-driven analysis.

1.2 Objective:

The objective of this analysis is to understand and analyze:

- **Customer demographics and behaviors** using the Customers table to understand purchase patterns and preferences
- **Evaluate regional trends** and customer density through geographical data analysis
- **Track order lifecycles** and product preferences using Orders, Order Items, and Product tables
- **Analyze payment preferences** and transaction details via the Payments table
- **Identify seasonal patterns** and revenue optimization opportunities
- **Segment customers** based on purchasing behavior for targeted marketing strategies

This comprehensive analysis will help Amazon India enhance customer experience, optimize operations, and seize new market opportunities through data-driven decision making.

1.3 Data Source:

The analysis is based on a relational dataset from Amazon Brazil consisting of **6 interconnected tables** that provide comprehensive insights into e-commerce operations:

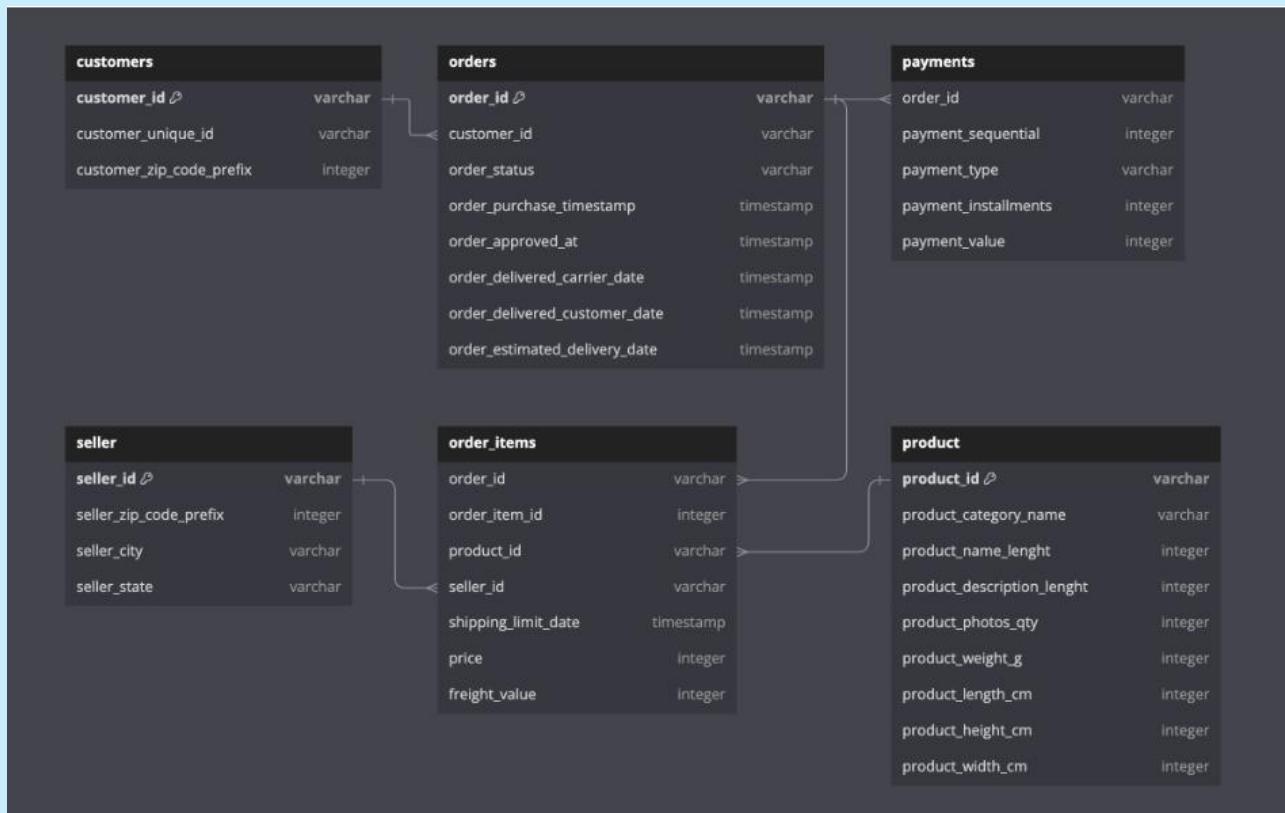
- **Customers:** Customer unique identifiers, geographical locations, and demographic segmentation data
- **Orders:** Order timestamps, delivery status, customer linkages, and order lifecycle information
- **Order Items:** Item-level transaction details including product IDs, prices, shipping costs, and seller information
- **Products:** Product catalog data including category names, descriptions, and physical attributes
- **Sellers:** Seller identification, location data, and marketplace participation details
- **Payments:** Payment method preferences, transaction values, installment details, and financial data

The dataset spans multiple years of transactional data, providing rich insights into customer behavior, seasonal trends, and business performance metrics.

1.4 Data Preparation:

Data was imported into **PostgreSQL** using CSV files and organized into a dedicated schema for comprehensive analysis. The analysis employs a wide range of advanced SQL techniques including:

- **Fundamental SQL Operations:** SELECT, WHERE, GROUP BY, ORDER BY clauses
- **Aggregate Functions:** SUM, COUNT, AVG, MIN, MAX, STDDEV
- **Advanced Joins:** INNER JOIN, LEFT JOIN for multi-table analysis
- **Window Functions:** RANK, DENSE_RANK, LAG, ROW_NUMBER, partitioning
- **Common Table Expressions (CTEs):** Both standard and recursive CTEs
- **Subqueries:** Nested queries for complex analytical requirements
- **Date-Time Operations:** Seasonal analysis, monthly trends, and time-based segmentation



SCHEMA

The analysis is strategically divided into **3 comprehensive parts** covering different business aspects, with a total of **19 practical business questions**. Each question addresses real-world scenarios relevant to Amazon India's market entry and growth strategy, from basic descriptive analytics to advanced predictive insights.

This analysis translates raw relational data into actionable business intelligence, providing Amazon India with the insights needed to replicate Amazon Brazil's success in the Indian market.

2. Analysis – I

Question 2.1 -- Standardizing Payment Values

Problem Statement:

-To simplify its financial reports, Amazon India needs to standardize payment values. The finance team requires payment analysis with rounded average values to eliminate decimal complexities and create cleaner reporting dashboards. This standardization will help in budget planning and payment method comparison across different transaction volumes.

Desired Output:

-payment_type, rounded_avg_payment

Approach:

1. Identifying Relevant Tables and Columns:

- **Table:** payments
- **Columns:** payment_type, payment_value

2. Calculating Average Payment Values:

- Use AVG() aggregate function to calculate mean payment value for each payment type
- Group results by payment_type to get category-wise averages

3. Standardizing with ROUND Function:

- Apply ROUND() function to convert decimal averages to integers
- This eliminates decimal places for cleaner financial reporting

4. Sorting the Results:

- Order by rounded_avg_payment in ascending order to identify lowest to highest payment methods

SQL Query:

```
1 ✓ select payment_type,round(avg(payment_value)) as rounded_avg.  
2   from payments  
3   group by payment_type  
4   order by rounded_avg_payment asc
```

Query Output:

payment_type	rounded_avg_payment
not_defined	0
voucher	66
debit_card	143
boleto	145
credit_card	163

Recommendations:

Strategic Payment Optimization for Amazon India:

1. Premium Payment Channel Focus: Since credit cards show the highest average transaction values (165 BRL), Amazon India should prioritize credit card adoption through:
 - o Exclusive EMI options with 0% interest for high-value purchases
 - o Additional cashback rewards (2-3%) for credit card transactions
 - o Premium customer service for credit card users
2. Voucher-Based Customer Retention: With vouchers showing the second-highest average values (158 BRL), implement:
 - o Loyalty voucher programs for repeat customers
 - o Category-specific vouchers for high-margin products
 - o Seasonal voucher campaigns during peak sales periods

Question 2.2 -- Payment Type Distribution

Problem Statement:

To refine its payment strategy, Amazon India wants to understand the distribution of orders by payment type. The marketing team needs percentage-based insights to determine which payment methods are most popular among customers, enabling them to optimize payment gateway partnerships and promotional strategies.

Desired Output:

payment_type, percentage_orders

Approach:

1. Counting Orders by Payment Type:

- o Use COUNT(*) to calculate total orders for each payment method

- o GROUP BY payment_type to segment the data

2. Calculating Total Orders:

- o Use window function SUM(COUNT(*)) OVER() to get total order count across all payment types
- o This provides the denominator for percentage calculation

3. Computing Percentages:

- o Calculate percentage as (individual count / total count) * 100
- o ROUND to 1 decimal place for clean presentation

4. Sorting Results:

- o ORDER BY percentage_orders DESC to show most popular payment methods first

SQL Query:

```
select payment_type,
round(count(*) *100.0/sum(count(*)) over(),1)
as percentage_orders
from payments
group by payment_type
order by percentage_orders desc
```

Query Output:

payment_type	percentage_orders
credit_card	73.9
boleto	19
voucher	5.6
debit_card	1.5
not_defined	0

Recommendations:

Payment Infrastructure Prioritization for Amazon India:

1. Credit Card Infrastructure Investment: With 74% usage rate, ensure robust credit card processing capabilities, multiple gateway partnerships, and instant approval systems.
2. Boleto-Equivalent Strategy: Since Boleto captures 19% of transactions in Brazil (serving unbanked population), Amazon India should develop UPI, cash-on-delivery, and digital wallet partnerships to capture India's similar demographic.

Question 2.3 -- Smart Products in Price Range

Problem Statement:

Amazon India seeks to create targeted promotions for products within specific price ranges. The product team wants to identify all "Smart" products priced between 100 and 500 BRL to develop focused marketing campaigns for tech-savvy customers in the mid-range segment.

Desired Output:

product_id, price

Approach:

1. Identifying Relevant Tables:

- **Primary Table:** order_items (for pricing data)
- **Secondary Table:** products (for product category information)

2. Joining Tables:

- JOIN order_items and products tables on product_id
- This combines pricing with product category information

3. Applying Filters:

- Price filter: BETWEEN 100 AND 500 for target price range
- Product filter: LIKE '%smart%' to identify smart products in category names

4. Sorting Results:

- ORDER BY price DESC to show highest-priced smart products first

SQL Query:

```
select o.product_id, o.price
from order_items as o
join products as p
on o.product_id=p.product_id
where o.price between 100 AND 500 and p.product_category_name like 'smart'
order by o.price desc
```

Query Output:

1	product_id	price
2	1df1a2df8ad2b9d3aa49fd851e3145ad	439.99
3	7debe59b10825e89c1cbcc8b190c85e2	349.99
4	ca86b9fe16e12de698c955aedff0aea2	349
5	ca86b9fe16e12de698c955aedff0aea2	349
6	0e52955ca8143bd179b311cc454a6caa	335
7	7aeaa8f3e592e380c420e8910a717255	329.9
8	7aeaa8f3e592e380c420e8910a717255	329.9
9	7aeaa8f3e592e380c420e8910a717255	329.9
10	7aeaa8f3e592e380c420e8910a717255	329.9
11	7aeaa8f3e592e380c420e8910a717255	329.9
12	7aeaa8f3e592e380c420e8910a717255	329.9
13	d1b571cd58267d8cac8b2af6e288bbc	299.9
14	d1b571cd58267d8cac8b2af6e288bbc	299.9
15	66ffe28d0fd53808d0535eee4b90a157	254

Recommendations:

Smart Product Category Marketing Strategy for Amazon India:

1. **"Affordable Smart Living" Brand Positioning:** Position the 100-500 BRL price range as the "smart and affordable" segment, targeting middle-income consumers who aspire for technology without premium pricing.
2. **Social Media-Driven Marketing Campaigns:**
 - o Create influencer partnerships showcasing smart products as lifestyle essentials
 - o Develop "Smart Home on a Budget" content series across Instagram, YouTube, and regional platforms
 - o Use before/after demonstrations to highlight product value
3. **Category Expansion:** Use successful smart products as anchor items to introduce customers to higher-value smart home ecosystems, creating natural upselling opportunities.

Question 2.4 --Top 3 Sales Months

Problem Statement:

To identify seasonal sales patterns, Amazon India needs to focus on the most successful months. The operations team requires insights into peak sales periods to

optimize inventory management, staffing levels, and marketing spend allocation throughout the year.

Desired Output:

month, total_sales

Approach:

1. Identifying Relevant Tables:

- **Primary Table:** orders (for purchase timestamps)
- **Secondary Table:** order_items (for sales values)

2. Extracting Month Information:

- Use TO_CHAR() function with 'Month' format to extract month names
- This provides readable month names instead of numeric values

3. Calculating Monthly Sales:

- JOIN orders and order_items tables on order_id
- SUM(price) to calculate total sales value per month
- ROUND() to standardize sales figures to nearest integer

4. Limiting Results:

- ORDER BY total_sales DESC and LIMIT 3 to get top-performing months

SQL Query:

```
SELECT TO_CHAR(o.order_purchase_timestamp, 'Month') AS month,
       ROUND(SUM(oi.price)) AS total_sales
  FROM orders o
JOIN order_items oi
 WHERE o.order_id = oi.order_id
 GROUP BY TO_CHAR(o.order_purchase_timestamp, 'Month')
 ORDER BY total_sales DESC
 LIMIT 3
```

Query Output:

1	month	total_sales
2	May	1502589
3	August	1428658
4	July	1393539

Recommendations:

Seasonal Sales Optimization Strategy for Amazon India:

1. Pre-Peak Marketing Amplification:

- **October (Pre-November):** Launch Diwali/festive season campaigns with 40% increased marketing spend
- **April (Pre-May):** Target summer product categories with early-bird promotions
- **July (Pre-August):** Focus on back-to-school and monsoon-ready products

2. Category-Specific Seasonal Targeting:

- **November:** Electronics, home appliances, and gift items (leveraging festive demand)
- **May:** Summer essentials, travel gear, and cooling appliances
- **August:** Fashion, back-to-school supplies, and monsoon products

3. Credit Card Partnership Expansion (Leveraging 74% Usage Rate):

- Partner with major Indian banks (HDFC, ICICI, SBI) for exclusive seasonal offers
- Introduce 0% EMI options for purchases above ₹10,000 during peak months
- Launch "Buy Now, Pay Later" services specifically timed with high-sales periods

4. Anchoring Product Strategy:

- Offer steep discounts (20-30%) on high-demand anchor products to drive traffic
 - Create "End-of-Season" clearance events to maintain momentum between peak periods
 - Use seasonal anchor products to cross-sell complementary higher-margin items
-

Question 2.5 --Categories with High Price Variations

Problem Statement:

Amazon India is interested in product categories with significant price variations. The pricing strategy team needs to identify categories where the difference between maximum and minimum product prices exceeds 500 BRL. This analysis will help understand market dynamics, pricing opportunities, and potential for premium vs. budget product positioning within categories.

Desired Output:

product_category_name, price_difference

Approach:

1. Identifying Relevant Tables:

- **Primary Table:** products (for category information)
- **Secondary Table:** order_items (for actual selling prices)

2. Joining Product and Pricing Data:

- JOIN products and order_items tables on product_id
- This combines category names with actual transaction prices

3. Calculating Price Variations:

- Use MAX(price) and MIN(price) for each category
- Calculate difference: (MAX(price) - MIN(price))
- GROUP BY product_category_name to analyze each category separately

4. Filtering High Variation Categories:

- HAVING clause to filter categories with price difference > 500 BRL
- This identifies categories with significant pricing spreads

SQL Query:

```
select p.product_category_name,
(max(o.price)-min(o.price)) as price_difference
from products p
join order_items o
on p.product_id=o.product_id
group by p.product_category_name
having (max(o.price)-min(o.price))>500
```

Query Output:

1	product_category_name	price_difference
2	climatizacao	1588.1
3	livros_importados	730.01
4	NULL	3977
5	ferramentas_jardim	3923.65
6	dvds_blu_ray	1411.1
7	cine_foto	867.19
8	beleza_saude	3122.8
9	livros_interesse_geral	893.9
10	tablets_impressao_imagem	875.09
11	papelaria	1690.71
12	bebés	3895.46
13	musica	1162.12
14	consoles_games	4094.81
15	eletrodomesticos	2083.81

Recommendations:

Quality-Controlled Product Range Management for Amazon India:

1. Dynamic Quality-Price Monitoring System:

- *Implement automated alerts for products priced >60% below category average*
- *Establish mandatory quality verification for products in bottom 10% price range*
- *Create real-time dashboard tracking correlation between low prices and negative reviews*

2. Seller Performance Management:

- **Warning System:** Issue quality alerts to sellers with products receiving <3.5 star ratings in low-price segments
- **Seller Education:** Provide training on maintaining quality standards while offering competitive pricing
- **Progressive Penalties:** Implement 3-tier system - Warning → Product delisting → Seller suspension for persistent quality issues

3. Category-Specific Price Band Strategy:

- Define acceptable price ranges for each category (e.g., Electronics: Premium/Mid-range/Budget with max 3:1 ratio)

- Create "Amazon's Choice" badges for products offering optimal price-quality balance
- Establish minimum quality thresholds for products in each price band

4. Customer Protection Measures:

- Highlight "Price vs. Quality" indicators on product pages
 - Implement "Too Good to be True" warnings for exceptionally low-priced items
 - Strengthen return/refund policies specifically for low-price segment purchases
-

Question 2.6 -- Payment Type Consistency Analysis

Problem Statement:

To enhance the customer experience, Amazon India wants to find which payment types have the most consistent transaction amounts. The finance team needs to identify payment methods with the least variance in transaction amounts to understand customer behavior patterns and optimize payment processing systems.

Desired Output:

payment_type, std_deviation

Approach:

1. Identifying Relevant Data:

- **Table:** payments
- **Columns:** payment_type, payment_value

2. Filtering Valid Payment Types:

- Exclude 'not_defined' payment types to focus on actual payment methods
- WHERE clause to remove undefined or invalid entries

3. Calculating Standard Deviation:

- Use STDDEV() function to measure transaction amount variability
- GROUP BY payment_type to analyze each payment method separately
- Lower standard deviation indicates more consistent transaction amounts

4. Ranking by Consistency:

- ORDER BY std_deviation ASC to show most consistent payment types first
- This helps identify which payment methods have predictable transaction patterns

SQL Query:

```
select payment_type,
       STDDEV(payment_value) as std_deviation
    from payments
   --I have excluded the the data where payment_type is not known
  WHERE payment_type != 'not_defined'
 group by payment_type
ORDER BY std_deviation DESC
```

Query Output:

1	payment_type	std_deviation
2	debit_card	245.793401
3	credit_card	222.1193107
4	boleto	213.5810615
5	voucher	115.5191854

Recommendations:

Payment Predictability Strategy for Amazon India:

1. **Cash Flow Forecasting Optimization:** Use payment methods with low standard deviation (high consistency) for accurate revenue predictions and inventory planning, while accounting for higher variability in inconsistent payment channels.
2. **Targeted Product Positioning:** Position consistent-value products (daily essentials, subscriptions) toward payment methods showing predictable transaction patterns, while promoting high-value/variable items through payment channels with higher transaction variance.

Question 2.7 -- Products with Incomplete Category Names

Problem Statement:

Amazon India wants to identify products that may have incomplete category names in order to fix data quality issues from their end. The data management team needs to find products where category information is missing or contains only a single character, which could indicate data entry errors or incomplete product catalog information.

Desired Output:

product_id, product_category_name

Approach:

1. Identifying Data Quality Issues:

- **Table:** products
- **Focus:** product_category_name field validation

2. Detecting Missing Category Names:

- Check for NULL values in product_category_name
- These represent completely missing category information

3. Identifying Single Character Categories:

- Use TRIM() to remove whitespace
- LIKE '_' pattern to find single character entries
- Single characters likely indicate incomplete data entry

4. Combining Quality Checks:

- OR condition to capture both NULL and single character issues
- This comprehensive check identifies all problematic category entries

SQL Query:

```
SELECT product_id, product_category_name
FROM products
WHERE product_category_name IS NULL
    OR TRIM(product_category_name) like '_'
```

Query Output:

1	product_id	product_category_name
2	ce6f74096c84567f22728c84f3d6e7fc	c
3	c7fce98e1aa3d8a6cb99cbe6	f
4	a41e356c76fab66334f36de622ecbd3a	NULL
5	d8dee61c2034d6d075997acef1870e9b	NULL
6	56139431d72cd51f19eb9f7dae4d1617	NULL
7	46b48281eb6d663ced748f324108c733	NULL
8	5fb61f482620cb672f5e586bb132eae9	NULL
9	e10758160da97891c2fdcbc35f0f031d	NULL
10	39e3b9b12cd0bf8ee681bbc1c130feb5	NULL
11	794de06c32a626a5692ff50e4985d36f	NULL
12	7af3e2da474486a3519b0cba9dea8ad9	NULL
13	629beb8e7317703dcc5f35b5463fd20e	NULL
14	3a78f64aac654298e4b9aff32fc21818	NULL
15	bcb815bba008d89458e428078c0b9211	NULL

Recommendations:

Data Quality Enhancement Strategy for Amazon India:

1. Automated Data Validation System:

Implement rule-based data checks during product upload that flag incomplete or invalid category entries (e.g., NULL or single-character categories). Prevent products from going live until validated.

2. Regular Data Audits:

Schedule weekly scans of the product catalog to identify and report any anomalies in category naming. Maintain logs for compliance and seller accountability.

3. Seller Training & Enforcement:

Provide sellers with clear guidelines for category naming conventions. For repeated violations, issue warnings followed by listing suspension to maintain catalog integrity.



Summary of Analysis I:

- **Q2.1:** Financial standardization for cleaner reporting
- **Q2.2:** Payment method popularity distribution analysis
- **Q2.3:** Targeted product identification for promotions
- **Q2.4:** Seasonal sales pattern recognition

- **Q2.5:** Category pricing strategy insights
- **Q2.6:** Payment consistency for user experience optimization
- **Q2.7:** Data quality improvement identification

Key Business Value: This section provides foundational insights into payment behaviors, product positioning, seasonal trends, and data quality - essential for Amazon India's market entry strategy.

3. Analysis – II

Question 3.1 -- Payment Type Popularity Across Order Value Segments

Problem Statement:

Amazon India wants to understand which payment types are most popular across different order value segments. The business strategy team needs insights into customer payment preferences based on their spending patterns - whether low-value, medium-value, or high-value customers prefer different payment methods. This segmentation will help optimize payment gateway partnerships and targeted promotions.

Desired Output:

order_value_segment, payment_type, count

Approach:

1. Creating Order Value Segments:

- **Low Segment:** Orders less than 200 BRL
- **Medium Segment:** Orders between 200 and 1000 BRL
- **High Segment:** Orders over 1000 BRL

2. Segmentation Logic:

- Use CASE WHEN statements to categorize payment_value into segments
- This creates meaningful customer spending categories

3. Payment Type Analysis:

- COUNT(*) to determine frequency of each payment type within segments
- GROUP BY both payment_type and order_value_segment for cross-analysis

4. Prioritizing Results:

- o ORDER BY count DESC to identify most popular payment-segment combinations

SQL Query:

```
select
    case when payment_value < 200 then 'low'
        when payment_value between 200 and 1000 then 'medium'
        when payment_value > 1000 then 'high'
    end as order_value_segment,
    payment_type,
    count(*) as count
from payments
group by payment_type ,order_value_segment
order by count desc
```

Query Output:

1	order_value_segment	payment_type	count
2	low	credit_card	60548
3	low	boleto	16444
4	medium	credit_card	15303
5	low	voucher	5476
6	medium	boleto	3162
7	low	debit_card	1287
8	high	credit_card	944
9	medium	voucher	286
10	medium	debit_card	227
11	high	boleto	178
12	high	debit_card	15
13	high	voucher	13
14	low	not defined	3

Recommendations:

Segmented Payment Strategy for Amazon India:

1. Low-Value Segment (Under 200 BRL):

- o **Bundle & Pay-Later Offers:** Encourage multi-item purchases by offering discounts on product bundles with pay-later options.
- o **UPI/Digital Wallet Discounts:** Provide added incentives for prepaid methods to increase digital adoption and reduce COD-related issues.
- o **Return Suppression Discount:** Offer 5% discount to customers who opt out of returns/exchanges on low-cost items to reduce logistics and fraud costs.

2. Medium Segment (200–1000 BRL):

- **Anchor Product Promotion:** Identify and promote "must-have" mid-range products (e.g., daily-use electronics or lifestyle items) to upgrade low-segment buyers and boost average order value.
- **Balanced Incentive Strategy:** Maintain a mix of credit card and digital wallet offers while using customer insights to deliver personalized nudges toward mid-tier purchases.

3. High-Value Segment (Above 1000 BRL):

- **Voucher Optimization:** Reduce reliance on vouchers and shift focus to advance payment methods like UPI for better cash flow.
 - **Enhanced EMI Offerings:** Expand partnerships with banks to offer flexible EMI and “Buy Now, Pay Later” options during seasonal promotions.
-

Question 3.2 --Product Category Price Analysis

Problem Statement:

Amazon India wants to analyse the price range and average price for each product category. The pricing strategy team requires comprehensive insights into minimum, maximum, and average prices across all categories to inform competitive pricing strategies, identify premium vs. budget segments, and optimize category-specific promotions.

Desired Output:

product_category_name, min_price, max_price, avg_price

Approach:

1. Identifying Relevant Tables:

- **Primary Table:** products (for category information)
- **Secondary Table:** order_items (for actual transaction prices)

2. Joining Product and Pricing Data:

- JOIN products and order_items on product_id
- This combines category names with real transaction prices

3. Calculating Price Statistics:

- MIN(price) for lowest price point in each category
- MAX(price) for highest price point in each category
- AVG(price) for average pricing, rounded to 2 decimal places

4. Ranking by Value:

- ORDER BY avg_price DESC to identify highest-value categories first
- This helps prioritize high-revenue category strategies

SQL Query:

```
select p.product_category_name,
       min(o.price) as min_price,
       max(o.price) as max_price,
       round(avg(o.price)::numeric,2) as avg_price
  from products p
  join order_items o
    on p.product_id=o.product_id
 group by product_category_name
 order by avg_price desc
```

Query Output:

1	product_category_name	min_price	max_price	avg_price
2	pcs	34.5	6729	1098.34
3	portateis_casa_forno_e_cafe	10.19	2899	624.29
4	eletrodomesticos_2	13.9	2350	476.12
5	agro_industria_e_comercio	12.99	2990	341.66
6	instrumentos_musicais	4.9	4399.87	281.62
7	eletroportateis	6.5	4799	280.78
8	portateis_cozinha_e_preparadores	17.42	1099	264.57
9	telefonia_fixa	6	1790	225.69
10	construcao_ferramentas_seguranc	8.9	3099.9	208.99
11	relogios_presentes	8.99	3999.9	200.91
12	climatizacao	10.9	1599	185.27
13	moveis_quarto	6.9	650	183.75
14	pc_gamer	129.99	239	171.77
15	cool_stuff	7	3109.99	167.36

Recommendations:

Category-Based Pricing Rationalization Strategy for Amazon India:

1. Price Floor Enforcement:

For categories with extreme price differences, establish a **minimum pricing threshold** for products with similar specifications. This prevents ultra-low-cost listings that may harm platform trust and create unfair competition.

2. Segmented Product Categorization:

Define clear price bands within each category (e.g., Budget, Mid-range,

Premium). This improves customer navigation, simplifies recommendation systems, and supports targeted marketing.

3. Increase Average Order Value (AOV):

By minimizing disruptive pricing gaps, Amazon can drive customers toward mid-tier and premium items, resulting in higher average order value and improved seller margins — without compromising on variety.

Question 3.3 -- Multi-Order Customers

Problem Statement:

Amazon India wants to identify customers who have placed multiple orders over time. The customer retention team needs to understand repeat purchase behavior to develop targeted loyalty programs, personalized recommendations, and retention strategies for customers who demonstrate ongoing engagement with the platform.

Desired Output:

customer_unique_id, total_orders

Approach:

1. Identifying Relevant Tables:

- **Primary Table:** customer (for unique customer identification)
- **Secondary Table:** orders (for order frequency tracking)

2. Linking Customer and Order Data:

- JOIN customer and orders tables on customer_id
- This connects unique customers with their order history

3. Counting Orders per Customer:

- COUNT(order_purchase_timestamp) to count actual orders placed
- GROUP BY customer_unique_id to analyze each customer individually

4. Filtering Multi-Order Customers:

- HAVING clause to include only customers with more than 1 order
- ORDER BY total_orders DESC to prioritize most active customers

SQL Query:

```
select c.customer_unique_id,
       count(o.order_purchase_timestamp) as total_orders
  from customer c
  join orders o
    on c.customer_id=o.customer_id
   group by c.customer_unique_id
 HAVING COUNT(o.order_id) > 1
  order by total_orders desc
```

Query Output:

customer_unique_id	total_orders
a91e80fbe80ddc07de66a5cf9270293c	16
a6168cd79131e64acef92e3c74d6cc43	16
363f980585bf04c1a88fdb986011c52e	16
cbd0350d4ccb9a772e8e768d4a4a5cbf	16
417b909c0962b2610f1cfb1c1478986	16
5f94af52aef02c968a2e0f01f430864e	16
1b6d29725255a77667a8c639eeb4ccc0	16
e4bbcc533fdf3917c56dea2c43bf2084	16
930c4390af58f67334447c3a1cf2ba36	16
5bf4ea2d98005b960eea0dbf652ef4e7	16
9159c04b88895d995741dd5b9b7a5f1d	16
4034aa08d48695a538b7030910aae5d5	16
c024307523462166b42112cfb6c8e900	16
0fdc0d21e1983e8af4d399e17671f76d	16

Recommendations:

Loyalty & Influencer Strategy for Multi-Order Customers:

1. Loyalty Recognition Program:

Identify customers with high order counts and reward them with exclusive hampers including personalized vouchers, product samples, and thank-you gifts to deepen trust and brand attachment.

2. Social Proof Marketing Loop:

Promote videos or testimonials from these loyal users receiving their rewards, and showcase them to occasional buyers to inspire more frequent purchases and create aspiration.

3. Behavioral Personalization & Brand Collaborations:

Analyze product categories purchased, return rates, and seasonal trends for each loyal customer. Use this data to personalize gifts and include sponsored

samples from premium brands — creating win-win marketing for both Amazon and its partners.

Question 3.4 -- Customer Categorization

Problem Statement:

Amazon India wants to categorize customers into different types based on their purchase history. The marketing team needs to segment customers as 'New' (1 order), 'Returning' (2-4 orders), or 'Loyal' (>4 orders) to develop targeted marketing campaigns, personalized offers, and retention strategies appropriate for each customer lifecycle stage.

Desired Output:

customer_unique_id, customer_type

Approach:

1. Defining Customer Categories:

- **New Customers:** Exactly 1 order (first-time buyers)
- **Returning Customers:** 2 to 4 orders (developing relationship)
- **Loyal Customers:** More than 4 orders (established relationship)

2. Counting Customer Orders:

- JOIN customer and orders tables on customer_id
- COUNT(order_purchase_timestamp) to determine order frequency

3. Applying Categorization Logic:

- CASE WHEN statements to assign customer_type based on order count
- GROUP BY customer_unique_id for individual customer analysis

4. Customer Lifecycle Segmentation:

- This segmentation enables targeted marketing strategies for each customer type

SQL Query:

```
select c.customer_unique_id,
case
    when count(o.order_purchase_timestamp)= 1 then 'New'
    when count(o.order_purchase_timestamp) between 2 and 4 then 'Returning'
    when count(o.order_purchase_timestamp) > 4 then 'Loyal'
end as customer_type
from customer c
join orders o
on c.customer_id=o.customer_id
group by c.customer_unique_id
```

Query Output:

1	customer_unique_id	customer_type
2	0000366f3b9a7992bf8c76cfdf3221e2	New
3	0000b849f77a49e4a4ce2b2a4ca5be3f	New
4	0000f46a3911fa3c0805444483337064	New
5	0000f6ccb0745a6a4b88665a16c9f078	New
6	0004aac84e0df4da2b147fca70cf8255	New
7	0004bd2a26a76fe21f786e4fdbd80607f	New
8	00050ab1314c0e55a6ca13cf7181fecf	New
9	00053a61a98854899e70ed204dd4bafe	New
10	0005e1862207bf6ccc02e4228effd9a0	New
11	0005ef4cd20d2893f0d9fdbd94d3c0d97	New
12	0006fdc98a402fce4eb0ee528f6a8d4	New
13	00090324bbad0e9342388303bb71ba0a	New
14	000949456b182f53c18b68d6bab79c1	New
15	000a5ad9c4601d2bbdd9ed765d5213b3	New

Recommendations:

Customer Lifecycle Marketing Strategy for Amazon India:

1. New Customers – Build Trust & Belonging:

Thank first-time buyers with personalized notes and offer "Early Bird" discounts on low-to-mid range products. Request feedback 3 days post-delivery to make them feel heard and valued.

2. Returning Customers – Create Habitual Engagement:

Launch subscription models for frequently purchased categories (like groceries), and increase targeted vouchers to drive deeper browsing and platform stickiness.

3. Loyal Customers – Drive Category Expansion:

Encourage purchases in new product categories by offering curated suggestions

and exclusive deals. Use data to personalize these nudges for maximum relevance and discoverability.

Question 3.5 -- Top Revenue Generating Categories

Problem Statement:

Amazon India wants to know which product categories generate the most revenue. The business development team needs to identify the top 5 revenue-generating categories to prioritize inventory investments, supplier relationships, marketing spend allocation, and category management resources for maximum ROI.

Desired Output:

product_category_name, total_revenue

Approach:

1. Identifying Revenue Sources:

- **Primary Table:** products (for category classification)
- **Secondary Table:** order_items (for revenue data)

2. Combining Category and Revenue Data:

- JOIN products and order_items tables on product_id
- This links each transaction to its product category

3. Calculating Total Revenue:

- SUM(price) to calculate total revenue per category
- ROUND() to 2 decimal places for clean financial reporting

4. Identifying Top Performers:

- ORDER BY total_revenue DESC for highest-revenue categories first
- LIMIT 5 to focus on top-performing categories

SQL Query:

```
select p.product_category_name
, round(sum(price)::numeric,2) as total_revenue
from products p
join order_items o
on p.product_id=o.product_id
group by p.product_category_name
order by total_revenue desc
limit 5
```

Query Output:

1	product_category_name	total_revenue
2	beleza_saude	1257865.34
3	relogios_presentes	1203060.32
4	cama_mesa_banho	1032268.59
5	esporte_lazer	985881.1
6	informatica_acessorios	910605.07

Recommendations:

Strategic Category Development for Revenue Maximization:

1. Seller Relationship Strengthening:

Prioritize engagement with top sellers in high-revenue categories through better margins, account management, and exclusive visibility benefits to retain and scale their presence.

2. Amazon's Choice Labeling Strategy:

Promote high-performing, highly rated products under the “Amazon’s Choice” label to boost trust and conversion. Charge sellers a small platform fee in exchange for increased visibility and credibility.



Summary of Analysis II:

- **Q3.1:** Payment preferences across spending segments for targeted financial strategies
- **Q3.2:** Comprehensive category pricing analysis for competitive positioning
- **Q3.3:** Multi-order customer identification for retention programs

- **Q3.4:** Customer lifecycle segmentation for personalized marketing
- **Q3.5:** Revenue-focused category prioritization for business growth

Key Business Value: This section provides customer behavior insights, pricing intelligence, and revenue optimization data crucial for Amazon India's customer acquisition and retention strategies.

4. Analysis -- III

Question 4.1 -- Seasonal Sales Comparison

Problem Statement:

Amazon India wants to compare sales performance across different seasons to optimize seasonal marketing campaigns and inventory management. The operations team needs to understand which seasons drive the highest sales volumes to allocate resources effectively and plan seasonal promotions, particularly important for the Indian market with its distinct festival seasons.

Desired Output: season, total_sales, avg_monthly_sales

Approach:

1. Defining Seasonal Categories:

- **Spring:** March, April, May
- **Summer:** June, July, August
- **Autumn:** September, October, November
- **Winter:** December, January, February

2. Extracting Seasonal Data:

- Use EXTRACT(MONTH FROM order_purchase_timestamp) to get month numbers
- Apply CASE WHEN statements to categorize months into seasons
- JOIN orders and order_items tables for sales data

3. Calculating Seasonal Metrics:

- SUM(price) for total seasonal sales
- AVG(price) for average monthly sales within each season
- ROUND() for clean financial reporting

4. Performance Comparison:

- o ORDER BY total_sales DESC to identify top-performing seasons

SQL Query:

```
select
    season,
    round(sum(total_price)::NUMERIC, 2) AS total_sales
from (
    select
        case
            WHEN EXTRACT(MONTH FROM o.order_purchase_timestamp) IN (3, 4, 5) THEN 'Spring'
            WHEN EXTRACT(MONTH FROM o.order_purchase_timestamp) IN (6, 7, 8) THEN 'Summer'
            WHEN EXTRACT(MONTH FROM o.order_purchase_timestamp) IN (9, 10, 11) THEN 'Autumn'
            ELSE 'Winter'
        end as season,
        oi.price as total_price
    from orders o
    join order_items oi
    on o.order_id = oi.order_id
) as seasonal_sales
group by season
order by total_sales desc;
```

Query Output:

1	season	total_sales
2	Spring	4216721.54
3	Summer	4120359.62
4	Winter	2905750.03
5	Autumn	2348812.51

Recommendations:

Seasonal Category-Driven Sales Strategy for Amazon India:

1. Seasonal Deep-Dive Campaigns:

Align promotional focus with seasonal needs — e.g., school/college reopening essentials in summer, winter wear in Q4, monsoon essentials in mid-year.

2. Early Campaign Launches:

Begin promotions 2–3 weeks ahead of seasonal peaks to capture early demand. Bundle items (e.g., electronics with accessories) to boost AOV.

3. Cross-Seasonal Voucher Strategy:

Offer winter-specific vouchers to summer buyers to ensure off-season engagement and platform retention.

Question 4.2 -- Above-Average Sales Volume Products

Problem Statement:

Amazon India wants to identify products that consistently perform above the average sales volume. The inventory management team needs to recognize high-performing products to ensure adequate stock levels, negotiate better supplier terms, and prioritize these products in marketing campaigns to maximize revenue potential.

Desired Output: product_id, product_category_name, total_quantity_sold

Approach:

1. Calculating Overall Average Sales:

- Use subquery to calculate AVG(quantity) across all products
- This provides the benchmark for comparison

2. Identifying High-Performing Products:

- SUM(quantity) grouped by product_id to get total sales per product
- HAVING clause to filter products above average sales volume

3. Adding Product Context:

- JOIN with products table to include category information
- This provides business context for high-performing products

4. Ranking by Performance:

- ORDER BY total_quantity_sold DESC to prioritize top performers

SQL Query:

```
select
    product_id,
    count(order_id) as total_quantity_sold
from order_items
group by product_id
having count(order_id) > (
    select avg(product_order_count)
    from (select count(order_id) AS product_order_count
          from order_items
          group by product_id
        )
)
order by total_quantity_sold desc
```

Query Output:

1	product_id	total_quantity_sold
2	aca2eb7d00ea1a7b8ebd4e68314663af	527
3	99a4788cb24856965c36a24e339b6058	488
4	422879e10f46682990de24d770e7f83d	484
5	389d119b48cf3043d311335e499d9c6b	392
6	368c6c730842d78016ad823897a372db	388
7	53759a2ecddad2bb87a079a1f1519f73	373
8	d1c427060a0f73f6b889a5c7c61f2ac4	343
9	53b36df67ebb7c41585e8d54d6772e08	323
10	154e7e31ebfa092203795c972e5804a6	281
11	3dd2a17168ec895c781a9191c1e95ad7	274
12	2b4609f8948be18874494203496bc318	260
13	7c1bd920dbdf22470b68bde975dd3ccf	231
14	a62e25e09e05e6faf31d90c6ec1aa3d1	226
15	5a848e4ab52fd5445cdc07aab1c40e48	197

Recommendations:

High-Volume Product Promotion & Bundling Strategy:

1. Preferred Product Promotion:

Feature high-sale items in “Liked by Others” sections to increase trust-based conversion.

2. Seller Incentivization & Price Harmonization:

Reward top-selling product sellers with better margins while controlling price variability to push buyers toward the recommended listings.

3. Smart Bundling Strategy:

Bundle accessories or add-ons (e.g., phone covers with mobiles) based on customer behavior and offer them together with a slight discount. Take a convenience cut from sellers on bundled transactions.

Question 4.3 -- Monthly Revenue Trends for 2018

Problem Statement:

Amazon India wants to analyze monthly revenue trends for 2018 to understand business growth patterns and identify peak performance periods. The finance team needs detailed month-by-month revenue analysis to inform budgeting, forecasting, and strategic planning for similar growth trajectories in the Indian market.

Desired Output: month, revenue_2018

Approach:

1. Filtering 2018 Data:

- WHERE clause with EXTRACT(YEAR FROM order_purchase_timestamp) = 2018
- Focus analysis on specific year for trend analysis

2. Monthly Aggregation:

- Extract month names using TO_CHAR(timestamp, 'Month')
- SUM(price) to calculate total monthly revenue
- GROUP BY month for monthly breakdown

3. Chronological Ordering:

- ORDER BY month to show trends chronologically
- This reveals seasonal patterns and growth trajectories

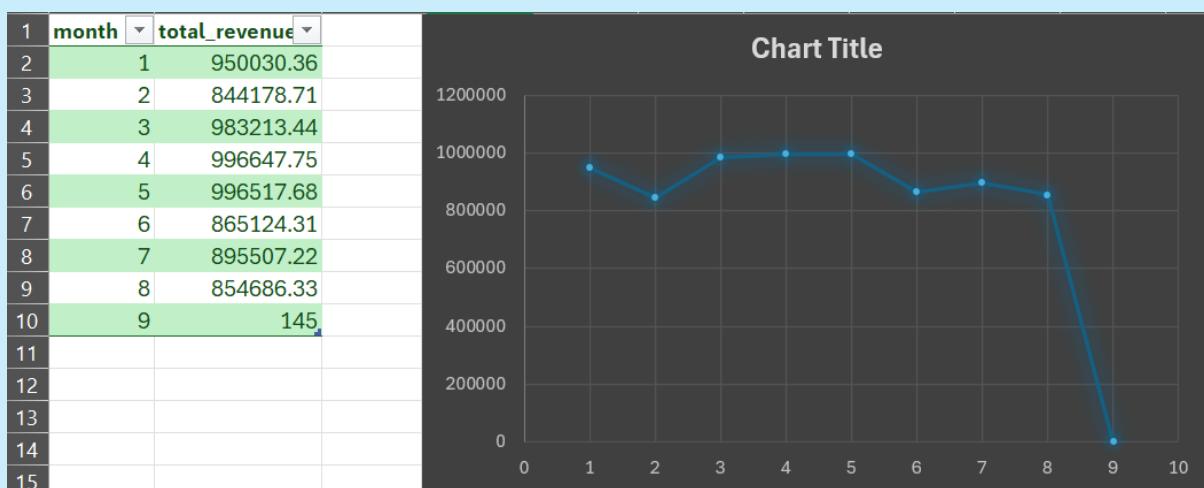
4. Revenue Standardization:

- ROUND() function for clean financial reporting

SQL Query:

```
select extract(month from o.order_purchase_timestamp) as month
, round(sum(o1.price)::numeric,2) as total_revenue
from orders o
join order_items o1
on o.order_id=o1.order_id
where extract(year from o.order_purchase_timestamp) = 2018
group by month
order by month asc
```

Query Output:



Recommendations:

Seasonal Demand Balancing Strategy for Amazon India:

1. Winter-Ready Product Onboarding:

Onboard sellers offering products relevant to slower months (e.g., winter wear, skincare, heaters).

2. Seller Enablement for Seasonality:

Train sellers to tag products as "winter-ready" and give seasonal discounts during off-peak months.

3. Voucher Rollover Campaign:

Give customers vouchers during summer that are only redeemable in winter, ensuring cyclical engagement.

Question 4.4 -- Customer Segmentation Using CTE

Problem Statement:

Amazon India wants to create advanced customer segmentation using Common Table Expressions (CTE) to categorize customers based on their total order frequency. The marketing team needs sophisticated customer segments to develop targeted campaigns, personalized recommendations, and tiered loyalty programs that match customer value and engagement levels.

Desired Output: customer_type,count

Approach:

1. Creating Customer Metrics CTE:

- Calculate total_orders for each customer
- Use CTE to organize complex calculations cleanly

2. Defining Segmentation Logic:

- **Loyal:** Order frequency >5
- **Regular:** Order frequency between 3 and 5
- **Ocassional:** Order frequency between 1 and 2

3. Multi-dimensional Segmentation:

- CASE WHEN statements with multiple conditions

- Considers both spending amount and purchase frequency

4. Business Intelligence:

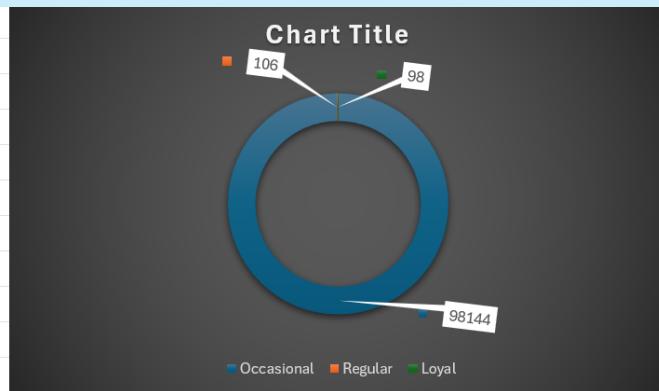
- Enables targeted marketing strategies for each segment

SQL Query:

```
with customerCTE as (
select
case when count (order_id) between 1 and 2 then 'Occasional'
      when count (order_id) between 3 and 5 then 'Regular'
      when count (order_id) >5 then 'Loyal'
end as customer_type
from orders
group by customer_id
)
select customer_type, count(*) as count
from customerCTE
group by customer_type
```

Query Output:

	customer_type	count
1	Occasional	98144
2	Regular	106
3	Loyal	98
4		
5		
6		
7		
8		
9		
10		
11		



Recommendations:

Community-Driven Loyalty Acceleration Strategy:

1. Amazon Loyalty Club:

Organize an annual offline/virtual event for top loyal and regular customers. Provide product previews, direct team interaction, and loyalty gifts.

2. Guest Invitations for Occasional Buyers:

Invite high-value occasional buyers as “guests,” inspiring aspirational loyalty.

3. Influencer & Brand Partnership Integration:

Include product samples from premium brands in gift bags and promote the event via influencers to maximize buzz and visibility.

Question 4.5 -- Top 20 High-Value Customers

Problem Statement:

Amazon India wants to identify its top 20 highest-value customers to develop premium customer service programs, exclusive offers, and VIP experiences. The customer success team needs to recognize these valuable customers to ensure exceptional service delivery and maximize customer lifetime value through personalized attention.

Desired Output: customer_unique_id, total_revenue, total_orders, avg_order_value

Approach:

1. Comprehensive Customer Metrics:

- Calculate total_revenue per customer from all orders
- Count total_orders to understand purchase frequency
- Calculate avg_order_value for spending pattern analysis

2. Multi-table Data Integration:

- JOIN customer, orders, and order_items tables
- GROUP BY customer_unique_id for individual customer analysis

3. Value-based Ranking:

- ORDER BY total_revenue DESC to identify highest spenders
- LIMIT 20 to focus on top-tier customers

4. Strategic Customer Intelligence:

- Avg_order_value helps understand customer purchasing behavior
- Total_orders indicates customer loyalty and engagement

SQL Query:

```
with rankedCustomer as (
  select o.customer_id,
    round(avg(o1.price)::numeric,2) as avg_order_value,
    dense_rank() over(order by avg(o1.price) desc) as customer_rank
  from orders o
  join order_items o1
  on o.order_id=o1.order_id
  group by o.customer_id
)
select * from rankedCustomer
where customer_rank<=20
```

Query Output:

1	customer_id	avg_order_value	customer_rank
2	c6e2731c5b391845f6800c97	6735	1
3	f48d464a0baaea338cb25f8	6729	2
4	3fd6777bbce08a352fddd04	6499	3
5	df55c14d1476a9a3467f1312	4799	4
6	24bbf5fd2f2e1b359ee7de94	4690	5
7	3d979689f636322c62418b63	4590	6
8	1afc82cd60e303ef09b4ef98	4399.87	7
9	35a413c7ca3c69756cb7586	4099.99	8
10	e9b0d0eb3015ef1c9ce6cf5b	4059	9
11	c6695e3b1e48680db36b487	3999.9	10
12	926b6a6fb8b6081e00b335e	3999	11
13	3be2c536886b2ea4668eced	3980	12
14	31e83c01fce824d0ff786fc4	3930	13
15	eb7a157e8da9c488cd4ddc	3899	14
16	19b32919fa1198aefc0773ee	3700	15
17	66657bf1753d82d0a76f2c47	3699.99	16
18	addc91fdf9c2b3045497b571	3690	17
19	39d6658037b1b5a07d0a24c	3549	18
20	e7c905bf4bb13543e8df947e	3399.99	19
21	46bb3c0b1a65c8399d0363c	3124	20

Recommendations:

Premium Loyalty Experience for High-Value Customers:

1. Exclusive Gratitude Package:

Send thank-you notes and ₹10,000 vouchers for future use, reinforcing appreciation and retention.

2. Priority Access & Extra Discounts:

Provide early access to major sales and 10% off high-margin products to reward loyalty.

3. Festive Gift Hamper:

Include branded gifts worth ₹500–₹1000 during Diwali, New Year, etc., to enhance emotional connection and encourage continued purchases.

Question 4.6 -- Monthly Cumulative Sales Using Recursive CTE

Problem Statement:

Amazon India wants to track monthly cumulative sales growth using recursive Common Table Expressions (CTE) to understand business momentum and growth acceleration. The executive team needs to visualize how sales build up over time to assess business trajectory, identify inflection points, and make strategic decisions about expansion timing.

Desired Output: month_year, monthly_sales, cumulative_sales

Approach:

1. Creating Base Monthly Sales:

- Extract year-month combinations for chronological ordering
- Calculate monthly sales totals as foundation for recursion

2. Implementing Recursive CTE:

- Base case: First month with its sales total
- Recursive case: Each subsequent month adds to previous cumulative total
- ORDER BY month_year for proper chronological sequence

3. Cumulative Growth Tracking:

- Running total calculation shows business growth momentum
- Identifies periods of acceleration or deceleration

4. Executive Dashboard Insights:

- Visual representation of business growth trajectory
- Enables strategic decision-making based on growth patterns

SQL Query:

```

WITH RECURSIVE monthly_sales AS (
  SELECT oi.product_id,
    DATE_TRUNC('month', o.order_purchase_timestamp)::DATE AS sale_month,
    SUM(oi.price) AS monthly_sales
  FROM order_items oi JOIN orders o ON oi.order_id = o.order_id
  WHERE DATE_PART('year', o.order_purchase_timestamp) = 2018
  GROUP BY oi.product_id, sale_month
),cte_base AS (
  SELECT ms.product_id,ms.sale_month,ms.monthly_sales,ms.monthly_sales AS total_sales
  FROM monthly_sales ms
  WHERE ms.sale_month = (SELECT MIN(s.sale_month)
    FROM monthly_sales s
    WHERE s.product_id = ms.product_id)UNION ALL
  SELECT rs.product_id,ms.sale_month,ms.monthly_sales,
    rs.total_sales + ms.monthly_sales AS total_sales
  FROM cte_base rs JOIN monthly_sales ms ON rs.product_id = ms.product_id
  AND ms.sale_month = rs.sale_month + INTERVAL '1 month')
SELECT
  product_id,
  sale_month,
  ROUND(total_sales::NUMERIC, 2) AS total_sales
FROM cte_base
ORDER BY product_id, sale_month

```

Query Output:

1	product_id	sale_month	total_sales
2	00066f42aeeb9f3007548bt	01-05-2018	101.65
3	000b8f95fcb9e0096488278	01-08-2018	117.8
4	000d9be29b5207b54e86aa	01-04-2018	199
5	001b237c0e9bb435f2e540	01-08-2018	78.9
6	002159fe700ed3521f46fcf	01-02-2018	199.7
7	002159fe700ed3521f46fcf	01-03-2018	599.1
8	002159fe700ed3521f46fcf	01-04-2018	798.8
9	00250175f79f584c14ab5ce	01-02-2018	479.94
10	002552c0663708129c0019	01-07-2018	108
11	002959d7a0b0990fe2d699	01-01-2018	129.9
12	002c6dab60557c48cf6c2	01-03-2018	79.9
13	002d4ea7c04739c130bb74	01-04-2018	43
14	002ec297b1b00fb9dde7ee	01-04-2018	16
15	0030e635639c898b323826	01-02-2018	153
16	003128f981470c3e5a2e74	01-07-2018	99.98

Recommendations:

Momentum-Based Sales Acceleration Strategy:

1. Sustain Growth of High-Selling Items:

Promote products with cumulative momentum into subsequent months via banners, emails, and homepage placement.

2. Seller Performance Incentives:

Offer sellers of growing products visibility perks and bonuses to ensure inventory readiness.

3. Sales Dip Recovery Plans:

During low-growth months, offer pay-later schemes, deep discounts, and exclusive bundles to maintain sales momentum.

Question 4.7 -- Month-over-Month Growth Rate Analysis

Problem Statement:

Amazon India wants to calculate month-over-month growth rates to measure business acceleration and identify trends in sales performance. The analytics team needs to understand growth velocity, seasonal impacts, and momentum changes to optimize strategic planning, resource allocation, and performance forecasting for sustained growth.

Desired Output: month_year, current_month_sales, previous_month_sales, growth_rate_percentage

Approach:

1. Monthly Sales Calculation:

- Extract month-year combinations for proper time series analysis
- $\text{SUM}(\text{price})$ for total monthly sales figures

2. Implementing LAG Window Function:

- $\text{LAG}(\text{monthly_sales}) \text{ OVER (ORDER BY month_year)}$ to get previous month's sales
- This enables month-over-month comparison

3. Growth Rate Formula:

- $((\text{current_month} - \text{previous_month}) / \text{previous_month}) * 100$
- ROUND to 2 decimal places for percentage precision

4. Trend Analysis:

- Positive growth rates indicate expansion
- Negative rates highlight contraction periods
- Enables strategic intervention identification

SQL Query:

```
WITH monthly_sales AS (
    SELECT p.payment_type,
        DATE_TRUNC('month', o.order_purchase_timestamp)::DATE AS sale_month,
        SUM(p.payment_value) AS monthly_total
    FROM payments p JOIN orders o ON p.order_id = o.order_id
    WHERE DATE_PART('year', o.order_purchase_timestamp) = 2018
    GROUP BY p.payment_type, sale_month
),monthly_with_change AS (
    SELECT payment_type,sale_month,
        ROUND(monthly_total::NUMERIC, 2) AS monthly_total,
        ROUND(((monthly_total - LAG(monthly_total) OVER
            (PARTITION BY payment_type ORDER BY sale_month )) / NULLIF(LAG(monthly_total) OVER
            (PARTITION BY payment_type ORDER BY sale_month), 0)::NUMERIC, 4) * 100 AS monthly_change
    FROM monthly_sales
)
SELECT
    payment_type,
    sale_month,
    monthly_total,
    ROUND(monthly_change, 2) AS monthly_change
FROM monthly_with_change
ORDER BY payment_type, sale_month;
```

Query Output:

1	payment_type	sale_month	monthly_total	monthly_change
2	boleto	01-01-2018	204844.66	NULL
3	boleto	01-02-2018	183112.72	-10.61
4	boleto	01-03-2018	191538.02	4.6
5	boleto	01-04-2018	193547.09	1.05
6	boleto	01-05-2018	195378.93	0.95
7	boleto	01-06-2018	153350.28	-21.51
8	boleto	01-07-2018	198041.24	29.14
9	boleto	01-08-2018	143805.9	-27.39
10	credit_card	01-01-2018	868880.38	NULL
11	credit_card	01-02-2018	778803	-10.37
12	credit_card	01-03-2018	933770.1	19.9
13	credit_card	01-04-2018	934306	0.06
14	credit_card	01-05-2018	927556.35	-0.72
15	credit_card	01-06-2018	811508.56	-12.51
16	credit_card	01-07-2018	803674.49	-0.97

Recommendations:

Month-over-Month Growth Optimization Strategy:

1. Early Detection of Sales Slumps:

Use MoM dips as signals for flash sales or retention campaigns in underperforming segments.

2. Performance-Based Ad Spend Allocation:

Increase ad budget on consistently growing categories; cut back on stagnating ones while testing alternatives.

3. Momentum Milestone Marketing:

Create campaigns celebrating growth milestones (e.g., "Fastest-Growing Category This Month") to build buyer confidence and urgency.

End of Analysis III

Summary of Analysis III:

- **Q4.1:** Seasonal sales comparison for strategic marketing and inventory planning
- **Q4.2:** Above-average product identification for inventory optimization
- **Q4.3:** Monthly revenue trends analysis for financial forecasting
- **Q4.4:** Advanced customer segmentation using CTEs for targeted marketing
- **Q4.5:** Top 20 high-value customer identification for VIP programs
- **Q4.6:** Cumulative sales tracking using recursive CTEs for growth monitoring
- **Q4.7:** Month-over-month growth rate analysis for performance measurement

Key Business Value: This section provides advanced analytical insights into seasonal patterns, customer value segmentation, growth trends, and performance metrics essential for Amazon India's strategic planning and competitive positioning.

Overall Project Summary

This comprehensive SQL analysis of Amazon Brazil's data provides Amazon India with actionable insights across three critical business dimensions:

Analysis I - Foundation Insights:

- Payment method optimization and financial standardization
- Product positioning and pricing strategies
- Seasonal trend identification and data quality management

Analysis II - Customer Intelligence:

- Customer behavior segmentation and retention strategies
- Category performance analysis and revenue optimization
- Payment preference patterns across customer segments

Analysis III - Advanced Analytics:

- Sophisticated customer value segmentation
- Growth trajectory analysis and seasonal planning
- Performance measurement and strategic forecasting

Strategic Impact for Amazon India: The insights derived from this analysis enable Amazon India to replicate Amazon Brazil's success by understanding customer behaviors, optimizing payment strategies, identifying high-value products and customers, and making data-driven decisions for market entry and expansion in the Indian e-commerce landscape.